Proceedings of CombLog'04

Workshop on Combination of Logics: Theory and Applications

Lisbon, July 2004

Walter Carnielli, F. Miguel Dionísio, Paulo Mateus (eds.)

CLC

Center for Logic and Computation

Department of Mathematics

Instituto Superior Técnico

Ficha Técnica

Editor: Departamento de Matemática - Instituto Superior Técnico

Compiladores: Walter A. Carnielli
F. Miguel Dionísio
Paulo Mateus

Título: Proceedings of CombLog'04
Workshop on Combination of Logics: Theory and Applications

ISBN: 972-99289-0-8

Depósito Legal: 1

Design: Eduardo R. Castro e Costa

Impressão/Acabamentos: Grafitese

Tiragem: 200

Lisboa, Julho de 2004

Contents

Forward

Combining logics and logicians

It is not any new desideratum of philosophers and logicians to build a paradise where several different logics could interact and cooperate, instead of clashing. The Czech philosopher and mathematician Bernard Bolzano had thought already in 1837, about mixing and combining different notions of consequence in his monumental *Wissenschaftslehre*. Bolzano is considered by some scholars to be philosophically akin in thought and fundamental conceptions to Gottfried Wilhelm Leibniz, to whom we owe the intentions of proposing a universal language, his *characteristica universalis*.

In Leibniz's conception, such a language when fully developed would be the basis for the greatest instrument of reason, the *calculus ratiocinator*: when there were disputes among persons, there would be place for an invitation for logical agreement: "Let us calculate, without further ado, and see who is right!" So, if it is not new that thinking should not be seen just as an one-sided monolith, the universe of applications that the research on combining logics and the development of mixed logics promises us should come as no surprise. Among the vast possibilities of procedures for combining logics, fibring occupies a central place. Its theoretical significance is due to the fact that it is more liable to results of completeness preservation than other combination procedures, in the sense that in many cases the completeness of a fibred logic can be obtained by the completeness of its fibring components. The fact that results of this sort have been obtained in the scope of higher-order, modal, relevance and non-truth-functional logics, and that refinements of the notion of fibring such as modulated fibring, have proved to be apt tools to solve some collapsing problems within the combination of logics justify the interest in fibring, although this does not make it the unique research target. The ambit of this workshop is thus naturally devoted to integrating, comparing and fostering other forms of composing and decomposing logics, such as fusion, splicing, splitting, synchronization and temporalization. Other, more ambitious, transference results between the whole and the component logics, such as preservation of completeness, interpolation properties and decidability are also envisaged at the present Workshop. Applications of the amply generous ideal of combining rationalities include software specification, knowledge representation, architectures for intelligent computing and quantum computing, applications to security protocols and authentication, secure computation and zero-knowledge proof protocols, with its connections to formal ethics and game protocols. The Workshop on Combination of Logics: Theory and Applications (CombLog'04) was held in the premises of the Center for Logic and Computation, at the Department of Mathematics of IST, Technical University of Lisbon, Portugal, from July 28-30, 2004. The sixteen selected extended abstracts that compose this book of proceedings, the result of a qualified consortium of philosophers, mathematicians and computer scientists, range from concrete cases of combined systems with a clear-cut intention, to the exposition of new formal semantics and abstract categorical treatments of formal ontology and institutions. The combinations of styles and trends, rather a combination of logicians than a combination of logics, is reflected in this collection of papers, which offers a taste of what has been done, and opens up main directions for further research.

Acknowledgement of sponsorship and support

The organizing committee in charge of CombLog'04 was composed by Walter A. Carnielli, CLE, University of Campinas, Marcelo E. Coniglio, CLE, University of Campinas, Paulo Mateus, CLC, IST, Technical University of Lisbon, Till Mossakowski, University of Bremen and Amílcar Sernadas (Chair), CLC, IST, Technical University of Lisbon.

The workshop was sponsored by ASL: The Association for Symbolic Logic (USA), CLC: Center for Logic and Computation at IST (Lisbon, Portugal), CLE: Centre for Logic, Epistemology and the History of Science at UNICAMP (Campinas, Brazil), and FLIRTS Interest Group.

Financial support was granted from EU FEDER POCTI Program, FCT: Fundação para a Ciência e a Tecnologia, Fundação Calouste Gulbenkian, and BPI.

The workshop was organized within the context of FCT/FEDER Project POCTI/2001/MAT/37239 FIBLOG (`http://clc.math.ist.utl.pt/fiblog.html`).

An electronic version of the publication will be held at CLE e-Prints (ISSN 1519-9681) of CLE-UNICAMP at Campinas.

Lisbon, July of 2004

The Editors

Walter A. Carnielli
F. Miguel Dionísio
Paulo Mateus

Reactive Kripke Semantics and Arc Accessibility

Dov Gabbay

Department of Computer Science, King's College London, U. K.

Abstract

Ordinary Kripke models are not reactive. When we evaluate (test/measure) a formula A at a model $\mathbf{m}$, the model does not react, respond or change while we evaluate. The model is static and unchanged. This paper studies Kripke models which react to the evaluation process and change themselves during the process. This is reminiscent of game theoretic semantics where the two sides react to each other. However, reactive Kripke models do not go as far as that. The only additional device we add to Kripke semantics to make it reactive is to allow the accessibility relation to access itself. Thus the accessibility relation $\mathcal{R}$ of a reactive Kripke model contains not only pairs $(a, b) \in \mathcal{R}$ of possible worlds (b is accessible to a, i.e. there is an accessibility *arc* from a to b) but also pairs of the form $(t, (a, b)) \in \mathcal{R}$, the arc (a, b) is accessible to t.
This new kind of Kripke semantics allows us to characterise more axiomatic modal logics (with one modality $\Box$) by a class of reactive frames. There are logics which cannot be characterised by ordinary frames but which can be characterised by reactive frames.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Motivation and Background

Traditional modal logic uses possible world semantics with accessibility relation R. When we evaluate a formula such as $B = \Box^2 p \wedge \Box^3 q$ in a Kripke model $\mathbf{m} = (S, R, a, h)$ (S is the set of possible worlds, $a \in S, R \subseteq S^2$ and h is the assignment) the model $\mathbf{m}$ does not change in the course of evaluation of B. We say the model $\mathbf{m}$ is not *reactive*. It stays the same during the process of evaluation.

To make this point absolutely clear, consider the situation in Figure 1 below

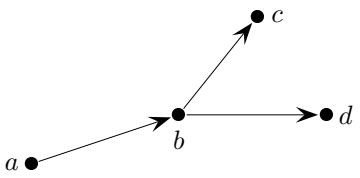


Figure 1

To evaluate $a \vDash \Box^3 q$, we have to check $b \vDash \Box^2 q$. We can also check another formula at b, say, $b \vDash \Box^2 p$. In either case the world accessible to b are c and d.

We *do not* say that since $b \vDash \Box^2 q$ started its evaluation at world a as $a \vDash \Box^3 q$ and continued to $b \vDash \Box^2 q$, then the accessible worlds to b are now different. In other words the model does not react to our starting the evaluation of $a \vDash \Box^3 q$ by changing the accessible worlds at b and therefore allowing us to see a different set of accessible worlds when we continue the evaluation of $b \vDash \Box^2 q$.

The evaluation of $\Box$ at b *does not* depend on how we "got" to b.

This paper addresses the case where the semantics does change (or react) under us as we evaluate a formula. This idea makes the evaluation of a wff at a world t dependent on the route leading to t. Thus we get a new kind of semantics, the reactive semantics. We shall explain exactly what we mean after we consider some case studies.

1.1 Airline example

We begin with a very simple and familiar example. Consider Figure 2

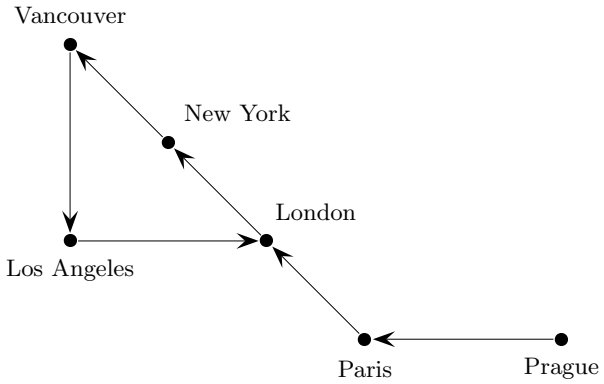


Figure 2

Figure 2 gives the possible flight routes for the aeroplanes of TUA (Trans Universal Airlines). It is well known that many features of a flight depend on the route. These include the cost of tickets, as well as the right to take passengers at an airport. The right to take passengers at an airport depends on the

flight route to that airport and on bilateral agreements between the airlines and governments. Thus, for example, flights to New York originating in London, may take on passengers in London to disembark in New York. However, a flight starting at Paris going to New York through London may not be allowed to pick up passengers in London to go onto New York. It is all a matter of agreements and landing rights. It is quite possible, however, that on the route Prague–Paris–London–New York, the airline is allowed to take passengers in London to disembark in New York. We can describe the above situation in Figure 3.

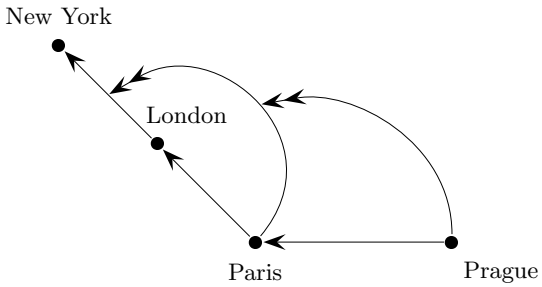


Figure 3

The double-headed arrow from Paris to the arc London→New York indicates a cancellation of the 'passenger' connection from London to New York. The double-headed arrow from Prague to the double arrow arc emanating from Paris indicates a cancellation of the cancellation.

Figure 3 looks like a typical reactive Kripke model, where we have arcs leading into arcs.

Let us see more examples of this.

1.2 Inheritance networks example

This example offers a different point of view of arc semantics, coming from the non-monotonic theory of inheritance networks. Consider Figure 4.

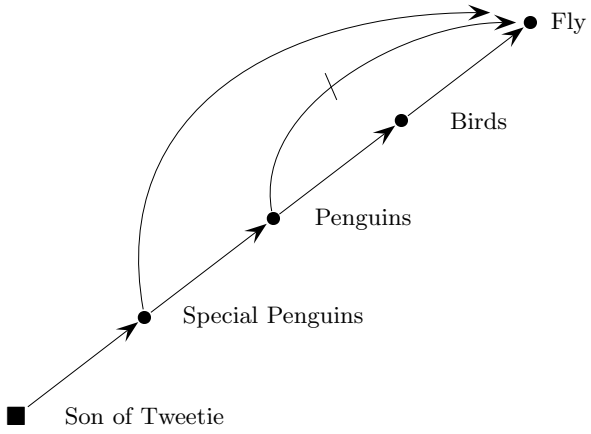


Figure 4

In Figure 4, the circular nodes are predicates, such as Fly, Birds, etc. The arrows indicate inheritance, so for example, we have $\forall x(\text{Bird}(x) \rightarrow \text{Fly}(x))$. The arrows with a bar indicate blockage, for example $\forall x(\text{Penguin}(x) \rightarrow \neg\text{Fly}(x))$. The square nodes indicate instantiation, so son of Tweetie is a special penguin.

Figure 4 is the kind of figure one finds in papers on inheritance networks. The figure indicates that Penguins are Birds, that Birds Fly but that Penguins do not Fly. However, special Penguins do Fly and

the son of Tweetie, a rare bird, is a special Penguin, and therefore does Fly. The arrow with the bar on it blocks the information from flowing from the Penguin node to the Fly node. The theory of inheritance networks spends a lot of effort on algorithms that allow us to choose between paths in the network so that we can come up with the desired intuitively correct answers. In the case of Figure 4 we want to get that the son of Tweetie does fly, since we have the most specific information about him. It is not important to us in this paper to take account of how inheritance theory deals with this example. We want to look at the example from our point of view, using our notation, as in Figure 5.

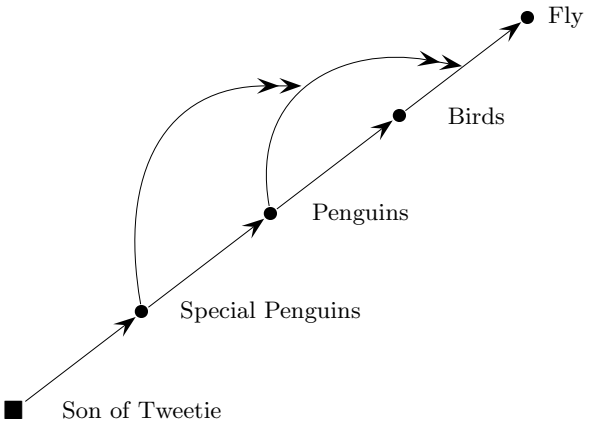


Figure 5

In Figure 5 the double headed arrow $\twoheadrightarrow$ emanating from Penguins attacks the arrow from Birds to Fly, and the double arrow emanating from Special penguins attacks the double arrow emanating from Penguins and attacking the arrow from Birds to Fly. This is not how inheritance theory would deal with this situation but we are not doing inheritance theory here. Our aim is to motivate our approach and what we need from the inheritance example is just the idea of the algorithmic flow of information during the dynamic evaluation process.

We have already put forward the reactive and dynamic idea of evaluation in earlier papers and lectures (see [3]). A typical example we give is to consider $t \vDash \Diamond A$. In modal logic this means that there is a possible world s such that we have $s \vDash A$ we take a more dynamic view of it.

We ask: where is s? How long does it take to get to it? and how much does it cost to get there? [1]

The reader should recall the way circumscription theory deals with the Tweetie example, see [4, section 4.1, especially page 324]. We write

- Birds $(x) \wedge \neg Ab_1(x) \rightarrow$ Fly(x)
- Penguins$(x) \rightarrow$ Birds(x)
- Penguins$(x) \wedge \neg Ab_2(x) \rightarrow Ab_1(x)$
- Special Penguins$(x) \rightarrow$ Penguins(x)
- Special Penguins (son of Tweetie)
- Special Penguins $(x) \rightarrow Ab_2(x)$.

"$Ab(x)$" stands for "x is abnormal". If the clause $C(x) \rightarrow B(x)$ represents the arc $C \rightarrow B$ then $C(x) \wedge \neg Ab(x) \rightarrow B(x)$ represents the situation in Figure 6

[1] It is our intention to explore whether our idea of double headed arrows cancelling other arrows can simplify inheritance theory algorithms.

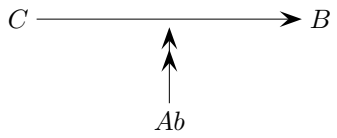


Figure 6

1.3 A technical example

It is now time to give a technical example. Consider Figure 7 below. This figure displays a past flow of time. The node t is the present moment and a single headed arrow from one node to another, say from s to t, means that t is in the immediate future of s. We use the modality $\Box$ to mean 'always in the immediate past'. Thus the accessibility relation R of figure 7 is as follows:

- $tRs, tRb, bRs, sRa, tRt, bRb, sRs$ and aRa.

The double-headed arrows cancel the accessibility relation.

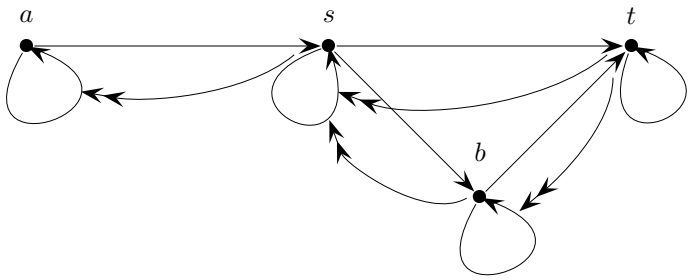


Figure 7

Let us calculate $t \vDash \Box^3 q$ in Figure 7.

Initial Position: Starting point is t and all arrows are active.
Step 1: Send double arrow signal from t to all destinations inverting the active/inactive status of all destination arrows. Then go to all accessible worlds (in this case s and b) and evaluate $\Box^2 q$ there. If the result is positive 1 and at all nodes, then send 'success' back to node t.
Step 2: Evaluate $\Box^2 q$ at nodes b and s.
Subcase 2.b. Evaluation at b: First we send a double arrow signal from b to all destinations reversing the activation status of these destinations. Thus the single arrow from s to s will be re-activated and we will evaluate $\Box q$ at s at the next step 3 (with s accessible to itself). b is not accessible to itself because its arrow has been deactivated by t at Step 1.
Subcase 2.s. Evaluation at s: First we send a double arrow signal to reverse the status arrow from a to a. Then we evaluate $\Box^2 q$ at s with s not accessible to itself, since the arrow from s to s was deactivated by t at step 1.

This can go on, but we shall not continue as we trust that the reader has got the idea by now.

Note that if we start at t and evaluate $B = \Box^3 q \wedge \Box^2 q$, we will get that $\Box q$ must be evaluated at s in two ways. One with s accessible to itself (coming from b via $t \vDash \Box^3 q$) and once with s not accessible to itself (coming from t via $t \vDash \Box^2 q$).

1.4 Tax example

Having explained the technical side of our reactive (changing) semantics, let us give some real examples.

House prices in London have gone up a great deal. An average upper middle class family is liable to pay inheritance tax on part of the value of their house (if the house is valued over £500,000, for example, then there is tax liability on £250,000). Some parents solved the problem by giving the house as a gift to their children. If at least one of the parents remains alive for seven years after the transaction, then current rules say that there is no tax. Consider therefore the following scenario:

1. current date is April 2004
2. parents gave house as a gift to children in 1996
3. parents continued to live in house as guests of the children

(1)–(3) above imply that (4):

4. if parents both die in March 2004, then no tax is liable.

To continue the story, there were rumours that the tax people were going to change the rules in April 2004, declaring that if parents remain living in the house after it was given as a gift, then the gift does not count as such an there is tax liability. The rumours also said that this law is going to apply *retrospectively.*[2]

Thus we have that (5) holds:

5. If parents both die on March 2005, then tax is liable.

We assume that (4) still holds even after the new law as we cannot imagine that the UK tax inspector would be opening closed old files and demanding more tax.

The way to represent (4) and (5) is to use two dimensional logic. We write $t \vDash_s A$ to mean at time t A is true given the point of view proposed or held at time s.

Thus $2005 \vDash_{2003} \neg(5)$ holds, because from the 2003 laws point of view (before legislation) no tax is liable ((5) says tax is liable).But $2005 \vDash_{2004} (5)$ also holds, because according to 2004 legislation tax is liable.

So far we have no formal problem and no need for our new semantics, because we can write

- $t \vDash_t \Box A$ iff for all future $s, s \vDash_s A$.

In other words we evaluate sentences at time t according to the point of view held at the very same time t.

The problem arises when we want to formalise the following scenario. The parents die in 2003. The lawyer is dealing with the estate. We do not know when he is going to finish. When he submits the paperwork then the tax liability at 2003 is judged according to the time of submission. Now the second index s in $t \vDash_s A$ behaves like a reactive model as we are evaluating

$$2003 \vDash_{\text{time lawyer submits}} (4).$$

2 Connection with hyper-modalities

In [2], we introduced hyper-modal logics. We showed that such modalities cannot always be characterised by a class of Kripke frames. However, there is hope that our new reactive semantics might provide frames for some of these modalities. This section and the next study the connection.

A hyper-modality $\Box$ is a modality which changes its nature depending on where in a formula it appears. So for example, in the formula $B = \Box^3 q \wedge \Box^2 q$, the inner modalities may not have the same meaning as the outer ones.

[2]Some countries, like Austria, for example, would *never* legislate retrospectively. They regard this as a cultural taboo.

To illustrate this point consider the arrangement of Figure 8

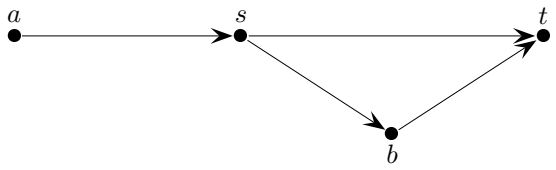


Figure 8

t is *now*, s is in the past of t and so are a and b. We consider two past operators

- $H_{\mathbf{K}}A$ saying A was true at all the immediately past moments of time

and

- $H_{\mathbf{T}}A$ saying $A \wedge H_{\mathbf{K}}A$

We let $\Box$ alternate between $H_{\mathbf{K}}$ and $H_{\mathbf{T}}$, starting with $H_{\mathbf{T}}$.

Thus $B = \Box^3 q \wedge \Box^2 q$ reads

$$H_{\mathbf{T}}H_{\mathbf{K}}H_{\mathbf{T}}q \wedge H_{\mathbf{T}}H_{\mathbf{K}}q.$$

Let us evaluate B at t. We write $t \vDash_{\mathbf{K}} A$ when we are evaluating A at a $\mathbf{K}$ mode and $t \vDash_{\mathbf{T}} A$ when we are evaluating A at the $\mathbf{T}$ mode.

Writing the above in full we have:

(*1) $t \vDash_{\mathbf{K}} \Box A$ iff for all immediately past points s we have $s \vDash_{\mathbf{T}} A$.

(*2) $t \vDash_{\mathbf{T}} \Box A$ iff $t \vDash_{\mathbf{K}} A$ and for all immediately past points s we have $s \vDash_{\mathbf{K}} A$.

Let us now evaluate $t \vDash (\Box^3 q \wedge \Box^2 q)$ in the flow of Figure 8. We have (remember we start with $\vDash_{\mathbf{T}}$):

- $t \vDash_{\mathbf{T}} \Box^3 q$ iff first $s \vDash_{\mathbf{K}} \Box^2 q$ and second $b \vDash_{\mathbf{K}} \Box^2 q$ and third $t \vDash_{\mathbf{K}} \Box^2 q$ iff first $a \vDash_{\mathbf{T}} \Box q$ and second $s \vDash_{\mathbf{T}} \Box q$ and third $b \vDash_{\mathbf{T}} \Box q$ and $s \vDash_{\mathbf{T}} \Box q$.
- $t \vDash_{\mathbf{T}} \Box^2 q$ iff first $s \vdash_{\mathbf{K}} \Box q$, and second $b \vDash_{\mathbf{K}} \Box q$ and third $t \vDash_{\mathbf{K}} \Box q$.

Since both $t \vDash_{\mathbf{T}} \Box^3 q$ and $t \vDash_{\mathbf{T}} \Box^2 q$ must hold, we see that we need to evaluate both $s \vDash_{\mathbf{T}} \Box q$ and $s \vDash_{\mathbf{K}} \Box q$.

This means that we cannot make the evaluation of $\Box q$ at s be dependent solely on the properties of the set $\{y \mid yRs\}$. We do need the dependency on the $\mathbf{T}$ and $\mathbf{K}$ modes.

Indeed, we axiomatise in [2] a modal logic with only the connective $\Box$ with the property that this logic can be characterised by the two $\mathbf{K}$ and $\mathbf{T}$ modes but it cannot be characterised by any class of frames. This shows that mode shifting is a genuinely stronger instrument of defining modal logics than imposing conditions on the accessibility relation R. We will to show in this paper that this logic can be characterised by a class of reactive models.

So much for a short survey of the ideas of [2]. See Appendix A for formal definitions of hyper-modal logics. Let us now proceed to show the connection with the reactive semantics of this paper.

First observe that the mode described above change the meaning of $\Box$. The modes do not change the semantics. In other words, the geometry of Figure 8 remained fixed. The model has not changed during the course of evaluation of $t \vDash B$. We want to show that we can achieve the same effect by changing the semantics as we evaluate.

Figure 9 describes the same flow of time as that of Figure 8 with the addition of the property that time is reflexive. Now suppose we have two modes of evaluation $\vDash_{\mathbf{T}}$, where we evaluate $\Box$ in the reflexive mode (i.e. Figure 9), and $\vDash_{\mathbf{K}}$, where we evaluate in the irreflexive model (i.e. in Figure 8).

Let us spell it out clearly:

(*3) $t \vDash_{\mathbf{K}} \Box A$ iff first deactivate all reflexive arrows in the accessibility relation R and then ask for $s \vDash_{\mathbf{T}} A$ to hold at every s which is accessible to t.

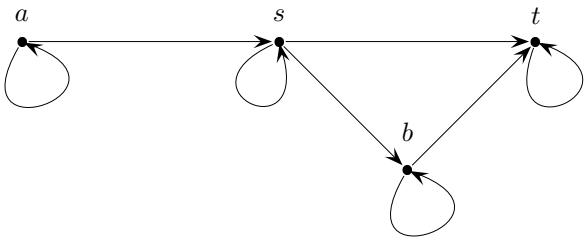


Figure 9

(*4) $t \vDash_{\mathbf{T}} \Box A$ iff first reactivate all cancelled reflexive arrows in R and then ask for $s \vDash_{\mathbf{K}} A$ to hold at every s accessible to t.

Clearly the evaluation of $t \vDash B$ will end up the same whether we view it as shifting the meaning of $\Box$ or shifting the underlying accessibility relation in the model.

Let us view the changing of the semantics as disconnecting or reconnecting arrows (accessibility) in the model.

This we have already done in the technical example of subsection 1.3. See Figure 7.

It is not difficult to see how a general arc model can be constructed in which $\Box$ alternates between a **T** and a **K** modality.

To do this properly, we need first a formal definition of the reactive semantics for modal logic.

3 Reactive Kripke models

We now give a definition of what we call reactive Kripke models, giving rise to what we call arc modal logics.

Definition 1. Let S be a set of possible worlds and $a \in S$ is the actual world. An arc-accessibility relation on S is defined as follows:

1. The set of all arcs $\mathcal{A}$ is defined by
   1.1. $S \subseteq \mathcal{A}$, these are 0 level arcs.
   1.2. If $\alpha \in \mathcal{A}$ is an n level arc and $s \in S$ then $\beta = (s \to \alpha)$ is an $n+1$ level arc.
2. A subset $\mathcal{R} \subseteq \mathcal{A}$ is an arc relation.
3. An arc-Kripke model has the form $(S, \mathcal{R}, \mathcal{R}^*, a, h)$, where $\mathcal{R}$ and $\mathcal{R}^*$ are an arc-accessibility relations and $\mathcal{R} \subseteq \mathcal{R}^*$ and h is an assignment giving to each $t \in S$ and each atom q a value $h(t,q) \in \{0,1\}$

Definition 2. Let $\mathbf{m} = (S, \mathcal{R}, \mathcal{R}^*, a, h)$ be a model.
Define $a \vDash A$, by structural induction

1. $a \vDash q$ if $h(a,q) = 1$, for q atomic.
2. $a \vDash A \wedge B$ iff $a \vDash A$ and $a \vDash B$.
3. $a \vDash \neg A$ iff $a \nvDash A$.
4. $a \vDash \Box A$ iff for all s in S such that $(a,s) \in \mathcal{R}$ we have that $s \vDash A$ in the model $\mathbf{m}_s = (S, \mathcal{R}_a, \mathcal{R}^*, s, h)$, where $\mathcal{R}_a$ is obtained from $\mathcal{R}$ as follows:

$$\mathcal{R}_a = \mathcal{R} - \{\alpha \mid (a \to \alpha) \in \mathcal{R} \wedge \alpha \in \mathcal{R}\} \cup \{\alpha \mid (a \to \alpha) \in \mathcal{R} \wedge \alpha \notin \mathcal{R} \wedge \alpha \in \mathcal{R}^*\}.$$

5. A model $\mathbf{m} = (S, \mathcal{R}, \mathcal{R}^*, a, h)$ is said to be of level $\leq n$, $n \geq 1$, if all its arcs in $\mathcal{R}^*$ are of level $\leq n$.

   For example, a model of level ≤ 2 can contain either arcs of the form $t \to s$ or of the form $r \to (t \to s)$, where $r, t, s, \in S$.

6. Let $\mathbf{K}^n_{\mathcal{A}}, n \geq 1$ be the set of wffs valid in the class of all reactive Kripke models of level $\leq n$. Note that $\mathbf{K}^1_{\mathcal{A}}$ is ordinary modal $\mathbf{K}$.

7. A modal logic $\mathbf{L}$ is said to be a reactive modal logic if for some class $\mathcal{K}$ of reactive models $\mathbf{L} = \{A \mid A$ is valid in all models of $\mathcal{K}\}$.

Example 3. Consider the two point model of Figure 10 with $S = \{a, b\}$:

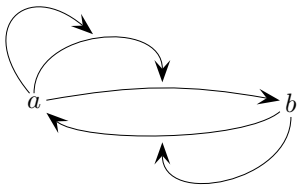


Figure 10

Let $\mathcal{R} = \mathcal{R}^* = \{a \to b, b \to a, a \to (a \to b), a \to (a \to (a \to b)), b \to (b \to a), b \to (b \to (b \to a))\}$. Consider a model

$$\mathbf{m} = (S, \mathcal{R}, \mathcal{R}^*, a, h).$$

Here $\mathcal{R}_a = \{b \to a, b \to (b \to a), a \to (a \to (a \to b))\}$.

$$\mathcal{R}_{a,b} = \{b \to (b \to a), a \to (a \to (a \to b))\}$$
$$\mathcal{R}_{a,b,a} = \{b \to (b \to a), a \to (a \to (a \to b)), a \to (a \to b)\}.$$

Note that $a \to b$ is not restored until $\mathcal{R}_{a,b,a,b,a}$.

Definition 4. Let (S, R) be a set S with a binary relation $R \subseteq S^2$. Let τ be a Horn clause theory in the language with R containing universal clauses of the form (universal closure) $(\bigwedge_{i=1}^{n} x_i R y_i \to xRy)$. Let $R^\tau \supseteq R$ be defined as the smallest extension of R such that $(S, R^\tau) \vDash \tau$. R^τ can be constructed by induction as the closure of R under all instances of τ as follows:

1. Let $R_0 = R$
2. Let $R_{n+1} = \{(a, b) \mid$ for some clause L in τ of the form (universal closure)$(\bigwedge x_i R y_i \to xRy)$ and a substitution θ to the variables of L such that $\theta(y_i), \theta(x_i) \in S, \theta(x) = a, \theta(y) = b$, we have that $\bigwedge \theta(x_i) R_n \theta(y_i)$ holds.
3. Let $R_\tau = \bigcup_n R_n$.

Definition 5.

1. Let $\mathbb{N}$ be the set of natural numbers $\{0, 1, 2, \ldots\}$. Let $\mathbb{N}^*$ be the set of all finite sequences of natural numbers including the empty sequence $\varnothing$. Define $\alpha < \beta$, for $\alpha, \beta \in \mathbb{N}^*$ by

   $$\alpha < \beta = (\text{ definition}) \text{ for some } m \in \mathbb{N}, \beta = \alpha * (m),$$

   where $*$ is concatenation.

   Let $<^*$ be the transitive closure of $<$.

2. A tree T is a nonempty subset of $\mathbb{N}^*$ such that if $\beta \in T$ and $\alpha < \beta$ then $\alpha \in T$.

3. Let τ be a Horn theory on $<$.

   Let $(T,<)$ be a tree and let $(T,<^\tau)$ be its τ-closure. Define $\mathcal{R}_\tau$ as follows:

   - $(\alpha \to \beta) \in \mathcal{R}_\tau$ if $(\alpha,\beta) \in <^\tau$.
   - $(\gamma \to (\alpha \to \beta)) \in \mathcal{R}_\tau$ whenever $\gamma <^\tau_* \alpha$ and $\gamma <^\tau_* \beta$ hold where $<^\tau_*$ is the transitive closure of $<^\tau$.

4. Let $\mathcal{K}_\tau$ be the class of all models of the form $(T,<,\mathcal{R}_\tau,\delta,h)$, where T is a tree and $\delta \in T$.

5. Let H_τ be a set of all wffs A such that A holds in any model of $\mathcal{K}_\tau$.

Example 6. Let $\tau = \{\forall x(xRx)\}$. Then the models of $\mathcal{K}_\tau$ have the form $(T,<,\mathcal{R}_\tau,\delta,h)$, where $\mathcal{R}_\tau = < \cup\{\gamma \to (\alpha \to \alpha) \mid \gamma \leqq^* \alpha\}$. This is so since $<^\tau$ is $< \cup\{(\alpha,\alpha) \mid \alpha \in T\}$.

It is easy to see that the meaning of $\Box$ alternates between **K** and **T** modalitie because $\mathcal{R}$ switches the reflexive arcs on and off. Thus the logic H_τ of our example is the same as the logic H_1^S of Section 3 of [2].

We know from [2] that the following is a Hilbert axiomatisation of H_τ. We make use of our irreflexivity rule, see [1].

Axioms: (E,F are wffs without $\Box$).

1. $A \wedge \Box A$, where A is a substitution instance of a truth functional tautology.
2. $\Box(A \to B) \to (\Box A \to \Box B)$
3. $\Box(\Box(A \to B) \to (\Box A \to \Box B))$
4. $\Diamond\top$
5. $\neg E \wedge \Box^2 E \wedge Y \to \Diamond(\neg E \wedge Y)$, where $Y = A$ or $Y = \Box A$, for A without $\Box$.
6. $\neg E \wedge \Box^2 E \wedge \Diamond(\neg E \wedge A) \wedge \Diamond(\neg E \wedge B) \to \Diamond(\neg E \wedge A \wedge B)$.
7. $\neg E \wedge \Box^2 E \to \Diamond(\neg E \wedge A) \vee \Diamond(\neg E \wedge \neg A)$
8. $\Box A \wedge \neg E \wedge \Box^2 E \to \Diamond(\neg E \wedge A)$.
9. $\Box X \wedge \neg E \wedge \Box^2 E \to \Diamond(\neg E \wedge \Box(\neg \wedge \Box^2 F \to \Diamond\neg F \wedge X))$
10. $\neg E \wedge \Box^2 E \wedge \Diamond A \wedge \Diamond(\neg E \wedge \Box Y \wedge \Box\Box Y' \wedge \neg A \wedge \Box\Box X) \to \Diamond(A \wedge X \wedge Y \wedge \Box Y')$ where Y, Y' are without $\Box$
11. $\neg E \wedge \Box^2 E \wedge \Diamond(C \wedge E \wedge Y) \to \Diamond(\neg E \wedge \Diamond(Y \wedge E \wedge \Diamond(C \wedge E))$

Rules

MP: $$\frac{\vdash A; \vdash A \to B}{\vdash B}$$

IRR: $$\frac{\vdash \neg q \wedge \Box^2 q \to A}{\vdash A}$$
where q is an atom not in A.

2-necessitation: $$\frac{\vdash A}{\vdash \Box^2 A}$$

IRR^n: $$\frac{\vdash \bigwedge_{m=1}^{n} \beta_m^m \to A}{\vdash A}$$
where β_m^m are as defined below and q_j^i are all not in A.

The following defines β_j^i:
Let q_j^i be a double indexed sequence of atoms. Let

1. $\beta_1^i(q_1^i) = \neg q_1^i \wedge \Box^2(q_1^i)$.
2. $\beta_2^i(q_1^i, q_2^i) = \neg q_2^i \wedge \Box^2 q_2^i \wedge \Diamond(\neg q_2^i \wedge \Box \beta_1^i)$.
3. $\beta_{n+1}^i(q_1^i, \ldots, q_{n+1}^i) = \neg q_{n+1}^i \wedge \Box^2 q_{n+1}^i \wedge \Diamond(\neg q_{n+1}^i \wedge \Box \beta_n^i)$

Example 7. We now give an example of a class of reactive Kripke models characterising a logic which cannot be presented as a hyper-modal logic. Consider one point models of the following form, see Figure 11

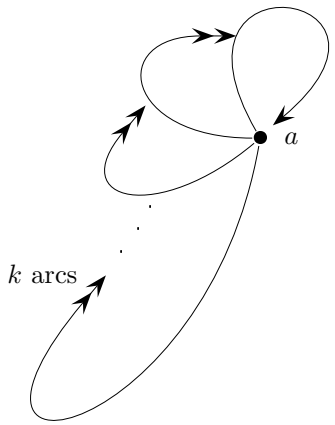


Figure 11

Let

$$\begin{aligned} &\text{arc}_0 = a \to a \\ &\vdots \\ &\text{arc}_{n+1} = (a \to \ \text{arc}_n) \end{aligned}$$

Consider the models $\mathbf{m}_k$ of the form

$$\mathbf{m}_k = (\{a\}, \{\text{arc}_0, \text{arc}_k\}, \{\text{arc}_n \mid n \leq k\}, a)$$

The model $\mathbf{m}_k$ is a one node model with arcs as in Figure 11, where only the arcs $a \to a$ and $a \to (a \to \ldots (a \to a) \ldots)$ (k arrows) are active.

The following points are clear for the class of models $\{\mathbf{m}_k\}$.

1. $\Diamond^n \Box \bot$ is consistent for all n.
2. The following holds in the logic $\mathbf{L}$ of this class

$$\vDash \Diamond^n \Box \bot \to \bigwedge_{m \leq n} (\Box^m A \Leftrightarrow A)$$

for all wffs A without $\Box$.

We claim the logic $\mathbf{L}$ cannot be presented as a hyper-modal logic.

For suppose there is a class of traditional Kripke models of the form (S, R, a) and a sequence $(\Psi_1, \ldots, \Psi_k)$ of conditions characteristing modalites $\Box_1, \ldots, \Box_k$ such that the evaluation of $\Diamond$ alternates according to this sequence (see Definition 9).

Then for a high enough n, the meaning of $\Diamond$ in $\Diamond^n \Box \bot$ starts repeating itself.

Let (S, R, a) be a model of $\Diamond^n \Box \bot$, for n large enough. By property 2, all $\Diamond^m, m \leq n$ satisfy $\vDash_a \Diamond^m A \Leftrightarrow A$, for A without $\Box$. Thus we must have $\Psi_i(a, R) = \{(a, a)\}$. Therefore, how can we also have $\vDash_a \Diamond^n \Box \bot$?

This example shows that there is a class of reactive models of finite level defining a modal logic $\mathbf{L}$ which is not a hyper-modal logic.

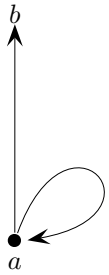


Figure 12

Example 8. We now exhibit a hyper-modal logic which cannot be characterised by a class of finite level reactive models. Consider the situation in Figure 12 Let $\Psi_1 = \{(a, a)\}$ and let $\Psi_2 = \{(a, a), (a, b)\}$. Consider the sequence (Ψ_1, Ψ_1, Ψ_2). This means that $\Box$ is interpreted as seeing only the a node twice and then it can also see b once before repeating. Let $\mathbf{L}$ be the hyper-modal logic defined by this set up.

To implement this logic by reactive models we need to switch the arc $a \to b$ on and off by other arcs in the repeating sequence $(-, -, +)$.

Figure 13 can help visualise the situation:

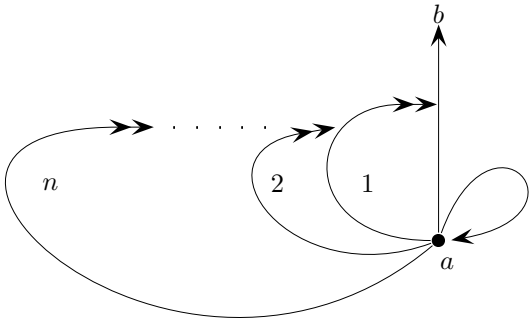


Figure 13

We note that if the connection $a \to b$ must alternate as $(-, -, +)$ then connection $a \to (a \to b)$ (i.e. arc_1 in Figure 13) must alternate $(-, +, +)$ and arc_2 must alternate $(+, -, +)$ and so on.

However in any finite level model, the highest level arcs cannot alternate. Hence modalities of the form $\Box^n$ for high enough n cannot be implemented in any given model. We conjecture that if we allow models of unbounded level then all reasonable hyper-modalities can be implemented. In our case the sequence $(-, -, +)$ can be implemented by a single model of infinite level.

Acknowledgments

I am grateful to David Makinson for valuable comments.

Bibliography

[1] D. M. Gabbay. An irreflexivity lemma with applications to axiomatizations of conditions on tense frames. In U. Monnich, editor, *Aspects of Philosophical Logic*, pages 67–89. D. Reidel, 1981.

[2] D. M. Gabbay. A theory of hypermodal logics: mode shifting in modal logic. *Journal of Philosophical Logic*, 31:211–243, 2002.

[3] D. M. Gabbay and V. Shehtman. Flow products of modal logics.

[4] V. Lifschitz. Circumscription. In C. J. Hogger D. M. Gabbay and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming. Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*. Oxford University Press, 1994.

Appendix

A Hypermodalities

Since we are comparing in this paper the notions of hyper-modal logics and reactive modal logics, we need to give here the exact definition of a hyper-modal logic.

Our starting point is a general Kripke model of the form $\mathbf{m} = (S, R, a, h)$. R is an arbitrary binary relation on S.

We are going to introduce evaluation modes into such semantics. It is convenient to regard the relation xRy as a classical formula $\Psi_{\mathbf{K}}(x, R, a, y)$ in the language of the relation R, the individual variables x, y and actual world constant a as follows

- $\Psi_{\mathbf{K}}(x, R, a, y) =_{\text{def}} xRy$

we have

- $t \vDash \Box A$ iff $\forall s(\Psi_{\mathbf{K}}(t, s) \rightarrow s \vDash A)$.

We can refer to $\Psi_{\mathbf{K}}$ as the mode of evaluation for $\Box$. It is fixed in the semantics and does not change. Intuitively it tells us, for a world x, how to evaluate $\Box A$ at x, namely where to look for worlds y where $y \vDash A$ must hold. The subscript $\mathbf{K}$ indicates that this formula is used in the case of $\mathbf{K}$ modality.

We can think of different formulas Ψ for the mode. Consider for example:

- $\Psi_{\mathbf{T}}(x, R, a, y) =_{\text{def}} xRy \vee x = y$
- $\Psi_{\mathbf{K4}}(x, R, a, y) =_{\text{def}} (\exists n \geq 1)xR^n y$.
  Where $xR^n y$ is defined by induction as:
  - $xR^0 y$ iff $x = y$
  - $xR^{n+1} y$ iff $\exists z(xRz \wedge zR^n y)$.
- $\Psi_{\mathbf{KB}}(x, R, a, y) =_{\text{def}} (xRy \vee yRx)$.

Clearly $\Psi_{\mathbf{K4}}(x, R, a, y)$ is not a first-order formula. It defines the transitive closure of R.

One can think of Ψ as changing the accessibility relation from R to $\lambda x \lambda y \Psi(x, y)$. Another way of looking at Ψ is that it gives us a new mode of how to use R in evaluating the truth value of $\Box A$. The latter view is more convenient to use because we will be shifting modes during the evaluation.

Let us write $\vDash_i$, to mean that the mode Ψ_i is used in the evaluation. Then $\vDash_{\mathbf{K}}$ for arbitrary frames (S, R, a) yields the logic $\mathbf{K}$, $\vDash_{\mathbf{T}}$ yields the logic $\mathbf{T}$, $\vDash_{\mathbf{KB}}$ yields the logic $\mathbf{KB}$ and $\vDash_{\mathbf{K4}}$ yields the logic $\mathbf{K4}$.

Note that our starting point is a frame (S, R, a) with an arbitrary R. We define $\Psi_i(x, R, a, y)$ as a binary relation and use it to evaluate $\Box$. Thus in traditional terms the frame we are using is (S, Ψ_i, a) not (S, R, a). When we shift modalities, i.e. change from $t \vDash_i \Box A$ to $s \vDash_j \Box A$ it is like shifting from (S, Ψ_i, a) to (S, Ψ_j, a). We now give a formal definition of hypermodality.

We treat the simple case is where the number of modes is a finite set μ and there is a function ε for shifting modes. This case is given in the next definition.

Definition 9 (Mode shifting). Let $\mu = \{\Psi_0, \ldots, \Psi_k\}$ be a set of modes and let ε be a function assigning to each $0 \leq i \leq k$ a value $0 \leq \varepsilon(i) \leq k$.

Let (S, R, a, h) be a Kripke model. We define the following (μ, ε) satisfaction in the model

- $t \vDash_i \Box A$ iff $\forall s(\Psi_i(t,s) \to s \vDash_{\varepsilon(i)} A)$.
- We say A is true in the model if $a \vDash_0 A$.

Definition 10.

1. Let $\mathcal{K}$ be a class of models of the form (S,R,a,h). Let (μ,ε) be a mode system. We write $\mathcal{K} \vDash_{(\mu,\varepsilon)} A$ iff for every model (S,R,a,h) in $\mathcal{K}$ we have $a \vDash_0 A$.
2. Let $\mathbf{L}$ be a logic complete for a class $\mathcal{K}$ of Kripke models of the form (S,R,a,h). Let $\mathcal{K}[\mu,\varepsilon]$ be $\{A \mid \mathcal{K} \vDash_{(\mu,\varepsilon)} A\}$. We sometimes write $\mathbf{L}[\mu,\varepsilon]$ for $\mathcal{K}[\mu,\varepsilon]$, when the implicit dependence on $\mathcal{K}$ is clear.

Obviously the nature of hypermodal logic depends on (μ,ε) and its abstract properties and also on the class $\mathcal{K}$ of models chosen.

B Traversing a graph

Definition 11. By a graph G we mean a set S with a binary relation $R \subseteq S^2$. Let $a \in S$ be the starting point. We write $G = (S,R,a)$.

Definition 12.

1. By a Horn closure condition in the language of R we mean a clasuse of the form
$$C : \bigwedge_i x_i R y_i \wedge \bigwedge_j u_j \neq v_j \to xRy.$$
2. A substitution θ from $Z = \{x_i, y_i, u_j, v_j, x, y\}$ into S is a function θ assigning values $\theta(z) \in S$ for each $z \in Z$.
3. We say a relation $R^* \subseteq S^2$ satisfies the clause C if for all θ, if $\theta(x_i) R^* \theta(y_i)$ holds and $\theta(u_j) \neq \theta(v_j)$ holds then $\theta(x) R^* \theta(y)$ also holds. We write $G^* = (S,R^*,a) \vDash C$.

   Let τ be a set of clauses. We say $G \vDash \tau$ iff $G \vDash C$ for all $C \in \tau$.

Lemma 13. Let $G = (S,R,a)$ be a graph and τ a set of clauses. Then there exists the smaller $R^* \supseteq R$ such that $G^* = (S,R^*,a) \vDash \tau$.

Definition 14. Let $G = (S,R,a)$ be a graph and let $\mu = (\tau_1, \ldots, \tau_k)$ be a sequence of sets of clauses. Let $R_1^*, \ldots, R_k^*$ be the closures of R under τ_i resp. Define a μ-path H through S as follows.

The first element of H is $a_0 = a$. The next element of H is a_1 such that $a_0 R_1^* a_1$ holds. We say a_1 is a τ_1 choice.

Assume a_n is a τ_n choice then a_{n+1} is such that $a_n R_{n+1}^* a_{n+1}$ where $R_{n+1}^* = R_{i+1}^*$ if a_n is a τ_i choice, $i < k$ and $R_{n+1}^* = R_1^*$ if a_n is a τ_k choice.

Definition 15 (A reactive graph). Let $G = (S,R,a)$ be a graph. Write $R = \{(x,y) \mid xRy \text{ holds}\}$. Define W as follows:

$$\begin{array}{lcll} W_0 & = & S \times S & \text{connections of level } 0 \\ W_{n+1} & = & S \times W_n & \text{connections of level } n+1 \\ W & = & \bigcup_n W_n & \end{array}$$

We consider any set of connections of level ≥ 1 as a *switch*.

A subset $\bar{R}$ of W can be used to describe S paths H as follows.

1. initial element of H is $a_0 = a$ initial set of $\bar{R}_1 = \bar{R}$.
2. Assume R_n is defined and $\bar{R}_n \subseteq \bar{R}$.
3. We define $\bar{R}_{n+1}$.

   Let $R_{n+1} = \bar{R}_n = \{\alpha \mid (a_n, \alpha) \in R_n\} \cup \{\beta \mid (a_n, \beta) \in (\bar{R}) - R_n\}$.

   We assume (a_n, a_{n+1}) is in $\bar{R}_n$.

Conjecture: Let H_μ be the all possible paths defined on $G = (S,R,a)$ using $\mu = (\tau_1, \ldots, \tau_k)$. Then there exists an $\bar{R} \subseteq W$ that yields the same paths.

Data, Schema and Ontology Integration

Joseph A. Goguen

Department of Computer Science and Engineering, University of California, San Diego, U.S.A.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction and Motivation

Data integration is emerging as a major challenge in the early 21st century. The rise of inexpensive storage media, data warehousing, and especially the web, have made available vast amounts of data. But unfortunately, it can be very difficult to find what you want, and to combine it properly to get what you need. Reasons for this difficulty include the highly variable structure and quality of data; for example, science labs and businesses often have data stored in spreadsheets, or even just formatted files, with little or no "meta-data" to document either format or meaning; moreover, some entries may be incomplete, corrupted, or inconsistent. If all documents had associated "schemas" (also called "data models") that accurately described their structure, and if fully automatic schema integration were feasible, then it would be possible to solve several interesting problems at the syntactic level [4]; however, these two assumptions are far from true. Moreover, format is only a small part of the difficulty, most of which is semantic and pragmatic, not syntactic. Although the so-called "semantic web" vision for the world wide web has received the most publicity, similar problems have appeared, and are being confronted, in other domains, including "workflows" to automate processing the enormous datasets that are increasingly common, not just in astrophysics, proteomics, high energy physics, etc., but also in ecology, agriculture, pharmacology, e-business, geology, and numerous other areas. The growing popularity of XML may make things easier, but it cannot solve the basic problems.

So called "ontologies" have been proposed as a solution. These are not philosophical (metaphysical) assertions about basic "world substances," but terminological systems, items from which can be attached to items in e-documents. They cannot capture real world semantics, but only logical relations between predicates, such as that all humans are mammals; the actual meanings of "human" and "mammal" remain unformalized. Moreover, a given domain may have competing ontologies, each in some ways incomplete and/or ambiguous, and possibly written in different ontology languages, which in turn may be based upon different logical systems. OWL and RDF are currently most prominent, but others include Ontologic, $\mathcal{ALC}$, KIF, KL-ONE, XSB, Flora, and OIL; specialized ontology languages, e.g., Ecolingua and EML for ecology, tend not to have a formal semantics. It follows from this that the ontology approach to data integration may require not just schema and ontology integration, but also ontology language integration, and even ontology logic integration, in such a way that semantics is respected throughout the entire "integration chain," from actual datasets or "documents," through schemas and ontologies, up to ontology logics. Institutions and their morphisms are promising for understanding the meaning of this integration chain, untangling some problems involved, and even suggesting some sound solutions; however, it should be noted that the deep difficulties of data integration have not only technical, but also social aspects.

Section 2 below provides informal background on databases, schemas, schema mappings, ontologies, workflows, and their role in data integration, written for logicians and others not especially familiar with current database technology. Section 3 describes the author's recent work on database theory and practice, including abstract schemas with constraints and their morphisms for m-to-n matches with semantic functions and conditions; this provides a new theory for integrating data in diverse formats. Section 3.1 briefly describes our tool SCIA, which implements this theory for data that has XML DTDs or SML Schemas. Section 4 sketches our approach to data integration via ontology integration, using prior work on institutions, institution morphisms, and Grothendieck institutions to formalize the logic level of the integration chain. Section 4 also includes a brief discussion of connections with the information flow (in the sense of Barwise and Seligman) approach to ontologies.

2 Background

Information comes from and goes to human beings: pixels, bits, marks on paper, etc. have no meaning in themselves, but must be interpreted. So what is stored in databases (or books, cave walls, DVDs, or other media) is not information, but data. Interpretation has been studied for a long time by numerous disciplines, including semiotics, hermeneutics, pattern recognition, literary criticism, ethnomethodology, statistics, media studies, machine learning, phenomenology, cognitive neuro-science, psychophysics, and more. What seems clear is that it is still poorly understood, in part because the narrow confines of the individual disciplines prevent a comprehensive perspective on a complex phenomenon that transcends such boundaries. It is distressingly common to sweep as much complexity as possible under the rug of "context," and it is a pervasive error to think that all the contextual information needed to interpret data

can be digitally encoded and mechanically applied. Human beings are the ground for all information and all interpretation, and human society is the matrix within which all meaning is embedded, and it is a side effect of that process to create a constantly shifting foreground and background, with the latter being called "context."

Moving up the chain a bit, and confining attention to digital data, we find a great variety of storage media, including tape, CD, DVD, flash memory, hard and scuzzy disk, RAM, stick, jukebox, and more. This level is usually taken for granted, but it is non-trivial, as anyone who has ever had to deal with the internals of device drivers can attest. Data at this level is already structured into bits, bytes, tracks, and other device-dependent subdivisions, oriented towards particular patterns of use.

Databases further organize storage media to facilitate operations that either interrogate or update the contents. This organization may involve data encodings according to data type, e.g., integer, character, string, floating point, etc., as well as further structuring, to make relationships between different data items explicit. Types determine what operations are available on the underlying bits, and provide humans with valuable clues about interpretation, while structuring makes it easier to find related data, (usually) provides associated names, and again facilitates interpretation. Unfortunately, several different ways of structuring data, called **meta-models**, are in common use. One, called **relational**, structures data into relations with fields, while another, called **object oriented**, structures data into objects with inheritance and attributes. With the rise of the web, XML is poised to overtake these; its approach is called **semi-structured**, hinting at its greater flexibility. There are also legacy databases that use so-called hierarchical meta-models.

Data about data is called **meta-data**. The most important, or at least best understood, meta-data for a database is its **schema**, which describes its conventions for structuring, typing, and naming data. Schemas for relational and object oriented databases are well established and well studied, as they are for XML, with its DTDs and XML Schemas. However, a great deal of critically important meta-data falls outside the scope of schemas. For example, while it may be possible to say that a certain item of data is measured in feet, it is not possible to say what a "foot" actually is, or to relate it to other units, such as meters. In some cases, the way measurements are taken, the so-called "protocol," can be very important. For example, in ecology, species density is defined as species count divided by area. But for marine species, volume may be the relevant denominator, and it will matter how species counts are obtained, e.g., by a net that is dragged for a certain amount of time, or by observation from some fixed point. The time of year and time of day may also be important, since species migrate at different times of the day and year, and of course variations in weather modify these patterns. The taxonomies used to classify species may also differ, e.g., different criteria may be used, different granularities of classes, etc. Different modes of observation may also have different inherent inaccuracies, some of which may be systematic, e.g., if certain colors are distorted or difficult to distinguish underwater. And all this is just one tip of an enormous iceberg of potentially critical information.

As the above discussion suggests, making use of data in one database may require integrating it with data from others. For example, to interpret a measurement of the density of gray whales at a certain time and place, we might want to know the ocean temperature, the path and time of migration for that species, and to compare current data with data from previous years, among other things. This requires knowing what factors are relevant, how important they are, where to get the necessary data, and how to process it. In general, the necessary data is stored in different databases at different sites, and although it might all be accessible over the internet, some of it may require passwords and/or special knowledge. Typically today, scientists import all the data they need into their own lab, massage it in various ways (such as averaging or interpolating time series to achieve compatability with other time series), and finally process the integrated data, often using hand coded *ad hoc* programs. All this is a far cry from writing a query in a standard language like SQL or XQuery, and running it over a single database. However, the goal of much current research is to bring data integration closer to that paradigm; some of this research is described in the next three subsections. It is not claimed that these three areas include everything that is relevant, or that they are likely to be sufficient for solving all problems of data integration.

2.1 Schema Integration

One standard approach to data integration is to construct a "global" database connected by "views" to the various "local" databases such that the global database contains all the data of interest from the local databases, and can be queried to obtain exactly what is needed for some particular purpose. However,

it is likely that the local databases are all changing over time, and it seems wasteful and inefficient to duplicate and continually update everything. So instead, the global database is only a "virtual" database, represented by a schema, and the views are actually schema mappings. Somewhat more formally, and using some language from category theory, views are schema morphisms, and the situation described above is a cone (and/or cocone) in the category of schemas, with apex a global schema G over local schema L_i. The case of a cone $v_i : G \to L_i$ is called "local as view" while the case of a cocone $u_i : L_i \to G$ is called "global as view." The latter is adequate for processing queries, but the former is also needed if updates are to be processed.

Unfortunately, it can take a lot of effort to construct the necessary views, effort that is usually not worthwhile for any single project. Moreover, the local databases and their schemas are in general evolving, as are the needs of the domain of activity involved, so that additional ongoing work is needed to "maintain" the views, i.e., to keep them up to date; often, no single research project will have the resources or motivation to do this additional work. This motivates the construction of tools to automatically construct schema mappings. Although there has been a good deal of effort in this area, one outcome is that total automation seems infeasible, so that some human intervention is needed to achieve quality results; see Section 3.1.

A different kind of gap is that the local databases often use different meta-models, while the notion of view is limited to schemas having the same meta-model. An expensive option is to "wrap" a database using one meta-model with a "mediator" providing a schema with a different meta-model; this may be practical for legacy databases using obsolete models, but is otherwise not usually worth the trouble. This implies that we need a way of defining views between schemas having different meta-models. To avoid an *ad hoc* approach in which there are n^2 different notions of view among n different kinds of meta-model, it would make sense to have a single notion of schema that can be used for databases having any meta-model; then we only need one notion of schema morphism, which should specialize to views for the individual meta-models. Such a notion is that of an abstract schema, described in Section 3; as far as we know, this work provides the first semantics for n-to-m matches with semantic functions and conditions, among schemas that may have different meta-models.

2.2 Ontologies

Ontologies, in the sense discussed in Section 1, are used to express logical relations among predicates, and in general may involve constants and terms, as well as logical connective, quantifiers, etc. The intention is to establish a standard terminology, with logical relations among terms, for use over a set of databases, in order to simplify both searching for and integrating data. Ontologies are often given as sets of Horn clauses, but as already noted, there are a number of competing formalisms. Moreover, not all ontology languages have a formal semantics, and among those that do, there are great differences in expressiveness.

These problems are analoguous to those already encountered at the meta-model level, as discussed in Section 2.1, and there is also an analoguous way to solve them, namely to formalize logics and morphisms among them in a uniform way. While logicians have seemed reluctant to formalize the notion of "a logic," computer scientists have been less shy, and there is now a considerable literature on "institutions" [10], which are an abstraction of satisfaction relations between sentences and models, in the style of Tarskian semantics, but parameterized over a category of signatures. A very high level of generality is achieved through the use of category theory. (It is unfortunate that the term "model" becomes so heavily overloaded in this collision of database theory and logic; the database notion that most closely corresponds to "model" in the sense of Tarskian model theory and institutional abstract logic is actually that of a database.)

2.3 Workflows

The kinds of data processing required for applications to both scientific research and e-commerce go well beyond what can be accomplished using only database query languages. Data must flow through a complex pipeline, being massaged and combined with other data in a great variety of ways; such processes are called **workflows**. In the case of scientific workflows, more than mere data translation or even complex data massaging are needed, including the use of powerful statistics packages, integration with complex scientific models, and display of selected data using visualization packages. Web-based business workflows have similar complexity, although the components are different. A number of languages

have been developed to describe workflow components and processes, and of course there are also many *ad hoc* solutions. In general, it is a major task to construct a workflow for a specific project using these technologies, and therefore it is a major open problem to develop more flexible and user-friendly approaches. Service integration is a generalization of data integration, and the problems described above also arise in this new and more complex setting.

3 Abstract Schema Morphisms and Schema Matching

We assume familiarity with many-sorted algebra, e.g., [12, 14]. A **signature** is a set S of **sorts** and a set Σ of operation symbols with their **rank**s, each a string of input sorts and a single output sort. The **term algebra** T_Σ contains all well-formed Σ-terms, indexed by their sort. Given a **specification** (Σ, E) where E is a set of Σ-equations, the quotient T_Σ/E by the ground instances of equations in E is **initial**, in that it has a unique Σ-homomorphism to any other (Σ, E)-algebra.

Schemas describe those aspects of a database invariant under transactions, including both structural and integrity constraints. Let D be a fixed **data algebra**, of basic data elements and operations upon them, having one sort for each type of element, and including all elements as constants in its signature. Let T^D be the set of types of D, called **basic types**; then D is a T^D-sorted algebra. Typical elements of T^D are `Bool`, `Int`, `Char`, and `String` (of characters). Elements of D serve as data values in schema instances, i.e., databases or "documents." We assume D has whatever constants, functions and relations are needed. Let P_i be a collection of algebraic specifications for **polymorphic types**, such as sets, bags, lists, or pairs; each P_i includes not only the constructors for that type, called **collectors**, but also **selectors** (which are inverses to constructors) and any needed predicates, given as Boolean valued functions. For each P_i, let c_i be an operation on types (not on data) with the same rank as the main constructor in P_i; examples are unary `SetOf` and `ListOf`, and binary product $\times$. The constructors in P_i structure databases, for example, a relation as a set of records, whereas the c_i are constructors for type expressions.

We assume a set N of **names** for database elements, given as strings, i.e., elements of $D_{\texttt{String}}$. The **type signature** has one sort, plus the operations c_i, with constants including N and the sorts T^D of D. Call the term algebra of this signature the **type algebra**, denoted $T(N)$; its elements, called **type expressions**[1], are fundamental for abstract schemas (see Definition 14 below).

Example 1 (Relational Databases). Let the name of the database itself be B, the names for its relations be $R_1, ..., R_n$, and the names for the fields of each R_i be F_{ij}. Also let P_1 be a theory of finite sets with $c_1 = \texttt{SetOf}$, and let P_k for $k > 1$ define k-tuples, so that c_2 is the binary infix product type constructor $\times$, c_3 is a constructor for types of 3-tuples, etc.[2] Then

$$\mathcal{S}(B) = R_1 \times ... \times R_n \qquad \mathcal{S}(R_i) = \texttt{SetOf}(F_{i1} \times ... \times F_{iK_i}) \qquad \mathcal{S}(F_{ij}) \in T^D$$

The first says a relational database has relations R_i, the second says each R_i is a set of records of type $F_{i1} \times ... \times F_{iK_i}$ having K_i fields F_{ij} which the third says all have values of a basic type. (If preferred, `BagOf` could be used instead of `SetOf`.) Any particular relational schema makes particular choices for these parameters, as illustrated below.

Example 2 (A Relational Student Database). There are three relations, and

$\mathcal{S}(B) = \texttt{Student} \times \texttt{Enrolled} \times \texttt{Course}$
$\mathcal{S}(\texttt{Student}) = \texttt{SetOf}(\texttt{StudentID} \times \texttt{Address} \times \texttt{Major} \times \texttt{GPA})$
$\mathcal{S}(\texttt{Enrolled}) = \texttt{SetOf}(\texttt{StudentID} \times \texttt{CourseId})$
$\mathcal{S}(\texttt{Course}) = \texttt{SetOf}(\texttt{CourseId} \times \texttt{Synopsis})$
$\mathcal{S}(\texttt{StudentID}) = \texttt{Int}$

We omit the remaining basic types for fields. The type algebra contains terms such as $\texttt{SetOf}(\texttt{CourseId} \times \texttt{Synopsis})$ and $\texttt{Student} \times \texttt{Enrolled} \times \texttt{Course}$, which uses the 3-tuple type constructor.

The **name graph** $G(\mathcal{S})$ of a partial function $\mathcal{S} : N \to T(N)$ with a given top name B has B as its root node, and if t is a node of $G(\mathcal{S})$ and if a name n appears in $\mathcal{S}(t)$, then $t.n$ is also a node of $G(\mathcal{S})$, and there is an edge from t to $t.n$ in $G(\mathcal{S})$. Then $\mathcal{S}$ is **acyclic** if its graph $G(\mathcal{S})$ is acyclic; this condition

[1] It seems not to be as widely known as it could be that it is often useful to regard type expressions as an initial algebra, and that initial algebras also elegantly capture the notion of abstract syntax; see [14].

[2] A more sophisticated approach uses only the binary pair constructor with an associative law, so that the n-tuple constructor is built with $n - 1$ pair constructors.

is useful for XML and similar structures, but not websites. Also, name $n \in N$ is **reachable** if it appears in $G(\mathcal{S})$, and $\mathcal{S}$ is **reachable** if every name in N is reachable; let N_* be the set of reachable names.

Definition 3. Given P, an **abstract schema** is a partial function $\mathcal{S} : N \to T(N)$ that is acyclic with respect to a designated **top** name $B \in N$, and n' is a **match to** n if n' occurs in $\mathcal{M}(n)$.

Having explicit element names and treating tags and attributes the same way allows an element to match a tag or attribute in another schema.

Example 4 (XML Schemas). The XML Schema spec is very complex (over 300 printed pages) and ugly, so we treat a simplification, which seems to agree with what Wadler calls "the essence of XML" [19], essentially an abstract schema with only one collector, `ListOf`, due to the inherent ordering in XML syntax. Here is the abstract schemas for a simple XML Schema for books, in which B is `Bib`, and

$$\mathcal{S}(\texttt{Bib}) = \texttt{ListOf}(\texttt{Book}) \qquad \mathcal{S}(\texttt{Book}) = \texttt{title} \times \texttt{year} \times \texttt{Author} \qquad \mathcal{S}(\texttt{Author}) = \texttt{ListOf}(\texttt{author})$$

If the specification for lists includes the empty list, then the constraint that `Author` lists are non-empty requires equation, for which see Example 12 below.

To define the databases that conform to a given abstract schema, we need a suitable signature; it is much larger than the signature of the type algebra.

Definition 5. For $\mathcal{S} : N \to T(N)$ an abstract schema, its **explicit type algebra** $T^{\#}(N_*)$ is built like its type algebra, but with N replaced by N_* and with a new unary type constructor $\#n$ added for each $n \in N_*$. Let E denote the set of **type equations** of $\mathcal{S}$, which are $n = \#n(\mathcal{S}(n))$ for each $n \in N_*$. Then the **signature** $\Sigma(\mathcal{S})$ **of** $\mathcal{S}$ has the initial algebra $T^{\#}(N_*)/E$ as its sorts, plus the signature of each P_i instantiated with each element of $T^{\#}(N_*)/E$.

The `#` operations make types explicit, so that, e.g., `#title(String)` and `#author(String)` are different; also, paths can be traced by following `#` operations in parse trees of explicit type expressions.

Example 6 (Bibliographic Schema Signature). The sorts are type expressions like `ListOf(Book)`, `Book(title` $\times$ `year` $\times$ `Author)`, and `ListOf(title` $\times$ `year` $\times$ `ListOf(author))`, which satisfy the type equations, so that `Bib` is equal `#Bib(ListOf(Book))` and to

`#Bib(ListOf(#Book(#title(String)` $\times$ `#year(Int)` $\times$ `ListOf(#Author(ListOf(#author(String)))))))`

Unused names like `GenomeRef` do not appear in $\Sigma(\mathcal{S})$ because $\Sigma(\mathcal{S})$ is based on N_* not N.

Definition 7. The **specification** of an abstract schema $\mathcal{S} : N \to T(N)$ has $\Sigma(\mathcal{S})$ as its signature, and has as its equations those of the P_i instantiated with all explicit type expressions, plus a spec for D.

It is also convenient to remove "unreachable" types, not in any type expression equal to the top sort.

Example 8 (Bibliographic Schema Specification). If the polymorphic specification P_1 for lists has binary constructor *cons* and selectors *head* and *tail*, then for any explicit type expression t, the equation $head(cons(a, L)) = a$ is included, for a a variable of sort t and L a variable of sort `ListOf`(t). Note that this gives an infinite set of equations; another such set has equations $tail(cons(a, L)) = L$; all are in E.

Definition 9. Let $I_{\mathcal{S}}$ be an initial algebra of the specification of $\mathcal{S}$. Then a **database** of $\mathcal{S}$ is an element of the carrier of the top sort of $I_{\mathcal{S}}$.

Note that such elements can be described by terms, but the implementation in $I_{\mathcal{S}}$ may be quite different.

Example 10 (Student Database). The explicit type expression for the top sort of Example 2 is

`#B(SetOf(#Student(#StudentID(Int)` $\times$ `#Address(String)` $\times$ `#GPA(Real))` $\times$
`SetOf(#Enrolled(#StudentID(Int)`$\times$
`#CourseId(String)))` $\times$ `SetOf(#CourseId(String)` $\times$ `#Synopsis(String))))`

Then a typical small student database for the schema of Example 2 is:

$\langle \{ \langle 215, \textit{Muir}\, 129, \textit{Math}, 3.2 \rangle, \langle 329, \textit{Revelle}\, 774, \textit{CS}, 3.8 \rangle \}, \{ \langle 215, 130 \rangle, \langle 215, 220 \rangle, \langle 329, 130 \rangle, \langle 329, 87 \rangle \}, \{ \langle 130, ... \rangle, \langle 87, ... \rangle, \langle 220, ... \rangle \} \rangle$

where ... indicates omitted text.

Integrity constraints and schema morphisms need notation for selectors in the P_i. If there is a standard notation, we will use it, e.g., `head` and `tail` for lists; for the product type constructor $\times$, we use its argument type names prefixed by `&`. Also, for $n \in N$, let `##`n denote the selector from B to n, using iteration of conjunction over elements in collections (called "mapcar" in LISP).

Definition 11. An **integrity constraint** for schema $\mathcal{S}$ is an equation of the form $(\forall d : B)\ t(d) = \texttt{true}$, using operations in the signature of $\mathcal{S}$. A **constrained abstract schema** is $(\mathcal{S}, C)$ where C is a set of constraints for $\mathcal{S}$. A database m of $\mathcal{S}$ **satisfies** a constraint if $t(m) = \texttt{true}$, and satisfies $(\mathcal{S}, C)$ if it satisfies each constraint in C, in which case we write $m \models (\mathcal{S}, C)$.

Example 12 (Integrity Constraints). The following are typical integrity constraints for Example 4:

$(\forall Y : \texttt{year})\ Y \leq 2004 \qquad (\forall A : \texttt{Author})\ |A| > 0$

where $|\_|$ is a length function assumed to be in the list specification (these do not look like equations, but they are with "$\leq$" and "$>$" as Boolean valued functions followed by an implicit "$= \texttt{true}$".) Even so, these do not have the form required by Definition 11. However, they can be put in that form as follows:

$(\forall d : \texttt{B})\ \texttt{\#\#year}(d) \leq 2004 \qquad (\forall d : \texttt{B})\ |\texttt{\#\#Author}(d)| > 0$

Typical constraints with more than one variable translate to forms with more than one `##`.

The example below has a mapping with both **semantic functions**, which manipulate data values (e.g., for converting meters to feet), and **conditions**, which restrict the application of mapping formulae.

Example 13 (Schema Mappings). Let the abstract schema $\mathcal{S}_1$ be like $\mathcal{S}$ of Example 4 except for adding the following type definitions:

$\mathcal{S}_1(\texttt{author}) = \texttt{fname} \times \texttt{lname} \qquad \mathcal{S}_1(\texttt{fname}) = \texttt{NString} \qquad \mathcal{S}_1(\texttt{lname}) = \texttt{NString}$

where `NString` is a basic type having no space character, and where `fname` and `lname` are elements for the first and last names of authors. Let $\mathcal{S}_2$ be a second abstract schema built on $\mathcal{S}$ of Example 4 by adding the following type definition and constraint:

$\mathcal{S}_2(\texttt{author}) = \texttt{ListOf}(\texttt{NString}) \qquad (\forall A : \texttt{author})\ 0 < |A| < 3$

Now $\mathcal{S}_1$ databases map to $\mathcal{S}_2$ databases by

$\mathcal{M}_{\texttt{author}}(A) = \texttt{\&fname}(A) \bullet \texttt{\&lname}(A)$

where `author` is the type name from $\mathcal{S}_2$, A is a variable of sort `author` from $\mathcal{S}_1$, `&fname` and `&lname` are selector functions for the `author` product type in $\mathcal{S}_1$, and $\bullet$ is the append operation on lists.

The converse mapping for these schemas involves both conditions and semantic functions:

$\mathcal{M}'_{\texttt{lname}}(A) = A \quad \texttt{if}\ |A| = 1 \qquad \mathcal{M}'_{\texttt{lname}}(A) = \texttt{tail}(A) \quad \texttt{if}\ |A| = 2$

$\mathcal{M}'_{\texttt{fname}}(A) = \perp \quad \texttt{if}\ |A| = 1 \qquad \mathcal{M}'_{\texttt{fname}}(A) = \texttt{head}(A) \quad \texttt{if}\ |A| = 2$

where A is a variable of type `author` from $\mathcal{S}_1$, and where $\perp$ is a null value. (Formally, each pair of equations should be one equation with a polymorphic `if_then_else_`)

There is an important duality, under which $\mathcal{M}$ maps databases one way, but maps queries the opposite way (although this paper does not treat queries). For N a name set and $n \in N$, let `%`n be a new variable symbol of sort n, and let `%`$N = \{$`%`$n \mid n \in N\}$. We are now ready for the main concept:

Definition 14. An **abstract schema morphism** from $\mathcal{S} : N \rangle T(N)$ to $\mathcal{S}' : N' \rightarrow T(N')$ is a partial function $\mathcal{M} : N'_* \rightarrow I_{\mathcal{S}}(\texttt{\%}N)$, which is $I_{\mathcal{S}}$ with `%`N adjoined as new constant symbols. Let $\mathcal{M}(m)$ denote the mapping of a $\mathcal{S}$ model m by $\mathcal{M}$. If $(\mathcal{S}, C)$ and $(\mathcal{S}', C')$ are constrained schemas, then a morphism $\mathcal{M}$ is a **constrained abstract schema morphism** if $m \models (\mathcal{S}, C)$ implies $\mathcal{M}(m) \models (\mathcal{S}', C')$.

Example 15 (continuation of Example 13). The schema morphism from $\mathcal{S}$ to $\mathcal{S}'$ in Example 13 is obtained from the formulas for $\mathcal{M}$ in that example, replacing A by `%author`, and assuming that any name not explicitly mentioned is assigned the "obvious" mapping, which is the identity function for leaf nodes, e.g., $\mathcal{M}_{\texttt{year}}(Y) = Y$ for Y the variable `%year` of sort `year` from $\mathcal{S}_1$, is the tupling function for the product constructor, and for the list constructor, is iteration ("`mapcar`") of a given function over list elements.

Proposition 16. Abstract schemas and schema morphisms form a category, with the identity morphism on $\mathcal{S}$ given by $1_{\mathcal{S}}(v) = v$, and with composition of morphisms $\mathcal{M} : N'_* \rightarrow I_{\mathcal{S}}(\texttt{\%}N)$ with $\mathcal{M}' : N''_* \rightarrow I_{\mathcal{S}}(\texttt{\%}N')$ defined by (for the cogniscenti, this is a Kleisli composition)

$\mathcal{M}'; \mathcal{M}(n'') = \mathcal{M}'(n'')[\texttt{\%}n' \leftarrow \mathcal{M}(n')]_{n' \in N'_*}$

which is the result of substituting $\mathcal{M}(\texttt{\%}n')$ for each occurrence of n' in $\mathcal{M}'(\texttt{\%}n)$. Constrained schemas and their morphisms also form a category.

This result makes available many powerful concepts and results from category theory, including a nice notion of equivalence via isomorphism, the correct notions of product, sum, and more generally, limit and colimit, for abstract schemas. All concepts apply to schemas of different kinds, and colimits give an elegant and extremely general notion of heterogeneous schema integration. See [1] for database motivation for the following, and Section 4 for some definitions:

Proposition 17. Abstract schemas as signatures (but with morphisms in the opposite direction), with integrity constraints as sentences, and with databases as models, form an institution. Moreover, particular kinds of schemas (XML, relational, etc.) form particular sub-institutions.

3.1 A Schema Mapping Tool

Our laboratory has designed and built a tool called SCIA which supports integration and transformation of databases having schemas in DTD and XML Schema format [15, 17, 20]. Since fully automatic schema mapping generation is infeasible, this tool attempts to minimize total user effort by identifying the critical decision points, where user input can yield the largest reduction of future matching effort. A **critical point** is where a core context has either no good matches, or else has more than one good 1-to-1 match, where **core contexts** are the most important contextualizing elements for tags within their subtrees. Core context elements typically have a large subtree, and can be found by heuristics and/or user input. In interactive mode, the tool solicites user input at critical points, and then iterates until both user and tool are satisfied; in automatic mode, it does just one pass using default strategies. Each pass has four steps: linguistic and data type matching; structural matching; context check; and combining match results. Other tools only try to find the easiest 1-to-1 matches, leaving all other difficult matches for the user to do by hand [18]; semantic functions, and conditions are not treated at all, or are left for a different tool for view generation, whereas our tool integrates these functions. A major finding is that this approach can significantly reduce total user effort.

4 Ontology Integration

An ontology is just a theory over a logic, i.e., a set of sentences in that logic. Using ontologies to integrate data raises issues analoguous to those discussed in Section 2.1: Mappings of ontologies over a single logic are well enough understood, so that cocones and colimits of ontologies can be used (as in [1, 3]), but to integrate ontologies over different logics, the notion of logic must be formalized, along with morphisms of theories over different logics, for which morphisms of logics will also be needed. Such issues can be addressed using **institutions** [10], which axiomatize the notion of logical system based on Tarski's idea that the *satisfaction* of a sentence by a model is fundamental. However, we need to parameterize sentences and models over signatures, rather than just assume a single fixed signature, as Tarski did. Institutions have been successfully applied to give semantics for powerful module systems [13], and multi-logic specification languages [8], databases [1], behavioral types and semantics for the object paradigm [11], as well as to generalize many results in classical model theory, such as Craig interpolation [7].

An **institution** consists of an abstract category $Sign$ of signatures, a functor $Sen: Sign \to Set$ for sentences, a functor $Mod: Sign^{op} \to Cat$ for models, and a satisfaction relation $\models_\Sigma$ between models and sentences such that for every signature morphism $f: \Sigma \to \Sigma'$, we have $f(M) \models_{\Sigma'} e$ iff $M \models_\Sigma f(e)$. A **theory** over an institution $\mathcal{I}$ is a pair (Σ, E) where E is a set of Σ-sentences. A Σ-model M **satisfies** (Σ, E) iff $M \models_\Sigma e$ for all $e \in E$. The **model class** of a theory, $(\Sigma, E)^\bullet$, is the class of all models that satisfy the theory, and the **theory** $\mathcal{M}^\bullet$ of a class $\mathcal{M}$ of models is the class of all sentences that are satisfied by all models in $\mathcal{M}$. This situation is a Galois connection, which gives us notions of **closed** theory, i.e., such that $(\Sigma, E)^{\bullet\bullet} = (\Sigma, E)$, and closed model class. Then a **theory morphism** $(\Sigma, E) \to (\Sigma', E')$ over $\mathcal{I}$ is a signature morphism $f: \Sigma \to \Sigma'$ such that $f(E) \subseteq E'^{\bullet\bullet}$, and theories with these morphisms form a category denoted $Th(\mathcal{I})$. This category has whatever colimits $Sign$ has [10].

An **institution morphism** from $\mathcal{I}$ to $\mathcal{I}'$ consists of a functor $\Phi: Sign \to Sign'$ and two natural transformations, $\alpha_\Sigma: Sen'(\Phi(\Sigma)) \Rightarrow Sen(\Sigma)$ and $\beta_\Sigma: Mod(\Sigma) \Rightarrow Mod'(\Phi(\Sigma))$, such that $M \models_\Sigma \alpha_\Sigma(e')$ iff $\beta_\Sigma(M) \models_{\Phi(\Sigma)} (e')$, for all signatures Σ in $\mathcal{I}$, Σ-models M in $\mathcal{I}$, and $\Phi(\Sigma)$-sentences e' in $\mathcal{I}'$. Institutions with these morphisms form a category, which we denote Ins.

There is an alternative approach to formalizing ontologies, pursued for example, in [16], using *local logics* in the sense of Barwise and Seligman [2]; however, local logics are actually a special case of

institutions, in which sets of "instances" are models, "types" are sentences, "classification" is satisfaction, the "consequence" relation give morphisms of sentences (as in a more general formulation in [10], but not in the definition above), and the category of signatures has just one object and one morphism. (Sentences in information flow theories are not just types, they are sequents of types.)

The logical heterogeneity of ontology languages creates a need to deal with multiple institutions at once. Let $\mathbb{O}$ be some (finite) subcategory of institutions. Then $GTh(O)$ has objects $(\mathcal{I}, \Sigma, E)$ where $\mathcal{I}$ is in $\mathbb{O}$ and (Σ, E) is a theory of $\mathcal{I}$. A **morphism** in $GTh(O)$ from $(\mathcal{I}, \Sigma, E)$ to $(\mathcal{I}', \Sigma', E')$ consists of an institution morphism $(\Phi, \alpha, \beta) : \mathcal{I} \to \mathcal{I}'$ and a signature morphism $f : \Sigma' \to \Phi(\Sigma)$ such that $E \subseteq \alpha_\Sigma(f(E'))^{\bullet\bullet}$. There is an "obvious" composition that makes $GTh(O)$ into a category; in fact, it is the Grothendieck category of the theory functor $Th : O \to Cat$, and it is also the theory category of the Grothendieck institution [8] of O viewed as a diagram, i.e., a (contravariant) functor from a small category to the category of institutions. The Grothendieck institution [8] is a remarkable construction of a single institution from an indexed family of institutions; its category of signatures is the Grothendieck category of the indexed category of signatures of the institutions involved.

It is known that if O satisfies some reasonable conditions, then $GTh(O)$ is cocomplete. An immediate benefit of this is a powerful "module system" for combining ontologies with heterogeneous logics. This arises from the fact that any cocomplete institution supports a rich collection of constructions on its theories, including instantiation of parameterized theories, sums of theories, and more; such features are important for structuring large and/or complex theories to better support reuse and reasoning. We thus get a powerful method for structuring ontologies into modules, including inheritance and sums of modules, shared submodules, and modules parameterized by other modules, in the style made popular by ML, but originating in the Clear language [6], and further developed under the name **parameterized programming** [9]. An elegant categorical semantics for this is given in [13]. Another benefit of the Grothendieck construction is its ability to lift Craig interpolation from the individual institutions to the whole [7]. This theory will be important when we extend our tool to take advantage of ontologies, providing a sound basis for its design, and an elegant semantics for its operation.

5 Conclusions and Future Work

We plan to extend our data integration tool so that it can handle relational schemas, spreadsheets, etc. This should be relatively straightforward, since it only requires writing a pre-processor to convert abstract schemas into the internal form of the tool (which consists of a tree and a graph in RDF notation).

A next step is to extend the tool to make use of ontologies. A significant issue for this task is connecting ontologies to abstract schemas; a good discussion of the setting for this problem is given in [5]. One natural approach is to translate ontologies into abstract schemas. Some information will be lost, but not as much as might be feared; for example, a logical implication can be translated into an inclusion relation on sorts, as in order sorted algebra [12]. The full generality of institutions is too great for this, so we restrict to description logics or a similar class, the theories of which can be encoded as abstract schemas, so that the mapping generation tool can do the translation. It would be interesting to abstractly characterize the institutions for which this works.

Some more theoretical issues also need further investigation, especially the properties of various categories discussed above, including abstract schemas, constrained abstract schemas, and Grothendieck categories of ontologies over heterogeneous logics. Relationships with approaches based on local logics should also be further explored.

Acknowledgments

I wish to thank Jenny Wang and Young-Kwang Nam for collaboration on the schema matching tool, Kai Lin and Vitaliy Zavesov for work on its implementation, and Bertram Ludäscher for valuable discussions. This material is based on work partially supported by the National Science Foundation under Grant No. ITR 0225676, the Science Environment for Ecological Knowledge (SEEK) project.

Bibliography

[1] Suad Alagic and Philip Bernstein. A model theory for generic model management. In Giorgio Ghelli and Gösta Grahne, editors, *Proc. Database Programming Languages 2001*, pages 228–246. Springer, 2002.

[2] Jon Barwise and Jerry Seligman. *Information Flow: Logic of Distributed Systems*. Cambridge, 1997. Tracts in Theoretical Computer Science 44.

[3] Trevor Bench-Capon and Grant Malcolm. Formalising ontologies and their relations. In *Proceedings of the 16th International Conference on Database and Expert Systems Applications (DEXA '99)*, pages 250–259. Springer, 1999. Lecture Notes in Computer Science, volume 1677.

[4] Philip Bernstein. Applying model management to classical meta data problems. In *Proc. Conf. on Innovative Database Research*, pages 209–220, 2003.

[5] Shawn Bowers and Bertram Ludäscher. An ontology-driven framework for data transformation in scientific workflows. In *Data Integration in the Life Sciences*. Springer, 2004.

[6] Rod Burstall and Joseph Goguen. Putting theories together to make specifications. In Raj Reddy, editor, *Proceedings, Fifth International Joint Conference on Artificial Intelligence*, pages 1045–1058. Department of Computer Science, Carnegie-Mellon University, 1977.

[7] Răzvan Diaconescu. Interpolation in Grothendieck institutions. *Theoretical Computer Science*, 311:439–461, 2004.

[8] Răzvan Diaconescu. Grothendieck institutions. *Applied Categorical Structures*, 10:383–402, 2002.

[9] Joseph Goguen. Principles of parameterized programming. In Ted Biggerstaff and Alan Perlis, editors, *Software Reusability, Volume I: Concepts and Models*, pages 159–225. Addison Wesley, 1989.

[10] Joseph Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, January 1992.

[11] Joseph Goguen and Răzvan Diaconescu. Towards an algebraic semantics for the object paradigm. In Hartmut Ehrig and Fernando Orejas, editors, *Proceedings, Tenth Workshop on Abstract Data Types*, pages 1–29. Springer, 1994. Lecture Notes in Computer Science, Volume 785.

[12] Joseph Goguen and Grant Malcolm. *Algebraic Semantics of Imperative Programs*. MIT, 1996.

[13] Joseph Goguen and Grigore Roşu. Composition of hidden information modules over inclusive institutions. In *From Object-Orientation to Formal Methods: Essays in Honor of Johan-Ole Dahl*. Springer, to appear 2003.

[14] Joseph Goguen, James Thatcher, Eric Wagner, and Jesse Wright. Initial algebra semantics and continuous algebras. *Journal of the Association for Computing Machinery*, 24(1):68–95, January 1977.

[15] Joseph Goguen, Guilian Wang, Young-Kwang Nam, and Kai Lin. Abstract schema morphisms and schema mapping generation. Technical report, Dept. Computer Science and Engineering, UCSD, 2004. Submitted for publication.

[16] Yannis Kalfoglou and Marco Schorlemmer. Information-flow-based ontology mapping. In Robert Meersman and Zahir Tari, editors, *Proc. Intl. Conf. on Ontologies, DataBases, and Applications of Semantics for Large Scale Information Systems*, volume 2519 of *Lecture Notes in Computer Science*, pages 1132–1151. Springer, 2002.

[17] Young-Kwang Nam, Joseph Goguen, and Guilian Wang. A metadata integration assistant generator for heterogeneous distributed databases. In Robert Meersman and Zahir Tari, editors, *Proc. Intl. Conf. on Ontologies, DataBases, and Applications of Semantics for Large Scale Information Systems*, volume 2519 of *Lecture Notes in Computer Science*, pages 1332–1344. Springer, 2002.

[18] Erhard Rahm and Philip Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[19] Jérôme Siméon and Philip Wadler. The essence of XML. In *Proc. Principles of Programming Languages*, pages 1–13. ACM, 2003.

[20] Guilian Wang, Joseph Goguen, Young-Kwang Nam, and Kai Lin. Critical points for interactive schema matching. In Jeffrey Xu Yu, Xuemin Lin, Hongjun Lu, and YanChun Zhang, editors, *Advanced Web Technologies and Applications*, pages 654–664. Springer, 2004.

Using Counterfactuals in Knowledge-Based Programming[1]

Joseph Y. Halpern[1] Yoram Moses[2]

[1] Department of Computer Science, Cornell University, Ithaca, NY 14853, U.S.A.
halpern@cs.cornell.edu

[2] Department of Electrical Engineering, Technion–Israel Institute of Technology, 32000 Haifa, Israel
moses@ee.technion.ac.il

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

[1] A preliminary version of this paper appeared in the Proceedings of the Seventh Conference on Theoretical Aspects of Rationality and Knowledge (TARK), 1998. The full version can be found at http://www.cs.cornell.edu/home/halpern and will appear in *Distributed Computing*.

Knowledge-based programs, first introduced by Halpern and Fagin [6] and further developed by Fagin, Halpern, Moses, and Vardi [3, 4], are intended to provide a high-level framework for the design and specification of protocols. The idea is that, in knowledge-based programs, there are explicit tests for knowledge. Thus, a knowledge-based program might have the form

$$\textbf{if } K(x = 0) \textbf{ then } y := y + 1 \textbf{ else } \mathsf{skip},$$

where $K(x = 0)$ should be read as "you know $x = 0$" and skip is the action of doing nothing. We can informally view this knowledge-based program as saying "if you know that $x = 0$, then set y to $y + 1$ (otherwise do nothing)".

Knowledge-based programs are an attempt to capture the intuition that what an agent does depends on what it knows. They have been used successfully in papers such as [1, 5, 7, 8, 11, 10, 13, 14] both to help in the design of new protocols and to clarify the understanding of existing protocols. However, as we show here, there are cases when, used naively, knowledge-based programs exhibit some quite counterintuitive behavior. We then show how this can be overcome by the use of *counterfactuals* [9, 16]. In this introduction, we discuss these issues informally, leaving the formal details to later sections of the paper.

Some counterintuitive aspects of knowledge-based programs can be understood by considering the *bit-transmission problem* from [3]. In this problem, there are two processes, a *sender* S and a *receiver* R, that communicate over a communication line. The sender starts with one bit (either 0 or 1) that it wants to communicate to the receiver. The communication line may be faulty and lose messages in either direction in any given round. That is, there is no guarantee that a message sent by either S or R will be received. Because of the uncertainty regarding possible message loss, S sends the bit to R in every round, until S receives an *ack* message from R acknowledging receipt of the bit. R starts sending the *ack* message in the round after it receives the bit, and continues to send it repeatedly from then on. The sender S can be viewed as running the program BT_S:

$$\textbf{if } \mathit{recack} \textbf{ then } \mathsf{skip} \textbf{ else } \mathsf{sendbit},$$

where *recack* is a proposition that is true if S has already received an *ack* message from R and false otherwise, while sendbit is the action of sending the bit.[2] Note that BT_S is a *standard* program—it does not have tests for knowledge. We can capture some of the intuitions behind this program by using knowledge. The sender S keeps sending the bit until an acknowledgment is received from the receiver R. Thus, another way to describe the sender's behavior is to say that S keeps sending the bit until it *knows* that the bit was received by R. This behavior can be characterized by the knowledge-based program BT'_S:

$$\textbf{if } K_S(\mathit{recbit}) \textbf{ then } \mathsf{skip} \textbf{ else } \mathsf{sendbit},$$

where *recbit* is a proposition that is true once R has received the bit. The advantage of this program over the standard program BT_S is that it abstracts away the mechanism by which S learns that the bit was received by R. For example, if messages from S to R are guaranteed to be delivered in the same round in which they are sent, then S knows that R received the bit even if S does not receive an acknowledgment.

We might hope to improve this even further. Consider a system where all messages sent are guaranteed to be delivered, but rather than arriving in one round, they spend exactly five rounds in transit. In such a system, a sender using BT_S will send the bit 10 times, because it will take 10 rounds to get the receiver's acknowledgment after the original message is sent. The program BT'_S is somewhat better; using it S sends the bit only five times, since after the fifth round, S will know that R got his first message. Nevertheless, this seems wasteful. Given that messages are guaranteed to be delivered, it clearly suffices for the sender to send the bit once. Intuitively, the sender should be able to stop sending the message as soon as it knows that the receiver will *eventually* receive a copy of the message; the sender should not have to wait until the receiver *actually* receives it.

It seems that there should be no problem handling this using knowledge-based programs. Let $\Diamond$ be the standard "eventually" operator from temporal logic [12]; $\Diamond\phi$ means that ϕ is eventually true, and let $\Box$ be its dual, "always". Now the following knowledge-based program BT^*_S for the sender should capture exactly what is required:

$$\textbf{if } K_S(\Diamond \mathit{recbit}) \textbf{ then } \mathsf{skip} \textbf{ else } \mathsf{sendbit}.$$

[2] Running such a program amounts to performing the statement repeatedly forever.

Unfortunately, BT^*_S does not capture our intuitions here. To understand why, consider the sender S. Should it send the bit in the first round? According to BT^*_S, the sender S should send the bit if S does not know that R will eventually receive the bit. But if S sends the bit, then S knows that R will eventually receive it (since messages are guaranteed to be delivered in 5 rounds). Thus, S should not send the bit. Similar arguments show that S should not send the bit at any round. On the other hand, if S never sends the bit, then R will never receive it and thus S *should* send the bit! It follows that according to BT^*_S, S should send the bit exactly if it will never send the bit. Obviously, there is no way S can follow such a program. Put another way, this program cannot be implemented by a standard program at all. This is certainly not the behavior we would intuitively have expected of BT^*_S.[3]

One approach to dealing with this problem is to change the semantics of knowledge-based programs. Inherent in the semantics of knowledge-based programs is the fact that an agent knows what standard protocol she is following. Thus, if the sender is guaranteed to send a message in round two, then she knows at time one that the message will be sent in the following round. Moreover, if communication is reliable, she also knows the message will later be received. If we weaken the semantics of knowledge sufficiently, then this problem disappears. (See [2] for an approach to dealing with the problem addressed in this paper along these lines.) However, it is not yet clear how to make this change and still maintain the attractive features of knowledge-based programs that we discussed earlier.

In this paper we consider another approach to dealing with the problem, based on counterfactuals. Our claim is that the program BT^*_S does not adequately capture our intuitions. Rather than saying that S should stop sending if S knows that R will eventually receive the bit, we should, instead, say that S should stop sending if it knows that *even if S does not send another message R* will eventually receive the bit.

How should we capture this? Let $do(i, \mathsf{a})$ be the formula that is true at a point (r, m) if process i performs a in the next round.[4] The most obvious way to capture "(even) if S does not send a message then R will eventually receive the bit" uses standard implication, also known as *material implication* or *material conditional* in philosophical logic: $do(S, \mathsf{skip}) \Rightarrow \mathit{recbit}$. This leads to a program such as $\mathsf{BT}^{\Rightarrow}_S$:

$$\textbf{if } K_S(do(S, \mathsf{skip}) \Rightarrow \Diamond \mathit{recbit}) \textbf{ then } \mathsf{skip} \textbf{ else } \mathsf{sendbit}.$$

Unfortunately, this program does not solve our problems. It too is not implementable by a standard program. To see why, suppose that there is some point in the execution of this protocol where S sends a message. At this point S knows it is sending a message, so S knows that $do(S, \mathsf{skip})$ is false. Thus, S knows that $do(S, \mathsf{skip}) \Rightarrow \Diamond \mathit{recbit}$ holds. As a result, $K_S(do(S, \mathsf{skip}) \Rightarrow \Diamond \mathit{recbit})$ is true, so that the test in $\mathsf{BT}^{\Rightarrow}_S$ succeeds. Thus, according to $\mathsf{BT}^{\Rightarrow}_S$, the sender S should *not* send a message at this point. On the other hand, if S *never* sends a message according to the protocol (under any circumstance), then S knows that it will never send a message (since, after all, S knows how the protocol works). But in this case, S knows that the receiver will never receive the bit, so the test fails. Thus, according to $\mathsf{BT}^{\Rightarrow}_S$, the sender S should send the message as its first action, this time contradicting the assumption that the message is never sent. Nothing that S can do is consistent with this program.

The problem here is the use of material implication ($\Rightarrow$). Our intuitions are better captured by using counterfactual implication, which we denote by $>$. A statement such as $\phi > \psi$ is read "if ϕ then ψ", just like $\phi \Rightarrow \psi$. However, the semantics of $>$ is very different from that of $\Rightarrow$. The idea, which goes back to Stalnaker [16] and Lewis [9] is that a statement such as $\phi > \psi$ is true at a world w if in the worlds "closest to" or "most like" w where ϕ is true, ψ is also true. This attempts to capture the intuition that the counterfactual statement $\phi > \psi$ stands for "if ϕ were the case, then ψ would hold". For example, suppose that we have a wet match and we make a statement such as "if the match were dry then it would light". Using $\Rightarrow$, this statement is trivially true, since the antecedent is false. However, with $>$, the situation is not so obvious. We must consider the worlds most like the actual world where the match is in fact dry and decide whether it would light in those worlds. If we think the match is defective for some reason, then even if it were dry, it would not light.

A central issue in the application of counterfactual reasoning to a concrete problem is that we need to specify what the "closest worlds" are. The philosophical literature does not give us any guidance on this point. We present some general approaches for doing so, motivated by our interest in modeling

[3] While intuitions may, of course, vary, some evidence of the counterintuitive behavior of this program is that it was used in a draft of [3]; it was several months before we realized its problematic nature.

[4] We assume that round m takes place between time $m-1$ and m. Thus, the next round after (r, m) is round $m+1$, which takes takes place between (r, m) and $(r, m+1)$.

counterfactual reasoning about what would happen if an agent were to deviate from the protocol it is following. We believe that this example can inform similar applications of counterfactual reasoning in other contexts.

There is a subtle technical point that needs to be addressed in order to use counterfactuals in knowledge-based programs. Traditionally, we talk about a knowledge-based program TELL being implemented by a protocol P. This is the case when the behavior prescribed by P is in accordance with what TELL specifies. To determine whether P implements TELL, the knowledge tests (tests for the truth of formulas of the form $K_i\phi$) in TELL are evaluated with respect to the points appearing in the set of runs of P. In this system, all the agents know that the properties of P (e.g. facts like process 1 always sending an acknowledgment after receiving a message from process 2) hold in all runs. But this set of runs does not account for what may happen if (counter to fact) some agents were to deviate from P. In counterfactual reasoning, we need to evaluate formulas with respect to a larger set of runs that allows for such deviations.

We deal with this problem by evaluating counterfactuals with respect to a system consisting of all possible runs (not just the ones generated by P). While working with this larger system enables us to reason about counterfactuals, processes no longer know the properties of P in this system, since it includes many runs not in P. In order to deal with this, we add a notion of likelihood to the system using what are called *ranking functions* [15]. Runs generated by P get rank 0; all other runs get higher rank. (Lower ranks imply greater likelihood.) Ranks let us define a standard notion of *belief*. Although a process does not *know* that the properties of P hold, it *believes* that they do. Moreover, when restricted to the set of runs of the original protocol P, this notion of belief satisfies the knowledge axiom $B_i\phi \Rightarrow \phi$, and coincides with the notion of knowledge we had in the original system. Thus, when the original protocol is followed, our notion of belief acts essentially like knowledge.

Using the counterfactual operator and this interpretation for belief, we get the program $\mathsf{BT}^{>}_S$:

$$\textbf{if } B_S(do(S, \mathsf{skip}) > \Diamond recbit) \textbf{ then } \mathsf{skip} \textbf{ else } \mathsf{sendbit}.$$

We show that using counterfactuals in this way has the desired effect here. If message delivery is guaranteed, then after the message has been sent once, under what seems to be the most reasonable interpretation of "the closest world" where the message is not sent, the sender believes that the bit will eventually be received. In particular, in contexts where messages are delivered in five rounds, using $\mathsf{BT}^{>}_S$, the sender will send one message.

As we said, one advantage of BT'_S over the standard program BT_S is that it abstracts away the mechanism by which S learns that the bit was received by R. We can abstract even further. The reason that S keeps sending the bit to R is that S wants R to know the value of the bit. Thus, intuitively, S should keep sending the bit until it knows that R knows its value. Let $K_R(bit)$ be an abbreviation for $K_R(bit = 0) \vee K_R(bit = 1)$, so $K_R(bit)$ is true precisely if R knows the value of the bit. The sender's behavior can be characterized by the following knowledge-based program, BT^K_S:

$$\textbf{if } K_S K_R(bit) \textbf{ then } \mathsf{skip} \textbf{ else } \mathsf{sendbit}.$$

Clearly when a message stating the value of the bit reaches the receiver, $K_R(bit)$ holds. But it also holds in other circumstances. If, for example, the $K_S K_R(bit)$ holds initially, then there is no need to send anything.

As above, it seems more efficient for the sender to stop sending when he knows that the receiver will *eventually* know the value of the bit. This suggests using the following program:

$$\textbf{if } K_S(do(S, \mathsf{skip}) \Rightarrow \Diamond K_R(bit)) \textbf{ then } \mathsf{skip} \textbf{ else } \mathsf{sendbit}.$$

However, the same reasoning as in the case of $\mathsf{BT}^{>}$ shows that this program is not implementable. And, again, using belief and counterfactuals, we can get a program $\mathsf{BT}^{\Diamond B}_S$ that does work, and uses fewer messages than $\mathsf{BT}^{>}_S$. In fact, the following program does the job:

$$\textbf{if } B_S(do(S, \mathsf{skip}) > \Diamond B_R(bit)) \textbf{ then } \mathsf{skip} \textbf{ else } \mathsf{sendbit},$$

except that now we have to take $B_R(bit)$ to be an abbreviation for $(bit = 0 \wedge B_R(bit = 0) \vee (bit = 1 \wedge B_R(bit = 1))$. Note that $K_R(bit)$, which was defined to be $K_R(bit = 0)) \vee K_R(bit = 1))$, is logically equivalent to $(bit = 0 \wedge K_R(bit = 0)) \vee (bit = 1 \wedge K_R(bit = 1))$, since $K_R\phi \Rightarrow \phi$ is valid for any formula

ϕ. But, in general, $B_R\phi \Rightarrow \phi$ is not valid, so adding the additional conjuncts in the case of belief makes what turns out to be quite an important difference. Intuitively, $B_R(bit)$ says that R has correct beliefs about the value of the bit.

In the full paper (which will appear in *Distributed Computing* and is available at http://www.cs.cornell.edu/home/halpern/papers/tark98.pdf) we provide a formal model of counterfactuals, and formally analyze the programs $\mathsf{BT}_S^{>}$ and $\mathsf{BT}_S^{\Diamond B}$ in this model, showing that they have the appropriate properties.

Acknowledgments

Work by the first author was supported in part by NSF under grant IRI-96-25901, IIS–0090145, and CTC-0208535, by the Air Force Office of Scientific Research under grant F49620-96-1-0323 and F48620-02-1-0101,and by ONR under grants N00014-00-1-03-41, N00014-01-1-0795, and by the DoD Multidisciplinary University Research Initiative (MURI) program administered by the ONR under grant N00014-01-1-0795.

Bibliography

[1] C. Dwork and Y. Moses. Knowledge and common knowledge in a Byzantine environment: crash failures. *Information and Computation*, 88(2):156–186, 1990.

[2] K. Engelhardt, R. van der Meyden, and Y. Moses. Knowledge and the logic of local propositions. In *Theoretical Aspects of Rationality and Knowledge: Proc. Seventh Conference (TARK 1998)*, pages 29–41. 1998.

[3] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, Mass., 1995.

[4] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. Knowledge-based programs. *Distributed Computing*, 10(4):199–225, 1997.

[5] V. Hadzilacos. A knowledge-theoretic analysis of atomic commitment protocols. In *Proc. 6th ACM Symp. on Principles of Database Systems*, pages 129–134, 1987.

[6] J. Y. Halpern and R. Fagin. Modelling knowledge and action in distributed systems. *Distributed Computing*, 3(4):159–179, 1989. A preliminary version appeared in *Proc. 4th ACM Symposium on Principles of Distributed Computing*, 1985, with the title "A formal model of knowledge, action, and communication in distributed systems: preliminary report".

[7] J. Y. Halpern, Y. Moses, and O. Waarts. A characterization of eventual Byzantine agreement. *SIAM Journal on Computing*, 31(3):838–865, 2001.

[8] J. Y. Halpern and L. D. Zuck. A little knowledge goes a long way: knowledge-based derivations and correctness proofs for a family of protocols. *Journal of the ACM*, 39(3):449–478, 1992.

[9] D. K. Lewis. *Counterfactuals*. Harvard University Press, Cambridge, Mass., 1973.

[10] M. S. Mazer. A link between knowledge and communication in faulty distributed systems. In *Theoretical Aspects of Reasoning about Knowledge: Proc. Third Conference*, pages 289–304. 1990.

[11] M. S. Mazer and F. H. Lochovsky. Analyzing distributed commitment by reasoning about knowledge. Technical Report CRL 90/10, DEC-CRL, 1990.

[12] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, Berlin/New York, 1992.

[13] Y. Moses and M. R. Tuttle. Programming simultaneous actions using common knowledge. *Algorithmica*, 3:121–169, 1988.

[14] G. Neiger and S. Toueg. Simulating real-time clocks and common knowledge in distributed systems. *Journal of the ACM*, 40(2):334–367, 1993.

[15] W. Spohn. Ordinal conditional functions: a dynamic theory of epistemic states. In W. Harper and B. Skyrms, editors, *Causation in Decision, Belief Change, and Statistics*, volume 2, pages 105–134. Reidel, Dordrecht, Netherlands, 1988.

[16] R. C. Stalnaker. A theory of conditionals. In N. Rescher, editor, *Studies in Logical Theory*, American Philosophical Quarterly Monograph Series, No. 2, pages 98–112. Blackwell, Oxford, U.K., 1968. Also appears in W. L. Harper, R. C. Stalnaker and G. Pearce (Eds.), *Ifs*. Dordrecht, Netherlands: Reidel, 1981.

Properties of Intuitionistic Provability and Preservativity Logics

Rosalie Iemhoff[1] Dick de Jongh[2] Chunlai Zhou[3]

[1]Institute for Discrete Mathematics and Geometry, Technical University Vienna
Wiedner Hauptstrasse 8-10 A-1040 Wien, Austria
iemhoff@logic.at

[2]Institute for Logic, Language and Computation, University of Amsterdam
Plantage Muidergracht 24 1018 TV Amsterdam, The Netherlands
dickdj@science.uva.nl

[3]Department of Mathematics, Indiana University
Bloomington IN 47405, U.S.A.
czhou@indiana.edu

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

In this paper we study some intuitionistic modal logics that arise from a specific mathematical interpretation of the modal operations:

$\Box\varphi$ "φ is provable in HA", i.e. $\mathsf{HA} \vdash \varphi$
$\varphi \rhd \psi$ "for all $\sigma \in \Sigma_1$: $\mathsf{HA} \vdash \sigma \to \varphi$ implies $\mathsf{HA} \vdash \sigma \to \psi$",

where HA is Heyting Arithmetic, the constructive counterpart of PA, and Σ_1 is the first level of the arithmetical hierarchy. All the logics we consider are part of *the provability or preservativity logic of* HA, the set of propositional schemes that HA proves about its provability predicate $\Box_{\mathsf{HA}}$ or its preservativity predicate $\rhd_{\mathsf{HA}}$. Provability logic, in the language $L_\Box$, can be considered to be part of preservativity logic, in the language $L_\rhd$, as $\Box A$ can be defined as $\top \rhd A$. Preservativity logic was introduced by [8] as a constructive alternative for interpretability logic. No axiomatization is known for the preservativity logic of HA, but over the last few years at least part (all?) of the logic has been axiomatized (see [4],[5]). In this paper we consider the following principles of the preservativity logic of HA ($\rhd$ and $\Box$ bind stronger than $\wedge, \vee$, that bind stronger than $\to$).

IPC	intuitionistic propositional logic		
$P1$	$A \rhd B \wedge B \rhd C \to A \rhd C$		
$P2$	$A \rhd B \wedge A \rhd C \to A \rhd (B \wedge C)$		
Dp	$A \rhd B \to (A \vee C) \rhd (B \vee C)$		
Mp	$A \rhd B \to (\Box C \to A) \rhd (\Box C \to B)$		
		K	$\Box(A \to B) \to (\Box A \to \Box B)$
$4p$	$A \rhd \Box A$	4	$\Box A \to \Box\Box A$
Lp	$(\Box A \to A) \rhd A$	L	$\Box(\Box A \to A) \to \Box A$
		Le	$\Box(A \vee B) \to \Box(A \vee \Box B)$
Rules:			
$Pres$	$A \to B \;/\; A \rhd B$	Nec	$A \;/\; \Box A$
MP	$A \;\; (A \to B) \;/\; B$		

iP^- denotes the logic given by IPC, the principles $P1, P2$, and the rules $Pres$ and MP. iP is the logic iP^- extended by Dp and is called the *basic preservativity logic* (it is complete for the natural frames, Theorem 4). By $iP4$ we denote the logic iP extended by the principle $4p$. Similarly for the other preservativity principles Lp, Mp, Wp. iK denotes the logic given by IPC, K, and the rules Nec and MP. The logic iK extended by the principle 4 is denoted by $iK4$. Conform tradition, iKL is denoted by iL, Similarly for Le; $iLLe$ denotes iL extended by Le. iPX denotes an arbitrary extension of iP. Lemma 1 below shows that all provability principles can be derived from the preservativity principles.

The non-logical axioms $K, 4, L$ are part of the provability logic GL of PA. This in contrast to Dp and Le, that do not belong to the preservativity logic of PA. The modal study of the principles above is interesting for two main reasons. First, these principles express principles of HA. Therefore, knowledge about them is likely to provide insights in HA, and might help in the search for a complete axiomatization of the provability and preservativity logic of HA. Second, as mentioned above, some of these principles do not belong to the logics regularly studied in intuitionistic modal logic and might therefore be a valuable addition to the field.

In [5] modal completeness results were presented for all logics given by some or all of these principles, except for iPL. In this paper we investigate the relation between the preservativity and provability logics (Section 3), and present fixed point theorems for both iPL and iL (Section 4). From the latter it follows that both the fixed point theorem and the Beth property hold for any extension of these logics in the appropriate language, in particular for the provability and preservativity logic of HA.

1.1 $\Box$-fragments

The $\Box$-fragment of a preservativity logic iPX in $L_\rhd$ is defined to be

$$iPX_\Box := \{A \text{ in } L_\Box \mid iPX \vdash A\}.$$

Here we ask ourselves what the $\Box$-fragment of a given preservativity logic is.
An obvious relation between $\Box$ and $\rhd$ is given by the following lemma ([5]).

Lemma 1. $iP^- \vdash \Box(A \to B) \to A \rhd B$ and $iP^- \vdash A \rhd B \to (\Box A \to \Box B)$.

The guiding idea behind the description of the $\Box$-fragments is the translation $^\circ$ on formulas that inductively replaces all occurrences of $A \rhd B$ by $\Box A \to \Box B$. All preservativity principles except Dp, Mp are derivable in iL under this translation ([5],[4]). It turns out that there are rules that cover the effect of Dp and Mp on the $\Box$-fragment of the preservativity logics that contain them:

$$\begin{array}{ll} DR & \Box A \to \Box B \;/\; \Box(A \vee C) \to \Box(B \vee C) \\ MoR & \Box A \to \Box B \;/\; \Box(\Box C \to A) \to \Box(\Box C \to B). \end{array}$$

We show that for all preservativity logics considered in this paper, these rules determine the $\Box$-fragment of a preservativity logic in the following way.

Theorem 2. (Numbers indicate the sections where the equality is proved.)

$$\begin{array}{lllll} iP_\Box & \overset{3.3}{=} & iK & \overset{3.3}{=} & iK + DR \\ iP4_\Box & \overset{3.1}{=} & iLe & \overset{3.1.1}{=} & iK4 + DR \\ iPL_\Box & \overset{3.2}{=} & iLLe & \overset{3.2}{=} & iL + DR \\ iPM_\Box & \overset{3.3}{=} & iK & \overset{3.3}{=} & iK + DR + MoR \end{array}$$

In particular, if X is one of $4p, Lp$ or empty, then $iPX_\Box = iKX^\circ + DR$. For $X = Mp$, $iPX_\Box = iK + DR + MoR = iK$.

2 Semantics for Preservativity and Modal Logic

Definition 3. *A frame* F is a triple $\langle W, R, \leq\rangle$, where W is a nonempty set of possible worlds, points or nodes, $\leq$ is a partial order and R is a binary relation satisfying $(\leq \circ R) \subseteq R$.
A model M is a quadruple $\langle W, R, \leq, \Vdash\rangle$ where $\langle W, R, \leq\rangle$ is a frame and $\Vdash$ is a forcing relation between points in W and propositional letters which satisfies the following condition, if $x \Vdash p$ and $x \leq y$, then $y \Vdash p$. *(persistence)*

The forcing relation extends to the connectives in IPC in the usual manner:
$M, w \Vdash A \wedge B \equiv_{def} M, w \Vdash A$ and $M, w \Vdash B$;
$M, w \Vdash A \vee B \equiv_{def} M, w \Vdash A$ or $M, w \Vdash B$;
$M, w \Vdash A \to B \equiv_{def} \forall v \geq w(M, v \Vdash A$ implies $M, v \Vdash B)$;
$M, w \Vdash \top$ for any w; $M, w \nVdash \bot$ for any w.
and to $\rhd$-formulas as follows:
$M, w \Vdash A \rhd B \equiv_{def}$ for any v such that wRv, if $M, v \Vdash A$, then $M, v \Vdash B$.
Then $M, w \Vdash \Box A$ iff for any v such that wRv, $M, v \Vdash A$.
Also one has persistence for all formulas.
As a matter of fact, given the persistence for propositional letters, the condition that $(\leq \circ R) \subseteq R$ is a *necessary and sufficient* condition to guarantee persistence for all formulas ([9]), which is different from the condition $(\leq \circ R) \subseteq (R \circ \leq)$ for intuitionistic modal logic (sometimes we write $R \circ \leq$ as $\bar{R}$).

Theorem 4. ([4]) $iP \vdash A$ iff A is valid on all (finite) frames.

It turns out that all intuitionistic *modal* logics iT that we will consider below are complete with respect to some class of frames satisfying additionally:

- (*brilliancy*) $(R \circ \leq) \subseteq R$.

In particular, iK is complete w.r.t the class of finite brilliant frames.

Next we give some basic propositions in preservativity logics. First the connection between the preservation rule and the more-often-used rule: necessitation.

Theorem 5. In any preservativity logic iT containing all theorems in iP^-, the preservation rule and the necessitation rule are equivalent.

The following substitution lemmas are used in our section on fixed points.

Lemma 6. (a) $T \vdash \boxdot(A \leftrightarrow B) \rightarrow (F[A/p] \leftrightarrow F[B/p])$, for $T = iP4$ or $T = iK4$,
(b) If p occurs only *modalized* in F, i.e. only under $\Box$ or $\rhd$, then
$T \vdash \Box(A \leftrightarrow B) \rightarrow (F[A/p] \leftrightarrow F[B/p])$, for $T = iP4$ or $T = iK4$.

Proof: We can prove (a) directly by induction on the complexity of F, and (b) by induction from (a). $\triangleleft$

As in classical provability logic 4 is derivable from L (and here also from Le).

Lemma 7. For $T = iL$ or $T = iLe$, $T \vdash \Box A \rightarrow \Box\Box A$.

3 Conservation Results

The rule BP (BoxPres): $\Box A \rightarrow \Box B / A \rhd B$ plays a dominant role in the following considerations. The rule is discussed in Section 5.2 of [4] where a short proof sketch is given for the admissibility of the rule for iPH. The admissibility of this rule is not automatically preserved for sublogics or superlogics. Each logic needs its own proof, but we will just give the basic one for $iP4$.

3.1 Conservation of $iP4$ over iLe

In the part of this subsection before subsubsection 3.1 we will repeat the argument of [4] because it is very characteristioc. A notational convention: Given a frame M, $[z) := \{w|$ there is a sequence of $w_0 S_0 w_1 \cdots w_n = w$ for some worlds $w_0, w_1, \cdots, w_n$ in M where $S_i \in \{R, \leq\}\}$. Thus $[z)$ stands for the subframe generated by z. The same notation applies to models.

Theorem 8. 1. In $L_\Box$, 4 corresponds to semi-transitivity: $(R \circ R) \subseteq (R \circ \leq)$.
2. In $L_\Box$, $\vdash_{iK4} A$ iff A is valid on all finite transitive frames.
3. The principle $4p$ corresponds to gatheringness: if $wRvRu$, then $v \leq u$.
4. $\vdash_{iP4} A$ iff A is valid on all finite gathering frames.
5. On finite frames Le corresponds to the Le-property:
$\forall wv(wRv \rightarrow \exists x(wRx \leq v \wedge \forall u(vRu \rightarrow x \leq u)))$.
6. $\vdash_{iLe} A$ iff A is valid on all finite brilliant Le-frames.
7. In $L_\Box$, $\vdash_{iLLe} A$ iff A is valid on all finite
transitive conversely well-founded brilliant Le-frames.

Lemma 9. Let $M := \langle W, R, \leq, \Vdash\rangle$ and $N := \langle W, R', \leq, \Vdash\rangle$ be two finite models. If $R' \subseteq R \subseteq (R' \circ \leq)$, then $M, w \Vdash B$ iff $N, w \Vdash B$ for any formula B in $L_\Box$ and any world $w \in W$.

Lemma 10. Let $M := \langle W, R, \leq, \Vdash\rangle$ be a finite Le brilliant model. Then there is a finite gathering model $N = \langle W, R', \leq, \Vdash\rangle$ such that $R' \subseteq R \subseteq (R' \circ \leq)$.

Proof: Assume that $M := \langle W, R, \leq, \Vdash\rangle$ is a finite Le brilliant model. Define:

$$wR'v \equiv_{def} wRv \text{ and } \forall u(vRu \rightarrow v \leq u) \text{ and } N := \langle W, R', \leq, \Vdash\rangle.$$

This model can be shown to have the right properties. $\triangleleft$

Theorem 11. $\vdash_{iLe} A$ iff A is valid on all finite gathering frames.

Proof: The right-to-left direction follows from the fact that Le is derivable in $iP4$ (observe that, by $4p$ and Dp, $\vdash_{iP4} (A \vee B) \rhd (A \vee \Box B)$, and apply Lemma 1). We just need to show the other direction. Suppose that $\nvdash_{iLe} A$. Then by the completeness of iLe, we know that there is a world b in some finite brilliant Le model $M = \langle W, R, \leq, \Vdash\rangle$ such that $M, b \nVdash A$. According to Lemma 10, there is another new finite gathering model $N = \langle W, R', \leq, \Vdash\rangle$ such that $R' \subseteq R \subseteq (R' \circ \leq)$. From Lemma 9, it follows that $N, b \nVdash A$. $\triangleleft$

Corollary 12 (Conservation). $\vdash_{iP4} A$ iff $\vdash_{iLe} A$, for all A in $L_\Box$.

iLe is equivalent to the logic $iK4$ with DR

Lemma 13. Let M be a model on a gathering frame and x, y be two worlds in this model such that xRy. If $y \Vdash A$, then, for any $z \in [y)$, $y \leq z$ and $z \Vdash \boxdot A$.

Lemma 14. $iP4$ satisfies BP: $\vdash_{iP4} A \rhd B$ iff $\vdash_{iP4} (\Box A \to \Box B)$.

Proof: The direction from left to right follows from Lemma 1. We prove the other direction by contraposition. Suppose that $iP4 \nvdash A \rhd B$. It follows that $A \rhd B$ is false at a point w of some finite gathering model M. Then there is a point v such that wRv, $v \Vdash A$ and $v \nVdash B$. Take $W' := \{w\} \cup [v)$, $R' = R \restriction_{W'}$, $\leq' = \leq \restriction_{W'}$, and $x \Vdash p$ iff $x \Vdash' p$ for any propositional variable p, for all $x \in W'$. Observe that M' has a gathering frame. Note that, for any $x \in [v)$ and for any formula B in $L_{\rhd}$, $M', x \Vdash B$ iff $M, x \Vdash B$.

It is clear that $M', w \nVdash \Box B$ because $wR'v$ and $M', v \nVdash B$. By the above lemma, we get that $M', w \Vdash \Box A$ because $R'[w] \subseteq [v)$ and for any $x \in [v), x \Vdash A$. So $M', w \Vdash \Box A$ but $M', w \nVdash \Box B$, which implies that $M', w \nVdash \Box A \to \Box B$. Therefore $\nvdash_{iP4} \Box A \to \Box B$. ◁

Lemma 15. If the rule BP is admissible for iPX, then DR is admissible for iPX, and whence for $iPX_{\Box}$. If in addition $iPX \vdash Mp$, then both DR and MoR are admissible for iPX, and whence for $iPX_{\Box}$.

Theorem 16. iLe is equivalent to the logic $iK4$ with the extra rule DR. Whence $iP4_{\Box} = iLe = iK4 + DR$.

Proof: First show that iLe is contained in $iK4 + DR$. We only need to show that Le is derivable in the latter logic. Since $iK4 + DR \vdash \Box A \to \Box\Box A$, we can get Le immediately by just applying DR. For the other direction, recall Lemma 7 that the principle 4 is derivable in iLe. Whence it remains to show that DR is admissible for iLe, which is the same as showing that it is admissible for $iP4_{\Box}$, by Corollary 12. That DR is admissible for $iP4_{\Box}$ follows from the previous lemma, by applying Lemma 15. ◁

3.2 Conservation of iPL

Lemma 17. The principle Lp corresponds to gatheringness plus converse well-foundedness of the modal relation. Similarly, L corresponds to semi-transitivity plus well-foundedness *([4])*.

Extending the argument of lemma 10 we obtain:

Lemma 18. $iLLe \vdash A$ iff A is valid on all finite gathering conversely well-founded frames.

Theorem 19 (Conservation). $iLLe$ is the $L_{\Box}$-fragment of iPL.

By a syntactic argument one gets

Lemma 20. 1. $iP4 \vdash \Box((\Box C \to C) \rhd C) \to (\Box C \to C) \rhd C$.

2. $iP4 \vdash \Box \boxdot L \leftrightarrow \boxdot L \leftrightarrow \Box L$ where L is $(\Box C \to C) \rhd C$.

The following lemma is then just a matter of careful checking.

Lemma 21 (Detour Lemma). $iPL \vdash A$ iff there exist $C_1, C_2, \cdots, C_n$ such that $iP4 \vdash \Box((\Box C_1 \to C_1) \rhd C_1) \wedge \cdots \wedge \Box((\Box C_n \to C_n) \rhd C_n) \to A$.

Lemma 14 extends to

Lemma 22. If $iP4 \vdash \Box C \to (\Box A \to \Box B)$, then $iP4 \vdash \Box C \to (A \rhd B)$, for all formulas C.

Theorem 23. iPL satisfies BP: $iPL \vdash \Box A \to \Box B$ iff $iPL \vdash A \rhd B$

Corollary 24. $iLLe$ is equivalent to the logic iL with the extra rule DR.
Whence $iPL_{\Box} = iLLe = iL + DR$.

3.3 Conservation of iPM over iK

There is an interesting but complicated model-theoretic proof that iK is the $L_\Box$-fragment of iPM (Theorem 27), again via showing the admissibility of BP. Here we skip this and just give a direct syntactic proof.
We give a proof of $iK + DR + MoR = iPM_\Box$ that uses the following translation on formulas which is related to the translation $^\circ$ given in the introduction.

Definition 25. The translation * from formulas in $L_\triangleright$ to those in $L_\Box$ is inductively defined as follows:

- For p, $\top$ and $\bot$, $p^* = p$, $\top^* = \top$ and $\bot^* = \bot$.
- For $\circ \in \{\vee, \wedge, \rightarrow\}$, $(A \circ B)^* = A^* \circ B^*$.
- $(\neg A)^* = \neg A^*$
- $(A \triangleright B)^* = \Box(A^* \rightarrow B^*)$.

Lemma 26. If $iK \vdash X^*$, then $iPX_\Box = iK$, where X is in $L_\triangleright$.

Proof: Clearly, $iK \subseteq iPX_\Box$. Thus it remains to show that $iPX_\Box \subseteq iK$. Assume that $iPX_\Box \vdash A$. Of course we can consider A as a formula in $L_\triangleright$ according to the fact that $\Box A \equiv (\top \triangleright A)$ in iP. It suffices to show that

$$\text{if } iPX \vdash A, \text{ then } iK \vdash A^* \ (*)$$

because, for any formula B in $L_\Box$, $B^* = B$.
Since $iPX \vdash A$, there is a finite sequence $s_1 s_2 \cdots s_n (= A)$ of formulas in $L_\triangleright$ in which, for any $s_i (1 \leq i \leq n)$,

1. either s_i is in the forms of P_1, P_2, Dp or X,
2. or there are some $A_1, A_2, s_j \in L_\Box$ $(j < i)$ such that $s_i = A_1 \triangleright A_2$ and $s_j = A_1 \rightarrow A_2$,
3. or there are some $s_j, s_k (j, k < i)$ such that $s_k = s_j \rightarrow s_i$.

The sequence $s_1^* s_2^* \cdots s_n^* (= A^*)$ of formulas in $L_\Box$ is a proof of A^* in iK. We treat the first case and leave the others to the reader. If s_i is an instance of P_1, P_2, Dp or X, then it is easy to see that s_i^* is a theorem of iK for the first three, and it follows by assumption for X. ◁

Theorem 27. $iPM_\Box = iK = iK + DR + MoR$.

Theorem 28. $iP_\Box = iK$.

4 Fixed Points and Beth Definability

In this section we will give the fixed point theorems for iL and iPL and point out connections with Beth's Definability Theorem. Let us remind the reader that fixed point theorems are of the form: for each formula $A(p)$ in which p occurs only modalized, there exists a unique B not containing p such that B and $A(B)$ are provably equivalent. The proof of the existence of fixed points in iL is an adaptation of the well-known proof of that property for GL; the proof of the existence of fixed points in iPL derives from the one for IL, the basic interpretability logic ([2]). The connections with Beth's theorem extend the work of [1] on interpretability logic (see also [3], Ch. 5).

A notational convention: AB is the result of substitution of B for p in the formula Ap.

Theorem 29 (Uniqueness Theorem). Suppose that p occurs modalized in A, then $\vdash_L (\boxdot(p \leftrightarrow Ap) \wedge \boxdot(q \leftrightarrow Aq)) \rightarrow (p \leftrightarrow q)$ where $L \in \{iL, iPL\}$.

The proof of the uniquenss theorem in e.g. [7] is intuitionistically acceptable. Proofs of the existence of fixed points for a system usually consist of proving the existence of fixed points for the basic formulas and proving an inductive step. For the inductive step for iPL, we may borrow the following (reformulated) Theorem 2.4 of [2], since its proof did not use classical logic. This means that for iL and iPL we can confine ourselves to proving the basic cases.

Theorem 30. Let U be any extension of iL or iPL satisfying:

FIX: Every formula Ap of the form $\Box Bp$ or $Bp \rhd Cp$ has a fixed point.

and the substitution lemmas (Lemma 6). Then, for every formula Ap with p modalized, there is a formula J such that p does not occur in J and $\vdash_U J \leftrightarrow AJ$.

4.1 Fixed Point Theorems for iL and iPL

Lemma 31. $iL \vdash \Box A\top \leftrightarrow \Box A\Box A\top$ for all formulas A.

Now an application of Theorem 30 suffices.

Theorem 32. If in C the propositional letter p occurs exclusively under $\Box$, then there is a formula D not containing p such that $iL \vdash D \leftrightarrow CD$.

The following theorem is proved similarly to the one for interpretability logic in [2]. To put it more precisely, the fixed point for the formula $A(p) \rhd B(p)$ in iPL is a mirror image of that for the formula $A(p) \rhd_i B(p)$ in IL. This is not surprising since classically $A(p) \rhd B(p)$ is equivalent to $\neg B(p) \rhd_i \neg A(p)$ in IL.

Theorem 33 (Fixed Point Theorem for $A(p) \rhd B(p)$). $\vdash A\Box B\top \rhd B\top \leftrightarrow A(A\Box B\top \rhd B\top) \rhd B(A\Box B\top \rhd B\top)$.

Since we have now proved FIX of Theorem 30 we can conclude

Theorem 34 (Fixed Point Theorem). For every formula Ap with p modalized, there is formula J such that p does not occur in J and $\vdash_{iPL} J \leftrightarrow AJ$.

In iPW, we have a simpler form of fixed point for $Ap \rhd Bp$.

Theorem 35. In iPW, the fixed point of $Ap \rhd Bp$ is $A\top \rhd B\top$.

4.2 Beth Definability and Fixed Points

For a large class of intuitionistic modal logics the Beth property (Definition 36) and the fixed point property (Definition 37) are equivalent. This theorem and its proof is an adaptation of the corresponding theorems and proofs of [1] concerning interpretability logic. An essential difference lies in a change in Maximova's trick ([6]) needed for intuitionistic logic to obtain the Beth property from the existence of fixed points.

Definition 36 (Beth Definability Property). A logic $\mathcal{L}$ has *the Beth Property* iff for all formulas $A(\bar{p}, r)$ the following holds:

- If $\vdash_L \boxdot A(\bar{p}, r) \wedge \boxdot A(\bar{p}, r') \rightarrow (r \leftrightarrow r')$, then there exists a formula $C(\bar{p})$ such that $\vdash_L \boxdot A(\bar{p}, r) \rightarrow (C(\bar{p}) \leftrightarrow r)$.

Definition 37 (Fixed Point Property). A logic $\mathcal{L}$ has the *fixed point property* iff, for any formula $A(\bar{p}, r)$ which is modalized in r, there exists a formula $F(\bar{p})$ such that

- (existence) $\vdash_L F(\bar{p}) \leftrightarrow A(\bar{p}, F(\bar{p}))$
- (uniqueness) $\vdash_L \boxdot(r \leftrightarrow A(\bar{p}, r)) \wedge \boxdot(r' \leftrightarrow A(\bar{p}, r')) \rightarrow (r \leftrightarrow r')$.

Theorem 38 (Equivalence of Beth Definability and Fixed Points). Let $\mathcal{L}$ be an intuitionistic logic with modal operators that extends iL and obeys the substition lemmas. Then $\mathcal{L}$ satisfies the Beth theorem iff $\mathcal{L}$ has the fixed point property.

Since iL and iPL have fixed points, so do their extensions. Hence

Corollary 39. Let $\mathcal{T}$ be an extension of iL or iPL satisfying the conditions of the above theorem. Then $\mathcal{T}$ has the Beth property.

5 Acknowledgements

The first author is supported by a Marie Curie fellowship of the European Union under grant HPMF-CT-2001-01383. The third author is very grateful to Prof. Johan van Benthem for his encouragement and his financial support through the Spinoza project "Logic in Action" during his master study in the Graduate Programme in Logic at ILLC.

Bibliography

[1] Carlos Areces, Eva Hoogland, and Dick de Jongh. Interpolation, definability and fixed points in interpretability logics. In M. Zakharyaschev, K. Segerberg, M. de Rijke, and H. Wansing, editors, *Advances in Modal Logic, Volume 2*, pages 53–76, 2000.

[2] Dick de Jongh and Albert Visser. Explicit fixed points in interpretability logic. *Studia Logica*, 50:39–50, 1991.

[3] Eva Hoogland. *Definability and Interpolation*. ILLC-dissertation series, Universiteit van Amsterdam, DS-2001-05 2001.

[4] Rosalie Iemhoff. *'Provability Logic and Admissible Rules'*. ILLC-dissertation series, Universiteit van Amsterdam, DS-2001-04 2001.

[5] Rosalie Iemhoff. Preservativity logic (an analogue of interpretability logic for constructive theories). *Mathematical Logic Quarterly*, 49(3):11–21, 2003.

[6] Larisa Maximova. Definability theorems in normal extensions of the provability logic. *Studia Logica*, 50:495–507, 1989.

[7] Craig Smoryński. *Self-reference and Modal Logic*. Springer-Verlag, Berlin, 1985.

[8] Albert Visser. Substitutions of Σ_1^0 sentences. *Annals of Pure and Applied Logic*, 114(1-3):227–271, 2002.

[9] Chunlai Zhou. Some intuitionistic preservativity logics and provability logics. ILLC-MoL series, Universiteit van Amsterdam, MoL-2003-01 2003.

Combining Interpreted Languages in Abstract Algebraic Logic[1]

Don Pigozzi

Department of Mathematics, Iowa State University, U.S.A.

Abstract

Algebraic logic can be viewed the part of logic that focuses on the logical equivalence of sentences rather than their validity. This is especially true of abstract algebraic logic where the way in which equivalence and validity interconnect is the central object of study. Algebraic methods have proved useful in several areas of logic when an abstract semantical approach is called for.

Abstract algebraic logic has developed along two distinct lines. In the *Boolean* or *semantic-based* approach, logical equivalence is taken to be a primitive notion, while in the *logistic* or *rule-based* approach it is obtained from a formal system of deduction by abstracting the classical *Lindenbaum-Tarski process*. This is a preliminary report on an ongoing project aimed at developing a general framework that combines these two approaches. Hopefully this will lead to a suitable domain for defining and investigating, in an algebraic context, fibring [15] and other forms of combining logics.[a]

We trace our understanding of the relation between equivalence and validity back to Frege's seminal insight, as interpreted by Church [6], that a (declarative) sentence is to be viewed as a proper name. In particular it *denotes* or *names* something (Church's rendering of the Frege's *bedeuten*). The thing it denotes, i.e., its *denotation* (*Bedeutung*), Church calls its *truth-value*. According to Frege a sentence also has a *sense* (*Sinn*). Carnap [5] makes a similar distinction between the *extension* and the *intension* of a sentence. However, while Carnap's notion of *extension* conforms closely to that of truth-value, for *intension* he clearly has in mind something different from Frege's sense. He uses it to abstract the property common to all sentences that are logically equivalent. We will use *extension*, *truth-value*, and *denotation* synonymously. We reserve the term *intension*, as does Carnap, for the property that abstracts the relation of logical equivalence.

[a] A similar, category-theoretic framework for abstract logic can be found in the theory of *institutions* [7, 10]. For a categorical approach to fibring see [13].

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

[1] Preliminary report on an ongoing project of H. Andréka, I. Németi, and I. Sain, and the author.

1 Languages

The algebraization of a propositional logic is essentially a one-step process, but the algebraization of a quantifier logic requires two steps, the first being a transformation into a propositional language. When starting with a quantifier logic, we assume that the transformation has already been done, so our development begins with a propositional logic and hence a propositional language. A *language type* is a pair $\langle \mathcal{C}, P\rangle$; members of $\mathcal{C}$ are called *logical connectives*, and members of P *atomic sentences*. The finite *rank*, or *arity*, of each $\omega \in \mathcal{C}$ is denoted by $\boldsymbol{\rho}\omega$. The set P of atomic sentences is called the *signature* of the language type. We assume $\mathcal{C} \cap P = \emptyset$.

Let $\langle \mathcal{C}, P\rangle$ be a language. The $(\mathcal{C}, P)$-*sentences*, P-*sentences* or simply *sentences* for short, are defined recursively in the usual way. The set of $(\mathcal{C}, P)$-sentences is denoted by $\mathrm{Se}_{\mathcal{C}}(P)$ or $\mathrm{Se}(P)$. In many contexts the set $\mathcal{C}$ of logical connectives is fixed while the set of atomic sentences varies. For example, in first-order logic the Boolean connectives, quantifiers, and equality are fixed, but the set of extralogical relation symbols often varies. By a *family of language types* we mean a possibly proper class $\big\langle \langle \mathcal{C}, P\rangle : P \subseteq \mathcal{P}\big\rangle$ languages of different signatures but with the same set $\mathcal{C}$ of logical connectives.

The $\mathcal{C}$-*sentential forms* are defined like sentences except that *sentential variables*, *variables* for short, take the place of atomic sentences. Finally, for each $P \subseteq \mathcal{P}$ the $(\mathcal{C}, P)$-*formulas*, or simply P–*formulas*, are defined like the sentential forms except that both sentential variables and atomic sentences of P are taken to be formulas. The class of sentential forms is denoted by $\mathrm{Sf}_{\mathcal{C}}$ or Sf and the class of formulas by $\mathrm{Fm}_{\mathcal{C}}(P)$ or $\mathrm{Fm}(P)$.

If φ is a sentential form, or more generally a P-formula, we write φ in the form $\varphi(v_0, \ldots, v_{n-1})$ to indicate that the variables that occur in φ must appear in the list $v_0, \ldots, v_{n-1}$. Then for any $\psi_0, \ldots, \psi_{n-1} \in \mathrm{Se}(P)$, we denote by $\varphi(\psi_0, \ldots, \psi_{n-1})$ the P-sentence (more precisely the $(\mathcal{C}, P)$-sentence) that results from φ by simultaneously substituting ψ_k for each occurrence of v_k.

For convenience, in the sequel we will also use *formula* as a generic term referring to sentences, formulas, or sentential forms. Technically, the class of formulas includes both sentences and sentential forms ($\mathrm{Fm}_{\mathcal{C}}(P) \supseteq \mathrm{Sf}_{\mathcal{C}} \cup \mathrm{Se}_{\mathcal{C}}(P)$), so no confusion is likely.

2 Languages as algebras

Let $\langle \mathcal{C}, P\rangle$ be an arbitrary language type. By an *algebra of language type* $\langle \mathcal{C}, P\rangle$, we mean a system $\boldsymbol{A} = \langle A, \omega^{\boldsymbol{A}}, \boldsymbol{p}^{\boldsymbol{A}}\rangle_{\omega\in\mathcal{C}, \boldsymbol{p}\in P}$, where A is a nonempty set (called the *universe* or *carrier* of $\boldsymbol{A}$), $\omega^{\boldsymbol{A}}$ is a $\boldsymbol{\rho}\omega$-ary operation on A (i.e., $\omega^{\boldsymbol{A}}\colon A^{\boldsymbol{\rho}\omega} \to A$) for each $\omega \in \mathcal{C}$, and, for each $\boldsymbol{p} \in P$, $\boldsymbol{p}^{\boldsymbol{A}}$ is a 0-ary operation of A, called a *distinguished element* of $\boldsymbol{A}$. $\mathcal{C}$ is called the *logical language type* of $\boldsymbol{A}$ and P is its *signature*. Note that the language type of $\boldsymbol{A}$ is a pair whose components are the logical language type and the signature of $\boldsymbol{A}$.

The *algebra of sentences of signature* P is $\langle \mathrm{Se}(P), \omega^{\mathbf{Se}(P)}, \boldsymbol{p}\rangle_{\omega\in\mathcal{C},\, \boldsymbol{p}\in S}$. It is denoted by $\mathbf{Se}(P)$. Corresponding to the family of languages $\big\langle \langle \mathcal{C}, P\rangle : P \subseteq \mathcal{P}\big\rangle$ we have a *family of sentence algebras* $\langle\, \mathbf{Se}(P) : P \subseteq \mathcal{P}\,\rangle$. The *algebra of sentential forms* $\mathbf{Sf} = \langle \mathrm{Sf}, \omega^{\mathbf{Sf}}\rangle_{\omega\in\mathcal{C}}$ and the *algebra of* P*-formulas*

$$\mathbf{Fm}(P) = \langle \mathrm{Fm}(P), \omega^{\mathbf{Fm}(P)}, \boldsymbol{p}\rangle_{\omega\in\mathcal{C},\, \boldsymbol{p}\in S}$$

are defined similarly. The signature of the first is empty and the signature of the second is P. We also have a *family of formula algebras* $\langle\, \mathbf{Fm}(P) : P \subseteq \mathcal{P}\,\rangle$.

3 Interpreted languages and the Leibniz relation

The language has been formalized; the next step is to formalize the notion of an interpretation and with it the notions of meaning, intension, and extension. A $(\mathcal{C}, P)$-*matrix* is a pair $\mathcal{A} = \langle \boldsymbol{A}, F_{\mathcal{A}}\rangle$ where $\boldsymbol{A}$ is a $(\mathcal{C}, P)$-algebra and $F_{\mathcal{A}}$ a subset of the universe A of $\boldsymbol{A}$ called the *designated filter* of $\mathcal{A}$. The sentences of a (loosely) interpreted language will take their meanings in the universe A of some $(\mathcal{C}, P)$-matrix $\mathcal{A}$. $F_{\mathcal{A}}$ is the set of meanings of true sentences (under the given interpretation), and its complement $A \setminus F_{\mathcal{A}}$ the set of meanings of the false sentences. The set of meanings inherits its algebraic structure from the sentence algebra via the compositionality of the meaning function. Thus attached to each $(\mathcal{C}, P)$-matrix $\mathcal{A} = \langle \boldsymbol{A}, F_{\mathcal{A}}\rangle$ is a canonical homomorphism from $\mathbf{Se}(P)$ to $\boldsymbol{A}$ that will carry the meanings of the sentences

in the interpretation. The *nominal* elements of $\mathcal{A}$ are those that are actually the meaning (under a given interpretation) of some sentence. $\mathcal{A}$ is *nominal* if each element if nominal.

Definition 1. Let $\langle \mathcal{C}, P\rangle$ be an arbitrary language type. By an (*algebraic*) *interpretation* of $\langle \mathcal{C}, P\rangle$ we will mean any nominal $(\mathcal{C}, P)$-matrix $\mathcal{A} = \langle \boldsymbol{A}, F_{\mathcal{A}}\rangle$. The canonical surjective homomorphism from $\mathbf{Se}(P)$ onto $\boldsymbol{A}$ is called the *meaning function* or *meaning homomorphism* of $\mathcal{A}$ and is denoted by $\mathrm{mng}_{\mathcal{A}}$. $\boldsymbol{A}$ is called the *underlying meaning algebra* and $F_{\mathcal{A}}$ the *truth set* of $\mathcal{A}$.

In a context in which $\mathcal{C}$ is fixed we often refer to a $(\mathcal{C}, P)$-interpretation as a *P-interpretation*.

Definition 2.

(i) By an (*algebraically*)*interpreted language* we mean a language type $\langle \mathcal{C}, P\rangle$ together with an single interpretation of $\langle \mathcal{C}, P\rangle$.

(ii) A *loosely* (*algebraically*)*interpreted language* is a language type together with a class of interpretations, called *admissible*.

(iii) A *loosely* (*algebraically*)*interpreted language family* is a language type family $\big\langle \langle \mathcal{C}, P\rangle : P \subseteq \mathcal{P}\big\rangle$ together with a class of *admissible* interpretations for each $\langle \mathcal{C}, P\rangle$, $P \subseteq \mathcal{P}$.

The consequence relation naturally associated with any loosely interpreted language has all the properties normally associated with such relations.

Definition 3. A sentence φ is said to be a *consequence* of a set of sentences Γ in a loosely interpreted language, in symbols $\Gamma \vDash \varphi$, if, for every admissible interpretation $\mathcal{A}$, $\mathrm{mng}^{\mathcal{A}}(\varphi) \in F_{\mathcal{A}}$ whenever $\mathrm{mng}^{\mathcal{A}}(\Gamma) \subseteq F_{\mathcal{A}}$.

With both the meaning and truth-value (extension) of a sentence clear in an interpreted language, we can define its intension in terms of the notion of indiscernibility of meanings in the interpretation.

Definition 4. Let $\mathcal{A} = \langle \boldsymbol{A}, F_{\mathcal{A}}\rangle$ be a nominal $(\mathcal{C}, P)$-matrix.

(i) Two elements $a, b \in A$ are said to be *$\mathcal{A}$-indiscernible* if, for every $\vartheta(v) \in \mathrm{Fm}(P)$ with a single sentential variable v, we have $\vartheta^{\boldsymbol{A}}(a) \in F_{\mathcal{A}}$ iff $\vartheta(b) \in F_{\mathcal{A}}$.

(ii) Let $\omega_{\mathcal{A}} = \{\, \langle a, b\rangle \in A^2 : a \text{ and } b \text{ are indiscernible in } \mathcal{A} \,\}$. $\omega_{\mathcal{A}}$ is called the *Leibniz relation* on $\mathcal{A}$.

Theorem 5. Let $\mathcal{A}$ be a nominal $(\mathcal{C}, P)$-matrix. Then the Leibniz relation is a congruence relation on $\boldsymbol{A}$ that is compatible with $F_{\mathcal{A}}$ in the sense that, if $a \equiv b \pmod{\omega_{\boldsymbol{A}}}$ and $a \in F_{\mathcal{A}}$, then $b \in F_{\mathcal{A}}$. Moreover, $\omega_{\mathcal{A}}$ is the largest congruence on $\boldsymbol{A}$ with this property.

The P-sentences φ and ψ have the same *intension* in a P-interpretation $\mathcal{A}$ if their meanings are $\mathcal{A}$-indiscernible, i.e., if $\mathrm{mng}^{\mathcal{A}}(\varphi) \equiv \mathrm{mng}^{\mathcal{A}}(\psi) \pmod{\omega_{\mathcal{A}}}$. They have the same *extension* if they have the same truth-value, i.e., either $\mathrm{mng}^{\mathcal{A}}(\varphi)$ and $\mathrm{mng}^{\mathcal{A}}(\psi)$ are both in $F_{\mathcal{A}}$ or are both in $\bar{F}_{\mathcal{A}}$, where $\bar{F}_{\mathcal{A}} = A \backslash F_{\mathcal{A}}$. Abstracting, the *intension* of a sentence φ is taken to be the equivalence class of its meaning under the Leibniz congruence, i.e., $\mathrm{mng}^{\mathcal{A}}(\varphi)/\omega_{\mathcal{A}}$. Its *extension* is its truth-value, and is true or false depending on whether of not $\mathrm{mng}^{\mathcal{A}}(\varphi) \in F_{\mathcal{A}}$.

An interpreted language is said to be *Fregean* if the extensionality relation is a congruence. In this case the intensionality and extensionality relations coincide. A loosely interpreted language is *Fregean* if each of its interpretations is Fregean.

An interpretation $\mathcal{A}$ is *intensional* if the meaning of every sentence is uniquely determined by its intension, i.e., if $\omega_{\mathcal{A}}$ is the identity relation. In an intensional interpretation the meaning of any sentence can be identified with its intension. Extending this terminology we say that a (loosely) interpreted language is *intensional* if its unique interpretation is intensional (all its interpretations are intensional). By the *intensional reduction* of an interpretation $\mathcal{A}$ we mean the quotient matrix $\mathcal{A}^* = \langle \boldsymbol{A}/\omega_{\mathcal{A}}, F_{\mathcal{A}}/\omega_{\mathcal{A}}\rangle$.

Theorem 6. An intentional interpreted language is Fregean iff its unique interpretation is a two-element matrix.

Definition 7. Two sentences φ and ψ in a loosely interpreted language are said to be *logically equivalent* if φ and ψ have the same intension, i.e., $\mathrm{mng}^{\mathcal{A}}(\varphi) \equiv \mathrm{mng}^{\mathcal{A}}(\psi) \pmod{\omega_{\mathcal{A}}}$, in every interpretation $\mathcal{A}$.

Since the intensionality and extensionality relations coincide for Fregean interpreted languages, we have that in a Fregean loosely interpreted langauge, two sentences are logically equivalent iff they have the same truth-value in every interpretation. So as one would expect, our notion of logical equivalence coincides with Carnap's for Fregean loosely interpreted languages.

Two sentences are logically equivalent in a loosely interpreted language iff they are logically equivalent in its intensional reduction. This follows easily from the fact that the Leibniz congruence is compatible with the truth set.

4 Rule-based interpreted languages

Rule-based interpretations are derived directly from the deductive mechanism of the logic. Let $\mathcal{C}$ be a set of logical connectives. A *rule over* $\mathcal{C}$ is an ordered pair of the form $\langle \Gamma, \varphi \rangle$ where $\Gamma \subseteq \mathrm{Sf}_{\mathcal{C}}$ and $\varphi \in \mathrm{Sf}_{\mathcal{C}}$. Its *cardinality* is the cardinality of its set of premisses, i.e., $|\Gamma|$. A rule of cardinality 0 is called an axiom and identified with its conclusion φ.

Definition 8. A *rule-based logic* over a logical language type $\mathcal{C}$ is a triple $\mathcal{D} = \langle \mathcal{C}, \mathrm{Ax}_{\mathcal{D}}, \mathrm{Ru}_{\mathcal{D}} \rangle$, where $\mathrm{Ax}_{\mathcal{D}}$ is a set of axioms and $\mathrm{Ru}_{\mathcal{D}}$ a set of rules over $\mathcal{C}$. The *cardinality* of $\mathcal{D}$ is the smallest cardinal greater than the cardinality of each rule in $\mathrm{Ru}_{\mathcal{D}}$.

An interpretation of a rule-based logic is obtained by adjoining extra-logical axioms to the logical axioms and rules. Let $\mathcal{D} = \langle \mathcal{C}, \mathrm{Ax}_{\mathcal{D}}, \mathrm{Ru}_{\mathcal{D}} \rangle$ be a rule-based logic and P a signature. A set T of P-sentences (more precisely $(\mathcal{C}, P)$-sentences) is called a *$\mathcal{D}$-theory over P* if it contains all substitution instances of the axioms and is closed under the inference rules of $\mathcal{D}$. T is *axiomatized* by a set Γ of sentences if it is the smallest theory including Γ.

Definition 9. Let $\mathcal{D}$ be a rule-based logic over $\mathcal{C}$. An interpretation of a language type $\langle \mathcal{C}, P \rangle$ is *$\mathcal{D}$-based* if it is the intensional reduction of a matrix of the form $\langle \mathbf{Se}(P), T \rangle$ for some theory $\mathcal{D}$-theory T over P. It is said to be *axiomatized* by a set of sentences if T axiomatized by Γ.

An interpretation is *rule-based* if it is $\mathcal{D}$-based for some rule-based logic $\mathcal{D}$ over $\mathcal{C}$.

Every rule-based logic $\mathcal{D}$ over a logical language type $\mathcal{C}$ defines an intensional loosely interpreted family of languages whose admissible interpretations are the $\mathcal{D}$-based interpretations of $\langle \mathcal{C}, P \rangle$ for every signature P. A sentence φ is said to be a *$\mathcal{D}$-consequence* of a set of sentences Γ, in symbols $\Gamma \models_{\mathcal{D}} \varphi$, if $\mathrm{mng}_{\mathcal{A}}(\varphi) \in F_{\mathcal{A}}$ where $\mathcal{A}$ is the $\mathcal{D}$-based interpretation axiomatized by Γ.

Theorem 10. Let $\mathcal{D}$ be rule-based logic over $\mathcal{C}$. The following are equivalent for all $\Gamma \cup \{\varphi\} \subseteq \mathrm{Se}(P)$.

(i) $\Gamma \models_{\mathcal{D}} \varphi$.

(ii) φ is a consequence of Γ in the loosely interpreted language defined by $\mathcal{D}$.

(iii) φ derivable from Γ by the logical axioms and rules of $\mathcal{D}$.

First-order predicate logic gives rise to a rule-based loosely interpreted language by eliminating individual variables and formalizing substitution for individual variables in the object language. The transformed language will have as its logical language type $\mathcal{C} = \{\rightarrow, \vee, \wedge, \neg, \top, \bot\} \cup \{ \Diamond_i : i < \omega \} \cup \{ \mathsf{d}_{ij} : i, j < \omega \}$, where the $\Diamond_i$ are modal connectives and the d_{ij} are constants representing respectively the (transforms of the) quantifiers $\exists x_i$ and the atomic equality formulas $x_i = x_j$. The signature P contains an atomic formula r for each predicate symbol R.[2]

5 Semantic-based interpreted languages

semantic-based interpretations are derived from the primitive notion of *class of models*—an arbitrary class—together with the primitive notion of *meaning*, a function that assigns to each element of the class and each sentence a "meaning". The other primitive notion involved in defining interpretations is the

[2]For details of the formalization of predicate logic as a rule-based interpreted language see [4, 11]. For a formulation of predicate logic where substitution is simulated in the object language see [12]. For a discussion of quantifiers as modal operators see [3, 14].

validity relation. This specifies which sentences are "true" in each model. The fundamental condition that connects meaning and validity is the following: if a sentence is true in a given model, then so is any sentence with the same meaning.

Definition 11. Let $\langle \mathcal{C}, P\rangle$ be a language type. Let $\mathcal{S} = \big\langle \langle \mathcal{C}, P\rangle, \mathsf{Mod}_{\mathcal{S}}, \mathrm{mng}_{\mathcal{S}}, \models_{\mathcal{S}}\big\rangle$ be an ordered four-tuple where

(i) $\mathsf{Mod}_{\mathcal{S}}$ is a class, called the *class of models* of $\mathcal{S}$,

(ii) $\mathrm{mng}_{\mathcal{S}}$ is a function with domain $\mathsf{Mod}_{\mathcal{S}} \times \mathrm{Se}(P)$, called the *meaning function* of $\mathcal{S}$, and

(iii) $\models_{\mathcal{S}}$ is a subclass of $\mathsf{Mod}_{\mathcal{S}} \times \mathrm{Se}(P)$, called the *validity relation* of $\mathcal{S}$.

We write $\mathfrak{M} \models_{\mathcal{S}} \varphi$ in place of $\langle \mathfrak{M}, \varphi\rangle \in \models_{\mathcal{S}}$ and $\mathrm{mng}^{\mathfrak{M}}_{\mathcal{S}}(\varphi)$ in place of $\mathrm{mng}_{\mathcal{S}}(\mathfrak{M}, \varphi)$.

$\mathcal{S}$ is called a *pre-semantical system* if the *conceptuality condition* holds, i.e.,

(iv) $\mathfrak{M} \models_{\mathcal{S}} \varphi$ and $\mathrm{mng}^{\mathfrak{M}}_{\mathcal{S}}(\varphi) = \mathrm{mng}^{\mathfrak{M}}_{\mathcal{S}}(\psi)$ implies $\mathfrak{M} \models_{\mathcal{S}} \psi$, for all $\mathfrak{M} \in \mathsf{Mod}_{\mathcal{S}}$ and $\varphi, \psi \in \mathrm{Se}(P)$.

$\mathcal{S}$ is called a *semantical system* if in addition the meaning function satisfies the *compositionality condition for meanings*, i.e.,

(v) For all $\mathfrak{M} \in \mathsf{Mod}_{\mathcal{S}}$, $\omega \in \mathcal{C}$, and $\varphi_0, \dots, \varphi_{\boldsymbol{\rho}\omega-1}, \psi_0, \dots, \psi_{\boldsymbol{\rho}\omega-1} \in \mathrm{Se}(P)$, if $\mathrm{mng}^{\mathfrak{M}}_{\mathcal{S}}(\varphi_i) = \mathrm{mng}^{\mathfrak{M}}_{\mathcal{S}}(\psi_i)$ for all $i < \boldsymbol{\rho}\omega$, then $\mathrm{mng}^{\mathfrak{M}}_{\mathcal{S}}(\omega\varphi_0 \cdots \varphi_{\boldsymbol{\rho}\omega-1}) = \mathrm{mng}^{\mathfrak{M}}_{\mathcal{S}}(\omega\psi_0 \cdots \psi_{\boldsymbol{\rho}\omega-1})$.

The validity relation $\models_{\mathcal{S}}$ defines a Galois connection between models and sentences in the usual way. For each set Γ of sentences define

$$\mathsf{Mo}\,\Gamma = \{\, \mathfrak{M} \in \mathsf{Mod} : \text{ for every } \varphi \in \Gamma,\ \mathfrak{M} \models_{\mathcal{S}} \varphi \,\},$$

and for each class K of models define

$$\mathrm{Th}\,\mathsf{K} = \{\, \varphi \in \mathrm{Se}(P) : \text{ for every } \mathfrak{M} \in \mathsf{K},\ \mathfrak{M} \models_{\mathcal{S}} \varphi \,\}.$$

$\mathsf{Mo}\,\Gamma$ is called the *class of models* of Γ and $\mathrm{Th}\,\mathsf{K}$ the *theory* of K. A sentence φ is said to be an $\mathcal{S}$-*consequence* of Γ, in symbols $\Gamma \models_{\mathcal{S}} \varphi$, if $\varphi \in \mathrm{Th}\,\mathsf{Mo}\,\Gamma$.

Let $\mathcal{S}$ be a semantical system, and let $\mathfrak{M}$ be a model of $\mathcal{S}$. We denote the range of the function $\mathrm{mng}^{\mathfrak{M}}$ for any $\mathfrak{M}$ by $\mathrm{Me}\,\mathfrak{M}$, called the *meaning set of* $\mathfrak{M}$. The structure of $\mathfrak{M}$ is embodied in its meaning set. The compositionality condition guarantees that $\mathrm{Me}\,\mathfrak{M}$ can be given the structure of a (unique) algebra.

Definition 12. Let $\mathcal{S}$ be a semantical system over the language type $\langle \mathcal{C}, P\rangle$, and let $\mathfrak{M} \in \mathsf{Mod}_{\mathcal{S}}$. By the *meaning algebra of* $\mathfrak{M}$ we mean the algebra of type $\langle \mathcal{C}, P\rangle$

$$\mathbf{Me}\,\mathfrak{M} = \langle \mathrm{Me}\,\mathfrak{M}, \omega^{\mathbf{Me}\,\mathfrak{M}}, \boldsymbol{p}^{\mathbf{Me}\,\mathfrak{M}}\rangle_{\omega\in\mathcal{C},\,\boldsymbol{p}\in P},$$

where $\omega^{\mathbf{Me}\,\mathfrak{M}}\big(\mathrm{mng}^{\mathfrak{M}}(\varphi_0), \dots, \mathrm{mng}^{\mathfrak{M}}(\varphi_{\boldsymbol{\rho}\omega-1})\big) = mng^{\mathfrak{M}}(\omega\varphi_0 \cdots \varphi_{\boldsymbol{\rho}\omega-1})$ for each $\omega \in \mathcal{C}$, all $\varphi_0, \dots, \varphi_{\boldsymbol{\rho}\omega-1} \in \mathrm{Se}(P)$, and $\boldsymbol{p}^{\mathbf{Me}\,\mathfrak{M}} = \mathrm{mng}^{\mathfrak{M}}(\boldsymbol{p})$ for each $\boldsymbol{p} \in P$.

The meaning homomorphism $\mathrm{mng}^{\mathfrak{M}} : \mathbf{Se}(P) \twoheadrightarrow \mathbf{Me}\,\mathfrak{M}$ is unique since $\mathbf{Se}(P)$ is a minimal algebra. It is also surjective. Hence $\mathbf{Me}\,\mathfrak{M}$ is also minimal.

In addition to the structure of the meaning algebra, the semantical system also specifices the sentences "true" in $\mathfrak{M}$ by means of the validity relation. Recall that *theory of* $\mathfrak{M}$ is defined to be the set $\mathrm{Th}\,\mathfrak{M} = \{\, \varphi \in \mathrm{Se}(P) : \mathfrak{M} \models \varphi \,\}$. Abstracting from first-order predicate logic we get the following definition of $\mathcal{S}$-elementary equivalence.

Definition 13. Let $\mathcal{S}$ be a pre-semantical system over the language $\langle \mathcal{C}, P\rangle$. Let $\mathfrak{M}$ and $\mathfrak{N}$ be models of $\mathcal{S}$. $\mathfrak{M}$ and $\mathfrak{N}$ are $\mathcal{S}$-*elementarily equivalent*, in symbols $\mathfrak{M} \equiv_{\mathcal{S}} \mathfrak{N}$, if they have the same theory, i.e., $\mathrm{Th}\,\mathfrak{M} = \mathrm{Th}\,\mathfrak{N}$.

The *truth filter* of $\mathfrak{M}$, in symbols $F_{\mathfrak{M}}$, is the set of meanings of sentences in the theory of $\mathfrak{M}$, i.e., $F_{\mathfrak{M}} = (\mathrm{mng}^{\mathfrak{M}})(\mathrm{Th}\,\mathfrak{M}) \subseteq \mathrm{Me}\,\mathfrak{M}$. We combine these two data to obtain the *meaning matrix of* $\mathfrak{M}$: $\mathcal{ME}\ \mathfrak{M} = \langle \mathbf{Me}\,\mathfrak{M}, F_{\mathfrak{M}}\rangle$.

Every semantical system $\mathcal{S}$ defines a generally loosely interpreted language whose admissible interpretations are the meaning matrices of $\mathcal{S}$.

Theorem 14. Let $\mathcal{S}$ be a semantical system over the language type $\langle \mathcal{C}, P\rangle$. For all $\Gamma \cup \{\varphi\} \subseteq \mathrm{Se}(P)$, $\Gamma \models_{\mathcal{S}} \varphi$ iff φ is a consequence of Γ in the loosely interpreted language defined by $\mathcal{S}$.

The admissible interpretations associated with a semantical system are not in general intensional, in contrast to the case for rule-based logics. Important information can be lost in passing from a meaning matrix to its intensional reduction.[3]

Two meaning matrices $\mathcal{ME}\ \mathfrak{M}$ and $\mathcal{ME}\ \mathfrak{N}$ are *isomorphic* if there is an isomorphism $h\colon \mathbf{Me}\,\mathfrak{M} \cong \mathbf{Me}\,\mathfrak{N}$ of the underlying algebras that preserves truth filters in the sense that $h(F_{\mathfrak{M}}) = F_{\mathfrak{N}}$. The meaning matrix captures essentially all the abstract structure of $\mathfrak{M}$ inherent in the semantical system in the sense that, if the meaning matrices of two models $\mathfrak{M}$ and $\mathfrak{N}$ of $\mathcal{S}$ are isomorphic, then $\mathfrak{M}$ and $\mathfrak{N}$ must be $\mathcal{S}$-elementarily equivalent. To see this we simply observe that, if h is a isomorphism between the meaning matrices $\mathcal{ME}\ \mathfrak{M}$ and $\mathcal{ME}\ \mathfrak{N}$, then for every $\varphi \in \mathrm{Se}(P)$, $\varphi \in \mathrm{Th}\,\mathfrak{M}$ iff $\mathrm{mng}^{\mathfrak{M}}(\varphi) \in F_{\mathfrak{M}}$ iff $h\big(\mathrm{mng}^{\mathfrak{M}}(\varphi)\big) \in F_{\mathfrak{N}}$ iff $\mathrm{mng}^{\mathfrak{N}}(\varphi) \in F_{\mathfrak{N}}$ (by the initiality of $\mathbf{Se}(P)$) iff $\varphi \in \mathrm{Th}\,\mathfrak{N}$. The converse does not hold in general. For this purpose a further abstraction is required, namely passage to the intensional reduction. The following algebraic characterization of $\mathcal{S}$-elementary equivalence is one of the main results of the elementary theory of abstract algebraic logic.

Theorem 15. Let $\mathcal{S}$ be a semantical system over the language type $\langle \mathcal{C}, P\rangle$. Then two models $\mathfrak{M}$ and $\mathfrak{N}$ of $\mathcal{S}$ are $\mathcal{S}$-elementarily equivalent iff their reduced meaning matrices are isomorphic, i.e., there exists a (unique) isomorphism $h\colon \mathbf{Me}^*\,\mathfrak{M} \cong \mathbf{Me}^*\,\mathfrak{N}$ of the reduced meaning algebras such that $h^\star F^*_{\mathfrak{M}} = F^*_{\mathfrak{N}}$.

Classical propositional logic, first-order predicate logic, and modal logic are just some of the logical system that can be formalized in a natural way as both rule-based and semantic-based interpreted languages. The modal logic is an example of a special class of semantic-based interpreted languages that arises as a abstraction of Kripke's "possible world" semantics.

Bibliography

[1] H. Andréka and I. Németi. General algebraic logic: A perspective on "what is logic". In [9], chapter 15, pages 485–569.

[2] H. Andréka, I. Németi, I. Sain, and Á Kurucz. General algebraic logic. In L. Csirmaz and D. Gabbay, editors, *Proc. Logic Coll'92, Veszprém, Hungary*, Studies in Logic, Language and Computation, pages 1–60. CSLI Publications, 1995.

[3] H. Andréka, J. van Benthem, and I. Németi. Back and forth between modal logic and classical logic. *J. of the IGPL*, 3(5):685–720, 1995.

[4] W. J. Blok and Don Pigozzi. *Algebraizable logics.* Number 396 in Mem. Amer. Math. Soc. Amer. Math. Soc., 1989.

[5] Rudolf Carnap. *Meaning and Necessity. A Study in Semantics and Modal Logic.* The University of Chicago Press, Chicago and London, 1956.

[6] Alonzo Church. *Introduction to Mathematical Logic*, volume 1. Princeton University Press, Princeton, New Jersey, 1956.

[7] J. Fiadeiro and A. Sernadas. Structuring theories on consequence. In D. Sannella and A. Tarlecki, editors, *Recent Treds in Data Type Specifications*, volume 332 of *Lecture Notes in Computer Science*, pages 44–72. Springer-Verlag, New York, 1988.

[8] J. M. Font and R. Jansana. On the sentential logics associated with strongly nice and semi-nice general logics. *Bull. Interest Group in Pure and Applied Logic*, 2:55–76, 1994.

[9] D. M. Gabbay, editor. *What is a logical system?*, volume 4 of *Studies in Logic and Computation.* Clarendon Press, Oxford, 1994.

[3] This is an important feature of semantic-based languages. The theory of semantic-based interpretations originated in [1, 2]. Rule- and semantic-based abstract algebraic logic are compared in [8].

[10] J. Goguen and R. Burstall. Introducing institutions: abstract model theory for specification and programming. *J. Assoc. Comput. Mach.*, 39:95–146, 1992.

[11] L. Henkin, J. D. Monk, and A. Tarski. *Cylindric Algebras. Part II.* North-Holland, Amsterdam, 1985.

[12] J. D. Monk. Substitutionless predicate logic with equality. *Arch. Math. Logik Grundlagenforsch.*, 7:102–121, 1965.

[13] A. Sernadas, C. Sernadas, and C. Caleiro. Fibring of logics as a categorical construction. *J. Logic Computat.*, 9(2):149–179, 1999.

[14] Y. Venema. Cylindric modal logic. *J. of Symbolic Logic*, 60:591–623, 1995.

[15] A. Zanardo, A. Sernadas, and C. Sernadas. Fibring: completeness preservation. *J. Symbolic Logic*, 66(1):414–439, March 2001.

Logics of Imperfect Information

Gabriel Sandu

Department of Philosophy, University of Helsinki, Finland

Abstract

The paper contains a survey of results and interpretations of incomplete information in predicate and modal logics.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Extensive games

It is customary to present games in classical game theory in *extensive form*. This is done by first fixing a set of actions A which represents the set of possible choices of the players in the game and then define an *extensive game* G_A of perfect information is a tuple

$$G_A = (N, H, Z, (I_i)_{i\in N}, P, (u_i)_{i\in N})$$

such that

(i) N is the set of players of the game;

(ii) H is a set of sequences of actions from A, which are called *histories*, or *plays* of the game. We require that:

(a) If $h \in H$, then any initial segment of h is in H too;

(b) There is a history h_0, the root of the game, which is an initial segment of every $h \in H$;

(iii) Z is the set of maximal histories of the game;

(iv) Each I_i is the information set of player i, which in the case of games of perfect information is a singleton.

(v) $P : H \setminus Z \to N$ is the player function which assigns to every non-terminal history the player whose turn is to move;

(vi) each u_i is the payoff function for player $i \in N$, that is, a function which specifies for each maximal history in Z what is the payoff for player i.

From the class of extensive games of perfect information, we single out a particular subclass, which is the class of z*ero-sum* (*win-loss*) games. These are games played by two players, that is, $\exists$ and $\forall$ and are defined in the standard way.

For any nonterminal history $h \in H$ we let $A(h)$ be the set of actions available to a player at the history h, i.e. $A(h) = \{x \in A : h \frown x \in H\}$, and $P^{-1}(\{i\})$ be the set of histories of H where it is player i's turn to move, as specified by the function P. A *strategy* for a player i is usually defined as any function

$$f_i : P^{-1}(\{i\}) \to A$$

such that $f_i(h) \in A(h)$. In other words, a strategy for a player in the game yields exactly one choice for any position where the player has to move.

An old result due to Zermelo is that every finite (i.e. game in which all the histories have finite length) extensive zero-sum game is determined: either $\exists$ or $\forall$ has a winning strategy in the game.

2 Strategies as plans of action

It is natural to introduce a more operative notion of strategy as a *plan of action*: only the *histories reached* in the game using the strategy matter; other counterfactual situations are ignored. We disregard the information sets of the players for the present case.

Fix a game $G_A = (\{\exists, \forall\}, H, Z, (I_i)_{i\in\{\exists,\forall\}}, P, (u_i)_{i\in\{\exists,\forall\}})$. A plan of action for a player in the game is a set of histories which contains the initial history, is closed with respect to the moves of the opponent, and yields, for every position reached by using the strategy, at least one possibility to continue the game.

More precisely, a *plan of action* for $\exists$ in the game G_A is a set $S_\exists \subseteq H$ which satisfies the following conditions:

(a) $h_0 \in S_\exists$.

For every nonterminal history h:

(b) If $h \in S_\exists$, and $P(h) = \forall$, then $h \frown x \in S_\exists$ for every $x \in A(h)$.

(c) If $h \in S_\exists$, and $P(h) = \exists$, then $h \frown x \in S_\exists$ for at least one $x \in A(h)$.

(d) The plan of action $S_\exists$ is a *winning* one, if in addition, for every maximal $h \in S_\exists \cap Z : u_\exists(h) = 1$.

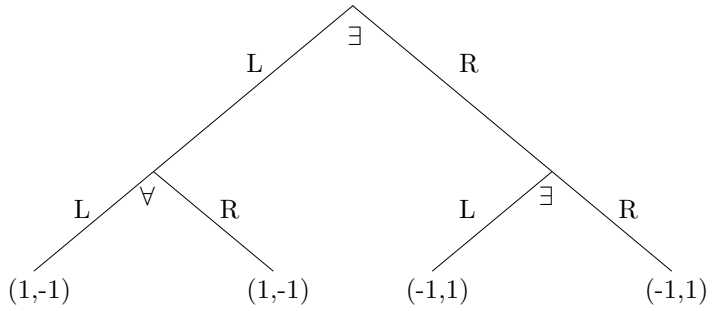
A plan of action for $\forall$ is defined in the same way, excep that '$\exists$' is replaced by '$\forall$' and in clause (d) we have: $u_\forall(h) = -1$.

A plan of action as defined above is in the general case *non-deterministic* in that it need not *functionally* pick up the choice to be made by the player in question. If we want it to be a function, then we need the following qualification. A set $S_\exists \subseteq H$ is a *deterministic plan of action* for $\exists$ if there is a function

$$f_\exists : P^{-1}(\{\exists\}) \cap S_\exists \to A$$

such that $S_\exists$ satisfies conditions (a)-(b) above and in addition condition (c) is replaced by (c') If $h \in S_\exists$, and $P(h) = \exists$, then $h \frown f_\exists(h) \in S_\exists$. A deterministic plan of action for $\forall$ is defined analogously.

Notice that in the example below, $\exists$ has a winning strategy as a plan of action but not in the traditional sense:



The deterministic winning strategy is $f_\exists(\varnothing) = L$.

3 Semantical games of perfect information

When the set of actions A consists of the set of the subformulas $Subf(\varphi)$ of a proposition sentence φ in negation normal form (otherwise negation is treated as role swapping) and a model M of the language of φ is fixed in such a way that

- Each history $h \in H$ is a sequence of subformulas of φ;
- The root of the tree is φ;
- When the last member of a history h is $\psi \wedge \theta$ ($\psi \vee \theta$), then both $h \frown (\psi)$ and $h \frown (\theta)$ belong to H and the corresponding move is by $\forall$ ($\exists$);
- A maximal history is a win for $\exists$ if its last member is true in M; otherwise it is a win for $\forall$

then we have a semantical game $G_{\varphi;M}$ for propositional logic. Game-theoretical truth and falsity in M are defined in an obvious way:

$$\begin{array}{lcl} M \vDash^{+}_{GT} \varphi & \iff & \text{there is a winning plan of action for } \exists \text{ in } G_{\varphi;M} \\ M \vDash^{-}_{GT} \varphi & \iff & \text{there is a winning plan of action for } \forall \text{ in } G_{\varphi;M}. \end{array}$$

A fairly simple proof shows that game theoretical truth coincide with Tarskian truth, and game-theoretical falsity coincides with Tarskian falsity.

When the formula φ is a formula of predicate logic such that the players choose not only conjunctions and disjunctions but also elements from the universe of the model corresponding to the universal and existential quantifiers of the formula, then we have a semantical game $G_{\varphi,g;M}$ where g is a partial assignment to the free variables of φ ($\varnothing$ is the empty assignment). In this case the root of the tree is (φ, g) and each nonmaximal history h is extended either with

$$(\chi, g), \chi \in \{\psi, \theta\},$$

or with

$$(\chi, g \cup \{(x_i, a)\}),\ a \in dom(M).$$

The details are straightforward.

Game-theoretical truth and game theoretical falsity are defined in analogy with the propositional case. It turns out that, if the *Axiom of Choice* is assumed, the existence of winning deterministic plans of action for $\exists$ and $\forall$ coincides with Tarskian truth and falsity, respectively.

When a formula is in prenex normal form, then any deterministic winning plan of action for $\exists$ is decomposable into *Skolem functions*; and if it is false, then any deterministic winning plan of action for player $\forall$ is decomposable into *Kreisel's counterexamples*.

4 Semantical games of imperfect information

The *extensive game* G_A of *imperfect information* is a tuple

$$G_A = (\{\exists, \forall\}, H, Z, P, (I_i)_{i\in\{\exists,\forall\}}, (u_i)_{i\in\{\exists,\forall\}})$$

where all the sets are as before, except for the information sets $(I_i)_{i\in\{\exists,\forall\}}$which are not any longer singletons. Each I_i is a partition of the set of histories where player i is to move, that is, a partition of the set $\{h \in H : P(h) = i\}$. The histories h, h' are *equivalent for player* i, $h \sim_i h'$, if there is $S \in I_i$ such that $h, h' \in S$.

There are usually two requirements on equivalent histories.

The Consistency condition (to equivalent histories there should correspond indistinguishable futures)

$$\text{For every } h, h' \in H, \text{ and } i \in N: \ h \sim_i h' \Rightarrow A(h) = A(h').$$

The *von Neumann & Morgenstern condition* (*to equivalent histories there should correspond indistinguishable pasts*)

$$\text{For every } h, h' \in H, \text{ and } i \in N : h \sim_i h' \Rightarrow length(h) = length(h')$$

A deterministic plan of action for player i is defined exactly as before, except that now the function f_i is required to be *uniform*:

$$\text{For every } h, h' \in S_i\text{: If } h \sim_i h' \Longrightarrow f_i(h) = f(h').$$

Imperfect information does at least three things:

- It introduces *indeterminacy* in the game.
- It allows for a phenomenon known in game theory as *signalling*.
- It introduces, in combination with contradictory negation, *paraconsistency* in the logic.

I shall discuss some examples in the full paper ([2]).

4.1 Imperfect information in predicate logic

We consider extensions of first-order logic with formulas like

$$\forall x_0(\exists x_1/\{x_0\})\varphi$$
$$\forall x_0 \exists x_1 \forall x_2(\exists x_3/\{x_0, x_1\})\psi$$

where φ and ψ are standard first-order formulas. The idea is that in the extensive form of the corresponding games (played on a fixed model M), any two histories

$$\langle\langle \forall x_0(\exists x_1/\{x_0\})\varphi, \varnothing\rangle, \langle(\exists x_1/\{x_0\})\varphi, \{(x_0, a)\}\rangle\rangle$$

and

$$\langle\langle \forall x_0(\exists x_1/\{x_0\})\varphi, \varnothing\rangle, \langle(\exists x_1/\{x_0\})\varphi, \{(x_0, b)\}\rangle\rangle$$

from the first game are indistinguishable for $\exists$. In the second example, any two histories (we omit the left side formulas and the root of the tree)

$$\langle\{(x_0, a)\}, \{(x_1, b)\}, \{(x_2, c)\}\rangle$$

and

$$\langle\{(x_0, a')\}, \{(x_1, b')\}, \{(x_2, c)\}\rangle$$

are indistinguishable for $\exists$. The idea in the general case

$$Qx_0Qx_2...Qx_{n-1}(Qx_n/W)\chi$$

with $W \subseteq \{x_1, ..., x_{n-1}\}$ is that the slash '/' introduces an equivalence relation on the set of histories of the relevant game and thus any two histories

$$\langle\{(x_0, a_0)\}, \{(x_1, a_1)\}, ..., \{(x_{n-1}, a_{n-1})\}\rangle$$

and

$$\langle\{(x_0, b_0)\}, \{(x_1, b_1)\}, ..., \{(x_{n-1}, b_{n-1})\}\rangle$$

are equivalent for player i exactly when

$$\text{For each } i \in (\{x_1, ..., x_{n-1}\} - W) : a_i = b_i.$$

The resulting logic (IF-predicate logic) is known to have the following properties:

1. **Effective disproof procedure**: The set of IF-contradictions is recursively axiomatizable.

2. **Compactness**: A set Γ of $IF-$sentences has a model if and only if every finite subset of Γ has a model.

3. **Interpolation property**. Let K_1 and K_2 be two class of structures definable by $IF-$sentences. If K_1 and K_2 are disjoint, then there is class of models K definable by a first-order sentence such that

$$K_1 \subseteq K \text{ and } K \cap K_2 = \varnothing.$$

4. **Expressive power**. Σ^1_1-logic has greater expressive power than standard first-order logic. For instance, the property of being a non-standard number, an infinite set. etc are definable by IF-sentences.

5. **Definability of truth**: $IF-$logic defines its own truth-predicate, in the sense that there is a $IF-$formula $\Phi(x)$, such that for every model M of PA, and every $IF-$sentence φ in the similarity type of PA:

$$M \vDash \Phi(\ulcorner\varphi\urcorner) \Leftrightarrow M \vDash \varphi.$$

I will discuss some of the foundational issues in the full paper ([1]).

4.2 Informational independence and restricted quantifiers

This kind of informational independence arises in typically two cases.

1. With restricted quantifiers in predicate logic, that is, with quantifiers of the form '$\exists x : R(x)$' and '$\forall x : Q(x)$' where '$R(x)$' and '$Q(x)$' are relations in the relevant model. In this case we shall have formulas of the form

$$(\forall x_0 : R_0(x_0))(\exists x_1 : R(x_1)/\{x_0\})\varphi$$
$$(\forall x_0 : R_0(x_0))(\exists x_1 : R(x_1))(\forall x_2 : R(x_2))(\exists x_3 : R(x_3)/\{x_0, x_1\})\psi$$

The games are exactly like before, the players' choosing individuals from the relevant model which belong to the extension of the appropriate relation R_i (when there is no legal move, the opponent wins right away). The assumption of imperfect information is implemented, as above, as the requirement of uniformity over equivalent strategies. The *Consistency Condition* or the *von Neumann & Morgenstern Condition* may be violated in this case. I shall discuss few examples in the full paper.

2. With modal operators which can be regarded as restricted quantifiers over accessibility relations. In the syntax, we add to standard modal logic formulas of the form

$$Q_1Q_2...Q_{n-1}(Q_n/W)\varphi$$

where each Q_i is one of the standard modal operators $\Box$ or $\Diamond$ and $W \subseteq \{1, ..., n-1\}$. Models have the form $M = (W, R, V)$. Games now will be played starting with a possible world w_0 and the players will choose possible worlds along the accessibility relation R (the play stops right away if a player cannot

continue with a legal move, with the other player winning the play right away). As in the previous case, the idea here is that any two histories

$$\langle w_0, w_1, ..., w_{n-1} \rangle$$

and

$$\langle w_0, w_1^{'}, ..., w_{n-1}^{'} \rangle$$

are equivalent for the relevant player exactly when

$$\text{For all } i \in (\{0, ..., n-1\} - W) : w_i = w_i^{'}.$$

It is known ([3]) that the resulting logic (IF-modal logic) is strictly stronger than standard modal logic.

In the full paper I will discuss some other interpretations of imperfect information than the uniformity of plans of action interpretation.

Bibliography

[1] Gabriel Sandu and Tapani Hyttinen. If logic and the foundations of mathematics. *Synthese*, pages 37–47, 2001.

[2] Gabriel Sandu and Ahti Pietarinen. Informationally independent connectives. In G. Mints and R. Muskens, editors, *Games, logic and constructive sets*, pages 23–41. CSLI publishers, Stanford, 2003.

[3] Tero Tulenheimo. Informationally independent connectives. In P. Balbiani et al., editor, *Advances in Modal Logic*, volume 4, pages 475–498. King's College Publications, London, 2003.

Software Specification and Development in Heterogeneous Environments

Andrzej Tarlecki

Institute of Informatics, Warsaw University and
Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

Nowadays, software tends to be enormously complex. This truth has a number of consequences for the methodology and practice of software development, and should be adequately reflected by the theories that capture and support program development process. In a typical program a number of facets coexist, giving rise to a number of its possible and useful views. Specification of a complex program necessarily involves presentation, often separate presentation, of many aspects of its behaviour; practically useful methodologies must support this. A successful example here is UML, where a good dozen of various kinds of diagrams are used. Diagrams of each kind present a different program view, and in fact (leaving aside any doubts about the formal underpinnings) offer a formalism to deal with one particular kind of program properties. None of these individual views captures all the aspects of the program in question, and only considering them together may lead to an adequate overall view.

Abstracting away from many details: to adequately describe a program we need to use a number of logical formalisms, each targeted at a special kind of program properties. Indeed, the evident proliferation of logical systems used in computer science in general, and in software specification and development in particular, is not just theoreticians' fancy, but a necessary consequence of the practical needs to capture various aspects of software, as well as dealing with various programming languages and paradigms.

Consequently, what we are after is not a single "best" framework to cater for all needs and capture all possible properties and kinds of software, but a truly *heterogeneous environment*, where a number of formalisms may coexist and complement each other. In such an environment, it should be possible to vary formalisms in use to deal with various aspects of software, meaningfully interpret resulting specifications based on a number of logical formalisms, use various logical (and programming) means to deal with different components of a single program, and switch between logical formalisms when progressing with program specification and development. Such an environment should also be *open*, meaning that a well-prescribed amount of work should be needed to add a new logical formalisms to it and make it available for its users. Last but not least, it should come with a good support system, facilitating the use of each individual formalisms incorporated in the environment, as well as the possibility to switch between these formalisms in the course of building specifications and then developing programs.

To achieve such an ambitious and far-fetching vision a good deal of foundational work is necessary, to precisely explicate the concepts and ideas involved, and to identify the potential problems and provide a solid basis to resolve them satisfactorily.

The first problem is a formalisation of the very concept of logical system as used here. A number of approaches are possible and provide potentially useful answers, including the rather popular in the theorem-proving community views of logics as captured in so-called *general logical frameworks* developed within appropriately rich type theories (with Edinburgh LF being perhaps the prime example [15]). We follow here a more semantic, model-theoretic view, very much in the spirit of abstract model theory [1] and the tradition of algebraic specification [11], where the notion of a logical system has been usefully formalised as an *institution* [13, 28]. The framework of institutions provides a solid basis here for sketching some key issues related to the development of a heterogeneous environment.

Some standard concepts of the theory of institutions are recalled in Sect. 2, covering the definition of an institution as well as notions that capture various ways of relating one institution with another [27, 14]. These are used in Sect. 3 to present structured heterogeneous specifications, built over a number of institutions, following [29]. Section 4 then shows how an abstract view of software development [24, 30] may incorporate the possibility of (and need for) switching from one institution to another, which leads to the discussion of *heterogeneous design*, where various components of the same program might be specified and developed using different formalisms, thus resulting in a heterogeneous version of *architectural specification* [3]. Section 5 offers some concluding remarks. Necessarily, the presentation here is very sketchy, trying to concentrate more on the problems and ideas than on precise answers, with statements of any technical results (and explicit examples!) being sacrificed for the sake of brevity — a more detailed and complete presentation will be given in a forthcoming full version of this paper.

At least two projects have been successfully undertaken in a similar vein of exploiting the theory of institutions to explicate and support practical use of multiple logical systems is software specification. Perhaps the first such project leading to a practical system was CafeOBJ based on a cube of logics formalised as institutions with institution morphisms between them [10]. The other one arose within the CoFI working group, with activities centered around Casl, an up-to-date algebraic specification formalism [2, 8]. This work used the concepts of institution theory to structure and facilitate the design

and formal description of CASL, to compare it with other specification formalisms, and to design its extensions [18, 8]. One of the recent CASL extensions is to provide a formalism to build heterogeneous specifications, with tool support offered by HETS, Heterogeneous Tool Set, and formal underpinnings developed in [17, 19, 20].

2 Institutional preliminaries

An *institution* [13] consists of a category $\mathbf{Sign}$ of *signatures*, functors $\mathbf{Sen}\colon \mathbf{Sign} \to \mathbf{Set}$ and $\mathbf{Mod}\colon \mathbf{Sign}^{op} \to \mathbf{Cat}$, and for each $\Sigma \in |\mathbf{Sign}|$, a Σ-*satisfaction relation* $\models_\Sigma \subseteq |\mathbf{Mod}(\Sigma)| \times \mathbf{Sen}(\Sigma)$. For $\Sigma \in |\mathbf{Sign}|$, $\mathbf{Sen}(\Sigma)$ is the set of Σ-*sentences*, and $\mathbf{Mod}(\Sigma)$ is the category of Σ-*models* and their morphisms. For any *signature morphism* $\sigma\colon \Sigma \to \Sigma'$ in $\mathbf{Sign}$, $\mathbf{Sen}(\sigma)$, written as $\sigma\colon \mathbf{Sen}(\Sigma) \to \mathbf{Sen}(\Sigma')$, is a translation of sentences, and $\mathbf{Mod}(\sigma)$, written as $\_|_\sigma\colon \mathbf{Mod}(\Sigma') \to \mathbf{Mod}(\Sigma)$ is a *reduct functor*. These are subject to the following *satisfaction condition*:

$$M'|_\sigma \models_\Sigma \varphi \iff M' \models_{\Sigma'} \sigma(\varphi)$$

where $\sigma\colon \Sigma \to \Sigma'$ in $\mathbf{Sign}$, $M' \in |\mathbf{Mod}(\Sigma')|$, $\varphi \in \mathbf{Sen}(\Sigma)$.

The requirements this definition imposes on a logical system are very mild, and examples of logical systems presented as institutions abound, including many typical logics such as various versions of equational and first-order logics (perhaps with partial operations, predicate symbols, error elements, subsorting, additional forms of higher-order constraints, etc). Just to hint that much more is covered, let us mention that modal logics, many-valued logics, and even programming language semantics may be formalized as institutions.

For any signature $\Sigma \in |\mathbf{Sign}|$, the satisfaction relation between Σ-models and Σ-sentences allows for reuse of standard logical concepts and results in an arbitrary institution. In particular, the definitions of the class of *models*[1] $Mod[\Phi] \subseteq |\mathbf{Mod}(\Sigma)|$ of a set of sentences $\Phi \subseteq \mathbf{Sen}(\Sigma)$, and of *semantic consequence* $\Phi \models_\Sigma \varphi$, for $\Phi \subseteq \mathbf{Sen}(\Sigma)$ and $\varphi \in \mathbf{Sen}(\Sigma)$, carry over without change.

In applications the institutions may be subject to further restrictions. For instance, in work on various forms of "putting together" signatures, theories and specifications [6], the category of signatures is assumed to be cocomplete, or at least to have pushouts, and things work smoothly if the amalgamation property holds over colimits of signatures (equivalently: the model functor is continuous, and so compatible model families over signature diagrams can be amalgamated to a model over the diagram colimit). Standard meta-logical properties may be abstractly formulated for an arbitrary institution, with various forms of interpolation over signature pushouts [26] being one important example. Moreover, one often assumes that the institutions of interest come with some additional structure; for instance, an additional "proof-theoretic" counterpart of the consequence relations may be required, captured by entailment relations $\vdash_\Sigma \subseteq \wp(\mathbf{Sen}(\Sigma)) \times \mathbf{Sen}(\Sigma)$, for $\Sigma \in |\mathbf{Sign}|$, which leads to the concept of a *general logic* [16] when each $\vdash_\Sigma$ is sound for the semantic consequence $\models_\Sigma$.

Given the formalisation of the concept of a logical system as an institution, much interesting work has been done to free the theory of algebraic specification and software development as originally developed for equational logic [11] from the limitations of using the equational logic only, leading to an abstract specification theory in an arbitrary but fixed institution, [24, 30].

We undertake now the next step: dealing with a number of institutions at the same time. Naturally, to make any sense of this, the institutions involved must be somehow linked with each other. A number of concepts of a formal mapping between institutions have been proposed, starting with the original *institution morphisms* in [13]. There is an obvious "taxonomy" for such mappings, based on the relative direction of translation of the three institution components (signatures, models, sentences), as perhaps first pointed out in [27] and then exploited to systematize the field and put forward key dualities in [14], see also [19]. Two such concepts seem of the crucial importance for our purposes.

Consider two institutions $\mathbf{I} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$ and $\mathbf{I}' = \langle \mathbf{Sign}', \mathbf{Sen}', \mathbf{Mod}', \langle \models'_{\Sigma'} \rangle_{\Sigma' \in |\mathbf{Sign}'|} \rangle$. An *institution morphism* $\mu\colon \mathbf{I}' \to \mathbf{I}$ consists of three components: a functor $\mu^{Sign}\colon \mathbf{Sign}' \to \mathbf{Sign}$ and two natural transformations, $\mu^{Mod}\colon \mathbf{Mod}' \to (\mu^{Sign})^{op};\mathbf{Mod}$ and $\mu^{Sen}\colon \mu^{Sign};\mathbf{Sen} \to \mathbf{Sen}'$, subject to the *satisfaction condition*:

$$M' \models'_{\Sigma'} \mu^{Sen}_{\Sigma'}(\varphi) \iff \mu^{Mod}_{\Sigma'}(M') \models_{\mu^{Sign}(\Sigma')} \varphi$$

[1] We apologize for the traditional overloading of the term "model" here.

where $\Sigma' \in |\mathbf{Sign}'|$, $M' \in |\mathbf{Mod}'(\Sigma')$, $\varphi \in \mathbf{Sen}(\mu^{Sign}(\Sigma'))$. Disregarding the sentence component (and so the satisfaction condition as well) yields an institution *semi-morphism*.

An *institution comorphism* $\rho\colon \mathbf{I} \to \mathbf{I}'$ consists of three components: a functor $\mu^{Sign}\colon \mathbf{Sign} \to \mathbf{Sign}'$ and two natural transformations, $\mu^{Mod}\colon (\mu^{Sign})^{op};\mathbf{Mod}' \to \mathbf{Mod}$ and $\rho^{Sen}\colon \mathbf{Sen} \to \mu^{Sign};\mathbf{Sen}'$, subject to the *satisfaction condition*:

$$M' \models'_{\rho^{Sign}(\Sigma)} \rho^{Sen}_{\Sigma}(\varphi) \iff \mu^{Mod}_{\Sigma}(M') \models_{\sigma} \varphi$$

where $\Sigma \in |\mathbf{Sign}|$, $M' \in |\mathbf{Mod}'(\rho^{Sign}(\Sigma))$, $\varphi \in \mathbf{Sen}(\Sigma)$. Disregarding the sentence component (and so the satisfaction condition as well) yields an institution *semi-comorphism*.

The rough and highly informal idea is that an institution morphism as above captures a way in which "richer" institution $\mathbf{I}'$ is built over more "poor" institution $\mathbf{I}$, by "extracting" simpler $\mathbf{I}$-signatures and models out of more complex $\mathbf{I}'$-signatures and models, and encoding $\mathbf{I}$-sentences as $\mathbf{I}'$-sentences. In turn, an institution comorphism as above captures a way in which more "poor" institution $\mathbf{I}$ is represented within "richer" institution $\mathbf{I}'$, by mapping simpler $\mathbf{I}$-signatures to some $\mathbf{I}'$-signatures so that $\mathbf{I}$-models can be extracted out of $\mathbf{I}'$-models and $\mathbf{I}$-sentences may be represented as $\mathbf{I}'$-sentences.

In the following we will often omit super- and sub-scripts identifying the exact component of an institution (co)morphism in use. Thus, for instance, if $\Sigma' \in |\mathbf{Sign}'|$ then $\mu(\Sigma')$ stands for $\mu^{Sign}(\Sigma')$ and if $\mathcal{M}' \subseteq |\mathbf{Mod}'(\Sigma')|$ then $\mu(\mathcal{M}')$ stands for the image of $\mathcal{M}'$ under $\mu^{Mod}_{\Sigma'}$.

3 Structured heterogeneous specifications

The use of an institution $\mathbf{I}$ in software specification and development is based on the overall idea that its signatures determine the syntax of the programs to be specified, its models represent (the semantics of) the programs, and its sentences are used to specify their properties. Following this, we adopt a model-theoretic view of specifications, as put forward in [22]. The meaning of any specification SP built over $\mathbf{I}$ is given by its *signature* $Sig[SP] \in |\mathbf{Sign}|$ and a class of its *models* $Mod[SP] \subseteq |\mathbf{Mod}(Sig[SP])|$.

One crucial insight going back to [6, 7] is that specifications must be built in some well-structured fashion; another one was that specific features of the logic at hand must not clutter the presentation and understanding of specification structure. Both points were further put forward in [22] by insisting that specifications are built from the simplest lists of axioms using some well-understood *specification-building operations* designed independently of the institution one works with. Following this, we consider a few very basic means to build specifications in an arbitrary institution:

Basic specification: indicates a signature and lists axioms to capture the desirable properties. For any $\Sigma \in |\mathbf{Sign}|$ and $\Phi \subseteq \mathbf{Sen}(\Sigma)$, $\langle \Sigma, \Phi \rangle$ is a specification with $Sig[\langle \Sigma, \Phi \rangle] = \Sigma$ and $Mod[\langle \Sigma, \Phi \rangle] = Mod[\Phi]$.

Union: combines constraints imposed by various specifications. For any SP_1 and SP_2 with $Sig[SP_1] = Sig[SP_2]$, $SP_1 \cup SP_2$ is a specification with $Sig[SP_1 \cup SP_2] = Sig[SP_1]$ and $Mod[SP_1 \cup SP_2] = Mod[SP_1] \cap Mod[SP_2]$.

Translation: renames and introduces new components, following a signature morphism. For any SP and $\sigma\colon PSP \to \Sigma'$, $\sigma(SP)$ is a specification with $Sig[\sigma(SP)] = \Sigma'$ and $Mod[\sigma(SP)] = \{M' \in |\mathbf{Mod}(\Sigma')| \mid M'|_{\sigma} \in Mod[SP]\}$.

Hiding: hides auxiliary components, receding to the source of a signature morphism. For any SP' and $\sigma\colon \Sigma \to PSP'$, $SP'|_{\sigma}$ is a specification with $Sig[SP'|_{\sigma}] = \Sigma$ and $Mod[SP'|_{\sigma}] = \{M'|_{\sigma} \mid M' \in Mod[SP']\}$.

The above definition exploits the structure of specifications to determine their semantics in a *compositional* manner. Compositionality is the key to effective use of large structured specifications; for instance, the structure of specifications may be used to guide a proof search for their semantic consequences defined as expected: $SP \models \varphi$, for a specification SP and $\varphi \in \mathbf{Sen}(Sig[SP])$, if φ holds in all models of SP. The following rules form a compositional proof system for structured specifications built as above in an institution with a proof-theoretic entailment $\langle \vdash_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|}$.

$$\frac{\varphi \in \Phi}{\langle \Sigma, \Phi \rangle \vdash \varphi} \qquad \frac{SP_1 \vdash \varphi}{SP_1 \cup SP_2 \vdash \varphi} \qquad \frac{SP_2 \vdash \varphi}{SP_1 \cup SP_2 \vdash \varphi}$$

$$\frac{SP \vdash \varphi}{\sigma(SP) \vdash \sigma(\varphi)} \qquad \frac{SP' \vdash \sigma(\varphi)}{SP'\big|_\sigma \vdash \varphi} \qquad \frac{\text{for } i \in \mathcal{I}, SP \vdash \varphi_i \quad \{\varphi_i\}_{i\in\mathcal{I}} \vdash \varphi}{SP \vdash \varphi}$$

Moreover, under reasonable assumptions about the underlying institution (we need pushouts of signatures, amalgamation and — most likely to raise problems — interpolation), the above rules extend any sound and complete entailments to a sound and complete proof system for consequences of structured specifications considered, see [5].

Heterogeneity of specifications may be achieved by introducing new specification building operations to move specifications from one institution to another. Of course, the institutions involved cannot be totally unrelated; at least their signatures and models must be linked (as these are the ingredients involved in the specification semantics). We will retain the property that each specification ultimately "resides" in a specific institution, with its semantics given in terms of signatures and models of this institution, even though some of its subspecifications may similarly reside in other institutions. We refer to specifications residing in an institution $\mathbf{I}$ in this sense as $\mathbf{I}$-specifications.

Institution semi-morphisms and semi-comorphisms offer a number of possible ways of moving specifications between institutions. It seems, however, that the following two *inter-institutional* constructs are most natural and useful:

Translation: introduces new structure to specification models, following an institution semi-comorphism. For any institution semi-comorphism $\rho\colon \mathbf{I} \to \mathbf{I}'$ and $\mathbf{I}$-specification SP, $\rho(SP)$ is an $\mathbf{I}'$-specification with $Sig[\rho(SP)] = \rho(Sig[SP])$ and $Mod[\rho(SP)] = \{M' \in |\mathbf{Mod}'(\rho(Sig[SP])| \mid \rho(M') \in Mod[SP]\}$.

Hiding: hides extra structure of specification models, following an institution semi-morphism. For any institution semi-morphism $\mu\colon \mathbf{I}' \to \mathbf{I}$ and $\mathbf{I}'$-specification SP', $SP'\big|_\mu$ is an $\mathbf{I}$-specification with $Sig[SP'\big|_\mu] = \mu(Sig[SP'])$ and $Mod[SP'\big|_\mu] = \{\mu(M') \mid M' \in Mod[SP']\}$.

We deliberately reuse here the terminology for the corresponding specification-building operations within an institution, as indeed, the intuition behind inter- and intra-institutional versions of translation and hiding is quite similar.

Translation of an $\mathbf{I}$-specification receding to an indicated signature in the source of institution semi-morphism $\mu\colon \mathbf{I}' \to \mathbf{I}$, and hiding for an $\mathbf{I}'$-specification receding to an indicated signature in the source of institution semi-comorphism $\rho\colon \mathbf{I} \to \mathbf{I}'$ may be defined similarly — we omit these here though.

Given the new inter-institutional specification-building operations, one can construct structured specifications that span a diagram of institutions linked by institution semi-morphisms and semi-comorphisms. Various parts of such a specification may be build in other institutions involved, and thus capture properties that may not be even expressible in the "surface" institution. Moreover, as before, the operations indicate how to understand and work with such structured heterogeneous specifications in a compositional way.

For instance, consider the problem of extending the above compositional proof system for structured specifications to heterogeneous specifications. Evidently, there is little one can do if the sentences of the institutions the specifications involve are not related. So, for this purpose we assume that the semi-morphisms and semi-comorphisms used to move specifications between institutions extend smoothly to sentences as well, that is, are in fact institution morphisms and comorphisms, respectively. Then the following two natural rules soundly extend the original system:

$$\frac{SP \vdash^{\mathbf{I}} \varphi}{\rho(SP) \vdash \rho(\varphi)} \qquad \frac{SP' \vdash^{\mathbf{I}'} \mu(\varphi)}{SP'\big|_\mu \vdash \varphi}$$

There is another, less direct way to introduce heterogeneous specifications. Namely, given a diagram of institutions linked by (semi-)morphisms, one can construct a single institution which incorporates all the institution in the diagram and captures links between them by signature morphisms. This is given essentially by a (more elaborate version of) the Grothendieck construction to flatted indexed categories, see [5]. Moreover, similar construction may be applied to diagrams of institutions linked by institution (semi-)comorphisms [17], and further generalised to institution diagrams where links of both kinds may be used [19]. Now, any structured specification built over the resulting Grothendieck institution using the usual intra-institutional specification-building operations may be viewed as a possibly heterogeneous specification built over the considered diagram of institutions. Translation and hiding w.r.t. certain morphisms in this Grothendieck institution correspond to inter-institutional translation and hiding as

explicitly introduced above. A warning is in order: the Grothendieck construction just puts the institutions involved next to each other, leaving them with their respective models and sentences, and only providing extra links between their signatures with institution morphisms used to define the induced model reducts and sentence translations. This has very little to do with the task of properly combining the institutions involved to form a new, more complex logical system, given in the most rudimentary way as the limit of the diagram in the (complete, see [26]) category of all institutions.

4 Heterogeneous software development

Given an institution $\mathbf{I}$ with models capturing the essential semantic aspects of programs we aim at, an $\mathbf{I}$-specification SP, whether heterogeneous or not, determines the programming task to produce a program that correctly implements it, that is, a program with the semantics yielding a model of SP. Simplifying a lot, the specified task is to build a model $M \in |\mathbf{Mod}(Sig[SP])|$ such that $M \in Mod[SP]$. The key idea is that this should not be carried out in a single jump; instead, one should proceed step by step, adding gradually more and more detail and incorporating more and more design and implementation decisions.

We adopt here a view proposed in [23], where each individual *refinement step* leading from one specification to another involves an additional component, a *constructor*. Intuitively, constructors correspond to generic modules, like Standard ML *functors* [21]; semantically they are functions that map models to models. In the realm of institutions, such constructors may be given as reducts w.r.t. signature morphisms, or via various forms of definitional extensions. Mixing the two leads to a powerful idea that a constructor may be given as a reduct w.r.t. a *derived* signature morphism [4], where symbols of the source signature may be mapped to terms, perhaps involving recursion, or even simply written in a programming language over the target signature.

With this concept in mind, we say that a specification SP' is a *constructor refinement* of SP via constructor $\kappa\colon |\mathbf{Mod}(Sig[SP'])| \to |\mathbf{Mod}(Sig[SP])|$, written $SP \overset{\kappa}{\rightsquigarrow\!\!>} SP'$, whenever $\kappa(Mod[SP']) \subseteq Mod[SP]$. Development process takes the form of a chain of such constructor refinements, with requirements on individual refinement steps ensuring the overall correctness of the result [23, 24].

There is a natural way to allow such a development process to switch from one institution to another. All that is needed are inter-institutional constructors. Under the semantic view of a constructor as a function that maps models to models, any model component of an institution semi-morphism or semi-comorphism can be used. Consequently, for any institution semi-morphism $\mu\colon \mathbf{I}' \to \mathbf{I}$, $\mathbf{I}'$-specification SP' with $Sig[SP'] = \Sigma'$, and $\mathbf{I}$-specification SP with $Sig[SP] = \mu^{Sign}(\Sigma')$, SP' is a constructor refinement of SP via μ (or, to be more precise, via $\mu^{Mod}_{\Sigma'}$),

$$SP \overset{\mu}{\rightsquigarrow\!\!>} SP'$$

provided that $\mu(Mod[SP']) \subseteq Mod[SP]$. Using such an inter-institutional refinement in a development process amounts to a decision to implement the requirements as captured by the $\mathbf{I}$-specification SP by first implementing the requirements captured by the $\mathbf{I}'$-specification SP' and then extracting an implementation of the original requirements from the result. A perhaps most typical case here has already been pointed out in [23]. The institution $\mathbf{I}$, where the original requirements specification is built, might be a standard institution of some usual algebraic logical system, like the institution of CASL [8] (in fact, SP then might be a CASL specification). Then the institution $\mathbf{I}'$, with a richer structure of models, might be an institution that captures some functional programming language, like Standard ML [21], with models corresponding to Standard ML structures, sentences to pieces of Standard ML code, and satisfaction relation capturing the semantics of Standard ML. A natural institution semi-morphism might then be given, which in essence abstracts away from some details of the semantics of Standard ML and views Standard ML structures as CASL models (many-sorted partial algebras). Then an inter-institutional refinement captures formally correct implementation of a CASL specification SP by a Standard ML specification (a program) SP'.

Note that there is no hope here to extend the institution semi-morphism to an institution morphism: this would require expressing Standard ML code using CASL sentences, which cannot be expected in general. The semantic presentation above does not require this: the institution semi-morphism is all that is needed. On the other hand, this does not provide any direct means to verify the correctness of inter-institutional refinement steps, as captured by the requirement $\mu(Mod[SP']) \subseteq Mod[SP]$. This

is a separate task (here: of verifying Standard ML code w.r.t. Casl axioms and specifications) and separate calculi and tools need to be provided to carry this out.

In the above formalisation of refinement steps we have entirely disregarded the possible internal structure of requirements specification. This was on purpose: in general there is no link between the structure used to capture requirements and the structure used to implement them. Quite simply, the two serve quite different purposes [12], which is perhaps even more visible in the context of heterogeneous specification and development. Using a number of institutions to build a requirements specification often reflects various views and aspects of the same program, and has nothing to do with identification of program components in any possible implementation.

Consequently, separate tools must be provided to design and capture in the development process the structure of program implementation. In Casl, these come in the form of *architectural specifications* [3], which might look as follows:

$$\begin{aligned}\textbf{arch spec } ASP = \textbf{units } & U_1 \colon SP_1 \\ & \dots \\ & U_n \colon SP_n \\ \textbf{result } & \kappa(U_1, \dots, U_n)\end{aligned}$$

The architectural specification ASP above capture a branching point in the development process, namely a design decision to build the system by providing separate implementations for specifications SP_1, ..., SP_n, (naming the resulting modules U_1, ..., U_n, respectively) and putting them together using a multi-argument constructor κ. In Casl architectural specifications, the result constructor κ is composed of simpler constructors on models of the underlying institution built into the formalism. Perhaps the most important among them (disregarding instantiation of generic units, which we omit here) are reducts w.r.t. a signature morphism and amalgamation of models to the union of their signatures. While reducts are quite straightforward and always safe, amalgamation causes extra problems, since not every two models can be amalgamated [25].

Heterogeneity may be added to architectural specifications simply by extending the repertoire of basic constructors by some inter-institutional constructors, as discussed above. These can be model translations given by institution semi-morphisms, with typical examples sketched above. This opens the possibility for component specifications SP_1, ..., SP_n to be given in various institutions, and their implementations to be developed using quite different programming paradigms. The resulting models, capturing implementations of individual system modules, can be translated to a common, typically more abstract framework (an institution) and combined there as required.

5 Final remarks

The aim of this note is to present some preliminary thoughts on possible ways of using multiple logical formalisms as well as programming languages in course of program specification and development. What emerges is a formal semantic view of a heterogeneous logical environment as a diagram of institutions linked by institution (semi-)morphisms and (semi-)comorphisms, which captures both, logical systems used to build requirements specifications and programming languages and paradigms used for implementation as well as their mutual semantic relationships. Given this, we sketch how heterogeneous specifications can be built and how heterogeneous programs may emerge, with various logics and programming formalisms used at subsequent stages of program development and for various components of the program. This semantic view seems perfectly adequate to provide guidelines for the real work yet to come.

The first task is to choose the logical and programming formalisms to be offered, formalise them as institutions and develop (co)morphisms to link some of them. Providing such links is never an easy task: we often look for relationships between some models and concepts developed to capture radically different views and intuitions concerning software. In a way, relating various semantic views of programs is often the most difficult task here.

Once this is given, there are further problems to solve. For instance, when discussing structured heterogeneous specifications, we presented a proof system for showing their consequences, which works in a satisfactory way under some technical assumptions. When those are not satisfied though (for instance, because intra- or inter-institutional interpolation property fails) completeness of the compositional proof

system must fail, and some non-compositional techniques may have to be used. Then, lifting this proof system to refinements between specifications requires an additional oracle for conservativity of extensions, perhaps even more troublesome in heterogeneous environments.

Our heterogeneous specifications ultimately reside in a specific institution, with other institutions in use playing in a sense an auxiliary role only. An interesting alternative to consider is to deal with diagrams of specifications distributed over the heterogeneous logical environment, understood as specifying a single program from a number of various perspectives. Issues of consistency and of "emerging features" seem to becoming even more important then.

Finally, the ease with which we treat heterogeneous development process is highly suspicious: we have in fact disregarded the true computational contents of implementations. While extracting an abstract model out of a program coded in a specific programming language is merely the task for the programming language semantics, combining two such models extracted from programs coded in possibly quite different programming languages, and especially resolving mutual dependencies between them in a computationally meaningful way cannot be restricted to semantic manipulation only, and must require some extra programming work to provide an interface between the languages involved.

Acknowledgments

The ideas sketched here have roots in joint work with Don Sannella on numerous issues related to abstract specification theory, with Michel Bidoit on architectural specifications, and with Till Mossakowski on various aspects of institution theory, including heterogeneity. Thanks also to Andrzej Gąsienica-Samek for "pączki" and "kodki" which made me wonder about heterogeneous programming from a less semantic perspective. This work has been partially supported by KBN grant 7T11C 002 21 and European AGILE project IST-2001-32747.

Bibliography

[1] J. Barwise. Axioms for abstract model theory. *Annals of Mathematical Logic*, 7:221–265, 1974.

[2] M. Bidoit and P. D. Mosses. CASL *User Manual*. LNCS 2900. Springer, 2004. With chapters by T. Mossakowski, D. Sannella, and A. Tarlecki.

[3] M. Bidoit, D. Sannella, and A. Tarlecki. Architectural specifications in CASL. *Formal Aspects of Comput.*, 13:252–273, 2002.

[4] M. Bidoit, D. Sannella, and A. Tarlecki. Global development via local observational construction steps. In *Proc. 27th Intl. Symp. Mathematical Foundations of Computer Science*, LNCS 2420, pages 1–24. Springer, 2002.

[5] T. Borzyszkowski. Logical systems for structured specifications. *Theoretical Comput. Sci.*, 286:197–245, 2002.

[6] R. Burstall and J. Goguen. Putting theories together to make specifications. In *Proc. 5th Intl. Joint Conference on Artificial Intelligence, Cambridge, Mass. (USA)*, pages 1045–1058, 1977.

[7] R. Burstall and J. Goguen. The semantics of CLEAR, a specification language. In *Proc. Copenhagen Winter School on Abstract Software Specification*, LNCS 86, pages 292–332. Springer, 1980.

[8] CoFI (The Common Framework Initiative). CASL *Reference Manual*. LNCS 2960 (IFIP Series). Springer, 2004.

[9] R. Diaconescu. Grothendieck institutions. *Applied Categorical Structures*, 10:383–402, 2002.

[10] R. Diaconescu and K. Futatsugi. Logical foundations of CAFEOBJ. *Theoretical Comput. Sci.*, 285:289–318, 2002.

[11] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1: Equations and Initial Semantics*. Springer, 1985.

[12] J. Fitzgerald and C. Jones. Modularizing the formal description of a database system. In *Proc. 3rd Intl. Symp. VDM Europe: VDM and Z, Formal Methods in Software Development*, LNCS 428, pages 189–210. Springer, 1990.

[13] J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *J. ACM*, 39:95–146, 1992.

[14] J. Goguen and G. Rosu. Institution morphisms. *Formal Aspects of Comput.*, 13:274–307, 2002.

[15] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *J. ACM*, 40:143–184, 1993.

[16] J. Meseguer. General logics. In *Logic Colloquium 87*, pages 275–329. North Holland, 1989.

[17] T. Mossakowski. Comorphism-based Grothendieck logics. In *Proc. 27th Intl. Symp. Mathematical Foundations of Computer Science*, LNCS 2420, pages 593–604. Springer, 2002.

[18] T. Mossakowski. Relating CASL with other specification languages: The institution level. *Theoretical Comput. Sci.*, 286:367–475, 2002.

[19] T. Mossakowski. Foundations of heterogeneous specification. In *Recent Trends in Algebraic Development Techniques, 16th Intl. Workshop, WADT 2002, Revised Selected Papers*, LNCS 2755, pages 359–375. Springer, 2003.

[20] T. Mossakowski. Heterogeneous specificiations and the Heterogeneous Tool Set. This volume, 2004.

[21] L. Paulson. *ML for the Working Programmer*. Cambridge University Press, 1991.

[22] D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76:165–210, 1988.

[23] D. Sannella and A. Tarlecki. Toward formal development of programs from algebraic specifications: Implementations revisited. *Acta Informatica*, 25:233–281, 1988.

[24] D. Sannella and A. Tarlecki. Essential concepts of algebraic specification and program development. *Formal Aspects of Comput.*, 9:229–269, 1997.

[25] L. Schröder, T. Mossakowski, A. Tarlecki, B. Klin, and P. Hoffman. Amalgamation in the semantics of CASL. *Theoretical Comput. Sci.* To appear.

[26] A. Tarlecki. Bits and pieces of the theory of institutions. In *Proc. Intl. Workshop on Category Theory and Computer Programming*, LNCS 240, pages 334–363. Springer, 1986.

[27] A. Tarlecki. Moving between logical systems. In *Recent Trends in Data Type Specifications. 11th Workshop on Specification of Abstract Data Types*, LNCS 1130, pages 478–502. Springer, 1996.

[28] A. Tarlecki. Institutions: An abstract framework for formal specifications. In E. Astesiano, H.-J. Kreowski, and B. Krieg-Brückner, editors, *Algebraic Foundations of Systems Specifications*, pages 105–130. Springer, 1999.

[29] A. Tarlecki. Towards heterogeneous specifications. In *Frontiers of Combining Systems 2, 1998*, pages 337–360. Research Studies Press, 2000.

[30] A. Tarlecki. Abstract specification theory: An overview. In *Models, Algebras and Logic of Engineering Software*, pages 43–79. IOS Press, 2003.

Why Are Combined Modal Logics So Robustly Undecidable?

Frank Wolter

Department of Computer Science, University of Liverpool,
Liverpool L69 7ZF, U.K.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

One of the main reasons for the success of modal logics in computer science is their unusual robust decidability. Indeed, standard modal logics like polymodal **K**, **S4**, and **S5**, temporal logics like **LTL** and **CTL**, dynamic logics like **PDL**, epistemic logics like **S5** with common knowledge, description logics like $\mathcal{ALC}$ and $\mathcal{SHIQ}$, and modal spatial logics like $\mathbf{S4}_u$, are decidable in EXPTIME [7, 4].

However, modern applications of modal languages often require rather complex formal models and corresponding languages that are capable of reflecting different features of an application domain. It is not sufficient to work with just one sort of modal operators (say, epistemic or temporal) but subtle combinations of different families of modal operators are required. For example,

- to analyze the behaviour of a multi-agent distributed system we may need a formalism containing both epistemic operators for capturing knowledge of agents and temporal operators for taking care of the development of this knowledge in time. In other words, we should construct a suitable combination of epistemic and temporal logics [2].

- To describe the behaviour of spatial objects (regions) changing over time we may need combinations of spatial modal logics with temporal logics;

- to analyze the dynamics of ontologies and the knowledge of agents about ontologies we may need combinations of description logics (as the underlying language for ontologies) with dynamic modal logics and epistemic logics.

- Fragments of first-order logic may be required to describe an application domain. In this case, instead of propositional modal logics, first-order modal logics are required; i.e., we have to combine (fragments) of first-order logic with propositional modal logics.

While decidability is preserved under forming combinations of modal logics without interaction axioms or constraints (i.e., fusions) [8, 1] this situation changes drastically as soon as some kind of interaction between the modalities is required. In contrast to standard modal logics, combined modal logics often exhibit rather nasty computational properties, and the standard toolkit of modal logic, e.g.,

- proving the finite model property by some kind of filtration;

- proving a variant of the tree-model property and applying automata-based decision procedures [7, 4];

- providing an embedding into a decidable fragment of first-order logic [4]

is no longer directly available. In fact, straightforward constructions of combined modal logics from simple one-dimensional ones will almost certainly result in computationally useless 'monsters'.

The aim of this presentation is twofold: (i) to discuss possible explanations for this phenomenon - *the robust undecidability of combined modal logics* - and (ii) to use the insights gained from this discussion in order to provide methodologies for constructing computationally well-behaved combined modal logics.

We will argue that the high computational complexity or even undecidability of most combined modal logics is explained by the two- or many-dimensional structure of their intended models [3]:

- The fact that almost all *three-dimensional* modal logics are undecidable can be intuitively explained by the undecidability of the product $\mathbf{S5}^3$ and its relation to the undecidable 3-variable fragment of first-order logic.

- *Two-dimensional* modal logics are usually at least NEXPTIME-hard. This can by intuitively explained by the NEXPTIME-completeness of the product $\mathbf{S5}^2$ and its relation to the 2-variable fragment of first-order logic. However, the computational properties of two-dimensional modal logics depend - in a way not yet completely understood - on the geometry of the models of the components. We will present partial results for two-dimensional modal logics as well as open problems based on [3].

- Products of **S5** with $\mathbf{CTL}^*$ and **CTL** may be regarded as $2\frac{1}{2}$-dimensional modal logics and show again the computational significance of the move from two to three dimensions: the product with $\mathbf{CTL}^*$ is undecidable while the product with **CTL** (or **LTL**) is decidable [5].

- The decidability/undecidability of fragments of first-order modal logics is closely related to the number of free variables allowed withing formulas starting with a modal operator; i.e., formulas of the form $\Box\varphi(\vec{x})$. While *monodic* fragments - which allows for just one free such variable - often exhibits 'good' computational properties, two variables usually lead to undecidable fragments already [3]. We will give an overview of explanations for this and discuss open problems.
- Finally, we discuss the method of E-connections to produce decidable and expressive combinationas of modal logics of many dimensions [6].

Acknowledgements

This presentation is based on joint work with Ian Hodkinson, Agnes Kurucz, Carsten Lutz and Michael Zakharyaschev.

Bibliography

[1] F. Baader, S. Ghilardi, and C. Tinelli. A new combination procedure for the word problem that generalizes fusion decidability results in modal logics. In D. Basin and M. Rusinowitch, editors, *Proceedings of the 2nd International Joint Conference on Automated Reasoning (IJCAR'04)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, 2004. To appear.

[2] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[3] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: theory and applications*. Elsevier, 2003.

[4] E. Grädel. Why are modal logics so robustly decidable? In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science. Entering the 21st Century*, pages 393–408. World Scientific, 2001.

[5] I. Hodkinson, F. Wolter, and M. Zakharyaschev. Decidable and undecidable fragments of first-order branching temporal logics. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (LICS'02)*, pages 393–402. IEEE, 2002.

[6] O. Kutz, C.Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artificial Intelligence*, 156:1 – 73, 2004.

[7] M. Vardi. Why is modal logic so robustly decidable? In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 31, pages 149–184. AMS, 1997.

[8] F. Wolter. Fusions of modal logics revisited. In M. Kracht, M. De Rijke, H. Wansing, and M. Zakharyaschev, editors, *Advances in Modal Logic*, volume 1, pages 361–379. CSLI, Stanford, 1997.

A Paradox in the Combination of Logics

Jean-Yves Béziau

Institute of Logic, University of Neuchâtel, Neuchâtel, Switzerland
Member of the LOCIA project (CNPq, Brazil)
jean-yves.beziau@unine.ch

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

In this paper we present a fact, surprising enough to be called a paradox, which shows that the central issue in combination of logic is still problematic. This issue has been described by Dov Gabbay in his book on fibration as follows "Combine $S1$ and $S2$ into a system S which is the smallest logical system for the combined language which is a conservative extension of both $S1$ and $S2$. The two systems are presented in totally different ways. How are we going to combine them." ([9], p.7)

Given two logics $L1$ and $L2$, let us call $L1 * L2$ the combination of $L1$ and $L2$ described by Gabbay, i.e. the smallest logic for the combined language which is a conservative extension of both $L1$ and $L2$. If we have a mechanism for combining semantics or proof systems, how can we be sure that this mechanism produces $L1 * L2$? If we have a technique to combine a Kripke semantics $K1$ generating a logic $L1$ and a Kripke semantics $K2$ generating a logic $L2$, we would like to be sure that the combination of $K1$ and $K2$ generates the combined logic $L1 * L2$. Modal logic is one of the favourite subject of logic combinators and it has been investigated since many years, so it is not surprising that people have found some techniques producing the expected result. But there are some other cases, where there is not yet a solution. The difficulty does not appear in a remote region of the logic land, e.g. the combination of super turbo polar fuzzy logics, but in a very simple case: good old classical propositional logic.

Consider the semantics SC for classical conjunction, given by the following usual condition: $b(F \wedge G) = 1$ iff $b(F) = 1$ and $b(G) = 1$. We call LC the consequence relation (logic, for short) generated by this condition using the usual method.

Similarly we consider the semantics SD for classical disjunction, given by the following usual condition: $b(F \vee G) = 1$ iff $b(F) = 1$ or $b(G) = 1$ and we call LD the generated logic.

Now if we put together the two conditions SC and SD in the natural way, we get a logic LCD which is not the expected one, it is not $LC * LD$, i.e. the smallest logic for the combined language which is a conservative extension of both LC and LD.

In the logic LCD generated by the combination of SC and SD, we have distributivity between conjunction and disjunction:

$(F \wedge G) \vee H \dashv\vdash (F \vee H) \wedge (G \vee H)$

$(F \vee G) \wedge H \dashv\vdash (F \wedge H) \vee (G \wedge H)$

The reader can check this with the truth-table method. But distributivity does not hold in $LC * LD$ by definition. Strangely enough, the combination of SC and SD produces something new, which was apparently neither in SC nor in SD. This kind of combination remembers biological phenomena and should perhaps better be called copulation. Note furthermore that here we have two logics which are presented in a very similar way, not heterogeneous presentations as suggested by Gabbay. So the challenge seems bigger than expected.

What can be said is that truth-functionality is not preserved by combination, since LC and LD are truth-functional (i.e. have a truth-functional semantics) but not $LC * LD$. The combination of SC and SD is a particular case of combination of logical matrices. What the paradox shows is that if we combine logical matrices in the natural way, we don't necessarily get what we want.

We find a similar problem at a proof-theoretical level. Let us consider the system GC which is the Gentzen system that we get by keeping only the structural rules and the two rules for conjunction of Gentzen's sequent system LK for classical logic. We consider in a similar way the system GD. Now the system GCD that we get by putting together the rules of GC and GD generates the logic LCD. In particular it is sound and complete for the combination of the semantics SC and SD (see [1]). It is in fact easy to prove distributivity in this system. Distributivity appears as derived rules. What shows the paradox here, is that if we put rules of two systems together, we may get more than expected, like if the rules were copulating.

It is possible to find a Gentzen sequent system for $LC * LD$. Consider the system $GC1$ which is the same as GC except that sequents must have one and only one formula on both sides. We define $GD1$ in a similar way. $GC1$ and $GD1$ generates respectively LC and LD, and when we put them together we get a system which generates $LC * KD$. They do not copulate, or if they do, this does not produce a fruit.

We may find several other examples where this kind of paradox appears. The paradox can be explained by the fact that in a semantics, or in a proof system, all features are not explicit. The implicit features may not manifest themselves isolately, but they may manifest, become active and produce something new by the combination process. Combination then turns into productive copulation.

The same logic can be generated by many different methods. But the fact that different methods generate the same logic does not mean that these methods are equivalent in general. LC can be generated

by a standard system of sequents, namely GC, and by a substructural one, $GC1$. This doesn't mean that substructural and non substructural systems of sequents are the same.

Acknowledgments

Work supported by a grant of the Swiss National Science Foundation.

Bibliography

[1] J.-Y.Béziau, 1995, *Recherches sur la Logique Universelle*, Department of Mathematics, University of Paris 7, Paris.

[2] D.Gabbay, 1999, Fibring logics, Clarendon, Oxford, 1999.

Finite Algebraizability Via Possible-Translations Semantics

J. Bueno Marcelo E. Coniglio Walter A. Carnielli

CLE and IFCH – Universidade Estadual de Campinas
P.O. Box 6133, 13081-970, Campinas, SP, Brazil
{juliana.bueno,coniglio,carniell}@cle.unicamp.br

Abstract

The general idea of combining logics also involves the concept of breaking logics into families of logics with lower semantical complexity. The possible-translations semantics (**PTS**'s) are a particularly apt tool for analyzing and providing semantical meaning and algebraic contents to certain complex logics as paraconsistent logics. This paper characterizes **PTS**'s and the concept of algebraizability via **PTS**'s in categorial terms, extending the concept of finitely-algebraizable (or Blok-Pigozzi algebraizable) logics.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

Introduction

In a broad view, combining logics includes, besides synthesizing logical systems by means of compositional procedures (as for instance fibring), also the direction of analysing a logic in terms of less complex logics. Such a reversing procedure is called *splitting*, as opposite to the process of *splicing* (cf. [6]).

In this paper we propose a categorial characterization of the process of splitting logics called *possible-translation semantics* (**PTS**'s, cf. [6]). Such semantics are adequate to providing interpretation to several non-standard logics (as to paraconsistent and to many-valued logics, for instance).

On the other hand, by combining algebraic techniques with **PTS**'s one obtains a new notion of algebraizability (cf. [3]) extending the method of finitely algebraizable logics due to W. Blok and D. Pigozzi in [1]. This extended notion of algebraization offers a solution to the question of obtaining an exact algebraic counterpart to certain logics which are not amenable to the method of Blok and Pigozzi, as it is the case of various paraconsistent logics.

The main results of this paper are the characterization of the possible-translations semantics (**PTS**) and the concept of algebraizability via **PTS**'s in categorial terms, and the proof of the Finiteness Preservation Lemma (13): it is proven that the product of finitely-algebraizable propositional logics is also finitely-algebraizable, under certain conditions.

This is achieved by defining the categories **PS** of propositional languages, and **CR** of propositional logics (defined through consequence relations), and showing that they are closed under arbitrary products. This permits to specify the category **ACR** of algebraizable logics as a subcategory of **CR** (cf. [10]). Examples and some research directions are discussed.

1 Propositional languages

Definition 1. A signature is a denumerable family $\Sigma = \{\Sigma_k\}_{k\in\omega}$, where each Σ_k is a set (of connectives of arity k) such that $\Sigma_k \cap \Sigma_n = \emptyset$ whenever $k \neq n$. The domain of Σ is the set $|\Sigma| = \bigcup_{n\in\omega} \Sigma_n$. We fix a denumerable set $\mathcal{V} = \{p_k \ : \ k \in \omega, k \geq 1\}$ of (propositional) variables such that $p_k \neq p_n$ whenever $k \neq n$. The (propositional) language generated by Σ, denoted by $L(\Sigma)$, is the algebra of type Σ freely generated by $\mathcal{V}$. Elements of $L(\Sigma)$ are called formulas. For every $n \geq 0$ consider the following sets:

$$L(\Sigma)[n] = \{\varphi \in L(\Sigma) \ : \ \text{the variables occurring in } \varphi \text{ are exactly } p_1, \ldots, p_n\},$$

$$L(\Sigma)(n) = \{\varphi \in L(\Sigma) \ : \ \text{the variables occurring in } \varphi \text{ are among } p_1, \ldots, p_n\}.$$

Definition 2. Let Σ be a signature. A substitution on $L(\Sigma)$ is a function $\sigma{:}\mathcal{V} \to L(\Sigma)$. We denote by $\widehat{\sigma}$ the unique extension of σ to an endomorphism $\widehat{\sigma} : L(\Sigma) \to L(\Sigma)$.

Given $\varphi \in L(\Sigma)(n)$ and σ such that $\sigma(p_i) = \alpha_i$ $(i = 1, \ldots, n)$ then $\widehat{\sigma}(\varphi)$ will be denoted by $\varphi(\alpha_1, \ldots, \alpha_n)$.

Definition 3. Let Σ and Σ' be signatures. A signature morphism f from Σ to Σ', denoted $\Sigma \xrightarrow{f} \Sigma'$, is a map $f : |\Sigma| \to L(\Sigma')$ such that, if $c \in \Sigma_n$ then $f(c) \in L(\Sigma')[n]$.

Given a signature morphism $\Sigma \xrightarrow{f} \Sigma'$, a map $\widehat{f} : L(\Sigma) \to L(\Sigma')$ can be defined in a natural way:

1. $\widehat{f}(p) = p$ if $p \in \mathcal{V}$;
2. $\widehat{f}(c) = f(c)$ if $c \in \Sigma_0$;
3. $\widehat{f}(c(\alpha_1, \ldots, \alpha_n)) = f(c)(\widehat{f}(\alpha_1), \ldots, \widehat{f}(\alpha_n))$ if $c \in \Sigma_n$ and $\alpha_1, \ldots, \alpha_n \in L(\Sigma)$.

The extension $\widehat{f}$ of f is unique: if f, f' are signature morphisms such that $\widehat{f} = \widehat{f'}$ then $f = f'$. Moreover, the propositional variables occurring in φ and in $\widehat{f}(\varphi)$ are the same.

Definition 4. Let $\Sigma \xrightarrow{f} \Sigma'$ and $\Sigma' \xrightarrow{g} \Sigma''$ be signature morphisms. The composition $g \centerdot f$ of f and g is the signature morphism $\Sigma \xrightarrow{g\centerdot f} \Sigma''$ given by the map $\widehat{g} \circ f : |\Sigma| \to L(\Sigma'')$.

Definition 5. The Category **PS** of (Propositional) languages is defined as follows:

- Objects: Propositional signatures (cf. Definition 1);
- Morphisms: Signature morphisms (cf. Definition 3);
- Composition: As in Definition 4;
- Identity morphisms: For every signature Σ the identity morphism $\Sigma \stackrel{id_\Sigma}{\to} \Sigma$ is defined by $id_\Sigma(c) = c$ (for $c \in \Sigma_0$) and $id_\Sigma(c) = c(p_1, \ldots, p_n)$ (for $c \in \Sigma_n$, $n \geq 1$).

Proposition 6. **PS** is s category with arbitrary (small) products.

The proof that **PS** a category is just a verification, as usual. **PS** is proven to have arbitrary (small) products by defining appropriate terminal signatures and using previous definitions.

2 Consequence relations

In this section we introduce the category **CR** of (propositional) logics defined through consequence relations.

Definition 7. A (propositional) logic is a pair $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}} \rangle$, where Σ is a signature (cf. Definition 1) and $\vdash_{\mathcal{L}}$ is a subset of $\wp(L(\Sigma)) \times L(\Sigma)$ satisfying the following properties, for every $\Gamma \cup \Theta \cup \{\varphi\} \subseteq L(\Sigma)$:

- If $\varphi \in \Gamma$ then $\Gamma \vdash_{\mathcal{L}} \varphi$ (Extensivity);
- If $\Gamma \vdash_{\mathcal{L}} \varphi$ and $\Gamma \subseteq \Theta$ then $\Theta \vdash_{\mathcal{L}} \varphi$ (Monotonicity);
- If $\Gamma \vdash_{\mathcal{L}} \varphi$ and $\Theta \vdash_{\mathcal{L}} \psi$ for all $\psi \in \Gamma$ then $\Theta \vdash_{\mathcal{L}} \varphi$ (Transitivity);
- If $\Gamma \vdash_{\mathcal{L}} \varphi$ then $\Delta \vdash_{\mathcal{L}} \varphi$ for some finite $\Delta \subseteq \Gamma$ (Finitariness);
- If $\Gamma \vdash_{\mathcal{L}} \varphi$ then $\widehat{\sigma}(\Gamma) \vdash_{\mathcal{L}} \widehat{\sigma}(\varphi)$ for every substitution σ (Structurality).

Definition 8. Let $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}} \rangle$ and $\mathcal{L}' = \langle \Sigma', \vdash_{\mathcal{L}'} \rangle$ be logics. A morphism of logics $\mathcal{L} \stackrel{f}{\to} \mathcal{L}'$ from $\mathcal{L}$ to $\mathcal{L}'$ is a **PS**-morphism $\Sigma \stackrel{f}{\to} \Sigma'$ which is a translation, that is, it satisfies, for every $\Gamma \cup \{\varphi\} \subseteq L(\Sigma)$:

$$\Gamma \vdash_{\mathcal{L}} \varphi \text{ implies } \widehat{f}(\Gamma) \vdash_{\mathcal{L}'} \widehat{f}(\varphi).$$

By defining composition of morphisms and identity morphisms as in **PS** we then obtain a category of (propositional) logics defined through consequence relations, called **CR**. A fundamental property of **CR** is the following:

Proposition 9. The category **CR** has arbitrary (small) products.

Proof: Let $\mathcal{F} = \{\mathcal{L}_i\}_{i \in I}$ be a family of logics, where I is a set and $\mathcal{L}_i = \langle \Sigma^i, \vdash_{\mathcal{L}_i} \rangle$ for every $i \in I$. If $I = \emptyset$ then, taking the terminal signature $S^{\mathbf{1}}$, $L^{\mathbf{1}} = \langle S^{\mathbf{1}}, \wp(L(S^{\mathbf{1}})) \times L(S^{\mathbf{1}}) \rangle$ is a terminal object in **CR** being, therefore, the product of $\mathcal{F}$; given a logic $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}} \rangle$ then the unique **PS**-morphism $\Sigma \stackrel{!}{\to} S^{\mathbf{1}}$ defines the unique morphism $\mathcal{L} \stackrel{!}{\to} L^{\mathbf{1}}$ in **CR**. If $I \neq \emptyset$ consider the product $\langle \Sigma^{\mathcal{F}}, \{\pi_i\}_{i \in I} \rangle$ of $\{\Sigma^i\}_{i \in I}$ in **PS** (cf. Proposition 6). Define a relation $\vdash_{\mathcal{F}} \subseteq \wp(L(\Sigma^{\mathcal{F}})) \times L(\Sigma^{\mathcal{F}})$ as follows: $\Gamma \vdash_{\mathcal{F}} \varphi$ iff there exists a finite set $\Delta \subseteq \Gamma$ such that $\widehat{\pi}_i(\Delta) \vdash_{\mathcal{L}_i} \widehat{\pi}_i(\varphi)$ for every $i \in I$. The rest of the proof consists in showing that $\mathcal{L}^{\mathcal{F}} = \langle \Sigma^{\mathcal{F}}, \vdash_{\mathcal{F}} \rangle$ is a logic and is not detailed here. ◁

3 Products of algebraizable logics

In this section we prove that, given a (small) family $\mathcal{F}$ of finitely algebraizable logics (in the sense of [1]) satisfying a finite bounding condition, then the product of $\mathcal{F}$ in **CR** is also an algebraizable logic. This will be used in Section 5.

We begin by briefly recalling the basic definitions of [1].

Definition 10. A logic $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}} \rangle$ is algebraizable (in the sense of Blok-Pigozzi) if there exists a finite set $\Delta = \{\Delta^i \ : \ 1 \leq i \leq n\}$ of formulas in $L(\Sigma)[2]$, and a finite set $\langle \varepsilon, \delta \rangle = \{\langle \varepsilon^i, \delta^i \rangle \ : \ 1 \leq i \leq m\}$ contained in $L(\Sigma)[1] \times L(\Sigma)[1]$ such that, for every $\varphi, \psi, \alpha \in L_\Sigma$:

1. $\vdash_{\mathcal{L}} \varphi \Delta \varphi$;
2. $\varphi \Delta \psi \vdash_{\mathcal{L}} \psi \Delta \varphi$;
3. $\varphi \Delta \psi$, $\psi \Delta \alpha \vdash_{\mathcal{L}} \varphi \Delta \alpha$;
4. $\varphi_1 \Delta \psi_1, \ldots, \varphi_k \Delta \psi_k \vdash_{\mathcal{L}} c(\varphi_1, \ldots, \varphi_k) \Delta c(\psi_1, \ldots, \psi_k)$ for every $c \in \Sigma_k$ and $\varphi_1, \ldots, \varphi_k, \psi_1, \ldots, \psi_k$ in $L(\Sigma)$;
5. $\varphi \vdash_{\mathcal{L}} \varepsilon(\varphi) \Delta \delta(\varphi)$, and $\varepsilon(\varphi) \Delta \delta(\varphi) \vdash_{\mathcal{L}} \varphi$.

We say that $\langle \Delta, \langle \varepsilon, \delta \rangle \rangle$ is an algebraizator for $\mathcal{L}$.

Some remarks on the notation adopted in Definition 10. For any $\varphi, \psi \in L(\Sigma)$ then $\varphi \Delta \psi$ denotes the set of formulas $\{\Delta^i(\varphi, \psi) \ : \ 1 \leq i \leq n\}$, and $\varepsilon(\varphi) \Delta \delta(\varphi)$ denotes the set $\{\Delta^j(\varepsilon^i(\varphi), \delta^i(\varphi)) : \ 1 \leq j \leq n \text{ and } 1 \leq i \leq m\}$. And given sets Γ, Θ of formulas then $\Gamma \vdash_{\mathcal{L}} \Theta$ means that $\Gamma \vdash_{\mathcal{L}} \varphi$ for every $\varphi \in \Theta$. Following [10] we define the category **ACR** of algebraizable logics.

Definition 11. The category **ACR** of algebraizable logics is the subcategory of **CR** defined as follows:

- Objects: logics $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}} \rangle$ which are algebraizable (cf. Definition 10);
- Morphisms: a morphism $\mathcal{L} \xrightarrow{f} \mathcal{L}'$ is a **CR**-morphism $\mathcal{L} \xrightarrow{f} \mathcal{L}'$ such that, if $\langle \Delta, \langle \varepsilon, \delta \rangle \rangle$ and $\langle \Delta', \langle \varepsilon', \delta' \rangle \rangle$ are algebraizators for $\mathcal{L}$ and $\mathcal{L}'$, respectively, then $p_1 \widehat{f}(\Delta) p_2 \vdash_{\mathcal{L}'} p_1 \Delta' p_2$ and $p_1 \Delta' p_2 \vdash_{\mathcal{L}'} p_1 \widehat{f}(\Delta) p_2$, where $p_1 \widehat{f}(\Delta) p_2$ denotes the set of formulas $\{\widehat{f}(\Delta^i)(p_1, p_2) \ : \ 1 \leq i \leq n\}$;
- Composition and identity morphisms: inherited from **CR**.

Remark 12. From [1] we get the following: let $\langle \Delta_1, \langle \varepsilon_1, \delta_1 \rangle \rangle$ and $\langle \Delta_2, \langle \varepsilon_2, \delta_2 \rangle \rangle$ be two algebraizators for a logic $\mathcal{L}$. Then $p_1 \Delta_2 p_2 \vdash_{\mathcal{L}} p_1 \Delta_1 p_2$ and $p_1 \Delta_1 p_2 \vdash_{\mathcal{L}} p_1 \Delta_2 p_2$. Therefore, a **CR**-morphism $\mathcal{L} \xrightarrow{f} \mathcal{L}'$ is a **ACR**-morphism iff there are algebraizators $\langle \Delta, \langle \varepsilon, \delta \rangle \rangle$ and $\langle \Delta', \langle \varepsilon', \delta' \rangle \rangle$ for $\mathcal{L}$ and $\mathcal{L}'$, respectively, such that $p_1 \widehat{f}(\Delta) p_2 \vdash_{\mathcal{L}'} p_1 \Delta' p_2$ and $p_1 \Delta' p_2 \vdash_{\mathcal{L}'} p_1 \widehat{f}(\Delta) p_2$. On the other hand, from [10] we have the following result: a **CR**-morphism $\mathcal{L} \xrightarrow{f} \mathcal{L}'$ is a **ACR**-morphism iff, for every algebraizator $\langle \Delta, \langle \varepsilon, \delta \rangle \rangle$ for $\mathcal{L}$, the pair $\langle \widehat{f}(\Delta), \langle \widehat{f}(\varepsilon), \widehat{f}(\delta) \rangle \rangle$ is an algebrizator for $\mathcal{L}'$.

Now we will prove that the product of a (small) family of algebraizable logics satisfying a finitely bounding condition is algebraizable.

Theorem 13 (Finiteness Preservation). Let $\mathcal{F} = \{\mathcal{L}_i\}_{i \in I}$ be a family of algebraizable logics, where I is a set and $\mathcal{L}_i = \langle \Sigma^i, \vdash_{\mathcal{L}_i} \rangle$ for every $i \in I$. Assume that $\mathcal{F}$ has the following property: there are natural numbers n and m such that, for every $i \in I$, there is an algebraizator $\langle \Delta_i, \langle \varepsilon_i, \delta_i \rangle \rangle$ for $\mathcal{L}_i$ such that Δ_i has at most n elements, and $\langle \varepsilon_i, \delta_i \rangle$ has at most m elements. Then, there exists the product in **ACR** of $\mathcal{F}$.

Proof: By hypothesis we can take, for any $i \in I$, finite sequences

$$\Delta_i^1 \cdots \Delta_i^n$$

$$\langle \varepsilon_i^1, \delta_i^1 \rangle \cdots \langle \varepsilon_i^m, \delta_i^m \rangle$$

such that $\langle \Delta_i, \langle \varepsilon_i, \delta_i \rangle \rangle$ is an algebraizator for $\mathcal{L}_i$, where

$$\Delta_i = \{\Delta_i^1, \ldots, \Delta_i^n\}$$

and $\langle \varepsilon_i, \delta_i \rangle = \{\langle \varepsilon_i^1, \delta_i^1 \rangle, \ldots, \langle \varepsilon_i^m, \delta_i^m \rangle\}$, for every $i \in I$. In fact, it is enough to take, for every $i \in I$, an algebraizator with at most n elements in Δ_i and at most m elements in $\langle \varepsilon_i, \delta_i \rangle$ and list their elements, repeating, if necessary, some elements, in order to define sequences of length n and m, respectively.

Now, consider the product $\langle \mathcal{L}^{\mathcal{F}}, \{\pi_i\}_{i \in I} \rangle$ in **CR** of family $\mathcal{F}$, and define the following formulas in $L(\Sigma^{\mathcal{F}})$:

- $\Delta^j = (\Delta_i^j)_{i\in I}(p_1, p_2)$ for $1 \leq j \leq n$;
- $\varepsilon^j = (\varepsilon_i^j)_{i\in I}(p_1)$ for $1 \leq j \leq m$;
- $\delta^j = (\delta_i^j)_{i\in I}(p_1)$ for $1 \leq j \leq m$.

Finally, let $\Delta = \{\Delta^i \ : \ 1 \leq i \leq n\}$ and $\langle \varepsilon, \delta\rangle = \{\langle \varepsilon^i, \delta^i\rangle \ : \ 1 \leq i \leq m\}$. It can be proven that $\langle \Delta, \langle \varepsilon, \delta\rangle\rangle$ is an algebraizator for $\mathcal{L}^{\mathcal{F}}$ (cf. Definition 10). The rest of the proof consists of proving (by induction on the length of formulas) that the clauses of Definition 10 are satisfied. ◁

4 Possible-translation semantics

Besides considering synthesis of given logics by means of a combination process (as, for instance, fibring) in order to obtain a new logic, it is also convenient to be able to split a logic into a family of simpler logics. This kind of 'reverse' technique is what was called *splitting logics*, as opposite to the process of *splicing logics* (cf. [6]). In this section we provide a categorial characterization of the process of splitting logics called *possible-translation semantics* (cf. [6]).

We begin by adapting the original definitions of [6] to our formalism.

Definition 14. Let $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}}\rangle$ be a logic, and let $\{\mathcal{L}_i\}_{i\in I}$ be a family of logics such that I is a set and $\mathcal{L}_i = \langle \Sigma^i, \vdash_{\mathcal{L}_i}\rangle$ for every $i \in I$. Let $\mathcal{L} \overset{f_i}{\rangle} \mathcal{L}^i$ be a **CR**-morphism for every $i \in I$. Then $P = \langle \{\mathcal{L}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle$ is a possible-translation for $\mathcal{L}$ (in short, a **PTS**) if, for every $\Gamma \cup \{\varphi\} \subseteq L(\Sigma)$,

$$\Gamma \vdash_{\mathcal{L}} \varphi \ \text{ iff there is a finite } \Delta \subseteq \Gamma \text{ such that } \widehat{f}_i(\Delta) \vdash_{\mathcal{L}_i} \widehat{f}_i(\varphi) \ \text{ for every } i \in I.$$

The meaning of having a **PTS** for a logic $\mathcal{L}$ is that $\mathcal{L}$ splits into the family $\{\mathcal{L}_i\}_{i\in I}$ through the translations $\{f_i\}_{i\in I}$.

Inspired by [8] we say that a **CR**-morphism $\mathcal{L} \overset{f}{\rangle} \mathcal{L}'$ is a *conservative translation* if, for every $\Gamma \cup \{\varphi\} \subseteq L(\Sigma)$,

$$\Gamma \vdash_{\mathcal{L}} \varphi \ \text{ iff } \ \widehat{f}(\Gamma) \vdash_{\mathcal{L}'} \widehat{f}(\varphi).$$

Using the results stated in the previous sections we characterize **PTS**'s in categorial terms.

We also show below that **PTS**'s for a logic induce conservative translations over products of families of logics in **CR**, and vice-versa. Such (induced) conservative translations turn out to be apt to extend the method of finite algebraizability with interesting applications, as shown in Section 5.

Theorem 15. Given a possible-translation semantics for a logic $\mathcal{L}$ there exists a conservative translations $\mathcal{L} \overset{f}{\rangle} \mathcal{L}'$, (where $\mathcal{L}'$ is a product in **CR** of some family of logics), and vice-versa.

Proof: Fix a logic $\mathcal{L} = \langle \Sigma, \vdash_{\mathcal{L}}\rangle$. If P is a **PTS** for $\mathcal{L}$ then it is possible to define a conservative translation $\mathcal{L} \overset{\mathbf{t}(P)}{\rangle} \mathbf{L}(P)$, where $\mathbf{L}(P)$ is a product in **CR** of some (small) family of logics, encoding P. Conversely, given a conservative translation $\mathcal{L} \overset{f}{\rangle} \mathcal{L}'$, where $\mathcal{L}'$ is a product of logics, we can define a **PTS** for $\mathcal{L}$ called $\mathbf{PTS}(f)$ encoding f such that these assignments ($\mathbf{t}$ and **PTS**) are one inverse of the other. Calculations are omitted here. ◁

5 Algebraizing logics by possible-translations semantics

The results above give support to a method for algebraizing logics using **PTS**'s which extends the well-known method of finite algebraizability of [1]. The idea, introduced in [3], is the following: consider a logic $\mathcal{L}$, and let $P = \langle \{\mathcal{L}_i\}_{i\in I}, \{f_i\}_{i\in I}\rangle$ be a **PTS** for $\mathcal{L}$. Suppose that every $\mathcal{L}_i$ is algebraizable, and assume that the family $\mathcal{F} = \{\mathcal{L}_i\}_{i\in I}$ satisfies the condition of Theorem 13. Then, by Theorem 15, the product $\langle \mathcal{L}^{\mathcal{F}}, \{\pi_i\}_{i\in I}\rangle$ of family $\mathcal{F}$ in **ACR** encodes P. Moreover, it is possible to build an algebraizator for $\mathcal{L}^{\mathcal{F}}$ from a bounded (in the sense of Theorem 13) family of algebraizators for $\mathcal{F}$. This shows that there exists a conservative translation $\mathcal{L} \overset{f}{\rangle} \mathcal{L}^{\mathcal{F}}$, where $\mathcal{L}^{\mathcal{F}}$ is an algebraizable logic. The conservative translation

f is a link between $\mathcal{L}$ and $\mathcal{L}^{\mathcal{F}}$ that preserves derivability (see comments after Definition 14) and so, using the algebraization for $\mathcal{L}^{\mathcal{F}}$, one obtains a kind of 'remote' algebraization for $\mathcal{L}$: in order to algebraically analyze $\mathcal{L}$, it is sufficient to translate $\mathcal{L}$ into $\mathcal{L}^{\mathcal{F}}$ and then analyze the result using the algebraic resources of $\mathcal{L}^{\mathcal{F}}$.

Here a concrete example is considered. Possible-translations semantics are used (cf. [6] and [12]) to obtain new semantics for the paraconsistent systems C_n introduced in [7]. Such semantics permit to characterize the logics C_n in terms of a family $\mathcal{F}_n$ of three-valued logics **LFI1** (cf.[4]), which are equivalent to the three-valued paraconsistent logic J_3 (introduced in [9]).

It turns out that J_3, **LFI1** and the three-valued Łukasiewicz logic Ł$_3$ are all finitely algebraizable with the same equivalent quasivariety, to wit, the quasivariety of the three-valued Moisil algebras, as shown in [2], p. 43.

It is clear then that logics J_3 (or **LFI1**) in the family $\mathcal{F}_n$ are finitely algebraizable and their algebraization satisfy the conditions of Finiteness Preservation Theorem 13. As a consequence, the product $\mathcal{L}^{\mathcal{F}_n}$ of the family $\mathcal{F}_n$ is also algebraizable, as argued in [3].

Furthermore, as a consequence of Theorem 15 there exists a conservative translations from each C_n into $\mathcal{L}^{\mathcal{F}_n}$.

This shows that our categorial characterization extends the concept of finite algebraizability in adequate terms, offering a non *ad hoc* solution to the question of algebraizing logics in general, as it amply extends the method of [1]. Other interesting questions, as the characterization of logics having the Craig interpolation property for consequence relation, can be recast here as a challenging problem: indeed, it is known (see [11] page 43 for a discussion) that a logic enjoying the deduction-detachment theorem has the Craig interpolation property iff its algebraization has the amalgamation property. Since the amalgamation property can be seen as a universal construction in the (sub)category **CR** of algebraizable logics, it remains to know whether this would correspond to any form of Craig interpolation property.

Acknowledgments

The first author wishes to acknowledge the financial support of CAPES (Brazil) and the invitation of the CLC (IST, Portugal) for an academic stay in Lisbon. The third author acknowledges financial support from CNPq (Brazil) and from the CLC (IST, Portugal) for a senior scientist research grant. We are also grateful to Xavier Caicedo, Carlos Caleiro, Renato Lewin and Paulo A. S. Veloso for discussions on the ideas of this paper.

Bibliography

[1] W. Blok and D. Pigozzi. *Algebraizable Logics*, volume 77 (396) of *Memoirs of the American Mathematical Society*. AMS, Providence, Rhode Island, 1989.

[2] W. J. Blok and D. Pigozzi. Abstract algebraic logic and the deduction theorem. To appear.

[3] J. Bueno and W.A. Carnielli. Possible-translations algebraic semantics: algebraizing paraconsistent logics. To appear, 2004.

[4] W. A. Carnielli, J. Marcos, and S. de Amo. Formal inconsistency and evolutionary databases. *Logic and Logical Philosophy*, 8:115–152, 2000.

[5] W.A. Carnielli. Possible-Translations Semantics for Paraconsistent Logics. In D. Batens, C. Mortensen, G. Priest, and J. P. Van Bendegem, editors, *Frontiers of Paraconsistent Logic: Proceedings of the I World Congress on Paraconsistency*, Logic and Computation Series, pages 149–163. Baldock: Research Studies Press, King's College Publications, 2000.

[6] W.A. Carnielli and M.E. Coniglio. A categorial approach to the combination of logics. *Manuscrito*, 22(2):69–94, 1999.

[7] N. C. A. da Costa. *Inconsistent Formal Systems* (in Portuguese). PhD thesis, Federal University of Parana, Curitiba, Brazil, 1963. Edited by Editora UFPR, Curitiba, 1993.

[8] J. J. da Silva, I. M. L. D'Ottaviano, and A. M. Sette. Translations between logics. In X. Caicedo and C. H. Montenegro, editors, *Models, Algebras and Proofs: Selected Papers of the X Latin American Symposium on Mathematical Logic Held in Bogota*, pages 435–448. Marcel Dekker, 1999.

[9] I.M.L. D'Ottaviano and N. C. A. da Costa. Sur un problème de Jaśkowski. *Comptes Rendus de l'Academie de Sciences de Paris (A-B)*, 270:1349–1353, 1970.

[10] V.L. Fernández and M.E. Coniglio. Fibring algebraizable consequence systems. To appear, 2004.

[11] J. M. Font, R. Jansana, and D. Pigozzi. A survey of abstract algebraic logic. *Studia Logica*, 74:13–97, 2003.

[12] J. Marcos. Semânticas de Traduções Possíveis (Possible Translations Semantics, in Portuguese). Master's thesis, IFCH-UNICAMP, Campinas, Brazil, 1999.
URL = http://www.cle.unicamp.br/pub/thesis/J.Marcos/.

Cryptofibring

Carlos Caleiro Jaime Ramos

CLC, Department of Mathematics, IST, Lisbon, Portugal

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

Fibring [13, 14, 30] is recognized as one of the main mechanisms for combining logics, namely because of the general preservation results that have been established for different metatheoretic properties, eg. completeness [32, 9]. However, fibring suffers from an anomaly usually known as "the collapsing problem" [2, 4]. Indeed, ever since the first accounts of fibring, it could be noticed that fibring the semantics of classical with intuitionistic logic would collapse into just classical logic. In [28], modulated fibring has been introduced and shown to avoid these collapses, by means of a very careful use of adjunctions between lattice structured models. Cryptofibring is a new structurally simpler alternative to solve the semantic collapse problem, by adopting a generalization of fibred semantics using cryptomorphisms. In particular, cryptofibring encompasses the original definition of fibred model, while admiting also amalgamated models that can be used to show that the above mentioned collapses are no longer present. In this presentation we focus only on propositional based logics, but cryptofibring can be smoothly generalized to cover a wider universe of logics.

2 Cryptofibred semantics

A *signature* is an $\mathbb{N}$-indexed family C. The elements of each C_k are known as *constructors* or *connectives* of arity k. Given a signature C the generated set of formulae is the carrier $L(C)$ of the free C-algebra. A *signature morphism* $g : C \to C'$ is an $\mathbb{N}$-indexed family of maps where each $g_k : C_k \to C'_k$.

The *denotation* $[\![\varphi]\!]_{\mathbf{A}}$ of $\varphi \in L(C)$ in a given a C-algebra $\mathbf{A} = \langle A, \cdot_{\mathbf{A}} \rangle$ is inductively defined, as usual, by $[\![c(\varphi_1, \ldots, \varphi_k)]\!]_{\mathbf{A}} = c_{\mathbf{A}}([\![\varphi_1]\!]_{\mathbf{A}}, \ldots, [\![\varphi_k]\!]_{\mathbf{A}})$. A *$C$-structure* is a pair $\mathbb{A} = \langle \mathbf{A}, T_{\mathbb{A}} \rangle$ where $\mathbf{A} = \langle A, \cdot_{\mathbf{A}} \rangle$ is a C-algebra and $T_{\mathbb{A}} \subseteq A$. The elements of A are called *truth-values* and those in $T_{\mathbb{A}}$ are known as *designated* truth-values. In the sequel, we write $[\![\varphi]\!]_{\mathbb{A}}$ for the denotation of φ in the underlying algebra. We denote the class of all C-structures by $Str(C)$. Given a signature morphism $g : C \to C'$ and a C'-structure $\mathbb{A}' = \langle \mathbf{A}', T_{\mathbb{A}'} \rangle$, the *reduct* of $\mathbb{A}'$ by g is the C-structure $\mathbb{A}'|_g = \langle \mathbf{A}'|_g, T_{\mathbb{A}'} \rangle$ where $\mathbf{A}'|_g = \langle A', \cdot_{\mathbf{A}'} \circ g \rangle$. Observe that $[\![\varphi]\!]_{\mathbb{A}'|_g} = [\![g(\varphi)]\!]_{\mathbb{A}'}$ for any $\varphi \in L(C)$. Given two C-structures $\mathbb{A}$ and $\mathbb{B}$, a *C-homomorphism* $h : \mathbb{A} \to \mathbb{B}$ is a C-algebra homomorphism $h : \mathbf{A} \to \mathbf{B}$ such that $T_{\mathbb{A}} = h^{-1}(T_{\mathbb{B}})$. Given a signature morphism $g : C \to C'$, a C-structure $\mathbb{A}$ and a C'-structure $\mathbb{A}'$, a *g-cryptomorphism* $h : \mathbb{A} \to \mathbb{A}'$ is a C-homomorphism $h : \mathbb{A} \to \mathbb{A}'|_g$.

An *interpretation system* is a tuple $\mathcal{I} = \langle C, \mathcal{M}, \alpha \rangle$ where C is a signature, $\mathcal{M}$ is a class and $\alpha : \mathcal{M} \to Str(C)$. The elements of $\mathcal{M}$ are known as *models*. In the sequel, we write $\mathbb{A}_m = \langle \mathbf{A}_m, T_m \rangle$ for $\alpha(m)$, $\cdot_m$ for $\cdot_{\mathbf{A}_m}$, and $[\![\varphi]\!]_m$ for $[\![\varphi]\!]_{\alpha(m)}$. Given $\Psi \subseteq L(C)$ and $\varphi \in L(C)$, we say that Ψ *entails* φ in $\mathcal{I}$, written $\Psi \vdash_{\mathcal{I}} \varphi$, if, for every $m \in \mathcal{M}$, $[\![\varphi]\!]_m \in T_m$ whenever $[\![\Psi]\!]_m \subseteq T_m$.

Let $\mathcal{I} = \langle C, \mathcal{M}, \alpha \rangle$ and $\mathcal{I}' = \langle C', \mathcal{M}', \alpha' \rangle$ be interpretation systems. An *interpretation system cryptomorphism* $f : \mathcal{I} \to \mathcal{I}'$ is a triple $f = \langle g, \mu, h \rangle$ where:

- $g : C \to C'$ is a signature morphism;
- $\mu : \mathcal{M}' \to \mathcal{M}$ is a map;
- $h = \{h_{m'}\}_{m' \in \mathcal{M}'}$ with each $h_{m'} : \mathbb{A}_{\mu(m')} \to \mathbb{A}_{m'}$ being a g-cryptomorphism.

Interpretation systems and cryptomorphisms constitute a category **cInt**.

We assume given two interpretation systems $\mathcal{I}' = \langle C', \mathcal{M}', \alpha' \rangle$ and $\mathcal{I}'' = \langle C'', \mathcal{M}'', \alpha'' \rangle$. Furthermore, we denote by C_0 the signature $C' \cap C''$ and we consider the interpretation system $\mathcal{I}_0 = \langle C_0, Str(C_0), Id \rangle$ endowed with the obvious cryptomorphisms $i'_0 : \mathcal{I}_0 \to \mathcal{I}'$ and $i''_0 : \mathcal{I}_0 \to \mathcal{I}''$. Then, the *cryptofibring* of $\mathcal{I}' = \langle C', \mathcal{M}', \alpha' \rangle$ and $\mathcal{I}'' = \langle C'', \mathcal{M}'', \alpha'' \rangle$ *constrained by sharing* C_0, is the interpretation system $\mathcal{I}' \circledast \mathcal{I}'' = \langle C' \cup C'', \mathcal{M}' \circledast \mathcal{M}'', \alpha_{\circledast} \rangle$ where:

- $C' \cup C''$ is the union of C' and C'', and $i' : C' \to C' \cup C''$ and $i'' : C'' \to C' \cup C''$ are the obvious inclusion morphisms;
- $\mathcal{M}' \circledast \mathcal{M}''$ is the class of tuples $\langle \mathbb{A}, m', m'', h', h'' \rangle$ such that:
  - $\mathbb{A} \in Str(C' \cup C'')$;
  - $m' \in \mathcal{M}'$ and $m'' \in \mathcal{M}''$;

  - $h' : \mathbb{A}_{m'} \to \mathbb{A}$ is a i'-cryptomorphism;
  - $h'' : \mathbb{A}_{m''} \to \mathbb{A}$ is a i''-cryptomorphism;
- $\alpha_{\circledast}(\langle \mathbb{A}, m', m'', h', h'' \rangle) = \mathbb{A}$.

When C_0 is the empty signature we say that the cryptofibring is *unconstrained*. Observe that $\mathcal{I}' \circledast \mathcal{I}''$ includes those structures that appear in the original notion of fibring: the class of all structures $\mathbb{A}$ such that (i) $\mathbb{A}|_{i'} \in \alpha'(\mathcal{M}')$ and (ii) $\mathbb{A}|_{i''} \in \alpha''(\mathcal{M}'')$. But $\mathcal{I}' \circledast \mathcal{I}''$ is much richer. It also includes amalgamated structures as described next. Given a C'-algebra $\mathbf{A}'$ and a C''-algebra $\mathbf{A}''$, their *amalgamation* is the $C' \cup C''$-algebra $\mathbf{A} = \mathbf{A}' \oplus \mathbf{A}'' = \mathcal{F}_{C' \cup C''}(A' \uplus A'')/_{\equiv}$ where $\equiv$ is the least congruence such that:

- $c'(a'_1, \ldots, a'_k) \equiv c'_{\mathbb{A}'}(a'_1, \ldots, a'_k)$ for $a'_1, \ldots, a'_k \in A'$ and $c' \in C'_k$;
- $c''(a''_1, \ldots, a''_k) \equiv c''_{\mathbb{A}''}(a''_1, \ldots, a''_k)$ for $a''_1, \ldots, a''_k \in A''$ and $c'' \in C''_k$.

This is however not a colimit in **cInt** as shown below.

In the sequel, we use the obvious amalgamation injections $j' : A' \to A$ and $j'' : A'' \to A$. Note that we can set up a flattened category **cAlg** of algebras and cryptomorphisms as follows. Each object is a pair $\langle C, \mathbf{A} \rangle$ where $\mathbf{A}$ is a C-algebra. Each morphism $\langle g, h \rangle : \langle C, \mathbf{A} \rangle \to \langle C', \mathbf{A}' \rangle$ is composed of a signature morphism $g : C \to C'$ and a C-algebra homomorphism $h : \mathbf{A} \to \mathbf{A}'|_g$. In this category, the pair $\langle C' \cup C'', \mathbf{A}' \oplus \mathbf{A}'' \rangle$ endowed with $\langle i', j' \rangle$ and $\langle i'', j'' \rangle$ is a coproduct of $\langle C', \mathbf{A}' \rangle$ and $\langle C'', \mathbf{A}'' \rangle$. Given two interpretation systems $\mathcal{I}' = \langle C', \mathcal{M}', \alpha' \rangle$ and $\mathcal{I}'' = \langle C'', \mathcal{M}'', \alpha'' \rangle$, their *amalgamation* is the interpretation system $\mathcal{I}' \oplus \mathcal{I}'' = \langle C' \cup C'', \mathcal{M}' \oplus \mathcal{M}'', \alpha_{\oplus} \rangle$ where:

- $\mathcal{M}' \oplus \mathcal{M}''$ is the class of tuples $\langle m', m'', T \rangle$ such that:
  - $m' \in \mathcal{M}'$ and $m'' \in \mathcal{M}''$;
  - T is a subset of the carrier set of $\mathbf{A}_{m'} \oplus \mathbf{A}_{m''}$ such that:
    - $j'^{-1}(T) = T_{m'}$;
    - $j''^{-1}(T) = T_{m''}$;

    where j' and j'' are the underlying amalgamation injections;
- $\alpha_{\oplus}(\langle m', m'', T \rangle) = \langle \mathbf{A}_{m'} \oplus \mathbf{A}_{m''}, T \rangle$.

Note that $\mathcal{M}' \oplus \mathcal{M}''$ includes, among others, the "minimal" models of the form

$$\langle m', m'', j'(T_{m'}) \cup j''(T_{m''}) \rangle.$$

It also includes models with more designated values, as long as they are chosen outside $j'(A_{m'}) \cup j''(A_{m''})$.

Proposition 1. *$\mathcal{I}' \oplus \mathcal{I}''$ is a pushout of $\{i'_0 : \mathcal{I}_0 \to \mathcal{I}', i''_0 : \mathcal{I}_0 \to \mathcal{I}''\}$ in **cInt**.*

However, cryptofibring of interpretation systems enjoys the following nice relationship with amalgamation.

Proposition 2. *$\vdash_{\mathcal{I}' \circledast \mathcal{I}''}$ coincides with $\vdash_{\mathcal{I}' \oplus \mathcal{I}''}$.*

Cryptofibred semantics, *per se*, may not be suitable as it encompasses so many models that soundness with respect to possible deduction systems associated with the interpretation systems at hand can easily be lost. Next, we adapt the construction to a suitable notion of logic system encompassing both a semantic and a deductive component, together with a soundness condition.

3 Cryptofibring

Assume given once and for all a set Ξ of *schema variables*. Given a signature C, the generated set of schema formulae is the carrier $SL(C)$ of the free C-algebra with generators Ξ. A *schema C-substitution* is a function $\sigma : \Xi \to SL(C)$. Given an schema formula δ, the *instance* of δ by the schema substitution σ is denoted by $\delta\sigma$ and is the result of simultaneously replacing each schema variable ξ in δ by $\sigma(\xi)$. A *(ground) C-substitution* is a function $\rho : \Xi \to L(C)$. The *instance* of δ by the substitution ρ is denoted by $\delta\rho$ and is the result of simultaneously replacing each schema variable ξ in δ by $\rho(\xi)$.

A *C-rule* is a pair $\langle \Upsilon, \eta \rangle$, where $\Upsilon \cup \{\eta\} \subseteq SL(C)$. A rule is said to be *finitary* when Υ is finite and is said to be *axiomatic* when Υ is empty. A *deduction system* is a pair $\mathcal{D} = \langle C, R \rangle$ where C is a signature and R is a set of finitary C-rules. The notion of proof is the usual. When there is a proof in $\mathcal{D}$ of δ from Γ, we write $\Gamma \vdash_{\mathcal{D}} \delta$. As usual we may also omit the set of premises when it is empty. Note that proofs are closed for substitutions: if $\Gamma \vdash_{\mathcal{D}} \delta$ then $\Gamma\sigma \vdash_{\mathcal{D}} \delta\sigma$, for any schema substitution σ. The *image* of a C-rule $r = \langle \Upsilon, \eta \rangle$ by a signature morphism $g : C \to C'$ is the C'-rule $g(r) = \langle g(\Upsilon), g(\eta) \rangle$. A *deduction system morphism* $g : \mathcal{D} \to \mathcal{D}'$ is a signature morphism $g : C \to C'$ such that $g(\Upsilon) \vdash_{\mathcal{D}'} g(\eta)$ for every $\langle \Upsilon, \eta \rangle \in R$.

We put the two components together into logic systems. A *logic system* is a tuple $\mathcal{L} = \langle C, R, \mathcal{M}, \alpha \rangle$ where $\mathcal{D}_{\mathcal{L}} = \langle C, R \rangle$ is a deduction system, and $\mathcal{I}_{\mathcal{L}} = \langle C, \mathcal{M}, \alpha \rangle$ is an interpretation system. In the sequel, we may write $\vdash_{\mathcal{L}}$ for $\vdash_{\mathcal{D}_{\mathcal{L}}}$ and $\vdash_{\mathcal{L}}$ for $\vdash_{\mathcal{I}_{\mathcal{L}}}$. A C-structure $\mathbb{A}$ is said to be *appropriate* for a set R of C-rules iff, for every $\langle \Upsilon, \eta \rangle \in R$ and C-substitution ρ, if $[\![\Upsilon\rho]\!]_{\mathbb{A}} \subseteq T_{\mathbb{A}}$ then $[\![\eta\rho]\!]_{\mathbb{A}} \in T_{\mathbb{A}}$. We denote by $App(R)$ the class of C-structures that are appropriate for R. A logic system $\mathcal{L} = \langle C, R, \mathcal{M}, \alpha \rangle$ is said to be:

- *sound* iff $\Psi \vdash_{\mathcal{L}} \varphi$ whenever $\Psi \vdash_{\mathcal{L}} \varphi$ for $\Psi \cup \{\varphi\} \subseteq L(C)$;
- *full* iff $\alpha(\mathcal{M}) = App(R)$;
- *complete* iff $\Psi \vdash_{\mathcal{L}} \varphi$ whenever $\Psi \vdash_{\mathcal{L}} \varphi$ for $\Psi \cup \{\varphi\} \subseteq L(C)$.

Clearly, a logic system is sound iff all its structures are appropriate for its rules. Therefore, every full logic system is sound. Furthermore, every full logic system is complete[1].

Let $\mathcal{L} = \langle C, R, \mathcal{M}, \alpha \rangle$ and $\mathcal{L}' = \langle C', R', \mathcal{M}', \alpha' \rangle$ be logic systems. A *logic system cryptomorphism* $f : \mathcal{L} \to \mathcal{L}'$ is a triple $f = \langle g, \mu, h \rangle$ such that:

1. $g : \mathcal{D}_{\mathcal{L}} \to \mathcal{D}_{\mathcal{L}'}$ is a deduction system morphism;
2. $f : \mathcal{I}_{\mathcal{L}} \to \mathcal{I}_{\mathcal{L}'}$ is an interpretation system cryptomorphism;
3. for every $m' \in \mathcal{M}'$, $\mathbb{A}_{m'} \in App(g(R))$ whenever $\mathbb{A}_{\mu(m')} \in App(R)$.

Condition 3 above is a reasonable requirement that guarantees the preservation of soundness by cryptomorphisms in the following sense: if $\mathcal{L} = \langle C, R, \mathcal{M}, \alpha \rangle$ is sound, so is $g(\mathcal{L}) = \langle C', g(R), \mathcal{M}', \alpha' \rangle$. Otherwise, valid rules when embedded in a larger language context might become unsound (see for instance [9]).

Logic systems and cryptomorphisms constitute a category **cLog**.

We assume given two deduction systems $\mathcal{D}' = \langle C', R' \rangle$ and $\mathcal{D}'' = \langle C'', R'' \rangle$, and again we denote by C_0 the subsignature $C' \cap C''$, and by $\mathcal{D}_0$ the canonical deduction system $\langle C_0, \emptyset \rangle$ endowed with the obvious morphisms $i' : \mathcal{D}_0 \to \mathcal{D}'$ and $i'' : \mathcal{D}_0 \to \mathcal{D}''$. We start by defining the cryptofibring of deduction systems: the *cryptofibring* of $\mathcal{D}'$ and $\mathcal{D}''$ constrained by sharing C_0 is the deduction system $\mathcal{D}' \circledast \mathcal{D}'' = \langle C' \cup C'', R' \cup R'' \rangle$.

We can finally define the notion of cryptofibring of logic systems. We assume given two logic systems $\mathcal{L}' = \langle C', R', \mathcal{M}', \alpha' \rangle$ and $\mathcal{L}'' = \langle C'', R'', \mathcal{M}'', \alpha'' \rangle$. The *cryptofibring* of $\mathcal{L}'$ and $\mathcal{L}''$ *constrained by sharing* C_0 is the logic system $\mathcal{L}' \circledR \mathcal{L}'' = \langle C' \cup C'', R' \cup R'', \mathcal{M}' \circledR \mathcal{M}'', \alpha_{\circledR} \rangle$ where:

- $\mathcal{M}' \circledR \mathcal{M}''$ is composed of every $\langle \mathbb{A}, m', m'', h', h'' \rangle$ in $M' \circledast M''$ such that:
  - $\mathbb{A} \in App(R' \cup R'')$ whenever $\mathbb{A}_{m'} \in App(R')$;
  - $\mathbb{A} \in App(R' \cup R'')$ whenever $\mathbb{A}_{m''} \in App(R'')$;
- $\alpha_{\circledR}(\langle \mathbb{A}, m', m'', h', h'' \rangle) = \mathbb{A}$.

Expectedly, this is not a colimit in **cLog**. As before, we define the amalgamation of logic systems. The *amalgamation* of $\mathcal{L}'$ and $\mathcal{L}''$ is $\mathcal{L}' \oplus_R \mathcal{L}'' = \langle C' \cup C'', R' \cup R'', \mathcal{M}' \oplus_R \mathcal{M}'', \alpha_{\oplus_R} \rangle$ where:

- $\mathcal{M}' \oplus_R \mathcal{M}''$ is composed of every $\langle m', m'', T \rangle$ in $\mathcal{M}' \oplus \mathcal{M}''$ such that:
  - T is closed for all ground instances of rules in $R' \cup R''$;

[1] Fullness underlies the completeness techniques used, for instance, in [32], which are in fact also applicable to cryptofibring since it includes all fibred models. We shall not dwell on completeness preservation by cryptofibring here, but the fact that cryptofibred semantics is richer opens the way to obtaining more general sufficient conditions for completeness preservation.

- $\alpha_{\oplus_R}(\langle m', m'', T\rangle) = \langle \mathbf{A}_{m'} \oplus \mathbf{A}_{m''}, T\rangle$.

Then, the following result holds.

Proposition 3. *$\mathcal{L}' \oplus_R \mathcal{L}''$ is a pushout of $\{i'_0 \colon \mathcal{L}_0 \to \mathcal{L}', i''_0 \colon \mathcal{L}_0 \to \mathcal{L}''\}$ in **cLog**.*

Fortunately, the strong relationship between the two entailments still holds.

Proposition 4. *$\vdash_{\mathcal{L}' \circledR \mathcal{L}''}$ coincides with $\vdash_{\mathcal{L}' \oplus_R \mathcal{L}''}$.*

4 Combining classical and intuitionistic logics

In this section, we use cryptofibring to combine the implicative fragments of classical propositional logic (*CPL*) and intuitionistic propositional logic (*IPL*), and show that the resulting logic does not collapse to *CPL*. To avoid any constraint in the combination we shall assume that *CPL* and *IPL* are based on disjoint denumerable sets P and Q, respectively, of propositional symbols.

The implicative fragment of *CPL* can be easily presented as a logic system with the usual semantics based on bivaluations $v : P \to \{0, 1\}$ with 1 designated, and the set of rules:

- $\langle \emptyset, (\xi_1 \Rightarrow^c (\xi_2 \Rightarrow^c \xi_1))\rangle$
  $\langle \emptyset, ((\xi_1 \Rightarrow^c (\xi_2 \Rightarrow^c \xi_3)) \Rightarrow^c ((\xi_1 \Rightarrow^c \xi_2) \Rightarrow^c (\xi_1 \Rightarrow^c \xi_3)))\rangle$
  $\langle \emptyset, (((\xi_1 \Rightarrow^c \xi_2) \Rightarrow^c \xi_1) \Rightarrow^c \xi_1)\rangle$
  $\langle \{\xi_1, (\xi_1 \Rightarrow^c \xi_2)\}, \xi_2\rangle$.

The implicative fragment of *IPL* can be presented with the usual Kripke semantics and the set of rules:

- $\langle \emptyset, (\xi_1 \Rightarrow^i (\xi_2 \Rightarrow^i \xi_1))\rangle$
  $\langle \emptyset, ((\xi_1 \Rightarrow^i (\xi_2 \Rightarrow^i \xi_3)) \Rightarrow^i ((\xi_1 \Rightarrow^i \xi_2) \Rightarrow^i (\xi_1 \Rightarrow^i \xi_3)))\rangle$
  $\langle \{\xi_1, (\xi_1 \Rightarrow^i \xi_2)\}, \xi_2\rangle$.

Note that the only difference between the calculi is the absence in *IPL* of *Peirce's law*, the third rule of *CPL*.

In order to show that *CPL*®*IPL* does not collapse into *CPL*, we have to find a model over the combined language that satisfies all the rules from both logics, together with cryptomorphisms from suitable models of each of the logics, in such a way that, for instance, the analogous of Peirce's rule for intuitionistic implication is not valid. We shall consider the model obtained by a natural extension of the usual Kripke semantics for intuitionistic logic. Given a partially-ordered Kripke frame $\langle W, \leq\rangle$, and letting $B_\leq$ be the set of all upper-subsets of W, any corresponding model $m = \langle W, \leq, V\rangle$ with $V : P \cup Q \to B_\leq$ induces an interpretation structure $\langle\langle B_\leq, \cdot_m\rangle, \{W\}\rangle$ with:

- $p_m = V(p)$ and $q_m = V(q)$;
- $\Rightarrow^c_m(b_1, b_2) = ((W \setminus b_1) \cup b_2)^c$;
- $\Rightarrow^i_m(b_1, b_2) = ((W \setminus b_1) \cup b_2)^i$,

where $X^c = \{w \in W : \text{there exists } x \in X \text{ such that } x \leq w\}$ and $X^i = \{w \in W : \{w' : w \leq w'\} \subseteq X\}$, given $X \subseteq W$. In the particular case when $W = \{u, v, w\}$ and $u \leq v$, only, and V is such that $V(p)$ is either $\emptyset$ or W, and $V(q_1) = \{v\}$, $V(q_2) = \emptyset$, the structure we obtain satisfies all the rules but $(((q_1 \Rightarrow_i q_2) \Rightarrow_i q_1) \Rightarrow_i q_1)$ does not hold. Indeed, $[\![(((q_1 \Rightarrow_i q_2) \Rightarrow_i q_1) \Rightarrow_i q_1)]\!]_m = \{v, w\}$. However, the reduct of this structure to $\Rightarrow_i$ and Q is a structure of *IPL* with the obvious cryptomorphism. Moreover, it is also possible to map the *CPL* structure based on the bivaluation $v : P \to \{0, 1\}$ such that $v(p) = 0$ if $V(p) = \emptyset$ and $v(p) = 1$ if $V(p) = W$, along the cryptomorphism that sends 0 to $\emptyset$ and 1 to W.

Acknowledgments

The authors are deeply grateful to Amílcar Sernadas and Cristina Sernadas for their essential contribution to the ideas underlying this work. This work was partially supported by FCT and EU FEDER, namely via the Project FibLog POCTI/MAT/37239/2001 of CLC.

Bibliography

[1] M. E. Coniglio, A. Sernadas, and C. Sernadas. Fibring logics with topos semantics. *Journal of Logic and Computation*, 13(4):595–624, 2003.

[2] L. Fariñas del Cerro and A. Herzig. Combining classical and intuitionistic logic. In F. Baader and K.U. Schulz, editors, *Frontiers of Combining Systems*, pages 93–102. Kluwer Academic Publishers, 1996.

[3] D. Gabbay. Fibred semantics and the weaving of logics: part 1. *Journal of Symbolic Logic*, 61(4):1057–1120, 1996.

[4] D. Gabbay. An overview of fibred semantics and the combination of logics. In F. Baader and K.U. Schulz, editors, *Frontiers of Combining Systems*, pages 1–55. Kluwer Academic Publishers, 1996.

[5] D. Gabbay. *Fibring logics*. Oxford University Press, 1999.

[6] A. Sernadas, C. Sernadas, and C. Caleiro. Fibring of logics as a categorial construction. *Journal of Logic and Computation*, 9(2):149–179, 1999.

[7] C. Sernadas, J. Rasga, and W. A. Carnielli. Modulated fibring and the collapsing problem. *Journal of Symbolic Logic*, 67(4):1541–1569, 2002.

[8] A. Zanardo, A. Sernadas, and C. Sernadas. Fibring: Completeness preservation. *Journal of Symbolic Logic*, 66(1):414–439, 2001.

Fibring Algebraizable Consequence Systems

Victor L. Fernández Marcelo E. Coniglio

CLE and IFCH – Universidade Estadual de Campinas
P.O. Box 6133, 13081-970, Campinas, SP, Brazil
{vlfernan,coniglio}@cle.unicamp.br

Abstract

In this article we study the process of (categorial) fibring of consequence systems which are algebraizable in the sense of Blok and Pigozzi. We investigate the question of preservation of algebraizability by fibring, proving that, under certain assumptions, we can preserve algebraizability by fibring of algebraizable consequence systems (with and without sharing connectives). We also study the particular case of algebraizable consequence systems which have Hilbert-style axiomatics. Finally, based on the work of Jánossy *et alia* we construct a category of equivalent algebraic semantics, proving that it is isomorphic to the subcategory of the category of algebraizable consequence systems in which there exists fibring. This suggess the possibility of fibring algebraic semantics.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

Introduction

Since the introduction of the concept of Fibring of Logics by D. Gabbay in the 90's (cf. [5]), such a name was used to define a very broad set of methodologies. With the purpose of formalize the techniques of fibring, Sernadas *et alia* introduced a categorial definition of fibring, (see, for instance, [8, 11, 3]). One of the main issues intrinsic to the idea of fibring is to find preservation results. That is, the problem to know whether certain properties of the logic systems are preserved through the fibring. In the present article, based on these techniques, we study the question of *preservation of algebraizability* by fibring.

As an answer to this question we will offer some conditions for the preservation of algebrizability via fibring. Thus, we will define the category of algebraizable consequence systems ALCO, which will be our framework wherein the problem of fibring will be analyzed. Then, we will obtain the principal preservation result: In an special subcategory of ALCO (called ALCO*) it is possible to obtain the fibring of algebraizable logics. After this we will study a particular case of fibring of algebraizable logics: The case when the logics are defined by means of Hilbert-type axiomatization. We conclude this article by proving the existence of an isomorphism between the category of equivalent algebraic semantics and ALCO*. This opens the possibility to perform the fibring of algebraic semantics, sharing or not function symbols.

1 Preliminaries

In this section we introduce the basic definitions which we propose to deal with: the category of propositional languages (which is also studied in [2]) and the category of algebraizable logics, as well as the categorial notion of fibring.

Definition 1. (i) Fix a denumerable set $\mathcal{V} = \{p_i\}_{i\in\mathbb{N}}$ of propositional variables. A propositional signature is a family of pairwise disjoint sets $C = \{C_k\}_{k\in\mathbb{N}}$ such that $|C| \cap \mathcal{V} = \emptyset$, where $|C| = \bigcup_{n\in\mathbb{N}} C_n$; elements of C_k are called connectives of arity k. The propositional language generated by C (denoted by $L(C)$) is the free algebra (of words) generated by $\mathcal{V}$ over C.
(ii) Given signatures C^1 and C^2, a signature morphism $h : C^1 \to C^2$ is a map $h : |C^1| \to L(C^2)$ such that, if $c \in C^1_n$ then $h(c) \in L(C^2)$ such that $h(c)$ depends exactly on $p_1, \ldots, p_n$.

If $h : C^1 \to C^2$ then there is exactly one extension $\widehat{h} : L(C^1) \to L(C^2)$ given by $\widehat{h}(p) = p$, if $p \in \mathcal{V}$, and $\widehat{h}(c(\beta_1, \ldots, \beta_k)) = h(c)(p_1/\widehat{h}(\beta_1), \ldots, p_k/\widehat{h}(\beta_k))$ if $c \in C^1_k$. By defining the composition of $C^1 \overset{h}{\to} C^2 \overset{k}{\to} C^3$ as $k \centerdot h = \widehat{k} \circ h$, and putting $id_C : C \to C$ as $id_C(c) = c(p_1, \ldots, p_n)$ then we obtain a category of propositional languages, called PLAN.

Definition 2. (i) The category ALCO of algebraizable logics is the category whose objects are consequence systems of the form $\mathcal{L} = \langle C, \vdash\rangle$, where C is a PLAN-object and $\vdash \subseteq \wp(L(C)) \times L(C)$ is a standard consequence relation (that is, extensive, transitive, monotonic, structural and finitary, cf. [30]), which is algebraizable in the sense of Blok-Pigozzi (see [1]). Given $\mathcal{L}_i = \langle C^i, \vdash_i\rangle$ $(i = 1, 2)$, an ALCO-morphism $h : \mathcal{L}_1 \to \mathcal{L}_2$ is a PLAN-morphism $h : C^1 \to C^2$ satisfying:
(a) if $\Gamma \vdash_1 \alpha$, then $h(\Gamma) \vdash_2 h(\alpha)$;
(b) there are algebraizators $\langle(\epsilon^i, \delta^i), \Delta^i\rangle$ for $\mathcal{L}_i$ $(i = 1, 2)$, respectively, such that, for every formula ϕ, ψ of $L(C^1)$, it holds $h(\phi)\Delta^2 h(\psi) \vdash_2 h(\phi)h(\Delta^1)h(\psi)$.
Composition and identity maps are as in PLAN.
(ii) The category ALCO* is the subcategory of ALCO such that the objects are the same, but the morphism are ALCO-morphisms satisfying:
(c) there are algebraizators $\langle(\epsilon^i, \delta^i), \Delta^i\rangle$ for $\mathcal{L}_i$ $(i = 1, 2)$, respectively, such that, for every formula ϕ, ψ of $L(C^1)$, it holds $h(\phi)\Delta^2 h(\psi) \dashv\vdash_2 h(\phi)h(\Delta^1)h(\psi)$.[1]

In the definitions above, we can substitute "there are algebraizators" by "for every algebraizator", because of the properties of algebraizators.

The category ALCO* is not a full subcategory of ALCO, as shows the following example:

Example 3. Let us consider Sette's propositional paraconsistent logic P^1, defined in [9] over a signature C such that $|C| = \{\neg, \vee, \wedge, \Rightarrow\}$. In the positive fragments, P^1 coincide with classical logic CPC. In [7]

[1] $\Gamma_1 \vdash \Gamma_2$ means $\Gamma_1 \vdash \gamma$ for every $\gamma \in \Gamma_2$; and $\Gamma_1 \dashv\vdash \Gamma_2$ means $\Gamma_1 \vdash \Gamma_2$ and $\Gamma_2 \vdash \Gamma_1$.

it was proven that P^1 is algebraizable, where $\Delta_{P^1} = \{p_1 \Rightarrow p_2, p_2 \Rightarrow p_1, \neg p_1 \Rightarrow \neg p_2, \neg p_2 \Rightarrow \neg p_1\}$[2]. Now, let $P^1_{\Rightarrow}$ be the $\{\Rightarrow\}$-fragment of P^1. Since it coincides with the implicational fragment of CPC, it is algebraizable with $\Delta' = \{p_1 \Rightarrow p_2, p_2 \Rightarrow p_1\}$. It is clear that, considering the inclusion map $inc : P^1_{\Rightarrow} \to P^1$, $inc(\Delta')$ cannot be an equivalence set for P^1. Therefore, inc is an ALCO-morphism, but not an ALCO*-morphism.

We adapt the categorial notion of fibring introduced in [8], defining the following:

Definition 4. Let $\mathcal{D}$ be a category of propositional logic system based on the category PLAN, that is, such that there is a forgetful functor $N : \mathcal{D} \to PLAN$. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two $\mathcal{D}$-objects.
(i) The unconstrained fibring of $\mathcal{L}_1$ and $\mathcal{L}_2$ (denoted by $\mathcal{L}_1 \bigoplus \mathcal{L}_2$) is the co-product in $\mathcal{D}$ of $\mathcal{L}_1$ and $\mathcal{L}_2$ (if it exists).
(ii) Let $D = \{i_j : C \to N(\mathcal{L}_j)\}_{j=1,2}$ be diagram in PLAN formed by monomorphisms. Suppose that there exists the coproduct $\mathcal{L}_1 \bigoplus \mathcal{L}_2$ in $\mathcal{D}$, and consider the coproduct $N(\mathcal{L}_1) \bigoplus N(\mathcal{L}_2) = N(\mathcal{L}_1 \bigoplus \mathcal{L}_2)$ of $N(\mathcal{L}_1), N(\mathcal{L}_2)$ in PLAN, with canonical injections $k_j : N(\mathcal{L}_j) \to N(\mathcal{L}_1 \bigoplus \mathcal{L}_2)$ $(j = 1, 2)$. The constrained fibring of $\mathcal{L}_1$ and $\mathcal{L}_2$ by sharing D is the codomain $\mathcal{L}$ of the cocartesian lifting $q : \mathcal{L}_1 \bigoplus \mathcal{L}_2 \to \mathcal{L}$ of the coequalizer $q : N(\mathcal{L}_1 \bigoplus \mathcal{L}_2) \to C'$ of $\{k_1 \centerdot i_1, k_2 \centerdot i_2\}$ in PLAN (if the cocartesian lifting exists).

2 Categorial Fibring in ALCO*

In this section we will prove that ALCO* has constrained and unconstrained fibring.

From [30] we know that the set $Stan(L(C))$ of standard consequence systems defined over the language $L(C)$, ordered by inclusion (of the consequence relations), is a complete lattice. Using this fact, we can define the following:

Definition 5. Let $\mathcal{L}_i = \langle C^i, \vdash_i \rangle$ $(i = 1, 2)$ be two consequence systems in $ALCO^*$. Consider the set $\mathcal{F}$ of standard consequence operators $\vdash$ over the coproduct $C = C^1 \bigoplus C^2$ such that:
(i) $\Gamma \vdash_i \alpha$ implies $\Gamma \vdash \alpha$ for every $\Gamma \cup \{\alpha\} \subseteq L(C^i)$ $(i = 1, 2)$;
(ii) $\varphi\Delta^i\psi \vdash \varphi\Delta^j\psi$, for some Δ^i and Δ^j equivalence sets of the consequence systems $\mathcal{L}_i$ and $\mathcal{L}_j$ $(i, j \in \{1, 2\})$.[3]
We define the consequence system $\mathcal{L}_{\mathcal{F}} = \langle C, \vdash_{\mathcal{F}} \rangle$ such that $\vdash_{\mathcal{F}}$ is the infimum of the family $\mathcal{F}$ in $Stan(L(C))$.

Then we can prove the following:

Proposition 6. Let $\mathcal{L}_i$ $(i = 1, 2)$, and $\mathcal{L}_{\mathcal{F}}$ as in Definition 5. Then $\mathcal{L}$ is an algebraizable consequence system.

Theorem 7. Let $\mathcal{L}_i$ $(i = 1, 2)$, and $\mathcal{L}_{\mathcal{F}}$ as above. Then $\langle \mathcal{L}_{\mathcal{F}}, inc_1, inc_2 \rangle$ is the ALCO*-coproduct of $\mathcal{L}_1$ and $\mathcal{L}_2$, where $inc_i : \mathcal{L}_i \to \mathcal{L}_{\mathcal{F}}$ is the canonical injection $(i = 1, 2)$.

With respect to constrained fibring, we have:

Theorem 8. Let $N : ALCO^* \longrightarrow PLAN$ the forgetful functor. Then, N is a cofibration.

From theorems 7 and 8 we obtain the following:

Theorem 9. ALCO* has both constrained and unconstrained fibring.

3 Fibring Algebraizable Hilbert Systems

In this section we analyze the special case of algebraizable consequence systems generated by axioms and inference rules, that is, Hilbert-style systems. A first approach to this problem was outlined in [22].

Definition 10. (i) Given a signature C and a fixed countable set $\Xi = \{\xi_i\}_{i \in \mathbb{N}}$ disjoint of $\mathcal{V} \cup |C|$, the schematic propositional language relative to C, denoted by $L(C, \Xi)$, is the free algebra generated by $\mathcal{V} \cup \Xi$ over C.

[2] And, implicitly, it was proven that $\{p_1 \Rightarrow p_2, p_2 \Rightarrow p_1\}$ is not an equivalence set for P^1.
[3] As before, it is equivalent to require "for every equivalence sets Δ^i and Δ^j".

(ii) A schematic inference rule is a pair $\langle \Upsilon, \beta \rangle$ such that $\Upsilon \cup \{\beta\}$ is a finite subset of $L(C, \Xi)$.
(iii) A Hilbert system is a pair $\mathcal{H} = \langle C, P \rangle$ where C is a signature and P is a set of schematic inference rules
(iv) A function $\sigma : \Xi \to L(C)$ is called an instantiation. Any instantiation can be extended to a unique homomorphism $\widehat{\sigma} : L(C, \Xi) \to L(C)$.
(v) Given a Hilbert system $\mathcal{H} = \langle C, P \rangle$, the consequence system $\mathcal{L}_{\mathcal{H}} = \langle C, \vdash_{\mathcal{H}} \rangle$ induced by $\mathcal{H}$ is defined as follows: $\Gamma \vdash_{\mathcal{H}} \alpha$ iff there exists a finite sequence $\phi_1, ..., \phi_m$ $(m \geq 1)$ in $L(C)$ such that $\phi_m = \alpha$ and, for every $1 \leq i \leq m$,
(a) $\phi_i \in \Gamma$, or
(b) there is an instantiation σ and a schematic inference rule $\langle \Upsilon, \beta \rangle$ such that

$$\widehat{\sigma}(\beta) = \phi_i \text{ and } \widehat{\sigma}(\Upsilon) \subseteq \{\phi_1, ..., \phi_{i-1}\}.$$

In order to define the fibring in ALCO of Hilbert system, we need to generalize the schema language, allowing symbols for schema connectives.

Definition 11. (i) Given a signature C, fix a family of pairwise disjoint sets $\Theta = \{\Theta_n\}_{n \in \mathbb{N}}$ such that $\Theta_n \cap (|C| \cup \mathcal{V} \cup \Xi) = \emptyset$, for every $n \in \mathbb{N}$; elements in Θ_n are called n-ary schema connectives. The language $L(C, \Theta; \Xi)$ is the free algebra generated over $\mathcal{V} \cup \Xi$ by $\{C_n \cup \Theta_n\}_{n \in \mathbb{N}}$.
(ii) A realization is a partial function $\rho : \bigcup_{n \in \mathbb{N}} \Theta_n \to |C|$ such that, if $C_n \neq \emptyset$, then $\rho(\kappa) \in C_n$ for every $\kappa \in \Theta_n$.
(iii) Given an instantiation $\sigma : \Xi \to L(C)$ and a realization $\rho : \bigcup_{n \in \mathbb{N}} \Theta_n \to |C|$, the extended instantiation induced by ρ and σ is the function $\chi : L(C, \Theta; \Xi) \to L(C)$ defined as follows:
(a) $\chi(p) = p$ for every $p \in \mathcal{V}$;
(b) $\chi(\xi) = \sigma(\xi)$ for every $\xi \in \Xi$;
(c) $\chi(\kappa(\phi_1, \ldots, \phi_n)) = \rho(\kappa)(\chi(\phi_1), \ldots, \chi(\phi_n))$, if $\kappa \in \Theta_n$ and $\rho(\kappa)$ is defined;
(d) $\chi(\kappa(\phi_1, \ldots, \phi_n)) = p_0$, if $\rho(\kappa)$ is not defined.
(iv) An extended Hilbert-system is a pair $\mathcal{H} = \langle C, P \rangle$ such that C is a signature and P is a set of inference rules as in Definition 10(ii), but now using $L(C, \Theta; \Xi)$.
(v) An extended Hilbert-system $\mathcal{H}$ induces a consequence system $\mathcal{L}_{\mathcal{H}} = \langle C, \vdash_{\mathcal{H}} \rangle$ as in Definition 10(v), but now using $L(C, \Theta; \Xi)$ and extended instantiations.

Definition 12. (i) The category ALHI$^-$ is the full subcategory of ALCO whose objects are consequence systems $\mathcal{L} = \langle C, \vdash \rangle$ of ALCO such that $\vdash$ is induced by some Hilbert system $\mathcal{H}$.
(ii) The category ALHI is the full subcategory of ALCO whose objects are consequence systems $\mathcal{L} = \langle C, \vdash \rangle$ of ALCO such that $\vdash$ is induced by some extended Hilbert system $\mathcal{H}$.

In order to obtain a sufficient condition for the existence of unconstrained fibring in ALHI, we recall the following result, due to [1]:

Proposition 13. Suppose that a standard consequence system $\mathcal{L} = \langle C, \vdash \rangle$ has a set $\Delta = \{\Delta_i(p_1, p_2)\}_{i=1}^n$ of formulas such that, for every formula φ, ψ:
(i) $\vdash \varphi \Delta \varphi$;
(ii) $\varphi \Delta \psi \vdash \psi \Delta \varphi$;
(iii) $\varphi \Delta \psi$, $\psi \Delta \alpha \vdash \varphi \Delta \alpha$;
(iv) For every connective $c \in C_k$, and every formula $\varphi_1, \ldots, \varphi_k, \psi_1, \ldots, \psi_k$:
$\varphi_1 \Delta \psi_1, \ldots, \varphi_k \Delta \psi_k \vdash c(\varphi_1, \ldots, \varphi_k) \Delta c(\psi_1, \ldots, \psi_k)$;
(v) $\varphi, \varphi \Delta \psi \vdash \psi$;
(vi) $\varphi, \psi \vdash \varphi \Delta \psi$.
Then $\mathcal{L}$ is algebraizable.

Theorem 14. If $\mathcal{L}_1$ and $\mathcal{L}_2$ are logics in ALHI satisfying the conditions of Proposition 13, then there exists the unconstrained fibring (i.e., the coproduct) $\mathcal{L}_1 \bigoplus \mathcal{L}_2$ of $\mathcal{L}_1$ and $\mathcal{L}_2$ in ALHI.

With respect to ALHI$^-$, we also obtain a sufficient condition for the existence of fibring.

Definition 15. : Let $\mathcal{L} = \langle C, \vdash \rangle$ be an algebraizable consequence system, and let $\langle (\delta, \epsilon), \Delta \rangle$ be an algebraizator of $\mathcal{L}$. We say that $\mathcal{L}$ is equivalence-expressing if there is a formula $(p_1 \leftrightarrow p_2) \in L(C)$ such that, for every $\phi, \psi \in L(C)$: $\phi \Delta \psi \dashv\vdash \phi \leftrightarrow \psi$.

Proposition 16. Let $\mathcal{L}_i$ $(i = 1, 2)$ be two ALHI$^-$-objects, which are equivalence-expressing. Then, there exists the constrained fibring of $\mathcal{L}_1$ and $\mathcal{L}_2$ by sharing $\leftrightarrow$, where $\{\leftrightarrow\}$ is the identification of $\leftrightarrow_1$ and $\leftrightarrow_2$.

4 The Category $ASEM$ of Equivalent Algebraic Semantics

Based on [6], now we define the category of classes of algebras that are equivalent algebraic semantics for some algebraizable consequence system. We will use languages in PLAN to represent terms of quasivarieties.

Definition 17. (i) Let K be a quasivariety and let $\models_K$ be the associated consequence relation (cf. [1]). Given sets of terms $\delta = \{\delta_i(x)\}_{i\leq n}$, $\varepsilon = \{\varepsilon_i(x)\}_{i\leq n}$ and $\Delta = \{\Delta_j(x,y)\}_{j\leq m}$, we say that $[(\delta,\varepsilon),\Delta]$ is a deductivizator of K if it verifies: $x \approx y \dashv\models_K \varepsilon(x\Delta y) \approx \delta(x\Delta y)$.
(ii) Given a quasivariety K such that K has a deductivizator, we define a equivalence relation $\simeq_K$ between deductivizators as follows:
$[(\delta,\varepsilon),\Delta] \simeq_K [(\delta',\varepsilon'),\Delta']$ iff $\varepsilon(x) \approx \delta(x) \dashv\models_K \varepsilon'(x) \approx \delta'(x)$. The equivalence class of $[(\delta,\varepsilon),\Delta]$ under $\simeq_K$ will be denoted by $[\varepsilon,\delta,\Delta]_K$.

Definition 18. The category ASEM of equivalent algebraic semantics is the category whose objects are triples $\mathfrak{A} = \langle C, K, [\varepsilon,\delta,\Delta]_K\rangle$, where C is a signature, K a quasivariety which have a deductivizator, and $[\varepsilon,\delta,\Delta]_K$ is the equivalence class under $\simeq_K$ of some deductivizator $[(\delta,\varepsilon),\Delta]$ of K. Given objects $\mathfrak{A}_i = \langle C^i, K_i, [\varepsilon^i,\delta^i,\Delta^i]_{K_i}\rangle$ $(i = 1,2)$, a morphism $h : \mathfrak{A}_1 \to \mathfrak{A}_2$ is a PLAN-morphism $h : C^1 \to C^2$ satisfying the following:
(a) If $\Gamma \models_{K_1} (\nu \approx \eta)$ then $h(\Gamma) \models_{K_2} h(\nu \approx \eta)$, for every set $\Gamma \cup \{\nu \approx \eta\}$ of K_1-equations;
(b) For every $\tau \in L(C^1)$, $\varepsilon^2(h(\tau)) \approx \delta^2(h(\tau)) \dashv\models_{K_2} h(\varepsilon^1(\tau)) \approx h(\delta^1(\tau))$.[4]

The fundamental result of this section is the following:

Theorem 19. ALCO* and ASEM are isomophic categories.

As a corollary we obtain the following:

Corollary 20. ASEM has coproducts, and the functor $N : ASEM \longrightarrow PLAN$ which associates each equivalent algebraic semantics with its underlying signature is a cofibration.

From the previous corollary, we can obtain new equivalent algebraic semantics (sharing or not function symbols) from previous ones. This result suggest the implicit notion of "fibring of equational languages", modifying the definitions of the previous sections.

Acknowledgments

The first author was supported by a grant from CAPES, Brazil.

Bibliography

[1] W. Blok and D. Pigozzi. *Algebraizable Logics*, volume 77 (396) of *Memoirs of the American Mathematical Society*. AMS, Providence, Rhode Island, 1989.

[2] J. Bueno, M.E. Coniglio, and W.A. Carnielli. Finite algebraizability via possible-translations semantics. Submitted.

[3] C. Caleiro. *Combining Logics*. Tese de Doutorado. IST-Lisboa, 2000.

[4] V.L. Fernández and M.E. Coniglio. Syntactic fibring of algebraizable logics. *XIII meeting of the Brazilian Society of Logic (Abstract)*, pages 61–62, 2003.

[5] D. Gabbay. *Fibring Logics*. Oxford Science Publications, 1999.

[6] A. Jánossy, A. Kurucz, and Á. Eiben. Combining algebraizable logics. *The Notre Dame Journal of Formal Logic*, 37(2):336–380, 1996.

[7] R. Lewin, I. Mikenberg, and M.G. Schwarze. Algebraization of paraconsistent logic P^1. *The Journal of Non-Classical Logics*, 7(1/2):79–88, 1990.

[4] This clause is well-defined, that is, it does not depend on the equivalence class representatives.

[8] A. Sernadas, C. Sernadas, and C. Caleiro. Fibring of logics as a categorical construction. *Journal of Logic and Computation*, 9 (2):149–179, 1999.

[9] A.M. Sette. On the propositional calculus P^1. *Mathematica Japonicae*, 18:173–180, 1973.

[10] R. Wójcicki. *Theory of Logical Calculi*. Synthese Library. Kluwer Academic Publishers, 1988.

[11] A. Zanardo, A. Sernadas, and C. Sernadas. Fibring: Completeness preservation. *The Journal of Symbolic Logic*, 66(1):414–439, 2001.

Formalizing Concurrent Common Knowledge as Product of Modal Logics

Vania Costa Mário Benevides

System Engineering and Computer Science Program, COPPE
Federal University of Rio de Janeiro (UFRJ)
Mailbox 68511, 21945-970 Rio de Janeiro - RJ, Brazil
{vaniac,mario}@cos.ufrj.br

Abstract

This paper introduces a two-dimensional modal logic to represent agents' *concurrent common knowledge* in distributed systems. Unlike common knowledge, concurrent common knowledge is a kind of agreement reachable in asynchronous environments. As a proper semantics to concurrent common knowledge we present the *closed subproduct of modal logics*. We axiomatize the presented logic issuing an idea of the soundness and completeness proofs.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

The common knowledge is a present phenomenon in a lot of situations in our social life. To coordinate actions, to establish agreements and in other typical behaviors, the individuals need a previous knowledge or the mutual understanding or even the common knowledge of certain facts. The knowledge about the conventions among all the members in a community is an example of common knowledge, once, for every stipulated fact, everybody knows this fact, and everybody knows that everybody knows such fact, and everybody knows that everybody knows that everybody knows the fact, and so on. In Computer Science, the analysis and the applications of the common knowledge and other knowledge types became a very active research field, especially in the last two decades, giving rise to the epistemic logics or logics of knowledge. However, it is proved in [8] that common knowledge requires coordinated actions and simultaneity to be attained. Hence, common knowledge can not be achieved in asynchronous systems, because simultaneity is not applicable in such environments.

We propose a logic to represent other concepts of knowledge that can be achieved in asynchronous environments, such as the *concurrent common knowledge* [12]. To illustrate the concept of concurrent common knowledge, suppose that we are attending the final game of the Soccer World Championship and our country is one of the teams. We can suppose there is a small gap of time in the arrival of the images in televisions around the country, in other words, suppose that the images reach first some places and some time later other places. As soon as the victory goal happens, some places begin to celebrate the title, knowing that in all the country, sooner or later, everybody will know about the victory. In this case, the knowledge about the winner team is not simultaneous, but everybody knows that, in some moment sooner or later, all the others will know it. Thus, we say that the team's victory is concurrent common knowledge among all.

2 A Model for Asynchronous Distributed Systems

Consider a model for asynchronous distributed systems based on Lamport's definitions of time and causality [10]: time is given by causality relations among events and consistent global states are *consistent cuts* in an asynchronous run hypergraph. The model consists in: a network of *fifo* channels with m agents; a set R of asynchronous runs; a set E of events; a set C of consistent cuts.

The hypergraph in Figure 1 illustrates one possible run of the PIF (propagation of information with feedback) algorithm for 3 agents. The goal of PIF algorithm is to make the message $\mathcal{M}$ known to all the agents in the system, and, assuming that just one agent initiates the algorithm, to inform the initiator when $\mathcal{M}$ has reached all of them.

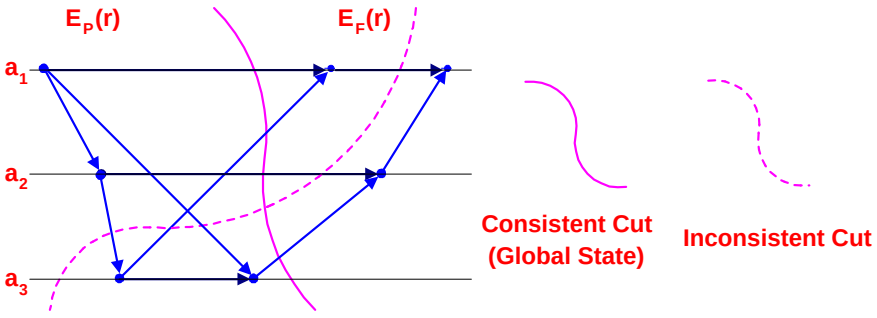


Figure 1. Consistent Cut

The dots represent **events** - when an agent sends and/or receives messages. The arrows establish a causality relation among events. A cut represents a global state and divides the graph into two sets of events, E_P and E_F, those which happen before (in the past of) and those which happen after (in the future of) the present cut. Intuitively, we can think about a **consistent cut** as a global state in which there are no messages from the future to the past.

In this model, an agent can not distinguish between two cuts if his local state is the same in both cuts. If so, the cuts are said to be *indistinguishable* according to the agent's point of view. There are distinct possible runs depending on the order in which messages reach the agents.

3 Products of Modal Logics

We think about asynchronous systems as a two-dimensional world. That is, taking into account the model of the previous section, we reason about the agents' knowledge under the perspective of a cartesian pair (r, c), a *run-cut* pair. In a modal logic approach, that means the interpretation of possible worlds are pairs (r, c) representing a state: a consistent cut c in an asynchronous run r.

The two-dimensional approach of knowledge can be formalized using the concept of *products of modal logics*. Like fibring, fusion, splitting and temporalization, which are forms to compose or decompose logics, products of modal logics is a technique to combine logics giving rise to many-dimensional or multidimensional logics. In multidimensional logics, the states are tuples representing dimensions where logical formulas are evaluated. The foundations of multidimensional logics are in Segerberg [13]. A complete overview on this subject and many results, including transference results between the whole and the component logics, were recently published in [6]. It follows some formal definitions and results on axiomatizing products of modal logics [14].

Definition 1. Let $F_1 = (W_1, R_1)$ and $F_2 = (W_2, R_2)$ be two propositional frames. The *product of frames* [14] is the frame $F_1 \times F_2 = (W_1 \times W_2, R_h, R_v)$, where:
$R_h = \{((x, z), (y, z))|xR_1y\}$ and $R_v = \{((z, x), (z, y))|xR_2y\}$.
Let L_1 and L_2 be modal logics, $\mathbf{F}(L_1)$ the class of frames validating L_1 and $\mathbf{F}(L_2)$ the class of frames validating L_2. The product of logics L_1 and L_2 is the logic $L_1 \times L_2 = \mathbf{L}(\mathbf{F}(L_1) \times \mathbf{F}(L_2))$ [1].

Definition 2. For L_1 n-modal and L_2 m-modal logics, let $[L_1, L_2] = L_1 * L_2 + C^1_{ij} + C^2_{ij}$, where:
$L_1 * L_2$ is the fusion of L_1 and L_2, $C^1_{ij} = (\Box_i\Box_{j+n}p \leftrightarrow \Box_{j+n}\Box_i p)$ and $C^2_{ij} = (\Diamond_i\Box_{j+n}p \rightarrow \Box_{j+n}\Diamond_i p)$, for $1 \leq i \leq n$, $1 \leq j \leq m$.
We say the logics L_1, L_2 are *commutative* if $L_1 \times L_2 = [L_1, L_2]$.

Definition 3. A modal formula is pseudo-transitive if it has the form:
$\bigtriangledown_1 \Box_k p \rightarrow \bigtriangleup_2 p$, where $p \in Prop$, $\bigtriangledown_1 = \Diamond_i, \ldots, \Diamond_j$, $\bigtriangleup_2 = \Box_i, \ldots, \Box_j$ are sequences of modal operators.
A PTC formula is a pseudo-transitive or closed formula.
A PTC logic is a modal logic axiomatized by PTC formulas.

Theorem 4. The logic resulting from the product of two PTC modal logics is *commutative* [14]. That is, if L_1 and L_2 are PTC then $L_1 \times L_2 = [L_1, L_2]$.

Many known modal logics are PTC, such as $\mathcal{D}, \mathcal{K}4, \mathcal{S}4, \mathcal{T}, \mathcal{B}, \mathcal{S}5$, and others. Thus, two-dimensional products like $\mathcal{T} \times \mathcal{T}$, $\mathcal{S}4 \times \mathcal{S}4$ or $\mathcal{S}5 \times \mathcal{S}5$ are *commutative*. We are interested, particularly, in the commutative product $\mathcal{S}5_m \times \mathcal{S}5_m$ which is used when axiomatizing our logic.

4 Semantics for Two-Dimensional Modal Logic of Knowledge

To model the desired two-dimensional knowledge approach we need a two-dimensional many-modal logic. The two dimensions refer to the runs' and cuts' dimensions, represented by the modal operators H_i and V_i, respectively. As usual, the semantics is based on Kripke's semantics of possible worlds, so we have possibility or accessibility relations for each dimension. These accessibility relations are equivalence relations, reflecting the concept of indistinguishable cuts and runs according to the agent's point of view. Hence, the accessibility relations are, in fact, equivalence relations for indistinguishability in each level of knowledge considered: the run dimension, the cut dimension, and a third relation for the transitive closure under the former. The closure relation gives us new features, for instance, representing knowledge properties according to indistinguishable pairs (r, c). Thus, the modal operator K_i related to the closure relation $\sim_i$represents the so-called agent's *concurrent knowledge*, that is, what he knows under indistinguishable consistent cuts in all possible runs.

We introduce the definition of *closed sub-product of modal logics* in order to formalize the kind of knowledge that we are interested in. The closed sub-product of modal logics is similar to the product, with two additional features: an extra relation for the transitive closure under the two basic relations, and a subset W_Δ of the cartesian product $R \times C$. To understand the set W_Δ, consider that there are some pairs (r, c) which, in fact, may not occur in the system. If so, we restrict the evaluation of the

[1] The modal logic $\mathbf{L}(\mathbf{F})$ for a class of frames $\mathbf{F}$ is defined as the intersection $\bigcap\{\mathbf{L}(F) \mid F \in \mathbf{F}\}$.

formulas to the so-called *reasonable* pairs, that is, the pairs (r, c) that really make sense. The subset $W_\Delta \subseteq R \times C$ denote these reasonable pairs. The idea is to make the modal operators H_i and V_i range only over the reasonable pairs in W_Δ, whereas the operators $\overline{H}_i$ and $\overline{V}_i$ range over the whole cartesian product $W = R \times C$.

Definition 5 (Closed Sub-product of Modal Logics). Let L_H be the smallest set of formulas containing the set of primitives $Prop_H$, closed under negation, conjunction and the modal operators H_i, $i = 1, \ldots, m$. Let L_V be the smallest set of formulas containing the set of primitives $Prop_V$, closed under negation, conjunction and the modal operators H_i, $i = 1, \ldots, m$.

Consider the frames $F_H = (R, \cong_i)$ and $F_V = (C, \asymp_j)$ for L_H and L_V, respectively. The *closed sub-product* of the frames F_H and F_V related to W_Δ is the frame $F_H \otimes F_V = (W, \simeq_i, \approx_j, \sim_k, W_\Delta)$, where:

1. $W = R \times C$ is the set of all the states (r, c);
2. $W_\Delta \subseteq W = R \times C$ is a subset of the states (r, c);
3. $\simeq_i = \{((r, c), (r', c)) | r \cong_i r'\}$;
4. $\approx_j = \{((r, c), (r, c')) | c \asymp_j c'\}$;
5. $\sim_k = (\simeq_i \cup \approx_j)$ *, where $(\simeq_i \cup \approx_j)$ * is the transitive closure under the union of $\simeq_i$ and $\approx_j$.

Let $\mathbf{F}(L_H)$ the class of frames validating L_H and $\mathbf{F}(L_V)$ the class of frames validating L_V. The *semantic sub-product* of the logics L_H and L_V is the logic $\mathbf{L}(\mathbf{F}(L_H) \otimes \mathbf{F}(L_V))$.

Definition 6 (Model for Closed Sub-product of Modal Logics). A *model* M over a closed sub-product frame $F = F_H \otimes F_V$ is a pair $M = (F, v)$, where v is a truth-value function for the primitive $Prop = Prop_H \cup Prop_V$. For each $p \in Prop$, $v(p)$ is the set of (r, c) where p is true, that is, $v(p) : Prop \to 2^{R \times C}$.

According to [12], to incorporate *concurrent common knowledge*, we need more three modalities:

. $P_i\alpha$ meaning "there is another consistent cut in the *same run* indistinguishable under the point of view of agent i where α is true". In our logic, the operator P_i is, in fact, the dual of V_i.

. $E_C\alpha$ meaning "everybody concurrently knows α", which is given by the formula $E_C\alpha = \bigwedge K_i P_i \alpha$.

. $C_C\alpha$ meaning "α is concurrent common knowledge". As usual, concurrent common knowledge implies that everybody concurrently knows α and everybody concurrently knows that everybody concurrently knows α and so on. Thus, $C_C\alpha$ is given by the formula $C_C\alpha \to E_C\alpha \wedge E_C^2\alpha \wedge E_C^3\alpha \wedge \ldots$.

We will use the same subscript i for the relations and modal operators, because we have m agents, and therefore, the product of two m-modal logics. It follows the formal semantics definitions.

Definition 7 (Satisfiability in L_m^2). Let L_m^2 be the smallest set of formulas containing Δ, the set of primitives $Prop = Prop_H \cup Prop_V$, closed under negation, conjunction and the modal operators $\overline{H}_i$, $\overline{V}_i$, K_i, E_C and C_C where $i = 1, \ldots, m$.

Suppose $\simeq_i$, $\approx_i$ and $\sim_i$ are equivalence relations in a closed sub-product of two modal frames, as defined in 5. Let $F = (W, \simeq_i, \approx_i, \sim_i, W_\Delta)$ be a frame for L_m^2 and let M be a model over F. A formula $\alpha \in L_m^2$ is true in $[M, (r, c)]$, $[M, (r, c)] \models \alpha$, for $(r, c) \in W = R \times C$, when: [2]

1. $[M, (r, c)] \models p \Leftrightarrow (r, c) \in v(p)$, where $p \in Prop$;
2. $[M, (r, c)] \models \alpha \wedge \beta \Leftrightarrow [M, (r, c)] \models \alpha$ and $[M, (r, c))] \models \beta$;
3. $[M, (r, c)] \models \neg\alpha \Leftrightarrow [M, (r, c)] \not\models \alpha$;
4. $[M, (r, c)] \models \overline{H}_i\alpha \Leftrightarrow \forall(r', c')\{((r, c) \simeq_i (r', c')) \Rightarrow [M, (r', c')] \models \alpha\}$;
5. $[M, (r, c)] \models \overline{V}_i\alpha \Leftrightarrow \forall(r', c')\{((r, c) \approx_i (r', c')) \Rightarrow [M, (r', c')] \models \alpha\}$;
6. $[M, (r, c)] \models \Delta \Leftrightarrow (r, c) \in W_\Delta \subseteq W = R \times C$;
7. $[M, (r, c)] \models H_i\alpha \Leftrightarrow [M, (r, c)] \models \Delta$ and $[M, (r, c)] \models \overline{H}_i\alpha$;
8. $[M, (r, c)] \models V_i\alpha \Leftrightarrow [M, (r, c)] \models \Delta$ and $[M, (r, c)] \models \overline{V}_i\alpha$;
9. $[M, (r, c)] \models Q_i\alpha \Leftrightarrow [M, (r, c)] \models H_i\alpha$ and $[M, (r, c)] \models V_i\alpha$;
10. $[M, (r, c)] \models K_i\alpha \Leftrightarrow [M, (r, c)] \models \Delta$ and $\forall(r', c')\{((r, c) \sim_i (r', c')) \Rightarrow [M, (r', c')] \models \alpha\}$;
11. $[M, (r, c)] \models P_i\alpha \Leftrightarrow [M, (r, c)] \models \neg V_i \neg\alpha$;
12. $[M, (r, c)] \models E_C\alpha \Leftrightarrow [M, (r, c)] \models \bigwedge K_i P_i \alpha$;
13. $[M, (r, c)] \models C_C\alpha \Leftrightarrow [M, (r, c)] \models E_C^k\alpha$ for all $k \geq 1$.

[2] Because of clarity we keep the semantic definitions for abbreviations such as $P_i\alpha$ and $E_C\alpha$

5 Axiomatic System $\mathcal{C}_m^2$

When the accessibility relations are equivalence relations, we know that logics as L_H and L_V are axiomatized by $\mathcal{S}5_m$ [8]. Furthermore, we know that the product $\mathcal{S}5_m \times \mathcal{S}5_m$ is commutative, and therefore, axiomatized by $[\mathcal{S}5_m, \mathcal{S}5_m]$, according to the theorem 4. Thus, we propose the system $\mathcal{C}_m^2$ in the table 13.1 as an axiomatization of the two-dimensional many-modal logic L_m^2.

The axioms 1 to 12 are axioms of $\mathcal{S}5_m$ for the horizontal and vertical dimensions, and also for the concurrent knowledge K_i. The axioms 13, 14 and 15 reflect the properties of the commutative product. The axioms 16 and 17 restrict the knowledge to the reasonable pairs. The axioms for concurrent knowledge K_i are 18, 19 and 20. The operator Q_i defined in axiom 18 is an auxiliary one which we call "inter-dimensional step". Remembering that P_i is the dual of V_i, axioms 21 and 22 define E_C, everybody concurrently knows. And finally, for concurrent common knowledge C_C, we have axiom 23.

Table 13.1. **System $\mathcal{C}_m^2$**

Axioms

1. $(\overline{H}_i\alpha \wedge \overline{H}_i(\alpha \to \beta)) \to \overline{H}_i\beta$	12. $\neg K_i\alpha \to K_i\neg K_i\alpha$
2. $\overline{H}_i\alpha \to \alpha$	13. $\overline{H}_i\overline{V}_j\alpha \leftrightarrow \overline{V}_j\overline{H}_i\alpha$
3. $\overline{H}_i\alpha \to \overline{H}_i\overline{H}_i\alpha$	14. $\neg\overline{H}_i\neg\overline{V}_j\alpha \to \overline{V}_j\neg\overline{H}_i\neg\alpha$
4. $\neg\overline{H}_i\alpha \to \overline{H}_i\neg\overline{H}_i\alpha$	15. $\neg\overline{V}_i\neg\overline{H}_j\alpha \to \overline{H}_j\neg\overline{V}_i\neg\alpha$
5. $(\overline{V}_i\alpha \wedge \overline{V}_i(\alpha \to \beta)) \to \overline{V}_i\beta$	16. $H_i\alpha \leftrightarrow \Delta \wedge \overline{H}_i\alpha$
6. $\overline{V}_i\alpha \to \alpha$	17. $V_i\alpha \leftrightarrow \Delta \wedge \overline{V}_i\alpha$
7. $\overline{V}_i\alpha \to \overline{V}_i\overline{V}_i\alpha$	18. $Q_i\alpha \leftrightarrow H_i\alpha \wedge V_i\alpha$
8. $\neg\overline{V}_i\alpha \to \overline{V}_i\neg\overline{V}_i\alpha$	19. $K_i\alpha \leftrightarrow Q_iK_i\alpha$
9. $(K_i\alpha \wedge K_i(\alpha \to \beta)) \to K_i\beta$	20. $K_i(\alpha \to Q_i\alpha) \to (\alpha \to K_i\alpha)$
10. $K_i\alpha \to \alpha$	21. $P_i\alpha \leftrightarrow \neg V_i\neg\alpha$
11. $K_i\alpha \to K_iK_i\alpha$	22. $E_C\alpha \leftrightarrow \bigwedge K_iP_i\alpha$
	23. $C_C\alpha \leftrightarrow E_C(\alpha \wedge C_C\alpha)$

Rules

R0. From $\vdash \alpha$ infer every uniform substitution for α
R1. From $\vdash \alpha, \alpha \to \beta$ infer β (*modus ponens*)
R2. From $\vdash \alpha$ infer $\overline{H}_i\alpha$ (*horizontal generalization*)
R3. From $\vdash \alpha$ infer $\overline{V}_i\alpha$ (*vertical generalization*)
R4. From $\vdash \alpha$ infer $K_i\alpha$ (*two-dimensional generalization*)
R5. From $\vdash \alpha \to E_C(\alpha \wedge \beta)$ infer $\alpha \to C_C\beta$ (*induction rule*)

where $i, j = 1, \ldots, m$.

Note: Axiom 10 can be deduced from axioms 19 and 20.

6 Conclusions

This work presents results on epistemic logics and many-dimensional logics with applications in the area of distributed multi-agent systems. We introduced the axiomatic system C_m^2 for concurrent common knowledge. The system is suitable to represent the properties of concurrent knowledge in distributed systems because the semantics is based on a model which considers consistent cuts and asynchronous runs to define time.

We have used the concept of multidimensional logics to deal with the two-dimensional approach of knowledge. Thus, the main contributions of this paper is in combination of logics in order to express the desired properties and the interactions among all the involved entities. The closed sub-product of modal logics was defined to make the necessary adjustments, resulting in a more powerful semantics.

As future developments, we would like to build a temporal version of the two-dimensional knowledge logic, which would better describe the evolution of knowledge acquisition over time.

Appendix

A Soundness and Completeness for $\mathcal{C}_m^2$

A.1 Soundness for $\mathcal{C}_m^2$

We prove soundness for system $\mathcal{C}_m^2$ with respect to the class of closed sub-product of modal frames for $\mathcal{S}5_m$. Thus, is it necessary to show that all the axioms of the system $\mathcal{C}_m^2$ are valid in this class of frames and the inference rules also preserve the validity. Let $\mathbf{F}$ be the class of closed sub-product of modal frames according to definition 5 and let M be a model over $F \in \mathbf{F}$. As $\simeq_i$, $\approx_i$ and $\sim_i$ are equivalence relations, the axioms 1 to 4 as well as the axioms 5 to 8 and 9 to 12 correspond to axioms from $\mathcal{S}5_m$, therefore the proofs can be found in [8]. As 13, 14, 15 are the axioms of Shehtman and Gabbay for the commutative product of logics, the soundness and completeness proofs can be found in [14]. For axioms 16, 17, 18, 21 and 22 soundness is straightforward from semantics rules 7, 8, 9, 11 and 12, respectively, in the satisfiability definition 7. The proofs for axioms 19 and 20 are not difficult and can be found in [4]. For the soundness proof of axiom 23, we propose a graph-theoretical characterization for concurrent common knowledge, as follows.

Definition 8. K_i-reachable, V_i-reachable, K_iV_i-reachable and KV-reachable in n KV-steps.
Let W_Δ be the set of states (r, c) such that $M, (r, c) \models \Delta$. Let $w, w', w'', w_n \in W_\Delta, n \geq 0$.
1. w' is K_i-reachable from w if and only if $w \sim_i w'$;
2. w' is V_i-reachable from w if and only if $w \approx_i w'$;
3. w' is K_iV_i-reachable from w if and only if there is a w'' such that $w \sim_i w''$ and $w'' \approx_i w'$;
4. w' is KV-reachable from w in n KV-steps if and only if there are $w_0, w_1, \ldots, w_n$ such that $w = w_0$, $w' = w_n$, and for all j, $0 \leq j \leq n-1$ we have w_{j+1} is K_iV_i-reachable from w_j, $i \in \{1, 2, \ldots, m\}$.

Proposition 9. Graph-theoretical Characterization for Concurrent Common Knowledge.
a) $M, w \models E_C\alpha$ iff, for $i \in \{1, 2, \ldots, m\}$, for all w' K_iV_i-reachable from w, $M, w' \models \alpha$;
b) $M, w \models E_C^j\alpha$ iff, for $i \in \{1, 2, \ldots, m\}$, for all w' KV-reachable from w in j KV-steps, $M, w' \models \alpha$;
c) $M, w \models C_C\alpha$ iff, for $i \in \{1, 2, \ldots, m\}$, for all w' KV-reachable from w in n KV-steps, for all $n > 0$, $M, w' \models \alpha$.

Proof for proposition 9: part *a*) follows from definition of E_C, everybody concurrently knows; considering that $M, w \models E_C^{j+1}\alpha \Leftrightarrow M, w \models E_C(E_C^j\alpha)$, part *b*) follows by induction on k; part *c*) is straightforward from *b*) and from the definition of concurrent common knowledge C_C.

Using proposition 9, it is easy to prove soundness for axiom 23: $M \models C_C\alpha \leftrightarrow E_C(\alpha \wedge C_C\alpha)$. For instance, for the direction ($\rightarrow$), suppose that $M, w \models C_C\alpha$. Thus, $M, w' \models \alpha$ for all w' KV-reachable from w in n KV-steps, $n > 0$. Particularly, if w'' is KV-reachable from w in one KV-step, we have $M, w'' \models \alpha$ and $M, w' \models \alpha$ for all w' KV-reachable from w'' in n KV-steps, $n > 0$. Therefore, $M, w'' \models \alpha \wedge C_C\alpha$ for all w'' KV-reachable from w in one KV-step. Hence, $M, w \models E_C(\alpha \wedge C_C\alpha)$. The proof for the converse direction is similar.

Regarding the inference rules of $\mathcal{C}_m^2$, it is easy to see that rules R1 to R4 preserve validity, and the proofs can be found in [4]. To prove soundness for rule R5, that is, to prove that if $M \models \alpha \rightarrow E_C(\alpha \wedge \beta)$, then $M \models \alpha \rightarrow C_C\beta$, we also use the graph-theoretical characterization of proposition 9. In fact, we show by induction on n, that for all w' KV-reachable from w in n KV-steps, $n > 0$, we have $M, w' \models \alpha \wedge \beta$, and, therefore, $M, w \models C_C(\alpha \wedge \beta)$. The complete proof is found in [4].

A.2 Completeness for $\mathcal{C}_m^2$

We prove completeness for $\mathcal{C}_m^2$ with respect to the class $\mathbf{F}$ of closed sub-product frames. Thus, it is necessary to show that every valid formula in the class $\mathbf{F}$ is a theorem from $\mathcal{C}_m^2$. Or, equivalently, we have to prove that for every formula φ $\mathcal{C}_m^2$-consistent there is a model based on a frame $F \in \mathbf{F}$ that satisfies φ. In [4] we build such finite models, that is, we prove that the system has the f.m.p. property, and therefore, as $\mathcal{C}_m^2$ is a finite axiomatization, we have, in addition, decidability. We also prove that the frames of such models are indeed frames in the class of closed sub-product frames $\mathbf{F}$.

The proof is standard, that is, the finite model is based on a frame $F^\varphi = (W^\varphi, \simeq_i^\varphi, \approx_i^\varphi, \sim_i^*, W_\Delta^\varphi)$ where $\varphi \in L_m^2$, W^φ contains all the φ-maximal $\mathcal{C}_m^2$-consistent sets, and the relations are defined as usual. The Truth Lemma is proved by induction on the length of the sub-formulas $\alpha \in Sub(\varphi)$. Consider, for

instance, that we want to prove $(M^{\varphi}, w) \models C_C\alpha \Rightarrow C_C\alpha \in w$. Suppose that $(M^{\varphi}, w) \models C_C\alpha$. As $w \in W^{\varphi}$ is φ-maximal $\mathcal{C}^2_m$-consistent, the conjunction of the sub-formulas in w is also a finite formula in L^2_m. Let $\hat{w}$ be the conjunction of the formulas in w. Consider the set $U = \{u \in W^{\varphi} | (M^{\varphi}, u) \models C_C\alpha\}$ of states which satisfy $C_C\alpha$. Let γ be the disjunction of such states, $\gamma = \bigvee_{u \in U} \hat{u}$. As U is finite, then γ is a formula of L^2_m and can be considered as the formula which characterizes the states where $C_C\alpha$ is true. Note that $\gamma \rightarrow E_C(\alpha \wedge \gamma)$ is $\mathcal{C}^2_m$-consistent and, therefore, we have $\vdash \gamma \rightarrow E_C(\alpha \wedge \gamma)$. By the induction rule R5, we have $\vdash \gamma \rightarrow C_C\alpha$. As $w \in U$, then $\vdash \hat{w} \rightarrow \gamma$, and thus $\vdash \hat{w} \rightarrow C_C\alpha$ (*). Hence, $C_C\alpha \in w$, otherwise $\neg C_C\alpha$ together with (*) would make w $\mathcal{C}^2_m$-inconsistent. The whole completeness proof, including the proof for the converse, is found in [4].

Bibliography

[1] R. J. Aumann. Agreeing to disagree. *Annals of Statistics*, 4(6):1236–1239, 1976.

[2] C. Caleiro, W. Carnielli, M. E. Coniglio, A. Sernadas, and C. Sernadas. Fibring non-truth-functional logics: Completeness preservation. *J. of Logic, Language and Information*, 12(2):183–211, 2003.

[3] M. E. Coniglio and W. Carnielli. Transfers between logics and their applications. *Studia Logica*, 72(3):367–400, 2002.

[4] V. Costa. *Uma Lógica Modal Bidimensional para Representação do Conhecimento em Sistemas Distribuídos Multiagentes*. PhD thesis, COPPE-UFRJ, Rio de Janeiro, Brazil, 2002.

[5] M. Finger and D. Gabbay. Combining temporal logic systems. *Notre Dame J. of Formal Logic*, 37(2):204–232, 1996.

[6] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: theory and applications*. Elsevier Science, 2003.

[7] J. Y. Halpern. Reasoning about knowledge: An overview. In *Proceedings of the 1st Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 1–17, Monterey, CA, U.S.A., 1986.

[8] J. Y. Halpern, R. Fagin, Y. Moses Y., and M. Y. Vardi. *Reasoning about knowledge*. MIT Press, 1995.

[9] J. Hintikka. *Knowledge and belief*. Cornell UP, Ithaca, NY, U.S.A., 1962.

[10] L. Lamport. Time, clocks and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[11] D. Lewis. *Convention*. Harvard University Press, Cambridge, MA, U.K., 1969.

[12] P. Panangaden and K. Taylor. Concurrent common knowledge: defining agreement for asynchronous systems. *Distributed Computing*, 6:73–93, 1992.

[13] K. Segerberg. Two-dimensional modal logics. *J. of Philosophical Logic*, 2:77–96, 1973.

[14] V. B. Shehtman and D. M. Gabbay. Products of modal logics, part 1. *Logical J. of the IGPL*, 6(1):73–146, 1998.

[15] F. Wolter. First order common knowledge logics. *Studia Logica*, 65:249–271, 2000.

Combining Possibility and Knowledge

Alexandre Costa-Leite

Institute of Logic, University of Neuchâtel, Neuchâtel - Switzerland
GTAL - UNICAMP - Brazil
{alexandre.costa,alexandrecl}@{unine.ch,cle.unicamp.br}

Abstract

This paper is an attempt to define a new modality with philosophical interest by combining the basic modal ingredients of possibility and knowledge. This combination is realized via product of modal frames so as to construct a *knowability* modality, which is a bidimensional constructor of arity one defined in a two-dimensional modal frame. A semantical interpretation for the operator is proposed, as well as an axiomatic system able to account for inferences related to this new modality. The resulting logic for knowability LK is shown to be sound and complete with respect to its class of modal-epistemic product models.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

According to [5], "as logic is being used more and more to formalise field problems in philosophy, language, artificial intelligence, logic programming and computer science, the kind of logics required become more and more complex". In this paper a particular method for combining logics is used to construct a logic able to express a philosophical concept as the concept of knowability, which has been a target of several articles related to the knowability paradox. This paradox is a modal argument which shows that given the monomodal logic KT extended with the verificationist principle $A \rightarrow \diamond KA$ it is possible to deduce $A \rightarrow KA$ causing the collapse of the knowledge modality. The motivation of this article is to propose a modal formalism able to account for the complex knowability modality which emerges in the presence of the verificationist principle. A natural conjecture underlying this proposal is that using the knowability modality it is possible to define an n-dimensional modal version of the logic KT where it is possible to add a bidimensional version of the verificationist principle without causing the collapse of the knowledge operator. This would be a nice solution to the trouble. Up to now, there is no definitive argument in this direction, but just a general clue.

The basic difference between the knowability modality and other usual modalities, in general terms, is that it is not possible to define knowability in monodimensional modal reality. Therefore, a knowability modality requires a two-dimensional environment defined by means of monomodal frames.

There are several different methods for combining modal logics: fibring [4] or synchronization as defined in [3] or, otherwise, one can use fusions [6] or products as defined in [5]. Also, given a particular method for combining logics, it is possible to give a categorial representation of the mechanism as in [2].

In this article a concrete case of product of modal frames is used to define a two-dimensional modal frame where it is possible to express a combined modality of *knowability*. In order to realize this task, the product of two modal frames (interpreted as alethic and epistemic, respectively) is defined as in [4]. Technical details of the construction will appear in the final version of the paper.

2 Bidimensional modal semantics for the knowability operator

It is well-known that the operation of product when applied to singular modal frames, defines many-dimensional modal frames. Many-dimensional frames are used to interpret modal languages for multi-dimensional modal logics [7]. In [4] there is the definition bellow, which is adequate for our account of knowability.

Definition 1. Consider two generalised frames:

$$F_1 = \langle W, R_1, ..., R_n \rangle$$
$$F_2 = \langle S, P_1, ..., P_n \rangle$$

Then, a *simple product of generalized frames* is defined as

$$F_1 \times F_2 = \langle W \times S, R'_1, ..., R'_n, P'_1, ..., P'_n \rangle$$

such that:

$$R'_i = \{\langle \langle x, y \rangle, \langle z, y \rangle \rangle : xR_iz, y \in S\}$$
$$P'_i = \{\langle \langle x, y \rangle, \langle x, z \rangle \rangle : yP_iz, x \in W\}$$

The notion of product can also be applied to classes of frames and logics.

We argue that it is reasonable to introduce bidimensional modal semantics to formalize the knowability operator: using a concrete case of the definition 1, we obtain:

Definition 2. Given two frames

$$F_1 = \langle W, R \rangle$$
$$F_2 = \langle S, P \rangle$$

where $F_1 = \langle W, R\rangle$ is interpreted as an alethic frame such that W is a set of possible worlds and R is an accessibility relation between worlds, while $F_2 = \langle S, P\rangle$ is interpreted as an epistemic frame composed by a set S of epistemic states and a plausibility relation P between epistemic states. The *simple product of a modal and an epistemic frame* is defined as:

$$F_1 \times F_2 = \langle W \times S, R', P'\rangle$$

where:

$$\langle w, s\rangle R' \langle w', s\rangle \text{ iff } wRw' \text{ and } s \in S$$
$$\langle w, s\rangle P' \langle w, s'\rangle \text{ iff } sPs' \text{ and } w \in W$$

Clearly, just single-agent frames are considered here. The elements of $W \times S$ are called *modal-epistemic states* while R' and P' are called *two-dimensional accessibility relations for knowability*. The basic difference between alethic and epistemic frames is that the accessibility relation does not have the notion of agents indexed to it. In this sense it is impossible to use accessibility relations and plausibility relations in the same way . Using the intuition to model 2-dimensional possibility and knowledge, we can suppose that we have just one operator called knowability.

Given a product of a modal and an epistemic frames, add in the usual way a valuation v to the frame in order to obtain a model. Then the formal semantics for the two knowability operators $\boxminus_L$ and $\boxminus_G$ are defined as:

Definition 3 (Local Knowability).
$\langle w, s\rangle \vDash \boxminus_L p$ iff $\exists w'(wRw' \wedge \langle w', s\rangle \vDash p)$ and $\forall s'(sPs' \Rightarrow \langle w', s'\rangle \vDash p)$.

Definition 4 (Global Knowability).
$\langle w, s\rangle \vDash \boxminus_G p$ iff $\exists w'(wRw' \wedge \langle w', s\rangle \vDash p)$ and $(w' \vDash p)$ and $\forall s'(sPs' \Rightarrow \langle w', s'\rangle \vDash p)$ and $(s' \vDash p)$.

Adding to the two above clauses the following bidimensional classical valuations to connectives:
$\langle w, s\rangle \vDash A$ iff $\langle w, s\rangle \in v(A)$, for A atomic.
$\langle w, s\rangle \vDash \neg p$ iff $\langle w, s\rangle \nvDash p$
$\langle w, s\rangle \vDash p \wedge q$ iff $\langle w, s\rangle \vDash p$ and $\langle w, s\rangle \vDash q$
$\langle w, s\rangle \vDash p \vee q$ iff $\langle w, s\rangle \vDash p$ or $\langle w, s\rangle \vDash q$
$\langle w, s\rangle \vDash p \rightarrow q$ iff $\langle w, s\rangle \nvDash p$ or $\langle w, s\rangle \vDash q$

These valuations are used to show that in each point $\langle w, s\rangle$ it is possible to reason classically.

The knowability operator is introduced here with the aims to modeling the concept of "it is possible to know "without using two modalities, but using instead just one complex modality. The global and local notions of knowability express the central property that agents may have different levels of knowledge, meaning that a proposition can be known in more than a single way.

It is important to note that the local knowability modality could be defined using 2-dimensional possibility and knowledge, but the same is not the case related to global knowability, which could be defined using both 2-dimensional modalities and 1-dimensional modalities. The basic difference between local and global knowability is that in the local case there are no interaction axioms between modalities of different dimensions, while in the global knowability it is indeed possible to have interaction between operators of different contexts. The next natural step is to find an axiomatization characterizing the two above semantical levels of knowability.

3 Axiomatic system for LK

A particular problem which arises in questions about combining modalities is that exposed in [5]: how to find an axiomatic system for a class of frames? An axiomatic system is proposed here in order to axiomatize the logic for the local knowability and global knowability. It is important to note that all these axioms could be represented in a powerset simple logic presentation, which is a better way to represent modal logics than simple logic system presentation, as showed in [3]. Note that the signature Σ_n of the LK is $\Sigma_1 = \{\boxminus_G, \boxminus_L, \neg\}$, for n=1.

Definition 5. The *axiomatic system for the logic of knowability* LK is:
Axioms
1. All tautologies from propositional classical logic;
2. $\boxminus_*(p \to q) \to (\boxminus_* p \to \boxminus_* q)$, $* \in \{L, G\}$
3. $\boxminus_G p \to \boxminus_L p$
Inference Rules:
4. MP
5. $\vdash p$ then $\vdash *p$, for $* \in \{\boxminus_L, \boxminus_G\}$

The axiomatic system above does not recognize in its language either $\diamond$ or K, but somehow conveys a common abstraction from such concepts. Both $\boxminus_L, \boxminus_G$ distributes over conjunction as the proof-system and the semantic show, given that it is possible to maintain a substantial part of classical reasoning for them.

4 Completeness result and transfer properties

A completeness proof is developed for the logic LK using the tools of [7] with some appropriate modifications in the notion of a perfect matrix.

Lemma 6. Let Γ be a set of formulas. Γ is satisfiable iff there is a $\boxminus$-perfect matrix for Γ.

Lemma 7. Let Γ be a set of formulas. Γ is consistent iff there is a $\boxminus$-perfect matrix for Γ.

In order to prove the two lemmas, we should make several adaptations in the methods and techniques proposed in [7], but saving the basic idea of the proof. In the following there is a general view on the completeness procedure: the first one establishes a relation between matrices and semantics. The second lemma is a bridge linking matrices and axioms. To prove the first lemma, define a matrix and show that it is, in fact, a $\boxminus$-perfect matrix for the set Γ. Soundness is a consequence of the fact that the function in the matrix assigns to each pair $\langle w, s\rangle$ a maximal consistent set. To prove the other direction we need to transform the $\boxminus$-perfect matrix into a model and then, by induction, to prove the truth lemma. To prove the second lemma, we must show that Γ is contained in a maximal consistent set. The other direction is the most difficult part. Given a consistent set, how is it possible to find its $\boxminus$-perfect matrix? The idea is to construct a matrix which should be a $\boxminus$-perfect matrix (without defects). Technical details of the proof will be given in the final version of the paper.

As a consequence of the two lemmas:

Theorem 8. LK is sound and complete with respect to its class of modal-epistemic product models.

In the scope of transferring theorems among logics it is already known that, if L_1 and L_2 are canonical logics, then the semantical product is also canonical. Given that every canonical logic is Kripke-complete, and given that the product logic is canonical, the desired result follows immediately. If the product is obtained from logics with known properties, then a general proof is also obtained by transferring properties of the given logics. This is not the case here, given that we do not know which are the logics used in the combination.

5 Conclusion

Methods for combining logics are an important tendency, as they are useful in the task of finding powerful logics able to map natural language (although the existence of the collapsing problem related to the most powerful mechanism for combining logics: fibring) . Such methods also have many applications in fields varying from philosophy to computer science. It is important to note that the process of combining, for example, two logical objects depends strongly on the nature of these objects. This means that we must start making a selection of a particular case of structure. In this sense, combining logics constitutes a chapter of something called Universal Logic, as defined in [1]. To illustrate these last three mysterious sentences, let me make reference to two articles: [2] and [3]. In [2], the authors show how to give categorical descriptions of methods, or mechanisms, for combining logics. But to realize this task, they must first choose a particular kind of structure: signatures, hilbertian calculi or interpretation systems for creating

the categories SIG, HIL and INT in order to represent the mechanisms in categorical terms. The same happens in [3], but now in order to define synchronization, parameterization and fibring it is necessary first to choose a particular kind of structure: consequence systems, simple logic systems, powerset logic systems, etc. This present article shows a way in which it is possible to define a bidimensional modal logic with respect to a knowability modality, without defining the kind of logic system associated, but, instead, departing from a syntactical approach by means of frames. The above construction is a clue that we should enter in the world of multi-dimensional modal logics and combination of logics to realize philosophical tasks.

In [4] some known logics are shown to be particular cases of fibring. Would it be possible to find two already known logics (one alethic and other epistemic, for example) showing that the logic LK can be obtained by some method of combination from these logics?

The problem of examining how to obtain products of particular modal logics with epistemic logics is another task, as it is the question of understanding further properties related to our construction. Results concerning families of logics for knowability are still under investigation, but the task seems to be promising.

Bibliography

[1] J-Y. Beziau. *Researchs on Universal Logic - excessivity, negation and sequents*. PhD thesis, Universite Denis Diderot, Paris 7, 1995.

[2] C. Caleiro, C. Sernadas, and A. Sernadas. Fibring of logics as a categorial construction. *J. Logic Computation*, 9(2):149–179, 1999.

[3] C. Caleiro, C. Sernadas, and A. Sernadas. Mechanisms for combining logics. Research report, Section of Computer Science, Department of Mathematics, Instituto Superior Tecnico, 1049-001 Lisboa, Portugal, 1999.

[4] Dov M. Gabbay. *Fibring Logics*. Oxford University Press, Clarendon Press, 1999.

[5] Dov. M. Gabbay and Valentin B. Shehtman. Products of modal logics. *Log. J. IGPL*, 6(1):73–146, 1998.

[6] Marcus Kracht and Frank Wolter. Properties of independently axiomatizable bimodal logics. *J. Symbolic Logic*, 56(4):1469–1485, 1991.

[7] Maarten Marx and Yde Venema. *Multi-dimensional Modal Logic*, volume 4 of *Applied Logic Series*. Kluwer Academic Publishers, Dordrecht, 1997.

Fusions of Normal and Non-normal Modal Logics

Marcelo Finger

Department of Computer Science, University of São Paulo, Brazil
mfinger@ime.usp.br

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

The fusion of two modal logics $\mathbf{M}_1$ and $\mathbf{M}_2$ is the smallest logic system that contains both $\mathbf{M}_1$ and $\mathbf{M}_2$. This combined logic system is represented by $\mathbf{M}_1 \otimes \mathbf{M}_2$, but in the literature it is also found as $\mathbf{M}_1 \circ \mathbf{M}_2$. The two modal logics $\mathbf{M}_1$ and $\mathbf{M}_2$ are assumed to share the same set of propositional atoms $\mathcal{P} = \{p_0, p_1, \ldots\}$ and the boolean connectives. The modal symbols of $\mathbf{M}_1$ and $\mathbf{M}_2$ are assumed to be distinct; if one is fusing a logic with itself, the combined system has two distinct set of copies of the modal symbols of the original system.

In this setting, fusion is a form of combining logics that can be seen as a restricted form of fibring logics [9, 16].

The study of fusion in particular, and of combination of logics in general, focuses on the *transference* of logical properties from the component logics to the combined system. Among the most studied properties are: soundness, completeness, decidability, compactness, finite model property, interpolation, etc.

If the component modal logics $\mathbf{M}_1$ and $\mathbf{M}_2$ are presented in terms of axiomatisations, the inference system of the fusion $\mathbf{M}_1 \otimes \mathbf{M}_2$ is simply obtained by taking the union of the two axiomatisations, with the care of renaming modal symbols appropriately if necessary. The study of this sort of independent axiomatisation is found in the literature since the work of Thomason [13], where the independent axiomatisation is shown to be a conservative extension of the two component modal systems.

A more systematic study of the transference of logical properties from the component logics to the combined system started with the works of Kracht and Wolter [12] and Fine and Schurz [6]. Both works considered only normal monomodal logics. The work of Kracht and Wolter, which perhaps coined the term *fusion*, dealt with the combination of only two modal logics, but considered the transference of a number of logical properties. The work of Fine and Schurz considered the fusion of any finite number of independent modal systems, but covered the transference of fewer logical properties.

The next step in the study of the fusion of modal logics was to extend such results to the fusion of normal, multimodal logic systems, where each modality could be an n-ary modality, $n \geq 1$. In this setting, each modal logic is given by its language, its inference system, a class of models and a semantic relation between formulas and models, which can be described as follows:

- There is a set of propositional symbols $\mathcal{P}$ which consist of *atomic formulas*.
- Each modality Δ_i in modal logic $\mathbf{M}$ is associated to a number of arguments $\text{arity}(\Delta_i) = n \geq 1$, so that if $A_1, \ldots A_n$ are well formed formulas, so is $\Delta_i(A_1, \ldots, A_n)$.
- The inference system contains a set of axioms, and the inference rules of Modus Ponens, Substitutivity and Normalisation.
- There is a class of models $\mathcal{K}$ and a semantic relation $\models$ such that for each model $\mathcal{M} \in \mathcal{K}$ and each element $w \in \mathcal{M}$ and each formula A, one can check if the formula A holds in the model $\mathcal{M}$ at that point w, that is, $\mathcal{M}, w \models A$.

Then the fusion of two modal logics will also define a language, an inference system, a class of models and a semantic relation, in the following way:

- The common propositional symbols are atomic formulas in the fusion.
- Each connective of the component system are present in the fusion, with the same formation rule.
- The inference system of the fusion is the union of the axiomatisations.
- If class of models $\mathcal{K}_1$ has models of the form $(W_1, R_1, \ldots R_{a_1})$ and class of models $\mathcal{K}_2$ has models of the form $(W_2, S_1, \ldots S_{a_2})$, the class of models of the fusion will have models of the form $(W, R_1, \ldots R_{a_1}, S_1, \ldots S_{a_2})$, and the semantic relation is defined using the rules of the component system.

In this setting, a problem arises on how to define *normality*. In the case of monomodal logics, where each modality is a unary $\Box$, it suffices that the system satisfies one of the versions of the normality axiom:

1. $\Box(p \to q) \to (\Box p \to \Box q)$; or
2. $\Box(p \wedge q) \leftrightarrow (\Box p \wedge \Box q)$; or

3. $\Diamond(p \vee q) \leftrightarrow (\Diamond p \vee \Diamond q)$

plus the normalisation inference rule $\frac{\vdash A}{\vdash \Box A}$ or its axiomatic correspondent, the validity of $\Box\top$.

However, when this notion of normality is extended to n-ary modalities, several distinct definition of normality are possible, generating fusions with more or less expressivity.

2 The Syntactic Approach to Normality

The syntactic approach was developed by Wolter [15]. It generalises the notion of normality above to n-ary multimodalities in the following way. If Δ is an n-ary modality, then for every argument position i, $1 \leq i \leq n$:

- $\vdash \neg\Delta(\ldots, \bot_i, \ldots)$;
- $\vdash \Delta(\ldots, A \vee B_i, \ldots) \leftrightarrow \Delta(\ldots, A_i, \ldots) \vee \Delta(\ldots, B_i, \ldots)$.

Note that Δ behaves like a diamond operator in version 3 of the definition of normality above. Such a definition of normality suited very well the algebraic approach employed by Wolter, and in [15] it was shown that soundness completeness and decidability transfer for normal modal systems that obey the restrictions above.

This approach relied heavily on the syntactical definition of normality and it did not suggest any obvious way to generalise the transference of logical properties to non-normal modalities.

It turns out that there are n-ary modalities in well known modal/temporal logics that were considered normal when they were proposed, but that do not fit in the definition above. One case in hand is Kamp's temporal operators *until* (U) and *since* (S) [11]. These binary connectives have an existential semantic in one argument and a universal semantic in the other; the latter fails the syntactic definition above. So a different approach to normality could be investigated.

3 The Semantic Approach to Normality

The semantic (or syntactic/semantic) approach to normality in the fusion of modal/temporal logics was presented in [8]; an earlier proposal in [5] dealt only with linear temporal logics.

This approach ignores how many arguments a modality can have. Instead, it focuses on the n-relational model in the underlying semantics. Suppose $(W, R_1, \ldots, R_n)$ is an acceptable model. The logic is normal according to the semantic approach if every R_i is associated with a definable connective $\Box_i$ that is normal, that is, $\vdash \Box_i(p \to q) \to (\Box_i p \to \Box_i q)$ and $\frac{\vdash A}{\vdash \Box_i A}$ are valid in the logic for every i, $1 \leq i \leq n$.

Note that $\Box_i$ does not have to be a primitive symbol in the language; it suffices that it be *definable* from the primitive n-ary connectives. This definition is strictly stronger than the previous one: every logic that is normal in the syntactic approach is clearly normal in this approach, and there are modal logics such as Kamp's US-temporal logic that are normal in this approach but not in the former one.

In [8], it was shown that fusion of n-ary multimodal logics according to the semantic approach do transfer the basic properties of soundness, completeness and decidability. One has to note that Wolter [1] provided a definition of *quasi-normal* modal logics that somehow overlapped with the semantic approach, and showed the transference of logical properties in the fusion of these systems.

Unlike the syntactic approach, which is algebraic, the semantic approach to the fusion of normal modal logics is based on Kripke semantics, and its proof strategy in the demonstration of transference could, in principle, be applied in the investigation of the fusion of non-normal modal logics.

4 Towards Fusion of Non-normal Modal Logics

The proof strategy for showing the transference of logic properties in the fusion of two logics according to the semantic approach consisted in three steps:

1. The *external application* of a modal logic system $\mathbf{M}$ to a generic logic $\mathbf{L}$, generating $\mathbf{M}(\mathbf{L})$, in a process called temporalisation [4] or modalisation [7]. The modalisation is a form of combining logics weaker than the fusion, and one proves it transfers logic properties.

2. The study of finite *iterated modalisations* of two modal logic systems $\mathbf{M}_1$ and $\mathbf{M}_2$, generating $\mathbf{M}_1(\mathbf{M}_2(\mathbf{M}_1(\ldots)))$. The aim here is to show that an outer application of $\mathbf{M}_1$ does not conflict with an inner application of $\mathbf{M}_1$, such that the resulting system transfers logic properties.

3. The view of fusion as the union of iterated modalisations, such that every formula in a fusion $\mathbf{M}_1 \otimes \mathbf{M}_2$ can be seen as a formula in some iterated modalisation. The transfer of logic properties follows from this fact and some additional considerations.

In the case of non-normal modal logic, the approach corresponding to Kripke semantics is a semantic based on *minimal models* [2], where a model is now given by (W, F, V), where $F : \mathcal{W} \rightarrow 2^{2^P}$ maps each point $w \in W$ to a set of sets of proposition, namely the set of propositions A such that $\Box A$ is true at w. On the proof-theoretic side, no axioms are required to hold in general, the only obligatory inference rule is the rule of *congruence*, namely $\frac{\vdash A \leftrightarrow B}{\vdash \Box A \leftrightarrow \Box B}$.

According to the proof strategy displayed above, the first step in the investigation of the fusion of non-normal modal logics was the external application (modalisation) of a non-normal modal logic $\mathbf{M}$ to a generic logic $\mathbf{L}$, generating $\mathbf{M}(\mathbf{L})$. In [3], it was shown that the non-normal modalisation preserves soundness, completeness and decidability.

The surprising result occurs when one tries to prove the transference of logic properties in the case of iterated modalisations $\mathbf{M}_1(\mathbf{M}_2(\mathbf{M}_1(\ldots)))$. It turns out that non-normal modal logics behave in ways not expected in normal modal logics.

The fusion of two modal logics, being the smallest logic that contain both its components, is not expected to contain any form of interaction between the two components. Forms of interactions do arise in the *product* of two modal logics [10], that satisfy the commutativity of two modalities, namely,

$$\vdash \Box_1 \Box_2 A \leftrightarrow \Box_2 \Box_1 A$$

and the convergence axiom

$$\vdash \Diamond_1 \Box_2 \rightarrow \Box_2 \Diamond_1 A$$

Note, however, that these two axioms do not always characterise the product of two logics, for the product does not in general transfer completeness and decidability [14].

In the fusion of two normal modal logics such interaction never occurs. Not so in the fusion of non-normal modal logics. To see that, consider a non-normal modal logic $\mathbf{M}_P$ containing the single axiom of *partition*.

$$(P) \quad \vdash_{\mathbf{M}_P} (\Box p \leftrightarrow p) \vee (\Box q \leftrightarrow \neg q)$$

In a normal modal logic we have $\Box \top \leftrightarrow \top$ so axiom P trivialises to $\Box p \leftrightarrow p$, which means that the modality can be eliminated.

Now consider the class $\mathcal{K}_P$ of minimal frames (W, F) such that for every $w \in W$, either:

1. $F(w) = \{X \subseteq W | w \in X\}$; or
2. $F(w) = \{X \subseteq W | w \notin X\}$.

We show (in an unpublished work with Rogerio Fajardo) that $\mathbf{M}_P$ is correct and complete with respect to $\mathcal{K}_P$. Furthermore, in $\mathbf{M}_P$ we show that $\Box$ and $\Diamond$ can be switched, that is, for every formula A.

$$\vdash_{\mathbf{M}_P} \Box A \leftrightarrow \Diamond A$$

Now consider the fusion of $\mathbf{M}_P$ with itself, $\mathbf{M}_{P1} \otimes \mathbf{M}_{P2}$, with modalities $\Box_1$ and $\Box_2$. We prove that $\mathbf{M}_{P1} \otimes \mathbf{M}_{P2}$satisfies the commutativity and confluence properties, that is, for every formula A:

$$\vdash_{\mathbf{M}_{P1} \otimes \mathbf{M}_{P2}} \Box_1 \Box_2 A \leftrightarrow \Box_2 \Box_1 A \qquad \vdash_{\mathbf{M}_{P1} \otimes \mathbf{M}_{P2}} \Diamond_1 \Box_2 \rightarrow \Box_2 \Diamond_1 A$$

We call this phenomenon a *strong interaction* arising in the fusion of two non-normal modal logics. This brings us problems in considering a formula in a fusion as a formula in an iterated modalisation.

To see that, consider the formula $\Box_1 \Box_2 p \wedge \Box_2 \Box_1 \neg p$. From the commutativity above we see that such a formula is $\mathbf{M}_{P1} \otimes \mathbf{M}_{P2}$-inconsistent. However, if we consider such formula as an iterated modalisation $\mathbf{M_P}_1(\mathbf{M_P}_2(\mathbf{M_P}_1))$, it is possible to build a model for it. The fact that an inconsistent formula is mapped into a consistent one precludes the view that considers fusion as a set of iterated modalisations.

Note however, that the logic $\mathbf{M}_{P1} \otimes \mathbf{M}_{P2}$ is *not* a counterexample for the transference of logic properties in the fusion of two non-normal modal logics.

5 Conclusion

We have shown how different definitions of normality may lead to stronger forms of fusions of modal logics.

In the case of the fusion of non-normal modal logics, we have shown that it is possible that there are surprisingly strong interactions between the modalities of the logics, which precludes the direct use of the proof strategies used for showing the transference of logic properties in the fusion.

Perhaps a different strategy to map a formula in a fusion to a formulas in an iterated modalisation could be explored, or a fusion of logics that respect the axiom $\vdash \Box\top$ could be shown to transfer the logic properties using the old strategy.

Bibliography

[1] F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research (JAIR)*, 16:1–58, 2002.

[2] B. F. Chellas. *Modal Logic — an Introduction*. Cambridge University Press, 1980.

[3] R. Fajardo and M. Finger. Non-normal modalisation. In *Advances in Modal Logic*, pages 316–325, Toulouse, France, 2002.

[4] M. Finger and D. Gabbay. Adding a Temporal Dimension to a Logic System. *Journal of Logic Language and Information*, 1:203–233, 1992.

[5] M. Finger and D. Gabbay. Combining Temporal Logic Systems. *Notre Dame Journal of Formal Logic*, 37(2):204–232, Spring 1996.

[6] K. Fine and G. Schurz. Transfer theorems for stratified multimodal logics. In J. Copeland, editor, *Logic and Reality: Proceedings of the Arthur Prior Memorial Conference*, pages 169–213. Cambridge University Press, 1996.

[7] M. Finger and M. A. Weiss. The unrestricted addition of a temporal dimension to a logic system. In *3rd International Conference on Temporal Logic (ICTL2000)*, Leipzig, Germany, 4–7 October 2000.

[8] M. Finger and M. A. Weiss. The unrestricted combination of temporal logic systems. *Logic Journal of the IGPL*, 10(2):165–190, March 2002.

[9] D. Gabbay. *Fibring logics*. Oxford University Press, 1999.

[10] D. Gabbay and V. Shehtman. Products of modal logics, part 1. *Logic Journal of the IGPL*, 6(1):73–146, 1998.

[11] H. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, UCLA, 1968.

[12] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4):1469–1485, 1991.

[13] S. K. Thomason. Independent Propositional Modal Logics. *Studia Logica*, 39:143–144, 1980.

[14] Y. Venema. *Many-Dimensional Modal Logic*. PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1991.

[15] F. Wolter. Fusions of modal logics revisited. In Marcus Kracht, Maarten de Rijke, Heinrich Wansing, and Michael Zakharyaschev, editors, *Advances in Modal Logic*, volume Volume 1 of *Lecture Notes 87*, pages 361–379. CSLI Publications, Stanford, CA, 1996.

[16] A. Zanardo, A. Sernadas, and C. Sernadas. Fibring: Completeness preservation. *Journal of Symbolic Logic*, 66(1):414–439, 2001.

Possible-Translations Semantics

João Marcos

CLC, Department of Mathematics, IST, Lisbon, Portugal
IFCH, Universidade Estadual de Campinas, Brazil
Center for Exact Sciences, UnilesteMG, Brazil
http://www.geocities.com/jm_logica/

Abstract

This text aims at providing a bird's eye view of possible-translations semantics ([10, 24]), defined, developed and illustrated as a very comprehensive formalism for obtaining or for representing semantics for all sorts of logics. With that tool, a wide class of complex logics will very naturally turn out to be (de)composable by way of some suitable combination of simpler logics. Several examples will be mentioned, and some related special cases of possible-translations semantics, among which are society semantics and non-deterministic semantics, will also be surveyed.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Logics, translations, possible-translations

Let a *logic* $\mathcal{L}$ be a structure of the form $\langle \mathcal{S}, \Vdash \rangle$, where $\mathcal{S}$ denotes its *language* (its set of *formulas*) and $\Vdash \subseteq \mathsf{Pow}(\mathcal{S}) \times \mathsf{Pow}(\mathcal{S})$ represents its associated *consequence relation* (cr), somehow defined so as to embed some formal model of reasoning. Call any subset of $\mathcal{S}$ a *theory*. As usual, capital Greek letters will denote theories, and lowercase Greek will denote formulas; a sequence such as $\Gamma, \alpha, \Gamma' \Vdash \Delta', \beta, \Delta$ should be read as asserting that $\Gamma \cup \{\alpha\} \cup \Gamma' \Vdash \Delta' \cup \{\beta\} \cup \Delta$.

Morphisms between any two of the above structures will be called *translations*. So, given any two logics, $\mathcal{L}_1 = \langle \mathcal{S}_1, \Vdash_1 \rangle$ and $\mathcal{L}_2 = \langle \mathcal{S}_2, \Vdash_2 \rangle$, a mapping $t : \mathcal{S}_1 \to \mathcal{S}_2$ will constitute a translation from $\mathcal{L}_1$ into $\mathcal{L}_2$ just in case the following holds:

(T1) $\Gamma \Vdash_1 \Delta \Rightarrow t(\Gamma) \Vdash_2 t(\Delta)$

A translation is said to be *conservative* in case the converse of (T1) holds, i.e.:

(T2) $\Gamma \Vdash_1 \Delta \Leftarrow t(\Gamma) \Vdash_2 t(\Delta)$

Given a logic $\mathcal{L} = \langle \mathcal{S}, \Vdash \rangle$, a *possible-translations representation* (ptr) over it is a structure of the form $\langle \mathsf{Log}, \mathsf{Tr}, \mathsf{Reg} \rangle$, where $\mathsf{Log} = \{\langle \mathcal{S}_j, \Vdash_j \rangle\}_{j \in J}$ is an indexed set of logics (also called *factors* or *ingredients* of this ptr), $\mathsf{Tr} = \{t_j : \mathcal{S} \to \mathcal{S}_j\}_{j \in J}$ is an indexed set of translations, and $\mathsf{Reg} \subseteq \mathsf{Pow}(\mathsf{Tr})$. To any such ptr one can immediately associate three levels of consequence relations: A *local* pt-cr, $\Vdash^j_{\mathsf{pt}} \subseteq \mathsf{Pow}(\mathcal{S}) \times \mathsf{Pow}(\mathcal{S})$, for each $t_j \in \mathsf{Tr}$, a *regional* pt-cr, $\Vdash^R_{\mathsf{pt}} \subseteq \mathsf{Pow}(\mathcal{S}) \times \mathsf{Pow}(\mathcal{S})$, for each $R \in \mathsf{Reg}$, and a *global* pt-cr, $\Vdash_{\mathsf{pt}} \subseteq \mathsf{Pow}(\mathcal{S}) \times \mathsf{Pow}(\mathcal{S})$. These relations will be defined by setting:

(L-pt) $\Gamma \Vdash^j_{\mathsf{pt}} \Delta$ iff $t_j(\Gamma) \Vdash_j t_j(\Delta)$

(R-pt) $\Gamma \Vdash^R_{\mathsf{pt}} \Delta$ iff $(\ni t_j \in R)[\Gamma \Vdash^j_{\mathsf{pt}} \Delta]$,
where $\ni$ is some (generalized) quantifier

(G-pt) $\Gamma \Vdash_{\mathsf{pt}} \Delta$ iff $(\forall R \in \mathsf{Reg})[\Gamma \Vdash^R_{\mathsf{pt}} \Delta]$

Obviously, (L-pt) is just a particular case of (R-pt). Taking $\mathsf{Reg} = \{\{t_j\} : t_j \in \mathsf{Tr}\}$ makes the regional pt-cr perfectly dispensable —we will call any ptr with that characteristics a *simple* ptr and write it more simply as $\langle \mathsf{Log}, \mathsf{Tr} \rangle$. There are usually many ways of obtaining the same global pt-cr. Suppose for instance that '$\ni = \forall$' in (R-pt). Then, $\Vdash_{\mathsf{pt}}$ will be exactly the same, for every Reg such that $\bigcup \mathsf{Reg} \supseteq \mathsf{Tr}$.

Given two logics $\mathcal{L}_1 = \langle \mathcal{S}_1, \Vdash_1 \rangle$ and $\mathcal{L}_2 = \langle \mathcal{S}_2, \Vdash_2 \rangle$, we will say that $\mathcal{L}_1$ is *sound* with respect to $\mathcal{L}_2$ in case $\Vdash_1 \subseteq \Vdash_2$. Similarly, we will say that $\mathcal{L}_1$ is *complete* with respect to $\mathcal{L}_2$ in case $\Vdash_1 \supseteq \Vdash_2$. Notice that translations can be endomorphisms. In particular, any logic is sound and complete with respect to itself, the identity endomorphism always constituting thus a trifling example of a ptr. A ptr over a logic $\mathcal{L} = \langle \mathcal{S}, \Vdash \rangle$ is said to be *adequate* in case $\mathcal{L}$ is sound and complete with respect to $\langle \mathcal{S}, \Vdash_{\mathsf{pt}} \rangle$. Thus, an adequate ptr can be seen as a way of combining a set of translations so as to obtain a very particular conservative translation. Finally, a *possible-translations semantics* (pts) is simply a possible-translations representation in which all factors are defined by 'semantic means' (in contrast to, say, 'abstract deductive' or 'proof-theoretical' means). This characterization certainly looks very vague, but I will show in more detail in the following subsections how the canonical semantic notions work and how they can be seen as special cases of simple pts, according to the above definitions.

One last methodological discrimination is sometimes useful. In case one starts with a logic $\mathcal{L}$ and then finds a set of factors for it in an adequate ptr, one will call the process *splitting logics*; in case one starts with the factors and then build a logic for which the corresponding ptr is adequate, the process will be called *splicing logics*. The immense majority of examples from the literature on *combining logics* is of a more synthetic character: More and more logics are spliced as time goes by. Here, on the contrary, it will be often natural to use ptr's in order to analyze some given logics, splitting them into simpler components in order to understand them. *Frango ut patefaciam.*

Digression 1. (*Categorial*) If one considers the category where logics are the objects and translations are the arrows, the diagrams we get for the ptr's all look like there were sunbeams irradiating from a common core. The logic that originates from the combination can be seen as the colimit of this diagram. In [11] the authors show how to generalize this construction for arbitrary diagrams. This should be compared to what is done in [30] in understanding *fibring* (a more general form of combination, check [23, 4]) as a categorial construction. A first advance in that direction, generalizing the basic construction of fibring, can be found in [16]. A different semantically driven generalization of fibring, *cryptofibring*, is categorially investigated in [7].

Digression 2. (*Historical*) Possible-translations semantics were first introduced in [9], restricted to the use of finite-valued factors. The embryo was then frozen for a period, and in between 1997 and 1998 it was publicized under the denomination 'non-deterministic semantics', in [12], and in several talks by Carnielli and a few by myself. Noticing that the non-deterministic element was but a particular accessory of the more general picture, from 1999 on the semantics retook its earlier denomination ([10, 24, 14, 15, 26]).

1.1 What is a logic?

This is a question that will *not* be answered in this section. Any number of answers to it can be found in the literature, if you dig hard enough. I will here instead show how some among the most popular answers can be recast in the present framework.

Given a logic $\mathcal{L} = \langle \mathcal{S}, \Vdash \rangle$ as above, we will call it *scottian* in case its cr is subject to the following restrictions (cf. [28]):

(C1)	$(\Gamma, \varphi \Vdash \varphi, \Delta)$	(overlap)
(C2)	$(\Gamma \Vdash \varphi, \Delta)$ and $(\Gamma', \varphi \Vdash \Delta') \Rightarrow (\Gamma', \Gamma \Vdash \Delta, \Delta')$	(cut)
(C3)	$(\Gamma \Vdash \Delta) \Rightarrow (\Gamma', \Gamma \Vdash \Delta, \Delta')$	(dilution)

Call any clause of the form $\Gamma \Vdash \Delta$ an *inference*. Theories that appear at the left-hand side of the $\Vdash$ are also dubbed *countertheories*, or *premises* assumed by the inference; theories that appear at the right-hand side of the $\Vdash$ are also called *alternatives* sanctioned by the inference. A *tarskian* cr (cf. [32]) is a particular case of a scottian cr, in which each inference has a single formula as alternative (no real 'alternative' in that case, is it?). Such alternative is often called *consequence* of the inference. Tarskian logics are also called *single-conclusion*, in contrast to the more symmetrical (*multiple-premise*) *multiple-conclusion* scottian logics. It would be just as natural, of course, to consider here a *countertarskian* logic to be defined by the same restrictions above, but on a single-premise-multiple-conclusion environment. Very uncommon in practice, the countertarskian case works pretty much like the tarskian case in most circumstances. Below I will only mention countertarskian logics explicitly, thus, when relevant.

Here are some degenerate examples of logics. Let a logic $\langle \mathcal{S}, \Vdash \rangle$ be called *overcomplete* in case its cr is characterized by one of the following universal properties:

(C0.0.0)	$(\Gamma \Vdash \Delta)$	(triviality)
(C0.0.1)	$(\Gamma, \alpha \Vdash \Delta)$	(nihilism)
(C0.1.0)	$(\Gamma \Vdash \beta, \Delta)$	(dadaism)
(C0.1.1)	$(\Gamma, \alpha \Vdash \beta, \Delta)$	(semitriviality)

Note, by the way, that THE trivial logic is characterized by the nonproper cr over the language $\mathcal{S}$. Clearly, tarskian logics must identify trivial and dadaistic logics, and identify nihilistic and semitrivial logics. When we talk about THE dadaistic logic in a given language we will be referring to the logic having a non-trivial dadaistic cr. Similarly, THE nihilistic logic will refer to the logic having a non-trivial nihilistic cr, and THE semitrivial logic will denote the logic having a non-dadaistic non-nihilistic cr.

A formula β of a logic $\mathcal{L}$ is said to be a *thesis* of this logic in case $(\Gamma \Vdash \beta, \Delta)$, for any choice of Γ and Δ; an *antithesis* of this logic is any formula α such that $(\Gamma, \alpha \Vdash \beta)$, for any choice of Γ and Δ. An arbitrary thesis is sometimes denoted by $\top$, and an arbitrary antithesis is sometimes denoted by $\bot$.

Theorem 3. (i) Every multiple-conclusion overcomplete logic is scottian. Every single-conclusion overcomplete logic is tarskian.
(ii) The empty language defines a unique scottian / tarskian logic.
(iii) Any arbitrary intersection of scottian / tarkian logics defined over some fixed language defines a scottian / tarskian logic.

Theorem 4. Fix some scottian / tarskian logic $\mathcal{L}$ over some non-empty language $\mathcal{S}$. Then:
(i) $\mathcal{L}$ is the trivial logic iff there is at least one formula in its language which is both a thesis and an antithesis of $\mathcal{L}$.
(ii) $\mathcal{L}$ is the nihilistic logic iff all of its formulas are antitheses of it.
(iii) $\mathcal{L}$ is the dadaistic logic iff all of its formulas are theses of it.
(iv) $\mathcal{L}$ is the semitrivial logic iff any formula implies any other (or the same) formula, but no antitheses nor theses are present in the language of this logic.

Several other restrictions and extensions of the above notion of logic are studied in [25], from an abstract viewpoint. As in that paper, a logic here will be called *minimally decent* in case it is not overcomplete.

1.2 What is the canonical notion of entailment?

Let $\mathcal{V}$ denote an arbitrary set of *truth-values*, where $\mathcal{D}^{\mathcal{V}} \subseteq \mathcal{V}$ denotes its subset of *designated* values (the 'true truth-values'), and $\mathcal{U}^{\mathcal{V}} = \mathcal{V} \setminus \mathcal{D}^{\mathcal{V}}$ denotes its subset of *undesignated* values (the 'false truth-values'). Given a language $\mathcal{S}$, let a *valuation* over it be any mapping $\S^{\mathcal{V}} : \mathcal{S} \to \mathcal{V}$. Call any collection of valuations over $\mathcal{S}$ a (*scottian*) *semantics* sem over $\mathcal{S}$. This semantics will be called *κ-valued* if κ is the greatest cardinality of truth-values of the valuations in sem, that is, $\kappa = \mathsf{sup}_{\S^{\mathcal{V}} \in \mathsf{sem}}(|\mathcal{V}|)$. To any valuation $\S^{\mathcal{V}}$ and any semantics sem one can associate *canonical* notions of *local entailment*, $\models^{\S}_{\mathsf{sem}} \subseteq \mathsf{Pow}(\mathcal{S}) \times \mathsf{Pow}(\mathcal{S})$ and *global entailment*, $\models_{\mathsf{sem}} \subseteq \mathsf{Pow}(\mathcal{S}) \times \mathsf{Pow}(\mathcal{S})$, by setting:

(L-ce) $\Gamma \models^{\S^{\mathcal{V}}}_{\mathsf{sem}} \Delta$ iff $(\S^{\mathcal{V}}(\Gamma) \cap \mathcal{U}^{\mathcal{V}} \neq \varnothing$ or $\S^{\mathcal{V}}(\Delta) \cap \mathcal{D}^{\mathcal{V}} \neq \varnothing)$

(G-ce) $\Gamma \models_{\mathsf{sem}} \Delta$ iff $(\forall \S^{\mathcal{V}} \in \mathsf{sem})[\Gamma \models^{\S^{\mathcal{V}}}_{\mathsf{sem}} \Delta]$

An *ordinary* scottian semantics is one in which a fixed cardinal of designated / undesignated values is set throughout all the valuations of the semantics. Obviously, any semantics can be made ordinary by just adding to each valuation a convenient number of truth-values that will not be used. Similarly to above, a *tarskian* (*ordinary*) *κ-valued semantics* will be defined just like a scottian (ordinary) κ-valued semantics, only that all inferences will have exactly one formula at their right-hand sides.

Theorem 5. (i) Any scottian / tarskian κ-valued semantics induces at least one scottian / tarskian logic by way of one of its associated canonical entailment relations.
(ii) Consider any covering of the valuations of a given scottian / tarskian semantics. Each layer of the covering can now be said to determine a new (universal) 'regional semantics', and the intersection of all the entailments associated to the latter gives you back the global entailment.

Given the above results, one sees that any semantic structure of the form $\langle \mathcal{S}, \models \rangle$ defines a scottian and a tarskian logic, and the logics corresponding to the global entailment relation can be obtained through the intersection of all local (or regional) entailment relations. As before, given a logic $\mathcal{L} = \langle \mathcal{S}, \Vdash \rangle$ and a semantics sem over $\mathcal{S}$, one can now very naturally talk about $\mathcal{L}$ being *locally sound* with respect to some $\S \in \mathsf{sem}$ in case $\Vdash \subseteq \models^{\S}_{\mathsf{sem}}$, and being *globally sound* with respect to sem in case $\Vdash \subseteq \models_{\mathsf{sem}}$. Similarly for local and global completeness and adequacy. The statement of the following result parallels that of Theorem 4.

Theorem 6. Here is how you can obtain adequate ordinary semantics for each sort of overcomplete logic:
(i) For the trivial logic, consider the empty semantics (empty set of truth-values).
(ii) For the nihilistic logic, consider some semantics whose valuations make everything false.
(iii) For the dadaistic logic, consider some semantics whose valuations make everything true.
(iv) For the semitrivial logic, consider some semantics whose valuations either make everything true or make everything false.

1.3 What can be done with translations between logics?

The general definitions of translation and of conservative translation that you found at the beginning of the present section were studied in detail in [12, 19], and interesting specializations of these notions were proposed in [20]. Typical examples of everyday translations are given by the endomorphisms that define uniform substitutions in a logic whose language is a free algebra (of formulas). One can here also easily check that:

Theorem 7. (i) A logic can always be conservatively translated into itself.
(ii) To check soundness or completeness of a given logic with respect to some scottian / tarskian semantics amounts to checking the identity mapping from the language into itself to be a translation.

Here are some degenerate examples of translations:

Theorem 8. For logics (not necessarily scottian nor tarskian) over some fixed language $\mathcal{S}$:
(i) Any logic is translatable into the trivial logic.
(ii) Any single-conclusion logic is translatable into any logic having a thesis. Any single-premise logic is translatable into any logic having an antithesis.
(iii) The trivial single-conclusion logic is conservatively translatable into any logic having a thesis. The trivial single-premise logic is conservatively translatable into any logic having an antithesis. The semitrivial logic is conservatively translatable into any logic having a thesis but no antitheses, or having an antithesis but no theses.
(iv) Given a logic with no (anti)theses at all, NO logic having a(n anti)thesis whatever is translatable into the former.
(v) Any logic having no theses nor antitheses is translatable into the semitrivial logic.

Problem: For more esoteric non-scottian logics, such as non-monotonic logics and other context-dependent applications it might seem more natural to work with a definition of translation that directly involves the inferences, instead of the formulas. In that case, a translation from $\langle \mathcal{S}_1, \Vdash_1 \rangle$ into $\langle \mathcal{S}_2, \Vdash_2 \rangle$ had better be defined, say, as a mapping $t : \mathsf{Pow}(\mathcal{S}_1) \to \mathsf{Pow}(\mathcal{S}_2)$ instead of $t : \mathcal{S}_1 \to \mathcal{S}_2$, as before. It might be better as well to think of a logic directly as a set of theories, instead of a set of formulas, endowed with a consequence relation. The properties of these sorts of definitions are yet to be investigated in more detail. An advance in that direction was already made in [17], where the authors conceive tarskian logics as two-sorted first-order structures (the sort of 'formulas' and the sort of 'theories'), and talk about 'transfers' as morphisms among those structures (of which translations between tarskian logics, in the above sense, are but particular cases).

1.4 What are possible-translations semantics?

We have defined above the notion of a possible-translations representation (ptr) based on the combination of a collection of factors through local ($\Vdash^{j}_{\mathsf{pt}}$), regional ($\Vdash^{R}_{\mathsf{pt}}$) and global ($\Vdash_{\mathsf{pt}}$) consequence relations (cr). A possible-translations semantics (pts) was then characterized as a ptr based on factors defined by 'semantic means'. Moreover, the above sections have shown a conventional rendering of the received notion of 'semantics', slightly generalized in accordance with the principles of the theory of valuations (cf. [18]) and of abstract multiple-conclusion deductive systems (cf. [33, 31]).

There are several ways of combining logics. In a very pleasant paper, [3], Blackburn and de Rijke survey the reasons one might have for splicing logics, and propose a catalogue of the forms of combination based on the increasing level of involvement of the ingredient logics: They come up with nice pictures for 'refining structures', then 'classification structures', then 'totally fibred structures'. Another taxonomy is proposed in [8, 4, 29], where 'synchronization' and 'parameterization' appear as distinguished special cases of 'fibring'. How would the general picture for the combination through a possible-translations representation look like?

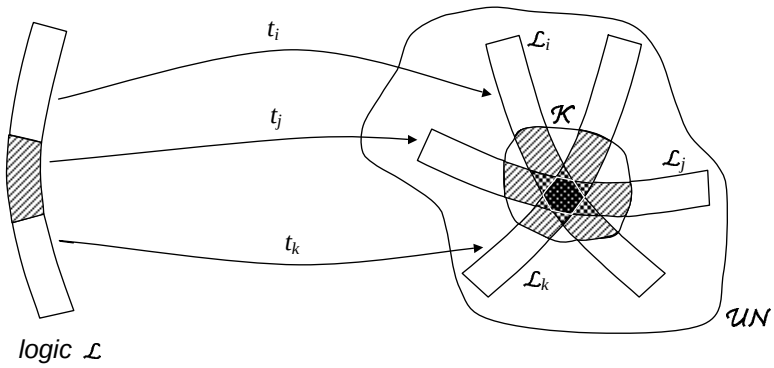


Figure 1. The logical Rosetta Stone.

An insightful analogy may be provided by concentrating on the situation in which a logic is split into its simpler components and comparing it to the deciphering of the 'Rosetta Stone' (cf. [15]). Carved in 196B.C. and found by Napoleon troops in July 1799 near the homonymous village (Rashid) located in

the western delta of the Nile, the Rosetta Stone is a basalt slab containing three different inscriptions of a text written by a group of priests to honor the Egyptian pharaoh. Why is it important? Because it finally allowed scholars to decipher the Hyerogliphic writing, a problem that had been open for several hundred years! After the work of Thomas Young, a British physicist, and Jean-François Champollion, a French Egyptologist, the code was finally broken, and a phonetic value was attached to hyeroglyphs that had previously been thought to have a purely symbolic value. How was it done? The three scripts in the stone were the Hyeroglyphic (used for important or religious documents), the Demotic (everyday Egyptian script) and the Greek (language of the rulers of Egypt at that time). With the aid of both Greek and Coptic (language of the Christian descendants of the ancient Egyptians), Champollion was able to decipher the Demotic writing, and from that he was able to trace back the meaning of the Hyerogliphic signs. But how did they know that the three scripts represented the same text, to start with? Because the stone *said so*, at the very end of its Greek inscription! Another beautiful example of self-reference, therefore.

Based on the above story, Figure 1 gives a schematic illustration of what is going on when a ptr is designed. The Rosetta Stone is the 'logical universe' $\mathcal{UN}$ where all ingredient logics can be found, resembling perhaps an egg with the sunny side up. The long curved format of the logic represents the form of reasoning sanctioned by it. You can see that the morphisms (possible-translations) are intended to preserve that format. At a distinguished hachured region of each logic you may find its circumstantial theses and antitheses. Each translation should in particular take theses into theses, and antitheses into antitheses. The region where they can be found in $\mathcal{UN}$ is at its yolk $\mathcal{K}$. The appetizing part is the one in which the ingredients are cooked together so as to give us the corresponding possible-translations structure.

The next result shows some simple examples of ptr and pts:

Theorem 9. (i) Any logic has an adequate possible-translations representation.
(ii) Any (scottian / tarskian) semantics can be seen as a possible-translations semantics with any positive number of factors.

One can count now on a more sophisticated interplay between local and global notions at hand: If a scottian / tarskian semantics can be seen as a general way of gluing arbitrary collections of valuations, a possible-translations semantics can be seen as a more general way of gluing collections of any arbitrary kind of previously given semantics.

Call a semantics *unitary* in case it is defined by way of a single valuation, or a single factor; call it *large* in case the cardinality of the set of valuations or the set of factors is at least as big as the cardinality of the underlying language. Obviously, any unitary semantics is ordinary from its very inception; unitary semantics can be made large, and large semantics can always be made ordinary at request, by the addition of redundant valuations or truth-values. We already knew from Theorem 5(ii) than any scottian / tarskian semantics can be reduced to the intersection of unitary scottian / tarskian semantics; the last result above suggests now that any semantics can ultimately and quite naturally be converted into a large possible-translations semantics whose factors are all unitary semantics themselves.

Moreover:

Theorem 10. If you are talking about logics characterized by scottian / tarskian entailments, or by simple possible-translations representations:
(i) Global soundness implies local soundness.
(ii) Local completeness implies global completeness.
In overcomplete logics:
(iii) Local soundness automatically transfers to global soundness.
(iii) Global completeness automatically transfers to local completeness.

1.5 Which logics have adequate semantics?

Right now we have two things called *scottian*: The abstract consequence relations characterized by way of clauses (C1)–(C3) in subsection 1.1 and the semantics to which canonical entailment relations were associated in subsection 1.2. A similar thing can be said about abstract tarskian consequence relations and tarskian semantics. The attentive reader will certainly have noticed, though, that we did not establish a relation between the homonymous creatures! This subsection will correct this slip for the benefit of the interested.

Consider first the tarskian case. Given a single-conclusion logic $\langle \mathcal{S}, \Vdash \rangle$ and a counter-theory $\Pi \subseteq \mathcal{S}$, the *right-closure* of Π, denoted by $\Pi^{\mathbf{c}}$, is the set of all of its derived consequences, that is, the set $\{\pi : \Pi \Vdash \pi\}$.

Theorem 11. (i) In any tarskian logic, $\Pi^{\mathbf{cc}} = \Pi^{\mathbf{c}}$, that is, $\Pi^{\mathbf{c}} \Vdash \pi \Leftrightarrow \Pi \Vdash \pi$.
(ii) In any tarskian logic $\langle \mathcal{S}, \Vdash \rangle$, given arbitrary $\Sigma \cup \Gamma \cup \{\varphi\} \subseteq \mathcal{S}$, to check whether $\Sigma, \Delta \Vdash \varphi$ holds is equivalent to checking whether $(\forall \gamma \in \Gamma)\Sigma \Vdash \gamma$ implies $\Sigma \Vdash \varphi$.

Theorem 12. (*Lindenbaum-like*) Each tarskian logic has at least as many (but no less than one) sound tarskian unitary semantics as the number of its right-closed theories.

Theorem 13. (*Wójcicki-like*) Every tarskian logic has an adequate semantics.

Corollary 14. Every tarskian logic $\langle \mathcal{S}, \Vdash \rangle$ has an adequate ordinary κ-valued semantics, with $\kappa \leq |\mathcal{S}|$.

The previous result is very general, but a κ-valued semantics is more interesting in case its truth-values are well-behaved with respect to the underlying language, for instance, in case one can count on truth-functionality. The contrast between designated and undesignated values casts though a shadow of *bivalence*. Indeed:

Theorem 15. (*Suszko-like*) Every tarskian logic has an adequate κ-valued tarskian semantics, for $\kappa \leq 2$.

Everything can be easily dualized to the counter-tarskian case. Only that now, given a single-premise logic $\langle \mathcal{S}, \Vdash \rangle$ and a theory $\Pi \subseteq \mathcal{S}$, you had better work with the *left-closure* of Π, denoted by ${}^{\mathbf{c}}\Pi$, as the set of all of its deriving premises, that is, the set $\{\pi : \pi \Vdash \Pi\}$. The rest is straightforward to adapt.

I will now briefly show how the above constructions can be modified for the scottian case (cf. [31]). As usual, call $\langle \Sigma, \Pi \rangle$ a *partition* of the set $\Theta \subseteq \mathcal{S}$ in case $\Sigma \cup \Pi = \Theta$ and $\Sigma \cap \Pi = \varnothing$.

Theorem 16. (*Cut for sets*) Given a scottian logic $\langle \mathcal{S}, \Vdash \rangle$:
If $\Gamma, \Sigma \Vdash \Pi, \Delta$, for every partition $\langle \Sigma, \Pi \rangle$ of Θ then $\Gamma \Vdash \Delta$.

Theorem 17. (*L-theorem*) Each scottian logic has some sound scottian unitary semantics.

Theorem 18. (*W-theorem*) Any scottian logic has an adequate semantics.

Corollary 19. Every scottian logic $\langle \mathcal{S}, \Vdash \rangle$ has an adequate ordinary κ-valued semantics, with $\kappa \leq |\mathcal{S}|$.

Theorem 20. (*S-theorem*) Every scottian logic has an adequate κ-valued scottian semantics, for $\kappa \leq 2$.

One can conclude from the above results that:

Theorem 21. (i) Every tarskian / scottian logic has an adequate possible-translations semantics, in fact even a possible-translations semantics based on 2-valued factors (copies of classical logic).
(ii) The local and the global consequence relations associated to any simple possible-translations representation or possible-translations semantics based on tarskian / scottian factors is tarskian / scottian.

It is noteworthy that the above results for canonical semantics have pretty much the same flavor of a pts: Each unitary semantics can be seen as determining a translation, and the intersection of all of the appropriate unitary semantics in each case gives you the desired conservative translation.

2 Further illustrations

We have seen, in the previous section, that every scottian / tarskian logic has an adequate scottian / tarskian (2-valued) semantics. Moreover, any logic (scottian, tarskian, or not) has an adequate possible-translations representation (ptr), and if it has an adequate semantics (scottian, tarskian, or not) then it can be given an adequate possible-translations semantics (pts).

What about other less trivial examples of possible-translations semantics, not obtained by plain use of brute force, as above? Indeed, notice that the previous adequacy results were often either uninformative (when a logic was used to represent itself) or non-constructive (when a κ-valued semantics was posited but no recursive method was presented so as to define it). The situation can be improved in a some cases. In the case of sufficiently expressive finite-valued truth-functional logics, for instance, a constructive method can be designed for the specification of a recursive set of clauses that describe the 2-valued semantics announced by Theorem 15 (cf. [6, 5]).

Moreover, to get even more concrete, one can use a ptr to provide, say, a pts based on a couple of well-behaved and well-known finite-valued truth-functional factors for logics having NO adequate finite-valued scottian / tarskian truth-functional semantics, as done in [10, 24, 14, 26] for several paraconsistent and paracomplete logics. Also, deductive limits for infinite hierarchies of logics can very naturally be spliced, and decidability transferred from the factors to the product, as in [24, 14]. Moreover, truth-functional finite-valued logics can themselves be split in terms of 2-valued logics, that is, fragments of classical logic ([24, 27]), copies of classical logic can be combined into fragments of modal logics, and so on and so forth.

The final version of the paper will display a few representative such examples in detail.

3 Some other related semantic structures

The advantage of possible-translations semantics lies in its generality. It is no overstatement to assert that pretty much anything that one might want to call a semantics can be recast in the present framework. This leads us immediately to the main disadvantage of possible-translations semantics: its generality! Anything that is universally true can easily turn out to be also universally irrelevant. It is very important thus to characterize some interesting subclasses of possible-translations semantics, defined by stricter terms. Clauses restricting the set of translations or the factors involved are often helpful, often inevitable. With that in mind, *society semantics* ([13, 24, 21, 22]), *dyadic semantics* ([6, 5]), and (dynamic and static) *non-deterministic semantics* ([2, 1]) can all be precisely characterized as specialized forms of possible-translations semantics.

This will be done in detail in the final version of the paper.

Acknowledgments

The author wishes to acknowledge the partial support of FCT (Portugal) and FEDER (European Union), namely, via the Project FibLog POCTI / MAT / 37239 / 2001 of the CLC (IST, Portugal) and the FCT PhD grant SFRH / BD / 8825 / 2002. I am also grateful to Arnon Avron, Beata Konikowska, Carlos Caleiro, and Walter Carnielli for helpful related discussions.

Bibliography

[1] Arnon Avron. Non-deterministic semantics for families of paraconsistent logics. Technical report, 2003. Presented at the III World Congress on Paraconsistency.

[2] Arnon Avron and Iddo Lev. Non-deterministic multiple-valued structures. Technical report, 2003. Submitted to publication.

[3] Patrick Blackburn and Maarten de Rijke. Why combine logics? *Studia Logica*, 59(1):5–27, 1997.

[4] Carlos Caleiro. *Combining Logics*. PhD thesis, IST, Universidade Técnica de Lisboa, PT, 2000.

[5] Carlos Caleiro, Walter A. Carnielli, Marcelo E. Coniglio, and João Marcos. Dyadic semantics for many-valued logics. Research report, CLC, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisbon, PT, 2003. Presented at III World Congress on Paraconsistency, Toulouse, France, July 28–31, 2003.

[6] Carlos Caleiro, Walter A. Carnielli, Marcelo E. Coniglio, and João Marcos. Suszko's Thesis and dyadic semantics. Research report, CLC, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisbon, PT, 2003. Presented at III World Congress on Paraconsistency, Toulouse, France, July 28–31, 2003.

[7] Carlos Caleiro and Jaime Ramos. Cryptomorphisms at work. Abstract, CLC, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisbon, PT, 2004. Presented at XVII International Workshop on Algebraic Development Techniques, March 27–30, 2004, Barcelona, Spain. Long version to be submitted.

[8] Carlos Caleiro, Cristina Sernadas, and Amílcar Sernadas. Mechanisms for combining logics. Research report, Section of Computer Science, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisbon, PT, 1999.

[9] Walter Carnielli. Many-valued logics and plausible reasoning. In *Proceedings of the* XX *International Congress on Many-Valued Logics,* held at the University of Charlotte / NC, US, 1990, pages 328–335. IEEE Computer Society, 1990.

[10] Walter A. Carnielli. Possible-translations semantics for paraconsistent logics. In D. Batens, C. Mortensen, G. Priest, and J. P. Van Bendegem, editors, *Frontiers of Paraconsistent Logic,* Proceedings of the 1st World Congress on Paraconsistency, held in Ghent, BE, July 29–August 3, 1997, pages 149–163. Research Studies Press, Baldock, UK, 2000.

[11] Walter A. Carnielli and Marcelo E. Coniglio. A categorial approach to the combination of logics. *Manuscrito—Revista Internacional de Filosofia*, XXII(2):69–94, October 1999.

[12] Walter A. Carnielli and Itala M. L. D'Ottaviano. Translations between logical systems: a manifesto. *Logique et Analyse (N.S.)*, 40(157):67–81, 1997.

[13] Walter A. Carnielli and Mamede Lima-Marques. Society semantics and multiple-valued logics. In W. Carnielli and I. M. L. D'Ottaviano, editors, *Advances in Contemporary Logic and Computer Science: Proceedings of the* XI *Brazilian Logic Conference on Mathematical Logic*, Salvador, BR, May 6–10, 1996, volume 235 of *Contemporary Mathematics*, pages 33–52. American Mathematical Society, 1999.

[14] Walter A. Carnielli and João Marcos. Limits for paraconsistent calculi. *Notre Dame Journal of Formal Logic*, 40(3):375–390, 1999.

[15] Walter A. Carnielli and João Marcos. *Ex contradictione non sequitur quodlibet.* In R. L. Epstein, editor, *Proceedings of the* II *Annual Conference on Reasoning and Logic*, held in Bucharest, RO, July 2000, volume 1, pages 89–109. Advanced Reasoning Forum, 2001.

[16] Marcelo E. Coniglio. Categorial combination of logics: Completeness preservation. Preprint, Section of Computer Science, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisbon, PT, 2001. Submitted for publication.

[17] Marcelo E. Coniglio and Walter A. Carnielli. Transfers between logics and their applications. *Studia Logica*, 72(3):367–400, 2002.

[18] Newton C. A. da Costa and Jean-Yves Béziau. Théorie de la valuation. *Logique et Analyse (N.S.)*, 37(146):95–117, 1994.

[19] Jairo J. da Silva, Itala M. L. D'Ottaviano, and Antônio M. Sette. Translations between logics. In C. H. Montenegro X. Caicedo, editor, *Models, Algebras and Proofs: Proceedings of the* X *Latin American Symposium on Mathematical Logic,* Bogotá, July 1995, pages 435–448. Marcel Dekker, New York, 1999.

[20] Richard L. Epstein. *Propositional Logics: The semantic foundations of logic.* Wadsworth-Thomson Learning, 2000.

[21] Victor L. Fernández. Society Semantics for n-valued Logics (in Portuguese). Master's thesis, State University of Campinas, BR, 2001.

[22] Victor L. Fernández and Marcelo E. Coniglio. Combining valuations with society semantics. *Journal of Applied Non-Classical Logics*, 13(1):21–46, 2003.

[23] Dov M. Gabbay. *Fibring Logics.* Oxford Logic Guides 38. Clarendon Press, 1999.

[24] João Marcos. Possible-Translations Semantics (in Portuguese). Master's thesis, State University of Campinas, BR, 1999.

[25] João Marcos. On negation: Pure local rules. *Journal of Applied Logic*, in print.

[26] João Marcos. Possible-translations semantics for some weak classically based paraconsistent logics. Technical report, CLC / IST, October 19, 2003. Submitted to publication.

[27] João Marcos and Jean-Yves Béziau. Many values, many semantics. Forthcoming.

[28] Dana S. Scott. On engendering an illusion of understanding. *Journal of Philosophy*, 68:787–807, 1971.

[29] Amílcar Sernadas and Cristina Sernadas. Combining logic systems: Why, how, what for? *CIM Bulletin*, 15:9–14, December 2003.

[30] Amílcar Sernadas, Cristina Sernadas, and Carlos Caleiro. Fibring of logics as a categorial construction. *Journal of Logic and Computation*, 9(2):149–179, 1999.

[31] D. J. Shoesmith and Timothy J. Smiley. *Multiple-Conclusion Logic*. Cambridge University Press, Cambridge–New York, 1978.

[32] Alfred Tarski. Über den Begriff der logischen Folgerung. *Actes du Congrès International de Philosophie Scientifique*, 7:1–11, 1936.

[33] Jan Zygmunt. *An Essay in Matrix Semantics for Consequence Relations*. Wydawnictwo Uniwersytetu Wrocławskiego, Wrocław, 1984.

Heterogeneous Specification and the Heterogeneous Tool Set

Till Mossakowski

BISS, Department of Computer Science, University of Bremen, Germany

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

For the specification of large software systems, heterogeneous multi-logic specifications are needed, since complex problems have different aspects that are best specified in different logics. True logic combinations (in the sense of fibring [8], or colimits of logical systems [14, 17, 4, 4]) work well for certain classes of logics. However, the true combination approach reaches its limits when logics involving very different features (like modalities, higher-order polymorphism, and calculi for concurrent systems) shall be combined. In such cases, a true combination of all the used logics will quickly become too complex. Hence, heterogeneous specification provide a weaker form of logic combination (corresponding to weighted colimits), where basically the logics are put side by side, but can interact via logic translations.

Using heterogeneous specifications, different approaches being developed at different sites can be related, i.e. there is a formal interoperability among languages and tools. In many cases, specialized languages and tools have their strengths in particular aspects. Using heterogeneous specification, these strengths can be combined with comparably small effort.

A general semantic framework for heterogeneous specification has been outlined in another paper in this volume [20]. The goal of the present work is to show how such a framework can be equipped with a proof calculus and tool support. Although some calculus for deriving theorems from a heterogeneous specification is already given in [20], some further work is required in order to obtain a calculus for proving refinements between heterogeneous specifications.

We show that Grothendieck institutions based on institution comorphisms can serve as a framework for developing such a proof calculus and building tools. In particular, we show how to extend the verification semantics given for structured specifications in [13] to the heterogeneous case. This semantics translates a heterogeneous specification into a kernel formalism called development graphs.

The heterogeneous tool set provides tool support for heterogeneous specification. Based on an object-oriented interface for institutions (using type classes in Haskell), it implements the Grothendieck institution and provides a heterogeneous parser, static analysis and proof support for heterogeneous specification. This is based on parsers, static analysers and proof support for the individual institutions, on the above mentioned heterogeneous verification semantics, and on a proof calculus for development graphs over the Grothendieck institution.

2 Institutions, Their (Co)Morphisms, and Heterogeneous Specifications

We only rather briefly recall the technical preliminaries, referring to [20] for more explanation.

Following [10], we formalize logics as institutions.

Definition 1. An *institution* $I = (\mathbf{Sign}^I, \mathbf{Sen}^I, \mathbf{Mod}^I, \models^I)$ consists of

- a category $\mathbf{Sign}^I$ of *signatures*,
- a functor $\mathbf{Sen}^I\colon \mathbf{Sign}^I \longrightarrow \mathbf{Set}$ giving, for each signature Σ, the set of *sentences* $\mathbf{Sen}^I(\Sigma)$, and for each signature morphism $\sigma\colon \Sigma \longrightarrow \Sigma'$, the *sentence translation map* $\mathbf{Sen}^I(\sigma)\colon \mathbf{Sen}^I(\Sigma) \longrightarrow \mathbf{Sen}^I(\Sigma')$, where often $\mathbf{Sen}^I(\sigma)(\varphi)$ is written as $\sigma(\varphi)$,
- a functor $\mathbf{Mod}^I\colon (\mathbf{Sign}^I)^{op} \longrightarrow \mathcal{CAT}$[1]giving, for each signature Σ, the category of *models* $\mathbf{Mod}^I(\Sigma)$, and for each signature morphism $\sigma\colon \Sigma \longrightarrow \Sigma'$, the *reduct functor*

$$\mathbf{Mod}^I(\sigma)\colon \mathbf{Mod}^I(\Sigma') \longrightarrow \mathbf{Mod}^I(\Sigma),$$

  where often $\mathbf{Mod}^I(\sigma)(M')$ is written as $M'|_\sigma$ (the σ-reduct of M'),
- a satisfaction relation $\models^I_\Sigma \subseteq |\mathbf{Mod}^I(\Sigma)| \times \mathbf{Sen}^I(\Sigma)$ for each $\Sigma \in \mathbf{Sign}^I$,

such that for each $\sigma\colon \Sigma \longrightarrow \Sigma'$ in $\mathbf{Sign}^I$ the following *satisfaction condition* holds:

$$M' \models^I_{\Sigma'} \sigma(\varphi) \Leftrightarrow M'|_\sigma \models^I_\Sigma \varphi$$

for each $M' \in \mathbf{Mod}^I(\Sigma')$ and $\varphi \in \mathbf{Sen}^I(\Sigma)$.

Given an institution, it is possible to define the semantics of structured specifications over it in an entirely institution independent way [16]:

***presentations*:** For any signature $\Sigma \in |\mathbf{Sign}|$ and finite set $\Gamma \subseteq \mathbf{Sen}(\Sigma)$ of Σ-sentences, the *presentation* $\langle \Sigma, \Gamma \rangle$ is a specification with:

$$\begin{aligned} Sig[\langle \Sigma, \Gamma \rangle] &:= \Sigma \\ \mathbf{Mod}[\langle \Sigma, \Gamma \rangle] &:= \{M \in \mathbf{Mod}(\Sigma) \mid M \models \Gamma\} \end{aligned}$$

***union*:** For any signature $\Sigma \in |\mathbf{Sign}|$, given Σ-specifications SP_1 and SP_2, their *union* $SP_1 \cup SP_2$ is a specification with:

$$\begin{aligned} Sig[SP_1 \cup SP_2] &:= \Sigma \\ \mathbf{Mod}[SP_1 \cup SP_2] &:= \mathbf{Mod}[SP_1] \cap \mathbf{Mod}[SP_2] \end{aligned}$$

***translation*:** For any signature morphism $\sigma\colon \Sigma \longrightarrow \Sigma'$ and Σ-specification SP, **translate** SP **by** σ is a specification with:

$$\begin{aligned} Sig[\textbf{translate } SP \textbf{ by } \sigma] &:= \Sigma' \\ \mathbf{Mod}[\textbf{translate } SP \textbf{ by } \sigma] &:= \{M' \in \mathbf{Mod}(\Sigma') \mid M'|_\sigma \in \mathbf{Mod}[SP]\} \end{aligned}$$

***hiding*:** For any signature morphism $\sigma\colon \Sigma \longrightarrow \Sigma'$ and Σ'-specification SP', **derive from** SP' **by** σ is a specification with:

$$\begin{aligned} Sig[\textbf{derive from } SP' \textbf{ by } \sigma] &:= \Sigma \\ \mathbf{Mod}[\textbf{derive from } SP' \textbf{ by } \sigma] &:= \{M'|_\sigma \mid M' \in \mathbf{Mod}[SP']\} \end{aligned}$$

We now come to the task of relating different institutions. Institution *morphisms* [10] relate two given institutions. A typical situation is that an institution morphism expresses the fact that a "larger" institution *is built upon* a "smaller" institution by *projecting* the "larger" institution onto the "smaller" one.

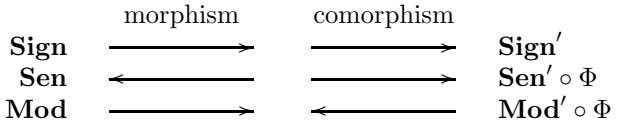
Given institutions I and J, an *institution morphism* [10] $\mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J$ consists of

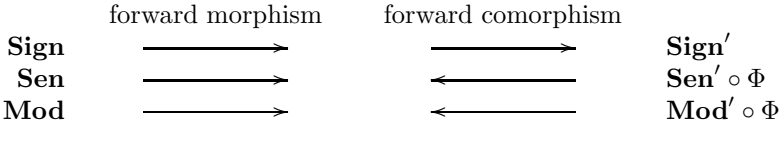
- a functor $\Phi\colon \mathbf{Sign}^I \longrightarrow \mathbf{Sign}^J$,
- a natural transformation $\alpha\colon \mathbf{Sen}^J \circ \Phi \longrightarrow \mathbf{Sen}^I$ and
- a natural transformation $\beta\colon \mathbf{Mod}^I \longrightarrow \mathbf{Mod}^J \circ \Phi^{op}$,

such that the following *satisfaction condition* is satisfied for all $\Sigma \in \mathbf{Sign}^I$, $M \in \mathbf{Mod}^I(\Sigma)$ and $\varphi' \in \mathbf{Sen}^J(\Phi(\Sigma))$:

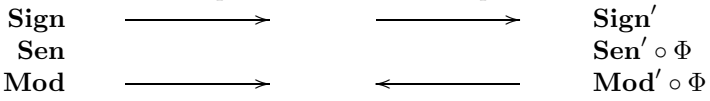
$$M \models^I_\Sigma \alpha_\Sigma(\varphi') \Leftrightarrow \beta_\Sigma(M) \models^J_{\Phi(\Sigma)} \varphi'$$

The notion of institution morphism can be varied in several ways by changing the directions of the arrows or even, in the case of semi-morphisms, omitting the arrows [9, 18]:





[1] $\mathcal{CAT}$ be the (quasi-)category of categories and functors.



The respective satisfaction conditions are quite obvious (note that for semi-(co)morphisms, none is required).

These various notions of institution translations naturally lead to the following heterogeneous specification constructs [19, 20]:

***heterogeneous translation*:** For any institution comorphism, forward comorphism or semi-comorphism $\mu = (\Phi, \alpha, \beta)\colon I \longrightarrow I'$ and Σ-specification SP in I, **translate** SP **by** μ is a specification with:

$$\begin{aligned} &Sig[\textbf{translate } SP \textbf{ by } \mu] &&:= \Phi(\Sigma) \\ &\textbf{Mod}[\textbf{translate } SP \textbf{ by } \mu] &&:= \{M' \in \textbf{Mod}(\Phi(\Sigma)) \mid \beta_\Sigma(M') \in \textbf{Mod}[SP]\} \end{aligned}$$

***heterogeneous hiding*:** For any institution morphism, forward morphism or semi-morphism $\mu = (\Phi, \alpha, \beta) : I \longrightarrow I'$ and Σ-specification SP in I, **derive from** SP **by** μ is a specification with:

$$\begin{aligned} &Sig[\textbf{derive from } SP \textbf{ by } \mu] &&:= \Phi(\Sigma) \\ &\textbf{Mod}[\textbf{derive from } SP \textbf{ by } \mu] &&:= \{\beta_\Sigma(M') \mid M' \in \textbf{Mod}[SP]\} \end{aligned}$$

3 Development Graphs

The notion of institutions gains much of its importance by the fact that one can design languages for structured specifications in a completely institution independent and even heterogeneous way, as explained in the previous section.

However, the standard proof calculi for proving entailment within and refinement between such structured specifications [2] rely on some form of the *Craig interpolation* property, which fails in some institutions (even for many-sorted logic, it holds only for sort-injective signature morphisms). Moreover, although Craig interpolation for heterogeneous specification has been studied [7], this of course relies on Craig interpolation for the individual logics, and some extra assumptions about Craig interpolation for the comorphisms that are not satisfied in many practical examples.

We hence propose to follow a different path, namely to use the formalism of *development graphs* [1]. The proof calculus for this does not rely on Craig interpolation; rather, is based on a different assumption, namely the existence of *weakly amalgamable cocones* [12]. Note that the latter property (although technically incomparable in strength with Craig interpolation) for practical purposes is a weaker assumption than Craig interpolation (cf. the results of [6]).

A development graph consists of a set of nodes (corresponding to whole structured specifications or parts thereof), and a set of arrows called definition links, indicating the dependency of each involved structured specification on its subparts.

Definition 2. Given an arbitrary but fixed institution I, a *development graph* $\mathcal{DG}$ over I is an acyclic directed graph $\mathcal{S} = \langle \mathcal{N}, \mathcal{L} \rangle$.

$\mathcal{N}$ is a set of nodes. Each node $N \in \mathcal{N}$ is a tuple (Σ^N, Γ^N) such that $\Sigma^N \in \textbf{Sign}^I$ is a signature and $\Gamma^N \subseteq \textbf{Sen}^I(\Sigma^N)$ is the set of *local axioms* of N.

$\mathcal{L}$ is a set of directed links, so-called *definition links*, between elements of $\mathcal{N}$. Each definition link from a node M to a node N is either

- *global* (denoted $M \xrightarrow{\sigma} N$), annotated with a signature morphism $\sigma : \Sigma^M \to \Sigma^N \in \textbf{Sign}^I$, or
- *hiding* (denoted $M \xrightarrow[h]{\sigma} N$), annotated with a signature morphism $\sigma : \Sigma^N \to \Sigma^M \in \textbf{Sign}^I$ *going against the direction of the link*. Typically, σ will be an inclusion, and the symbols of Σ^M not in Σ^N will be hidden.

What is the meaning of such development graphs? Development graphs without hiding have a theory-level semantics, see [1]. For development graphs with hiding, a model-level semantics seems to be more appropriate:

Definition 3. Given a node $N \in \mathcal{N}$ in a development graph $\mathcal{DG}$, its associated class $\mathbf{Mod}_{\mathcal{S}}(N)$[2] of models (or N-models for short) consists of those Σ^N-models n for which

- n satisfies the local axioms Γ^N,
- for each $K \xrightarrow{\sigma} N \in \mathcal{DG}$, $n|_\sigma$ is a K-model, and
- for each $K \xrightarrow[h]{\sigma} N \in \mathcal{DG}$, n has a σ-expansion k (i.e. $k|_\sigma = n$) which is a K-model.

Complementary to definition and hiding links, which *define* the theories of related nodes, we introduce the notion of a *theorem link* with the help of which we are able to *postulate* relations between different theories. Global theorem links[3] (denoted by $N -\stackrel{\sigma}{-}\!\!\succ M$, where $\sigma\colon \Sigma^N \longrightarrow \Sigma^M$) are the central data structure to represent proof obligations arising in formal developments.

Definition 4. Let $\mathcal{DG}$ be a development graph. $\mathcal{DG}$ *implies* a global theorem link $N -\stackrel{\sigma}{-}\!\!\succ M$ (denoted $\mathcal{DG} \models N -\stackrel{\sigma}{-}\!\!\succ M$), iff for all $m \in \mathbf{Mod}_{\mathcal{DG}}(M)$, $m|_\sigma \in \mathbf{Mod}_{\mathcal{DG}}(N)$.

4 Grothendieck Institutions and Spans of Comorphisms

Heterogeneous specification can be viewed as structured specification over a Grothendieck construction. Diaconescu's Grothendieck institution construction [5] basically flattens a diagram of institution and morphisms. We here recall the Grothendieck institution for the comorphism-based case [12]:

Definition 5. An *indexed coinstitution* is a functor $\mathcal{I}\colon Ind^{op} \longrightarrow \mathbf{CoIns}$ into the category **CoIns** of institutions and institution comorphisms[4].

The basic idea of the Grothendieck institution is that all signatures of all institutions are put side by side, and a signature morphism in this large realm of signatures consists of an intra-institution signature morphism plus an inter-institution translation (along some institution comorphism). The other components are then defined in a straightforward way.

Definition 6. Given an *indexed coinstitution* $\mathcal{I}\colon Ind^{op} \longrightarrow \mathbf{CoIns}$, define the *Grothendieck institution* $\mathcal{I}^\#$ as follows:

- signatures in $\mathcal{I}^\#$ are pairs (Σ, i), where $i \in |Ind|$ and Σ a signature in the institution $\mathcal{I}(i)$,
- signature morphisms $(\sigma, e)\colon (\Sigma_1, i) \longrightarrow (\Sigma_2, j)$ consist of a morphism $e\colon j \longrightarrow i \in Ind$ and a signature morphism $\sigma\colon \Phi^{\mathcal{I}(e)}(\Sigma_1) \longrightarrow \Sigma_2$ (here, $\mathcal{I}(e)\colon \mathcal{I}(i) \longrightarrow \mathcal{I}(j)$ is the institution comorphism corresponding to the arrow $e\colon j \longrightarrow i$ in the indexed coinstitution, and $\Phi^{\mathcal{I}(e)}$ is its signature translation component),
- the (Σ, i)-sentences are the Σ-sentences in $\mathcal{I}(i)$, and sentence translation along (σ, e) is the composition of sentence translation along σ with sentence translation along $\mathcal{I}(e)$,
- the (Σ, i)-models are the Σ-models in $\mathcal{I}(i)$, and model reduction along (σ, e) is the composition of model translation along $\mathcal{I}(e)$ with model reduction along σ, and
- satisfaction w.r.t. (Σ, i) is satisfaction w.r.t. Σ in $\mathcal{I}(i)$.

[2] $\mathbf{Mod}_{\mathcal{DG}}$ is not to be confused with the model functor **Mod** of the institution.

[3] There are also local and hiding theorem links, which are omitted here for simplicity.

[4] Indeed, the name is justified by the fact that the category of institutions and institution comorphisms is isomorphic to the category of coinstitutions and coinstitution morphisms. A coinstitution is an institution with model translations covariant to signature morphisms, while sentence translations are contravariant.

While comorphism-based Grothendieck institutions are a good semantic basis for heterogeneous specifications involving institution comorphisms, the question arises what to do with the other kinds of translations, like institution morphisms or semi-morphisms. Rather than complicate the Grothendieck construction with different kinds of translations, instead we represent the other kinds of translations as *spans* of institution comorphisms.

The span construction works as follows:

Each institution morphism $\mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J = I \xrightarrow{\Phi} \xleftarrow{\alpha} \xrightarrow{\beta} J$ can be translated into a span $I \xleftarrow{\mu^-} J \circ \Phi \xrightarrow{\mu^+} J$ of institution comorphisms as follows:

$$\begin{array}{ccccc}
\mathbf{Sign}^I & \xleftarrow{id} & \mathbf{Sign}^I & \xrightarrow{\Phi} & \mathbf{Sign}^J \\
\mathbf{Sen}^I & \xleftarrow{\alpha} & \mathbf{Sen}^J \circ \Phi & \xrightarrow{id} & \mathbf{Sen}^J \circ \Phi \\
\mathbf{Mod}^I & \xrightarrow{\beta} & \mathbf{Mod}^J \circ \Phi & \xleftarrow{id} & \mathbf{Mod}^J \circ \Phi
\end{array}$$

Here, the "middle" institution $J \circ \Phi$ is the institution with signature category inherited from I, but sentences and models inherited from J via Φ.

Consider now a semi-comorphism $I \xrightarrow{\Phi} \xleftarrow{\beta} J$. It can be translated into a span $I \xleftarrow{\mu^-} I^{\emptyset} \xrightarrow{\mu^+} J$ of comorphisms

$$\begin{array}{ccccc}
\mathbf{Sign} & \xleftarrow{id} & \mathbf{Sign}^I & \xrightarrow{\Phi} & \mathbf{Sign}^J \\
\mathbf{Sen}^I & \xleftarrow{incl} & \emptyset & \xrightarrow{incl} & \mathbf{Sen}^J \circ \Phi \\
\mathbf{Mod}^I & \xrightarrow{id} & \mathbf{Mod}^I & \xleftarrow{\beta} & \mathbf{Mod}^J \circ \Phi
\end{array}$$

while a semi-morphism $I \xrightarrow{\Phi} \xrightarrow{\beta} J$ is translated into a span $I \xleftarrow{\mu^-} J \circ \Phi^{\emptyset} \xrightarrow{\mu^+} J$ of comorphisms

$$\begin{array}{ccccc}
\mathbf{Sign} & \xleftarrow{id} & \mathbf{Sign}^I & \xrightarrow{\Phi} & \mathbf{Sign}^J \\
\mathbf{Sen}^I & \xleftarrow{incl} & \emptyset & \xrightarrow{incl} & \mathbf{Sen}^J \circ \Phi \\
\mathbf{Mod}^I & \xrightarrow{\beta} & \mathbf{Mod}^J \circ \Phi & \xleftarrow{id} & \mathbf{Mod}^J \circ \Phi
\end{array}$$

where in each case the "middle" institution has the indicated components.

Forward comorphisms $\mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J = I \xrightarrow{\Phi} \xleftarrow{\alpha} \xleftarrow{\beta} J$ are translated into spans of form $I \xleftarrow{\mu^-} J \circ \Phi^{\mathbf{Sen}} \xrightarrow{\mu^+} J$ consisting of institution comorphisms as follows:

$$\begin{array}{ccccc}
\mathbf{Sign}^I & \xleftarrow{id} & \mathbf{Sign}^I & \xrightarrow{\Phi} & \mathbf{Sign}^J \\
\mathbf{Sen}^I & \xleftarrow{\alpha} & \mathbf{Sen}^J \circ \Phi & \xrightarrow{id} & \mathbf{Sen}^J \circ \Phi \\
\mathbf{Mod}^I & \xrightarrow{id} & \mathbf{Mod}^I & \xleftarrow{\beta} & \mathbf{Mod}^J \circ \Phi
\end{array}$$

The "middle" institution $J \circ \Phi^{\mathbf{Sen}}$ inherits signatures and models from I, but sentences (via Φ) from J. The satisfaction relation $M \models^{J \circ \Phi^{\mathbf{Sen}}}_{\Sigma} \varphi$ holds iff $\beta_\Sigma(M) \models^I_\Sigma \varphi$ in I.

Dually, a forward morphism $\mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J = I \xrightarrow[\alpha]{\Phi} J \xleftarrow{\beta}$ can be translated into a span $I \xleftarrow{\mu^-} J \circ \Phi^{\mathbf{Mod}} \xrightarrow{\mu^+} J$ of institution comorphisms as follows:

$$\begin{array}{ccccc}
\mathbf{Sign}^I & \xleftarrow{id} & \mathbf{Sign}^I & \xrightarrow{\Phi} & \mathbf{Sign}^J \\
\mathbf{Sen}^I & \xleftarrow{id} & \mathbf{Sen}^I & \xrightarrow{\alpha} & \mathbf{Sen}^J \circ \Phi \\
\mathbf{Mod}^I & \xrightarrow{\beta} & \mathbf{Mod}^J \circ \Phi & \xleftarrow{id} & \mathbf{Mod}^J \circ \Phi
\end{array}$$

The "middle" institution $J \circ \Phi^{\mathbf{Mod}}$ inherits signatures and sentences from I, but models (via Φ) from J. The satisfaction relation $M \models^{J\circ\Phi^{\mathbf{Mod}}}_{\Sigma} \varphi$ holds iff $M \models^{J}_{\Phi(\Sigma)} \alpha_\Sigma(\varphi)$ in J.

5 The Heterogeneous Verification Semantics

The purpose of the heterogeneous verification semantics, which follows a similar structured verification semantics in [13], is to provide a proof calculus for heterogeneous specifications. Some heterogeneous calculus rules have been given already in [20]; however, they cover only the question whether a heterogeneous specification entails a sentence, but not the question whether a heterogeneous specification entails (or refines to) another one.

General assumption. We assume to work with an indexed coinstitution that contains all the "middle" institutions as well as the μ^- and μ^+ comorphisms given by the constructions of the previous section, applied to those morphisms, semi-morphisms etc. that we expect to occur in our heterogeneous specifications. Of course, we assume that all institutions and comorphisms that are used directly are included as well.

The heterogeneous verification semantics now takes a heterogeneous specification and translates it into a development graph over the Grothendieck institution induced by the indexed coinstitution given by the above assumption.

In the sequel, if we want to extend a given development graph $\mathcal{DG}$, we use a suggestive concise notation like $\mathcal{DG}' = \mathcal{DG} \uplus \{N' := (\Sigma', \Gamma);\ N \xrightarrow{\sigma} N'\ \}$ which should be largely self-explanatory (in particular, '$N' := (\Sigma', \Gamma)$' means that we introduce a new node N' with $\Sigma^{N'} = \Sigma'$ and $\Gamma^{N'} = \Gamma$).

Furthermore, by abuse of notation, we identify institutions and comorphisms with their respective indices in the index category of the indexed coinstitution (in general, it is expected that the indexed coinstitution is an embedding of categories; hence this abuse of notation will not lead to ambiguities).

The verification semantics uses judgements of form

$$\vdash SP \rhd\!\!\rhd\!\!\rhd (N, \mathcal{DG})$$

which read as: the specification SP is translated to the node N in development graph $\mathcal{DG}$.

$$\frac{\Sigma \text{ is a signature in } I}{\vdash \langle \Sigma, \Gamma\rangle \rhd\!\!\rhd\!\!\rhd (N, \{N := ((\Sigma, I), \Gamma)\})}$$

$$\frac{\begin{array}{c}\vdash SP_1 \rhd\!\!\rhd\!\!\rhd (N_1, \mathcal{DG}_1) \\ \vdash SP_2 \rhd\!\!\rhd\!\!\rhd (N_2, \mathcal{DG}_2)\end{array}}{\vdash SP_1 \cup SP_2 \rhd\!\!\rhd\!\!\rhd (K, \mathcal{DG}_1 \uplus \mathcal{DG}_2 \uplus \{K := (\Sigma^{N_1}, \emptyset)\} \uplus \{\ N_i \xrightarrow{id} K \mid i = 1, 2\})}$$

$$\frac{\begin{array}{c}\vdash SP \rhd\!\!\rhd\!\!\rhd (N, \mathcal{DG}) \\ N = (\Sigma, I)\end{array}}{\vdash \textbf{translate } SP \textbf{ by } \sigma\colon \Sigma \longrightarrow \Sigma' \rhd\!\!\rhd\!\!\rhd (K, \mathcal{DG} \uplus \{\ N \xrightarrow{(\sigma, id_I)} K\ := ((\Sigma', I), \emptyset)\})}$$

$$\frac{\vdash SP \rhd\!\!\rhd\!\!\rhd (N, \mathcal{DG}) N = (\Sigma', I)}{\vdash \textbf{derive from } SP' \textbf{ by } \sigma\colon \Sigma \longrightarrow \Sigma' \rhd\!\!\rhd\!\!\rhd (K, \mathcal{DG} \uplus \{\ N \xrightarrow[h]{(\sigma, id_I)} K\ := ((\Sigma, I), \emptyset)\})}$$

$$\frac{\vdash SP \mathrel{|\!\!\Rightarrow\!\!|} (N, \mathcal{DG}) \quad N = (\Sigma, I) \quad \mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J \text{ is a comorphism}}{\vdash \textbf{translate } SP \textbf{ by } \mu\colon I \longrightarrow J \mathrel{|\!\!\Rightarrow\!\!|} (K, \mathcal{DG} \uplus \{ N \xrightarrow{(id,\mu)} K \ := ((\Phi(\Sigma), J), \emptyset)\})}$$

$$\frac{\vdash SP \mathrel{|\!\!\Rightarrow\!\!|} (N, \mathcal{DG}) \quad N = (\Sigma, I) \quad \mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J \text{ is a morphism} \quad \mathcal{DG}' = \mathcal{DG} \uplus \{ N \xrightarrow[h]{(id,\mu^-)} K \ := (\Sigma, J \circ \Phi), \emptyset);\ \ K \xrightarrow{(id,\mu^+)} P \ := ((\Phi(\Sigma), J), \emptyset)\}}{\vdash \textbf{derive from } SP \textbf{ by } \mu\colon I \longrightarrow J \mathrel{|\!\!\Rightarrow\!\!|} (P, \mathcal{DG}')}$$

$$\frac{\vdash SP \mathrel{|\!\!\Rightarrow\!\!|} (N, \mathcal{DG}) \quad N = (\Sigma, I) \quad \mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J \text{ is a semi-comorphism} \quad \mathcal{DG}' = \mathcal{DG} \uplus \{ N \xrightarrow[h]{(id,\mu^-)} K \ := (\Sigma, I^{\emptyset}), \emptyset);\ \ K \xrightarrow{(id,\mu^+)} P \ := ((\Phi(\Sigma), J), \emptyset)\}}{\vdash \textbf{translate } SP \textbf{ by } \mu\colon I \longrightarrow J \mathrel{|\!\!\Rightarrow\!\!|} (P, \mathcal{DG}')}$$

$$\frac{\vdash SP \mathrel{|\!\!\Rightarrow\!\!|} (N, \mathcal{DG}) \quad N = (\Sigma, I) \quad \mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J \text{ is a semi-morphism} \quad \mathcal{DG}' = \mathcal{DG} \uplus \{ N \xrightarrow[h]{(id,\mu^-)} K \ := (\Sigma, J \circ \Phi^{\emptyset}), \emptyset);\ \ K \xrightarrow{(id,\mu^+)} P \ := ((\Phi(\Sigma), J), \emptyset)\}}{\vdash \textbf{translate } SP \textbf{ by } \mu\colon I \longrightarrow J \mathrel{|\!\!\Rightarrow\!\!|} (P, \mathcal{DG}')}$$

$$\frac{\vdash SP \mathrel{|\!\!\Rightarrow\!\!|} (N, \mathcal{DG}) \quad N = (\Sigma, I) \quad \mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J \text{ is a forward comorphism} \quad \mathcal{DG}' = \mathcal{DG} \uplus \{ N \xrightarrow[h]{(id,\mu^-)} K \ := (\Sigma, J \circ \Phi^{\mathbf{Sen}}), \emptyset);\ \ K \xrightarrow{(id,\mu^+)} P \ := ((\Phi(\Sigma), J), \emptyset)\}}{\vdash \textbf{translate } SP \textbf{ by } \mu\colon I \longrightarrow J \mathrel{|\!\!\Rightarrow\!\!|} (P, \mathcal{DG}')}$$

$$\frac{\vdash SP \mathrel{|\!\!\Rightarrow\!\!|} (N, \mathcal{DG}) \quad N = (\Sigma, I) \quad \mu = (\Phi, \alpha, \beta)\colon I \longrightarrow J \text{ is a forward morphism} \quad \mathcal{DG}' = \mathcal{DG} \uplus \{ N \xrightarrow[h]{(id,\mu^-)} K \ := (\Sigma, J \circ \Phi^{\mathbf{Mod}}), \emptyset);\ \ K \xrightarrow{(id,\mu^+)} P \ := ((\Phi(\Sigma), J), \emptyset)\}}{\vdash \textbf{translate } SP \textbf{ by } \mu\colon I \longrightarrow J \mathrel{|\!\!\Rightarrow\!\!|} (P, \mathcal{DG}')}$$

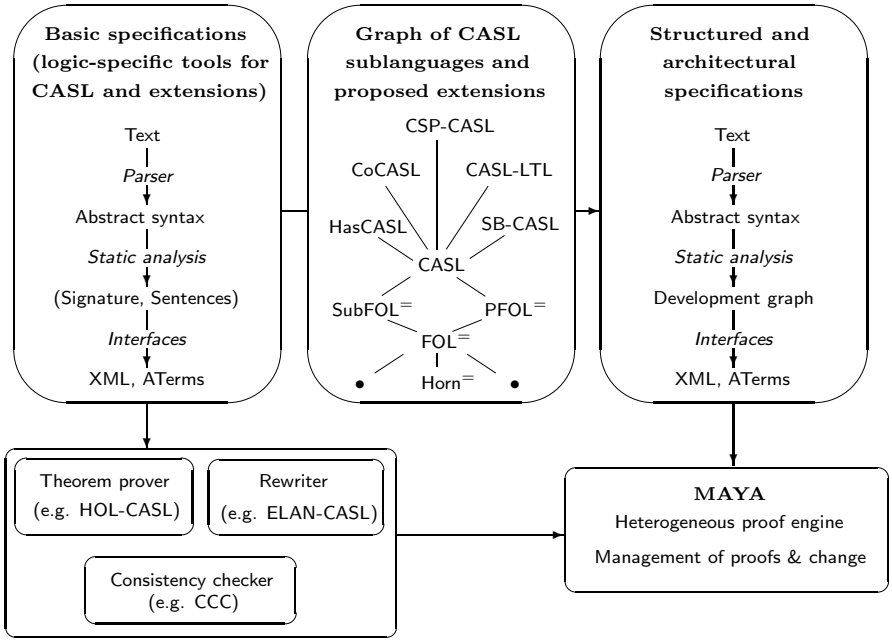
We now state the important property of the heterogeneous verification semantics:

Theorem 7. The heterogeneous verification semantics preserves model classes of heterogeneous specifications. More precisely, given a heterogenous specification SP with $\vdash SP \mathrel{|\!\!\Rightarrow\!\!|} (N, \mathcal{DG})$, we have

$$\mathbf{Mod}[SP] = \mathbf{Mod}_{\mathcal{DG}}(N)$$

By inserting theorem links into a development graph generated by the heterogeneous verification semantics, it is now possible to tackle the problem of refinement between heterogeneous specifications. A proof calculus for such theorem links in the context of Grothendieck institutions has been given in [12]. This proof calculus relies on proof calculi (formalized as so-called entailment systems [11]) for the individual institutions, as well as some technical conditions on both the institutions (namely, the existence of weakly amalgamable cocones) as well as the comorphisms.

A word concerning the difference between morphisms and semi-morphisms is in order: although they are treated quite similarly in the heterogeneous verification semantics, their difference shows up when using the proof calculus: the "middle" institution is much poorer in the case of semi-morphisms (it has



no sentences). The latter makes it much harder to conduct heterogeneous proofs; indeed, for proofs along semi-morphisms, typically both the source and target institution have to be translatable into some common target institution (or some special proof rules for the particular semi-morphism has to be added). Of course, a similar remark applies to semi-comorphisms as well.

6 The Heterogeneous Tool set (HETS)

The Heterogeneous Tool Set HETS is a tool implementing the theory developed so far. Its architecture is depicted above. HETS has an abstract interface corresponding to concept of institution (or more precisely, entailment system — since model theory is not directly implementable) in Haskell.

HETS implements this by providing a type class `Logic`. `Logic` is a multiparameter type classes with functional dependencies [15]. Such a type class can be thought of as a formal parameter signature. `Logic` contains types for signatures, signature morphisms, sentences, abstract syntax of basic specifications etc., and functions for parsing, printing, static analysis, and proving. Based on this abstract interface, we have implemented *heterogeneous* tools for parsing and static analysis of heterogeneous CASL(using a semantics similar to the verification semantics in Section 5 above). The static semantic analysis yields a development graph over the Grothendieck institution, and we are currently implementing the corresponding proof calculus.

Technically, heterogeneity is realized as follows. On top of the type class `Logic`, an existential datatype is constructed. Usually, existential types are used to realize e.g. heterogeneous lists, where each element may have a different type. We use lists of (components of) institutions and comorphisms instead. This leads to an implementation of the Grothendieck institution over an indexed coinstitution.

We have instantiated this general framework with institution-specific analysis tools for CASL, HASCASL, Haskell, CSP-CASL and MODALCASL. Future work will interface existing theorem proving tools with specific institutions in HETS. We already have implemented an experimental interface to the theorem prover Isabelle.

The Heterogeneous Tool Set is available at www.tzi.de/cofi/hets.

Acknowledgements

Thanks to Andrzej Tarlecki, Joseph Goguen, Grigore Rosu, Serge Autexier and Dieter Hutter for useful cooperation and discussions, and to Răzvan Diaconescu for inventing Grothendieck institutions.

This work has been supported by the *Deutsche Forschungsgemeinschaft* under Grant KR 1191/5-2.

Bibliography

[1] S. Autexier, D. Hutter, H. Mantel, and A. Schairer. Towards an evolutionary formal software-development using CASL. In C. Choppy and D. Bert, editors, *Recent Trends in Algebraic Development Techniques, 14th International Workshop, WADT'99, Bonas, France*, volume 1827 of *Lecture Notes in Computer Science*, pages 73–88. Springer-Verlag, 2000.

[2] T. Borzyszkowski. Logical systems for structured specifications. *Theoretical Computer Science*, 286:197–245, 2002.

[3] C. Caleiro, W. A. Carnielli, M. E. Coniglio, A. Sernadas, and C. Sernadas. Fibring non-truth-functional logics: Completeness preservation. *Journal of Logic, Language and Information*, 12(2):183–211, 2003.

[4] C. Caleiro, P. Mateus, J. Ramos, and A. Sernadas. Combining logics: Parchments revisited. *Lecture Notes in Computer Science*, 2267:48–??, 2001.

[5] R. Diaconescu. Grothendieck institutions. *Applied categorical structures*, 10:383–402, 2002.

[6] R. Diaconescu. An institution-independent proof of Craig Interpolation Property. *Studia Logica*, 76(3), 2004.

[7] R. Diaconescu. Interpolation in Grothendieck Institutions. *Theoretical Computer Science*, 311(1–3):439–461, Jan. 2004.

[8] D. M. Gabbay. *Fibring Logics*. Oxford University Press, Oxford, 1999.

[9] J. Goguen and G. Rosu. Institution morphisms. *Formal aspects of computing*, 13:274–307, 2002.

[10] J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39:95–146, 1992. Predecessor in: LNCS 164, 221–256, 1984.

[11] J. Meseguer. General logics. In *Logic Colloquium 87*, pages 275–329. North Holland, 1989.

[12] T. Mossakowski. Comorphism-based Grothendieck logics. In K. Diks and W. Rytter, editors, *Mathematical foundations of computer science*, volume 2420 of *LNCS*, pages 593–604. Springer, 2002.

[13] T. Mossakowski, P. Hoffman, S. Autexier, and D. Hutter. CASL logic. In P. D. Mosses, editor, *CASL Reference Manual*, volume 2960 of *Lecture Notes in Computer Science*, part ÏV. Springer Verlag, London, 2004. Edited by T. Mossakowski.

[14] T. Mossakowski, A. Tarlecki, and W. Pawłowski. Combining and representing logical systems using model-theoretic parchments. In F. Parisi Presicce, editor, *Recent trends in algebraic development techniques. Proc. 12th International Workshop*, volume 1376 of *Lecture Notes in Computer Science*, pages 349–364. Springer, 1998.

[15] S. Peyton Jones, M. Jones, and E. Meijer. Type classes: exploring the design space. In *Haskell Workshop*. 1997.

[16] D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Information and Computation*, 76:165–210, 1988.

[17] A. Sernadas, C. Sernadas, C. Caleiro, and T. Mossakowski. Categorical fibring of logics with terms and binding operators. In D. Gabbay and M. d. Rijke, editors, *Frontiers of Combining Systems 2*, Studies in Logic and Computation, pages 295–316. Research Studies Press, 2000.

[18] A. Tarlecki. Moving between logical systems. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specifications. 11th Workshop on Specification of Abstract Data Types*, volume 1130 of *Lecture Notes in Computer Science*, pages 478–502. Springer Verlag, 1996.

[19] A. Tarlecki. Towards heterogeneous specifications. In D. Gabbay and M. d. Rijke, editors, *Frontiers of Combining Systems 2, 1998*, Studies in Logic and Computation, pages 337–360. Research Studies Press, 2000.

[20] A. Tarlecki. Software specification and development in heterogeneous environments. This volume, 2004.

Exogenous Quantum Logic

P. Mateus and A. Sernadas

CLC, Department of Mathematics, IST, Lisbon, Portugal

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Context

Most of the work on quantum logic (since the seminal paper [4]) has continued to adopt the lattice of closed subspaces of a Hilbert space as the basis for its semantics [11, 8].

Here we take a quite different approach, what we call the exogenous approach. The key idea is to keep the models of the classical logic (say propositional logic) as they are, to produce models for the envisaged quantum logic as superpositions of classical models, and, finally, to design a suitable language for constraining such superpositions.

The exogenous approach is a variation of the possible worlds approach originally proposed by Kripke [14] for modal logic, and it is also akin to the society semantics introduced in [7] for many-valued logic and to the possible translations semantics proposed in [6] for paraconsistent logic. In fact, Kripke structures can be described as binary relations between classical models, the models in [7] are just collections of classical models, and the models in [6] are obtained using translation maps into the original logic(s). The possible worlds approach was also used in [19, 20] for probabilistic logic[1]: the models turn out to be probability spaces of classical models, as first recognized in [10].

The difference between the possible worlds approach and the exogenous approach is subtle but full of consequences. The exogenous approach is used in [16] to develop a probabilistic version of any given logic where, as in [10], each model is a probability space of the original models, but where the connectives are different from those of the logic being probabilized. As explained in Section 2, the global semantics of the new connectives arises naturally when using the new models.

Note that the endogenous approach to probabilistic logic is also useful and, actually, widely used. By endogenous approach we mean that we tinker with the classical models in order to make them suitable for a specific type of probabilistic reasoning. For instance, if we want a logic for reasoning about probabilistic transition systems (probabilistic automata) we can modify the Kripke models of dynamic logic by labelling the transition pairs (pairs of the accessibility relation) with probabilities [12, 15]. As another example of the endogenous approach, consider the probabilization of first-order logic obtained by enriching the domain of individuals with a probability distribution [1], having in mind notions like almost everywhere.

Returning to quantum logic, the exogenous approach seems promising for several reasons: (i) it can be applied to any given logic[2]; (ii) it settles once and for all the issue of the nature of the quantum models (as superpositions of models from the original logic); and, finally, (iii) it also guides the design of the quantum language (that should provide the means to write assertions about both the original models and the quantum models). Furthermore, since quantum logic involves probabilistic reasoning (because of the stochastic nature of the results of observing quantum systems), the successful exogenous development of probabilistic logic in [16] reinforces this idea.

It is to be expected that the lattice approach to quantum logic will play a similar role to the one played by modal algebras in modal logic, by Heyting algebras in intuitionistic logic, by Boolean algebras in classical logic, etc. But, as in those cases, the algebraic approach is not the right source of inspiration for discovering the linguistic ingredients of the envisaged logic. For instance, modal algebras appeared much later than Kripke structures, well after the modal language was widely accepted.

In Section 2, we start by showing how to set up the exogenous probabilistic version of propositional logic. Afterwards, we continue the process, first, in Section 3, towards the exogenous quantum version of propositional logic (including the probabilistic one), and finally, in Section 4, towards a dynamic version of the latter (for reasoning also about changes in the quantum state of a system).

2 EPPL

Assume a fixed set $\{\mathcal{P}_k : k \in \mathbb{N}\}$ of propositional constants. Recall that a classical valuation is a map $v : \{\mathcal{P}_k : k \in \mathbb{N}\} \to \{0, 1\}$. The exogenous approach to probabilistic logic suggests that we identify a probabilistic valuation with a probability space of classical valuations. A Nilsson[3] structure $\mathbf{V}$ is a tuple $\langle V, \mathcal{B}, \nu\rangle$ where: V is a nonempty set of classical valuations; $\mathcal{B}$ is a σ-algebra over V (that is, $\mathcal{B} \subseteq \wp V$ and $\mathcal{B}$ is closed under complements and countable unions) such that $\{v \in V : v \Vdash \mathcal{P}_k\} \in \mathcal{B}$ for each $k \in \mathbb{N}$; and ν is a map from $\mathcal{B}$ to $[0, 1]$ such that $\nu(V) = 1$ and $\nu(\bigcup_{j\in\mathbb{N}} B_j) = \sum_{j\in\mathbb{N}} \nu(B_j)$ whenever $B_{j_1} \cap B_{j_2} = \emptyset$

[1] Later on, the relationship to epistemic logic was made clear in [3, 2].

[2] Herein, we just address the problem of designing the quantum version of classical propositional logic, but it is feasible to repeat the process for any given logic fulfilling some minimal requirements as it was done in [16] for probabilistic logic.

[3] In recognition of the significance of [19] for the development of probabilistic logic.

for every $j_1 \neq j_2 \in \mathbb{N}$. In short, a Nilsson structure is a probability space where outcomes are classical valuations and the extent of every propositional constant is among the events.

This exogenous semantics for probabilistic reasoning suggests that we adopt the probabilistic language composed of formulae of the form[4]

$$\gamma := \omega_k \curlyvee \varphi \curlyvee (t \leq t) \curlyvee (\boxminus \gamma) \curlyvee (\gamma \sqsupset \gamma)$$

where φ is a classical formula of the form

$$\varphi := \xi_k \curlyvee \mathcal{P}_k \curlyvee (\neg \varphi) \curlyvee (\varphi \Rightarrow \varphi)$$

and t is a real term of the form

$$t := \theta_k \curlyvee r \curlyvee (\textstyle\int \varphi) \curlyvee (\int \varphi \,|\, \varphi) \curlyvee (t + t) \curlyvee (t\, t)$$

where r is a computable real number. The ξ's, ω's and θ's are variables (to be used in rules) that can be the target of substitutions respecting the syntactic categories. An expression is said to be ground if it does not contain any such variable. As usual, other (classical and probabilistic) connectives can be used as abbreviations. Furthermore, we write $(t_1 = t_2)$ for $((t_1 \leq t_2) \sqcap (t_2 \leq t_1))$.

Clearly, for each ground classical formula φ, $[\varphi]_{\mathbf{V}} = \{v \in V : v \Vdash \varphi\} \in \mathcal{B}$. So, from the probabilistic point of view, it is worthwhile to look at classical formulae as describing events. The denotation of ground terms is inductively defined as follows:

- $[\![r]\!]_{\mathbf{V}} = r$;
- $[\![(\int \varphi)]\!]_{\mathbf{V}} = \nu([\varphi]_{\mathbf{V}})$;
- $[\![(\int \varphi_2 \,|\, \varphi_1)]\!]_{\mathbf{V}} = \begin{cases} \frac{\nu([\varphi_1]_{\mathbf{V}} \cap [\varphi_2]_{\mathbf{V}})}{\nu([\varphi_1]_{\mathbf{V}})} & \text{if } \nu([\varphi_1]_{\mathbf{V}}) \neq 0 \\ 1 & \text{otherwise} \end{cases}$;
- $[\![(t_1 + t_2)]\!]_{\mathbf{V}} = [\![t_1]\!]_{\mathbf{V}} + [\![t_2]\!]_{\mathbf{V}}$;
- $[\![(t_1 t_2)]\!]_{\mathbf{V}} = [\![t_1]\!]_{\mathbf{V}} \times [\![t_2]\!]_{\mathbf{V}}$.

According to the exogenous approach, the satisfaction of formulae by $\mathbf{V}$ and ground substitution ρ is as follows:

- $\mathbf{V}\rho \Vdash \omega_j$ iff $\mathbf{V}\rho \Vdash \omega_j\rho$;
- $\mathbf{V}\rho \Vdash \varphi$ iff $v \Vdash \varphi\rho$ for every $v \in V$;
- $\mathbf{V}\rho \Vdash (t_1 \leq t_2)$ iff $[\![t_1\rho]\!]_{\mathbf{V}} \leq [\![t_2\rho]\!]_{\mathbf{V}}$;
- $\mathbf{V}\rho \Vdash (\boxminus \gamma)$ iff $\mathbf{V}\rho \nVdash \gamma$;
- $\mathbf{V}\rho \Vdash (\gamma_1 \sqsupset \gamma_2)$ iff $\mathbf{V}\rho \nVdash \gamma_1$ or $\mathbf{V}\rho \Vdash \gamma_2$.

The global nature of satisfaction is a key ingredient of the exogenous approach (contrarily to the local nature of satisfaction within the possible worlds approach). The exogenous approach has the advantage of uncoupling the two layers: the probabilistic connectives are defined independently of the original logic. Note that the global probabilistic connectives are still classical but should not be confused with the connectives of the original logic. Indeed, consider the following probabilistic formulae where φ is a classical formula: (i) $(\varphi \vee (\neg \varphi))$; (ii) $(\varphi \sqcup (\boxminus \varphi))$; and (iii) $(\varphi \sqcup (\neg \varphi))$. Clearly, (i) and (ii) hold in every Nilsson structure for every ground substitution, while (iii) does not hold in general.

Observe also that the exogenous probabilization procedure applied above to classical propositional logic is generic in the sense that it can be applied to any given logic as explained in [16]. So, probabilization is akin to temporalization and other cases of parameterization of logics. The same comment applies to the procedure described in Section 3 for building quantum logic.

Finally, the notion of probabilistic entailment is introduced as follows: $\Gamma \vdash \delta$ iff, for every $\mathbf{V}$ and ground ρ, $\mathbf{V}\rho \Vdash \delta$ whenever $\mathbf{V}\rho \Vdash \gamma$ for each $\gamma \in \Gamma$. This entailment enjoys the properties that one would expect from classical and probabilistic reasoning, such as:

[4]When defining languages, we use the abstract Backus-Naur notation [17], but adopting $\curlyvee$ instead of the traditional | in order to avoid confusions with the object language.

- $\vdash ((0 \leq (\int \varphi)) \sqcap ((\int \varphi) \leq 1))$;
- $\vdash ((\int(\neg\, \varphi)) = (1 - (\int \varphi)))$;
- $\vdash ((\int(\varphi_1 \vee \varphi_2)) = (((\int \varphi_1) + (\int \varphi_2)) - (\int(\varphi_1 \wedge \varphi_2))))$;
- $\varphi \vdash ((\int \varphi) = 1)$;
- $(\varphi_1 \Leftrightarrow \varphi_2) \vdash ((\int \varphi_1) = (\int \varphi_2))$;
- $(\varphi_1 \Rightarrow \varphi_2) \vdash ((\int \varphi_2 \,|\, \varphi_1) = 1)$.

A Hilbert calculus is proposed in [16] that is sound and weakly complete[5] with respect to the above semantics, with the following rules on probability:

PM	$\vdash ((\int \mathbf{t}) = 1)$;
FA	$\vdash (((\int(\neg(\xi_1 \wedge \xi_2))) = 1) \sqsupset ((\int(\xi_1 \vee \xi_2)) = ((\int \xi_1) + (\int \xi_2))))$;
CP	$\vdash (((\int \xi_2 \,|\, \xi_1)(\int \xi_1)) = (\int(\xi_1 \wedge \xi_2)))$;
UCP	$\vdash (((\int \xi_1) = 0) \sqsupset ((\int \xi_2 \,|\, \xi_1) = 1))$;
MON	$\vdash ((\xi_1 \Rightarrow \xi_2) \sqsupset ((\int \xi_1) \leq (\int \xi_2)))$;
MP	$\omega_1, (\omega_1 \sqsupset \omega_2) \vdash \omega_2$.

Observe that EPPL, the exogenous probabilistic propositional logic just described, allows us to work with both classical and probabilistic assertions. The former constrain the outcome space of the Nilsson structure and the latter constrain the probability measure. This feature was already present in the possible worlds probabilistic logics proposed in [10]. For a comparison between the two approaches see [16].

3 EQPL

Taking into account the postulates of quantum physics as stated for example in [18], the exogenous approach to quantum logic suggests that we should identify a quantum state (or quantum valuation) with a unit superposition of classical valuations. That is, the envisaged quantum logic should provide the means for reasoning about a quantum system composed of a denumerable set of qubits (one for each propositional constant $\mathcal{P}_k$) and where the (classical projective) observation values are classical valuations. The basic idea is to find a suitable Hilbert space containing the envisaged superpositions of classical valuations. Given a nonempty set V of observable classical valuations, $\mathcal{H}(V)$ is the following inner product space over $\mathbb{C}$:

- each element is a map $|w\rangle : V \to \mathbb{C}$ such that:
  - $\mathrm{supp}(|w\rangle) = \{v : |w\rangle(v) \neq 0\}$ is countable;
  - $\sum_{v \in \mathrm{supp}(|w\rangle)} ||w\rangle(v)|^2 < \infty$.
- $|w_1\rangle + |w_2\rangle = \lambda v.\, |w_1\rangle(v) + |w_2\rangle(v)$.
- $\alpha|w\rangle = \lambda v.\, \alpha|w\rangle(v)$.
- $\langle w_1|w_2\rangle = \sum_{v \in V} |w_1\rangle(v)\overline{|w_2\rangle(v)}$.

[5] Since we are using only finitary rules, strong completeness is out of question because probabilistic entailment (as defined herein) is not compact.

As usual, the inner product induces the norm $|||w\rangle|| = \sqrt{\langle w|w\rangle}$ and, so, the distance $d(|w_1\rangle, |w_2\rangle) = |||w_1\rangle - |w_2\rangle||$. Since $\mathcal{H}(V)$ is complete for this distance, $\mathcal{H}(V)$ is a Hilbert space[6]. Clearly, $\{|v\rangle : v \in V\}$ is an orthonormal basis of $\mathcal{H}(V)$ where $|v\rangle(v) = 1$ and $|v\rangle(v') = 0$ for every $v' \neq v$.

A quantum structure $\mathbf{w}$ is a pair $\langle V, |w\rangle\rangle$ where: V is a nonempty set of classical valuations; and $|w\rangle \in \mathcal{H}(V)$ such that $|||w\rangle|| = 1$. This structure provides the means for reasoning about a quantum system composed of a denumerable set of qubits (one for each $\mathcal{P}_k$) such that by observing it we get a classical valuation in V. The current state of the system is the unit vector $|w\rangle$ (a unit superposition of the observable classical valuations). The stochastic result of observing the system at that quantum state is described by the Nilsson structure $\mathcal{N}(\mathbf{w}) = \langle V, \wp V, \nu_{|w\rangle}\rangle$ where, for each $U \subseteq V$, $\nu_{|w\rangle}(U) = \sum_{u \in U} |\langle u|w\rangle|^2$.

Given a set S of propositional constants (qubits), we denote by $V_{[S]}$ the set $\{v|_S : v \in V\}$ and by $V_{]S[}$ the set $\{v|_{S^c} : v \in V\}$. The Hilbert spaces $\mathcal{H}(V_{[S]})$ and $\mathcal{H}(V_{]S[})$ will be useful when asserting facts about a target set S of qubits. Clearly, $\mathcal{H}(\mathcal{V}) = \mathcal{H}(\mathcal{V}_{[S]}) \otimes \mathcal{H}(\mathcal{V}_{]S[})$ where $\mathcal{V}$ is the set of all classical valuations. But, $\mathcal{H}(V) \subseteq \mathcal{H}(V_{[S]}) \otimes \mathcal{H}(V_{]S[})$ where equality does not hold in general. When it does, we say that the quantum system is composed of two independent subsystems (one with the qubits in S and the other with rest of the qubits). Furthermore, given a unit $|w\rangle \in \mathcal{H}(V)$, if there are unit $|w'\rangle \in \mathcal{H}(V_{[S]})$ and unit $|w''\rangle \in \mathcal{H}(V_{]S[})$ such that $|w\rangle = |w'\rangle \otimes |w''\rangle$ then we say that, in state $|w\rangle$, the qubits in S are not entangled with the qubits not in S and, therefore, that the qubits in S are independent of the other qubits at that state $|w\rangle$.

This exogenous semantics for reasoning about a quantum system (and its subsystems) suggests that we adopt the quantum language composed of formulae of the form[7]

$$\gamma := \omega_k \curlyvee \varphi \curlyvee (t \leq t) \curlyvee ([S] \diamond \overrightarrow{\psi : u}) \curlyvee (\boxminus \gamma) \curlyvee (\gamma \sqsupset \gamma)$$

where φ is a classical formula, t is a real term, u is a complex term, S is a nonempty recursive set of propositional constants (qubits), and ψ is a classical formula over S. The classical language is as before. The enriched set of real terms and the set of complex terms are jointly defined as follows:

$$\begin{cases} t := \theta_k \curlyvee r \curlyvee (\int \varphi) \curlyvee (\int \varphi \,|\, \varphi) \curlyvee (t + t) \curlyvee (t\, t) \curlyvee \mathrm{Re}(u) \curlyvee \mathrm{Im}(u) \curlyvee \arg(u) \curlyvee |u| \\ u := \upsilon_k \curlyvee (t + it) \curlyvee te^{it} \curlyvee \overline{u} \curlyvee (u + u) \curlyvee (u\, u) \end{cases}.$$

The denotation at $\mathbf{w}$ of ground terms is straightforward, but it is worthwhile to mention that the probability terms are interpreted using $\mathcal{N}(\mathbf{w})$. For instance: $[\![(\int \varphi)]\!]_{\mathbf{w}} = \nu_{|w\rangle}([\varphi]_{\mathbf{w}})$. The satisfaction of formulae by $\mathbf{w}$ and ground substitution ρ is as follows:

- $\mathbf{w}\rho \Vdash \omega_j$ iff $\mathbf{w}\rho \Vdash \omega_j\rho$;
- $\mathbf{w}\rho \Vdash \varphi$ iff $v \Vdash \varphi\rho$ for every $v \in V$;
- $\mathbf{w}\rho \Vdash (t_1 \leq t_2)$ iff $[\![t_1\rho]\!]_{\mathbf{w}} \leq [\![t_2\rho]\!]_{\mathbf{w}}$;
- $\mathbf{w}\rho \Vdash ([S] \diamond \psi_1 : u_1, \ldots, \psi_n : u_n)$ iff there are unit $|w'\rangle \in \mathcal{H}(V_{[S]})$ and unit $|w''\rangle \in \mathcal{H}(V_{]S[})$ such that $|w\rangle = |w'\rangle \otimes |w''\rangle$ and there are distinct $v_1, \ldots, v_n \in \mathrm{supp}(|w'\rangle)$ such that $v_k \Vdash \psi_k\rho$ and $|w'\rangle(v_k) = [\![u_k\rho]\!]_{\mathbf{w}}$ for $k = 1, \ldots, n$;
- $\mathbf{w}\rho \Vdash (\boxminus \alpha)$ iff $\mathbf{w}\rho \nVdash \alpha$;
- $\mathbf{w}\rho \Vdash (\alpha_1 \sqsupset \alpha_2)$ iff $\mathbf{w}\rho \nVdash \alpha_1$ or $\mathbf{w}\rho \Vdash \alpha_2$.

The language of EQPL, the exogenous quantum propositional logic just introduced, is quite powerful. Some abbreviations are useful for expressing some important derived concepts:

- $(\diamond \varphi_1 : u_1, \ldots, \varphi_n : u_n)$ for $([\{\mathcal{P}_k : k \in \mathbb{N}\}] \diamond \varphi_1 : u_1, \ldots, \varphi_n : u_n)$;

[6] Actually, $\mathcal{H}(V)$ is isomorphic to L^2 over the counting measure on V. Since we have in mind applications to quantum computation and information where all bits have the same importance, the choice of the counting measure is appropriate. However, other applications may require a different approach leading to a quite different logic. For instance, when looking at a quantum system with a physical real quantity such that each $\mathcal{P}_k$ reflects the k-th bit in its binary representation (and we could use sequences of other propositional symbols for representing other quantities), the Hilbert space proposed above would not be suitable. We should use instead L^2 over the Lebesgue measure on $[0, 1]$.

[7] Here we need to extend the Backus-Naur notation: we write $\vec{\delta}$ for a finite sequence of elements of the form δ.

- $[S]$ for $([S] :)$ — qubits in S are not entangled with those outside S;
- $(\Diamond\varphi)$ for $((\int\varphi) > 0)$ and $(\Box\varphi)$ for $((\int\varphi) = 1)$;
- $(\bigwedge_F A)$ for $((\wedge_{p_k\in A}\,\mathcal{P}_k) \wedge (\wedge_{p_k\in(F\setminus A)}\,(\neg\,\mathcal{P}_k)))$ whenever F is a finite set of propositional constants and $A \subseteq F$.

As an illustration, consider the following consistent assertions about an enriched Schrödinger's cat (where **cat-in-box**, **cat-alive** and **cat-moving** are propositional constants):

- $[\mathbf{cat\text{-}in\text{-}box}, \mathbf{cat\text{-}alive}, \mathbf{cat\text{-}moving}]$;
- $(\mathbf{cat\text{-}in\text{-}box} \wedge (\mathbf{cat\text{-}moving} \Rightarrow \mathbf{cat\text{-}alive}))$;
- $((\Diamond\mathbf{cat\text{-}alive}) \sqcap (\Diamond(\neg\,\mathbf{cat\text{-}alive})))$;
- $([\mathbf{cat\text{-}alive}, \mathbf{cat\text{-}moving}] \sqcap (\boxminus[\mathbf{cat\text{-}alive}]))$;
- $(((\int\mathbf{cat\text{-}alive}) = \frac{1}{3}) \sqcap ((\int\mathbf{cat\text{-}moving}\,|\,\mathbf{cat\text{-}alive}) = \frac{1}{2}))$;
- $([\mathbf{cat\text{-}alive}, \mathbf{cat\text{-}moving}] \Diamond \mathbf{cat\text{-}alive} : \frac{1}{\sqrt{6}}, \mathbf{cat\text{-}alive} : \frac{1}{\sqrt{6}})$;
- $([\mathbf{cat\text{-}alive}, \mathbf{cat\text{-}moving}] \Diamond (\mathbf{cat\text{-}alive} \wedge \mathbf{cat\text{-}moving}) : \frac{1}{\sqrt{6}},$
  $(\mathbf{cat\text{-}alive} \wedge (\neg\,\mathbf{cat\text{-}moving})) : \frac{1}{\sqrt{6}},$
  $((\neg\,\mathbf{cat\text{-}alive}) \wedge (\neg\,\mathbf{cat\text{-}moving})) : e^{i\frac{\pi}{3}}\sqrt{\frac{2}{3}})$.

The notion of quantum entailment is introduced as expected: $\Gamma \vdash \delta$ iff, for every quantum structure $\mathbf{w}$ and ground substitution ρ, $\mathbf{w}\rho \Vdash \delta$ whenever $\mathbf{w}\rho \Vdash \gamma$ for each $\gamma \in \Gamma$. Our ultimate goal is to establish a deduction calculus complete in some useful sense with respect to this semantics. Meanwhile, the following are examples of interesting entailments:

- $\vdash (\boxminus([S] \Diamond \psi : 0))$;
- $\vdash (([S] \Diamond \psi_1 : u_1, \ldots, \psi_n : u_n) \sqsupset ([S] \Diamond \psi_k : e^{it}u_k))$ for $k = 1, \ldots, n$;
- $\vdash (([S] \Diamond (\psi \vee \psi') : u) \equiv (([S] \Diamond \psi : u) \sqcup ([S] \Diamond \psi' : u)))$;
- $\vdash (([S] \Diamond \psi_1 : u_1, \ldots, \psi_n : u_n) \sqsupset (((|u_1|^2 + \cdots + |u_n|^2) \leq (\int(\psi_1 \vee \ldots \vee \psi_n)))))$;
- $\vdash (([F] \Diamond (\bigwedge_F A) : u) \sqsupset ((\int(\bigwedge_F A)) \leq |u|^2))$;
- $\vdash ((\psi_1 \Rightarrow \psi_2) \sqsupset (([S] \Diamond \psi_1 : u) \leq ([S] \Diamond \psi_2 : u)))$.

4 DEQPL

For reasoning about changes in the state of a quantum system (including transitions resulting from projective observations of a single qubit), we need to enrich the language. Following [18], according to postulate 2 of quantum physics, the evolution of a closed quantum system is described by a unitary operator. And postulate 3 tells us what is the resulting state when a qubit is projectively observed with result $|b\rangle$ in $\mathcal{H}(2)$. We need transition terms for denoting all such state transitions, of the form

$$Z := \tau_k \curlyvee U \curlyvee P \curlyvee (Z \circ Z)$$

where U is a unitary operator term of the form

$$U := \mathbf{I} \curlyvee \mathbf{H}_k \curlyvee \mathbf{S}_k \curlyvee \left(\frac{\pi}{\mathbf{8}}\right)_k \curlyvee \mathbf{X}_k \curlyvee \mathbf{Y}_k \curlyvee \mathbf{Z}_k \curlyvee \mathbf{cN}_{k_2}^{k_1} \curlyvee U^{-1} \curlyvee (U \circ U)$$

and P is a projective observation transition term of the form

$$P := \mathbf{P}_k^{c|\mathbf{0}\rangle + c|\mathbf{1}\rangle}$$

where c is a complex number of the form $r+ir$ where r is, as before, a computable real number. The eight symbols in U denote the eight basic unitary operators (identity, Hadamard, phase, $\pi/8$, Pauli X, Y, Z, and control not, respectively). Any finitary, unitary operator can be approximated as close as desired by a finite composition of these basic operators [9]. We also need transition formulae of the form[8]

$$H := \{\gamma\}\, Z\, \{\gamma\} \bigvee \{\gamma\}\, \Omega Z$$

where γ is a quantum formula as defined in the previous section.

The denotation $[\![Z]\!]$ of a transition term Z is a partial map from the unit circle of $\mathcal{H}(\mathcal{V})$ to itself (recall that $\mathcal{V}$ is the set of all classical valuations). In the case of every unitary operator term this map is total. Partiality only arises for observation transitions (as illustrated below).

Observe that, given $V \subseteq \mathcal{V}$, it may happen that $[\![Z]\!]|w\rangle \notin \mathcal{H}(V)$ even when $|w\rangle \in \mathcal{H}(V)$ and $[\![Z]\!]$ is defined on $|w\rangle$. We must keep this in mind when defining the satisfaction of transition formulae[9]:

- $V\rho \Vdash \{\gamma_1\}\, Z\, \{\gamma_2\}$ iff, for every $|w\rangle \in \mathcal{H}(V)$, if $\langle V, |w\rangle\rangle\rho \Vdash \gamma_1$ then $\langle V, [\![Z]\!]|w\rangle\rangle\rho \Vdash \gamma_2$ whenever $[\![Z]\!]|w\rangle\!\downarrow$ and $[\![Z]\!]|w\rangle \in \mathcal{H}(V)$;
- $V\rho \Vdash \{\gamma\}\, \Omega Z$ iff, for every $|w\rangle \in \mathcal{H}(V)$, if $\langle V, |w\rangle\rangle\rho \Vdash \gamma$ then $[\![Z]\!]|w\rangle\!\downarrow$ and $[\![Z]\!]|w\rangle \in \mathcal{H}(V)$.

That is, $\{\gamma_1\}\, Z\, \{\gamma_2\}$ means that if the quantum system evolves by Z from a state where γ_1 holds to a legitimate state (that is, in $\mathcal{H}(V)$) then γ_2 holds at the resulting state. If the resulting state is not legitimate the transition formula is vacuously satisfied. And $\{\gamma\}\, \Omega Z$ means that the quantum system reaches a legitimate state when it evolves by Z from a state where γ holds.

It is worthwhile to spell out in detail the semantics of the basic unitary operators. To this end, we need the notion of the dual of a valuation on a qubit: $\overline{v}^k$ is the valuation that agrees with v on all propositional symbols barring $\mathcal{P}_k$ and gives the other Boolean value to $\mathcal{P}_k$. For instance:

- $[\![\mathbf{H}_k]\!]|w\rangle(v) = \begin{cases} \frac{1}{\sqrt{2}}(|w\rangle(v) + |w\rangle(\overline{v}^k)) & \text{if } v \nVdash \mathcal{P}_k \\ \frac{1}{\sqrt{2}}(|w\rangle(\overline{v}^k) - |w\rangle(v)) & \text{otherwise} \end{cases}$;
- $[\![\mathbf{S}_k]\!]|w\rangle(v) = \begin{cases} |w\rangle(v) & \text{if } v \nVdash \mathcal{P}_k \\ i|w\rangle(v) & \text{otherwise} \end{cases}$;
- $[\![\mathbf{cN}_{k_2}^{k_1}]\!]|w\rangle(v) = \begin{cases} |w\rangle(v) & \text{if } v \nVdash \mathcal{P}_{k_1} \\ |w\rangle(\overline{v}^{k_2}) & \text{otherwise} \end{cases}$.

Before describing the semantics of the projective observation operators, we need some notation. Given a set S of propositional constants (qubits), we denote by $I_{[S]}$ the identity operator on $\mathcal{H}(\mathcal{V}_{[S]})$ and by $I_{]S[}$ the identity operator on $\mathcal{H}(\mathcal{V}_{]S[})$. Given $|b\rangle = \alpha_0|\mathbf{0}\rangle + \alpha_1|\mathbf{1}\rangle$ in $\mathcal{H}(2)$ we also need to use the projector along $|b\rangle$, the operator $|b\rangle\langle b|$ on $\mathcal{H}(2)$ defined by the following matrix:

$$\begin{pmatrix} \alpha_0\overline{\alpha_0} & \alpha_0\overline{\alpha_1} \\ \alpha_1\overline{\alpha_0} & \alpha_1\overline{\alpha_1} \end{pmatrix}.$$

Letting $P_k^{|b\rangle}$ be the projector along $|b\rangle$ for qubit k in $\mathcal{H}(\mathcal{V})$ that is given by $I_{[\{\mathcal{P}_0,\ldots,\mathcal{P}_{k-1}\}]} \otimes |b\rangle\langle b| \otimes I_{]\{\mathcal{P}_0,\ldots,\mathcal{P}_k\}[}$, the semantics of the projective observation transition terms is as follows:

- $[\![\mathbf{P}_k^{c_0|\mathbf{0}\rangle+c_1|\mathbf{1}\rangle}]\!]|w\rangle = \frac{P_k^{c_0|\mathbf{0}\rangle+c_1|\mathbf{1}\rangle}|w\rangle}{||P_k^{c_0|\mathbf{0}\rangle+c_1|\mathbf{1}\rangle}|w\rangle||}$.

Observe that $[\![\mathbf{P}_k^{c_0|\mathbf{0}\rangle+c_1|\mathbf{1}\rangle}]\!]$ is undefined at $|w\rangle$ if $||P_k^{c_0|\mathbf{0}\rangle+c_1|\mathbf{1}\rangle}|w\rangle|| = 0$. In particular, $[\![\mathbf{P}_k^{|\mathbf{0}\rangle}]\!]$ is undefined at $|w\rangle$ whenever $|w\rangle \Vdash ((\int(\neg\, \mathcal{P}_k)) = 0)$. In fact, it is not possible to observe 0 on $\mathcal{P}_k$ when all valuations in the support of the state of the system satisfy $\mathcal{P}_k$.

The projective observation transition terms play the role of qubit assignments in quantum computation since they impose the superposition of the target qubit in the resulting state. But, contrarily to classical

[8] Adapting from the Hoare (pre- and post-condition) triplets in the logic of imperative programs [13].

[9] As usual when dealing with partial maps, we write $[\![Z]\!]|w\rangle\!\downarrow$ for asserting that $[\![Z]\!]$ is defined on $|w\rangle$.

computation, an assignment to qubit $\mathcal{P}_k$ may also affect other qubits (those that were entangled with $\mathcal{P}_k$). For instance, the transition formula

$$\begin{array}{l}\{([\mathbf{Earth\text{-}cat\text{-}alive}, \mathbf{Mars\text{-}cat\text{-}alive}] \Diamond \\ \qquad\qquad (\mathbf{Earth\text{-}cat\text{-}alive} \wedge \mathbf{Mars\text{-}cat\text{-}alive}) : \tfrac{1}{\sqrt{2}}, \\ \qquad\qquad ((\neg\, \mathbf{Earth\text{-}cat\text{-}alive}) \wedge (\neg\, \mathbf{Mars\text{-}cat\text{-}alive})) : \tfrac{1}{\sqrt{2}})\} \\ \mathbf{P}^{|\mathbf{0}\rangle}_{\mathbf{Earth\text{-}cat\text{-}alive}} \\ \{([\mathbf{Mars\text{-}cat\text{-}alive}] \Diamond (\neg\, \mathbf{Mars\text{-}cat\text{-}alive}) : 1)\}\end{array}$$

states, among other things, that if the two cats are entangled then after observing the Earth cat dead we end up in a state where the Mars cat is also dead.

The proposed quantum transition language is quite powerful. We envisage to set up a relatively complete calculus for DEQPL (the dynamic exogenous quantum propositional logic with the above semantics). The key ingredients of this calculus will be the rules for the primitive operators since dealing with composition and relating with valid formulae of EQPL will be straightforward.

5 Concluding remarks

The proposed exogenous approach to quantum reasoning provided us with a working semantics for a powerful quantum logic. The resulting logic is promising and interesting in itself, but further work is necessary, namely towards a complete axiomatization and a clarification of the relationship to other quantum logics.

Since it is feasible to repeat the construction starting from any other logic (fulfilling some weak requirements), it seems worthwhile to investigate the construction of quantum logics as a form of parameterization of logics (as defined in [5]), like it is done for probabilistic logic in [16].

Assessing the effective role of the chosen basis for $\mathcal{H}(V)$ is also an interesting line of research. Indeed, EQPL satisfaction, as defined herein, strongly relies upon using the orthonormal basis $\{|v\rangle : v \in V\}$. One wonders if we can relax the semantics, while preserving the intended entailment, in order to be able to deal with classical formulae when we do not know V but we are just given a Hilbert space isomorphic to $\mathcal{H}(V)$.

Acknowledgments

The authors wish to express their deep gratitude to João Marcos who carefully proofread the draft and to the regular participants in the QCI Seminar who suffered early presentations of this work and gave very useful feedback that helped us to get over initial difficulties and misunderstandings of quantum physics, specially Jorge Buescu, Manuel Ricou, António Serra, Gabriel Pires, Pedro Resende, Vítor Rocha Vieira, Walter Carnielli, João Pimentel Nunes, and, last but not least, José Cidade Mourão.

This work was partially supported by FCT and FEDER through POCTI, namely via FibLog 2001/ MAT/37239 Project and within the recent QuantLog initiative of CLC.

Bibliography

[1] M. Abadi and J. Y. Halpern. Decidability and expressiveness for first-order logics of probability. *Information and Computation*, 112(1):1–36, 1994.

[2] F. Bacchus. On probability distributions over possible worlds. In *Uncertainty in Artificial Intelligence, 4*, volume 9 of *Machine Intelligence and Pattern Recognition*, pages 217–226. North-Holland, 1990.

[3] F. Bacchus. *Representing and Reasoning with Probabilistic Knowledge*. MIT Press Series in Artificial Intelligence. MIT Press, 1990.

[4] G. Birkhoff and J. von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37(4):823–843, 1936.

[5] C. Caleiro, C. Sernadas, and A. Sernadas. Parameterisation of logics. In J. Fiadeiro, editor, *Recent Trends in Algebraic Development Techniques - Selected Papers*, volume 1589 of *Lecture Notes in Computer Science*, pages 48–62. Springer-Verlag, 1999.

[6] W. A. Carnielli. Possible-translations semantics for paraconsistent logics. In *Frontiers of Paraconsistent Logic (Ghent, 1997)*, volume 8 of *Studies in Logic and Computation*, pages 149–163. Research Studies Press, 2000.

[7] W. A. Carnielli and M. Lima-Marques. Society semantics and multiple-valued logics. In *Advances in Contemporary Logic and Computer Science (Salvador, 1996)*, volume 235 of *Contemporary Mathematics*, pages 33–52. AMS, 1999.

[8] M. L. D. Chiara, R. Giuntini, and R. Greechie. *Reasoning in Quantum Theory.* Kluwer Academic Publishers, 2004.

[9] D. P. DiVincenzo. Two-bit gates are universal for quantum computation. *Physics Reviews A*, 51(2):1015–1022, 1995.

[10] R. Fagin, J. Y. Halpern, and N. Megiddo. A logic for reasoning about probabilities. *Information and Computation*, 87(1-2):78–128, 1990.

[11] D. J. Foulis. A half-century of quantum logic. What have we learned? In *Quantum Structures and the Nature of Reality*, volume 7 of *Einstein Meets Magritte*, pages 1–36. Kluwer Acad. Publ., 1999.

[12] S. Hart and M. Sharir. Probabilistic propositional temporal logics. *Information and Control*, 70(2-3):97–155, 1986.

[13] C. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12:576–583, 1969.

[14] S. A. Kripke. Semantical analysis of modal logic. I. Normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[15] P. Mateus, A. Pacheco, J. Pinto, A. Sernadas, and C. Sernadas. Probabilistic situation calculus. *Annals of Mathematics and Artificial Intelligence*, 32(1/4):393–431, 2001.

[16] P. Mateus, A. Pacheco, A. Sernadas, and C. Sernadas. Exogenous probabilization of an arbitrary logic. In preparation, 2004.

[17] P. Naur. Revised report on the algorithmic language Algol 60. *The Computer Journal*, 5:349–367, 1963.

[18] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[19] N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–87, 1986.

[20] N. J. Nilsson. Probabilistic logic revisited. *Artificial Intelligence*, 59(1-2):39–42, 1993.

Preservation of Interpolation by Fibring

Walter A. Carnielli[1] Cristina Sernadas[2] Alberto Zanardo[3]

[1] CLE and Department of Philosophy/IFCH, UNICAMP, Brazil

[2] CLC, Department Mathematics, IST, Portugal

[3] Department of Pure and Applied Mathematics, University of Padova, Italy

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

The method of fibring for combining logics as originally proposed by Gabbay [13, 14], includes some other methods as fusion [29] as a special case. Albeit fusion is the best developed mechanism, mainly in what concerns preservation of properties as soundness, weak completeness, semantic Craig interpolation and decidability (see [31, 17]), fibring in general raises some difficulties at the semantic level [26]. The general quest for *preservation* represents one of the main research trends in fibring.

Although preservation of soundness and completeness has been investigated in the context of propositional-based logics [32, 28], first-order quantification [27], higher-order features [9], non truth-functional semantics [4], sequent and other deduction calculi [16, 24], other forms of preservation are still to be investigated.

We outline here some results on preservation of interpolation in the context of propositional-based logics endowed with a Hilbert calculus coping with global and local derivability consequences (see [8] for further details).

What is now generally known as Craig interpolation is a heritage of the classical results by W. Craig [10] for first-order logic. Several abstractions have been considered either in proof-theoretical way (e.g. [5]) or in (non-constructive) model-theoretical style (e.g. for modal and positive logics as in [20, 21], for intuitionistic logic as in [15] and for hybrid logic as in [1, 2]). The importance of Craig interpolation for some fundamental problems of complexity theory as shown in [22] and further developed in [23] associates the rate of growth of the interpolant and measures of complexity.

Interpolation properties are known to be related with properties of model theory as exemplified by the correspondence between Craig interpolation and joint consistency properties for classical propositional logic. This correspondence is mediated in the classical case by finite algebraizability and by the familiar (global) metatheorem of deduction.

In the general case of deducibility relations, however, specially in those where the peculiarities of local and global deduction interfere, this correspondence opens difficult and challenging problems. We refer here to *careful reasoning* when the distinction of global and local deduction is relevant: careful reasoning may lead to other forms of interpolation even at propositional level. In the sequel we use the labels l-property and g-property when it makes sense to distinguish between local and global versions of that property; we write d-property when there is no distinction between them.

The importance of a general form of metatheorem of deduction is stressed, proving that Craig interpolation implies another form of interpolation proposed by S. Maehara [19], thus showing that the mediation of metatheorem of deduction plays a central role. Moreover, in certain specific cases, local interpolation implies global interpolation.

A result on preservation of Craig interpolation for fusion of modal logics restricted to global reasoning was obtained in [17]. Herein we study this question in a much broader sense for a wide-scoped fibring combinations covering global and local reasoning and encompassing several logics besides the modal ones. We establish sufficient conditions for preservation of interpolation by fibring using a translation of formulae from the fibring to the components. Along the way we show the preservation of a generalized version of the deduction metatheorem.

General techniques for obtaining interpolation are not known in general; Craig interpolation for example fails unexpectedly in all Łukasiewicz logics L_n with n finite or infinite see [18], and also in all Gödel logics G_n for $n \geq 4$, see [3]. Developing constructive proofs of interpolation is still a harder problem. We obtain here constructive methods of Craig interpolation for special logics as it is the case of some many-valued logics and logics of formal inconsistency (as studied in [6]).

2 Preliminaries

A *signature* C is an $\mathbb{N}$-indexed family of countable sets. The elements of each C_k are called *constructors* of arity k. Let $sL(C, \Xi)$ be the free algebra over C generated by Ξ. We denote by $\mathrm{var}(\varphi)$ the set of elements of Ξ occurring in φ and $\mathrm{var}(\Gamma) = \cup_{\gamma\in\Gamma}\mathrm{var}(\gamma)$. A *substitution* is any map $\sigma : \Xi \to sL(C, \Xi)$. Substitutions can inductively extended to formulae: $\sigma(\gamma)$ is the formula where each $\xi \in \Xi$ is replaced by $\sigma(\xi)$ and also to sets: $\sigma(\Gamma) = \{\sigma(\gamma) : \gamma \in \Gamma\}$. When $\mathrm{var}(\varphi) = \{\xi_1, \ldots, \xi_n\}$ and $\sigma(\xi_i) = \psi_i$ for $i = 1, \ldots, n$, we may use $\varphi(\psi_1, \ldots, \psi_n)$ to denote $\sigma(\varphi)$.

A *rule* over C is a pair $r = \langle\Theta, \eta\rangle$ where $\Theta \cup \{\eta\} \subseteq sL(C, \Xi)$ and Θ is finite. A *deductive system* is a triple $\mathcal{D} = \langle C, R_l, R_{\mathrm{g}}\rangle$ where C is a signature and both R_l and R_{g} are sets of rules over C such that

$R_l \subseteq R_{\rm g}$. The rules in R_l are called *local rules* and those in $R_{\rm g}$ are called *global rules*. The distinction between local and global rules is imparted in the concept of careful reasoning and is reflected when investigating metatheoretical properties and their preservation.

A *global derivation* of $\varphi \in sL(C,\Xi)$ from $\Gamma \subseteq sL(C,\Xi)$, indicated by $\Gamma \vdash^{\rm g}_{\mathcal{D}} \varphi$, is a sequence $\psi_1 \dots \psi_n$ such that ψ_n is φ and each ψ_i is either an element of Γ or there are a rule $r = \langle\{\theta_1,\dots,\theta_m\},\eta\rangle \in R_{\rm g}$ and a substitution σ such that ψ_i is $\sigma(\eta)$ and $\sigma(\theta_j)$ appears in $\psi_1 \dots \psi_{i-1}$ for every $j = 1,\dots,m$. A *local derivation* of $\varphi \in sL(C,\Xi)$ from $\Gamma \subseteq sL(C,\Xi)$, indicated by $\Gamma \vdash^{l}_{\mathcal{D}} \varphi$, is a sequence $\psi_1 \dots \psi_n$ such that ψ_n is φ and each ψ_i is either an element of Γ, is globally derivable from the empty set or there are a rule $r = \langle\{\theta_1,\dots,\theta_m\},\eta\rangle \in R_l$ and a substitution σ such that ψ_i is $\sigma(\eta)$ and $\sigma(\theta_j)$ appears in $\psi_1 \dots \psi_{i-1}$ for every $j = 1,\dots,m$. We use the notation $\Gamma \vdash^{d}_{\mathcal{D}} \varphi$ when stating properties that hold either for global or for local derivation. We will extend the derivations to sets: $\Gamma \vdash^{\rm g}_{\mathcal{D}} \Psi$ with $\Psi \subseteq sL(C,\Xi)$ iff $\Gamma \vdash^{\rm g}_{\mathcal{D}} \psi$ for every $\psi \in \Psi$.

Several deduction metatheorems can be considered as indicated in [12]: they generalize the usual deduction metatheorems that require the existence of an implication in the signature. Herein, we consider extended versions taking into account global and local reasoning as follows: A deduction system $\mathcal{D}$ has the *d-metatheorem of deduction* (d-MTD) if there is a finite set of formulae $\Delta \subseteq sL(C,\{\xi_1,\xi_2\})$ such that if $\Gamma,\varphi_1 \vdash^{d}_{\mathcal{D}} \varphi_2$ then $\Gamma \vdash^{d}_{\mathcal{D}} \Delta(\varphi_1,\varphi_2)$. And it has the *d-metatheorem of modus ponens* (d-MTMP) if there is a finite set of formulae $\Delta \subseteq sL(C,\{\xi_1,\xi_2\})$ such that the converse holds.

3 Interpolation

We recast here some forms of interpolation taking into account the distinction between local and global deduction. A deductive system has the *d-extension interpolation property* (d-EIP) whenever $\Gamma,\Psi \vdash^{d}_{\mathcal{D}} \varphi$ implies that there is $\Gamma' \subseteq sL(C,\mathrm{var}(\Psi)\cup\mathrm{var}(\varphi))$ such that $\Gamma \vdash^{d}_{\mathcal{D}} \Gamma'$ and $\Gamma',\Psi \vdash^{d}_{\mathcal{D}} \varphi$, for every $\Gamma,\Psi \subseteq sL(C,\Xi)$ and $\varphi \in sL(C,\Xi)$. Although as mentioned before Łukasiewicz logics L_n with $n \geq 3$ do not have CIP they do enjoy EIP.

A deductive system has the *d-Craig interpolation property* (d-CIP) whenever $\Gamma \vdash^{d}_{\mathcal{D}} \varphi$ and $\mathrm{var}(\Gamma)\cap\mathrm{var}(\varphi) \neq \emptyset$ implies that there is $\Gamma' \subseteq sL(C,\mathrm{var}(\Gamma)\cap\mathrm{var}(\varphi))$ such that $\Gamma \vdash^{d}_{\mathcal{D}} \Gamma'$ and $\Gamma' \vdash^{d}_{\mathcal{D}} \varphi$, for every $\Gamma \subseteq sL(C,\Xi)$ and $\varphi \in sL(C,\Xi)$.

The relevance of careful reasoning is measured by the fact that in some cases it is also possible to relate local and global CIP. That is the case of deductive systems which share with modal and first-order logics the important property that we call *careful-reasoning-by-cases*. A deduction system $\mathcal{D}$ is said to enjoy *careful-reasoning-by-cases* iff whenever $\Gamma \vdash^{\rm g}_{\mathcal{D}} \varphi$ then there is Ψ such that $\Gamma \vdash^{\rm g}_{\mathcal{D}} \Psi$, $\mathrm{var}(\Psi) \subseteq \mathrm{var}(\Gamma)$ and $\Psi \vdash^{l}_{\mathcal{D}} \varphi$.

Theorem 1. A deductive system $\mathcal{D}$ enjoying careful-reasoning-by-cases has g-Craig interpolation whenever it has l-Craig interpolation property.

A deductive system has the *d-Maehara interpolation property* (d-MIP) whenever $\Gamma,\Psi \vdash^{d} \varphi$ and $sL(C,\mathrm{var}(\Gamma)\cap(\mathrm{var}(\Psi)\cup\mathrm{var}(\varphi))) \neq \emptyset$ implies that there is $\Gamma' \subseteq sL(C,\mathrm{var}(\Gamma)\cap(\mathrm{var}(\Psi)\cup\mathrm{var}(\varphi)))$ such that $\Gamma \vdash^{d} \Gamma'$ and $\Gamma',\Psi \vdash^{d} \varphi$, for every $\Gamma,\Psi \subseteq sL(C,\Xi)$ and $\varphi \in sL(C,\Xi)$. It is proved in [11] that a deductive system has g-MIP iff it has g-EIP and g-CIP. Careful reasoning allows to the following improvement that together with the result in [11] shows that d-MTD and d-EIP are provable from each other.

Theorem 2. A deductive system with d-MTD, d-MTMP and d-CIP has d-MIP.

4 Preserving metaproperties

Given two deductive systems $\mathcal{D}'$ and $\mathcal{D}''$, their *fibring* is the deductive system $\mathcal{D}$ defined as follows: $C_k = C'_k \cup C''_k$ for every $k \in \mathbb{N}$, $R_l = {R_l}' \cup {R_l}''$ and $R_{\rm g} = {R_{\rm g}}' \cup {R_{\rm g}}''$. Observe that the deductive system induced by $\mathcal{D}$ is not the union (in the sense of [30]) of consequence systems induced by $\mathcal{D}'$ and $\mathcal{D}''$ neither for local nor for global derivation. Moreover taking $\Gamma' \subseteq sL(C',\Xi)$ and $\Gamma'' \subseteq sL(C'',\Xi)$ in general we get that ${\Gamma'}^{\vdash'^{d}} \subset {\Gamma'}^{\vdash^{d}}$ and ${\Gamma''}^{\vdash''^{d}} \subset {\Gamma''}^{\vdash^{d}}$.

In order to analyze the preservation of MTD it is easier to provide an alternative characterization involving derivations in the object logic. We start by presenting a necessary and sufficient condition for a deductive system to have MTMP.

Lemma 3. A deductive system has d-MTMP iff there is a finite set $\Delta \subseteq sL(C, \{\xi_1, \xi_2\})$ of formulae such that $\Delta, \xi_1 \vdash^d_{\mathcal{D}} \xi_2$.

Lemma 4. A deductive system with d-MTMP has d-MTD iff there is a finite set $\Delta \subseteq sL(C, \{\xi_1, \xi_2\})$ of formulae such that: (1) $\{\xi_1\} \vdash^l_{\mathcal{D}} \Delta(\xi_2, \xi_1)$; (2) $\vdash^d_{\mathcal{D}} \Delta(\xi_1, \xi_1)$; and (3) $\Delta(\xi_1, \theta_1) \cup \cdots \cup \Delta(\xi_1, \theta_m) \vdash^d_{\mathcal{D}} \Delta(\xi_1, \eta)$ for each rule $r = \langle \{\theta_1, \ldots, \theta_m\}, \eta \rangle \in R_l$.

Theorem 5. d-MTMP and d-MTD are preserved by fibring deductive systems with the same Δ.

The relationship between global and local Craig interpolation in the fibring can be investigated. The first step is to be able to *transform* derivations in the fibring $\mathcal{D}$ into derivations in the components $\mathcal{D}'$ and $\mathcal{D}''$ by using a "ghost" technique and transformation maps τ' and τ'' over the enrichments of $\mathcal{D}'$ and $\mathcal{D}''$ with the ghosts.

Theorem 6. Careful-reasoning-by-cases is preserved by fibring deductive systems.

A deductive system has *conjunction* if there is a constructor $\wedge \in C_2$ and $\langle \{(\xi_1 \wedge \xi_2)\}, \xi_1 \rangle, \langle \{(\xi_1 \wedge \xi_2)\}, \xi_2 \rangle, \langle \{\xi_1, \xi_2)\}, (\xi_1 \wedge \xi_2) \rangle \in R_{\mathrm{g}}$.

Theorem 7. g-Craig interpolation is preserved by the fibring $\mathcal{D}$ of two deductive systems $\mathcal{D}'$ and $\mathcal{D}''$ with conjunction, g-MTMP, g-MTMP and an axiom that can be instanced with any finite number of variables.

For instance, in modal logic the axiom $(\xi_1 \Rightarrow (\xi_2 \Rightarrow \xi_1))$ can be instanced with any finite number of variables: the instance $((\wedge_{j=1}^k \xi_j) \Rightarrow (\xi_2 \Rightarrow (\wedge_{j=1}^k \xi_j)))$ includes $\xi_1, \ldots, \xi_k$.

Craig interpolation can also be investigated when no hypothesis are present and we are dealing with systems that have implication connective. A deductive system has the *implication connective* iff $\Rightarrow \in C_2$ and the d-MTD and the d-MTMP hold with $\Delta = \{(\xi_1 \Rightarrow \xi_2)\}$. A deductive system with implication has *imp-Craig interpolation* if $\vdash^{\mathrm{g}}_{\mathcal{D}} (\varphi_1 \Rightarrow \varphi_2)$ and $\mathrm{var}(\varphi_1) \cup \mathrm{var}(\varphi_2) \neq \emptyset$ then there is $\psi \in sL(C, \mathrm{var}(\varphi_1) \cup \mathrm{var}(\varphi_2))$ such that $\vdash^{\mathrm{g}}_{\mathcal{D}} (\varphi_1 \Rightarrow \psi)$ and $\vdash^{\mathrm{g}}_{\mathcal{D}} (\psi \Rightarrow \varphi_2)$.

Theorem 8. Imp-Craig interpolation is preserved by the fibring $\mathcal{D}$ of two deductive systems $\mathcal{D}'$ and $\mathcal{D}''$ both with the same implication and conjunction.

The preservation of l-Craig interpolation requires more assumptions on the component logics, namely a refinement of the notion of careful-reasoning-by-cases. A deductive system $\mathcal{D}$ has *localized-careful-reasoning-by-cases* if $\Gamma, \Psi \vdash^{\mathrm{g}}_{\mathcal{D}} \varphi$, Ψ is finite and with a derivation where rules in R_{g} are only applied to hypotheses in Ψ then there is a finite $\Omega \in sL(C, \mathrm{var}(\Psi))$ such that $\Omega \subseteq \Psi^{\vdash^{\mathrm{g}}_{\mathcal{D}}}$ and $\Omega \vdash^l_{\mathcal{D}} \varphi$. Both modal and first-order logics have this property.

Theorem 9. l-Craig interpolation is preserved by the fibring $\mathcal{D}$ of two deductive systems $\mathcal{D}'$ and $\mathcal{D}''$ both with localized-careful-reasoning-by-cases, conjunction, l-MTMP, l-MTD and an axiom that can be instanced with any finite number of variables.

Theorem 10. d-Maehara interpolation (d-MIP) is preserved by fibring deductive systems $\mathcal{D}'$ and $\mathcal{D}''$ both with d-MTMP, d-MTD, conjunction and an axiom that can be instanced with any number of variables (and with localized-careful-reasoning-by-cases).

5 Some logics with constructive Craig interpolation

Some constructive proofs of Craig interpolation can be given for deductive systems that enjoy certain properties. For this purpose we need a few semantic notions. A *matrix* is a triple $\langle B, \cdot, D \rangle$ where $m = \langle B, \cdot \rangle$ is an algebra over C (of values) and $D \subseteq B$ (the set of distinguished values). A *valuation* is any map from Ξ to B. The denotation of formula $[\![\varphi]\!]^m_v$ is defined inductively in the expected way. A formula φ is a *global semantic consequence* of Γ, written $\Gamma \vdash^{\mathrm{g}} \varphi$ if $[\![\varphi]\!]^m_v \in D$ whenever $[\![\gamma]\!]^m_v \in D$ for every $\gamma \in \Gamma$. When a logic has a unique matrix up to isomorphism we may use $v(\varphi)$ to refer to $[\![\varphi]\!]^m_v$.

A deductive system has *binary supremum* with respect to global derivation iff there is $\vee \in C_2$ such that (i) $\xi_i \vdash^{\mathrm{g}}_{\mathcal{D}} (\xi_1 \vee \xi_2)$ and (ii) if $\xi_i \vdash^{\mathrm{g}}_{\mathcal{D}} \psi$ then $(\xi_1, \vee \xi_2) \vdash^{\mathrm{g}}_{\mathcal{D}} \psi$ for $i = 1, 2$. A deductive system with binary supremum has any finite suprema as well. We represent by $(\xi_1 \vee \ldots \xi_n)$ the suprema of $\xi_1 \ldots, \xi_n$.

A deductive system is said to be *suitable* iff and there are $\delta_i \in sL(C, \{\xi_1\})$ with $i = 1, \ldots, n$ satisfying the following conditions:

1. for every $\varphi \in sL(C,\Xi)$ and δ_i there is $\theta_i^{\varphi} \in sL(C,\Xi)$ such that: (a) $\mathrm{var}(\theta_i^{\varphi}) \subseteq \mathrm{var}(\varphi) \setminus \{\xi_1\}$; (b) $\varphi, \delta_i \vdash^{\mathrm{g}}_{\mathcal{D}} \theta_i^{\varphi}$ and $\delta_i, \theta_i^{\varphi} \vdash^{\mathrm{g}}_{\mathcal{D}} \varphi$; (c) $\varphi, \delta_i \vdash^{\mathrm{g}}_{\mathcal{D}} \psi$ for every $i = 1, \ldots, n$ then $\varphi \vdash^{\mathrm{g}}_{\mathcal{D}} \psi$.

2. If $\gamma_1, \gamma \vdash^{\mathrm{g}}_{\mathcal{D}} \gamma_2$ and $\mathrm{var}(\gamma) \cap \mathrm{var}(\gamma_1) = \mathrm{var}(\gamma) \cap \mathrm{var}(\gamma_2) = \emptyset$ then $\gamma_1 \vdash^{\mathrm{g}}_{\mathcal{D}} \gamma_2$.

Conditions 1(a) and (b) mean that we can separate the formula into two equivalent formulae without sharing variables. Condition 1(c) indicates that in order to analyze φ it is enough to analyze δ_i for every $i = 1, \ldots, n$. We call property 2 the *omitting symbols property* that holds in every free algebra $sL(C,\Upsilon)$ where Υ is a finite subset of the set of variables Ξ and $sL(C,\Xi)$ is the union of such algebras [30] (Section 1.1). In classical logic, the omitting symbols property holds iff γ is a contradiction. If the deductive system is complete with respect to matrix semantics we can provide sufficient conditions for 2. Observe that if $\varphi \vdash^{\mathrm{g}} \psi$, φ is satisfiable and $\mathrm{var}(\varphi) \cap \mathrm{var}(\psi) = \emptyset$ then $\vdash^{\mathrm{g}} \psi$.

Lemma 11. Let $\mathcal{D}$ be a complete deductive system having implication. Assume that $\mathrm{var}(\gamma) \cap \mathrm{var}(\gamma_1) = \mathrm{var}(\gamma) \cap \mathrm{var}(\gamma_2) = \emptyset$, γ is satisfiable and for every formula α, $v(\alpha) = v'(\alpha)$ whenever $v(\xi) = v'(\xi)$ for each $\xi \in \mathrm{var}(\alpha)$. If $\gamma_1, \gamma \vdash^{\mathrm{g}}_{\mathcal{D}} \gamma_2$ then $\gamma_1 \vdash^{\mathrm{g}}_{\mathcal{D}} \gamma_2$.

Theorem 12. Every suitable deductive system with binary supremum and implication has g-Craig interpolation.

Constructive Craig interpolation can also be given for deductive systems allowing the possibility of expressing all truth values at the syntactical level. A logic $\mathcal{L}$ is *syntactically faithful* if its deductive system has binary supremum $\vee$ and there are $\beta_1, \ldots, \beta_n$ depending at most upon the variables $\xi_1, \ldots, \xi_n$ such that $v(\beta_i) = b_i$ for every valuation v and for every truth value $b_i \in B$. Several finite-valued and non-truth functional logics share this property. Besides Rosser-Turquette deductive systems [25] and Post systems, several logics of formal inconsistency such as **mbC**, **bC**, **Ci** and da Costa's $\mathcal{C}_n$ for $n \in \mathbb{N}$ are also syntactically faithful, see [7], and thus enjoy g-Craig interpolation.

Theorem 13. Every syntactically faithful logic has g-Craig interpolation.

Acknowledgments

This work was partially supported by *Fundação para a Ciência e a Tecnologia* and EU FEDER, namely via the Project FibLog POCTI/MAT/37239/2001 at CLC (Center for Logic and Computation) and by Conselho Nacional de Desenvolvimento Científico e Tecnológico CNPq in Brasil. The first author also acknowledges financial support from the CLC (IST, Portugal) for a senior scientist research grant.

Bibliography

[1] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3):977–1010, 2001.

[2] C. Areces, P. Blackburn, and M. Marx. Repairing the interpolation theorem in quantified modal logic. *Annals of Pure and Applied Logic*, 124(1-3):287–299, 2003.

[3] M. Baaz and H. Veith. Interpolation in fuzzy logic. *Archive for Mathematical Logic*, 38(7):461–489, 1999.

[4] C. Caleiro, W. A. Carnielli, M. E. Coniglio, A. Sernadas, and C. Sernadas. Fibring non-truth-functional logics: Completeness preservation. *Journal of Logic, Language and Information*, 12(2):183–211, 2003.

[5] A. Carbone. Interpolants, cut elimination and flow graphs for the propositional calculus. *Annals of Pure and Applied Logic*, 83(3):249–299, 1997.

[6] W. A. Carnielli, M. E. Coniglio, and J. Marcos. Logics of formal inconsistency. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 12. Kluwer Academic Publishers, in print.

[7] W. A. Carnielli and J. Marcos. A taxonomy of C-systems. In *Paraconsistency (São Sebastião, 2000)*, volume 228 of *Lecture Notes in Pure and Applied Mathematics*, pages 1–94. Dekker, New York, 2002.

[8] W. A. Carnielli and C. Sernadas. Preservation of interpolation features by fibring. Preprint, CLC, Department of Mathematics, Instituto Superior Técnico, 1049-001 Lisboa, Portugal, 2004. Submitted for publication.

[9] M. E. Coniglio, A. Sernadas, and C. Sernadas. Fibring logics with topos semantics. *Journal of Logic and Computation*, 13(4):595–624, 2003.

[10] W. Craig. Linear reasoning. A new form of the Herbrand-Gentzen theorem. *Journal of Symbolic Logic*, 22:250–268, 1957.

[11] J. Czelakowski and D. Pigozzi. Amalgamation and interpolation in abstract algebraic logic. In *Models, algebras, and proofs*, volume 203 of *Lecture Notes in Pure and Applied Mathematics*, pages 187–265. Dekker, 1999.

[12] J. M. Font, R. Jansana, and D. Pigozzi. A survey of abstract algebraic logic. *Studia Logica*, 74(1-2):13–97, 2003. Abstract algebraic logic, Part II (Barcelona, 1997).

[13] D. Gabbay. Fibred semantics and the weaving of logics: part 1. *Journal of Symbolic Logic*, 61(4):1057–1120, 1996.

[14] D. Gabbay. *Fibring Logics*. Oxford University Press, 1999.

[15] Dov M. Gabbay. Craig interpolation theorem for intuitionistic logic and extensions. III. *Journal of Symbolic Logic*, 42(2):269–271, 1977.

[16] G. Governatori, V. Padmanabhan, and A. Sattar. On fibring semantics for BDI logics. In S. Flesca and G. Ianni, editors, *Logics in computer science*, volume 2424 of *Lecture Notes in Artificial Intelligence*, pages 198–210. Springer Verlag, 2002.

[17] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *Journal of Symbolic Logic*, 56(4):1469–1485, 1991.

[18] P. S. Krzystek and S. Zachorowski. Ł ukasiewicz logics have not the interpolation property. *Rep. Math. Logic*, (9):39–40, 1977.

[19] S. Maehara. On the interpolation theorem of Craig. *Sûgaku*, 12:235–237, 1960/1961.

[20] L. Maksimova. Interpolation in superintuitionistic predicate logics with equality. *Algebra i Logika*, 36(5):543–561, 600, 1997.

[21] L. Maksimova. Complexity of interpolation and related problems in positive calculi. *Journal of Symbolic Logic*, 67(1):397–408, 2002.

[22] D. Mundici. Complexity of Craig's interpolation. *Fundamenta Informaticae*, 5(3-4):261–278, 1982.

[23] D. Mundici. NP and Craig's interpolation theorem. In *Logic colloquium '82 (Florence, 1982)*, volume 112 of *Stud. Logic Found. Math.*, pages 345–358. North-Holland, Amsterdam, 1984.

[24] J. Rasga, A. Sernadas, C. Sernadas, and L. Viganò. Fibring labelled deduction systems. *Journal of Logic and Computation*, 12(3):443–473, 2002.

[25] J. B. Rosser and A. R. Turquette. *Many-valued Logics*. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam, 1951.

[26] A. Sernadas, C. Sernadas, and C. Caleiro. Fibring of logics as a categorial construction. *Journal of Logic and Computation*, 9(2):149–179, 1999.

[27] A. Sernadas, C. Sernadas, and A. Zanardo. Fibring modal first-order logics: Completeness preservation. *Logic Journal of the IGPL*, 10(4):413–451, 2002.

[28] C. Sernadas, J. Rasga, and W. A. Carnielli. Modulated fibring and the collapsing problem. *Journal of Symbolic Logic*, 67(4):1541–1569, 2002.

[29] R. H. Thomason. Combinations of tense and modality. In *Handbook of Philosophical Logic, Vol. II*, volume 165 of *Synthese Library*, pages 135–165. Reidel, 1984.

[30] R. Wójcicki. *Theory of Logical Calculi*, volume 199 of *Synthese Library*. Kluwer Academic Publishers Group, 1988.

[31] F. Wolter. Fusions of modal logics revisited. In *Advances in Modal Logic, Vol. 1*, volume 87 of *CSLI Lecture Notes*, pages 361–379. CSLI Publ., 1998.

[32] A. Zanardo, A. Sernadas, and C. Sernadas. Fibring: Completeness preservation. *Journal of Symbolic Logic*, 66(1):414–439, 2001.

Abstract Modalities and Institutions

Răzvan Diaconescu [1] Petros Stefaneas [2]

[1] Institute of Mathematics "Simion Stoilow", PO Box 1-764, Bucharest 70700, Romania
Razvan.Diaconescu@imar.ro

[1] National Technical University Athens, Greece
petros@math.ntua.gr

Abstract

We define abstract modal semantics using institutions. Modalities can then be generated over a wide variety of logics. Using tools from institution-independent model theory we state a preservation result for the modal satisfaction.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

Institutions [8] formalize the intuitive notion of logical system, using concepts from category theory. Their initial goal was "to do as much computing science as possible" independent of what the underlying logic may be. Recently, this 'institution-independent' approach found applications in model theory [5, 4, 2]. Our paper introduces a Kripke semantics at the level of institutions. Also, we state a fundamental preservation result, that each modal sentence is preserved by ultraproducts of frames. Following [5], logical connectives and quantifiers can be developed internally to any institution. Knowledge on institution-independent ultraproducts [5] is required.

2 Kripke frames

Definition 1. Assume an institution morphism $(\Phi^\Delta, \alpha^\Delta, \beta^\Delta) : \quad (\mathbb{S}ign, Sen, \text{Mod}, \models) \to (\mathbb{S}ign^\Delta, Sen^\Delta, \text{Mod}^\Delta, \models^\Delta)$. Given a signature Σ in $\mathbb{S}ign$, a Σ-*frame* (W, R) consists of

- a family of Σ-models $W : \ I_W \to |\text{Mod}(\Sigma)|$ such that *sharing condition* $\beta^\Delta_\Sigma(W^i) = \beta^\Delta_\Sigma(W^{i'})$ holds for each $i, i' \in I_W$, and
- an binary *accessibility relation* R on the index set I_W.

A Σ-*frame homomorphism* $(h^W, h^I) : \ (W, R) \to (W', R')$ consists of

- a function $h^I : \ I_W \to I_{W'}$ between the index sets which is a relation homomorphism, i.e. $\langle i, j\rangle \in R$ implies $\langle h^I(i), h^I(j)\rangle \in R'$, and
- a natural transformation $h^W : \ W \Rightarrow h^I; W'$ such that $\beta^\Delta_\Sigma((h^W)^i) = \beta^\Delta_\Sigma((h^W)^{i'})$ for each $i, i' \in I_W$.

For each $\text{K} \in \{T, S4, S5\}$, the Σ-frames and their homomorphisms form a category denoted K^Δ-Mod(Σ).

Definition 2. Given a signature morphism $\varphi : \ \Sigma \to \Sigma'$, each Σ'-frame (W', R') can be reduced to the Σ-frame $(W'; \text{Mod}(\varphi), R')$. Similarly, each frame homomorphism (h^W, h^I) can be reduced to $(h^W \text{Mod}(\varphi), h^I)$. This defines a frame model functor K^Δ-Mod$: \ \mathbb{S}ign^{op} \to \mathbb{C}at$.

3 Modal sentences

The definition below extends the internal logic of [5] with modalities.

Definition 3. Given any institution $(\mathbb{S}ign, Sen, \text{Mod}, \models)$, a functor M-$Sen : \ \mathbb{S}ign \to \mathbb{S}et$ is a *modal sentence functor over Sen* when

- for each signature Σ, each sentence in M-$Sen(\Sigma)$ can be obtained from the sentences of $Sen(\Sigma)$ by iterative application of
  - usual logical connectives (i.e. conjunction, disjunction, negation, implication, etc.),
  - modal (unary) connectives (i.e. necessity $\Box\_$, possibility $\Diamond\_$),
  - universal or existential quantification along signature morphisms, i.e. $(\forall\chi)\rho$ is a Σ-sentence when ρ is a Σ_1-sentence and $\chi : \ \Sigma \to \Sigma_1$ is a signature morphism.
- for each signature morphism $\phi : \ \Sigma \to \Sigma_1$,
  - M-$Sen(\phi)(\rho) = Sen(\phi)(\rho)$ when $\rho \in Sen(\Sigma)$,
  - M-$Sen(\phi)(\rho_1 \wedge \rho_2) =$ M-$Sen(\phi)(\rho_1) \wedge$ M-$Sen(\phi)(\rho_2)$; similarly for other logical connectives,
  - M-$Sen(\phi)(\Box\rho) = \Box$(M-$Sen(\phi)(\rho)$); similarly for $\Diamond$, and
  - M-$Sen(\phi)((\forall\chi)\rho) = (\forall\chi')$M-$Sen(\phi_1)(\rho)$ where

$$\begin{array}{ccc} \Sigma & \xrightarrow{\phi} & \Sigma' \\ {\scriptstyle\chi}\downarrow & & \downarrow{\scriptstyle\chi'} \\ \Sigma_1 & \xrightarrow[\phi_1]{} & \Sigma'_1 \end{array}$$

  is a pushout of signatures; similarly for existential quantification.

4 Modal satisfaction

Definition 4. Given a signature Σ, for each Σ-frame (W, R) and each modal sentence $\rho \in \text{M-}Sen(\Sigma)$ we define the *satisfaction of ρ in (W, R) at the possible world $i \in I_W$*, denoted $(W, R) \models^i \rho$ by

- $(W, R) \models^i \rho$ if and only if $W^i \models \rho$ when $\rho \in Sen(\Sigma)$,
- $(W, R) \models^i \rho_1 \wedge \rho_2$ if and only if $(W, R) \models^i \rho_1$ and $(W, R) \models^i \rho_2$; similarly for other logical connectives,
- $(W, R) \models^i \Box\rho$ if and only if $(W, R) \models^j \rho$ for each $\langle i, j\rangle \in R$, and
- $(W, R) \models^i (\forall\chi)\rho$ if and only if $(W', R) \models^i \rho$ for each expansion (W', R) of (W, R) along χ.

For each Σ-frame (W, R) and each Σ-modal sentence ρ, then (W, R) *satisfies* ρ, denoted $(W, R) \models^m \rho$, if and only if $(W, R) \models^i \rho$ at each possible world $i \in I_W$.

Theorem 5. Assume that both the base institution and the domain institution are semi-exact [EB.+ED.], and that the mapping between their categories of signatures preserve pushouts [ET.].

Then for any modal sentence functor M-*Sen* over *Sen*, and for each $\text{K} \in \{T, S4, S5\}$, $(\mathbb{S}ign, \text{M-}Sen, K^{\Delta}\text{-Mod}, \models^m)$ is an institution.

Proposition 6. For each signature Σ and ρ and ρ' any Σ-sentences,

- $\models^m \Box\rho \rightarrow \rho$,
- $\models^m \Box(\rho \rightarrow \rho') \rightarrow (\Box\rho \rightarrow \Box\rho')$,
- $\rho \models^m \Box\rho$.

when $\text{K} \in \{T, S4, S5\}$,

- $\models^m \Box\rho \rightarrow \Box\Box\rho$

when $\text{K} \in \{S4, S5\}$,

- $\models^m \Diamond\rho \rightarrow \Box\Diamond\rho$

when $\text{K} = S5$.

For any signature morphism $\chi : \Sigma \rightarrow \Sigma'$ and each Σ'-sentence ρ,

- $(\forall\chi)\Box\rho \leftrightarrow \Box(\forall\chi)\rho$, and
- $(\exists\chi)\Box\rho \rightarrow \Box(\exists\chi)\rho$

when $\text{K} \in \{T, S4, S5\}$.

Notice however that due to the generality of quantification along signature morphisms, in the example of classical modal logic the last two rules hold also for second order quantification.

5 Reduced products of frames

Let $(\Phi^{\Delta}, \alpha^{\Delta}, \beta^{\Delta}) : (\mathbb{S}ign, Sen, \text{Mod}, \models) \rightarrow (\mathbb{S}ign^{\Delta}, Sen^{\Delta}, \text{Mod}^{\Delta}, \models^{\Delta})$ be an institution morphism from a base institution to a domain institution such that

EB. the base institution $(\mathbb{S}ign, Sen, \text{Mod}, \models)$ is semi-exact,
ED. the domain institution $(\mathbb{S}ign^{\Delta}, Sen^{\Delta}, \text{Mod}^{\Delta}, \models^{\Delta})$ is semi-exact,
ET. $\Phi^{\Delta} : \mathbb{S}ign \rightarrow \mathbb{S}ign^{\Delta}$ preserves pushouts, and
RP. for each signature Σ the category of Σ-models $\text{Mod}(\Sigma)$ has small products and directed colimits and β^{Δ}_{Σ} preserves them.
LI. for any signature Σ, β^{Δ}_{Σ} *lifts isomorphisms*, i.e. if $\beta^{\Delta}_{\Sigma}(M)$ is isomorphic to N' there exists N isomorphic to M such that $N' = \beta^{\Delta}_{\Sigma}(N)$.

In connection to Proposition 6 we have seen conditions [EB.], [ED.], and [ET.] as very easy, while the condition [LI.] is almost trivial, all examples of domain institution for first order logic discussed above satisfying it easily. Condition [RP.] is also very easy being very similar to the preservation of reduced products by model reducts corresponding to signature morphisms in an institution.

Proposition 7. For each $K \in \{T, S4, S5\}$ and for each signature Σ, the category of frames K^{Δ}-MOD(Σ) has reduced products .

Proof: Let us make the construction of the reduced products of frames explicit. For each $J \in F$ we denote the frame product $\prod_{j\in J}(W_j, R_j)$ by (W_J, R_J). Then

- $(I_{W_J}, R_J) = \prod_{j\in J}(I_{W_j}, R_j)$ in the category of first order models for a signature with only one sort and only one binary relation symbol; then each $k \in I_{W_J}$ can be represented as $(k_j)_{j\in J}$ with $k_j \in I_{W_j}$ for each $j \in J$, and we have that

$$\langle k, k'\rangle \in R_J \text{ if and only if } \langle k_j, k'_j\rangle \in R_j \text{ for each } j \in J$$

- for each $k = (k_j)_{j\in J} \in I_{W_J}$ we have $W_J^k = \prod_{j\in J} W_j^{k_j}$.

For each $J \subseteq J'$ where $J, J' \in F$, let $p_{J',J}$ denote the canonical projection $(W_{J'}, R_{J'}) \to (W_J, R_J)$. The reduced product (W_F, R_F) of $\{(W_i, R_i) \mid i \in I\}$ modulo F is the colimit of the directed diagram made of all these projections $p_{J',J}$.

$$\begin{array}{ccc} (W_{J'}, R_{J'}) & \xrightarrow{\quad p_{J',J} \quad} & (W_J, R_J) \\ & {\scriptstyle \mu_{J'}} \searrow \quad \swarrow {\scriptstyle \mu_J} & \\ & (W_F, R_F) & \end{array}$$

This colimit is constructed in two steps. We first do the reduced product (I_{W_F}, R_F) of the family of first order models $\{(I_{W_i}, R_i) \mid i \in I\}$ modulo F

$$\begin{array}{ccc} (I_{W_{J'}}, R_{J'}) & \xrightarrow{\quad p^I_{J',J} \quad} & (I_{W_J}, R_J) \\ & {\scriptstyle \mu^I_{J'}} \searrow \quad \swarrow {\scriptstyle \mu^I_J} & \\ & (I_{W_F}, R_F) & \end{array}$$

and then for each $i \in I_{W_F}$ we define W_F^i as the colimit of the (directed) diagram constituted of the canonical projections $p_{k',k} : W_{J'}^{k'} \to W_J^k$ for each $J \subseteq J'$ in F, and each $k \in {\mu^I_J}^{-1}(i)$ and $k' \in {\mu^I_{J'}}^{-1}(i)$ with $p^I_{J',J}(k') = k$

$$\begin{array}{ccc} W_{J'}^{k'} & \xrightarrow{\quad p_{k',k} \quad} & W_J^k \\ & {\scriptstyle \mu_{k'}} \searrow \quad \swarrow {\scriptstyle \mu_k} & \\ & W_F^i & \end{array}$$

Notice that for each $i, i' \in I_{W_F}$, $\langle i, i'\rangle \in R_F$ if and only if there exists $J \in F$ and there exists $k \in \mu_J^{-1}(i)$ and $k' \in \mu_J^{-1}(i')$ such that $\langle k, k'\rangle \in R_J$. Based on this remark, we can further notice that if R_i is reflexive, symmetric, or transitive for each $i \in I$, then R_F is reflexive, symmetric, respectively transitive.

Also, by conditions [RP.] and [LI.] we can see that W_F^i can be chosen such that $W_F^i = W_F^{i'}$ for each i and i' in I_{W_F}. ◁

We can also notice that for each $i \in I_{W_F}$, W_F^i can be presented as a reduced product.

Lemma 8. For each $i \in I_{W_F}$, and each $(k_j)_{j\in I} \in \mu_I^{-1}(i)$, W_F^i is the reduced product modulo F of the family $\{W_j^{k_j} \mid j \in I\}$.

Proof: For each $k \in \mu_I^{-1}(i)$ and each $J \in F$, let $k_J = p_{I,J}(k)$. Then the diagram constituted by the projections $p_{k_{J'},k_J}$ for all $J \subseteq J'$ in F is a final sub-diagram of the diagram defining W_F^i. ◁

6 Modal fundamental theorem

Let $(\mathbb{S}ign, \text{M-}Sen, K^{\Delta}\text{-}\textsc{Mod}, \models^m)$ be a modal institution over the institution morphism

$$(\Phi^{\Delta}, \alpha^{\Delta}, \beta^{\Delta}) : (\mathbb{S}ign, Sen, \textsc{Mod}, \models) \to (\mathbb{S}ign^{\Delta}, Sen^{\Delta}, \textsc{Mod}^{\Delta}, \models^{\Delta})$$

satisfying the conditions [EB.], [ED.], [ET.], [RP.], and [LI.].

Definition 9. Let $\mathcal{F}$ be a class of filters. For a signature Σ, a modal sentence ρ is

- *modally preserved by $\mathcal{F}$-reduced factors* when for each $i \in I_{W_F}$, $\prod_F(W_i, R_i) = (W_F, R_F) \models^i \rho$ implies "there exists $J \in F$ and $k \in \mu_J^{-1}(i)$ such that $(W_j, R_j) \models^{k_j} \rho$ for each $j \in J$", and
- *modally preserved by $\mathcal{F}$-reduced products* when for each $i \in I_{W_F}$, "there exists $J \in F$ and $k \in \mu_J^{-1}(i)$ such that $(W_j, R_j) \models^{k_j} \rho$ for each $j \in J$" implies $\prod_F(W_i, R_i) = (W_F, R_F) \models^i \rho$.

for each filter $F \in \mathcal{F}$ over a set I and for each family $\{(W_i, R_i)\}_{i \in I}$ of Σ-frames.

The following modal fundamental ultraproducts theorem represents a modal extension of the main result of [5].

Theorem 10. For any modal institution $(\mathbb{S}ign, \text{M-}Sen, K^{\Delta}\text{-}\textsc{Mod}, \models^m)$ over an institution morphism $(\Phi^{\Delta}, \alpha^{\Delta}, \beta^{\Delta}) : (\mathbb{S}ign, Sen, \textsc{Mod}, \models) \to (\mathbb{S}ign^{\Delta}, Sen^{\Delta}, \textsc{Mod}^{\Delta}, \models^{\Delta})$

1. Each sentence of the base institution which is preserved by $\mathcal{F}$-reduced products is also modally preserved by $\mathcal{F}$-reduced products of frames.
2. Each sentence of the base institution which is preserved by $\mathcal{F}$-reduced factors is also modally preserved by $\mathcal{F}$-reduced factors of frames.
3. The set of sentences modally preserved by $\mathcal{F}$-reduced products of frames is closed under possibility $\Diamond$.
4. The set of sentences modally preserved by $\mathcal{F}$-reduced factors of frames is closed under possibility $\Diamond$.
5. The set of sentences modally preserved by $\mathcal{F}$-reduced products of frames is closed under existential χ-quantification, when χ is conservative and preserves $\mathcal{F}$-reduced products in the base institution.
6. The set of sentences modally preserved by $\mathcal{F}$-reduced factors of frames is closed under existential χ-quantification, when χ lifts $\mathcal{F}$-reduced products of frames.
7. The set of sentences modally preserved by $\mathcal{F}$-reduced factors of frames and the set of sentences modally preserved by $\mathcal{F}$-reduced products of frames are both closed under (finite) conjunction.
8. The set of sentences modally preserved by $\mathcal{F}$-reduced products of frames is closed under infinite conjunctions.
9. If a sentence is modally preserved by $\mathcal{F}$-reduced factors of frames then its negation is modally preserved by $\mathcal{F}$-reduced products of frames.

And finally, if we further assume that $\mathcal{F}$ contains only ultrafilters,

10. If a sentence is modally preserved by $\mathcal{F}$-reduced products of frames then its negation is modally preserved by $\mathcal{F}$-reduced factors of frames.
11. The set of sentences modally preserved by both $\mathcal{F}$-reduced products and factors of frames is closed under negation.

Theorem 11. Each modal sentence obtained from the Łoś-sentences of the base institution by iterative application of logical connectives, necessity $\Box$, possibility $\Diamond$, and χ-quantifications for which χ is conservative, preserves reduced products of models (in the base institution), and lifts reduced products of frames

1. is modally preserved by ultraproducts and ultrafactors, and
2. is preserved by ultraproducts.

Note that (ordinary) preservation by ultrafactors is not a property to be in general expected for modal satisfaction, since, unlike in the case of ultraproducts, for ultrafactors the (ordinary) preservation cannot be established as a consequence of the modal preservation. This seems to be one of the important particularities of modal satisfaction.

Similarly to the corresponding result from [5], the only conditions that narrow the set of modal sentences which are preserved by ultraproducts refer to the quantifiers. Except lifting of reduced frames, the other conditions are at the level of the base institution and they have been previously analysed. Therefore, the key condition is the lifting of reduced frames. However the result below reduces it to lifting of models (in the base institution).

Definition 12. A signature morphism $\chi : \Sigma \to \Sigma'$ is Φ^{Δ}-*exact* when the square of the naturality of β^{Δ} for χ is pullback:

$$\begin{array}{ccc} \Sigma & & \mathrm{Mod}(\Sigma) \xleftarrow{\beta^{\Delta}_{\Sigma}} \mathrm{Mod}'(\Sigma\Phi^{\Delta}) \\ {\scriptstyle\chi}\downarrow & & \uparrow{\scriptstyle \mathrm{Mod}(\chi)} \qquad\qquad \uparrow{\scriptstyle \mathrm{Mod}(\chi\Phi^{\Delta})} \\ \Sigma' & & \mathrm{Mod}(\Sigma') \xleftarrow[\beta^{\Delta}_{\Sigma'}]{} \mathrm{Mod}'(\Sigma'\Phi^{\Delta}) \end{array}$$

Proposition 13. Let $\chi : \Sigma \to \Sigma'$ be a signature morphism in the base institution. If χ is Φ^{Δ}-exact and lifts and preserves reduced products of models, then it lifts reduced products of frames.

By Proposition 13 we get the following sufficient condition for lifting of reduced products of frames.

Corollary 14. A signature morphism lifts reduced products of frames when it is finitary representable conservative, Φ^{Δ}-exact, and preserves reduced products of models (in the base institution).

Example 15. We may recall that for the institution first order logic of first order logic, each injective signature morphism adding only a finite number of constants as new symbols is finitary representable. Moreover, such signature morphism is also Φ^{Δ}-exact when we take the sub-institution with signatures are sets of sort symbols plus constant declarations as the domain institution. This shows how the preservation of modal sentences by ultraproducts in conventional modal first order logic becomes a special case of Theorem 11.

Let us derive a compactness property for modal institutions as application of our modal preservation result.

Definition 16. A set of sentences E for a signature Σ is *consistent* if E^* is not empty.

An institution is *model compact* if each set of sentences is consistent when all its finite subsets are consistent.

If for each set of sentences E and each sentence e, $E \models e$ implies the existence of a finite subset $E_f \subseteq E$ such that $E_f \models e$, then we say that the institution is *compact*.

Remark 17. Each model compact institution having negation is compact and each compact institution having **false**[1] is model compact.

Recall from [5] the the following result:

Proposition 18. Any institution in which each sentence is preserved by ultraproducts is model compact.

When we apply this to modal institutions we obtain:

Corollary 19. If the base institution is a Łoś-institution and each modal sentence is obtained from the sentences of the base institution by iterative application of logical connectives, necessity $\Box$, possibility $\Diamond$, and χ-quantifications for which χ is finitary representable, conservative, Φ^{Δ}-exact, and preserves reduced products of models (in the base institution), then the modal institution is model compact.

Example 20. A typical concrete instance of Corollary 19 is conventional modal first order logic, which is therefore model compact. By Remark 17 modal first order logic is compact too.

[1] A sentence which is not satisfied by any model.

7 Conclusions and Future Research

We have defined abstract 'modal' institutions which has frames defined from the models of the base institution, modal sentences as extensions of the sentences of the base institution with the usual modal operators as sentences, and a modal satisfaction between frames and sentences. We can extent our work to the multi - modal case. Other logical systems such as epistemic logic, action logic, dynamic logic or deontic logic which use Kripke models may benefit from our approach. Grothendiek institutions may be used to combine logics with abstract Kripke semantics. Applications in formal ethics can be found.

Bibliography

[1] R. Diaconescu. Extra theory morphisms for institutions: logical semantics for multi-paradigm languages. *Applied Categorical Structures*, 6(4):427–453, 1998.

[2] R. Diaconescu. Elementary diagrams in institutions. Technical Report 7-2002, Institute of Mathematics of the Romanian Academy, 2002. Submitted to publication.

[3] R. Diaconescu. Grothendieck institutions. *Applied Categorical Structures*, 10(4):383–402, 2002.

[4] R. Diaconescu. An institution-independent proof of Craig interpolation property. Technical Report 8-2002, Institute of Mathematics of the Romanian Academy, 2002. Submitted to publication.

[5] R. Diaconescu. Institution-independent ultraproducts. Technical Report 5-2002, Institute of Mathematics of the Romanian Academy, 2002. Submitted to publication.

[6] R. Diaconescu and K. Futatsugi. Logical foundations of CafeOBJ. *Theoretical Computer Science*, 285:289–318, 2002.

[7] R. Diaconescu, J. Goguen, and P. Stefaneas. Logical support for modularisation. In G. Huet and G. Plotkin, editors, *Logical Environments*, pages 83–130, 1993. Proceedings of a Workshop held in Edinburgh, Scotland, May 1991.

[8] J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.

[9] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24:1–15, 1959.

[10] S. MacLane. *Categories for the Working Mathematician*. Springer-Verlag, 1998. Second edition.

[11] J. Meseguer. A logical theory of concurrent objects and its realization in the Maude language. In G. Agha, P. Wegner, and A. Yonezawa, editors, *Research Directions in Concurrent Object-Oriented Programming*. The MIT Press, 1993.

[12] T. Mossakowski. Relating CASL with other specification languages: the institution level. *Theoretical Computer Science*, 286:367–475, 2002.

[13] A. Tarlecki. Quasi-varieties in abstract algebraic institutions. *Journal of Computer and System Sciences*, 33(3):333–360, 1986. Original version, University of Edinburgh, Report CSR-173-84.

[14] A. Tarlecki. Towards heterogeneous specifications. In D. Gabbay and M. van Rijke, editors, *Frontiers of Combining Systems (FroCoS'98)*, pages 337–360. 2000.

Analysis of Two Fragments of the Logic of Residuated Lattices

Félix Bou[1] Àngel García-Cerdaña[2] Ventura Verdú[1]

[1] Department of Logic, History and Philosophy of Science, University of Barcelona, Spain
{bou,verdu}@mat.ub.es

[2] Institute of Investigation in Artificial Intelligence, CSIC, Bellaterra, Spain
angel@iiia.csic.es

Abstract

The logic of (commutative integral bounded) residuated lattices is known under different names in the literature: monoidal logic [9], intuitionistic logic without contraction [1], H_{BCK} [13], etc.
This paper contains a summary of the results obtained in [6] about the $\langle \vee, *, \neg, 0, 1\rangle$-fragment and the $\langle \vee, \wedge, *, \neg, 0, 1\rangle$-fragment of the logic of residuated lattices.
. As regards the algebraic aspects of this study, we define two new varieties: the variety of commutative integral bounded semilatticed pseudocomplemented monoids, denoted by $\mathbb{CIBPM}^{s\ell}$, and the variety of commutative integral bounded latticed pseudocomplemented monoids, denoted by $\mathbb{CIBPM}^{\ell}$. We show that every $\mathbb{CIBPM}^{s\ell}$-algebra and every $\mathbb{CIBPM}^{\ell}$-algebra is embeddable into a residuated lattice.
As regards the logical aspects of this study, we introduce two sequent calculi: $\mathbf{FL_{ew}}[\vee, *, \neg]$ and $\mathbf{FL_{ew}}[\vee, \wedge, *, \neg]$. It can be shown that $\mathbb{CIBPM}^{s\ell}$ ($\mathbb{CIBPM}^{\ell}$) is the equivalent variety semantics [14, 8] of the intuitionistic Gentzen system associated to the sequent calculi $\mathbf{FL_{ew}}[\vee, *, \neg]$ ($\mathbf{FL_{ew}}[\vee, \wedge, *, \neg]$). Moreover we show a generalization of [11, Corollary 9]: the external deductive system $\mathcal{S}_e[\vee, *, \neg]$ ($\mathcal{S}_e[\vee, \wedge, *, \neg]$) associated to $\mathbf{FL_{ew}}[\vee, *, \neg]$ ($\mathbf{FL_{ew}}[\vee, \wedge, *, \neg]$) is the $\langle \vee, *, \neg, 0, 1\rangle$-fragment ($\langle \vee, \wedge, *, \neg, 0, 1\rangle$-fragment) of the logic of residuated lattices. We also show that $\mathcal{S}_e[\vee, *, \neg]$ and $\mathcal{S}_e[\vee, \wedge, *, \neg]$ are decidable.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Preliminary concepts

In this section we recall some concepts and results about Gentzen systems and the algebraization of these systems which will be used in this paper (for more information see [14], [15], [8]).

1.1 Gentzen systems

Let $\mathcal{L}$ be a propositional language. We will denote by $Fm_{\mathcal{L}}$ the set of $\mathcal{L}$-formulas and by $\mathcal{F}m_{\mathcal{L}}$ the algebra of $\mathcal{L}$-formulas. Let α and β be some subsets of the set ω of natural numbers. A *$\mathcal{L}$-sequent of type* (α,β) is a pair (Γ,Δ) of finite sequences of $\mathcal{L}$-formulas such that the length of Γ belongs to α and the length of Δ belongs to β, that is, $(\Gamma,\Delta)\in Fm_{\mathcal{L}}^m\times Fm_{\mathcal{L}}^n$ for some $m\in\alpha$ and $n\in\beta$. We will write $\Gamma\Rightarrow\Delta$ instead of (Γ,Δ). We will denote by $Seq_{\mathcal{L}}^{(\alpha,\beta)}$ the set of $\mathcal{L}$-sequents of type (α,β).

A consequence relation $\vdash_{\mathcal{G}}$ on the set $Seq_{\mathcal{L}}^{(\alpha,\beta)}$ is said to be *invariant under substitutions* if, for every $h\in Hom(Fm_{\mathcal{L}},Fm_{\mathcal{L}})$,

$$\{\Gamma_i\Rightarrow\Delta_i : i\in I\}\vdash_{\mathcal{G}}\Gamma\Rightarrow\Delta \text{ implies } \{h(\Gamma_i\Rightarrow\Delta_i) : i\in I\}\vdash_{\mathcal{G}} h(\Gamma\Rightarrow\Delta),$$

where $h(\varphi_0,\ldots,\varphi_{m-1}\Rightarrow\psi_0,\ldots,\psi_{n-1})$ stands for $h\varphi_0,\ldots,h\varphi_{m-1}\Rightarrow h\psi_0,\ldots,h\psi_{n-1}$.

A *Gentzen system of type* (α,β) is a pair $\mathcal{G}=\langle\mathcal{L},\vdash_{\mathcal{G}}\rangle$ where $\vdash_{\mathcal{G}}$ is a relation which is finitary and invariant under substitutions on the set $Seq_{\mathcal{L}}^{(\alpha,\beta)}$. If $T\cup\{\Gamma\Rightarrow\Delta,\Pi\Rightarrow\Lambda\}\subseteq Seq_{\mathcal{L}}^{(\alpha,\beta)}$, we will write $T,\Gamma\Rightarrow\Delta\vdash_{\mathcal{G}}\Pi\Rightarrow\Lambda$ for $T\cup\{\Gamma\Rightarrow\Delta\}\vdash_{\mathcal{G}}\Pi\Rightarrow\Lambda$.

We will say that two sequents $\Gamma\Rightarrow\Delta$ and $\Pi\Rightarrow\Lambda$ are *$\mathcal{G}$-equivalent* and we will write $\Gamma\Rightarrow\Delta\dashv\vdash_{\mathcal{G}}\Pi\Rightarrow\Lambda$ if it holds that $\Gamma\Rightarrow\Delta\vdash_{\mathcal{G}}\Pi\Rightarrow\Lambda$ and $\Pi\Rightarrow\Lambda\vdash_{\mathcal{G}}\Gamma\Rightarrow\Delta$. A sequent $\Gamma\Rightarrow\Delta\in Seq_{\mathcal{L}}^{(\alpha,\beta)}$ is *$\mathcal{G}$-derivable* if $\emptyset\vdash_{\mathcal{G}}\Gamma\Rightarrow\Delta$.

Clearly, a deductive system $\mathcal{S}$ can be seen as a Gentzen system $\mathcal{G}_{\mathcal{S}}$ of type $(\{0\},\{1\})$, if we identify a formula φ with the sequent $\emptyset\Rightarrow\varphi$. Furthermore, if K is a class of algebras such that the equational consequence relation determined by K is finitary, then $S_K=\langle\mathcal{L},\models_K\rangle$ will be considered as a Gentzen system $\mathcal{G}_K$ of type $(\{1\},\{1\})$, if we identify an equation $\varphi\approx\psi$ with the sequent $\varphi\Rightarrow\psi$.

1.2 Fragments

Let $\mathcal{G}$ be a Gentzen system of type (α,β) and let $\mathcal{L}'$ be a sublanguage of $\mathcal{L}$. The *$\mathcal{L}'$-fragment* of $\mathcal{G}$ is the Gentzen system $\mathcal{G}'=\langle\mathcal{L}',\vdash_{\mathcal{G}'}\rangle$ of type (α,β) defined by:

$$T\vdash_{\mathcal{G}'}\Gamma\Rightarrow\Delta \text{ iff } T\vdash_{\mathcal{G}}\Gamma\Rightarrow\Delta, \text{ for all } T\cup\{\Gamma\Rightarrow\Delta\}\subseteq Seq_{\mathcal{L}'}^{(\alpha,\beta)}.$$

Let $\alpha'\subseteq\alpha$ and $\beta'\subseteq\beta$. The *(α',β')-fragment* de $\mathcal{G}$ is the Gentzen system $\mathcal{G}'=\langle\mathcal{L},\vdash_{\mathcal{G}'}\rangle$ of type (α',β') defined by:

$$T\vdash_{\mathcal{G}'}\Gamma\Rightarrow\Delta \text{ iff } T\vdash_{\mathcal{G}}\Gamma\Rightarrow\Delta, \text{ for all } T\cup\{\Gamma\Rightarrow\Delta\}\subseteq Seq_{\mathcal{L}}^{(\alpha',\beta')}.$$

1.3 Sequent calculus

An *$(\mathcal{L},(\alpha,\beta))$-sequent calculus* is a set of $(\mathcal{L},(\alpha,\beta))$-rules. Every $(\mathcal{L},(\alpha,\beta))$-sequent calculus LX determines a Gentzen system $\mathcal{G}_{LX}=\langle\mathcal{L},\vdash_{LX}\rangle$ of type (α,β) in the following way:

Let $T\cup\{\Gamma\Rightarrow\Delta\}\subseteq Seq_{\mathcal{L}}^{(\alpha,\beta)}$. We will say that $\Gamma\Rightarrow\Delta$ is *deduced* from T in the Gentzen system $\mathcal{G}_{LX}$ and we will write $T\vdash_{LX}\Gamma\Rightarrow\Delta$ iff there is a finite sequence of $\mathcal{L}$-sequents $\Gamma_0\Rightarrow\Delta_0,\ldots,\Gamma_{n-1}\Rightarrow\Delta_{n-1}$ (which is called a *proof* of $\mathbf{\Gamma}\Rightarrow\Delta$ from T) such that $\Gamma_{n-1}\Rightarrow\Delta_{n-1}=\Gamma\Rightarrow\Delta$ and for each $i<n$ one of the following conditions holds:

(i) $\Gamma_i\Rightarrow\Delta_i$ is an instance of an axiom of LX;

(ii) $\Gamma_i\Rightarrow\Delta_i\in T$;

(iii) $\Gamma_i\Rightarrow\Delta_i$ is obtained from $\{\Gamma_j\Rightarrow\Delta_j : j<i\}$ by using a rule r of LX, i. e., $\dfrac{T}{\Gamma_i\Rightarrow\Delta_i}\in r$ for some $T\subseteq\{\Gamma_j\Rightarrow\Delta_j : j<i\}$.

In this case we will say that $\mathcal{G}_{LX}$ is the *Gentzen system determined by the sequent calculus LX*. Note that we use the rules of the calculus to obtain sequents from sets of sequents (and so not only from the empty set).

1.4 Equivalence of Gentzen systems

From now on let $\mathcal{G}_1$ and $\mathcal{G}_2$ be Gentzen systems over the same language $\mathcal{L}$ of type (α_1, β_1) and (α_2, β_2) respectively. A $((\alpha_1, \beta_1), (\alpha_2, \beta_2))$-*translation* is a set

$$\tau = \left\{\tau_{(m,n)} : m \in \alpha_1, n \in \beta_1\right\},$$

where every $\tau_{(m,n)}$ is a finite set of sequents $Seq_{\mathcal{L}}^{(\alpha_2,\beta_2)}$ in $m+n$ variables $p_0, \ldots, p_{m-1}$, $q_0, \ldots, q_{n-1}$ (at most) if $(m,n) \neq (0,0)$, and in one variable p_0 (at most) if $(m,n) = (0,0)$.

Given $\Gamma \Rightarrow \Delta = (\varphi_0, \ldots, \varphi_{m-1}) \Rightarrow (\psi_0, \ldots, \psi_{n-1}) \in Seq_{\mathcal{L}}^{(\alpha_1,\beta_1)}$, we will write $\tau(\Gamma \Rightarrow \Delta) = \tau_{(m,n)}(\Gamma \Rightarrow \Delta)$, the result of replacing the variable p_i by φ_i, $i < m$, and the variable q_j by ψ_j, $j < n$, in every sequent of $\tau_{(m,n)}$. If $\emptyset \Rightarrow \emptyset \in Seq_{\mathcal{L}}^{(\alpha_1,\beta_1)}$, we will write $\tau(\emptyset \Rightarrow \emptyset) = \tau_{(0,0)}(p_0)$. If $T \subseteq Seq_{\mathcal{L}}^{(\alpha_1,\beta_1)}$, $\tau(T)$ will denote the set $\cup\{\tau(\Gamma \Rightarrow \Delta) : \Gamma \Rightarrow \Delta \in T\}$.

We say that $\mathcal{G}_1$ and $\mathcal{G}_2$ are *equivalent* if there is a translation τ of the set of sequents $Seq_{\mathcal{L}}^{(\alpha_1,\beta_1)}$ in the set of sequents $Seq_{\mathcal{L}}^{(\alpha_2,\beta_2)}$ and a translation ρ of the set of sequents $Seq_{\mathcal{L}}^{(\alpha_2,\beta_2)}$ in the set of sequents $Seq_{\mathcal{L}}^{(\alpha_1,\beta_1)}$, such that

(i) $T \vdash_{\mathcal{G}_1} \Gamma \Rightarrow \Delta$ iff $\tau(T) \vdash_{\mathcal{G}_2} \tau(\Gamma \Rightarrow \Delta)$ for all $T \cup \{\Gamma \Rightarrow \Delta\} \subseteq Seq_{\mathcal{L}}^{(\alpha_1,\beta_1)}$.

(ii) $T \vdash_{\mathcal{G}_2} \Gamma \Rightarrow \Delta$ iff $\rho(T) \vdash_{\mathcal{G}_1} \rho(\Gamma \Rightarrow \Delta)$ for all $T \cup \{\Gamma \Rightarrow \Delta\} \subseteq Seq_{\mathcal{L}}^{(\alpha_2,\beta_2)}$.

(iii) $\Gamma \Rightarrow \Delta \dashv\vdash_{\mathcal{G}_2} \tau\rho(\Gamma \Rightarrow \Delta)$ for all $\Gamma \Rightarrow \Delta \in Seq_{\mathcal{L}}^{(\alpha_2,\beta_2)}$.

(iv) $\Gamma \Rightarrow \Delta \dashv\vdash_{\mathcal{G}_1} \rho\tau(\Gamma \Rightarrow \Delta)$ for all $\Gamma \Rightarrow \Delta \in Seq_{\mathcal{L}}^{(\alpha_1,\beta_1)}$.

In fact this definition is redundant because (i) and (iii) are equivalent to (ii) and (iv) (see [15, Proposition 2.1]).

1.5 Algebraization of Gentzen systems

Let K be a class of algebras. We will denote by $\mathcal{S}_K$ the equational system associated with K. Gentzen system $\mathcal{G}$ is said to be *algebraizable with equivalent algebraic semantics* K if $\mathcal{G}$ and $\mathcal{S}_K$ are equivalent as Gentzen systems.

If K is an equivalent algebraic semantics for a Gentzen system $\mathcal{G}$, then so is the quasivariety K^Q generated by K [15, Corollary 2.2]. Moreover, if K are K' are equivalent algebraic semantics for $\mathcal{G}$, then K and K' generates the same quasivariety [15, Corollary 2.4]. This quasivariety is called *the equivalent quasivariety semantics* for $\mathcal{G}$.

2 The calculi $\mathbf{FL_{ewc}}$, $\mathbf{FL_{ew}}$, $\mathbf{FL_{ew}}[\vee, *, \neg]$, $\mathbf{FL_{ew}}[\vee, \wedge, *, \neg]$

In this section we recall the definition of the sequent calculi (cut free) $\mathbf{FL_{ewc}}$ and $\mathbf{FL_{ew}}$ (Cf. [10], [12]) in the language $\mathcal{L} = \{\vee, \wedge, *, \rightarrow, \neg, 0, 1\}$ and we define the calculi $\mathbf{FL_{ew}}[\vee, *, \neg]$ and $\mathbf{FL_{ew}}[\vee, \wedge, *, \neg]$ obtained by restriction of the calculus $\mathbf{FL_{ew}}$ to the languages $\{\vee, *, \neg, 0, 1\}$ and $\{\vee, \wedge, *, \neg, 0, 1\}$ respectively.

Definition 1. Let $\mathcal{L} = \{\vee, \wedge, *, \rightarrow, \neg, 0, 1\}$ be a propositional language of type (2, 2, 2, 2, 1, 0, 0). Let φ, ψ be $\mathcal{L}$-formulas; Γ, Π, Σ finite sequences (possibly empty) of $\mathcal{L}$-formulas and Δ a sequence of at most one formula. $\mathbf{FL_{ewc}}$ is the calculus of $\mathcal{L}$-sequents of type $(\omega, \{0,1\})$ defined by the following axioms and rules:

Axioms::

$$\varphi \Rightarrow \varphi \quad \text{(Axiom 1)} \qquad 0 \Rightarrow \emptyset \quad \text{(Axiom 2)} \qquad \emptyset \Rightarrow 1 \quad \text{(Axiom 3)}$$

Structural rules::

Cut:

$$\frac{\Gamma \Rightarrow \varphi \qquad \Sigma, \varphi, \Pi \Rightarrow \Delta}{\Sigma, \Gamma, \Pi \Rightarrow \Delta} \ (\text{Cut})$$

Exchange:

$$\frac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta} \ (e \Rightarrow)$$

Weakening:

$$\frac{\Sigma, \Gamma \Rightarrow \Delta}{\Sigma, \varphi, \Gamma \Rightarrow \Delta} \ (w \Rightarrow) \qquad \frac{\Gamma \Rightarrow \emptyset}{\Gamma \Rightarrow \varphi} \ (\Rightarrow w)$$

Contraction:

$$\frac{\Sigma, \varphi, \varphi, \Gamma \Rightarrow \Delta}{\Sigma, \varphi, \Gamma \Rightarrow \Delta} \ (c \Rightarrow)$$

Rules of introduction of connectives:

$$\frac{\Sigma, \varphi, \Gamma \Rightarrow \Delta \qquad \Sigma, \psi, \Gamma \Rightarrow \Delta}{\Sigma, \varphi \vee \psi, \Gamma \Rightarrow \Delta} \ (\vee \Rightarrow) \qquad \frac{\Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \varphi \vee \psi} \ (\Rightarrow \vee_1) \qquad \frac{\Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \vee \psi} \ (\Rightarrow \vee_2)$$

$$\frac{\Sigma, \varphi, \Gamma \Rightarrow \Delta}{\Sigma, \varphi \wedge \psi, \Gamma \Rightarrow \Delta} \ (\wedge_1 \Rightarrow) \qquad \frac{\Sigma, \psi, \Gamma \Rightarrow \Delta}{\Sigma, \varphi \wedge \psi, \Gamma \Rightarrow \Delta} \ (\wedge_2 \Rightarrow) \qquad \frac{\Gamma \Rightarrow \varphi \qquad \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \wedge \psi} \ (\Rightarrow \wedge)$$

$$\frac{\Sigma, \varphi, \psi, \Gamma \Rightarrow \Delta}{\Sigma, \varphi * \psi, \Gamma \Rightarrow \Delta} \ (* \Rightarrow) \qquad \frac{\Gamma \Rightarrow \varphi \qquad \Pi \Rightarrow \psi}{\Gamma, \Pi \Rightarrow \varphi * \psi} \ (\Rightarrow *)$$

$$\frac{\Gamma \Rightarrow \varphi \qquad \Sigma, \psi, \Pi \Rightarrow \Delta}{\Sigma, \varphi \to \psi, \Gamma, \Pi \Rightarrow \Delta} \ (\to \Rightarrow) \qquad \frac{\varphi, \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \to \psi} \ (\Rightarrow \to)$$

$$\frac{\Gamma \Rightarrow \varphi}{\neg\varphi, \Gamma \Rightarrow \emptyset} \ (\neg \Rightarrow) \qquad \frac{\varphi, \Gamma \Rightarrow \emptyset}{\Gamma \Rightarrow \neg\varphi} \ (\Rightarrow \neg)$$

Definition 2. $\mathbf{FL_{ew}}$ is the calculus of $\mathcal{L}$-sequents of type $(\omega, \{0, 1\})$ obtained from $\mathbf{FL_{ewc}}$ by deleting the rule *$(c \Rightarrow)$*.

Theorem 3 ([10, Theorem 6]). Cut elimination holds for $\mathbf{FL_{ewc}}$ and $\mathbf{FL_{ew}}$.

Definition 4. The calculus obtained by deleting from $\mathbf{FL_{ew}}$ the rules of introduction of the additive conjunction and the implication will be denoted by $\mathbf{FL_{ew}}[\vee, *, \neg]$[1], and the calculus obtained by deleting from $\mathbf{FL_{ew}}$ the rules for the implication will be denoted by $\mathbf{FL_{ew}}[\vee, \wedge, *, \neg]$. The Gentzen systems associated to the calculi $\mathbf{FL_{ew}}[\vee, *, \neg]$ and $\mathbf{FL_{ew}}[\vee, \wedge, *, \neg]$ will be denoted by $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ and $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$, respectively.

Theorem 5. $\mathbf{FL_{ew}}[\vee, *, \neg]$ and $\mathbf{FL_{ew}}[\vee, \wedge, *, \neg]$ satisfy the cut elimination theorem.

3 Equivalence between $\mathcal{G}_{\mathbf{FL_{ew}}}$, the deductive system $\widehat{IPC}^*\backslash c$ and the variety of residuated lattices

In this section we will define the deductive system $\widehat{IPC}^*\backslash c$. It is easy to see that this system is definitionally equivalent to $IPC^*\backslash c$ [1], H_{BCK} [13] and ML [9]. We will also state the algebraization of the Gentzen system $\mathcal{G}_{\mathbf{FL_{ew}}}$.

Definition 6. $\widehat{IPC}^*\backslash c$ is the deductive system in the language $\mathcal{L} = \{\vee, \wedge, *, \to, \neg, 0, 1\}$ of type (2, 2, 2, 2, 1, 0, 0), defined by the Modus Ponens rule and the following axioms:

[1]It would be more accurate to talk about $\mathbf{FL_{ew}}[\vee, *, \neg, 0, 1]$ because this is the language where this calculus is given, but for the sake of simplicity we will not do so.

(A1) $\varphi \to 1$

(A2) $(\varphi \to \psi) \to ((\gamma \to \varphi) \to (\gamma \to \psi))$ (B)

(A3) $(\varphi \to (\psi \to \gamma)) \to (\psi \to (\varphi \to \gamma))$ (C)

(A4) $\varphi \to (\psi \to \varphi)$ (K)

(A5) $(\varphi \wedge \psi) \to \varphi$

(A6) $(\varphi \wedge \psi) \to \psi$

(A7) $\varphi \to (\psi \to (\varphi \wedge \psi))$

(A8) $((\gamma \to \varphi) \wedge (\gamma \to \psi)) \to (\gamma \to (\varphi \wedge \psi))$

(A9) $\psi \to (\varphi \vee \psi)$

(A10) $\varphi \to (\varphi \vee \psi)$

(A11) $(\varphi \to \gamma) \to ((\psi \to \gamma) \to ((\varphi \vee \psi) \to \gamma))$

(A12) $\varphi \to (\psi \to (\varphi * \psi))$

(A13) $(\varphi \to (\psi \to \gamma)) \to ((\varphi * \psi) \to \gamma)$

(A14) $0 \to \varphi$

(A15) $\neg\varphi \to (\varphi \to \psi)$

(A16) $(\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$

(A17) $\varphi \to \neg\neg\varphi$

Let us recall the definition of residuated lattice:

Definition 7. The class $\mathbb{RL}$ of *commutative integral bounded residuated lattice*, or *residuated lattice* for short, is the class of algebras $\mathcal{A} = \langle A, \vee, \wedge, *, \to, \neg, 0, 1\rangle$ of type $(2, 2, 2, 2, 1, 0, 0)$ satisfying the following conditions:

1. $\langle A, \vee, \wedge, 0, 1\rangle$ is a bounded lattice with minimum element 0 and maximum element 1,
2. $\langle A, *, 1\rangle$ is a commutative monoid,
3. $\forall\ a, b \in A$, $a \to b = \max\{c \in A : a * c \leq b\}$ *(*i.e. $x * z \leq y \Leftrightarrow z \leq x \to y$*)*
4. $\forall a \in A$, $\neg a = \neg a \to 0$.

It is well known that $\mathbb{RL}$ is a variety. It was proved in [1, Theorem 22] that $\mathcal{G}_{LJ^*\backslash c}$ and $IPC^*\backslash c$ are equivalent and it was also proved in [1, Theorem 22] that the Gentzen system $\mathcal{G}_{LJ^*\backslash c}$ is algebraizable with equivalent algebraic semantics the variety of residuated lattices. As $\mathcal{G}_{\mathbf{FL_{ew}}}$ and $\widehat{IPC}^*\backslash c$ are essentially equivalent to $\mathcal{G}_{LJ^*\backslash c}$ and $IPC^*\backslash c$, respectively, we have the following results:

Theorem 8. $\mathcal{G}_{\mathbf{FL_{ew}}}$ and $\widehat{IPC}^*\backslash c$ are equivalent, with the translations τ from $\mathcal{G}_{\mathbf{FL_{ew}}}$ to $\widehat{IPC}^*\backslash c$ and ρ from $\widehat{IPC}^*\backslash c$ to $\mathcal{G}_{\mathbf{FL_{ew}}}$ defined as follows:

$$\tau_{(m,1)}(p_0, ..., p_{m-1} \Rightarrow q_0) = \begin{cases} p_0 \to (p_1 \to (... \to (p_{m-1} \to q_0)...)), & \text{if } m \geq 1 \\ q_0, & \text{if } m = 0 \end{cases}$$

$$\tau_{(m,0)}(p_0, ..., p_{m-1} \Rightarrow \emptyset) = \begin{cases} p_0 \to (p_1 \to (... \to (p_{m-1} \to 0)...)), & \text{if } m \geq 1 \\ 0, & \text{if } m = 0 \end{cases}$$

$\rho(p_0) = \{\emptyset \Rightarrow p_0\}$.

That is, the following conditions are satisfied:

(*i*) For every $\Sigma \cup \{\varphi\} \subseteq Fm_{\mathcal{L}}$, $\Sigma \vdash_{\widehat{IPC}^*\setminus c} \varphi$ iff $\{\rho(\sigma) : \sigma \in \Sigma\} \vdash_{\mathbf{FL_{ew}}} \rho(\varphi)$.

(*ii*) For every $\Gamma \Rightarrow \Delta \in Seq_{\mathcal{L}}^{(\omega,\{0,1\})}$, $\Gamma \Rightarrow \Delta \dashv\vdash_{\mathbf{FL_{ew}}} \rho\tau(\Gamma \Rightarrow \Delta)$.

Theorem 9. $\mathcal{G}_{\mathbf{FL_{ew}}}$ is algebraizable with equivalent variety semantics the variety $\mathbb{RL}$ with translations τ from $\mathcal{G}_{\mathbf{FL_{ew}}}$ to $\mathcal{S}_{\mathbb{RL}}$ (the equational system associated to $\mathbb{RL}$) and ρ from $\mathcal{S}_{\mathbb{RL}}$ to $\mathcal{G}_{\mathbf{FL_{ew}}}$ defined as follows:

$$\tau_{(m,1)}(p_0,...,p_{m-1} \Rightarrow q_0) = \begin{cases} p_0 \to (p_1 \to (... \to (p_{m-1} \to q_0)...)) \approx 1, & \text{if } m \geq 1 \\ q_0 \approx 1, & \text{if } m = 0 \end{cases}$$

$$\tau_{(m,0)}(p_0,...,p_{m-1} \Rightarrow \emptyset) = \begin{cases} p_0 \to (p_1 \to (... \to (p_{m-1} \to 0)...)) \approx 1, & \text{if } m \geq 1 \\ 0 \approx 1, & \text{if } m = 0 \end{cases}$$

$\rho(p_0 \approx p_1) = \{p_0 \Rightarrow p_1; p_1 \Rightarrow p_0\}$

Remark 1. Observe that since in $\mathbb{RL}$ the following holds

- $(x_0 * x_1 * \ldots * x_n) \to y \approx x_0 \to (x_1 \to (\ldots \to (x_n \to y)\ldots))$
- $x \vee y \approx y$ iff $x \leq y$ iff $x \to y \approx 1$,

then in Theorem 9 we could take the following translations:

$\tau_{(m,1)}(p_0,...,p_{m-1} \Rightarrow q_0) = \{((p_0 * p_1 * ... * p_{m-1}) \vee q_0) \approx q_0\}$
$\tau_{(m,0)}(p_0,...,p_{m-1} \Rightarrow \emptyset) = \{p_0 * p_1 * ... * p_{m-1} \approx 0\}$.

4 The varieties $\mathbb{CIBPM}^{s\ell}$ and $\mathbb{CIBPM}^{\ell}$

In this section we study the class of commutative integral bounded semilatticed pseudocomplemented monoids ($\mathbb{CIBPM}^{s\ell}$ for short) and the class of commutative integral bounded semilatticed pseudocomplemented monoids ($\mathbb{CIBPM}^{\ell}$ for short). We introduce the notion of pseudocomplement applied to the operation $*$ of the commutative integral bounded semilatticed monoids. This notion is a generalization of the concept of pseudocomplement defined traditionally in the context of the bounded distributive lattices (see for instance [3]).

Definition 10. A *commutative semilatticed monoid* (*commutative $s\ell$-monoid* for short) is an algebra $\mathcal{A} = \langle A, \vee, *, 1\rangle$ of type $(2, 2, 0)$ such that:

1. $\langle A, \vee\rangle$ is a semilattice,
2. $\langle A, *, 1\rangle$ is a commutative monoid,
3. $\mathcal{A} \vDash (x \vee y) * z \approx (x * z) \vee (y * z)$.

A *commutative latticed monoid* (*commutative ℓ-monoid* for short) is an algebra $\mathcal{A} = \langle A, \vee, \wedge, *, 1\rangle$ of type $(2, 2, 2, 0)$ such that $\langle A, \vee, \wedge\rangle$ is a lattice and such that $\langle A, \vee, *, 1\rangle$ is a commutative $s\ell$-monoid.

Definition 11. An algebra $\mathcal{A} = \langle A, \vee, *, 0, 1\rangle$ of type $(2, 2, 0, 0)$ is a *commutative integral bounded $s\ell$-monoid* if the following conditions are satisfied:

1. $\langle A, \vee, *, 1\rangle$ is a commutative $s\ell$-monoid,
2. $\mathcal{A} \vDash 0 \vee x \approx x$ (i.e. $0 \leq x$),
3. $\mathcal{A} \vDash x \vee 1 \approx 1$ (i.e. $x \leq 1$).

An algebra $\mathcal{A} = \langle A, \vee, \wedge, *, 0, 1\rangle$ of type $(2, 2, 2, 0, 0)$ is a *commutative integral bounded ℓ-monoid* if $\langle A, \vee, \wedge\rangle$ is a lattice and $\langle A, \vee, *, 0, 1\rangle$ is a commutative integral bounded $s\ell$-monoid.

We will denote by $\mathbb{CIBM}^{s\ell}$ the class of commutative integral bounded $s\ell$-monoids and by $\mathbb{CIBM}^{\ell}$ the class of commutative integral bounded ℓ-monoids. Obviously these classes of algebras are varieties. The variety of bounded distributive lattices, $\mathbb{BDL}$ for short, is the subvariety of $\mathbb{CIBM}^{s\ell}$ defined by the equation $x * x \approx x$.

Theorem 12. The variety $\mathbb{BDL}$ of bounded distributive lattices is the subvariety of $\mathbb{CIBM}^{s\ell}$ defined by the equation $x * x \approx x$.

Next we will introduce the notion of *pseudocomplement* with respect to the monoidal operation of $\mathcal{A} \in \mathbb{CIBM}^{s\ell}$ or $\mathcal{A} \in \mathbb{CIBM}^{\ell}$. This notion is a generalization of the notion of pseudocomplement in the context of meet-semilattices. Let us recall this definition for distributive lattices with minimum:

Definition 13 ([3, Definition 1, p. 152]). Let $\mathcal{A} = \langle A, \wedge, \vee, 0\rangle$ be a distributive lattice with minimum element 0. An element $a \in A$ is called *pseudocomplemented* if the set $\{b \in A : a \wedge b = 0\}$ has a maximum. In this case this maximum is denoted by $\neg a$ and is called the *pseudocomplement* of a. A *pseudocomplemented distributive lattice* is a distributive lattice with minimum element 0 in which every element has a pseudocomplement.

Let us recall that if $\mathcal{A}$ is a pseudocomplemented distributive lattice, then $a \leq \neg 0$ for all $a \in A$, so A has a maximum element $1 = \neg 0$. Hence we can define the class of pseudocomplemented distributive lattices, which we denote by $\mathbb{PDL}$, in the language $\{\vee, \wedge, \neg, 0, 1\}$.

Definition 14. Let $\lambda \in \{s\ell, \ell\}$ and $\mathcal{A} \in \mathbb{CIBM}^{\lambda}$. An element $a \in A$ is called $*$-*pseudocomplemented* if the set $\{b \in A : a * b = 0\}$ has a maximum with respect to the order of the semilattice. In this case this maximum is denoted by $\neg a$ and is called the $*$-*pseudocomplement* of a.

Definition 15. The class of *commutative integral bounded pseudocomplemented λ-monoids*, $\mathbb{CIBPM}^{\lambda}$ for short, is the class of algebras $\mathcal{B} = \langle \mathcal{A}, \neg\rangle$ such that:

1. $\mathcal{A} \in \mathbb{CIBM}^{\lambda}$
2. For every $a \in A$, $\neg a = \max\{b : a * b = 0\}$

Obviously the classes $\mathbb{CIBPM}^{s\ell}$ and $\mathbb{CIBPM}^{\ell}$ are quasivarieties and in fact they are also varieties. We thank Roberto Cignoli for his personal communication stating this result.

Theorem 16. The class $\mathbb{CIBPM}^{s\ell}$ is the equational class of algebras $\mathcal{A} = \langle A, \vee, *, \neg, 0, 1\rangle$ of type $(2, 2, 1, 0, 0)$ satisfying the following equations:

(1) The set of equations defining the commutative integral bounded $s\ell$-monoids.

(2) $\neg 1 \approx 0$.

(3) $\neg 0 \approx 1$.

(4) $(x * \neg(y * x)) \vee \neg y \approx \neg y$.

As an immediate consequence of Theorem 16 we obtain an equational definition of the class $\mathbb{CIBPM}^{\ell}$.

Theorem 17. The class $\mathbb{CIBPM}^{\ell}$ is the equational class of algebras $\mathcal{A} = \langle A, \vee, \wedge, *, \neg, 0, 1\rangle$ of type $(2, 2, 2, 1, 0, 0)$ satisfying the set of equations defining the commutative integral bounded ℓ-monoids and the equations (2), (3) and (4) in Theorem 16.

In the next result we characterize the variety of pseudocomplemented distributive lattices as a subvariety of $\mathbb{CIBPM}^{s\ell}$.

Theorem 18. The variety $\mathbb{PDL}$ of pseudocomplemented distributive lattices can be defined as the subvariety of $\mathbb{CIBPM}^{s\ell}$ defined by the equation $x * x \approx x$.

It is well known that every pseudocomplemented distributive lattice is isomorphic to a subreduct of a Heyting algebra [4, Proof of Theorem 2.6]. The following two theorems are generalizations of this result.

Theorem 19. Every $\mathbb{CIBPM}^{s\ell}$-algebra is embeddable into a complete residuated lattice. Thus, the class $\mathbb{CIBPM}^{s\ell}$ is the closure under isomorphism and subalgebras of the class of algebras $\{\mathcal{A} : \mathcal{A} = \langle A, \vee, *, \neg, 0, 1\rangle$ and there are operations $\wedge, \rightarrow$ such that $\langle A, \vee, \wedge, *, \rightarrow, \neg, 0, 1\rangle \in \mathbb{RL}\}$.

Theorem 20. Every $\mathbb{CIBPM}^{\ell}$-algebra is embeddable into a complete residuated lattice. Therefore, the class $\mathbb{CIBPM}^{\ell}$ is the closure under isomorphism and subalgebras of the class of algebras $\{\mathcal{A} : \mathcal{A} = \langle A, \vee, \wedge, *, \neg, 0, 1\rangle$ and there is an operation $\rightarrow$ such that $\langle A, \vee, \wedge, *, \rightarrow, \neg, 0, 1\rangle \in \mathbb{RL}\}$.

The following theorem concerns decidability.

Theorem 21. The quasiequational theories of $\mathbb{CIBPM}^{s\ell}$ and $\mathbb{CIBPM}^{\ell}$ are decidable.

We also show that $\mathbb{CIBPM}^{s\ell}$ and $\mathbb{CIBPM}^{\ell}$ are not the equivalent algebraic semantics of any deductive system.

Theorem 22. Let $\lambda \in \{s\ell, \ell\}$. Let $\vDash_{\mathbb{CIBPM}^{\lambda}}$ be the equational consequence relation determined by $\mathbb{CIBPM}^{\lambda}$. Then $\mathcal{S}_{\mathbb{CIBPM}^{\lambda}} = \langle \mathcal{L}, \vDash_{\mathbb{CIBPM}^{\lambda}}\rangle$ is not the equivalent algebraic semantics of any deductive system.

5 Algebraization of the Gentzen systems determined by the sequent calculi $\mathbf{FL_{ew}}[\vee, *]$ and $\mathbf{FL_{ew}}[\vee, \wedge, *]$

In the following result we show the equivalence between $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ and $\mathcal{S}_{\mathbb{CIBPM}^{s\ell}}$ as Gentzen systems (see Section 1.5) by means of the translations τ and ρ defined in Theorem 9 but writing τ in the form considered in Note 1. That is, we show the algebraization of $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$.

Theorem 23. The Gentzen system $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ determined by $\mathbf{FL_{ew}}[\vee, *, \neg]$ is algebraizable, with equivalent algebraic semantics the variety $\mathbb{CIBPM}^{s\ell}$, with translations τ from $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ to $\mathcal{S}_{\mathbb{CIBPM}^{s\ell}}$ and ρ from $\mathcal{S}_{\mathbb{CIBPM}^{s\ell}}$ to $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ defined as follows:

$$\tau_{(m,1)}(p_0, ..., p_{m-1} \Rightarrow q_0) = \begin{cases} (p_0 * ... * p_{m-1}) \vee q_0 \approx q_0, & \text{if } m \geq 1 \\ 1 \approx q_0, & \text{if } m = 0 \end{cases}$$

$$\tau_{(m,0)}(p_0, ..., p_{m-1} \Rightarrow \emptyset) = \begin{cases} p_0 * ... * p_{m-1} \approx 0, & \text{if } m \geq 1 \\ 1 \approx 0, & \text{if } m = 0 \end{cases}$$

$$\rho(p_0 \approx p_1) = \{p_0 \Rightarrow p_1, p_1 \Rightarrow p_0\}$$

We also show that the result obtained in [15] (see also [14]) stating that the variety of pseudocomplemented distributive lattices is the equivalent algebraic semantics of the Gentzen system determined by the calculus obtained by deleting the implication rules from LJ (this Gentzen system is essentially the same as $\mathcal{G}_{\mathbf{FL_{ewc}}[\vee,*,\neg]}$) can be easily obtained from Theorem 23.

Theorem 24. The Gentzen system $\mathcal{G}_{\mathbf{FL_{ewc}}[\vee,*,\neg]}$ determined by the calculus $\mathbf{FL_{ewc}}[\vee, *, \neg]$ is algebraizable, with equivalent algebraic semantics the variety $\mathbb{PDL}$, with the translations τ from $\mathcal{G}_{\mathbf{FL_{ewc}}[\vee,*,\neg]}$ to $\mathcal{S}_{\mathbb{PDL}}$ and ρ from $\mathcal{S}_{\mathbb{PDL}}$ to $\mathcal{G}_{\mathbf{FL_{ewc}}[\vee,*,\neg]}$ defined in Theorem 23.

We also have that the variety $\mathbb{CIBPM}^{\ell}$ is the equivalent algebraic semantics of the Gentzen system $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$.

Theorem 25. The Gentzen system $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$ is algebraizable and the variety $\mathbb{CIBPM}^{\ell}$ is its equivalent algebraic semantics, with the translations τ from $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$ to $\mathcal{S}_{\mathbb{CIBPM}^{\ell}}$ and ρ from $\mathcal{S}_{\mathbb{CIBPM}^{\ell}}$ to $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$ defined in Theorem 23.

To finalize this section we state two straightforward consequences of the algebraization of $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ and $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$. The first one concerns the contraction rule.

Theorem 26. The contraction rule is not admissible in $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ and $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$.

The second consequence concerns the impossibility of obtaining equivalent Hilbert-style formulations of the Gentzen systems $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ and $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$. Using Theorem 22 and the algebraization results we obtain:

Theorem 27. The Gentzen systems $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ and $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$ are not equivalent to any deductive system.

6 The external deductive systems associated to $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ and $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$

In this section we study the external deductive systems associated to the Gentzen systems $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ and $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$ denoted here by $\mathcal{S}_e[\vee,*,\neg]$ and $\mathcal{S}_e[\vee,\wedge,*,\neg]$, respectively, and we study their position in the Abstract Algebraic Logic hierarchy. First let us recall the definition of external deductive system associated to a Gentzen system.

Definition 28. The *external deductive system*[2] associated to a Gentzen system $\mathcal{G}$ of type (α,β), with $0\in\alpha$ and $1\in\beta$, is the deductive system $\mathcal{S}_e(\mathcal{G})=\langle \mathcal{F}m_{\mathcal{L}},\vdash_{\mathcal{S}_e(\mathcal{G})}\rangle$ defined in the following way: for all $\Sigma\cup\{\varphi\}\subseteq \mathcal{F}m_{\mathcal{L}}$, $\Sigma\vdash_{\mathcal{S}_e(\mathcal{G})}\varphi$ iff there is a finite subset $\{\varphi_1,\ldots,\varphi_n\}\subseteq\Sigma$ such that $\emptyset\Rightarrow\varphi_1,\ldots,\emptyset\Rightarrow\varphi_n\vdash_{\mathcal{G}}\emptyset\Rightarrow\varphi$.

To show that $\mathcal{S}_e[\vee,*,\neg]$ and $\mathcal{S}_e[\vee,\wedge,*,\neg]$ are fragments of $\widehat{IPC}^*\backslash c$ we show first that $\widehat{IPC}^*\backslash c$ is the external deductive system of $\mathbf{FL_{ew}}$ and using this result, the algebraization theorems and the results about subreducts, we show that the Gentzen systems $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ and $\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$ are fragments of $\mathcal{G}_{\mathbf{FL_{ew}}}$.

Lemma 2. *$\widehat{IPC}^*\backslash c$ is the external deductive system of $\mathcal{G}_{\mathbf{FL_{ew}}}$.*

Lemma 3. *$\mathcal{G}_{\mathbf{FL_{ew}}[\vee,*,\neg]}$ is the $\{\vee,*,\neg,0,1\}$-fragment of $\mathcal{G}_{\mathbf{FL_{ew}}}$, that is, if $\mathcal{L}=\{\vee,*,\neg,0,1\}$ and $T\cup\{\Gamma\Rightarrow\Delta\}\subseteq Seq_{\mathcal{L}}^{(\omega,\{0,1\})}$, then:*

$$T\vdash_{\mathbf{FL_{ew}}}\Gamma\Rightarrow\Delta \quad \textit{iff} \quad T\vdash_{\mathbf{FL_{ew}}[\vee,*,\neg]}\Gamma\Rightarrow\Delta.$$

Lemma 4. *$\mathcal{G}_{\mathbf{FL_{ew}}[\vee,\wedge,*,\neg]}$ is the $\{\vee,\wedge,*,\neg,0,1\}$-fragment of $\mathcal{G}_{\mathbf{FL_{ew}}}$.*

Using the two last results we prove that $\mathcal{S}_e[\vee,*,\neg]$ and $\mathcal{S}_e[\vee,\wedge,*,\neg]$ are fragments of $\widehat{IPC}^*\backslash c$.

Theorem 29. For all $\Sigma\cup\{\varphi\}\subseteq Fm_{\{\vee,*,\neg,0,1\}}$, we have that

$$\Sigma\vdash_{\widehat{IPC}^*\backslash c}\varphi \text{ iff } \Sigma\vdash_{\mathcal{S}_e[\vee,*,\neg]}\varphi.$$

Theorem 30. For all $\Sigma\cup\{\varphi\}\subseteq Fm_{\{\vee,\wedge,*,\neg,0,1\}}$we have that

$$\Sigma\vdash_{\widehat{IPC}^*\backslash c}\varphi \text{ iff } \Sigma\vdash_{\mathcal{S}_e[\vee,\wedge,*,\neg]}\varphi.$$

Let us denote by $IPC_{\{\vee,*,\neg,0,1\}}$ the implication-less fragment of the intuitionistic propositional logic (we use the symbol $*$ to the additive conjunction). We prove that $\mathcal{S}_e[\vee,*,\neg]$, the fragment without conjunction and without implication of $\widehat{IPC}^*\backslash c$, is a proper subsystem of $IPC_{\{\vee,*,\neg,0,1\}}$.

Theorem 31. $\mathcal{S}_e[\vee,*,\neg]\subsetneq IPC_{\{\vee,*,\neg,0,1\}}$.

Let us denote by $IPC^*_{\{\vee,\wedge,*,\neg,0,1\}}$ the implication-less fragment of the intuitionistic propositional logic (with $*=\wedge$). We then obtain the following:

Theorem 32. $\mathcal{S}_e[\vee,\wedge,*,\neg]\subsetneq IPC^*_{\{\vee,\wedge,*,\neg,0,1\}}$.

Finally we will explain the position of our deductive systems in the Abstract Algebraic Logic hierarchy. As an immediate consequence of Theorem 27, we have that the varieties $\mathbb{CIBPM}^{s\ell}$ and $\mathbb{CIBPM}^{\ell}$ cannot be the equivalent algebraic semantics of $\mathcal{S}_e[\vee,*,\neg]$ and $\mathcal{S}_e[\vee,\wedge,*,\neg]$ respectively. In fact we prove that these external systems are not proto-algebraic. Recall that a deductive system $\mathcal{S}$ is protoalgebraic if the Leibnitz operator Ω is monotonic on the set of $\mathcal{S}$-theories (see for example [7]). This condition is equivalent to the fact that there is a set of formulas $P(p,q)$ (in two variables at most) such that:

[2] We use the name *extern* for this deductive system following A.Avron (see for instance [2]).

$$\begin{array}{ll} \emptyset \vdash_{\mathcal{S}} P(p,p), & \text{(Reflexivity)} \\ \{p\} \cup P(p,q) \vdash_{\mathcal{S}} q, & \text{(Modus Ponens)} \end{array}$$

As a consequence, if $\mathcal{S}$ is not protoalgebraic, there is no defined binary connective $\rightarrow$ such that:

$$\begin{array}{ll} \emptyset \vdash_{\mathcal{S}} p \rightarrow p, & \text{(Identity)} \\ p, p \rightarrow q \vdash_{\mathcal{S}} q, & \text{(Modus Ponens).} \end{array}$$

Theorem 33. Neither the deductive system $\mathcal{S}_e[\vee, *, \neg]$ nor $\mathcal{S}_e[\vee, \wedge, *, \neg]$ are protoalgebraic.

So these two deductive systems are not algebraizable. Nevertheless it can be seen that the algebraization results for our Gentzen systems give the following completeness statements:

Theorem 34. The variety $\mathbb{CIBPM}^{s\ell}$ is an algebraic semantics for $\mathcal{S}_e[\vee, *, \neg]$ with defining equation $p \approx 1$.

Theorem 35. The variety $\mathbb{CIBPM}^{\ell}$ is an algebraic semantics for $\mathcal{S}_e[\vee, \wedge, *, \neg]$ with defining equation $p \approx 1$.

From these results and the decidability of the quasivarieties $\mathbb{CIBPM}^{s\ell}$ and $\mathbb{CIBPM}^{\ell}$ (Theorem 21) we obtain the following:

Theorem 36. $\mathcal{S}_e[\vee, *, \neg]$ and $\mathcal{S}_e[\vee, \wedge, *, \neg]$ are decidable.

Acknowledgements

The authors thank Roberto Cignoli for his personal communication stating Theorem 16 and Francesc Esteva for his suggestions. This study is partially supported by Grant BFM2001-3329 from the Spanish DGES and Grant 2001SGR-00017 from the Generalitat de Catalunya.

Bibliography

[1] R. Adillon and V. Verdú. On a contraction-less intuitionistic propositional logic with conjunction and fusion. *Studia Logica, Special Issue on Abstract Algebraic Logic*, 65(1):11–30, 2000.

[2] A. Avron. The semantics and proof theory of linear logic. *Theoretical Computer Science*, 570:161–184, 1988.

[3] R. Balbes and P. Dwinger. *Distributive lattices*. University of Missouri Press, Columbia (Missouri), 1974.

[4] W. J. Blok and D. Pigozzi. *Algebraizable logics*, volume 396 of *Mem. Amer. Math. Soc.* A.M.S., Providence, January 1989.

[5] W. J. Blok and C. J. van Alten. The finite embeddability property for residuated lattices, pocrims and BCK-algebras. *Algebra Universalis*, 48(3):253–271, 2002.

[6] F. Bou, A. García-Cerdanya, and V. Verdú. Analysis of two fragments with negation and without implication of the logic of residuated lattices. Manuscript, 2003.

[7] J. Czelakowski. *Protoalgebraic logics*, volume 10 of *Trends in Logic—Studia Logica Library*. Kluwer Academic Publishers, Dordrecht, 2001.

[8] A. J. Gil, A. Torrens, and V. Verdú. On Gentzen systems associated with the finite linear MV-algebras. *Journal of Logic and Computation*, 7(4):473–500, 1997.

[9] U. Höhle. Commutative, residuated l-monoids. In U. Höhle and E. P. Klement, editors, *Non-classical logics and their applications to fuzzy subsets (Linz, 1992)*, volume 32 of *Theory Decis. Lib. Ser. B Math. Statist. Methods*, pages 53–106. Kluwer Acad. Publ., Dordrecht, 1995.

[10] H. Ono. Proof-theoretic methods for nonclassical logic - an introduction. In M. Takahashi, M. Okada, and M. Dezani-Ciancaglini, editors, *Theories of Types and Proofs*, MSJ Memoirs 2, pages 207–254. Mathematical Society of Japan, 1998.

[11] H. Ono. Closure operators and complete embeddings of residuated lattices. *Studia Logica*, 74(3):427–440, 2003.

[12] H. Ono. Substructural logics and residuated lattices - an introduction. In V. F. Hendricks and J. Malinowski, editors, *50 Years of Studia Logica*, volume 21 of *Trends in Logic—Studia Logica Library*, pages 193–228. Dordrecht, 2003.

[13] H. Ono and Y. Komori. Logics without the contraction rule. *The Journal of Symbolic Logic*, 50(1):169–201, 1985.

[14] J. Rebagliato and V. Verdú. On the algebraization of some Gentzen systems. *Fundamenta Informaticae, Special Issue on Algebraic Logic and its Applications*, 18:319–338, 1993.

[15] J. Rebagliato and V. Verdú. Algebraizable Gentzen systems and the deduction theorem for Gentzen systems. Mathematics Preprint Series 175, University of Barcelona, June 1995.

On Filter Logics for 'Most' and Special Predicates

Paulo A. S. Veloso Sheila R. M. Veloso

COPPE, UFRJ, Caixa Postal 68511, 21945-970 Rio de Janeiro, RJ, Brasil
{veloso, sheila}@cos.ufrj.br

Abstract

Logics for 'generally' were introduced for handling, by non-standard generalized quantifiers, assertions with vague notions (important issues in Logic and in Artificial Intelligence). Filter logic is intended to address (some versions) of 'most'. We show that filter logic can be faithfully embedded into a first-order theory of compatible predicates. We also use representative predicates to eliminate the generalized quantifier. These devices permit using classical first-order methods to reason about consequence in filter logic and help clarifying the role of such logics for 'generally'.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

In this paper we show that filter logic (FL) can be faithfully embedded into a first-order logic theory of certain predicates. We thus provide a framework where the semantic intuitions of FL (for 'most') can be combined with proof methods for classical first-order logic (CFOL). This framework supports theorem proving in FL, as it permits using proof procedures and theorem provers for CFOL. It will also help clarifying the role of such extensions of CFOL for 'generally'.

Some logics for 'generally' (LG) were introduced for handling assertions with some vague notions, such as 'generally' and 'rarely', by non-standard generalized quantifiers [2, 9].[1] Their expressive power is quite convenient and they have sound and complete deductive systems. This, however, still leaves open the question of theorem proving, namely theorem provers for them. We will show that special predicates (representative and compatible predicates) allow one to use existing theorem provers (for CFOL) for this task.[2]

This paper is structured as follows. In section 2 we review some ideas about LG. Section 3 introduces representative sets and functions. In section 4 we examine the translation of ∇ by predicates and compatibility. Section 5 introduces representative axioms to internalize our translation. Section 6 exhibits a framework for reducing filter reasoning to CFOL theories of compatible predicates and examines some applications. Section 6 contains some concluding remarks about our approach.

2 Logics for Reasoning about 'Generally'

We now examine some ideas about LG: motivations and some technical aspect

We first briefly consider some motivations underlying LG [11, 12]. Assertions and arguments involving some vague notions, such as 'generally', 'rarely', 'most', 'several', etc., occur often in ordinary language and in some branches of science. One often meets assertions such as "Several bodies expand when heated", "Most birds fly" and "Metals rarely are liquid under ordinary conditions". The assertions "Whoever likes sports watches SporTV" and "Boys generally like sports" appear to lead to "Boys generally watch SporTV". Qualitative reasoning about such notions often occurs in everyday life. We can express some assertions in CFOL: e. g. "All birds fly" and "Some birds fly". But, what about vague assertions like "Birds generally fly"? We wish to express such assertions and reason about them in a precise manner. Logics for some vague notions can be obtained by extending CFOL with an operator ∇ and axioms [9]. So, one can express "Birds generally fly" by $\nabla v F(v)$.

Logics for 'generally' extend CFOL by generalized quantifiers, interpreted as 'generally' [9, 12]. We now briefly review FL. We use $\mathsf{L}(\rho)$ for the usual CFOL language (with equality $\simeq$) of a given signature ρ. It is convenient to have a fixed, though arbitrary, ordering for the set V of variables: in each list of variables, they will be listed according to this ordering. We will use $\mathsf{L}^{\nabla}(\rho)$ for the extension of $\mathsf{L}(\rho)$ by the new operator ∇. The formulas of $\mathsf{L}^{\nabla}(\rho)$ are built by the usual formation rules and a new variable-binding formation rule giving *generalized formulas*: for each variable v, if φ is a formula in $\mathsf{L}^{\nabla}(\rho)$ then so is $\nabla v\varphi$. Other syntactic notions, like substitution ($\varphi[v/t]$), can be easily adapted.

We provide semantic interpretation for 'generally' by enriching structures with filters and extending the definition of satisfaction to ∇. A *filter structure* $\mathcal{A}^{\mathcal{F}} = \langle \mathcal{A}, \mathcal{F} \rangle$ for signature ρ consists of a usual structure $\mathcal{A}$ for ρ together with a filter $\mathcal{F}$ over the universe A of $\mathcal{A}$. We extend the usual definition of *satisfaction* of a formula in a structure under assignment $\underline{a}$ to its (free) variables, using the extension $\mathcal{A}^{\mathcal{F}}[\varphi(\underline{a}, z)] := \{b \in A : \mathcal{A}^{\mathcal{F}} \models \varphi(\underline{a}, z)[\underline{a}, b]\}$, as follows: $\mathcal{A}^{\mathcal{F}} \models \nabla z \varphi(\underline{a}, v)[\underline{a}]$ iff $\mathcal{A}^{\mathsf{u}}[\varphi(\underline{a}, z)]$ is in $\mathcal{F}$. Other semantic notions, such as reduct, model ($\mathcal{A}^{\mathcal{F}} \models \Gamma$) and validity, are as usual [4, 7].[3] The notion of *filter consequence* is as expected: $\Gamma \models^{\mathcal{F}} \tau$ iff $\mathcal{A}^{\mathcal{F}} \models \tau$, for every filter model $\mathcal{A}^{\mathcal{F}} \models \Gamma$.

We can formulate deductive systems for our LG by adding schemas to a calculus for CFOL. To set up a deductive system $\vdash^{f}$ for FL, we take a sound and complete deductive calculus for CFOL, with Modus Ponens as the sole inference rule (as in [7]), and extend its set Λ_ρ of axiom schemas by adding a set Δ^ρ of new axiom schemas (coding properties of filters), to form $\Lambda^\rho := \Lambda_\rho \cup \Delta^\rho$. This set Δ^ρ consists of all the universal generalizations of the following five schemas (where φ, ψ and θ are formulas of $\mathsf{L}^{\nabla}(\rho)$):

[1] These logics are related to variants of default logic [1], but they are quite different logical systems, both technically and in terms of intended interpretations [10]. The expressive power of our generalized quantifiers [3] paves the way for other possible applications where it may be helpful: e. g. expressing some fuzzy concepts [12].

[2] We will concentrate on FL (for 'most'), but the main lines can be adapted to some other LG (cf. [13, 8]).

[3] Satisfaction of a formula hinges only on the realizations assigned to its symbols.

$[\nabla\exists]$ $\nabla v\varphi \to \exists v\varphi$ $\qquad$ $[\forall\nabla]$ $\forall v\varphi \to \nabla v\varphi$

$[\nabla\alpha]$ $\nabla v\varphi \leftrightarrow \nabla u\varphi[v := u]$ $\qquad$ for a new variable u not occurring in φ

$[\to \nabla]$ $\forall v(\psi \to \theta) \to (\nabla v\psi \to \nabla v\theta)$ $\qquad$ $[\nabla\wedge]$ $(\nabla v\psi \wedge \nabla v\theta) \to \nabla v(\psi \wedge \theta)$

These schemas express properties of filters, with $[\nabla\alpha]$ covering alphabetic variants. Derivations are CFOL derivations from the schemas: $\Gamma \vdash^f \varphi$ iff $\Gamma \cup \Lambda^\rho \vdash \varphi$. Other usual deductive notions, such as (maximal) consistent sets, witnesses and conservative extension [4, 7], can be easily adapted.

We have a sound and complete deductive system for our logic ($\models^{\mathcal{F}} = \vdash^f$), which is a proper conservative extension of CFOL [6, 12, 11].[4] In the sequel, we show that we can reason about such 'most'-assertionss entirely within CFOL by means of special predicates: we will show that FL can be faithfully embedded into a CFOL theory of compatible predicates.[5]

3 Representative Sets and Functions

We will now introduce the notions of representative sets and functions.

We first examine some ideas about representative sets. "Most birds fly" ($\nabla vF(v)$) is intended to mean that the set of flying birds is an 'important' set of birds. As such, we cannot infer from it that an arbitrary bird flies (we do not have instantiations: $\nabla z\varphi \to \varphi[z/t]$ is not valid). We may consider as typical a bird with the properties that most birds have; so, we can instantiate generalized formulas to typical objects: for each typical bird b (if any) we have $\nabla vF(v) \to F(b)$. Such typical objects are somewhat elusive: they may fail to exist. (What would be a typical natural number? Considering 'most' as cofinite, no standard natural can be typical.) Now, what about the converse: from which set of birds can we infer $\nabla vF(v)$? This appears problematic, like experimental induction. Let us call a set S of birds sufficiently representative when one can infer $\nabla vF(v)$ from $F(b)$, for all $b \in S$. These ideas motivate calling a set S representative (with respect to flying) when $\nabla vF(v)$ is equivalent to $F(b)$, for all $b \in S$.

We now examine representative objects in a filter structure $\mathcal{A}^{\mathcal{F}}$. A *representative set* for a generalized sentence $\nabla z\varphi$ is a subset $S \subseteq A$ such that $\mathcal{A}^{\mathcal{K}} \models \nabla z\varphi$ iff, for every $a \in S$, $\mathcal{A}^{\mathcal{K}} \models \varphi[a]$. We can extend this idea to representative functions for formulas with free variables. A *representative function* for a generalized formula $\nabla z\varphi$ with m free variable is an m-ary function $\underline{r} : A^m \to \wp(A)$, assigning to each m-tuple $\underline{a} \in A^m$ a representative set $\underline{r}(\underline{a}) \subseteq A$: $\mathcal{A}^{\mathcal{K}} \models \nabla z\varphi[\underline{a}]$ iff, for every $b \in \underline{r}(\underline{a})$, $\mathcal{A}^{\mathcal{K}} \models \varphi[\underline{a}, b]$.[6]

In the sequel, we will show that we can translate 'most'-assertions by means of special predicates. We will use expansions of signatures by new predicates. Given a signature σ, consider for each $n \in \mathbb{N}$, a new $(n+1)$-ary predicate symbol p_n not in σ, and form the expansion $\sigma[P] := \sigma \cup P$, obtained by adding the set $P := \{p_n : n \in \mathbb{N}\}$. We will translate ∇ by universal relativizations to these predicates.

4 Translation of 'Most' to First-Order

We will now consider the translation of ∇ by predicates and examine the idea of compatibility.

We translate ∇ by universal relativizations to new predicate symbols in the set $P := \{p_n : n \in N\}$. We transform $\nabla z\theta(\underline{u}, z)$, with list $\underline{u}$ of m free variables, to $(\forall z : p_m|\underline{u})\theta(\underline{u}, z)$, which abbreviates $\forall z[p_m(\underline{u}, z) \to \theta(\underline{u}, z)]$. Now, we eliminate ∇ from a formula φ of $\mathsf{L}^\nabla(\sigma[P])$ by replacing (inside-out) each subformula $\nabla z\theta(\underline{u}, z)$ of φ by $(\forall z : p_{|\underline{u}|}(\underline{u}))\theta(\underline{u}, z)$ to obtain a formula of $\mathsf{L}(\sigma[P])$: its P-transform φ_P. This process defines a ∇-eliminating P-translation $()_P : \mathsf{L}^\nabla(\sigma[P]) \to \mathsf{L}(\sigma[P])$.

We now introduce compatibility. Our transformation should preserve provability. For this purpose, we need some (FO) information connecting the new predicates symbols: their compatibility. A motivation for compatibility comes from examining the translation of the filter schemas into FO: each instance of the schemas $[\forall\nabla]$ and $[\nabla\alpha]$ becomes logically valid, but no so for the other schemas. Compatibility will provide a way to handle schemas $[\nabla\exists]$, $[\nabla\wedge]$ and $[\to \nabla]$. Our requirements are related to the P-translation of generalized formulas: $(\nabla z\varphi)P := (\forall z : p_m|\underline{u})\varphi_P$, for $\nabla z\varphi$ with list $\underline{u}$ of m free variables. We also consider the list $\underline{v}_n$ of the n variables $v_1, \ldots, v_n$, for each $n \in N$.

[4] Soundness is clear and completeness can be established by adapting Henkin's familiar proof for CFOL. It is not difficult to see that we have conservative extensions of CFOL. These extensions are proper, because some sentences, such as $\exists u\nabla zu \simeq z$, cannot be expressed without ∇ [12, 11].

[5] The crucial issue here is having a faithful interpretation. An interpretation into CFOL can be obtained by replacing throughout ∇ by $\forall$: each filter axiom is translated to a logical validity, but this interpretation is not faithful.

[6] We may view an m-ary function $\underline{r} : A^m \to \wp(A)$ as an $(m+1)$-ary relation $r \subseteq A^m \times A$ given by $\langle \underline{a}, b\rangle \in r$ iff $b \in \underline{r}(\underline{a})$.

The inclusion in the translation is vacuously satisfied by a void predicate. So, our first requirement $\underline{p}_n(a) \neq \emptyset$ is the *non-voidness axiom* $\exists(p_n)$: $\forall \underline{v}_n \exists v_0 p_n(\underline{v}_n, v_0)$. The intuition of having a representative set for each n-tuple suggests a second compatibility requirement, $\underline{p}_{n+1}(\underline{a}, b) \subseteq \underline{p}_n(\underline{a})$, expressed by the *decreasing axiom* $\downarrow(p_n)$: $\forall \underline{v}_n \forall v_{n+1} \forall v_0 [p_{n+1}(\underline{v}_n, v_{n+1}, v_0) \rightarrow p_n(\underline{v}_n, v_0)]$. The third requirement mirrors the idea that new free variables do not matter, as expressed by the *restriction axiom* $(p^{\underline{v}}_n\varphi)$: $\forall \underline{u} \forall \underline{v} [\forall z (p_{m+k}(\underline{u}, \underline{v}, z) \rightarrow \varphi) \rightarrow \forall z (p_m(\underline{u}, z) \rightarrow \varphi)]$ (where $\underline{v}$ is a list of k variables other than $\underline{u}$ and z).

We will extend these axioms to schemas for sets. The *non-voidness* and *decreasing schemas* are the sets $\exists[P] := \{\exists(p_n) : p_n \in P\}$ and $\downarrow[P] := \{\downarrow(p_n) : p_n \in P\}$, respectively. The *restriction schema* for $\mathsf{L}^\nabla(\sigma)$ is the set $[P \uparrow \sigma^\nabla]$ consisting of the restriction axioms for the P-translation of every generalized formula $\nabla z\varphi$ in $\mathsf{L}^\nabla(\sigma)$. We shall also use the *compatibility schema* $\Omega^\star := \exists[P] \cup \downarrow[P] \cup [P \uparrow \sigma^\nabla]$.

We now examine some properties of compatible predicates.

Lemma 1. The P-translations of the filter axioms in Δ^σ follow from the compatibility schema $\Omega^\star$.

Proof: The schema $\exists[P]$ yields the translation of $[\ \nabla\exists\ ]$ and the P-translation of each axiom in $[\nabla\wedge\ ]$ and $[\rightarrow \nabla\ ]$ follows from the other two schemas (which permit adjusting the free variable). ◁

We see that our translation interprets FL into a CFOL theory of compatible predicates.

Proposition 2. The P-translation $()_P$ restricted to $\mathsf{L}^\nabla(\sigma)$ interprets FL (for signature σ) into any CFOL theory $\Sigma \subseteq \mathsf{L}(\sigma[P])$ where the new predicate symbols in P are compatible (i. e. $\Omega^\star \subseteq Cn(\Sigma)$): for each set $\Gamma \cup \{\tau\}$ of sentences of $\mathsf{L}^\nabla(\sigma)$, if $\Gamma \vdash^f \tau$ then $\Sigma \cup \Gamma_P \vdash \tau_P$.

Proof: The assertion follows from the preceding lemma, giving $[\Delta^\sigma]_P \subseteq Cn(\Omega^\star)$. ◁

5 Representative Predicates and Axioms

We now introduce representative axioms to internalize our translation of ∇ by special predicates.

We first formulate representative predicates by sentences in the expanded language $\mathsf{L}^\nabla(\sigma[P])$.

Given a generalized formula $\nabla z\varphi$ of $\mathsf{L}^\nabla(\sigma[P])$ with list $\underline{u}$ of m free variables, the formula $(\forall z : p_m|\underline{u})\varphi$ is in $\mathsf{L}^\nabla(\sigma[P])$, and the *representative axiom* $\partial(p_m \nabla z\varphi)$ for $\nabla z\varphi$ is the universal closure of the formula $\nabla z\varphi \leftrightarrow (\forall z : p_m|\underline{u})\varphi$ of $\mathsf{L}^\nabla(\sigma[P])$. We extend this idea to sets of formulas. The *representative schema* for a set Ψ of formulas of $\mathsf{L}^\nabla(\sigma[P])$ is the set $|\partial[\Psi]$ consisting of the representative axioms for every generalized formula $\nabla z\varphi$ in Ψ. When Ψ is the set of all the (generalized) formulas of signature σ, we use $|\partial[\sigma^\nabla] := |\partial[\mathsf{L}^\nabla(\sigma)]$ for the *representative schema* for $\mathsf{L}^\nabla(\sigma)$.

We can now see that the representative axioms internalize our P-translation $()_P$ into $\mathsf{L}(\sigma[P])$.

Proposition 3. The representative schema $|\partial[\Psi]$ for a set Ψ of formulas of $\mathsf{L}^\nabla(\sigma[P])$, closed under subformulas, yields the equivalence between a formula ψ in Ψ and its P-transform ψ_P: $|\partial[\Psi] \vdash^f \psi \leftrightarrow \psi_P$, for $\psi \in \Psi$.

Proof: By induction on the structure of the formulas.

Thus, representative axioms enable the elimination of the new quantifier ∇ in favor of representative predicates. In particular, $|\partial[\sigma^\nabla] \vdash^f \tau \leftrightarrow \tau_P$, for every sentence τ of $\mathsf{L}^\nabla(\sigma)$. We can now see that the restriction axioms are filter consequences of the representative schema. ◁

Corollary 4. The restriction axioms in $[P \uparrow \sigma^\nabla]$ are filter consequences of the representative schema $|\partial[\sigma^\nabla]$.

Proof: Each restriction axiom $(p^{\underline{v}}_n\varphi)$ is a filter consequence of two representative axioms. ◁

We will now establish the conservativeness of extensions by compatible and representative axioms.

Representative sets are elusive: many structures fail to have them. Nevertheless, we can add them conservatively into theories. To establish this fact, we introduce some terminology and show that a filter structure has compatible relations that are representative for a finite set of generalized formulas.

Consider a filter structure $\mathcal{A}^\mathcal{F} = \langle \mathcal{A}, \mathcal{F} \rangle$. We call an n-ary function $\underline{r} : A^n \rightarrow \wp(A)$ *strong* iff $\underline{r}(\underline{a}) \in \mathcal{F}$, for every $\underline{a} \in A^n$. Given an $(n+1)$-ary function $\underline{s} : A^{n+1} \rightarrow \wp(A)$, we call $\underline{s}$ *inclusive* with respect to n-ary function $\underline{r}$ iff, for every $\langle a_1, \ldots, a_n, b\rangle \in A^{n+1}$, $\underline{s}(a_1, \ldots, a_n, b) \subseteq \underline{r}(a_1, \ldots, an)$. We will call a set R of relations on A *adequate* iff the associated functions into $\wp(A)$ are strong and inclusive.

A filter structure has adequate relations that are representative for a finite set of formulas.

Lemma 5. A filter structure $\mathcal{A}^{\mathcal{F}}$ has adequate relations that are representative for a finite set Φ of formulas.

Proof: The functions are intersections of extensions in $\mathcal{F}$: with v_m as the highest free variable in the generalized formulas of Φ, take $\underline{r}_n(\underline{a})$ as the intersection of the extensions $\mathcal{A}^{\mathcal{F}}[\varphi(\underline{a}, z)]$ in $\mathcal{F}$ for $\nabla z\varphi$ in Φ with free variables up to v_n, for $0 \leq n \leq m$. The associated relations are adequate and representative. ◁

So, one can always conservatively extend a theory by compatible and representative predicates.

Proposition 6. Given a set Γ of sentences of $\mathsf{L}^{\nabla}(\sigma)$, the theory $\Gamma^{\star}[\partial] := \Gamma \cup \Omega^{\star} \cup \partial[\sigma^{\nabla}] \subseteq \mathsf{L}^{\nabla}(\sigma[P])$ is a conservative extension of $\Gamma \subseteq \mathsf{L}^{\nabla}(\sigma)$: $\Gamma^{\star}[\partial] \vdash^{f} \tau$ iff $\Gamma \vdash^{f} \tau$, for each sentence $\tau \in \mathsf{L}^{\nabla}(\sigma)$.

Proof: The assertion follows from the previous lemma and corollary: $\Gamma \cup \exists[P] \cup \downarrow [P] \cup \partial[\sigma^{\nabla}]$ is a conservative extension of Γ (by compactness) and $[P \uparrow \sigma^{\nabla}] \subseteq Cn^{f}(\partial[\sigma^{\nabla}])$. ◁

6 Framework for Reasoning with 'Most' in First-Order

Now we exhibit our reduction framework based on a faithful interpretation.

We first put together our results to see that we have a common (conservative) extension.

$$\begin{array}{c} \dfrac{\mathsf{L}^{\nabla}(\sigma)}{\Gamma} \\ \Lambda \\ \dfrac{\Gamma \cup \Omega^{\star} \cup |\partial[\sigma^{\nabla}]}{\mathsf{L}^{\nabla}(\sigma[P])} \end{array} \quad \frac{1}{4} \quad \begin{array}{c} \dfrac{\mathsf{L}^{\nabla}(\sigma[P])}{\Gamma_P \cup \Omega^{\star}} \\ \cap \\ \dfrac{\Gamma_P \cup \Omega^{\star} \subseteq \mathsf{L}(\sigma[P])}{\mathsf{L}^{\nabla}(\sigma[P])} \end{array}$$

Corollary 7. Given a set Γ of sentences of $\mathsf{L}^{\nabla}(\sigma)$, the theory $\Gamma^{\star}[\partial|] := \Gamma \cup \Omega^{\star} \cup |\partial[\sigma^{\nabla}] \subseteq \mathsf{L}^{\nabla}(\sigma[P])$ is a conservative extension of $\Gamma \subseteq \mathsf{L}^{\nabla}(\sigma)$ which extends the CFOL theory $\Gamma_P \cup \Omega^{\star} \subseteq \mathsf{L}(\sigma[P])$.

Proof: The assertion follows from the previous results: the propositions in 5. ◁

So, we have a faithful interpretation of FL into the theory of compatible predicates.

Theorem 8. The P-translation $()_P$ restricted to $\mathsf{L}^{\nabla}(\sigma)$ interprets faithfully FL (for σ) into the CFOL theory $\Omega^{\star} \subseteq \mathsf{L}(\sigma[P])$: $\Gamma \vdash^{f} \tau$ iff $\Gamma_P \cup \Omega^{\star} \vdash \tau_P$, for a set $\Gamma \cup \{\tau\}$ of sentences of $\mathsf{L}^{\nabla}(\sigma)$.

Proof: The assertion follows from the previous results: the proposition in 4 and the preceding corollary. ◁

We thus have a sound and complete reduction of filter consequence to CFOL derivability with compatible predicates: establishing $\Gamma \models^{\mathcal{K}} \tau$ amounts to showing that $\Gamma_P \cup \Omega^{\star} \vdash \tau_P$.

We will now examine some examples illustrating the application of our reduction procedure.

As a simple example, we see that $\nabla z \forall u L(u, z) \vdash^{f} \forall u \nabla z L(u, z)$, since the translated conclusion $[\forall u \nabla z L(u, z)]_P$ is a CFOL consequence of the translated hypothesis $[\nabla z \forall u L(u, z)]_P$ together with the decreasing axiom $\downarrow (p_0)$. Similarly, we can reduce $\exists v \nabla z \varphi \vdash^{f} \nabla z \exists v \varphi$ to CFOL consequence (from $[P \uparrow \sigma^{\nabla}]$). We also see that $\{\nabla z \psi, \nabla z(\psi \to \theta)\} \vdash^{f} \nabla z \theta$, because $[\nabla z \theta]_P$ is a CFOL consequence of the translated hypotheses $[\nabla z \psi]_P$ and $[\nabla z(\psi \to \theta)]_P$ together with the schemas $\downarrow [P]$ and $[P \uparrow \sigma^{\nabla}]$. For an induction-like example, consider a universe of emeralds and imagine every emerald examined to be green. If we also assume that most emeralds are similar to those examined and that similarity generally transfers colors, then we can infer that most emeralds are green: $\nabla z G(z)$. Here Γ consists of ε: $\forall u(E(u) \to G(u))$, δ: $\nabla z \exists u(E(u) \wedge S(u, z))$ and γ: $\nabla z \nabla u[(E(u) \wedge S(u, z)) \to (G(u) \to G(z))]$.

In many practical cases (as in databases, for instance), we deal only with finitely many formulas. As the preceding examples indicate, in such cases one needs only a finite number of new predicates and axioms related to generalized subformulas of the formulas involved. These ideas are also useful for investigating provability in FL, as we will now illustrate with some cases involving simply generalized formulas: those of the form $\nabla z \varphi$, for φ without ∇.

Proposition 9. Consider a set Σ of sentences and a formula ψ in $\mathsf{L}(\sigma)$.

1. For each formula $\theta \in \mathsf{L}(\sigma)$: $\Sigma \cup \nabla z\psi \vdash^f \nabla z\theta$ iff $\Sigma \vdash \forall z(\psi \to \theta)$ and $\Sigma \vdash^f \nabla z\theta$ iff $\Sigma \vdash \forall z\theta$.
2. For each sentence $\tau \in \mathsf{L}(\sigma)$: $\Sigma \cup \nabla z\psi \vdash^f \tau$ iff $\Sigma \cup \exists z\psi \vdash \tau$.
3. There exists a sentence $\tau \in \mathsf{L}(\sigma)$ so that $\Sigma \vdash^f \nabla z\psi \leftrightarrow \tau$ iff $\Sigma \vdash \exists z\psi \to \forall z\psi$.

Proof: Reduce to compatible axioms. For (1): ($\Rightarrow$), choose compatible functions giving the nonempty z-extension of $\psi \wedge \neg\theta$ in some model $\mathcal{M} \models \Sigma$. Part (2) is like (1) and they yield (3). ◁

7 Conclusion

Logics for 'generally' were introduced for handling assertions with vague notions, such as 'generally', 'most', 'several'. We have established that FL (filter logic) can be faithfully embedded into a CFOL theory of compatible predicates. This provides a CFOL reduction of filter consequence, thus allowing the use of methods for CFOL. So, there are many proof procedures and theorem provers at one's disposal [5]. In addition, this approach helps to clarify the place of FL: despite its semantics based on filters, this extension of CFOL can be regarded as (part of) a CFOL theory of compatible predicates.

The development has concentrated on FL (for 'most'), but its main lines can be adapted to some other LG.[7] Our framework is not meant as a competitor to non-monotonic logics, although it does solve monotonically various problems (e. g., generic reasoning) addressed by non-monotonic approaches.

As special predicates enable using any available classical proof methods, we expect to have paved the way for theorem proving in FL for 'most'. In fact, our framework permits combining the intuitions about 'important' subsets of the universe, extensions of CFOL by non-standard generalized quantifiers and proof methods for CFOL. Such combinations appear to be very fruitful, deserving further consideration.

Acknowledgements

Partial financial support from the Brazilian agencies FAPERJ (E-26/152.395/2002 and E-26/131.180/2003) and CNPq (471608/03-3 and Locia Project). The authors gratefully acknowledge helpful conversations with Walter Carnielli.

Bibliography

[1] G. Antoniou. *Nonmonotonic Reasoning*. Cambridge, MA: MIT Press, 1997.

[2] J. Barwise and R. Cooper. Generalized quantifiers and natural language. *Ling. & Philosophy* 4: 159–219, 1981.

[3] J. Barwise and S. Feferman (eds.) *Model-Theoretic Logics*. New York: Springer, 1985.

[4] C. C. Chang and H. J. Keisler. *Model Theory*. Amsterdam: North-Holland, 1973.

[5] C. Chang and R. Lee. *Symbolic Logic and Mechanical Theorem Proving*. New York: Academic, 1973.

[6] W. A. Carnielli and P. A. S. Veloso. Ultrafilter logic and generic reasoning. In G. Gottlob, A. Leitsch and D. Mundici, (eds.) *Computational Logic and Proof Theory* (LNCS 1289): 34–53, Berlin: Springer, 1997.

[7] H. B. Enderton. *A Mathematical Introduction to Logic*. New York: Academico, 1972.

[8] A. Fuhrmann. Some remarks on ultrafilter logic. *Studia Logica* 73: 197–207, 2003.

[9] M. Grácio. *Modulated Logics and Reasoning under Uncertainty* (in Port.). D. Sc. diss., Unicamp, Campinas, 1999.

[7] The compatibility axioms in the reduction for FL are somewhat more modular than in the case of ultrafilter logic, where the axioms of the special functions also depend on the source language [13]. Other approaches add a predicate or function symbol for each generalized formula (like Skolem functions).

[10] A. M. Sette, W. A. Carnielli and P. A. S. Veloso. An alternative view of default reasoning and its logic. In E. H. Haeusler and L. C. Pereira (eds.) *Pratica: Proofs, Types and Categories*: 127–158, Rio: PUC-Rio, 1999.

[11] P. A. S. Veloso. On 'almost all' and some presuppositions. *Manuscrito* XXII: 469–505, 1999

[12] P. A. S. Veloso and W. A. Carnielli. Logics for qualitative reasoning. *CLE e-Prints* 1, 2001.

[13] S. R. M.Veloso and P. A. S. Veloso. On special functions and theorem proving in logics for 'generally'. In G. Bittencourt and G. Ramalho (eds.) *Advances in Artificial Intelligence* (LNAI 2507): 1–10, Berlin: Springer, 2002.

Towards a Metalogic for Security Protocol Analysis

Carlos Caleiro[1] Luca Viganò[2] David Basin[2]

[1] CLC, Department of Mathematics, IST, Lisbon, Portugal
cs.math.ist.utl.pt/ccal.html

[2] Department of Computer Science, ETH Zurich, Switzerland
www.infsec.ethz.ch/∼vigano www.infsec.ethz.ch/∼basin

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

Many security protocols have been proposed to help build secure distributed systems. Given how difficult it is for humans to predict all possible ways for distributed computation to proceed, it is not so surprising that attacks have been found on many protocols that were originally believed to be secure. Due to the subtlety of the problem, the use of formal methods for analyzing security protocols has been gaining popularity. These include approaches based on process algebras, e.g. [5, 14, 20, 25], which support elegant models but lack a suitable logical language to express protocol properties; model-checking and related techniques, such as [2, 3, 11, 22, 26], which are suitable for automation but rely on simplifying assumptions (to yield finite models) and hence are difficult to use reliably on different applications; special-purpose epistemic logics like BAN, e.g. [10], which provide for high-level knowledge-based formalizations of protocols and their properties, but whose semantics is complex, restricted, or simply lacking; and inductive theorem proving, like [24], which are general, but require time consuming interactive theorem proving by experienced researchers.

In this paper, we report on our work-in-progress on the formalization of a suitable version of temporal logic for communicating agents which provides both an *object level tool*, where we can specify and reason about specific protocols, and a *metalevel tool* for the compared analysis of security protocol models and properties. Our starting point is the work of [15, 17], which focus on the expressibility of properties from the local point of view of each agent, and that we extend in order to express also global properties. Besides its very clean interpretation structures, which provide a nice and intuitive model of distributed systems, our reasons for using this logic are primarily threefold. First, its temporal dimension can be effectively used to formalize and reason about *interleaved* protocol executions; this is in contrast to other approaches based on epistemic or doxastic logics, which are not well-suited for reasoning about such interleavings but consider only single protocol executions. Second, its distributed dimension, with explicit agent identification, supports formalizing the different security properties that the protocols have been designed to achieve, such as secrecy of information and different notions of authentication between agents. Finally, it is well-suited for specifying communicating agents in distributed systems.

Using the logic we are able to specify a protocol-independent distributed communication model, on top of which protocols can be formally defined and analyzed. For instance, we have used the logic to analyze a number of protocols and properties the protocols are supposed to establish. In particular, we have analyzed the well-known Needham-Schroeder Public-Key Protocol (NSPK) [18]. We have thereby been able both to find the usual man-in-the-middle attack to the NSPK and to show that authentication properties hold for the corrected version NSL (given by Lowe to prevent the man-in-the-middle attack).

The principal aim of our work, however, is not merely the concrete analysis of specific protocols. Rather, our long-term objective is to use our logic as a *metalevel tool* for the compared analysis of security protocol models and properties. Our logic provides a basis to rigorously investigate general metalevel properties of different protocol models, by establishing modeling and analysis simplification techniques that may contribute to the sound design of effective protocol validation tools. In this regard, we believe that our logic can contribute to clarifying the concepts involved through a natural representation of the underlying computational models. We anticipate several applications. The most direct consists of a rigorous account of widely used simplification techniques, namely by reasoning about (formally proving) the correctness of widely used simplification principles, like bounds on the number of principals involved, the adequacy of the intruder trace as an abstraction of the hostile communication channel, step-compression and other reduction, abstraction and approximation techniques (see, for example, [1, 3, 4, 6, 9, 12, 16, 23]). A number of promising preliminary results are described in [7, 8].

2 Distributed temporal logic

DTL [15] is a logic for reasoning about temporal properties of distributed systems from the local point of view of its agents, which are assumed to execute sequentially and to interact by means of synchronous event sharing. Distribution is implicit, making it easier to state the properties of an entire system through the local properties of its component agents and their interaction. Herein, we introduce a version of DTL tailored to allow also for the smooth spelling and proof of global properties.

The logic is defined in the context of a given *signature* of a distributed system,

$$\langle Id, \{Act_i\}_{i \in Id}, \{Prop_i\}_{i \in Id} \rangle$$

where Id is a finite set (of agent identifiers) and, for each $i \in Id$, Act_i is a set (of local action symbols) and $Prop_i$ is a set (of local state propositions). The *global language* $\mathcal{L}$ is defined by the grammar

$$\mathcal{L} ::= @_i[\mathcal{L}_i] \mid \bot \mid \mathcal{L} \Rightarrow \mathcal{L}\,,$$

where the *local languages* $\mathcal{L}_i$ for each $i \in Id$ are defined by

$$\mathcal{L}_i ::= Act_i \mid Prop_i \mid \bot \mid \mathcal{L}_i \Rightarrow \mathcal{L}_i \mid \mathcal{L}_i \,\mathsf{U}\, \mathcal{L}_i \mid \mathcal{L}_i \,\mathsf{S}\, \mathcal{L}_i \mid @_j[\mathcal{L}_j]\,,$$

with $j \in Id$. Locally for an agent, U and S are respectively the *until* and *since* temporal operators. Furthermore, actions correspond to true statements of an agent when they have just occurred, whereas state propositions characterize the current local states of the agents. Note that $@_j[\varphi]$ means different things depending on the context. If it is a global formula, it means that φ holds at the current local state of agent j. If it is a local formula appearing inside an $@_i$-formula, it means that agent i has just communicated with agent j for whom φ held.

We can then define a number of other operators as abbreviations as usual, e.g. $\neg$, $\top$, $\vee$, $\wedge$, $\Leftrightarrow$, as well as:

$\mathsf{X}\,\varphi$	$\equiv \bot \,\mathsf{U}\, \varphi$	next	$\dagger$	$\equiv \neg \bigcirc \top$	in the end
$\mathsf{Y}\,\varphi$	$\equiv \bot \,\mathsf{S}\, \varphi$	previous	$*$	$\equiv \neg \mathsf{Y} \top$	in the beginning
$\mathsf{F}\,\varphi$	$\equiv \top \,\mathsf{U}\, \varphi$	sometime in the future	$\mathsf{F}_\circ\,\varphi$	$\equiv \varphi \vee \Diamond_F \varphi$	now or sometime in the future
$\mathsf{P}\,\varphi$	$\equiv \top \,\mathsf{S}\, \varphi$	sometime in the past	$\mathsf{P}_\circ\,\varphi$	$\equiv \varphi \vee \mathsf{P}\,\varphi$	now or sometime in the past
$\mathsf{G}\,\varphi$	$\equiv \neg \Diamond_F \neg \varphi$	always in the future	$\mathsf{G}_\circ\,\varphi$	$\equiv \varphi \wedge \Box_F \varphi$	now and always in the future
$\mathsf{H}\,\varphi$	$\equiv \neg \mathsf{P} \neg \varphi$	always in the past	$\mathsf{H}_\circ\,\varphi$	$\equiv \varphi \wedge \mathsf{H}\,\varphi$	now and always in the past

The interpretation structures of $\mathcal{L}$ are built upon adequate forms of Winskel's *event structures* [27]. A *local life-cycle* (of agent $i \in Id$) is a pair $\lambda_i = (Ev_i, \rightarrow_i)$ where Ev_i is a set (of local events) and $\rightarrow_i \subseteq Ev_i \times Ev_i$ is a (local successor) relation such that $\rightarrow_i^*$ defines a well-founded total order of causality on Ev_i. Of course, Ev_i can be finite or infinite, but it is always denumerable. A *local configuration* of λ_i is any finite set $\xi_i \subseteq Ev_i$ such that if $e' \in \xi_i$ and $e \rightarrow_i e'$ then $e \in \xi_i$. We denote the set of all local configurations of λ_i by Ξ_i. Clearly, every local configuration $\xi_i \neq \emptyset$ has a maximum event that we denote by $last(\xi_i)$. Moreover, for every local configuration $\xi_i \neq Ev_i$ there exists a unique next event $next(\xi_i)$, corresponding to the minimum event in $Ev_i \setminus \xi_i$, such that $\xi_i \cup \{next(\xi_i)\}$ is a local configuration.

A *distributed life-cycle* is a family $\lambda = \{\lambda_i\}_{i \in Id}$, where $\lambda_i = (Ev_i, \rightarrow_i)$ is the local life-cycle of each agent i, such that $\rightarrow^* = (\bigcup_{i \in Id} \rightarrow_i)^*$ defines a partial order of causality on the set $Ev = \bigcup_{i \in Id} Ev_i$ of all events. Note that each event may be shared by several agents at communication points. Therefore, the condition that $\rightarrow^*$ defines a global ordering of the set of events amounts to requiring that communication does not introduce cycles among the different local causality orderings. A *global configuration* of λ is any set $\xi \subseteq Ev$ such that if $e' \in \xi$ and $e \rightarrow e'$ then $e \in \xi$. We denote the set of all global configurations of λ by Ξ. Clearly, every global configuration ξ contains a local configuration $\xi|_i = \xi \cap Ev_i$ for each $i \in Id$. Moreover, given $E \subseteq Ev$ finite, $(E\downarrow) = \{e' \in Ev \mid e' \rightarrow^* e \text{ for some } e \in E\}$ is a global configuration. Given a global configuration ξ and $E \not\subseteq \xi$, $(\xi + E)$ stands for $\xi \cup (E\downarrow)$. If $E = \{e\}$ we just write $(e\downarrow)$ and $(\xi + e)$. The following lemma shows that the configurations of any distributed life-cycle λ can be built by consecutively adding events.

Lemma 1. If $\xi, \xi' \in \Xi$ and $\xi \subsetneq \xi'$ then there exists $e \in \xi' \setminus \xi$ such that $\xi \cup \{e\} \in \Xi$.

An *interpretation structure* for $\mathcal{L}$ is a suitably labeled distributed life-cycle, that is, a triple $\mu = \langle \lambda, \sigma, \alpha \rangle$ where λ is a distributed life-cycle, $\sigma = \{\sigma_i \mid \Xi_i \rightarrow 2^{Prop_i}\}_{i \in Id}$ is an agent-indexed family of maps that associate a local state to each local configuration, and $\alpha = \{\alpha_i \mid Ev_i \rightarrow 2^{Act_i}\}_{i \in Id}$, with $\alpha_i(e) \neq \emptyset$ and finite for every $e \in Ev_i$, is an agent-indexed family of maps that associate a non-empty finite set of local actions to each local event. We can now define the satisfaction relation at a given global configuration ξ of μ

$$\mu, \xi \Vdash @_i[\varphi] \text{ if } \mu, \xi|_i \Vdash_i \varphi; \quad \mu, \xi \nVdash \bot; \quad \text{and} \quad \mu, \xi \Vdash \gamma \Rightarrow \delta \text{ if } \mu, \xi \nVdash \gamma \text{ or } \mu, \xi \Vdash \delta,$$

where local satisfaction is defined by

- $\mu, \xi_i \Vdash_i act$ if $\xi_i \neq \emptyset$ and $\alpha_i(last(\xi_i)) = act$;
- $\mu, \xi_i \Vdash_i p$ if $p \in \sigma_i(\xi_i)$;

- $\mu, \xi_i \not\Vdash_i \bot$;
- $\mu, \xi_i \Vdash_i \varphi \Rightarrow \psi$ if $\mu, \xi_i \not\Vdash_i \varphi$ or $\mu, \xi_i \Vdash_i \psi$;
- $\mu, \xi_i \Vdash_i \varphi \,\mathsf{U}\, \psi$ if there exists $\xi_i'' \in \Xi_i$ with $\xi_i \subsetneq \xi_i''$ such that $\mu, \xi_i'' \Vdash_i \psi$, and $\mu, \xi_i' \Vdash_i \varphi$ for every $\xi_i' \in \Xi_i$ with $\xi_i \subsetneq \xi_i' \subsetneq \xi_i''$;
- $\mu, \xi_i \Vdash_i \varphi \,\mathsf{S}\, \psi$ if there exists $\xi_i'' \in \Xi_i$ with $\xi_i'' \subsetneq \xi_i$ such that $\mu, \xi_i'' \Vdash_i \psi$, and $\mu, \xi_i' \Vdash_i \varphi$ for every $\xi_i' \in \Xi_i$ with $\xi_i'' \subsetneq \xi_i' \subsetneq \xi_i$; and
- $\mu, \xi_i \Vdash_i @_j[\varphi]$ if $\xi_i \neq \emptyset$, $last(\xi_i) \in Ev_j$ and $\mu, (last(\xi_i)\downarrow)|_j \Vdash_j \varphi$.

As usual, we say that μ is a model of $\Gamma \subseteq \mathcal{L}$ if $\mu, \xi \Vdash \gamma$ for every global configuration ξ of μ and every $\gamma \in \Gamma$.

We now present some useful lemmas about the logic.

Lemma 2 (Local properties). Let $\varphi \in \mathcal{L}_i$ be a local formula and μ an interpretation structure. If $\xi, \xi' \in \Xi$ are such that $\xi|_i = \xi'|_i$ then $\mu, \xi \Vdash @_i[\varphi]$ if and only if $\mu, \xi' \Vdash @_i[\varphi]$.

In the sequel, if $\varphi \in \mathcal{L}_i$ does not include @ subformulas then it is called a *private formula*. Furthermore, if φ is also free of the temporal operators U and S then it is called a *state formula*. The following is a useful lemma concerning private formulas.

Lemma 3 (Private properties). Let $\varphi \in \mathcal{L}_i$ be a private formula and μ, μ' interpretation structures with $\mu_i = \mu_i'$. If $\xi \in \Xi$ and $\xi' \in \Xi'$ are such that $\xi|_i = \xi'|_i$ then $\mu, \xi_i \Vdash @_i[\varphi]$ if and only if $\mu', \xi_i' \Vdash @_i[\varphi]$.

The logic allows us to state the following invariance rule for global properties.

Proposition 4 (Global invariance rule). Let $\gamma \in \mathcal{L}$ be a global formula, μ an interpretation structure and $\xi \in \Xi$ a global configuration. If both $\mu, \xi \Vdash \gamma$, and $\mu, \xi' \Vdash \gamma$ implies $\mu, \xi' \cup \{e\} \Vdash \gamma$ for every $\xi' \in \Xi$ and $e \in Ev \setminus \xi'$ such that $\xi \subseteq \xi'$ and $\xi' \cup \{e\} \in \Xi$, then $\mu, \xi' \Vdash \gamma$ for every $\xi' \in \Xi$ such that $\xi \subseteq \xi'$.

For local state properties, the invariance rule can be stated in the following more familiar way.

Proposition 5 (Local invariance rule). Let $\varphi \in \mathcal{L}_i$ be a local state formula, μ an interpretation structure and $\xi_i \in \Xi_i$ a local configuration. If both $\mu, \xi_i \Vdash_i \varphi$, and $\mu, \xi_i' \Vdash_i \varphi$ implies $\mu, \xi_i' \cup \{next(\xi_i')\} \Vdash \varphi$ for every $\xi_i' \in \Xi_i$ such that $\xi_i \subseteq \xi_i' \subsetneq Ev_i$, then $\mu, \xi_i' \Vdash_i \varphi$ for every $\xi_i' \in \Xi_i$ such that $\xi_i \subseteq \xi_i'$, or equivalently, $\mu, \xi_i \Vdash_i \mathsf{G}_\circ \varphi$.

Hence, μ is a model of $@_i[\varphi]$ if and only if μ is a model of both $@_i[* \Rightarrow \varphi]$ and $@_i[(\varphi \wedge \mathsf{X}\top) \Rightarrow \mathsf{X}\varphi]$, or equivalently, $@_i[(\varphi \wedge \mathsf{X}\, act) \Rightarrow \mathsf{X}\varphi]$ for every $act \in Act_i$.

3 The network model

We provide the specification of a generic open network where agents interact by exchanging messages through an insecure public channel. A *network signature* is a pair $\langle Pr, Nam\rangle$, where Pr is a finite set of principal identifiers $A, B, C, \ldots$, and Nam is a family $\{Nam_A\}_{A \in Pr}$ of pairwise disjoint finite sets of *names*, corresponding to the possible aliases used by each principal (the importance of aliases will become clearer below. We write A' to denote a name used by principal A. By abuse of notation, we also use $Nam = \bigcup_{A \in Pr} Nam_A$. Furthermore, we assume fixed two sets *Non* and *Key* of "numbers" that can be used as *nonces* and *keys*, respectively, and whose members we denote by N and K, possibly with annotations. In general, we assume that several kinds of keys can coexist and that each key K has its own inverse key K^{-1}. *Messages*, which we denote by M possibly with annotations, are built inductively from *atomic messages* (names and "numbers"), by concatenation ($\_;\_$), which we assume to be associative, and encryption under a key K ($\{\_\}_K$). The set *Msg* of messages is thus defined by

$$Msg ::= Nam \mid Non \mid Key \mid Msg; Msg \mid \{Msg\}_{Key}\,.$$

Given a network signature $\langle Pr, Nam\rangle$, we obtain a distributed signature by taking $Id = Pr \uplus \{Ch\}$, where *Ch* is the communication channel (used to model asynchronous communication), and letting the local alphabet of each agent (the principals and the channel) be defined as follows. The signature of a principal A requires actions Act_A and state propositions $Prop_A$, where Act_A includes

- $send(M, B')$ — sending of the message M to B';
- $rec(M)$ — reception of the message M;
- $spy(M)$ — eavesdropping of the message M; and
- $nonce(N)$ — generation of the fresh nonce N,

and $Prop_A$ includes

- $knows(M)$ — knowledge of the message M.

For the channel Ch we do not require any state propositions, i.e. $Prop_{Ch} = \emptyset$, whereas the actions Act_{Ch} include

- $in(M, A')$ — arrival at the channel of the message M addressed to A';
- $out(M, A')$ — delivery of M from the channel to principal A; and
- $leak$ — leaking of messages.

In the network model, principals can send and receive messages, at will, always through the channel. If the principal A sends a message to B', then the message synchronously arrives at the channel, where it is stored for future delivery to B. If delivery ever happens, it must be synchronized with the corresponding receive action of B. However, principal A can only send M to B' if A knows both B' and M. As usual, the knowledge of principals is not static. In addition to their initial knowledge, principals gain knowledge from the messages they receive and the nonces they generate. Principals may also spy on messages being leaked by the channel and learn their content. We do not allow principals to explicitly divert messages, but we also do not guarantee that messages delivered to the channel are ever received.

To ensure that principals do not learn messages in an ad hoc fashion, we specify that the *knows* propositions only hold where strictly necessary. We follow the idea underlying Paulson's inductive model [24], in accordance with the usual assumption of *perfect cryptography*. We restrict attention to those interpretation structures μ such that, for every principal A, the following condition holds for all messages M and global configurations $\xi \in \Xi$ such that $\xi|_A \neq \emptyset$:

(K) $\mu, \xi \Vdash_A knows(M)$ iff

$$M \in synth(analyz(\{M' \mid \mu, \xi \Vdash_A (\mathsf{Y}\, knows(M')) \vee rec(M') \vee spy(M') \vee nonce(M')\}))\,,$$

where *analyz* and *synth* are the functions representing how principals analyze or synthesize messages from a given set of messages (see, e.g., [24]).

To guarantee the freshness and uniqueness of the nonces generated by each principal, we further require the axioms

(N1) $@_A[nonce(N) \Rightarrow \mathsf{Y} \neg\, knows(M_N)]$,
(N2) $@_A[nonce(N)] \Rightarrow \bigwedge_{B \in Pr \setminus \{A\}} @_B[\neg\, knows(M_N)]$,

where M_N ranges over all the messages containing the nonce N. Together with **(K)**, **(N1)** and **(N2)** guarantee that every nonce is generated at most once, if at all, in each model, and always freshly (taking also into account the initial knowledge of all agents). The specification of the network model also comprises a number of axioms that characterize the behavior of the channel and of each principal $A \in Pr$:

(C1) $@_{Ch}[in(M, A') \Rightarrow \bigvee_{B \in Pr} @_B[send(M, A')]]$;
(C2) $@_{Ch}[out(M, A') \Rightarrow \mathsf{P}\ in(M, A')]]$; and
(C3) $@_{Ch}[out(M, A') \Rightarrow @_A[rec(M)]]$,
(P1) $@_A[send(M, B') \Rightarrow \mathsf{Y}(knows(M) \wedge knows(B'))]$;
(P2) $@_A[send(M, B') \Rightarrow @_{Ch}[in(M, B')]]$;
(P3) $@_A[rec(M) \Rightarrow @_{Ch}[\bigvee_{A' \in Nam_A} out(M, A')]]$;
(P4) $@_A[spy(M) \Rightarrow @_{Ch}[leak \wedge \mathsf{P} \bigvee_{B' \in Nam} in(M, B')]]$;
(P5) $@_A[\bigwedge_{B \in Pr \setminus \{A\}} \neg @_B[\top]]$; and
(P6) $@_A[nonce(N) \Rightarrow \neg\, @_{Ch}[\top]]$.

The channel axioms **(C1–C3)** are straightforward. They state that a message addressed to A' only arrives at the channel if it is sent to A' by some principal B; that the channel only delivers a message to A' if such a message for A' has previously arrived; and that if the channel delivers a message to A' then A receives it. The principal axioms are also simple. **(P1)** is a precondition for sending a message, stating that the sender must know both the message and the recipient's name beforehand. The next three formulas are interaction axioms. **(P2)** and **(P3)** state that the sending and receiving of messages, respectively, must be shared with the corresponding arrival and delivery actions of the channel. **(P4)** guarantees that a spied message must have arrived at the channel, addressed to some recipient. The two final axioms limit the possible amount of interaction: **(P5)** guarantees that principals never communicate directly (only through the channel), and **(P6)** states that nonce generating actions are not communication actions.

4 Protocol modeling and analysis

Protocols are usually informally described by short sequences of messages that are exchanged by principals in order to achieve particular security goals in open, hostile environments. We illustrate protocol modeling on top of our network by using a standard example: the (flawed) simplified Needham-Schroeder Public-Key Protocol NSPK [18], which we present as the following sequence of message exchange steps.

$$\begin{array}{lcll} a \to b & : & (n_1). & \{n_1; a\}_{K_b} \\ b \to a & : & (n_2). & \{n_1; n_2\}_{K_a} \\ a \to b & : & & \{n_2\}_{K_b} \end{array}$$

In this notation a and b are variables of sort name that denote each of the roles played in one execution of the protocol, and n_1 and n_2 are variables of sort nonce. The arrows represent communication, from sender to receiver. The parenthesized nonces prefixing the first and second messages exchanges signify that these nonces must be freshly generated before the subsequent message is sent. Moreover, it is assumed that an underlying "infrastructure" of *public* and *private* keys exists: K_a represents the public key of a, whose inverse key should be private, i.e. known by no one but the principal using that name. Although other possibilities could be easily added to the model, we refrain from doing so here, for simplicity, and assume that these are the only existing keys.

Formalizing a protocol like the above involves defining the sequences of actions (*send*, *rec*, and *nonce*) taken by honest agents executing the protocol. Namely, for each role, we formalize the actions taken and the order in which they must be taken. In the case of NSPK there are two roles: an initiator role *Init*, represented by a, and a responder role *Resp*, represented by b. Given distinct names A' and B', of principals A and B respectively, and nonces N_1 and N_2, the role instantiations should correspond to the execution, by principal A, of the sequence of actions $\mathrm{run}_A^{Init}(A', B', N_1, N_2)$:

$$\langle \mathit{nonce}(N_1).\mathit{send}(\{N_1; A'\}_{K_{B'}}, B').\mathit{rec}(\{N_1; N_2\}_{K_{A'}}).\mathit{send}(\{N_2\}_{K_{B'}}, B')\rangle\,,$$

and to the execution, by principal B, of the sequence $\mathrm{run}_B^{Resp}(A', B', N_1, N_2)$:

$$\langle \mathit{rec}(\{N_1; A'\}_{K_{B'}}).\mathit{nonce}(N_2).\mathit{send}(\{N_1; N_2\}_{K_{A'}}, A').\mathit{rec}(\{N_2\}_{K_{B'}})\rangle\,.$$

If $\mathrm{run}_A^i(\overline{M}) = \langle act_1 \dots act_n\rangle$ then we can consider the local formula $\mathrm{role}_A^i(\overline{M})$:

$$act_n \wedge \mathsf{P}(act_{n-1} \wedge \mathsf{P}(\dots \wedge \mathsf{P}\, act_1)\dots)\,.$$

For example, it should be clear that $\mu, \xi \Vdash @_A[\mathrm{role}_A^{Init}(A', B', N_1, N_2)]$ if and only if A has just completed at ξ the required sequence of actions.

4.1 Honesty

We take an external view of the system, supported by the *one-intruder* reduction reported in [7, 8], and consider a *protocol signature* to be a triple $\langle Hn, \{Z\}, Nam\rangle$ where Hn is the set of *honest* of principals and $Z \notin Hn$ is the *intruder*, and $\langle Hn \cup \{Z\}, Nam\rangle$ is a network signature such that every honest principal has exactly one name and never plays two distinct roles in the same protocol run. Without loss of generality, we assume that $Nam_A = \{A\}$ for every $A \in Hn$. We assume also that the private key of each honest principal is initially only known by that principal. This can be achieved by the axioms **(Key1)** and **(Key2)** below, where $A \in Hn$:

(Key1) $@_A[* \Rightarrow \mathit{knows}(K_A^{-1})]$; and

(Key2) $@_B[* \Rightarrow \neg\, \mathit{knows}(M)]$, for every $B \in Pr \setminus \{A\}$ and M containing K_A^{-1}.

Models of a protocol will be those network models where, furthermore, all honest principals follow the rules of the protocol. That is, for every $A \in Hn$, if the local life-cycle of A is $e_1 \rightarrow_A e_2 \rightarrow_A e_3 \rightarrow_A \ldots$, then the corresponding (possibly infinite) sequence of actions $\langle \alpha_A(e_1).\alpha_A(e_2).\alpha_A(e_3)\ldots\rangle$ must be an interleaving of prefixes of possible protocol runs, but using distinct fresh nonces in each of them. In the case of NSPK, this means that the life-cycle of an honest agent must be built by interleaving prefixes of sequences of the form $\mathrm{run}_A^{Init}(A, B', N_1, N_2)$ or $\mathrm{run}_A^{Resp}(B', A, N_1, N_2)$, such that no two initiator runs can have the same N_1, no two responder runs can have the same N_2, and the N_1 of any initiator run must be different from the N_2 of any responder run.

4.2 Security goals

The aim of protocol analysis is to prove (or disprove) the correctness of a protocol with respect to the security goals that it is supposed to achieve. For instance, *secrecy* of the critical data exchanged during an execution of the protocol among its participants is certainly a goal to be achieved. In addition, an honest principal running the protocol should be able to *authenticate* the identities of its protocol partners through the examination of the messages he receives. Below, we show how to formulate the required secrecy and authentication goals of protocols in the general case, illustrating them by means of the NSPK protocol.

As usual, we call an *attack* to the protocol, and specifically to a given security goal, any protocol model μ and configuration ξ for which the formula expressing the goal does not hold. Let us start with secrecy.

Secrecy We can formalize that the messages in a finite set S will remain a shared secret between the participants after a complete execution of the protocol with participants $A_1, \ldots, A_j$ by the formula secr_S:

$$\bigwedge_{i=1}^{j} @_{A_i}[\mathsf{P}_\circ\ \mathrm{role}^i_{A_i}(\overline{M})] \Rightarrow \bigwedge_{B \in Pr\setminus\{A_1,\ldots,A_j\}} \bigwedge_{M \in S} @_B[\neg\, \mathit{knows}(M)].$$

Of course, this property can only be expected to hold in particular situations. Assume that all the participants in a complete run of the protocol are honest. One should then expect that the "critical" nonces generated during that run will remain a secret shared only by the participating principals. Indeed, being honest, they will not reuse those nonces in further protocol runs. Using the logic, we can check the property $\mathrm{secr}_F(\overline{M})$ for the relevant set F of fresh nonces. In the case of NSPK, this would amount to requiring $\mathrm{secr}_{\{N_1,N_2\}}(A, B, N_1, N_2)$, with A and B both honest.

Authentication There are many possible shades of authentication (see, e.g., [19]). However, most authors agree that authentication should be expressed as some kind of correspondence property between the messages an agent receives in a protocol run and the messages that other participants of the same run are supposed to send. The typical authentication goal states that if an honest principal A completes his part of a run of a protocol in role i, with certain partners and data, then it must be the case that these partners have also been actively involved by sending to A the messages that he received. The property that A authenticates a partner B in role j at step q of the protocol can be defined in our logic by the formula $\mathrm{auth}^{i,j,q}_{A,B}(\overline{M})$, which is

$$@_A[\mathrm{role}^i_A(\overline{M})] \Rightarrow @_B[\mathsf{P}_\circ\ \mathit{send}(M, A)],$$

assuming that the protocol step q requires that B, in role j, sends message M to A, in role i. We should therefore require $\mathrm{auth}^{i,j,q}_{A,B}(\overline{M})$ to hold whenever step q is considered essential. In the case of NSPK, we could require for honest A acting as initiator, the authentication of the responder at step 2 using $\mathrm{auth}^{Init,Resp,2}_{A,B}(A, B', N_1, N_2)$, and for honest B acting as responder, the authentication of the initiator at step 3 using $\mathrm{auth}^{Resp,Init,3}_{B,A}(A', B, N_1, N_2)$. The latter fails in the man-in-the-middle attack to NSPK [18], as we explain below.

4.3 Analysis

To evaluate the cogency of our approach, we have analyzed the well-known Needham-Schroeder Public-Key Protocol (NSPK) and the protocol NSL, the corrected version given by Lowe to prevent the man-in-the-middle attack on NSPK [18]. We have applied our logic to these and other similar examples, and have thereby been able both to:

- find the usual man-in-the-middle attack to NSPK (which results from the failure of the proof of the mentioned authentication formula), and
- show that the authentication properties hold for NSL.

A detailed account of our analysis of these and other protocols and properties can be found in [7, 8].

5 Conclusion

We have been applying our logic also to investigate general metatheoretic properties of the underlying protocol models and model simplification techniques that may contribute to the sound design of effective protocol analysis tools. Such results also help simplify the underlying protocol model and thereby simplify the analysis of properties such as the ones considered in the previous section. Namely, in [7, 8], we prove a general lemma about *secret data* that is similar to the secrecy theorems of [13, 21]. We also obtain soundness and completeness results, with respect to typical security goals, for two model-simplification techniques: *one intruder is enough*, in the lines of [12], and the *predatory-intruder*, a bound on the behavior of the intruder that goes in the direction of the trace models used in practice, e.g. [24]. While these results, mutatis mutandis, have already been shown for other particular formalisms, our logic provides a means for proving them in a general and uniform way, within the same formalism, which opens the way for further general investigations. Our formalization has also allowed us to clarify aspects of these simplification properties that are often neglected or cannot be specified in the first place (e.g. concerning principals' identities and the way security properties are established). We have also begun applying our logic to other metatheoretical investigations, such as the development of appropriate partial-order techniques that may reduce the (potentially infinite) state-space exploration involved in model-checking protocol properties (cf. [3]). This is work in progress and the first results are promising.

Acknowledgments

This work was partially supported by FCT and EU FEDER via the Project FibLog POCTI/MAT/37239/2001 of CLC, and by the FET Open Project IST-2001-39252 and the BBW Project 02.0431, "AVISPA: Automated Validation of Internet Security Protocols and Applications".

Bibliography

[1] A. Armando and L. Compagna. Abstraction-driven SAT-based Analysis of Security Protocols. In *Proc. SAT 2003*, LNCS 2919. Springer-Verlag, 2003. Available at http://www.avispa-project.org.

[2] A. Armando, L. Compagna, and P. Ganty. SAT-based Model-Checking of Security Protocols using Planning Graph Analysis. In *Proc. FME'2003*, LNCS 2805. Springer-Verlag, 2003.

[3] D. Basin, S. Mödersheim, and L. Viganò. An On-The-Fly Model-Checker for Security Protocol Analysis. In E. Snekkenes and D. Gollmann, editors, *Proc. ESORICS'03*, LNCS 2808, pages 253–270. Springer-Verlag, 2003. Available at http://www.avispa-project.org.

[4] D. Basin, S. Mödersheim, and L. Viganò. Constraint Differentiation: A New Reduction Technique for Constraint-Based Analysis of Security Protocols. In V. Atluri and P. Liu, editors, *Proc. CCS'03*, pages 335–344. ACM Press, 2003. Available at http://www.avispa-project.org.

[5] C. Bodei, P. Degano, R. Focardi, and C. Priami. Primitives for authentication in process algebras. *Theoretical Computer Science*, 283(2), 2002.

[6] L. Bozga, Y. Lakhnech, and M. Perin. Pattern-based abstraction for verifying secrecy in protocols. In *Proc. TACAS 2003*, LNCS 2619. Springer-Verlag, 2003.

[7] C. Caleiro, L. Viganò, and D. Basin. Distributed Temporal Logic for Security Protocol Analysis. In preparation, 2004.

[8] C. Caleiro, L. Viganò, and D. Basin. Metareasoning about Security Protocols using Distributed Temporal Logic. To appear, 2004.

[9] I. Cervesato, C. Meadows, and P. F. Syverson. Dolev-Yao is no better than Machiavelli. In P. Degano, editor, *Proc. of WITS'00*, 8–9 July 2000.

[10] I. Cervesato and P. F. Syverson. The logic of authentication protocols. In *Foundations of Security Analysis and Design*, LNCS 2171, pages 63–136. Springer-Verlag, 2001.

[11] Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols. In E. Brinksma and K. G. Larsen, editors, *Proc. CAV'02*, LNCS 2404, pages 324–337. Springer-Verlag, 2002.

[12] H. Comon-Lundh and V. Cortier. Security properties: two agents are sufficient. In *Proc. ESOP'2003*, LNCS 2618, pages 99–113. Springer-Verlag, 2003.

[13] V. Cortier, J. Millen, and H. Rueß. Proving secrecy is easy enough. In *Proc. of CSFW'01*. IEEE Computer Society, 2001.

[14] B. Donovan, P. Norris, and G. Lowe. Analyzing a library of security protocols using Casper and FDR. In *Proc. FLOC'99 Workshop on Formal Methods and Security Protocols (FMSP'99)*, 1999.

[15] H.-D. Ehrich and C. Caleiro. Specifying communication in distributed information systems. *Acta Informatica*, 36:591–616, 2000.

[16] T. Genet and F. Klay. Rewriting for cryptographic protocol verification. In *Proc. CADE'00*, LNCS 1831, pages 271–290. Springer-Verlag, 2000.

[17] K. Lodaya, R. Parikh, R. Ramanujam, and P. Thiagarajan. A logical study of distributed transition systems. *Information and Computation*, 119(1):91–118, 1995.

[18] G. Lowe. Breaking and Fixing the Needham-Shroeder Public-Key Protocol Using FDR. In T. Margaria and B. Steffen, editors, *Proc. TACAS'96*, LNCS 1055, pages 147–166. Springer-Verlag, 1996.

[19] G. Lowe. A hierarchy of authentication specifications. In *Proc. CSFW'97*. IEEE Computer Society Press, 1997.

[20] G. Lowe. Casper: a Compiler for the Analysis of Security Protocols. *Journal of Computer Security*, 6(1):53–84, 1998.

[21] J. Millen and H. Rueß. Protocol-independent secrecy. In *2000 IEEE Symposium on Security and Privacy*. IEEE Computer Society, May 2000.

[22] J. Millen and V. Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proc. of CCS'01*, pages 166–175. ACM Press, 2001.

[23] F. Oehl, G. Cécé, O. Kouchnarenko, and D. Sinclair. Automatic approximation for the verification of cryptographic protocols. In *Proc. Conference on Formal Aspects of Security*, LNCS 2629. Springer-Verlag, 2003.

[24] L. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6:85–128, 1998.

[25] P. Ryan, S. Schneider, M. Goldsmith, G. Lowe, and B. Roscoe. *Modelling and Analysis of Security Protocols*. Addison Wesley, 2000.

[26] D. Song, S. Berezin, and A. Perrig. Athena: a novel approach to efficient automatic security protocol analysis. *Journal of Computer Security*, 9:47–74, 2001.

[27] G. Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Applications and Relationships to Other Models of Concurrency*, LNCS 255, pages 325–392. Springer-Verlag, 1987.

Combining linear orders with modalities for possible histories

Valentin Goranko [1] Alberto Zanardo [2]

[1] Department of Mathematics, Rand Afrikaans University, Johannesburg.
vfg@rau.ac.za.

[2] Department of Pure and Applied Mathematics, University of Padova.
azanardo@math.unipd.it.

Prooceedings of CombLog'04
Workshop on Combination of Logics:
Theory and Applications

1 Introduction

Two main areas of temporal logics are those of *linear time* and of *branching time*. In the former, time is viewed as a linear sequence of *moments* (see e.g. [5]), while, in the latter, time is pictured in a tree-like fashion: the past of any moment is linearly ordered, but there might be incomparable moments in its future (see e.g. [2, 4]). Linear orders, though, play a crucial role also in logics for branching time. Prior's Ockhamist and Peircean semantical rules for branching time [6], in fact, involve quantification over *histories* in tree-like structures, where histories are maximal linearly ordered sets of moments. Moreover, this quantification can be viewed as (and in Ockhamist logic is) the result of the application of a modal operator (see e.g. [8]). This means that the language and semantics for branching time can be obtained as the combination of languages and semantics for linear time with a modality for possible histories. In this paper, we study various degrees of combining linear time and modal operators and semantics, and we discuss the problem of transferring logical properties from linear to branching time.

1.1 Technical preliminaries

The propositional language for logics of linear time consists of the usual Boolean part plus the temporal operators P and F (with dual operators $H = \neg P \neg$ and $G = \neg F \neg$). The semantics of these operators is the usual Kripke semantics for modal logics: given any linear order $\mathcal{L} = \langle X, < \rangle$ and any evaluation V of the propositional variables in X, truth of Boolean combinations of formulae in the model $M = \langle \mathcal{L}, V \rangle$ is defined in the usual way, while, for tensed formulae, we set

$$\begin{array}{lcl} M, t \models F\theta & \Leftrightarrow & M, t' \models \theta\,, \text{ for some } t' > t \\ M, t \models P\theta & \Leftrightarrow & M, t' \models \theta\,, \text{ for some } t' < t \end{array} \tag{1}$$

Given any class C of linear orders, the temporal (P, F) logic of C will be denoted by $L^t(C)$ and the modal logic representing its future fragment by $L^m(C)$.

In the context of this paper, a tree is an irreflexive and transitive order $\mathrm{T} = \langle T, \prec \rangle$ with the *left-linearity* property: if $x \prec y$ and $z \prec y$, then either $x \prec z$, or $z \prec x$, or $x = z$. A *history* in the tree T is a subset of T linearly ordered by $\prec$, which is maximal for inclusion. The set of all histories in the tree T will be written as $\mathrm{H}(\mathrm{T})$ and the set of histories passing through t will be written as $\mathrm{H}_t(\mathrm{T})$.

Given any model $M = \langle \mathrm{T}, V \rangle$ based on the tree T, any history h in T and any moment $t \in h$, the truth of a PF-temporal formula θ in M at $\langle h, t \rangle$, denoted $M, h, t \models \theta$, is defined as truth of θ at t in the linear order $\langle h, \prec_{|h} \rangle$, with the evaluation $V' = V_{|h}$.

In a branching time context, the modality $\Diamond$ is read as 'for some possible history' passing through the current moment. Thus, we can set

$$M, h, t \models \Diamond\theta \quad \Leftrightarrow \quad M, h', t \models \theta\,, \text{ for some } h' \in \mathrm{H}_t(\mathrm{T}) \tag{2}$$

It is worth observing here that the truth of $\Diamond\theta$ at $\langle t, h \rangle$ in M does not depend on h and hence the right part of (2) is often written as $M, t \models \Diamond\theta$. Formulae of this kind are called *state formulae*. Formulae of the form $F\theta$, whose truth is history dependent, are called *path formulae*. As always, we write $\Box$ for the dual operator $\neg\Diamond\neg$.

Given a class of linear orders C we denote by $\mathrm{T}(C)$ the class of trees in which every history belongs to C. We say that $\mathrm{T}(C)$ is the class of trees *generated by* C. Note that $C \subseteq \mathrm{T}(C)$. Moreover, for a set $C = \{\langle X_i, <_i \rangle : i \in I\}$ of linear orders such that $\langle \cup_{i \in I} X_i, \cup_{i \in I} <_i \rangle$ is a tree T, it might happen that $\mathrm{T} \notin T(C)$: T might have *emerging* histories which do not belong to C.

The converse operation is that of passing from a class $\mathcal{C}$ of trees to the class $\mathrm{H}(\mathcal{C}) = \cup_{\mathrm{T} \in \mathcal{C}} \mathrm{H}(\mathrm{T})$. In this way, the expressions $\mathrm{H}(\mathrm{T})$ and $\mathrm{H}(\{\mathrm{T}\})$ are different names for the same set.

2 A unified approach to branching time semantics

Given a class of linear orders C we can define a hierarchy of modal and temporal logics over the class of trees $\mathrm{T}(C)$ by combining in different degree the temporal operators F, P with the modality $\Diamond$ - see [6], [3], [4], [1], and other works. Here we indicate the natural combinations of modalities and the resulting logic when these are added to propositional logic.

- $\Diamond F$: the ordinary modal logic $L^m_{\mathrm{Prior}}(\mathrm{T}(C))$ on $\mathrm{T}(C)$;
- $\Diamond F, \Diamond P$: the ordinary Priorean temporal logic $L^t_{\mathrm{Prior}}(\mathrm{T}(C))$ on $\mathrm{T}(C)$;
- $\{\Diamond, \Box\} \times \{F\}$: the future fragment of the Peircean logic $L^m_{\mathrm{Peirce}}(\mathrm{T}(C))$ over $\mathrm{T}(C)$;
- $\{\Diamond, \Box\} \times \{F, P\}$: the full Peircean logic $L^t_{\mathrm{Peirce}}(\mathrm{T}(C))$ over $\mathrm{T}(C)$;
- $\{\Diamond, F\}$: the future fragment $L^m_{\mathrm{Ockham}}(\mathrm{T}(C))$ of the Ockhamist logic over $\mathrm{T}(C)$;
- $\{\Diamond, F, P\}$: the full Ockhamist logic $L^t{}_{\mathrm{Ockham}}(\mathrm{T}(C))$ over $\mathrm{T}(C)$.

In Prior and Peircean logics tenses and modalities occur only as part of composed operators. Thus, we will adopt the following notation:

$$\begin{array}{llll} \mathbf{F} := \Box F & \mathbf{f} := \Diamond F & \mathbf{G} := \neg\mathbf{f}\neg & \mathbf{g} := \neg\mathbf{F}\neg \\ \mathbf{P} := \Box P & \mathbf{p} := \Diamond P & \mathbf{H} := \neg\mathbf{p}\neg & \mathbf{h} := \neg\mathbf{P}\neg \end{array}$$

According to (1) and (2), given any branching time model $M = \langle \mathrm{T}, V\rangle$ and any moment t in T, the truth conditions for $\mathbf{F}$ and $\mathbf{f}$ are

$$\begin{array}{lcl} M, t \models \mathbf{F}\theta & \Leftrightarrow & \forall h \in \mathrm{H}_t\, \exists t' \in h : t \prec t' \text{ and } M, t' \models \theta \\ M, t \models \mathbf{f}\theta & \Leftrightarrow & \exists h \in \mathrm{H}_t\, \exists t' \in h : t \prec t' \text{ and } M, t' \models \theta \end{array} \tag{1}$$

The truth conditions for $\mathbf{P}$ and $\mathbf{p}$ can be expressed similarly, by replacing $\prec$ with $\succ$. It must be observed, however, that the left-linearity property of trees implies that the interpretations of $\mathbf{P}$ and $\mathbf{p}$ coincide. These two operators actually agree with the linear time operator P. For this reason, in the sequel we will use only $\mathbf{P}$ and $\mathbf{H}$.

Priorean and Peircean validity (in symbol, $\vDash_{\mathrm{Prior}}$ and $\vDash_{\mathrm{Peirce}}$) are defined on the basis of the above truth conditions, by means of the obvious quantifications over moments and models. On Prior formulae θ, $\vDash_{\mathrm{Prior}}$ and $\vDash_{\mathrm{Peirce}}$ are trivially equivalent.

The Ockhamist logic operators are just the linear time operators P and F, plus the modality $\Diamond$, and their truth conditions are given by (1) and (2), which define the notion of Ockhamist validity ($\vDash_{\mathrm{Ockham}}$). It must be observed that, in general, the truth of Ockhamist formulae in a model is relative to pairs $\langle t, h\rangle$.

Note that for each of the Peircean and Ockhamist logic, and their future fragments, there are (at least) two different natural semantics: over *bundled trees* ([3], [8]), and over *complete bundled trees* which is equivalent to the one considered above. In the bundled tree semantics, the quantifications over H(T) is replaced by a quantification over an arbitrary subset (bundle) $\mathcal{B}$ of H(T) with the property that $\cup\mathcal{B} = T$.

The Ockhamist formula $\Box G\Diamond F\Box p \rightarrow \Diamond GF\Box p$ ([3]) is valid in all complete trees, but can be falsified in some bundled tree. On Peircean validity, instead, the two semantics coincide [4]. This does not mean, though, that any formula which is valid in a specific tree T is also valid in any bundled tree based on T. The formula $\mathbf{G}(p \rightarrow \mathbf{f}p) \wedge \mathbf{f}p \rightarrow \mathbf{gf}p$, for instance, is Peirce-valid on all complete binary ω-trees[1] but fails on any bundled ω-tree where exactly one history is removed and p is true precisely at all moments of that history.

3 Translations

The linear time PF-language (or its future fragment) can be embedded into the languages of Priorean, Peircean, and Ockhamist logics. In the case of Priorean logic, the obvious embedding is given by $F \mapsto \mathbf{f}$ and $P \mapsto \mathbf{P}$. Also for Ockhamist logic there is only one possible choice, namely, the identical embedding of the linear time operators.

The situation is more interesting with Peircean logic where we have many possible choices. In principle, each occurrence of F in a linear time formula θ can be translated either to $\mathbf{F}$ or to $\mathbf{f}$ and hence, if θ has n occurrences of F, then there are 2^n possible ways of translating it to a Peircean formula.

[1] A binary ω-tree is a tree in which every history is isomorphic to the set ω of natural numbers and every moment has exactly two immediate successors.

In general, arbitrary translations of linear time formulae to Peircean formulae do not preserve validity. The simplest example is the formula

$$Fp \wedge Fq \rightarrow F(p \wedge Fq) \vee F(q \wedge Fp) \vee F(p \wedge q) \tag{1}$$

which expresses that time is linear towards the future. Both constant translations, $F \mapsto \mathbf{F}$ and $F \mapsto \mathbf{f}$, transform this formula into non-valid Peircean formulas. On the other hand, linearity towards the future can also be expressed by the formula

$$Fp \rightarrow G(p \vee Fp \vee Pp) \tag{2}$$

and the translation $\mathbf{F}p \rightarrow \mathbf{G}(p \vee \mathbf{F}p \vee \mathbf{P}p)$ of this formula is Peirce valid, as well as the translation $\mathbf{f}p \rightarrow \mathbf{g}(p \vee \mathbf{f}p \vee \mathbf{P}p)$.

Thus, a natural question arises to determine the syntactic conditions under which a translation preserves validity, and, in general, the semantical behaviour of translated formulas. Two limit translations can be considered, for instance. The *weakest translation* τ^w (of linear time language to Peircean language) replaces every *positive* occurrence of F by $\mathbf{f}$ and every *negative* occurrence by $\mathbf{F}$. The *strongest translation* τ^s works the other way around.

Every point-wise valuation V (assigning a set of moments to every propositional variable) over a tree T, restricted to any $h \in \mathrm{H(T)}$ determines a valuation V_h on that linear order. Thus, for any Prior formula θ, it makes sense to set, as an auxiliary notion,

$$\mathrm{T}, V, t \vDash_{\mathrm{Peirce}} \Diamond\theta \quad \text{iff} \quad \exists h \in \mathrm{H}_t(\mathrm{T}) : h, V_h, t \vDash_{\mathrm{Prior}} \theta$$

and similarly for $\mathrm{T}, V, t \vDash_{\mathrm{Peirce}} \Box\theta$. On the basis of this definition, we have that, for every linear time formula θ,

$$\vDash_{\mathrm{Peirce}} \Diamond\theta \rightarrow \tau^w(\theta) \quad \text{and} \quad \vDash_{\mathrm{Peirce}} \tau^s(\theta) \rightarrow \Box\,\theta \tag{3}$$

Indeed, note first that after driving all negations in $\tau^w(\theta)$ inside the temporal operators, only operators $\mathbf{f}$ and $\mathbf{g}$ will occur in $\tau^w(\theta)$, i.e. every temporal operator will be prefixed by a $\Diamond$. Then, it suffices to use the validities $\vDash \Diamond\alpha \vee \Diamond\beta \leftrightarrow \Diamond(\alpha \vee \beta)$, $\vDash \Diamond(\alpha \wedge \beta) \rightarrow \Diamond\alpha \wedge \Diamond\beta$, $\vDash \Diamond F\alpha \rightarrow \Diamond F\Diamond\alpha$, and $\vDash \Diamond G\alpha \rightarrow \Diamond G\Diamond\alpha$ to pull all $\Diamond's$ in front of the formula and eventually show that $\vDash \Diamond\theta \rightarrow \tau^w(\theta)$. Likewise, but dually, for $\tau^s(\theta)$.

A first consequence of (3) is that, for any Priorean formula θ, tree T, valuation V on T, and $t \in T$:

$$h, V_h, t \vDash_{\mathrm{Prior}} \theta \text{ for some } h \in \mathrm{H(T)} \quad \Rightarrow \quad \mathrm{T}, V, t \vDash_{\mathrm{Peirce}} \tau^w(\theta) \tag{4}$$

and, taking into account that $\neg\tau^w(\theta) \equiv \tau^s(\neg\theta)$,

$$\mathrm{T}, V, t \vDash_{\mathrm{Peirce}} \tau^s(\theta) \quad \Rightarrow \quad h, V_h, t \vDash_{\mathrm{Prior}} \theta \text{ for every } h \in \mathrm{H(T)} \tag{5}$$

These results imply in turn that, for any Priorean formula θ and any class of linear orders C,

$$C \vDash_{\mathrm{Prior}} \theta \quad \text{iff} \quad \mathrm{T}(C) \vDash_{\mathrm{Peirce}} \tau^w(\theta) \tag{6}$$

Proof: The implication from right to left is immediate, because $C \subset \mathrm{T}(C)$ and both semantics coincide on linear orders. Now, suppose $\mathrm{T}(C) \nvDash_{\mathrm{Peirce}} \tau^w(\theta)$, i.e. $\mathrm{T}, V, t \vDash_{\mathrm{Peirce}} \neg\tau^w(\theta)$, hence $\mathrm{T}, V, t \vDash_{\mathrm{Peirce}} \tau^s(\neg\theta)$ for some $\mathrm{T} \in \mathrm{T}(C)$. Then (5) implies $h, V_h, t \vDash_{\mathrm{Prior}} \neg\theta$ for every $h \in \mathrm{H(T)}$, but $\mathrm{H(T)} \subseteq C$. ◁

As we see from the proof of (6), the claim can be strengthened. In this paper we will establish a syntactic characterization of the translations for which that claim still holds.

In general, the set of translations of a given Prior formula θ can be endowed with a lattice structure by letting $\tau \leq \tau'$ whenever $\tau(\theta)$ can be obtained from $\tau'(\theta)$ by replacing some (possibly no) positive occurrence of $\mathbf{f}$ by $\mathbf{F}$ and some (possibly no) negative occurrence of $\mathbf{F}$ by $\mathbf{f}$. In this way τ^w and τ^s turn out to be the top and the bottom or the lattice, respectively. On the basis of the Peirce validity $\mathbf{F}\alpha \rightarrow \mathbf{f}\alpha$ it can be proved that

$$\tau \leq \tau' \quad \Rightarrow \quad \vDash_{\mathrm{Peirce}} \tau(\theta) \rightarrow \tau'(\theta)$$

4 Transfer of properties

In this paper we are investigating transfer of logical properties, such as definability, axiomatizations and decidability between various linear time logics $L^t(C)$ (or $L^m(C)$) and their branching time counterparts $L^t(\mathrm{T}(C))$ with Priorean, or Peircean, or Ockhamist semantics.

4.1 Transfer of definability

Which modally definable properties of linear orders in a given class C transfers to properties of all histories in $\mathrm{T}(C)$? More specifically, given a property of linear orders and a formula θ which defines that property in the class C, we wonder whether some translation of θ defines the same property for all histories in $\mathrm{T}(C)$.

The answer is trivial when Ockhamist logic is considered because Ockhamist tense operators are linear time operators and hence any definable property is transferred to a property which is definable in tree by means of the same formula.

Passing from Peircean definability to linear time definability is trivial as well. If the Peircean formula ϕ defines a given property of histories in some $\mathrm{T}(C)$, then, for any translation τ, $\tau^{-1}(\phi)$ defines that property in C. In fact, the elements of C are particular trees in $\mathrm{T}(C)$ and, on linear orders, the interpretations of both $\mathbf{F}$ and $\mathbf{f}$ coincide with the interpretation of F.

General transferability results from linear time logics to Peircean logics have not been established yet. As a further consequence of (3), we have that, if θ defines the property $\mathcal{P}$ in the class of linear orders, then, for every tree T:

(1) $\mathrm{T}, t \vDash_{\mathrm{Peirce}} \tau^w(\theta)$ whenever a history passing through t has $\mathcal{P}$,

and

(2) $\mathrm{T}, t \vDash_{\mathrm{Peirce}} \tau^s(\theta)$ implies that every history passing through t has $\mathcal{P}$.

Here are some examples of definable properties of linear orders which are also definable in trees by Peircean formulas.

Density	$\mathbf{f}p \to \mathbf{ff}p$ (or $\mathbf{F}p \to \mathbf{FF}p$)
Discreteness	$(\mathbf{f}\top \to ((p \wedge \mathbf{H}p) \to \mathbf{FH}p)) \wedge$ $(\mathbf{P}\top \to ((p \wedge \mathbf{g}p)) \to \mathbf{Pg}p)$
Dedekind Continuity	$\mathbf{FG}\neg p \wedge \mathbf{F}p \to \mathbf{F}(\mathbf{g}\neg p \wedge \mathbf{Hf}p)$
Isomorphism to $\mathbb{Z}$	$\mathbf{H}p \to \mathbf{P}p \wedge \mathbf{G}p \to \mathbf{F}p \wedge$ $\mathbf{H}(\mathbf{H}p \to p) \to (\mathbf{PH}p \to \mathbf{H}p) \wedge$ $\mathbf{G}(\mathbf{G}p \to p) \to (\mathbf{FG}p \to \mathbf{G}p)$

4.2 Transfer of satisfiability/validity, axiomatizations, decidability

How do satisfiability and validity transfer between linear time logics and their branching time counterparts? Accordingly, when are axiomatizations and decidability results preserved in passing between these?

With Ockhamist logic, for instance, the translation τ is the identical embedding and Ockhamist PF-validity is linear time validity. Thus, any axiomatization of $L^t(C)$ is also an axiomatization of the PF fragment of $L^t_{\mathrm{Ockham}}(\mathrm{T}(C))$. Decidability transfers similarly.

The situation is quite different, though, if we adopt a different definition of $\mathrm{T}(C)$, and we let this class be the set of bundled trees $\langle \mathrm{T}, \mathcal{B} \rangle$ in which $\mathcal{B} \subseteq C$. This new perspective is actually suggested by the *Kamp frames* considered in [7]. With this new definition of $\mathrm{T}(C)$, emerging histories might appear and, at this stage, no proof of the preservation results considered above seems to be available.

5 Concluding remarks

This paper aims at systematic investigation of the logical aspects and virtues of combining linear orders as semantics for modal and temporal logics, with modalities for possible histories, resulting into a variety of branching time logics.

Bibliography

[1] K. Bowen and D. de Jongh. Some complete logics for branched time, Part 1: Well-founded time, forward looking operators. ILLC report, 1986.

[2] J. Burgess. The unreal future. *Theoria*, 44:157–179, 1978.

[3] J. Burgess. Logic and time. *J. of Symbolic Logic*, 44:556–582, 1979.

[4] J. Burgess. Decidability for branching time. *Studia Logica*, 39:203–218, 1980.

[5] R. Goldblatt. *Logic of Time and Computation.* Lecture Notes. CSLI, 1987.

[6] A. Prior. *Past, Present and Future.* Clarendon, Oxford, 1967.

[7] R. Thomason. Combination of tense and modality. In D. Gabbay and F. Guenthner, editors, *The Handbook of Philosophical Logic*, volume II, chapter 3, pages 135–165. Reidel, Dordrecht, 1984.

[8] A. Zanardo. Branching-time logic with quantification over branches: the point of view of modal logic. *J. of Symbolic Logic*, 61(1):1–39, 1996.