A Pointfree approach to Constructive Analysis in Type Theory

Jan Cederquist

Abstract

The first paper in this thesis presents a machine checked formalisation, in Martin-Löf's type theory, of pointfree topology with applications to domain theory. In the other papers pointfree topology is used in an approach to constructive analysis. The continuum is defined as a formal space from a base of rational intervals. Then the closed rational interval $[a, b]$ is defined as a formal space, in terms of the continuum, and the Heine-Borel covering theorem is proved constructively. The basic definitions for a pointfree approach to functional analysis are given in such a way that the linear functionals from a seminormed linear space to the reals are points of a particular formal space, and in this setting the Alaoglu and the Hahn-Banach theorems are proved in an entirely constructive way. The proofs have been carried out in intensional Martin-Löf type theory with one universe and finitary inductive definitions, and the proofs have also been mechanically checked in an implementation of that system.

Acknowledgements

First I want to thank my supervisor Jan Smith. A few years ago he suggested to me that I start studying pointfree topology as an approach to constructive mathematics. Since then he has repeatedly encouraged me to continue investigating this area. The suggestions and the response I have got from him during our discussions have been valuable for me. But most importantly, he has always shown enthusiasm in my work.

Two more people have influenced my work considerably. Many ideas that I have used are due to Thierry Coquand. Thierry also invented the logical framework Half that was suitable for my kind of formalisation. The last two years I have also had an inspiring collaboration with Sara Negri, one that I hope to continue.

Dan Synek is worthy a great praise for implementing a type-checker and an `emacs`-interface to Half. He made the system fast enough that my formalisations could be carried through.

I am grateful to Giovanni Sambin for generously inviting me to Padova. Giovanni has developed the particular notion of formal space that I have been using. His papers and lectures during his visits in Gothenburg have also contributed to my work.

On the invitation of Jan von Plato, I have paid two very pleasant visits to the Department of Philosophy in Helsinki. Both times I had interesting discussions with Jan and his colleagues.

The programming methodology group in Gothenburg has offered a creative atmosphere and I have had fruitful discussions with many of its members. I would in particular like to thank Michael Hedberg, Henrik Persson, Mary Sheeran and Martin Weichert for interesting discussions and useful comments.

Introduction

The papers in this thesis all have *pointfree topology* in common. Some of them also describe machine assisted formalisations in *type theory*. I will here briefly describe what the papers contain. But in order to make this introduction self-contained, I first say a few words about type theory and pointfree topology, followed by brief descriptions of the two proof checkers, ALF and Half, that have been used and the specific type theory that each proof checker is based on. In [Mag95] there is a detailed description of ALF, and Half is described in more detail in the third paper in this thesis.

Type theory

Martin-Löf's type theory [Mar72, NPS90] is a typed functional programming language with dependent types. But in type theory, unlike the usual functional programming languages, there is also the possibility of developing proofs. The aim of type theory was originally to serve as a foundation for constructive mathematics. However, because of the close relation between constructive proofs and computations, it may also be suitable as a foundation for ordinary programming.

In constructive mathematics the notion of function is primitive and a function from a set A to a set B is a method that, when given an element in A produces an element in B. So functions in constructive mathematics are computable and can be seen as programs. To prove a proposition constructively means to have a method of proving it. For instance to prove $(\forall x \in A)(\exists y \in B)(P(x, y))$ constructively means to give a function f that when applied to an element a in A gives an element b in B such $P(a, b)$ holds. For a presentation of the ideas of constructive mathematics we refer to [Bis67, Dum77, TvD88]. A constructive proof can thus be seen as a program (cf. [Con82, Bis70, Mar82, Moh86, NS84]), and in type theory execution of the program corresponds to normalisation of the proof.

The basic idea behind using type theory for developing proofs and programs is the Curry-Howard isomorphism [How80], where propositions (specifications) are identified with types and proofs of a proposition (programs satisfying a specification) are identified with objects of the corresponding type. Proof checking is then the same as type checking.

In type theory we have the possibility of introducing new types, which makes type theory into an open theory, and a type is formed by prescribing what has to be done in order to construct an object of that type. There are basically two ways of introducing new types in Martin-Löf's type theory: by inductive definitions and (dependent) function types. Proofs in Martin-Löf's type theory are represented by proof objects. They are formed by natural deduction, which reflects the ways the types are introduced.

Expressions in type theory "live" at different levels. For instance, we distinguish between sets and types; if we denote the type of sets by **Set** then **Set** is a proper type. Martin-Löf's type theory is *predicative*. For instance, when defining a set (or type) inductively, we do not quantify over objects of the set (type) we are defining and we do not compress one level into a lower level. This means, in particular that, given a set $X \in \mathbf{Set}$, the power set $\mathcal{P}(X)$ cannot, in general, be formed as an object of **Set**.

The type theory used here is called *intensional*; by this we mean that the equality used is the definitional equality. Intensional type theory should be compared to *extensional*, which uses a weaker propositional equality. In Martin-Löf's intensional type theory, equality and type checking are decidable; whereas extensionality leads to undecidability.

Formal spaces

In the usual (set-theoretic) topology we start from a space of points. Given a set X, a topology ΩX on the space X is a family of subsets of X which is closed under finite intersection and arbitrary union. The subsets in ΩX are called open sets.

The intuition behind pointfree topology is that points are abstractions: a point is an ideal object consisting of a non-contradictory collection of pieces of information. So the pointfree approach is sort of reversal of the usual set-theoretical approach: the notion of open set (or rather just open) is taken as primitive and what interests us is the algebraic structure (*frame*) that a topology forms.

A frame is a lattice $(A, \leq, \wedge, \bigvee)$, with arbitrary join ($\bigvee$), in which meet ($\wedge$) distributes over join:

$$a \wedge \bigvee S = \bigvee \{a \vee b : b \in S\}.$$

Points are then defined as a suitable collections of opens (completely prime filters). Frames can thus be seen as "generalised" topological spaces. These spaces may fail to have points (for non-trivial examples see [FG82, JT84]).

For the early development of this area we refer to [Joh82, Joh83], where Johnstone gives a comprehensive bibliography over the development of abstract algebra and pointfree topology. Recently this topic has appeared in locale theory [Joh82], it has also been used in applications to domain theory [Sco82, Mar83, Vic89].

Below we use the term *locale* interchangeably with frame. In the literature (see for instance [Joh82, Vic89]), the word locale is often used when there is explicit mention of the points. Categorically speaking, in the category of frames the morphisms are the *frame homomorphisms* (functions preserving arbitrary join and finite meet). The category of locales is the opposite category. Continuous maps between locales are determined by the corresponding frame homomorphisms, except that they go in the opposite directions. The morphisms in the category of locales thus correspond exactly to the continuous maps. The word *space* is also often used when referring to the points of a frame (locale).

A set-theoretic topology ΩX can always be presented using one of its bases. So a more "basic" way to present a general topological space is to describe the structure of its base. Moreover, since the definition of frame above contains a possibly infinitary join, it is more convenient to consider, for instance, semilattices for the development of pointfree topology. Semilattices are purely algebraic they can be formulated as commutative and idempotent monoids. When using predicative type theory to formalise topology it is also more general to start from a base, since the neighbourhoods may form a set (in type theory) whereas the opens do not.

A problem, that was solved by Johnstone, is whether frames in general can be presented by semilattices. For finitary algebraic theories presentations always present algebras. (For an overview of the method of presenting algebras from a set of *generators* and relations see [Man76] or [Vic89].) This is not always the case for infinitary algebraic theories (see [Joh82]). For frames, however, it is the case, as shown in [Joh82]. In the "coverage theorem" Johnstone gives an explicit description of a frame being presented from a set of generators and relations. In this way, he also gives an explicit description of arbitrary frame coproduct.

Fourman and Grayson [FG82] introduced the name *formal space*, for models of a propositional theory, to be the locale presented by the theory. They gave an effective presentation of topologies by generating topologies from preorders with conditional meet and closing an entailment relation under the axioms for a Grothendieck topology. A formalisation, in Martin-Löf's type theory, of this method of generating topologies is however not immediate.

Martin-Löf and Sambin then introduced *formal topologies* [Sam87] as a constructive approach to (pointfree) topology, in the tradition of Johnstone's coverages and, Fourman and Grayson's formal spaces, but using a constructive set theory based on Martin-Löf's type theory.

A formal topology is a structure $(S, \cdot, \lhd)$, where S is a set (base) of neighbourhoods such that $(S, \cdot)$ forms a semilattice (or alternatively a commutative monoid). The operation $\cdot$ corresponds to finite meet and instead of join we have the cover relation $\lhd$ between elements and subsets of S satisfying the following rules

- if $a \in U$ then $a \lhd U$,
- if $a \in U$ and $U \lhd V$ then $a \lhd V$, where $U \lhd V$ means that $x \lhd V$ for all $x \in U$,
- if $a \in U$ then $a \cdot b \lhd U$,
- if $a \in U$ and $a \lhd V$ then $a \lhd U \cdot V$, where $U \cdot V$ is the set of all $x \cdot y$ such that $x \in U$ and $y \in V$.

In terms of spaces as sets of points, the monoid operation can be understood as intersection and $a \lhd U$ means $a \subseteq \bigcup U$.

In the definition of formal topology, given in [Sam87], the base monoid has a unit and there is a positivity predicate for neighbourhoods (the intuition of a positive neighbourhood is that it is inhabited). In the papers in this thesis we sometimes drop these requirements. For formal topologies, as for frames, there are notions of point and morphism. Moreover, each formal topology represents a frame and each frame can be represented by a formal topology.

Generation of formal spaces have also been studied by several authors other than those already mentioned, see for example [Dra88, Gra83, Sig90].

In this thesis we use formal topology as an approach to pointfree topology and as a step towards a complete formalisation in Martin-Löf's type theory. Pointfree topology is used here because of constructiveness; to formulate and prove constructively theorems that in their usual set-theoretic formulation are not constructively true. In a pointfree formulation of a theorem, properties of points are replaced by the corresponding properties of their neighbourhoods (finite approximations) and sometimes constructions needing the axiom of choice or non-constructive arguments in terms of points get a constructive proof when stated in a pointfree way.

An example of such a theorem is Tychonoff's theorem, which says that a product of compact spaces is compact. This example is perhaps particularly interesting since, in the usual formulation, Tychonoff's theorem is equivalent to the axiom of choice (see [Kel50]). Johnstone used his presentation of frames to give a localic proof of Tychonoff's theorem without the axiom of choice. Other localic proofs of Tychonoff's theorem have also been given and, because of the similarities to the framework in which the proofs in this thesis are developed, we mention two of them here. Coquand presents in [Coq92b] an intuitionistic proof (which besides being constructive only uses inductive definitions) and, following Coquand's idea, Negri and Valentini [NV97] gave a proof in the framework of formal topologies.

Johnstone gives a few more examples of constructive proofs of theorems formulated in a pointfree way in [Joh83], and for further examples see [Coq92a, Coq95].

Implementations of type theory

ALF (Another Logical Framework)

ALF [Mag95] is an implementation of Martin-Löf's monomorphic type theory [NPS90] extended with pattern matching [Coq92c]. In the type theory used in ALF there are two basic levels: sets and types. Sets are formed by induction. The types consist of the type **Set** (whose objects are the sets), the type of elements of a set (in **Set**) and function types. Objects of a type are formed from constants and variables using application and abstraction (as in ordinary typed λ-calculus).

There are three ways of defining constants:

1. Inductive definitions of sets and families of sets, which consist of a formation rule and introduction rules prescribing how the canonical elements are formed.
2. Explicit definitions, which simply are abbreviations of well typed expressions.
3. Implicit definitions, which offer the possibility of defining functions using pattern matching. These may also be recursive.

Judgements are made relative to a *context*

$$a : A\ [\Gamma],$$

where a is an expression of type A and Γ a context. A context is a dependent list of declarations

$$[\,] : \mathit{Context} \qquad\qquad \frac{\Gamma : \mathit{Context} \qquad \alpha : \mathit{type}\ [\Gamma]}{[\Gamma; x : \alpha] : \mathit{Context}}$$

where x does not occur free in Γ and $[\Gamma; x : \alpha]$ is the extension of Γ with the clause $x : \alpha$.

Using a context we can represent an abstract algebraic structure. This is used in the first paper in this thesis. Betarte also gives a more detailed discussions about this approach in [Bet93]. A concrete structure can then be proved to be an instance of the abstract structure using a *substitution*, that is, an assignment of objects of appropriate types to the variables in the context. Substitutions are introduced by the rules

$$\{\,\} : [\,]\ [\Delta] \qquad\qquad \frac{\gamma : \Gamma\ [\Delta] \qquad \alpha : \mathit{type}\ [\Gamma] \qquad a : \alpha\gamma\ [\Delta]}{\{\gamma; x := a\} : [\Gamma; x : \alpha]\ [\Delta]}$$

where $\{\,\}$ is the empty substitution, $\gamma : \Gamma\ [\Delta]$ means that the substitution γ *fits* the context Γ in the context Δ, $\alpha\gamma$ is the substitution γ applied to the expression α and $\{\gamma; x := a\}$ is the extension of the substitution γ with the assignment $x := a$. Using a substitution γ fitting the context Γ, propositions stated relative to Γ can be instantiated:

$$\frac{a : A\ [\Gamma] \qquad \gamma : \Gamma\ [\Delta]}{a\gamma : A\gamma\ [\Delta]}$$

Tasistro explains substitutions in detail in [Tas93].

Half

The Half system, developed by Thierry Coquand, is a successor to ALF. It is a logical framework based on Martin-Löf's polymorphic type theory with one universe [Mar72], extended by a *theory* mechanism (similar to the theory mechanism in PVS [OSR93]) and *let-expressions* (see [Bar92, Bru91, Coq96]).

The system has three levels; **Set**, **Type** and **Kind**. **Set** is an element and a subset of **Type**. Elements can be formed in both **Set** and **Type**; both **Set** and **Type** are closed under function types (Π-types) and disjoint union (Σ-types) and allow recursive definitions. There is also a type **Theory** for theories. **Kind** consists of the types **Set**, **Type** and **Theory**, and function types.

The recursive definitions in Half are linear inductive definitions, that is dependencies between the parameters cannot be introduced in a recursive definition. It turned out that pattern matching together with non-linear inductive definitions is a non-conservative extension of Martin-Löf's type theory (see [Hof93]).

Theories are used to collect definitions and lemmas that logically belong together. They are often used together with the function type. By defining functions giving theories as result, a notion of parametrised theory is obtained.

Compared to ALF, Half has some features that make proof development easier and the resulting proofs more readable. The presence of both **Set** and **Type**, where **Set** corresponds to a universe, allows more abstract reasoning than is possible in ALF. For instance, Half allows type abbreviations. Furthermore, Σ-types and theories for grouping definitions and proofs simplify the structure of an implementation. Then, for local lemmas and abbreviations, Half have *let-expressions*.

There is a more detailed description of Half in the third paper below.

The Papers in the thesis

The thesis consists of the following papers:

1. J. Cederquist. *A machine assisted formalization of pointfree topology in type theory*, Chalmers University of Technology and University of Göteborg, Sweden, 1994.

2. J. Cederquist, S. Negri. *A constructive proof of the Heine-Borel covering theorem for formal reals*, In S. Berardi and C. Coppo eds., "Types for Proofs and Programs", Lecture Notes in Computer Science 1158, pp. 62–75, 1996.

3. J. Cederquist, *An implementation of the Heine-Borel covering theorem in type theory*, Chalmers University of Technology and University of Göteborg, Sweden, 1997.

4. J. Cederquist, T. Coquand, S. Negri *The Hahn-Banach Theorem in Type Theory*, To be published in the proceedings of Twenty five years of Constructive Type Theory, G. Sambin and J. Smith eds., Oxford University Press, 1997.

5. J. Cederquist, *A Machine Assisted Proof of the Hahn-Banach Theorem*, Chalmers University of Technology and University of Göteborg, Sweden, 1997.

The first paper is basically my licentiate thesis [Ced94] (the paper was changed slightly due to minor errors found after the printing and comments during the defence). We describe a formalisation of pointfree topology in Martin-Löf's type theory, using ALF. The work follows closely parts of the work by Sambin in [Sam87], and by Sambin, Valentini and Virgili in [SVV96].

A general formal topology is defined as a context `TOP`. Here we use a different definition of formal topology from that presented in [Sam87]. Ours is shorter and easier to handle as a context. To show that a concrete structure is a formal topology, the notion of substitution is used. By substitution, properties proved for the general topology `TOP` then also hold for all the instances.

Relative to the assumptions in `TOP`, *meet* and *join*, for subsets of the base, are defined and we prove that the subsets form a frame.

The formal points form in general a proper type and can therefore not be defined as a set in ALF, instead properties about points are proved by giving the point-rules as parameters to the proofs.

A Scott topology `SCOTTTOP` is defined by extending `TOP` with some extra rules. Then we define predicates needed in order to state when a space of points forms a Scott domain and we prove that the points of a Scott topology form a Scott domain.

The main problem with this formalisation was the lack of a universe. It would, for instance, have been desirable to form a type of formal topologies and also, given a topology, to form the type of its points. This was not possible, so to define abstract formal topologies we instead used contexts. But note that the context `TOP` is one arbitrary formal topology and not a template for formal topologies.

This formalisation of pointfree topology was used by Persson [Per96]. Following a proof by Sambin [Sam95], Persson developed a machine assisted, constructive completeness proof for intuitionistic predicate logic, using models based on formal topology.

In the second paper the continuum is defined as a formal topology from a base of rational intervals, using only finitary inductive definitions. Then a localic version of the Heine-Borel covering theorem is constructively proved. Given two rational numbers a and b we define the closed interval $[a, b]$ as a formal space, i.e. a space whose points are the formal real numbers between a and b (the reals corresponding to a and b included). Then we show that this space is compact.

We also give a proof that the formal real numbers are in a 1-1 correspondence to the real numbers as Cauchy sequences à la Bishop [Bis67], which is interesting since Bishop's real numbers form a set in type theory. The continuum as a formal topology is further explored by Negri and Soravia in [NS96]; the formal reals are here also compared to axiomatisations of the reals as Dedekind cuts and Martin-Löf's maximal approximations [Mar70].

In the third paper we describe an implementation in type theory, using Half, of the proof of the Heine-Borel covering theorem above.

The notion of formal space is here defined as a Σ-type $space(A, =, \cdot, \lhd)$. An element of $space(A, =, \cdot, \lhd)$ is thus a structure containing proofs that the base set A with the equality relation $=$, the operation $\cdot$ and the cover relation $\lhd$ satisfies the rules of formal topologies. Properties of a general formal space are then collected in a theory, *theory_space*, parametrised over a set A, an equality $=$, a binary operation $\cdot$, a relation $\lhd$ between elements and subsets of A, and an element of $space(A, =, \cdot, \lhd)$. In this theory we also define what it means for this formal space to be compact.

The continuum is defined and proved to be a formal topology, i.e. we form an element $\mathcal{R}$ of type $space(Q \times Q, =_{Q \times Q}, \cdot_{\mathcal{R}}, \lhd_{\mathcal{R}})$, where Q is the rational numbers, $=_{Q\times Q}$ is the equality on intervals, $\cdot_{\mathcal{R}}$ is the particular dot-operation on intervals and $\lhd_{\mathcal{R}}$ is the particular cover relation used on rational intervals. Having $\mathcal{R}$ we also get access to the definitions and lemmas in *theory_space*.

For the formal space $[a, b]$ we use the same base, equality and dot-operation. The cover $\lhd_{[a,b]}$ is defined in terms of $\lhd_{\mathcal{R}}$ and is easily proved to be a cover relation. $[a, b]$ is thus a formal topology and we get access to the definition of compactness in *theory_space*.

The rational numbers are defined as an object of an abstract data type, using a sigma set, as a general linear ordering. In fact the continuum, the definition of $[a, b]$ and their properties are proved under the assumption that there is an element in this sigma set.

In the fourth paper the basic definitions for a formal approach to functional analysis are given and in this setting the Alaoglu and the Hahn-Banach theorems are proved entirely constructively.

Given a seminormed linear space A we define a formal space whose formal points correspond to the linear functionals of norm ≤ 1 from A to the reals. Let M be a subspace A and F a linear functional on A of norm ≤ 1. Then, in terms of points, the Hahn-Banach theorem says that the restriction function

$$F \longmapsto F_{|M}$$

is surjective. To state the theorem in a pointfree way, we use the usual definition of surjectivity on points as formal injectivity (see for example [MM92]).

In this work we were influenced by earlier pointfree proofs of the Hahn-Banach theorem by Mulvey and Pelletier in [MP91] and Vermeulen in [Ver86]. The proof in [MP91] shows the theorem in any Grothendieck topos and the argument relies on Barr's theorem (which is not justified constructively) and the proof in [Ver86] is done in topos theory with a natural number object (and thus relies on impredicative quantification). Our proof, on the other hand, is developed using only finitary inductive definitions. It follows also rather closely the standard proof of the Hahn-Banach theorem.

Bishop gave a constructive proof of the Hahn-Banach theorem, based on points, in [Bis67]. In his formulation of the theorem, the norm of a linear functional can be preserved to an arbitrary degree by an extension. Bishop also gave a counterexample that shows that the norm, in general, is not preserved exactly. In the pointfree formulation, one works with finite approximations of functionals rather than with the functionals themselves (and classical arguments are of course needed to show the existence of an extended linear functional).

In the last paper the proof of the Hahn-Banach theorem is formalised in Half. A slightly more general definition of the formal space of linear functionals than that given in [4] is used here. This does not affect the informal proof in [4], but it greatly simplifies the implementation.

The implementation was partially developed simultaneously with [4] and this, in fact, influenced the informal proof. Quite important steps in the original development were changed due to errors found during the implementation.

The rational numbers are here defined abstractly, as in [3].

Conclusions and related work

We have used the formal topology introduced by Martin-Löf and Sambin to prove some theorems in constructive analysis. To define the formal topologies we have used only finite inductive definitions. The proofs have also been implemented using a logical framework based on Martin-Löf's intensional type theory with one universe and inductive definitions. For the implementation we have defined both the abstract notion of formal topology, as a type, and proved properties of formal topologies in general, as well as concrete formal topologies as instances.

Bishop [Bis67] had the traditional approach with points to constructive mathematics. He not only showed that fundamental parts of classical mathematics can be rebuilt constructively, he also demonstrated that constructive mathematics can be formulated as elegantly as classical mathematics.

In [Mar70] Martin-Löf presented another approach to constructive analysis, which is more similar to the pointfree one. Spaces are here represented by sets of neighbourhoods (which are assumed to be positive), approximations are certain well behaved recursively enumerated sets of neighbourhoods and constructive points are defined as *maximal* approximations. Open and closed sets are certain sets of neighbourhoods and *recursive functionals* between spaces are mappings of neighbourhoods.

We have taken a pointfree approach and chosen to develop our proofs in a predicative version of type theory using inductive definitions. We have also shown that, using this approach, rather substantial proofs can be developed using a proof assistant.

Let us now draw the attention to some related computer aided formalisations. Considerable parts of mathematics have been formalised in the systems Coq [Dow91] and LEGO [Pol94], which are both based on the calculus of constructions [CG88], and in the Nuprl system [Con86], which is based on an extensional version of Martin-Löf's type theory. Jones [Jon93] uses LEGO for some theorems of constructive analysis in the extended calculus of constructions [Luo89]. She considers the construction of the reals from the rationals and the existence of a completion of a metric space. Chirimar and Howe describe in [CH92] a formalisation, in Nuprl, of parts of constructive real analysis. They follow Bishop in the development of real numbers, which are represented by Cauchy sequences. The main theorem is the completeness theorem for reals. Jackson [Jac94] presents some steps taken in implementing abstract data types in Nuprl, and he uses Nuprl to explore how well suited constructive type theory is for reasoning about abstract data types. Harrison [Har96] uses the HOL system [GM93] in a discussion about real numbers in theorem proving. The reals are constructed using a version of Cantor's method (Cauchy sequences where the terms are scaled up and everything is done using naturals). Harrison describes a formalisation of significant parts of real analysis.

References

[Bar92] H. Barendregt. *Lamda calculi with types*, In S. Abramsky, D.M. Gabbay and T.S.E. Maibaum eds., "Handbook of Logic in Computer Science, Vol. 2", Oxford University Press, Oxford, 1992.

[Bet93] G. Betarte. "A case study in machine-assisted proofs: The Integers form an Integral Domain", Licentiate Thesis, Chalmers University of Technology and University of Göteborg, 1993.

[Bis67] E. Bishop. "Foundations of Constructive Analysis", McGraw-Hill, New York, 1967.

[Bis70] E. Bishop. *Mathematics as a numerical language*, In Myhill, Kino, and Vesley eds., "Intuitionism and Proof Theory", pp. 53–71, North-Holland, Amsterdam, 1970.

[Bru91] N.G. de Bruijn. *A plea for weaker frameworks*, In G. Huet and G. Plotkin eds., "Logical Frameworks", pp. 40–68, Cambridge University Press, Cambridge, 1991.

[Ced94] J. Cederquist. "A machine assisted formalization of pointfree topology in type theory", Licentiate Thesis, Chalmers University of Technology and University of Göteborg, 1994.

[CH92] J. Chirimar, D.J. Howe. *Implementing constructive real analysis: Preliminary report*, In J.P. Myers Jr. and M.J O'Donnell eds., "Constructivity in Computer Science", Lecture Notes in Computer Science 613, pp. 165–178, Springer-Verlag, 1992.

[Con82] R.L. Constable. *Programs as Proofs*, Technical report 82-532, Dept. of Computer Science, Cornell University, 1982.

[Con86] R.L. Constable, et al. "Implementing Mathematics with the Nuprl Development System", Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[Coq92a] T. Coquand. *Constructive Topology and Combinatorics*, In J.P. Myers Jr. and M.J. O'Donnell eds., "Constructivity in Computer Science", Lecture Notes in Computer Science 613, pp. 159–164, 1992.

[Coq92b] T. Coquand. *An Intuitionistic Proof of Tychonoff's Theorem*, The Journal of Symbolic Logic 57, pp. 28–32, 1992.

[Coq92c] T. Coquand. *Pattern Matching with Dependent Types*, "Proceeding from the logical framework workshop at Båstad", 1992.

[Coq95] T. Coquand. *A constructive topological proof of Van der Waerden's theorem*, Journal of Pure and Applied Algebra 105(3), pp. 251–259, 1995.

[Coq96] T. Coquand. *An algorithm for type-checking dependent types*, Science of Computer Programming 26, pp. 167–177, Elsevier, 1996.

[CG88] T. Coquand, G. Huet. *The Calculus of Constructions*, Information and Computation 76 (2/3), pp. 95–120, 1988.

[Dow91] G. Dowek, A. Felty, H. Herbelin, H. Huet, G.P. Murthy, C. Parent, C. Paulin-Mohring, B. Werner, *The Coq Proof Assistant User's Guide Version 5.6*, Rapport Technique 134, INRIA, 1991.

[Dra88] A.G. Dragalin. "Mathematical Intuitionism : Introduction to Proof Theory", Translations of Mathematical Monographs 67, AMS, 1988.

[Dum77] M. Dummet. "Elements of Intuitionism", Clarendon Press, Oxford, 1977.

[FG82] M.P. Fourman, R.J. Grayson. *Formal spaces*, In A. S. Troelstra and D. van Dalen eds., "The L. E. J. Brouwer Centenary Symposium", pp. 107–122, North-Holland, Amsterdam, 1982.

[GM93] M.J.C. Gordon and T.F. Melham eds. "Introduction to HOL: a theorem proving environment for higher order logic", Cambridge University Press, 1993.

[Gra83] R.J. Grayson. *Forcing in intuitionistic systems without power-set*, The Journal of Symbolic Logic 48, pp. 670–682, 1983.

[Har96] J.R. Harrison. "Theorem Proving with Real Numbers", University of Cambridge, PhD Thesis, 1996.

[Hof93] M. Hofmann. *A model of intensional Martin-Löf type theory in which unicity of identity proofs does not hold*, Technical report, Dept. of Computer Science, University of Edinburgh, 1993.

[How80] W.A. Howard. *The Formulae-as-types notion of construction*, In J.P. Seldin and J.R. Hindley eds., "To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism" pp. 479–490, Academic Press, London, 1980.

[Jac94] P. Jackson. *Exploring Abstract Algebra in Constructive Type Theory*, A. Bundy, ed., "Automated Deduction CADE-12", Lecture Notes in Artificial Intelligence 814, pp. 590–604, Springer-Verlag, New York, 1994.

[Joh82] P.T. Johnstone. "Stone Spaces", Cambridge University Press, 1982.

[Joh83] P.T. Johnstone. *The point of pointless topology*, Bull. Amer. Math. Soc. vol. 8, pp. 41–53, 1983.

[Jon93] C. Jones. *Completing the rationals and metric spaces in LEGO*, In G. Huet, G. Plotkin and C. Jones eds., "Logical Frameworks", pp. 209–222, Cambridge University Press, 1991.

[JT84] A. Joyal, M. Tierney. "Extension of the Galois Theory of Grothendieck", Memoirs of the AMS 309, Providence, 1984.

[Kel50] J.L. Kelley. *The Tychonoff product theorem implies the axiom of choice*, Fundamenta Mathematicae 37, pp. 75–76, 1950.

[Luo89] Z. Luo. *ECC, an extended calculus of constrution*, In "Proceedings of the Fourth Annual Conference on Logic in Computer Science", Asilomar, California, 1989.

[MM92] S. MacLane, L. Moerdijk. "Sheaves in Geometry and Logic : A First Introduction to Topos Theory", Springer-Verlag, New York, 1992.

[Mag95] L. Magnusson. "The Implementation of ALF - a Proof Editor based on Martin-Löf's Monomorphic Type Theory with Explicit Substitution", Chalmers University of Technology and University of Göteborg, PhD Thesis, 1995.

[Man76] E.G. Manes. "Algebraic Theories", Graduate Texts in Mathematics 26, Springer-Verlag, 1976.

[Mar70] P. Martin-Löf. "Notes on Constructive Mathematics", Almqvist & Wiksell, Stockholm, 1970.

[Mar72] P. Martin-Löf. *An Intuitionistic Theory of Types* (1972), To be published in the proceedings of Twentyfive years of Constructive Type Theory, G. Sambin and J. Smith eds., Oxford University Press.

[Mar82] P. Martin-Löf. *Constructive Mathematics and Computer Programming*, In "Logic, Methodology and Philosophy of Science VI", L.J. Cohen, J. Loś, H. Pfeiffer and K. Podewski eds., pp. 153–175, North-Holland, 1982.

[Mar83] P. Martin-Löf. *The Domain Interpretation of Type Theory*, In K. Karlsson and K. Petersson eds., "Proceedings of Workshop on Semantics of Programming Languages", Chalmers University of Technology and University of Göteborg, 1983.

[Moh86] C. Mohring. *Algorithm Delvelopment in the Calculus of Constructions*, "Proceedings Symposium on Logic in Computer Science", Cambridge, Mass., pp. 84–91, 1986.

[MP91] C.J. Mulvey, J.W. Pelletier. *A globalization of the Hahn-Banach theorem*, Advances in Mathematics 89, pp. 1–60, 1991.

[NS96] S. Negri, D. Soravia. *The continuum as a formal space*, submitted for publication, 1996.

[NV97] S. Negri, S. Valentini. *Tychonoff's theorem in the framework of formal topologies*, The Journal of Symbolic Logic, in press.

[NPS90] B. Nordström, K. Petersson, J. Smith. "Programming in Martin-Löf's Type Theory", Oxford University Press, 1990.

[NS84] B. Nordström, J. Smith. *Propositions, Types and Specifications in Martin-Löf's Type Theory*, BIT 24(3), pp. 288–301, 1984.

[OSR93] S. Owre, N. Shankar, J. M. Rushby. *The PVS Specification Language (Beta Release)*, Computer Science Laboratory, SRI International, Menlo Park, CA 94025, USA, 1993.

[Per96] H. Persson. "A Formalization of a Constructive Completeness Proof for Intuitionistic Predicate logic", Licentiate Thesis, Chalmers University of Technology, 1996.

[Pol94] R. Pollack. "The Theory of LEGO, A Proof Checker for the Extended Calculus of Constructions", PhD Thesis, University of Edinburgh, 1994.

[Sam87] G. Sambin, *Intuitionistic formal spaces – a first communication*, In D. Skordev ed., "Mathematical Logic and its Applications", pp. 187–204, Plenum Press, New York, 1987.

[Sam95] G. Sambin. *Pretopologies and completeness proofs*, The Journal of Symbolic Logic 60, pp. 861–878, 1995.

[SVV96] G. Sambin, S. Valentini, P. Virgili, *Constructive domain theory as a branch of intuitionistic pointfree topology*, Theoretical Computer Science 159, pp. 319–341, 1996.

[Sig90] I. Sigstam. "On Formal Spaces and their Effective Presentations", Uppsala University, PhD Thesis, 1990.

[Sco82] D. Scott. *Domains for Denotational Semantics*, "20th International Colloquium on Automata, Languages and Programming", Lecture Notes in Computer Science 140, pp. 577–613, Springer-Verlag, 1982.

[Tas93] A. Tasistro. "Formulation of Martin-Löf's Theory of Types with Explicit Substitution", Licentiate Thesis, Chalmers University of Technology and University of Göteborg, 1993.

[TvD88] A.S. Troelstra, D. van Dalen, " Constructivism in Mathematics. An introduction", Volume I and II, North-Holland, 1988.

[Ver86] J.J.C. Vermeulen. "Constructive Techniques in Functional Analysis", PhD Thesis, University of Sussex, 1986.

[Vic89] S. Vickers. "Topology Via Logic", Cambridge University Press, 1989.

A machine assisted formalization of pointfree topology in type theory

Jan Cederquist
Department of Computing Science
University of Göteborg
S-412 96 Göteborg, Sweden
email: ceder@cs.chalmers.se

Abstract

We will present a formalization of pointfree topology in Martin-Löf's type theory. A notion of point will be introduced and we will show that the points of a Scott topology form a Scott domain. This work follows closely the intuitionistic approach to pointfree topology and domain theory, developed mainly by Martin-Löf and Sambin. The important difference is that the definitions and proofs are machine checked by the proof assistant ALF.

Contents

1 Introduction

The traditional motivation for topology relies on abstracting first from Euclidean spaces to metric spaces, and then abstracting out certain properties of their open sets.

A topology ΩX for a set X is a family of subsets of X which is closed under finite intersection and arbitrary union. X is the space of the topology and $\langle X, \Omega X\rangle$ is a topological space. The elements in X are called points and the sets in ΩX are called open sets. For the development of general topology see for instance Kelley [8].

In pointfree topology (locale theory), Johnstone [7], one considers the open sets, and not the points, as primitive entities and studies those properties of a topological space that can be expressed without any mention of points. By abstracting from the fact that open sets are subsets of points one only looks at the algebraic structure, called a *frame*, that the open sets form.

A frame is a partially ordered set A with two operations *meet* $\wedge$ and *join* $\vee$, operating on subsets of A, corresponding to intersection and union, respectively. *Meet* gives to each finite subset the infimum and *join* gives to each subset the supremum. In particular, A contains two elements $true$ and $false$ which correspond to empty meet and join, respectively. Binary meets must also distribute over join. We call the elements in a frame for *opens*.

On the computer science side the motivation for topology relies on connections to domain theory. Scott [14] has showed that by describing only the elements that contain a finite amount of information, the computational content of a domain can be described topologically. This emphasis makes the open sets independent of the points of the topological space, leading to pointfree topology.

Given a frame, points can be defined uniquely from the opens as *completely prime filters* [16]. As an example, take the special case when the opens are open sets (and $true$, $\wedge$, $\vee$ and $\leq$ are the whole space, $\bigcap$, $\bigcup$ and $\subseteq$, respectively). A completely prime filter F then corresponds to the points in the intersection of the open sets in F. In other words, given a point x the corresponding completely prime filter is the set of all open sets containing x. Let $\langle A, \wedge, \vee\rangle$ be a frame and let $F \subseteq A$ be upper closed, that is if $a \in F$ and $a \leq b$ then $b \in F$, then

> F is a *filter* iff it is closed under finite meets:
> $true \in F$ and if $a, b \in F$ then $a \wedge b \in F$,
>
> a filter F is *completely prime* iff it is inaccessible by joins:
> if $S \subseteq A$ and $\bigvee S \in F$ then $s \in F$ for some $s \in S$.

Another motivation for pointfree topology is constructiveness; sometimes the use of pointfree topology makes it possible to replace non constructive reasoning using the axiom of choice by constructive proofs, see for instance Coquand [2, 4].

This work is a machine assisted formalization, in type theory [11], of (a part of) the intuitionistic approach to pointfree topology and domain theory, developed by Martin-Löf [10], Sambin [12], and by Sambin, Valentini and Virgili in [13]. All definitions and proofs are checked by the proof assistant ALF [9]. In [12, 13] the constructivity is guaranteed by adopting Martin-Löf's type theory, but in this paper we will by type theory mean the formalization in ALF. We will prove that this formalization really defines a frame, where the opens are defined as equivalence classes of subsets. A closure operator will be defined and we will prove that each equivalence class contains exactly one closed subset. As a concrete example of a pointfree topology we will look at the neighbourhoods of the natural numbers. A notion of point equivalent to completely prime filter will be introduced. Then we will look at a pointfree version of *Scott topology*, and show that the points of this topology form a *Scott domain*.

Another formalization of constructive domain theory, in ALF, is presented in Hedberg[6]. Hedberg has implemented a cartesian closed category of semilattices and approximable mappings.

In Martin-Löf's type theory, which is implemented in ALF, there are two basic levels: types and sets. The sets are inductively defined and correspond to what is usually called types in a programming language. The types are formed by the type `Set`, the types of elements of sets in `Set`, and function types. In type theory propositions are identified with sets and proofs of propositions are identified with elements of sets; in order to prove that a proposition is true we need to find an element in the corresponding set. Three different forms of definitions (apart from definitions of contexts and substitutions, which will be explained later) will be used in this paper:

1. Inductive definitions of sets or families of sets, which consist of a formation rule and introduction rules prescribing how its canonical elements are formed.
2. Explicit definitions, which are names for well typed expressions.
3. Implicit definitions, which provide the possibility of defining functions using pattern matching [3]. These may be recursive.

All type theory expressions will be written in ALF-syntax and in typewriter font. For example, the set of natural numbers is inductively defined by

```
N : Set
  zero : N
  succ : (n:N)N
```

Addition can then be implicitly defined, using pattern matching:

```
add : (n:N;m:N)N
  add(zero,m) = m
  add(succ(n1),m) = succ(add(n1,m))
```

And a function that doubles its argument can be explicitly defined:

```
double = [n]add(n,n) : (n:N)N
```

In this paper, we will often refer to appendices about proofs. However, most proof terms are too long for a comprehensible presentation, so we have decided to omit many of them entirely and only present their types. Instead all proofs can be obtained by ftp; ftp.cs.chalmers.se: /pub/users/ceder/formtop/*.

2 Formalization of pointfree topology in ALF

2.1 Formal topology

Following Sambin [12, 13], a structure $\langle S, \wedge, 1, \lhd, pos\rangle$ is called a *formal topology* if it satisfies the following requirements:

1. S is a formal base, that is, a set with the binary operation $\wedge$ and element 1 such that $\langle S, \wedge, 1\rangle$ forms a meet semilattice (that is, an algebra with a unit element and a binary operator satisfying commutativity, associativity, unit law and idempotence). The elements in S are called formal basic neighbourhoods.

2. $\lhd$ is a covering relation, that is, a relation between elements of S and subsets of S which for arbitrary $a, b \in S$ and $U, V \subseteq S$ satisfies:

$$\frac{a \in U}{a \lhd U} \text{ reflexivity}$$

$$\frac{a \lhd U \quad (\forall b \in U)(b \lhd V)}{a \lhd V} \text{ transitivity}$$

$$\frac{a \lhd U}{a \wedge b \lhd U} \text{ $\wedge$-left1}$$

$$\frac{a \lhd U \quad a \lhd V}{a \lhd \{b \wedge c : b \in U, c \in V\}} \text{ $\wedge$-right}$$

3. pos is a consistency predicate, that is, a predicate on the elements of S which for arbitrary $a \in S$ and $U \subseteq S$ satisfies:

$$\frac{pos(a) \quad a \lhd U}{(\exists b \in U)\ pos(b)} \text{ monotonicity}$$

$$\frac{pos(a) \rightarrow a \lhd U}{a \lhd U} \text{ positivity}$$

If we extend $\wedge$, $\lhd$ and pos for arbitrary $U, V \subseteq S$ by the definitions

$$U \bigwedge V \equiv \{a \wedge b : a \in U, b \in V\}$$

$$U \lhd V \equiv (\forall a \in U)(a \lhd V)$$

$$POS(U) \equiv (\exists a \in U)pos(a)$$

then transitivity, $\wedge$-right, and monotonicity can be written

$$\frac{a \lhd U \quad U \lhd V}{a \lhd V} \text{ transitivity}$$

$$\frac{a \lhd U \quad a \lhd V}{a \lhd U \bigwedge V} \text{ $\wedge$-right}$$

$$\frac{pos(a) \quad a \lhd U}{POS(U)} \text{ monotonicity}$$

which are also closer to the forthcoming definitions in type theory. (Observe the introduction of the new symbols $\bigwedge$, $\lhd$ and POS. The reason not to let $\wedge$, $\lhd$ and pos be overloaded is that all of them will be defined in ALF and ALF does not support overloading.)

In order to define $\langle S, \wedge, 1\rangle$ to be a semilattice, an ordering or equality between the elements in S is needed. To avoid that, one can notice that if (2) in the definition above holds then $\langle S, \wedge, 1\rangle$ is a semilattice iff for arbitrary $a \in S$ and $U \subseteq S$

$$\frac{b \lhd U}{a \wedge b \lhd U}\ \wedge\text{-left2}$$

and

$$\frac{}{a \lhd \{1\}}\ \wedge\text{-1}$$

hold.

Proof: First, assume that $\langle S, \wedge, 1\rangle$ is a semilattice with equality $=_S$, $\wedge$-left2 then follows from $\wedge$-left1 and commutativity of $\wedge$, and $\wedge$-1 is proved by

$$\frac{\dfrac{\dfrac{\dfrac{}{1 \in \{1\}}}{1 \lhd \{1\}}\ \text{reflexivity}}{a \wedge 1 \lhd \{1\}}\ \wedge\text{-left2} \qquad \dfrac{}{a \wedge 1 =_S a}\ \text{unit}}{a \lhd \{1\}}\ \lhd\ \mathit{must\ respect}\ =_S$$

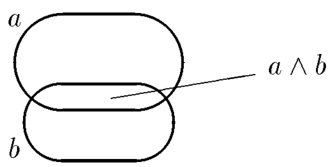
Second, if $\wedge$-left2 and $\wedge$-1 hold then, then we can define an equality between elements in S such that two element are equal if they are covered by each other's singleton sets. Commutativity, associativity, unit law and idempotence for $\wedge$, with respect to that equality are then easily proved; so $\langle S, \wedge, 1\rangle$ form a semilattice with 1 as top element (for more details see appendix D, that $\langle S, \wedge, 1\rangle$ form a semilattice is proved in ALF after the notion of formal topology is defined in type theory).

By exchanging the requirement that $\langle S, \wedge, 1\rangle$ should form a semilattice for the two new rules, we get a definition which is equivalent to the standard definition of formal topology. The reason for this exchange is that it makes the formalization shorter; it is easier to state the new rules than to define a semilattice.

In the definition of formal topology, a subset of S is a propositional function with argument ranging over S. For instance, a is considered as an element in U iff $a \in S$ and $U(a)$ holds. In section 2.3 there is a little theory of these subsets.

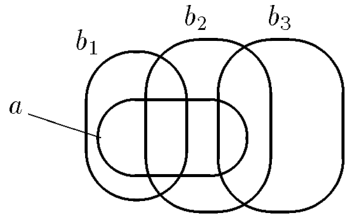
2.2 Explanations of the definition of formal topology

We can think of the elements of S as containing information represented by regions, in such a way that a neighbourhood corresponding to a subregion of another is more informative (it contains more specific information). By $a \wedge b$ we mean the conjunction of the information represented by the intersection of the corresponding regions



and a subset U of S as the disjunction of the information in its elements, represented by the union of the regions of its elements. Then the covering can be understood by a picture: $a \lhd U$

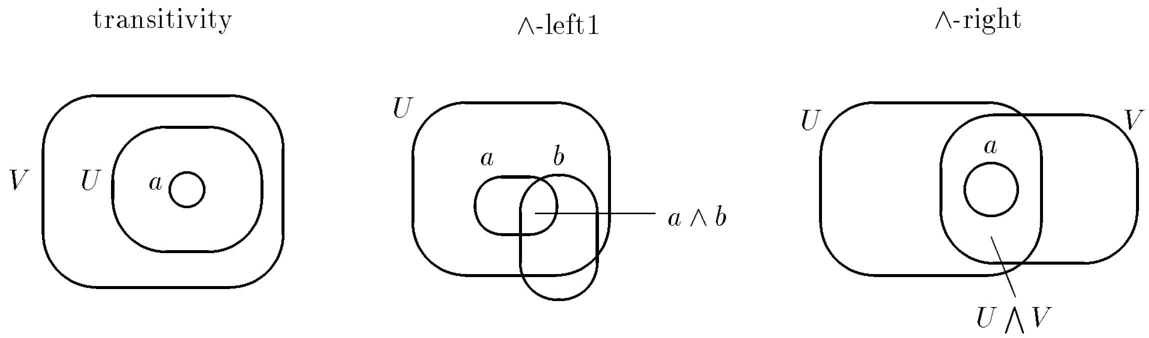
iff the region of a is covered by the region of U.



$a \lhd U$

where $U = \{b_1, b_2, b_3\}$

Transitivity, $\wedge$-left and $\wedge$-right can now be understood by the pictures



Thinking of the neighbourhoods in terms of information we can understand the information in a positive neighbourhood as meaningful or not contradictory. Monotonicity then says that if a is positive and a is covered by U, then U must contain something meaningful. Positivity says exactly that only positive elements contribute to the covering since positivity is equivalent to

$$\frac{a \lhd U}{a \lhd U^+}\;\text{openness} \qquad\qquad \text{where } U^+ \equiv \{b \in U : pos(b)\}$$

Proof: We first assume positivity and show openness:

$$\dfrac{a \lhd U \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[b \in U] \quad [pos(b)]}{b \in U^+}\;\text{def of } U^+}{b \lhd U^+}\;\text{reflexivity}}{pos(b) \to b \lhd U^+}\;\to\text{-intro}}{b \lhd U^+}\;\text{positivity}}{(\forall b \in U)(b \lhd U^+)}\;\forall\text{-intro}}{U \lhd U^+}\;\text{def}}{a \lhd U^+}\;\text{transitivity}$$

Then by assuming openness, positivity is proved by

$$\dfrac{\dfrac{\dfrac{\dfrac{}{a \in \{a\}}}{a \lhd \{a\}}\;\text{reflexivity}}{a \lhd \{a\}^+}\;\text{openness} \qquad \dfrac{\dfrac{\dfrac{pos(a) \to a \lhd U \qquad \dfrac{[b \in \{a\}^+]}{b = a \;\&\; pos(b)}\;\text{def}}{b \lhd U}\;\text{subst,}\to\text{-elim}}{(\forall b \in \{a\}^+)(b \lhd U)}\;\forall\text{-intro}}{\{a\}^+ \lhd U}\;\text{def}}{a \lhd U}\;\text{transitivity}$$

For the moment, regard the elements in S as being neighbourhoods of concrete points; $x \epsilon a$ will be used here to mean that a is a neighbourhood of the point x. Then 1 corresponds to the whole space, $\wedge$ corresponds to intersection, $a \lhd U$ means "the set of points forming a is included in the union of U", and $pos(a)$ means that a is inhabited. For this special case, we can actually prove monotonicity and positivity. For monotonicity: $pos(a)$ implies that there is a point, say x, in a and since $a \lhd U$ there exists a b in U such that $x \epsilon b$, that is $(\exists b \in U)\ pos(b)$. For positivity:

$$\dfrac{\dfrac{\dfrac{[x \epsilon a] \qquad \dfrac{\dfrac{\dfrac{[x \epsilon a]}{pos(a)}\ \text{pos-intro} \qquad pos(a) \to a \lhd U}{a \lhd U}\ \to\text{-elim}}{a \subseteq \bigcup U}\ \text{def}}{x \epsilon \bigcup U}\ \subseteq\text{-elim}}{a \subseteq \bigcup U}\ \subseteq\text{-intro}}{a \lhd U}\ \text{def}$$

2.3 Subsets as propositional functions

As mentioned before, we use propositional functions over the base set S as subsets of S. If U is a propositional function over S and a an element in S, then a is considered to be an element in U iff $U(a)$ holds. We extend this to explain when a subset (propositional function) is a subset of another subset of S. Let U and V be propositional functions over S, then U is a subset of V iff for all a in S, $U(a)$ implies $V(a)$. This can be defined by an introduction rule:

```
subset : (S:Set;U:(S)Set;V:(S)Set)Set

subsetintro : (S:Set;
               U:(S)Set;
               V:(S)Set;
               (a:S;U(a))V(a))
                 subset(S,U,V)
```

U and V are considered equal (as sets) iff they are subsets of each other:

```
eqsubset = [S,U,V]Product(subset(S,U,V),subset(S,V,U)) :
           (S:Set;U:(S)Set;V:(S)Set)Set
```

where `Product` is conjunction.

2.4 Using a context to formalize pointfree topology

We will now represent a formal topology by a list of assumptions (type declarations), in which we assume sets and functions ranging over these sets as well as express the axioms that describe the properties of the formal topology. Lists of type declarations are formalized as contexts, constructions which are governed by the following rules

$$[\,] : Context \qquad\qquad \dfrac{\Gamma : Context \quad \alpha : type\ [\Gamma]}{[\Gamma; x : \alpha] : Context}$$

where x does not occur free in Γ and $[\Gamma; x : \alpha]$ is the extension of Γ with the clause $x : \alpha$.

In the implementation `one` will be used for 1, `meet` and `MEET` for $\wedge$ and $\bigwedge$, respectively, `cov` and `COV` for $\triangleleft$ and $\lhd$, respectively.

First, `MEET`, `COV` and `POS` must be defined since they will be used inside the context defining the topology. They depend on `S`, `meet`, `cov` and `pos`, so `S`, `meet`, `cov` and `pos` occur as parameters in `MEET`, `COV` and `POS`. By this way `MEET`, `COV` and `POS` can be used to different contexts defining formal topologies. But standing for themselves, without such a context, they have of course not the intended meaning. `MEET`, `COV` and `POS` could be explicitly defined, using quantifiers, but introduction rules makes the proofs easier:

```
MEET : (S:Set;meet:(S;S)S;U:(S)Set;V:(S)Set;S)Set

MEETintro : (S:Set;meet:(S;S)S;U:(S)Set;V:(S)Set;a:S;b:S;U(a);V(b))
            MEET(S,meet,U,V,meet(a,b))

COV : (S:Set;cov:(S;(S)Set)Set;U:(S)Set;V:(S)Set)Set

COVintro : (S:Set;cov:(S;(S)Set)Set;U:(S)Set;V:(S)Set;(a:S;U(a))cov(a,V))
           COV(S,cov,U,V)

POS : (S:Set;pos:(S)Set;U:(S)Set)Set

POSintro : (S:Set;pos:(S)Set;U:(S)Set;b:S;U(b);pos(b))POS(S,pos,U)
```

Our context also makes use of singleton sets, which are explicitly defined using propositional equality:

```
Sing = Id : (S:Set;S;S)Set
```

Finally, the formal topology `TOP` is defined as a context which contains the following assumptions: `S` is a set with a particular element `1` and a binary operator `meet`, `cov` is a relation between elements and subsets of `S` and `pos` is a predicate on the elements of `S`, followed by the list of properties (corresponding to the rules in the definition of formal topology in section 2.1) that `S`, `1`, `meet`, `cov` and `pos` must have.

```
TOP is [S:Set; one:S; meet:(S;S)S; cov:(S;(S)Set)Set; pos:(S)Set;
        covmeet1:(a:S)cov(a,Sing(S,one));
        covrefl:(a:S;U:(S)Set;U(a))cov(a,U);
        covtrans:(a:S;
                  U:(S)Set;
                  V:(S)Set;
                  cov(a,U);
                  f:COV(S,cov,U,V))cov(a,V);
        covmeetl1:(a:S;b:S;U:(S)Set;cov(a,U))cov(meet(a,b),U);
        covmeetl2:(a:S;b:S;U:(S)Set;cov(b,U))cov(meet(a,b),U);
```

```
covmeetr:(a:S;
          U:(S)Set;
          V:(S)Set;
          cov(a,U);
          cov(a,V))cov(a,MEET(S,meet,U,V));
mono:(a:S;U:(S)Set;pos(a);cov(a,U))POS(S,pos,U);
posi:(a:S;U:(S)Set;(pos(a))cov(a,U))cov(a,U)]
```

However, using contexts to represent algebraic structures have some drawbacks. For instance, the definition above gives us no template for making new topologies; a proof or definition that involve several algebraic structures require as many contexts. That means that reasoning using many algebraic structures is tedious. In Betarte [1] there is a more detailed discussion about this.

2.4.1 Concrete topology as substitution

We also want to express that some structure is an instance of the definition of formal topology. For that we use the notion of substitution, that is an assignment of objects of appropriate types to the variables in a context. Substitutions are introduced by the following rules

$$\{\ \} : [\,]\ [\Gamma] \qquad \frac{\gamma : \Delta\ [\Gamma] \quad \alpha : type\ [\Delta] \quad a : \alpha\gamma\ [\Gamma]}{\{\gamma; x := a\} : [\Delta; x : \alpha]\ [\Gamma]}$$

where $\{\ \}$ is the empty substitution and $\{\gamma; x := a\}$ is the extension of the substitution γ with the assignment $x := a$. This will be used in the example below. In Tasistro [15], substitutions are explained in more detail.

2.4.2 Example: Neighbourhoods of the natural numbers

As an example, given by Sambin [12], of a concrete pointfree topology we take the set SN of neighbourhoods of the natural numbers given by the rules

$$\frac{}{\overline{N} \in SN}$$

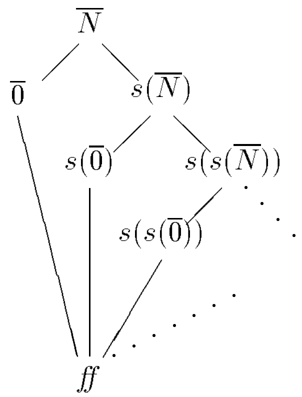
$$\frac{}{\overline{0} \in SN}$$

$$\frac{a \in SN}{s(a) \in SN}$$

$$\frac{}{\mathit{ff} \in SN}$$

and if a and b are two neighbourhoods of a number then, their intersection, $a \wedge_{nat} b$ is a neighbourhood of the same number. Furthermore, a neighbourhood is positive if it is a neighbourhood of a number.

The intended meaning is that $s^n(\overline{N})$, where $n \in N$, is a neighbourhood of all numbers in $\{n, n+1, n+2, ...\}$, $s^n(\overline{0})$ is a neighbourhood only of $s^n(0)$, and no number has ff as neighbourhood (ff is needed to make sure that given two neighbourhoods a and b, $a \wedge_{nat} b$ is a neighbourhood). The figure illustrates the structure that the neighbourhoods form:



The figure is not complete, there are also an infinite number of empty neighbourhoods of the form $s(...s(\mathit{ff})...)$, which are not identical to ff but are equal to ff in the sense that they are all non positive and therefore also covered by each other's singleton sets.

Formalized in type theory, SN is a set with four constructors:

```
SN : Set
onenat : SN
zero : SN
s : (SN)SN
ff : SN
```

where `onenat` and `zero` correspond to $\overline{N}$ and $\overline{0}$, respectively. $\wedge_{nat}$ (`meetnat`) can be implicitly defined, using pattern matching:

```
meetnat : (a:SN;b:SN)SN
        meetnat(onenat,b) = b
        meetnat(zero,onenat) = zero
        meetnat(zero,zero) = zero
        meetnat(zero,s(h)) = ff
        meetnat(zero,ff) = ff
        meetnat(s(h),onenat) = s(h)
        meetnat(s(h),zero) = ff
        meetnat(s(h),s(h1)) = s(meetnat(h,h1))
        meetnat(s(h),ff) = ff
        meetnat(ff,b) = ff
```

We define a neighbourhood to be positive if it is a neighbourhood of a number, thus ff, $s(\mathit{ff})$, $s(s(\mathit{ff}))$, ... are the only non-positive neighbourhoods:

```
posnat : (a:SN)Set
        posnat(onenat) = N1
        posnat(zero) = N1
        posnat(s(h)) = posnat(h)
        posnat(ff) = Empty
```

where N1 is the set containing tt as only element, that is, a true proposition and Empty is the empty set, that is, a false proposition.

Before defining the covering relation we define a partial order $\leq_{nat}$ on the neighbourhoods by

$$a \leq_{nat} b \quad \text{iff} \quad a \wedge_{nat} b = a$$

This is the same ordering as in a semilattice (and in the figure above), which the neighbourhoods in fact form even though we have not proved it yet. In type theory $\leq_{nat}$ is explicitly defined using propositional equality:

```
leqnat = [a,b]Id(SN,meetnat(a,b),a) : (a:SN;b:SN)Set
```

Now we can define the covering relation, for arbitrary $a \in S$ and $U \subseteq S$, by

$$a \lhd U \quad \text{iff} \quad a \textit{ is not positive} \quad \text{or} \quad (\exists b \in U)(a \leq_{nat} b)$$

But instead the following definition by introduction rules will be used

```
covnat : (a:SN;U:(SN)Set)Set
covnati1 : (a:SN;U:(SN)Set;(posnat(a))Empty)covnat(a,U)
covnati2 : (a:SN;U:(SN)Set;b:SN;U(b);leqnat(a,b))covnat(a,U)
```

It is easy to see that the two definitions of covering above ($\lhd$ and covnat) are equivalent. The reason not to define covnat explicitly, using existential quantification, is that the definition by introduction rules makes the proofs easier and shorter.

In order to show that SN, onenat, meetnat, covnat and posnat is a formal topology one must prove that all the properties of formal topology (the properties listed in the definition of TOP) are satisfied. Consult appendix C for more details.

The proof that the neighbourhoods of the natural numbers is a formal topology is then completed by the substitution TOPNAT:

```
TOPNAT is {S:=SN; one:=onenat; meet:=meetnat; cov:=covnat; pos:=posnat;
           covmeet1:=covmeetnat1; covrefl:=covreflnat;
           covtrans:=covtransnat; covmeetl1:=covmeetnatl1;
           covmeetl2:=covmeetnatl2; covmeetr:=covmeetnatr;
           mono:=mononat; posi:=posinat} : TOP      □
```

2.5 Properties of a formal topology

In this section we will concentrate on definitions and types, not on the proofs. The proof terms of the types are too long for a readable presentation, they can however be obtained by ftp (see the introduction). For a description of the proofs see Sambin [12]. The definitions and results of this section are not used in the rest of the paper.

2.5.1 Frames and complete Heyting algebras

Here we show that a formal topology defines a frame in such a way that equivalence classes of subsets (the equality will soon be defined) are the opens, COV corresponds to the partial order and MEET corresponds to the meet operation.

First we define the equality relation between subsets such that two subsets are equal iff they cover each other:

```
EQS = [U,V]Product(COV(S,cov,U,V),COV(S,cov,V,U)) :
      (U:(S)Set;V:(S)Set)Set     TOP
```

Note here that we are doing all this in the context TOP. That EQS is an equivalence relation is easily proved (see appendix E: EQSsymm, EQSrefl, EQStrans).

The opens (equivalence classes of subsets) are difficult to define in ALF and so are ordering, meet- and join-operations for opens, instead we will rely on the fact that the ordering respects EQS and that EQS respects meet and join, which are defined on subsets. Of course that has to be proved, the types of the proof is in appendix E: COVrespEQS, EQSrespMEET, EQSrespJOIN.

For the ordering COV is used, which is a partial order on the family of subsets of S (appendix E: COVtrans, COVrefl, antisymmetry follows directly from the definition of the equality EQS).

For the meet operation we use MEET, which gives the infimum (appendix E: MEETisinfl1, MEETisinfl2, MEETisinfr).

Join is defined as a union:

```
JOIN = [T,I,U]union(S,T,I,U) : (T:Set;I:(T)Set;U:(T;S)Set;S)Set     TOP
```

We postpone the definition of union to section 3.4. JOIN gives the supremum (appendix E: JOINissup1, JOINissup2).

Finally the infinite distributivity

```
(T:Set;I:(T)Set;V:(S)Set;U:(T;S)Set)
 EQS(MEET(S,meet,V,JOIN(T,I,U)),
     JOIN(T,I,[i]MEET(S,meet,V,U(i))))     TOP
```

holds (appendix E: infdistr).

This far we have proved that a formal topology defines a frame. Implication can then be defined in the frame so it becomes a *complete Heyting algebra*.

A *complete Heyting algebra* is a complete lattice A where, for every $a, b \in A$, there is an element $a \to b$ satisfying

$c \leq a \to b$ iff $c \wedge a \leq b$.

In a frame $\to$ is defined by

$a \to b \equiv \bigvee\{c : c \wedge a \leq b\}$.

For a proof that this definition of implication gives a complete Heyting algebra see for instance [16].

The definition of implication translated to our case becomes

```
cHaimply = [U,V,a]COV(S,cov,MEET(S,meet,U,Sing(S,a)),V) :
           (U:(S)Set;V:(S)Set;a:S)Set     TOP
```

cHaimply respects EQS and satisfies the implication property (see appendix E: cHaimplyrespEQS, cHaimplyprop1, cHaimplyprop2). This completes the proof that a formal topology defines a complete Heyting algebra.

2.5.2 Closure operator

In the previous subsection it was shown that the equivalence classes of subsets form a frame. Now we will define a *closure operator*, that is an operator that given a subset U returns its *downward closure*. The downward closure of a subset U is the subset of all neighbourhoods which are covered by U.

We will show that each equivalence class contains a closed set and that the closed sets form a frame which is isomorphic to the frame formed by the equivalence classes in such way that each equivalence class is represented by its closed set.

A closure operator, Cl, is an operator acting on subsets and satisfying the following properties

$$U \subseteq Cl(U)$$
$$U \subseteq V \to Cl(U) \subseteq Cl(V)$$
$$Cl(Cl(U)) = Cl(U)$$

Here `Cl` is explicitly defined by

```
Cl = [U,a]cov(a,U) : (U:(S)Set;a:S)Set      TOP
```

`Cl` satisfy the closure operator properties (appendix F: `Clprop1`,2,3).

We then say that a subset is closed or saturated if it is equal, as a subset, to its closure:

```
sat = [U]eqsubset(S,U,Cl(U)) : (U:(S)Set)Set      TOP
```

Since

```
(U:(S)Set)EQS(U,Cl(U))      TOP
```

holds, any equivalence class contains a closed subset. Given two subsets in the same equivalence class, their closures are equal

```
(U:(S)Set;V:(S)Set;EQS(U,V))eqsubset(S,Cl(U),Cl(V))      TOP,
```

so any equivalence class contains exactly one closed subset. Thus the closed subsets form a frame which is isomorphic to the frame formed by the equivalence classes.

```
(U:(S)Set;V:(S)Set;COV(S,cov,U,V))subset(S,Cl(U),Cl(V))      TOP
```

and

```
(U:(S)Set;V:(S)Set;subset(S,Cl(U),Cl(V)))COV(S,cov,U,V)      TOP
```

hold, so the order in this frame is the subset order.

`cHaimply(U,V)` is closed for any two subsets `U` and `V`, and `Cl` preserves implication, so the closed sets form a cHa which is isomorphic to the one formed by the equivalence classes (appendix F: `satcHaimply`, `ClprescHaimply`).

In appendix F meet- and join-operations are also defined. In appendix F there are also proofs of that `Cl` is a cHa isomorphism.

2.6 Points

A formal point (Sambin [13]) of a formal topology $\langle S, \wedge, 1, \triangleleft, pos \rangle$ is a subset p of S which, for arbitrary $a, b \in S$, satisfies

1. $\overline{1 \in p}$

2. $\dfrac{a \in p \quad b \in p}{a \wedge b \in p}$

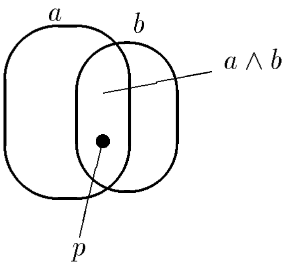
3. $\dfrac{a \in p \quad a \lhd U}{(\exists b \in U)(b \in p)}$

4. $\dfrac{a \in p}{pos(a)}$

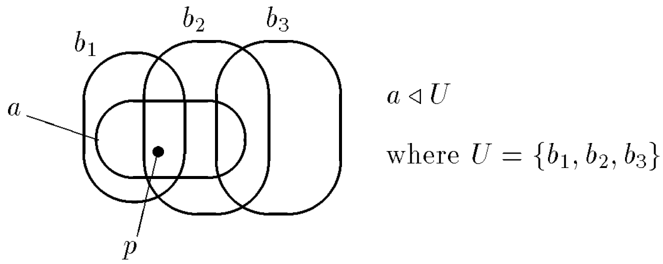
Even though a point is a subset, the intuition of a subset as an open and a subset as a point are not the same. A subset (recall section 2.2) we regard as union of the regions of its elements, while we can understand a point as something in the intersection of all neighbourhoods in it. So an informal understanding of $a \in p$ (where a is a neighbourhood and p a point) might be "p is a point in a". Then we can understand the definition of points in the following way.

1. Any point p is in the space (since 1 corresponds to the whole space).

2. If p is in both a and b, then p is in the intersection of a and b.



3. If p is in a and a is covered by U, then U must contain a neighbourhood containing p.



4. If p is in a then a is meaningful.

To avoid the existential quantification, in rule 3 of the definition of formal point, we make the following definition

```
P : (p:(S)Set;U:(S)Set)Set      TOP

Pintro : (p:(S)Set;U:(S)Set;b:S;U(b);p(b))P(p,U)      TOP
```

Informally: $P(p,U)$ holds iff $(\exists b \in U)(b \in p)$.

From rule (3) one can see that a definition by introduction rules of points impossible:

```
point : (p:(S)Set)Set     TOP

pointintro : (p:(S)Set;
              p(one);
              (a:S;b:S;p(a);p(b))p(meet(a,b));
              (a:S;U:(S)Set;p(a);cov(a,U))P(p,U);
              (a:S;p(a))pos(a))
                point(p)     TOP
```

For instance it does not follow the general scheme (given in [5]) of an inductive definition: `U` is of function type and is not a parameter to the definition. So a neat definition of formal points in type theory seems to be impossible. Instead we can do the following: in order to prove that a subset `p` is a point we prove

`p(one)` ,

`(a,b:S;p(a);p(b))p(meet(a,b))` ,

`(a:S;U:(S)Set;p(a);cov(a,U))P(p,U)`

and

`(a:S;p(a))pos(a)`.

And in order to prove that, given a point `p`, some property `C(p)` holds, we assume all properties a point must have:

```
(p:(S)Set;
 p(one);
 (a:S;b:S;p(a);p(b))p(meet(a,b));
 (a:S;U:(S)Set;p(a);cov(a,U))P(p,U);
 (a:S;p(a))pos(a))
   C(p)     TOP
```

The above definition of points corresponds exactly to the definition of points as completely prime filters. For details see appendix G.

3 The points of a Scott topology form a Scott domain

In this section we will show that the formal points of a *Scott topology* form a *Scott domain*.

3.1 Scott domain

By a Scott domain we mean an algebraic cpo in which every family of elements which is bounded above has a least upper bound. Observe that we use the word family and not subset: in general the points do not form a proper set in the type theoretic sense. In the following definitions, which are adopted from Sambin [13], there is a distinction between *sets*, *collections* and *families*. Sets are inductively defined and families are subcollections indexed by sets or subsets (propositional functions).

Let $\mathcal{D} = \langle D, \sqsubseteq \rangle$ be a partially ordered collection. A family $(x_i)_{i \in I}$ of elements in D is *bounded* whenever there exist an element $x \in D$ such that $(\forall i \in I)(x_i \sqsubseteq x)$ and *directed* if I is inhabited and $(\forall i, j \in I)(\exists k \in I)(x_i \sqsubseteq x_k \;\&\; x_j \sqsubseteq x_k)$. $\mathcal{D}$ is called a *complete partial order* (*cpo*) if D has a minimum element $\bot$ and every directed family has a supremum. The supremum of a directed family $(x_i)_{i \in I}$ will be denoted $\bigsqcup_{i \in I} x_i$.

An element a of a cpo $\mathcal{D}$ is called *compact* if, for any directed family $(x_i)_{i \in I}$ of elements in D, $a \sqsubseteq \bigsqcup_{i \in I} x_i$ implies that $(\exists k \in I)(a \sqsubseteq x_k)$. We will write $K(D)$ for the collection of compact elements of D.

A cpo $\mathcal{D}$ is called *algebraic* if, for every $x \in D$, the collection $\{a \in K(D) : a \sqsubseteq x\}$ of compact lower bounds of x is a directed family of elements $(a_i)_{i \in I}$, for a suitable index set I, such that $x = \bigsqcup_{i \in I} a_i$. This definition is stronger than the traditional, normally it is only required that $x = \bigsqcup \{a \in K(D) : a \sqsubseteq x\}$ since $\{a \in K(D) : a \sqsubseteq x\}$ is directed. But here we also require that the compact elements of a domain must form a family.

From Sambin [13] it follows that any algebraic cpo such that any bounded pair of compact elements has a supremum is a Scott domain. This is the property that we will show that the points satisfy.

3.2 Scott topology

In the following definition we mean by set, set in classical set theory. Let $\langle X, \sqsubseteq \rangle$ be a poset of points. The *Scott topology* on $\langle X, \sqsubseteq \rangle$ consists of all sets $U \subseteq X$ that satisfy

- U is upward closed, that is if $x \in U$ and $x \sqsubseteq y$ then $y \in U$.
- U is inaccessible by directed joins, that is if $V \subseteq X$ is directed and $\bigvee V \in U$ then $(\exists x \in V)(x \in U)$.

Now let $\langle X, \sqsubseteq \rangle$ be a Scott domain and consider its Scott topology. It can be shown (Sambin [13]) that the subsets $O_U = \{x \in X : (\forall a \in U)(a \sqsubseteq x)\}$, for $U \subseteq_f K(X)$ ($\subseteq_f$ means finite subset), form a base for this topology. Moreover if O_U is inhabited and $O_U \subseteq \bigcup_{i \in I} O_{U_i}$ then $(\exists i \in I)(O_U \subseteq O_{U_i})$.

Proof: Assume O_U is inhabited and $O_U \subseteq \bigcup_{i \in I} O_{U_i}$. From O_U inhabited it follows that U is bounded above and since X is a Scott domain U has a supremum $\bigsqcup U$, for which $(\forall a \in U)(a \sqsubseteq \bigsqcup U)$ holds, that is $\bigsqcup U \in O_U$. Now

$$\begin{aligned} O_U \subseteq \bigcup_{i \in I} O_{U_i} &\Rightarrow \bigsqcup U \in \bigcup_{i \in I} O_{U_i} \\ &\Leftrightarrow (\exists i \in I)(\bigsqcup U \in O_{U_i}) \\ &\Leftrightarrow (\exists i \in I)(\forall a \in U_i)(a \sqsubseteq \bigsqcup U). \end{aligned}$$

Then take an arbitrary $x \in O_U$. Since $\langle X, \sqsubseteq \rangle$ is algebraic, x is equal to the supremum of its compact lower bounds, hence $\bigsqcup U \sqsubseteq x$. By transitivity of $\sqsubseteq$, $(\forall a \in U_i)(a \sqsubseteq x)$ for some $i \in I$, that is $x \in O_{U_i}$ for some $i \in I$. So $(\exists i \in I)(O_U \subseteq O_{U_i})$.

This property of Scott topologies is taken as definition in the pointfree approach, thinking of the base $\{O_U : U \subseteq_f K(X)\}$ as a formal base. A formal topology is called *Scott* if it satisfies

$$\frac{a \lhd U \quad pos(a)}{(\exists b \in U)(a \lhd \{b\})}\ \text{scott}$$

By the definition

```
SCOTTOP is TOP + [scott:(a:S;
                         U:(S)Set;
                         cov(a,U);
                         pos(a))
                            Exists(S,[b]Product(U(b),cov(a,Sing(S,b))))]
```

SCOTTOP is the context TOP extended with the scott property.

3.2.1 Example: Neighbourhoods of the natural numbers

Recall the example in section 2.4.2. It is easy to see that the Scott property is also satisfied, so `SN`, `onenat`, `meetnat`, `covnat` and `posnat` actually form a Scott topology. For a full proof in ALF see appendix I.

3.3 Points of a Scott topology

If the Scott property holds then the covering relation $\lhd$ can be replaced by the simpler relation $\leq$ defined by

$$a \leq b \equiv a \lhd \{b\}$$

A formal point is the same as a subset p satisfying

1. $\dfrac{}{1 \in p}$

2. $\dfrac{a \in p \quad b \in p}{a \wedge b \in p}$

3. $\dfrac{a \in p \quad a \leq b}{b \in p}$

4. $\dfrac{a \in p}{pos(a)}$

If we first define the new order

```
leq = [a,b]cov(a,Sing(S,b)) : (a:S;b:S)Set     TOP
```

then the points can be defined (there is no longer a quantification over subsets):

```
scpoint : (p:(S)Set)Set     SCOTTOP

scpintro : (p:(S)Set;
            p(one);
            (a:S;b:S;p(a);p(b))p(meet(a,b));
            (a:S;b:S;p(a);leq(a,b))p(b);
            (a:S;p(a))pos(a))
              scpoint(p)     SCOTTOP
```

It is easy to prove that if the Scott property holds then a subset `p` is a point iff `scpoint(p)` holds. For details see appendix J (`point2scpoint`, `scpoint2point`).

3.4 Definition of union of subsets, directed families and compactness

Before we present the proof, that the points of a Scott topology form a Scott domain, some more definitions are needed. In the proofs we will look at families of points where the index set itself is a point (propositional function), so in the definitions the index set must be a general subset. Formally, the family $\{p_i\}_{i\in I}$ consists of three parts:

```
T:Set
I:(T)Set
p:(T;S)Set
```

The intended meaning is: if `i:T` then `p(i)` is a member of the family iff `I(i)` holds.

Union of subsets:

```
union : (S:Set;T:Set;I:(T)Set;U:(T;S)Set;S)Set

unionintro : (S:Set;
              T:Set;
              I:(T)Set;
              U:(T;S)Set;
              i:T;
              I(i);
              a:S;
              U(i,a))
                union(S,T,I,U,a)
```

The order in the domain is the subset order, so in the definition of directed families and compactness we use `subset`.

Directed families:

```
directed : (S:Set;T:Set;I:(T)Set;p:(T;S)Set)Set

directedintro : (S:Set;
                 T:Set;
                 I:(T)Set;
                 p:(T;S)Set;
                 i0:T;
                 I(i0);
                 (i:T;j:T;I(i);I(j))
                   Exists(T,[k]Product(I(k),
                                       Product(subset(S,p(i),p(k)),
                                               subset(S,p(j),p(k)))))
                     directed(S,T,I,p)
```

As was the case with the points (section 2.6), a definition by introduction rules of compactness is impossible. It would contain quantifications over function types:

```
compact : ((S)Set)Set

compactintro : (q:(S)Set;
                (T:Set;
                 I:(T)Set;
                 p:(T;S)Set;
                 directed(S,T,I,p);
                 subset(S,q,union(S,T,I,p)))
                   Exists(T,[i]Product(I(i),subset(S,q,p(i)))))
                     compact(q)
```

In order to prove that a point q in a cpo is compact, one has to show that the function type

```
(T:Set;
 I:(T)Set;
 p:(T;S)Set;
 directed(S,T,I,p);
 subset(S,q,union(S,T,I,p)))
   Exists(T,[i]Product(I(i),subset(S,q,p(i))))
```

is inhabited. And in order to prove that, given a compact point q, some property C(q) holds then all the properties of a compact point has to be assumed:

```
(q:(S)Set;
 scpoint(q);
 (T:Set;
  I:(T)Set;
  p:(T;S)Set;
  directed(S,T,I,p);
  subset(S,q,union(S,T,I,p)))Exists(S,[i]Product(I(i),subset(S,q,p(i)))))
    C(q)
```

3.5 The points of a Scott topology form a Scott domain

Unless otherwise stated, the types for the propositions in this section can be found in appendix L. The formal proofs can be obtained by ftp.

3.5.1 The points form an algebraic cpo

In order to show that the points of a Scott topology form a cpo, which we denote by $\mathcal{P}t(S)$ (where S is the formal base), we need the extra property that 1 is positive. Points are subsets of positive neighbourhoods and 1 is contained in all points, so without the knowledge that 1 is positive we cannot show that there are any points at all, particularly no bottom element, and consequently no cpo.

The order is the subset order (subset) which, of course, is reflexive (subsetrefl in appendix A), transitive (subsettrans in appendix A) and antisymmetric (by definition of the equality, eqsubset).

Next we need a bottom element, a point which is a subset of all other points. Given a positive element $a \in S$, its upper closure $\uparrow a = \{c \in S : a \leq c\}$ can easily be shown to be a point. In our type theoretic notation, the upper closure of a neighbourhood `a` is `leq(a)`. `genscp` is a proof that given an arbitrary positive neighbourhood `a`, `leq(a)` is a point; in particular if `pos(one)` holds then `leq(one)` is a point. Intuitively, since all points contains 1 and all points are upper closed, the upper closure of 1 is a least element; `leqonemin` is a formal proof of that.

The union of a family of subsets is, of course, an upper bound of the family (`unionsup1`), it is less than or equal to all upper bounds (`unionsup2`) and the union of a directed family of points is a point itself (`unionpoint`). So given a directed family of points its supremum is formed by taking the union of all points in the family.

That $\mathcal{P}t(S)$ is algebraic is proved as follows. Given two arbitrary points p and q, we have

$$(\uparrow a)_{a \in p} \textit{ is directed} \tag{1}$$

and

$$q \subseteq p \;\&\; q \textit{ compact} \Leftrightarrow (\exists a \in p)(q = \uparrow a) \tag{2}$$

which implies that the family of compact lower bounds to p is $(\uparrow a)_{a \in p}$. Finally the union of $(\uparrow a)_{a \in p}$ (which is the supremum) is equal to p:

$$\bigcup_{a \in p} \uparrow a = p \tag{3}$$

Proof of 1: $1 \in p$ so p is inhabited. If $a, b \in p$ then $a \wedge b \in p$ and $\uparrow a, \uparrow b \subseteq \uparrow (a \wedge b)$.

Proof of 2 $\Rightarrow$: Assume that $q \subseteq p$ and q is compact. The family $(\uparrow a)_{a \in q}$ is directed (follows from 1). Clearly $q \subseteq \bigcup_{a \in q} \uparrow a$, so by the definition of compactness $(\exists a \in q)(q \subseteq \uparrow a)$ follows. But if $a \in q$ then $\uparrow a \subseteq q$, so we have $(\exists a \in q)(q = \uparrow a)$ and from the assumption $q \subseteq p$ we get $(\exists a \in p)(q = \uparrow a)$.

$\Leftarrow$: Assume that $(\exists a \in p)(q = \uparrow a)$. Clearly $q \subseteq p$ holds. By existential elimination $q = \uparrow a$, for some $a \in p$. Let $(r_i)_{i \in I}$ be a directed family such that $q \subseteq \bigcup_{i \in I} r_i$. By substitution we have $\uparrow a \subseteq \bigcup_{i \in I} r_i$. Now

$$\begin{aligned} \uparrow a \subseteq \bigcup_{i \in I} r_i \quad &\Leftrightarrow \quad a \in \bigcup_{i \in I} r_i \\ &\Leftrightarrow \quad (\exists i \in I)(a \in r_i) \\ &\Leftrightarrow \quad (\exists i \in I)(\uparrow a \subseteq r_i). \end{aligned}$$

And by substituting back $(\exists i \in I)(q \subseteq r_i)$ follows. So q is compact.

Proof of 3: It is easy to see that $p \subseteq \bigcup_{a \in p} \uparrow a$. Let $b \in \bigcup_{a \in p} \uparrow a$ then $(\exists a \in p)(b \in \uparrow a)$. But if $a \in p$ then $\uparrow a \subseteq p$ so $b \in p$. Thus $\bigcup_{a \in p} \uparrow a \subseteq p$.

Again, the proof terms are too long so we only present the types. 1 is proved by

```
genscpdir : (p:(S)Set;scpoint(p))directed(S,S,p,leq)     SCOTTOP
```

The implication from left to right in 2 follows from

```
scpcomplb1 : (q:(S)Set;
              scpoint(q);
              (T:Set;
               I:(T)Set;
               r:(T;S)Set;
               directed(S,T,I,r);
               subset(S,q,union(S,T,I,r)))
                 Exists(T,[x]Product(I(x),subset(S,q,r(x))));
              p:(S)Set;
              subset(S,q,p))
                Exists(S,[a]Product(p(a),eqsubset(S,q,leq(a))))     SCOTTOP
```

The first conjunct in the implication from right to left follows from

```
scpcomplb2a : (p:(S)Set;
               q:(S)Set;
               scpoint(p);
               Exists(S,[a]Product(p(a),eqsubset(S,q,leq(a)))))
                 subset(S,q,p)     SCOTTOP
```

and the second conjunct from

```
scpcomplb2b : (p:(S)Set;
               q:(S)Set;
               Exists(S,[a]Product(p(a),eqsubset(S,q,leq(a))));
               T:Set;
               I:(T)Set;
               r:(T;S)Set;
               (i:T;I(i))scpoint(r(i));
               subset(S,q,union(S,T,I,r)))
                 Exists(T,[x]Product(I(x),subset(S,q,r(x))))     SCOTTOP
```

3 is proved by

```
supcompscp : (p:(S)Set;scpoint(p))eqsubset(S,p,union(S,S,p,leq))     SCOTTOP
```

3.5.2 Every bounded pair of compact points has a supremum

In order to prove that every bounded pair of compact points has a supremum we first notice that

$$\textit{if a point } p \textit{ is compact then } (\exists a \in p)(p = \uparrow a) \tag{4}$$

The proof of this is similar to the proof of 2. (In fact the converse also holds). Now take two arbitrary compact points p_1 and p_2, we then know that there exists positive a and b such that $p_1 = \uparrow a$ and $p_2 = \uparrow b$. If p_1 and p_2 are bounded, by say the point r, then $a, b \in r$ and since r is a point $a \wedge b \in r$. Again, since r is a point, $a \wedge b$ is positive so $\uparrow(a \wedge b)$ is a point. $\uparrow a, \uparrow b \subseteq \uparrow(a \wedge b)$ and if $\uparrow a, \uparrow b \subseteq q$ then $\uparrow(a \wedge b) \subseteq q$. Hence the supremum of p_1 and p_2 is $\uparrow(a \wedge b)$, provided they are bounded.

4 follows from

```
gencompscp : (p:(S)Set;
              scpoint(p);
              (T:Set;
               I:(T)Set;
               p2:(T;S)Set;
               directed(S,T,I,p2);
               subset(S,p,union(S,T,I,p2)))
                 Exists(T,[h]Product(I(h),subset(S,p,p2(h)))))
                Exists(S,[h]Product(p(h),eqsubset(S,p,leq(h))))
```

Now the fact that every two compact points, which are bounded above, have a supremum is proved by

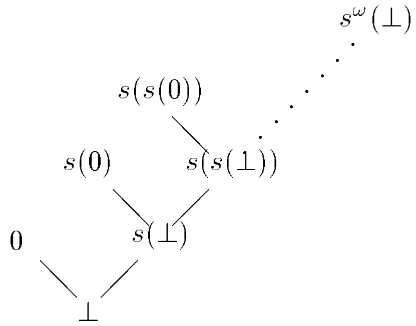
```
psuptoleqp : (p1:(S)Set;
              p2:(S)Set;
              scpoint(p1);
              scpoint(p2);
              (T:Set;
               I:(T)Set;
               q:(T;S)Set;
               directed(S,T,I,q);
               subset(S,p1,union(S,T,I,q)))
                 Exists(T,[h]Product(I(h),subset(S,p1,q(h))));
              (T:Set;
               I:(T)Set;
               q:(T;S)Set;
               directed(S,T,I,q);
               subset(S,p2,union(S,T,I,q)))
                 Exists(T,[h]Product(I(h),subset(S,p2,q(h))));
              r:(S)Set;
              scpoint(r);
              subset(S,p1,r);
              subset(S,p2,r);
              q:(S)Set;
              scpoint(q))
                Exists(S,[x]Product(Product(scpoint(leq(x)),
                                            Product(subset(S,p1,leq(x)),
                                                    subset(S,p2,leq(x)))),
                                    Imply(Product(subset(S,p1,q),
                                                  subset(S,p2,q)),
                                          subset(S,leq(x),q))))     SCOTTOP
```

The meaning of the type above is the following: if p_1, p_2 and q are compact points, such that p_1 and p_2 are bounded above, then

$$(\exists x \in S)(\uparrow x \textit{ is a point } \& \ p_1, p_2 \subseteq \uparrow x \ \& \ (p_1, p_2 \subseteq q \ \rightarrow \uparrow x \subseteq q)).$$

3.5.3 Example: Natural numbers

Recall again the example in section 2.4.2 and 3.2.1, on formal neighbourhoods of the natural numbers. From the definition of points in a Scott topology it is easy to see that the domain formed by the points in our case is



where

$$
\begin{array}{ccl}
\bot & \text{is} & \{\overline{N}\} \\
0 & \text{is} & \{\overline{N}, \overline{0}\} \\
s(\bot) & \text{is} & \{\overline{N}, s(\overline{N})\} \\
s(0) & \text{is} & \{\overline{N}, s(\overline{N}), s(\overline{0})\} \\
s(s(\bot)) & \text{is} & \{\overline{N}, s(\overline{N}), s(s(\overline{N}))\} \\
. & & \quad . \\
. & & \quad . \\
s^\omega(\bot) & \text{is} & \{\overline{N}, s(\overline{N}), s(s(\overline{N})), ...\}
\end{array}
$$

and all points except of $s^\omega(\bot)$ are finite.

4 Discussion

4.1 Subsets as propositional functions

By using propositional functions to represent subsets we can form subsets that cannot be constructed by using sets of type `Set` as subsets: if P is a predicate over the set S then we also have the subset $\{x \in S : P(x)\}$, in general there is no way to effectively produce the elements of this subset, and the same element can occur in several subsets.

Another possibility is to use $\sum$-sets: let $a \in \sum(S, U)$ iff there exist some $b : U(a)$ such that $\langle a, b\rangle : \sum(S, U)$. But to create $\sum(S, U)$ we need the predicate U and when we use an element a of a subset we first have to pick out a from the pair $\langle a, b\rangle$.

4.2 The consistency predicate

The rule of positivity,

$$\frac{pos(a) \to a \lhd U}{a \lhd U} \quad ,$$

has not been used in any formal proof. The rule of monotonicity,

$$\frac{pos(a) \quad a \lhd U}{(\exists b \in U)\ pos(b)}\ ,$$

has been used to show: if a is a positive neighbourhood in a Scott topology, then the upper closure of a, $\uparrow a$, is a point. This, in turn, is frequently used in the proof that the points of a Scott topology form a Scott domain, but the reason for that is that the points, by definition, consist of positive neighbourhoods.

By simply removing the consistency predicate and its rules, we can still show that "any formal topology defines a frame" and "the points of a Scott topology form a Scott domain". For Scott topologies, however, the Scott property,

$$\frac{a \lhd U \quad pos(a)}{(\exists b \in U)(a \lhd \{b\})}\ ,$$

must be changed; if a is the least element then $a \lhd U$ even if U is empty. A new condition for a Scott topology might be

$$a \lhd U \quad \leftrightarrow \quad (\forall b \in S)(a \lhd \{b\}) \ \bigvee\ (\exists b \in U)(a \lhd \{b\})$$

or, even better, remove the old condition and replace the covering $\lhd$ by $\leq$ (defined by $a \leq b \equiv a \lhd \{b\}$) in all the other rules. So one might ask whether the consistency predicate is needed. The category of Scott topologies (with consistency predicate) is equivalent to the category of Scott domains (for a proof see [13]), we cannot expect that this equivalence still holds if we remove some rules from the definition of formal topology.

4.3 Problems

One of the main problems that occured when formalizing pointfree topology and domain theory in ALF, was that we did not find any internal definition of points (section 2.6) and compactness (section 3.4). As a consequence many types have become long and hard to read; it is cumbersome to say that something is a point/compact element, both as assumption and as result of a proposition. Another consequence is that the proof terms, even for trivial lemmas, have become unreadable; they contain many variables and several of them are of function type.

Another problem is that some properties are difficult to express inside the theory. To show the statements "the equivalence classes of subsets form a frame" and "the points of a Scott topology form a Scott domain" we have proved a lot of properties, which together implies that one can understand, outside the theory, that the statements are correct.

5 Acknowledgements

I would like to thank my supervisor Jan Smith for all his support. I also would like to thank Thierry Coquand, Bengt Nordström, Randy Pollack, Mikael Rittri and Björn von Sydow for discussions and valuable comments on earlier versions of this thesis.

References

[1] Gustavo Betarte. A case study in machine-assisted proofs: The integers form an integral domain. Licentiate Thesis, Chalmers University of Technology and University of Göteborg, Sweden, November 1993.

[2] Thierry Coquand. Constructive topology and combinatorics. In *proceeding of the conference Constructivity in Computer Science, San Antonio, LNCS 613*, pages 28–32, 1992.

[3] Thierry Coquand. Pattern matching with dependent types. In *Proceeding from the logical framework workshop at Båstad*, June 1992.

[4] Thierry Coquand. An intuitionistic proof of tychonoff's theorem. *Journal of Symbolic Logic*, pages 28–32, Volume 57, 1992.

[5] Peter Dybjer. An inversion principle for Martin-Löf's type theory. In *Proceedings of the Workshop on Programming Logic, Båstad*, May 1989. Accepted (subject to revision) for publication in *Formal Aspects of Computing.*

[6] Michael Hedberg. *Type Theory and the External Logic of Programs.* PhD thesis, Department of Computing Science, Chalmers University of Technology, Göteborg, Sweden, May 1994.

[7] Peter T. Johnstone. *Stone spaces.* Cambridge University Press, 1982.

[8] John L. Kelley. *General Topology.* Van Nostrand, 1955.

[9] Lena Magnusson. The new Implementation of ALF. In *The informal proceeding from the logical framework workshop at Båstad, June 1992*, 1992.

[10] Per Martin-Löf. The Domain Interpretation of Type Theory, Lecture Notes. In Kent Karlsson and Kent Petersson, editors, *Workshop on Semantics of Programming Languages, Abstracts and Notes*, Chalmers University of Technology and University of Göteborg, August 1983. Programming Methodology Group.

[11] Bengt Nordström, Kent Petersson, and Jan M. Smith. *Programming in Martin-Löf's Type Theory. An Introduction.* Oxford University Press, 1990.

[12] Giovanni Sambin. Intuitionistic Formal Spaces - A First Communication. In *The Proceedings of Conference on Logic and its Applications, Bulgaria.* Plenum Press, 1986.

[13] Giovanni Sambin, Silvio Valentini, and Paolo Virgili. Constructive domain theory as a branch of intuitionistic pointfree topology. Technical report, Dip.di Matematica Pura e Appl., Univ. di Padova, August 1992.

[14] Dana Scott. Lectures on a Mathematical Theory of Computation. Technical Monograph, Prg 19, Oxford University Computing Laboratory, May 1981.

[15] Alvaro Tasistro. Formulation of Martin-Löf's Theory of Types with Explicit Substitution. Licentiate Thesis, Chalmers University of Technology and University of Göteborg, Sweden, May 1993.

[16] Steven Vickers. *Topology Via Logic.* Cambridge University Press, 1989.

A Subsets as propositional functions

```
subset : (S:Set;U:(S)Set;V:(S)Set)Set     []    C

subsetintro : (S:Set;
               U:(S)Set;
               V:(S)Set;
               (a:S;U(a))V(a))
                 subset(S,U,V)     []    C


eqsubset = [S,U,V]Product(subset(S,U,V),subset(S,V,U)) :
           (S:Set;U:(S)Set;V:(S)Set)Set     []



subsetrefl : (S:Set;U:(S)Set)subset(S,U,U)     []    I

    subsetrefl(S,U) = subsetintro(S,U,U,[a,h]h)


subsettrans : (S:Set;
               U:(S)Set;
               V:(S)Set;
               W:(S)Set;
               subset(S,U,V);
               subset(S,V,W))
                 subset(S,U,W)     []    I

    subsettrans(S,U,V,W,subsetintro(_,_,_,h2),subsetintro(_,_,_,h)) =
        subsetintro(S,U,W,[a,h1]h(a,h2(a,h1)))
```

B Formal topology

```
COV : (S:Set;cov:(S;(S)Set)Set;U:(S)Set;V:(S)Set)Set     []    C

COVintro : (S:Set;
            cov:(S;(S)Set)Set;
            U:(S)Set;
            V:(S)Set;
            (a:S;U(a))cov(a,V))
              COV(S,cov,U,V)     []    C
```

```
MEET : (S:Set;meet:(S;S)S;U:(S)Set;V:(S)Set;S)Set     []    C

MEETintro : (S:Set;
             meet:(S;S)S;
             U:(S)Set;
             V:(S)Set;
             a:S;
             b:S;
             U(a);
             V(b))
               MEET(S,meet,U,V,meet(a,b))     []    C


POS : (S:Set;pos:(S)Set;U:(S)Set)Set     []    C

POSintro : (S:Set;
            pos:(S)Set;
            U:(S)Set;
            b:S;
            U(b);
            pos(b))
              POS(S,pos,U)     []    C


Sing = Id : (S:Set;S;S)Set     []


TOP is [S:Set; one:S; meet:(S;S)S; cov:(S;(S)Set)Set; pos:(S)Set;
        covmeet1:(a:S)cov(a,Sing(S,one));
        covrefl:(a:S;U:(S)Set;U(a))cov(a,U);
        covtrans:(a:S;U:(S)Set;V:(S)Set;cov(a,U);f:COV(S,cov,U,V))cov(a,V);
        covmeetl1:(a:S;b:S;U:(S)Set;cov(a,U))cov(meet(a,b),U);
        covmeetl2:(a:S;b:S;U:(S)Set;cov(b,U))cov(meet(a,b),U);
        covmeetr:(a:S;
                  U:(S)Set;
                  V:(S)Set;
                  cov(a,U);
                  cov(a,V))
                    cov(a,MEET(S,meet,U,V));
        mono:(a:S;U:(S)Set;pos(a);cov(a,U))POS(S,pos,U);
        posi:(a:S;U:(S)Set;(pos(a))cov(a,U))cov(a,U)]
```

C A concrete topology: Neighbourhoods of the natural numbers

```
SN : Set     []    C

onenat : SN     []    C
zero : SN     []    C
s : (SN)SN     []    C
ff : SN     []    C


meetnat : (a:SN;b:SN)SN     []    I
        meetnat(onenat,b) = b
        meetnat(zero,onenat) = zero
        meetnat(zero,zero) = zero
        meetnat(zero,s(h)) = ff
        meetnat(zero,ff) = ff
        meetnat(s(h),onenat) = s(h)
        meetnat(s(h),zero) = ff
        meetnat(s(h),s(h1)) = s(meetnat(h,h1))
        meetnat(s(h),ff) = ff
        meetnat(ff,b) = ff


leqnat = [a,b]Id(SN,meetnat(a,b),a) : (a:SN;b:SN)Set     []


posnat : (a:SN)Set     []    I
        posnat(onenat) = N1
        posnat(zero) = N1
        posnat(s(h)) = posnat(h)
        posnat(ff) = Empty


covnat : (a:SN;U:(SN)Set)Set     []    C

covnati1 : (a:SN;U:(SN)Set;(posnat(a))Empty)covnat(a,U)     []    C
covnati2 : (a:SN;U:(SN)Set;b:SN;U(b);leqnat(a,b))covnat(a,U)     []    C
```

We present only the types, because of the length of the proofs:

```
covmeetnat1 : (a:SN)covnat(a,Sing(SN,onenat))     []

covreflnat : (a:SN;U:(SN)Set;U(a))covnat(a,U)     []

covmeetnatl1 : (a:SN;b:SN;U:(SN)Set;covnat(a,U))covnat(meetnat(a,b),U)     []
```

```
covmeetnatl2 : (a:SN;b:SN;U:(SN)Set;covnat(b,U))covnat(meetnat(a,b),U)     []

covmeetnatr : (a:SN;
               U:(SN)Set;
               V:(SN)Set;
               covnat(a,U);
               covnat(a,V))
                 covnat(a,MEET(SN,meetnat,U,V))     []

covtransnat : (a:SN;
               U:(SN)Set;
               V:(SN)Set;
               covnat(a,U);
               f:COV(SN,covnat,U,V))
                 covnat(a,V)     []

mononat : (a:SN;U:(SN)Set;posnat(a);covnat(a,U))POS(SN,posnat,U)     []

posinat : (a:SN;U:(SN)Set;(posnat(a))covnat(a,U))covnat(a,U)     []


TOPNAT is {S:=SN; one:=onenat; meet:=meetnat; cov:=covnat; pos:=posnat;
           covmeet1:=covmeetnat1; covrefl:=covreflnat;
           covtrans:=covtransnat; covmeetl1:=covmeetnatl1;
           covmeetl2:=covmeetnatl2; covmeetr:=covmeetnatr;
           mono:=mononat;posi:=posinat} : TOP     []
```

D Semilattice

Order between the formal neighbourhoods:

`leq = [a,b]cov(a,Sing(S,b)) : (a:S;b:S)Set     TOP`

Equality between the formal neighbourhoods:

`eqs = [a,b]Product(leq(a,b),leq(b,a)) : (S;S)Set     TOP`

$\langle$`S,meet,one`$\rangle$ with order `leq` form a semilattice. We only present the types, since the proofs are too long:

```
meetcomm : (a:S;b:S)eqs(meet(a,b),meet(b,a))     TOP

meetassoc : (a:S;b:S;c:S)eqs(meet(a,meet(b,c)),meet(meet(a,b),c))     TOP

meetunit : (a:S)eqs(meet(one,a),a)     TOP

meetidem : (a:S)eqs(meet(a,a),a)     TOP
```

E A formal topology defines a frame/complete Heyting algebra

Equivalence between subsets:

```
EQS = [U,V]Product(COV(S,cov,U,V),COV(S,cov,V,U)) : (U:(S)Set;V:(S)Set)Set     TOP
```

Join in the formal topology:

```
JOIN = [T,I,U]union(S,T,I,U) : (T:Set;I:(T)Set;U:(T;S)Set;S)Set     TOP
```

(`union` is defined in appendix K).

Implication in complete Heyting algebra:

```
cHaimply = [U,V,a]COV(S,cov,MEET(S,meet,U,Sing(S,a)),V) :
           (U:(S)Set;V:(S)Set;a:S)Set     TOP
```

A formal topology defines a frame/complete Heyting algebra. We only present the types, because of the length of the proof terms.

`EQS` is an equivalence relation:

```
EQSrefl : (U:(S)Set)EQS(U,U)     TOP

EQSsymm : (U:(S)Set;V:(S)Set;EQS(U,V))EQS(V,U)     TOP

EQStrans : (U:(S)Set;V:(S)Set;W:(S)Set;EQS(U,V);EQS(V,W))EQS(U,W)     TOP
```

`COV` is a partial order on the subsets of `S` (antisymmetri is direct from the definition of `EQS`):

```
COVrefl : (U:(S)Set)COV(S,cov,U,U)     TOP

COVtrans : (U:(S)Set;
            V:(S)Set;
            W:(S)Set;
            COV(S,cov,U,V);
            COV(S,cov,V,W))
              COV(S,cov,U,W)     TOP
```

`COV` respects `EQS`:

```
COVrespEQS : (U:(S)Set;
              V:(S)Set;
              U':(S)Set;
              V':(S)Set;
              COV(S,cov,U,V);
              EQS(U,U');
              EQS(V,V'))
                COV(S,cov,U',V')     TOP
```

EQS respects MEET:

```
EQSrespMEET : (U:(S)Set;
               U':(S)Set;
               V:(S)Set;
               V':(S)Set;
               EQS(U,U');
               EQS(V,V'))
                 EQS(MEET(S,meet,U,V),MEET(S,meet,U',V'))     TOP
```

EQS respects JOIN:

```
EQSrespJOIN : (T:Set;
               I:(T)Set;
               U:(T;S)Set;
               U':(T;S)Set;
               (i:T;I(i))EQS(U(i),U'(i)))
                 EQS(JOIN(T,I,U),JOIN(T,I,U'))     TOP
```

The following proofs show that MEET, JOIN and cHaimply have the correct properties:

```
MEETisinfl1 : (U:(S)Set;V:(S)Set)COV(S,cov,MEET(S,meet,U,V),U)     TOP

MEETisinfl2 : (U:(S)Set;V:(S)Set)COV(S,cov,MEET(S,meet,U,V),V)     TOP

MEETisinfr : (W:(S)Set;
              U:(S)Set;
              V:(S)Set;
              COV(S,cov,W,U);
              COV(S,cov,W,V))
                COV(S,cov,W,MEET(S,meet,U,V))     TOP

MEETempty : (U:(S)Set)COV(S,cov,U,Sing(S,one))     TOP

JOINissup1 : (T:Set;I:(T)Set;U:(T;S)Set;i:T;I(i))
               COV(S,cov,U(i),JOIN(T,I,U))     TOP

JOINissup2 : (T:Set;
              I:(T)Set;
              U:(T;S)Set;
              V:(S)Set;
              (i:T;I(i))COV(S,cov,U(i),V))
                COV(S,cov,JOIN(T,I,U),V)     TOP

infdistr : (T:Set;
            I:(T)Set;
            V:(S)Set;
            U:(T;S)Set)EQS(MEET(S,meet,V,JOIN(T,I,U)),
                           JOIN(T,I,[i]MEET(S,meet,V,U(i))))     TOP
```

```
cHaimplyrespEQS : (U:(S)Set;
                    U':(S)Set;
                    V:(S)Set;
                    V':(S)Set;
                    EQS(U,U');
                    EQS(V,V'))
                      EQS(cHaimply(U,V),cHaimply(U',V'))     TOP
```

```
cHAimplyprop1 : (W:(S)Set;
                  U:(S)Set;
                  V:(S)Set;
                  COV(S,cov,W,cHaimply(U,V)))
                    COV(S,cov,MEET(S,meet,W,U),V)     TOP
```

```
cHaimplyprop2 : (W:(S)Set;
                  U:(S)Set;
                  V:(S)Set;
                  COV(S,cov,MEET(S,meet,W,U),V))
                    COV(S,cov,W,cHaimply(U,V))     TOP
```

F Closure operator

Closure operator:

`Cl = [U,a]cov(a,U) : (U:(S)Set;a:S)Set     TOP`

Meet for the closed sets:

`MEETsat = [U,V,a]Product(U(a),V(a)) : (U:(S)Set;V:(S)Set;a:S)Set     TOP`

Join for the closed sets:

`JOINsat = [T,I,U]Cl(JOIN(T,I,U)) : (T:Set;I:(T)Set;U:(T;S)Set;S)Set     TOP`

Predicate for closed sets:

`sat = [U]eqsubset(S,U,Cl(U)) : (U:(S)Set)Set     TOP`

We only present the types, because of the length of the proofs.

The covering order between closed subsets is the subset order:

`Cllemma1a : (U:(S)Set;V:(S)Set;COV(S,cov,U,V))subset(S,U,Cl(V))     TOP`

`Cllemma1b : (U:(S)Set;V:(S)Set;subset(S,U,Cl(V)))COV(S,cov,U,V)     TOP`

Each equivalence class contains exactly one closed subset:

`Cllemma2 : (U:(S)Set)EQS(U,Cl(U))     TOP`

`Cllemma3 : (U:(S)Set;V:(S)Set;EQS(U,V))eqsubset(S,Cl(U),Cl(V))     TOP`

`Cl` is a closure operator:

```
Clprop1 : (U:(S)Set)subset(S,U,Cl(U))     TOP

Clprop2 : (U:(S)Set;V:(S)Set;subset(S,U,V))subset(S,Cl(U),Cl(V))     TOP

Clprop3 : (U:(S)Set)eqsubset(S,Cl(Cl(U)),Cl(U))     TOP
```

The closed subsets form a frame which is isomorphic to the frame formed by the equivalence classes and `Cl` is a cHa isomorphism:

```
ClJOIN2JOINsat : (T:Set;
                  I:(T)Set;
                  U:(T;S)Set)
                    eqsubset(S,
                             Cl(JOIN(T,I,U)),
                             JOINsat(T,I,[i]Cl(U(i))))     TOP

ClMEET2MEETsatempty : (a:S)Cl(Sing(S,one),a)     TOP

ClMEET2MEETsatbin : (U:(S)Set;
                     V:(S)Set)
                       eqsubset(S,
                                Cl(MEET(S,meet,U,V)),
                                MEETsat(Cl(U),Cl(V)))     TOP

satcHaimply : (U:(S)Set;V:(S)Set)sat(cHaimply(U,V))     TOP

ClprescHaimply : (U:(S)Set;
                  V:(S)Set)
                    eqsubset(S,Cl(cHaimply(U,V)),cHaimply(Cl(U),Cl(V)))     TOP
```

G Points of a formal topology

`P` is a function that given a point returns a completely prime filter:

```
P : (p:(S)Set;U:(S)Set)Set     TOP     C

Pintro : (p:(S)Set;U:(S)Set;b:S;U(b);p(b))P(p,U)     TOP     C
```

We only present the types, because of the length of the proof terms.

Any point defines a completely prime filter:

```
point2filter1 : (p:(S)Set;
                 p(one);
                 (a:S;b:S;p(a);p(b))p(meet(a,b));
                 (a:S;U:(S)Set;p(a);cov(a,U))P(p,U);
                 (a:S;p(a))pos(a);
                 U:(S)Set;
                 V:(S)Set;
                 COV(S,cov,U,V);
                 P(p,U))
                   P(p,V)     TOP

point2filter2 : (p:(S)Set;
                 p(one);
                 (a:S;b:S;p(a);p(b))p(meet(a,b));
                 (a:S;U:(S)Set;p(a);cov(a,U))P(p,U);
                 (a:S;p(a))pos(a))
                   P(p,Sing(S,one))     TOP

point2filter3 : (p:(S)Set;
                 p(one);
                 (a:S;b:S;p(a);p(b))p(meet(a,b));
                 (a:S;U:(S)Set;p(a);cov(a,U))P(p,U);
                 (a:S;p(a))pos(a);
                 U:(S)Set;
                 V:(S)Set;
                 P(p,U);
                 P(p,V))
                   P(p,MEET(S,meet,U,V))     TOP

point2filter4 : (p:(S)Set;
                 p(one);
                 (a:S;b:S;p(a);p(b))p(meet(a,b));
                 (a:S;U:(S)Set;p(a);cov(a,U))P(p,U);
                 (a:S;p(a))pos(a);
                 T:Set;
                 I:(T)Set;
                 U:(T;S)Set;
                 P(p,JOIN(T,I,U)))
                   Exists(T,[i]Product(I(i),P(p,U(i))))     TOP
```

```
point2filter5 : (p:(S)Set;
                 p(one);
                 (a:S;b:S;p(a);p(b))p(meet(a,b));
                 (a:S;U:(S)Set;p(a);cov(a,U))P(p,U);
                 (a:S;p(a))pos(a);
                 U:(S)Set;
                 P(p,U))
                   POS(S,pos,U)     TOP
```

Any completely prime filter defines a point:

```
filter2point1 : (F:((S)Set)Set;
                 (U:(S)Set;V:(S)Set;COV(S,cov,U,V);F(U))F(V);
                 F(Sing(S,one));
                 (U:(S)Set;V:(S)Set;F(U);F(V))F(MEET(S,meet,U,V));
                 (T:Set;I:(T)Set;U:(T;S)Set;F(JOIN(T,I,U)))
                   Exists(T,[i]Product(I(i),F(U(i))));
                 (U:(S)Set;F(U))POS(S,pos,U))
                   F(Sing(S,one))     TOP

filter2point2 : (F:((S)Set)Set;
                 (U:(S)Set;V:(S)Set;COV(S,cov,U,V);F(U))F(V);
                 F(Sing(S,one));
                 (U:(S)Set;V:(S)Set;F(U);F(V))F(MEET(S,meet,U,V));
                 (T:Set;I:(T)Set;U:(T;S)Set;F(JOIN(T,I,U)))
                   Exists(T,[i]Product(I(i),F(U(i))));
                 (U:(S)Set;F(U))POS(S,pos,U);
                 a:S;
                 b:S;
                 F(Sing(S,a));
                 F(Sing(S,b)))
                   F(Sing(S,meet(a,b)))     TOP

filter2point3 : (F:((S)Set)Set;
                 (U:(S)Set;V:(S)Set;COV(S,cov,U,V);F(U))F(V);
                 F(Sing(S,one));
                 (U:(S)Set;V:(S)Set;F(U);F(V))F(MEET(S,meet,U,V));
                 (T:Set;I:(T)Set;U:(T;S)Set;F(JOIN(T,I,U)))
                   Exists(T,[i]Product(I(i),F(U(i))));
                 (U:(S)Set;F(U))POS(S,pos,U);
                 a:S;
                 U:(S)Set;
                 cov(a,U);
                 F(Sing(S,a)))
                   P([x]F(Sing(S,x)),U)     TOP
```

```
filter2point4 : (F:((S)Set)Set;
                 (U:(S)Set;V:(S)Set;COV(S,cov,U,V);F(U))F(V);
                 F(Sing(S,one));
                 (U:(S)Set;V:(S)Set;F(U);F(V))F(MEET(S,meet,U,V));
                 (T:Set;I:(T)Set;U:(T;S)Set;F(JOIN(T,I,U)))
                   Exists(T,[i]Product(I(i),F(U(i))));
                 (U:(S)Set;F(U))POS(S,pos,U);
                 a:S;
                 F(Sing(S,a)))
                   pos(a)     TOP
```

P is a bijection:

```
pfbij1a : (p:(S)Set;
           p(one);
           (a:S;b:S;p(a);p(b))p(meet(a,b));
           (a:S;U:(S)Set;p(a);cov(a,U))P(p,U);
           (a:S;p(a))pos(a);
           a:S;
           p(a))
             P(p,Sing(S,a))     TOP

pfbij1b : (p:(S)Set;
           p(one);
           (a:S;b:S;p(a);p(b))p(meet(a,b));
           (a:S;U:(S)Set;p(a);cov(a,U))P(p,U);
           (a:S;p(a))pos(a);
           a:S;
           P(p,Sing(S,a)))
             p(a)     TOP

pfbij2a : (F:((S)Set)Set;
           (U:(S)Set;V:(S)Set;COV(S,cov,U,V);F(U))F(V);
           F(Sing(S,one));
           (U:(S)Set;V:(S)Set;F(U);F(V))F(MEET(S,meet,U,V));
           (T:Set;I:(T)Set;U:(T;S)Set;F(JOIN(T,I,U)))
             Exists(T,[i]Product(I(i),F(U(i))));
           (U:(S)Set;F(U))POS(S,pos,U);
           U:(S)Set;
           F(U))
             P([a]F(Sing(S,a)),U)     TOP
```

```
pfbij2b : (F:((S)Set)Set;
           (U:(S)Set;V:(S)Set;COV(S,cov,U,V);F(U))F(V);
           F(Sing(S,one));
           (U:(S)Set;V:(S)Set;F(U);F(V))F(MEET(S,meet,U,V));
           (T:Set;I:(T)Set;U:(T;S)Set;F(JOIN(T,I,U)))
             Exists(T,[i]Product(I(i),F(U(i))));
           (U:(S)Set;F(U))POS(S,pos,U);
           U:(S)Set;
           P([a]F(Sing(S,a)),U))
             F(U)     TOP
```

H Scott topology

```
SCOTTOP is TOP + [scott:(a:S;
                         U:(S)Set;
                         cov(a,U);
                         pos(a))
                           Exists(S,[b]Product(U(b),cov(a,Sing(S,b))))]

SCOTTOP1 is SCOTTOP + [pos1:pos(one)]
```

I A concrete Scott topology: Neighbourhoods of the natural numbers

Here the example from appendix C continues.

```
scottnat : (a:SN;
            U:(SN)Set;
            covnat(a,U);
            posnat(a))
              Exists(SN,[b]Product(U(b),covnat(a,Sing(SN,b))))     []    I

    scottnat(a,U,covnati1(_,_,h2),h1) =
        case0([h]Exists(SN,[b]Product(U(b),covnat(a,Sing(SN,b)))),h2(h1))

    scottnat(a,U,covnati2(_,_,b,h2,h3),h1) =
        Exists_intro(SN,
                     [b']Product(U(b'),covnat(a,Sing(SN,b'))),
                     b,
                     pair(U(b),
                          covnat(a,Sing(SN,b)),
                          h2,
                          covnati2(a,Sing(SN,b),b,id(SN,b),h3)))
```

```
TOPNAT2 is {S:=SN; one:=onenat; meet:=meetnat; cov:=covnat; pos:=posnat;
            covmeet1:=covmeetnat1; covrefl:=covreflnat; covtrans:=covtransnat;
            covmeetl1:=covmeetnatl1; covmeetl2:=covmeetnatl2;
            covmeetr:=covmeetnatr; mono:=mononat; posi:=posinat;
            scott:=scottnat} : SCOTTOP     []
```

J Points of a Scott topology

Points of a Scott topology:

```
scpoint : (p:(S)Set)Set     SCOTTOP    C

scpintro : (p:(S)Set;
            p(one);
            (a:S;b:S;p(a);p(b))p(meet(a,b));
            (a:S;b:S;p(a);leq(a,b))p(b);
            (a:S;p(a))pos(a))
              scpoint(p)     SCOTTOP    C
```

In a Scott topology, scpoint is the same as point:

```
point2scpoint = [p,h,h1,h2,h3]
    scpintro(p,
             h,
             h1,
             [a,b,h4,h5]Exists_elim(S,
                                    [x]Product(Sing(S,b,x),
                                               p(x)),
                                    [h6]p(b),
                                    [a',b']idsubst'(S,
                                                    [b1]p(b1),
                                                    b,
                                                    a',
                                                    proj1(Id(S,b,a'),p(a'),b'),
                                                    proj2(Sing(S,b,a'),p(a'),b')),
                                    h2(a,Sing(S,b),h4,h5)),
             h3) :
    (p:(S)Set;
     p(one);
     (a:S;b:S;p(a);p(b))p(meet(a,b));
     (a:S;U:(S)Set;p(a);cov(a,U))Exists(S,[x]Product(U(x),p(x)));
     (a:S;p(a))pos(a))
       scpoint(p)     SCOTTOP    I
```

```
scpoint2point : (p:(S)Set;
                 scpoint(p);
                 a:S;
                 U:(S)Set;
                 p(a);
                 cov(a,U))
                   Exists(S,[h]Product(U(h),p(h)))     SCOTTOP    I
    scpoint2point(p,scpintro(_,h3,h4,h5,h6),a,U,h1,h2) =
      Exists_elim(S,
                  [b]Product(U(b),cov(a,Sing(S,b))),
                  [z]Exists(S,[h]Product(U(h),p(h))),
                  [a',b]Exists_intro(S,
                                     [h]Product(U(h),p(h)),
                                     a',
                                     pair(U(a'),
                                          p(a'),
                                          proj1(U(a'),cov(a,Sing(S,a')),b),
                                          h5(a,a',h1,proj2(U(a'),leq(a,a'),b)))),
                  scott(a,U,h2,h6(a,h1)))
```

K Union of subsets and directed families

Union of subsets, where the index set itself is a subset:

```
union : (S:Set;T:Set;I:(T)Set;U:(T;S)Set;S)Set     []    C

unionintro : (S:Set;
              T:Set;
              I:(T)Set;
              U:(T;S)Set;
              i:T;
              I(i);
              a:S;
              U(i,a))
                union(S,T,I,U,a)     []    C
```

Directed families, where the index set is a subset:

```
directed : (S:Set;T:Set;I:(T)Set;p:(T;S)Set)Set     []    C

directedintro : (S:Set;
                 T:Set;
                 I:(T)Set;
                 p:(T;S)Set;
                 i0:T;
                 I(i0);
                 (i:T;j:T;I(i);I(j))
                   Exists(T,[k]Product(I(k),
                                       Product(subset(S,p(i),p(k)),
                                               subset(S,p(j),p(k))))))
                     directed(S,T,I,p)     []    C
```

L The points in a Scott topology form a Scott domain

We only present the types, since the proofs are too long.

The points in a Scott topology form a cpo:

```
leqonemin : (p:(S)Set;scpoint(p))subset(S,leq(one),p)     SCOTTOP

unionpoint : (T:Set;
              I:(T)Set;
              p:(T;S)Set;
              (i:T;I(i))scpoint(p(i));
              directed(S,T,I,p))
                scpoint(union(S,T,I,p))     SCOTTOP

unionsup1 : (T:Set;
             I:(T)Set;
             p:(T;S)Set;
             i:T;
             I(i))
               subset(S,p(i),union(S,T,I,p))     TOP

unionsup2 : (T:Set;
             I:(T)Set;
             p:(T;S)Set;
             q:(S)Set;
             (i:T;I(i))subset(S,p(i),q))
               subset(S,union(S,T,I,p),q)     TOP
```

which is algebraic and every two points has a least upper bound:

```
genscp : (a:S;pos(a))scpoint(leq(a))     SCOTTOP
```

```
genscpdir : (p:(S)Set;scpoint(p))directed(S,S,p,leq)     SCOTTOP

gencompscp : (p:(S)Set;
              scpoint(p);
              (T:Set;
               I:(T)Set;
               p2:(T;S)Set;
               directed(S,T,I,p2);
               subset(S,p,union(S,T,I,p2)))
                 Exists(T,[h]Product(I(h),subset(S,p,p2(h)))))
                   Exists(S,[h]Product(p(h),eqsubset(S,p,leq(h))))     SCOTTOP

scpcomplb1 : (q:(S)Set;
              scpoint(q);
              (T:Set;
               I:(T)Set;
               r:(T;S)Set;
               directed(S,T,I,r);
               subset(S,q,union(S,T,I,r)))
                 Exists(T,[x]Product(I(x),subset(S,q,r(x))));
              p:(S)Set;
              subset(S,q,p))
                Exists(S,[a]Product(p(a),eqsubset(S,q,leq(a))))     SCOTTOP

scpcomplb2a : (p:(S)Set;
               q:(S)Set;
               scpoint(p);
               Exists(S,[a]Product(p(a),eqsubset(S,q,leq(a)))))
                 subset(S,q,p)     SCOTTOP

scpcomplb2b : (p:(S)Set;
               q:(S)Set;
               Exists(S,[a]Product(p(a),eqsubset(S,q,leq(a))));
               T:Set;
               I:(T)Set;
               r:(T;S)Set;
               (i:T;I(i))scpoint(r(i));
               subset(S,q,union(S,T,I,r)))
                 Exists(T,[x]Product(I(x),subset(S,q,r(x))))     SCOTTOP

supcompscp : (p:(S)Set;scpoint(p))eqsubset(S,p,union(S,S,p,leq))     SCOTTOP
```

```
psuptoleqp : (p1:(S)Set;
              p2:(S)Set;
              scpoint(p1);
              scpoint(p2);
              (T:Set;
               I:(T)Set;
               q:(T;S)Set;
               directed(S,T,I,q);
               subset(S,p1,union(S,T,I,q)))
                 Exists(T,[h]Product(I(h),subset(S,p1,q(h))));
              (T:Set;
               I:(T)Set;
               q:(T;S)Set;
               directed(S,T,I,q);
               subset(S,p2,union(S,T,I,q)))
                 Exists(T,[h]Product(I(h),subset(S,p2,q(h))));
              r:(S)Set;
              scpoint(r);
              subset(S,p1,r);
              subset(S,p2,r);
              q:(S)Set;
              scpoint(q))
                Exists(S,[x]Product(Product(scpoint(leq(x)),
                                            Product(subset(S,p1,leq(x)),
                                                    subset(S,p2,leq(x)))),
                                    Imply(Product(subset(S,p1,q),
                                                  subset(S,p2,q)),
                                          subset(S,leq(x),q))))     SCOTTOP
```

A Constructive Proof of the Heine-Borel Covering Theorem for Formal Reals

Jan Cederquist[1] and **Sara Negri**[2]
[1] Department of Computing Science
University of Göteborg
S-412 96 Göteborg, Sweden
[2] Dipartimento di Matematica Pura ed Applicata
Via Belzoni 7 - 35131 Padova, Italy
Department of Computing, Imperial College
180 Queen's Gate, SW7 2BZ London, U.K.
e-mail: ceder@cs.chalmers.se,
negri@pdmat1.math.unipd.it

Abstract

The continuum is here presented as a formal space by means of a finitary inductive definition. In this setting a constructive proof of the Heine-Borel covering theorem is given.

1 Introduction

It is well known that the usual classical proofs of the Heine-Borel covering theorem are not acceptable from a constructive point of view (cf. [vS, F]). An intuitionistic alternative proof that relies on the fan theorem was given by Brouwer (cf. [B, H]). In view of the relevance of constructive mathematics for computer science, relying on the connection between constructive proofs and computations, it is natural to look for a completely constructive proof of the theorem in its most general form, namely for intervals with real-valued endpoints.

By using formal topology the continuum, as well as the closed intervals of the real line, can be defined by means of finitary inductive definitions. This approach allows a proof of the Heine-Borel theorem that, besides being constructive, can also be completely formalized and implemented on a computer. Formal topology can be expressed in terms of Martin-Löf's type theory; a complete formalization of formal topology in the ALF proof editor has been given in [JC]. A development of mathematical results in formal topology will then be a preliminary work for a complete formalization of these results. On the basis of the present work, the first author has implemented the proof of the Heine-Borel theorem for rational intervals.

Moreover, here as elsewhere (see for instance [C, C2, N, NV]), the use of a pointfree approach allows to replace non-constructive reasoning by constructive proofs.

We point out that a proof similar in spirit to our work was given by Martin-Löf in [ML].

The paper is organized as follows: in Section 2 we provide all the preliminary definitions on formal topology to make the exposition self-contained; in Section 3 the continuum is defined as a formal space by means of an inductive definition, equivalent to the one given in [NS] but more suitable for our purpose. As an aside, the definition provides an explicit description of its Stone compactification (cf. [N]). Formal reals are also proved to be equivalent to real numbers à la Bishop. In the following section, the formal space of a closed interval with rational endpoints is defined. Formal intervals are then proved to coincide, when considered in the extensional way as sets of points, with the usual intervals of the real line. Finally, the Heine-Borel covering theorem is proved and the same is done, without any substantial difference, for intervals with real-valued endpoints.

2 Preliminaries

We recall here the basic theoretical background concerning formal topology. Further general information can be found in [S, SVV], whereas in [N, NV] the constructive character of this approach to topology is testified by applications to constructive pointfree proofs. In [NS], the theory of real numbers in the framework of formal topology is developed, but we also provide here all the definitions needed.

Formal topologies were introduced by Per Martin-Löf and Giovanni Sambin ([S, S1]) as a constructive approach to (pointfree) topology, in the tradition of Johnstone's version of the *Grothendieck topologies* [J] and Fourman and Grayson's *Formal Spaces* [FG], but using simpler technical devices and a constructive set theory based on Martin Löf's constructive type theory.

The definition of a formal topology is obtained by abstracting from the definition of a topological space $\langle X, \Omega(X)\rangle$, without mentioning the points. Since a point-set topology can always be presented using one of its bases, the abstract structure that we will consider is a commutative monoid $\langle S, \cdot_{\mathcal{S}}, 1_{\mathcal{S}}\rangle$ where the set S corresponds to the base of the point-set topology $\Omega(X)$, $\cdot_{\mathcal{S}}$ corresponds to the operation of intersection between basic subsets, and $1_{\mathcal{S}}$ corresponds to the whole collection X.

In a point-set topology any open set is obtained as a union of elements of the base, but union does not make sense if we refuse reference to points; hence we are naturally led to think that an open set may directly correspond to a subset of the set S. Let c^* denote the element of the base which corresponds to the formal basic open c. Since there may be many different subsets of basic elements whose union is the same open set, we need an equivalence relation $\cong_{\mathcal{S}}$ between two subsets U and V of S such that $U \cong_{\mathcal{S}} V$ holds if and only if the opens $U^* \equiv \cup_{a \in U} a^*$ and $V^* \equiv \cup_{b \in V} b^*$ are equal. For this purpose we introduce an infinitary relation $\lhd_{\mathcal{S}}$, called *cover*, between a basic element a of S and a subset U of S whose intended meaning is that $a \lhd_{\mathcal{S}} U$ when $a^* \subseteq U^*$. The conditions we require of this relation are a straightforward rephrasing of the analogous set-theoretic situation.

Besides the notion of cover, we introduce a predicate $Pos_{\mathcal{S}}(a)$ $[a \in S]$ to express positively (that is without using negation) the fact that a basic open is not empty.

Definition 2.1 (Formal topology) *A* formal topology *over a set S is a structure*

$$\mathcal{S} \equiv \langle S, \cdot_{\mathcal{S}}, 1_{\mathcal{S}}, \lhd_{\mathcal{S}}, Pos_{\mathcal{S}}\rangle$$

where $\langle S, \cdot_{\mathcal{S}}, 1_{\mathcal{S}}\rangle$ is a commutative monoid with unit, $\lhd_{\mathcal{S}}$ is a relation, called cover*, between elements and subsets of S such that, for any $a, b \in S$ and $U, V \subseteq S$, the following conditions hold:*

$$(\textit{reflexivity}) \qquad \frac{a \in U}{a \lhd_{\mathcal{S}} U}$$

$$(\textit{transitivity}) \qquad \frac{a \lhd_{\mathcal{S}} U \qquad U \lhd_{\mathcal{S}} V}{a \lhd_{\mathcal{S}} V} \qquad \textit{where} \quad U \lhd_{\mathcal{S}} V \equiv (\forall u \in U)\ u \lhd_{\mathcal{S}} V$$

$$(\cdot\ \textit{- left}) \qquad \frac{a \lhd_{\mathcal{S}} U}{a \cdot_{\mathcal{S}} b \lhd_{\mathcal{S}} U}$$

$$(\cdot\ \textit{- right}) \qquad \frac{a \lhd_{\mathcal{S}} U \qquad a \lhd_{\mathcal{S}} V}{a \lhd_{\mathcal{S}} U \cdot_{\mathcal{S}} V} \qquad \textit{where} \quad U \cdot_{\mathcal{S}} V \equiv \{u \cdot_{\mathcal{S}} v \mid u \in U, v \in V\}$$

and $Pos_{\mathcal{S}}$ *is a predicate on* S*, called* positivity predicate*, satisfying:*

$$(\textit{monotonicity}) \quad \frac{Pos_{\mathcal{S}}(a) \qquad a \lhd_{\mathcal{S}} U}{(\exists b \in U)\ Pos_{\mathcal{S}}(b)}$$

$$(\textit{positivity}) \qquad a \lhd_{\mathcal{S}} \{a\}^{+} \qquad \textit{where} \qquad U^{+} \equiv \{b \in U \mid Pos_{\mathcal{S}}(b)\}\ .$$

All the conditions, except positivity, are a straightforward rephrasing of the preceding intuitive considerations. One reason to introduce positivity is that any non-positive basic open is covered by everything. Indeed, when $Pos_{\mathcal{S}}$ is a decidable predicate, positivity is equivalent to

$$\frac{\neg Pos_{\mathcal{S}}(a)}{a \lhd_{\mathcal{S}} \emptyset}$$

and this will be the case both for the topology of formal reals and for the topology of intervals with rational endpoint. Technically, positivity also allows proof by cases on $Pos_{\mathcal{S}}(a)$ for deductions involving covers (for a detailed discussion cf. [SVV]).

We point out that we can dispense with the unit in the definition of formal topology without any substantial difference in the development of the theory. This choice will be pursued in the sequel.

In order to connect our pointfree approach to classical point-set topology, the notion of point has to be recovered. Since we reverse the usual conceptual order between points and opens, and take the opens as primitive, points will be defined as particular, well behaved, collections of opens. We recall here the definition of a (formal) point of a formal topology:

Definition 2.2 *Let* $\mathcal{A} \equiv \langle S, \cdot, 1, \lhd, Pos\rangle$ *be a formal topology. A subset* α *of* S *is said to be a* formal point *if for all* $a, b \in S$*,* $U \subseteq S$ *the following conditions hold:*

1. $1 \in \alpha$;
2. $\dfrac{a \in \alpha \quad b \in \alpha}{a \cdot b \in \alpha}$;
3. $\dfrac{a \in \alpha \quad a \lhd U}{(\exists b \in U)(b \in \alpha)}$;
4. $\dfrac{a \in \alpha}{Pos(a)}$.

In order to maintain the usual intuition on points, in the sequel we will write $\alpha \Vdash a$ (α forces a, or α is a point in a) in place of $a \in \alpha$. Moreover, when a singleton set occurs we will sometimes omit curly brackets, and write $a \lhd b$ for $a \lhd \{b\}$, and $U \cdot b$ for $U \cdot \{b\}$.

3 The Continuum as a Formal Space

Formal real numbers can be obtained as formal points of a suitable formal topology based on the rationals (cf. [NS]). We are adopting here a somewhat different approach to formal reals in comparison with the one given in [NS]. We have the same monoid operation and positivity predicate, and the covering relations are equivalent, but we dispense with the unit. By this approach we avoid adding top and bottom to the rational numbers. The following definition was proposed by Thierry Coquand in order to make inductive arguments easier. Technically, it is a *finitary inductive definition*, since each rule involved has only finitely many premises (cf. [A]). In fact, we do not need to close under the cover rules. Moreover, as we will see, the definition provides a simple presentation of the Stone compactification for the cover (cf. [N]).

Definition 3.1 *The* formal topology of formal reals *is the structure*

$$\mathcal{R} \equiv \langle Q \times Q, \cdot, \lhd, Pos \rangle \ ,$$

where Q is the set of rational numbers, $S \equiv Q \times Q$ is the Cartesian product. The monoid operation is defined by $(p,q) \cdot (r,s) \equiv (max(p,r), min(q,s))$; the cover $\lhd$ is defined by

$$(p,q) \lhd U \equiv (\forall p', q')(p < p' < q' < q \to (p', q') \lhd_f U) \ ,$$

where the relation $\lhd_f$ is inductively defined by

1. $\dfrac{q \leq p}{(p,q) \lhd_f U}$;
2. $\dfrac{(p,q) \in U}{(p,q) \lhd_f U}$;
3. $\dfrac{(p,s) \lhd_f U \quad (r,q) \lhd_f U \quad p \leq r < s \leq q}{(p,q) \lhd_f U}$;
4. $\dfrac{(p',q') \lhd_f U \quad p' \leq p < q \leq q'}{(p,q) \lhd_f U}$.

The positivity predicate is defined by

$$Pos(p,q) \equiv p < q \ .$$

According to the intuitive set-theoretic reading of the definition of formal topology, the above definition amounts to the following: A basic open (p,q) is covered by a family U of basic opens if and only if all (p',q') strictly included in (p,q) are included in the union of a finite subfamily of U. The rest of this section will be devoted to proving that the above definition really defines a formal topology whose formal points correspond to constructive real numbers.

The usual definition of formal point of a formal topology, given in Section 2, specializes to the following one when considering the formal topology of formal reals $\mathcal{R}$.

Definition 3.2 *A subset α of S is a* formal point of $\mathcal{R}$ *if it satisfies*

1. $(\exists p, q)(\alpha \Vdash (p,q))$;

2. $\dfrac{\alpha \Vdash (p,q) \quad \alpha \Vdash (p',q')}{\alpha \Vdash (p,q)\cdot(p',q')}$;

3. $\dfrac{\alpha \Vdash (p,q) \quad (p,q) \lhd U}{(\exists (p',q') \in U)(\alpha \Vdash (p',q'))}$;

4. $\dfrac{\alpha \Vdash (p,q)}{Pos(p,q)}$.

We observe here that, since $Pos(p,q)$ is decidable, the fourth rule is provable from the third. Let $Pt(\mathcal{R})$ denote the formal points of $\mathcal{R}$, called *formal reals.*

We will now prove that both $\lhd$ and $\lhd_f$ are covers, the latter being the Stone compactification of the former.

Proposition 3.3 *The relation $\lhd_f$ is a cover.*

Proof. Before proving the cover rules for $\lhd_f$, we observe that the rule of $\cdot$ - right follows from the rule of *localization* $\frac{a \lhd U}{a\cdot b \lhd U\cdot b}$ since the base is a semilattice.

Reflexivity: By definition.

Transitivity: Suppose $(p,q) \lhd_f U$ and $U \lhd_f V$. Then it is straightforward by induction on the derivation of $(p,q) \lhd_f U$ that $(p,q) \lhd_f V$.

$\cdot$ - Left: By the fourth axiom since $p \leq max(p,r)$ and $min(q,s) \leq q$.

Localization: Suppose $(p,q) \lhd_f U$. Then we prove, by induction on the derivation of $(p,q) \lhd_f U$, that $(p,q)\cdot(r,s) \lhd_f U\cdot(r,s)$. We first observe that we can assume $r < s$, because if $s \leq r$ the claim follows trivially by the first rule. If $(p,q) \lhd_f U$ is derived by the first or the second axiom the claim is trivial. Suppose it is derived by the third axiom with the assumptions $p \leq t < v \leq q$, $(p,v) \lhd_f U$ and $(t,q) \lhd_f U$. If $s \leq t$ then $min(v,s) = min(q,s)$ and therefore $(p,v)\cdot(r,s) = (p,q)\cdot(r,s)$. From $(p,v) \lhd_f U$, by induction hypothesis, we have $(p,v)\cdot(r,s) \lhd_f U\cdot(r,s)$ thus $(p,q)\cdot(r,s) \lhd_f U\cdot(r,s)$. If $v \leq r$ then $max(t,r) = max(p,s)$ and the conclusion follows as above by applying inductive hypothesis to the premiss $(t,q) \lhd_f U$. Otherwise $max(t,r) < min(v,s)$ and we have, by induction hypothesis and the same rule, $(p,q)\cdot(r,s) \lhd_f U\cdot(r,s)$. If it comes from $(p',q') \lhd_f U$, with $p' \leq p < q \leq q'$, then by induction hypothesis we get $(p',q')\cdot(r,s) \lhd_f U\cdot(r,s)$ and since $max(p',r) \leq max(p,r)$ and $min(q,s) \leq min(q',s)$ we obtain by the same rule $(p,q)\cdot(r,s) \lhd_f U\cdot(r,s)$. □

Moreover we have the following essential result:

Proposition 3.4 *The relation $\lhd_f$ is a Stone cover, i.e., a cover with the property that, for arbitrary $(p,q) \in S$ and $U \subseteq S$, $(p,q) \lhd_f U$ implies the existence of a finite subset U_0 of U such that $(p,q) \lhd_f U_0$.*

Proof. Suppose $(p,q) \lhd_f U$. Then we can find a finite subset U_0 of U such that $(p,q) \lhd_f U_0$ by induction on the derivation of $(p,q) \lhd_f U$. □

The following lemma is used to prove that $\lhd$ is a cover.

Lemma 3.5 *Suppose $(p,q) \lhd_f U$, $U \lhd V$ and let $p < p' < q' < q$. Then $(p',q') \lhd_f V$.*

Proof. By induction on the derivation of $(p,q) \lhd_f U$. If $p \geq q$ and $p < p' < q' < q$ we have $(p',q') \lhd_f U$ by axioms 1 and 4. If $(p,q) \in U$ then by the assumption $U \lhd V$ we have $(p,q) \lhd V$ and therefore if $p < p' < q' < q$, $(p',q') \lhd_f V$. If $p \leq r < s \leq q$, $(p,s) \lhd_f U$

and $(r, q) \lhd_f U$ we distinguish two cases according to the position of r, s with respect to p', q'. In the first case $r < p'$ or $q' < s$, in the second $p' \leq r < s \leq q'$. Suppose $r < p'$, then $r < p' < q' < q$ so from the assumptions $(r, q) \lhd_f U$ and $U \lhd V$ we get, by induction hypothesis, $(p', q') \lhd_f V$. If $q' < s$ we conclude symmetrically. If $p' \leq r < s \leq q'$ we can find r', s' such that $r < r' < s' < s$. Therefore we have $p < p' < s' < s$ and $r < r' < q' < q$. By induction hypothesis the former, together with $(p, s) \lhd_f U$ and $U \lhd V$ gives $(p', s') \lhd_f V$ and the latter together with $(r, q) \lhd_f U$ and $U \lhd V$ gives $(r', q') \lhd_f V$. Since $p' \leq r' < s \leq q'$ we get the conclusion $(p', q') \lhd_f V$. If $(p, q) \lhd_f U$ is derived by the fourth rule we just apply induction hypothesis to the premiss and the fourth rule again. □

Proposition 3.6 *The relation* $\lhd$ *is a cover.*

Proof. Reflexivity: Let $(p, q) \in U$, then $(p, q) \lhd_f U$ and so if $p < p' < q' < q$ we have $(p', q') \lhd_f U$. Therefore $(p, q) \lhd U$.

Transitivity: Let $p < p' < q' < q$. Then there exist p'' and q'' such that $p < p'' < p' < q' < q'' < q$ and $(p'', q'') \lhd_f U$. By the lemma above we have $(p', q') \lhd_f V$ and therefore $(p, q) \lhd V$.

$\cdot$ - Left: Suppose $(p, q) \lhd U$, then $(p, q) \cdot (r, s) \lhd U$ follows directly from the definitions since $max(p, r) < p' < q' < min(q, s)$ implies $p < p' < q' < q$.

$\cdot$ - Right: Straightforward from the validity of $\cdot$ - right for $\lhd_f$. □

Finally, it is straightforward to prove monotonicity and positivity for Pos, thus completing the proof that $\mathcal{R}$ is a formal topology.

We will now prove that the cover $\lhd_f$ is the Stone compactification of the cover $\lhd$. We point out that this result is not needed in the proof of the Heine-Borel theorem.

Proposition 3.7 *If* $(p, q) \lhd U$ *and* U *is finite, then* $(p, q) \lhd_f U$.

Before proving Proposition 3.7, observe we can assume that, for all $(r, s) \in U$, $Pos((p, q) \cdot (r, s))$ holds. In fact, if this is not the case, from $(p, q) \lhd U$ we have $(p, q) \lhd ((p, q) \cdot U)^+$, and from $(p, q) \lhd_f ((p, q) \cdot U)^+$, by $\cdot$ - left and transitivity, $(p, q) \lhd_f U$. The following lemmas will allow a proof of Proposition 3.7 by induction on the number of elements of U.

Lemma 3.8 *For positive* (p, q), $(p, q) \lhd_f (r, s)$ *implies* $r \leq p < q \leq s$.

Proof. By induction on the derivation of $(p, q) \lhd_f (r, s)$. If $(p, q) \lhd_f (r, s)$ is derived by the first or the second axiom, the claim holds trivially. If it is derived by the third axiom from $p \leq u < v \leq q$, $(p, v) \lhd_f (r, s)$, $(u, q) \lhd_f (r, s)$, then by induction hypothesis we have $r \leq p < v \leq s$, $r \leq u < q \leq s$ and therefore $r \leq p < q \leq s$. If it follows from $p' \leq p < q \leq q'$ and $(p', q') \lhd_f (r, s)$ by the fourth axiom, then by induction hypothesis $r \leq p' < q' \leq s$ and therefore $r \leq p < q \leq s$. □

Corollary 3.9 $(p, q) \lhd (r, s)$ *implies* $(p, q) \lhd_f (r, s)$.

Proof. Let $(p, q) \lhd (r, s)$. Then, for all p', q' such that $p < p' < q' < q$, we have $r \leq p' < q' \leq s$, and therefore $r \leq p < q \leq s$, hence $(p, q) \lhd_f (r, s)$. □

Lemma 3.10 *Suppose that* $p < q$ *and* $(p, q) \lhd U$, *where* U *is finite and for all* $(r, s) \in U$, $Pos((p, q), (r, s))$ *holds. Then there exists* $(p_1, q_1) \in U$ *such that* $p_1 \leq p < q_1$.

Proof. Let (p_1, q_1) be an element of U such that p_1 is the smallest (with respect to the usual order of the rational numbers) of all the first projections of elements of U. Then $p_1 \leq p$. In fact, for all $(p', q') \in U$, $p_1 \leq max(p', p) < min(q', q) \leq q$, that implies $U \cdot (p, q) \lhd_f (p_1, q)$. Since $(p, q) \lhd U \cdot (p, q)$, we have by transitivity $(p, q) \lhd (p_1, q)$, and therefore, by Corollary 3.9 and Lemma 3.8, we get $p_1 \leq p < q$. Then, by the assumption that for, all $(r, s) \in U$, $Pos((p, q) \cdot (r, s))$ holds, we have $p_1 \leq p < q_1$. □

Lemma 3.11 *Suppose that $(p, q) \lhd_f U$, and let $p < u < q$. Then there exists $(r, s) \in U$ such that $r < u < s$.*

Proof. Straightforward by induction on the derivation of $(p, q) \lhd_f U$. □

Corollary 3.12 *Suppose that $(p, q) \lhd U$, and let $p < u < q$. Then there exists $(r, s) \in U$ such that $r < u < s$.*

Proof. If $p < u < q$, there exist p', q' such that $p < p' < u < q' < q$ and therefore $(p', q') \lhd_f U$. Then the conclusion follows by Lemma 3.11. □

Lemma 3.13 *Suppose that $(p, q) \lhd U$, and let $(r, s) \in U$ with $\neg Pos((p, q) \cdot (r, s))$. Then $(p, q) \lhd U \setminus \{(r, s)\}$.*

Proof. From $(p, q) \lhd U$ we have, by positivity and $\cdot$ - right, $(p, q) \lhd (U \cdot (p, q))^+$. Since $\neg Pos((p, q) \cdot (r, s))$ holds, we have $(U \cdot (p, q))^+ \subseteq (U \setminus \{(r, s)\}) \cdot (p, q)$ and therefore $(p, q) \lhd (U \setminus \{(r, s)\}) \cdot (p, q)$, thus a fortiori $(p, q) \lhd U \setminus \{(r, s)\}$. □

Proof of Proposition 3.7. The proof is by induction on the number of elements of U. If $U = \{(r, s)\}$ the claim follows by Corollary 3.9. Suppose the result holds for $|U| = n$ and suppose that $(p, q) \lhd U_{n+1}$, where $|U_{n+1}| = n+1$. By Lemma 3.10 there exists $(p_1, q_1) \in U_{n+1}$ such that $p_1 \leq p < q_1$. If $q \leq q_1$ then $p_1 \leq p < q \leq q_1$ and therefore $(p, q) \lhd_f (p_1, q_1)$, so by reflexivity and transitivity $(p, q) \lhd_f U_{n+1}$. Otherwise $q_1 < q$, hence by Corollary 3.12 there exists $(p_2, q_2) \in U_{n+1}$ such that $p_2 < q_1 < q_2$. So we can find r, s such that $q_1 < r < s < q_2$. Since $p \leq r$ and $(p, q) \lhd U_{n+1}$, $(r, q) \lhd U_{n+1}$. From $q_1 < r$, we have $\neg Pos((r, q) \cdot (p_1, q_1))$ and therefore, by Lemma 3.13, we have $(r, q) \lhd U_{n+1} \setminus \{(p_1, q_1)\}$, so that by induction hypothesis $(r, q) \lhd_f U_{n+1} \setminus \{(p_1, q_1)\}$. Then a fortiori $(r, q) \lhd_f U_{n+1}$. Since $(p, s) \lhd_f \{(p_1, q_1), (p_2, q_2)\}$, we also have $(p, s) \lhd_f U_{n+1}$ and therefore $(p, q) \lhd_f U_{n+1}$. □

We conclude this section with observing that formal reals offer an alternative approach to constructive analysis; they have been used in the treatment of the Hahn-Banach theorem (cf. [CCN]) and of the Cantor and Baire theorems (cf. [N1], [NS]). Moreover, we can show that they are equivalent to real numbers à la Bishop. First we recall the following (cf. [Bi]):

Definition 3.14 *A* real number *is a sequence of rational numbers $(x_n)_n$ such that*

$$|x_m - x_n| \leq m^{-1} + n^{-1} \qquad (m, n \in N^+) \ .$$

Two real numbers, $(x_n)_n$ and $(y_n)_n$, are equal if

$$|x_n - y_n| \leq 2n^{-1} \qquad (n \in N^+) \ .$$

We have:

Proposition 3.15 *There exists a bijective correspondence between formal reals and real numbers à la Bishop.*

Proof. Let α be a formal real. By the rules in Definition 3.2, α contains arbitrarily small intervals, in particular (p,q) with $q-p \leq 2/3$. Since $\frac{2x+y}{3} < \frac{x+2y}{3}$ again by the rules in Definition 3.2, $\alpha \Vdash (x,y)$ implies $\alpha \Vdash (x, \frac{x+2x}{3}) \ \vee \ \alpha \Vdash (\frac{2x+y}{3}, y)$. Now we can recursively generate a sequence of intervals $((x_n, y_n))_n$, by case-analysis:

$$\begin{array}{rcl} (x_1, y_1) & \equiv & (p,q) \\ (x_{i+1}, y_{i+1}) & \equiv & \begin{cases} (x_i, \frac{x_i+2y_i}{3}) & \text{if } \alpha \Vdash (x_i, \frac{x_i+2y_i}{3}) \\ (\frac{2x_i+y_i}{3}, y_i) & \text{if } \alpha \Vdash (\frac{2x_i+y_i}{3}, y_i) \ . \end{cases} \end{array}$$

It can be verified that the sequences $(x_n)_n$ and $(y_n)_n$ are real numbers according to Definition 3.14.

Conversely, if $(x_n)_n$ is a real number à la Bishop, then the set defined by

$$\alpha \equiv \bigcup_{n \in N^+} \{(p,q) : p < x_n - 2/n < x_n + 2/n < q\}$$

is a formal real.

Moreover the correspondence thus established is bijective. $\Box$

4 The Formal Space $[a, b]$

Given two rational numbers a, b such that $a < b$, we will define a formal space whose formal points are the formal points of $\mathcal{R}$ between a and b. We will follow the standard way to build, from an open U of a space X, a space classically corresponding to the closed subspace $X \backslash U$. Indeed, we will define a cover relation $\lhd_{[a,b]}$ and the intended meaning of $(p,q) \lhd_{[a,b]} U$ is that the part of (p,q) inside the closed interval $[a,b]$ is covered by U. By classical set-theoretic reasoning we have that $(p,q) \cap [a,b] \subseteq \cup U$ is the same as

$$(p,q) \subseteq (\cup U) \cup \{(r,a) \mid r < a\} \cup \{(b,s) \mid b < s\} \ .$$

An interval (p,q) is then positive in the space $[a,b]$ iff the part of (p,q) inside $[a,b]$ is positive. This justifies the following:

Definition 4.1 *Let* $\mathcal{R} \equiv \langle Q \times Q, \cdot, \lhd, Pos \rangle$ *be the formal topology of formal reals and let* $[a,b]$ *be defined by*

$$[a,b] \equiv \langle Q \times Q, \cdot, \lhd_{[a,b]}, Pos_{[a,b]} \rangle$$

where the relation $\lhd_{[a,b]}$ *is defined by*

$$(p,q) \lhd_{[a,b]} U \equiv (p,q) \lhd U \cup \{(r,a) \mid r < a\} \cup \{(b,s) \mid b < s\} \ ,$$

and the predicate $Pos_{[a,b]}$ *is defined by*

$$Pos_{[a,b]}(p,q) \equiv Pos((p,q) \cdot (a,b)) \ .$$

In the sequel we will use the notation $\mathcal{C}[a,b]$ for $\{(r,a) \mid r < a\} \cup \{(b,s) \mid b < s\}$ and we will understand $\mathcal{C}[a,b]$ as the complement of $[a,b]$.

By the following proposition and by the immediate verification that $Pos_{[a,b]}$ is a positivity predicate, the above does indeed define a formal topology.

Proposition 4.2 *The relation* $\lhd_{[a,b]}$ *is a cover.*

The proposition follows from the following lemma:

Lemma 4.3 *Let* $\lhd$ *be a cover on the base* S *and let* $V \subseteq S$. *Then the relation* $\lhd_V$ *defined by*

$$a \lhd_V U \equiv a \lhd U \cup V$$

is a cover.

Proof. Reflexivity, transitivity, $\cdot$ - left are straightforward, and $\cdot$ - right follows from the fact that in general $(U \cup V) \cdot (W \cup V) \lhd (U \cdot W) \cup V$. $\square$

As in Section 3, the general definition of formal point of a formal topology can be specialized to $[a, b]$:

Definition 4.4 *A subset* α *of* S *is a* formal point of $[a, b]$ *if it satisfies*

1. $(\exists p, q)(\alpha \Vdash (p, q))$;

2. $\dfrac{\alpha \Vdash (p, q) \quad \alpha \Vdash (p', q')}{\alpha \Vdash (p, q) \cdot (p', q')}$;

3. $\dfrac{\alpha \Vdash (p, q) \quad (p, q) \lhd_{[a,b]} U}{(\exists (p', q') \in U)(\alpha \Vdash (p', q'))}$;

4. $\dfrac{\alpha \Vdash (p, q)}{Pos_{[a,b]}(p, q)}$.

As was the case in Definition 3.2 the fourth rule is provable from the third, since $Pos_{[a,b]}(p, q)$ is decidable. We will denote with $Pt([a, b])$ the collection of formal points of $[a, b]$, called *formal reals of the interval* $[a, b]$.

We recall here the definition of order for $Pt(\mathcal{R})$ (cf. [NS]):

$$\alpha < \beta \equiv (\exists (p, q), (r, s) \in S)(\alpha \Vdash (p, q) \ \& \ \beta \Vdash (r, s) \ \& \ q < r) \ ;$$

$$\alpha \leq \beta \equiv \neg(\beta < \alpha) \ .$$

Let $\bar{a}$ denote the formal point $\{(p, q) \mid p < a < q\}$, corresponding to the rational a. Then we have $\alpha < \bar{a} \Leftrightarrow (\exists (p, q) \in S)(\alpha \Vdash (p, q) \ \& \ q < a)$.

The following proposition says that the formal space $[a, b]$ really corresponds to the closed interval $[a, b]$, i.e., the definition of the formal space $[a, b]$ is correct:

Proposition 4.5 $\alpha \in Pt([a, b]) \Leftrightarrow \alpha \in Pt(R) \ \& \ \bar{a} \leq \alpha \leq \bar{b}$.

Proof. $\Rightarrow$: Let $\alpha \in Pt([a, b])$. It is immediate that $\alpha \in Pt(R)$ since $(p, q) \lhd U$ implies $(p, q) \lhd_{[a,b]} U$. To show that $\bar{a} \leq \alpha$, suppose $\alpha < \bar{a}$. Then by definition $(\exists (p, q) \in S)(\alpha \Vdash (p, q) \ \& \ q < a)$ and therefore $\neg Pos_{[a,b]}(p, q)$, against the assumption. Hence $\bar{a} \leq \alpha$. The inequality $\alpha \leq \bar{b}$ is proved symmetrically.

$\Leftarrow$: Let $\alpha \in Pt(R) \ \& \ \bar{a} \leq \alpha \leq \bar{b}$. Clauses 1 and 2 are obvious.

3. Let $\alpha \Vdash (p, q)$ and $(p, q) \lhd_{[a,b]} U$. Then there exists $(r, s) \in U \cup \mathcal{C}[a, b]$ such that $\alpha \Vdash (r, s)$. Since $Pos_{[a,b]}(r, s)$ holds, it cannot be $(r, s) = (r, a)$ or $(r, s) = (b, s)$ and therefore $(r, s) \in U$. $\square$

5 The Heine-Borel Covering Theorem for $[a, b]$

Here we will prove the Heine-Borel covering theorem asserting that any open cover of a closed and bounded interval has a finite sub-cover. We will use the notation $[a, b] \lhd_{[a,b]} U$ for $(\forall p, q)((p, q) \lhd_{[a,b]} U)$, meaning that U covers the whole space $[a, b]$.

Theorem 5.1 *The formal space* $[a, b]$ *is compact, i.e.*

$$[a, b] \lhd_{[a,b]} U \Rightarrow (\exists U_0 \subseteq_\omega U)([a, b] \lhd_{[a,b]} U_0) \ .$$

The proof uses the following lemma:

Lemma 5.2 $[a, b] \lhd_{[a,b]} U \Leftrightarrow (\exists r, s)(r < a < b < s \ \& \ (r, s) \lhd_f U \cup \mathcal{C}[a, b])$.

Proof. $\Rightarrow$: By the hypothesis $[a, b] \lhd_{[a,b]} U$, in particular there exist p, q such that $p < a < b < q$ and $(p, q) \lhd_{[a,b]} U$. Then by definition

$$(\forall p', q')(p < p' < q' < q \rightarrow (p', q') \lhd_f U \cup \mathcal{C}[a, b]) \ .$$

By choosing r and s such that $p < r < a < b < s < q$, we can thus conclude $(\exists r, s)(r < a < b < s \ \& \ (r, s) \lhd_f U \cup \mathcal{C}[a, b])$.

$\Leftarrow$: Observe that $(r, s) \lhd_f U \cup \mathcal{C}[a, b]$ implies $(r, s) \lhd U \cup \mathcal{C}[a, b]$, that is $(r, s) \lhd_{[a,b]} U$. If $r < a < b < s$, for all (p, q), $(p, q) \lhd \{(p, a), (r, s), (b, q)\}$ holds, and therefore $(p, q) \lhd_{[a,b]} \{(r, s)\}$. The claim follows by transitivity of $\lhd_{[a,b]}$. $\square$

Proof of Theorem 5.1. Suppose $[a, b] \lhd_{[a,b]} U$. Then by Lemma 5.2 there exists r and s such that $r < a < b < s \ \& \ (r, s) \lhd_f U \cup \mathcal{C}[a, b]$ and by Proposition 3.4 there exists a finite subset W_0 of $U \cup \mathcal{C}[a, b]$ such that $(r, s) \lhd_f W_0$. Now, since W_0 is a finite subset of $U \cup \mathcal{C}[a, b]$, we can find a finite subset U_0 of U such that $W_0 \subseteq_\omega U_0 \cup \mathcal{C}[a, b]$. We get $(r, s) \lhd_f U_0 \cup \mathcal{C}[a, b]$. So, by Lemma 5.2 again, $[a, b] \lhd_{[a,b]} U_0$. $\square$

6 The Formal Space $[\alpha, \beta]$

Generalizing the formal space $[a, b]$ that corresponds to an interval with rational endpoints, we will define the formal space $[\alpha, \beta]$, with α and β formal reals with $\alpha < \beta$, that corresponds to an interval with real endpoints. The cover for the formal space $[\alpha, \beta]$ is defined starting from $\lhd$, similarly to the cover for $[a, b]$:

Definition 6.1 *Let* $\lhd_{[\alpha,\beta]}$ *be the relation defined by*

$$(p, q) \lhd_{[\alpha,\beta]} U \ \equiv (p, q) \lhd U \cup \mathcal{C}[\alpha, \beta] \ ,$$

where $\mathcal{C}[\alpha, \beta] \equiv \{(r, a) \mid r < a < \alpha\} \cup \{(b, s) \mid \beta < b < s\}$.

Proposition 6.2 *The relation* $\lhd_{[\alpha,\beta]}$ *is a cover.*

The proof is immediate by Lemma 4.3.

Definition 6.3 *A subset* γ *of* S *is a* formal point *of* $[\alpha, \beta]$ *if it satisfies*

1. $(\exists p, q)(\gamma \Vdash (p, q))$;

2. $\dfrac{\gamma \Vdash (p,q) \quad \gamma \Vdash (p',q')}{\gamma \Vdash (p,q)\cdot(p',q')}$;

3. $\dfrac{\gamma \Vdash (p,q) \quad (p,q) \lhd_{[\alpha,\beta]} U}{(\exists (p',q') \in U)(\gamma \Vdash (p',q'))}$;

4. $\dfrac{\gamma \Vdash (p,q)}{p < q \ \& \ \alpha < \bar{q} \ \& \ \bar{p} < \beta}$.

We remark that the property $p < q \ \& \ \alpha < \bar{q} \ \& \ \bar{p} < \beta$ of the basic neighbourhood (p,q) expresses the fact that (p,q) has positive intersection with the interval $[\alpha,\beta]$. Nevertheless, we do not call it a positivity predicate, since the property of positivity does not seem to be constructively valid for this predicate.

The collection of formal points of $[\alpha,\beta]$ will be denoted $Pt([\alpha,\beta])$. As in the case of the formal space $[a,b]$ we have:

Proposition 6.4 $\gamma \in Pt([\alpha,\beta]) \ \Leftrightarrow \ \gamma \in Pt(R) \ \& \ \alpha \leq \gamma \leq \beta$.

Proof. $\Rightarrow$: If $\gamma \in Pt([\alpha,\beta])$ it is immediate to show $\gamma \in Pt(R)$ since $(p,q) \lhd U$ implies $(p,q) \lhd_{[\alpha,\beta]} U$. Now suppose $\gamma < \alpha$. Then by definition

$$(\exists (p_\gamma,q_\gamma),(p_\alpha,q_\alpha) \in S)(\gamma \Vdash (p_\gamma,q_\gamma) \ \& \ \alpha \Vdash (p_\alpha,q_\alpha) \ \& \ q_\gamma < p_\alpha) \ .$$

From $\gamma \Vdash (p_\gamma,q_\gamma)$, by the fourth rule, we obtain that $\alpha < \bar{q}_\gamma$ which contradicts $\bar{q}_\gamma < \bar{p}_\alpha < \alpha$. Hence $\alpha \leq \gamma$. We obtain $\gamma \leq \beta$ symmetrically.

$\Leftarrow$: 1 and 2 are direct.

3. Let $\gamma \Vdash (p,q)$ and $(p,q) \lhd_{[\alpha,\beta]} U$. By definition we have $(p,q) \lhd U \cup \mathcal{C}[\alpha,\beta]$ and by the third rule for $Pt(\mathcal{R})$ we get $(\exists (p',q') \in U \cup \mathcal{C}[\alpha,\beta])(\gamma \Vdash (p',q'))$. If $\gamma \Vdash (p',q')$, by the fourth rule for $Pt([\alpha,\beta])$ (which is proved below), $\alpha < \bar{q}'$ and $\bar{p}' < \beta$ and therefore $(p',q') \in U$. Hence $(\exists (p',q') \in U)(\gamma \Vdash (p',q'))$.

4. Let $\gamma \Vdash (p,q)$. Then by the fourth rule for $Pt(\mathcal{R})$ we have $p < q$. If $\gamma \Vdash (p,q)$ we also have $\gamma < \bar{q}$ and since $\alpha < \gamma$ we get $\alpha < \bar{q}$. The inequality $\bar{p} < \beta$ is proved symmetrically. □

7 The Heine-Borel Covering Theorem for $[\alpha,\beta]$

Here we will prove the Heine-Borel covering theorem for closed intervals with real-valued endpoints. We introduce the notation:

$$[\alpha,\beta] \lhd_{[\alpha,\beta]} U \equiv (\forall p,q)((p,q) \lhd_{[\alpha,\beta]} U) \ .$$

Theorem 7.1 *The formal space $[\alpha,\beta]$ is compact, i.e.*

$$[\alpha,\beta] \lhd_{[\alpha,\beta]} U \Rightarrow (\exists U_0 \subseteq_\omega U)([a,b] \lhd_{[\alpha,\beta]} U_0) \ .$$

The proof uses the following lemma:

Lemma 7.2 $[\alpha,\beta] \lhd_{[\alpha,\beta]} U \Leftrightarrow (\exists r,s)(\bar{r} < \alpha < \beta < \bar{s} \ \& \ (r,s) \lhd_f U \cup \mathcal{C}[\alpha,\beta])$.

Proof. $\Rightarrow$: Given $[\alpha, \beta] \lhd_{[\alpha,\beta]} U$, there exist p, q such that $\bar{p} < \alpha < \beta < \bar{q}$ and $(p, q) \lhd_{[\alpha,\beta]} U$. By definition

$$(\forall p', q')(p < p' < q' < q \rightarrow (p', q') \lhd_f U \cup \mathcal{C}[\alpha, \beta]) .$$

Now we can choose r, s such that $\bar{p} < \bar{r} < \alpha < \beta < \bar{s} < \bar{q}$. Hence we obtain $(\exists r, s)(\bar{r} < \alpha < \beta < \bar{s} \;\&\; (r, s) \lhd_f U \cup \mathcal{C}[\alpha, \beta])$.

$\Leftarrow$: Choose (r, s) such that $\bar{r} < \alpha < \beta < \bar{s}$ and $(r, s) \lhd_f U \cup \mathcal{C}[\alpha, \beta]$. For any a, b with $\bar{r} < \bar{a} < \alpha < \beta < \bar{b} < \bar{s}$ we get, for all (p, q), $(p, q) \lhd_f \{(p, a), (r, s), (b, q)\}$. We have $(p, a) \lhd_f U \cup \mathcal{C}[\alpha, \beta]$ because if $p < a$ then $(p, a) \in U \cup \mathcal{C}[\alpha, \beta]$ otherwise $(p, a) \lhd_f U \cup \mathcal{C}[\alpha, \beta]$ by axiom. By symmetry we have $(b, q) \lhd_f U \cup \mathcal{C}[\alpha, \beta]$, and therefore, by transitivity, $(p, q) \lhd_f U \cup \mathcal{C}[\alpha, \beta]$. This also means that $(p, q) \lhd_{[\alpha,\beta]} U$ and, since (p, q) is arbitrary, $[\alpha, \beta] \lhd_{[\alpha,\beta]} U$. $\Box$

Proof of Theorem 7.1. Suppose $[\alpha, \beta] \lhd_{[\alpha,\beta]} U$. Then, by Lemma 7.2, there exist r and s such that $\bar{r} < \alpha < \beta < \bar{s} \;\&\; (r, s) \lhd_f U \cup \mathcal{C}[\alpha, \beta]$ and by Proposition 3.3 there exists a finite subset W_0 of $U \cup \mathcal{C}[a, b]$ such that $(r, s) \lhd_f W_0$. Then we can find a finite subset U_0 of U such that $W_0 \subseteq_\omega U_0 \cup \mathcal{C}[\alpha, \beta]$ and we get $(r, s) \lhd_f U_0 \cup \mathcal{C}[\alpha, \beta]$. Using Lemma 7.2 again, $[\alpha, \beta] \lhd_{[\alpha,\beta]} U_0$. $\Box$

8 Acknowledgements

We wish to thank Thierry Coquand and Jan Smith for helpful suggestions and remarks.

References

[A] P. Aczel. *An Introduction to Inductive Definitions*, in *Handbook of Mathematical Logic*, J. Barwise ed., North-Holland (1977) 739–782.

[B] L. E. J. Brouwer. *Die intuitionistische Form des Heine-Borelschen Theorems*, in *L. E. J. Brouwer Collected Works*, A. Heyting ed., North-Holland, Amsterdam (1975) vol. 1, 350–351, 1926C.

[Bi] E. Bishop. “Foundations of Constructive Analysis”, Mc Graw Hill, 1967.

[JC] J. Cederquist. *A machine assisted formalization of pointfree topology in type theory*, Chalmers University of Technology and University of Göteborg, Sweden, Licentiate Thesis (1994).

[CCN] J. Cederquist, T. Coquand, S. Negri *Helly-Hahn-Banach in formal topology*, forthcoming.

[C] T. Coquand. *An intuitionistic proof of Tychonoff's theorem*, The Journal of Symbolic Logic vol. 57, no. 1 (1992) 28–32.

[C2] T. Coquand. *Constructive Topology and Combinatorics*, proceeding of the conference Constructivity in Computer Science, San Antonio, LNCS 613 (1992) 159–164.

[FG] M. P. Fourman, R.J. Grayson. *Formal Spaces*, in “The L. E. J. Brouwer Centenary Symposium”, A. S. Troelstra and D. van Dalen (eds), North-Holland, Amsterdam (1982) 107–122.

[F] M. Franchella. "L. E. J. Brouwer pensatore eterodosso. L'intuizionismo tra matematica e filosofia", Guerini Studio (1994).

[H] A. Heyting. "Intuitionism, an introduction", North-Holland (1971).

[J] P. T. Johnstone. "Stone Spaces", Cambridge University Press (1982).

[M] L. Magnusson, "The Implementation of ALF - a Proof Editor based on Martin-Löf's Monomorphic Type Theory with Explicit Substitution", Chalmers University of Technology and University of Göteborg, PhD thesis (1995).

[ML] P. Martin-Löf. "Notes on Constructive Mathematics", Almqvist & Wiksell, Stockholm (1970).

[ML1] P. Martin-Löf. "Intuitionistic type theory", notes by Giovanni Sambin of a series of lectures given in Padua, June 1980, Bibliopolis, Napoli (1984).

[N] S. Negri. *Stone bases, alias the constructive content of Stone representation*, "Logic and Algebra", A. Ursini and P. Aglianò eds., Dekker, New York (1996) 617–636.

[N1] S. Negri. "Dalla topologia formale all'analisi", Ph.D. thesis, University of Padua (1996).

[NS] S. Negri, D. Soravia. *The continuum as a formal space*, Rapporto Interno n.4, 17-7-95, Dipartimento di Matematica Pura e Applicata, Università di Padova.

[NV] S. Negri, S. Valentini. *Tychonoff's theorem in the framework of formal topologies*, The Journal of Symbolic Logic (in press).

[NPS] B. Nordström, K. Peterson, J. Smith, "Programming in Martin-Löf's Type Theory", Oxford University Press (1990).

[S] G. Sambin. *Intuitionistic formal spaces – a first communication*, in Mathematical logic and its applications, D. Skordev ed., Plenum (1987) 187–204.

[S1] G. Sambin. *Intuitionistic formal spaces and their neighbourhoods*, in "Logic Colloquium 88", R. Ferro et al. eds., North-Holland, Amsterdam (1989) 261–285.

[SVV] G. Sambin, S. Valentini, P. Virgili. *Constructive Domain Theory as a branch of Intuitionistic Pointfree Topology*, Theoretical Computer Science (in press).

[vS] W. P. van Stigt. "Brouwer's Intuitionism", Studies in the History and Philosophy of Mathematics, vol. 2, North-Holland (1990).

An implementation of the Heine-Borel covering theorem in Type Theory

Jan Cederquist
Department of Computing Science
University of Göteborg
S-412 96 Göteborg, Sweden
e-mail: ceder@cs.chalmers.se

Abstract

We describe an implementation, in type theory, of a pointfree proof of the Heine-Borel covering theorem for intervals with rational endpoints.

1 Introduction

The proof presented here is a complete formalisation of the proof presented in "*A constructive proof of the Heine-Borel covering theorem for formal reals*" [CN]. We describe an implementation, in type theory, of a pointfree proof of the Heine-Borel covering theorem for intervals with rational endpoints. The implementations also contain a definition of formal spaces as a type, and definitions of the continuum and the closed rational interval as instances of that type.

The paper is organised as follows: in section 2 we describe the proof-checker Half, in which the implementation has been done, and the type theory it is based on. The rest of the paper is devoted to formal definitions and the proof of the Heine-Borel covering theorem. In section 3 some general definitions are given. In section 4 we define a general formal topology, we also define the notion of compact space and Stone space. Then the rational numbers are are defined as an object of an abstract data type. In section 6 the continuum is defined as a formal space and some of its properties are proved. Then the closed rational interval [a,b] is defined as a formal space and compactness of this space is proved.

In order to make this paper readable, we concentrate on the definitions and many proofs (or lemmas and definitions used in these proofs) are left out. In the code these omitted proofs are replaced by the ellipsis ....[1] However, all identifiers used in the proofs presented are defined and the main theorem is given with all details.

2 Description of the proof-checker Half

The implementation has been done in the proof-checker Half, developed by Thierry Coquand, using a type-checker and an `emacs`-interface implemented by Dan Synek.

[1]The complete proofs are obtainable from the URL:
`ftp://ftp.cs.chalmers.se/pub/users/ceder/heineb/hb.tar`.

The Half system is a successor to ALF [M]. It is a logical framework based on Martin-Löf's polymorphic type theory with one universe [ML], extended by a *theory* mechanism (similar to the theory mechanism in PVS [OSR]) and *let-expressions* (cf. [C, Br, Ba]).

The system has three levels; **Set**, **Type** and **Kind**. **Set** is an element and a subset of **Type**. Elements can be formed in both **Set** and **Type**; both **Set** and **Type** are closed under function types (Π-types) and disjoint union (Σ-types) and allow recursive definitions. There is also a type **Theory** for theories. **Kind** consists of the types **Set**, **Type** and **Theory**, and function types.

A proof (program) in Half consists of a list of definitions and proofs, having the form $f(x_1 : T_1, \ldots, x_n : T_n) = e : T$, where the type T_i may depend on the parameters $x_1, \ldots, x_{i-1}$ and e is an expression of type T.

The Π-type is used for expressing dependent function spaces. Given two types A and B, the Π-type for functions from A to B is written $(x : A) \to B$. Elements of $(x : A) \to B$ are functions $\lambda x \to e$, where the abstracted variable x has type A and e is an expression of type B. The elimination form for elements of Π-types is application.

A recursive data type is defined using the reserved word **data**:

$$\begin{array}{l}\mathbf{data}\{\\ \quad c_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}),\\ \qquad\qquad \vdots\\ \quad c_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk})\},\end{array}$$

where A_{ij} is an arbitrary type. Elements are introduced using the constructors c_i

$$c_i\ a_{i1} \cdots a_{ij}$$

and the elimination form, for objects of a recursively defined data type, is the *case-expression*

$$\begin{array}{l}\mathbf{case}\ x\ \mathbf{of}\ \{\\ \quad c_1\ a_{11} \cdots a_{1m} \to e_1,\\ \qquad\quad \vdots\\ \quad c_n\ a_{n1} \cdots a_{nn} \to e_n\},\end{array}$$

where $e_1, \ldots, e_n$ are expressions of the same type (the type of the case-expression). For example, the set of finite lists may be defined by

$$\mathit{list}(A : \mathbf{Set}) = \mathbf{data}\{Nil, Cons(x : A,\ xs : \mathit{list}\ A)\} : \mathbf{Set}$$

and a list can then be analysed using a *case-expression* as in the following definition of append:

$$\begin{array}{l}\mathit{append}(A : \mathbf{Set},\ l_1 : \mathit{list}\ A,\ l_2 : \mathit{list}\ A) =\\ \quad \mathbf{case}\ l_1\ of\ \{\\ \qquad Nil \to l_2,\\ \qquad Cons\ x\ xs \to Cons\ x\ (append\ A\ xs\ l_2)\} : \mathit{list}\ A.\end{array}$$

Note that, using these recursive definitions on functional form, non-linear inductive types cannot be defined, i.e. dependencies between the parameters cannot be introduced. It turned out that pattern matching together with non-linear inductive definitions is a non-conservative extension of Martin-Löf's type theory (see [H]). The approach taken in Half is to allow only linear inductive definitions. As a consequense, the *Id*-type

$$\frac{a \in A}{id(A, a) \in Id(A, a, a)}$$

is not definable: without dependencies between the parameters there is no way of saying that the two elements are the same. Therefore, for abstract sets, instead of working with sets and the *Id*-type, we work in a more general setting using *setoids*, i.e. sets with equivalence relations. For concrete sets, equalities are explicitly defined. This is also closer to the usual mathematical approach where a set comes together with an equality relation.

A Σ-type is a dependent record $\textbf{sig}\{t_1 : T_1, \ldots, t_n : T_n\}$, where the type T_i may depend on $t_1, \ldots, t_{i-1}$. An object of a Σ-type is formed by constructing objects of the types T_i, $\textbf{struct}\{t_1 = e_1, \ldots, t_n = e_n\}$, where e_i is an expression of type T_i. The elimination rule for Σ-types is projection; if M is of type $\textbf{sig}\{t_1 : T_1, \ldots, t_n : T_n\}$, the value of its i'th component is accessed by $M.t_i$.

Adding Σ-types to the system is a conservative extension of the system; it does not affect the strength of the theory, equivalent definitions can always be obtained using recursive definitions with one constructor. However, to analyse objects of a recursively defined set, case-analysis is required, even if there is only one case to consider.

Theories are lists of definitions and proofs:

$$\begin{array}{l} th = \textbf{theory}\{ \\ \qquad f_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}) = e_1 : T_1, \\ \qquad\qquad \vdots \\ \qquad f_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk}) = e_n : T_n\} \\ \quad : \textbf{Theory} \end{array}$$

Theories are used to collect definitions and lemmas that logically belong together. Identifiers defined in a theory can be accessed from outside: if th is a theory and f_i an identifier defined in th, then the value of f_i is reached by $th.f_i$.

By defining functions giving theories as result, a notion of parametrised theory is obtained. Identifiers defined in a parametrised theory can then be accessed from outside, provided they are given proper parameters. Also the notion of (parametrised) theory is a conservative extension of the system: functions occuring in a parametrised theory can always be parametrised themselves and defined outside the theory.

The *let-expressions* are used for local lemmas and abbreviations:

$$\textbf{let } x = e_1 : T \textbf{ in } e_2$$

In the environment ρ, the expression above computes to $e_2(\rho, x = e_1\rho)$, i.e. the value of e_2 in the environment ρ extended with $x = e_1\rho$.

Expressions of this language are thus formed by

sorts	**Set**, **Type** and **Theory**
Π-types	$(x : A) \to B$

abstractions	$\lambda x \to e$
applications	$a\ b$
Σ-types	$\mathbf{sig}\{a_1 : A_1, \ldots, a_n : A_n\}$
structures	$\mathbf{struct}\{a_1 = e_1, \ldots, a_n = e_n\}$
projections	$b.a_i$
rec. def. types	$\mathbf{data}\{$ $c_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}),$ $\vdots$ $c_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk})\}$
constructors	c_i
case expressions	$\mathbf{case}\ x\ \mathbf{of}\,\{$ $c_1\ a_{11} \cdots a_{1m} \to e_1,$ $\vdots$ $c_n\ a_{n1} \cdots a_{nn} \to e_n\}$
let expressions	$\mathbf{let}\ x = e_1 : T\ \mathbf{in}\ e_2$
theories	$\mathbf{theory}\{$ $f_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}) = e_1 : T_1,$ $\vdots$ $f_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk}) = e_n : T_n\}$
projections	$th.f_i$
variables	x

The system also allow mutual recursive definitions. But this has not been used in the proofs in this paper, we have also avoided mutual recursion between a function f and functions locally defined in f.

There is a “size check” for inductively defined types. The type

$$\begin{array}{l}\mathbf{data}\{\\ \quad c_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}),\\ \quad\quad\quad\quad \vdots\\ \quad c_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk})\}\end{array}$$

lives in **Set** or **Type** if all A_{ij}'s live in **Set** or **Type**, respectively.

The definitional equality is a combination of structural equality and equal by name; for checking equality of “complex” structures, i.e. **data**, **sig**, **struct** and **case**, comparision “by name” is used. This means for instance that in

$$\begin{aligned} Bool &= \mathbf{data}\{False, True\} : \mathbf{Set}, \\ Bool' &= \mathbf{data}\{False, True\} : \mathbf{Set}, \\ Bool'' &= Bool : \mathbf{Set} \end{aligned}$$

Bool and *Bool’* are not equal, but *Bool* and *Bool”* are. This is the approach taken for several strongly typed languages.

The presence of both **Set** and **Type** in Half, where **Set** corresponds to a universe, allows a more abstract reasoning than is possible in a system without a universe. We show this by a small example with subsets of a set represented as propositional functions. First we give a name for the type of predicates over a type A:

$$pred(A : \mathbf{Type}) = (x : A) \to \mathbf{Set} : \mathbf{Type}.$$

The predicates over A are objects in the function space from A to **Set**. This function space does not form a set in predicative type theory (it has the type **Type**). In the same way, given a type A, we form the type for relations on A:

$$rel(A : \mathbf{Type}) = (x : A,\, y : A) \to \mathbf{Set} : \mathbf{Type}.$$

Now we represent subsets of a set A as predicates over A. We say that U is a subset of A if U is a propositional function ranging over A and an element a of A is a member of U iff $U(a)$ holds. A propositional function U is then a subset of another propositional function V provided that Ux implies Vx for all x of type A:

$$subset(A : \mathbf{Set}) = \lambda U\ V \to (x : A,\, h : Ux) \to Vx : rel\ (pred\ A).$$

Note that in the type we can see that, given a set A, *subset* A is a relation on predicates of A. Also note that, in the last definition, A must be a set, since by the definition of *rel*, $(x : A,\, h : Ux) \to Vx$ has to be a set. The system checks this for us.

3 Preliminary definitions

From now on, Half-code in `typewriter font` is mixed with comments, motivations and less formal definitions and proofs.

First some definitions about relations, predicates and operations:

```
rel(A:Type) = (x:A,y:A) -> Set : Type,
pred(A:Type) = (x:A) -> Set : Type,
bin(A:Type) = (x:A,y:A) -> A : Type,
op(A:Type) = (x:A) -> A : Type,
ref(A:Set,R:rel A) = (x:A) -> R x x : Set,
sym(A:Set,R:rel A) = (x:A,y:A,p:R x y) -> R y x : Set,
trans(A:Set,R:rel A) = (x:A,y:A,z:A,p:R x y,q:R y z) -> R x z : Set,
```

Note that `rel`, `pred`, `bin` and `op` also are applicable to elements of type `Set`.

The propositions *false* and *true* are defined as the empty set and a singleton set, respectively:

```
n0 = data{} : Set,
n1 = data{$True} : Set,
```

(In Half-code the constructors start with `$`). The connectives are defined by

```
not (A:Set) = (x:A) -> n0 : Set,
and (A:Set,B:Set) = sig{fst:A,snd:B} : Set,
or (A:Set,B:Set) = data{$Inl (x:A),$Inr (y:B)} : Set,
```

The definition of `and` should be compared to the equivalent definition using a *pair-constructor*,

```
and2(A:Set,B:Set) = data{$pair(a:A,b:B)} : Set.
```

However, a case-expression is required in order to analyse an object of type `and2 A B`, whereas for `and`, the proofs of `A` and `B` are obtained directly using the names `fst` and `snd`, respectively.

The existential quantifier is here defined using a Σ-*set* and, as was the case with `and`, the elimination rules are first and second projection:

```
exists(A:Set,B:pred A) = sig{fst:A,snd:B fst} : Set,
```

A set A is dense with respect to a relation R, if for all related x and y in A, there exists a z in A such that $R\ x\ z$ and $R\ z\ y$. (If `e` is an expression then `\x -> e` is the notation for a lambda abstraction.)

```
dense(A:Set,R:rel A) =
  (x:A,y:A,h:R x y) -> exists A (\z -> and (R x z) (R z y)) : Set,
```

A set A is decidable if $A \vee \neg A$ holds, and a relation R on A is decidable, if for all x and y, $R\ x\ y$ is decidable:

```
dec(A:Set) = or A (not A) : Set,
dec_rel(A:Set,R:rel A) = (x:A,y:A) -> dec (R x y) : Set,
```

A setoid is a set with an equivalence relation:

```
setoid(A:Set,R:rel A) =
  sig{isref:ref A R,issym:sym A R,istrans:trans A R} : Set,
```

A monoid is a setoid with a binary operation satisfying congruence, commutativity and associativity:

```
monoid(A:Set,eq:rel A,add:bin A) =
  sig{issetoid:setoid A eq,
      iscong:(x:A,y:A,z:A,t:A,h1:eq x z,h2:eq y t) ->
               eq (add x y) (add z t),
      iscom:(x:A,y:A) -> eq (add x y) (add y x),
      isassoc:(x:A,y:A,z:A) -> eq (add x (add y z)) (add (add x y) z)}
   : Set,
```

In the definition of formal space (section 4), propositional functions are used as subsets. Below we define what it means for a propositional function to be a subset of another propositional function. In general predicates do not respect equality, therefore the second (weaker) definition, that takes the equality relation as parameter, is used at some places. To justify the second subset relation, consider the following example: let $=$ be an equality defined on a set containing the elements x and y, and let U and V be predicates over that set; then using the first definition we do not in general have $x = y \ \&\ Ux \ \&\ U \subseteq V \Rightarrow Vx$. Moreover, in a formal topology the cover relation respects the equality relation. So, the second definition below is just as strong as it needs to be.

```
Subset(A:Set) = \U V -> (x:A,h:U x) -> V x : rel (pred A),

Subset2(A:Set,eq:rel A) =
  \U V -> (x:A,h:U x) -> exists A (\y -> and (eq x y) (V y))
  : rel (pred A),
```

Binary union for subsets as predicates is defined using disjunction:

```
Union(A:Set) = \U V -> \x -> or (U x) (V x) : bin (pred A),
```

We now define finite lists and use the concept of parametrised theory to collect some definitions and lemmas for lists.

```
list(A:Set) = data{$Nil,$Cons (x:A,xs:list A)} : Set,

theory_fin_list(A:Set,R:rel A) = theory{
  mem_list(x:A) = \xs -> case xs of {
                          $Nil -> n0,
                          $Cons y xs1 -> or (R x y) (mem_list x xs1)}
                : pred (list A),
   ...
```

(Just observe here that several definitions and lemmas are left out. We take the liberty of freely writing text inside theories like this.)

```
} : Theory {- end of theory_fin_list -}
```

An easy way to handle finite subsets is to use lists. But since lists of a type A and predicates over A have different types, a method for converting lists into predicates is needed when mixing the two notions. To transform a list into a predicate we simply abstract a variable belonging to the list (see **finset** below). The meaning of **finsubset l U** is "l is a finite subset of U". Finally, **findpart** takes a list X and a proof that X is a subset of a union, and finds the sublist Y of X belonging to the first subset in the union.

```
theory_subsets(A:Set,eq:rel A,issetoid:setoid A eq) = theory {
  th_fin_list = theory_fin_list A eq : Theory,

  finset(l:list A) = \x -> th_fin_list.mem_list x l : pred A,

  finsubset(l:list A,U:pred A) =
    case l of {
     $Nil -> n1,
     $Cons x xs -> and (U x) (finsubset xs U)} : Set,

  subset2 = Subset2 A eq:rel (pred A),

  findpart(X:list A,U:pred A,V:pred A,h:finsubset X (union U V)) = ...
    :exists
     (list A)
     (\Y->and (finsubset Y U) (subset2 (finset X) (union (finset Y) V)))
   ...
} : Theory,
```

We conclude this section by giving a type for intervals and a theory for intervals. Given a set, an interval is simply the pair of its endpoints:

```
interval(A:Set) = sig{lp:A,rp:A} : Set,
```

Given a set `A` and a relation `R` on `A`, we define the corresponding relation `S`, for intervals of `A`. This is used in the definition of the continuum (section 6), where $=$, $<$ and $\leq$ for rational intervals are defined from the corresponding relations on the rational numbers.

```
theory_interval(A:Set,R:rel A) = theory{
  B = interval A : Set,
  S = \I J -> and (R J.lp I.lp) (R I.rp J.rp) : rel B,
   ...
} : Theory
```

4 Formal spaces

We recall the definition of *formal topology* given by Giovanni Sambin [S]. A formal topology over a set A is a structure

$$\langle A, =, \cdot, \lhd \rangle$$

where $\langle A, =, \cdot \rangle$ is a commutative monoid, $\lhd$ is a relation, called *cover*, between elements and subsets of A such that, for any $x, y \in A$ and $U, V \subseteq A$, the following conditions hold:

$$\text{(substitutivity)} \quad \frac{x = y \qquad y \lhd U}{x \lhd U}$$

$$\text{(reflexivity)} \quad \frac{x \in U}{x \lhd U}$$

$$\text{(transitivity)} \quad \frac{x \lhd U \qquad U \lhd V}{x \lhd V} \qquad \text{where} \quad U \lhd V \equiv (\forall u \in U)(u \lhd V)$$

$$\text{(dot - left)} \quad \frac{x \lhd U}{x \cdot y \lhd U}$$

$$\text{(dot - right)} \quad \frac{x \lhd U \qquad x \lhd V}{x \lhd U \cdot V} \qquad \text{where} \quad U \cdot V \equiv \{u \cdot v \mid u \in U, v \in V\}.$$

Subsets of the base A are represented by propositional functions ranging over A (see the previous sections).

We point out that, in contrast to the definition of formal topology given in [S], we do not require the base monoid to have a unit, nor do we have the positivity predicate used in [S]. The equality relation on the base set is also explicit here.

A formal topology is here defined as a Σ-*type*: The set A with the relation $=$, the binary operation $\cdot$ and the relation $\lhd$ form a formal space; if A, $=$, $\cdot$ form a monoid and the rules of a formal topology (substitutivity, reflexivity, transitivity, dot-left, dot-right) are satisfied. In the implementation `eq`, `dot` and `cov` are used for $=$, $\cdot$ and $\lhd$, respectively. `DOT` and `COV` are used for the generalisations of $\cdot$ and $\lhd$, respectively, to subsets. Since `DOT` and `COV` are used both in the definition of the formal space and in the theory for formal spaces, they are defined globally; and, since $\cdot$ and $\lhd$ for subsets, depend on $\cdot$ and $\lhd$ for elements, `DOT` and `COV` have `A`, `eq`, `dot` and `cov` as parameters.

```
COV(A:Set,cov:(x:A,U:pred A)->Set)=
  \U V -> (x:A,p:U x) -> cov x V:rel (pred A),
DOT(A:Set,eq:rel A,dot:bin A)=
  \U V z -> sig{x:A,y:A,px:U x,py:V y,iseq:eq z (dot x y)}
  :bin (pred A),
```

If x and y are elements in U and V, respectively, then, immediate by the definition of $\cdot$ for subsets, $x \cdot y$ is an element in $U \cdot V$.

```
lemDOT(A:Set,eq:rel A,isref:ref A eq,dot:bin A,
       x:A,y:A,U:pred A,V:pred A,p:U x,q:V y)=
  struct{x=x,y=y,px=p,py=q,iseq=isref (dot x y)}
  :DOT A eq dot U V (dot x y),

space(A:Set,eq:rel A,dot:bin A,cov:(x:A,U:pred A)->Set)=
  sig{ismonoid:monoid A eq dot,
      ax0:(x:A,y:A,h2:eq x y,U:pred A,h3:cov y U)->cov x U,
      ax1:(x:A,U:pred A,h2:U x)->cov x U,
      ax2:(x:A,U:pred A,V:pred A,h2:cov x U,h3:COV A cov U V)->cov x V,
      ax3:(x:A,y:A,U:pred A,h2:cov x U)->cov (dot x y) U,
      ax4:(x:A,U:pred A,V:pred A,h2:cov x U,h3:cov x V)->
            cov x (DOT A eq dot U V)} : Type,
```

In the theory below some general facts of formal spaces are proved and some definitions are given. Later on we will define concrete formal spaces as instances of this theory.

```
theory_space (A:Set,eq:rel A,dot:bin A,cov:(x:A,U:pred A)->Set,
              s:space A eq dot cov) =
theory{
  union=Union A:bin (pred A),
  subset2=Subset2 A eq:rel (pred A),
  Cov=COV A cov:rel (pred A),
  Dot=DOT A eq dot:bin (pred A),

  lemDot=lemDOT A eq isref dot
       :(x:A,y:A,U:pred A,V:pred A,p:U x,q:V y)->Dot U V (dot x y),
```

The following lemmas say that $U \subseteq V \Rightarrow U \lhd V$ and $(U \cup V) \cdot (W \cup V) \lhd (U \cdot W) \cup V$, respectively.

```
  lem12(U:pred A,V:pred A)=
    \h->\x p->let {h1=h x p:exists A (\y ->and (eq x y) (V y))}
              in s.ax0 x h1.fst V h1.snd.fst (s.ax1 h1.fst V h1.snd.snd)
    :(h:subset2 U V)->Cov U V,

  lem7(U:pred A,V:pred A,W:pred A)=
    \x p->s.ax0
          x
          (dot p.x p.y)
          (union (Dot U W) V)
          p.iseq
          (case p.px of {
            $Inl x1->
              case p.py of {
```

```
              $Inl x2->
                s.ax1 (dot p.x p.y) (union (Dot U W) V)
                  ($Inl (lemDot p.x p.y U W x1 x2)),
              $Inr y->
                ax3l p.x p.y (union (Dot U W) V)
                  (s.ax1 p.y (union (Dot U W) V) ($Inr y))},
           $Inr y->
             s.ax3 p.x p.y (union (Dot U W) V)
                (s.ax1 p.x (union (Dot U W) V) ($Inr y))})
    :Cov (Dot (union U V) (union W V)) (union (Dot U W) V),
```

Given a formal space (note that we still are inside the theory `theory_space`), we now define the space induced by a subset. In our implementation that is achieved by a nested theory, in which the induced cover is defined and the cover-rules are proved.

```
theory_indspace(V:pred A) = theory{
```

Let $\lhd$ be a cover. The cover induced by the subset V is defined by $a \lhd_V U \equiv a \lhd U \cup V$:

```
covind=\x U->cov x (union U V):(x:A,U:pred A)->Set,
```

In order to prove that the space induced by V really is a formal space, an object of `space A eq dot covind` is constructed:

```
indspace=
  struct{
   ismonoid=s.ismonoid,
   ax0=\x y h2 U h3->s.ax0 x y h2 (union U V) h3,
   ax1=\x U h2->s.ax1 x (union U V) ($Inl h2),
   ax2=\x U V1 h2 h3->
         s.ax2
         x
         (union U V)
         (union V1 V)
         h2
         (\x1 p->case p of {
                  $Inl x2->h3 x1 x2,
                  $Inr y->s.ax1 x1 (union V1 V) ($Inr y)}),
   ax3=\x y U h2->s.ax3 x y (union U V) h2,
   ax4=\x U V1 h2 h3->
         s.ax2
         x
         (Dot (union U V) (union V1 V))
         (union (Dot U V1) V)
         (s.ax4 x (union U V) (union V1 V) h2 h3)
         (lem7 U V V1)}
  :space A eq dot covind
}:Theory,  {- end of theory_indspace -}
```

In order to define compactness and Stone spaces, a notion of finite subset is needed. For that purpose finite lists are used. Given a list, the function `finset` returns the corresponding subset, and the meaning of `finsubset l U` is "`l` is a finite subset of `U`".

```
th_subs=theory_subsets A eq issetoid:Theory,
finset=th_subs.finset:(l:list A)->pred A,
finsubset=th_subs.finsubset:(l:list A,U:pred A)->Set,
```

The following predicate says that, given a subset U and predicate P for subsets, there exists a finite subset of U for which P holds.

```
    existsFin(U:pred A,P:pred (pred A))=
      exists (list A) (\l->and (finsubset l U) (P (finset l))):Set,
```

isCover U is an abbreviation for "U covers the whole space":

```
    isCover=\U->(x:A)->cov x U:pred (pred A),
```

Now compactness, saying that if a subset U covers the whole space then there exists a finite subset of U that covers the whole space, and Stone cover (see [S]), saying that if the element x is covered by U then there exists a finite subset of U that covers x, are easily defined:

```
    compact=(U:pred A)->(h:isCover U)->existsFin U isCover:Type,
    stone=(x:A,U:pred A,h:cov x U)->existsFin U (\U0->cov x U0):Type
  }:Theory,  {- end of theory_space -}
```

We conclude this section by "covering" the definitions above with one more level of abstraction.

```
  SPACE = sig{A:Set,eq:rel A,dot:bin A,cov:(x:A,U:pred A)->Set,
              is_a_space:space A eq dot cov} : Type,
```

Given a space s, Token s returns the base of s

```
  Token(s:SPACE) = s.A : Set,
```

and Open s returns the subsets of the space s.

```
  Open(s:SPACE) = pred (Token s) : Type,
```

Given a space s and a subset U in s, indSpace U s forms the space where the cover in s is induced by U.

```
  indSpace(s:SPACE,U:Open s) =
    let {th1=theory_space s.A s.eq s.dot s.cov s.is_a_space:Theory,
         th2=th1.theory_indspace U:Theory}
    in struct{
        A=s.A,
        eq=s.eq,
        dot=s.dot,
        cov=th2.covind,
        is_a_space=th2.indspace}:SPACE,
```

CompactSPACE is a predicate of over all spaces (SPACE), saying that the space is compact.

```
  CompactSPACE(s:SPACE)=
    let {th=theory_space s.A s.eq s.dot s.cov s.is_a_space:Theory}
    in th.compact:Type,
```

Using a Σ-type, a compact space is a space which is compact.

```
  COMPACTSPACE=sig{s:SPACE,iscompact:CompactSPACE s}:Type,
```

We also define the corresponding for Stone spaces:

```
  StoneSPACE(s:SPACE)=
    let {th=theory_space s.A s.eq s.dot s.cov s.is_a_space:Theory}
    in th.stone:Type,

  STONESPACE=sig{s:SPACE,isstone:StoneSPACE s}:Type
```

5 Linear ordering

The rational numbers are formed abstractly as an unbounded, dense, decidable linear ordering. Following von Plato [vP], we start with the order relation $<$ and state the axioms $\neg(x < y \;\&\; y < x)$ and $x < y \Rightarrow (x < z \vee z < y)$.

```
islinear(A:Set,lt:rel A) =
 sig{LO1:(x:A,y:A,p:lt x y,q:lt y x) -> n0,
     LO2:(x:A,y:A,z:A,p:lt x y) -> or (lt x z) (lt z y)}
 : Set,
```

Less-then-or-equal (or rather *not-greater-than*) is defined as $x \leq y \equiv \neg(y < x)$. The equality $x = y \equiv x \leq y \;\&\; y \leq x$ then satisfy reflexivity, symmetry and transitivity.

```
leq = \x y -> not (lt y x) : rel A,
eq = \x y -> and (leq x y) (leq y x) : rel A,
```

To this ordering decidability $(x < y \vee y \leq x)$ is added:

```
isdeclinear(A:Set,lt:rel A)=sig{DLO1:islinear A lt,
                                DLO2:dec_rel A lt}:Set,
```

Then `max` and `min` can be defined by analysing the proof of $x < y \vee y \leq x$.

The rationals also form an unbounded $((\forall a)(\exists x)(x < a)$ and $(\forall a)(\exists x)(a < x))$ and dense $(x < y \Rightarrow (\exists z)(x < z < y))$ set.

```
isdenseunbdeclinear(A:Set,lt:rel A)=
  sig{isdeclin:isdeclinear A lt,
      nolb:(a:A)->exists A (\x->lt x a),
      noub:(a:A)->exists A (\x->lt a x),
      isdense:dense A lt}:Set,
```

Now we collect the definitions above in the following theory:

```
theoryUnboundedDenseDecidableLinear(
    A:Set,lt:rel A,isdudl:isdenseunbdeclinear A lt)=theory{
  leq=...:rel A,
  eq=...:rel A,
  min=...:bin A,
  max=...bin A,
   ...
} : Theory
```

6 The continuum as a formal space

The *topology of formal reals* is the structure

$$\langle Q \times Q, =_{Q\times Q}, \cdot, \lhd \rangle ,$$

where Q is the set of rational numbers. The monoid operation is defined by $(p, q) \cdot (r, s) \equiv (max(p, r), min(q, s))$; the cover $\lhd$ is defined by

$$(p, q) \lhd U \equiv (\forall p', q')(p < p' \;\&\; q' < q \Rightarrow (p', q') \lhd_f U) ,$$

where the relation $\lhd_f$ is inductively defined by

1. $\dfrac{q \leq p}{(p,q) \lhd_f U}$

2. $\dfrac{(p,q) \in U}{(p,q) \lhd_f U}$

3. $\dfrac{r < s \quad (p,s) \lhd_f U \quad (r,q) \lhd_f U}{(p,q) \lhd_f U}$

4. $\dfrac{(p',q') \lhd_f U \quad p' \leq p \quad q \leq q'}{(p,q) \lhd_f U}$.

For properties of this formal space we refer to [CN].

Starting from the linear ordering of the previous section, the continuum is here to be defined as a formal space. In the following theory, $\cdot$, $\lhd_f$ and $\lhd$ are defined (having the names `dot`, `covf` and `cov`, respectively). We also prove that $\lhd_f$ is a Stone cover and that $\lhd$ is a cover relation.

```
theory_continuum(Q:Set,ltQ:rel Q,
                 isdudl:isdenseunbdeclinear Q ltQ)=theory{
  th_dudl=theoryUnboundedDenseDecidableLinear Q ltQ isdudl
         :Theory,
  leqQ=th_dudl.leq:rel Q,
  max=th_dudl.max:bin Q,
  min=th_dudl.min:bin Q,
  eqQ=th_dudl.eq:rel Q,
  eqQsym=...:sym Q eqQ,
```

The base consists of the rational intervals:

```
  QxQ=interval Q:Set,
  int(p:Q,q:Q)=struct{lp=p,rp=q}:QxQ,
```

By instantiating the theory `theory_interval` with `Q` and a relation on `Q`, the corresponding relation on intervals is obtained.

```
  th_int_eqQ=theory_interval Q eqQ:Theory,
  eqQxQ=th_int_eqQ.S:rel QxQ,
  th_int_ltQ=theory_interval Q ltQ:Theory,
  ltQxQ=th_int_ltQ.S:rel QxQ,
  th_int_leqQ=theory_interval Q leqQ:Theory,
  leqQxQ=th_int_leqQ.S:rel QxQ,
  eqQxQref=...:ref QxQ eqQxQ,
```

The dot-operation is defined as intersection:

```
  dot=\x y -> int (max x.lp y.lp) (min x.rp y.rp):bin QxQ,
```

The formalised version of $\lhd_f$ is recursively defined by the following:

```
  covf(I:QxQ,U:pred QxQ)=
    data{$C1(h:leqQ I.rp I.lp),
         $C2(h:U I),
         $C3(J:QxQ,h1:ltQ J.lp J.rp,
             h2:covf (int I.lp J.rp) U,
             h3:covf (int J.lp I.rp) U),
         $C4(J:QxQ,h1:leqQxQ I J,h2:covf J U)}:Set,
```

`(QxQ,eqQxQ,dot)` forms a commutative monoid:

```
    ismonoid=struct{
              issetoid=struct{
                        isref=...,
                        issym=...,
                        istrans=...},
              iscong=...,
              iscom=...,
              isassoc=...}:monoid QxQ eqQxQ dot,
```

`(QxQ,eqQxQ,dot,covf)` is a formal space:

```
    Rf=struct{
        ismonoid=ismonoid,
        ax0=...,
        ax1=\x U h2->$C2 h2,
        ax2=...,
        ax3=...,
        ax4=...}:space QxQ eqQxQ dot covf,
    th_Rf=theory_space QxQ eqQxQ dot covf Rf:Theory,
```

The formalised version of $\lhd$ is explicitly defined by the following:

```
    cov(I:QxQ,U:pred QxQ)=(J:QxQ,h:ltQxQ J I)->covf J U:Set,
```

`(QxQ,eqQxQ,dot,cov)` forms a formal space:

```
    R=struct{
       ismonoid=ismonoid,
       ax0=...,
       ax1=...,
       ax2=...,
       ax3=...,
       ax4=...}
      :space QxQ eqQxQ dot cov,
    th_R=theory_space QxQ eqQxQ dot cov R:Theory,


  th_subs=theory_subsets QxQ eqQxQ (ismonoid.issetoid):Theory,
  finset=th_subs.finset:(l:list QxQ) -> pred QxQ,
  finsubset=th_subs.finsubset:(l:list QxQ,U:pred QxQ)->Set,
```

`covf` is a Stone cover:

```
  covfSc(I:QxQ,U:pred QxQ)=...
    :(h:covf I U)->
       exists
       (list QxQ)
       (\U0->and (finsubset U0 U) (covf I (finset U0))),

  isstone=covfSc:th_Rf.stone,
```

and `(QxQ,eqQxQ,dot,covf)` form a Stone space:

```
isStoneSpace=struct{
              s=struct{
                 A=QxQ,
                 eq=eqQxQ,
                 dot=dot,
                 cov=covf,
                 is_a_space=Rf},
              isstone=covfSc}:STONESPACE
} :Theory  {- end of theory_continuum -}
```

7 The Heine-Borel covering theorem

We now define the closed rational interval $[a, b]$ as a formal space and prove the Heine-Borel covering theorem, i.e, if U is a subset that covers $[a, b]$ then there exists a finite subset of U that covers $[a, b]$.

Let $\mathcal{R} \equiv \langle Q \times Q, =_{Q\times Q}, \cdot, \lhd_{\mathcal{R}} \rangle$ be the formal topology of formal reals and let

$$[a, b] \equiv \langle Q \times Q, =_{Q\times Q}, \cdot, \lhd_{[a,b]} \rangle$$

where the relation $\lhd_{[a,b]}$ is defined by

$$(p, q) \lhd_{[a,b]} U \equiv (p, q) \lhd_{\mathcal{R}} U \cup \{(r, a) \mid r \in Q\} \cup \{(b, s) \mid s \in Q\}.$$

Intuitively, the interval (p, q) is covered by U in the space $[a, b]$, if (p, q) intersected with *the closed interval* $[a, b]$ is covered by U in the space $\mathcal{R}$. In the sequel we will use the notation $\mathcal{C}[a, b]$ for $\{(r, a) \mid r \in Q\} \cup \{(b, s) \mid s \in Q\}$ and we understand $\mathcal{C}[a, b]$ as the complement of $[a, b]$.

```
theory_heineborel(Q:Set,ltQ:rel Q,
                  isdudl:isdenseunbdeclinear Q ltQ,a:Q,b:Q)=
theory{
  th_c=theory_continuum Q ltQ isdudl:Theory,
  th_R=th_c.th_R:Theory,
  th_Rf=th_c.th_Rf:Theory,
  th_subs=th_c.th_subs:Theory,

  eqQ=th_c.eqQ:rel Q,

  QxQ=th_c.QxQ:Set,
  eqQxQ=th_c.eqQxQ:rel QxQ,
  int=th_c.int:(p:Q,q:Q)->QxQ,
  dot=th_c.dot:bin QxQ,
  covR=th_c.cov:(I:QxQ,U:pred QxQ)->Set,
  covRf=th_c.covf:(I:QxQ,U:pred QxQ)->Set,

  Rf=th_c.Rf:space QxQ eqQxQ dot (th_c.covf),

  union=th_R.union:bin (pred QxQ),
  finset=th_subs.finset:(l:list QxQ)->pred QxQ,

  ltQxQ=th_c.ltQxQ:rel QxQ,
```

The cover on $[a, b]$ is defined in the following way: I is covered by U in $[a, b]$ if I is covered by the union of U and the the complement of $[a, b]$ in $\mathcal{R}$. The fact that $[a, b]$ really is a formal space is immediate, since the cover is an instance of a cover induced by a subset.

```
Cab=union (\x->(eqQ a x.rp)) (\x->(eqQ b x.lp)):pred QxQ,

covab(I:QxQ,U:pred QxQ)=covR I (union U Cab):Set,

th_ind=th_R.theory_indspace Cab:Theory,
ab=th_ind.indspace:space QxQ eqQxQ dot covab,
th_ab=theory_space QxQ eqQxQ dot covab ab:Theory,
```

`hblem1,2` below prove the equivalence

$$(\forall I)(I \lhd_{[a,b]} U) \Leftrightarrow (\exists r, s)(r < a \ \& \ b < s \ \& \ (r, s) \lhd_{\mathcal{R}_f} U \cup \mathcal{C}[a, b]).$$

$\Rightarrow$: By the axiomatisation of the rational numbers, there exist p and q such that $p < a$ and $b < q$. So given $(\forall I)(I \lhd_{[a,b]} U) \equiv (\forall I)(I \lhd_{\mathcal{R}} U \cup \mathcal{C}[a, b])$, then in particular $(p, q) \lhd_{\mathcal{R}} U \cup \mathcal{C}[a, b])$. Again by the axioms, there exists r and s such that $p < r < a$ and $b < s < q$. Then by the definition of $\lhd_{\mathcal{R}}$, $(r, s) \lhd_{\mathcal{R}_f} U \cup \mathcal{C}[a, b]$.

```
hblem1(U: pred QxQ)=
  \h->let {p=isdudl.no_lb a:exists Q (\x->ltQ x a),
           q=isdudl.no_ub b:exists Q (\x->ltQ b x),
           r=isdudl.isdense p.fst a p.snd
            :exists Q (\x->and (ltQ p.fst x) (ltQ x a)),
           s=isdudl.isdense b q.fst q.snd
            :exists Q (\x->and (ltQ b x) (ltQ x q.fst))}
      in struct{
          fst=int r.fst s.fst,
          snd=struct{
               fst=struct{
                    fst=r.snd.snd,
                    snd=s.snd.fst},
               snd=h (int p.fst q.fst) (int r.fst s.fst)
                    (struct{
                      fst=r.snd.fst,
                      snd=s.snd.snd})}}
  :(h:(I:QxQ)->covab I U)->
     exists QxQ (\x->and (ltQxQ (int a b) x) (covRf x (union U Cab))),
```

$\Leftarrow$: It is enough to prove $(p, q) \lhd_{\mathcal{R}_f} U \cup \mathcal{C}[a, b]$, for arbitrary p and q. Since $r < a$ and $b < s$, $(p, q) \lhd_{\mathcal{R}_f} \{(p, a), (r, s), (b, q)\}$. Since $(p, a) \in \mathcal{C}[a, b]$, $(r, s) \lhd_{\mathcal{R}_f} U \cup \mathcal{C}[a, b]$ and $(b, q) \in \mathcal{C}[a, b]$, $\{(p, a), (r, s), (b, q)\} \lhd_{\mathcal{R}_f} U \cup \mathcal{C}[a, b]$. The claim now follows by transitivity.

```
hblem2(U:pred QxQ)=
  \h->\I->\J->\h1->
    let {rs=h.fst:QxQ,
         U1=finset
            ($Cons rs ($Cons (int J.lp a) ($Cons (int b J.rp) $Nil)))
           :pred QxQ,
         h1=$C3
            (int rs.lp a)
            h.snd.fst.fst
```

```
              (Rf.ax1 (int J.lp a) U1
                 ($Inr ($Inl (th_c.eqQxQref (int J.lp a)))))
              ($C3
               (int b rs.rp)
               h.snd.fst.snd
               (Rf.ax1 (int h.fst.lp h.fst.rp) U1
                  ($Inl (th_c.eqQxQref rs)))
               (Rf.ax1 (int b J.rp) U1
                  ($Inr ($Inr ($Inl (th_c.eqQxQref (int b J.rp)))))))
             :th_c.covf J U1,
           h2=\J1->\p->
                case p of {
                 $Inl x -> Rf.ax0 J1 rs (union U Cab) x h.snd.snd,
                 $Inr y ->
                   case y of {
                    $Inl x ->Rf.ax1
                             J1
                             (union U Cab)
                             ($Inr ($Inl (th_c.eqQsym J1.rp a x.snd))),
                    $Inr y ->
                      case y of {
                       $Inl x -> Rf.ax1 J1 (union U Cab)
                                   ($Inr ($Inr x.fst)),
                       $Inr y -> case y of {}}}}
             :th_c.Covf U1 (union U Cab)}
      in (Rf.ax2 J U1 (union U Cab) h1 h2)
    :(h:exists QxQ (\rs->and
                       (ltQxQ (int a b) rs)
                       (covRf rs (union U Cab))))->
       (I:QxQ)->covab I U,
```

The Heine-Borel covering theorem: $[a, b]$ is compact, i.e.

$$(\forall I)(I \lhd_{[a,b]} U) \Rightarrow (\exists U_0 \subseteq_f U)(\forall I)(I \lhd_{[a,b]} U_0),$$

where $U_0 \subseteq_f U$ means that U_0 is a finite subset of U.

The proof goes as follows. Given $(\forall I)(I \lhd_{[a,b]} U)$, by `hblem1`, $(\exists r, s)(r < a \ \& \ b < s \ \& \ (r, s) \lhd_{\mathcal{R}_f} U \cup \mathcal{C}[a, b])$. Since $\lhd_{\mathcal{R}_f}$ is a Stone cover, there exists $W_0 \subseteq_f U \cup \mathcal{C}[a, b]$ such that $(r, s) \lhd_{\mathcal{R}_f} W_0$. By `findpart`, in `theory_subsets`, the part of W_0 belonging to U can be extracted, thus $(\exists U_0 \subseteq_f U)(\exists r, s)(r < a \ \& \ b < s \ \& \ (r, s) \lhd_{\mathcal{R}_f} U_0 \cup \mathcal{C}[a, b])$. The claim then follows by `hblem2`.

```
subset2=Subset2 QxQ eqQxQ:rel (pred QxQ),

heine_borel=\U h->
  let {rs=hblem1 U h
         :exists
          QxQ
          (\rs->and (ltQxQ (int a b) rs) (covRf rs (union U Cab))),
       W0=th_c.covfSc (rs.fst) (union U Cab) rs.snd.snd
         :exists (list QxQ) (\W0->and
                                  (th_subs.finsubset W0 (union U Cab))
                                  (covRf rs.fst (finset W0))),
```

```
        U0=th_subs.findpart W0.fst U Cab W0.snd.fst
          :exists
           (list QxQ)
           (\U0->and
                 (th_subs.finsubset U0 U)
                 (subset2 (finset W0.fst) (union (finset U0) Cab))),
        h1=Rf.ax2
           rs.fst
           (finset W0.fst)
           (union (finset U0.fst) Cab)
           W0.snd.snd
           (th_Rf.lem12
            (finset W0.fst)
            (union (finset U0.fst) Cab)
            U0.snd.snd)
          :covRf rs.fst (union (finset U0.fst) Cab)}
   in struct{
       fst=U0.fst,
       snd=struct{
            fst=U0.snd.fst,
            snd=hblem2 (finset U0.fst) (struct{
                                          fst=rs.fst,
                                          snd=struct{
                                               fst=rs.snd.fst,
                                               snd=h1}})}}
   :th_ab.compact
}:Theory  {- end of theory_heineborel -}
```

8 Acknowledgement

Parts of the formalisation in section 4 are due to Thierry Coquand. Sara Negri, Jan von Plato and Jan Smith have also contributed with useful remarks to this paper.

References

[Ba] H. Barendregt. *Lamda calculi with types*, In S. Abramsky, D.M. Gabbay and T.S.E. Maibaum eds., "Handbook of Logic in Computer Science, Vol. 2", Oxford University Press, Oxford, 1992.

[Br] N.G. de Bruijn. *A plea for weaker frameworks*, In G. Huet and G. Plotkin eds., "Logical Frameworks", pp. 40-68, Cambridge University Press, Cambridge, 1991.

[CN] J. Cederquist, S. Negri. *A constructive proof of the Heine-Borel covering theorem for formal reals*, In S. Berardi and M. Coppo eds., "Types for Proofs and Programs", Logic in Computer Science 1158, pp. 62–75, Springer-Verlag, 1996.

[C] T. Coquand. *An algorithm for type-checking dependent types*, Science of Computer Programming 26, pp. 167-177, Elsevier, 1996.

[H] M. Hofmann. *A model of intensional Martin-Löf type theory in which unicity of identity proofs does not hold*, Technical report, Dept. of Computer Science, University of Edinburgh, 1993.

[M] L. Magnusson. "The Implementation of ALF - a Proof Editor based on Martin-Löf's Monomorphic Type Theory with Explicit Substitution", Chalmers University of Technology and University of Göteborg, PhD Thesis, 1995.

[ML] P. Martin-Löf. *An Intuitionistic Theory of Types* (1972), To be published in the proceedings of Twentyfive years of Constructive Type Theory, G. Sambin and J. Smith eds., Oxford University Press.

[OSR] S. Owre, N. Shankar, J. M. Rushby. *The PVS Specification Language (Beta Release)*, Computer Science Laboratory, SRI International, Menlo Park, CA 94025, USA, 1993.

[vP] J. von Plato. *A memorandum on the constructive axioms of linear order*, Dept. of Philosophy, University of Helsinki, 1995.

[S] G. Sambin. *Intuitionistic formal spaces – a first communication*, In D. Skordev ed., "Mathematical logic and its applications", pp. 187-204, Plenum Press, 1987.

The Hahn-Banach Theorem in Type Theory

Jan Cederquist[1], **Thierry Coquand**[1], **Sara Negri**[2]
[1] Department of Computing Science
University of Göteborg
S-412 96 Göteborg, Sweden
[2] Department of Philosophy
PL 24, Unioninkatu 40 B
00014 University of Helsinki, Finland
e-mail: ceder@cs.chalmers.se,
coquand@cs.chalmers.se,
negri@helsinki.fi

Abstract

We give the basic definitions for pointfree functional analysis and present constructive proofs of the Alaoglu and Hahn-Banach theorems in the setting of formal topology.

1 Introduction

We present the basic concepts and definitions needed in a pointfree approach to functional analysis via formal topology. Our main results are the constructive proofs of localic formulations of the Alaoglu and Helly-Hahn-Banach[1] theorems.

Earlier pointfree formulations of the Hahn-Banach theorem, in a topos-theoretic setting, were presented by Mulvey and Pelletier in [13, 14] and by Vermeulen in [19]. A constructive proof based on points was given by Bishop [2]. In the formulation of his proof, the norm of the linear functional is preserved to an arbitrary degree by the extension and a counterexample shows that the norm, in general, is not preserved exactly.

As usual in pointfree topology, our guideline is to define the objects under analysis as formal points of a suitable formal space. After this has been accomplished for the reals, we consider the formal topology $\mathcal{L}(A)$ obtained as follows: To the formal space of mappings from a normed vector space A to the reals, we add the linearity and norm conditions in the form of covering axioms. The linear functionals of norm ≤ 1 from A to the reals then correspond to the formal points of this formal topology.

Given a subspace M of A, the classical Helly-Hahn-Banach theorem says that the restriction mapping from the linear functionals on A of norm ≤ 1 to those on M is surjective. In terms of covers, conceived as deductive systems, it becomes a conservativity statement (cf. [14]): Whenever a is an element and U is a subset of the base of the formal space $\mathcal{L}(M)$ and we have a derivation in $\mathcal{L}(A)$ of $a \lhd U$, then we can find a derivation in $\mathcal{L}(M)$ with the same conclusion.

[1] As explained in [11], the main idea in the usual proof of what is called the Hahn-Banach theorem is due to Helly. Since this is also the key idea in our derivation, we here rename the theorem in this way.

With this formulation it is quite natural to look for a proof by induction on covers. Moreover, as already pointed out in [14], it is possible to simplify the problem greatly, since it is enough to prove it for coherent spaces of which $\mathcal{L}(A)$ and $\mathcal{L}(M)$ are retracts. Then, in a derivation of a cover, we can assume that only finite subsets occur on the right-hand side of the cover relation. A global proof transformation makes it possible to change a derivation in $\mathcal{L}(A)$ into a derivation in $\mathcal{L}(M)$, since only a finite-dimensional extension of the space M has to be taken into account. In consequence, as in the case for the classical proof for the one-dimensional extension, no use of Zorn's lemma is needed.

There exist already two pointfree proofs of Hahn-Banach's theorem. The proof in [14] shows Hahn-Banach's theorem in any Grothendieck topos. However, the argument relies on Barr's theorem, for which no constructive justification has been given so far. The proof of Vermeulen [19] is done in the framework of topos theory with a natural number object, and thus, a priori, relies on the use of impredicative quantification.

Here as elsewhere (cf. [7, 8, 9, 16]) the use of formal topology allows for elementary and constructive proofs of pointfree formulations of classical results.

The two main contributions of this paper are the following:

- Our proof of the pointfree version of the Hahn-Banach theorem, following rather closely the original proof by Helly.
- This proof can actually be expressed in Martin-Löf's Type Theory (cf. [12]). In fact, on the basis of our proof, the first author has done a formalisation of the Hahn-Banach theorem in an implementation of the intensional version of Type Theory with one universe and finitary inductive definitions (cf. [6]).

2 Preliminaries

We recall here the definition of formal topology introduced by Per Martin-Löf and Giovanni Sambin [18]. We remark that, in contrast to the definition given in [18] and without any substantial difference in the development of the theory, we do not require the base monoid to have a unit. Nor do we have the positivity predicate used in [18].

Definition 2.1 (Formal topology) *A* formal topology *over a set S is a structure $\langle S, \cdot, \lhd \rangle$, where $\langle S, \cdot \rangle$ is a commutative monoid, $\lhd$ is a relation, called* cover, *between elements and subsets of S such that, for any $a, b \in S$ and $U, V \subseteq S$, the following conditions hold:*

(reflexivity) $$\frac{a \in U}{a \lhd U}$$

(transitivity) $$\frac{a \lhd U \qquad U \lhd V}{a \lhd V} \qquad \textit{where} \qquad U \lhd V \equiv (\forall u \in U)(u \lhd V)$$

(dot - left) $$\frac{a \lhd U}{a \cdot b \lhd U}$$

(dot - right) $$\frac{a \lhd U \qquad a \lhd V}{a \lhd U \cdot V} \qquad \textit{where} \qquad U \cdot V \equiv \{u \cdot v \mid u \in U, v \in V\}.$$

For readability reasons, when a singleton set occurs we will sometimes omit curly brackets, and write $a \lhd b$ for $a \lhd \{b\}$, and $U \cdot b$ for $U \cdot \{b\}$.

Since we dropped the unit element and the positivity predicate in the definition of formal topology, the definition of formal points given in [18] has to be revised as follows:

Definition 2.2 *Let $\langle S, \cdot, \lhd \rangle$ be a formal topology. A subset α of S is said to be a* formal point *if for all $a, b \in S$, $U \subseteq S$ the following conditions hold:*

1. $(\exists a \in S)(a \in \alpha)$;
2. $\dfrac{a \in \alpha \quad b \in \alpha}{a \cdot b \in \alpha}$;
3. $\dfrac{a \in \alpha \quad a \lhd U}{(\exists b \in U)(b \in \alpha)}$.

In order to maintain the usual intuition on points, in the sequel we will write $\alpha \Vdash a$ (α forces a, or α is a point in a) in place of $a \in \alpha$.

As an instance of the abstract definition of formal topology the continuum can be defined as a formal space. The proofs that this definition satisfies the rules of a formal topology can all be found in [7]. Further properties of the continuum as a formal space can also be found in [17], where a slightly different definition adopting the unit is given.

Definition 2.3 *The* formal topology of formal reals *is the structure*

$$\mathcal{R} \equiv \langle Q \times Q, \cdot, \lhd_{\mathcal{R}} \rangle,$$

where Q is the set of rational numbers. The monoid operation is defined by $(p, q) \cdot (r, s) \equiv (max(p, r), min(q, s))$, the cover $\lhd_{\mathcal{R}}$ by

$$(p, q) \lhd_{\mathcal{R}} U \equiv (\forall p', q')(p < p' < q' < q \rightarrow (p', q') \lhd_{\mathcal{R}_f} U)$$

where the relation $\lhd_{\mathcal{R}_f}$ is inductively defined by

1. $\dfrac{q \leq p}{(p, q) \lhd_{\mathcal{R}_f} U}$;
2. $\dfrac{(p, q) \in U}{(p, q) \lhd_{\mathcal{R}_f} U}$;
3. $\dfrac{p \leq r < s \leq q \quad (p, s) \lhd_{\mathcal{R}_f} U \quad (r, q) \lhd_{\mathcal{R}_f} U}{(p, q) \lhd_{\mathcal{R}_f} U}$;
4. $\dfrac{p' \leq p < q \leq q' \quad (p', q') \lhd_{\mathcal{R}_f} U}{(p, q) \lhd_{\mathcal{R}_f} U}$.

We denote with $Pt(\mathcal{R})$ the collection of formal points of $\mathcal{R}$, called *formal reals.*

In [7] it is proved that $\lhd_{\mathcal{R}_f}$ and $\lhd_{\mathcal{R}}$ are cover relations. Moreover, $\lhd_{\mathcal{R}_f}$ is a *Stone cover*, since we have:

Proposition 2.4 *If $(p, q) \lhd_{\mathcal{R}_f} U$, then there exists a finite subset U_0 of U such that $(p, q) \lhd_{\mathcal{R}_f} U_0$.*

Proof: See [7]. $\Box$

In [7] it is also proved that whenever $(p, q) \lhd_{\mathcal{R}} U$ and U is finite then $(p, q) \lhd_{\mathcal{R}_f} U$ and therefore $\lhd_{\mathcal{R}_f}$ is the *Stone compactification* of $\lhd_{\mathcal{R}}$ (cf. [18, 15]), but this stronger result will not be used in the sequel.

With $I = (p, q)$ and $J = (r, s)$, we write $I < J$ (resp. $I \leq J$) to express that $r < p < q < s$ (resp. $r \leq p < q \leq s$). Thus $I \lhd_{\mathcal{R}} U$ means $J \lhd_{\mathcal{R}_f} U$ for all $J < I$. Moreover, we use the notations $I + J$ for $(p + r, q + s)$, and tI for (tp, tq) when $t \geq 0$ and for (tq, tp) when $t < 0$.

The following lemma will be used in section 3.

Lemma 2.5 *If $J < I$ and $I \lhd_{\mathcal{R}_f} U$, then there exists a subset V of $Q \times Q$ such that $(\forall K \in V)(\exists L \in U)(K < L)$ and $J \lhd_{\mathcal{R}_f} V$.*

Proof: Straightforward by induction on the derivation of $I \lhd_{\mathcal{R}_f} U$. $\square$

3 Formal linear functionals

In this section we define the space of linear functionals of norm ≤ 1 from a seminormed space to the reals. This space is obtained by means of an inductive definition giving rise to a formal topology: Basic neighbourhoods correspond to basic opens in the *weak topology* (cf. [3]) and formal points correspond to the linear functionals of norm ≤ 1 from the given seminormed space to the space of formal reals.

Moreover, the formal cover for this space is defined explicitly from a Stone cover, and therefore a simple proof of Alaoglu's theorem is obtained.

3.1 The dual of a seminormed space as a formal space

We start by defining a seminormed space as in [14]. The seminorm is defined by means of formal open balls centred around the origin.

Definition 3.1 *A* seminormed space *A on the rationals Q is a linear space A on Q together with a mapping*

$$N : Q^{+} \longrightarrow \mathcal{P}(A)$$

from the positive rationals to the subsets of A satisfying the following conditions for $x, x' \in A$, $q, q' \in Q^{+}$:

N1. $x \in N(q) \to (\exists q' < q)(x \in N(q'))$;

N2. $(\exists q)(x \in N(q))$;

N3. $x \in N(q)\ \&\ x' \in N(q') \to x + x' \in N(q + q')$;

N4. $x \in N(q') \to qx \in N(qq')$;

N5. $x \in N(q) \to -x \in N(q)$;

N6. $0 \in N(q)$.

We limit ourselves to the presentation of the formal space $\mathcal{L}(A)$ of linear functionals of norm ≤ 1. The basic opens are finite sets of the form

$$w \equiv \{\langle x_1 \in I_1\rangle, \ldots, \langle x_n \in I_n\rangle\} \ ,$$

where $x_1, \ldots, x_n \in A$ and $I_1, \ldots, I_n$ are rational intervals. The intuitive reading of a basic open is that of a neighbourhood of functionals in the weak topology. We will in the sequel use the notation $\langle x_1 \in I_1, \ldots, x_n \in I_n\rangle$ for $\{\langle x_1 \in I_1\rangle, \ldots, \langle x_n \in I_n\rangle\}$.

We obtain with the operation

$$w_1 w_2 \equiv w_1 \cup w_2$$

a commutative and idempotent monoid with unit given by the empty set. We will denote with $S_{\mathcal{L}(A)}$ such a base of $\mathcal{L}(A)$.

Let $w \equiv \langle x_1 \in I_1, \ldots, x_n \in I_n\rangle$, then define

$$\begin{aligned} w \leq \langle x \in I\rangle \ \equiv \ & (\exists \langle x_{i_1} \in I_{i_1}\rangle, \ldots, \langle x_{i_p} \in I_{i_p}\rangle \in w) \\ & (x_{i_1} = \ldots = x_{i_p} = x \ \& \ I_{i_1} \cdot \ldots \cdot I_{i_p} \leq I) \end{aligned}$$

and

$$w \leq w' \ \equiv \ (\forall \langle x \in I\rangle \in w')(w \leq \langle x \in I\rangle).$$

Then, without assuming decidability of equality in A, $\leq$ is a reflexive and transitive relation.

Equality between basic neighbourhoods is subset equality

$$w = w' \equiv (\forall \langle x \in I\rangle)(\langle x \in I\rangle \in w \Leftrightarrow \langle x \in I\rangle \in w') \ .$$

Notice that $w = w'$ implies $w \leq w'$. This defines an equivalence relation on the type of basic neighbourhoods. We follow Bishop [2] in working systematically with types with an equivalence relation. Whenever we define a predicate, we have to be careful to check that this predicate respects the equivalence relation.

The finitary cover $\lhd_f$ for $\mathcal{L}(A)$ is inductively defined as follows:

C1 $$\frac{w \in U}{w \lhd_f U} \ ;$$

C2 $$\frac{w \leq w' \quad w' \lhd_f U}{w \lhd_f U} \ ;$$

C3 $$\frac{V \textit{ finite} \quad I \lhd_{\mathcal{R}_f} V \quad (\forall J \in V)(\langle x \in J\rangle w' \lhd_f U)}{\langle x \in I\rangle w' \lhd_f U} \ ;$$

C4 $$\frac{\langle x + y \in I + J\rangle w' \lhd_f U}{\langle x \in I, y \in J\rangle w' \lhd_f U} \ ;$$

C5 $$\frac{r \neq 0 \quad \langle rx \in rI\rangle w' \lhd_f U}{\langle x \in I\rangle w' \lhd_f U} \ ;$$

C6 $$\frac{x \in N(1) \quad \langle x \in (-1, 1)\rangle w \lhd_f U}{w \lhd_f U} \ .$$

A motivation for the above definition can be given as follows: conditions *C1*-*C3* define formal functionals from A to the formal reals, *C4* and *C5* impose linearity and *C6* says that we only consider functionals of norm ≤ 1.

Notice that, by *C2*, we have $w \lhd_f U$ whenever $w = w'$ and $w' \lhd_f U$.

In order to get a finitary inductive definition, the subset V in clause *C3* has to be finite. This however is not a restriction since $\lhd_{\mathcal{R}_f}$ is a Stone cover and such a finite set can always be found.

We have:

Proposition 3.2 *The relation* $\lhd_f$ *is a cover.*

Proof: Reflexivity holds by definition, *dot-left* follows from *C2* since $w_1 w_2 \leq w_1$, *transitivity* and *localization* are straightforward by induction on the derivation of $w \lhd_f U$. $\Box$

Then $\lhd$ is defined by

$$\langle x_1 \in I_1, \ldots, x_n \in I_n \rangle \lhd U \quad \equiv \quad (\forall J_1 < I_1, \ldots, J_n < I_n)(\langle x_1 \in J_1, \ldots, x_n \in J_n \rangle \lhd_f U).$$

Next, we prove that $\lhd$ is a cover, but before that some lemmas are needed.

Lemma 3.3 *If* $\langle x_1 \in K_1, \ldots, x_n \in K_n \rangle \leq \langle y_1 \in L_1, \ldots, y_m \in L_m \rangle$ *and* $J_1 < K_1, \ldots, J_n < K_n$, *then* $(\exists L'_1 < L_1, \ldots, L'_m < L_m)(\langle x_1 \in J_1, \ldots, x_n \in J_n \rangle \leq \langle y_1 \in L'_1, \ldots, y_m \in L'_m \rangle)$.

Proof: Starting from the hypotheses $\langle x_1 \in K_1, \ldots, x_n \in K_n \rangle \leq \langle y_1 \in L_1, \ldots, y_m \in L_m \rangle$ and $J_1 < K_1, \ldots, J_n < K_n$, the intervals L'_i $(1 \leq i \leq m)$ are constructed in the following way. By definition, for each $i \leq m$ there exists $p_1, \ldots, p_k$ such that $x_{p_1} = \ldots = x_{p_k} = y_i$ and $K_{p_1} \cdot \ldots \cdot K_{p_k} \leq L_i$. Since $J_1 < K_1, \ldots, J_n < K_n$ we have $J_{p_1} \cdot \ldots \cdot J_{p_k} < L_i$. Thus we put $L'_i = J_{p_1} \cdot \ldots \cdot J_{p_k}$. $\Box$

The following lemma makes the definition of $\lhd$ legitimate.

Lemma 3.4 *The relation* $\lhd$ *respects equality between basic neighbourhoods.*

Proof: We will show that *C2* also holds for $\lhd$. The conclusion then follows since $w = w'$ implies $w \leq w'$.

Suppose $\langle x_1 \in I_1, \ldots, x_n \in I_n \rangle \leq \langle y_1 \in J_1, \ldots, y_m \in J_m \rangle$ and $\langle y_1 \in J_1, \ldots, y_m \in J_m \rangle \lhd U$. Let $I'_1 < I_1, \ldots, I'_n < I_n$. Then, by lemma 3.3, there exist $J'_1 < J_1, \ldots, J'_m < J_m$ such that $\langle x_1 \in I'_1, \ldots, x_n \in I'_n \rangle \leq \langle y_1 \in J'_1, \ldots, y_m \in J'_m \rangle$. For such $J'_1, \ldots, J'_m$ we have $\langle y_1 \in J'_1, \ldots, y_m \in J'_m \rangle \lhd_f U$ and, by *C2*, $\langle x_1 \in I'_1, \ldots, x_n \in I'_n \rangle \lhd_f U$. Therefore $\langle x_1 \in I_1, \ldots, x_n \in I_n \rangle \lhd U$. $\Box$

Lemma 3.5 *If* $w \lhd_f U$ *then* $w \lhd U$.

Proof: Suppose $\langle x_1 \in I_1, \ldots, x_n \in I_n \rangle \lhd_f U$ and let $J_1 < I_1, \ldots, J_n < I_n$. Then $\langle x_1 \in J_1, \ldots, x_n \in J_n \rangle \leq \langle x_1 \in I_1, \ldots, x_n \in I_n \rangle$ so, by *C2*, $\langle x_1 \in J_1, \ldots, x_n \in J_n \rangle \lhd_f U$. Therefore $\langle x_1 \in I_1, \ldots, x_n \in I_n \rangle \lhd U$. $\Box$

The following lemma is used in the proof of *transitivity* for $\lhd$.

Lemma 3.6 *If* $\langle x_1 \in K_1, \ldots, x_n \in K_n \rangle \lhd_f U$, $J_1 < K_1, \ldots, J_n < K_n$ *and* $U \lhd V$, *then* $\langle x_1 \in J_1, \ldots, x_n \in J_n \rangle \lhd_f V$.

Proof: The proof is by induction on the derivation of $\langle x_1 \in K_1, \ldots, x_n \in K_n\rangle \lhd_f U$.

1. If $\langle x_1 \in K_1, \ldots, x_n \in K_n\rangle \in U$ then $\langle x_1 \in K_1, \ldots, x_n \in K_n\rangle \lhd V$, thus by definition, $\langle x_1 \in J_1, \ldots, x_n \in J_n\rangle \lhd_f V$.

2. If $\langle x_1 \in K_1, \ldots, x_n \in K_n\rangle \lhd_f U$ is derived by *C2* from $\langle x_1 \in K_1, \ldots, x_n \in K_n\rangle \leq \langle y_1 \in L_1, \ldots, y_n \in L_m\rangle$ and $\langle y_1 \in L_1, \ldots, y_n \in L_m\rangle \lhd_f U$, then by lemma 3.3 there exist $L_1' < L_1, \ldots, L_m' < L_m$ such that $\langle x_1 \in J_1, \ldots, x_n \in J_n\rangle \leq \langle y_1 \in L_1', \ldots, y_n \in L_m'\rangle$. For such $L_1', \ldots, L_m'$, by induction hypothesis, $\langle y_1 \in L_1', \ldots, y_n \in L_m'\rangle \lhd_f V$. Then, by *C2*, $\langle x_1 \in J_1, \ldots, x_n \in J_n\rangle \lhd_f V$.

3. Let $\langle x_1 \in K_1, \ldots, x_n \in K_n\rangle \lhd_f U$ be derived by *C3* from $K_1 \lhd_{\mathcal{R}_f} V'$ and $(\forall J' \in V')(\langle x_1 \in J', x_2 \in K_2, \ldots, x_n \in K_n\rangle \lhd_f U)$, where V' is finite. Since $J_1 < K_1$ and $K_1 \lhd_{\mathcal{R}_f} V'$, by lemma 2.5, we can construct a finite subset V'' such that $J_1 \lhd_{\mathcal{R}_f} V''$ and $(\forall J'' \in V'')(\exists J' \in V')(J'' < J')$. Then, by induction hypothesis, $(\forall J'' \in V'')(\langle x_1 \in J'', x_2 \in J_2, \ldots, x_n \in J_n\rangle \lhd_f V)$ and, by *C3*, $\langle x_1 \in J_1, \ldots, x_n \in J_n\rangle \lhd_f V$.

4. Let $\langle x_1 \in K_1, \ldots, x_n \in K_n\rangle \lhd_f U$ be derived by *C4* from $\langle x_1 + x_2 \in K_1 + K_2, x_3 \in K_3 \ldots, x_n \in K_n\rangle \lhd_f U$. Then $J_1 + J_2 < K_1 + K_2$ so, by induction hypothesis, $\langle x_1 + x_2 \in J_1 + J_2, x_3 \in J_3 \ldots, x_n \in J_n\rangle \lhd_f V$ and, by *C4*, $\langle x_1 \in J_1, \ldots, x_n \in J_n\rangle \lhd_f V$.

5. Similar to 4.

6. Let $\langle x_1 \in K_1, \ldots, x_n \in K_n\rangle \lhd_f U$ be derived by *C6* from $\langle x \in (-1, 1), x_1 \in K_1 \ldots, x_n \in K_n\rangle \lhd_f U$ and $x \in N(1)$. By *N1* there exists an $r < 1$ such that $x \in N(r)$. For such an r, by induction hypothesis, $\langle x \in (-r, r), x_1 \in J_1 \ldots, x_n \in J_n\rangle \lhd_f V$. Now the claim follows by *C6*. □

We are now ready to prove:

Proposition 3.7 *The relation $\lhd$ is a cover.*

Proof: Reflexivity: Immediate from *reflexivity* of $\lhd_f$ and lemma 3.5.

Dot-left, dot-right: Immediate from the definition of $\lhd$ and the corresponding properties of $\lhd_f$.

Transitivity: Suppose $\langle x_1 \in I_1, \ldots, x_n \in I_n\rangle \lhd U$ and $U \lhd V$. If $J_1 < I_1, \ldots, J_n < I_n$, then we can find $K_1, \ldots, K_n$ such that $J_1 < K_1 < I_1, \ldots, J_n < K_n < I_n$ and, by definition of $\langle x_1 \in I_1, \ldots, x_n \in I_n\rangle \lhd U$, we get $\langle x_1 \in K_1, \ldots, x_n \in K_n\rangle \lhd_f U$. Lemma 3.6 now gives $\langle x_1 \in J_1, \ldots, x_n \in J_n\rangle \lhd_f V$. Thus $\langle x_1 \in I_1, \ldots, x_n \in I_n\rangle \lhd V$. □

Finally, the following lemma will be useful in section 4.

Lemma 3.8 *The axioms for $\lhd_f$ are valid rules for $\lhd$.*

Proof: 1: Given $w \in U$, $w \lhd U$ is immediate from *C1* and lemma 3.5.

2: See the proof of lemma 3.4.

3: Suppose $I_1 \lhd_R V$ and $(\forall J \in V)(\langle x_1 \in J\rangle\langle x_2 \in I_2, \ldots, x_n \in I_n\rangle \lhd U)$. Let $I_1' < I_1, \ldots, I_n' < I_n$. Then there exists I'' such that $I_1' < I'' < I_1$. For such an I'', $I'' \lhd_{\mathcal{R}_f} V$ so by lemma 2.5 there exists a subset V' such that $I_1' \lhd_{\mathcal{R}_f} V'$ and $(\forall L' \in V')(\exists L \in V)(L' < L)$. But then, since $\lhd_{\mathcal{R}_f}$ is a Stone cover, there exists a finite subset $V_0' \subseteq V'$ such that $I_1' \lhd_{\mathcal{R}_f} V_0'$ and $(\forall L' \in V_0')(\exists L \in V)(L' < L)$. Now, by definition of $\lhd$, $(\forall J' \in V_0')(\langle x_1 \in J'\rangle\langle x_2 \in I_2', \ldots, x_n \in I_n'\rangle \lhd_f U)$ and, by *C3*, $\langle x_1 \in I_1', \ldots, x_n \in I_n'\rangle \lhd_f U$. Therefore $\langle x_1 \in I_1, \ldots, x_n \in I_n\rangle \lhd_f U$.

4 and 5 follows easily from the definition of $\lhd_f$, *C4* and *C5*, respectively.

6: Suppose $x \in N(1)$ and $\langle x \in (-1, 1), x_1 \in I_1, \ldots, x_n \in I_n\rangle \lhd U$. Let $J_1 < I_1, \ldots, J_n < I_n$. By *N1*, there exists $r < 1$ such that $x \in N(r)$. For such an r, $\frac{1}{r}x \in N(1)$. Now, by *C5* and the definition of $\lhd$, $\langle \frac{1}{r}x \in (-1,1), x_1 \in J_1, \ldots, x_n \in J_n \rangle \lhd_f U$ and, by *C6*, $\langle x_1 \in J_1, \ldots, x_n \in J_n \rangle \lhd_f U$. Therefore $\langle x_1 \in I_1, \ldots, x_n \in I_n \rangle \lhd U$. $\Box$

3.2 Linear functionals as formal points

In this section, which is not needed for the proof of the Hahn-Banach theorem, we will show how linear functionals of norm ≤ 1 from A to the formal space of reals are obtained in our setting. We denote the collection of formal points of $\mathcal{L}(A)$ with $Pt(\mathcal{L}(A))$. Given $F \in Pt(\mathcal{L}(A))$ and $x \in A$, let

$$F^*(x) \equiv \{(p,q) \in Q \times Q : F \Vdash \langle x \in (p,q) \rangle\}.$$

The following propositions are easily proved from the definitions (see also [13]):

Proposition 3.9 *F^* is a linear functional from A to $Pt(\mathcal{R})$ of norm ≤ 1, that is:*

1. *For all $x \in A$, $F^*(x) \in Pt(\mathcal{R})$;*
2. *$F^*(x+y) = F^*(x) + F^*(y)$;*
3. *$F^*(tx) = tF^*(x)$;*
4. *$x \in N(1) \Rightarrow -1 < F^*(x) < 1$.*

Conversely, given a linear functional f from A to $Pt(\mathcal{R})$, that is, a map satisfying *1–4* as in proposition 3.9, let f° be the subset of $S_{\mathcal{L}(A)}$ defined by

$$f^\circ \equiv \{\langle x_1 \in I_1, \ldots, x_n \in I_n \rangle : f(x_1) \Vdash I_1, \ldots, f(x_n) \Vdash I_n\}.$$

Then we have:

Proposition 3.10 *$f^\circ \in Pt(\mathcal{L}(A))$.*

The correspondence stated above is biunivocal since we have, with the notation used above, $(f^\circ)^* = f$ and $(F^*)^\circ = F$.

3.3 Alaoglu's theorem

The cover $\lhd_f$ is a Stone cover, since we have:

Proposition 3.11 *If $w \lhd_f U$, then there exists a finite subset U_0 of U such that $w \lhd_f U_0$.*

Proof: The proof is straightforward by induction on the derivation of $w \lhd_f U$. We will only consider the case when $w \lhd_f U$ is obtained by *C3* from $w \equiv \langle x \in I \rangle w'$, $I \lhd_{\mathcal{R}_f} V$ and $(\forall J \in V)(\langle x \in J \rangle w' \lhd_f U)$, for a finite subset V. By induction hypothesis, given an element $J \in V$, there exists a finite subset U_J of U such that $\langle x \in J \rangle w' \lhd_f U_J$. Let U_0 be the union of all such U_J's. Then by *C3*, $w \lhd_f U_0$. $\Box$

As an immediate consequence of the definition of the cover for $\mathcal{L}(A)$, we obtain Alaoglu's theorem, asserting that the unit ball of the space of linear and continuous functionals is compact in the weak topology.

Corollary 3.12 *The formal space $\mathcal{L}(A)$ is compact.*

Proof: Given $\langle\, \rangle \lhd U$, by definition of $\lhd$, $\langle\, \rangle \lhd_f U$. Then, since $\lhd_f$ is a Stone cover, $\langle\, \rangle \lhd_f U_0$ for some finite subset U_0 of U. Therefore, by lemma 3.5, $\langle\, \rangle \lhd U_0$. $\Box$

4 The Helly-Hahn-Banach theorem

In this section we prove the localic version of the Helly-Hahn-Banach theorem in the form of a conservativity result of the theory for the cover in $\mathcal{L}(A)$ over that in $\mathcal{L}(M)$. At the end of the section we also provide an informal motivation why this result is the localic statement of the usual point-set theorem.

The definition of $\lhd$ in terms of $\lhd_f$ allows us to replace $\lhd$ with $\lhd_f$ for proving conservativity. If M is a linear subspace of A, we say that $\lhd_f$ for $\mathcal{L}(A)$ is *conservative* over $\lhd_f$ for $\mathcal{L}(M)$ if, whenever w is an element and U is a subset of the base of $\mathcal{L}(M)$ and $w \lhd_f U$ in $\mathcal{L}(A)$, then $w \lhd_f U$ in $\mathcal{L}(M)$. Conservativity of $\lhd$ for $\mathcal{L}(A)$ over $\lhd$ for $\mathcal{L}(M)$ is defined in the same way. Then we have:

Proposition 4.1 *If $\lhd_f$ for $\mathcal{L}(A)$ is conservative over $\lhd_f$ for $\mathcal{L}(M)$, then $\lhd$ for $\mathcal{L}(A)$ is conservative over $\lhd$ for $\mathcal{L}(M)$.*

Proof: If $\langle x_1 \in I_1, \ldots, x_n \in I_n\rangle \lhd U$ in $\mathcal{L}(A)$, then, for all $J_1 < I_1, \ldots, J_n < I_n$, $\langle x_1 \in J_1, \ldots, x_n \in J_n\rangle \lhd_f U$, hence by conservativity for $\lhd_f$, for all $J_1 < I_1, \ldots, J_n < I_n$, $\langle x_1 \in J_1, \ldots, x_n \in J_n\rangle \lhd_f U$ in $\mathcal{L}(M)$ and hence $\langle x_1 \in I_1, \ldots, x_n \in I_n\rangle \lhd U$ in $\mathcal{L}(M)$. $\Box$

The localic statement of the Helly-Hahn-Banach theorem in the framework of formal topologies is, "$\lhd$ *for $\mathcal{L}(A)$ is conservative over $\lhd$ for $\mathcal{L}(M)$*". By the above proposition it thus reduces to the following:

Theorem 4.2 *$\lhd_f$ for $\mathcal{L}(A)$ is conservative over $\lhd_f$ for $\mathcal{L}(M)$.*

Since $\lhd_f$ is a Stone cover, we can transform any derivation of $a \lhd_f U$ in $\mathcal{L}(A)$ into one in which only finite subsets occur on the right-hand side of the cover relation. Therefore we can assume that only a finite number of elements not belonging to the base of $\mathcal{L}(M)$ are involved. Thus, arguing by induction, the problem is reduced to showing that if $a \lhd_f U$ is derived in $\mathcal{L}(M')$, where $M' \equiv [M + x]$ and x is an element in the base of $\mathcal{L}(A)$, then there exists also a derivation of $a \lhd_f U$ in $\mathcal{L}(M)$. The claim of theorem 4.2 thus becomes:

Proposition 4.3 *$\lhd_f$ for $\mathcal{L}(M')$ is conservative over $\lhd_f$ for $\mathcal{L}(M)$.*

We will use the notations $w \lhd_{M_f} U$ and $w \lhd_{M'_f} U$ to express that $w \lhd_f U$ in $\mathcal{L}(M)$ and $\mathcal{L}(M')$, respectively. Before proving the proposition, some auxiliary lemmas will be needed.

Lemma 4.4 $\langle y + z \in (p, q)\rangle \lhd \{\langle y \in (r, s), z \in (r', s')\rangle : p \leq r + r' < s + s' \leq q\}$

Proof: Since $(p'', q'') < (p, q)$ implies that

$$\langle y + z \in (p'', q'')\rangle \in \{\langle y + z \in (p', q')\rangle : p < p' < q' < q\} ,$$

we have by definition of $\lhd$ that

$$\langle y + z \in (p, q)\rangle \lhd \{\langle y + z \in (p', q')\rangle : p < p' < q' < q\}.$$

By *N2* and 3.8(6) we have

$$\langle\ \rangle \lhd \{\langle y \in (r, s)\rangle : r < s\} ,$$

and similarly

$$\langle\ \rangle \lhd \{\langle z \in (r', s')\rangle : r' < s'\} ,$$

so by stability

$$\langle y+z \in (p,q)\rangle \lhd \{\langle y+z \in (p',q'),\ \ y \in (r,s), z \in (r',s')\rangle : \\ p<p'<q'<q, r<s, r'<s'\}\ .$$

Since every interval can be covered by arbitrarily small intervals, using 3.8(3), we get

$$\begin{aligned}\langle y+z \in (p,q)\rangle \lhd \{\langle y+z \in (p',q'),\ \ & y \in (r,s), z \in (r',s')\rangle : \\ & p<p'<q'<q, r<s, r'<s', \\ & s-r \leq min(p'-p, q-q')/2, \\ & s'-r' \leq min(p'-p, q-q')/2\}\end{aligned}$$

and by 3.8(4)

$$\begin{aligned}\langle y+z \in (p,q)\rangle \lhd \{\langle y+z \in (p',q'),\ \ & y \in (r,s), z \in (r',s'), y+z \in (r+r', s+s')\rangle : \\ & p<p'<q'<q, r<s, r'<s', \\ & s-r \leq min(p'-p, q-q')/2, \\ & s'-r' \leq min(p'-p, q-q')/2\}\ .\end{aligned}$$

Now, if $r+r'<p$ then $s+s'<p'$ and, by 3.8(2),

$$\langle y+z \in (p',q'), y+z \in (r+r', s+s')\rangle \lhd \langle y+z \in (p', s+s')\rangle.$$

Then, since $s+s'<p'$, the right-hand side is covered by anything, in particular the empty set. If $q<s+s'$ we obtain symmetrically $\langle y+z \in (p',q'), y+z \in (r+r', s+s')\rangle \lhd \langle y+z \in (r+r', q')\rangle$, where $q'<r+r'$. We thus obtain

$$\langle y+z \in (p,q)\rangle \lhd \{\langle y \in (r,s), z \in (r',s')\rangle : p \leq r+r' < s+s' \leq q\}. \ \square$$

Let I be a finite index set, $\{a_n\}$ and $\{t_n\}$ sequences of A and Q, respectively, such that $(\forall i \in I)(a_i \in M \ \&\ a_i + t_i x \in N(1))$. Then, for rational q, define

$$P_q = \langle \ldots, a_i \in (-1+t_i q, 1+t_i q), \ldots \rangle.$$

The following lemma is the core of our proof. Intuitively it says that we can find a rational approximation for the value of $u(x)$, where u is a generic linear functional. We observe that the argument is similar to the one used by Helly.

Lemma 4.5 $\langle\ \rangle \lhd_{M_f} \{P_q : q \in Q\}$.

Proof: First observe that for all $i \in I$ such that $t_i = 0$, we can apply *C6* and *dot-right*. So, in the rest of this proof, we can assume that $t_i \neq 0$ for all i. For any $i, j \in I$, by the rules of N we have

$$\left\{\begin{array}{c} a_i + t_i x \in N(1) \\ a_j + t_j x \in N(1) \end{array}\right. \ \Rightarrow\ \left\{\begin{array}{c} a_i/t_i + x \in N(1/|t_i|) \\ -a_j/t_j - x \in N(1/|t_j|) \end{array}\right.$$

$$\Rightarrow\ a_i/t_i - a_j/t_j \in N(1/|t_i| + 1/|t_j|)\ .$$

By the rules for $\lhd_{M_f}$ we get

$$\langle\ \rangle \lhd_{M_f} \langle a_i/t_i - a_j/t_j \in (-1/|t_i| - 1/|t_j|, 1/|t_i| + 1/|t_j|)\rangle,$$

by the lemmas 3.5 and 4.4

$$\begin{array}{lll}\langle\,\rangle & \lhd_M & \{\langle a_i/t_i \in (r_i,s_i), a_j/t_j \in (r_j,s_j)\rangle : \\ & & -1/|t_i| - 1/|t_j| \le r_i - s_j < s_i - r_j \le 1/|t_i| + 1/|t_j|\}.\end{array}$$

From the definition of $\lhd$ we obtain the same for $\lhd_f$, and by *dot-right*

$$\begin{array}{lll}\langle\,\rangle & \lhd_{M_f} & \{\langle \ldots, a_i/t_i \in (r_i,s_i), \ldots\rangle : \\ & & (\forall i,j \in I)(-1/|t_i| - 1/|t_j| \le r_i - s_j < s_i - r_j \le 1/|t_i| + 1/|t_j|)\}\end{array}$$

$$\Rightarrow$$

$$\langle\,\rangle \lhd_{M_f} \{\langle \ldots, a_i/t_i \in (r_i,s_i), \ldots\rangle : (\forall i,j \in I)(s_i - 1/|t_i| \le r_j + 1/|t_j|)\}$$

$$\Rightarrow$$

$$\langle\,\rangle \lhd_{M_f} \{\langle \ldots, a_i/t_i \in (r_i,s_i), \ldots\rangle : \bigvee_{i\in I}(s_i - 1/|t_i|) \le \bigwedge_{i\in I}(r_i + 1/|t_i|)\}.$$

Now there are two cases depending on whether t_i is positive or negative; we only consider the case when t_i is positive.

$$\langle\,\rangle \lhd_{M_f} \{\langle \ldots, a_i \in (t_i r_i, t_i s_i), \ldots\rangle : \bigvee_{i\in I}(s_i - 1/|t_i|) \le \bigwedge_{i\in I}(r_i + 1/|t_i|)\}.$$

If $q = \bigvee_{i\in I}(s_i - 1/|t_i|)$, then $-1 + qt_i \le t_i r_i < t_i s_i \le 1 + qt_i$, because $s_i - 1/|t_i| \le q \le r_i + 1/|t_i|$, so we get

$$\langle\,\rangle \lhd_{M_f} \{P_q : q \equiv \bigvee_{i\in I}(s_i - 1/|t_i|) \le \bigwedge_{i\in I}(r_i + 1/|t_i|)\}.$$

and *a fortiori* $\langle\,\rangle \lhd_{M_f} \{P_q : q \in Q\}$. $\square$

Let I be a finite set as above. We define $\lhd'_{M'_f}$ as $\lhd_{M'_f}$, where the axiom $C6$ is replaced by

$$C6' : \ \frac{i \in I \quad \langle a_i + t_i x \in (-1,1)\rangle w \lhd'_{M'_f} U}{w \lhd'_{M'_f} U}.$$

Let $w \equiv \langle \ldots, a_i + t_i x \in (r_i,s_i), \ldots\rangle$ be an element in the base of M' and let q be a rational number. Then define

$$\bar{w} \equiv \langle \ldots, a_i \in (r_i + t_i q, s_i + t_i q), \ldots\rangle,$$

$$\bar{U} \equiv \{\bar{w} : w \in U\}.$$

Note that $\bar{w}$ depends on the proof of $w \in S_{\mathcal{L}(M')}$, and if $x \in M$ there are many ways of writing $a_i + t_i x$ and thus many proofs of $w \in S_{\mathcal{L}(M')}$.

We have:

Lemma 4.6 $w \lhd'_{M'_f} U \Rightarrow x \in M \ \vee \ (\forall q \in Q)(P_q \bar{w} \lhd_{M_f} \bar{U})$.

Proof: By induction on the derivation of $w \lhd'_{M'_f} U$. We only consider the cases *C1'* and *C6'*.

C1': Given $w \in U$, since $U \subseteq S_{\mathcal{L}(M')}$ we have two proofs of $w \in S_{\mathcal{L}(M')}$ and two possibly different values of $\bar{w}$, $\langle \ldots, a_i \in (r_i + t_i q, s_i + t_i q), \ldots\rangle$ and $\langle \ldots, b_i \in (r_i + u_i q, s_i + u_i q), \ldots\rangle$. Then there are two cases; if $(\forall i)(t_i = u_i)$, then $(\forall i)(a_i = b_i)$ and $\bar{w} \in \bar{U}$, thus $(\forall q \in Q)(P_q \bar{w} \lhd_{M_f} \bar{U})$; if $t_i \neq u_i$, then $x = \frac{a_i - b_i}{u_i - t_i} \in M$.

C6′: By inductive hypothesis we have

$$x \in M \ \vee \ (\forall q \in Q)(P_q \langle a_i \in (-1 + t_i q, 1 + t_i q) \rangle \bar{w} \lhd_{M_f} \bar{U})$$

that gives, since $\langle a_i \in (-1 + t_i q, 1 + t_i q) \rangle \in P_q$,

$$x \in M \ \vee \ (\forall q \in Q)(P_q \bar{w} \lhd_{M_f} \bar{U}) \ . \ \square$$

Observe here that we do not require decidability of membership of M. In lemma 4.6 a large part of the effort is devoted for the possibility of different values of $\bar{w}$ for different proofs of $w \in S_{\mathcal{L}(M')}$. This problem only occurs when $x \in M$. So decidability would simplify the proof; if $x \in M$ then $M' = M$ and there is nothing to prove, and if $x \notin M$ then the formulation and the proof of lemma 4.6 are easier.

If w is an element and U is a subset of the base of $\mathcal{L}(M)$, then $\bar{w} = w$ and $\bar{U} = U$, so as a corollary to lemma 4.6 we obtain:

Corollary 4.7 *Let w be an element, U a subset of the base of $\mathcal{L}(M)$, then*

$$w \lhd'_{M'_f} U \Rightarrow x \in M \ \vee \ (\forall q \in Q)(P_q w \lhd_{M_f} U).$$

Proof of proposition 4.3: Suppose $w \lhd_{M'_f} U$, where w is an element and U a subset in the base of $\mathcal{L}(M)$. First, by examining the axioms of the form *C6* in the proof of $w \lhd_{M'_f} U$, we can find finite sequences $\{a_n\}$ and $\{t_n\}$ in A and in Q, respectively, such that $(\forall i)(a_i \in M \ \& \ a_i + t_i x \in N(1))$. Then, the proof of $w \lhd_{M'_f} U$ can be converted into a proof of $w \lhd'_{M'_f} U$. By corollary 4.7 we have $x \in M \vee (\forall q \in Q)(P_q w \lhd_{M_f} U)$. If $x \in M$, $M' = M$ and there is nothing to prove. If $(\forall q \in Q)(P_q w \lhd_{M_f} U)$ then $\{P_q : q \in Q\} \cdot w \lhd_{M_f} U$ and, since $\langle\ \rangle \lhd_{M_f} \{P_q : q \in Q\}$ (lemma 4.5), we obtain $w \lhd_{M_f} U$ by *localization* and *transitivity.* $\square$

We include a proof in this setting of the following application of the Helly-Hahn-Banach theorem (which is proved indirectly in [13], whereas it has a direct proof in [19]).

Proposition 4.8 *If $\langle\ \rangle \lhd \langle x \in (-1,1) \rangle$, then $x \in N(1)$.*

The above proposition is proved by interpreting the neighbourhoods of $\mathcal{L}([x])$ as subsets of $Q \times Q$. First we define, for $r \in Q$ and $I = (p, q)$,

$$r \in I \equiv p < r < q.$$

Then we put

$$\langle t_1 x \in I_1, \ldots, t_n x \in I_n \rangle' \equiv \begin{cases} \emptyset & \text{if } (\exists i)(t_i = 0 \ \& \ 0 \notin I_i) \\ Q \times Q & \text{if } (\forall i)(t_i = 0 \ \& \ 0 \in I_i) \\ \{\bigwedge_{1 \le i \le n, t_i \neq 0} \frac{1}{t_i} I_i\} & \text{otherwise} \end{cases}$$

and let U' be the union of all w' such that $w \in U$.

Then we have:

Lemma 4.9 *If $w \lhd_f U$ in $\mathcal{L}([x])$, then*

$$(\exists r \in Q)(x \in N(r) \;\&\; (-r,r)\cdot w' \lhd_{\mathcal{R}_f} U').$$

Proof: The proof is by induction on the derivation of $w \lhd_f U$.

1. Let $w \lhd_f U$ be derived by *C1* from $w \in U$. Then, by the definition of $'$, $w' \subseteq U'$ so $w' \lhd_{\mathcal{R}_f} U'$. By *N2*, $(\exists r)(x \in N(r))$ and for such an r, $(-r,r)\cdot w' \lhd_{\mathcal{R}_f} U'$. Thus $(\exists r)(x \in N(r) \;\&\; (-r,r)\cdot w' \lhd_{\mathcal{R}_f} U')$.

For 2–6 below we argue by case analysis on the way $\langle t_1 x \in I_1, \ldots, t_n x \in I_n\rangle'$ is constructed. The cases $(\exists i)(t_i = 0 \;\&\; 0 \notin I_i)$ and $(\forall i)(t_i = 0 \;\&\; 0 \in I_i)$ are easy, so below we only consider the third case for which

$$\langle t_1 x \in I_1, \ldots, t_n x \in I_n\rangle' \;\equiv\; \{\bigwedge_{1\le i\le n, t_i\neq 0} \frac{1}{t_i} I_i\}\;.$$

2. Let $w_1 \lhd_f U$ be derived by *C2* from $w_1 \le w_2$ and $w_2 \lhd_f U$. From the definition of $\le$ and $'$, $w_1' = \{I\}$ and $w_2' = \{J\}$, for some $I \le J$, thus $w_1' \lhd_{\mathcal{R}_f} w_2'$. By induction hypothesis $(\exists r)(x \in N(r) \;\&\; (-r,r)\cdot w_2' \lhd_{\mathcal{R}_f} U')$. For such an r, $(-r,r)\cdot w_1' \lhd_{\mathcal{R}_f} (-r,r)\cdot w_2'$ so, by *transitivity*, $(-r,r)\cdot w_1' \lhd_{\mathcal{R}_f} U'$. Thus the claim.

3. Suppose $\langle tx \in I\rangle w \lhd_f U$ is derived by *C3* from V *finite*, $I \lhd_f V$ and $(\forall J \in V)((\langle tx \in J\rangle w \lhd_f U)$. Here also we only consider the case for which w' is a singleton set of an interval, the other cases being easy. By induction hypothesis $(\forall J \in V)(\exists r)(x \in N(r) \;\&\; (-r,r)\cdot ((\langle tx \in J\rangle w)' \lhd_{\mathcal{R}_f} U')$ and since V is finite we can take the smallest such r and obtain $x \in N(r) \;\&\; (\forall J \in V)((-r,r)\cdot ((\langle tx \in J\rangle w)' \lhd_{\mathcal{R}_f} U')$. We assume V to be nonempty, since if V is empty, I is negative and the claim follows easily. Since $((\langle tx \in J\rangle w)' = \frac{1}{t} I \cdot w'$, $(\forall K \in (-r,r)\cdot \frac{1}{t} w' \cdot V)(K \lhd_{\mathcal{R}_f} U')$. Moreover $(-r,r)\cdot \frac{1}{t} w' \cdot I \lhd_{\mathcal{R}_f} (-r,r)\cdot \frac{1}{t} w' \cdot V$ so, by *transitivity*, $(-r,r)\cdot \frac{1}{t} w' \cdot I \lhd_{\mathcal{R}_f} U'$. Thus the claim.

4. Suppose $\langle t_1 x \in I_1, t_2 x \in I_2\rangle w \lhd_f U$ is derived by *C4* from $\langle t_1 x + t_2 x \in I_1 + I_2\rangle w \lhd_f U$. By induction hypothesis $(\exists r)(x \in N(r) \;\&\; (-r,r)\cdot ((\langle t_1 x + t_2 x \in I_1 + I_2\rangle w)' \lhd_{\mathcal{R}_f} U')$. Then we just notice that $\langle t_1 x \in I_1, t_2 x \in I_2\rangle' \le \langle t_1 x + t_2 x \in I_1 + I_2\rangle'$ and therefore the claim.

5. Similar to 4.

6. Suppose $w \lhd_f U$ is derived by *C6* from $tx \in N(1)$ and $\langle tx \in (-1,1)\rangle w \lhd_f U$. Again, we only consider the case in which w' is a singleton set of an interval. By *N4* and *N5*, $x \in N(\frac{1}{|t|})$, and by induction hypothesis,

$$(\exists r)(x \in N(r) \;\&\; (-r,r)\cdot ((\langle tx \in (-1,1)\rangle w)' \lhd_{\mathcal{R}_f} U').$$

We have $((\langle tx \in (-1,1)\rangle w)' = \frac{1}{t}(-1,1)\cdot w'$. If $r \le \frac{1}{|t|}$ then $(-r,r) \le (-r,r)\cdot \frac{1}{t}(-1,1)$ and therefore $(-r,r)\cdot w' \lhd_{\mathcal{R}_f} U'$; if $r > \frac{1}{|t|}$ we have $\frac{1}{t}(-1,1)\cdot w' \lhd_{\mathcal{R}_f} U'$ and the claim follows from $x \in N(\frac{1}{|t|})$. Thus $(\exists r)(x \in N(r) \;\&\; (-r,r)\cdot w' \lhd_{\mathcal{R}_f} U')$. □

Proof of proposition 4.8: Suppose $\langle\,\rangle \lhd_{\mathcal{L}(A)} \langle x \in (-1,1)\rangle$. Then, by the (localic) Helly-Hahn-Banach theorem, we have $\langle\,\rangle \lhd_{\mathcal{L}([x])} \langle x \in (-1,1)\rangle$ and, by definition of $\lhd$, $\langle\,\rangle \lhd_{\mathcal{L}([x])_f} \langle x \in (-1,1)\rangle$. Thus by, lemma 4.9, there exists an $r \in Q^+$ such that $x \in N(r)$ and $(-r,r) \lhd_{\mathcal{R}_f} (-1,1)$. Therefore (cf. [7], lemma 10) $(-r,r) \le (-1,1)$, that is, $r \le 1$, thus $x \in N(1)$. □

The usual Hahn-Banach theorem states that the restriction mapping

$$\begin{array}{rcl} Pt(\mathcal{L}(A)) & \to & Pt(\mathcal{L}(M)) \\ f & \mapsto & \overline{f} \equiv f \cap S_{\mathcal{L}(M)} \end{array}$$

is surjective.

We can obtain this result from ours plus an assumption of extensionality, up to an arbitrarily good approximation. Namely we can prove that given g in $Pt(\mathcal{L}(M)$ and given $\omega \in g$ there exists $f \in Pt(\mathcal{L}(A))$ such that $\omega \in \overline{f}$: By contradiction, suppose that no such f exists. Then by extensionality $\omega \lhd_{\mathcal{L}(A)} \emptyset$ and therefore, by conservativity, $\omega \lhd_{\mathcal{L}(M)} \emptyset$, that contradicts $\omega \in g$.

5 Conclusion

The basic problem of the Helly-Hahn-Banach theorem can be formulated as follows. We have a normed vector space M, a linear functional u on M of norm ≤ 1, and we want to extend u to $[M + x_0]$. For this we consider two families

$$\begin{cases} A_x = u(x) - \|x - x_0\| \,, & x \in M \\ B_y = u(y) + \|y - x_0\| \,, & y \in M. \end{cases}$$

We have $A_x \leq B_y$ for all $x, y \in M$ and the core of the question is to find a real number r such that $A_x \leq r \leq B_y$ for all x and y. Classically this is possible by taking $r = sup(A_x)$. Intuitionistically, there is no reason why A_x should have a supremum. There are then two alternatives:

Bishop [2] and Bridges-Richman [4]:

One finds instead an extension u' of u of norm $\leq 1 + \epsilon$ for any given $\epsilon > 0$. For this, given $\epsilon > 0$, we change A_x and B_y into

$$\begin{cases} A'_x = u(x) - (1+\epsilon)\|x - x_0\| \\ B'_y = u(x) + (1+\epsilon)\|y - x_0\|. \end{cases}$$

We still have $A'_x \leq B'_y$, but the difference is that now $sup(A'_x)$ exists, because A'_x goes to $-\infty$ when x is big, and we can therefore restrict $sup(A'_x)$ to a compact subset of M.

So here we build a linear functional that extends the given one, but with norm $\leq 1 + \epsilon$ instead of ≤ 1.

Formal topology:

One reduces the problem to a finite collection of x and y. We can then take $sup\ A_x$, over a finite subset of M.

We never build a functional, only a finite approximation.

From this analysis, it is quite unlikely that one can easily deduce the Bishop version from the formal one, nor do we think that the proof in formal topology follows from Bishop's formulation. On the other hand, the formal one may be as good as Bishop's result for applications, since in an application one will only need finite approximations of a functional.

As we said, our proof can be represented in Martin-Löf's type theory, and indeed, proof-theoretically, we use only the notion of finitary inductive definition. An interesting question at this point is the connection of our work to [5]. For instance, our spaces are absolute, see [10], only for separable vector spaces, which explains the separability restriction in [5] that does not appear here.

One aim of this approach is to provide an alternative to Bishop's treatment of functional analysis in constructive mathematics. For this, it will be important to find concrete instances of the use of the Alaoglu and Hahn-Banach theorems, for which our proof can help to extract their computational content.

References

[1] P. Aczel. *An introduction to inductive definitions*, in "Handbook of Mathematical Logic", J. Barwise ed., pp. 739–782, North-Holland, Amsterdam, 1977.

[2] E. Bishop. "Foundations of Constructive Analysis", McGraw Hill, New York, 1967.

[3] H. Brezis. "Analyse fonctionelle – Théorie et applications", Masson Editeur, Paris, 1983.

[4] D. Bridges, F. Richman. "Varieties of Constructive Mathematics", London Mathematical Society Lecture Notes Series 97, Cambridge University Press, 1987.

[5] D.K. Brown, S.G. Simpson. *Which set existence axioms are needed to prove the separable Hahn-Banach theorem?*, Annals of Pure and Applied Logic 31, pp. 123–144, 1986.

[6] J. Cederquist. *A Machine Assisted Proof of the Hahn-Banach Theorem*, submitted for publication, 1997.

[7] J. Cederquist, S. Negri. *A constructive proof of the Heine-Borel covering theorem for formal reals*, in "Types for Proofs and Programs", S. Berardi and M. Coppo eds., Lecture Notes in Computer Science 1158, pp. 62–75, Springer, 1996.

[8] T. Coquand. *Constructive topology and combinatorics*, Lecture Notes in Computer Science 613, pp. 159–164, 1992.

[9] T. Coquand. *Minimal invariant spaces in formal topology*, The Journal of Symbolic Logic, in press.

[10] M.P. Fourman, R.J. Grayson. *Formal spaces*, in "The L. E. J. Brouwer Centenary Symposium", A. S. Troelstra and D. van Dalen eds., pp. 107-122, North Holland, Amsterdam, 1982.

[11] H. Hochstad. *Eduard Helly, Father of the Hahn-Banach theorem*, Mathematical Intelligencer, vol. 2, no. 3, pp. 123–125, 1980.

[12] P. Martin-Löf, *An Intuitionistic Theory of Types*, University of Stockholm, Stockholm, 1972, to be published in this volume.

[13] C.J. Mulvey, J.W. Pelletier. *The dual locale of a seminormed space*, Cahiers de topologie et geometrie differentielle, vol. 23, no. 1, pp. 73–92, 1987.

[14] C.J. Mulvey, J.W. Pelletier. *A globalization of the Hahn-Banach theorem*, Advances in Mathematics 89, pp. 1–60, 1991.

[15] S. Negri. *Stone bases, alias the constructive content of Stone representation*, in "Logic and Algebra", A. Ursini and P. Aglianò eds., pp. 617–636, Dekker, New York, 1996.

[16] S. Negri, S. Valentini. *Tychonoff's theorem in the framework of formal topologies*, The Journal of Symbolic Logic, in press.

[17] S. Negri, D. Soravia. *The continuum as a formal space*, submitted for publication, 1996.

[18] G. Sambin. *Intuitionistic formal spaces – a first communication*, in "Mathematical Logic and its Applications", D. Skordev ed., pp. 187–204, Plenum Press, New York, 1987.

[19] J.J.C. Vermeulen. "Constructive Techniques in Functional Analysis", Ph. D. Thesis, University of Sussex, 1986.

A Machine Assisted Proof of the Hahn-Banach Theorem

Jan Cederquist
Department of Computing Science
University of Göteborg
S-412 96 Göteborg, Sweden
e-mail: ceder@cs.chalmers.se

Abstract

We describe an implementation of a pointfree proof of the Alaoglu and the Hahn-Banach theorems in Type Theory. The proofs described here are formalisations of the proofs presented in "*The Hahn-Banach Theorem in Type Theory*" [4]. The implementation was partially developed simultaneously with [4] and it was a help in the development of the informal proofs.

1 Introduction

We present a machine assisted formalisation of pointfree topology in Martin-Löf's type theory. The continuum and the basic definitions needed in a pointfree approach to functional analysis are given and in this setting we describe implementations of localic formulations of the Alaoglu and the Hahn-Banach theorems.

The classical Hahn-Banach theorem says that, if M is a subspace of a normed linear space A and f is a bounded linear functional on M, then f can be extended to a linear functional F on A so that $\|F\| = \|f\|$. (In our proof we use the equivalent formulation: if $\|f\| \leq 1$ then f can be extended to F so that $\|F\| \leq 1$.) A constructive proof of the Hahn-Banach theorem, based on points, was given by Bishop in [2]. In his formulation of the theorem, the norm of the linear functional is preserved to an arbitrary degree by the extension and a counterexample shows that the norm, in general, is not preserved exactly.

In a pointfree formulation of the theorem, one works with finite approximations of the functionals rather than with the functionals themselves. Pointfree formulations of the Hahn-Banach theorem were presented by Mulvey and Pelletier [13] and by Vermeulen [20]. The proof in [13] shows the Hahn-Banach theorem in any Grothendieck topos. However, the argument relies on Barr's theorem, which is not justified constructively. The proof in [20] is done in the framework of topos theory with a natural number object, and thus relies on the use of impredicative quantification. The proof in [4], which this formalisation is based on, uses more basic concepts; in fact, the formalisation in this paper is done in type theory (cf. section 7.1) with only finitary inductive definitions.

This formalisation should not be seen only as a correctness check of the proofs presented in [4], but also as a help in the development of the informal proofs of the Hahn-Banach theorem. The implementation was partially developed simultaneously with [4]. Some parts in the proof were harder than we first expected, some steps were even wrong, and quite important steps in the original development were changed due to errors found during this formalisation.

The paper is organised as follows. First we provide all the preliminary definitions (formal topology and the continuum). Then follows a definition of the formal space of linear functionals of norm ≤ 1 and proofs of the Alaoglu and the Hahn-Banach theorems. Some further details specific to the implementation are then given, this includes axiomatically produced rational numbers and linear spaces. For further details of the proofs we then refer to the actual implementation.

We do not present all the lemmas occuring in the implementation, but we try to present the most important ones and show the overall structure of the implementation. Many properties of a defined object are obvious from an abstract point of view; we think that by presenting all of them the more important proofs would get drown.

The proofs presented in section 4–6 are basically the same as in [4]. The proofs here however have a direct correspondence to the implementation: they are proved in the same way and also presented in the same order as the implementation was developed, partly top-down. Here we also have a slightly more general definition of the space of the formal linear functionals than that given in [4]. This modification does not affect the informal proofs but it greatly simplifies the implementation.

2 Formal Spaces

Formal topologies were introduced by Per Martin-Löf and Giovanni Sambin [19] as a constructive approach to (pointfree) topology, in the tradition of Johnstone's version of the *Grothendieck topologies* [9] and Fourman and Grayson's *Formal Spaces* [7], but using a constructive set theory based on Martin Löf's type theory.

In pointfree topology one considers the open sets, and not the points, as primitive entities and studies those properties of a topological space that can be expressed without mentioning the points. By abstracting from the fact that open sets are subsets of points one only looks at the algebraic structure, called a *frame*, that the open sets form.

Since a point-set topology always can be presented using one of its bases, the abstract structure that we will consider is a commutative monoid $\langle S, \cdot \rangle$ where the set S corresponds to a base of the point-set topology $\Omega(X)$ and $\cdot$ corresponds to the operation of intersection between basic subsets.

In a point-set topology any open set is obtained as a union of elements of the base, but this union does not make sense if we refuse reference to points; hence we are naturally led to think that an open set may directly correspond to a subset of the set S. For this purpose we introduce a relation $\lhd$, called *cover*, between elements and subsets S whose intuitive meaning is that $a \lhd U$ when $a^* \subseteq \cup_{b \in U} b^*$, where a^* is the set of points of the neighbourhood a. The conditions we require of this relation can all be justified by this analogue to the point-set theoretic situation.

DEFINITION 2.1 *A* formal topology *over a set S is a structure*

$$\langle S, \cdot, \lhd \rangle$$

where $\langle S, \cdot \rangle$ is a commutative monoid, $\lhd$ is a relation, called cover*, between elements and subsets of S such that, for any $x, y \in S$ and $U, V \subseteq S$, the following conditions hold:*

$$\textit{(reflexivity)} \quad \frac{x \in U}{x \lhd U}$$

$$\textit{(transitivity)} \quad \frac{x \lhd U \qquad U \lhd V}{x \lhd V} \quad \textit{where} \quad U \lhd V \equiv (\forall u \in U)(u \lhd V)$$

$$\textit{(dot-left)} \quad \frac{x \lhd U}{x \cdot y \lhd U}$$

$$\textit{(dot-right)} \quad \frac{x \lhd U \qquad x \lhd V}{x \lhd U \cdot V} \quad \textit{where} \quad U \cdot V \equiv \{u \cdot v \mid u \in U, v \in V\}.$$

We point out that, in contrast to the definition of formal topology given by Sambin in [19], we do not require the base monoid to have a unit. The role of the unit element is that of the whole space and is here taken over by the subset of all elements in the base. Nor do we have the positivity predicate used in [19].

Subsets of S are here represented by predicates over S. We say that U is a subset of the base set S, $U \subseteq S$, if U is a propositional function ranging over S and we say that an element a of S is a member of the subset U, $a \in U$, iff $U(a)$ holds.

We work with sets with equalities and whenever a relation is defined, we check that this relation respects the equality. For a formal topology this means that we justify the definition of the cover by showing

$$\textit{(substitutivity)} \quad \frac{x = y \qquad y \lhd U}{x \lhd U} \ .$$

Several concrete formal topologies are defined in this paper and for each of them the equality relation is explicitly given, but the substitutivity rule is used implicitly in the proofs presented here.

The following derived rules are frequently used below:

$$\textit{(stability)} \quad \frac{x \lhd U \qquad y \lhd V}{x \cdot y \lhd U \cdot V} \qquad\qquad \textit{(localisation)} \quad \frac{x \lhd U}{x \cdot y \lhd U \cdot y}$$

and so is the property of the cover relation respecting the subset relation:

$$\frac{x \lhd U \qquad U \subseteq V}{x \lhd V}.$$

We write $x \lhd y$ for $x \lhd \{y\}$, $x \cdot U$ for $\{x\} \cdot U$, etc. Also note that dot-right follows from stability and that stability is derivable without dot-right if localisation holds. This means that in a proof that a structure is a formal topology we can show localisation (or stability) instead of dot-right.

We conclude this section by defining a notion of formal *Stone cover* [19] and *compact formal space*:

DEFINITION 2.2 *A cover $\lhd$ is called a* Stone cover *if, whenever a is an element and U a subset of the base, $a \lhd U$ implies $a \lhd U_0$ for some finite subset U_0 of U.*

DEFINITION 2.3 *Let $\mathcal{S} \equiv \langle S, \cdot, \lhd \rangle$ be a formal topology. We say that a subset U of the base S* covers *the whole space $\mathcal{S}$, $\mathcal{S} \lhd U$, if $(\forall x \in S)(x \lhd U)$. The space $\mathcal{S}$ is* compact *if, for any subset U of S, $\mathcal{S} \lhd U$ implies $\mathcal{S} \lhd U_0$ for some finite subset U_0 of U.*

Note that in a formal topology $\langle S, \cdot, \lhd \rangle$ with a unit element 1, compactness can be stated equivalently as follows. For any subset U of the base S, if $1 \lhd U$ then $1 \lhd U_0$ for some finite subset U_0 of U.

3 The Continuum as a Formal Space

Real numbers can be obtained as points of a locale where the subbasic elements are open rational intervals (cf. Johnstone [9]). Here the continuum is obtained as a certain formal topology based on the rational numbers.[1] The following definition was suggested by Thierry Coquand.

DEFINITION 3.1 *The* topology of formal reals *is the structure*

$$\mathcal{R} \equiv \langle Q \times Q, =_{Q \times Q}, \cdot, \lhd_R \rangle,$$

where Q is the set of rational numbers. Two intervals are equal if their respective endpoints are equal. The monoid operation is defined by $(p,q) \cdot (r,s) \equiv (max(p,r), min(q,s))$. The cover $\lhd_R$ is defined by

$$(p,q) \lhd_R U \equiv (\forall p', q')(p < p' \ \& \ q' < q \rightarrow (p', q') \lhd_{R_f} U),$$

where the relation $\lhd_{R_f}$ is inductively defined by

1. $\dfrac{q \leq p}{(p,q) \lhd_{R_f} U}$

2. $\dfrac{(p,q) \in U}{(p,q) \lhd_{R_f} U}$

3. $\dfrac{r < s \quad (p,s) \lhd_{R_f} U \quad (r,q) \lhd_{R_f} U}{(p,q) \lhd_{R_f} U}$

4. $\dfrac{(p',q') \lhd_{R_f} U \quad p' \leq p \quad q \leq q'}{(p,q) \lhd_{R_f} U}.$

With $I = (p,q)$ and $J = (r,s)$, we write $I < J$ (resp. $I \leq J$) to express that $r < p \ \& \ q < s$ (resp. $r \leq p \ \& \ q \leq s$). Thus $I \lhd_{\mathcal{R}} U$ means $J \lhd_{\mathcal{R}_f} U$ for all $J < I$. Moreover, we use the notations $I + J$ for $(p+r, q+s)$, and tI for (tp, tq) when $t \geq 0$ and for (tq, tp) when $t < 0$.

Now we will show that $\mathcal{R}_f \equiv \langle Q \times Q, =_{Q \times Q}, \cdot, \lhd_{R_f} \rangle$ and $\mathcal{R} \equiv \langle Q \times Q, =_{Q \times Q}, \cdot, \lhd_R \rangle$ are formal topologies. First note that $\langle Q \times Q, \cdot \rangle$ forms a commutative monoid under the relation $=_{Q \times Q}$. Then it is enough to prove that $\lhd_{R_f}$ and $\lhd_R$ really are cover relations.

LEMMA 3.2 *The relation $\lhd_{R_f}$ is a cover.*

Proof: By the fourth rule, $\lhd_{R_f}$ respects the equality. Reflexivity holds by definition. To prove transitivity, suppose $I \lhd_{R_f} U$ and $U \lhd_{R_f} V$. Then it is straightforward by induction on the derivation of $I \lhd_{R_f} U$ that $I \lhd_{R_f} V$. Dot-left follows by the fourth rule. Localisation follows by induction on the derivation of the premiss. □

The following lemma is used to prove transitivity of $\lhd_R$.

[1] For a description of the formal points of this formal topology we refer to [14].

LEMMA 3.3 *Suppose $I < J$, $J \lhd_{R_f} U$ and $U \lhd_R V$. Then $I \lhd_{R_f} V$.*

Proof: By induction on the derivation of $J \lhd_{R_f} U$. $\Box$

LEMMA 3.4 *The relation $\lhd_R$ is a cover.*

Proof: By the definition of $\lhd_R$ and the fact that $<$ on intervals respects the equality, $\lhd_R$ respects the equality relation. To prove transitivity, suppose that $I \lhd_R U$ and $U \lhd_R V$. To prove $I \lhd_R V$ we must then show that $J \lhd_{R_f} V$ holds for all $J < I$. So take a $J < I$. Then we can find J' such that $J < J' < I$. By definition of $\lhd_R$, $J' \lhd_{R_f} U$ and then, by lemma 3.3, $J \lhd_{R_f} V$. Dot-left follows from the definition of $\lhd_R$, the definition of the dot-operation and the fourth rule of $\lhd_{R_f}$. Dot-right is straightforward from the validity of dot-right of $\lhd_{R_f}$. $\Box$

We have thus proved

PROPOSITION 3.5 *$\mathcal{R}_f \equiv \langle Q \times Q, =_{Q\times Q}, \cdot, \lhd_{R_f}\rangle$ and $\mathcal{R} \equiv \langle Q \times Q, =_{Q\times Q}, \cdot, \lhd_R\rangle$ are formal topologies.*

We include here some lemmas used to prove properties of the space of formal linear functionals in the next section.

LEMMA 3.6 *Let $I, J, J_1, \ldots, J_n$ be intervals such that $I < J$ and $J \lhd_{R_f} \{J_1, \ldots, J_n\}$. Then*

$$(\exists I_1, \ldots, I_n)((\forall i)(I_i < J_i) \;\&\; I \lhd_{R_f} \{I_1, \ldots, I_n\}).$$

Proof: By induction on the derivation of $I \lhd_{R_f} \{J_1, \ldots, J_n\}$. $\Box$

LEMMA 3.7 *Let V be a finite subset of rational intervals. Then*

$$(p, q) \lhd_{R_f} V \Rightarrow (p + k, q + k) \lhd_{R_f} \{(r + k, s + k) : (r, s) \in V\}.$$

Proof: By induction on the derivation of $(p, q) \lhd_{R_f} V$. $\Box$

LEMMA 3.8 *Let (p, q) be a rational interval and r a rational number greater than 0. Then there is a finite set U of rational intervals, shorter than or equal to r, such that $(p, q) \lhd_{R_f} U$.*

Proof: We cover (p, q) with an overlapping family of intervals of length r.

For an arbitrary rational number $r > 0$ and a natural number n, let

$$U_n \equiv \{(p + (i - 1)r/2, p + (i + 1)r/2) : 0 \leq i \leq n\}.$$

We prove by induction on n that $(p, p + (n + 1)r/2) \lhd_{R_f} U_n$. Then, given a rational interval (p, q), by the Archimedian axiom, there exists a natural number n such that $n(r/2) > q - p$. For such an n, since $q < p + (n + 1)r/2$,

$$(p, q) \lhd_{R_f} U_n.$$

If $n = 0$ then, since $(p, p + r/2) \leq (p - r/2, p + r/2)$, by the fourth rule of $\lhd_{R_f}$,

$$(p, p + r/2) \lhd_{R_f} U_0.$$

Otherwise, by induction hypothesis,

$$(p, p + n(r/2)) \lhd_{R_f} U_{n-1}$$

and, since $U_{n-1} \subseteq U_n$,

$$(p, p + n(r/2)) \lhd_{R_f} U_n.$$

By reflexivity, $(p+(n-1)r/2, p+(n+1)r/2) \lhd_{R_f} U_n$. Then, since $p+(n-1)r/2 < p+n(r/2)$, by rule three for $\lhd_{R_f}$,

$$(p, p + (n + 1)r/2) \lhd_{R_f} U_n. \ \Box$$

The following easily proved property is used implicitly in the rest of this paper.

LEMMA 3.9 *Let I be a rational interval and U a subset of intervals. Then*

$$I \lhd_{R_f} U \Rightarrow I \lhd_R U.$$

For this and other properties of the formal space $\mathcal{R}$ we refer to [5] and [14].

4 Formal Linear Functionals

We start by defining a seminormed space as in [13]. Here a seminorm on a linear space X is not defined as a function $\|\cdot\|$ from X to the non-negative reals, but as a function N from the rationals to $\mathcal{P}(X)$. The rules we require for N can be justified, from those of a seminorm $\|\cdot\|$, by putting $N(q) \equiv \{x : \|x\| < q\}$. (For the representation of N and seminormed linear spaces in type theory see section 7.6.)

DEFINITION 4.1 *A* seminormed space *X on the rationals Q is a linear space X on Q together with a mapping*

$$N : Q \longrightarrow \mathcal{P}(X)$$

from the rationals to the subsets of X satisfying the following conditions for $x, y \in X$, $p, q \in Q$:

N0. $q \leq 0 \Rightarrow x \notin N(q)$

N1. $x \in N(q) \Rightarrow (\exists p < q)(x \in N(p))$

N2. $(\exists q)(x \in N(q))$

N3. $x \in N(q)$ & $y \in N(p) \Rightarrow x + y \in N(q + p)$

N4. $x \in N(q)$ & $p > 0 \Rightarrow px \in N(pq)$

N5. $x \in N(q) \Rightarrow -x \in N(q)$

N6. $x = y$ & $x \in N(q) \Rightarrow y \in N(q)$

N7. $x \in N(p)$ & $p = q \Rightarrow x \in N(q)$.

Given a seminormed linear space X, we will now define the formal space $\mathcal{L}(A)$ of linear functionals of norm ≤ 1, generated by a linear subspace A of X.[2] The basic opens are finite sets of the form $w \equiv [\langle x_1, I_1\rangle, \ldots, \langle x_n, I_n\rangle]$, where $x_1, \ldots, x_n$ are elements of X and $I_1, \ldots, I_n$ are rational intervals. The intuitive meaning of an element w is that of a neighbourhood of functionals in the weak topology [18], we say that w is a neighbourhood of f iff for all i, I_i is a neighbourhood of $f(x_i)$. In the sequel we use the notation $\langle x_1 \in I_1, \ldots, x_n \in I_n\rangle$ for $[\langle x_1, I_1\rangle, \ldots, \langle x_n, I_n\rangle]$.

DEFINITION 4.2 *Let X be a linear space and A a linear subspace of X. The base S of $\mathcal{L}(A)$ consists of finite lists of the form $\langle x_1 \in I_1, \ldots, x_n \in I_n\rangle$, where $x_1, \ldots, x_n$ are elements in X and $I_1, \ldots, I_n$ are rational intervals.*

Let $w \equiv \langle x_1 \in I_1, \ldots, x_n \in I_n\rangle$ and $w' \equiv \langle y_1 \in J_1, \ldots, y_m \in J_m\rangle$. Then define

$$
\begin{aligned}
w = w' &\equiv n = m \;\&\; (\forall i)(x_i = y_i \;\&\; I_i = J_i),\\
w < w' &\equiv n = m \;\&\; (\forall i)(x_i = y_i \;\&\; I_i < J_i),\\
w \leq \langle x \in I\rangle &\equiv (\exists \langle x_{i_1} \in I_{i_1}, \ldots, x_{i_p} \in I_{i_p}\rangle \subseteq w)\\
&\qquad (x_{i_1} = \cdots = x_{i_p} = x \;\&\; I_{i_1} \cdot \cdots \cdot I_{i_p} \leq I),\\
w \leq w' &\equiv (\forall \langle x \in I\rangle \in w')(w \leq \langle x \in I\rangle).
\end{aligned}
$$

Let w be an element and U a subset of S. Then $w \lhd_{A_f} U$ is inductively defined by

$$C1\ \frac{w \in U}{w \lhd_{A_f} U}$$

$$C2\ \frac{w \leq w' \quad w' \lhd_{A_f} U}{w \lhd_{A_f} U}$$

$$C3\ \frac{x \in A \quad w = \langle x \in I\rangle w' \quad I \lhd_{\mathcal{R}_f} V \quad (\forall J \in V)(\langle x \in J\rangle w' \lhd_{A_f} U)}{w \lhd_{A_f} U}$$

where V is a finite subset

$$C4\ \frac{x, y \in A \quad w = \langle x \in I, y \in J\rangle w' \quad \langle x + y \in I + J\rangle w' \lhd_{A_f} U}{w \lhd_{A_f} U}$$

$$C5\ \frac{x \in A \quad w = \langle x \in I\rangle w' \quad r \neq 0 \quad \langle rx \in rI\rangle w' \lhd_{A_f} U}{w \lhd_{A_f} U}$$

$$C6\ \frac{x \in A \quad x \in N(1) \quad \langle x \in (-1,1)\rangle w \lhd_{A_f} U}{w \lhd_{A_f} U}.$$

Then $w \lhd_A U$ is defined by

$$w \lhd_A U \equiv (\forall w' < w)(w' \lhd_{A_f} U).$$

[2] The formal points of $\mathcal{L}(A)$ correspond to the linear functionals of norm ≤ 1 from A to the reals, see [4].

A motivation for the above definition can be given as follows: conditions *C1-C3* define formal functionals from A to the formal reals, *C4* and *C5* impose linearity and *C6* says that we only consider functionals of norm ≤ 1.

Observe that in order to get a finitary inductive definition, the subset V in clause *C3* has to be finite. In fact, this inductive definition cannot be made in predicative type theory if V is an arbitrary subset.

The formal space $\mathcal{L}(A)$ of neighbourhoods of linear functionals of norm ≤ 1 from A to the reals is defined as:

DEFINITION 4.3 *Let* $\mathcal{L}(A_f) \equiv \langle S, =_\subseteq, \cdot, \lhd_{A_f} \rangle$ *and* $\mathcal{L}(A) \equiv \langle S, =_\subseteq, \cdot, \lhd_A \rangle$, *where the equality between neighbourhoods is the subset equality:* $w_1 =_\subseteq w_2 \equiv w_1 \subseteq w_2 \ \&\ w_2 \subseteq w_1$, *and the monoid operation is union.*

Observe that the base S is independent of the subspace A. Different subspaces of X give rise to different formal spaces of linear functionals, but all of these formal spaces have the same base. This will greatly simplify the implementation. The alternative (as suggested in [4]) is to let the vectors $x_1, \ldots, x_n$ occuring in a neighbourhood in S belong to the specific subspace A. A neighbourhood will then be a list of triples $\langle x, h, I \rangle$, where h is a proof that x is an element of A. The problem is now that the elements have to be transformed (the proof objects have to be changed) between formal spaces generated from different linear subspaces. This problem already appears in the formulation of the Hahn-Banach theorem (theorem 6.1) and building the proofs upon this definition, these transformations would soon occupy a considerable part of the proofs.

To express that the vectors occuring in a neighbourhood w belong to a certain linear space A, the expression $w \in S$ can thus not be used. For this purpose we introduce the relation *live*:

DEFINITION 4.4 *Let* $w \equiv \langle x_1 \in I_1, \ldots, x_n \in I_n \rangle$. *We say that* w lives *in the space generated by* A *if* $(\forall i)(x_i \in A)$; *and a subset* U lives *in* $\mathcal{L}(A)$ *if all its elements* live *in* $\mathcal{L}(A)$.

If we forget about the proof objects, then the definition used here and the corresponding definition in [4] are equivalent[3], so our simplification does not change the informal proofs.

We continue by proving that $\mathcal{L}(A_f)$ and $\mathcal{L}(A)$ are formal topologies. First note that $\langle S, \cdot \rangle$ form a commutative monoid under the subset equality, it is then enough to show that $\lhd_{A_f}$ and $\lhd_A$ are cover relations.

LEMMA 4.5 *The relation* $\lhd_{A_f}$ *is a cover.*

Proof: If $w =_\subseteq w'$ then $w \leq w'$ so, by *C2*, $\lhd_{A_f}$ respects equality. Reflexivity holds by definition. Dot-left follows from *C2*, since $w_1 w_2 \leq w_1$. Transitivity and localisation are straightforward by induction on the proof of $w \lhd_{A_f} U$ and, as noted before, dot-right follows from localisation. □

The following lemma is used in the proof of transitivity of $\lhd_A$.

[3] If w and U live in $\mathcal{L}(A)$ and $w \lhd_{A_f} U$, then it is easy to *dress* w and U with proof objects into w' and U' and, by induction on $w \lhd_{A_f} U$, $w' \lhd'_{A_f} U'$ follows, where $\lhd'_{A_f}$ is the corresponding definition in [4]. Conversely, if $w' \lhd'_{A_f} U'$, then the corresponding w and U live in $\mathcal{L}(A)$ and $w \lhd_{A_f} U$. The same then also holds for $\lhd_A$.

LEMMA 4.6 *Let $w_1 < w$, $w \lhd_{A_f} U$ and $U \lhd_A V$. Then $w_1 \lhd_{A_f} V$.*

Proof: By induction on the derivation of $w \lhd_{A_f} U$.

C1: If $w \in U$ then, since $U \lhd_A V$, $w \lhd_A V$ and, by definition of $\lhd_A$, $w_1 \lhd_{A_f} V$.

C2: $w \lhd_{A_f} U$ is derived from $w \leq w'$ and $w' \lhd_{A_f} U$. Since $w_1 < w$ and $w \leq w'$, there is a partition w_2 of w_1 such that $w_1 \leq w_2$ and $w_2 < w'$. So, by induction hypothesis, $w_2 \lhd_{A_f} V$ and, by *C2*, $w_1 \lhd_{A_f} V$.

C3: $w \lhd_{A_f} U$ is derived from $w = \langle x \in I\rangle w_2$, $I \lhd_{R_f} \{K_1, \ldots, K_m\}$ and $(\forall i)(\langle x \in K_i\rangle w_2 \lhd_{A_f} U)$. Since $w_1 < w = \langle x \in I\rangle w_2$, $w_1 \equiv \langle y \in J\rangle w_3$ for some $y = x$, $J < I$ and $w_3 < w_2$. By lemma 3.6 there is a set $\{K'_1, \ldots, K'_m\}$ such that $J \lhd_{R_f} \{K'_1, \ldots, K'_m\}$ and $(\forall i)(K'_i < K_i)$. For such a set, by induction hypothesis,

$$(\forall i)(\langle y \in K'_i\rangle w_3 \lhd_{A_f} V).$$

Then $w_1 \lhd_{A_f} V$ follows by *C3*.

C4: $w \lhd_{A_f} U$ is derived from $w = \langle x_1 \in I_1, x_2 \in I_2\rangle w_2$ and $\langle x_1 + x_2 \in I_1 + I_2\rangle w_2 \lhd_{A_f} U$. Since $w_1 < w = \langle x_1 \in I_1, x_2 \in I_2\rangle w_2$, $w_1 \equiv \langle y_1 \in J_1, y_2 \in J_2\rangle w_3$ for some $y_1 = x_1$, $y_2 = x_2$, $J_1 < I_1$, $J_2 < I_2$ and $w_3 < w_2$, and thus $J_1 + J_2 < I_1 + I_2$. Then, by induction hypothesis,

$$\langle y_1 + y_2 \in J_1 + J_2\rangle w_3 \lhd_{A_f} V.$$

$w_1 \lhd_{A_f} V$ now follows by *C4*.

C5: $w \lhd_{A_f} U$ is derived from $w = \langle x \in I\rangle w_2$, $r \neq 0$ and $\langle rx \in rI\rangle w_2 \lhd_{A_f} U$. Since $w_1 < w = \langle x_1 \in I\rangle w_2$, $w_1 \equiv \langle y \in J\rangle w_3$, for some $y = x$, $J < I$ and $w_3 < w_2$, and thus $rJ < rI$. Then, by induction hypothesis,

$$\langle ry \in rJ\rangle w_3 \lhd_{A_f} V.$$

$w_1 \lhd_{A_f} V$ now follows by *C5*.

C6: $w \lhd_{A_f} U$ is derived from $x \in N(1)$ and $\langle x \in (-1,1)\rangle w \lhd_{A_f} U$. By *N1*, $x \in N(r)$ for some $r < 1$. Then, by induction hypothesis,

$$\langle x \in (-r, r)\rangle w_1 \lhd_{A_f} V.$$

$w_1 \lhd_{A_f} V$ now follows by *C6* and the rules of N. □

LEMMA 4.7 *The relation $\lhd_A$ is a cover.*

Proof: We start by justifying the definition of $\lhd_A$, by showing that $\lhd_A$ respects the equality relation $=_\subseteq$. To prove $w_1 \lhd_A U$ from $w_1 =_\subseteq w_2$ and $w_2 \lhd_A U$, we must show $w'_1 \lhd_{A_f} U$ for an arbitrary $w'_1 < w_1$. Given $w'_1 < w_1$ and $w_1 =_\subseteq w_2$ we can find a partition w'_2 of w'_1 such that $w'_2 < w_2$ and $w'_2 \subseteq w'_1$. Moreover $w'_2 \subseteq w'_1$ implies $w'_1 \leq w'_2$ and, by definition of $\lhd_A$, $w'_2 \lhd_{A_f} U$. So, by *C2*, $w'_1 \lhd_{A_f} U$.

Reflexivity: follows from reflexivity of $\lhd_{A_f}$, since $w \lhd_{A_f} U$ implies $w \lhd_A U$ (lemma 4.9).

Dot-left: Suppose $w_1 \lhd_A U$. To prove $w_1 w_2 \lhd_A U$ we must show that $w \lhd_{A_f} U$ holds for an arbitrary $w < w_1 w_2$. Given $w < w_1 w_2$ we can find w'_1 and w'_2 such that $w = w'_1 w'_2$, $w'_1 < w_1$ and $w'_2 < w_2$. Then, by definition of $\lhd_A$, $w'_1 \lhd_{A_f} U$ and, by dot-left of $\lhd_{A_f}$, $w'_1 w'_2 \lhd_{A_f} U$. $w \lhd_{A_f} U$ now follows since $w = w'_1 w'_2$ implies $w =_\subseteq w'_1 w'_2$.

Transitivity: Suppose $w \lhd_A U$ and $U \lhd_A V$. We must show that $w' \lhd_{A_f} V$ holds for an arbitrary $w' < w$. The relation $<$ on neighbourhoods is dense, so given $w' < w$ there exists a w'' such that $w' < w'' < w$. For such a w'', by definition of $\lhd_A$, $w'' \lhd_{A_f} U$. Then, by lemma 4.6, $w' \lhd_{A_f} V$.

Dot-right: Immediately from the definition of $\lhd_A$ and the validity of dot-right of $\lhd_{A_f}$. $\Box$

We have thus proved

PROPOSITION 4.8 *If A is a normed linear space then $\mathcal{L}(A_f) \equiv \langle S, \cdot, =_{\subseteq}, \lhd_{A_f} \rangle$ and $\mathcal{L}(A) \equiv \langle S, \cdot, =_{\subseteq}, \lhd_A \rangle$ are formal topologies.*

The following two easily proved properties will be used implicitly without any remarks in the proofs.

LEMMA 4.9 *Let M be a linear space, w a neighbourhood and U a subset of the base. Then*

$$w \lhd_{M_f} U \Rightarrow w \lhd_M U.$$

LEMMA 4.10 *Let M be a linear space and U a subset of the base. Then*

$$\langle\rangle \lhd_M U \Rightarrow \langle\rangle \lhd_{M_f} U.$$

Another easily proved property is the following:

LEMMA 4.11 *Let M be a linear subspace of the linear space A. Then*

$$w \lhd_{M_f} U \Rightarrow w \lhd_{A_f} U.$$

Proof: $w \lhd_{A_f} U$ follows by induction on the derivation of $w \lhd_{M_f} U$, since $x \in M$ implies $x \in A$. $\Box$

5 Alaoglu's Theorem

As an immediate consequence of the definition of the cover for $\mathcal{L}(A)$, we obtain Alaoglu's theorem, asserting that the unit ball of the space of linear and continuous functionals is compact in the weak topology. We need the following result:

PROPOSITION 5.1 *The cover $\lhd_{A_f}$ is a Stone cover.*

Proof: Suppose $w \lhd_{A_f} U$. The proof is straightforward by induction on the derivation of $w \lhd_{A_f} U$. We will only consider the case when $w \lhd_{A_f} U$ is obtained by *C3* from $w = \langle x \in I \rangle w'$, $I \lhd_{\mathcal{R}_f} V$ and $(\forall J \in V)(\langle x \in J \rangle w' \lhd_{A_f} U)$, for a finite subset V. By induction hypothesis, given an element $J \in V$, there exists a finite subset U_J of U such that $\langle x \in J \rangle w' \lhd_{A_f} U_J$. Let U_0 be the union of all such U_J's. Then, by *C3*, $w \lhd_{A_f} U_0$. $\Box$

Alaoglu's theorem then follows as a corollary:

THEOREM 5.2 *The formal space $\mathcal{L}(A)$ is compact.*

Proof: If $\langle\ \rangle \lhd_A U$ then, by definition of $\lhd_A$, $\langle\ \rangle \lhd_{A_f} U$ and, since $\lhd_{A_f}$ is a Stone cover, $\langle\ \rangle \lhd_{A_f} U_0$ for some finite subset U_0 of U. For such a U_0, by lemma 4.9, $\langle\ \rangle \lhd_A U_0$. $\Box$

6 The Hahn-Banach Theorem

The motivation to our localic formulation of the Hahn-Banach theorem uses a reasoning with points. Let M be a subspace of the seminormed linear space A and F a linear functional on A of norm ≤ 1, then the restriction $F_{|M}$ of F to M is also a linear functional of norm ≤ 1. The Hahn-Banach theorem says that the restriction function

$$F \longmapsto F_{|M}$$

is surjective. For T_1 spaces, surjectivity on points follows from formal injectivity (see for instance MacLane and Moerdijk [10]). Classically, the T_1 property on the space of functionals follows easily from the fact that $\mathcal{R}$ is Hausdorff. The localic Hahn-Banach theorem is then formulated as formal injectivity:

THEOREM 6.1 *Let M be a linear subspace of the linear space A, w an element and U a subset of the base, both living in $\mathcal{L}(M)$. Then*

$$w \lhd_A U \Rightarrow w \lhd_M U.$$

The Hahn-Banach theorem is a conservativity statement. If M is a subspace of A, we say that $\lhd_A$ is *conservative* over $\lhd_M$ if, whenever w is an element and U is a subset living in $\mathcal{L}(M)$, $w \lhd_A U$, then $w \lhd_M U$. Conservativity of $\lhd_{A_f}$ over $\lhd_{M_f}$ is defined in the same way. The definition of $\lhd_A$ in terms of $\lhd_{A_f}$ allows us to replace $\lhd_A$ by $\lhd_{A_f}$ for proving conservativity, since we have:

PROPOSITION 6.2 *If $\lhd_{A_f}$ is conservative over $\lhd_{M_f}$, then $\lhd_A$ is conservative over $\lhd_M$.*

Proof: To prove $w \lhd_M U$ from $w \lhd_A U$, where w and U live in $\mathcal{L}(M)$, we must show that $(\forall w' < w)(w' \lhd_{M_f} U)$ holds. But if $w' < w$ then w' also lives in $\mathcal{L}(M)$ and by definition of $\lhd_A$, $w' \lhd_{M_f} U$. The result then follows by conservativity of $\lhd_{A_f}$ over $\lhd_{M_f}$. $\square$

The Hahn-Banach theorem is thus reduced to:

LEMMA 6.3 *Let M be a linear subspace of the linear space A, w an element and U a subset living in $\mathcal{L}(M)$. Then*

$$w \lhd_{A_f} U \Rightarrow w \lhd_{M_f} U.$$

Now we will use the fact that the definition of $\lhd_{A_f}$ is finitary and show that only a finite number of elements of A is relevant to the proof of $w \lhd_{A_f} U$. Then we argue by induction on this finite set and reduce lemma 6.3 to the following lemma. (If M is a linear space, the notation $[M + x]$ is used for the linear extension of M with x.)

LEMMA 6.4 *Let M be a linear space, w an element and U a subset living in $\mathcal{L}(M)$. Then*

$$w \lhd_{[M+x]_f} U \Rightarrow w \lhd_{M_f} U.$$

To prove lemma 6.3 from lemma 6.4 the two additional lemmas 6.5 and 6.7 are used:

LEMMA 6.5 *Let A be a linear space. Then*

$$w \lhd_{A_f} U \Rightarrow (\exists A_0 \subseteq_f A)(w \lhd_{[A_0]_f} U).$$

Proof: The proof is by induction on the derivation of $w \lhd_{A_f} U$. For each introduction rule the finite set A_0 is constructed, the claim then follows by the same rule using the induction hypothesis.

C1: A_0 can be chosen to be the empty set.

C2: A_0 is taken from the induction hypothesis.

C3: First form the union of the sets from the induction hypothesises, then add the new element x to this union.

C4: Add the new elements x and y to the set from the induction hypothesis.

C5: Add the new element x to the set from the induction hypothesis.

C6: Suppose $w \lhd_{A_f} U$ is derived by *C6* from $x \in A$, $x \in N(1)$ and $\langle x \in (-1,1)\rangle w \lhd_{A_f} U$. By the induction hypothesis there is a finite set A_0' such that $\langle x \in (-1,1)\rangle w \lhd_{[A_0']_f} U$. Let A_0 be the set $A_0' \cup \{x\}$. Clearly $x \in A_0$ and, since $A_0' \subseteq A_0$, we have $\langle x \in (-1,1)\rangle w \lhd_{[A_0]_f} U$. $w \lhd_{[A_0]_f} U$ now follows by *C6*. □

Since $w \lhd_{[A_0]_f} U$ implies $w \lhd_{[B+A_0]_f} U$, we obtain as a corollary to lemma 6.5

COROLLARY 6.6 *Let B be a linear subspace of the linear space A, w an element and U a subset living in $\mathcal{L}(B)$. Then*

$$w \lhd_{A_f} U \Rightarrow (\exists A_0 \subseteq_f A)(w \lhd_{[B+A_0]_f} U).$$

LEMMA 6.7 *Let M be a linear subspace of the linear space B, A_0 a finite set of vectors, w an element and U a subset living in $\mathcal{L}(M)$. Then*

$$w \lhd_{[B+A_0]_f} U \Rightarrow w \lhd_{B_f} U.$$

Proof: The proof is by induction on the length of A_0 (Assuming that A_0 is a list). If A_0 is empty there is nothing to prove. If $A_0 \equiv x :: A_1$, then by induction hypothesis $w \lhd_{[B+x]_f} U$ and, since w and U live in $\mathcal{L}(M)$ they also live in $\mathcal{L}(B)$, so by 6.4, $w \lhd_{B_f} U$. □

Lemma 6.3 now follows by combining corollary 6.6 and lemma 6.7, for $B = M$.

To prove lemma 6.4 a proof of $w \lhd_{[M+x]_f} U$, where w and U live in $\mathcal{L}([M+x])$, must be transformed into a proof of $w \lhd_{M_f} U$. To do this we transform the elements and the subsets occuring in the proof of $w \lhd_{[M+x]_f} U$ in the following way: The sub-basic elements having form $\langle a + tx \in (r,s)\rangle$, where $a \in M$, are transformed into $\langle a \in (r+tq, s+tq)\rangle$. The rational number q can be understood as an approximation of $f(x)$, where f is the linear functional. The proof transformation is then done by induction on the derivation of $w \lhd_{[M+x]_f} U$. There are however some obstacles that have to be overcome.

To justify the definitions and lemmas below, we show what happens if we try to prove $w \lhd_{[M+x]_f} U \Rightarrow \overline{w} \lhd_{M_f} \overline{U}$ (where w and U live in $\mathcal{L}([M+x])$ and $\overline{w}$ and $\overline{U}$ are the corresponding transformed objects), directly by induction on the derivation of $w \lhd_{[M+x]_f} U$. First consider the axiom *C1*. We have $w \in U$, but $w \in U$ does not necessarily imply $\overline{w} \in \overline{U}$. To see this we must have a closer look at the transformation operation. The value of $\overline{w}$ does not only depend on w and the rational number q, but also the proof that w lives in $\mathcal{L}([M+x])$ and there may be different values of $\overline{w}$ for different proofs. This will happen when x is already a member of M, since then we can have $a_1 + t_1x = a_2 + t_2x$ with $a_1 \neq a_2$ and $t_1 \neq t_2$. Now, if w and U live in $\mathcal{L}([M+x])$ and $w \in U$, there are in fact to two proofs that w lives in $\mathcal{L}([M+x])$, the second one comes from the proof that U lives in $\mathcal{L}([M+x])$. If these two proof objects result in different values of $\overline{w}$, then $\overline{w}$ does not belong to $\overline{U}$. But, as we will see, $w \in U$ implies $\overline{w} \in \overline{U} \bigvee x \in M$ and if $x \in M$ then $[M+x]$ is no proper extension of M. Observe here that we do not require decidability of membership of M. Decidability would simplify the proof; if $x \in M$ then $[M+x] = M$ and there is nothing to prove, and if $x \notin M$ then there is only one value of $\bar{w}$ (for a specific rational number q). Then look at axiom *C6*; $w \lhd_{[M+x]_f} U$ is derived from $a+tx \in N(1)$ and $\langle a+tx \in (-1,1)\rangle w \lhd_{[M+x]_f} U$, where $a \in M$. By induction hypothesis $\langle a \in (-1+tq, 1+tq)\rangle \overline{w} \lhd_{M_f} \overline{U}$, but we do not have $a \in N(1)$ so the axiom *C6* is not applicable again. Yet another problem is that an efficient method of finding a sufficient good rational approximation q, used in the transformation from $\mathcal{L}([M+x])$ to $\mathcal{L}(M)$, does not seem to exist. (The problems described above were actually found during the implementation.)

The last problem is solved by quantifying over all rational numbers q. Before solving the other problems, let us define the transformation from $\mathcal{L}([M+x])$ to $\mathcal{L}(M)$ precisely.

DEFINITION 6.8 *Let q be a rational number and let w be an element living in $\mathcal{L}([M+x])$, i.e. we have a proof h_w of $w = \langle a_1+t_1x \in (r_1,s_1), \ldots, a_n+t_nx \in (r_n,s_n)\rangle$, where $a_1,\ldots,a_n \in M$. Then*

$$\overline{w_{q,h_w}} \equiv \langle a_1 \in (r_1+t_1q, s_1+t_1q), \ldots, a_n \in (r_n+t_nq, s_n+t_nq)\rangle$$

and if h_U is a proof that the subset U lives in $\mathcal{L}([M+x])$ then

$$\overline{U_{q,h_U}} \equiv \{\overline{w_{q,h_U(w,h)}} : h \text{ is a proof that } w \in U\}.$$

As indicated before, the problem of $\overline{w_{q,h_w}}$ being dependent of the proof h_w that w lives in $\mathcal{L}([M+x])$ is solved by the following:

LEMMA 6.9 *Let h_w and h_U be proofs that w and U live in $\mathcal{L}([M+x])$, respectively. Then*

$$w \in U \;\Rightarrow\; (\forall q)(\overline{w_{q,h_w}} \in \overline{U_{q,h_U}}) \;\vee\; x \in M.$$

Proof: We start by showing that, if there are two proofs h_1 and h_2 that $\langle a \in I\rangle$ lives in $\mathcal{L}([M+x])$ then $(\forall q)(\overline{\langle a \in I\rangle_{q,h_1}} = \overline{\langle a \in I\rangle_{q,h_2}}) \;\vee\; x \in M$.

By h_1 and h_2 there exists m_1, t_1, m_2, t_2 such that $m_1, m_2 \in M$ and $a = m_1+t_1x = m_2+t_2x$. We now use decidability of equality on the rational numbers. If $t_1 = t_2$ then also $m_1 = m_2$ and thus $(\forall q)(m_1 \in I+(tq,tq) = m_2 \in I+(tq,tq))$. If $t_1 \neq t_2$ then $x = (m_2-m_1)/(t_1-t_2) \in M$.

Then, if h_1 and h_2 are two proofs that w lives in $\mathcal{L}[M+x])$, it is easy to prove, by induction on the length of w, that $(\forall q)(\overline{w_{q,h_1}} = \overline{w_{q,h_2}}) \;\vee\; x \in M$. This is also the situation in the original problem. There are two proofs that w lives in $\mathcal{L}([M+x])$: h_w and $h_U(w,h)$, where h is the proof that w is an element in U. □

To solve the problem arising from axiom *C6*, we first define a new cover relation, which is defined as $\lhd_{[M+x]_f}$ but for the last axiom where only a finite number of elements from [M+x] are allowed.

DEFINITION 6.10 *Let $\boldsymbol{a} \equiv \{a_1,\ldots,a_n\}$ and $\boldsymbol{t} \equiv \{t_1,\ldots,t_n\}$ be finite sequences in M and in Q, respectively, such that $(\forall i)(a_i+t_ix \in N(1))$. Then let $\lhd_{\boldsymbol{a},\boldsymbol{t}}$ be defined as $\lhd_{[M+x]_f}$, with the last axiom replaced by*

$$C6' : \; \frac{a \in \boldsymbol{a} \quad t \in \boldsymbol{t} \quad \langle a+tx \in (-1,1)\rangle w \lhd_{\boldsymbol{a},\boldsymbol{t}} U}{w \lhd_{\boldsymbol{a},\boldsymbol{t}} U}.$$

Then the neighbourhood P in the following definition is used in order to "absorb" the element $\langle a \in (-1+tq, 1+tq)\rangle$ which appears in the induction hypothesis, in the case *C6*.

DEFINITION 6.11 *Let q be a rational number and let $\boldsymbol{a} \equiv \{a_1, \ldots, a_n\}$ and $\boldsymbol{t} \equiv \{t_1, \ldots, t_n\}$ be finite sequences in M and in Q, respectively, such that $(\forall i)(a_i + t_i x \in N(1))$. Then define*

$$P_{q,\boldsymbol{a},\boldsymbol{t}} \equiv \langle a_1 \in (-1+t_1 q, 1+t_1 q), \ldots, a_n \in (-1+t_n q, 1+t_n q)\rangle.$$

The proof that $w \lhd_{M_f} U$ follows from $w \lhd_{[M+x]_f} U$ (lemma 6.4) uses of the following two lemmas. First a proof of $w \lhd_{[M+x]_f} U$ is transformed into a proof of $w \lhd_{\boldsymbol{a},\boldsymbol{t}} U$, for some $\boldsymbol{a}$ and $\boldsymbol{t}$:

LEMMA 6.12 *If $w \lhd_{[M+x]_f} U$, then there exist finite sequences $\boldsymbol{a}$ and $\boldsymbol{t}$ in M and in Q, respectively, such that $(\forall i)(a_i + t_i x \in N(1))$ and $w \lhd_{\boldsymbol{a},\boldsymbol{t}} U$.*

Proof: The proof is by induction on the derivation of $w \lhd_{[M+x]_f} U$. For each introduction rule we find appropriate sequences $\boldsymbol{a}$ and $\boldsymbol{t}$, then the corresponding rule of $\lhd_{\boldsymbol{a},\boldsymbol{t}}$ is applied.

C1: Let $\boldsymbol{a}$ and $\boldsymbol{t}$ be empty.

C2, C4, C5: Take $\boldsymbol{a}$ and $\boldsymbol{t}$ directly from the induction hypothesis.

C3: Form $\boldsymbol{a}$ and $\boldsymbol{t}$ by appending all the sequences from the induction hypothesises.

C6: $w \lhd_{[M+x]_f} U$ is derived from $a+tx \in N(1)$ and $\langle a+tx \in (-1,1)\rangle w \lhd_{[M+x]_f} U$, where $a \in M$. Take the sequences $\boldsymbol{a'}$ and $\boldsymbol{t'}$ from the induction hypothesis, then let $\boldsymbol{a} \equiv a :: \boldsymbol{a'}$ and $\boldsymbol{t} \equiv t :: \boldsymbol{t'}$. *C6'* now gives $w \lhd_{\boldsymbol{a},\boldsymbol{t}} U$. □

Then the proof of $w \lhd_{\boldsymbol{a},\boldsymbol{t}} U$ is then transformed into a proof of $w \lhd_{M_f} U$:

LEMMA 6.13 *Let $\boldsymbol{a}$ and $\boldsymbol{t}$ be finite sequences in M and in Q, respectively, such that $(\forall i)(a_i + t_i x \in N(1))$ and let w be an element and U a subset living in $\mathcal{L}(M)$. Then*

$$w \lhd_{\boldsymbol{a},\boldsymbol{t}} U \;\Rightarrow\; w \lhd_{M_f} U \;\vee\; x \in M.$$

Proof of lemma 6.4: Suppose w and U live in $\mathcal{L}([M+x])$ and $w \lhd_{[M+x]_f} U$. By combining the lemmas 6.12 and 6.13, we get $w \lhd_{M_f} U \;\vee\; x \in M$. If $w \lhd_{M_f} U$, there is nothing more to prove. If $x \in M$ then $[M+x] \subseteq M$ and, since $w \lhd_{[M+x]_f} U$, $w \lhd_{M_f} U$ follows. □

The Hahn-Banach theorem is thus reduced to lemma 6.13. Before proving this lemma some intermediate results are needed. The first one is the core of the Hahn-Banach theorem; intuitively it tells us that, if f is a linear functional on M, then we can find a rational approximation q for the value of $f(x)$. The proof is given later.

LEMMA 6.14 *Let $\boldsymbol{a}$ and $\boldsymbol{t}$ be finite sequences in M and in Q, respectively, such that $(\forall i)(a_i + t_i x \in N(1))$. Then*

$$\langle\rangle \lhd_{M_f} \{P_{q,\boldsymbol{a},\boldsymbol{t}} : q \in Q\}.$$

The following two lemmas describe some properties of the transformation of neighbourhoods from $\mathcal{L}([M+x])$ to $\mathcal{L}(M)$ and the relation $\leq$ between neighbourhoods.

LEMMA 6.15 *Let h_{w_1} be a proof that w_1 lives in $\mathcal{L}([M+x])$. Then*

$$w_1 \leq w_2$$
$$\Rightarrow$$
$$(\exists h_{w_2} : w_2 \textit{ lives in } \mathcal{L}([M+x]))(\forall q)(\overline{w_1}_{q,h_{w_1}} \leq \overline{w_2}_{q,h_{w_2}}) \;\vee\; x \in M.$$

Proof: We show that if $w_1 \leq \langle a \in I\rangle$ then

$$(\exists h : \ \langle a \in I\rangle \ \textit{lives in} \ \mathcal{L}([M+x]))(\forall q)(\overline{w_{1q,h_{w_1}}} \leq \overline{\langle a \in I\rangle_{q,h}}) \ \vee \ x \in M.$$

The claim then follows by induction on the length of w_2.

By the definition of $\leq$, there exists $\langle a_1 \in I_1, \ldots, a_n \in I_n\rangle \subseteq w_1$ such that $a_1 = \cdots = a_n = a$ and $I_1 \cdot \cdots \cdot I_n \leq I$. Since w_1 lives in $\mathcal{L}([M+x])$ there exist elements $m_1, \ldots, m_n$ in M and rational numbers $t_1, \ldots, t_n$ such that $a_1 = m_1 + t_1 x, \ldots, a_n = m_n + t_n x$. Now there are two cases. If $t_1 = \cdots = t_n = t$ then also $m_1 = \cdots = m_n = m$ and, since $I_1 \cdot \cdots \cdot I_n \leq I$, $(I_1 + (tq,tq)) \cdot \cdots \cdot (I_n + (tq,tq)) \leq I + (tq,tq)$. If there exists an i such that $t_i \neq t$ then, since $m_i + t_i x = m + tx$, $x = (m - m_i)/(t_i - t) \in M$. $\square$

LEMMA 6.16 Localisation *holds for* $\leq$*:*

$$w_1 \leq w_2 \Rightarrow w_1 \cdot w \leq w_2 \cdot w.$$

Proof: Follows from the definition of $\leq$, by using properties of the subset relation and intersection of rational intervals. $\square$

The next lemma is the one that actually performs the proof transformation from $\mathcal{L}([M+x])$ to $\mathcal{L}(M)$.

LEMMA 6.17 *Let* $\boldsymbol{a}$ *and* $\boldsymbol{t}$ *be finite sequences in* M *and in* Q*, respectively, such that* $(\forall i)(a_i + t_i x \in N(1))$*, and let* h_w *and* h_U *be proofs that* w *is an element and* U *a subset living in* $\mathcal{L}([M+x])$*, respectively. Then*

$$w \lhd_{\boldsymbol{a},\boldsymbol{t}} U \ \Rightarrow \ (\forall q)(\overline{w_{q,h_w}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}) \ \vee \ x \in M.$$

Proof: By induction on the derivation of $w \lhd_{\boldsymbol{a},\boldsymbol{t}} U$.

C1: $w \in U$ and, by lemma 6.9,

$$(\forall q)(\overline{w_{q,h_w}} \in \overline{U_{q,h_U}}) \ \vee \ x \in M.$$

If $(\forall q)(\overline{w_{q,h_w}} \in \overline{U_{q,h_U}})$ then, by reflexivity and dot-left,

$$(\forall q)(\overline{w_{q,h_w}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}).$$

If $x \in M$ then there is nothing more to prove.

C2: $w \leq w'$ and $w' \lhd_{\boldsymbol{a},\boldsymbol{t}} U$. By lemma 6.15,

$$(\exists h_{w'} : w' \ \textit{lives in} \ \mathcal{L}([M+x]))(\forall q)(\overline{w_{q,h_w}} \leq \overline{w'_{q,h_{w'}}}) \ \vee \ x \in M.$$

If $(\exists h_{w'} : w' \ \textit{lives in} \ \mathcal{L}([M+x]))(\forall q)(\overline{w_{q,h_w}} \leq \overline{w'_{q,h_{w'}}})$ then for such an $h_{w'}$, by induction hypothesis,

$$(\forall q)(\overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}) \ \vee \ x \in M.$$

If $(\forall q)(\overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}})$ then, by lemma 6.16 on $\overline{w_{q,h_w}} \leq \overline{w'_{q,h_{w'}}}$,

$$\overline{w_{q,h_w}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \leq \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}}$$

and then, by *C2'*,

$$(\forall q)(\overline{w_{q,h_w}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}).$$

C3: $I \lhd_{R_f} V$ and $(\forall J \in V)(\langle a + tx \in J\rangle w' \lhd_{\boldsymbol{a},\boldsymbol{t}} U)$. By induction hypothesis

$$(\forall J \in V)((\forall q)(\langle a \in J + (tq, tq)\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}) \ \vee \ x \in M)$$

and, since V is finite, the disjunction can be moved outside the universal quantifier. We get

$$(\forall J \in V)(\forall q)(\langle a \in J + (tq, tq)\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}) \ \vee \ x \in M$$

which is the same as

$$(\forall q)(\forall J \in V + (tq, tq))(\langle a \in J\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}) \ \vee \ x \in M.$$

If $(\forall q)(\forall J \in V + (tq, tq))(\langle a \in J\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}})$, then, by *C3'*, since $I \lhd_{R_f} V$ implies $I + (tq, tq) \lhd_{R_f} V + (tq, tq)$ (see lemma 3.7),

$$(\forall q)(\langle a \in I + (tq, tq)\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}).$$

C4: $\langle a + t_a x + b + t_b x \in I_a + I_b\rangle w' \lhd_{\boldsymbol{a},\boldsymbol{t}} U$ so, by induction hypothesis,

$$\begin{gathered}(\forall q)(\langle a + b \in I_a + (t_a q, t_b q) + I_b + (t_a q, t_b q)\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}) \\ \vee \\ x \in M.\end{gathered}$$

If $(\forall q)(\langle a + b \in I_a + (t_a q, t_b q) + I_b + (t_a q, t_b q)\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}})$ then, by *C4'*,

$$(\forall q)(\langle a \in I_a + (t_a q, t_b q), b \in I_b + (t_a q, t_b q)\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}).$$

C5: $\langle r(a + tx) \in rI\rangle w' \lhd_{\boldsymbol{a},\boldsymbol{t}} U$ so, by induction hypothesis,

$$(\forall q)(\langle ra \in r(I + (tq, tq))\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}) \ \vee \ x \in M.$$

If $(\forall q)(\langle ra \in r(I + (tq, tq))\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}})$ then, by *C5'*,

$$(\forall q)(\langle a \in I + (tq, tq)\rangle \overline{w'_{q,h_{w'}}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}).$$

C6: $a + tx \in N(1)$ and $\langle a + tx \in (-1, 1)\rangle w \lhd_{\boldsymbol{a},\boldsymbol{t}} U$ so, by induction hypothesis,

$$(\forall q)(\langle a \in (-1 + tq, 1 + tq)\rangle \overline{w_{q,h_w}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}) \ \vee \ x \in M.$$

If $(\forall q)(\langle a \in (-1 + tq, 1 + tq)\rangle \overline{w_{q,h_w}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}})$ then, since $\langle a \in (-1 + tq, 1 + tq)\rangle$ is a member of $P_{q,\boldsymbol{a},\boldsymbol{t}}$,

$$(\forall q)(\overline{w_{q,h_w}} \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} \overline{U_{q,h_U}}). \ \square$$

If w lives in $\mathcal{L}(M)$, then there is a trivial proof h that w lives in $\mathcal{L}([M + x])$ (for which the coefficient of x is always zero). For this proof object w will not change during the transformation from $\mathcal{L}([M + x])$ to $\mathcal{L}(M)$ $((\forall q)(w = \overline{w_{q,h}}))$. So as a corollary to lemma 6.17 we obtain:

COROLLARY 6.18 *Let* $\boldsymbol{a}$ *and* $\boldsymbol{t}$ *be finite sequences in* M *and in* Q*, respectively, such that* $(\forall i)(a_i + t_i x \in N(1))$. *Let* w *be an element and* U *a subset living in* $\mathcal{L}(M)$. *Then*

$$w \lhd_{\boldsymbol{a},\boldsymbol{t}} U \;\Rightarrow\; (\forall q)(w \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} U) \;\vee\; x \in M.$$

Proof of lemma 6.13: Suppose $w \lhd_{\boldsymbol{a},\boldsymbol{t}} U$. Then by corollary 6.18,

$$(\forall q)(w \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} U) \;\vee\; x \in M.$$

If $(\forall q)(w \cdot P_{q,\boldsymbol{a},\boldsymbol{t}} \lhd_{M_f} U)$ then, using transitivity,

$$w \cdot \{P_{q,\boldsymbol{a},\boldsymbol{t}} : q \in Q\} \lhd_{M_f} U.$$

Moreover from lemma 6.14 and localisation we get

$$w \lhd_{M_f} w \cdot \{P_{q,\boldsymbol{a},\boldsymbol{t}} : q \in Q\}$$

so, using transitivity again,

$$w \lhd_{M_f} U. \;\square$$

The Hahn-Banach theorem is now reduced to lemma 6.14. To prove lemma 6.14, we start by restricting it to the case when all rational numbers in the sequence $\boldsymbol{t}$ are distinct from 0.

LEMMA 6.19 *Let* $\boldsymbol{a}$ *and* $\boldsymbol{t}$ *be finite sequences in* M *and in* Q*, respectively, such that* $(\forall i)(a_i + t_i x \in N(1)\ \&\ t_i \neq 0)$. *Then*

$$\langle\rangle \lhd_{M_f} \{P_{q,\boldsymbol{a},\boldsymbol{t}} : q \in Q\}.$$

Proof of lemma 6.14: Split the sequences $\boldsymbol{a}$,$\boldsymbol{t}$ into $\boldsymbol{a}'$,$\boldsymbol{t}'$ and $\boldsymbol{a}''$,$\boldsymbol{t}''$ in such a way that $\boldsymbol{a}'$,$\boldsymbol{t}'$ contains all elements a_i,t_i for which $t_i \neq 0$ and $\boldsymbol{a}''$,$\boldsymbol{t}''$ contains all elements a_i,t_i for which $t_i = 0$. To $\boldsymbol{a}'$ and $\boldsymbol{t}'$ lemma 6.19 is applied. By induction on the length of $\boldsymbol{a}''$ and $\boldsymbol{t}''$ and by using *C6*, we have $\langle\rangle \lhd_{M_f} \{\textit{the list of all } a_i \in (-1,1) \textit{ for which } t_i = 0\}$. $\langle\rangle \lhd_{M_f} \{P_{q,\boldsymbol{a},\boldsymbol{t}} : q \in Q\}$ then basically follows by dot-right. $\square$

By transitivity, lemma 6.19 follows from the next two lemmas.

LEMMA 6.20 *Let* $\boldsymbol{a}$ *and* $\boldsymbol{t}$ *be non empty sequences in* M *and in* Q*, respectively, such that* $(\forall i)(a_i + t_i x \in N(1)\ \&\ t_i \neq 0)$. *Then*

$$\begin{aligned}\langle\rangle \quad \lhd_{M_f} \quad & \{\langle a_1/t_1 \in (r_1,s_1),\ldots,a_n/t_n \in (r_n,s_n)\rangle : \\ & max(s_i - 1/|t_i|) \leq min(r_i + 1/|t_i|)\}.\end{aligned}$$

LEMMA 6.21 *Let* $\boldsymbol{a}$ *and* $\boldsymbol{t}$ *be non empty sequences in* M *and in* Q*, respectively, such that* $(\forall i)(a_i + t_i x \in N(1)\ \&\ t_i \neq 0)$. *Then*

$$\begin{gathered}\{\langle a_1/t_1 \in (r_1,s_1),\ldots,a_n/t_n \in (r_n,s_n)\rangle : \\ max(s_i - 1/|t_i|) \leq min(r_i + 1/|t_i|)\} \\ \lhd_{M_f} \\ \{P_{q,\boldsymbol{a},\boldsymbol{t}} : q \in Q\}.\end{gathered}$$

Several lemmas are used in order to prove the lemmas 6.20 and 6.21.

LEMMA 6.22 *Given x and y in M and a rational interval (p,q), we have*

(a) $\langle x+y \in (p,q)\rangle \lhd_M \{\langle x \in (r,s), y \in (r',s')\rangle : p \leq r+r' \ \&\ s+s' \leq q\}$

(b) $\langle x-y \in (p,q)\rangle \lhd_M \{\langle x \in (r,s), y \in (r',s')\rangle : p \leq r-s' \ \&\ s-r' \leq q\}$.

Proof: First we show that, given an element x and a positive rational d, we have

$$(c) \quad \langle\rangle \lhd_{M_f} \{\langle x \in (r,s)\rangle : s-r \leq d\}.$$

By $N2$ there exists a rational number q such that $x \in N(q)$. $C6$ and the rules of N then gives

$$\langle\rangle \lhd_{M_f} \langle x \in (-q,q)\rangle.$$

By lemma 3.8 there exists a finite subset U of rational intervals, not longer than d, such that

$$(-q,q) \lhd_{R_f} U.$$

For all I in U, $\langle x \in I\rangle \in \{\langle x \in (r,s)\rangle : s-r \leq d\}$ and thus

$$\langle x \in I\rangle \lhd_{M_f} \{\langle x \in (r,s)\rangle : s-r \leq d\}.$$

So, by $C3$,

$$\langle x \in (-q,q)\rangle \lhd_{M_f} \{\langle x \in (r,s)\rangle : s-r \leq d\}.$$

The claim now follows by transitivity.

(a) By definition of $\lhd_M$,

$$\langle x+y \in (p,q)\rangle \lhd_M \{\langle x+y \in (p',q')\rangle : p < p' \ \&\ q' < q\}.$$

By transitivity it is enough to show that

$$\begin{gathered}\{\langle x+y \in (p',q')\rangle : p < p' \ \&\ q' < q\} \\ \lhd_M \\ \{\langle x \in (r,s), y \in (r',s')\rangle : p \leq r+r' \ \&\ s+s' \leq q\}.\end{gathered}$$

So consider an element $\langle x+y \in (p',q')\rangle$, where $p < p'$ and $q' < q$, and let

$$\begin{cases} d \equiv min(p'-p, q-q')/2 \\ V_x \equiv \{\langle x \in (r,s)\rangle : s-r \leq d\} \\ V_y \equiv \{\langle y \in (r',s')\rangle : s'-r' \leq d\}. \end{cases}$$

Then, since $d > 0$, by (c), $\langle\rangle \lhd_M V_x$ and $\langle\rangle \lhd_M V_y$. Stability now gives

$$\begin{gathered}\langle x+y \in (p',q')\rangle \\ \lhd_M \\ \langle x+y \in (p',q')\rangle \cdot V_x \cdot V_y \\ \lhd_M \\ \{\langle x \in (r,s), y \in (r',s'), x+y \in (p',q')\rangle : s-r \leq d \ \&\ s'-r' \leq d\}.\end{gathered}$$

Now consider an element $\langle x \in (r,s), y \in (r',s'), x+y \in (p',q')\rangle$, where $s-r \leq d$ and $s'-r' \leq d$. By $C4$,

$$\begin{gathered}\langle x \in (r,s), y \in (r',s'), x+y \in (p',q')\rangle \\ \lhd_{M_f} \\ \langle x+y \in (r+r', s+s'), x+y \in (p',q')\rangle.\end{gathered}$$

There are three cases to consider. If $r+r' < p$ then $s+s' \leq p'$ and

$$\begin{gathered}\langle x+y \in (r+r', s+s'), x+y \in (p', q')\rangle \\ \lhd_{M_f} \\ \langle x+y \in (p', p')\rangle \\ \lhd_{M_f} \\ \{\langle x \in (r,s), y \in (r',s')\rangle : p \leq r+r' \ \& \ s+s' \leq q\},\end{gathered}$$

the last step since $p' \leq p'$. Similarly, if $q < s+s'$ then $r+r' \geq q'$ and

$$\begin{gathered}\langle x+y \in (r+r', s+s'), x+y \in (p', q')\rangle \\ \lhd_{M_f} \\ \langle x+y \in (q', q')\rangle \\ \lhd_{M_f} \\ \{\langle x \in (r,s), y \in (r',s')\rangle : p \leq r+r' \ \& \ s+s' \leq q\}.\end{gathered}$$

Otherwise, $p \leq r+r'$ and $s+s' \leq q$, and then

$$\begin{gathered}\langle x \in (r,s), y \in (r',s'), x+y \in (p'q')\rangle \\ \lhd_{M_f} \\ \langle x \in (r,s), y \in (r',s')\rangle \\ \lhd_{M_f} \\ \{\langle x \in (r,s), y \in (r',s')\rangle : p \leq r+r' \ \& \ s+s' \leq q\},\end{gathered}$$

the last step is by reflexivity.

(b) From (a) we get

$$\langle x-y \in (p,q)\rangle \lhd_M \{\langle x \in (r,s), -y \in (r',s')\rangle : p \leq r+r' \ \& \ s+s' \leq q\}.$$

We proceed by using transitivity. So take an element $\langle x \in (r,s), -y \in (r',s')\rangle$ of $\{\langle x \in (r,s), -y \in (r',s')\rangle : p \leq r+r' \ \& \ s+s' \leq q\}$ and let

$$\begin{cases} r'' \equiv -s' \\ s'' \equiv -r'. \end{cases}$$

By *C5*, we get

$$\langle x \in (r,s), -y \in (r',s')\rangle \lhd_M \langle x \in (r,s), y \in (r'',s'')\rangle$$

and the claim follows since $\langle x \in (r,s), y \in (r'',s'')\rangle$ is an element of $\{\langle x \in (r,s), y \in (r'',s'')\rangle : p \leq r-s'' \ \& \ s-r'' \leq q\}$. $\Box$

LEMMA 6.23 *Let a and b be elements in M, t_a and t_b rational numbers distinct from* 0 *such that $a+t_a x \in N(1)$ and $b+t_b x \in N(1)$. Then*

$$a/t_a - b/t_b \in N(1/|t_a| + 1/|t_b|).$$

Proof: Easily proved using the rules of N. $\Box$

LEMMA 6.24 *Let a and b be elements in M, t_a and t_b rational numbers distinct from* 0 *such that $a+t_a x \in N(1)$ and $b+t_b x \in N(1)$. Then*

$$\begin{aligned}\langle\rangle \quad \lhd_M \quad & \{\langle a/t_a \in (r,s), b/t_b \in (r',s')\rangle : \\ & -(1/|t_a| + 1/|t_b|) \leq r-s' \ \& \ s-r' \leq 1/|t_a| + 1/|t_b|\}.\end{aligned}$$

Proof: Lemma 6.23 gives

$$a/t_a - b/t_b \in N(1/|t_a| + 1/|t_b|)$$

then, by *C6*, the rules of N and the fact that $w \lhd_{M_f} U \Rightarrow w \lhd_M U$,

$$\langle\rangle \lhd_M \langle a/t_a - b/t_b \in (-(1/|t_a| + 1/|t_b|), 1/|t_a| + 1/|t_b|)\rangle.$$

The claim now follows by transitivity and lemma 6.22(b). □

LEMMA 6.25 *Let a be an element in M and t a rational number distinct from* 0 *such that $a + tx \in N(1)$. Then*

$$\langle\rangle \lhd_M \{\langle a/t \in (r,s)\rangle : s - r \leq 1/|t| + 1/|t|\}.$$

Proof: By *N2* there exists a rational number q such that $a/t \in N(q)$ and thus, by *C6* and the rules of N,

$$\langle\rangle \lhd_{M_f} \{\langle a/t \in (-q,q)\rangle : q \in Q\}.$$

By transitivity it is now enough to show that

$$\langle a/t \in (-q,q)\rangle \lhd_{M_f} \{\langle a/t \in (r,s)\rangle : s - r \leq 1/|t| + 1/|t|\},$$

which follows from *C3*, using the fact that there exists a finite set V of rational intervals, not longer than $1/|t| + 1/|t|$, such that $(-q,q) \lhd_{R_f} V$ (see lemma 3.8). □

LEMMA 6.26 *Let $a, a_1, \ldots, a_n$ be elements in M, $t, t_1, \ldots, t_n$ rational numbers distinct from* 0 *such that $a + tx \in N(1)$ and $(\forall i)(a_i + t_i x \in N(1))$. Then*

$$\begin{aligned} \langle\rangle \quad \lhd_M \quad & \{\langle a/t \in (r,s), a_1/t_1 \in (r_1,s_1), \ldots, a_n/t_n \in (r_n,s_n)\rangle : \\ & s - r \leq 1/|t| + 1/|t| \ \& \\ & (\forall i)(s - r_i \leq 1/|t| + 1/|t_i| \ \& \ s_i - r \leq 1/|t_i| + 1/|t|)\}. \end{aligned}$$

Proof: By induction on the length of the sequences. If $n = 0$ then the claim follows by lemma 6.25. Otherwise let

$$\left\{ \begin{array}{lll} \mathcal{A} & \equiv & \{\langle a/t \in (r',s'), a_1/t_1 \in (r_1,s_1)\rangle : \\ & & \ s' - r_1 \leq 1/|t| + 1/|t_1| \ \& \ s_1 - r' \leq 1/|t_1| + 1/|t|\} \\ \mathcal{B} & \equiv & \{\langle a/t \in (r'',s''), a_2/t_2 \in (r_2,s_2), \ldots, a_n/t_n \in (r_n,s_n)\rangle : \\ & & \ s'' - r'' \leq 1/|t| + 1/|t| \ \& \\ & & \ (\forall i)(s'' - r_i \leq 1/|t| + 1/|t_i| \ \& \ s_i - r'' \leq 1/|t_i| + 1/|t|)\} \\ \mathcal{C} & \equiv & \{\langle a/t \in (r,s), a_1/t_1 \in (r_1,s_1), \ldots, a_n/t_n \in (r_n,s_n)\rangle : \\ & & \ s - r \leq 1/|t| + 1/|t| \ \& \\ & & \ (\forall i)(s - r_i \leq 1/|t| + 1/|t_i| \ \& \ s_i - r \leq 1/|t_i| + 1/|t|)\}. \end{array} \right.$$

Lemma 6.24 gives $\langle\rangle \lhd_{M_f} \mathcal{A}$ and, by induction hypothesises, $\langle\rangle \lhd_{M_f} \mathcal{B}$. Thus, by dot-right, $\langle\rangle \lhd_{M_f} \mathcal{A} \cdot \mathcal{B}$. By transitivity, it is enough to show $\mathcal{A} \cdot \mathcal{B} \lhd_{M_f} \mathcal{C}$. So take a

$$\begin{aligned} w \ \equiv \ & \langle a/t \in (r',s'), a_1/t_1 \in (r_1,s_1)\rangle \cdot \\ & \langle a/t \in (r'',s''), a_2/t_2 \in (r_2,s_2), \ldots, a_n/t_n \in (r_n,s_n)\rangle \in \mathcal{A} \cdot \mathcal{B} \end{aligned}$$

and let

$$w' \equiv \langle a/t \in (r,s), a_1/t_1 \in (r_1,s_1), \ldots, a_n/t_n \in (r_n,s_n)\rangle,$$

where $(r,s) = (r',s') \cdot (r'',s'')$. Then $w \leq w'$ and $w' \in \mathcal{C}$, since

$$\begin{cases} s-r &= min(s',s'') - max(r',r'') \leq s''-r'' \leq 1/|t| + 1/|t| \\ s-r_1 &= min(s',s'') - r_1 \leq s'-r_1 \leq 1/|t| + 1/|t_1| \\ s-r_i &= min(s',s'') - r_i \leq s''-r'_i \leq 1/|t| + 1/|t_i|, & i \geq 2 \\ s_1-r &= s_1 - max(r',r'') \leq s_1 - r' \leq 1/|t_1| + 1/|t| \\ s_i - r &= s_i - max(r',r'') \leq s_i - r'' \leq 1/|t_i| + 1/|t|, & i,j \geq 2. \end{cases}$$

Thus, by *C2*, $w \lhd_{M_f} \mathcal{C}$. □

LEMMA 6.27 *Let $a_1, \ldots, a_n$ be elements in M, $t_1, \ldots, t_n$ rational numbers distinct from 0 such that $(\forall i)(a_i + t_i x \in N(1))$. Then*

$$\begin{aligned} \langle\rangle \quad \lhd_M \quad & \{\langle a_1/t_1 \in (r_1,s_1), \ldots, a_n/t_n \in (r_n,s_n)\rangle : \\ & (\forall i,j)(s_i - r_j \leq 1/|t_i| + 1/|t_j|)\}. \end{aligned}$$

Proof: By induction on the length of the sequences. The base case follows by reflexivity. For the inductive step let

$$\begin{cases} \mathcal{A} \equiv & \{\langle a_1/t_1 \in (r'_1,s'_1), \ldots, a_n/t_n \in (r'_n,s'_n)\rangle : \\ & s'_1 - r'_1 \leq 1/|t_1| + 1/|t_1| \ \& \\ & (\forall i \geq 2)(s'_1 - r'_i \leq 1/|t_1| + 1/|t_i| \ \& \ s'_i - r'_1 \leq 1/|t_i| + 1/|t|)\} \\ \mathcal{B} \equiv & \{\langle a_2/t_2 \in (r''_2,s''_2), \ldots, a_n/t_n \in (r''_n,s''_n)\rangle : \\ & (\forall i,j \geq 2)(s''_i - r''_j \leq 1/|t_i| + 1/|t_j|)\} \\ \mathcal{C} \equiv & \{\langle a_1/t_1 \in (r_1,s_1), \ldots, a_n/t_n \in (r_n,s_n)\rangle : \\ & (\forall i,j)(s_i - r_j \leq 1/|t_i| + 1/|t_j|)\}. \end{cases}$$

Lemma 6.26 gives $\langle\rangle \lhd_{M_f} \mathcal{A}$ and, by induction hypothesises, $\langle\rangle \lhd_{M_f} \mathcal{B}$. Thus, by dot-right, $\langle\rangle \lhd_{M_f} \mathcal{A} \cdot \mathcal{B}$. By transitivity, it is enough to show $\mathcal{A} \cdot \mathcal{B} \lhd_{M_f} \mathcal{C}$. So take a

$$\begin{aligned} w \quad \equiv \quad & \langle a_1/t_1 \in (r'_1,s'_1), \ldots, a_n/t_n \in (r'_n,s'_n)\rangle \cdot \\ & \langle a_2/t_2 \in (r''_2,s''_2), \ldots, a_n/t_n \in (r''_n,s''_n)\rangle \in \mathcal{A} \cdot \mathcal{B} \end{aligned}$$

and let

$$w' \equiv \langle a_1/t_1 \in (r_1,s_1), \ldots, a_n/t_n \in (r_n,s_n)\rangle,$$

where

$$\begin{cases} (r_1,s_1) \equiv (r'_1,s'_1) \\ (r_i,s_i) \equiv (r'_i,s'_i) \cdot (r''_i,s''_i), \quad \text{for } i \geq 2. \end{cases}$$

Then $w \leq w'$ and $w' \in \mathcal{C}$, since

$$\begin{cases} s_1 - r_1 &= s'_1 - r'_1 \leq 1/|t_1| + 1/|t_1| \\ s_1 - r_i &= s'_1 - max_{i \geq 2}(r'_i, r''_i) \leq s'_1 - r'_i \leq 1/|t_1| + 1/|t_i|, & i \geq 2 \\ s_1 - r_i &= min_{i \geq 2}(s'_i, s''_i) - r'_i \leq s'_i - r'_1 \leq 1/|t_i| + 1/|t_1|, & i \geq 2 \\ s_i - r_j &= min_{i \geq 2}(s'_i, s''_i) - max_{j \geq 2}(r'_j, r''_j) \leq \\ & \quad s''_i - r''_j \leq 1/|t_i| + 1/|t_j|, & i,j \geq 2. \end{cases}$$

Thus, by *C2*, $w \lhd_{M_f} \mathcal{C}$. □

LEMMA 6.28 *Let n be a natural number greater than 0, $a_1, \ldots, a_n$ be elements in M, $t_1, \ldots, t_n$ rational numbers distinct from 0 and $(r_1, s_1), \ldots, (r_n, s_n)$ rational intervals. Then*

$$(\forall i, j)(s_i - r_j \leq 1/|t_i| + 1/|t_j|) \;\Rightarrow\; max(s_i - 1/|t_i|) \leq min(r_i + 1/|t_i|).$$

Proof: Easily proved by induction on the length of the sequences. □

Proof of lemma 6.20: Lemma 6.27 gives

$$\begin{array}{ccc} \langle\rangle & \lhd_M & \{\langle a_1/t_1 \in (r_1, s_1), \ldots, a_n/t_n \in (r_n, s_n)\rangle : \\ & & (\forall i, j)(s_i - r_j \leq 1/|t_i| + 1/|t_j|)\} \end{array}$$

and by lemma 6.28,

$$(\forall i, j)(s_i - r_j \leq 1/|t_i| + 1/|t_j|) \;\Rightarrow\; max(s_i - 1/|t_i|) \leq min(r_i + 1/|t_i|)$$

which means that

$$\begin{array}{c} \{\langle a_1/t_1 \in (r_1, s_1), \ldots, a_n/t_n \in (r_n, s_n)\rangle : \\ (\forall i, j)(s_i - r_j \leq 1/|t_i| + 1/|t_j|)\} \\ \subseteq \\ \{\langle a_1/t_1 \in (r_1, s_1), \ldots, a_n/t_n \in (r_n, s_n)\rangle : \\ max(s_i - 1/|t_i|) \leq min(r_i + 1/|t_i|)\}. \end{array}$$

The claim now follows since, in general, $a \lhd U \;\&\; U \subseteq V \Rightarrow a \lhd V$. □

LEMMA 6.29 *Let a be an element in M, t a rational number distinct from 0, r, s and q rational numbers such that $s - 1/|t| \leq q \leq r + 1/|t|$. Then*

$$\langle a/t \in (r, s)\rangle \lhd_{M_f} \langle a \in (-1 + tq, 1 + tq)\rangle.$$

Proof: From $s - 1/|t| \leq q \leq r + 1/|t|$, we get

$$t(r, s) \leq (-1 + tq, 1 + tq)$$

and, since $t(a/t) = a$,

$$\langle t(a/t) \in t(r, s)\rangle \leq \langle a \in (-1 + tq, 1 + tq)\rangle.$$

C2 now gives

$$\langle t(a/t) \in t(r, s)\rangle \lhd_{M_f} \langle a \in (-1 + tq, 1 + tq)\rangle$$

and by *C5*

$$\langle a/t \in (r, s)\rangle \lhd_{M_f} \langle a \in (-1 + tq, 1 + tq)\rangle. \; □$$

LEMMA 6.30 *Let $\{a_1, \ldots, a_n\}$ and $\{t_1, \ldots, t_n\}$ be non empty sequences in M and in Q, respectively, such that $(\forall i)(a_i + t_i x \in N(1) \;\&\; t_i \neq 0)$. Let q be a rational number such that $max(s_i - 1/|t_i|) \leq q \leq min(r_i + 1/|t_i|)$. Then*

$$\begin{array}{c} \langle a_1/t_1 \in (r_1, s_1), \ldots, a_n/t_n \in (r_n, s_n)\rangle \\ \lhd_{M_f} \\ \langle a_1 \in (-1 + t_1 q, 1 + t_1 q), \ldots, a_n \in (-1 + t_n q, 1 + t_n q)\rangle. \end{array}$$

Proof: The proof is by induction on the lenght of the sequences. We have

$$s_1 - 1/|t_1| \leq max(s_i - 1/|t_i|) \leq q \leq min(r_i + 1/|t_i|) \leq r_1 + 1/|t_1|.$$

So if $n = 1$, there is nothing more to prove. Otherwise, lemma 6.24 gives

$$\langle a_1/t_1 \in (r_1, s_1)\rangle \lhd_{M_f} \langle a_1 \in (-1 + t_1 q, 1 + t_1 q)\rangle.$$

Moreover

$$\begin{array}{l} max_{i\geq 2}(s_i - 1/|t_i|) \leq \\ max(s_i - 1/|t_i|) \leq \\ q \leq \\ min(r_i + 1/|t_i|) \leq \\ min_{i\geq 2}(r_i + 1/|t_i|) \end{array}$$

and, by induction hypothesis,

$$\begin{array}{c} \langle a_2/t_2 \in (r_2, s_2), \ldots, a_n/t_n \in (r_n, s_n)\rangle \\ \lhd_{M_f} \\ \langle a_2 \in (-1 + t_2 q, 1 + t_2 q), \ldots, a_n \in (-1 + t_n q, 1 + t_n q)\rangle. \end{array}$$

The claim now follows, since in general,

$$x_1 \lhd y_1 \;\&\; x_2 \lhd y_2 \;\Rightarrow\; x_1 \cdot x_2 \lhd y_1 \cdot y_2. \;\square$$

Proof of lemma 6.21: Given an element

$$\langle a_1/t_1 \in (r_1, s_1), \ldots, a_n/t_n \in (r_n, s_n)\rangle$$

in

$$\begin{array}{c} \{\langle a_1/t_1 \in (r_1, s_1), \ldots, a_n/t_n \in (r_n, s_n)\rangle : \\ max(s_i - 1/|t_i|) \leq min(r_i + 1/|t_i|)\}, \end{array}$$

let $q \equiv max(s_i - 1/|t_i|)$. Then the claim follows easily using lemma 6.30. $\square$

7 The Implementation

In this section we give some implementation specific details.

7.1 Description of the Proof-Checker Half

The implementation has been done in the proof-checker Half, developed by Thierry Coquand, using a type-checker and an `emacs`-interface implemented by Dan Synek.

The Half system is a successor to ALF [11]. It is a logical framework based on Martin-Löf's polymorphic type theory with one universe [12], extended by a *theory* mechanism (similar to the theory mechanism in PVS [16]) and *let-expressions* (cf. [6, 3, 1]).

The system has three levels; **Set**, **Type** and **Kind**. **Set** is an element and a subset of **Type**. Elements can be formed in both **Set** and **Type**; both **Set** and **Type** are closed under function types (Π-types) and disjoint union (Σ-types) and allow recursive definitions. There is also a type **Theory** for theories. **Kind** consists of the types **Set**, **Type** and **Theory**, and function types.

A proof (program) in Half consists of a list of definitions and proofs, having the form $f(x_1 : T_1, \ldots, x_n : T_n) = e : T$, where the type T_i may depend on the parameters $x_1, \ldots, x_{i-1}$ and e is an expression of type T.

The Π-type is used for expressing dependent function spaces. Given two types A and B, the Π-type for functions from A to B is written $(x : A) \rightarrow B$. Elements of $(x : A) \rightarrow B$ are functions $\lambda x \rightarrow e$, where the abstracted variable x has type A and e is an expression of type B. The elimination form for elements of Π-types is application.

A recursive data type is defined using the reserved word **data**:

$$\begin{array}{l}\mathbf{data}\{\\ \quad c_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}),\\ \qquad\qquad \vdots\\ \quad c_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk})\},\end{array}$$

where A_{ij} is an arbitrary type. Elements are introduced using the constructors c_i

$$c_i\ a_{i1} \cdots a_{ij}$$

and the elimination form, for objects of a recursively defined data type, is the *case-expression*

$$\begin{array}{l}\mathbf{case}\ x\ \mathbf{of}\ \{\\ \quad c_1\ a_{11} \cdots a_{1m} \rightarrow e_1,\\ \qquad\qquad \vdots\\ \quad c_n\ a_{n1} \cdots a_{nn} \rightarrow e_n\},\end{array}$$

where $e_1, \ldots, e_n$ are expressions of the same type (the type of the case-expression). For example, the set of finite lists may be defined by

$$list(A : \mathbf{Set}) = \mathbf{data}\{Nil, Cons(x : A,\ xs : list\ A)\} : \mathbf{Set}$$

and a list can then be analysed using a *case-expression* as in the following definition of append:

$$\begin{array}{l}append(A : \mathbf{Set},\ l_1 : list\ A,\ l_2 : list\ A) =\\ \quad \mathbf{case}\ l_1\ of\ \{\\ \qquad Nil \rightarrow l_2,\\ \qquad Cons\ x\ xs \rightarrow Cons\ x\ (append\ A\ xs\ l_2)\} : list\ A.\end{array}$$

Note that, using these recursive definitions on functional form, non-linear inductive types cannot be defined, i.e. dependencies between the parameters cannot be introduced. It turned out that pattern matching together with non-linear inductive definitions is a non-conservative extension of Martin-Löf's type theory (see [8]). The approach taken in Half is to allow only linear inductive definitions. As a consequense, the *Id*-type

$$\frac{a \in A}{id(A, a) \in Id(A, a, a)}$$

is not definable: without dependencies between the parameters there is no way of saying that the two elements are the same. Therefore, for abstract sets, instead of working with sets and the *Id*-type, we work in a more general setting using *setoids*, i.e. sets with equivalence relations. For concrete sets, equalities are explicitly defined. This is also closer to the usual mathematical approach where a set comes together with an equality relation.

A Σ-type is a dependent record $\mathbf{sig}\{t_1 : T_1, \ldots, t_n : T_n\}$, where the type T_i may depend on $t_1, \ldots, t_{i-1}$. An object of a Σ-type is formed by constructing objects of the types T_i, $\mathbf{struct}\{t_1 = e_1, \ldots, t_n = e_n\}$, where e_i is an expression of type T_i. The elimination rule for Σ-types is projection; if M is of type $\mathbf{sig}\{t_1 : T_1, \ldots, t_n : T_n\}$, the value of its i'th component is accessed by $M.t_i$.

Adding Σ-types to the system is a conservative extension of the system; it does not affect the strength of the theory, equivalent definitions can always be obtained using recursive definitions with one constructor. However, to analyse objects of a recursively defined set, case-analysis is required, even if there is only one case to consider.

Theories are lists of definitions and proofs:

$$\begin{array}{l} th = \mathbf{theory}\{ \\ \qquad f_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}) = e_1 : T_1, \\ \qquad\qquad \vdots \\ \qquad f_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk}) = e_n : T_n\} \\ \quad : \mathbf{Theory} \end{array}$$

Theories are used to collect definitions and lemmas that logically belong together. Identifiers defined in a theory can be accessed from outside: if th is a theory and f_i an identifier defined in th, then the value of f_i is reached by $th.f_i$.

By defining functions giving theories as result, a notion of parametrised theory is obtained. Identifiers defined in a parametrised theory can then be accessed from outside, provided they are given proper parameters. Also the notion of (parametrised) theory is a conservative extension of the system: functions occuring in a parametrised theory can always be parametrised themselves and defined outside the theory.

The *let-expressions* are used for local lemmas and abbreviations:

$$\mathbf{let}\ x = e_1 : T\ \mathbf{in}\ e_2$$

In the environment ρ, the expression above computes to $e_2(\rho,\ x = e_1\rho)$, i.e. the value of e_2 in the environment ρ extended with $x = e_1\rho$.

Expressions of this language are thus formed by

sorts	**Set**, **Type** and **Theory**
Π-types	$(x : A) \to B$
abstractions	$\lambda x \to e$
applications	$a\ b$
Σ-types	$\mathbf{sig}\{a_1 : A_1, \ldots, a_n : A_n\}$
structures	$\mathbf{struct}\{a_1 = e_1, \ldots, a_n = e_n\}$
projections	$b.a_i$

rec. def. types	$\mathbf{data}\{$ $c_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}),$ $\vdots$ $c_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk})\}$
constructors	c_i
case expressions	$\mathbf{case}\ x\ \mathbf{of}\ \{$ $c_1\ a_{11} \cdots a_{1m} \rightarrow e_1,$ $\vdots$ $c_n\ a_{n1} \cdots a_{nn} \rightarrow e_n\}$
let expressions	$\mathbf{let}\ x = e_1 : T\ \mathbf{in}\ e_2$
theories	$\mathbf{theory}\{$ $f_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}) = e_1 : T_1,$ $\vdots$ $f_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk}) = e_n : T_n\}$
projections	$th.f_i$
variables	x

The system also allow mutual recursive definitions. But this has not been used in the proofs in this paper, we have also avoided mutual recursion between a function f and functions locally defined in f.

There is a "size check" for inductively defined types. The type

$$\begin{array}{l}\mathbf{data}\{\\ \quad c_1(a_{11} : A_{11}, \ldots, a_{1m} : A_{1m}),\\ \qquad\qquad \vdots\\ \quad c_n(a_{n1} : A_{n1}, \ldots, a_{nk} : A_{nk})\}\end{array}$$

lives in **Set** or **Type** if all A_{ij}'s live in **Set** or **Type**, respectively.

The definitional equality is a combination of structural equality and equal by name; for checking equality of "complex" structures, i.e. **data**, **sig**, **struct** and **case**, comparision "by name" is used. This means for instance that in

$$\begin{array}{l}Bool = \mathbf{data}\{False, True\} : \mathbf{Set},\\ Bool' = \mathbf{data}\{False, True\} : \mathbf{Set},\\ Bool'' = Bool : \mathbf{Set}\end{array}$$

Bool and *Bool'* are not equal, but *Bool* and *Bool"* are. This is the approach taken for several strongly typed languages.

The presence of both **Set** and **Type** in Half, where **Set** corresponds to a universe, allows a more abstract reasoning than is possible in a system without a universe. We show this by a small example with subsets of a set represented as propositional functions. First we give a name for the type of predicates over a type A:

$$pred(A : \textbf{Type}) = (x : A) \to \textbf{Set} : \textbf{Type}.$$

The predicates over A are objects in the function space from A to **Set**. This function space does not form a set in predicative type theory (it has the type **Type**). In the same way, given a type A, we form the type for relations on A:

$$rel(A : \textbf{Type}) = (x : A,\, y : A) \to \textbf{Set} : \textbf{Type}.$$

Now we represent subsets of a set A as predicates over A. We say that U is a subset of A if U is a propositional function ranging over A and an element a of A is a member of U iff $U(a)$ holds. A propositional function U is then a subset of another propositional function V provided that Ux implies Vx for all x of type A:

$$subset(A : \textbf{Set}) = \lambda U\ V \to (x : A,\, h : Ux) \to Vx : rel\ (pred\ A).$$

Note that in the type we can see that, given a set A, *subset* A is a relation on predicates of A. Also note that, in the last definition, A must be a set, since by the definition of *rel*, $(x : A,\, h : Ux) \to Vx$ has to be a set. The system checks this for us.

7.2 Subsets and Finite Subsets

A standard way in type theory to represent subsets of a base set X is to use predicates over X; in other words the predicate U represents the subset $\{x \in X : U(x)\ \textit{is true}\}$. Using the Half notation, a subset of the base set X has the type $(x : X) \to \textbf{Set}$. We say that an element a of X is a member of the subset U, $a \in U$, iff $U(a)$ holds, i.e. there exists a proof p of $U(a)$. Moreover, U is a subset of the subset V, $U \subseteq V$, provided that for all elements x, $U(x) \Rightarrow V(x)$. Predicates, however, do not in general respect equality, therefore we also define a weaker subset relation that takes the equality relation on elements as parameter:

$$U \subseteq_= V \equiv (\forall x)(U(x) \Rightarrow (\exists y)(x = y \mathbin{\&} V(y))).$$

In a formal topology the cover relation respects the equality relation; so, this second definition is just as strong as it needs to be.

The representation of subsets as predicates indicates that all subsets and subset forming operations are formed by abstracting a variable x from a proposition $P(x)$ (we usually use the notation $\{x : P(x)\}$ for this subset). For instance the union of U and V, $U \cup V$, is defined by the abstraction $\lambda x.U(x) \vee V(x)$. The properties of the operation, which are the expected ones, then follows immediately from the properties of the corresponding connective.

For the development of the theories used in this paper, a notion of finite subset is also needed. An easy way to handle finite subsets is to use lists. But since lists of a set X and predicates over X have different types, a method of converting lists into predicates is needed when mixing the two notions. To transform a list into a predicate we here simply abstract a variable belonging to the list:

$$\lambda x.x \in l,$$

where $\in$ is the membership relation on lists. We also need a relation $\subseteq_f$ between lists of type X and predicates over X, where the meaning of $l \subseteq_f U$ is that l is a finite subset of

U. Since lists are defined recursively and proofs about lists are by induction, the best way to define $l \subseteq_f U$ is by induction on l:

$$\begin{array}{rl} l \subseteq_f U \equiv & \mathit{case}\ l\ \mathit{of} \\ & \quad \mathit{nil} \Rightarrow \{\mathit{True}\}, \\ & \quad x :: xs \Rightarrow U(x)\ \&\ xs \subseteq_f U. \end{array}$$

Another way to handle finite subsets is to use functions from a finite set to the base X. We start with the following canonical finite sets.

$$\begin{cases} N_0 & \equiv \emptyset \\ N_{k+1} & \equiv N_k + \{\mathit{True}\}. \end{cases}$$

A natural number k together with a function f of type $N_k \longrightarrow X$ represents the finite set

$$\{x \in X : (\exists i \in N_k)(f(i) = x)\}.$$

So to obtain a predicate of X from the pair $\langle k, f\rangle$ we abstract a variable belonging to the image of N_k under f:

$$\lambda x.(\exists i \in N_k)(f(i) = x).$$

The two ways of representing finite subsets are equivalent, but they have different advantages (and disadvantages). Using lists, we have the possibility of using case-analysis over lists. Using functions from finite sets, we have the possibility of quantifying over the "indices" of the finite subsets.

In the implementation the list version of finite subset is used everywhere, except in definition of the cover in definition 4.2, where functions are used to represent finite subsets. In the rule *C3* the subset V is represented by a natural number k and a function f from N_k to $Q \times Q$. The reason is that it makes proofs by induction on the derivation of $w \lhd_{A_f} U$ easier. The list approach would here force a local lemma and mutual recursion; while using the chosen approach, the induction hypothesis can be applied directly inside the assumption $i \in N_k$.

7.3 Formal Spaces

A natural way of collecting general properties of a mathematical structure is to use the notion of parametrised theory. First a type for the structure is defined and the lemmas for this type are then collected inside a theory, parametrised over objects of the type. Given an instance of the mathematical structure, we can then get access to the theory and use the proofs inside it.

A formal space is here defined as a Σ-type: The set S with the equality relation $=$, the binary operation $\cdot$ and the relation $\lhd$ form a formal space; if $\langle S, =, \cdot\rangle$ form a monoid, $\lhd$ respects the equality relation and the conditions of a formal topology (reflexivity, transitivity, dot-left, dot-right) are satisfied. We denote this Σ-type by $\mathit{space}(S, =, \cdot, \lhd)$.

Properties of a general formal space are then collected in a theory parametrised over a set S, a relation $=$, a binary operation $\cdot$, a relation $\lhd$ and a proof that $\langle S, =, \cdot, \lhd\rangle$ form a formal space (i.e. an object of $\mathit{space}(S, =, \cdot, \lhd)$).

7.4 Rational Numbers

The rational numbers Q are here formed abstractly. Following von Plato [17], we start with the order relation $<$ and, for element $p, q, r \in Q$, state the axioms

$$\neg(p < q \ \& \ q < p)$$

$$p < q \Rightarrow p < r \vee r < q.$$

For this order, weak order and equality is defined by

$$p \leq q \equiv \neg(q < p)$$

$$p = q \equiv p \leq q \ \& \ q \leq p.$$

The set of rational numbers form a decidable, unbounded and dense order:

$$p < q \vee \neg(p < q)$$

$$(\forall p)(\exists q)(q < p)$$

$$(\forall p)(\exists q)(p < q)$$

$$p < q \Rightarrow (\exists r)(p < r < q).$$

Then we add the elements 0 and 1; the binary operations $+$ and $*$; unary $-$ and the inverse operation $(\cdot)^{-1}$ for nonzero elements such that $\langle Q, 0, +, -, 1, *, (\cdot)^{-1}, =\rangle$ form a field. Finally we need axioms for relating $<$ to $+$ and $<$ to $*$

$$p < q \Rightarrow p + r < q + r$$

$$p < q \ \& \ 0 < r \Rightarrow p * r < q * r$$

and the Archimedian axiom

$$(\forall p, q > 0)(\exists n \in N)(\bar{n} * q > p),$$

where $n \longmapsto \bar{n}$ is the embedding of N into Q.

So we define a Σ-type, *ABSTR_Q*, consisting of a set Q with the relation $<$, such that Q and $<$ form a dense, decidable, unbounded linear order, with all the elements and operations described above, and such that all the axioms hold. The "true" set of rational numbers should be an element of *ABSTR_Q*. Future definitions and proofs are parametrised over *ABSTR_Q*.

7.5 The Continuum

Given the rational numbers (or rather an element of *ABSTR_Q*), the continuum can now be defined as described in section 3. Once it is proved that $\mathcal{R}$ satisfies the conditions of a formal topology we have an instance of the formal space, i.e. an object of the type $\mathit{space}(Q \times Q, =_{Q \times Q}, \cdot, \lhd_R)$.

7.6 Seminormed Linear Spaces

A linear space over the the rationals is a set of vectors X with a zero element 0, binary addition $+$ and negation $-$, such that $\langle X, 0, +, - \rangle$ constitutes an Abelian group. Moreover, to each pair p and x, where p is a rational number and x is a vector, there corresponds a vector $p * x$, called the product of p and x, such that

$$p * (q * x) = (p * q) * x$$

$$p = q \;\&\; x = y \Rightarrow p * x = q * y$$

$$1 * x = x$$

$$(p + q) * x = p * x + q * x$$

$$p * (x + y) = p * x + p * y.$$

A seminormed linear space is then a linear space equipped with a seminorm, a predicate N on $Q \times X$ (for readability the notation $x \in N(q)$ is used for $N(q, x)$).

Linear spaces and seminormed linear spaces are treated in the same way as the rational numbers. They are defined as certain Σ-types describing the elements, operations and axioms of a linear space and a seminormed linear space, respectively.

A subspace of the linear space X is a linear space A which is a subset of X. A linear subspace A of the linear space X is here a propositional function A over X (we use the notation $x \in A$ for $A(x)$ *true*) satisfying

$$0 \in A$$

$$x \in A \;\&\; x = y \Rightarrow y \in A$$

$$x \in A \Rightarrow p * x \in A$$

$$x \in A \;\&\; y \in A \Rightarrow x + y \in A.$$

To define the subspaces we state, using a Σ-set, under what circumstances a predicate A of X is a proper linear subspace. We denote this Σ-set by *seminolinsubsp*(A). To express that a linear space M is a subspace of another linear space the usual subset relation between subsets is used, since M and A are predicates.

Linear spaces can then be generated in the following ways. The linear space only containing the 0-*vector*:

$$[\,] \equiv \{y : y = 0\}.$$

Given a linear space M, the extension of M with x:

$$[M + x] \equiv \{y : (\exists z \in M, t \in Q)(y = z + tx)\}.$$

Given a linear space M, the extension of M with a finite set M_0 (represented by a list):

$$\begin{aligned}[M + M_0] \equiv\ & \mathit{case}\ M_0\ \mathit{of} \\ & \quad \mathit{nil} \Rightarrow M, \\ & \quad x :: M_1 \Rightarrow [[M + M_1] + x].\end{aligned}$$

The linear space spanned by a finite set M_0:

$[M_0] \equiv [[\ ] + M_0]$.

For each one of these generated subspaces it must be proved that they really are linear spaces, i.e. objects of *seminolinsubsp*([ ]), *seminolinsubsp*($[M + x]$), etc., have to be found, but this follows easily from the way the predicates are generated.

7.7 Formal Linear Functionals

Let X be a seminormed linear space. The subbasic elements of the formal topology of linear functionals of norm ≤ 1, consist of pairs of the form $\langle x, I \rangle$, where x is an element of X and I a rational interval. The base S consists of finite lists of these subbasic elements. The dot-operation is then *append*. Using the subset relation on lists, the proofs that the base form a monoid all follow from general properties of lists and so does idempotence.

7.8 Structure of the Implementation

The entire implementation consists of definitions and proofs divided between several theories and files[4]. Many of them are not specific to the development of the proof of the Hahn-Banach theorem. We here briefly describe what the files contain.

core.half : General data types and definitions about relations.

dec.half [5]: Decidable propositions and relations.

monoid.half : Definition of congruent and commutative monoid.

interval.half [5]: Definition of interval defined as a pair of its endpoints. A theory containing some properties of relations on interval.

linear.half : Definition of a general linear ordering.

declinear.half [5]: Decidability is here added to the linear ordering.

unbounddensedec.half : The ordering above is here also unbounded and dense.

fin.half : Definition of finite lists and a theory containing general properties of lists.

subsets.half : Definitions of the subset relation and union for subsets (predicates). A theory containing general properties of subsets and finite subsets represented by lists. (see section 7.2)

space.half [5]: Definition of formal space (definition 2.1). A theory containing some general properties of formal spaces, such as stability, localisation and the fact that a cover respects the subset relation (see section 2), definition of formal Stone cover (definition 2.2) and compact formal space (definition 2.3). A lemma saying that stability is derivable without dot-right if localisation holds (see section 2.1).

[4]The files are obtainable from the URL:
`ftp://ftp.cs.chalmers.se/pup/users/ceder/hahnbanach/hhb.tar`.

[5]Parts of the definitions and proofs in this file are due to Thierry Coquand.

continuum.half : Definition of the continuum R as a formal space (definition 3.1). The definition of the cover $\lhd_R$ goes via a finitely inductively defined cover $\lhd_{R_f}$. Proofs that R really is a formal space (the lemmas 3.2–3.4 and proposition 3.5).

finset.half : Finite subsets of a set A are here defined as functions from canonical finite sets to A (see section 7.2).

algstruct.half : Definitions of some algebraic structures used to define the abstract rational numbers in section 7.4.

fin2.half : A theory containing some further properties of lists. Provided that the equality relation on elements are reflexive and transitive, we prove that append is congruent, commutative, associative and idempotent with respect to the subset equality defined on lists.

RatnumAndLinalg.half : Definitions of the abstract rational numbers *ABSTR_Q* (see section 7.4) and linear spaces (see section 7.6).

The rational numbers are defined as a Σ-type consisting of a set Q with the relation $<$, such that Q and $<$ form a dense, decidable, unbounded linear order, and with all the elements, operations and properties described in section 7.4. Included in this Σ-type are also several very elementary properties of the rational numbers which are assumed to hold. These properties do follow from the axiomatisation. Intuitively they are trivial and there should be no difficulties in proving them formally either. The reason why we do not prove these properties instead of assuming them is that it would be tedious and the purpose of this work has not been to prove elementary lemmas about rational numbers. Observe that these assumptions are not unproved lemmas: elements of `ABSTR_Q` are records and the assumptions are part of the axiomatisation.

The linear spaces are then treated in the same way.

Here are also some additional properties of the cover $\lhd_{R_f}$ proved: the lemmas 3.6–3.8.

hhb.half : The implementation of the Hahn-Banach theorem is structured in theories in the following way. The proof consists of a single theory, `theory_semi_no_linsp`, parametrised over seminormed linear spaces (definition 4.1). Apart from proofs of some basic properties of the norm; definition of seminormed linear subspace (*seminolinsubsp* section 7.6); definition of the base S (definition 4.2); definition and properties of the relations $=$, $<$, $\leq$, $=_{\subseteq}$ on S; `theory_semi_no_linsp` also contains two nested theories: `theory_LINSUBSP` and `theoryHHB`.

`theory_LINSUBSP` is parametrised over a seminormed linear subspace A, it contains definitions and proofs specific to one seminormed linear subspace. Here are for instance the cover relations $\lhd_{A_f}$ and $\lhd_A$ defined (definition 4.2); and proofs that $\langle S, =_{\subseteq}, \cdot, \lhd_{A_f}\rangle$ and $\langle S, =_{\subseteq}, \cdot, \lhd_A\rangle$ are formal topologies (proposition 4.8). Here are also proofs that $\lhd_{A_f}$ is a Stone cover (proposition 5.1) and Alaoglu's theorem (corollary 5.2).

`theoryHHB` is parametrised over two seminormed linear subspaces, M and A, and a proof that M is a subspace of A. Here we find a proof of the conservativity property (proposition 6.2); definitions of how to extend a a seminormed linear subspace (see section 7.6); the lemmas 6.3–6.7, and finally the Hahn-Banach theorem 6.1. In this theory there is also a nested theory, `theory_key`, containing definitions and lemmas for the proof of lemma 6.4.

`theory_key` is parametrised over a seminormed linear subspace M and an element x, here we find definitions and proofs related to the transformation of neighbourhoods from $\mathcal{L}([M+x])$ to $\mathcal{L}(M)$. `theory_key` contains for instance definitions of P (definition 6.11); the key lemma 6.14 (and 6.19) and the lemmas 6.20–6.21, 6.23–6.26, 6.12. The theory `theory_key` contains yet another nested theory `theory_cof` with definitions and proofs specific to the cover in definition 6.10. `theory_cof` is parametrised over a list of elements in $[M+x]$ of norm ≤ 1. Apart from the definition of the cover in definition 6.10, `theory_cof` also contains a definition of the transformation of neighbourhoods and subsets from $\mathcal{L}([M+x])$ to $\mathcal{L}(M)$ (definition 6.8); the lemma that performs the transformation (lemma 6.17); and the lemmas 6.13, 6.18.

7.9 Cross Reference List

def/lem	formal name	located in theory	file
def 2.1	`space`		`space.half`
def 2.2	`stone`	`theory_space`	"
def 2.3	`compact`	"	"
sec 2			
stability	`stab`	"	"
localisation	`loc`	"	"
$\lhd$ *respects* $\subseteq$	`lem14`, `lem15`	"	"
stability follows from localisation	`STAB`		"
def 3.1			
$Q \times Q$	`QxQ`	`theory_continuum`	`continuum.half`
$=_{Q\times Q}$	`eqQxQ`	"	"
$\cdot$	`dot`	"	"
$\lhd_{R_f}$	`covf`	"	"
$\lhd_R$	`cov`	"	"
lem 3.2	`axiom0f`, `reflf`, `transf`, `dotlf`, `dotrf`	"	"
lem 3.3	`translem`	"	"
lem 3.4	`axiom0`, `refl`, `trans`, `dotl`, `dotr`	"	"
prop 3.5	`Rf`, `R`	"	"
lem 3.6	`ltQxQcovRflem`	`theory_ABSTR_Q`	`RatnumAndLinalg.half`
lem 3.7	`addQxQcovRflem`	"	"
lem 3.8	`fincovRf`	"	"
lem 3.9	`covftocov`	`theory_continuum`	`continuum.half`
def 4.1	`semi_no_linsp`		`hhb.half`
def 4.2			
S	`S`	`theory_semi_no_linsp`	"
$=$	`eqS`	"	"
$<$	`ltS`	"	"
$\leq$	`leqS`	"	"
$\lhd_{A_f}$	`covf`	"	"
$\lhd_A$	`cov`	"	"
def 4.3			
$\cdot$	`dot`	"	"
$=_{\subseteq}$	`eqS2`	"	"
def 4.4	`lives`, `subs_lives`	`theory_semi_no_linsp`, `theory_LINSUBSP`	"
lem 4.5	`axiom0f`, `reflf`, `transf`, `dotlf`, `dotrf`	"	"
lem 4.6	`translem`	"	"
lem 4.7	`axiom0`, `refl`, `trans`, `dotl`, `dotr`	"	"
prop 4.8	`isspacef`, `isspace`	"	"
lem 4.9	`covftocov`	"	"
lem 4.10	`covtocovf`	"	"
lem 4.11	`convcovf2`	`theory_semi_no_linsp`, `theoryHHB`	"

def/lem	formal name	located in theory	file
prop 5.1	isStone	theory_semi_no_linsp, theory_LINSUBSP	hhb.half
thrm 5.2	Alaoglu	"	"
thrm 6.1	HHB	theory_semi_no_linsp, theoryHHB	"
prop 6.2	conserveprop	"	"
lem 6.3	HHBf	"	"
lem 6.4	HHBf2	"	"
lem 6.5	HHBf0	"	"
cor 6.6	HHBf1	"	"
lem 6.7	HHBf3	"	"
def 6.8	bar, barsubs	theory_semi_no_linsp, theoryHHB, theory_key, theory_cof	"
lem 6.9	barlem4	"	"
def 6.10	cof	"	"
def 6.11	P	theory_semi_no_linsp, theoryHHB, theory_key	"
lem 6.12	covftocof	"	"
lem 6.13	coftocovf2	theory_semi_no_linsp, theoryHHB, theory_key, theory_cof	"
lem 6.14	hhbkey2	theory_semi_no_linsp, theoryHHB, theory_key	"
lem 6.15	liveslem6	theory_semi_no_linsp, theoryHHB, theory_key, theory_cof	"
lem 6.16	leqSlem9	theory_semi_no_linsp	"
lem 6.17	coftocovfbar	theory_semi_no_linsp, theoryHHB, theory_key, theory_cof	"
cor 6.18	coftocovf1	"	"
lem 6.19	hhbkey1	theory_semi_no_linsp, theoryHHB, theory_key	"
lem 6.20	hhbkey1lem1	"	"
lem 6.21	hhbkey1lem2	"	"
lem 6.22	splitsum1, splitsum2	theory_semi_no_linsp, theory_LINSUBSP	"
lem 6.23	Nlem3	theory_semi_no_linsp, theoryHHB, theory_key	"
lem 6.24	covlem1	"	"
lem 6.25	covlem2lem2	"	"
lem 6.26	covlem2	"	"
lem 6.27	covlem3	"	"
lem 6.28	maxsminrlem	"	"
lem 6.29	covflem1	"	"
lem 6.30	covflem2	"	"

def/lem	formal name	located in theory	file
sec 7.2			
$\subseteq$	`Subset`		`subsets.half`
$\subseteq_=$	`Subset2`		”
$\cup$	`Union`		”
$\subseteq_f$	`finsubset`	`theory_subsets`	”
N	`NN`		`finset.half`
sec 7.4			
order	`islinear`		`declinear.half`
$\leq$	`leq`	`theoryLinear`	`linear.half`
$=$	`eq`	”	”
dec order	`isdeclinear`		`unbounddencedec.half`
dec, unbounded, dense order	`isdenseunbdeclinear`		”
ABSTR_Q	`ABSTR_Q`		`RatnumAndLinalg.half`
sec 7.6			
linear spaces	`linsp`		”
semi no lin sp	`semi_no_linsp`		`hhb.half`
$[\,]$	`ZERO`	`theory_semi_no_linsp`, `theoryHHB`	”
$[M+x]$	`finext1`	”	”
$[M+M_0]$	`finext`	”	”
$[M_0]$	`mk_lin`	”	”

8 Acknowledgement

I wish to thank Thierry Coquand for useful suggestions to the formalisation and Dan Synek who implemented the type checker and the `emacs`-interface to Half. I also wish to thank Sara Negri and Jan Smith for helpful suggestions to this paper.

References

[1] H. Barendregt. *Lamda calculi with types*, In S. Abramsky, D.M. Gabbay and T.S.E. Maibaum eds., “Handbook of Logic in Computer Science, Vol. 2”, Oxford University Press, Oxford, 1992.

[2] E. Bishop. “Foundations of Constructive Analysis”, McGraw Hill, New York, 1967.

[3] N.G. de Bruijn. *A plea for weaker frameworks*, In G. Huet and G. Plotkin eds., “Logical Frameworks”, pp. 40-68, Cambridge University Press, Cambridge, 1991.

[4] J. Cederquist, T. Coquand, S. Negri. *The Hahn-Banach Theorem in Type Theory*, To be published in the proceedings of Twentyfive years of Constructive Type Theory, G. Sambin and J. Smith eds., Oxford University Press, 1997.

[5] J. Cederquist, S. Negri. *A constructive proof of the Heine-Borel covering theorem for formal reals*, In S. Berardi and M. Coppo eds., Logic in Computer Science 1158, pp. 62-75, 1996.

[6] T. Coquand. *An algorithm for type-checking dependent types*, Science of Computer Programming 26, pp. 167-177, Elsevier, 1996.

[7] M.P. Fourman, R.J. Grayson. *Formal Spaces*, In A.S. Troelstra and D. van Dalen eds, "The L. E. J. Brouwer Centenary Symposium", pp. 107-122, North Holland, Amsterdam, 1982.

[8] M. Hofmann. *A model of intensional Martin-Löf type theory in which unicity of identity proofs does not hold*, Technical report, Dept. of Computer Science, University of Edinburgh, 1993.

[9] P.T. Johnstone. "Stone Spaces", Cambridge University Press, 1982.

[10] S. MacLane, L. Moerdijk. "Sheaves in Geometry and Logic : A First Introduction to Topos Theory", Springer-Verlag, New York, 1992.

[11] L. Magnusson. "The Implementation of ALF - a Proof Editor based on Martin-Löf's Monomorphic Type Theory with Explicit Substitution", Chalmers University of Technology and University of Göteborg, PhD Thesis, 1995.

[12] P. Martin-Löf. *An Intuitionistic Theory of Types* (1972), To be published in the proceedings of Twentyfive years of Constructive Type Theory, G. Sambin and J. Smith eds., Oxford University Press, 1997.

[13] C.J. Mulvey, J.W. Pelletier. *A globalization of the Hahn-Banach theorem*, Advances in Mathematics 89, pp. 1-60, 1991.

[14] S. Negri, D. Soravia. *The continuum as a formal space*, submitted for publication, 1996.

[15] B. Nordström, K. Petersson, J. Smith. "Programming in Martin-Löf's Type Theory", Oxford University Press, 1990.

[16] S. Owre, N. Shankar, J. M. Rushby. *The PVS Specification Language (Beta Release)*, Computer Science Laboratory, SRI International, Menlo Park, CA 94025, USA, 1993.

[17] J. von Plato. *A memorandum on the constructive axioms of linear order*, Dept. of Philosophy, University of Helsinki, 1995.

[18] W. Rudin. "Functional analysis", 2nd ed., McGraw-Hill, 1991.

[19] G. Sambin. *Intuitionistic formal spaces – a first communication*, In D. Skordev ed., "Mathematical logic and its applications", pp. 187-204, Plenum Press, 1987.

[20] J.J.C. Vermeulen. "Constructive Techniques in Functional Analysis", PhD Thesis, University of Sussex, 1986.