

A Labeled Tableaux System for the Distributed Temporal Logic DTL

David Basin

Department of Computer Science, ETH Zurich, Switzerland
basin@inf.ethz.ch

Carlos Caleiro Jaime Ramos

SQIG-Instituto de Telecomunicações and Department of Mathematics, IST, TU Lisbon, Portugal
{ccal | jabr}@math.ist.utl.pt

Luca Viganò

Department of Computer Science, University of Verona, Italy
luca.vigano@univr.it

Abstract

DTL is a distributed temporal logic for reasoning about temporal properties of distributed systems from the local point of view of the system's agents, which are assumed to execute sequentially and to interact by means of synchronous event sharing. We present a sound and complete labeled tableaux system for future-time DTL. To achieve this, we first formalize a labeled tableaux system for reasoning locally at each agent, which provides a system for full future-time LTL, and afterwards we combine the local systems into a global one by adding rules that capture the distributed nature of DTL.

1 Introduction

The distributed temporal logic DTL [8] is a logic for reasoning about temporal properties of distributed systems from the local point of view of the system's agents, which are assumed to execute sequentially and to interact by means of synchronous event sharing. Distribution is implicit and properties of entire systems are formulated in terms of the local properties of the agents and their interaction. DTL is closely related to the family of temporal logics whose semantics are based on the models of true concurrency introduced and developed in [14, 15, 19]. In particular, the semantics of these logics are based on a conflict-free version of Winskel's event structures [26], enriched with information about sequential agents. Different versions have been given, reflecting different perspectives on how non-local information can be accessed by each agent.

DTL was first proposed in [8] as a logic for specifying

and reasoning about distributed information systems. The logic has also been used in the context of security protocol analysis for reasoning about the interplay between protocol models and security properties [3, 4]. However, all of the previous results have been obtained directly by semantic arguments. It would be reassuring, and useful in general, to have a usable deductive system for DTL. An attractive possibility in this regard is a labeled tableaux system as deductions will then closely follow semantic arguments.

In this paper, we present a sound and complete labeled tableaux system for future-time DTL. Our main contributions are two-fold. First, we introduce a labeled tableaux system for full future-time LTL, where reasoning is local. Due to lack of space, we do not consider here past-time operators like since, although all the results presented in this paper extend straightforwardly to full LTL (and thus to full DTL) as we show in [1].

Second, we combine the LTL systems local to each agent with rules that capture the distributed nature of DTL, via communication. The tableaux systems for local reasoning (in LTL) are, as expected, built from formulas labeled with local state information and relations between these labels (order and equality). We integrate these systems into a system for global reasoning, where we introduce an additional relation expressing synchronization. We prove the soundness and completeness of the system with respect to DTL entailment and provide examples of its use. We have not, for now, addressed the question of efficient proof search and have included an infinite closure rule that captures eventualities which are always delayed. Note, in this regard, that DTL entailment is decidable as it can be reduced to entailment in LTL by a suitable polynomial-time syntactic translation and linearization of the distributed models.

However, the distributed properties expressible in DTL are *trace-consistent*, or invariant under different linearizations, and this suggests developing a simpler, dedicated deductive system for DTL along the lines that we follow here.

We proceed as follows. In §2, we introduce DTL. In §3, we present our tableaux system for local reasoning and establish its soundness and completeness with respect to entailment. In §4, we extend our local system into a system for global reasoning by including a new synchronization relation between local labels and we also prove soundness and completeness with respect to entailment. We conclude, in §5, by comparing with related work and discussing future work. Due to lack of space, examples have been shortened and proofs have been omitted; full details are given in [1]. Applications of the logic are also presented in [1], as well as in [3, 4, 8, 9].

2 The distributed temporal logic DTL

2.1 The syntax and semantics of DTL

DTL is defined over a *distributed signature* $\Sigma = \langle Id, \{Prop_i\}_{i \in Id} \rangle$ of a system, where Id is a finite set of *agents* and, for each $i \in Id$, $Prop_i$ is a set of *local state propositions*. The *global language* \mathcal{L}_{DTL} is defined by the grammar

$$\mathcal{L}_{DTL} ::= @_{i_1}[\mathcal{L}_{i_1}] \mid \cdots \mid @_{i_n}[\mathcal{L}_{i_n}],$$

for $Id = \{i_1, \dots, i_n\}$. The *local languages* \mathcal{L}_i , for each $i \in Id$, are in turn defined by

$$\begin{aligned} \mathcal{L}_i ::= & Prop_i \mid \neg \mathcal{L}_i \mid \mathcal{L}_i \Rightarrow \mathcal{L}_i \mid \mathbf{F} \mathcal{L}_i \mid \mathbf{G} \mathcal{L}_i \mid \mathbf{X} \mathcal{L}_i \mid \\ & \mathcal{L}_i \mathbf{W} \mathcal{L}_i \mid @_j[\mathcal{L}_j], \end{aligned}$$

with $j \in Id$. A global formula $@_i[\varphi]$ means that φ holds for agent i . Local formulas, as the name suggests, hold locally for the different agents. For instance, locally for an agent i , the operators \mathbf{F} , \mathbf{G} , \mathbf{X} , and \mathbf{W} are the usual *sometime in the future*, *always in the future*, *next*, and *weak until (unless)* temporal operators, respectively, while the *communication formula* $@_j[\psi]$ means that agent i has just communicated (synchronized) with agent j , for whom ψ held¹. We will use \mathcal{L}_i° to denote the set of all purely temporal formulas of \mathcal{L}_i , that is, excluding communication formulas.

Other logical connectives (conjunction, disjunction, etc.) and temporal operators can be defined as abbreviations, e.g.:

$\varphi \mathbf{U} \psi$	$\equiv (\mathbf{F} \psi) \wedge (\varphi \mathbf{W} \psi)$	strong until;
$\mathbf{F}_\circ \varphi$	$\equiv \varphi \vee \mathbf{F} \varphi$	now or sometime in the future;
$\mathbf{G}_\circ \varphi$	$\equiv \varphi \wedge \mathbf{G} \varphi$	now and always in the future;
$\varphi \gg_j \psi$	$\equiv \varphi \Rightarrow @_j[\psi]$	calling.

¹Note that the DTL syntax here differs from, and improves upon, the original presentation in [8]. Previously, the operator $@_i$ was overloaded with $@_i$ and its interpretation was therefore context dependent.

Here we use the subscript \circ to denote the reflexive versions of the operators. Note also that *calling* is specific to DTL as it involves communication: $@_i[\varphi \gg_j \psi]$ means that if φ holds for agent i then he calls (synchronizes with) agent j , for whom ψ must hold.

A *local life-cycle* of an agent i is a countable (finite or infinite), discrete, well-founded, total order $\lambda_i = \langle E_i, \leq_i \rangle$, where E_i is the set of *local events* and \leq_i the *local order of causality*. We define the corresponding *local successor relation* $\rightarrow_i \subseteq E_i \times E_i$ to be the relation such that $e \rightarrow_i e'$ if $e <_i e'$ and there is no e'' such that $e <_i e'' <_i e'$. As a consequence, we have that $\leq_i = \rightarrow_i^*$, i.e., \leq_i is the reflexive and transitive closure of \rightarrow_i .

A *distributed life-cycle* is a family $\lambda = \{\lambda_i\}_{i \in Id}$ of local life-cycles such that $\leq = (\bigcup_{i \in Id} \leq_i)^*$ defines a partial order of *global causality* on the set of all events $E = \bigcup_{i \in Id} E_i$. Note that communication is modeled by event sharing and thus for some event e we may have $e \in E_i \cap E_j$ for $i \neq j$. In that case, requiring \leq to be a partial order amounts to requiring that the local orders are globally compatible, thus excluding the existence of another $e' \in E_i \cap E_j$ where both $e <_i e'$ and $e' <_j e$.

A *local state* of agent i is a finite set $\xi \subseteq E_i$ that is downward-closed for local causality, i.e., if $e \leq_i e'$ and $e' \in \xi$ then also $e \in \xi$. The set Ξ_i of all local states of an agent i is totally ordered by inclusion and has \emptyset as the minimal element. In general, each non-empty local state ξ is reached from the local state $\xi \setminus \{last_i(\xi)\}$ by the occurrence of an event that we call $last_i(\xi)$. The local states of each agent are totally ordered as a consequence of the total order on local events. Since they are discrete and well-founded, we enumerate them as follows: \emptyset is the 0th state; $\{e\}$, where e is the minimum of $\langle E_i, \leq_i \rangle$, is the first state; and, in general, if ξ is the k th state of agent i and $last_i(\xi) \rightarrow_i e'$, then $\xi \cup \{e'\}$ is the $(k+1)$ th state of agent i . We denote by ξ_i^k the k th state of agent i . $\xi_i^0 = \emptyset$ is the initial state and ξ_i^k is the state reached from the initial state after the occurrence of the first k events. In fact, ξ_i^k is the only state of agent i that contains k elements, i.e., where $|\xi_i^k| = k$. Given $e \in E_i$, observe that $(e \downarrow i) = \{e' \in E_i \mid e' \leq_i e\}$ is always a local state. Furthermore, if ξ is non-empty, then $(last_i(\xi) \downarrow i) = \xi$.

An *interpretation structure* $\mu = \langle \lambda, \sigma \rangle$ consists of a distributed life-cycle λ and a family $\sigma = \{\sigma_i\}_{i \in Id}$ of labeling functions. For each $i \in Id$, $\sigma_i : \Xi_i \rightarrow \wp(Prop_i)$ associates a set of local state propositions to each local state. We denote $\langle \lambda_i, \sigma_i \rangle$ by μ_i and define the *global satisfaction relation* by

- $\mu \Vdash_{DTL} @_i[\varphi]$ iff $\mu_i \Vdash_i \varphi$ iff $\mu_i, \xi \Vdash_i \varphi$ for every $\xi \in \Xi_i$,

where the local satisfaction relations at local states are defined by

- $\mu_i, \xi \Vdash_i p$ if $p \in \sigma_i(\xi)$;
- $\mu_i, \xi \Vdash_i \neg \varphi$ if $\mu_i, \xi \not\Vdash_i \varphi$;
- $\mu_i, \xi \Vdash_i \varphi \Rightarrow \psi$ if $\mu_i, \xi \not\Vdash_i \varphi$ or $\mu_i, \xi \Vdash_i \psi$;

- $\mu_i, \xi \Vdash_i \mathbf{F} \varphi$ if $|\xi| = k$ and there exists $\xi_i^n \in \Xi_i$ such that $k < n$ with $\mu_i, \xi_i^n \Vdash_i \varphi$;
- $\mu_i, \xi \Vdash_i \mathbf{G} \varphi$ if $|\xi| = k$ and $\mu_i, \xi_i^n \Vdash_i \varphi$ for every $\xi_i^n \in \Xi_i$ such that $k < n$;
- $\mu_i, \xi \Vdash_i \mathbf{X} \varphi$ if $|\xi| = k$, ξ_i^{k+1} exists and $\mu_i, \xi_i^{k+1} \Vdash_i \varphi$;
- $\mu_i, \xi \Vdash_i \varphi \mathbf{W} \psi$ if $|\xi| = k$ and $\mu_i, \xi_i^n \Vdash_i \varphi$ for every $\xi_i^n \in \Xi_i$ with $k < n$; or there exists $\xi_i^n \in \Xi_i$ such that $k < n$ with $\mu_i, \xi_i^n \Vdash_i \psi$, and $\mu_i, \xi_i^m \Vdash_i \varphi$ for every $k < m < n$;
- $\mu_i, \xi \Vdash_i \odot_j[\varphi]$ if $|\xi| > 0$, $\text{last}_i(\xi) \in E_j$, and $\mu_j, (\text{last}_i(\xi) \downarrow j) \Vdash_j \varphi$.

The satisfaction conditions for the connectives and the temporal operators defined as abbreviations can be obtained from the primitive ones in a straightforward way. For example, for strong until we have that

- $\mu_i, \xi \Vdash_i \varphi \mathbf{U} \psi$ if $|\xi| = k$ and there exists $\xi_i^n \in \Xi_i$ such that $k < n$ with $\mu_i, \xi_i^n \Vdash_i \psi$, and $\mu_i, \xi_i^m \Vdash_i \varphi$ for every $k < m < n$.

We say that μ is a *model* of $\Gamma \subseteq \mathcal{L}_{\text{DTL}}$ if μ globally satisfies each of the formulas in Γ , and given $\delta \in \mathcal{L}_{\text{DTL}}$ we say that Γ *entails* δ , written $\Gamma \models_{\text{DTL}} \delta$, if every global model of Γ is also a model of δ . Given $\Phi \cup \{\psi\} \subseteq \mathcal{L}_i$, we write $\Phi \models_i \psi$ to denote the fact that every local model of Φ is also a model of ψ or, equivalently, that $\{\@_i[\varphi] \mid \varphi \in \Phi\} \models_{\text{DTL}} \@_i[\psi]$.

For instance, $\@_i[p \Rightarrow \mathbf{F} \odot_j[Xq]]$ holds in a model if whenever proposition p holds locally at a state of agent i then there must be a future state of agent i where he has just synchronized with agent j , for whom q will hold in the next state. Additional examples can be found in [1].

Note that, as is well known, the expressive power of the \mathbf{U} operator is exactly the same of the set $\{\mathbf{F}, \mathbf{G}, \mathbf{X}, \mathbf{W}\}$. This can be seen from the following equivalences.

$\mathbf{F} \varphi$	$\equiv \top \mathbf{U} \varphi$	sometime in the future;
$\mathbf{G} \varphi$	$\equiv \neg \mathbf{F} \neg \varphi$	always in the future;
$\mathbf{X} \varphi$	$\equiv \perp \mathbf{U} \varphi$	tomorrow (next);
$\varphi \mathbf{W} \psi$	$\equiv (\mathbf{G} \varphi) \vee (\varphi \mathbf{U} \psi)$	weak until (unless).

2.2 Decidability of DTL

It is not difficult to show, as suggested in [8], that DTL is decidable by means of a translation to LTL. We omit this translation for the sake of space and refer the reader to [1]. Since LTL is decidable, any decision procedure for LTL entailment can thus also be used for DTL (and since our translation is polynomial, the asymptotic complexity is identical). The translation relies on the fact that given a distributed life-cycle $\lambda = \{\langle E_i, \leq_i \rangle\}_{i \in Id}$, it is always possible to *linearize* the global order on events $\langle E, \leq \rangle$: one can define an injective function $f : E \rightarrow \mathbb{N}$ that preserves the global causality relation, i.e., if $e < e'$ then $f(e) < f(e')$. However, the result given in [1] is actually independent of the chosen linearization function f and in general there may be many such functions. This means that DTL is *trace-consistent* in the sense of [24]; namely, DTL properties can be checked by considering one arbitrary linearization of the

distributed model, as opposed to checking all possible linearizations. This makes DTL properties particularly well-suited for efficient model checking using partial-order reduction techniques [17], which was explored in [9]. Furthermore, this fact suggests that proof techniques tailored directly to DTL might have advantages over off-the-shelf techniques operating on the translations in LTL. This is yet another motivation for developing a labeled tableaux system for DTL.

3 Tableaux for local reasoning

We start by developing a labeled tableaux system for reasoning locally at each agent. This essentially amounts to a tableaux system for full future-time LTL (see [1] for the extension with past-time operators).

From now on, we consider fixed a distributed signature Σ . Our tableaux for local reasoning handles four kinds of *local judgments* for each agent $i \in Id$: labeled local formulas (excluding communication), equality between labels, inequality between labels, and a special judgment indicating absurdity. Local labels denote the local states of agent i . To define the language of labels, for the given signature Σ , we assume fixed a family $\mathcal{V} = \{\mathcal{V}_i\}_{i \in Id}$ of sets of *label variables* and also use a family $\mathcal{F} = \{\mathcal{F}_i\}_{i \in Id}$ of sets of *Skolem function symbols* defined as follows:

$$\mathcal{F}_i = \{\mathbf{f}_{\varphi \mathbf{W} \psi} \mid \varphi, \psi \in \mathcal{L}_i^{\odot}\} \cup \{\mathbf{f}_{\neg(\varphi \mathbf{W} \psi)} \mid \varphi, \psi \in \mathcal{L}_i^{\odot}\}.$$

Note that while \mathbb{N} denotes the natural numbers with 0, we write \mathbb{N}^+ to denote the positive natural numbers, $\mathbb{N} \setminus \{0\}$. The syntax of the *local labels* of agent i is defined by

$$\begin{aligned} \mathcal{T}_i &::= \mathbb{N} \mid \mathcal{V}_i + \mathbb{Z} \mid \mathcal{F}_i(\mathcal{T}_i) + \mathbb{Z}, \\ \mathcal{S}_i &::= (i, \mathcal{T}_i). \end{aligned}$$

Labels involving the Skolem function symbols will be used in the tableaux to guarantee the existence of certain local states associated with the satisfaction of formulas involving the weak until. Although the use of fresh variables suffices in some cases, weak until, as well as its negation, may require the existence of states in the model with specific properties. This requirement makes the use of the Skolem functions an essential ingredient of our system. We write v to denote an arbitrary label variable, x, y , and z to denote arbitrary label terms, and s_i to denote an arbitrary element of \mathcal{S}_i . We abbreviate $x + 0$ as x and if $c \in \mathbb{N}^+$ then we write $x - c$ instead of $x + (-c)$, as usual.

The syntax of *local judgments* of agent i can now be defined by

$$\mathcal{J}_i ::= \mathcal{S}_i : \mathcal{L}_i^{\odot} \mid \mathcal{S}_i = \mathcal{S}_i \mid \mathcal{S}_i < \mathcal{S}_i \mid \text{CLOSED}.$$

When convenient, we write $s_i < s'_i < s''_i$ instead of $s_i < s'_i$ and $s'_i < s''_i$. The intended meaning of a labeled formula

$s_i : \varphi$ is that φ holds at the local state (denoted by) s_i of agent i . Equalities and inequalities of local labels of agent i are interpreted directly over the causality ordering. To make this formal, we extend our notion of interpretation structure with information concerning labels: we interpret labels as natural numbers in such a way that the interpretation of a given local label identifies, by its value, the local state of the corresponding agent. An *assignment on label variables* is a function $\rho_i : \mathcal{V}_i \rightarrow \mathbb{N}$. We also need to consider a fixed interpretation structure μ . The *denotation of labels* over μ and $\rho = \{\rho_i\}_{i \in Id}$, for each agent $i \in Id$, in symbols $\llbracket \cdot \rrbracket_{\mu, \rho} : \mathcal{S}_i \rightarrow \mathbb{N}$, is defined as the following partial function:

- $\llbracket (i, k) \rrbracket_{\mu, \rho} = k$;
- $\llbracket (i, v) \rrbracket_{\mu, \rho} = \rho_i(v)$;
- $\llbracket (i, f_{\varphi W\psi}(x)) \rrbracket_{\mu, \rho} = n$ provided that
 - $\llbracket (i, x) \rrbracket_{\mu, \rho}$ is defined;
 - $n > \llbracket (i, x) \rrbracket_{\mu, \rho}$ is the least number, if it exists, such that
 - $\xi_i^n \in \Xi_i$ and $\mu_i, \xi_i^n \Vdash_i \varphi$ and $\mu_i, \xi_i^n \not\Vdash_i \psi$;
 - $\mu_i, \xi_i^k \Vdash_i \varphi$, for every k such that $\llbracket (i, x) \rrbracket_{\mu, \rho} < k < n$;
- $\llbracket (i, f_{\neg(\varphi W\psi)}(x)) \rrbracket_{\mu, \rho} = n$, provided that
 - $\llbracket (i, x) \rrbracket_{\mu, \rho}$ is defined;
 - $n > \llbracket (i, x) \rrbracket_{\mu, \rho}$ is the least number, if it exists, such that
 - $\xi_i^n \in \Xi_i$, $\mu_i, \xi_i^n \not\Vdash_i \varphi$ and $\mu_i, \xi_i^n \Vdash_i \psi$;
 - $\mu_i, \xi_i^k \not\Vdash_i \psi$, for every k such that $\llbracket (i, x) \rrbracket_{\mu, \rho} < k < n$;
- $\llbracket (i, x + k) \rrbracket_{\mu, \rho} = \llbracket (i, x) \rrbracket_{\mu, \rho} + k$, provided that $\llbracket (i, x) \rrbracket_{\mu, \rho}$ is defined and $\llbracket (i, x) \rrbracket_{\mu, \rho} + k \geq 0$.

For simplicity, when $\llbracket (i, x) \rrbracket_{\mu, \rho}$ depends only on ρ , we may write $\rho_i(x)$.

One reason why the denotation of labels is partial is that we do not consider negative values. This is, however, unproblematic as the labels appearing in relevant places in our tableaux will always denote non-negative values. A second reason for the partiality is due to the interpretations of the Skolem functions. The interpretations of the function symbols for negated weak until $f_{\neg(\varphi W\psi)}$ are defined depending on the satisfaction of the corresponding formulas $\neg(\varphi W\psi)$, in which case the interpretations will have the value of the first state in the future where φ does not hold. The interpretations of the function symbols for weak until (i.e., $f_{\varphi W\psi}$) do not mimic the satisfaction of the corresponding formulas so closely. Actually, it is enough for our purposes that they are only defined under the assumption that φ does not hold forever in the future. In this case, their interpretations will take the value of the first state where ψ holds. In any case, the relevant labels of this form appearing in our tableaux will always arise in contexts where their denotation is defined.

We can now define the *satisfaction of local judgments* of agent i at μ , given an assignment ρ :

- $\mu, \rho \Vdash s_i : \varphi$ if $\llbracket s_i \rrbracket_{\mu, \rho}$ is defined, $\xi_i^{\llbracket s_i \rrbracket_{\mu, \rho}} \in \Xi_i$, and $\mu_i, \xi_i^{\llbracket s_i \rrbracket_{\mu, \rho}} \Vdash_i \varphi$;

- $\mu, \rho \Vdash s_i = s'_i$ if $\llbracket s_i \rrbracket_{\mu, \rho}$ and $\llbracket s'_i \rrbracket_{\mu, \rho}$ are both defined and $\llbracket s_i \rrbracket_{\mu, \rho} = \llbracket s'_i \rrbracket_{\mu, \rho}$;
- $\mu, \rho \Vdash s_i < s'_i$ if $\llbracket s_i \rrbracket_{\mu, \rho}$ and $\llbracket s'_i \rrbracket_{\mu, \rho}$ are both defined and $\llbracket s_i \rrbracket_{\mu, \rho} < \llbracket s'_i \rrbracket_{\mu, \rho}$;
- $\mu, \rho \not\Vdash \text{CLOSED}$.

Recall that $\xi_i^{\llbracket s_i \rrbracket_{\mu, \rho}}$ denotes the $\llbracket s_i \rrbracket_{\mu, \rho}^{\text{th}}$ local state of agent i in μ . We can finally define our tableaux for local reasoning.

Definition 3.1 The *local tableaux system* \mathcal{T}_i , for agent $i \in Id$, built over sets of local judgments in \mathcal{J}_i , consists of the rules shown in Figures 1–2. \square

We assume that the reader is familiar with standard terminology and notation for tableaux, for example from [7]. As usual, a branch of a (possibly infinite) tableau is *exhausted* if no more rules are applicable, *closed* if it contains CLOSED, and *open* if it is exhausted but not closed. A tableau is *closed* if all of its branches are closed.

Figure 1 contains the rules for the logical connectives, which are straightforward, and the rules for the temporal operators, most of which are standard. For instance, the rule (F) guarantees that in order for $F\varphi$ to hold at state x , there must exist a future state v where φ holds, which is enforced by the freshness of v (i.e., v is a new variable, not occurring in the tableaux). In contrast, the rule ($\neg F$) concludes that if $\neg F\varphi$ holds at state x then φ cannot hold in any state y in the future of x . The additional premise $(i, y) : \psi$ is there only to control the introduction of labeled formulas. The rules (G) and ($\neg G$) are justified similarly. The rule (X) simply requires the existence of a suitable next state. The rule ($\neg X$) follows a pattern similar to the ones above. The rules for weak until follow closely the operator’s semantics, but some explanation is needed to clarify the use of the Skolem function symbols. Rule (W_1) splits the satisfaction of $\varphi W\psi$ at state x into two cases: either φ holds always in the future, or there is a future state $f_{\varphi W\psi}(x)$ where ψ holds. Of course, this future state, which we have required to be the earliest possible, defines together with x an interval where φ must hold. These requirements are then imposed by rule (W_2), hence justifying the use of the Skolem function $f_{\varphi W\psi}$. The rules for negated weak until ($\neg W_1$) and ($\neg W_2$) are similar.

The rules in Figure 2 define the properties of the relations. Note that we use $\theta(i, x)$ to denote any local judgment of agent i where x occurs. The rule (POS) states that the values of the labels are either 0 or greater than 0. The rule (CONG) expresses the congruence of $=$, i.e., if two labels (i, x) and (i, y) denote the same local state, then we may replace some occurrences of x by occurrences of y in any judgment. Similarly, rule (REFL) asserts the reflexivity of equality. With rule (FILL), we “fill down” the set of states: if (i, x) denotes a state and if (i, y) is smaller than (i, x) , then it should also denote a state (which we express by having truth hold there). (TR) forces the order relation to be

$$\begin{array}{c}
\frac{s_i : \neg\neg\varphi}{s_i : \varphi} (\neg\neg) \quad \frac{s_i : \varphi \quad s_i : \neg\varphi}{\text{CLOSED}} (\text{ABS}) \quad \frac{s_i : \varphi \Rightarrow \psi}{s_i : \neg\varphi \mid s_i : \psi} (\Rightarrow) \quad \frac{s_i : \neg(\varphi \Rightarrow \psi)}{s_i : \varphi, s_i : \neg\psi} (\neg\Rightarrow) \\
\\
\frac{(i, x) : \mathbf{F}\varphi}{(i, x) < (i, v), (i, v) : \varphi} (\mathbf{F}) [v \text{ fresh}] \quad \frac{(i, x) : \neg\mathbf{F}\varphi \quad (i, x) < (i, y) \quad (i, y) : \psi}{(i, y) : \neg\varphi} (\neg\mathbf{F}) \\
\\
\frac{(i, x) : \mathbf{G}\varphi \quad (i, x) < (i, y) \quad (i, y) : \psi}{(i, y) : \varphi} (\mathbf{G}) \quad \frac{(i, x) : \neg\mathbf{G}\varphi}{(i, x) < (i, v), (i, v) : \neg\varphi} (\neg\mathbf{G}) [v \text{ fresh}] \\
\\
\frac{(i, x) : \mathbf{X}\varphi}{(i, x+1) : \varphi} (\mathbf{X}) \quad \frac{(i, x) : \neg\mathbf{X}\varphi \quad (i, x) < (i, y) \quad (i, y) : \psi}{(i, x+1) : \neg\varphi} (\neg\mathbf{X}) \\
\\
\frac{(i, x) : \varphi \mathbf{W}\psi}{(i, x) : \mathbf{G}\varphi \mid (i, x) < (i, f_{\varphi\mathbf{W}\psi}(x)), (i, f_{\varphi\mathbf{W}\psi}(x)) : \psi} (\mathbf{W}_1) \quad \frac{(i, x) < s_i < (i, f_{\varphi\mathbf{W}\psi}(x))}{s_i : \varphi, s_i : \neg\psi} (\mathbf{W}_2) \\
\\
\frac{(i, x) : \neg(\varphi \mathbf{W}\psi)}{(i, x) < (i, f_{\neg(\varphi\mathbf{W}\psi)}(x)), (i, f_{\neg(\varphi\mathbf{W}\psi)}(x)) : \neg\varphi, (i, f_{\neg(\varphi\mathbf{W}\psi)}(x)) : \neg\psi} (\neg\mathbf{W}_1) \quad \frac{(i, x) < s_i < (i, f_{\neg(\varphi\mathbf{W}\psi)}(x))}{s_i : \neg\psi, s_i : \varphi} (\neg\mathbf{W}_2)
\end{array}$$

Figure 1. Rules for the logical connectives and the temporal operators.

$$\begin{array}{c}
\frac{\theta(i, x)}{(i, x) = (i, 0) \mid (i, 0) < (i, x)} (\text{POS}) \quad \frac{(i, x) = (i, y) \quad \theta(i, x)}{\theta(i, y)} (\text{CONG}) \quad \frac{\theta(i, x)}{(i, x) = (i, x)} (\text{REFL}) \quad \frac{s_i : \varphi \quad s'_i < s_i}{s'_i : \top} (\text{FILL}) \\
\\
\frac{\theta(i, x) \quad \theta(i, y)}{(i, x) < (i, y) \mid (i, x) = (i, y) \mid (i, y) < (i, x)} (\text{TR}) \quad \frac{(i, x) < (i, y) \quad \theta(i, y + c)}{(i, x) < (i, y + c)} (\text{MON}) [c > 0] \quad \frac{(i, x) < (i, y) < (i, z)}{(i, x) < (i, z - 1)} (\text{DTRANS}) \\
\\
\frac{(i, x) < (i, y) \quad \theta(i, y + c)}{(i, x + c) < (i, y + c)} (\text{RSHIFT}) [c > 0] \quad \frac{(i, x) < (i, y) \quad \theta(i, x + c)}{(i, x + c) < (i, y + c)} (\text{LSHIFT}) [c < 0] \quad \frac{\theta(i, x + 1)}{(i, x) < (i, x + 1)} (\text{SUCC}) \\
\\
\frac{(i, 0) < (i, x)}{(i, x - 1) < (i, x)} (\text{PRED}) \quad \frac{(i, x) < (i, x + c)}{\text{CLOSED}} (\text{NLOOP}) [c \leq 0] \quad \frac{(i, x + c) < (i, y) \quad \exists^\infty c \geq 0}{\text{CLOSED}} (\text{INF})
\end{array}$$

Figure 2. Rules for the relations.

trichotomic². (MON) is a form of transitivity, given that y precedes $y + c$ when $c > 0$. (DTRANS) is discrete transitivity: if (i, x) is smaller than (i, y) and (i, y) is smaller than (i, z) , then (i, x) is also smaller than (i, z) . In fact, our rule is more specific and formalizes that (i, x) is actually smaller than $(i, z - 1)$. (RSHIFT) and (LSHIFT) shift the precedence order along with addition, taking care that no new states are introduced. (SUCC) and (PRED) order successive states in appropriate conditions. The closure rule (NLOOP) states that x cannot precede $x + c$ when $c \leq 0$. Finally, (INF) is an infinitary closure rule: if in a branch there are infinitely many, distinct, non-negative constants that when added to (i, x) denote a value smaller than (i, y) , then the branch is closed.

We illustrate the use of the tableaux system with an ex-

²Note that the trichotomy rule may lead to considerable branching in the tableaux. In [1], we have replaced this rule by a set of rules that permit only controlled forms of trichotomy.

ample. To prove that $((\varphi\mathbf{W}\psi) \wedge \mathbf{X}(\neg\psi)) \Rightarrow \mathbf{X}\varphi$ is a theorem, we give a closed tableau in Figure 3 for the negated formula. Further examples are provided in [1].

Note that we do not claim the independence of all the proposed rules of our tableaux system \mathcal{T}_i . For instance, it is easy to obtain the rule (NLOOP) from (INF) by infinitely many applications of (RSHIFT) and (DTRANS). We leave the identification of a minimal, equivalent, set of rules for future work and instead we now establish the soundness and completeness of \mathcal{T}_i . We first consider soundness, where, as usual, a rule is *sound* if every model that satisfies its premises also satisfies at least one of its conclusions. Of course, a closure rule, i.e., a rule whose conclusion is CLOSED, is sound if no model satisfies its premises.

Proposition 3.2 *The rules of \mathcal{T}_i are sound.*

Before we proceed to the completeness result, we recall [18] some technical results about *integer constraints*

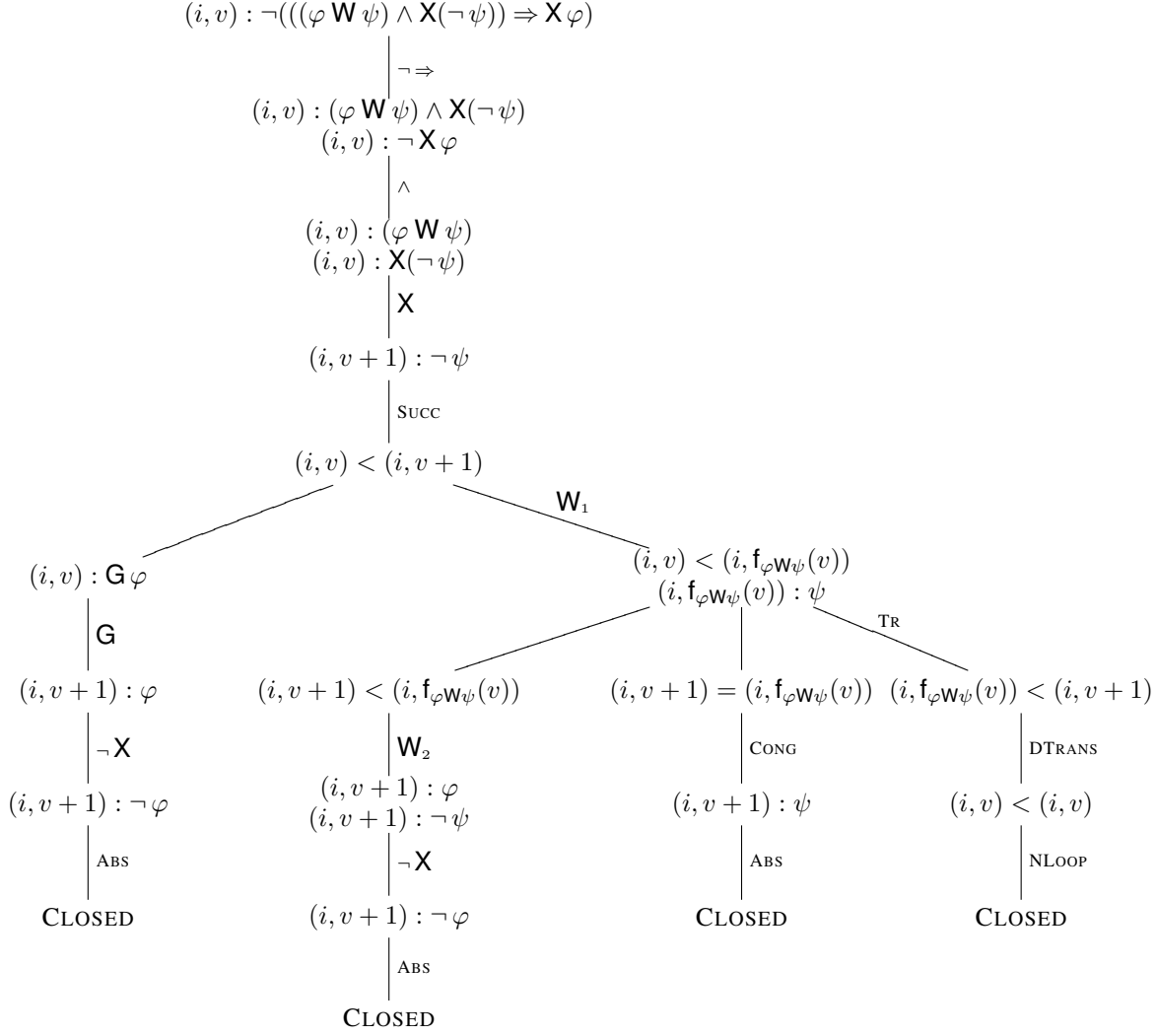


Figure 3. Tableau for $\neg(((\varphi \mathbf{W} \psi) \wedge \mathbf{X}(\neg \psi)) \Rightarrow \mathbf{X} \varphi)$.

of the form $x \leq y$, where (i, x) and (i, y) are local labels in \mathcal{S}_i . It is clear that any such constraint is of the form $u_1 + n \leq u_2 + m$, where u_1 and u_2 are either label variables, label terms whose head is a Skolem function, or 0. Let $A = \{A_1, A_2, \dots\}$ be a (possibly infinite) set of such constraints. The *constraint graph* for A is a weighted, directed graph $G_A = \langle V_A, E_A \rangle$ constructed as follows:

- $V_A = \mathcal{V}(A) \cup \{0\}$, where $\mathcal{V}(A)$ is the set of \mathcal{V}_i variables and of label terms headed by a Skolem function occurring in A^3 ;
- $E_A = \{u_1 \xrightarrow{m-n} u_2 \mid u_1 + n \leq u_2 + m \in A\} \cup \{0 \xrightarrow{0} u \mid u \in \mathcal{V}(A)\}$.

³At this point, labels whose head is a Skolem function symbol are treated as if they were simply variables.

As notation, $u_1 \xrightarrow{c} u_2$ represents the directed edge (u_1, u_2) with weight c . Intuitively, this means that u_1 is at most c larger than u_2 . So, for instance, edges of the second kind, $0 \xrightarrow{0} u$, express that $0 \leq u + 0$, which is satisfied when u is non-negative, i.e., a natural number. As usual, a *path* p in a graph is a finite sequence of vertices u_1, \dots, u_n , where (u_i, u_{i+1}) is an edge, for all i such that $1 \leq i \leq n$. The weight of a path is the sum of the weights of its edges.

Lemma 3.3 *A (possibly infinite) set of constraints A is satisfiable if and only if for each non-zero node in G_A , there exists a minimum-weight path in G_A among all the paths from 0 to that node.*

In our tableaux, every judgment of the form $(i, x) < (i, y)$ can be equivalently stated as a constraint of the form

$x \leq y - 1$. Similarly, a judgment of the form $(i, x) = (i, y)$ can be equivalently formalized as the pair of constraints $x \leq y$ and $y \leq x$.

We can now prove our completeness result for the tableaux system \mathcal{T}_i . Since the Skolem function symbols are intended to be used only as an internal tool of the system, we will assume that the initial set of judgments contains no Skolem functions at all. Note, still, that this requirement could be dropped if we added additional constraints to the graphs G_A , (1) imposing the required ordering between labels whose head is a Skolem function and their subterms (e.g., stating that $(i, s_i) < (i, f_{\varphi W\psi}(s_i))$), and (2) splitting the rules (\mathbf{W}_1) and ($\neg\mathbf{W}_1$) in two, one for introducing the Skolem symbols and another for introducing their properties.

Proposition 3.4 *Let Θ be a set of local judgments without Skolem functions. Then Θ is satisfiable if and only if there is a \mathcal{T}_i -tableau for Θ with an open branch.*

We can then reason about entailment in the logic:

Corollary 3.5 *Given $\Phi \cup \{\psi\} \in \mathcal{L}_i$, $\Phi \models_i \psi$ if and only if every exhausted \mathcal{T}_i -tableau for $\{(i, 0) : \mathbf{G}_o \varphi \mid \varphi \in \Phi\} \cup \{(i, v) : \neg\psi\}$ is closed.*

The tableau in Figure 3 shows that $\models_i ((\varphi \mathbf{W} \psi) \wedge \mathbf{X}(\neg\psi)) \Rightarrow \mathbf{X}\varphi$.

4 Tableaux for global reasoning

Our aim is to build a tableaux system \mathcal{T} for full DTL by capitalizing on the local tableaux systems for each agent $i \in Id$. We will now introduce an additional kind of *global judgment*: synchronization between labels. Labeled local formulas will also be unrestricted, i.e., communication formulas are allowed. Of course, the language of labels is now distributed. That is, if $Id = \{i_1, \dots, i_n\}$ then

$$\mathcal{S} ::= \mathcal{S}_{i_1} \mid \dots \mid \mathcal{S}_{i_n},$$

where the *local labels* of agent i are defined, as before, by

$$\mathcal{T}_i ::= \mathbb{N} \mid \mathcal{V}_i + \mathbb{Z} \mid \mathcal{F}_i(\mathcal{T}_i) + \mathbb{Z},$$

$$\mathcal{S}_i ::= (i, \mathcal{T}_i),$$

but where the *Skolem function symbols* extend now also to formulas involving communication, that is,

$$\mathcal{F}_i = \{f_{\varphi W\psi} \mid \varphi, \psi \in \mathcal{L}_i\} \cup \{f_{\neg(\varphi W\psi)} \mid \varphi, \psi \in \mathcal{L}_i\}.$$

The syntax of *global judgments* can now be defined by

$$\mathcal{J} ::= \mathcal{J}_{i_1} \mid \dots \mid \mathcal{J}_{i_n} \mid \mathcal{S}_i \bowtie \mathcal{S}_j,$$

where the local judgments are extended to also incorporate communication formulas

$$\mathcal{J}_i ::= \mathcal{S}_i : \mathcal{L}_i \mid \mathcal{S}_i = \mathcal{S}_i \mid \mathcal{S}_i < \mathcal{S}_i \mid \text{CLOSED}.$$

The intended meaning of a *synchronization judgment* $(i, x) \bowtie (j, y)$ is that the event leading to state x of agent i is synchronized with the event leading to state y of agent j . Semantically, we require a distributed *assignment on label variables* $\rho = \{\rho_i\}_{i \in Id}$. The *denotation of labels* is defined as before. The satisfaction of judgments is also just extended with

- $\mu, \rho \Vdash s_i \bowtie s_j$ if $\xi_i^{\llbracket s_i \rrbracket \mu, \rho} \neq \emptyset$, $\xi_j^{\llbracket s_j \rrbracket \mu, \rho} \neq \emptyset$ and $\text{last}_i(\xi_i^{\llbracket s_i \rrbracket \mu, \rho}) = \text{last}_j(\xi_j^{\llbracket s_j \rrbracket \mu, \rho})$.

We finally define our tableaux for global reasoning, which we show to be sound and complete.

Definition 4.1 The *global tableaux system* \mathcal{T} for DTL, built over sets of global judgments in \mathcal{J} , consists of the rules of \mathcal{T}_i for each agent $i \in Id$, together with the global rules in Figure 4.

Figure 4 contains the rules for communication and synchronization. The rules for communication, (\mathbf{C}) and ($\neg\mathbf{C}$), follow closely the semantics. Consider, for instance, (\mathbf{C}): if agent i , in state x , just communicated with agent j , for whom φ held, then the event leading to state x is synchronized with an event leading to some state v of agent j , where φ holds (and where v is fresh). The rules for synchronization are also quite intuitive. For instance, (\mathbf{EVT}) guarantees that since there is no event leading to the initial state, synchronization is not possible. The rules (\mathbf{SYM}), (\mathbf{SELF}), and (\mathbf{TRANS}) are straightforward. The rule (\mathbf{ORDER}) guarantees that local orders are globally compatible. If there is a chain of synchronizations linking two events of agent i , then these two events preserve the ordering imposed by the synchronization chain. For instance, assume that the events leading to states s_i and s'_i of agent i have just synchronized with the events leading to states s_j and s'_j of agent j , respectively. Furthermore, assume that s_j precedes s'_j . Then this order must be reflected in agent i and so s_i must precede s'_i . This extends to more than two agents in a straightforward way.

Proposition 4.2 *The rules of \mathcal{T} are sound.*

The completeness of \mathcal{T} builds on the completeness result for each of the local systems \mathcal{T}_i .

Proposition 4.3 *\mathcal{T} is complete, i.e., a set of global judgments Θ without Skolem functions is satisfiable if and only if there is a \mathcal{T} -tableau for Θ with an open branch.*

We can then reason about entailment in DTL:

$$\begin{array}{c}
\frac{(i, x) : \odot_j[\varphi]}{(j, v) : \varphi, (i, x) \bowtie (j, v)} \quad (\odot) [v \text{ fresh}] \qquad \frac{(i, x) : \neg \odot_j \varphi \quad (i, x) \bowtie (j, y)}{(j, y) : \neg \varphi} \quad (\neg \odot) \\
\\
\frac{(i, x) \bowtie (i, y)}{(i, 0) < (i, x)} \quad (\text{EVT}) \qquad \frac{s_i \bowtie s_j}{s_j \bowtie s_i} \quad (\text{SYM}) \qquad \frac{s_i \bowtie s'_i}{s_i = s'_i} \quad (\text{SELF}) \qquad \frac{s_i \bowtie s_j \quad s_j \bowtie s_k}{s_i \bowtie s_k} \quad (\text{TRANS}) \\
\\
\frac{s_{i_1} \bowtie s_{i_2} \quad s_{i_2} < s'_{i_2} \quad s'_{i_2} \bowtie s_{i_3} \quad s_{i_3} < s'_{i_3} \quad \dots \quad s'_{i_p} \bowtie s'_{i_1}}{s_{i_1} < s'_{i_1}} \quad (\text{ORDER})
\end{array}$$

Figure 4. Rules for communication and synchronization.

Corollary 4.4 *Given $\Gamma \cup \{\@_i[\varphi]\} \in \mathcal{L}$, $\Gamma \models_{\text{DTL}} \@_i[\varphi]$ if and only if every exhausted \mathcal{T} -tableau for $\{(j, 0) : \mathbf{G}_0 \psi \mid \@_j[\psi] \in \Gamma\} \cup \{(i, v) : \neg \varphi\}$ is closed.*

For example, as is done in [1], we can show that $\{\@_i[\odot_j[\top] \Rightarrow \odot_j[\mathbf{X} \odot_k[\top]]], \@_j[\odot_k[\top] \Rightarrow \odot_k[\mathbf{X} \odot_i[\top]]]\} \models_{\text{DTL}} \@_i[\odot_j[\top] \Rightarrow \mathbf{F} \odot_k[\top]]$ by building a closed \mathcal{T} -tableau for the corresponding judgments.

5 Related and future work

We have given the first sound and complete tableaux system for the distributed temporal logic DTL. To do so, we first gave a system for reasoning locally (in LTL) at each agent and afterwards we combined the local systems into one for global reasoning. We again note that our tableaux system can be smoothly extended to deal also with past operators, like since, as shown in the full paper [1].

A number of tableaux and other deductive systems have been given for fragments of LTL, e.g., [5, 7, 10, 11, 12, 13, 16, 20, 21, 22, 23, 25, 27] to name a few. In particular, different kinds of labels are employed to guide the proof search in the different labeled systems. For instance, [13] considers time points as labels for formulas, while [5, 20, 21] consider time intervals. It is interesting to note that Schmitt and Goubault-Larrecq employ constraint graphs to reason about the completeness of their rules, where labels are time intervals, similar to what we did for our time-point labels. Most importantly, different fragments of the logic are considered in the different systems to cope with the difficulties of the full logic, e.g., the difficulties of formalizing rules for until and since. The manuscript [20] is an attempt to give a labeled tableaux system for the full logic, but unfortunately it has never been completed.

We have designed our systems with the aim of providing tableaux for full DTL, including past, but it is interesting to note that our system for local reasoning seems to be closely related to the natural deduction system for future-time LTL of [2], which was developed in parallel with our work. We have begun investigating whether the rules of [2] would also be suited for the extension to global reasoning in

past and future-time DTL, and plan to report on this soon. This might be useful since, as noted in the introduction, we have not yet addressed the question of efficient proof search and have thus considered an infinite closure rule that captures eventualities which are always delayed. We will address alternative rules, leading to more efficient deduction, in future work, by adding loop checking as is usually done. When doing so, we also expect to be able to capitalize on having based our tableaux system on constraint graphs. Actually, in the finite case, our Lemma 3.3 is well known to amount to checking that there are no cycles with negative sum in the graph [18], which can be done efficiently using the Bellman-Ford algorithm [6].

Another direction for future work will be to extend our system to the *Distributed Temporal Protocol Logic* DTPL that we have devised to reason about models and properties of security protocols. In [3, 4], we have applied DTPL in two different ways: first to verify (or refute) that security protocols provide the security properties they have been designed for, and second to prove metatheoretic properties of protocol models that can be used to simplify the verification of protocols or to search for attacks against them. All of these results have been obtained directly by semantic arguments. Hence, extending the tableaux system given here to DTPL will allow us to formalize, and possibly implement, (meta)reasoning about security protocols. We will report on this in a forthcoming paper.

Acknowledgments The first author was partially supported by the Hasler Foundation, ManCom project 2071. The second and third authors were partially supported by FCT and EU FEDER, via the projects KLog PTDC/MAT/68723/2006 of SQIG-IT, and Quant-Log POCI/MAT/55796/2004 of CLC. The fourth author was partially supported by the FP7-ICT-2007-1 Project no. 216471, “AVANTSSAR: Automated Validation of Trust and Security of Service-oriented Architectures” (www.avantssar.eu). We thank Matthias Schmalz for useful comments on a draft of this paper.

References

- [1] D. Basin, C. Caleiro, J. Ramos, and L. Viganò. Labelled Tableaux for Distributed Temporal Logic. Submitted for publication, 2008. Available online at <http://wslc.math.ist.utl.pt/ftp/pub/CaleiroC/07-BCRV-dtl-tableaux.pdf>.
- [2] A. Bolotov, O. Grigoriev, and V. Shagin. Automated Natural Deduction for Propositional Linear-Time Temporal Logic. In *Proc. TIME07*, pages 47–58. IEEE Computer Society Press, 2007.
- [3] C. Caleiro, L. Viganò, and D. Basin. Metareasoning about Security Protocols using Distributed Temporal Logic. In *Proc. ARSPA'04*, pages 67–89. ENTCS 125(1), 2005.
- [4] C. Caleiro, L. Viganò, and D. Basin. Relating strand spaces and distributed temporal logic for security protocol analysis. *Logic Journal of the IGPL*, 13(6):637–664, 2005.
- [5] S. Cerrito and M. Cialdea Mayer. Labelled tableaux for propositional linear time logic over finite frames. In D. Basin, M. D'Agostino, D. M. Gabbay, S. Matthews, and L. Viganò, editors, *Labelled Deduction*. Kluwer Academic Publishers, 2000.
- [6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. MIT Press, 2001.
- [7] M. D'Agostino, D. M. Gabbay, R. Hähnle, and J. Posegga, editors. *Handbook of Tableau Methods*. Kluwer Academic Publishers, 1999.
- [8] H.-D. Ehrich and C. Caleiro. Specifying communication in distributed information systems. *Acta Informatica*, 36:591–616, 2000.
- [9] H.-D. Ehrich, M. Kollmann, and R. Pinger. Checking object system designs incrementally. *Journal of Universal Computer Science*, 9(2):106–119, 2003.
- [10] M. Fisher. Implementing Temporal Logics: Tools for Execution and Proof. In *Proc. CLIMA VI*, LNAI 3900, pages 129–142. Springer-Verlag, 2006.
- [11] M. Fisher, D. M. Gabbay, and L. Vila, editors. *Handbook of Temporal Reasoning in Artificial Intelligence I*. Elsevier, 2005.
- [12] G. D. Gough. Decision procedures for temporal logic. Technical Report UMCS-89-10-1, Department of Computer Science, University of Manchester, 1984.
- [13] R. Hähnle and O. Ibens. Improving Temporal Logic Tableaux Using Integer Constraints. In *Proc. ICTL'94*, LNAI 827. Springer-Verlag, 1994.
- [14] K. Lodaya, R. Ramanujam, and P. Thiagarajan. Temporal logics for communicating sequential agents: I. *Intern. Journal of Foundations of Computer Science*, 3(1):117–159, 1992.
- [15] K. Lodaya and P. Thiagarajan. A modal logic for a subclass of event structures. In *Proc. ICALP 14*, LNCS 267, pages 290–303. Springer-Verlag, 1987.
- [16] Z. Manna and A. Pnueli, editors. *Temporal Verification of Reactive Systems: Safety*. Springer-Verlag, 1995.
- [17] D. Peled. All from one, one for all: on model checking using representatives. In *Proc. CAV '93*, pages 409–423. Springer-Verlag, 1993.
- [18] V. Pratt. Two easy theories whose combination is hard. Technical report, MIT, Cambridge, 1977.
- [19] R. Ramanujam. Locally linear time temporal logic. In *Proc. LICS 11*, pages 118–127. IEEE Computer Society Press, 1996.
- [20] P. H. Schmitt and J. Goubault-Larrecq. A tableau system for full linear temporal logic. Unpublished manuscript.
- [21] P. H. Schmitt and J. Goubault-Larrecq. A Tableau System for Linear-TIME Temporal Logic. In *Proc. TACAS'97*, LNCS 1217, pages 130–144. Springer-Verlag, 1997.
- [22] S. Schwendimann. A New One-Pass Tableau Calculus for PLTL. In *Proc. Tableaux'98*, LNAI 1397, pages 277–291. Springer-Verlag, 1998.
- [23] R. Scott, M. Fisher, and J. Keane. Parallel Temporal Tableaux. In *Proc. Euro-Par'98*, LNAI 1470, pages 852–861. Springer-Verlag, 1998.
- [24] P. S. Thiagarajan. A Trace Consistent Subset of PTL. In *Proc. CONCUR'95*, LNCS 962, pages 438–452. Springer-Verlag, 1995.
- [25] L. Viganò and M. Volpe. Labeled Natural Deduction Systems for a Family of Tense Logics. This volume.
- [26] G. Winskel. Event structures. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Applications and Relationships to Other Models of Concurrency*, LNCS 255, pages 325–392. Springer-Verlag, 1987.
- [27] P. Wolper. The tableau method for temporal logic: An overview. *Logique et Analyse*, 110:119–136, 1985.