A Decidable Logic for Time Intervals: Propositional Neighborhood Logic

Angelo Montanari
University of Udine, Italy
Department of Mathematics and Computer Science
Via delle Scienze 206, 33100 Udine, Italy
ph. +39 0432 558477, fax +39 0432 558499
montana@dimi.uniud.it

Guido Sciavicco
University of Udine, Italy
Department of Mathematics and Computer Science
Via delle Scienze 206, 33100 Udine, Italy
ph. +39 0432 558451, fax +39 0432 558499
sciavicc@dimi.uniud.it

Abstract

Logics for time intervals provide a natural framework for representing and reasoning about timing properties in various areas of artificial intelligence and computer science. Unfortunately, most time interval logics proposed in the literature are (highly) undecidable. Decidable fragments of these logics have been obtained by imposing severe restrictions on their expressive power. In this paper, we focus our attention on the propositional fragment of Neighborhood Logic (PNL for short). We show that PNL is expressive enough to capture meaningful timing properties and that it is decidable. Decidability is proved by developing an original tableau decision method for PNL. We conclude the paper by pointing out interesting relationships between PNL and compass logics for spatial reasoning.

Introduction

Logics for time intervals provide a natural framework for dealing with timing properties in various areas of computer science and artificial intelligence, such as planning and natural language processing, where reasoning about time intervals rather than time points is far more natural and closer to common sense (point-based and interval-based temporal logics are systematically analyzed in (van Benthem 1991)).

Unfortunately, most interval temporal logics and duration calculi proposed in the literature, such as Moszkowski's Interval Temporal Logic (ITL) (Halpern, Manna, & Moszkowski 1983), Halpern and Shoham's Modal Logic of Time Intervals (HS) (Halpern & Shoham 1991), Venema's CDT logic (Venema 1991), Chaochen and Hansen's Neighborhood Logic (NL) (Chouchen & Hansen 1998), and Chaochen, Hoare, and Ravn's Duration Calculus (DC) (Chaochen, Hoare, & Ravn 1991), are (highly) undecidable.

ITL is provided with the two modal operators $\bigcirc$ (*next*) and ; (*chop*). An ITL interval is a finite or infinite sequence of states. Given two formulas φ, ψ and an interval $s_0, \ldots, s_n$, $\bigcirc\varphi$ holds over $s_0, \ldots, s_n$ if and only if φ holds over $s_1, \ldots, s_n$, while $\varphi ; \psi$ holds over $s_0, \ldots, s_n$ if and only if there exists i, with $0 \leq i \leq n$, such that φ holds over $s_0, \ldots, s_i$ and ψ holds over $s_i, \ldots, s_n$. The undecidability of Propositional ITL has been proved by a reduction from the problem of testing the emptiness of the intersection of two grammars in Greibach form (Moszkowski 1983). HS features three basic operators $\langle \mathrm{A} \rangle$ (after), $\langle \mathrm{B} \rangle$ (begin), and $\langle \mathrm{E} \rangle$ (end), together with their duals $\langle \bar{\mathrm{A}} \rangle$, $\langle \bar{\mathrm{B}} \rangle$, and $\langle \bar{\mathrm{E}} \rangle$. Given a formula φ and an interval $[a, b]$, $\langle \mathrm{A} \rangle\varphi$ holds at $[a, b]$ if and only if φ holds at $[b, c]$, for some $c > b$, $\langle \mathrm{B} \rangle\varphi$ holds at $[a, b]$ if and only if φ holds at $[a, c]$, for some $c < b$, and $\langle \mathrm{E} \rangle\varphi$ holds at $[a, b]$ if and only if φ holds at $[c, b]$, for some $c > a$. A number of other temporal operators can be defined by means of the basic ones. As an example, the subinterval operator $\langle D \rangle$ such that $\langle D \rangle\phi$ holds at a given interval $[a, b]$ if and only if ϕ holds at a proper subinterval $[c, d]$ of $[a, b]$ can be defined as $\langle \mathrm{B} \rangle\langle \mathrm{E} \rangle\phi$ or, equivalently, $\langle \mathrm{E} \rangle\langle \mathrm{B} \rangle\phi$. HS has been shown to be undecidable by coding the halting problem in it. CDT has three binary operators C, D, and T, which informally deal with the situations generated by adding an extra point in one of the three possible positions with respect to the two points delimiting an interval (before, in between, and after). Since HS is a subsystem of CDT, the undecidability of the latter easily follows. NL is a first-order interval logic with two expanding modalities $\Diamond_l$ and $\Diamond_r$ and a special symbol l which denotes the length of the current interval. Given a formula φ and an interval $[a, b]$, $\Diamond_l\varphi$ holds at $[a, b]$ if and only if φ holds at $[c, a]$, for some $c \leq a$, $\Diamond_r\varphi$ holds at $[a, b]$ if and only if φ holds at $[b, c]$, for some $c \geq b$, and the valuation of l over $[a, b]$ is $b - a$. NL undecidability can be easily proved by embedding HS in it. Finally, DC extends ITL by adding *temporal variables* (also called state expressions) as integrals of state variables in order to model dynamic systems in a continuous time. Temporal variables make it possible to represent the duration of intervals as well as numerical constants. As an example (Sørensen, Ravn, & Rischel 1990), the specification of the behavior of a gas burner can include conditions as the following one: "for any period of 30 seconds the gas may leak, that is, flow and not burn, only once and for 4 seconds at most". Such a condition is expressed by the DC formula: $l > 30 \vee ((\int(\neg Gas \vee Flame) = l); (\int(Gas \wedge \neg Flame) = l \wedge l \leq 4); (\int(\neg Gas \vee Flame) = l))$, where *Gas* (the gas is flowing) and *Flame* (the gas is burning) are two state variables. In (Chaochen, Hansen, & Sestoft 1993) Chouchen et al. showed that DC is undecidable, the main source of undecidability being the fact that state changes in real-time systems can occur at any time point.

The problem of finding decidable fragments of these logics has been raised by several authors, including Halpern and Shoham (cf. Problem 4 in (Halpern & Shoham 1991)) and Venema (cf. Question 3.20 in (Venema 1991)). In general, decidable fragments have been obtained by imposing severe restrictions on the expressive power of the logics, e.g., (Moszkowski 1983; Bowman & Thompson 1998). As an example, Moszkowski (Moszkowski 1983) proves the decidability of the fragment of Propositional ITL with Quantification (over propositional variables) obtained by imposing a suitable *locality* constraint. Such a constraint states that each propositional variable is true over an interval if and only if it is true at its first state. This allows one to collapse all the intervals starting at the same state into the single interval consisting of the first state only. By exploiting such a constraint, decidability of Local ITL can be easily proved by embedding it into Quantified Propositional Linear Temporal Logic.

In this paper, we focus our attention on the propositional fragment of Neighborhood Logic (PNL for short). Even though PNL does not involve any locality constraint, its satisfiability problem is decidable and its language is expressive enough to capture meaningful timing properties. Decidability is proved by developing an original tableau decision method for PNL. Such a tableau method can be classified as an "explicit" system (a detailed account of the existing tableau methods can be found in (D'Agostino *et al.* 1999)). Its general structure as well as the form of the proofs are inspired by standard tableau procedures for first-order logics with free variables and for modal logics. At the best of our knowledge, there exist only two tableau methods for time interval logics in the literature. In (Bowman & Thompson 1998), Bowman and Thompson consider an extension of Local ITL, which, besides the chop operator ;, contains a projection operator $proj$ and an empty interval modal constant $empty$. They introduce a normal form for the formulas of the resulting logic that allows them to exploit a classical tableau method, devoid of any mechanism for constraint label management. In (Chetcuti-Serandio & Fariñas del Cerro 2000), the authors identify a decidable fragment of DC, which is expressive enough to model the above-given condition on the behavior of a gas burner, that imposes no restriction on state expressions, but encompasses a proper subset of DC operators, namely, $\wedge$, $\vee$ and ; (chop). The tableau construction for the resulting logic mixes the application of the rules of classical tableaus and that of a suitable constraint resolution algorithm.

The rest of the paper is organized as follows. We first introduce PNL and we discuss its expressive power. Then, we focus on decidability issues for PNL. We develop an original tableau method for PNL and we prove that it is terminating, sound, and complete. Finally, we establish a connection between PNL and suitable fragments of HS and of full compass logic. In the conclusions, we provide an assessment of the work and outline further research directions.

Propositional Neighborhood Logic

In this section, we define the syntax and semantics of PNL. We also discuss its strength and limitations in expressive power.

PNL is a proper fragment of NL. The language $\mathcal{L}$ for PNL consists of a set of propositional variables $\mathcal{AP} = \{P, Q, \ldots\}$, of the propositional connectives $\neg$ and $\wedge$ ($\vee$ and $\rightarrow$ can be defined in the usual way), the *left neighborhood modality* $\Diamond_l$, and the *right neighborhood modality* $\Diamond_r$. PNL *formulas*, denoted as $\phi, \psi, \ldots$, are defined according to the following abstract syntax:

$$\phi ::= P | \neg\phi | \phi \wedge \psi | \Diamond_l \phi | \Diamond_r \phi.$$

Examples of well-formed PNL formulas are $P \wedge \Diamond_r P$ and $\Diamond_l (P \rightarrow Q)$. In the following we will use the modalities $\Box_l$ and $\Box_r$ as abbreviations of $\neg\Diamond_l\neg$ and $\neg\Diamond_r\neg$, respectively. We define the *length* of a PNL formula ϕ as the number of its modal and classical operators, and we denote it by $|\phi|$; for instance, $|(P \wedge (\Diamond_r Q \vee \Box_l P))| = 4$.

From a semantic point of view, we assume our domain to be a nonempty point-based set $\mathbb{D}$ with a total ordering $<$. Examples of possible domains are $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$. Given a time domain $\mathbb{D}$, the set of all intervals $\mathbb{I}$ over $\mathbb{D}$ is given by

$$\mathbb{I} = \{[a, b] : a, b \in \mathbb{D} \wedge a \leq b\}.$$

The meaning of propositional variables is given through a *valuation function*, or *value assignment*, $V : \mathbb{I} \mapsto 2^{\mathcal{AP}}$, namely, for any propositional variable P, if $P \in V([a, b])$, then P is *true* in $[a, b]$, otherwise it is false. We shall call the pair $M = (\mathbb{D}, V)$ an *interval model* or, simply, a *model*. $M, [a, b] \Vdash \phi$ stands for the formula ϕ being satisfied over the interval $[a, b]$ (called *valuation* or *starting interval*) with respect to the model M. *Satisfiability* can be defined in the standard way by induction on the structure of the formulas:

1. $M, [a, b] \Vdash P$ iff $P \in V([a, b])$, where P is a propositional letter;
2. $M, [a, b] \Vdash \phi \wedge \psi$ iff $M, [a, b] \Vdash \phi$ and $M, [a, b] \Vdash \psi$;
3. $M, [a, b] \Vdash \neg\psi$ iff it is not the case that $M, [a, b] \Vdash \psi$;
4. $M, [a, b] \Vdash \Diamond_l \psi$ iff there exists a point $c \leq a$ such that $M, [c, a] \Vdash \psi$;
5. $M, [a, b] \Vdash \Diamond_r \psi$ iff there exists a point $c \geq b$ such that $M, [b, c] \Vdash \psi$.

We say that ϕ is *valid* (denoted by $\models \phi$) if and only if for any model M and interval $[a, b]$, $M, [a, b] \Vdash \phi$.

It is worth noticing that, as in HS, ITL, and CDT, we assume intervals to be *closed* (this implies, for instance, that two meeting intervals, in Allen's terminology (Allen & Ferguson 1994), share a point). Notice also that $\Diamond_l$ and $\Diamond_r$ are the reflexive versions of $\langle \overline{A} \rangle$ and $\langle A \rangle$ of HS, respectively. We shall later show that the proposed tableau method can actually be adapted to decide the subset of HS containing $\langle A \rangle$ and $\langle \overline{A} \rangle$ only.

A sound and complete axiom system for PNL can be easily tailored from that for NL (Barua, Roy, & Chaochen 2000). The axiom system for PNL consists of the following set of axioms:

1. axioms of propositional logic;
2. $\Diamond_r(\phi \vee \psi) \rightarrow \Diamond_r \phi \vee \Diamond_r \psi$ (distributivity of modalities);
3. $\Diamond_r \Diamond_l \phi \rightarrow \Box_r \Diamond_l \phi$ (same point of starting);

4. $\Diamond_r\Diamond_r\Diamond_r\phi \leftrightarrow \Diamond_r\Diamond_r\phi$ (sum of intervals);
5. axioms 2, 3, and 4 with left modalities substituted for right modalities, and vice versa,

and the following set of rules ($\vdash \phi$ stands for ϕ is derivable in PNL):

1. if $\vdash \phi \rightarrow \psi$, then $\vdash \Diamond_r\phi \rightarrow \Diamond_r\psi$ (monotonicity);
2. if $\vdash \phi$, then $\vdash \Box_r\phi$ (necessity);
3. if $\vdash \phi$ and $\vdash \phi \rightarrow \psi$, then $\vdash \psi$ (modus ponens);
4. schemes 1 and 2 with left modalities substituted for right modalities.

By exploiting the results given in (Barua, Roy, & Chaochen 2000), we can prove the following result.

Theorem 1 *For all $\phi \in PNL$, $\vdash \phi$ if and only if $\models \phi$.*

PNL is expressive enough to capture relevant timing properties. As an example, conditions of the form "From now on, it will be true that any occurrence of *stop* is always preceded by an occurrence of *start*" are quite common in the area of formal specifications of reactive systems (we found it in the context of the specification of a time-triggered protocol which allows a fixed number of stations to communicate via a shared bus). Such conditions can be expressed in PNL as follows:

$$\Box_r(\Diamond_r(stop \rightarrow \Diamond_l\Diamond_l start))$$

Furthermore, a wholistic version of the *until* operator $\phi\langle U\rangle\psi$ can be expressed in PNL by means of the formula $\psi \equiv \Diamond_r(\phi \wedge \Diamond_r\psi)$. As an example, conditions like "each flight from Milan to Moscow initiates a period of time during which the traveller is in Moscow" can be expressed as follows:

$$\Box_r\Box_r\Box_l\Box_l(Milan\text{-}to\text{-}Moscow\ \langle U\rangle\ stay\text{-}in\text{-}Moscow)$$

A wholistic version of the *since* operator can be obtained in a similar way. Notice that a decomposable version of these operators would require to force homogeneity either implicitly (via the assumption of the homogeneity principle (Allen & Ferguson 1994)) or explicitly (by means of subinterval operators).

As for the limitations in the expressive power of PNL, it is possible to show that there exist operators of time interval logics which cannot be expressed in it. As an example, a bisimulation argument suffices to show that the *proper subinterval* operator $\langle D\rangle$ cannot be expressed in PNL. Consider two models $M = (\mathbb{D}, V)$ and $M' = (\mathbb{D}', V')$ such that $\mathbb{D} = \{a,b,c,d\}, \{\lessdot(a,b), \lessdot(b,c), \lessdot(c,d)\}, V([b,c]) = \{P\}$ and $\mathbb{D} = \{a',b'\}, \{<(a',b')\}$, $V([a',b']) = \{P\}$. It is easy to see that $M,[a,b] \models \Diamond_r P$, $M,[c,d] \models \Diamond_l\ P$, $M',[a',a'] \models \Diamond_r P$, and $M',[b'b'] \models \Diamond_l P$. The function $f : M \mapsto M'$ such that $f(a) = f(b) = a', f(c) = f(d) = b'$ can be extended to a bisimulation relation between intervals and the "formula" $\langle D\rangle P$ (which obviously does not belong to $\mathcal{L}$) is satisfied in M, but not in M'.

The Tableau Method for PNL

In this section, we develop a new tableau decision procedure for checking the satisfiability of PNL formulas. The tableau we propose is an *explicit tableau*. This means that the accessibility relation is maintained by some external device and that the nodes of the tableau contain *labeled* formulas. Labels are built over the language we describe below. We assume the existence of enumerable sets $\mathcal{V} = \{x, y, z, \ldots\}$ of variables, $\mathcal{C} = \{a, b, c, \ldots\}$ of constants, and $\mathcal{F} = \{f_1, f_2, \ldots\}$ of function symbols. *Interval terms* (or, simply, *terms*) are defined as constants, variables, or function symbols applied to non-constant terms, e.g., $x, c, f_1(f_2(x))$ are terms, $f_1(c)$ is not. The set of all interval terms is denoted by $\mathcal{T}$. All formulas in the tableau are labeled by a pair of terms $[i, j]$, called *reference interval*, which can be viewed as the tableau counterpart of valuation intervals. A reference interval is called *local* if it does not contain variables, *non local* otherwise. Terms and reference intervals reflect the semantics of PNL formulas. As an example, evaluating $(\phi, [a, x])$ means evaluating ϕ in all intervals beginning with a. A term of the form $f(x)$ denotes a point to be placed in a suitable way with respect to the point denoted by x (it comes into play, for instance, in the tableau for the PNL-formula $\Box_r\Diamond_r\phi$). In the standard way, we partition PNL formulas in four *syntactic types*: the *conjunctive* type (called α), the *disjunctive* type (called β), and the *universal* and *existential* types (resp. called γ and δ). In the following, we will use the notation $\phi \in \tau$, where $\tau \in \{\alpha, \beta, \gamma, \delta\}$, to state that ϕ is of type τ. Moreover, we will indicate with $\phi(\psi)$ the fact that ψ is a subformula of ϕ. Table 1 shows the *immediate subformulas* of a given formula, together with their reference intervals. Evaluating formulas of types δ and γ involves *new terms*.

The key notion of the tableau method for PNL is that of *suitable substitution*.

Definition 2 *A set of* term constraints $\mathcal{H}$ *is a set of inequalities of the form $i \leq j$, where i, j are terms.*

As an example, suppose to evaluate $\neg\Diamond_r\phi$ over the reference interval $[a, b]$, with $\mathcal{H} = \{a \leq b\}$. According to Table 1, this means that $\neg\phi$ has to be evaluated over the reference interval $[b, x]$, with $\mathcal{H} = \{a \leq b, b \leq x\}$.

Definition 3 *A partial function $\rho : \mathcal{V} \mapsto \mathcal{T}$ is a* suitable substitution *if and only if $\mathcal{H}\rho \subseteq \mathcal{H}$, provided that all expressions of the form $i \leq i$ have been eliminated*[1].

Consider, for instance, the set of term constraints $\mathcal{H} = \{b \geq a, c \leq a, d \geq b, x \geq a\}$, graphically depicted in Figure 1. According to Definition 3, the substitution ρ such that

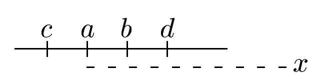

Figure 1: The set of term constraints $\mathcal{H}$.

$\rho(x) = b$ is a suitable substitution. On the contrary, neither

[1]With an abuse of notation, we use $\mathcal{H}\rho$ to indicate the application of ρ to the variables of $\mathcal{H}$.

type	labeled formulas	labeled immediate subformulas	new terms
α	$\neg\neg\psi, [i,j]$	$\psi, [i,j]$	–
	$\psi_1 \wedge \ldots \wedge \psi_n, [i,j]$	$\psi_1, [i,j], \ldots, \psi_n, [i,j]$	–
	$\neg(\psi_1 \vee \ldots \vee \psi_n), [i,j]$	$\neg\psi_1, [i,j], \ldots, \neg\psi_n, [i,j]$	–
	$\neg(\psi \rightarrow \varphi), [i,j]$	$\psi, [i,j], \neg\varphi, [i,j]$	–
β	$\psi_1 \vee \ldots \vee \psi_n, [i,j]$	$\psi_1, [i,j], \ldots, \psi_n, [i,j]$	–
	$\neg(\psi_1 \wedge \ldots \wedge \psi_n), [i,j]$	$\neg\psi_1, [i,j], \ldots, \neg\psi_n, [i,j]$	–
	$\psi \rightarrow \varphi, [i,j]$	$\neg\psi, [i,j], \varphi, [i,j]$	–
γ	$\Box_r\psi, [i,j]$	$\psi, [j,x]$	$x \geq j$
	$\neg\Diamond_r\psi, [i,j]$	$\neg\psi, [j,x]$	$x \geq j$
	$\Box_l\psi, [i,j]$	$\psi, [x,i]$	$x \leq i$
	$\neg\Diamond_l\psi, [i,j]$	$\neg\psi, [x,i]$	$x \leq i$
δ	$\Diamond_r\psi, [i,c]$ $(\Diamond_r\psi, [i,y])$	$\psi, [c,d]$ $(\psi, [y, f(y)])$	$d \geq c$ $(f(y) \geq y)$
	$\Diamond_l\psi, [c,j]$ $(\Diamond_l\psi, [x,j])$	$\psi, [d,c]$ $(\psi, [f(x), x])$	$d \leq c$ $(f(x) \leq x)$
	$\neg\Box_r\psi, [i,c]$ $(\neg\Box_r\psi, [i,y])$	$\neg\psi, [c,d]$ $(\neg\psi, [y, f(y)])$	$d \geq c$ $(f(y) \geq y)$
	$\neg\Box_l\psi, [c,j]$ $(\neg\Box_l\psi, [x,j])$	$\neg\psi, [d,c]$$(\neg\psi, [f(x), x])$	$d \leq c$ $(f(x) \leq x)$

Table 1: types, labeled formulas, labeled immediate subformulas, and new terms.

the substitution ρ' such that $\rho'(x) = c$ nor the substitution ρ'' such that $\rho''(x) = d$ is a suitable substitution, because both the constraint $c \geq a$ and the constraint $d \geq a$ do not belong to $\mathcal{H}$.

As a matter of fact, suitable substitutions look for contradictions over a given interval. The inclusion condition, that identifies the finite set of suitable substitutions, prevents us both from collapsing distinct intervals and from introducing new intervals (through the transitivity of the ordering relation) over which the given formula does not state anything. Furthermore, the elimination of any constraint of the form $i \leq i$ follows from the reflexivity of the PNL operators. As an example, suitable substitutions must take into account that an expression of the form $\Box_r\Box_r\phi$ states that ϕ holds over all future intervals, including those which are met by the current one. In order to simplify the notation, in the following we will use $[i,j]\rho$ to indicate the reference interval obtained by applying ρ to i and/or j if i and/or j are variables.

A tableau for a PNL formula ϕ is a pair $(T, \mathcal{H})$, where T is a finitely-branching tree and $\mathcal{H}$ is a set of term constraints. *Ancestors* in the tree are defined in the standard way. The tree T is generated by the expanding rule below. The *nodes* of T are labeled formulas of the form $(\phi, [i,j])$, where ϕ is a PNL formula and $[i,j]$ is a reference interval. We say that a node is local or non local depending on its reference interval being local or not. The basic operation of the expanding rule consists in extending the branch B with a finite path of one or more nodes $n_1, \ldots, n_k$, denoted by $B \oplus n_1 \oplus \ldots \oplus n_k$. Furthermore, we will use the notation $B \oplus n_1 | \ldots | n_k$ to denote the result of adding k sons to B. Finally, we will denote by $B := B'$ the operation of replacing the branch B by the branch B'.

We term *fresh* a non local node $n = (\phi, [i,j])$ if and only if n has no ancestor $n' = (\psi(\phi), [i,j])$, with $\psi \in \beta$. A node on which the expanding rule has been applied is said to be *used*; a modal operator $\nabla \in \{\Box_r, \Box_l, \Diamond_r, \Diamond_l\}$ is *used* if and only if there is at least a used node of the form $(\psi(\nabla\phi), [i,j])$ or $(\psi(\neg\nabla\phi), [i,j])$.

The *expanding rule* for a node $n = (\phi, [i,j])$ and term constraint set $\mathcal{H}$ consists of the following steps:

(a) case ϕ of

- α for all branches B containing n, $B := B \oplus (\phi_1, [i,j]) \oplus \ldots \oplus (\phi_k, [i,j])$, where $\phi_1, \ldots, \phi_k$ are the immediate sub-formulas of ϕ;
- β for all branches B containing n, $B := B \oplus (\phi_1, [i,j]) | \ldots | (\phi_k, [i,j])$, where $\phi_1, \ldots, \phi_k$ are the immediate sub-formulas of ϕ;
- γ, δ for all branches B containing n, $B := B \oplus (\phi', [i', j'])$, where ϕ' is the immediate subformula of ϕ and $[i', j']$ is the correspondent reference interval; then update $\mathcal{H}$ accordingly (cf. Table 1);

(b) if n is non local and fresh, then for all branches B containing n and all suitable substitutions $\rho_1, \ldots, \rho_k$, $B := B \oplus (\phi, [i,j]\rho_1) \oplus \ldots \oplus (\phi, [i,j]\rho_k)$;

(c) if n is non local and fresh, then for all branches B containing n, if there is at least one not used modal operator in B, then $B := B \oplus (\phi, [i,j])$.

The intuition behind steps b and c of the expanding rule is the following one. Consider a node $(\phi, [c,x])$. Such a node states that ϕ must hold over every interval beginning with c. Hence, all suitable substitutions ρ must be considered, and, for each of them, $(\phi, [c, \rho(x)])$ must be added to all branches B including the node (step b). Suppose also that there is a non used modal operator in a branch B. In such a case, it can be the case that step b has to be repeated later on. Step c guarantees that a fresh copy of the node $(\phi, [c,x])$ occurs in the branch B.

Definition 4 *A* tableau *for the* PNL-*formula ϕ is a pair $(T, \mathcal{H})$ generated by the following algorithm:*

1. *given an input formula ϕ, build a tree with one node (the root) labeled with $(\phi, [a,b])$;*
2. *let $\mathcal{H} = \{a \leq b\}$;*

3. *while there is at least a non used node $(\psi, [i, j])$, apply the expanding rule on it;*
4. *the output is the resulting pair $(T, \mathcal{H})$.*

$$\mathcal{H} = \{a \leq b, b \leq c, b \leq x\} \text{ and } T =$$

$$\begin{array}{ccccr}
 & (\Box_r(P \vee Q) \wedge \Diamond_r\neg P, [a, b]) & & & (1) \\
 & | & & & \\
 & (\Box_r(P \vee Q), [a, b]) & & & (2) \\
 & | & & & \\
 & (\Diamond_r\neg P, [a, b]) & & & (2) \\
 & | & & & \\
 & ((P \vee Q), [b, x]) & & & (3) \\
 & | & & & \\
 & (\neg P, [b, c]) & & & (4) \\
 & / \quad \backslash & & & \\
(P, [b, x]) & & (Q, [b, x]) & & \\
| & & | & & (5) \\
((P \vee Q), [b, c]) & & (P \vee Q, [b, c]) & & \\
/ \quad \backslash & & / \quad \backslash & & \\
(P, [b, c]) \; (Q, [b, c]) & & (P, [b, c]) \; (Q, [b, c]) & & (6) \\
| & & | & & \\
\textit{closed} & & \textit{closed} & &
\end{array}$$

Figure 2: The tableau for $\Box_r(P \vee Q) \wedge \Diamond_r\neg P$.

If Γ is a set of PNL formulas and there exists a tableau for all $\phi \in \Gamma$, we say that there exists a tableau for Γ.

A *contradiction* is a pair of nodes of the forms $(P, [i, j])$ and $(\neg P, [i, j])$, where P is a propositional letter. We name *open* a branch of a tableau T if there is no contradiction on it, and *closed* otherwise. Accordingly, a tableau is open if and only if it has an open branch, and closed otherwise.

As an example, in Figure 2 we show the tableau for the PNL-formula $\Box_r(P \vee Q) \wedge \Diamond_r\neg P$. The choice of the node to expand has been done according to an "uppermost leftmost" policy. As another example, the tableau $(T, \mathcal{H})$ for the formula $\Diamond_l(P \wedge \Box_r Q) \wedge \Diamond_r\neg Q$ is given in Figure 3. It is worth noting that the resulting set $\mathcal{H}$ of term constraints is the set of constraints represented in Figure 1. Furthermore in step 8 we substituted b, but not d, for x. Indeed, $x \mapsto d$ is not a suitable substitution, because it would introduce the new constraint $a \leq d$.

The proposed algorithm always terminates. Termination easily follows from two observations: (1) the application of step (a) to a node $n = (\phi, [i, j])$ produces only nodes $n' = (\psi, [i', j'])$ such that $|\psi| < |\phi|$, and (2) steps (b) and (c) can be applied only a finite number of times to any given node (being finite the number of possible terms in $\mathcal{H}$).

The proof of the soundness and completeness of the proposed tableau method takes advantage of the following notion of Hintikka set for PNL.

Definition 5 *An* Hintikka Set for PNL $(S, \mathcal{H})$ *consists of a downward saturated set of nodes S without contradictions and the corresponding set of term constraints $\mathcal{H}$ (hereafter*

$$\mathcal{H} = \{a \leq b, b \leq d, c \leq a, a \leq x\} \text{ and } T =$$

$$\begin{array}{cr}
(\Diamond_l(P \wedge \Box_r Q) \wedge \Diamond_r\neg Q, [a, b]) & (1) \\
| & \\
(\Diamond_l(P \wedge \Box_r Q), [a, b]) & (2) \\
| & \\
(\Diamond_r\neg Q, [a, b]) & (2) \\
| & \\
((P \wedge \Box_r Q), [c, a]) & (3) \\
| & \\
(\neg Q, [b, d]) & (4) \\
| & \\
(P, [c, a]) & (5) \\
| & \\
(\Box_r Q, [c, a]) & (6) \\
| & \\
(Q, [a, x]) & (7) \\
| & \\
(Q, [a, b]) & (8)
\end{array}$$

Figure 3: The tableau for $\Diamond_l(P \wedge \Box_r Q) \wedge \Diamond_r\neg Q$.

we write "Hintikka set" for "Hintikka set for PNL*"). It can be generated by the following inductive clauses:*

1. *if S contains $(\phi, [i, j])$ then $\mathcal{H}$ contains the term constraint $i \leq j$;*
2. *if S contains $(\phi, [i, j])$ and $\phi \in \alpha$, then for all ϕ_i immediate subformulas of ϕ, S contains $(\phi_i, [i, j])$;*
3. *if S contains $(\phi, [i, j])$ and $\phi \in \beta$, then S contains at least one $(\phi_i, [i, j])$, where ϕ_i is an immediate subformula of ϕ;*
4. *if S contains $(\phi, [i, j])$ and ϕ belongs to γ or δ, then S contains $(\phi', [i', j'])$, where ϕ' is the immediate subformula of ϕ and $[i', j']$ is the corresponding reference interval, and $\mathcal{H}$ contains the corresponding constraint on i', j';*
5. *if S contains a non local node $(\phi, [i, j])$ and does not contain any node $(\psi(\phi), [i, j])$ with $\psi \in \beta$ then, for all suitable substitutions $\rho_1, \ldots, \rho_k$, S contains $(\phi, [i, j]\rho_1), \ldots, (\phi, [i, j]\rho_k)$;*
6. *S contains no pair of local nodes $(P, [i, j])$ and $(\neg P, [i, j])$.*

It is possible to prove the following lemma (the proof is rather straightforward and thus omitted).

Lemma 6 *If ϕ is a satisfiable* PNL *formula, then there exists an Hintikka set containing $(\phi, [a, b])$.*

Observation 7 *An Hintikka set is the (set-theoretic) semantic counterpart of an open branch in a tableau for* PNL*. In particular, step (b) and (c) of the expanding rule are covered by rules (4) and (5) for Hintikka sets, while rule (6) is exactly the definition of open branch.*

Lemma 8 (Hintikka' s Lemma for PNL**)** *If $(S, \mathcal{H})$ is an Hintikka set, then there exists a model satisfying all* PNL *formulas* ϕ *such that* $(\phi, [i,j]) \in S$ *(we say that* $(S, \mathcal{H})$ *is* satisfiable*).*

Proof.

Let $(S, \mathcal{H})$ be an Hintikka set. Define $S' \subseteq S$ as the set of all nodes $(\phi, [i,j])$ in S such that $|\phi|$ is at most t, and $\mathcal{H}' \subseteq \mathcal{H}$ as the term constraint set associated to S'. By definition, $(S', \mathcal{H}')$ is downward saturated. We show by induction that $(S', \mathcal{H}')$ is satisfiable for every $t \in \mathbb{N}$.

Base case. Let ϕ be a PNL formula such that $(\phi, [i,j]) \in S'$ and $|\phi| = 0$. Clearly, $\phi \equiv P$, with P propositional variable. By definition of Hintikka set (absence of contradictions), S does not contain $(\neg P, [i,j])$; thus, a model stating P for all intervals denoted by $[i,j]$, with the possible exception of those local intervals $[c,c']$ such that $(\neg P, [c,c']) \in S'$ (S' may contain at most a finite number of "local" contradictions), satisfies ϕ.

The inductive case ($t > 0$) depends on the syntactic type of ϕ:

- $\phi \in \alpha$. By definition, S' contains all pairs $(\phi_i, [i,j])$, with ϕ_i immediate subformula of ϕ. By the inductive hypothesis, there is a model satisfying all ϕ_i, and thus ϕ is satisfiable as well.
- $\phi \in \beta$. By definition, S' contains at least one pair $(\phi_i, [i,j])$, with ϕ_i immediate subformula of ϕ. By the inductive hypothesis, ϕ_i is satisfiable, and the satisfiability of ϕ follows.
- $\phi \in \gamma$. Let us consider $\phi \equiv \Box_r \psi$. By the inductive hypothesis, ψ is satisfiable at the reference interval $[j,x]$, that is, ψ is satisfiable at all intervals beginning with j, and thus there is a model satisfying ϕ (the other cases can be treated in the same way).
- $\phi \in \delta$. Let us consider $\phi \equiv \Diamond_r \psi$. By the inductive hypothesis, ψ is satisfiable with respect to the reference interval $[j,c]$ (or $[j, f(j)]$ is j is a variable), and thus there is a model satisfying ϕ (the other cases can be treated in the same way).

■

Theorem 9 (Soundness and Completeness) *ϕ is a satisfiable* PNL *formula iff there exists an open tableau* $(T, \mathcal{H})$ *for* ϕ.

Proof.

First suppose that ϕ is satisfiable. By Lemma 6, there is an Hintikka set $(S, \mathcal{H})$ containing $(\phi, [a,b])$. By exploiting Observation 7, it follows that a tableau for ϕ contains at least an open branch. The opposite direction is proved by contradiction. Suppose that there exists an open tableau $(T, \mathcal{H})$ for ϕ and ϕ is not satisfiable. By exploiting Observation 7, the open branch in $(T, \mathcal{H})$ corresponds to an Hintikka set $(S, \mathcal{H})$ containing $(\phi, [a,b])$. By Lemma 8, it follows that ϕ is satisfiable, which is a contradiction. ■

In order to make clearer how to build a model for a satisfiable PNL formula, consider the example of Figure 2. In this case, the leftmost open branch contains $(P, [b,x])$, $(\neg P, [b,c])$, and $(Q, [b,c])$. A model for the starting formula can be built on as follows: take a pair $(\mathbb{D}, <)$; assign arbitrary (distinct) values in the domain $\mathbb{D}$ to the constants a, b, c (respecting the ordering relation); assign P to all intervals beginning with b, except for $[b,c]$ over which Q must hold.

Current Relation	$\langle A \rangle$	$\langle \overline{A} \rangle$
e	$\{mi\}$	$\{m\}$
mi	$\{bi\}$	$\{f, e, fi\}$
m	$\{s, e, si\}$	$\{b\}$
bi	$\{bi\}$	$\{bi, mi, oi, si, di\}$
b	$\{b, o, m, fi, di\}$	$\{b\}$
fi	$\{mi\}$	$\{b\}$
f	$\{mi\}$	$\{d, s, o\}$
di	$\{b\}$	$\{bi\}$
d	$\{d, s, o\}$	$\{d, f, oi\}$
oi	$\{d, s, o\}$	$\{bi\}$
o	$\{d, f, oi\}$	$\{b\}$
si	$\{bi\}$	$\{m\}$
s	$\{d, f, oi\}$	$\{b\}$

Table 2: IA relations and the $\langle A \rangle / \langle \overline{A} \rangle$ fragment of HS.

PNL and HS

The tableau method for PNL can be easily adapted for the HS fragment provided with the two operators $\langle A \rangle$ and $\langle \overline{A} \rangle$ only, interpreted over linear structures $(\mathbb{D}, <)$. To obtain a terminating, sound, and complete method for this logic, it suffices to replace the symbol $\leq$ by the symbol $<$ in term constraints and to replace Definition 3 by the following one:

Definition 3′ A partial function $\rho : \mathcal{V} \mapsto \mathcal{T}$ is a *suitable substitution* if and only if $\mathcal{H}\rho \subseteq \mathcal{H}$.

Notice that the difference between the two logics is not trivial. As an example, the PNL formula $\phi_1 \equiv \Diamond_r P \wedge \Box_r \Box_r \neg P$ is not satisfiable, while the formula $\phi_2 \equiv \langle \mathrm{A} \rangle P \wedge [\mathrm{A}]\,[\mathrm{A}]\,\neg P$ is satisfiable in the $\langle A \rangle / \langle \overline{A} \rangle$ fragment of HS. The tableau for ϕ_1 includes $\mathcal{H}_1 = \{a \leq b, b \leq c, b \leq x, x \leq y\}$. The substitution ρ such that $\rho(x) = b$ and $\rho(y) = c$ is suitable, since the constraint $b \leq b$ must be eliminated from $\mathcal{H}_1 \rho$. The fact that x can assume the value b forces us to include the intervals of the form $[b, y]$ in the set of intervals at which the subformula $\Box_r \Box_r P$ must be evaluated. The tableau for ϕ_2 includes $\mathcal{H}_2 = \{a < b, b < c, b < x, x < y\}$. According to Definition 3′, the above substitution ρ is not suitable (the constraint $b < b$ is new). Indeed, in HS, as well as in all its fragments, $\langle \mathrm{A} \rangle$ and $[\mathrm{A}]$ range only over *non degenerate* intervals, and thus the subformula $[\mathrm{A}]\,[\mathrm{A}]\,P$ must not be evaluated at any interval which is met by the current one.

It is worth comparing such a fragment of HS with Allen's Interval Algebra (IA) (Allen & Ferguson 1994). It is well known that all the relations of IA can be captured in HS. This is not the case with the $\langle A \rangle / \langle \overline{A} \rangle$ fragment of HS which is not able to express full IA, but only part of it. As an example, the condition $[b,c]$ is *met* by $[a,b]$ can be expressed by the formula $\langle A \rangle \top$, where $\top$ stands for $P \vee \neg P$. However, there exist properties of intervals, which are not expressible in IA due to its syntactic restrictions, that can be specified in the $\langle A\rangle/\langle\overline{A}\rangle$ fragment. Consider, for instance, three intervals I_1, I_2, and I_3. The condition $I_1\{b\}I_2$ *or* $I_3\{m\}I_2$, which cannot be represented in IA, can be codified by the formula $\langle\overline{A}\rangle\langle\overline{A}\rangle\top \vee \langle\overline{A}\rangle\top$.

The fragment of IA which is captured by the $\langle A\rangle/\langle\overline{A}\rangle$ fragment of HS can be characterized as follows. Let us consider a formula of the form $\langle X_1\rangle \ldots \langle X_k\rangle\top$, where, for all $i = 1, \ldots, k$, $\langle X_i\rangle$ is either $\langle A\rangle$ or $\langle\overline{A}\rangle$. If $k = 0$, the formula is evaluated at (identifies) the starting interval $[a, b]$. If $k = 1$, the formula $\langle A\rangle\top$ (resp. $\langle\overline{A}\rangle\top$) is evaluated at a time interval $[b, c]$ (resp. $[c, a]$) which *is met by* (resp. *meets*) the interval $[a, b]$. Hence, the formulas $\langle A\rangle\top$ and $\langle\overline{A}\rangle\top$ respectively capture Allen's relations mi and m between the interval reached by applying $\langle A\rangle$ (resp. $\langle\overline{A}\rangle$) and the starting interval. If $k > 1$, let $\{r_1, \ldots, r_n\}$ be the set of Allen's relations that may possibly hold between the interval reached by applying the sequence $\langle X_1\rangle \ldots \langle X_{k-1}\rangle$ and the starting one. Allen's relations between the interval reached by applying the sequence $\langle X_1\rangle \ldots \langle X_k\rangle$ and the starting one can be determined by composing the set of relations $\{r_1, \ldots, r_n\}$ either with mi (if $X_k = \langle A\rangle$) or with m (if $X_k = \langle\overline{A}\rangle$). The whole set of composition rules is given in Table 2.

PNL and Compass Logics

There exist interesting relationships between PNL and a well-known class of logics for spatial reasoning, namely, compass logics. Compass logics have been originally proposed by Venema in (Venema 1990) and later studied by Marx and Reynolds in (Marx & Reynolds 1999). Full compass logics are provided with four operators $\Diamond^{\rightarrow}$, $\Diamond^{\leftarrow}$, $\Diamond^{\uparrow}$, and $\Diamond^{\downarrow}$. They are interpreted over pairs of linearly ordered domains $M = ((\mathbb{D}, <) \times (\mathbb{D}, <), V)$, where V is the valuation function, with the standard semantics for propositional formulas and the following semantics for modal formulas:

1. $M, (t, u) \Vdash \Diamond^{\rightarrow}\phi$ iff there exists $t' > t$ such that $M, (t', u) \Vdash \phi$;
2. $M, (t, u) \Vdash \Diamond^{\leftarrow}\phi$ iff there exists $t' < t$ such that $M, (t', u) \Vdash \phi$;
3. $M, (t, u) \Vdash \Diamond^{\uparrow}\phi$ iff there exists $u' > u$ such that $M, (t, u') \Vdash \phi$;
4. $M, (t, u) \Vdash \Diamond^{\downarrow}\phi$ iff there exists $u' < u$ such that $M, (t, u') \Vdash \phi$.

Full compass logic has been shown to be undecidable in (Marx & Reynolds 1999), where some variants and fragments of full compass logic are introduced.

It is possible to establish an interesting connection between time interval logics and compass logics. On the one hand, the interpretation domain must be restricted to the northwestern halfplane defined by the first diagonal. On the other hand, additional modal operators must be provided. Consider the case of a compass logic, interpreted over northwestern halfplane, provided with the operators $\Diamond^{\uparrow}$ and $\Diamond^{\leftarrow}$, and an additional pair of *projection* operators defined as follows:

- $M, (t, u) \Vdash J_1\phi$ iff $M, (t, t) \Vdash \phi$;

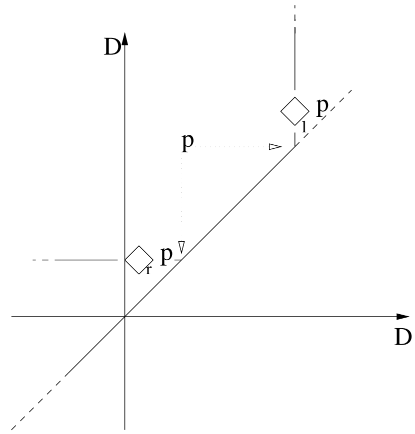

Figure 4: PNL as a fragment of compass logic

- $M, (t, u) \Vdash J_2\phi$ iff $M, (u, u) \Vdash \phi$.

We can define a suitable fragment of such a logic provided with a pair of operators $J_1\Diamond^{\uparrow} \equiv \Diamond_r$ and $J_2\Diamond^{\leftarrow} \equiv \Diamond_l$, graphically depicted in Figure 4, whose decidability immediately follows from the decidability result obtained for PNL.

Conclusions

In this work, we studied the propositional fragment of Neighborhood Logic (PNL). PNL does not involve any locality constraint, and it is expressive enough to capture interesting timing properties. We proved that the decidability problem for PNL is decidable by developing a terminating, sound, and complete tableau method. Furthermore, we established interesting connections between PNL, HS, and a class of spatial logics. In particular, we showed that PNL can be viewed as a variant of the $\langle A\rangle/\langle\overline{A}\rangle$ fragment of HS. We are currently investigating expressiveness and decidability issues for other, more expressive fragments of HS. In particular, we are studying the possibility of extending the proposed decision algorithm to a fragment of HS including the operators $\langle \mathrm{D}\rangle$ and $\langle\bar{D}\rangle$. Furthermore, we are considering alternative approaches to the problem of identifying decidable fragments of (undecidable) interval temporal logics. On the one hand, we are exploring syntactic characterizations of meaningful interval logics via guarded fragments of first-order logic; on the other hand, we are studying possible semantic restrictions of well-known interval modalities/structures.

References

Allen, J. F., and Ferguson, G. 1994. Actions and events in interval temporal logic. *Journal of Logic and Computation* 4(5):531–579.

Barua, R.; Roy, S.; and Chaochen, Z. 2000. Completeness of neighbourhood logic. *Journal of Logic and Computation* 10(2):271–295.

Bowman, H., and Thompson, S. 1998. A tableau method for interval temporal logic with projection. *Lecture Notes in Computer Science* 1397:108–134.

Chaochen, Z.; Hansen, M. R.; and Sestoft, P. 1993. Decidability results for duration calculus. In Enjalbert, P.; Finkel, A.; and Wagner, K. W., eds., *STACS*, volume 665 of *Lecture Notes in Computer Science*, 58–68. Springer.

Chaochen, Z.; Hoare, C. A. R.; and Ravn, A. P. 1991. A calculus of durations. *Information Processing Letters* 40(5):269–276.

Chetcuti-Serandio, N., and Fariñas del Cerro, L. 2000. A mixed decision method for duration calculus. *Journal of Logic and Computation* 10:877–895.

Chouchen, Z., and Hansen, M. R. 1998. An adequate first order interval logic. *Lecture Notes in Computer Science* 1536:584–608.

D'Agostino, M.; Gabbay, D.; Hähnle, R.; and Posegga, J., eds. 1999. *Handbook of Tableau Methods*. Kluwer, Dordrecht.

Halpern, J. Y., and Shoham, Y. 1991. A propositional modal logic of time intervals. *Journal of the ACM* 38(4):935–962.

Halpern, J. Y.; Manna, Z.; and Moszkowski, B. 1983. A hardware semantics based on temporal intervals. In Diaz, J., ed., *10th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 154 of *Lecture Notes in Computer Science*, 278–292. Barcelona, Spain: Springer.

Marx, M., and Reynolds, M. 1999. Undecidability of compass logic. *Journal of Logic and Computation* 9(6):897–914.

Moszkowski, B. 1983. *Reasoning about Digital Circuits*. Ph.D. Dissertation, Stanford University, Stanford, CA.

Sørensen, E.; Ravn, A.; and Rischel, H. 1990. Control program for a gas burner. part i: Informal requirements, process case study 1. Technical report, ProCoS Report ID/DTH EVS2.

van Benthem, J. 1991. *The Logic of Time (2nd Edition)*. Kluwer Academic Press.

Venema, Y. 1990. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic* 31(4):529–547.

Venema, Y. 1991. A modal logic for chopping intervals. *Journal of Logic and Computation* 1(4):453–476.