A tableau-based decision procedure for Right Propositional Neighborhood Logic

Davide Bresolin and Angelo Montanari

Department of Mathematics and Computer Science, University of Udine, Italy
E-mail: {bresolin|montana}@dimi.uniud.it

Abstract. Propositional interval temporal logics are quite expressive temporal logics that allow one to naturally express statements that refer to time intervals. Unfortunately, most such logics turned out to be (highly) undecidable. To get decidability, severe syntactic and/or semantic restrictions have been imposed to interval-based temporal logics that make it possible to reduce them to point-based ones. The problem of identifying expressive enough, yet decidable, new interval logics or fragments of existing ones which are *genuinely* interval-based is still largely unexplored. In this paper, we make one step in this direction by devising an original tableau-based decision procedure for the future fragment of Propositional Neighborhood Interval Temporal Logic, interpreted over natural numbers.

1 Introduction

Propositional interval temporal logics are quite expressive temporal logics that provide a natural framework for representing and reasoning about temporal properties in several areas of computer science. Among them, we mention Halpern and Shoham's Modal Logic of Time Intervals (HS) [6], Venema's CDT logic [10], Moszkowski's Propositional Interval Temporal Logic (PITL) [9], and Goranko, Montanari, and Sciavicco's Propositional Neighborhood Logic (PNL) [2] (an up-to-date survey of the field can be found in [4]). Unfortunately, most such logics turned out to be (highly) undecidable. To get decidability, severe syntactic and/or semantic restrictions have been imposed to make it possible to reduce them to point-based ones, thus leaving the problem of identifying expressive enough, yet decidable, new interval logics or fragments of existing ones which are *genuinely* interval-based largely unexplored. In this paper, we make one step in this direction by devising an original tableau-based decision procedure for the future fragment of PNL, interpreted over natural numbers.

Interval logics make it possible to express properties of *pairs* of time points (think of intervals as constructed out of points), rather than *single* time points. In most cases, this feature prevents one from the possibility of reducing interval-based temporal logics to point-based ones. However, there are a few exceptions where the logic satisfies suitable *syntactic* and/or *semantic restrictions*, and such a reduction can be defined, thus allowing one to benefit from the good computational properties of point-based logics [8].

 Proceedings of TABLEAUX 2005: 14th Conference on Automated Reasoning with Analytic Tableaux and Related Methods, Koblenz, Germany, September 2005.

One can get decidability by making a suitable choice of the interval modalities. This is the case with the $\langle B\rangle\langle\overline{B}\rangle$ and $\langle E\rangle\langle\overline{E}\rangle$ fragments of HS. Given a formula ϕ and an interval $[d_0, d_1]$, $\langle B\rangle\phi$ (resp. $\langle\overline{B}\rangle\phi$) holds over $[d_0, d_1]$ if ϕ holds over $[d_0, d_2]$, for some $d_0 \leq d_2 < d_1$ (resp. $d_1 < d_2$), and $\langle E\rangle\phi$ (resp. $\langle\overline{E}\rangle\phi$) holds over $[d_0, d_1]$ if ϕ holds over $[d_2, d_1]$, for some $d_0 < d_2 \leq d_1$ (resp. $d_2 < d_1$). Consider the case of $\langle B\rangle\langle\overline{B}\rangle$ (the case of $\langle E\rangle\langle\overline{E}\rangle$ is similar). As shown by Goranko et al. [4], the decidability of $\langle B\rangle\langle\overline{B}\rangle$ can be obtained by embedding it into the propositional temporal logic of linear time LTL[F,P] with temporal modalities F (sometime in the future) and P (sometime in the past). The formulae of $\langle B\rangle\langle\overline{B}\rangle$ are simply translated into formulae of LTL[F,P] by a mapping that replaces $\langle B\rangle$ by P and $\langle\overline{B}\rangle$ by F. LTL[F,P] has the finite model property and is decidable.

As an alternative, decidability can be achieved by constraining the classes of temporal structures over which the interval logic is interpreted. This is the case with the so-called Split Logics (SLs) investigated by Montanari et al. in [7]. SLs are propositional interval logics equipped with operators borrowed from HS and CDT, but interpreted over specific structures, called split structures. The distinctive feature of split structures is that every interval can be 'chopped' in at most one way. The decidability of various SLs has been proved by embedding them into first-order fragments of monadic second-order decidable theories of time granularity (which are proper extensions of the well-known monadic second-order theory of one successor S1S).

Another possibility is to constrain the relation between the truth value of a formula over an interval and its truth value over subintervals of that interval. As an example, one can constrain a propositional variable to be true over an interval if and only if it is true at its starting point (*locality*) or can constrain it to be true over an interval if and only it it is true over all its subintervals (*homogeneity*). A decidable fragment of PITL extended with quantification over propositional variables (QPITL) has been obtained by imposing the *locality* constraint. By exploiting such a constraint, decidability of QPITL can be proved by embedding it into quantified LTL. (In fact, as already noted by Venema, the locality assumption yields decidability even in the case of quite expressive interval logics such as HS and CDT.)

A major challenge in the area of interval temporal logics is thus to identify *genuinely* interval-based decidable logics, that is, logics which are not explicitly translated into point-based logics and not invoking locality or other semantic restrictions. In this paper, we propose a tableau-based decision procedure for the future fragment of (strict) Propositional Neighborhood Logic, that we call Right Propositional Neighborhood Logic (RPNL^- for short), interpreted over natural numbers. While various tableau methods have been developed for linear and branching time point-based temporal logics, not much work has been done on tableau methods for interval-based temporal logics. One reason for this disparity is that operators of interval temporal logics are in many respects more difficult to deal with [5]. As an example, there exist straightforward inductive definitions of the basic operators of point-based temporal logics, while inductive definitions of interval modalities turn out to be much more complex. In [3,5], Goranko et al. propose a general tableau method for CDT, interpreted over partial orders. It combines features of the classical tableau method for first-order logic with those of explicit tableau methods for modal logics with constraint label management, and it can be easily tailored to most propositional interval temporal logics proposed in the literature. However, it only provides a semi-decision procedure for unsatisfiability. By combining syntactic restrictions (future temporal operators) and semantic ones (the domain of natural numbers), we succeeded in devising a tableau-based decision procedure for RPNL$^-$. Unlike the case of the $\langle B\rangle\langle\overline{B}\rangle$ and $\langle E\rangle\langle\overline{E}\rangle$ fragments, in such a case we cannot abstract way from the left endpoint of intervals: there can be contradictory formulae that hold over intervals that have the same right endpoint but a different left one. The proposed tableau method partly resembles the tableau-based decision procedure for LTL [11]. However, while the latter takes advantage of the so-called fix-point definition of temporal operators which makes it possible to proceed by splitting every temporal formula into a (possibly empty) part related to the current state and a part related to the next state, and to completely forget the past, our method must also keep track of universal and (pending) existential requests coming from the past.

The rest of the paper is organized as follows. In Section 2, we introduce the syntax and semantics of RPNL$^-$. In Section 3, we give an intuitive account of the proposed method. In Section 4, we present our decision procedure, we prove its soundness and completeness, and we address complexity issues. In Section 5, we show our procedure at work on a simple example. Conclusions provide an assessment of the work and outline future research directions.

2 The Logic RPNL$^-$

In this section, we give syntax and semantics of RPNL$^-$ interpreted over natural numbers or over a prefix of them. To this end, we introduce some preliminary notions. Let $\mathbb{D} = \langle D, <\rangle$ be a strict linear order, isomorphic to the set $\mathbb{N}$ of natural numbers or to a prefix of them. A strict *interval* on $\mathbb{D}$ is an ordered pair $[d_i, d_j]$ such that $d_i, d_j \in D$ and $d_i < d_j$. The set of all strict intervals on $\mathbb{D}$ will be denoted by $\mathbb{I}(\mathbb{D})^-$ (notice that every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$ contains only a finite number of points). The pair $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle$ is called an *interval structure*.

RPNL$^-$ is a propositional interval temporal logic based on the neighborhood relation between intervals. Its formulae consist of a set AP of propositional letters $p, q, \ldots$, the Boolean connectives $\neg$ and $\vee$, and the future temporal operator $\langle A\rangle$. The other Boolean connectives, as well as the logical constants $\top$ (true) and $\bot$ (false), are defined in the usual way. Furthermore, we introduce the temporal operator $[A]$ as a shorthand for $\neg\langle A\rangle\neg$. The *formulae* of RPNL$^-$, denoted by $\varphi, \psi, \ldots$, are recursively defined by the following grammar:

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle A\rangle\varphi.$$

We denote by $|\varphi|$ the length of φ, that is, the number of symbols in φ (in the following, we shall use $|\ |$ to denote the cardinality of a set as well). Whenever there are no ambiguities, we call an RPNL$^-$ formula just a formula. A formula of the form $\langle A\rangle\psi$ or $\neg\langle A\rangle\psi$ is called a *temporal formula* (from now on, we identify $\neg\langle A\rangle\psi$ with $[A]\neg\psi$), while a formula devoid of temporal operators is called a *state formula* (state formulae are formulae of propositional logic).

A *model* for an RPNL$^-$ formula is a tuple $\mathbf{M}^- = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{V}\rangle$, where $\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle$ is an interval structure and $\mathcal{V} : \mathbb{I}(\mathbb{D})^- \longrightarrow 2^{AP}$ is the *valuation function* that assigns to every interval the set of propositional letters true on it.

Let $\mathbf{M}^- = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{V}\rangle$ be a model and let $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$. The semantics of RPNL$^-$ is defined recursively by the *satisfiability relation* $\Vdash$ as follows:

- for every propositional letter $p \in AP$, $\mathbf{M}^-, [d_i, d_j] \Vdash p$ iff $p \in \mathcal{V}([d_i, d_j])$;
- $\mathbf{M}^-, [d_i, d_j] \Vdash \neg\psi$ iff $\mathbf{M}^-, [d_i, d_j] \nVdash \psi$;
- $\mathbf{M}^-, [d_i, d_j] \Vdash \psi_1 \vee \psi_2$ iff $\mathbf{M}^-, [d_i, d_j] \Vdash \psi_1$, or $\mathbf{M}^-, [d_i, d_j] \Vdash \psi_2$;
- $\mathbf{M}^-, [d_i, d_j] \Vdash \langle A\rangle\psi$ iff $\exists d_k \in D$, $d_k > d_j$, such that $\mathbf{M}^-, [d_j, d_k] \Vdash \psi$.

Let d_0 be the initial element of D and let d_1 be its successor. The satisfiability problem for an RPNL$^-$ formula φ with respect to the *initial interval* $[d_0, d_1]$ of the structure is defined as follows: φ is satisfiable in a model $\mathbf{M}^- = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{V}\rangle$ if and only if $\mathbf{M}^-, [d_0, d_1] \Vdash \varphi$.

3 The proposed solution

In this section we give an intuitive account of the proposed tableau-based decision procedure for RPNL$^-$. To this end, we introduce the main features of a model building process that, given a formula φ to be checked for satisfiability, generates a model for it (if any) step by step. Such a process takes into consideration one element of the temporal domain at a time and, at each step, it progresses from one time point to the next one. For the moment, we completely ignore the problem of termination. In the next section, we shall show how to turn this process into an effective procedure.

Let $D = \{d_0, d_1, d_2, \ldots\}$ be the temporal domain, which we assumed to be isomorphic to $\mathbb{N}$ or to a prefix of it. The model building process begins from time point d_1 by considering the initial interval $[d_0, d_1]$. It associates with $[d_0, d_1]$ the set $C_{[d_0,d_1]}$ of all and only the formulae which hold over it.

Next, it moves from d_1 to its immediate successor d_2 and it takes into consideration the two intervals ending in d_2, namely, $[d_0, d_2]$ and $[d_1, d_2]$. As before, it associates with $[d_1, d_2]$ (resp. $[d_0, d_2]$) the set $C_{[d_1,d_2]}$ (resp. $C_{[d_0,d_2]}$) of all and only the formulae which hold over $[d_1, d_2]$ (resp. $[d_0, d_2]$). Since $[d_1, d_2]$ is a right neighbor of $[d_0, d_1]$, if $[A]\psi$ holds over $[d_0, d_1]$, then ψ must hold over $[d_1, d_2]$. Hence, for every formula $[A]\psi$ in $C_{[d_0,d_1]}$, it puts ψ in $C_{[d_1,d_2]}$. Moreover, since every interval which is a right neighbor of $[d_0, d_2]$ is also a right neighbor of $[d_1, d_2]$, and vice versa, for every formula ψ of the form $\langle A\rangle\xi$ or $[A]\xi$, ψ holds over $[d_0, d_2]$ if and only if it holds over $[d_1, d_2]$. Accordingly, it requires that $\psi \in C_{[d_0,d_2]}$ if and only if $\psi \in C_{[d_1,d_2]}$. Let us denote by $\mathrm{REQ}(d_2)$ the set of formulae of the form $\langle A\rangle\psi$ or $[A]\psi$ which hold over an interval ending in d_2 (by analogy, let $\mathrm{REQ}(d_1)$ be the set of formulae of the form $\langle A\rangle\psi$ or $[A]\psi$ which hold

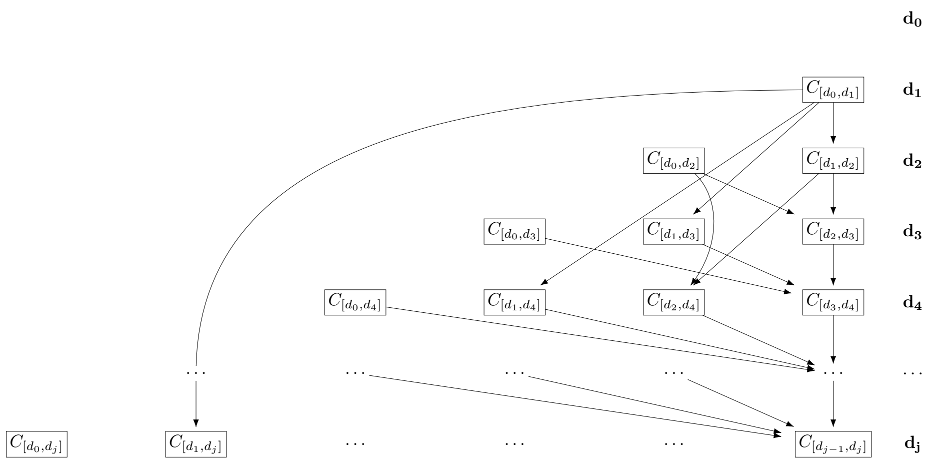

Fig. 1. The layered structure

over an interval ending in d_1, that is, the formulae $\langle A\rangle\psi$ or $[A]\psi$ which hold over $[d_0, d_1]$).

Next, the process moves from d_2 to its immediate successor d_3 and it takes into consideration the three intervals ending in d_3, namely, $[d_0, d_3]$, $[d_1, d_3]$, and $[d_2, d_3]$. As at the previous steps, for $i = 0, 1, 2$, it associates the set $C_{[d_i,d_3]}$ with $[d_i, d_3]$. Since $[d_1, d_3]$ is a right neighbor of $[d_0, d_1]$, for every formula $[A]\psi \in \mathrm{REQ}(d_1)$, $\psi \in C_{[d_1,d_3]}$. Moreover, $[d_2, d_3]$ is a right neighbor of both $[d_0, d_2]$ and $[d_1, d_2]$, and thus for every formula $[A]\psi \in \mathrm{REQ}(d_2)$, $\psi \in C_{[d_2,d_3]}$. Finally, for every formula ψ of the form $\langle A\rangle\xi$ or $[A]\xi$, we have that $\psi \in C_{[d_0,d_3]}$ if and only if $\psi \in C_{[d_1,d_3]}$ if and only if $\psi \in C_{[d_2,d_3]}$.

Next, the process moves from d_3 to its successor d_4 and it repeats the same operations; then it moves to the successor of d_4, and so on.

The layered structure generated by the process is graphically depicted in Figure 1. The first layer correspond to time point d_1, and for all $i > 1$, the i-th layer corresponds to time point d_i. If we associate with each node $C_{[d_i,d_j]}$ the corresponding interval $[d_i, d_j]$, we can interpret the set of edges as the neighborhood relation between pairs of intervals. As a general rule, given a time point $d_j \in D$, for every $d_i < d_j$, the set $C_{[d_i,d_j]}$ of all and only the formulae which hold over $[d_i, d_j]$ satisfies the following conditions:

- since $[d_i, d_j]$ is a right neighbor of every interval ending in d_i, for every formula $[A]\psi \in \mathrm{REQ}(d_i)$, $\psi \in C_{[d_i,d_j]}$;
- since every right neighbor of $[d_i, d_j]$ is also a right neighbor of all intervals $[d_k, d_j]$ belonging to layer d_j, for every formula ψ of the form $\langle A\rangle\xi$ or $[A]\xi$, $\psi \in C_{[d_i,d_j]}$ if and only if it belongs to all sets $C_{[d_k,d_j]}$ belonging to the layer.

As we shall show in the next section, the layers of the structure depicted in Figure 1 will become the (macro)nodes of the tableau for φ, whose edges will connect ordered pairs of nodes corresponding to consecutive layers.

4 A tableau-based decision procedure for RPNL$^-$

4.1 Basic notions

Let φ be an RPNL$^-$ formula to check for satisfiability and let AP be the set of its propositional variables. For the sake of brevity, we use $(A)\psi$ as a shorthand for both $\langle A\rangle\psi$ and $[A]\psi$.

Definition 1. *The* closure $\mathrm{CL}(\varphi)$ *of a formula* φ *is the set of all subformulae of* φ *and of their single negations (we identify* $\neg\neg\psi$ *with* ψ*).*

Lemma 1. *For every formula* φ*,* $|\,\mathrm{CL}(\varphi)| \leq 2 \cdot |\varphi|$.

Lemma 1 can be easily proved by induction on the structure of φ.

Definition 2. *The set of temporal requests of a formula* φ *is the set* $\mathrm{TF}(\varphi)$ *of all temporal formulae in* $\mathrm{CL}(\varphi)$*, that is,* $\mathrm{TF}(\varphi) = \{(A)\psi \in \mathrm{CL}(\varphi)\}$.

We are now ready to introduce the key notion of atom.

Definition 3. *Let* φ *be a formula of RPNL*$^-$*. A* φ-atom *is a pair* (A, C)*, with* $A \subseteq \mathrm{TF}(\varphi)$ *and* $C \subseteq \mathrm{CL}(\varphi)$*, such that:*
- *for every* $(A)\psi \in \mathrm{TF}(\varphi)$*, if* $(A)\psi \in A$*, then* $\neg(A)\psi \notin A$*;*
- *for every* $\psi \in \mathrm{CL}(\varphi)$*,* $\psi \in C$ *iff* $\neg\psi \notin C$*;*
- *for every* $\psi_1 \vee \psi_2 \in \mathrm{CL}(\varphi)$*,* $\psi_1 \vee \psi_2 \in C$ *iff* $\psi_1 \in C$ *or* $\psi_2 \in C$*;*
- *for every* $[A]\psi \in A$*,* $\psi \in C$*.*

Temporal formulae in A *are called* active requests*, while formulae in* C *are called* current formulae*.*

We denote the set of all φ-atoms by A_φ. We have that $|A_\varphi| \leq 2^{2|\varphi|}$. As we shall later show, the proposed tableau method identifies any interval $[d_i, d_j]$ with an atom (A, C), where A includes all universal formulae $[A]\psi \in \mathrm{REQ}(d_i)$ as well as those existential formulae $\langle A\rangle\psi \in \mathrm{REQ}(d_i)$ which do not hold over any interval $[d_i, d_k]$, with $d_k < d_j$, and C includes all formulae $\psi \in \mathrm{CL}(\varphi)$ which hold over $[d_i, d_j]$. Moreover, for all $[A]\psi \in A$, $\psi \in C$, while for any $\langle A\rangle\psi \in A$, it may happen that $\psi \in C$, but this is not necessarily the case.

Atoms are connected by the following binary relation.

Definition 4. *Let* X_φ *be a binary relation over* A_φ *such that, for every pair of atoms* $(A, C), (A', C') \in A_\varphi$*,* $(A, C)X_\varphi(A', C')$ *if (and only if):*
- $A' \subseteq A$*;*
- *for every* $[A]\psi \in A$*,* $[A]\psi \in A'$*;*
- *for every* $\langle A\rangle\psi \in A$*,* $\langle A\rangle\psi \in A'$ *iff* $\neg\psi \in C$*.*

In the next section we shall show that for any pair $i < j$, the relation X_φ connects the atom associated with the interval $[d_i, d_j]$ to the atom associated with the interval $[d_i, d_{j+1}]$.

4.2 Tableau construction, fulfilling paths, and satisfiability

To check the satisfiability of a formula φ, we build a graph, called a *tableau* for φ, whose nodes are *sets of atoms*, associated with a layer of Figure 1 (namely, with a set of intervals that end at the same point), and whose edges represent the relation between a layer and the next one, that is, between a point and its immediate successor. We shall take advantage of such a construction to reduce the problem of finding a model for φ to the problem of finding a path in the tableau that satisfies suitable properties.

Definition 5. *A* node *is a set of φ-atoms $\mathcal{N}$ such that, for any pair $(A, C), (A', C') \in \mathcal{N}$ and any $\langle A\rangle\psi \in \mathrm{TF}(\varphi)$, $\langle A\rangle\psi \in C \Leftrightarrow \langle A\rangle\psi \in C'$.*

We denote by $\mathcal{N}_\varphi$ the set of all nodes that can be built from A_φ, by $Init(\mathcal{N}_\varphi)$ the set of all *initial nodes*, that is, the set $\{\{(\emptyset, C)\} \in \mathcal{N}_\varphi\}$, and by $Fin(\mathcal{N}_\varphi)$ the set of all *final nodes*, that is, the set $\{\mathcal{N} \in \mathcal{N}_\varphi : \forall (A, C) \in \mathcal{N}, \forall \langle A\rangle\psi \in \mathrm{CL}(\varphi)(\langle A\rangle\psi \notin C)\}$. Furthermore, for any node $\mathcal{N}$, we denote by $\mathrm{REQ}(\mathcal{N})$ the set $\{\langle A\rangle\psi : \exists (A, C) \in \mathcal{N}(\langle A\rangle\psi \in C)\}$ (or, equivalently, $\{\langle A\rangle\psi : \forall (A, C) \in \mathcal{N}(\langle A\rangle\psi \in C)\}$). From Definition 5, it follows that $|\mathcal{N}_\varphi| \leq 2^{2^{2\cdot|\varphi|}}$.

Definition 6. *The* tableau *for a formula φ is a directed graph $T_\varphi = \langle \mathcal{N}_\varphi, E_\varphi\rangle$, where for any pair $\mathcal{N}, \mathcal{M} \in \mathcal{N}_\varphi$, $(\mathcal{N}, \mathcal{M}) \in E_\varphi$ if and only if $\mathcal{M} = \{(A_\mathcal{N}, C_\mathcal{N})\} \cup \mathcal{M}'_\mathcal{N}$, where*

1. *$(A_\mathcal{N}, C_\mathcal{N})$ is an atom such that $A_\mathcal{N} = \mathrm{REQ}(\mathcal{N})$;*
2. *for every $(A, C) \in \mathcal{N}$, there exists $(A', C') \in \mathcal{M}'_\mathcal{N}$ such that $(A, C) X_\varphi (A', C')$;*
3. *for every $(A', C') \in \mathcal{M}'_\mathcal{N}$, there exists $(A, C) \in \mathcal{N}$ such that $(A, C) X_\varphi (A', C')$.*

Definition 7. *Given a finite path $\pi = \mathcal{N}_1 \ldots \mathcal{N}_n$ in T_φ, an* atom path *in π is a sequence of atoms $(A_1, C_1), \ldots, (A_n, C_n)$ such that:*

- *for every $1 \leq i \leq n$, $(A_i, C_i) \in \mathcal{N}_i$;*
- *for every $1 \leq i < n$, $(A_i, C_i) X_\varphi (A_{i+1}, C_{i+1})$.*

Given a node $\mathcal{N}$ and an atom $(A, C) \in \mathcal{N}$, we say that the atom (A', C') is a descendant of (A, C) *if and only if there exists a node $\mathcal{M}$ such that $(A', C') \in \mathcal{M}$ and there exists a path π from $\mathcal{N}$ to $\mathcal{M}$ such that there is an atom path from (A, C) to (A', C') in π.*

The search for a φ model can by reduced to the search for a suitable path in T_φ.

Definition 8. *A* pre-model *for φ is a (finite or infinite) path $\pi = \mathcal{N}_1\mathcal{N}_2\mathcal{N}_3 \ldots$ in T_φ such that:*

- *$\mathcal{N}_1 (= \{(\emptyset, C)\}) \in Init(\mathcal{N}_\varphi)$ and $\varphi \in C$;*
- *if π is finite and $\mathcal{N}_n$ is the last node of π, then $\mathcal{N}_n \in Fin(\mathcal{N}_\varphi)$.*

Let $\mathbf{M}^- = \langle\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{V}\rangle$ be a model for φ. For every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$, we define an atom $(A_{[d_i,d_j]}, C_{[d_i,d_j]})$ such that:

$$A_{[d_i,d_j]} = \{[A]\psi \in \mathrm{REQ}(d_i)\} \cup \{\langle A\rangle\psi \in \mathrm{REQ}(d_i) : \forall d_i < d_l < d_j(\mathbf{M}^-, [d_i, d_l] \Vdash \neg\psi)\};$$
$$C_{[d_i,d_j]} = \{\psi \in \mathrm{CL}(\varphi) : \mathbf{M}^-, [d_i, d_j] \Vdash \psi\}.$$

For every $j \geq 1$ (and $j < |D|$, if $|D|$ is finite), we have that:

$$(A_{[d_i,d_j]}, C_{[d_i,d_j]}) X_\varphi (A_{[d_i,d_{j+1}]}, C_{[d_i,d_{j+1}]}).$$

For every $d_j \in D$, with $j \geq 1$, let $\mathcal{N}_j = \{(A_{[d_i,d_j]}, C_{[d_i,d_j]}) : i < j\}$ and $\pi_{\mathbf{M}^-} = \mathcal{N}_1\mathcal{N}_2\mathcal{N}_3 \ldots$. We have that for all $j \geq 1$, $\mathcal{N}_j$ is a node; moreover, $\pi_{\mathbf{M}^-}$ is a pre-model for φ.

Conversely, for every pre-model for φ $\pi = \mathcal{N}_1\mathcal{N}_2\mathcal{N}_3 \ldots$ in T_φ, we build an interval structure $\langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle$ and a set of interpretations $\mathbb{M}_\pi$ such that $\mathbf{M}^- = \langle \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{V} \rangle \in \mathbb{M}_\pi$ if and only if:

– $\langle D \setminus \{d_0\}, < \rangle$ and π are isomorphic;
– for every $p \in AP$ and $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$, $p \in \mathcal{V}([d_i, d_j])$ if and only if $\mu([d_i, d_j]) = (A, C)$ and $p \in C$, where $\mu : \mathbb{I}(\mathbb{D})^- \to A_\varphi$ maps every interval $[d_i, d_j]$ into an atom $\mu([d_i, d_j])$ in such a way that:
  1. $\mathcal{N}_1 = \{\mu([d_0, d_1])\}$;
  2. for every $d_i \in D$, with $d_1 < d_i$, $\mu([d_{i-1}, d_i]) = (A, C) \in \mathcal{N}_i$, with $A = \mathrm{REQ}(\mathcal{N}_{i-1})$;
  3. for every $d_i, d_j \in D$, with $d_i < d_{j-1}$, $\mu([d_i, d_j]) = (A, C) \in \mathcal{N}_j$, with $\mu([d_i, d_{j-1}]) X_\varphi (A, C)$;

Intuitively, μ assigns to every interval $[d_i, d_j]$ an atom belonging to the j-th node $\mathcal{N}_j$ in such a way that the interval relations depicted in Figure 1 are respected.

However, being π a pre-model for φ does not imply that there exists a model satisfying φ in $\mathbb{M}_\pi$, because formulae of the form $\langle A\rangle\psi$ are not necessarily satisfied by interpretations $\langle \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{V} \rangle \in \mathbb{M}_\pi$. To overcome this problem, we restrict our attention to fulfilling paths.

Definition 9. *Let $\pi = \mathcal{N}_1\mathcal{N}_2\mathcal{N}_3 \ldots$ be a pre-model for φ in T_φ. π is a* fulfilling path *for φ in T_φ if and only if, for every $\mathcal{N}_i$ and every $(A, C) \in \mathcal{N}_i$, if $\langle A\rangle\psi \in A$, then there exist $\mathcal{N}_j$, with $i \leq j$, and $(A', C') \in \mathcal{N}_j$, such that (A', C') is a descendant of (A, C) in π and $\psi \in C'$.*

Theorem 1. *For any formula φ, φ is satisfiable if and only if there exists a fulfilling path for φ in T_φ.*

Proof. Let φ be a satisfiable formula and $\mathbf{M}^-$ be a model for it. It is easy to show that the pre-model $\pi_{\mathbf{M}^-} = \mathcal{N}_1\mathcal{N}_2 \ldots$ is a fulfilling path for φ in T_φ. Let $(A_{[d_i,d_j]}, C_{[d_i,d_j]}) \in \mathcal{N}_j$ be an atom such that $\langle A\rangle\psi \in A_{[d_i,d_j]}$. By definition of $\pi_{\mathbf{M}^-}$, we have that $\langle A\rangle\psi \in \mathrm{REQ}(d_i)$ and, for all $d_i < d_l < d_j$, $\mathbf{M}^-, [d_i, d_l] \Vdash \neg\psi$. Since $\mathbf{M}^-$ is a model for φ, there must exist an interval $[d_i, d_k]$, with $d_k \geq d_j$, satisfying ψ. Hence, by definition, $\psi \in C_{[d_i,d_k]}$.

As for the converse, let $\pi = \mathcal{N}_1\mathcal{N}_2 \ldots$ be a fulfilling path for φ, and let $\mathbb{M}_\pi$ be the corresponding set of interpretations. We show that there exists a *fulfilling interpretation* $\mathbf{M}^- = \langle \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{V} \rangle \in \mathbb{M}_\pi$, that is, an interpretation such that, for every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$, if $\mu([d_i, d_j]) = (A, C)$ and $\langle A\rangle\psi \in C$, then there exists an interval $[d_j, d_k] \in \mathbb{I}(\mathbb{D})^-$ such that $\mu([d_j, d_k]) = (A', C')$, with $\psi \in C'$. We define such an interpretation by induction on the index of the nodes in π, that is, we first show how to fulfill the $\langle A\rangle$-formulae in $\text{REQ}(\mathcal{N}_1)$ (base case) and then we show how to fulfill those in $\text{REQ}(\mathcal{N}_j)$ provided that we have already fulfilled those in $\text{REQ}(\mathcal{N}_i)$ for $i = 1, \ldots, j-1$ (inductive step).

We begin from the initial node $\mathcal{N}_1 = \{(\emptyset, C)\}$. By the definition of μ, we have that $\mu([d_0, d_1]) = (\emptyset, C)$. Let $\langle A\rangle\psi_1, \langle A\rangle\psi_2, \ldots, \langle A\rangle\psi_n$ be the ordered list of the $\langle A\rangle$-formulae in $\text{REQ}(\mathcal{N}_1)$, if any (assume that they have been totally ordered on the basis of some syntactical criterion). We start with $\langle A\rangle\psi_1$. By Definition 6, there exists $(A_2, C_2) \in \mathcal{N}_2$ such that $\langle A\rangle\psi_1 \in A_2$. Since π is a fulfilling path for φ, by Definition 9 there exist $\mathcal{N}_k$, with $k \geq 2$, and $(A_k, C_k) \in \mathcal{N}_k$ such that (A_k, C_k) is a descendant of (A_2, C_2) in π and $\psi_1 \in C_k$. Let $(A_2, C_2) \ldots (A_k, C_k)$ be the atom path that leads from (A_2, C_2) to (A_k, C_k) in π. By putting $\mu([d_1, d_i]) = (A_i, C_i)$ for every $2 \leq i \leq k$, we meet the fulfilling requirement for $\langle A\rangle\psi_1$. Then we move to formula $\langle A\rangle\psi_2$. Two cases may arise: if there exists (A_i, C_i), with $2 \leq i \leq k$, such that $\psi_2 \in C_i$, then we are already done. Otherwise, we have that, for every (A_i, C_i), with $2 \leq i \leq k$, $\neg\psi_2 \in C_i$, and thus, by Definition 4, $\langle A\rangle\psi_2 \in A_i$ for $2 \leq i \leq k$. Since π is fulfilling, there exist $\mathcal{N}_h$, with $h > k$, and $(A_h, C_h) \in \mathcal{N}_h$ such that (A_h, C_h) is a descendant of (A_k, C_k) in π and $\psi_2 \in C_h$. Let $(A_k, C_k) \ldots (A_h, C_h)$ be the atom path that leads from (A_k, C_k) to (A_h, C_h) in π. As before, by putting $\mu([d_1, d_i]) = (A_i, C_i)$ for every $k+1 \leq i \leq h$, we meet the fulfilling requirement for $\langle A\rangle\psi_2$. By repeating this process for the remaining formulae $\langle A\rangle\psi_3, \ldots, \langle A\rangle\psi_n$, we meet the fulfilling requirement for all $\langle A\rangle$-formulae in $\text{REQ}(\mathcal{N}_1)$.

Consider now a node $\mathcal{N}_j \in \pi$, with $j > 1$, and assume that, for every $i < j$, the fulfilling requirements for all $\langle A\rangle$-formulae in $\text{REQ}(\mathcal{N}_i)$ have been met. We have that for any pair of atoms $(A, C), (A', C') \in \mathcal{N}_j$ and any $\langle A\rangle\psi \in \text{CL}(\varphi)$, $\langle A\rangle\psi \in C$ iff $\langle A\rangle\psi \in C'$ iff $\langle A\rangle\psi \in \text{REQ}(\mathcal{N}_j)$. Moreover, no one of the intervals over which μ has been already defined can fulfill any $\langle A\rangle\psi \in \text{REQ}(\mathcal{N}_j)$ since their left endpoints strictly precede d_j. We proceed as in the case of $\mathcal{N}_1$: we take the ordered list of $\langle A\rangle$-formulae in $\text{REQ}(\mathcal{N}_j)$ and we orderly fulfill them.

As for the intervals $[d_i, d_j]$ which are not involved in the fulfilling process, it suffices to define $\mu([d_i, d_j])$ so that it satisfies the general constraints on μ.

To complete the proof, it suffices to show that a fulfilling interpretation $\mathbf{M}^- = \langle\langle\mathbb{D}, \mathbb{I}(\mathbb{D})^-\rangle, \mathcal{V}\rangle$ is a model for φ. We show that for every $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$ and every $\psi \in \text{CL}(\varphi)$, $\psi \in C$, with $\mu([d_i, d_j]) = (A, C)$, if and only if $\mathbf{M}^-, [d_i, d_j] \Vdash \psi$. We prove this by induction on the structure of ψ.

- If ψ is the propositional letter p, then $p \in C \overset{\mathcal{V}\text{ def.}}{\Longleftrightarrow} p \in \mathcal{V}([d_i, d_j]) \Leftrightarrow \mathbf{M}^-, [d_i, d_j] \Vdash p$.
- If ψ is the formula $\neg\xi$, then $\neg\xi \in C \overset{\text{atom def.}}{\Longleftrightarrow} \xi \notin C \overset{\text{ind. hyp.}}{\Longleftrightarrow} \mathbf{M}^-, [d_i, d_j] \nVdash \xi \Leftrightarrow \mathbf{M}^-, [d_i, d_j] \Vdash \neg\xi$.
- If ψ is the formula $\xi_1 \vee \xi_2$, then $\xi_1 \vee \xi_2 \in C \overset{\text{atom def.}}{\Longleftrightarrow} \xi_1 \in C$ or $\xi_2 \in C \overset{\text{ind. hyp.}}{\Longleftrightarrow}$ $\mathbf{M}^-, [d_i, d_j] \Vdash \xi_1$ or $\mathbf{M}^-, [d_i, d_j] \Vdash \xi_2 \Leftrightarrow \mathbf{M}^-, [d_i, d_j] \Vdash \xi_1 \vee \xi_2$.

– Let ψ be the formula $\langle A\rangle\xi$. Suppose that $\langle A\rangle\xi \in C$. Since $\mathbf{M}^-$ is a fulfilling interpretation, there exists an interval $[d_j, d_k] \in \mathbb{I}(\mathbb{D})^-$ such that $\mu([d_j, d_k]) = (A', C')$ and $\xi \in C'$. By the inductive hypothesis, we have that $\mathbf{M}^-, [d_j, d_k] \Vdash \xi$, and thus $\mathbf{M}^-, [d_i, d_j] \Vdash \langle A\rangle\xi$. As for the opposite implication, we assume by contradiction that $\mathbf{M}^-, [d_i, d_j] \Vdash \langle A\rangle\xi$ and $\langle A\rangle\xi \notin C$. By atom definition, this implies that $\neg\langle A\rangle\xi = [A]\neg\xi \in C$. By definition of μ, we have that $\mu([d_j, d_k]) = (A', C')$ and $[A]\neg\xi \in A'$ for every $d_k > d_j$, and thus $\neg\xi \in C'$. By the inductive hypothesis, this implies that $\mathbf{M}^-, [d_j, d_k] \Vdash \neg\xi$ for every $d_k > d_j$, and thus $\mathbf{M}^-, [d_i, d_j] \Vdash [A]\neg\xi$, which contradicts the hypothesis that $\mathbf{M}^-, [d_i, d_j] \Vdash \langle A\rangle\xi$.

Since π is a fulfilling path for φ, $\varphi \in C_{[d_0,d_1]}$, and thus $\mathbf{M}^-, [d_0, d_1] \Vdash \varphi$. □

4.3 Maximal strongly connected components and decidability

In the previous section, we reduced the satisfiability problem for RPNL$^-$ to the problem of finding a fulfilling path in the tableau for the formula φ to check. However, fulfilling paths may be infinite, and thus we must show how to finitely establish their existence.

Let $\mathcal{C}$ be a subgraph of T_φ. We say that $\mathcal{C}$ is a *strongly connected component* (*SCC* for short) of T_φ if for any two different nodes $\mathcal{N}, \mathcal{M} \in \mathcal{C}$ there exists a path in $\mathcal{C}$ leading from $\mathcal{N}$ to $\mathcal{M}$.

Let $\pi = \mathcal{N}_1\mathcal{N}_2\mathcal{N}_3\ldots$ be an infinite fulfilling path in T_φ. Let $Inf(\pi)$ be the set of nodes that occurs infinitely often in π. It is not difficult to see that the subgraph defined by $Inf(\pi)$ is an SCC. We show that the search for a fulfilling path can be reduced to the search for a suitable SCC in T_φ. More precisely, we show that it is suffices to consider the *maximal strongly connected components* (*MSCC* for short) of T_φ, namely, SCC which are not properly contained in any other SCC.

Definition 10. *Let $\mathcal{C}$ be an SCC in T_φ. $\mathcal{C}$ is* self-fulfilling *if for every node $\mathcal{N} \in \mathcal{C}$, every atom $(A, C) \in \mathcal{N}$, and every formula $\langle A\rangle\psi \in A$, there exists a descendant (A', C') of (A, C) in $\mathcal{C}$ such that $\psi \in C'$.*

Let $\pi = \mathcal{N}_1\mathcal{N}_2\ldots$ be a fulfilling path for φ in T_φ. π starts from an initial node $\mathcal{N}_1$. If π is finite, it reaches a final node that belongs to a self-fulfilling SCC. If π is infinite, it reaches the SCC defined by $Inf(\pi)$ which is self-fulfilling as well.

The following lemma proves that being self-fulfilling is a monotone property.

Lemma 2. *Let $\mathcal{C}$ and $\mathcal{C}'$ be two non empty SCCs such that $\mathcal{C} \subseteq \mathcal{C}'$. If $\mathcal{C}$ is self-fulfilling, then $\mathcal{C}'$ is self-fulfilling too.*

Proof. Let $\mathcal{C} \subset \mathcal{C}'$. Suppose that there exist a node $\mathcal{N} \in \mathcal{C}'$ and an atom $(A, C) \in \mathcal{N}$ such that $\langle A\rangle\psi \in C$. Since $\mathcal{C}'$ is a SCC, there exists a path in $\mathcal{C}'$ that connects the node $\mathcal{N}$ to a node $\mathcal{M}$ in $\mathcal{C}$.

By Definition 6, there exists an atom $(A', C') \in \mathcal{M}$ which is a descendant of (A, C). Two cases may arise:

– if $\langle A\rangle\psi \notin A'$, then there exist a node $\mathcal{M}'$ in the path from $\mathcal{N}$ to $\mathcal{M}$ and a descendant (A'', C'') of (A, C) in $\mathcal{M}'$ with $\psi \in C''$;
– if $\langle A\rangle\psi \in A'$, since $\mathcal{C}$ is self-fulfilling, there exists (A'', C'') which is a descendant of (A', C') (and thus of (A, C)) with $\psi \in C''$.

In both cases, the formula $\langle A\rangle\psi$ gets fulfilled. □

On the basis of Lemma 2, we define a simple algorithm searching for fulfilling paths, that progressively removes from T_φ *useless* MSCCs, that is, MSCCs that cannot participate in a fulfilling path. We call *transient state* an MSCC consisting of a single node $\mathcal{N}$ devoid of self loops, i.e., such that the edge $(\mathcal{N}, \mathcal{N}) \notin E_\varphi$.

Definition 11. *Let $\mathcal{C}$ be an MSCC in T_φ. $\mathcal{C}$ is* useless *if one of the following conditions holds:*

1. *$\mathcal{C}$ is not reachable from any initial node;*
2. *$\mathcal{C}$ is a transient state which has no outgoing edges and is not a final node;*
3. *$\mathcal{C}$ has no outgoing edges and it is not self-fulfilling.*

Algorithm 1 *Satisfiability checking procedure.*

$\langle \mathcal{N}_0, E_0\rangle \leftarrow T_\varphi$
$i \leftarrow 0$
while $\langle \mathcal{N}_i, E_i\rangle$ is not empty and contains useless MSCC **do**
  let $\mathcal{C} = \langle \mathcal{N}, E\rangle$ be a useless MSCC
  $i \leftarrow i + 1$
  $\mathcal{N}_i \leftarrow \mathcal{N}_{i-1} \setminus \mathcal{N}$
  $E_i \leftarrow E_{i-1} \cap (\mathcal{N}_i \times \mathcal{N}_i)$
if $\exists \mathcal{N} \in Init(\mathcal{N}_i)$ such that $\mathcal{N} = \{(\emptyset, C)\}$ with $\varphi \in C$ **then**
  return *true*
else
  return *false*

Let us denote by $\langle \mathcal{N}^*, E^*\rangle$ the structure computed by Algorithm 1. The correctness of the algorithm is based on the following lemma.

Lemma 3. *π is a fulfilling path for φ in $\langle \mathcal{N}_i, E_i\rangle$ if and only if it is a fulfilling path for φ in $\langle \mathcal{N}_{i+1}, E_{i+1}\rangle$.*

Proof. A fulfilling path for φ starts from an initial node and reaches either a final node that belongs to a self-fulfilling SCC (finite path) or the self-fulfilling SCC defined by $Inf(\pi)$ (infinite path). By Lemma 2, we know that being self-fulfilling is monotone and thus by removing useless MSCC from $\langle \mathcal{N}_i, E_i\rangle$ we cannot remove any fulfilling path. □

Theorem 2. *For any formula φ, φ is satisfiable if and only if Algorithm 1 returns* true*.*

Proof. By Theorem 1, we have that φ is satisfiable if and only if there exists a fulfilling path in $T_\varphi = \langle \mathcal{N}_0, E_0\rangle$. By Lemma 3, this holds if and only if there exists a fulfilling path in $\langle \mathcal{N}^*, E^*\rangle$, that is, there exists a finite path $\pi = \mathcal{N}_1\mathcal{N}_2 \ldots \mathcal{N}_k$ in $\langle \mathcal{N}^*, E^*\rangle$ such that:

– $\mathcal{N}_1 = \{(\emptyset, C)\}$ is an initial node with $\varphi \in C$;
– $\mathcal{N}_k$ belongs to a self-fulfilling MSCC.

Since $\langle \mathcal{N}^*, E^* \rangle$ does not contain any useless MSCC, this is equivalent to the fact that there exists an initial node $\mathcal{N} = \{(\emptyset, C)\}$ in $\mathcal{N}^*$, with $\varphi \in C$. Thus, the algorithm correctly returns *true* if and only if φ is satisfiable. □

As for computational complexity, we have:
– $|\mathcal{N}_\varphi| \leq 2^{2^{2|\varphi|}}$ and thus $|T_\varphi| = 2^{2^{O(|\varphi|)}}$;
– the decomposition of T_φ into MSCCs can be done in time linear in $|T_\varphi|$;
– the algorithm takes time polynomial in $|A_\varphi| \cdot |T_\varphi|$.

Hence, checking the satisfiability of a formula φ has an overall time bound of $2^{2^{O(|\varphi|)}}$, that is, doubly exponential in the length of φ.

4.4 Improving the complexity: an EXPSPACE algorithm

In this section we describe an improvement of the proposed solution that exploits nondeterminism to find a fulfilling path in the tableau and thus to decide the satisfiability of a given formula φ, which is based on the definition of *ultimately periodic pre-model*.

Definition 12. *An infinite pre-model for φ $\pi = \mathcal{N}_1\mathcal{N}_2\ldots$ is* ultimately periodic*, with prefix l and period $p > 0$, if and only if, for all $i \geq l$, $\mathcal{N}_i = \mathcal{N}_{i+p}$.*

Theorem 3. *Let T_φ be the tableau for a formula φ. There exists an infinite fulfilling path in T_φ if and only if there exists an infinite fulfilling path that is ultimately periodic with prefix $l \leq |\mathcal{N}_\varphi|$ and period $p \leq |\mathcal{N}_\varphi|^2$.*

Proof. Let π be an infinite fulfilling path in T_φ. Consider now the SCC defined by $Inf(\pi)$ and the path σ connecting the initial node of π to it. An ultimately periodic fulfilling path satisfying the conditions of the theorem can be built as follows:

– let $\sigma = \mathcal{N}_1\mathcal{N}_2\ldots\mathcal{N}_n$. If $n > |\mathcal{N}_\varphi|$, take a path σ' from $\mathcal{N}_1$ to $\mathcal{N}_n$ of length $|\sigma'| \leq |\mathcal{N}_\varphi|$. Otherwise, take $\sigma' = \sigma$.
– Since $\mathcal{N}_n \in Inf(\pi)$, take a path σ_{loop} from $\mathcal{N}_n$ to $\mathcal{N}_n$. To guarantee the condition of fulfilling, we constrain σ_{loop} to contain all nodes in $Inf(\pi)$. By the definition of SCC, this loop exists and its length is less than or equal to $|\mathcal{N}_\varphi|^2$.

The infinite path $\pi' = \sigma'\sigma_{loop}\sigma_{loop}\sigma_{loop}\ldots$ is an ultimately periodic fulfilling path with prefix $l \leq |\mathcal{N}_\varphi|$ and period $p \leq |\mathcal{N}_\varphi|^2$. □

The following algorithm exploits Theorem 3 to nondeterministically guess a fulfilling path satisfying the formula.

First, the algorithm guesses two numbers $l \leq |\mathcal{N}_\varphi|$ and $p \leq |\mathcal{N}_\varphi|^2$. If $p = 0$, it searches for a finite pre-model of length l. Otherwise, it takes l as the prefix and p as the period of the ultimately periodic pre-model. Next, the algorithm guesses the first node $\mathcal{N}_1$ of the pre-model, taking $\mathcal{N}_1 = \{(\emptyset, C)\}$ with $\varphi \in C$. Subsequently, it guesses the next node $\mathcal{N}_2$, incrementing a counter and checking that the edge $(\mathcal{N}_1, \mathcal{N}_2)$ is in E_φ. The algorithm proceeds in this way, incrementing the counter for every node it adds to the pre-model.

When the counter reaches l, two cases are possible: if $p = 0$, then the current node is the last node of a finite pre-model, and the algorithm checks if it is a self-fulfilling final node. If $p > 0$, the algorithm must guess the period of the pre-model. To this end, it keeps in $\mathcal{N}_p$ the current node (that is, the first node of the period), and it guesses the other nodes of the period by adding a node and by incrementing the counter at every step. Furthermore, for every atom $(A, C) \in \mathcal{N}_p$ and for every formula $\langle A \rangle \psi \in A$, it checks if the formula gets fulfilled in the period. When the counter reaches p, it checks if there exists an edge from the current node to $\mathcal{N}_p$ and if all $\langle A \rangle$-formulae in $\mathcal{N}_p$ has been fulfilled.

By Theorem 3, it follows that the algorithm returns *true* if and only if φ is satisfiable. Furthermore, the algorithm only needs to store:

– the numbers l, p and a counter ranging over them;
– the initial node $\mathcal{N}_1$;
– the current node and the next guessed node of the pre-model;
– the first node of the period $\mathcal{N}_p$;
– the set of $\langle A \rangle$-formulae that needs to be fulfilled.

Since the counters are bounded by $|\mathcal{N}_\varphi|^2 = 2^{2^{O(|\varphi|)}}$, and since the number of nodes is bounded by $2^{2^{O(|\varphi|)}}$), the algorithm needs an amount of space which is exponential in the length of the formula.

Theorem 4. *The satisfiability problem for RPNL$^-$ is in EXPSPACE.*

5 The decision procedure at work

In this section we apply the proposed decision procedure to the satisfiable formula $\varphi = \langle A \rangle p \wedge [A]\langle A \rangle p$ (which does not admit finite models). We show only a portion of the entire tableau, which is sufficiently large to include a fulfilling path for φ and thus to prove that φ is satisfiable.

Let $\mathbf{M}^- = \langle \langle \mathbb{D}, \mathbb{I}(\mathbb{D})^- \rangle, \mathcal{V} \rangle$ be a model that satisfies φ. Since $\mathbf{M}^-, [d_0, d_1] \Vdash \varphi$ we have that $\mathbf{M}^-, [d_0, d_1] \Vdash [A]\langle A \rangle p$ and $\mathbf{M}^-, [d_0, d_1] \Vdash \langle A \rangle p$. It is easy to see that this implies that, for every interval $[d_i, d_j] \in \mathbb{I}(\mathbb{D})^-$, $\mathbf{M}^-, [d_i, d_j] \Vdash [A]\langle A \rangle p$ and $\mathbf{M}^-, [d_i, d_j] \Vdash \langle A \rangle p$. For this reason, we can consider (when searching for a fulfilling path for φ) only atoms obtained by combining one the following set of active requests with one of the following set of current formulae:

$$\begin{array}{ll} A_0 = \emptyset; & C_0 = \{\varphi, [A]\langle A \rangle p, \langle A \rangle p, p\}; \\ A_1 = \{[A]\langle A \rangle p\}; & C_1 = \{\varphi, [A]\langle A \rangle p, \langle A \rangle p, \neg p\}. \\ A_2 = \{\langle A \rangle p, [A]\langle A \rangle p\}; & \end{array}$$

As an example, consider the initial node $\mathcal{N}_1 = \{(\emptyset, C_0)\}$. Figure 2 depicts a portion of T_φ which is reachable from $\mathcal{N}_1$. An edge reaching a boxed set of nodes means that there is an edge reaching every node in the box, while an edge leaving from a box means that there is an edge leaving from every node in the box.

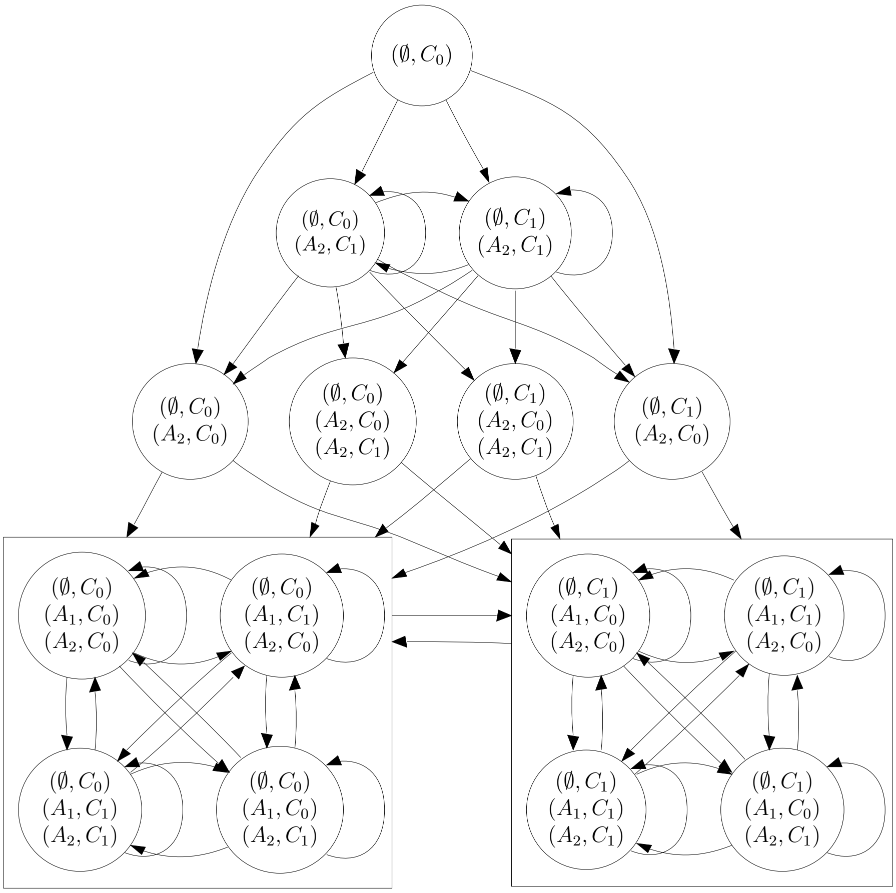

Fig. 2. A portion of the tableau for $\langle A\rangle p \wedge [A]\langle A\rangle p$.

The only atoms with $\langle A\rangle$-formulae in their set of active requests are (A_2, C_0) and (A_2, C_1), since $\langle A\rangle p \in A_2$. The atom (A_2, C_0) immediately fulfills $\langle A\rangle p$, since $p \in C_0$. The atom (A_2, C_1) does not fulfill $\langle A\rangle p$, but we have that $(A_2, C_1) X_\varphi$ (A_2, C_0).

Consider now the two boxed set of nodes of Figure 2. They define an SCC $\mathcal{C}$ such that every node in $\mathcal{C}$ that contains the atom (A_2, C_1) has a descendant that contains the atom (A_2, C_0), which fulfills the request $\langle A\rangle p$. This means that $\mathcal{C}$ is a self-fulfilling SCC which is reachable from the initial node $\mathcal{N}_1$ and thus our decision procedure correctly concludes that the formula φ is satisfiable.

6 Conclusions and further work

In this paper we proposed an original tableau-based decision procedure for $RPNL^-$ interpreted over $\mathbb{N}$ (or over a prefix of it). We also provided an EXPSPACE upper bound to the complexity of the satisfiability problem for $RPNL^-$. We do not know yet whether it is EXPSPACE-complete or not. As for possible extensions of the proposed method, we generalized it to branching time (where every timeline is isomorphic to $\mathbb{N}$) [1]. To this end, the definition of T_φ must be modified to take into account that every point of the tree (and, thus, every node) may have many immediate successors. Furthermore, the decision algorithm, instead of searching for fulfilling paths, has to test whether, for every formula $\langle A \rangle \psi$ that belongs to the set C of current formulae of an atom (A, C) of a node $\mathcal{N}$, there exists a successor $\mathcal{M}$ of $\mathcal{N}$ that fulfills its request. If this is the case, a model satisfying the formula can be obtained simply by taking the tree obtained through the unfolding of the tableau, starting from the initial node. The extension of the method to full PNL^- turns out to be more difficult. In such a case the definition of nodes and edges of the tableau, as well as the definition of fulfilling path, must be revised to take into account past operators.

References

1. D. Bresolin, A. Montanari, A tableau-based decision procedure for Right Propositional Neighborhood Logic, Research Report 02, Dipartimento di Matematica e Informatica, Università di Udine, Italy, 2005.
2. V. Goranko, A. Montanari, G. Sciavicco, Propositional interval neighborhood temporal logics, Journal of Universal Computer Science 9 (9) (2003) 1137–1167.
3. V. Goranko, A. Montanari, G. Sciavicco, A general tableau method for propositional interval temporal logics, in: Proc. of the International Conference TABLEAUX 2003, Vol. 2796 of LNAI, Springer, 2003, pp. 102–116.
4. V. Goranko, A. Montanari, G. Sciavicco, A road map of interval temporal logics and duration calculi, Journal of Applied Non-Classical Logics 14 (1-2) (2004) 9–54.
5. V. Goranko, A. Montanari, G. Sciavicco, P. Sala, A general tableau method for propositional interval temporal logics: theory and implementation, Journal of Applied Logic, 2005 (to appear).
6. J. Halpern, Y. Shoham, A propositional modal logic of time intervals, Journal of the ACM 38 (4) (1991) 935–962.
7. A. Montanari, G. Sciavicco, N. Vitacolonna, Decidability of interval temporal logics over split-frames via granularity, in: Proc. of the European Conference on Logic in Artificial Intelligence 2002, LNAI 2424, Springer, 2002, 259–270.
8. A. Montanari, Propositional interval temporal logics: some promising paths, Proc. of the 12th International Symposium on Temporal Representation and Reasoning (TIME), IEEE Computer Society Press, 2005, 201-203.
9. B. Moszkowski, Reasoning about digital circuits, Ph.D. thesis, Dept. of Computer Science, Stanford University, Tech. Rep. STAN-CS-83-970, Stanford, CA, 1983.
10. Y. Venema, A modal logic for chopping intervals, Journal of Logic and Computation 1 (4) (1991) 453–476.
11. P. Wolper, The tableau method for temporal logic: An overview, Logique et Analyse 28 (1985) 119–136.