

Linear-time Temporal Logics with Presburger Constraints: an Overview

Stéphane Demri

▶ **To cite this version:**

Stéphane Demri. Linear-time Temporal Logics with Presburger Constraints: an Overview. Journal of Applied Non-Classical Logics, 2006, 16 (3-4), pp.311-347. 10.3166/jancl.16.311-347 . hal-03189932

HAL Id: hal-03189932
https://hal.science/hal-03189932

Submitted on 5 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Linear-time Temporal Logics with Presburger Constraints: an Overview [1]

Stéphane Demri

LSV, CNRS & INRIA Futurs projet SECSI & ENS Cachan
61, av. Pdt. Wilson, 94235 Cachan Cedex, France
email: demri@lsv.ens-cachan.fr

RÉSUMÉ. *Ce papier présente un panorama de logiques temporelles du temps linéaire avec contraintes de Presburger dont les modèles sont des séquences de tuples d'entiers. De tels langages formels permettent de spécifier et vérifier certaines propriétés des systèmes à compteurs. Le papier rappelle le cadre général de LTL avec domaines concrets et présente les principaux résultats de décidabilité et complexité des fragments de LTL avec contraintes de Presburger. Une comparaison avec des formalismes alternatifs ou voisins est aussi ébauchée.*

ABSTRACT. *We present an overview of linear-time temporal logics with Presburger constraints whose models are sequences of tuples of integers. Such formal specification languages are well-designed to specify and verify systems that can be modelled with counter systems. The paper recalls the general framework of LTL over concrete domains and presents the main decidability and complexity results related to fragments of Presburger LTL. Related formalisms are also briefly presented.*

MOTS-CLÉS : *logique temporelle, arithmétique de Presburger, automate de Büchi, complexité*

KEYWORDS: *temporal logic, Presburger arithmetic, Büchi automaton, computational complexity*

1. Invited talk given at the INTAS worskhop held in Olsztyn, Poland, june 2006, for the INTAS project "Algebraic and Deduction Methods in NonClassical Logics and their Applications to Computer Science". Work partially supported by the ACI "Sécurité et Informatique" CORTOS.

1. Introduction

Temporal logics. Temporal logics are standard formal specification languages to verify and specify complex systems/programs, see e.g. (Pnueli, 1977; Bérard *et al.*, 2001). Such languages allow the user to specify by means of temporal formulae the properties which a program must satisfy and to verify that the specifications are met. Temporal logics are well-studied formalisms to specify the behavior of finite-state systems and the computational complexity of the model-checking problems is nowadays well-known, see e.g. a survey in (Schnoebelen, 2003). However, it is known that many systems such as communication protocols have infinitely many configurations and usually the techniques for the finite case cannot be applied directly. Unfortunately, for numerous classes of infinite-state systems, the model-checking problem for the linear-time temporal logic LTL can be easily shown to be undecidable (counter automata, hybrid automata and more general constraint automata (Revesz, 2002, Chapter 6)). Actually, simpler problems such as reachability are already undecidable. Nevertheless, remarkable classes of infinite-state systems admit decidable model-checking problems, such as timed automata (Alur *et al.*, 1994a) and subclasses of counter automata (Ibarra, 1978; Boigelot, 1998; Boigelot *et al.*, 2002; Leroux *et al.*, 2005). More specifically, fragments of LTL with Presburger constraints have been shown decidable over appropriate counter automata in (Čerāns, 1994; Comon *et al.*, 2000; Demri *et al.*, 2003). We recall that Presburger arithmetic (Presburger, 1929) is the first-order theory of integers with addition (but without multiplication) and, it can be shown to be decidable by quantifiers elimination.

Content of the paper. The paper presents an overview of linear-time temporal logics over constraint systems and more precisely of fragments of Presburger arithmetic. Such logical formalisms allow to express effects of basic instructions in programs. For instance, a formula of the form $\mathsf{F}(x = \mathsf{X}y)$ states that eventually the value of the variable x is going to be equal to the next value of y. Decidability and complexity issues for satisfiability and model-checking about constrained automata (viewed as models for systems/programs) are discussed along the paper and we provide a generous amount of examples. Hence, we present the main framework for linear-time temporal logics over constraints systems, extending the standard logic LTL (Sistla *et al.*, 1985). We explain why symbolic representation of concrete models of the logics is needed and how symbolic models can be recognized by automata accepting ω-words languages. Most of our decidability results are based on the automata-based approach from (Vardi *et al.*, 1994) but we cannot apply this method directly since the models of the logics are ω-sequences of valuations (providing an infinite alphabet incompatible with standard Büchi automata). Decidability and complexity results for fragments and for extensions of Presburger LTL are thoroughly presented whereas some undecidable logics are precisely placed in the arithmetical and analytical hierarchies. Finally, we illustrate the main concepts on simple examples and the results presented herein in details are compared to alternative formalisms.

Most of the results stated in the paper are presented in details in other papers. We have taken a special care to make uniform the notations and to provide a great amount of relevant bibliographical references.

Plan of the paper. The structure of the paper is the following. Section 2 introduces the framework of linear-time temporal logics over constraint systems and presents fragments of Presburger arithmetic useful in the rest of the paper. In Section 3, we present various fragments of LTL over constraint systems (some of them based on fragments of Presburger arithmetic) whose satisfiability problem is undecidable. In most cases, runs of Minsky machines can be simulated. The main proof technique to show decidability is presented in Section 4 where the relationships between satisfiability and symbolic models are clearly stated. This unifies known decidability proofs. As explained in Section 4, it still remains to check properties assumed in the main theorem. Decidability results are stated in Section 5 including those for which the constraint system has integer periodicity constraints or for which the class of symbolic models for a given formula is not necessarily ω-regular. A quite remarkable result is that model-checking one-counter automata with LTL over quantifier-free Presburger arithmetic is PSPACE-complete. Decidability and complexity results in presence of the freeze operator are presented in Section 6. In particular the presence of this powerful binding mechanism with the poor constraint system $\langle \mathbb{N}, = \rangle$ already produces highly undecidable problems. On the decidability side, effective procedures can be designed with one rigid variable and finite models, or with integer periodicity constraints, to quote a few examples. Related formalisms are compared with the ones presented in the paper in Section 7. The main goal of this section is to provide the reader useful bibliographical references to other works related to (temporal) logics and Presburger constraints.

2. LTL over constraint systems

Linear-time temporal logic LTL equipped with "next-time" operator X, "until" operator U and their past-time counterparts is known to be equivalent to first-order theory of successors (Kamp, 1968). Satisfiability and model-checking problems for LTL (even with past-time operators) are also known to be PSPACE-complete (Sistla *et al.*, 1985). In spite of these nice features, it is worth recalling that a propositional variable p only represents a property of the current state of the system. For instance, p may hold true whenever the value of the variable x is greater than the value of the variable y after running the current instruction. A more satisfying solution is to include in the logical language the possibility to express directly constraints between variables of the program, whence giving up the standard abstraction made with propositional variables. When the variables are typed, they may be interpreted in some specific domain like integers, real numbers, strings and so on. Hence, a proposition like "x is greater than the next value of y" can be encoded in such extended temporal logics by $x < \mathsf{X}y$ but this time the models are made of structured states. This means that each state comes with a valuation for variables. Hence, the basic idea behind the design of

temporal logics over constraint systems is to refine the language of atomic formulae and to allow the possibility to compare values of variables at successive positions of the execution of programs, see e.g., in (Comon *et al.*, 2000) a representative example of such logics.

2.1. *How to refine LTL with the help of constraint systems*

Let $\mathrm{VAR} = \{x_0, x_1, \ldots\}$ be a countably infinite set of variables. A constraint system is a pair $\mathcal{D} = \langle D, (R_\alpha)_{\alpha \in I} \rangle$ where D is a specific domain of interpretation for variables and $(R_\alpha)_{\alpha \in I}$ is a countable family of relations on the elements of D. So, a constraint system is simply a relational structure. An atomic $\mathcal{D}$-constraint is a term of the form $\mathsf{R}(x_1, \ldots, x_n)$ where R is interpreted as a relation of the domain and n is the arity of this relation. A D-valuation is a function $v : \mathrm{VAR} \to D$ that assigns to every variable a value in D. A constraint is satisfied by a D-valuation v, denoted by $v \models_{\mathcal{D}} \mathsf{R}(x_1, \ldots, x_n)$, if $(v(x_1), \ldots, v(x_n)) \in R$, R being the relation in $\mathcal{D}$ associated to the symbol R.

The consistency problem consists in checking whether there is a valuation satisfying every atomic $\mathcal{D}$-constraint from a finite set. Similarly, the maximal consistency problem checks for maximal consistency with respect to a finite set of variables and a finite set of relations. For instance, the maximal consistency problem for the constraint system $\langle \mathbb{R}, <, = \rangle$ is NLOGSPACE-complete since it amounts to detect cycles in a finite graph. The implication problem consists in checking whether every valuation satisfying a finite set X of atomic $\mathcal{D}$-constraints satisfies an atomic $\mathcal{D}$-constraint A (written $X \models_{\mathcal{D}} A$). For instance, with $\mathcal{D} = \langle \mathbb{R}, < \rangle$, we have $\{x < y, y < z\} \models_{\mathcal{D}} \{x < z\}$.

The logic $\mathrm{CLTL}(\mathcal{D})$ is defined as an extension of LTL where the propositional variables are refined by atomic $\mathcal{D}$-constraints over terms (called atomic temporal $\mathcal{D}$-constraints). A term is defined as a variable x_j prefixed by a certain number i of X "next" symbol X and is denoted by $\mathsf{X}^i x_j$ (its encoding requires $\mathcal{O}(i + log\ j)$ bits). The term $\mathsf{X}^i x_j$ is interpreted as the value of x_j at the ith next state. The "next" symbol X is overloaded in this paper since X is also used below as a logical temporal operator. The $\mathrm{CLTL}(\mathcal{D})$ formulae are defined as follows :

$$\phi ::= \mathsf{R}(\mathsf{X}^{l_1} x_{j_1}, \ldots, \mathsf{X}^{l_n} x_{j_n}) \mid \phi \wedge \phi \mid \neg\phi \mid \mathsf{X}\phi \mid \phi \mathsf{U} \phi.$$

The symbols X and U are respectively the classical operators next-time and until from LTL. We use the notations $\mathsf{F}\phi$ and $\mathsf{G}\phi$ as the abbreviations for $\top \mathsf{U} \phi$ and $\neg \mathsf{F} \neg \phi$. A one-step constraint is an atomic formula of the form $\mathsf{R}(\mathsf{X}^{l_1} x_{j_1}, \ldots, \mathsf{X}^{l_n} x_{j_n})$ such that $l_1, \ldots, l_n \leq 1$. Given a $\mathrm{CLTL}(\mathcal{D})$ formula ϕ we define its X-length $|\phi|_{\mathsf{X}}$ as the maximal number i such that a term of the form $\mathsf{X}^i x$ occurs in ϕ. Intuitively, the X-length defines the size of a frame of consecutive states that can be compared. The models of $\mathrm{CLTL}(\mathcal{D})$ are sequences of D-valuations $\sigma : \mathbb{N} \to (\mathrm{VAR} \to D)$ and the satisfaction relation is defined as for LTL except at the atomic level :

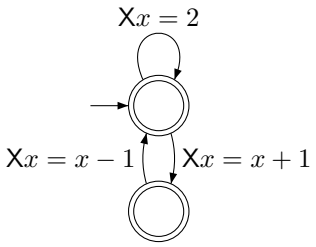


Figure 1. *A 1-variable* $\langle \mathbb{N}, +1, = 2 \rangle$*-automaton* $\mathcal{A}_0$

– $\sigma, i \models \mathsf{R}(\mathsf{X}^{l_1} x_{j_1}, \ldots, \mathsf{X}^{l_n} x_{j_n})$ iff $\langle \sigma(i + l_1)(x_{j_1}), \ldots, \sigma(i + l_n)(x_{j_n})) \rangle \in R$,

– $\sigma, i \models \phi \wedge \phi'$ iff $\sigma, i \models \phi$ and $\sigma, i \models \phi'$,

– $\sigma, i \models \neg\phi$ iff $\sigma, i \not\models \phi$,

– $\sigma, i \models \mathsf{X}\phi$ iff $\sigma, i + 1 \models \phi$,

– $\sigma, i \models \phi \mathsf{U} \phi'$ iff there is $j \geq i$ such that $\sigma, j \models \phi'$ and for every $i \leq l < j$, we have $\sigma, l \models \phi$.

As usual, a formula $\phi \in \mathrm{CLTL}(\mathcal{D})$ is satisfiable whenever there exists a model σ such that $\sigma, 0 \models \phi$. We write $\mathrm{CLTL}^l_k(\mathcal{D})$ to denote the restriction of $\mathrm{CLTL}(\mathcal{D})$ to formulae with at most k variables and X-length less or equal to l. Standard LTL (Sistla *et al.*, 1985) can be viewed as $\mathrm{CLTL}(\{\top, \bot\}, True)$ where $True = \{\top\}$.

Lemma 2.1. Let $\mathcal{D}$ be a constraint system with equality predicate. There is a logarithmic space reduction from $\mathrm{CLTL}(\mathcal{D})$ satisfiability into $\mathrm{CLTL}^1_\omega(\mathcal{D})$ satisfiability.

The proof of Lemma 2.1 is done by renaming terms and requires an unbounded amount of variables in $\mathrm{CLTL}^1_\omega(\mathcal{D})$.

We define the model-checking problem for the class of $\mathcal{D}$-automata that are Büchi automata with transitions labeled with one-step constraints. A k-variable $\mathcal{D}$-automaton $\mathcal{A}$ is a structure $\langle Q, \delta, I, F \rangle$ such that Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states and δ is a subset of $Q \times \Sigma \times Q$ where Σ is a finite subset of $1SC_k$, the set of Boolean combinations of one-step constraints built over the set of variables $\{x_1, \ldots, x_k\}$. We use the notation $q \xrightarrow{c} q'$ as an abbreviation for $\langle q, c, q' \rangle \in \delta$. When the constraint system is a fragment of Presburger arithmetic, $\mathcal{D}$-automata form a special class of counter automata. Figure 1 contains a graphical representation of a 1-variable $\langle \mathbb{N}, +1, = 2 \rangle$-automaton where "$= 2$" is the unary predicate interpreted by $\{2\}$ which corresponds to a test-for-two.

The language accepted by a $\mathcal{D}$-automaton is denoted by $\mathrm{L}(\mathcal{A})$. A $\mathrm{CLTL}(\mathcal{D})$ model σ is said to realize an ω-sequence of $\mathrm{CLTL}(\mathcal{D})$ formulae $\phi_0 \phi_1 \ldots$ iff for every $i \geq 0$, we have $\sigma, i \models \phi_i$. We write $\mathrm{L}^{\mathcal{D}}(\mathcal{A})$ to denote the class of $\mathrm{CLTL}(\mathcal{D})$ models that realize some ω-word in $\mathrm{L}(\mathcal{A})$.

The model-checking problem [1] for $\mathrm{CLTL}(\mathcal{D})$ is defined as follows :

input : a $\mathcal{D}$-automaton $\mathcal{A}$ and a $\mathrm{CLTL}(\mathcal{D})$ formula ϕ,

output : 1 iff there is a model σ in $\mathrm{L}^{\mathcal{D}}(\mathcal{A})$ such that $\sigma, 0 \models \phi$ (denoted by $\mathcal{A} \models_{\exists} \phi$).

As expected, this amounts to check that there is a computation of $\mathcal{A}$ that satisfies the property/formula ϕ. A universal version of the problem (more used in verification) can be also defined analogously (we then use $\mathcal{A} \models_{\forall} \phi$). For the restriction of the model-checking to $\mathrm{CLTL}^{l}_{k}(\mathcal{D})$, ϕ belongs to $\mathrm{CLTL}^{l}_{k}(\mathcal{D})$ and $\mathcal{A}$ is a k-variable $\mathcal{D}$-automaton.

For all non-trivial constraint systems presented in the paper, satisfiability can be shown equivalent to the existential version of the model-checking problem and validity can be shown equivalent to its universal version. For instance, $\mathcal{A}_0 \models_{\exists} (x = 0) \wedge \mathsf{GF}(x = 0)$ with $\mathcal{A}_0$ defined in Figure 1. Indeed, $((\mathsf{X}x = x + 1) \cdot (\mathsf{X}x = x - 1))^{\omega} \in \mathrm{L}(\mathcal{A}_0)$ and $(0 \cdot 1)^{\omega}$ realizes this ω-word.

2.2. *Fragments of Presburger LTL*

2.2.1. *Presburger arithmetic*

Presburger arithmetic (PA) is the first-order theory of $\langle \mathbb{Z}, + \rangle$ of integers with addition (Presburger, 1929). This theory is decidable in triple exponential-time (Fischer *et al.*, 1974) and many fragments can be decided even with a lower complexity, see below. In (Ginsburg *et al.*, 1966), it is shown that PA defines precisely semilinear sets.

Given a Presburger formula $A(x_1, \ldots, x_n)$ with free variables in $\vec{x} = \langle x_1, \ldots, x_n \rangle$, and $\vec{a} = \langle a_1, \ldots, a_n \rangle \in \mathbb{Z}^n$, the truth of $A(x_1, \ldots, x_n)$ with respect to the interpretation $\vec{a}$ is denoted by $\vec{a} \models A(\vec{x})$. We write $sol(A(\vec{x}))$ to denote the elements $\vec{a} \in \mathbb{Z}^n$ satisfying the formula $A(\vec{x})$. In the sequel, any (possibly infinite) fragment FPA of PA (understood as a subset of PA) defines implicitely a constraint system

$$\mathcal{D}_{\mathrm{FPA}} = \langle \mathbb{Z}, (sol(A(\vec{x}))_{A(\vec{x}) \in \mathrm{FPA}} \rangle.$$

Hence, the constraint system $\mathcal{D}_{\mathrm{FPA}}$ may contain an infinite set of relations. We shall write $\mathrm{CLTL}(\mathrm{FPA})$ instead of $\mathrm{CLTL}(\mathcal{D}_{\mathrm{FPA}})$. For instance the constraints of the form $x \geq c$ with $c \in \mathbb{Z}$ induces the constraint system

$$\langle \mathbb{Z}, (\{n \in \mathbb{Z} : n \geq c\})_{c \in \mathbb{Z}} \rangle,$$

with a countably infinite set of relations.

In the next subsection, we present fragments of Presburger arithmetic useful in the rest of the paper.

---

1. Unlike what is done in the journal paper, we consider the existential version for the Σ^1_1 results.

2.2.2. *Integer periodicity constraints*

We define below languages of the first-order theory of integer periodicity constraints. The *constraint language IPC* is defined by the grammar below :

$$A ::= x \equiv_k y + c \mid x \equiv_k c \mid A \wedge A \mid \neg A,$$

where $k, c \in \mathbb{N}$, $x, y \in \mathrm{VAR}$. Given $X \subseteq \{\exists, [], <, \mathtt{Eq}\}$, we define an extension of IPC, namely IPC^X, by adding clauses to the definition of IPC :

– if $\exists \in X$, then the clause $\exists\, x\ A$ is added (existential quantification),

– if $[] \in X$, then the clause $x \equiv_k y + [c_1, c_2]$ with $c_1, c_2 \in \mathbb{N}$ is added,

– if $\mathtt{Eq} \in X$, then the clause $x = y$ with $x, y \in \mathrm{VAR}$ is added,

– if $< \in X$, then the clauses $x < d \mid x > d \mid x = d$ with $x \in \mathrm{VAR}$ and $d \in \mathbb{Z}$ are added.

Observe that the presence of "$\mathtt{Eq}$" in X allows to express equality constraints between two variables whereas the presence of "$<$" allows to express equality constraints between a variable and a constant. By way of example, we provide the semantics for the formula $x \equiv_k y + [c_1, c_2] : v \models x \equiv_k y + [c_1, c_2] \stackrel{\text{def}}{\Leftrightarrow}$ there are $c \in [c_1, c_2]$ and $l \in \mathbb{Z}$ such that $(v(x) - v(y)) = l \times k + c$.

Below, IPC^+ denotes $\mathrm{IPC}^{\{\exists, [], <\}}$ whereas IPC^{++} denotes $\mathrm{IPC}^{\{\exists, [], <, \mathtt{Eq}\}}$. IPC^{++} is the extension of the language of the first-order theory of integer periodicity constraints introduced in (Toman *et al.*, 1998) but with the inclusion of negation as considered in (Bertino *et al.*, 1996). Observe that what is called "IPC" in (Toman *et al.*, 1998) is precisely defined by $A ::= x \equiv_k y + c \mid x \equiv_k c \mid A \wedge A \mid \exists x\ A$.

As Presburger arithmetic, IPC^{++} enjoys a quantifier elimination property but the complexity of the consistency problem is lower.

Theorem 2.2. (Demri, 2006)

(I) IPC^{++} consistency is PSPACE-complete.

(II) Given a constraint A in IPC^{++}, one can compute an equivalent quantifier-free A' in polynomial space in $|A|$ and $|A'|$ is in $\mathcal{O}(2^{|A|})$.

2.2.3. *Difference logic*

Let DL be the following fragment of Presburger arithmetic made of difference constraints :

$$A ::= x \sim y + d \mid x \sim d \mid A \wedge A \mid \neg A$$

where $x, y \in \mathrm{VAR}$, $\sim \in \{<, =\}$ and $d \in \mathbb{Z}$. We use the notations $x \leq y$, $x \geq y$ and $x > y$ as the respective abbreviations of $x < y \vee x = y$, $\neg(x < y)$ and $\neg(x \leq y)$. Given a valuation $v : \mathrm{VAR} \to \mathbb{Z}$, the satisfaction relation $v \models c$ is defined in the obvious way. DL is a proper fragment of quantifier-free Presburger arithmetic. Indeed, periodicity constraints of the form $x \equiv_k c$ or comparisons of the form $x + y + z < 5$ are not part of DL.

The satisfiability problem for $\mathrm{CLTL}(\mathrm{DL})$ can be placed in the analytical hierarchy in Σ_1^1.

2.2.4. *Quantifier free Presburger arithmetic*

The last fragment of Presburger arithmetic we consider in this paper is quantifier-free Presburger arithmetic QFP that is mainly used in Section 5.6. It is defined as follows :

$$A ::= \sum_{i\in I} a_i x_i = d \mid \sum_{i\in I} a_i x_i < d \mid \sum_{i\in I} a_i x_i \equiv_k c \mid \neg A \mid A \wedge A$$

where $a_i, d \in \mathbb{Z}$, $k, c \in \mathbb{N}$ and I is a finite subset of $\mathbb{N}$. Observe that QFP is as expressive as Presburger arithmetic but it is less concise (consistency is in NP).

3. Quantitative constraints and undecidability

In this section, we present undecidable fragments of Presburger LTL even though some drastic syntactic restrictions are sometimes considered.

3.1. *Minsky machines*

Undecidability of LTL with Presburger constraints shall be mainly established by reducing the halting problem or the recurrence problem for nondeterministic Minsky machines. That is why, we recall below what are these problems. A nondeterministic Minsky machine M consists of two counters C_1 and C_2, and a sequence of $n \geq 1$ instructions, each of which may increment or decrement one of the counters, or jump conditionally upon of the counters being zero. After the execution of a non-jump instruction, M proceeds nondeterministically to one of two specified instructions. The l^{th} instruction has one of the following forms :

l **:** $C_i := C_i + 1$; goto l' or goto l''

l **:** if $C_i = 0$ then goto l' else $C_i := C_i - 1$; goto l' or goto l''

The configurations of M are triples $\langle l, c_1, c_2\rangle$, where $1 \leq l \leq n$, $c_1 \geq 0$, and $c_2 \geq 0$ are the current values of the location counter and the two counters C_1 and C_2, respectively. The consecution relation on configurations is defined in the obvious way. A computation of M is an ω-sequence of related configurations, starting with the initial configuration $\langle 1, 0, 0\rangle$. The computation is recurring if it contains infinitely many configurations with the value of the location counter being 1. The recurrence problem is to decide, given a nondeterministic Minsky machine M, whether M has a recurring computation. This problem is Σ_1^1-hard (Alur *et al.*, 1994b, Section 4.1). Similarly, the problem of checking whether from $\langle 1, 0, 0\rangle$ a configuration with location counter 1 can be reached in at least one step is undecidable (Minsky, 1967) –halting problem.

3.2. *Constraint systems with counting mechanism*

High undecidability of $\mathrm{CLTL}^1_\omega(\mathbb{N}, =, +1)$ satisfiability can be easily shown by reducing the recurrence problem for nondeterministic Minsky machines. More generally, we define below three abstract conditions for a constraint system to admit implicitly a counting mechanism which leads to undecidability.

Definition 3.1. A constraint system $\mathcal{D}$ is said to admit an implicit counting mechanism if the following conditions are met :

1) $\mathcal{D}$ contains the equality predicate,

2) $\mathcal{D}$ contains a binary predicate R such that

a) $R = \{\langle a, b\rangle \in D^2 : f(a) = b\}$ for some injective map $f : D \to D$,

b) $\langle D, R\rangle$ is a DAG.

∇

So, whenever $\mathcal{D}$ has an implicit counting mechanism, for any $a \in D$, $a \xrightarrow{R} f^1(a) \xrightarrow{R} \ldots f^i(a) \xrightarrow{R} \ldots$ is isomorphic to $\langle \mathbb{N}, <\rangle$. For instance, for every $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$ and for every $i \in D \setminus \{0\}$, the constraint system $\langle D, =, =_{+i}\rangle$ has an implicit counting mechanism, where $n =_{+i} n'$ iff $n = n' + i$. Similarly, the constraint system $\langle D \setminus \{0\}, =, =_{\times i}\rangle$ admits an implicit counting mechanism where $n =_{\times i} n'$ iff $n = n' \times i$ with $i \neq -1$.

Theorem 3.1. (Demri *et al.*, 2003) The satisfiability problem for $\mathrm{CLTL}^1_\omega(\mathcal{D})$ is Σ^1_1-hard for every constraint system $\mathcal{D}$ with an implicit counting mechanism.

The proof is by reducing the recurrence problem for nondeterministic Minsky machines (see Section 3.1) as done in the proof of (Comon *et al.*, 2000, Theorem 3). Indeed, computations of such machines can be encoded as models of some $\mathrm{CLTL}^1_\omega(\mathcal{D})$ formula. It is then easy to express that a location counter appears infinitely often. This is the way $\mathrm{CLTL}^1_3(\mathrm{DL})$ satisfiability can be shown undecidable, see e.g. (Comon *et al.*, 2000). For instance the instruction

– $l : C_1 := C_1 + 1$; goto l' or goto l''

can be encoded by a temporal formula of the form

$$\mathsf{G}(x_l = y_l \Rightarrow (\mathsf{X}z_1 = z_1 + 1 \wedge \mathsf{X}(x_{l'} = y_{l'} \vee x_{l''} = y_{l''}))),$$

where z_1 is a variable dedicated to the counter C_1.

Recall that Σ^1_1-hardness implies that no recursively enumerable axiomatization exists. As corollaries we obtain :

Corollary 3.2. For every $D \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{Q}_+\mathbb{R}, \mathbb{R}_+\}$,

(I) for every $i \in D \setminus \{0\}$, satisfiability for $\mathrm{CLTL}^1_\omega(D, =, =_{+i})$ is Σ^1_1-hard.
(II) for every $i \in D \setminus \{0, 1, -1\}$, satisfiability $\mathrm{CLTL}^1_\omega(D, =, =_{\times i})$ is Σ^1_1-hard.

3.3. *Restricting syntactic resources*

In Section 3.2, we have seen that $\mathrm{CLTL}^1_\omega(\mathbb{N}, =, +1)$ and variants are highly undecidable. The obvious way to encode Minsky machines requires an unbounded amount of variables in order to encode their control states. In this section, we provide evidence that restricting further the syntactic ressources preserves undecidability. For instance, Lemma 3.3 states how to reduce the X-length of formulae and Lemma 3.4 how to reduce the number of variables.

Lemma 3.3. (Demri *et al.*, 2006b) For all $k, l, k', l' \in \mathbb{N} \setminus \{0\}$ and constraint systems $\mathcal{D}$, there is an exponential time reduction from $\mathrm{CLTL}^l_k(\mathcal{D})$ into $\mathrm{CLTL}^{l'}_{k'}(\mathcal{D})$ whenever $k \times l = k' \times l'$ and $k' = k \times m$ for some $m \geq 2$.

The idea of the proof is to encode m states from a $\mathrm{CLTL}^l_k(\mathcal{D})$ model into a single state in a $\mathrm{CLTL}^{l'}_{k'}(\mathcal{D})$ model. For instance, if $k' = 2k$, then the $\mathrm{CLTL}^l_k(\mathcal{D})$ model below

$$\begin{pmatrix} 1 \\ 2 \\ \dots \\ k \end{pmatrix} \begin{pmatrix} k+1 \\ k+2 \\ \dots \\ 2k \end{pmatrix} \begin{pmatrix} 2k+1 \\ 2k+2 \\ \dots \\ 3k \end{pmatrix} \dots$$

is encoded as the $\mathrm{CLTL}^{l'}_{k'}(\mathcal{D})$ model below

$$\begin{pmatrix} 1 \\ 2 \\ \dots \\ 2k \end{pmatrix} \begin{pmatrix} 2k+1 \\ 2k+2 \\ \dots \\ 4k \end{pmatrix} \dots$$

Lemma 3.4 below states how to reduce the number of variables in formulae.

Lemma 3.4. (Demri *et al.*, 2006b) For all $k, l, k', l' \in \mathbb{N} \setminus \{0\}$ and constraint systems $\mathcal{D}$ with equality and at least three elements, there is a logspace reduction from $\mathrm{CLTL}^l_k(\mathcal{D})$ into $\mathrm{CLTL}^{l'}_{k'}(\mathcal{D})$ whenever $3k \times l \leq k' \times l'$ and $k = k' \times m$ for some $m \geq 2$.

The idea of the proof is to encode one state from a $\mathrm{CLTL}^l_k(\mathcal{D})$ model into $3m$ states in a $\mathrm{CLTL}^{l'}_{k'}(\mathcal{D})$ model. Only one state over three encodes values. Intermediate states are used to know when a sequence of $3m$ states in $\mathrm{CLTL}^{l'}_{k'}(\mathcal{D})$ corresponds to

a state in a $\mathrm{CLTL}_k^l(\mathcal{D})$ model. For instance, if $k' = 1$ and $k = 2$, then the $\mathrm{CLTL}_k^l(\mathcal{D})$ model below

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \end{pmatrix} \cdots$$

is encoded as the $\mathrm{CLTL}_{k'}^{l'}(\mathcal{D})$ model

$$\overbrace{1 = b \neq b \neq 2 \neq b \neq b \neq}^{\text{position 0}} \overbrace{3 = b \neq b \neq 4 \neq b \neq b \ldots}^{\text{position 1}}$$

where b denotes arbitrary values satisfying the mentioned relations with its neighbors (each occurrence of b corresponds to a possibly distinct value). In order to ensure that such values exist, we assume that the domain has at least three distinct elements. The beginning of the encoding of some state from the $\mathrm{CLTL}_k^l(\mathcal{D})$ model is such that two consecutive values of the variable are identical.

The flat fragment of $\mathrm{CLTL}_\omega^1(\mathrm{DL})$ is shown decidable in (Comon *et al.*, 2000) with a decision procedure of complexity at least the one of Presburger arithmetic. The flat fragment is defined by restricting the use of the until operator : the left arguments do not contain temporal operators. Undecidability of $\mathrm{CLTL}_3^1(\mathrm{DL})$ is shown in (Comon *et al.*, 2000) by reducing the halting problem for Minsky machines. Similarly, undecidability of $\mathrm{CLTL}_2^\omega(\mathrm{DL})$ is shown in (Demri *et al.*, 2003) with constraints of the form $x = y$ and $x = y + 1$. It is possible to refine these results.

Theorem 3.5. (Demri *et al.*, 2006b) Satisfiability for $\mathrm{CLTL}_1^2(\mathrm{DL})$ is Σ_1^1-complete.

As a consequence of Theorem 3.5 and Lemma 3.3, we can improve the Σ_1^1-hardness of $\mathrm{CLTL}_3^1(\mathrm{DL})$ established in (Comon *et al.*, 2000).

Corollary 3.6. The satisfiability problem for $\mathrm{CLTL}_2^1(\mathrm{DL})$ is Σ_1^1-complete.

Consequently, the logic $\mathcal{L}_p$ from (Comon *et al.*, 2000) restricted to two variables is also highly undecidable. Moreover, the satisfiability problem can easily be reduced to the model-checking problem since $\phi \in \mathrm{CLTL}(\mathrm{DL})$ is satisfiable iff $\mathcal{A}_\top \models \phi$ where $\mathcal{A}_\top$ is the one-state $\mathcal{D}$-automaton $\langle \{q\}, \delta, \{q\}, \{q\} \rangle$ with unique transition $q \xrightarrow{\top} q$ (i.e. every model is accepted by $\mathcal{A}_\top$). So we get the following corollary.

Corollary 3.7. (Demri *et al.*, 2006b) The model-checking problems for $\mathrm{CLTL}_2^1(\mathrm{DL})$ and $\mathrm{CLTL}_1^2(\mathrm{DL})$ are Σ_1^1-complete.

By close inspection of the proofs of Theorem 3.5 and Corollary 3.6, one can show that satisfiability and model-checking for $\mathrm{CLTL}_2^1(\mathrm{DL})$ and $\mathrm{CLTL}_1^2(\mathrm{DL})$ but restricted to the sometime operator F (instead of until) are also Σ_1^1-hard.

3.4. $\mathrm{CLTL}_1^1(\mathrm{QFP})$ *satisfiability*

Extending $\mathrm{CLTL}(\mathrm{DL})$ with constraints of the form $ax + by = 0$ where $a, b \in \mathbb{Z}$ results to an undecidable logic even restricted to one variable and X-length one.

Theorem 3.8. (Demri *et al.*, 2006b) $\mathrm{CLTL}_1^1(\mathrm{QFP})$ satisfiability is undecidable.

Indeed, the values of two counters $\langle c_1, c_2 \rangle$ in the configuration of a Minsky machine can be encoded by the value $2^{c_1}3^{c_2}$ for the variable. Zero tests, increments and decrements can be encoded with constraints of the form $x \equiv_2 0$, $x \equiv_3 0$, $\mathsf{X}x = 2x$ (incrementation of the first counter) etc. The value of the instruction counter is encoded for instance by repeating the same configuration. As a consequence, the model-checking problem is undecidable even for the fragment restricted to one variable and X-length one.

4. General schema with symbolic models

In this section, we explain the main approach to establish decidability of model-checking and satisfiability problems for $\mathrm{CLTL}(\mathcal{D})$. Unless otherwise stated, $\mathcal{D}$ is a constraint system $\mathcal{D} = \langle D, (R_\alpha)_{\alpha \in I} \rangle$ with I possibly infinite.

4.1. ω*-regularity via abstraction*

Given an LTL formula ϕ built over the propositional variables in $\{p_1, \ldots, p_s\}$, the LTL models of ϕ can be viewed as ω-sequences $\sigma : \mathbb{N} \to \Sigma$ with $\Sigma = \mathcal{P}(\{p_1, \ldots, p_s\})$. The automata-based approach advocated in (Vardi *et al.*, 1994) establishes that one can effectively and efficiently build a Büchi automaton $\mathcal{A}_\phi$ such that the language accepted by $\mathcal{A}_\phi$ contains precisely the models of ϕ, whence providing a method to solve the satisfiability and model-checking problems for LTL. Indeed, the non-emptiness problem for Büchi automata is NLOGSPACE-complete. We recall that a Büchi automaton is a structure $\mathcal{A} = \langle \Sigma, S, S_0, \rho, F \rangle$ such that Σ is a finite set of symbols (the alphabet), S is a finite set of states, $S_0 \subseteq S$ is the set of initial states, $\rho : S \times \Sigma \to \mathcal{P}(S)$ is the transition function and, $F \subseteq S$ is a set of final (or accepting) states. A run is an infinite sequence $s_0 \stackrel{a_0}{\to} s_1 \stackrel{a_1}{\to} s_2 \ldots$ such that for every $i \geq 0$, $s_{i+1} \in \rho(s_i, a_i)$ and $s_0 \in S_0$. Given a run c, we write $inf(c)$ to denote the set of states that occur infinitely often in c. An accepting run c verifies $inf(c) \cap F \neq \emptyset$. The word $\sigma \in \Sigma^\omega$ is accepted by the accepting run $c = s_0 \stackrel{a_0}{\to} s_1 \stackrel{a_1}{\to} s_2 \ldots$ whenever $\sigma = a_0 a_1 a_2 \ldots$. The word σ is also said to be accepted by the automaton $\mathcal{A}$ and we write $\mathrm{L}(\mathcal{A})$ to denote the language of infinite words accepted by the Büchi automaton $\mathcal{A}$.

Given an $\mathrm{CLTL}(\mathcal{D})$ formula ϕ built over the variables $x_1, \ldots, x_k$, the $\mathrm{CLTL}(\mathcal{D})$ models of ϕ are sequences $\mathbb{N} \to D^k$. The product D^k is not necessarily finite and

therefore $\mathrm{CLTL}(\mathcal{D})$ models *cannot* be viewed as ω-sequences over a *finite* alphabet. In order to reuse results for Büchi automata, one strategy consists in abstracting $\mathrm{CLTL}(\mathcal{D})$ models of ϕ as sequences $\mathbb{N} \to \{0,1\}^m$ for some $m \geq 1$ depending on ϕ and on the constraint system $\mathcal{D}$. Each element of $\{0,1\}^m$ can be viewed as a finite set of local properties (for instance "the next value of y is equal to the current value of x"). Given a formula ϕ and a model $\sigma : \mathbb{N} \to D^k$, we can define an abstract/symbolic model $abs(\phi, \sigma) : \mathbb{N} \to \{0,1\}^m$. In order to solve $\mathrm{CLTL}(\mathcal{D})$ as it is done for LTL, the best situation we can hope for, is that the set of abstract models derived from a given $\mathrm{CLTL}(\mathcal{D})$ formula is ω-regular and the corresponding Büchi automaton can be built effectively and efficiently.

The rest of this section is dedicated to the presentation of the main steps to switch from the concrete models to the symbolic ones.

4.2. *Syntactic measure*

In order to check the satisfiability status of a given $\mathrm{CLTL}(\mathcal{D})$ formula ϕ, we only need to take into account relevant syntactic resources in ϕ. For instance, for LTL formulae, the number of propositional variables occurring in the formula is the appropriate criterion. Since we are dealing with constraint systems, we need to extend this notion.

Definition 4.1. A (syntactic) measure μ is a triple $\langle k, l, X\rangle$ such that $k \in \mathbb{N} \setminus \{0\}$ (the number of variables from VAR), $l \in \mathbb{N}$ (the X-length) and X is a finite subset of relation symbols related to $\mathcal{D}$. ∇

For instance, given an $\mathrm{CLTL}(\mathbb{N}, <, =)$ formula ϕ, a measure $\langle k, l, X\rangle$ for ϕ can be defined such that k is the number of distinct variables occurring in ϕ (assumed to be $x_1, \ldots, x_k$), $l = |\phi|_{\mathsf{X}}$, and $X = \{<, =\}$. More generally, when I is finite, we can assume without any loss of generality that the relations in a measure are exactly those appearing in $\mathcal{D}$.

4.3. *Symbolic state*

The set of atomic temporal $\mathcal{D}$-constraints defined from a measure $\mu = \langle k, l, X\rangle$ is denoted by CONS^μ and is defined as the set

$$\{\mathsf{R}(t_1, \ldots, t_n) : \mathsf{R} \in X\}$$

where each term t_i is of the form $\mathsf{X}^{l'} x_{k'}$ with $0 \leq l' \leq l$ and $1 \leq k' \leq k$. Cardinality of CONS^μ is at most exponential in $k + l + |X|$. In the case the arity of constraint relations in $\mathcal{D}$ is bounded (as for instance in $\langle \mathbb{R}, <, =\rangle$), the cardinality of CONS^μ is polynomial in $k + l + |X|$. We write FOR^μ to denote the subset of $\mathrm{CLTL}(\mathcal{D})$

formulae restricted to atomic formulae in CONS^μ. We define below a symbolic state as a finite set of (local) properties satisfied at a given position of a model. It abstracts a D-valuation whose range can have values arbitrarily large.

Definition 4.2. A symbolic state with respect to the measure μ is a subset of CONS^μ. ∇

Observe that not every symbolic state is [resp. maximally] consistent. The set of symbolic states is denoted by SYMB^μ. A symbolic model ρ for $\mathrm{CLTL}(\mathcal{D})$ with respect to μ is a sequence $\rho : \mathbb{N} \to \mathrm{SYMB}^\mu$ (it is of the form $\mathbb{N} \to \{0,1\}^m$ for some $m \geq 1$, see Section 4.1). We are now in position to define the symbolic satisfaction relation $\models_\mu$ with respect to the measure μ. Formulae are in FOR^μ and symbolic models are ω-words over the alphabet SYMB^μ. Boolean and temporal operators are defined homomorphically as for $\models$. Only the atomic case requires a special treatment that is indeed similar to what happens for LTL : for any $\phi \in \mathrm{CONS}^\mu$, we have $\rho, i \models \phi \stackrel{\text{def}}{\Leftrightarrow} \phi \in \rho(i)$.

4.4. *Abstraction*

Given a concrete model σ of $\mathrm{CLTL}(\mathcal{D})$ and a measure μ, we write ρ^μ_σ (it corresponds to $abs(\phi, \sigma)$ in Section 4.1) to denote the symbolic model wrt μ such that for every $i \geq 0$,

$$\rho^\mu_\sigma(i) \stackrel{\text{def}}{=} \{\phi \in \mathrm{CONS}^\mu : \sigma, i \models \phi\}.$$

Lemma 4.1. Let μ be a measure, σ be a model, ϕ be a formula in FOR^μ and $i \in \mathbb{N}$. If $\sigma, i \models \phi$, then $\rho^\mu_\sigma, i \models_\mu \phi$.

Definition 4.3. An abstraction for $\mathrm{CLTL}(\mathcal{D})$ is a computable function f from the set of formulae to the set of measures. ∇

An abstraction is complete whenever for every formula ϕ, for all models σ and $i \in \mathbb{N}$, $\sigma, i \models \phi$ iff $\rho^{f(\phi)}_\sigma, i \models_{f(\phi)} \phi$.

Theorem 4.2. Let f be a complete abstraction. Then ϕ is $\mathrm{CLTL}(\mathcal{D})$ satisfiable iff there is a symbolic model ρ such that

(I) $\rho, 0 \models_{f(\phi)} \phi$,

(II) there is a concrete model σ such that $\rho = \rho^{f(\phi)}_\sigma$.

In the proof, if $\sigma, 0 \models \phi$, then by Lemma 4.1, $\rho^{f(\phi)}_\sigma, 0 \models_{f(\phi)} \phi$. Obviously, $\rho^{f(\phi)}_\sigma$ has a concrete model. Conversely, if ρ satisfies (I) and (II), then by completeness of the abstraction, $\sigma, 0 \models \phi$.

In order to establish that $\mathrm{CLTL}(\mathcal{D})$ satisfiability is in PSPACE, we often show that

– $\mathrm{CLTL}(\mathcal{D})$ admits a complete abstraction f where consistent elements of $\mathrm{SYMB}^{f(\phi)}$ can be encoded in polynomial space and checking whether $X \subseteq \mathrm{CONS}^{f(\phi)}$ is consistent can be done in polynomial space,

– the symbolic models satisfying Condition (I) in Theorem 4.2 can be defined with a Büchi automaton computable in polynomial space (as for LTL, see Section 4.5),

– the symbolic models satisfying Condition (II) in Theorem 4.2 can be defined with a Büchi automaton computable in polynomial space.

In order to preserve the PSPACE upper bound, Büchi automata can be replaced by any class of operational models whose non-emptiness problem can be check in NLOGSPACE, see e.g. (Demri *et al.*, 2006b). Examples of PSPACE upper bounds established with such an approach can be found in (Demri *et al.*, 2003; Demri *et al.*, 2005a; Gascon, 2005; Demri, 2006; Demri *et al.*, 2006b). However, the class of symbolic models satisfying Condition (I) is usually ω-regular, unlike the class of symbolic models satisfying Condition (II), see e.g. (Demri *et al.*, 2003).

4.5. *Büchi automaton for symbolic satisfaction*

We define below a Büchi automaton accepting the symbolic models wrt $f(\phi)$ symbolically satisfying ϕ (Condition (I) in Theorem 4.2). This slightly extends the standard translation from LTL formulae into Büchi automata (Vardi *et al.*, 1994).

We define $cl(\phi)$ the closure of ϕ with a slight modification to consider atomic constraints and an atom of ϕ is a maximally consistent subset of $cl(\phi)$. As usual $\psi_1 \mathsf{U} \psi_2 \in cl(\phi)$ implies $\mathsf{X}(\psi_1 \mathsf{U} \psi_2) \in cl(\phi)$ and $\psi_1 \mathsf{U} \psi_2$ belongs to an atom X iff either $\phi_2 \in X$ or $\phi_1, \mathsf{X}(\psi_1 \mathsf{U} \psi_2) \in X$. Let $\mathcal{A}_\phi^{f(\phi)}$ be the generalized Büchi automaton defined by the tuple $\langle Q, \delta, I, F \rangle$ over the alphabet $\mathrm{SYMB}^{f(\phi)}$ such that :

– Q is the set of atoms of ϕ and $I = \{X \in Q : \phi \in X\}$,

– $X \xrightarrow{Z} Y$ iff

- for every atomic temporal formula A in X, $Z \models_{\mathcal{D}} A$ (instance of the implication problem),

- for every $\mathsf{X}\psi \in cl(\phi)$, $\mathsf{X}\psi \in X$ iff $\psi \in Y$,

– Let $\{\psi_1 \mathsf{U} \phi_1, \ldots, \psi_n \mathsf{U} \phi_n\}$ be the set of until formulae in $cl(\phi)$. We pose $F = \{F_1, \ldots, F_n\}$ where $F_i = \{X \in Q : \psi_i \mathsf{U} \phi_i \notin X \text{ or } \phi_i \in X\}$ for every $i \in \{1, \ldots, n\}$.

It is then easy to show the following result.

Lemma 4.3. $\mathrm{L}(\mathcal{A}_\phi^{f(\phi)}) = \{\rho : \mathbb{N} \to \mathrm{SYMB}^{f(\phi)} \mid \rho, 0 \models_{f(\phi)} \phi\}$.

5. Decidability results

In this section, we provide various examples of decidable fragments of Presburger LTL and more generally of decidable LTL over constraint systems.

5.1. *Completion property*

A general problem about the class of logics $\mathrm{CLTL}(\mathcal{D})$ is to identify sufficient conditions on the constraint system $\mathcal{D}$ for which model-checking and satisfiability for $\mathrm{CLTL}(\mathcal{D})$ admit effective decision procedures, ideally in polynomial space as for plain LTL. Very often this amount to check whether $\mathcal{D}$ has a complete abstraction and whether for every measure μ, the set $\{\rho^\mu_\sigma : \sigma\ \mathrm{CLTL}(\mathcal{D})-\mathrm{model}\}$ of symbolic models wrt μ is ω-regular or not. In this section, we present a class of constraint systems such that the above set is indeed ω-regular and there is an easy way to characterize it with a Büchi automaton.

A symbolic model ρ wrt μ is one-step consistent iff for every $i \geq 0$,

– $\rho(i)$ is maximally consistent wrt μ,

– for every atomic formula $\mathsf{R}(\mathsf{X}^{l_1}x_{j_1}, \ldots, \mathsf{X}^{l_n}x_{j_n}) \in \mathrm{CONS}^\mu$ with $l_1, \ldots, l_n \geq 1$, $\mathsf{R}(\mathsf{X}^{l_1}x_{j_1}, \ldots, \mathsf{X}^{l_n}x_{j_n}) \in \rho(i)$ iff $\mathsf{R}(\mathsf{X}^{l_1-1}x_{j_1}, \ldots, \mathsf{X}^{l_n-1}x_{j_n}) \in \rho(i+1)$.

For the constraint system $\langle \mathbb{R}, <, = \rangle$ and the measure $\langle 1, 2, \{<, =\}\rangle$, the symbolic model wrt μ below is one-step consistent :

$$\rho_0 = \{x > \mathsf{X}x, \mathsf{X}x > \mathsf{XX}x, x > \mathsf{XX}x, x = x, \mathsf{X}x = \mathsf{X}x, \mathsf{XX}x = \mathsf{XX}x\}^\omega.$$

One-step consistency is a necessary condition for a symbolic model to belong to $\{\rho^\mu_\sigma : \sigma\ \mathrm{CLTL}(\mathcal{D})-\mathrm{model}\}$. We define below a class of constraint systems for which this condition is also sufficient.

A constraint system $\mathcal{D}$ has the completion property (Demri *et al.*, 2003) iff for every measure $\mu = \langle k, 0, X\rangle$, for every maximally consistent set $Y \subseteq \mathrm{CONS}^\mu$ (made of atomic $\mathcal{D}$-constraints) and $1 \leq k' < k$, if

– $Y' = \{\phi \in Y : \phi \in \mathrm{CONS}^{\langle k', 0, X\rangle}\}$ (restriction of Y to atomic $\mathcal{D}$-constraints with $\{x_1, \ldots, x_{k'}\}$) and,

– $v : \{x_1, \ldots, x_{k'}\} \to D$ such that $Y' = \{\phi \in \mathrm{CONS}^{\langle k', 0, X\rangle} : v \models \phi\}$,

then there is D-valuation $v' : \{x_1, \ldots, x_k\} \to D$ such that v' is an extension of v and $Y = \{\phi \in \mathrm{CONS}^\mu : v' \models \phi\}$. Similar properties have been also introduced in (Dechter, 1992; Balbiani *et al.*, 2002; Lutz *et al.*, 2005a) known as global consistency.

When $\mathcal{D}$ is a constraint system satisfying the completion property, the set of symbolic models obtained by abstraction can be easily characterized.

Lemma 5.1. Let $\mathcal{D}$ be a constraint system with the completion property and μ be a measure. A symbolic model ρ wrt μ has a concrete model iff ρ is one-step consistent.

For instance, $\langle \mathbb{R}, <, = \rangle$, $\langle \mathbb{R}_+, <, = \rangle$, $\langle \mathbb{Q}, <, = \rangle$, $\langle \mathbb{Q}_+, <, = \rangle$ as well as $\langle D, = \rangle$ for any non-empty set D satisfy the completion property. The above-mentioned symbolic model ρ_0 has therefore a concrete model. By constrast, ρ_0 has no concrete model with the constraint system $\langle \mathbb{N}, <, = \rangle$ because of well-foundedness.

Let μ be a measure and ϕ be a formula in FOR^{μ}. We write $\mathrm{L}^{\mu}(\phi)$ to denote the set of symbolic models ρ wrt μ such that $\rho, 0 \models_{\mu} \phi$ and ρ has a concrete model. Observe that ϕ is satisfiable iff $\mathrm{L}^{\mu}(\phi)$ is non-empty.

Lemma 5.2. Let $\mathcal{D}$ be a constraint system admitting a complete abstraction f and satisfying the completion property. Then, for every $\mathrm{CLTL}(\mathcal{D})$ formula ϕ, $\mathrm{L}^{f(\phi)}(\phi)$ is ω-regular.

The Büchi automaton accepting the language $\mathrm{L}^{f(\phi)}(\phi)$ can be defined as the intersection of the Büchi automaton accepting one-step consistent symbolic models wrt $f(\phi)$ and, $\mathcal{A}_{\phi}^{f(\phi)}$.

Based on these properties, we can establish the following complexity result.

Theorem 5.3. (Balbiani *et al.*, 2002; Demri *et al.*, 2002) Let $\mathcal{D}$ be a constraint system with a finite amount of relations. Whenever $\mathcal{D}$ satisfies the completion property and, the implication and the maximally consistency problems are in PSPACE, then $\mathrm{CLTL}(\mathcal{D})$ satisfiability and model-checking are in PSPACE.

The proof in (Balbiani *et al.*, 2002) uses arguments analogous to the ones used in (Sistla *et al.*, 1985) to show that LTL is in PSPACE whereas the proof in (Demri *et al.*, 2002) takes advantage of the automata-based approach for LTL (Vardi *et al.*, 1994).

Corollary 5.4. Model-checking and satisfiability for $\mathrm{CLTL}(\mathbb{R}, <, =)$, $\mathrm{CLTL}(\mathbb{Q}, <, =)$ and $\mathrm{CLTL}(D, =)$ for any set D with at least two elements, are PSPACE-complete.

Adding to $\mathrm{CLTL}(\mathbb{Q}, <, =)$ comparisons with constants in $\mathbb{Q}$ (encoded with a binary representation) of the form $x \sim c$ with $\sim \in \{<, =\}$ and $c \in \mathbb{Q}$ preserves the PSPACE upper bound. Indeed, constants in a formula can be encoded by variables that remain equal along the model and we need only to specify how the constants compare between each other.

5.2. *Finite model case*

For another class of constraint systems, we can provide complexity upper bounds for $\mathrm{CLTL}(\mathcal{D})$ problems, namely when $\mathcal{D}$ is finite. Finiteness means that $\mathcal{D}$ is of the

form $\langle D, R_1, \ldots, R_N \rangle$ where D is a finite set $\{d_1, \ldots, d_M\}$. It should not come as a surprise that for such a restricted case, $\mathrm{CLTL}(\mathcal{D})$ satisfiability is in PSPACE since one can design a simple logspace reduction from $\mathrm{CLTL}(\mathcal{D})$ satisfiability to LTL satisfiability.

Theorem 5.5. Let $\mathcal{D}$ be a finite constraint system. The satisfiability problem for $\mathrm{CLTL}(\mathcal{D})$ is decidable in PSPACE.

A proof consists in introducing an auxiliary constraint system

$$\mathcal{D}' = \langle D, P_1, \ldots, P_M \rangle$$

such that $P_i = \{d_i\}$ for each i. A logspace reduction from $\mathrm{CLTL}(\mathcal{D})$ to $\mathrm{CLTL}(\mathcal{D}')$ can be designed and $\mathrm{CLTL}(\mathcal{D}')$ satisfiability can be shown in PSPACE since $\mathcal{D}'$ admits an easy complete abstraction and satisfies the completion property, see also (Demri, 2004, Theorem 4). The reduction is indeed homomorphic for Boolean and temporal operators and $\mathsf{R}(t_1, \ldots, t_n)$ is translated into

$$\bigvee_{R(d_{i_1}, \ldots, d_{i_n})} P_{i_1}(t_1) \wedge \cdots \wedge P_{i_n}(t_n).$$

This atomic step still guarantees that the reduction is in logspace since the arities of the relations R_i's and the cardinal of $\mathcal{D}$ are parameters of the logic. PSPACE-hardness can be also easily shown by reducing LTL satisfiability [resp. model-checking] whenever $\mathcal{D}$ is non-trivial. This means that there is a relation R in $\mathcal{D}$ of arity $n \geq 1$ such that either $R \neq \emptyset$ or $R \neq D^n$.

5.3. *Integer periodicity constraints*

The language IPC^{++} is a quite expressive fragment of Presburger arithmetic. This is witnessed by the PSPACE-completeness of its consistency problem whereas $\langle \mathbb{N}, <, = \rangle$ is only NLOGSPACE-complete. For instance, formulae of $\mathrm{CLTL}(\mathrm{IPC}^{++})$ can encode calendars and slices from (Niezette *et al.*, 1992). A calendar C can be viewed as an ordered partition $X_1, X_2, \ldots$ of $\mathbb{N}$ such that (the partition can be finite but we omit this case here)

(ordering) for all i, x and y, $x \in X_i$ and $y \in X_{i+1}$ imply $x < y$,

(consecution) for every i, there are $x \in X_i$ and $y \in X_{i+1}$ such that $y = x + 1$.

A calendar $C = X_1, X_2, \ldots$ can be represented in $\mathrm{CLTL}(\mathrm{IPC}^{++})$ by the interpretation of a variable x in an $\mathrm{CLTL}(\mathrm{IPC}^{++})$ model $\sigma : \mathbb{N} \times \mathrm{VAR} \to \mathbb{Z}$ such that consecutive positions in σ having the same value for x belongs to the same class :

$$\underbrace{\sigma(0,x) = \sigma(1,x) = \ldots = \sigma(i_1,x)}_{X_1 = \{0,\ldots,i_1\}} \neq \underbrace{\sigma(i_1+1,x) = \ldots = \sigma(i_2,x)}_{X_2 = \{i_1+1,\ldots,i_2\}} \neq \ldots$$

In most cases, $\{\sigma(i,x) : i \in \mathbb{N}\}$ is naturally finite (minuts, hours, days in a week, months). This means that a class of such calendars can be alternatively encoded as consecutive positions having the same value modulo some integer.

Theorem 5.6. (Demri, 2004) Satisfiability and model-checking for $\mathrm{CLTL}(\mathrm{IPC}^{++})$ is PSPACE-complete.

A key argument for such a proof is that IPC^{++} admits a complete abstraction and it is possible to encode in polynomial-space the maximally consistent symbolic states. As an application of Theorem 5.6, we can characterize the complexity of the equivalence problem for extended single-string automata defined in (Lago *et al.*, 2001, Sect.5), see other related automata in (Bresolin *et al.*, 2004) and (Puppis, 2006). This problem is central to check whether two time granularities are equivalent (see also (Wijsen, 2000)) when granularities are encoded by such automata that can be viewed as Büchi automata recognizing exactly one ω-word. Guards on transitions expressed by integer periodicity constraints and update maps on transitions provide conciseness of such constraint automata. Similarly, in (Combi *et al.*, 2002), the authors advocate the need to design an extension of LTL that expresses quantitative temporal requirements, such as periodicity constraints. $\mathrm{CLTL}(\mathrm{IPC}^{++})$ with past-time operators (that is also PSPACE-complete) provides such an extension.

Let IPC' be the fragment of $\mathrm{IPC}^{\{\exists\}}$ containing Boolean combinations of atomic constraints of the form either $x \equiv_k c$ or $\exists z\,(x \equiv_k z \wedge y \equiv_{k'} z)$. An update map g for the variable x_i is defined as an expression of the form either $x_i := x_i + c$ or $x_i := c$ with $c \in \mathbb{Z}$. We write $\mathrm{UP}_{x_1,\ldots,x_n}$ to denote the set of update maps that uses variables from $\{x_1,\ldots,x_n\}$. An extended single-string automaton $\mathcal{A}$ (ESSA) over the finite set of variables $\{x_1,\ldots,x_n\}$ (Lago *et al.*, 2001) is a structure of the form $\langle Q, q_0, \overline{v_0}, \Sigma, \delta\rangle$ where

– Q is a finite set of states and $q_0 \in Q$ (initial state),

– $\vec{v_0} \in \mathbb{Z}^n$ (initial value of the variables $x_1,\ldots,x_n$),

– Σ is a finite alphabet,

– $\delta \subseteq Q \times \Sigma \times Q \times (\{\top\} \cup \mathrm{IPC}') \times \mathcal{P}(\mathrm{UP}_{x_1,\ldots,x_n})$ and for every $q \in Q$, there are exactly two u such that $\langle q, u\rangle \in \delta$, say u_1 and u_2, and in that case u_1 is of the form $\langle a_1, q_1, A, X_1\rangle$, u_2 is of the form $\langle a_2, q_2, \neg A, X_2\rangle$ where A is a constraint in IPC^* built over variables in $\{x_1,\ldots,x_n\}$ and in both X_1 and X_2 exactly one update map for x_i is present.

The elements of δ are also denoted by $q \xrightarrow{a,A,X} q'$ (A is the guard and X is the global update map).

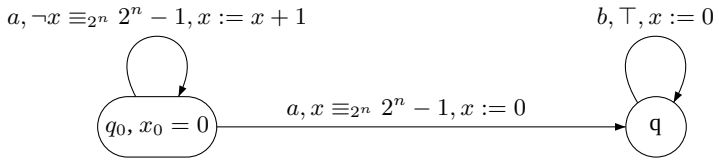
A configuration is a member $\langle q, \vec{v}\rangle \in Q \times \mathbb{Z}^n$. We define the one-step relation $\xrightarrow{a}$ for $a \in \Sigma$ as follows : $\langle q, \vec{v}\rangle \xrightarrow{a} \langle q', \vec{v'}\rangle$ iff there is $q \xrightarrow{a,A,X} q' \in \delta$ such that $[x_1 \leftarrow v_1, \ldots, x_n \leftarrow v_n] \models A$ (in IPC^{++}) and for every $g \in X$,

– if g is $x_i := x_i + c$ then $v'_i = v_i + c$;

– if g is $x_i := c$ then $v'_i = c$.

It is easy to check that there is exactly one sequence $w = a_1 a_2 \ldots \in \Sigma^\omega$ such that $\langle q_0, \vec{v_0}\rangle \stackrel{a_1}{\rightarrow} \langle q_1, \vec{v_1}\rangle \stackrel{a_2}{\rightarrow} \ldots$. The unique ω-sequence generated from the ESSA $\mathcal{A}$ is denoted by $w_\mathcal{A}$. The equivalence problem for ESSA consists in checking whether $w_\mathcal{A} = w_{\mathcal{A}'}$, given two ESSA $\mathcal{A}$ and $\mathcal{A}'$. This problem introduced in (Lago *et al.*, 2001) is central to check the equivalence of time granularities when granularities are encoded by such automata.

For instance, the ω-word associated with the ESSA below is $a^{2^n} \cdot b^\omega$ with initial value 0 :



Theorem 5.7. (Demri, 2004) The equivalence problem for ESSA is PSPACE-complete.

In order to establish the PSPACE upper bound, given two ESSA $\mathcal{A}$ and $\mathcal{A}'$, one can build an $\mathrm{IPC}^{\{\exists\}}$)-automaton $\mathcal{B}$ in logspace such that equivalent to $\mathcal{B} \models_\exists \top$ iff $w_\mathcal{A} = w_{\mathcal{A}'}$. Actually $l(\mathcal{B})$ will contain at most one ω-word. $\mathcal{B}$ is indeed a product between $\mathcal{A}$ and $\mathcal{A}'$. The PSPACE lower bound is obtained by adequately reducing QBF. The proof entails that checking whether $w_\mathcal{A} = w_{\mathcal{A}'}$ can be done in time

$$\mathcal{O}(2^{2 \times maxsize^2 \times n} \times |Q| \times |Q'|),$$

where n is the number of variables used in $\mathcal{A}, \mathcal{A}'$ and $maxsize$ is the size of the greatest integer k in $\equiv_k$-guards occurring in $\mathcal{A}, \mathcal{A}'$. Hence, the greatest integer occuring in $\mathcal{A}, \mathcal{A}'$ has value in $\mathcal{O}(2^{maxsize})$. Actually, the proof also entails that the problem remains PSPACE-hard when the only integer k in $\equiv_k$-guards occurring in $\mathcal{A}, \mathcal{A}'$ is 2 or when the integers are encoded with a unary representation. Similarly, the problem remains PSPACE-hard when only two distinct variables are used.

Another simpler problem which arises when dealing with time granularities, is to find the nth occurrence of a given symbol in a string (Lago *et al.*, 2003, Sect. 4). Here is the definition of the occurrence problem for ESSA :

input : An ESSA $\mathcal{A}$, $a \in \Sigma$ and $n, m \in \mathbb{N}$ (with a binary representation).

question Is the nth occurrence of a in $w_\mathcal{A}$ in position less than m ?

The proof of Theorem 5.7 can be easily adapted to prove the result below.

Theorem 5.8. (Demri, 2006) The occurrence problem for ESSA is PSPACE-complete.

5.4. *Decidability of* $\mathrm{CLTL}(\mathbb{Z}, <, =)$

Even though the consistency problems for $\langle \mathbb{Z}, <, = \rangle$ and $\langle \mathbb{R}, <, = \rangle$ are identical, and both $\mathrm{CLTL}(\mathbb{Z}, <, =)$ and $\mathrm{CLTL}(\mathbb{R}, <, =)$ admit complete abstractions, respectively, the satisfiability problem for $\mathrm{CLTL}(\mathbb{Z}, <, =)$ requires more sophisticated techniques to be shown in PSPACE than for $\mathrm{CLTL}(\mathbb{R}, <, =)$. Indeed, not only $\langle \mathbb{Z}, <, = \rangle$ does not satisfy the completion property, but also ω-regularity is not guaranteed which may invalidate the use of Büchi automata.

Lemma 5.9. (Demri *et al.*, 2003) There are a measure μ and a formula ϕ in FOR^{μ} such that $\mathrm{L}^{\mu}(\phi)$ is not ω-regular.

The very problem comes from the fact that for some measure μ, the class of one-step consistent symbolic models wrt μ that admit a concrete model is not necessarily ω-regular. In order to circumvent this difficult, we introduced in (Demri *et al.*, 2003) an over-approximation of this class that is ω-regular and we show that any ultimately periodic one-step consistent symbolic model wrt μ satisfying this new condition admit a concrete model. This allows us to show the result below.

Theorem 5.10. Satisfiability and model-checking for $\mathrm{CLTL}(\mathbb{N}, <, =)$ and $\mathrm{CLTL}(\mathbb{Z}, <, =)$ are PSPACE-complete.

The above result has been extended in (Demri *et al.*, 2005a). The set of IPC^* constraints A is defined as follows :

$$A ::= A' \mid x < y \mid A \wedge A \mid \neg A$$

$$A' ::= x \equiv_k [c_1, c_2] \mid x \equiv_k y + [c_1, c_2] \mid x = y \mid x < d \mid x = d \mid A' \wedge A' \mid \neg A' \mid \exists x\ A'$$

where $x, y \in \mathrm{VAR}$, $k \in \mathbb{N} \setminus \{0\}$, $c_1, c_2 \in \mathbb{N}$ and $d \in \mathbb{Z}$. IPC^* extends both $\langle \mathbb{Z}, <, = \rangle$ and IPC^{++}. Observe that the constraint language is defined in two layers in order to avoid an occurrence of $x < y$ in the scope of $\exists$ which would make possible the encoding of incrementation.

So far, IPC^* is the optimal known class of qualitative constraints on $\mathbb{Z}$ such that $\mathrm{CLTL}(\mathrm{IPC}^*)$ is decidable in polynomial-space. By a qualitative constraint, we mean for instance a constraint that is interpreted as a non-deterministic binary relation, like $x < y$ and $x \equiv_{2^k} y + 5$ (the relationship between x and y is not sharp). Constraint automata with qualitative constraints on $\mathbb{Z}$ are quite attractive operational models since they can be viewed as abstractions of counter automata where incrementations and decrementations are abstracted by operations modulo some power of two. Common programming languages perform arithmetic operators for integer types modulo 2^k (Müller-Olm *et al.*, 2005), typically k is either 32 or 64. For example, $x = y + 1$ can be abstracted by $x \equiv_{2^k} y + 1 \ \wedge \ y < x$ which is expressible in IPC^*.

By extending adequately proof techniques from (Demri *et al.*, 2003; Demri, 2004), we can characterize $\mathrm{CLTL}(\mathrm{IPC}^*)$ complexity. Observe that even though IPC^* contains both $\langle \mathbb{Z}, <, = \rangle$ and IPC^{++}, the successor relation cannot be defined in IPC^*.

Theorem 5.11. (Demri *et al.*, 2005a) Satisfiability and model-checking for $\mathrm{CLTL}(\mathrm{IPC}^*)$ are PSPACE-complete.

Surprisingly this allows to characterize the complexity of model-checking integral relational automata.

Corollary 5.12. The model-checking problem for integral relational automata restricted to the LTL fragment of CCTL* introduced in (Čerāns, 1994) is in PSPACE.

5.5. $\mathrm{CLTL}_1^1(\mathrm{DL})$

We have shown previously that both $\mathrm{CLTL}_2^1(\mathrm{DL})$ and $\mathrm{CLTL}_1^2(\mathrm{DL})$ are indecidable. When the number of variables and the X-length are both restricted to one, we regain the PSPACE upper bound.

Theorem 5.13. (Demri *et al.*, 2006b) Model-checking and satisfiability for $\mathrm{CLTL}_1^1(\mathrm{DL})$ are PSPACE-complete.

The proof is based on the fact that DL admits a complete abstraction and the class of symbolic models wrt to some measure of the form $\langle 1, 1, X \rangle$ that admit a concrete model can be recognized by one-counter automata where

– the counter is interpreted in $\mathbb{Z}$,

– there are zero tests and sign tests,

– accepted words are ω-sequences (Büchi acceptance condition),

– updates of the counter are among 0,-1,1.

Standard Büchi automata form a specific subclass of such one-counter automata. Additionally, in order to get the PSPACE upper bound one needs to show that non-emptiness problem for this class of one-counter automata is NLOGSPACE-complete (Demri *et al.*, 2006b).

5.6. *Model-checking* $\mathrm{CLTL}_1^1(\mathrm{QFP})$

We have seen in Section 3.4 that $\mathrm{CLTL}_1^1(\mathrm{QFP})$ satisfiability is highly undecidable. Consequently, so is $\mathrm{CLTL}_1^1(\mathrm{QFP})$ model-checking. However, there is a relevant restriction of the $\mathrm{CLTL}_1^\omega(\mathrm{QFP})$ model-checking problem that is decidable.

Indeed, consider as operational models one-counter automata where the counter is interpreted in $\mathbb{Z}$, there are zero tests and sign tests, accepted words are ω-sequences and updates of the counter are in $\mathbb{Z}$.

Theorem 5.14. (Demri *et al.*, 2006b) The model-checking problem for $\mathrm{CLTL}_1^{\omega}(\mathrm{QFP})$ over one-counter automata with updates in $\mathbb{Z}$ is PSPACE-complete.

5.7. *Other decidable extensions*

One of the nice features of Theorem 4.2 rests on the distinct treatments between symbolic satisfaction and existence of concrete models for symbolic models. Hence, let $\mathcal{D}$ be a constraint system such that $\mathrm{CLTL}(\mathcal{D})$ has been stated in PSPACE previously and LTL^+ be an extension of LTL for which formulae can be translated into Büchi automata in polynomial space. This includes for instance extensions

– with past-time operators previous and since, see e.g. (Lichtenstein *et al.*, 2000),

– more generally, with a finite amount of MSO-definable temporal operators (Gastin *et al.*, 2003),

– with automata-based operators (Wolper, 1983),

– with fixpoint operator, see e.g. (Vardi, 1988).

A quite remarkable feature of our proof technique is the following.

Theorem 5.15. $\mathrm{CLTL}^+(\mathcal{D})$ model-checking and satisfiability are also in PSPACE.

Extensions of $\mathrm{CLTL}(\mathcal{D})$ by addition of MSO-definable temporal operators (not necessarily a finite amount) are decidable if $\mathrm{CLTL}(\mathcal{D})$ can be proved decidable with the above-mentioned proof techniques. It suffices to adapt the definition of $\mathcal{A}_{\phi}^{f(\phi)}$ from plain LTL to LTL^+.

6. Ubiquous freeze operator

Atomic temporal $\mathcal{D}$-constraints allow us to compare values of variables at bounded distance as in $x < \mathsf{X}^2 y$. The languages presented previously have not the possibility to state a property of the form : "there is $i \geq 0$ such that $x < \mathsf{X}^i y$ holds true" which can be written "$\bigvee_i x < \mathsf{X}^i y$" with a generalized disjunction. Similarly, a property like "all the future values of x are different from the current value of x", which may be written $\bigwedge_{i>0} \neg(x = \mathsf{X}^i x)$, cannot be expressed in the previous languages. In this section, we investigate extensions of $\mathrm{CLTL}(\mathcal{D})$ that can express such properties by adding the so-called freeze operator that allows to store a value from D (typically the value of a variable) and to test it later but possibly at some unbounded distance from the position it has been stored. This is a very weak form of existential first-order

quantification. However, it is worth recalling that first-order LTL is known to be highly undecidable (Abadi, 1989; Kröger, 1990) even in the case the uninterpreted domains are finite (Trahtenbrot, 1963). The extension of $\mathrm{CLTL}(\mathcal{D})$ with addition of the freeze quantifier $\downarrow$ is denoted by $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$.

6.1. *Definition*

In order to define $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ formally, we divide the set VAR into the (countably infinite) set of rigid variables (VAR_r) and the (countably infinite) set of flexible variables (VAR_f). The clause $\downarrow_{y=\mathsf{X}^j x}\ \phi$ with $y \in \mathrm{VAR}_r$ and $x \in \mathrm{VAR}_f$ is added to the definition of $\mathrm{CLTL}(\mathcal{D})$. Atomic formulae are of the form $\mathsf{R}(t_1, \ldots, t_n)$ where each t_i is either a rigid variable or a term of the form $\mathsf{X}^i x$ with $x \in \mathrm{VAR}_f$. A model σ for $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ is an infinite sequence of valuations $\sigma : \mathbb{N} \times \mathrm{VAR}_f \rightarrow D$ over the set of flexible variables and the satisfaction relation is indexed by an environment $e : \mathrm{VAR}_r \rightarrow D$. The definition of $\models_e$ is extended as follows :

$$\sigma, i \models_e \downarrow_{y=\mathsf{X}^j x}\ \phi \overset{\text{def}}{\Leftrightarrow} \sigma, i \models_{e'} \phi \text{ where } e' \text{ is obtained from } e \text{ by only modifying the value for } y : e'(y) = \sigma(i+j)(x).$$

Satisfaction of atomic formulae uses both σ and e depending whether variables are in VAR_f or VAR_r. Without any loss of generality, we can also assume that for all the formulae ϕ in $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$, the free variables in ϕ are necessarily flexible.

A similar binding mechanism can be found in other logical formalisms :

– in real-time logics (Henzinger, 1990; Alur *et al.*, 1994b),

– in modal logics with λ-abstraction (Fitting, 2002; Lisitsa *et al.*, 2005),

– in first-order logic over data words (David, 2004; Bojańczyk *et al.*, 2006),

– in quantified propositional temporal logic with repeating (French, 2003).

A more detailed comparison can be found in (Demri *et al.*, 2006d).

The formula below from $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ states that in the model all the values of the variable x are distinct :

$$\varphi_0 = \mathsf{G} \downarrow_{y=x} \mathsf{XG}(x \neq y).$$

We can indeed show that the freeze quantifier is really powerful since every satisfiable formula ϕ in $\mathrm{CLTL}(\mathbb{N}, =)$ has a model which contains only finitely many distinct values (Demri *et al.*, 2006d). By contrast, all the models of φ_0 contain an infinite amount of distinct values.

6.2. *Undecidable fragments*

Surprisingly, adding the freeze operator leads to undecidability even for the poor constraint system $\langle \mathbb{N}, = \rangle$. Hence, the effects of adding the freeze operator in full generality are quite devastating. Recall that $\langle \mathbb{N}, = \rangle$ satisfies the completion property and $\mathrm{CLTL}(\mathcal{D})$ admits a complete abstraction.

Theorem 6.1. (Demri *et al.*, 2005b) Satisfiability for $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ restricted to two rigid variables and one flexible variable is Σ_1^1-complete.

Undecidability with three rigid variables has been independently shown in (Lisitsa *et al.*, 2005). The Σ_1^1 upper bound is easily obtained. In order to get the hardness result, first one can show that satisfiability for $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ can be reduced in logspace into its restriction to a unique flexible variable. Then, the recurrence problem for nondeterministic 2-counter machines can be reduced to satisfiability for $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ restricted to two rigid variables and one flexible variable. Undecidability is a consequence of the fact that the freeze operator can express that if there are n consecutive distinct values (of the flexible variable) between two special markers (encoding that a counter has value n) then there are $n + 1$ consecutive distinct values between the next occurrences of the two special markers and the subsequence of n last distinct values is equal to the first sequence of n distinct values. This requires exactly the use of two rigid variables.

The above problem is also undecidable with finite models. When a single rigid variable is considered undecidability still holds but with a lower degree.

Theorem 6.2. (Demri *et al.*, 2006c) Satisfiability for $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ restricted to one rigid variable and one flexible variable is Π_1^0-complete.

The proof is by reduction from infinitary nonemptiness problem for incrementing counter automata. This new class of automata are the counterpart of lossy counter automata (a special class of lossy channel systems (Schnoebelen, 2002)) in which counters can increment without notice.

6.3. *Decidability but not PR*

The undecidability results presented in Section 6.2 does not leave many hopes to use the freeze operator while being on the decidability side. Indeed, these results are obtained with quite poor constraint systems. By contrast, finite models and one rigid variable lead to decidability with very high complexity.

Theorem 6.3. (Demri *et al.*, 2006c) Satisfiability for $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ restricted to one rigid variable over finite models is decidable but not primitive recursive.

The decidability proof is done in three steps :

1) There is a logspace reduction from $\mathrm{CLTL}^{\downarrow}(\mathbb{N},=)$ restricted to one rigid variable to $\mathrm{CLTL}^{\downarrow}(\mathbb{N},=)$ restricted to one rigid variable, one flexible variable y and the freeze operator is only used in subformulae of the form $\downarrow_{y=x} \psi$ (Demri *et al.*, 2005b).

2) Every formula in this fragment can be reduced to an equivalent alternating 1-register automaton (Demri *et al.*, 2006c).

3) Non-emptiness for alternating 1-register automaton can be reduced to decidable finitary non-emptiness problem for incrementing counter automata (Demri *et al.*, 2006c).

Non primitive recursiveness is proved in two steps :

1) Finitary nonemptiness for incrementing counter automata can be shown non primitive recursive by adapting the proof in (Schnoebelen, 2002).

2) This latter problem can be reduced in logspace to satisfiability in $\mathrm{CLTL}^{\downarrow}(\mathbb{N},=)$ restricted to one rigid variable.

6.4. *Finiteness, flatness and periodicity constraints*

Theorem 5.5 has a counterpart in presence of the freeze operator possibly at the cost of one more exponential in space. Let $\mathcal{D}$ be a constraint system $\langle D, R_1, \ldots, R_N \rangle$ where D is a finite set.

Theorem 6.4. (Demri *et al.*, 2005b) Let $\mathcal{D}$ be a finite constraint system. The satisfiability problem for $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ is decidable in EXPSPACE.

The proof is by designing an exponential-time reduction from $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ satisfiability into $\mathrm{CLTL}(\mathcal{D}')$ for some other finite constraint system $\mathcal{D}'$ and then to take advantage of Theorem 5.5. It is natural to wonder whether the exponential space upper bound is optimal. After all, $\mathrm{CLTL}(\mathcal{D})$ satisfiability is only in polynomial space. For most domains $\mathcal{D}$, the answer is positive as stated below.

Theorem 6.5. (Demri *et al.*, 2005b) Let $\mathcal{D}$ be a constraint system with equality such that the underlying domain D contains at least two elements. $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ satisfiability is EXPSPACE-hard.

$\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ satisfiability can be also shown to be decidable when restricted to so-called flat formulae. For the rest of this section, $\mathcal{D}$ is not necessarily finite. Flat fragments of plain LTL versions have been studied in (Dams, 1999; Comon *et al.*, 2000) (see also in (Ibarra *et al.*, 2001, Section 5) the design of a flat logical temporal language for model-checking pushdown machines) and the definition of flatness takes advantage of the polarity of 'until' subformulae occurring in a formula.

We say that the occurrence of a subformula in a formula is positive if it occurs under an even number of negations, otherwise it is negative. The flat fragment of $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ is the restriction of $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ where, for any subformula $\phi_1 \mathsf{U} \phi_2$, if it is positive then $\downarrow$ does not occur in ϕ_1, and if it is negative then $\downarrow$ does not occur in ϕ_2.

This concept of flatness restricts the interplay between future-time operators and the freeze quantifier as done in (Bouajjani *et al.*, 1996; Comon *et al.*, 2000; ten Cate *et al.*, 2005) to limit the interaction between modalities and freeze-like quantifiers. In order to understand why flat formulae are more manageable, in a formula like $\downarrow_{y=x} \mathsf{F}\phi$ that is flat, only the current value of x needs to be stored. By contrast, in a formula like $\mathsf{G} \downarrow_{y=x} \phi$ that is not flat, one needs to store as many values of x as there are future positions.

We assume that the flexible variables of $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ are $\{x_0, x_1, \ldots\}$ and the rigid variables of $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ are $\{y_0, y_1, \ldots\}$. For ease of presentation, we assume that the flexible variables of $\mathrm{CLTL}(\mathcal{D})$ are composed of the following two disjoint sets : $\{x_0, x_1, \ldots\}$ and $\{y_0^{\text{new}}, y_1^{\text{new}}, \ldots\}$. We define a map t from the flat fragment $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ into $\mathrm{CLTL}(\mathcal{D})$ as follows : t replaces each y_j by y_j^{new} in atomic formulae, it is homomorphic for Boolean and temporal operators, and

$$t(\downarrow_{y=\mathsf{X}^n x} \psi) \stackrel{\text{def}}{=} y^{\text{new}} = \mathsf{X}^n x \wedge \mathsf{G}(y^{\text{new}} = \mathsf{X} y^{\text{new}}) \wedge t(\psi)$$

It is easy to show that $t(\phi)$ can be computed in logspace in the size of ϕ.

Lemma 6.6. Let $\mathcal{D}$ be a constraint system with equality. For any formula ϕ of the flat fragment of $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$, ϕ is $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ satisfiable iff $t(\phi)$ is $\mathrm{CLTL}(\mathcal{D})$ satisfiable.

As a corollary, we obtain the following result.

Theorem 6.7. (Demri *et al.*, 2005b) Flat fragments of $\mathrm{CLTL}^{\downarrow}(\mathbb{Z}, <, =)$, $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, <, =)$, $\mathrm{CLTL}^{\downarrow}(\mathbb{R}, <, =)$, and $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ with $\mathcal{D}$ finite are PSPACE-complete.

Even though flat $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ satisfiability can be reduced in logspace to $\mathrm{CLTL}(\mathbb{N}, =)$ satisfiability, $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ is more expressive than $\mathrm{CLTL}(\mathbb{N}, =)$, see details in (Demri *et al.*, 2006d).

Finally, using ideas and techniques useful to prove above-mentioned results, we can show that the following result.

Theorem 6.8. (Demri, 2006) Satisfiability for $\mathrm{CLTL}^{\downarrow}(\mathrm{IPC}^{+})$ is EXPSPACE-complete.

By contrast, $\mathrm{CLTL}^{\downarrow}(\mathrm{IPC}^{++})$ is undecidable as a consequence of Theorem 6.1 and $\mathrm{CLTL}(\mathrm{IPC}^{+})$ is "only" PSPACE-complete.

An alternative binding mechanism is the existential quantification. In order to define $\mathrm{CLTL}^{\exists}(\mathcal{D})$ formally, we consider the same syntactic categories as for $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ except that we add the clause $\exists\ y\ \phi$ with $y \in \mathrm{VAR}_r$ to define formulae. $\mathrm{CLTL}^{\exists}(\mathcal{D})$ and $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ have the same models and the satisfaction relation is indexed by an environment $e : \mathrm{VAR}_r \to D$. The relation $\models_e$ is defined as follows :

$$\sigma, i \models_e \exists\ y\ \phi \stackrel{\text{def}}{\Leftrightarrow} \text{there is } a \in D \text{ such that } \sigma, i \models_{e'} \phi \text{ where } e' \text{ is obtained from } e \text{ only modifying the value of } y : e'(y) = a.$$

When $\mathcal{D}$ contains equality, $\mathrm{CLTL}^{\downarrow}(\mathcal{D})$ can be viewed as a fragment of $\mathrm{CLTL}^{\exists}(\mathcal{D})$ but it is not always the case as in IPC^{+}. Indeed, $\downarrow_{y=x} \phi$ is then equivalent to $\exists\ y\ x = y \wedge \phi$.

Theorem 6.9. (Demri, 2006) Satisfiability for $\mathrm{CLTL}^{\exists}(\mathrm{IPC}^{+})$ is EXPSPACE-complete.

The simple fragment of $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ with past-time operators is restricted to formulae with one flexible variable x and one rigid variable y such that

– the only temporal operators are X, X^{-1}, XXF and $\mathsf{X}^{-1}\mathsf{X}^{-1}\mathsf{F}^{-1}$,

– every occurrence of such temporal operators is immediately preceded by $\downarrow_{y=x}$,

– the only terms occurring in atomic formulae are x and y.

X^{-1} is the "previous" operator and F^{-1} is the "sometimes in the past" operator. This simple fragment is known to be equivalent to a first-order logic with two variables, see e.g. (Demri *et al.*, 2006c) and consequently we can establish the following result by using the decidability for this first-order language shown in (Bojańczyk *et al.*, 2006).

Theorem 6.10. (Demri *et al.*, 2006c) Satisfiability for the simple fragment of $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ with past-time operators is decidable.

7. Related work

In this section, we provide useful pointers to works dealing with temporal logics and Presburger constraints in order to facilitate the comparison between the present framework and existing logical formalisms. Section 7.1 deals with first-order temporal logics whereas Section 7.2 presents LTL variants with quantitative aspects. Section 7.3 refers to quantitative branching-time temporal logics whereas Section 7.4 is dedicated to specification languages for expressing properties about counter systems. Logics with Presburger constraints on the number of children in tree-like models are described in Section 7.5 (see also the automata-based counterparts). Sections 7.6 and 7.7 present alternative temporal logics over concrete domains such as spatio-temporal logics and description logics over concrete domains.

7.1. *First-order temporal logics*

The different variants of LTL over constraint systems presented in this paper can be viewed as fragments of first-order LTL where the domain of interpretation of variables is fixed as well as the interpretation of predicate symbols. Moreover, flexible variables in $\mathrm{CLTL}(\mathcal{D})$ correspond to unary predicate symbols interpreted as singletons (the values of variables). However, $\mathrm{CLTL}(\mathcal{D})$ has no quantification over elements of the domain and in that sense it corresponds to a very weak first-order extension of plain LTL. It is worth recalling that first-order LTL is highly undecidable (Abadi, 1989; Kröger, 1990) even in the case the uninterpreted domains are finite (Trahtenbrot, 1963). Similarly, first-order LTL over finite time structures is highly undecidable (Cerrito *et al.*, 1999). In particular, the Σ_1^1-hardness implies that no axiomatization with a countable set of axioms exists for this logic.

Since the freeze binding mechanism is a first-order quantification over a singleton set, the freeze quantifier can be expressed in first-order temporal logics (Degtyarev *et al.*, 2002; Wolter *et al.*, 2002; Hodkinson *et al.*, 2003; Gabbay *et al.*, 2003). Indeed, $\mathrm{CLTL}^{\downarrow}(\mathbb{N}, =)$ satisfiability can be reduced to first-order temporal logic $\mathcal{TL}$ satisfiability over the linear structure $\langle \mathbb{N}, < \rangle$ (the latter logic was introduced in (Gabbay *et al.*, 2003, Chapter 11)). To each flexible variable x one associates a unary predicate symbol P_x in such a way that P_x is interpreted as the singleton set containing the value of x.

A variant of first-order LTL has been introduced in (Deutsch *et al.*, 2004) to verify data-driven web applications. The interplay between temporal operators and first-order quantifiers is restricted since no quantification can occur in the scope of temporal operators, which guarantees better computational properties.

7.2. *Quantitative versions of LTL*

The logics presented in this paper belong to the long tradition of quantitative versions of LTL. LTL-like logics having models non isomorphic to ω can be found in (Alur *et al.*, 1996; Reynolds, 2003; Hirshfeld *et al.*, 2003; Hirshfeld *et al.*, 2004; Lutz *et al.*, 2005b; Demri *et al.*, 2005c). Temporal operators in the real-time logics from (Alur *et al.*, 1996; Hirshfeld *et al.*, 2004; Lutz *et al.*, 2005b) are indexed by intervals. By contrast, Constrained LTL defined in (Bouajjani *et al.*, 1995) have standard LTL models but the logical language is enriched with Presburger constraints about the number of occurrences of events. This extension of LTL is undecidable and decidable fragments are introduced in (Bouajjani *et al.*, 1995).

7.3. *Branching-time temporal Logics*

Integral relational automata defined in (Čerāns, 1994) form a subclass of IPC^*-automata and we have seen that the model-checking problem for $\mathrm{CLTL}(\mathrm{IPC}^*)$ is

PSPACE-complete. By contrast, it is shown in (Čerāns, 1994) that model-checking integral relational automata with a CTL version of $\mathrm{CLTL}(\mathrm{IPC}^*)$ is undecidable roughly because quantification over natural numbers can be simulated by a quantification over paths. However, model-checking for the existential fragment of the CTL* version of $\mathrm{CLTL}(\mathrm{IPC}^*)$ is shown decidable in (Bozzelli *et al.*, 2006) extending a weaker result from (Čerāns, 1994). The decidability proof is based on techniques for well-structured systems.

Very few classes of counter systems are decidable for CTL* (see e.g. (Finkel *et al.*, 1997) for one-counter systems). Another logical formalism close to the ones presented in this paper is studied in (Bultan *et al.*, 1997) where an undecidable temporal logic with CTL-like operators and atomic formulae in Presburger arithmetic is introduced and the models are counter systems.

7.4. *Verification of counter systems*

Analyzing the reachability problem for counter systems is ubiquitous for the verification of infinite-state systems, see e.g. (Ibarra *et al.*, 2000) (reversal-bounded systems), (Comon *et al.*, 1998) (flat systems), (Finkel *et al.*, 2002) (flat Presburger transition systems), (Dang *et al.*, 2003) (discrete timed automata), see also the decidability of reachability for classes of 2-counter systems (Finkel *et al.*, 2000). It is worth noting that, even though decidability can be obtained only at the cost of making drastic restrictions on counter systems, there is a remarkable class of counter systems that are sufficiently expressive for modelling different case studies. For instance, the flattable systems (Leroux *et al.*, 2005) admit a flat finite unfolding of the control graph with the same reachability set (see also (Demri *et al.*, 2006a) for properties other than accessibility). On the logical side, temporal logics with Presburger constraints have been developed in (Čerāns, 1994; Bouajjani *et al.*, 1995; Bultan *et al.*, 1997; Comon *et al.*, 2000; Bruyère *et al.*, 2003), some of which have quite expressive decidable fragments. However, undecidability of the reachability problem can be proved for quite restricted counter systems, see e.g. (Cortier, 2002; Potapov, 2004).

The subclass of one-counter automata is obviously equivalent to pushdown systems with a singleton stack alphabet and many results on such systems can be therefore applied to one-counter automata. For instance, model-checking one-counter automata with the modal μ-calculus has been shown in EXPTIME in (Walukiewicz, 2001) and this result has been refined in (Serre, 2004, Section 7.2) (see also (Serre, 2006)) where it is proved that the problem is in PSPACE. It is worth mentioning that in these logics the atomic formulae can only speak about the control states and not about the content of the counter, a major difference with formalisms involved in Theorem 5.14. Similarly, in (Bouajjani *et al.*, 1997) model-checking pushdown automata over the linear μ-calculus is proved in EXPTIME. Surprisingly, one-counter automata have applications in various parts of theoretical computer science :

– for verification of cryptographic protocols (Lafourcade *et al.*, 2005),

– for validating XML streams (string representations of XML documents) by encoding recursive DTDs as one-counter automata (Chitic *et al.*, 2004),

– to solve the identification problem (Wakatsuki *et al.*, 2004).

7.5. *Constraints on the number of children*

In order to query XML documents with Presburger and/or regular constraints on the children of nodes, logical and automata-based formalisms have been recently introduced (Seidl *et al.*, 2004; Ohsaki *et al.*, 2005; Zilio *et al.*, 2006; Boneva *et al.*, 2005; Demri *et al.*, 2006e) leading to various expressiveness and complexity results about logics and specialized tree automata. As usual, XML documents are viewed as finite labeled, unranked ordered trees. For instance, the computational complexity of a logic with fixed-point operators, Presburger and regularity constraints is studied in (Seidl *et al.*, 2004), improving results for description logics with qualified number restrictions (Hollunder *et al.*, 1991; Calvanese *et al.*, 2005). Graded modal logics from (Fine, 1972; Barnaba *et al.*, 1985; van der Hoek *et al.*, 1995) have very elementary Presburger constraints compared to modal logic extended with all quantifier-free Presburger constraints that is shown to be in PSPACE in (Demri *et al.*, 2006e), see also related logics in (Pacuit *et al.*, 2004; Schröder *et al.*, 2006). In those logics, Presburger formulae express constraints on the number of children.

7.6. *Spatio-temporal logics*

Spatio-temporal logics, see e.g. (Wolter *et al.*, 2000; Gabbay *et al.*, 2003; Hodkinson *et al.*, 2003; Gabelaia *et al.*, 2003) are examples of versions of LTL with constraint systems for which the constraint system has indeed a spatial structure. Examples of such spatial systems verifying the completion property and examples of systems not verifying this property can be found in (Balbiani *et al.*, 2002). Complexity and decidability results for such logics can be found in (Gabbay *et al.*, 2003).

7.7. *Description logics over concrete domains*

The introduction of concrete domains in description logics is due to (Baader *et al.*, 1991) and since then, such logic-based formalisms for knowledge representation have been intensively studied, see e.g. (Lutz, 2003; Lutz, 2004). As expected, concept satisfiability with respect to general TBoxes with an arithmetic concrete domain is undecidable which is comparable to the undecidability of $\mathrm{CLTL}(\mathbb{N}, =, +1)$ satisfiability. Hence, LTL over concrete domains can be technically viewed as a subclass of description logics over concrete domain in which the models are linear. Among the differences, it is worth observing that in LTL it is possible to access to any successor position with the always operator G whose counterpart in description logics is the presence of a transitive closure operator. Motivations are also quite different. In-

deed, the introduction of LTL over Presburger constraints is motivated by the need to model-check counter systems whereas concrete domains in description logics have been introduced to integrate concrete qualities into description logic concepts.

8. Concluding remarks

In this paper, we have presented an overview of linear-time temporal logics with Presburger constraints and we have compared them with alternative formalisms. Because of lack of space, we have not provided the full technical details and they can be found in the original papers. It is worth observing that a great amount of work remains to be done to push the decidability border further in order to get effective procedures to verify programs with variables interpreted in fixed domains.

Among the open problems, let us mention a few of them. The first track is certainly to have more decidability results for richer classes of constraint systems. For instance, what is the decidability status of $\mathrm{CLTL}(\{0,1\}^*, \subset, =)$ where $\subset$ is the subword [resp. prefix, factor] relation ? Obviously, $\mathrm{CLTL}(\{0\}^*, \subset, =)$ behaves as $\mathrm{CLTL}(\mathbb{N}, <, =)$. Developing model-checking techniques for such a logic might help to verify programs manipulating strings. Another line of research consists in extending known decidability proofs to heterogeneous constraint systems, for instance mixing strings and natural numbers. Finally, let us quote other interesting problems :

– How to refine the classification of constraint systems $\mathcal{D}$ such that $\mathrm{CLTL}(\mathcal{D})$ is decidable ? in PSPACE ?

– What makes decidable branching-time extensions of decidable $\mathrm{CLTL}(\mathcal{D})$?

– How to restrict further the use of the freeze binding quantifier in order to regain decidability ?

Acknowledgements : Numerous results presented in this paper have been obtained with colleagues. I would like to thank Deepak D'Souza, Régis Gascon, Ranko Lazić and David Nowak for fruitful and friendly collaborations. I am grateful to the INTAS project members for their kind invitation to present such an overview. Finally, I thank the anonymous referee for suggestions that help me to improve the readability of this document.

9. Bibliographie

Abadi M., « The power of temporal proofs », *Theoretical Computer Science*, vol. 65, p. 35-83, 1989.

Alur R., Dill D., « A theory of timed automata », *Theoretical Computer Science*, vol. 126, p. 183-235, 1994a.

Alur R., Feder T., Henzinger T., « The benefits of relaxing punctuality », *Journal of the Association for Computing Machinery*, vol. 43, p. 116-146, 1996.

Alur R., Henzinger T., « A really temporal logic », *Journal of the Association for Computing Machinery*, vol. 41, n° 1, p. 181-204, 1994b.

Baader F., Hanschke P., « A scheme for integrating concrete domains into concept languages », *12th International Joint Conference on Artificial Intelligence, Sydney, Australia*, p. 452-457, 1991.

Balbiani P., Condotta J., « Computational Complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning », *in* A. Armando (ed.), *Frontiers of Combining Systems (FroCoS'02)*, vol. 2309 of *Lecture Notes in Artificial Intelligence*, Springer, Berlin, p. 162-173, 2002.

Barnaba M. F., Caro F. D., « Graded Modalities », *Studia Logica*, vol. 44, n° 2, p. 197-221, 1985.

Bérard B., Bidoit M., Finkel A., Laroussinie F., Petit A., Petrucci L., Schnoebelen P., *Systems and Software Verification, Model-Checking Techniques and Tools*, Springer-Verlag, 2001.

Bertino E., Bettini C., Ferrari E., Samarati P., « Supporting Periodic Authorizations and Temporal Reasoning in Database Access Control », *22nd VLDB, Bombay, India*, p. 472-483, 1996.

Boigelot B., Symbolic methods for exploring infinite state spaces, PhD thesis, Université de Liège, 1998.

Boigelot B., Wolper P., « Representing arithmetic constraints with finite automata : an overview », *in* P. Stuckey (ed.), *ICLP'02*, vol. 2401 of *Lecture Notes in Computer Science*, Springer, p. 1-19, 2002.

Bojańczyk M., Muscholl A., Schwentick T., Segoufin L., David C., « Two-variable logic on words with data », *LICS'06*, IEEE, p. 7-16, 2006.

Boneva I., Talbot J., « Automata and Logics for Unranked and Unordered Trees », *RTA'05*, vol. 3467 of *Lecture Notes in Computer Science*, Springer, p. 500-515, 2005.

Bouajjani A., Echahed R., Habermehl P., « On the verification problem of nonregular properties for nonregular processes », *LICS'95*, p. 123-133, 1995.

Bouajjani A., Esparza J., Maler O., « Reachability Analysis of Pushdown Automata : Application to Model Checking », *CONCUR'97*, vol. 1243 of *LNCS*, Springer, p. 135-150, 1997.

Bouajjani A., Habermehl P., « Constrained Properties, Semilinear Sets, And Petri Nets », *CONCUR'96*, vol. 1119 of *Lecture Notes in Computer Science*, Springer, p. 481-497, 1996.

Bozzelli L., Gascon R., « Branching-time temporal logic extended with Presburger constraints », *LPAR'06*, Lecture Notes in Computer Science, Springer, p. ? ?- ? ?, 2006. to appear.

Bresolin D., Montanari A., Puppis G., « Time Granularities and Ultimately Periodic Automata », *JELIA'04*, vol. 3229 of *Lecture Notes in Computer Science*, Springer, p. 513-525, 2004.

Bruyère V., Dall'Olio E., Raskin J., « Durations, Parametric Model-Checking in Timed Automata with Presburger Arithmetic », *STACS'03*, vol. 2607 of *Lecture Notes in Computer Science*, Springer, p. 687-698, 2003.

Bultan T., Gerber R., Pugh W., « Symbolic model checking of infinite state systems using Presburger arithmetic », *CAV'97*, vol. 1254 of *Lecture Notes in Computer Science*, Springer, p. 400-411, 1997.

Calvanese D., Giacomo G. D., « Expressive Description Logics », *Description Logics Handbook*, Cambridge University Press, p. 178-218, 2005.

Čerāns K., « Deciding Properties of Integral Relational Automata », *ICALP'94*, vol. 820 of *Lecture Notes in Computer Science*, Springer, p. 35-46, 1994.

Cerrito S., Mayer M. C., Praud S., « First-Order Linear Temporal Logic over Finite Time Structures », *in* H. Ganzinger, D. McAllester, A. Voronkov (eds), *6th Int. Conference on Logic Programming and Automated Reasoning, Tbilisi, Republic of Georgia (LPAR'99)*, vol. 1705 of *Lecture Notes in Computer Science*, Springer, p. 62-76, 1999.

Chitic C., Rosu D., « On validation of XML streams using finite state machines », *WebDB, Paris*, p. 85-90, 2004.

Combi C., Franceschet M., Peron A., « A logical approach to represent and reason about calendars », *Int. Symposium on Temporal Representation and Reasoning*, IEEE Computer Society Press, p. 134-140, 2002.

Comon H., Cortier V., « Flatness is not a weakness », *CSL'00*, vol. 1862 of *Lecture Notes in Computer Science*, Springer, p. 262-276, 2000.

Comon H., Jurski Y., « Multiple counters automata, safety analysis and Presburger arithmetic », *in* A. Hu, M. Vardi (eds), *Proc. Computer Aided Verification, Vancouver*, vol. 1427 of *Lecture Notes in Computer Science*, Springer, Berlin, p. 268-279, 1998.

Cortier V., « About the Decision of Reachability for Register Machines », *Theoretical Informatics and Applications*, vol. 36, n° 4, p. 341-358, 2002.

Dams D., « Flat fragments of CTL and CTL* : separating the expressive and distinguishing powers », *Logic Journal of the IGPL*, vol. 7, n° 1, p. 55-78, 1999.

Dang Z., Pietro P. S., Kemmerer R., « Presburger Liveness Verification of Discrete Timed Automata », *Theoretical Computer Science*, vol. 299, p. 413-438, 2003.

David C., *« Mots et données infinies »*, Master's thesis, LIAFA, 2004.

Dechter R., « From local to global consistency », *Artificial Intelligence*, vol. 55, p. 87-107, 1992.

Degtyarev A., Fisher M., Lisitsa A., « Equality and monodic first-order temporal logic », *Studia Logica*, vol. 72, p. 147-156, 2002.

Demri S., « LTL over Integer Periodicity Constraints (Extended Abstract) », *in* I. Walukiewicz (ed.), *FOSSACS'04*, vol. 2987 of *Lecture Notes in Computer Science*, Springer, Berlin, p. 121-135, 2004.

Demri S., « LTL over integer periodicity constraints », *Theoretical Computer Science*, vol. 360, n° 1–3, p. 96-123, 2006.

Demri S., D'Souza D., « An automata-theoretic approach to constraint LTL », *in* M. Agrawal, A. Seth (eds), *FST&TCS'02, Kanpur*, vol. 2556 of *Lecture Notes in Computer Science*, Springer, Berlin, p. 121-132, 2002.

Demri S., D'Souza D., An automata-theoretic approach to constraint LTL, Technical Report n° LSV-03-11, LSV, August, 2003. 40 pages. Extended version of (Demri *et al.*, 2002). Under submission.

Demri S., Finkel A., Goranko V., van Drimmelen G., « Towards a model-checker for counter systems », *Proceedings of the 4th International Symposium on Automated Technology for Verification and Analysis (ATVA'06)*, vol. 4218 of *Lecture Notes in Computer Science*, Springer, Beijing, ROC, p. 493-507, 2006a.

Demri S., Gascon R., « Verification of Qualitative $\mathbb{Z}$-Constraints », *in* M. Abadi, de L. Alfaro (eds), *Proceedings of the 16th International Conference on Concurrency Theory (CONCUR'05)*, vol. 3653 of *Lecture Notes in Computer Science*, Springer, San Francisco, CA, USA, p. 518-532, August, 2005a.

Demri S., Gascon R., The effects of bounding syntactic resourses on Presburger LTL, Technical Report n° LSV-06-5, LSV, 2006b. 36 pages.

Demri S., Lazić R., « LTL with the freeze quantifier and register automata », *LICS'06*, IEEE, p. 17-26, 2006c.

Demri S., Lazić R., Nowak D., « On the freeze quantifier in constraint LTL : decidability and complexity », *12th Int. Symp. on Temporal Representation and Reasoning, Burlington, Vermont*, IEEE Computer Society Press, p. 113-121, 2005b.

Demri S., Lazić R., Nowak D., « On the freeze quantifier in constraint LTL : decidability and complexity », *Information and Computation*, 2006d. To appear. Extended version of (Demri *et al.*, 2005b).

Demri S., Lugiez D., « Presburger Modal Logic is PSPACE-complete », *3rd International Joint Conference on Automated Reasoning (IJCAR'06), Seattle*, vol. 4130 of *Lecture Notes in Computer Science*, Springer, p. 541-556, 2006e.

Demri S., Nowak D., « Reasoning about transfinite sequences (extended abstract) », *ATVA'05*, vol. 3707 of *Lecture Notes in Computer Science*, Springer, p. 248-262, 2005c.

Deutsch A., Sui L., Vianu V., « Specification and verification of data-driven web services », *PODS'04, Paris*, p. 71-82, 2004.

Fine K., « In so many possible worlds », *Notre Dame Journal of Formal Logic*, vol. 13, n° 4, p. 516-520, 1972.

Finkel A., Leroux J., « How to compose Presburger accelerations : Applications to broadcast protocols », *in* M. Agrawal, A. Seth (eds), *FST&TCS'02, Kanpur*, vol. 2256 of *Lecture Notes in Computer Science*, Springer, Berlin, p. 145-156, 2002.

Finkel A., Sutre G., « Decidability of reachability problems for classes of two counters automata », *17th Ann. Symp. Theoretical Aspects of Computer Science*, vol. 2256 of *Lecture Notes in Computer Science*, Springer-Verlag, p. 346-357, 2000.

Finkel A., Willems B., Wolper P., « A Direct Symbolic Approach to Model Checking Pushdown Systems (Extended Abstract) », *INFINITY'97*, vol. 9 of *ENTCS*, Elsevier Science, 1997.

Fischer M., Rabin M., « Super-exponential complexity of Presburger arithmetic », *in* R. Karp (ed.), *Complexity of Computation*, vol. 7 of *SIAM-AMS proceedings*, American Mathematical Society, p. 27-42, 1974.

Fitting M., « Modal logic between propositional and first-order », *Journal of Logic and Computation*, vol. 12, n° 6, p. 1017-1026, 2002.

French T., « Quantified Propositional Temporal Logic with Repeating States », *TIME-ICTL 2003*, IEEE Computer Society, p. 155-165, 2003.

Gabbay D., Kurucz A., Wolter F., Zakharyaschev M., *Many-dimensional modal logics : theory and practice*, Cambridge University Press, 2003.

Gabelaia D., Kontchakov R., Kurucz A., Wolter F., Zakharyaschev M., « On the computational complexity of spatio-temporal logics », *FLAIRS'03, St Augustine, Florida*, p. 460-464, 2003.

Gascon R., « Verifying qualitative and quantitative properties with LTL over concrete domains », *Proceedings of the 4th Workshop on Methods for Modalities (M4M'05)*, vol. 194 of *Informatik Bericht*, Humboldt Universität zu Berlin, p. 54-61, 2005.

Gastin P., Kuske D., « Satisfiability and model checking for MSO-definable temporal logics are in PSPACE », *CONCUR'03*, vol. 2761 of *Lecture Notes in Computer Science*, Springer, p. 222-236, 2003.

Ginsburg S., Spanier E., « Semigroups, Presburger formulas and languages », *Pacific Journal of Mathematics*, vol. 16, n° 2, p. 285-296, 1966.

Henzinger T., « Half-order modal logic : how to prove real-time properties », *9th Annual ACM Symposium on Principles of Distributed Computing*, ACM Press, p. 281-296, 1990.

Hirshfeld Y., Rabinovich A., « Future temporal logic needs infinitely many modalities », *Information and Computation*, vol. 187, n° 2, p. 196-208, 2003.

Hirshfeld Y., Rabinovich A., « Logics for real time : decidability and complexity », *Fundamenta Informaticae*, vol. 62, p. 1-28, 2004.

Hodkinson I., Kontchakov R., Kurucz A., Wolter F., Zakharyaschev M., « On the computational complexity of decidable fragments of first-order linear temporal logics », *10th Int. Symp. Temporal Representation and Reasoning and 4th Int. Conf. Temporal Logic (TIME-ICTL)*, IEEE, p. 91-98, 2003.

Hollunder B., Baader F., « Qualifying number restrictions in concept languages », *KR'91*, p. 335-346, 1991.

Ibarra O., « Reversal-bounded multicounter machines and their decision problems », *Journal of the Association for Computing Machinery*, vol. 25, n° 1, p. 116-133, 1978.

Ibarra O., Dang Z., « On removing the stack from reachability constructions », *in* P. Eades, T. Takaoka (eds), *International Symposium on Algorithms and Computation (ISAAC'01)*, vol. 2223 of *Lecture Notes in Computer Science*, Springer, p. 244-256, 2001.

Ibarra O., Su J., Dang Z., Bultan T., Kemmerer A., « Counter Machines : Decidable Properties and Applications to Verification Problems », *MFCS'00*, vol. 1893 of *Lecture Notes in Computer Science*, Springer, p. 426-435, 2000.

Kamp J., Tense Logic and the theory of linear order, PhD thesis, UCLA, USA, 1968.

Kröger F., « On the interpretability of arithmetic in temporal logic », *Theoretical Computer Science*, vol. 73, p. 47-61, 1990.

Lafourcade P., Lugiez D., Treinen R., « Intruder deduction for AC-like equational theories with homomorphisms », *RTA'05*, vol. 3467 of *Lecture Notes in Computer Science*, Springer, p. 308-322, 2005.

Lago U. D., Montanari A., « Calendars, time granularities, and automata », *Int. Symposium on Spatial and Temporal Databases*, vol. 2121 of *Lecture Notes in Computer Science*, Springer, Berlin, p. 279-298, 2001.

Lago U. D., Montanari A., Puppis G., « Towards compact and tractable automaton-based representations of time granularities », *ICTCS 2003*, vol. 2841 of *Lecture Notes in Computer Science*, Springer, Berlin, p. 72-85, 2003.

Leroux J., Sutre G., « Flat counter systems are everywhere ! », *ATVA'05*, vol. 3707 of *Lecture Notes in Computer Science*, Springer, p. 489-503, 2005.

Lichtenstein O., Pnueli A., « Propositional Temporal Logics : Decidability and Completeness », *Logic Journal of the IGPL*, vol. 8, n° 1, p. 55-85, 2000.

Lisitsa A., Potapov I., « Temporal logic with predicate λ-abstraction », *12th Int. Symp. on Temporal Representation and Reasoning, Burlington, Vermont*, IEEE Computer Society Press, p. 147-155, 2005.

Lutz C., « Description Logics with Concrete Domains—A Survey », *Advances in Modal Logics Volume 4*, King's College Publications, p. 265-296, 2003.

Lutz C., « NEXPTIME-complete description logics with concrete domains », *ACM Transactions on Computational Logic*, vol. 5, n° 4, p. 669-705, 2004.

Lutz C., Miličić M., « A Tableau algorithm for description logics with concrete domains and GCIs », *TABLEAUX'05*, vol. 3702 of *Lecture Notes in Computer Science*, Springer, p. 201-216, 2005a.

Lutz C., Walther D., Wolter F., « Quantitative temporal logics : PSPACE and below », *TIME'05*, p. 138-146, 2005b.

Minsky M., *Computation, Finite and Infinite Machines*, Prentice Hall, 1967.

Müller-Olm M., Seidl H., « Analysis of Modular Arithmetic », *ESOP'05*, vol. 3444 of *Lecture Notes in Computer Science*, Springer, p. 46-60, 2005.

Niezette M., Stevenne J., « An efficient symbolic representation of periodic time », *Proc. of the International Conference on Information and Knowledge Management, Baltimore, Maryland*, vol. 752 of *Lecture Notes in Computer Science*, Springer, p. 161-168, 1992.

Ohsaki H., Talbot J., Tison S., Roos Y., « Monotone AC-Tree Automata », *12th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR'05), Montego Bay (Jamaica)*, vol. 3835 of *Lecture Notes in Artificial Intelligence*, Springer, p. 337-351, 2005.

Pacuit E., Salame S., « Majority logic », *KR'04*, AAAI Press, p. 598-605, 2004.

Pnueli A., « The temporal logic of programs », *18th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society Press, p. 46-57, 1977.

Potapov I., « From Post Systems to the Reachability Problems for Matrix Semigroups and Multicounter Automata », *DTL'04*, vol. 3340 of *Lecture Notes in Computer Science*, Springer, p. 345-356, 2004.

Presburger M., « Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt », *Comptes Rendus du premier congrès de mathématiciens des Pays Slaves, Warszawa*, p. 92-101, 1929.

Puppis G., Automata for branching and layered temporal structures, PhD thesis, Universitá Degli Studi Di Udine, March, 2006.

Revesz P., *Introduction to Constraint Databases*, Springer, New York, 2002.

Reynolds M., « The complexity of the temporal logic with until over general linear time », *Journal of Computer and System Sciences*, vol. 66, n° 2, p. 393-426, 2003.

Schnoebelen P., « Verifying lossy channel systems has nonprimitive recursive complexity », *Information Processing Letters*, vol. 83, n° 5, p. 251-261, 2002.

Schnoebelen P., « The complexity of temporal logic model checking », *Advances in Modal Logic, vol. 4, selected papers from 4th Conf. Advances in Modal Logic (AiML'2002), Sep.-Oct. 2002, Toulouse, France*, King's College Publication, p. 437-459, 2003.

Schröder L., Pattinson D., « PSPACE bounds for rank-1 modal logics », *LICS'06*, IEEE, p. 231-240, 2006.

Seidl H., Schwentick T., Muscholl A., Habermehl P., « Counting in Trees for free », *ICALP'04*, vol. 3142 of *Lecture Notes in Computer Science*, p. 1136-1149, 2004. Long version available at `http://www.mathematik.uni-marburg.de/~tick/`.

Serre O., Contribution à l'étude des jeux sur des graphes de processus de pile, PhD thesis, Université Paris 7 - Denis Diderot, 2004.

Serre O., « Parity games played on transition graphs of one-counter processes », *FOSSACS'06*, vol. 3921 of *Lecture Notes in Computer Science*, Springer, p. 337-351, 2006.

Sistla A., Clarke E., « The complexity of propositional linear temporal logic », *Journal of the Association for Computing Machinery*, vol. 32, n° 3, p. 733-749, 1985.

ten Cate B., Franceschet M., « On the complexity of hybrid logics with binders », *CSL'05*, vol. 3634 of *Lecture Notes in Computer Science*, Springer, p. 339-354, 2005.

Toman D., Chomicki J., « DATALOG with integer periodicity constraints », *Journal of Logic Programming*, vol. 35, n° 3, p. 263-290, 1998.

Trahtenbrot B., « Impossibility of an algorithm for the decision problem in finite classes », *American Mathematical Society - Translation Series*, vol. 23, n° 2, p. 1-5, 1963.

van der Hoek W., de Rijke M., « Counting objects », *Journal of Logic and Computation*, vol. 5, n° 3, p. 325-345, 1995.

Vardi M., « A temporal fixpoint calculus », *15th Annual ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages, San Diego*, ACM, p. 250-259, 1988.

Vardi M., Wolper P., « Reasoning about Infinite Computations », *Information and Computation*, vol. 115, p. 1-37, 1994.

Wakatsuki M., Teraguchi K., Tomita E., « Polynomial time identification of strict deterministic restricted one-counter automata in some class from positive data », *International Colloquium on Grammatical Inference (ICGI 2004), Athens*, vol. 3264 of *Lecture Notes in Artificial Intelligence*, Springer, p. 260-272, 2004.

Walukiewicz I., « Pushdown processes : games and model-checking », *Information and Computation*, vol. 164, n° 2, p. 234-263, 2001.

Wijsen J., « A string based-model for infinite granularities », *AAAI Workshop on Spatial and Temporal Granularity*, AAAI Press, p. 9-16, 2000.

Wolper P., « Temporal logic can be more expressive », *Information and Computation*, vol. 56, p. 72-99, 1983.

Wolter F., Zakharyaschev M., « Spatio-temporal representation and reasoning based on RCC-8 », *KR'00*, p. 3-14, 2000.

Wolter F., Zakharyaschev M., « Axiomatizing the monodic fragment of first-order temporal logic », *Annals of Pure and Applied Logic*, 2002.

Zilio S. D., Lugiez D., « XML schema, tree logic and sheaves automata », *Applicable Algebra in Engineering, Communication and Computing (AAECC)*, 2006. To appear.