Verifying qualitative and quantitative properties with LTL over concrete domains

Régis Gascon

Laboratoire Spécification et Vérification
CNRS & ENS Cachan
FRANCE

gascon@lsv.ens-cachan.fr

Abstract. We introduce different extensions of LTL where propositional variables are replaced by constraints interpreted in $\mathbb{Z}$. We show different decidability and complexity results for the satisfiability and model checking problems of these logics. The extension of LTL over a wide set of *qualitative* constraints is shown to be PSPACE-complete. When introducing some *quantitative* constraints, we must consider strong restrictions to regain decidability.

1 Introduction

Model checking counter automata. The verification of infinite-state systems has benefited from the numerous decidable model-checking problems. But though decidability can be obtained via numerous proof techniques, showing undecidability of model-checking is often easier since the halting problem for Minsky machines [Min67] is already undecidable. Decidability is difficult to establish and sometimes it can be regained by naturally restricting the class of models or by considering fragments of the specification language. Symbolic representations of infinite sets of states are often the key argument to get decidability (see for instance [HMR05]).

Structures with a finite set of control states augmented with a finite set of variables interpreted as integers (counter automata) are ubiquitous in computer science. They are used as operational models of numerous infinite-state systems, including broadcast protocols [FL02], and even their restrictions to one counter have found applications to verify cryptographic protocols [LLT05] or to validate XML streams [CR04]. However, the class of counter machines has numerous undecidable model-checking problems such as the reachability problem. This does not end the story since many subclasses admit a decidable reachability problem as reversal-bounded multicounter machines [Iba78] or flat counter systems [Boi98,FL02].

Motivations. Classical problems studied on counter automata include reachability issues and verification of properties on the control states. We aim at checking richer properties by introducing constraints on counters in temporal logic. Defining decidable model-checking problems with such logics is difficult since introducing some *quantitative* constraints (i.e. counting mechanism) like $y = x + 1$ allows to encode a two counter machine. A first alternative is to consider only *qualitative* constraints where the relation between the constrained terms is not sharp, like $y \leq x$ or $y \equiv_k x + 1$. Another option is to consider decidable fragments of undecidable logics by restricting some resources of the formulae. For instance, bounding the number of variables could be a good restriction since we need to store the value of three different counters to encode a two-counter machine (the two counters and the instruction counter).

Contributions. Extensions of the temporal logic LTL over concrete domains have already been considered in [WZ00,BC02,DD02]. We introduce new extensions of LTL over concrete domains where variables are interpreted in $\mathbb{Z}$. The propositional variables are replaced by constraints (induced by the concrete domain) between terms representing the variables at different states of the models. For instance, the constraint $x < \mathsf{XX}x$ means that the current value of x is smaller than the value of x two states further. Such constraints can also be found in description logics over concrete domains [Lut04]. We consider here linear models but the constraint language is very rich.

First, we show the PSPACE-completeness of an extension of LTL over a wide set of qualitative constraints, including periodicity constraints of the form $y \equiv_k x + 1$ and comparisons of the form $y < x$. Such constraints are used for instance in calendar formalisms [LM01] when one needs to consider the periodicity of time or in abstraction of counter automata where operations are defined modulo some power of two [LS01]. Indeed, common programming languages perform arithmetical operations modulo 2^k [MOS05] (k is equal to 16 or 32).

Then we consider another extension introducing quantitative constraints. This full formalism can easily be shown to be undecidable and so we study restrictions of the logic bounding syntactic resources of the formulae. The resources we consider are the number of variables and the distance between the positions (in the model) of the terms that are compared. For example, the distance between $\mathsf{X}x$ and $\mathsf{XXX}x$ is 2. Even with these restrictions the decidability is very difficult to regain. We show that the fragment with one variable and distance two and the fragment with two variables and distance one are undecidable. However, we can prove that the model checking of the fragment with one variable and distance one over one counter automata is PSPACE-complete.

2 LTL over concrete domains

Let $V = \{x_0, x_1, \dots\}$ be a countable infinite set of variables. A *concrete domain* is a pair $\mathcal{D} = \langle D, (R_\alpha)_{\alpha \in I} \rangle$ where D is a specific domain of interpretation for the variables and $(R_\alpha)_{\alpha \in I}$ is a countable family of relations on the elements of D. An *atomic $\mathcal{D}$-constraint* is an expression of the form $\mathsf{R}(x_1, \dots, x_n)$ where R is interpreted as a relation of the domain and n is the arity of this relation. A *D-valuation* is a function $v : V \rightarrow D$ that assigns to every variable a value in

D and a constraint is satisfied by v, denoted by $v \models \mathsf{R}(x_1, \ldots, x_n)$, whenever $(v(x_1), \ldots, v(x_n)) \in R$, R being the relation in $\mathcal{D}$ associated to the symbol R.

We define the logic CLTL($\mathcal{D}$) as the extension of LTL where the propositional variables are replaced by atomic $\mathcal{D}$-constraints over *terms* representing different states of the variables, denoted by $\mathsf{R}(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_n}x_n)$. The formulae of CLTL($\mathcal{D}$) are defined by the following grammar:

$$\phi ::= \mathsf{R}(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k) \mid \phi \wedge \phi \mid \neg\phi \mid \mathsf{X}\phi \mid \phi\mathsf{U}\phi.$$

The symbols X and U are the classical operators next and until of LTL. A term $\mathsf{X}^i x$ (abbreviation for $\mathsf{X}\ldots\mathsf{X}x$) represents the value of x at the i^{th} next state. A *one-step constraint* is an atomic formula of the form $\mathsf{R}(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)$ such that $i_1, \ldots, i_k \leq 1$. Given a 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 a frame of consecutive states that can be compared. The models of CLTL($\mathcal{D}$) are sequences of D-valuations $\sigma : \mathbb{N} \to (V \to D)$ and the satisfaction relation is defined like the LTL satisfaction relation except at the atomic level:

- $\sigma, i \models \mathsf{R}(\mathsf{X}^{j_1}x_1, \ldots, \mathsf{X}^{j_n}x_n)$ iff $(\sigma(i+j_1)(x_1), \ldots, \sigma(i+j_n)(x_n)) \in R$,
- $\sigma, i \models \phi \wedge \phi'$ 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 $j \geq i$ s.t. $\sigma, j \models \phi'$ and for every $i \leq k < j$, $\sigma, k \models \phi$.

As usual, a formula $\phi \in$ CLTL($\mathcal{D}$) is satisfiable whenever there exists a model σ such that $\sigma, 0 \models \phi$. The *satisfiability problem* takes as input a formula and checks whether it is satisfiable.

The *model checking problem* is defined for the class of $\mathcal{D}$-*automata* that are Büchi automata with transitions labeled by one-step constraints. Formally, 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 $Q \times 1SC_k \times Q$ where $1SC_k$ is the set of Boolean combinations of one-step constraints built over the set of variables $\{x_1, \ldots, x_k\}$. $\mathcal{D}$-automata have no input alphabet: we do not use them as language acceptors but we are rather interested in the behaviors of such systems. A *configuration* of $\mathcal{A}$ is a tuple $\langle q, \boldsymbol{c}\rangle \in Q \times D^k$. Let $\boldsymbol{c}_{[i]}$ denote the i^{th} value of $\boldsymbol{c}$. The one-step transition relation $\to \subseteq (Q \times D^k)^2$ is defined by: $\langle q, \boldsymbol{c}\rangle \to \langle q', \boldsymbol{c'}\rangle \overset{\text{def}}{\Leftrightarrow}$ there is a transition $\langle q, \varphi, q'\rangle \in \delta$ such that $[x_1 \leftarrow \boldsymbol{c}_{[1]}, \ldots, x_k \leftarrow \boldsymbol{c}_{[k]}, \mathsf{X}x_1 \leftarrow \boldsymbol{c'}_{[1]}, \ldots, \mathsf{X}x_k \leftarrow \boldsymbol{c'}_{[k]}] \models \varphi$.

An infinite path w is a map $\mathbb{N} \to (Q \times D^k)$ such that for every $i \in \mathbb{N}$, we have $w(i) \to w(i+1)$. An accepting run for $\mathcal{A}$ is an infinite path w such that $w(0)$ is of the form $\langle q, \boldsymbol{c}\rangle$ with $q \in I$ and the set of configurations in w of the form $\langle q_f, \boldsymbol{c}\rangle$ with $q_f \in F$ is infinite.

Given an infinite path w, let σ_w be the CLTL($\mathcal{D}$) model such that for all $i \in \mathbb{N}$ and $j \in \{1, \ldots, k\}$, $\sigma_w(i)(x_j) = \boldsymbol{c}_{[j]}$ where $w(i) = \langle q, \boldsymbol{c}\rangle$. The model-checking problem for CLTL($\mathcal{D}$) takes as input a CLTL($\mathcal{D}$) formula ϕ and a $\mathcal{D}$-automaton $\mathcal{A}$ and checks whether there is an accepting run w of $\mathcal{A}$ such that

$\sigma_w, 0 \models \phi$. It is easy to show that the satisfiability problem can be reduced to the model-checking problem.

3 Verifying qualitative constraints

We introduce in this section an extension of LTL over a concrete domain with qualitative constraints only. By qualitative, we mean that the relation between the constrained terms is non deterministic. We distinguish two types of constraints: comparisons and periodicity constraints that are congruence modulo relations. We consider the language of constraints IPC*, which is an extension of the language IPC of [Dem04], defined as following:

$$\begin{array}{rl} \varphi ::= & \varphi_{mod} \mid x < y \mid \varphi \wedge \varphi \mid \neg \varphi \\ \varphi_{mod} ::= & x \equiv_k [c_1, c_2] \mid x \equiv_k y + [c_1, c_2] \mid x = y \mid x < d \mid x = d \mid \\ & \varphi_{mod} \wedge \varphi_{mod} \mid \neg\, \varphi_{mod} \mid \exists x\ \varphi_{mod} \end{array}$$

where $x, y \in V$, $k \in \mathbb{N} \setminus \{0\}$, $c_1, c_2 \in \mathbb{N}$ and $d \in \mathbb{Z}$. The models of IPC* are $\mathbb{Z}$-valuations of the form $v : V \to \mathbb{Z}$ and the satisfaction relation is defined by:

- $v \models x \sim y \overset{\text{def}}{\Leftrightarrow} v(x) \sim v(y)$ and $v \models x \sim d \overset{\text{def}}{\Leftrightarrow} v(x) \sim d$ where $\sim \in \{<, =\}$,
- $v \models x \equiv_k [c_1, c_2] \overset{\text{def}}{\Leftrightarrow} v(x)$ equals c modulo k for some $c_1 \leq c \leq c_2$,
- $v \models x \equiv_k y + [c_1, c_2] \overset{\text{def}}{\Leftrightarrow} v(x) - v(y)$ equals c modulo k for some $c_1 \leq c \leq c_2$,
- $v \models p \wedge p' \overset{\text{def}}{\Leftrightarrow} v \models p$ and $v \models p'$,
- $v \models \neg p \overset{\text{def}}{\Leftrightarrow}$ not $v \models p$,
- $v \models \exists x\ p \overset{\text{def}}{\Leftrightarrow}$ there exists $z \in \mathbb{Z}$ such that $v[x \leftarrow z] \models p$ where $v[x \leftarrow z]$ is defined by $v[x \leftarrow z](x') = v(x')$ for every $x \neq x'$ and $v[x \leftarrow z](x) = z$.

In the following, a set of constraints is satisfiable if there is a valuation satisfying all its elements.

We write CLTL(IPC*) to denote the extension of LTL over the concrete domain induced by the constraint language IPC*. For instance, $\mathsf{X}x \equiv_5 \mathsf{X}y + [0, 2]$ or $\mathsf{X}x < x$ are one-step constraints of CLTL(IPC*) and $(\mathsf{X}x < x)\mathsf{U}(x = 0)$ is a CLTL(IPC*) formula. The satisfiability and model-checking problems for CLTL(IPC*) are reducible to each other in logarithmic space following techniques from [SC85]. So the results below about the satisfiability problem also extend to the model-checking problem. Note that extending IPC* by allowing constraints of the form $x < y$ in the scope of $\exists$ leads to undecidability since the successor relation for integers is then definable and the halting problem for Minsky machines can be easily encoded by a formula of the corresponding extension of LTL.

Following the approach in [VW94], we have shown in [DG05] that given a CLTL(IPC*) formula ϕ, we can build a standard Büchi automaton $\mathcal{A}_\phi$ such that ϕ is satisfiable iff $\mathrm{L}(\mathcal{A}_\phi)$ is non-empty. Moreover, checking emptiness of $\mathrm{L}(\mathcal{A}_\phi)$ can be done in polynomial space in $|\phi|$.

Symbolic models. The language recognized by the Büchi automaton $\mathcal{A}_\phi$ is not a set of models but a set of symbolic models. We briefly explain below how we define a symbolic representation of the models (see details in [DG05]). Given a formula ϕ, a symbolic valuation is a set of constraints depending of the least common multiple of the integers k_i such that a relation $\equiv_{k_i}$ occurs in ϕ, the set of constants d occurring in constraints of the form $x \sim d$ and the finite set of variables occurring in ϕ. This symbolic representation of a $\mathbb{Z}$-valuation contains all the relevant information to evaluate constraints. Symbolic valuations define an equivalence relation between $\mathbb{Z}$-valuations, like regions for timed automata.

Let ρ be an infinite symbolic valuation sequence. We say that ρ is satisfiable iff there is a model $\sigma : \mathbb{N} \to (V \to \mathbb{Z})$ such that for every $i \in \mathbb{N}$ we have $\sigma, i \models \rho(i)$ (and we note $\sigma \models \rho$). A symbolic model for the formula ϕ is a *one-step consistent* sequence of symbolic valuation wrt ϕ. A sequence is one-step consistent if the constraints are propagated along the sequence: for instance if $\mathsf{X}^a x \sim \mathsf{X}^b y + d$ is a constraint of $\rho(i)$ and both $a > 0$ and $b > 0$, it must be the case that the constraint $\mathsf{X}^{a-1} x \sim \mathsf{X}^{b-1} y + d$ occurs at the next position of the sequence (i.e. $\rho(i+1)$). We need this property to eliminate some irrelevant sequences since non one-step consistent sequences are not satisfiable. We define a symbolic satisfaction relation between symbolic models and formulae denoted $\rho \models_{\text{symb}} \phi$. In order to prove the decidability we use the fact below:

Lemma. *A* CLTL(IPC*) *formula ϕ is satisfiable iff there are a symbolic model ρ and a model σ such that $\sigma \models \rho$ and $\rho \models_{\text{symb}} \phi$.*

Approximation. The main difficulty in the construction of $\mathcal{A}_\phi$ is to characterize by means of automata the set of satisfiable symbolic models (one-step consistence is not enough), which is not ω-regular. We introduce an ω-regular condition $(\mathcal{C})$ (see definition in [DG05]) that over-approximate this set: every satisfiable symbolic model verifies $(\mathcal{C})$ but some symbolic models verifying $(\mathcal{C})$ are not satisfiable. However this condition is equivalent to satisfiability whenever the symbolic model is *ultimately periodic*. A symbolic model is ultimately periodic if it is of the form $\gamma \cdot (\delta)^\omega$ (the suffix δ is repeated infinitely often).

Construction of the automaton. The automaton $\mathcal{A}_\phi$ is defined as the intersection $\mathcal{A}_{\text{LTL}} \cap \mathcal{A}_\mathcal{C}$ of Büchi automata where $\mathrm{L}(\mathcal{A}_{\text{LTL}})$ is the set of symbolic models satisfying ϕ and $\mathrm{L}(\mathcal{A}_\mathcal{C})$ is the set of symbolic models verifying $(\mathcal{C})$. We can prove that a CLTL(IPC*) formula ϕ is satisfiable iff $\mathrm{L}(\mathcal{A}_\phi)$ is non-empty. Like in [DD02], the main trick of the proof is that if $\mathrm{L}(\mathcal{A}_\phi)$ is non-empty then there exists an ultimately periodic ω-sequence $\rho \in \mathrm{L}(\mathcal{A}_\phi)$. Thus ρ is satisfiable since it is ultimately periodic and verifies $(\mathcal{C})$.

Given a formula ϕ we can effectively build $\mathcal{A}_\phi$ and check whether $\mathrm{L}(\mathcal{A}_\phi)$ is empty. So the satisfiability problem for CLTL(IPC*) is decidable. We can also establish the PSPACE upper bound by using the fact that all the components of the automaton can be built in PSPACE and subtle arguments from complexity theory and [Saf89].

Theorem. [DG05] *The satisfiability problem and the model checking problem for the logic* CLTL(IPC*) *are* PSPACE*-complete.*

Since the operators of CLTL(IPC*) are definable in the monadic second order logic (MSO), we can also prove using [GK03] that any extension of CLTL(IPC*) obtained by adding MSO-definable operators remains in PSPACE. We only need to update the automaton $\mathcal{A}_{\mathrm{LTL}}$.

4 Verifying quantitative constraints

In this section, we introduce another instance of LTL over a concrete domain where the variables are interpreted in $\mathbb{Z}$ and the underlying constraint language is a fragment of Presburger arithmetic. The main difference with IPC* is the introduction of quantitative constraints of the form $x = y + d$ where $d \in \mathbb{Z}$. We show that such constraints easily lead to undecidability, even when restricting strongly the syntactic resources of the formulae. More details will be available in [DG]. The constraint language $\mathcal{P}_r$ is defined by:

$$\varphi ::= x \sim y + d \mid x \sim d$$

where $x, y \in V$, $\sim \in \{<, =\}$ and $d \in \mathbb{Z}$. The variables of $\mathcal{P}_r$ are interpreted in $\mathbb{Z}$ and given a valuation $v : V \to \mathbb{Z}$, the satisfaction relation $v \models \varphi$ is defined in the obvious way. Like in the previous section, the extension of LTL over the concrete domain induced by $\mathcal{P}_r$ is denoted by CLTL($\mathcal{P}_r$).

Counter Automata. The class of $\mathcal{P}_r$-automata contains many known classes of counter automata and can easily simulate two-counter non-deterministic Minsky machines. A k-counter automaton is a $\mathcal{P}_r$-automaton such that the constraints on transitions are conjunctions of constraints expressing zero tests, sign tests and updates of the counters of the form $\mathsf{X}x = x + d$ where $d \in \{-1, 0, 1\}$. The existence of an accepting run for one-counter automata is a more difficult question than similar questions studied for instance in [LLT05] since we deal with a Büchi acceptance condition, the counter is interpreted in $\mathbb{Z}$ and zero test and sign test are allowed.

Decidability status. Full CLTL($\mathcal{P}_r$) can easily be shown to be undecidable. We consider restrictions of the syntactic resources of the formulae while preserving the full strength of the logical operators. We denote with CLTL^l_k($\mathcal{P}_r$) the fragment of CLTL($\mathcal{P}_r$) where the number of variables of the formulae is bounded by k and the X-length by l. Known results show that the fragments CLTL$^\omega_2$($\mathcal{P}_r$) and CLTL$^1_3$($\mathcal{P}_r$) have undecidable satisfiability problems (see [CC00] and [DD02]). We refine these undecidability results: the decidability status of the different fragments are shown in the following table (our results are in bold characters):

$k \setminus l$	1	2	ω
1	**pspace-complete**	**undecidable**	**undecidable**
2	**undecidable**	undecidable	undecidable [DD02]
3	undecidable [CC00]	undecidable	undecidable

We show that bounding the number of variables is not enough to regain decidability by proving the undecidability of $\mathrm{CLTL}_1^{\omega}(\mathcal{P}_r)$. We also establish sharp decidability limits of these fragments by showing the undecidability of $\mathrm{CLTL}_1^2(\mathcal{P}_r)$ and $\mathrm{CLTL}_2^1(\mathcal{P}_r)$. These undecidability results can be proved by encoding two-counter automata and reducing different fragments of the logic.

To complete the picture, we show that the fragment $\mathrm{CLTL}_1^1(\mathcal{P}_r)$ is PSPACE-complete. As in the previous section, the proof relies on the construction of a particular one-counter automaton recognizing symbolic representations of the models. This complexity result uses an auxiliary result about reachability in one-counter automata. We prove that the existence of an accepting run in such automata can be decided in NLOGSPACE.

To conclude, we mention that the extension $\mathcal{P}_r'$ of $\mathcal{P}_r$ with constraints of the form $ax + by = 0$ where $a, b \in \mathbb{Z}$ leads to undecidability even for the fragment $\mathrm{CLTL}_1^1(\mathcal{P}_r')$. 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 are encoded by constraints of the form $x \equiv_2 0$, $x \equiv_3 0$, $\mathsf{X}x = 2x$ (incrementation of the first counter) etc. So, as far as decidability is concerned, the underlying fragment of Presburger arithmetic in $\mathrm{CLTL}(\mathcal{P}_r)$ is quite optimal with respect to the restrictions we consider.

References

[BC02] Ph. Balbiani and J.F. Condotta. Computational complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning. In *FroCoS'02*, volume 2309 of *LNAI*, pages 162–173. Springer, 2002.

[Boi98] B. Boigelot. *Symbolic methods for exploring infinite state spaces.* PhD thesis, Université de Liège, 1998.

[CC00] H. Comon and V. Cortier. Flatness is not a weakness. In *CSL'00*, volume 1862 of *LNCS*, pages 262–276. Springer, 2000.

[CR04] C. Chitic and D. Rosu. On validation of XML streams using finite state machines. Technical Report CSRG-489, University of Toronto, 2004.

[DD02] S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. In *FST&TCS'02*, volume 2256 of *LNCS*, pages 121–132. Springer, 2002.

[Dem04] S. Demri. LTL over integer periodicity constraints. In *FOSSACS'04*, volume 2987 of *LNCS*, pages 121–135. Springer, 2004.

[DG] S. Demri and R. Gascon. The Effects of Bounding Syntactic Resources on Presburger LTL. Submitted.

[DG05] S. Demri and R. Gascon. Verification of qualitative $\mathbb{Z}$-constraints. In *CONCUR'05*, volume 3653 of *LNCS*, pages 518–532. Springer, 2005.

[FL02] A. Finkel and J. Leroux. How to compose Presburger accelerations: Applications to broadcast protocols. In *FST&TCS'02*, volume 2256 of *LNCS*, pages 145–156. Springer, 2002.

[GK03] P. Gastin and D. Kuske. Satisfiability and model checking for MSO-definable temporal logics are in PSPACE. In *CONCUR'03*, volume 2761 of *LNCS*, pages 222–236. Springer, 2003.

[HMR05] Th. Henzinger, R. Majumdar, and J.F. Raskin. A classification of symbolic transitions systems. *ACM Transactions on Computational Logic*, 6(1):1–32, 2005.

[Iba78] O. Ibarra. Reversal-bounded multicounter machines and their decision problems. *JACM*, 25(1):116–133, 1978.

[LLT05] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. In *RTA'05*, volume 3467 of *LNCS*, pages 308–322. Springer, 2005.

[LM01] U. Dal Lago and A. Montanari. Calendars, time granularities, and automata. In *Int. Symposium on Spatial and Temporal Databases*, volume 2121 of *LNCS*, pages 279–298. Springer, 2001.

[LS01] G. Logothetis and K. Schneider. Abstraction from counters: an application on real-time systems. In *TIME'01*, pages 214–223. IEEE, 2001.

[Lut04] C. Lutz. NEXPTIME-complete description logics with concrete domains. *ACM Transactions on Computational Logic*, 5(4):669–705, 2004.

[Min67] M. Minsky. *Computation, Finite and Infinite Machines.* Prentice Hall, 1967.

[MOS05] M. Müller-Olm and H. Seidl. Analysis of modular arithmetic. In *ESOP'05*, volume 3444 of *LNCS*, pages 46–60. Springer, 2005.

[Saf89] S. Safra. *Complexity of Automata on Infinite Objects*. PhD thesis, The Weizmann Institute of Science, 1989.

[SC85] A. Sistla and E. Clarke. The complexity of propositional linear temporal logic. *JACM*, 32(3):733–749, 1985.

[VW94] M. Vardi and P. Wolper. Reasoning about infinite computations. *I & C*, 115:1–37, 1994.

[WZ00] F. Wolter and M. Zakharyaschev. Spatio-temporal representation and reasoning based on RCC-8. In *KR'00*, pages 3–14, 2000.