Temporal Logics with Local Constraints*

Claudia Carapelle[1] and Markus Lohrey[2]

1 **University of Leipzig, Germany**
`carapelle@informatik.uni-leipzig.de`

2 **University of Siegen, Germany**
`lohrey@eti.uni-siegen.de`

Abstract

Recent decidability results on the satisfiability problem for temporal logics, in particular LTL, CTL^* and ECTL^*, with constraints over external structures like the integers with the order or infinite trees are surveyed in this paper.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Temporal logics with constraints, concrete domains, LTL, CTL^*, ECTL^*

Digital Object Identifier 10.4230/LIPIcs.CSL.2015.2

Category Invited Talk

1 Linear Time Temporal Logic with Constraints

Temporal logics are a very popular family of logical languages, used to specify properties of abstracted systems. Pnueli [27] was the first who used linear temporal logic, briefly denoted by LTL, for reasoning about reactive systems. Since then, LTL has become one of the most prominent specification languages used in verification and model checking. Both, model-checking and satisfiability for LTL are PSPACE-complete [28].

In the last few years, many extensions of temporal logics have been proposed in order to address the need to express more than just abstract properties, see for instance [2, 4, 16, 17, 32]. In some of these studies we can find languages which allow to reason about time intervals, space regions, data values from dense domains like the real numbers or discrete domains like the integers or natural numbers.

A general approach for creating such formalisms is described by Demri and D'Souza in [15], where they show how to extend LTL with the ability to express properties of data values from an arbitrary relational structure $\mathcal{D}$, which is often called a *concrete domain*. An example of a concrete domain can be $(\mathbb{Z}, <)$, whose universe is the set $\mathbb{Z}$ of integers and $<$ is the standard linear order on $\mathbb{Z}$, viewed as a binary relation. The approach from [15] is also used in the field of description logics (DLs), where Baader and Hanschke first described a way to integrate arbitrary concrete domains into the knowledge-representation language $\mathcal{ALC}$ [3].

The logic defined in [15] is called constraint LTL, briefly CLTL. The idea behind this language is the following: Fix a set of variables $\mathcal{X}$ and another one of propositions $\mathcal{P}$, both countably infinite, for the rest of the paper. Moreover, fix a relational signature σ, which is a set of relational symbols R, each having an arity a_R. We assume that σ is either finite or countably infinite. A σ-structure is a tuple $\mathcal{D} = (D, (R^{\mathcal{D}})_{R\in\sigma})$, where $R^{\mathcal{D}} \subseteq D^{a_R}$ is a relation

* This work was partially supported by the DFG Research Training Group 1763 (QuantLA).

of arity a_R. In the following, we always identify the relational symbol R with the associated relation $R^{\mathcal{D}}$ if the structure $\mathcal{D}$ is clear from the context. Then the set of CLTL-formulas over $\mathcal{D}$ is defined by the following syntax, where $p \in \mathcal{P}$, $R \in \sigma$, $k = a_R$, $i_1, \ldots, i_k \in \mathbb{N}$, and $x_1, \ldots, x_k \in \mathcal{X}$:

$$\varphi \ ::= \ p \ \mid \ R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k) \ \mid \ \neg\varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \mathsf{X}\varphi \ \mid \ \varphi \mathsf{U} \varphi \tag{1}$$

A formula of the form

$$R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k) \tag{2}$$

is called a $\mathcal{D}$-*constraint*, or simply constraint if $\mathcal{D}$ is clear from the context. We do not assume that the variables $x_1, \ldots, x_k$ are pairwise distinct. A CLTL-formula over $\mathcal{D}$ is interpreted over an infinite word

$$w = (A_0, \eta_0)(A_1, \eta_1)(A_2, \eta_2) \cdots, \tag{3}$$

where for $i \geq 0$, $A_i \subseteq \mathcal{P}$ is a set of propositions and $\eta_i : \mathcal{X} \to D$ assigns a value from D to every variable. One can think of $\mathcal{D}$-registers attached to the system states. Words of the form (3) are also known as *multi-data words*. For $i \geq 0$ we define the suffix $w[i:]$ as the multi-data word $(A_i, \eta_i)(A_{i+1}, \eta_{i+1})(A_{i+2}, \eta_{i+2}) \cdots$.

The satisfaction relation $w \models \varphi$ where $w = (A_0, \eta_0)(A_1, \eta_1)(A_2, \eta_2) \cdots$ is a multi-data word and φ is a CLTL-formulas over $\mathcal{D}$ is inductively defined as follows (all cases except for the case that φ is a constraint of the form (2) are as for ordinary LTL without constraints):

- $w \models p$ iff $p \in A_0$ for $p \in \mathcal{P}$.
- $w \models R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)$ iff $(\eta_{i_1}(x_1), \ldots, \eta_{i_k}(x_k)) \in R$.
- $w \models \neg\varphi$ iff $w \models \varphi$ does not hold.
- $w \models \varphi_1 \wedge \varphi_2$ iff $w \models \varphi_1$ and $w \models \varphi_2$.
- $w \models \mathsf{X}\varphi$ iff $w[1:] \models \varphi$.
- $w \models \varphi_1 \mathsf{U} \varphi_2$ iff there is an $i \geq 0$ with $w[i:] \models \varphi_2$ and $w[j:] \models \varphi_1$ for all $0 \leq j \leq i-1$.

▶ **Example 1.** Take the structure $\mathcal{D} = (\mathbb{Z}, <, =, (=_a)_{a \in \mathbb{Z}})$, where $<$ is the order relation defined above, and $=_a$ is the unary predicate that only holds for a. Instead of $=_a(x)$ we write $x = a$. The CLTL-formula $(x < \mathsf{X}^1 y) \, \mathsf{U} \, (y = 100)$ holds on a multi-data word if and only if there is a position where variable y holds the value 100 and for all previous positions t, the value of x at position t is strictly smaller than the value of y at position $t+1$.

2 Satisfiability for Linear Time Temporal Logic with Constraints

A CLTL-formula φ over $\mathcal{D}$ is satisfiable if there exists a multi-data word w of the form (3) such that $w \models \varphi$. Of course, if $\mathcal{P}_\varphi \subseteq \mathcal{P}$ (resp. $\mathcal{X}_\varphi \subseteq \mathcal{X}$) is the finite set of propositions (resp., variables) that occur in φ then we can assume that $A_i \subseteq \mathcal{P}_\varphi$ and $\eta_i : \mathcal{X}_\varphi \to D$ in (3).

Balbiani and Condotta [4] proved a general decidability result for CLTL over concrete domains $\mathcal{D}$ satisfying certain properties. The following outline follows [15], where the result of Balbiani and Condotta is reproved in an automata theoretic framework. First of all, let us fix a concrete domain $\mathcal{D} = (D, R_1, \ldots, R_n)$ with only finitely many relations. If $i_j = 0$ for all $1 \leq j \leq k$ in (2), then we call the $\mathcal{D}$-constraint a *point $\mathcal{D}$-constraint* (since it refers to one time point). For a point $\mathcal{D}$-constraint $R(x_1, \ldots, x_k)$ and a mapping $\eta : V \to D$, where $x_1, \ldots, x_k \in V \subseteq \mathcal{X}$ we write $\eta \models R(x_1, \ldots, x_k)$ if $(\eta(x_1), \ldots, \eta(x_k)) \in R$. Given a finite subset $V \subseteq \mathcal{X}$ of variables and a mapping $\eta : V \to D$ we denote with $\mathsf{frame}(V, \eta)$ the set of all constraints $R(x_1, \ldots, x_k)$ with $x_1, \ldots, x_k \in V$ and $\eta \models R(x_1, \ldots, x_k)$. A *frame* over the finite subset $V \subseteq \mathcal{X}$ is a set of constraints of the form $\mathsf{frame}(V, \eta)$ for some mapping $\eta : V \to \mathcal{D}$. In other words, a frame over V is a maximal set of constraints which is still satisfiable. We say that *frame-checking* is decidable for $\mathcal{D}$ if there exists an algorithm, whose input is a finite set of constraints C and which checks, whether C is a frame.

For a set of constraints C and a set of variables $U \subseteq \mathcal{X}$ we denote with $C\restriction_U \subseteq C$ the set of all constraints $R(x_1, \ldots x_k) \in C$ such that $x_1, \ldots, x_k \in U$. A structure $\mathcal{D}$ has the *completion property*, if for every frame C over V, every subset $V' \subseteq V$, and every mapping $\eta' : V' \to D$ such that $C\restriction_{V'} = \mathsf{frame}(V', \eta')$ there exists an extension η of η' (meaning that $\eta'(x) = \eta(x)$ for all $x \in V'$) such that $C = \mathsf{frame}(V, \eta)$.

Now we can present the result of Balbiani and Condotta [4] in the form stated in [15]:

▶ **Theorem 2** ([4, 15]). *Let $\mathcal{D} = (D, R_1, \ldots, R_n)$ be a structure having the completion property. If frame-checking is decidable (resp., in* PSPACE*), then satisfiability for* CLTL *over $\mathcal{D}$ is decidable (resp.,* PSPACE*-complete).*

Recall that satisfiability for ordinary LTL (without constraints) is already PSPACE-complete. For the proof of Theorem 2 one follows the classical translation of an LTL-formula (without constraints) to a Büchi automaton. In addition to a set of subformulas, the Büchi automaton also has to store a frame over the variables appearing in the CLTL-formula. Along its run, the Büchi automaton checks whether successive frames fit together in the sense that they can be extended to a single frame. Moreover, the automaton checks whether the constraints that belong to the current set of subformulas hold in that combined frame (for this, one has to assume that $i_1, \ldots, i_k \leq 1$, which indeed can be enforced by adding further variables). The resulting Büchi automaton accepts a non-empty language if and only if the CLTL-formula is satisfiable. The completion property ensures the correctness of the construction.

Instances of domains with the completion property and decidable frame-checking are $(D, <, =)$ with $D = \mathbb{R}$ or $D = \mathbb{Q}$, and $(\mathbb{R}^2, sw, s, se, w, e, nw, n, ne, =)$, where the nine relations illustrate the mutual position of two points in the Cartesian plane (eg. (a, b) sw (c, d) iff $a < c$ and $b < d$). In these cases, the dense structure of the real and rational numbers is fundamental for the completion property. On the other hand, the structures $(\mathbb{N}, <)$ and $(\mathbb{Z}, <)$ do not have the completion property: Take for instance the frame consisting of the constraints $x < y, y < z, x < z$. Then the mapping $\eta' : \{x, z\} \to \mathbb{N}$ with $\eta'(x) = 1$ and $\eta'(z) = 2$ cannot be extended to a mapping $\eta : \{x, y, z\} \to \mathbb{N}$ such that $\mathsf{frame}(\{x, y, z\}, \eta) = \{x < y, y < z, x < z\}$. In fact, it was shown in [15] that a structure $(D, <, =)$ where $(D, <)$ is an infinite linear order satisfies the completion property if and only if $(D, <)$ is dense and has neither a smallest nor a largest element. This originated the question whether satisfiability of CLTL over $(\mathbb{Z}, <, =)$ or $(\mathbb{N}, <, =)$ is still decidable. Demri and D'Souza [15] finally answered this question positively (as in Example 1, $=_a$ denotes the unary relation $\{a\}$):

▶ **Theorem 3** ([15]). *Satisfiability for* CLTL *over the structures $(\mathbb{Z}, <, =, (=_a)_{a\in\mathbb{Z}})$ and $(\mathbb{N}, <, =, (=_a)_{a\in\mathbb{N}})$ is* PSPACE*-complete.*

In [16], Demri and Gascon extend this result to CLTL with so called IPC*-constraints. If we disregard succinctness aspects, this logic is equivalent to CLTL over the structure

$$\mathcal{Z} = (\mathbb{Z}, <, =, (=_a)_{a\in\mathbb{Z}}, (\equiv_{a,b})_{0\leq a<b}), \tag{4}$$

where $\equiv_{a,b}$ denotes the unary relation $\{a + xb \mid x \in \mathbb{Z}\}$ (expressing that an integer is congruent to a modulo b). The main result from [16] states that satisfiability of CLTL with IPC*-constraints is still PSPACE-complete.

Constraints over the above structure (4) do not allow to express the successor relation $y = x + 1$, which would be very useful for analyzing counter systems. There is a good reason for this: Using successor constraints it is easy to reduce the halting problem (and even the Σ_1^1-complete recurrent reachability problem) for two-counter machines to the satisfiability problem for CLTL over $(\mathbb{Z}, \{(x, x+1) \mid x \in \mathbb{Z}\})$. In [15] the authors extend this observation by showing undecidability of satisfiability for CLTL over every structure with a so called *implicit counting mechanism*. A relational structure $\mathcal{D}$ with universe D has an implicit counting mechanism if it contains the equality relation and a binary relation R such that (i) $R = \{(x, y) \in D \times D \mid f(x) = y\}$, where $f : D \to D$ is injective and (ii) (D, R) is acyclic.

▶ **Theorem 4** ([15]). *If $\mathcal{D}$ has an implicit counting mechanism, then satisfiability for* CLTL *over $\mathcal{D}$ is hard for Σ_1^1 (the first existential level of the analytical hierarchy).*

Using the structure $\mathcal{Z}$ from (4) one can still specify an abstracted version of increment operations. For example $x = y+1$ can be abstracted by $(y > x) \wedge \bigvee_{i=-2^k}^{2^k-1} (\equiv_{i,2^k}(x) \wedge \equiv_{i+1,2^k}(y))$ where k is a large natural number. This is why CLTL over $\mathcal{Z}$ seems to be a good compromise between (unexpressive) total abstraction and (undecidable) high concretion.

3 Branching Time Temporal Logics with Constraints

In the same way as outlined for LTL in Section 1, constraints can be also added to branching time logics like CTL* and even ECTL* (extended computation tree logic), obtaining CCTL* and CECTL*, respectively. Formulas from these logics are interpreted over *decorated* Kripke structures. Fix again a σ-structure $\mathcal{D} = (D, (R^{\mathcal{D}})_{R \in \sigma})$. A $\mathcal{D}$-decorated Kripke structure is a tuple $\mathcal{K} = (V, R, \lambda, \zeta)$, where V is the set of nodes (or states), $R \subseteq V \times V$ is a binary edge relation such that for every $v \in V$ there exists $v' \in V$ with $(v, v') \in R$, and for every node $v \in V$, $\lambda(v) \subseteq \mathcal{P}$ is the set of propositions that hold in v, whereas $\zeta(v) : \mathcal{X} \to D$ assigns values from D to the variables (or registers). Instead of $(\zeta(v))(x)$ we also write $\zeta(v, x)$. For $v \in V$, a $(\mathcal{K}, v)$-path is an infinite sequence of nodes $\rho = (v_0, v_1, v_2, \ldots)$ such that $v_0 = v$ and $(v_i, v_{i+1}) \in R$ for $i \geq 0$. As for multi-data words we define $\rho[i:] = (v_i, v_{i+1}, v_{i+2}, \ldots)$.

The syntax of CCTL* over $\mathcal{D}$ is given by the following grammar, where $p \in \mathcal{P}$, $R \in \sigma$, $k = a_R$ is the arity of R, $i_1, \ldots, i_k \in \mathbb{N}$, and $x_1, \ldots, x_k \in \mathcal{X}$:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathsf{E}\psi \tag{5}$$
$$\psi \quad ::= \quad \varphi \mid R(\mathsf{X}^{i_1} x_1, \ldots, \mathsf{X}^{i_k} x_k) \mid \neg\psi \mid \psi \wedge \psi \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi \tag{6}$$

Formulas of the form (5) are state formulas and are interpreted over nodes of a $\mathcal{D}$-decorated Kripke structure, whereas formulas of the form (6) are path formulas and are interpreted over paths in $\mathcal{K}$. Note that constraints are path formulas. Here is the inductive definition of the satsifiability relation, where $\mathcal{K} = (V, R, \lambda, \zeta)$ is a $\mathcal{D}$-decorated Kripke structure, $v \in V$ and $\rho = (v_0, v_1, v_2, \ldots)$ is a $(\mathcal{K}, v_0)$-path (we omit here the obvious cases for the boolean operators $\neg$ and $\wedge$):

- $(\mathcal{K}, v) \models p$ iff $p \in \lambda(v)$ for $p \in \mathcal{P}$.
- $(\mathcal{K}, v) \models \mathsf{E}\psi$ iff there is a $(\mathcal{K}, v)$-path ϱ such that $(\mathcal{K}, \varrho) \models \psi$.
- $(\mathcal{K}, \rho) \models \varphi$ if $(\mathcal{K}, v_0) \models \varphi$ for a state formula φ.
- $(\mathcal{K}, \rho) \models R(\mathsf{X}^{i_1} x_1, \ldots, \mathsf{X}^{i_k} x_k)$ if $(\zeta(v_{i_1}, x_1), \ldots, \zeta(v_{i_k}, x_k)) \in R$.
- $(\mathcal{K}, \rho) \models \mathsf{X}\varphi$ iff $(\mathcal{K}, \rho[1:]) \models \varphi$.
- $(\mathcal{K}, \rho) \models \varphi_1 \mathsf{U} \varphi_2$ iff there is an $i \geq 0$ with $(\mathcal{K}, \rho[i:]) \models \varphi_2$ and $(\mathcal{K}, \rho[j:]) \models \varphi_1$ for all $0 \leq j \leq i-1$.

A CCTL* state formula ψ over $\mathcal{D}$ is satisfiable if there exists a $\mathcal{D}$-decorated Kripke structure $\mathcal{K}$ and a node v from $\mathcal{D}$ such that $(\mathcal{K}, v) \models \psi$.

A weak form of CCTL* over $\mathcal{Z}$, where only integer variables at the same state can be compared, was first introduced in [13] and used to describe properties of so called relational automata. It was shown in [13] that the model checking problem for the above fragment of CCTL* over relational automata is undecidable.

Demri and Gascon [16] asked whether satisfiability of CCTL* over the structure $\mathcal{Z}$ from (4) is decidable. This problem was further investigated in [6, 20], where several partial results where shown: If we replace in $\mathcal{Z}$ the binary predicate $<$ by unary predicates $<_c = \{x \mid x < c\}$ for $c \in \mathbb{Z}$, then satisfiability of CCTL* was shown to be decidable in [20]. For the full structure $\mathcal{Z}$ satisfiability was shown to be decidable for CEF$^+$, a fragment of CCTL* which contains both the existential and the universal fragment of CCTL*, see [6] for details. Later in [7] Bozzelli and Pinchinat proved that satisfiability of the existential and universal fragment of CCTL* over the domain $(\mathbb{Z}, =, <)$ are PSPACE-complete. Finally, in [11], we answered the question of Demri and Gascon positively:

▶ **Theorem 5** ([11]). CCTL* *over* $\mathcal{Z}$ *is decidable.*

Before we explain the proof techniques from [11], let us first discuss an extension of Therem 5 to ECTL* (extended CTL*) with constraints over $\mathcal{Z}$, which was shown in the long version [12] of [11] using a straightforward extension of the techniques from [11].

The logic ECTL* (without constraints) is a proper extension of CTL* (see [29, 31]) in which path formulas are defined by Büchi-automata or, equivalently, MSO-formulas. In contrast, CTL* can only specify LTL-properties or, equivalently, first-order properties along paths. To define the constraint version of ECTL* over $\mathcal{D} = (D, (R^{\mathcal{D}})_{R\in\sigma})$ we first have to define a constraint version of MSO (monadic second-order logic) over infinite words, which is interpreted over multi-data words. We speak of CMSO (constraint MSO) over $\mathcal{D}$. Fix a countably infinite set $\mathcal{V}_{\mathsf{el}}$ (resp., $\mathcal{V}_{\mathsf{set}}$) of element variables (resp., set variables). The set of CMSO-formulas over $\mathcal{D}$ is defined by the following grammar, where $y, y_1, y_2 \in \mathcal{V}_{\mathsf{el}}$, $Y \in \mathcal{V}_{\mathsf{set}}$, $p \in \mathcal{P}$, $R \in \sigma$, $k = a_R$ is the arity of R, $i_1, \ldots, i_k \in \mathbb{N}$, and $x_1, \ldots, x_k \in \mathcal{X}$:

$$\varphi \ ::= \ p(y) \ \mid \ y_1 < y_2 \ \mid \ y \in Y \ \mid \ [R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)](y) \ \mid \ \neg\varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \exists y\, \varphi \ \mid \ \exists Y\, \varphi$$

To define the semantics of CMSO over $\mathcal{D}$ we need interpretation functions $I_1 : \mathcal{V}_{\mathsf{el}} \to \mathbb{N}$ and $I_2 : \mathcal{V}_{\mathsf{set}} \to 2^{\mathbb{N}}$. Then, for a multi-data word $w = (A_0, \eta_0)(A_1, \eta_1)(A_2, \eta_2)\cdots$ we define $(w, I_1, I_2) \models \varphi$ inductively as follows (again we omit the obvious cases for boolean operators):

- $(w, I_1, I_2) \models p(y)$ iff $p \in A_j$ where $j = I_1(y)$.
- $(w, I_1, I_2) \models y_1 < y_2$ iff $I_1(y_1) < I_2(y_2)$.
- $(w, I_1, I_2) \models y \in Y$ iff $I_1(y) \in I_2(Y)$.
- $(w, I_1, I_2) \models [R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)](y)$ iff $(\eta_{j+i_1}(x_1), \ldots, \eta_{j+i_k}(x_k)) \in R$, where $j = I_1(y)$.
- $(w, I_1, I_2) \models \exists y\, \varphi$ iff there exists $j \in \mathbb{N}$ such that $(w, I_1[y \mapsto j], I_2) \models \varphi$.
- $(w, I_1, I_2) \models \exists y\, \varphi$ iff there exists $J \subseteq \mathbb{N}$ such that $(w, I_1, I_2[Y \mapsto J]) \models \varphi$.

Here, the function $I_1[y \mapsto j]$ is defined by $I_1[y \mapsto j](y) = j$ and $I_1[y \mapsto j](y') = I_1(y')$ for $y' \neq y$, and similarly for $I_2[Y \mapsto J]$. Moreover, for a pair $(\mathcal{K}, \rho)$ consisting of a $\mathcal{D}$-decorated Kripke structure $\mathcal{K} = (D, R, \lambda, \zeta)$ and a $(\mathcal{K}, v_0)$-path $\rho = (v_0, v_1, v_2, \ldots)$ (for some node v_0 of $\mathcal{K}$) we write $(\mathcal{K}, \rho, I_1, I_2) \models \varphi$ if $(w, I_1, I_2) \models \varphi$, where w is the multi-data word $(\lambda(v_0), \zeta(v_0))(\lambda(v_1), \zeta(v_1))(\lambda(v_2), \zeta(v_2))\cdots$.

Now we can define CECTL* (constraint ECTL*) over $\mathcal{D} = (D, (R^{\mathcal{D}})_{R\in\sigma})$ as follows, where φ is an arbitrary CMSO-formula over $\mathcal{D}$ in which only the set variables $Y_1, \ldots, Y_n$ occur freely:

$$\varphi \ ::= \ \neg\varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \mathsf{E}\varphi[Y_1/\varphi, \ldots, Y_n/\varphi]$$

Such a formula is evaluated in a node $v \in D$ of a $\mathcal{D}$-decorated Kripke structure $\mathcal{K} = (V, R, \lambda, \zeta)$ by the following rule (the definition for boolean operator is the obvious one): $(\mathcal{K}, v_0) \models \mathsf{E}\varphi[Y_1/\varphi_1, \ldots, Y_n/\varphi_n]$ iff there is a $(\mathcal{K}, v_0)$-path $\rho = (v_0, v_1, v_2, \ldots)$ such that $(\mathcal{K}, \rho, I_1, I_2) \models \varphi$, where I_1 is arbitrary (note that φ is not allowed to have free element variables) and I_2 satisfies $I_2(Y_i) = \{j \mid (\mathcal{K}, v_j) \models \varphi_i\}$. Note that for a CMSO-formula φ without free variables, $\mathsf{E}\varphi$ is a CECTL* formula that holds in a node v if there is a path starting in v along which φ holds. Satisfiability for CECTL*-formulas over $\mathcal{D}$ is defined as for CCTL*. The main result of [12] is:

▶ **Theorem 6** ([12]). *Satisfiability for* CECTL* *over* $\mathcal{Z}$ *is decidable.*

In the next section, we explain the method that we use to obtain the results from [11, 12], which we call the EHD-*method*.

4 EHD-method

The EHD-method yields sufficient conditions on a relational structure $\mathcal{D}$ which guarantee that satisfiability of CECTL* over $\mathcal{D}$ is decidable. Then, one can show that the structure $\mathcal{Z}$ satisfies these properties.

The structure $\mathcal{D} = (D, (R^{\mathcal{D}})_{R\in\sigma})$ is *negation closed*, if for every $R \in \sigma$ the complement of $R^{\mathcal{D}}$ is definable in positive existential first-order logic over $\mathcal{D}$. Moreover, since σ can be countably infinite we have to require that a positive existential first-order formula for the complement of $R^{\mathcal{D}}$ is computable from the relational symbol $R \in \sigma$. For instance $(\mathbb{Z}, =, <)$ is negation closed, because $\neg x < y$ iff $(x = y \ \vee \ y < x)$ and $\neg x = y$ iff $(x < y \ \vee \ y < x)$. Negation closure is needed in order to achieve a strong kind of negation normal form for CECTL*, in which the constraints only appear positively.

The second condition on $\mathcal{D}$, the EHD-*property*, expresses the fact that we can provide a characterization of all structures which allow a homomorphism into $\mathcal{D}$ using a suitable logical language. Let $\tau \subseteq \sigma$ be a subsignature of σ. A homomorphism $h : \mathcal{C} \to \mathcal{D}$ from a τ-structure $\mathcal{C} = (C, (R^{\mathcal{C}})_{R\in\tau})$ to the σ-structure $\mathcal{D}$ is a mapping $h : C \to D$ such that for every $R \in \tau$ and every tuple $(c_1, \ldots, c_k) \in R^{\mathcal{C}}$ (where $k = a_R$ is the arity of R) we have $(h(c_1), \ldots, h(c_k)) \in R^{\mathcal{D}}$. We say that the σ-structure $\mathcal{D}$ has the property EHD($\mathcal{L}$) for some logic $\mathcal{L}$ if and only if there is a computable function that maps a finite subsignature $\tau \subseteq \sigma$ to an $\mathcal{L}$-sentence φ_τ such that for any countable τ-structure $\mathcal{C}$ one has:

$$\exists h : \mathcal{C} \to \mathcal{D} \text{ homomorphism} \iff \mathcal{C} \models \varphi_\tau \,.$$

To make use of this condition for proving satisfiability of CECTL*, the logic $\mathcal{L}$ has to satisfy two properties: (i) it has to be at least as expressive as MSO and (ii) the satisfiability problem over the class of infinite node labelled rooted trees has to be decidable for $\mathcal{L}$. In [11, 12] we used for $\mathcal{L}$ the logic Bool(MSO, WMSO+B) (in short BMWB), whose formulas are all boolean combinations of MSO and WMSO+B formulas. Here, WMSO+B is the extension of weak monadic second-order logic (where only quantification over finite subsets is allowed) with the bounding quantifier B: A formula $\mathsf{B}X\,\varphi$ holds in a structure $\mathcal{A}$ if and only if there exists a bound $b \in \mathbb{N}$ such that for every finite subset B of the domain of $\mathcal{A}$ with $\mathcal{A} \models \varphi(B)$ we have $|B| \leq b$. Recently, Bojańczyk and Toruńczyk have shown that satisfiability of WMSO+B over infinite node-labeled trees is decidable [5]. They translate WMSO+B-formulas into a certain kind of tree automata, which they call puzzles. Since puzzles are equipped with a parity acceptance condition, it follows easily from [5] that satisfiability over infinite node labelled trees remains decidable for BMWB. The technical main result from [12] is:

▶ **Theorem 7** ([12]). *Let $\mathcal{D}$ be a relational structure which is (i) negation closed and (ii) has the property* EHD(BMWB)*. Then satisfiability of* CECTL^* *over $\mathcal{D}$ is decidable.*

Let us sketch the proof of this result for CCTL^*, which is notationally a bit simpler than the proof for CECTL^*. So, let φ be a CCTL^* state formula. Using negation closure, we can assume that φ is in a strong negation normal form where negations only appear directly in front of atomic propositions $p \in \mathcal{P}$. For this we have to add dual operators to the logic (e.g., the universal path quantifier A). Negation closure allows to eliminate a negation in front of a constraint. Let r be 1+ the number of subformulas of φ of the form $\mathsf{E}\theta$.

Next, it is easy to show that φ is satisfiable if and only if it has a model $\mathcal{T} = (V, R, \lambda, \zeta)$, where (V, R) is a rooted tree of degree r, meaning that $(\mathcal{T}, v_0) \models \varphi$, where v_0 is the root of (V, R). We call such a structure a $\mathcal{D}$-decorated r-tree. The proof for this tree model property is the same as for classical CTL^*.

We use the following notation for ancestors in a tree: Let (V, R) be a rooted tree and let $v \in V$. Then we denote with v^1 the parent node of v if it exists. Moreover, for $i \geq 0$ let $v^0 = v$ and $v^{i+1} = (v^i)^1$ (the latter does not necessarily exist). So, v^i is the i-th ancestor of v if it exists.

Next we define an abstracted version of φ, where every occurrence of a constraint $\theta = R(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)$ is replaced by $\mathsf{X}^d p_\theta$, where p_θ is a fresh proposition associated with θ. Here $d = \max\{i_1, \ldots, i_k\}$ is the *depth* of the constraint. We call the resulting formula φ^a; it is a pure CTL^*-formula. For a $\mathcal{D}$-decorated r-tree $\mathcal{T} = (V, R, \lambda, \zeta)$ we also define an abstracted version $\mathcal{T}^a = (V, R, \lambda^a)$ (we call it an undecorated r-tree), which is obtained from $\mathcal{T}$ by removing the decoration mapping ζ and adding propositions. More precisely, the labelling function $\lambda^a : V \to 2^{\mathcal{P}}$ is defined as follows: For every node $v \in V$, $\lambda^a(v)$ is the union of $\lambda(v)$ and the set of all fresh propositions p_θ, where θ is a constraint in φ of depth d and θ holds in the path starting in v^d and passing through v. Since θ looks only d steps into the future, it does not matter how the chosen path continues from v downwards in the tree; only the initial segment from the d-th ancestor of v to v is relevant. Note that if the $\mathcal{D}$-decorated r-tree $\mathcal{T}$ is a model for φ, then $\mathcal{T}^a$ is a model for φ^a, but the converse does in general not hold. In order to get an equivalence, we have to add a further condition.

Let $\mathcal{P}_0$ be the set of propositions that appear in the abstracted formula φ^a (which contains the fresh propositions p_θ for constraints θ) and let $\mathcal{X}_0 \subseteq \mathcal{X}$ be the finite set of variables that occur in the initial CCTL^*-formula φ. Assume now that $\mathcal{S} = (V, R, \lambda)$ is an undecorated r-tree, where the propositions from $\mathcal{P}_0$ occur. We define a σ-structure $\mathcal{C}_\mathcal{S}$ as follows (σ is the signature of $\mathcal{D}$): Its universe is the set of all pairs $(v, x) \in V \times \mathcal{X}_0$. Moreover, for every k-ary relation symbol $R \in \sigma$ let $R^{\mathcal{C}_\mathcal{S}}$ (the interpretation of R in the structure $\mathcal{C}_\mathcal{S}$) consist of all tuples $((v^{d-i_1}, x_1), \ldots, (v^{d-i_k}, x_k))$ such that $v \in V$, $\lambda(v)$ contains the proposition θ, θ is the constraint $R(\mathsf{X}^{i_1}(x_1), \ldots, \mathsf{X}^{i_k}(x_k))$, and d is the depth of the constraint. The intuition here is the following: The universe of $\mathcal{C}_\mathcal{S}$ is obtained by attaching to each node of $\mathcal{S}$ copies of the variables from $\mathcal{X}_0$. That a node v is labelled with the proposition p_θ indicates that in the unabstracted version of $\mathcal{S}$ (imagine $\mathcal{S}$ is the abstracted version $\mathcal{T}^a$ of a $\mathcal{D}$-decorated r-tree $\mathcal{T}$) the constraint θ holds in the path starting in v^d and passing through v. The relation $R^{\mathcal{C}_\mathcal{S}}$ contains therefore all tuples that are forced to exist due to the labels p_θ for the constraints θ.

With the above definitions, it is not difficult to prove that the following two statements are equivalent (here, we actually need the fact that constraints only occur positively in φ):

- There is a $\mathcal{D}$-decorated r-tree $\mathcal{T}$ with root v_0 such that $(\mathcal{T}, v_0) \models \varphi$
- There is an undecorated r-tree $\mathcal{S}$ with root v_0 and a homomorphism $h : \mathcal{C}_\mathcal{S} \to \mathcal{D}$ such that $(\mathcal{S}, v_0) \models \varphi^a$.

Now it is fairly easy to finish the proof of Theorem 7. By the EHD(BMWB)-property, there exists a BMWB-sentence ψ such that $\mathcal{C}_{\mathcal{S}} \models \psi$ if and only if there is a homomorphism $h : \mathcal{C}_{\mathcal{S}} \to \mathcal{D}$. From the definition of the structure $\mathcal{C}_{\mathcal{S}}$ it is easy to see that it can be obtained by a so called copying first-order transduction from $\mathcal{S}$. One can therefore construct from the BMWB-sentence ψ another BMWB-sentence ψ' such that $\mathcal{C}_{\mathcal{S}} \models \psi$ if and only if $\mathcal{S} \models \psi'$. Finally, since CTL* can be translated into MSO, one can construct an MSO sentence ψ'' such that $(\mathcal{S}, v_0) \models \varphi^a$ if and only if $\mathcal{S} \models \psi''$. Altogether we have that our initial CCTL*-formula φ is satisfiable if and only if there exists an r-tree $\mathcal{S}$ such that $\mathcal{S} \models \psi' \wedge \psi''$. By the result of Bojańczyk and Toruńczyk [5] the latter is decidable. This concludes our proof sketch for Theorem 7.

By Theorem 7, to prove Theorem 6 it suffices to show that the structure $\mathcal{Z}$ from (4) is negation closed and has the property EHD(BMWB). Negation closure is straightforward to show. For the EHD(BMWB)-property let us briefly argue why the reduct $(\mathbb{Z}, <)$ of $\mathcal{Z}$ has the property EHD(BMWB). For a structure $\mathcal{C} = (C, R)$ (where R is an arbitrary binary relation on C) one can show that there exists a homomorphism from $\mathcal{C}$ to $(\mathbb{Z}, <)$ if and only if (i) R is acyclic and (ii) for all $a, b \in C$ there is a bound on the length of all paths in $\mathcal{C}$ from a to b. These properties can be easily expressed in WMSO+B, which shows that $(\mathbb{Z}, <)$ has the property EHD(BMWB). Moreover, the above characterization of the existence of homomorphisms to $(\mathbb{Z}, <)$ can be extended to $\mathcal{Z}$. Let us only remark that one has to consider also structures $\mathcal{C}$ (over the same signature as $\mathcal{Z}$), where the equality symbol $=$ is not interpreted by the identity relation on the universe of $\mathcal{C}$.

Our proof that $\mathcal{Z}$ has the property EHD(BMWB) actually only needs rather weak assumptions on the unary predicates (which are satisfied for the unary relations $=_a$ and $\equiv_{a,b}$). In particular, Theorem 6 can be extended to expansions of $\mathcal{Z}$ that contain additional unary predicates like the set of primes and even some undecidable subsets of $\mathbb{Z}$, see [12] for details.

The EHD-method is quite general, and it is tempting to try applying it to other structures. An interesting candidate in this context (as mentioned in [16]) is the infinite order tree

$$\mathcal{T}_\infty = (\mathbb{N}^*, <, =, \perp),$$

where $<$ denotes the prefix order on $\mathbb{N}^*$ and $\perp$ denotes the incomparability relation with respect to $<$. We add the latter relation in order to obtain a negation closed structure. Unfortunately, using an Ehrenfeucht-Fraïssé-game for WMSO+B, we proved in [10] that $\mathcal{T}_\infty$ does not satisfy the property EHD(BMWB):

▶ **Theorem 8** ([10]). *There is no* BMWB*-sentence ψ such that for every countable structure $\mathcal{A}$ over the signature $\{<, =, \perp\}$ one has: $\mathcal{A} \models \psi$ if and only if there is a homomorphism $h : \mathcal{A} \to \mathcal{T}_\infty$.*

In other words, BMWB is not expressive enough to distinguish between those $\{<, =, \perp\}$-structures which can be mapped homomorphically to the infinite order tree and those that cannot.

This shows that the EHD-method cannot be applied to the concrete domain $\mathcal{T}_\infty$ (or, equivalently, to the infinite binary tree), but it does not imply that satisfiability for CECTL* over $\mathcal{T}_\infty$ is undecidable. In fact, recently Demri and Deters [14] gave a positive answer for CCTL* over $\mathcal{T}_\infty$:

▶ **Theorem 9** ([14]). *Satisfiability for* CCTL* *over $\mathcal{T}_\infty$ is decidable and* PSPACE*-completeness for the corresponding* CLTL*-fragment.*

The result for CLTL has been recently reproved in [21] using a direct automata theoretic approach. Demri and Deters prove their results actually for a richer logic, which allows to compare the length of the longest common prefix for pairs of elements from $\mathcal{T}_\infty$. Decidability is obtained by a reduction to the satisfiability problem of CLTL (resp., CCTL*) over the domain $(\mathbb{N}, =, <, (=_a)_{a\in\mathbb{N}})$, which is PSPACE-complete (resp., decidable) by [16] (resp., [11]). We conjecture that the decidability result for CCTL* over $\mathcal{T}_\infty$ can be extended to CECTL*.

Despite the fact that the EHD-method fails for $\mathcal{T}_\infty$, one can apply it to other tree-like structures, such as semi-linear orders, ordinal trees, and infinitely branching trees of a fixed height. *Semi-linear orders* are partial orders that are tree-like in the sense that for every element x the set of all smaller elements $\downarrow x$ forms a linear suborder. If this linear suborder $\downarrow x$ is an ordinal (for every x) then one has an *ordinal tree*. Ordinal trees are widely studied in descriptive set theory and recursion theory. Note that a tree is a particular instance of a semi-linear order which has a smallest element and where for every x the set $\downarrow x$ is finite.

So far, we have investigated satisfiability for CECTL* over one fixed structure $\mathcal{D}$. For semi-linear orders and ordinal trees it is more natural to consider satisfiability with respect to a class of concrete domains Γ (over a fixed signature σ): The question becomes, whether for a given CECTL*-formula φ there is a concrete domain $\mathcal{D} \in \Gamma$ such that φ holds in a $\mathcal{D}$-decorated Kripke structure. If a class Γ has a universal structure[1] $\mathcal{U}$, then satisfiability with respect to the class Γ is equivalent to satisfiability with respect to $\mathcal{U}$ because there is a $\mathcal{D} \in \Gamma$ such that φ holds in a $\mathcal{D}$-decorated Kripke structure if and only if φ holds in a $\mathcal{U}$-decorated Kripke structure. A typical class with a universal structure is the class of all countable linear orders, for which $(\mathbb{Q}, <)$ is universal. Similarly, for the class of all countable trees the tree $\mathcal{T}_\infty$ as well as the infinite binary tree are universal.

Using the EHD-method, we proved the following decidability results in [10]:

▶ **Theorem 10** ([10]). *Satisfiability of* CECTL* *over each of the following classes is decidable:*
- *the class of all semi-linear orders,*
- *the class of all ordinal trees, and*
- *for each $h \in \mathbb{N}$, the class of all order trees of height h.*

5 Adding Non-Local Constraints

Notice that the constraints of the form $R(\mathsf{X}^{i_1}x_1, \dots, \mathsf{X}^{i_k}x_k)$ which we have considered so far are *local*, in the sense that they can compare data values in an n-sized neighborhood of the state in which they are evaluated, where $n = \max\{i_1, \dots, i_k\} + 1$. Other proposed extensions of temporal logics have the ability to compare data values at arbitrary distance. Metric temporal logic (MTL) [2] and FreezeLTL [17] are two prominent examples of such logics. In [15], Demri and D'Souza ask whether satisfiability of CLTL with constraints over the integers is preserved when adding non-local constraints of the form $x = \mathsf{F}y$, stating that there exists a future state where the value of y is equal to the current value of x. Using a reduction from the Π^0_1-complete problem of deciding the existence of an infinite accepting run of an incrementing counter automaton (see [17]), we answered this question negatively in [12]:

▶ **Theorem 11** ([12]). *Satisfiability for* CLTL *over* $(\mathbb{Z}, =, <)$ *extended with non-local constraints of the form* $x = \mathsf{F}y$ *is undecidable.*

On the other hand, if one adds non-local constraints of the form $x < \mathsf{F}y$ and $\mathsf{F}x < y$ to CLTL, then one still gets a decidable logic:

---

[1] A structure $\mathcal{U}$ is universal for the class Γ if (i) $\mathcal{U} \in \Gamma$ and (ii) every structure from Γ is an induced substructure of $\mathcal{U}$.

▶ **Theorem 12.** *Satisfiability for* CLTL *over* $\mathcal{Z}$ *from* (4) *extended with non-local order constraints of the form* $x < \mathsf{F}y$ *and* $\mathsf{F}x < y$ *is* PSPACE*-complete.*

This result was shown in the PhD thesis of the first author [8], where it is shown that non-local order constraints of the form $x < \mathsf{F}y$ and $\mathsf{F}x < y$ can be replaced by local order constraints. It remains open, whether CCTL* over $\mathcal{Z}$ (or the reduct $(\mathbb{Z}, =, <)$) extended with non-local constraints of the form $x < \mathsf{F}y$ and $\mathsf{F}x < y$ is still decidable.

6 Related Work

In the area of knowledge representation, extensions of description logics with constraints on different concrete domains have been intensively studied, see [23] for a survey. In [24], it was shown that the extension of the description logic $\mathcal{ALC}$ with constraints from $(\mathbb{Q}, <, =)$ has a decidable (EXPTIME-complete) satisfiability problem even in the presence of general TBoxes. A TBox can be seen as a second $\mathcal{ALC}$-formula that has to hold in all nodes of a model. Our decidability proof is partly inspired by the construction from [24], which in contrast to our proof is purely automata-theoretic. Further results for description logics and concrete domains can be found in [25, 26].

There are other extensions of temporal logics that allow to reason about structures with data values, especially in the linear time setting. Logical languages like MTL [22, 2] and TPTL [1] are extensions of LTL often used to specify properties of *timed words*, i.e. data words over the real numbers in which the data sequence is monotonically growing, or monotonic data words over the natural numbers. These logics have however also received some attention on non-monotonic data words [9, 19]. In general, as soon as one drops the monotonicity requirement, satisfiability for these logics becomes undecidable and research has been concentrating on some decidable fragments. An example is freezeLTL, a syntactical restriction of TPTL that has the ability to check data values only for equality. Satisfiability for freezeLTL has been shown to be decidable over finite data words, but undecidable over infinite data words [17]. In contrast to CLTL, the constraints of freezeLTL are of a global nature.

7 Open Problems

The most important open problem that remains from this work concerns the complexity of satisfiability for CCTL* over $(\mathbb{Z}, <, =)$ (or even $\mathcal{Z}$). Clearly, this problem is 2EXPTIME-hard, since satisfiability of (unconstrained) CTL* is 2EXPTIME-complete [18, 30]. To get an upper complexity bound, one should investigate the complexity of the emptiness problem for puzzles (the tree automata used in [5] to show that satisfiability of WMSO+B over infinite node labelled trees is decidable). The WMSO+B-properties used in our decidability result for the structure $(\mathbb{Z}, <, =)$ are very simple, in particular there quantifier nesting depth is small. One may hope to derive a reasonable complexity bound from this observation. At the same time, we believe that our decidability result based on the EHD-method, whose upside is its general nature, may not be the most effective way to devise an efficient decidability procedure for the specific case of the structure $(\mathbb{Z}, <, =)$. The reason behind this statement is the following: Recall from our proof sketch for Theorem 7 that we have to check whether there exists a tree $\mathcal{S}$ which (i) is a model of the abstracted CTL*-formula φ^a and (ii) such that there is a homomorphism from the structure $\mathcal{C}_{\mathcal{S}}$ to $\mathcal{D}$. Property (i) can be checked in 2EXPTIME. The complexity theoretic bottleneck in our proof is property (ii). We simply translate it to BMWB over trees, for which we have no complexity bound. But $\mathcal{C}_{\mathcal{S}}$ has some interesting properties:

It has bounded degree and bounded tree-width, where the tree-width is determined by the number of variables occurring in the input formula. Maybe one can exploit this fact to come up with a more efficient solution. Let us remark that for CECTL^* satisfiability over any concrete domain is non-elementary since path properties are specified in MSO (for which satisfiability is non-elementary). Of course one may replace MSO by Büchi automata (as in [29, 31]), which might then lead to an elementary complexity bound.

Another interesting question is whether there exists a linear order $(A, <)$ such that satisfiability for CECTL^* (or even CCTL^* or CLTL) over $(A, <, =)$ is undecidable. Such a linear order must be necessarily scattered, i.e., $(\mathbb{Q}, <)$ cannot be embedded into $(A, <)$. If $(\mathbb{Q}, <)$ can be embedded into $(A, <)$, then satisfiability over $(A, <, =)$ is equivalent to satisfiability over $(\mathbb{Q}, <, =)$. Finally, it would be interesting to know, whether there exists a concrete domain $\mathcal{D}$ such that satisfiability for CLTL over $\mathcal{D}$ is decidable but the more expressive CCTL^* over $\mathcal{D}$ is undecidable.

References

1 Rajeev Alur and Thomas A. Henzinger. A really temporal logic. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS 1989)*, pages 164–169. IEEE Computer Society Press, 1989.

2 Rajeev Alur and Thomas A. Henzinger. Real-time logics: complexity and expressiveness. *Information and Computation*, 104:390–401, 1993.

3 Franz Baader and Philipp Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 1991)*, volume 1, pages 452–457. Morgan Kaufmann, 1991.

4 Philippe Balbiani and Jean-François Condotta. Computational complexity of propositional linear temporal logics based on qualitative spatial or temporal reasoning. In *Proceedings of the 4th International Workshop on Frontiers of Combining Systems (FroCos 2002)*, volume 2309 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 2002.

5 M. Bojańczyk and S. Toruńczyk. Weak MSO+U over infinite trees. In *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS 2012)*, volume 14 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 648–660. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012.

6 Laura Bozzelli and Régis Gascon. Branching-time temporal logic extended with qualitative presburger constraints. In *Proceedings of the 13th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2006)*, volume 4246 of *Lecture Notes in Computer Science*, pages 197–211. Springer, 2006.

7 Laura Bozzelli and Sophie Pinchinat. Verification of gap-order constraint abstractions of counter systems. *Theoretical Computer Science*, 523:1–36, 2014.

8 Claudia Carapelle. *On the Satisfiability of Temporal Logics with Concrete Domains*. PhD thesis, University of Leipzig, 2015. submitted.

9 Claudia Carapelle, Shiguang Feng, Oliver Fernández Gil, and Karin Quaas. Satisfiability for MTL and TPTL over Non-monotonic Data Words. In *Proceedings of the 8th International Conference on Language and Automata Theory and Applications (LATA 2014)*, volume 8370 of *Lecture Notes in Computer Science*. Springer, 2014.

10 Claudia Carapelle, Shiguang Feng, Alexander Kartzow, and Markus Lohrey. Satisfiability of ECTL* with Tree Constraints. In *Proceedings of the 10th International Computer Science Symposium in Russia (CSR 2015)*, volume 9139 of *Lecture Notes in Computer Science*, pages 94–108. Springer, 2015.

11 Claudia Carapelle, Alexander Kartzow, and Markus Lohrey. Satisfiability of CTL* with Constraints. In *Proceedings of the 24th International Conference on Concurrency The-*

ory (CONCUR 2013), volume 8052 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 2013.

12 Claudia Carapelle, Alexander Kartzow, and Markus Lohrey. Satisfiability of ECTL* with Constraints, 2015. submitted for publication, `http://www.eti.uni-siegen.de/ti/veroeffentlichungen/ectl-with-constraints.pdf`.

13 Karlis Cerans. Deciding properties of integral relational automata. In *Proceedings of the 21st International Colloquium on Automata, Languages and Programming (ICALP 1994)*, volume 820 of *Lecture Notes in Computer Science*, pages 35–46. Springer-Verlag, 1994.

14 Stéphane Demri and Morgan Deters. Temporal logics on strings with prefix relation. *Journal of Logic and Computation*, 2015. to appear.

15 Stéphane Demri and Deepak D'Souza. An automata-theoretic approach to constraint LTL. *Information and Computation*, 205(3):380–415, 2007.

16 Stéphane Demri and Régis Gascon. Verification of qualitative Z constraints. *Theoretical Computer Science*, 409(1):24–40, 2008.

17 Stéphane Demri and Ranko Lazić. LTL with the freeze quantifier and register automata. *ACM Transactions on Computational Logic*, 10(3):16:1–16:30, 2009.

18 E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs. *SIAM Journal on Computing*, 29(1):132–158, 1999.

19 Shigunag Feng, Markus Lohrey, and Karin Quaas. Path-checking for MTL and TPTL over data words. In *Proceedings of the 19th International Conference on Developments in Language Theory (DL 2015)*, 2015. to appear.

20 Régis Gascon. An automata-based approach for CTL* with constraints. *Electronic Notes in Theoretical Computer Science*, 239:193–211, 2009.

21 Alexander Kartzow and Thomas Weidner. Model checking constraint LTL over trees. Technical report, arxiv.org, 2015. `http://arxiv.org/abs/1504.06105`.

22 Ron Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

23 Carsten Lutz. Description logics with concrete domains – a survey. In *Advances in Modal Logic 4*, pages 265–296. King's College Publications, 2003.

24 Carsten Lutz. Combining interval-based temporal reasoning with general TBoxes. *Artificial Intelligence*, 152(2):235–274, 2004.

25 Carsten Lutz. NEXP TIME-complete description logics with concrete domains. *ACM Transactions on Computational Logic*, 5(4):669–705, 2004.

26 Carsten Lutz and Maja Milicic. A tableau algorithm for description logics with concrete domains and general TBoxes. *Journal of Automated Reasoning*, 38(1-3):227–259, 2007.

27 Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS 1977)*, pages 46–57. IEEE Computer Society, 1977.

28 A. Prasad Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

29 Wolfgang Thomas. Computation tree logic and regular omega-languages. In *REX Workshop*, pages 690–713, 1988.

30 Moshe Y. Vardi and Larry J. Stockmeyer. Improved upper and lower bounds for modal logics of programs: Preliminary report. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, STOC 1985*, pages 240–251. ACM, 1985.

31 Moshe Y. Vardi and Pierre Wolper. Yet another process logic (preliminary version). In *Logic of Programs*, pages 501–512, 1983.

32 Frank Wolter and Michael Zakharyaschev. Spatio-temporal representation and reasoning based on RCC-8. In *Proceedings of the 7th Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, pages 3–14. Morgan Kaufmann, 2000.