

# Constraint Automata on Infinite Data Trees: From $\text{CTL}(\mathbb{Z})/\text{CTL}^*(\mathbb{Z})$ To Decision Procedures

Stéphane Demri<sup>1</sup> and Karin Quaas<sup>2</sup>

<sup>1</sup>Université Paris-Saclay, CNRS, ENS Paris-Saclay,  
Laboratoire Méthodes Formelles, 91190, Gif-Sur-Yvette, France

<sup>2</sup> Universität Leipzig, Fakultät für Mathematik und Informatik

**Abstract.** We introduce the class of **tree constraint automata with data values in  $\mathbb{Z}$**  (equipped with the less than relation and equality predicates to constants) and we show that the nonemptiness problem is  $\text{EXPTIME}$ -complete. Using an automata-based approach, we establish that the satisfiability problem for  $\text{CTL}(\mathbb{Z})$  ( $\text{CTL}$  with constraints in  $\mathbb{Z}$ ) is  $\text{EXPTIME}$ -complete and the satisfiability problem for  $\text{CTL}^*(\mathbb{Z})$  is  $2\text{EXPTIME}$ -complete, solving a longstanding open problem (only decidability was known so far). By-product results with other concrete domains and other logics, such as description logics with concrete domains, are also briefly presented.

## 1 Introduction

*Logics with concrete domains.* Many works are dedicated to reasoning about temporal logics on concrete domains (relational structures with a fixed non-empty domain and a family of relations), see e.g. [BG06,DHV14,Car15], so that temporal reasoning is done about the evolution of typed variables (for instance interpreted by integers or by strings, to cite a few examples). Typically, logical formalisms for reasoning about concrete domains contain atomic formulae expressing constraints about the concrete domains and the models are Kripke-style structures, see e.g. [BH91,BC02,Lut04b,FHP22]. By way of example, such formalisms include description logics with concrete domains, see e.g. [Lut02,CT16,Lab21,BR22] as well as temporal logics, see e.g. the recent works [GGG22,FHP22,FMW22b]. In this former family of logics, handling ontologies with values from concrete domains means that the domain elements are enriched with tuples of values, see e.g. [Lut03,Lut04b,LM07]. Furthermore, combining reasoning in your favourite logic with reasoning in a relevant concrete domain reveals to be essential for numerous applications, for instance for reasoning about ontologies, see e.g. [Lut04b,LOS20], or about data-aware systems, see e.g. [DHV14,FMW22a]. A brief survey can be found in [DQ21].

*Automata-based approach.* The automata-based approach for solving decision problems for temporal/modal/description logics can be adapted in the presence of concrete domains, following seminal works, for instance for MSO logic [Büc62] and for temporal logics, see e.g. [VW86,VW94,KVW00]. This popular approach consists of reducing logical problems (satisfiability, model-checking) to automata-based decision problems while taking advantage of existing results and decision procedures from automata theory, see e.g. [VW08]. In the presence of concrete domains, one can distinguish two approaches. The first one consists in designing constraint automata (see e.g. [Rev02]) accepting directly structures with data values. This is the approach explicitly followed in [ST11,KW15,PQ22], see also automata accepting finite data trees in [Fig12]. The translation from logics with concrete domains to constraint automata often smoothly follows the plain case with temporal logics, see e.g. [VW94] and the difficult part is to design

decision procedures for checking nonemptiness of constraint automata (and this is done only once). The second approach consists in reducing the satisfiability problem into the nonemptiness problem for automata handling finite alphabets, see e.g. [Lut01,Lut04a,DD07,Gas09,LOS20]. In this approach, the main effort is put on the design of the translation (based on abstractions for tuples of data values) as the decision procedures for the target automata are already well-studied.

*Our motivations.* The class of concrete domains handled in [LM07,Gas09,BR22] and leading to decidability results excludes the ubiquitous concrete domain  $(\mathbb{Z}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{Z}})$  (set of integers with the less than relation and equality predicate to constants). By contrast, decidability results for logics with concrete domain  $\mathbb{Z}$  requires dedicated proof techniques, see e.g. [BG06,DD07,ST11,LOS20]. In particular, *fragments* of  $\text{CTL}^*$  with concrete domain  $\mathbb{Z}$  (written  $\text{CTL}^*(\mathbb{Z})$ ) are shown decidable in [BG06] using integral relational automata from [Čer94]. Another important breakthrough came with the decidability of  $\text{CTL}^*(\mathbb{Z})$  [CKL16, Theorem 32] by designing a reduction to a decidable second-order logic, whose formulae are made of Boolean combinations of formulae from MSO and from WMSO+U [BT12], where U is the unbounding second-order quantifier, see e.g. [Boj04,BC06]. This is all the more remarkable as the decidability result is part of a powerful general approach [CKL16], but no sharp complexity upper bound can be concluded from it. More recently, the condition  $\mathcal{C}_{\mathbb{Z}}$  [DD07] to approximate the set of satisfiable symbolic models of a given  $\text{LTL}(\mathbb{Z})$  formula (in a problematic way, not necessarily an  $\omega$ -regular language) is extended to the branching case in [LOS20] leading to the EXPTIME-easiness of the satisfiability problem w.r.t. a TBox for the description logic  $\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c)$ . Indeed, a new (approximation) condition  $(\star)$  is proposed in [LOS20] generalising  $\mathcal{C}_{\mathbb{Z}}$ . More importantly,  $(\star)$  corresponds to an  $\omega$ -regular tree language.

It is well-known that decision procedures for  $\text{CTL}^*$  are difficult to design (for instance in the automata-based approach, Safra’s determinisation of Büchi automata can be used) and therefore the combination with the concrete domain  $\mathbb{Z}$  is definitely challenging. The simple idea that initiates this work is to wonder whether the product of the automaton capturing the condition  $(\star)$  in [LOS20] with the automaton for a propositional abstraction of  $\text{CTL}^*(\mathbb{Z})$  formulae would not lead to an elementary complexity for the satisfiability problem for  $\text{CTL}^*(\mathbb{Z})$ , if not an optimal complexity. For instance, in [Gas09, Appendix D] such abstractions for  $\text{CTL}^*(\mathbb{R})$  formulae are designed with alternating parity tree automata based on a construction for  $\text{CTL}^*$  formulae from [KVV00]. Before pursuing this appealing idea, the question of determining the right automata models (on finite alphabets) for abstractions of  $\text{CTL}^*(\mathbb{Z})$  formulae needs to be answered (alternating versus deterministic ones, type of abstraction, etc.), in particular when computational complexity enters into play. However, we aim at proposing a general framework and therefore we do not wish for every new logic with concrete domain to study again and again what is the proper way to define products of automata leading to optimal complexity (in [LOS20], this is done for the description logic  $\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c)$ ). That is why our main goal in this work is to investigate the existence of a model of *tree constraint automata* (understood as a target formalism in the pure tradition of the automata-based approach), for which the nonemptiness problem is decidable, hopefully leading to an elementary complexity upper bound for  $\text{CTL}^*(\mathbb{Z})$  but also to be easily reused. The structures accepted by such automata should be *infinite* data trees in which nodes are labelled by a letter from a finite alphabet and a tuple in  $\mathbb{Z}^{\beta}$  for some  $\beta \geq 1$  (this would exclude the automata designed in [Fig10, Fig12] dedicated to finite trees and no predicate  $<$  is involved). Decision problems for alternating automata over infinite alphabets are often undecidable, see e.g. [NSV04,LW08,DL09], and therefore we advocate the introduction of *nondeterministic* constraint automata without alternation. In view of the complexity of the existing works for deciding  $\text{CTL}^*$  formulae, as well as for logics with

concrete domain  $\mathbb{Z}$ , we shall do our best to provide a self-contained presentation of our results, which is definitely a challenging task as shown in this paper.

*Our contributions.* In this paper, we consider  $(\mathbb{Z}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{Z}})$  (abbreviated by  $\mathbb{Z}$ ) as our working concrete domain.

- In Section 3.1, we introduce the class of tree constraint automata accepting infinite finite-branching trees with nodes labelled by letters from a finite alphabet and finite tuples from  $\mathbb{Z}$ . Definition 1 naturally extends the definition of constraint automata for words (see e.g. [Čer94, Rev02, ST11, KW15]) and as far as we know, the extension to infinite trees in the way done herein has not been considered earlier in the literature. In Section 3.2, we show how to translate  $\text{CTL}(\mathbb{Z})$  (CTL with concrete domain  $\mathbb{Z}$ ) formulae into tree constraint automata, adapting the standard automata-based approach developed in [VW86] (see also [Wol87]). However, we modify the construction from [VW08] not only to handle nondeterministic tree automata but also we provide refinements to the translation. In particular, we need to establish a tree model property for  $\text{CTL}(\mathbb{Z})$  with a strict discipline on the witness paths or successors (Proposition 3).
- In Section 4, we show that the nonemptiness problem for tree constraint automata is  $\text{EXPTIME}$ -complete (Theorem 2 and Theorem 4) and this is a key result in our investigations. The  $\text{EXPTIME}$  lower bound is by reduction from the acceptance problem for alternating Turing machines running in polynomial-space. In order to show the  $\text{EXPTIME}$  upper bound, we adapt results from [LOS20, Lab21] (originally expressed in the context of interpretations for description logics) and we take advantage of several automata-based constructions for Rabin/Streett tree automata. In particular, we observed that the condition  $(\star)$  on framed constraint graphs in [LOS20, Lab21] needs to be modified leading to the condition  $(\star^c)$  on the graphs  $G_{\mathfrak{t}}^c$  (see Section 4.3 for all the details), requiring to prove Proposition 4. In a way, our initial idea to use off-the-shelf the condition  $(\star)$  from [LOS20, Lab21] was not realistic (see all the reasons in Section 4.3). As a corollary, we establish that the satisfiability problem for  $\text{CTL}(\mathbb{Z})$  is  $\text{EXPTIME}$ -complete (Theorem 5), which is one of the main results of the paper. It is worth noting that no elementary complexity upper bound for the satisfiability problem for  $\text{CTL}(\mathbb{Z})$  was known since the decidability of  $\text{CTL}^*(\mathbb{Z})$  established in [Car15, CKL16].
- The proof of Proposition 4 is given in Section 7, as the above-mentioned condition  $(\star^c)$  is central in our paper. We establish Lemma 25 to characterise the class of satisfiable symbolic data value trees. Then, the main ideas of the final (lengthy) part of the proof of Proposition 4 are due to [LOS20, Lab21], sometimes adapted and completed to meet our needs (for instance, we introduce a simple and explicit taxonomy on paths).
- In Section 5.2, as a by-product, we are able to conclude that the satisfiability problem w.r.t. a TBox for the description logic  $\mathcal{ALCF}^p(\mathbb{Z}_c)$  is in  $\text{EXPTIME}$  (Proposition 5), a result already known since [LOS20]. Though it may sound that we turned upside down the material from [LOS20] to regain the  $\text{EXPTIME}$  upper bound, the major difference herein is that we have now a pivot formalism, namely tree constraint automata for which we can smoothly apply the automata-based approach.
- In Section 6, we show that the satisfiability problem for  $\text{CTL}^*(\mathbb{Z})$  can be solved in  $2\text{EXPTIME}$  by using Rabin tree constraint automata (also introduced herein). We follow the challenging automata-based approach for  $\text{CTL}^*$ , see e.g. [ES84, EJ00], but adapted to tree automata with constraints, leading to the  $2\text{EXPTIME}$  upper bound. Not only we had to check that the essential steps for  $\text{CTL}^*$  can be lifted to  $\text{CTL}^*(\mathbb{Z})$  but also that computationally we are in a position to provide an optimal upper bound. In Section 6.1, we establish a special form for  $\text{CTL}^*(\mathbb{Z})$  formulae from which tree automata are defined, adapting the developments from [ES84]. Moreover, determinisation of

nondeterministic (Büchi) word constraint automata with Rabin word constraint automata is proved in Section 6.3 following developments from [Saf89, Chapter 1] but carefully adapted to the context of constraint automata. The main contribution of the present work is definitely the characterisation of the complexity for  $\text{CTL}^*(\mathbb{Z})$  satisfiability, for which only decidability was known from [Car15,CKL16] with no identified complexity upper bound.

In Figure 1, we present a graphical representation of the dependencies between the different results in the paper (prefixed by the corresponding sections). Variations in the thickness of the lines are designed to self-evaluate the difficulty of our contributions in the corresponding sections.

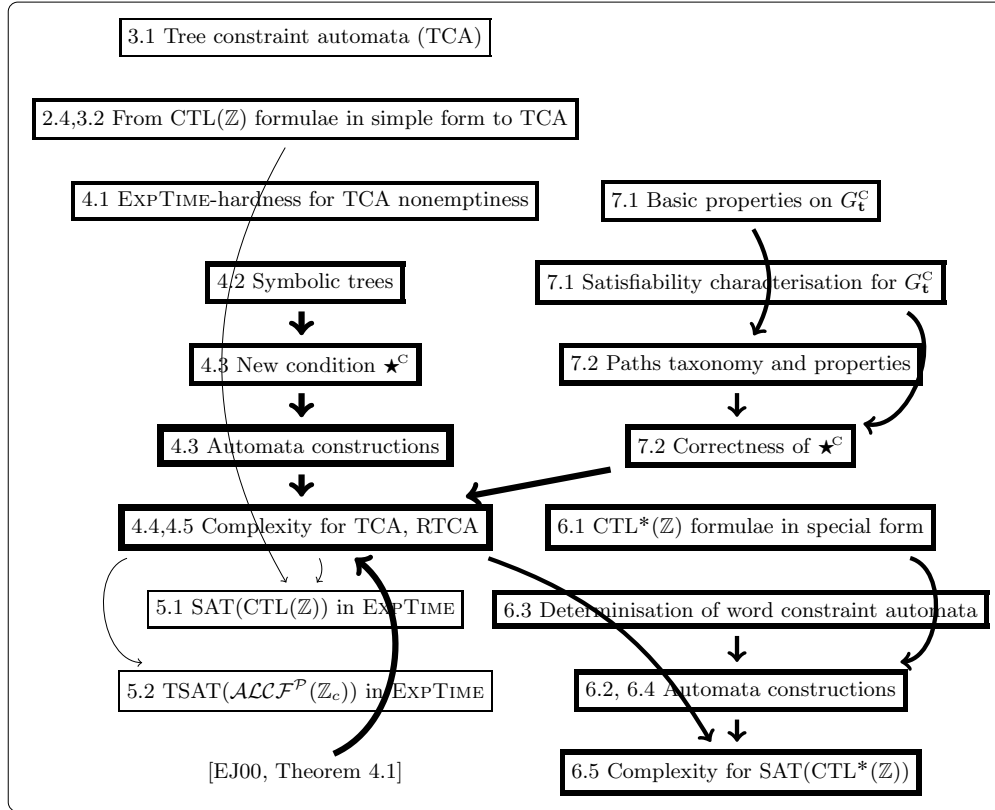


Fig. 1: Roadmap for the main results.

## 2 Temporal Logics with Numerical Domains

### 2.1 Domain $(\mathbb{Z}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{Z}})$ and Kripke-style structures

*Concrete domain  $\mathbb{Z}$ .* In the sequel, we consider the concrete domain  $(\mathbb{Z}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{Z}})$  (also written  $\mathbb{Z}$ ), where  $=_{\mathfrak{d}}$  is a unary predicate stating the equality with the constant  $\mathfrak{d}$  and,  $<$  and  $=$  are the usual binary predicates on  $\mathbb{Z}$ . We also consider the concrete domain  $(\mathbb{N}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{N}})$  (written  $\mathbb{N}$ ), which is defined similarly to  $\mathbb{Z}$ .

Let  $\text{VAR} = \{\mathbf{x}, \mathbf{y}, \dots\}$  be a countably infinite set of variables. A *term*  $\mathbf{t}$  over  $\text{VAR}$  is an expression of the form  $\mathbf{X}^i \mathbf{x}$ , where  $\mathbf{x} \in \text{VAR}$  and  $\mathbf{X}^i$  is a (possibly empty) sequence of  $i$  symbols ‘ $\mathbf{X}$ ’. A term  $\mathbf{X}^i \mathbf{x}$  should be understood as a variable (that needs to be interpreted) but, later on, we will see that the prefix  $\mathbf{X}^i$  will have a temporal interpretation. We write  $T_{\text{VAR}}$  to denote the set of all terms over  $\text{VAR}$ . For all  $i \in \mathbb{N}$ , we write  $T_{\text{VAR}}^{\leq i}$  to denote the subset of terms of the form  $\mathbf{X}^j \mathbf{x}$ , where  $j \leq i$ . For instance,  $T_{\text{VAR}}^{\leq 0} = \text{VAR}$ . An *atomic constraint*  $\theta$  over  $T_{\text{VAR}}$  is an expression of one of the forms below:

$$\mathbf{t} < \mathbf{t}' \quad \mathbf{t} = \mathbf{t}' \quad =_{\mathfrak{d}}(\mathbf{t}) \text{ (also written } \mathbf{t} = \mathfrak{d}),$$

where  $\mathfrak{d} \in \mathbb{Z}$  and  $\mathbf{t}, \mathbf{t}' \in T_{\text{VAR}}$ . A *constraint*  $\Theta$  is defined as a Boolean combination of atomic constraints; we use  $\wedge$ ,  $\vee$  and  $\neg$  for the standard Boolean connectives. Constraints are interpreted on valuations  $\mathbf{v} : T_{\text{VAR}} \rightarrow \mathbb{Z}$  that assign elements from  $\mathbb{Z}$  to the terms in  $T_{\text{VAR}}$ , so that  $\mathbf{v}$  *satisfies*  $\theta$ , written  $\mathbf{v} \models \theta$ , if and only if, the interpretation of the terms in  $\theta$  makes true  $\theta$  in  $\mathbb{Z}$  in the usual way. The Boolean connectives are interpreted as usual. A constraint  $\Theta$  is *satisfiable*  $\stackrel{\text{def}}{\iff}$  there is a valuation  $\mathbf{v} : T_{\text{VAR}} \rightarrow \mathbb{Z}$  such that  $\mathbf{v} \models \Theta$ . Similarly, a constraint  $\Theta_1$  *entails* a constraint  $\Theta_2$  (written  $\Theta_1 \models \Theta_2$ )  $\stackrel{\text{def}}{\iff}$  for all valuations  $\mathbf{v}$ , we have  $\mathbf{v} \models \Theta_1$  implies  $\mathbf{v} \models \Theta_2$ . Note that satisfiability problem restricted to finite conjunctions of atomic constraints can be solved in PTIME and entailment is in coNP. In the sequel, quite often, the valuations  $\mathbf{v}$  are of the form  $\{\mathbf{x}_1, \dots, \mathbf{x}_{\beta}\} \rightarrow \mathbb{Z}$  when we are only interested in the values for the variables in  $\{\mathbf{x}_1, \dots, \mathbf{x}_{\beta}\}$ .

*$\mathbb{Z}$ -decorated Kripke structures.* To define logics with the concrete domain  $\mathbb{Z}$ , its semantical structures (typically, Kripke-style structures) are enriched with valuations that interpret the variables by elements in  $\mathbb{Z}$ . A  *$\mathbb{Z}$ -decorated Kripke structure* (or *Kripke structure* for short)  $\mathcal{K}$  is a triple  $(\mathcal{W}, \mathcal{R}, \mathbf{v})$ , where  $\mathcal{W}$  is a set of *worlds*,  $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$  is the accessibility relation and  $\mathbf{v} : \mathcal{W} \times \text{VAR} \rightarrow \mathbb{Z}$  is a valuation function. A Kripke structure  $\mathcal{K}$  is *total* whenever for all  $w \in \mathcal{W}$ , there is  $w' \in \mathcal{W}$  such that  $(w, w') \in \mathcal{R}$ . Totality is a standard property for defining classes of models for temporal logics such as CTL and CTL\*. Given a Kripke structure  $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathbf{v})$  and a world  $w \in \mathcal{W}$ , an *infinite path*  $\pi$  from  $w$  is an  $\omega$ -sequence  $w_0, \dots, w_n, \dots$  such that  $w_0 = w$  and for all  $i \in \mathbb{N}$ , we have  $(w_i, w_{i+1}) \in \mathcal{R}$ . Finite paths are defined accordingly. We say that a world  $w'$  is *reachable from*  $w$  if there exists a finite path  $w_0, w_1, \dots, w_n$  such that  $w_0 = w$  and  $w_n = w'$ .

*Infinite trees.* Given  $D \geq 1$ , a *tree*  $\mathbf{t}$  of *degree*  $D$  is an infinite, prefix-closed set  $\text{dom}(\mathbf{t}) \subseteq [0, D-1]^*$ , that is, if  $\mathbf{n} \cdot j \in \text{dom}(\mathbf{t})$  for some  $\mathbf{n} \in [0, D-1]^*$  and  $j \in [0, D-1]$ , then  $\mathbf{n} \in \text{dom}(\mathbf{t})$  and  $\mathbf{n} \cdot i \in \text{dom}(\mathbf{t})$  for all  $0 \leq i < j$ , too. The elements of  $\mathbf{t}$  are called *nodes*. The empty word  $\varepsilon$  is the *root node* of  $\mathbf{t}$ . For every node  $\mathbf{n} \in \text{dom}(\mathbf{t})$ , the elements  $\mathbf{n} \cdot i$  (where  $i \in [0, D-1]$ ) are called the *children nodes of*  $\mathbf{n}$ , and  $\mathbf{n}$  is called the *parent node of*  $\mathbf{n} \cdot i$ . We say that the tree  $\mathbf{t}$  is a *full  $D$ -ary tree* if every node  $\mathbf{n}$  has exactly  $D$  children  $\mathbf{n} \cdot 0, \dots, \mathbf{n} \cdot (D-1)$ . We often label the nodes of a tree with elements from some (possibly infinite) alphabet  $\Sigma$ . In that case, we define a tree to be a mapping  $\mathbf{t} : \text{dom}(\mathbf{t}) \rightarrow \Sigma$ . Given a tree  $\mathbf{t}$  and

a node  $\mathbf{n}$  in  $\text{dom}(\mathbf{t})$ , an infinite *path* in  $\mathbf{t}$  starting from  $\mathbf{n}$  is an infinite sequence  $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \dots$ , where  $j_i \in [0, D-1]$  and  $\mathbf{n} \cdot j_1 \dots j_i \in \text{dom}(\mathbf{t})$  for all  $i \geq 1$ .

A *tree Kripke structure*  $\mathcal{K}$  is a Kripke structure  $(\mathcal{W}, \mathcal{R}, \mathbf{v})$  such that  $(\mathcal{W}, \mathcal{R})$  is a tree (not necessarily a full  $D$ -ary tree). Tree Kripke structures  $(\mathcal{W}, \mathcal{R}, \mathbf{v})$  such that  $(\mathcal{W}, \mathcal{R})$  is isomorphic to the tree induced by  $[0, D-1]^*$  are represented by maps of the form  $\mathbf{t} : [0, D-1]^* \rightarrow \mathbb{Z}^\beta$ . This assumes that we only care about the value of the variables  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$  and  $\mathbf{t}(\mathbf{n}) = (\mathbf{d}_1, \dots, \mathbf{d}_\beta)$  encodes that for all  $i \in [1, \beta]$ , we have  $\mathbf{v}(\mathbf{n}, \mathbf{x}_i) = \mathbf{d}_i$ .

## 2.2 The logic $\text{CTL}^*(\mathbb{Z})$

We introduce the logic  $\text{CTL}^*(\mathbb{Z})$  extending the branching-time temporal logic  $\text{CTL}^*$  from [EH86] but with data values in  $\mathbb{Z}$ . *State formulae*  $\phi$  and *path formulae*  $\Phi$  of  $\text{CTL}^*(\mathbb{Z})$  are defined below

$$\phi := \neg\phi \mid \phi \wedge \phi \mid \mathbf{E}\Phi \quad \Phi := \phi \mid \mathbf{t} = \mathbf{d} \mid \mathbf{t}_1 = \mathbf{t}_2 \mid \mathbf{t}_1 < \mathbf{t}_2 \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathbf{X}\Phi \mid \Phi \mathbf{U}\Phi,$$

where  $\mathbf{t}, \mathbf{t}_1, \mathbf{t}_2 \in \mathbf{T}_{\text{VAR}}$ . We use also the standard temporal connectives  $\mathbf{G}, \mathbf{F}$  and the universal path quantifier  $\mathbf{A}$  ( $\mathbf{F}\Phi \stackrel{\text{def}}{=} \top \mathbf{U} \Phi$  with  $\top$  equal to  $\mathbf{E}(\mathbf{x} = \mathbf{x})$ ,  $\mathbf{G}\Phi \stackrel{\text{def}}{=} \neg\mathbf{F}\neg\Phi$  and  $\mathbf{A}\Phi \stackrel{\text{def}}{=} \neg\mathbf{E}\neg\Phi$ ). No propositional variables occur in  $\text{CTL}^*(\mathbb{Z})$  formulae, but it is easy to simulate them with atomic formulae of the form  $\mathbf{E}(\mathbf{x} = 0)$ . State formulae are interpreted on worlds from a Kripke structure, whereas path formulae are interpreted on infinite paths. The two satisfaction relations are defined as follows (we omit the standard clauses for Boolean connectives), where  $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathbf{v})$  is a total Kripke structure, and  $w \in \mathcal{W}$ .

- $\mathcal{K}, w \models \mathbf{E}\Phi \stackrel{\text{def}}{\iff}$  there is an infinite path  $\pi$  from  $w$  such that  $\mathcal{K}, \pi \models \Phi$ .

Let  $\pi = w_0, w_1, \dots$  be an infinite path of  $\mathcal{K}$ . Let us define  $\mathbf{v}(\pi, \mathbf{X}^j \mathbf{x}) \stackrel{\text{def}}{=} \mathbf{v}(w_j, \mathbf{x})$ , for all terms of the form  $\mathbf{X}^j \mathbf{x}$ .

- $\mathcal{K}, \pi \models \mathbf{t} = \mathbf{d} \stackrel{\text{def}}{\iff} \mathbf{v}(\pi, \mathbf{t}) = \mathbf{d}$ ,
- $\mathcal{K}, \pi \models \mathbf{t}_1 \sim \mathbf{t}_2 \stackrel{\text{def}}{\iff} \mathbf{v}(\pi, \mathbf{t}_1) \sim \mathbf{v}(\pi, \mathbf{t}_2)$  for all  $\sim \in \{<, =\}$ ,
- $\mathcal{K}, \pi \models \mathbf{X}\Phi \stackrel{\text{def}}{\iff} \mathcal{K}, \pi[1, +\infty) \models \Phi$ , where for any  $n$ ,  $\pi[n, +\infty)$  is the suffix of  $\pi$  truncated by the  $n$  first worlds,
- $\mathcal{K}, \pi \models \Phi \mathbf{U}\Psi \stackrel{\text{def}}{\iff}$  there is  $j \geq 0$  such that  $\mathcal{K}, \pi[j, +\infty) \models \Psi$  and for all  $j' \in [i, j-1]$ , we have  $\mathcal{K}, \pi[j', +\infty) \models \Phi$ .

For example, the formula  $\phi = \mathbf{AFEGF}(\mathbf{x} < \mathbf{X}\mathbf{x})$  states that on all paths, there is a position from which there is a path for which infinitely often the value for the variable  $\mathbf{x}$  is strictly less than the next value of  $\mathbf{x}$  along this path. The paper aims at proposing a decision procedure to determine the satisfiability status of the formula  $\phi$ . More precisely, the *satisfiability problem for  $\text{CTL}^*(\mathbb{Z})$* , written  $\text{SAT}(\text{CTL}^*(\mathbb{Z}))$ , is defined as follows.

**Input:** A  $\text{CTL}^*(\mathbb{Z})$  state formula  $\phi$ .

**Question:** Are there a total Kripke structure  $\mathcal{K}$  and a world  $w$  such that  $\mathcal{K}, w \models \phi$ ?

Decidability of strict fragments of  $\text{CTL}^*(\mathbb{Z})$  is shown in [BG06]. It is only recently in [Car15, CKL16] that decidability has been established for the full logic using a translation into a decidable second-order logic. Fragments of the model-checking problem involving a temporal logic very similar to  $\text{CTL}^*(\mathbb{Z})$  are also investigated in [Čer94, BG06, DDS18, FMW22a] (see also [CKP15, ADG20]). However, note that model-checking problems with  $\text{CTL}^*(\mathbb{Z})$ -like languages can be easily undecidable, see e.g. [Čer94, Theorem 1].

**Proposition 1.** [CKL16, Theorem 32]  $\text{SAT}(\text{CTL}^*(\mathbb{Z}))$  is decidable.

The proof of Proposition 1 does not provide a complexity upper bound because the target decidable second-order logic admits an automata-based decision procedure with open complexity.

### 2.3 The logic $\text{CTL}(\mathbb{Z})$

The logic  $\text{CTL}(\mathbb{Z})$  is a fragment of  $\text{CTL}^*(\mathbb{Z})$  and its formulae are of the form below

$$\phi := \text{E } \Theta \mid \text{A } \Theta \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \text{EX}\phi \mid \text{E}\phi\text{U}\phi \mid \text{E}\phi\text{R}\phi \mid \text{AX}\phi \mid \text{A}\phi\text{U}\phi \mid \text{A}\phi\text{R}\phi,$$

where  $\Theta$  is a Boolean combination of atomic constraints built over terms in  $\text{T}_{\text{VAR}}$ . Below, we mainly handle formulae in negation normal form (negation occurs only in Boolean combinations of constraints), that is why we have to guarantee that every connective has its dual counterpart. Moreover, we recall that  $\mathcal{K}, w \models \text{E } \phi_1\text{R}\phi_2$  holds exactly when  $\mathcal{K}, w \models \text{E } \neg(\neg\phi_1\text{U}\neg\phi_2)$  holds (by definition), a similar definition exists for  $\text{A } \phi_1\text{R}\phi_2$ . We use also the following standard abbreviations:  $\text{EG}\phi \stackrel{\text{def}}{=} \text{E}(\perp \text{R}\phi)$  and  $\text{AG}\phi \stackrel{\text{def}}{=} \text{A}(\perp \text{R}\phi)$  with  $\perp$  equal to  $\text{E}(\mathbf{x} < \mathbf{x})$ .

A  $\text{CTL}(\mathbb{Z})$  formula is in *simple form* iff it is in negation normal form and terms are restricted to those in  $\text{T}_{\text{VAR}}^{\leq 1}$ . When it comes to translate  $\text{CTL}(\mathbb{Z})$  formulae to tree constraint automata (see Section 3.2), we need to preprocess the initial formulae to ease the translation while being computationally harmless.

**Proposition 2.** For every  $\text{CTL}(\mathbb{Z})$  formula  $\phi$ , one can construct in polynomial-time in the size of  $\phi$  a  $\text{CTL}(\mathbb{Z})$  formula  $\phi'$  in simple form such that  $\phi$  is satisfiable iff  $\phi'$  is satisfiable.

The proof of Proposition 2 first establishes a tree model property (in the standard way using unfoldings) and then uses a renaming technique (see e.g. [Sco62]) to flatten the constraints but tailored to  $\text{CTL}(\mathbb{Z})$ .

*Proof.* First, we establish that  $\text{CTL}(\mathbb{Z})$  has the tree model property using the standard unfolding technique for Kripke structures. Then, we show that the restriction to formulae in simple form is possible using the renaming technique (correctness is guaranteed by the tree model property).

Let  $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathbf{v})$  be a total Kripke structure and  $w \in \mathcal{W}$ . We write  $\hat{\mathcal{K}}_w$  to denote the (total) Kripke structure  $(\hat{\mathcal{W}}, \hat{\mathcal{R}}, \hat{\mathbf{v}})$  defined as follows (understood as the unfolding of  $\mathcal{K}$  from the world  $w$ ).

- $\hat{\mathcal{W}}$  is the set of finite paths in  $\mathcal{K}$  starting from the world  $w$ .
- The relation  $\hat{\mathcal{R}}$  contains all the pairs of the form  $(\pi, \pi')$  such that  $\pi$  is of the form  $w_1 \cdots w_n$  and  $\pi'$  is of the form  $w_1 \cdots w_{n+1}$  with  $(w_n, w_{n+1}) \in \mathcal{R}$ .
- For all paths  $\pi = w_1 \cdots w_n$ , for all variables  $\mathbf{x} \in \text{VAR}$ , we have  $\hat{\mathbf{v}}(\pi, \mathbf{x}) \stackrel{\text{def}}{=} \mathbf{v}(w_n, \mathbf{x})$ .

Satisfaction of all the state formulae is preserved using classical arguments. More precisely, for all  $\pi = w_1 \cdots w_n \in \hat{\mathcal{W}}$ , for all state formulae  $\phi$  in  $\text{CTL}(\mathbb{Z})$ , we have  $\mathcal{K}, w_n \models \phi$  iff  $\hat{\mathcal{K}}_w, \pi \models \phi$ . The proof is by structural induction using the correspondence between paths in  $\mathcal{K}$  from some  $w' \in \mathcal{W}$  such that  $w'$  is reachable from  $w$ , and paths in  $\hat{\mathcal{K}}_w$ . As a conclusion, the logic  $\text{CTL}(\mathbb{Z})$  has the tree model property since the structures of the form  $\hat{\mathcal{K}}_w$  are trees.

Now, we move to the construction of  $\phi'$  in simple form. Let us introduce below the natural notion of forward degree. Given a term  $\mathbf{t} = \text{X}^i \mathbf{x}$  for some  $i \in \mathbb{N}$ , the *forward degree* of  $\mathbf{t}$ , written  $\text{fd}(\mathbf{t})$ , is equal to  $i$ . Given a constraint  $\Theta$ , we write  $\text{fd}(\Theta)$  to denote the *forward degree* of  $\Theta$  defined as the maximal forward degree of any term occurring in  $\Theta$ . For instance,  $\text{fd}((\text{X}\mathbf{x}_1 < \text{XXX}\mathbf{x}_2) \wedge (\mathbf{x}_1 = \mathbf{x}_2)) = 3$ . By extension, given

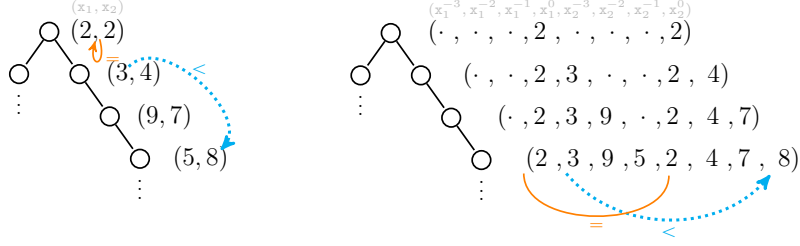


Fig. 2: A tree over two variables  $x_1$  and  $x_2$  (left) and the corresponding tree over eight variables  $x_1^{-3}, x_1^{-2}, x_1^{-1}, x_1^0, x_2^{-3}, x_2^{-2}, x_2^{-1}, x_2^0$  (right) for illustrating the translation of the formula  $\text{EX E}((Xx_1 < XXXx_2) \wedge (x_1 = x_2))$  into its simple form.

a state formula  $\phi$  in  $\text{CTL}(\mathbb{Z})$ , we write  $\text{fd}(\phi)$  to denote the *forward degree* of  $\phi$  defined as the maximal forward degree of any constraint occurring in  $\phi$ . For instance,  $\text{fd}(\text{EX E}((Xx_1 < XXXx_2) \wedge (x_1 = x_2))) = 3$  and  $\text{fd}(\text{E}((XXx_1 < XXXXx_2) \wedge (Xx_1 = Xx_2))) = 4$ .

Let  $\phi$  be a state formula in  $\text{CTL}(\mathbb{Z})$  in negation normal form such that  $\text{fd}(\phi) = N$  and the variables occurring in  $\phi$  are among  $x_1, \dots, x_\beta$ . Below, we build a formula  $\phi'$  over  $x_1^{-N}, \dots, x_1^0, \dots, x_\beta^{-N}, \dots, x_\beta^0$  with  $\text{fd}(\phi') \leq 1$  such that  $\phi$  is satisfiable in a tree Kripke structure iff  $\phi'$  is satisfiable in a tree Kripke structure and  $\phi'$  can be computed in polynomial-time in the size of  $\phi$ . The above-mentioned variables can be obviously renamed but the current naming is helpful to grasp the correctness of the whole enterprise. In short, the value for  $x_j^{-k}$  on a node should be understood as the value for  $x_j$  exactly  $k$  nodes behind along the branch leading to the node. Thanks to the tree structure,  $x_j^{-k}$  takes a unique value. Note that we could not work with forward values (say with  $x_j^{+k}$  to keep the same rule for naming) because the structures are branching and therefore  $k$  steps ahead does not lead necessarily to a unique world/state.

Given  $\text{E } \Theta$  or  $\text{A } \Theta$  occurring in  $\phi$  with  $\text{fd}(\Theta) = M$  (so  $M \leq N$ ), we write  $\text{jump}(\Theta, M)$  to denote the constraints built over  $x_1^{-N}, \dots, x_1^0, \dots, x_\beta^{-N}, \dots, x_\beta^0$  such that

- $\text{jump}(\cdot, M)$  is homomorphic for Boolean connectives,
- $\text{jump}(t_1 < t_2, M) \stackrel{\text{def}}{=} \text{jump}(t_1, M) < \text{jump}(t_2, M)$ ,
- $\text{jump}(t_1 = t_2, M) \stackrel{\text{def}}{=} \text{jump}(t_1, M) = \text{jump}(t_2, M)$ ,
- $\text{jump}(t = \mathfrak{d}, M) \stackrel{\text{def}}{=} \text{jump}(t, M) = \mathfrak{d}$  with  $\text{jump}(X^i x_j, M) \stackrel{\text{def}}{=} x_j^{i-M}$ .

For instance,  $\text{jump}((Xx_1 < XXXx_2) \wedge (x_1 = x_2), 3) = (x_1^{-2} < x_2^0) \wedge (x_1^{-3} = x_2^{-3})$ . In short,  $\text{jump}(\Theta, M)$  corresponds to the constraint equivalent to  $\Theta$  if we evaluate it exactly  $M$  steps ahead. To do so, we therefore need to evoke in  $\text{jump}(\Theta, M)$  variables capturing previous local values along the branch. Let  $t$  be the translation map that is homomorphic for Boolean connectives and temporal connectives such that

- $t(\text{E } \Theta) \stackrel{\text{def}}{=} (\text{EX})^M \text{jump}(\Theta, M)$ ,
- $t(\text{A } \Theta) \stackrel{\text{def}}{=} (\text{AX})^M \text{jump}(\Theta, M)$ ,

where  $\text{fd}(\Theta) = M$ . Let  $\phi'$  be defined as follows:

$$t(\phi) \wedge \text{AG A} \left( \bigwedge_{j \in [1, \beta], k \in [0, N-1]} x_j^{-k} = Xx_j^{-k-1} \right).$$



The second conjunct in the definition of  $\phi'$  corresponds to the renaming part. Let us provide hints to understand why  $\phi$  is satisfiable in a tree Kripke structure iff  $\phi'$  is satisfiable in a tree Kripke structure.

First, suppose that  $\mathcal{K}, w \models \phi$ , where  $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathbf{v})$  is a tree Kripke structure with root  $w$ . Let  $\mathcal{K}' = (\mathcal{W}, \mathcal{R}, \mathbf{v}')$  be the Kripke structure that differs from  $\mathcal{K}$  only in the definition of the valuation. Given a node  $w' \in \mathcal{W}$  reachable from  $w$  via the branch  $w_0 \cdots w_n$  with  $w_0 = w$  and  $w_n = w'$  for some  $n \geq 0$ , for all  $k \in [0, N]$  and  $j \in [1, \beta]$ , we require  $\mathbf{v}'(w', \mathbf{x}_j^{-k}) \stackrel{\text{def}}{=} \mathbf{v}(w_{n-k}, \mathbf{x}_j)$  if  $n - k \geq 0$ , otherwise  $\mathbf{v}'(w', \mathbf{x}_j^{-k}) \stackrel{\text{def}}{=} 0$  (arbitrary value). In order to verify that  $\mathcal{K}', w \models \phi'$ , it boils down to check the three properties below (for which we omit the proofs at this stage).

- $\mathcal{K}', w \models \text{AG } A(\bigwedge_{j \in [1, \beta], k \in [0, N-1]} \mathbf{x}_j^{-k} = \mathbf{x}_j^{-k-1})$  (this holds thanks the definition of  $\mathbf{v}'$  and the tree structure of  $\mathcal{K}$ ).
- If  $\text{E } \Theta$  occurs in  $\phi$  and  $\mathcal{K}, w' \models \text{E } \Theta$ , then  $\mathcal{K}', w' \models (\text{EX})^M \text{jump}(\Theta, M)$ , where  $\text{fd}(\Theta) = M$ .
- If  $\text{A } \Theta$  occurs in  $\phi$  and  $\mathcal{K}, w' \models \text{A } \Theta$ , then  $\mathcal{K}', w' \models (\text{AX})^M \text{jump}(\Theta, M)$ , where  $\text{fd}(\Theta) = M$ .

For the other direction, suppose that  $\mathcal{K}, w \models \phi'$  and  $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathbf{v})$  is a tree Kripke structure with root  $w$ . Let  $\mathcal{K}' = (\mathcal{W}, \mathcal{R}, \mathbf{v}')$  be the Kripke structure that differs from  $\mathcal{K}$  only in the definition of the valuation. More precisely, for all  $w' \in \mathcal{W}$  and all  $j \in [1, \beta]$ , we have  $\mathbf{v}'(w', \mathbf{x}_j) \stackrel{\text{def}}{=} \mathbf{v}(w', \mathbf{x}_j^0)$ . By structural induction, one can show that  $\mathcal{K}, w \models \mathbf{t}(\phi)$  implies  $\mathcal{K}', w \models \phi$ .  $\square$

From now on, we assume that the  $\text{CTL}(\mathbb{Z})$  formulae are in simple form. Given a  $\text{CTL}(\mathbb{Z})$  formula  $\phi$  in simple form, we write  $\text{sub}(\phi)$  to denote the smallest set such that

- $\phi \in \text{sub}(\phi)$ ;  $\text{sub}(\phi)$  is closed under subformulae,
- for all  $\mathcal{Q} \in \{\text{E}, \text{A}\}$  and  $\text{Op} \in \{\text{U}, \text{R}\}$ , if  $\mathcal{Q} \phi_1 \text{ Op } \phi_2 \in \text{sub}(\phi)$ , then  $\mathcal{Q}X \mathcal{Q} \phi_1 \text{ Op } \phi_2 \in \text{sub}(\phi)$ .

The cardinality of  $\text{sub}(\phi)$  is at most twice the number of subformulae of  $\phi$ . Given  $X \subseteq \text{sub}(\phi)$ , we say that  $X$  is *propositionally consistent* iff the conditions below hold.

- If  $\phi_1 \vee \phi_2 \in X$ , then  $\{\phi_1, \phi_2\} \cap X \neq \emptyset$ ; if  $\phi_1 \wedge \phi_2 \in X$ , then  $\{\phi_1, \phi_2\} \subseteq X$ .
- If  $\text{E}\phi_1 \text{U}\phi_2 \in X$ , then  $\phi_2 \in X$  or  $\{\phi_1, \text{EXE}\phi_1 \text{U}\phi_2\} \subseteq X$ .
- If  $\text{A}\phi_1 \text{U}\phi_2 \in X$ , then  $\phi_2 \in X$  or  $\{\phi_1, \text{AXA}\phi_1 \text{U}\phi_2\} \subseteq X$ .
- If  $\text{E}\phi_1 \text{R}\phi_2 \in X$ , then  $\phi_2 \in X$  and  $\{\phi_1, \text{EXE}\phi_1 \text{R}\phi_2\} \cap X \neq \emptyset$ .
- If  $\text{A}\phi_1 \text{R}\phi_2 \in X$ , then  $\phi_2 \in X$  and  $\{\phi_1, \text{AXA}\phi_1 \text{R}\phi_2\} \cap X \neq \emptyset$ .

We write  $\text{sub}_{\text{EX}}(\phi)$  to denote the set of formulae in  $\text{sub}(\phi)$  of the form  $\text{EX}\psi$ . Similarly, we write  $\text{sub}_{\text{EU}}(\phi)$  (resp.  $\text{sub}_{\text{AU}}(\phi)$ ) to denote the set of formulae in  $\text{sub}(\phi)$  of the form  $\text{E}\psi_1 \text{U}\psi_2$  (resp.  $\text{A}\psi_1 \text{U}\psi_2$ ). Finally, we write  $\text{sub}_{\text{E}}(\phi)$  to denote the set of formulae of the form  $\text{E } \Theta$  in  $\text{sub}(\phi)$ .

## 2.4 Refining the tree model property for $\text{CTL}(\mathbb{Z})$

Let  $\phi$  be a  $\text{CTL}(\mathbb{Z})$  formula in simple form built over the variables  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$  for some  $\beta \geq 1$ , and set  $D = \text{card}(\text{sub}_{\text{EX}}(\phi)) + \text{card}(\text{sub}_{\text{E}}(\phi))$ . A *direction map*  $\iota$  for  $\phi$  is a bijection

$$\iota : (\text{sub}_{\text{EX}}(\phi) \cup \text{sub}_{\text{E}}(\phi)) \rightarrow [1, D].$$

We say that a tree model  $\mathbf{t} : [0, D]^* \rightarrow \mathbb{Z}^\beta$  of  $\phi$  (i.e.  $\mathbf{t}, \varepsilon \models \phi$ ) *obeys a direction map*  $\iota$  if for all nodes  $\mathbf{n} \in [0, D]^*$ , (1.)–(3.) hold.

1. For every  $\text{EX}\phi_1 \in \text{sub}_{\text{EX}}(\phi)$ , if  $\mathbf{t}, \mathbf{n} \models \text{EX}\phi_1$ , then  $\mathbf{t}, \mathbf{n} \cdot j \models \phi_1$  with  $j = \iota(\text{EX}\phi_1)$ .

2. For every  $E\phi_1 U\phi_2 \in \text{sub}_{EU}(\phi)$ , if  $\mathbf{t}, \mathbf{n} \models E\phi_1 U\phi_2$ , then there exists some  $k \geq 0$  such that  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_2$  and  $\mathbf{t}, \mathbf{n} \cdot j^i \models \phi_1$  for all  $0 \leq i < k$ , where  $j = \iota(\text{EXE}\phi_1 U\phi_2)$ . In other words, the path  $\mathbf{n}, \mathbf{n} \cdot j, \mathbf{n} \cdot j^2 \dots \mathbf{n} \cdot j^k$  satisfies  $\phi_1 U\phi_2$ .
3. For every  $E \Theta \in \text{sub}_E(\phi)$ , if  $\mathbf{t}, \mathbf{n} \models E \Theta$ , then  $\mathbb{Z} \models \Theta(\mathbf{t}(\mathbf{n}), \mathbf{t}(\mathbf{n} \cdot j))$  with  $\iota(E \Theta) = j$ .

Here,  $\mathbb{Z} \models \Theta(z, z')$  with  $z, z'$  is a shortcut for  $[\vec{x} \leftarrow z, \vec{x}' \leftarrow z'] \models \Theta$  where  $[\vec{x} \leftarrow z, \vec{x}' \leftarrow z']$  is a valuation  $\mathbf{v}$  on the variables  $\{\mathbf{x}_j, \mathbf{x}'_j \mid j \in [1, \beta]\}$  with for all  $j \in [1, \beta]$ ,  $\mathbf{v}(\mathbf{x}_j) = z(j)$  and  $\mathbf{v}(\mathbf{x}'_j) = z'(j)$ .

**Proposition 3.** *Let  $\phi$  be a  $\text{CTL}(\mathbb{Z})$ -formula in simple form and  $\iota$  be a direction map  $\iota$  for  $\phi$ . Then,  $\phi$  is satisfiable if, and only if,  $\phi$  has a tree model with branching width equal to  $\text{card}(\text{sub}_{EX}(\phi)) + \text{card}(\text{sub}_E(\phi)) + 1$  that obeys  $\iota$ .*

*Proof.* The direction from right to left (“if”) is trivial. So let us prove the direction from left to right (“only if”): suppose that  $\phi$  is a satisfiable  $\text{CTL}(\mathbb{Z})$  formula in simple form and built over the terms  $\mathbf{x}_1, \dots, \mathbf{x}_\beta, \mathbf{X}\mathbf{x}_1, \dots, \mathbf{X}\mathbf{x}_\beta$ . We recall that the terms  $\mathbf{X}\mathbf{x}_1, \dots, \mathbf{X}\mathbf{x}_\beta$  are also sometimes denoted by the primed variables  $\mathbf{x}'_1, \dots, \mathbf{x}'_\beta$ . Let  $\iota : (\text{sub}_{EX}(\phi) \cup \text{sub}_E(\phi)) \rightarrow [1, D]$  be a direction map for  $\phi$ , where  $D = \text{card}(\text{sub}_{EX}(\phi)) + \text{card}(\text{sub}_E(\phi))$ .

Let  $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathbf{v})$  be a total Kripke structure,  $w_{\text{in}} \in \mathcal{W}$  be a world in  $\mathcal{K}$  such that  $\mathcal{K}, w_{\text{in}} \models \phi$ . Since  $\phi$  contains only the variables in  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$ , the map  $\mathbf{v}$  can be restricted to the variables among  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$ . Furthermore, below, we can represent  $\mathbf{v}$  as a map  $\mathcal{W} \rightarrow \mathbb{Z}^\beta$  such that  $\mathbf{v}(w)(i)$  for some  $i \in [1, \beta]$  is understood as the value of the variable  $\mathbf{x}_i$  on  $w$ . We construct a tree  $\mathbf{t} : [0, D]^* \rightarrow \mathbb{Z}^\beta$  such that  $\mathbf{t}$  obeys  $\iota$  and  $\mathbf{t}, \varepsilon \models \phi$ .

We introduce an auxiliary map  $g : [0, D]^* \rightarrow \mathcal{W}$  such that  $g(\varepsilon) \stackrel{\text{def}}{=} w_{\text{in}}$ ,  $\mathbf{t}(\varepsilon) \stackrel{\text{def}}{=} \mathbf{v}(g(\varepsilon))$  and more generally, we require that for all  $\mathbf{n} \in [0, D]^*$ , we have  $\mathbf{t}(\mathbf{n}) \stackrel{\text{def}}{=} \mathbf{v}(g(\mathbf{n}))$ . The definition of  $g$  is performed by picking the smallest element  $\mathbf{n} \cdot j \in [0, D]^*$  with respect to the lexicographical ordering such that  $g(\mathbf{n})$  is defined and  $g(\mathbf{n} \cdot j)$  is undefined. Let  $\mathbf{n} \cdot j$  be the smallest node such that  $g(\mathbf{n})$  is defined and  $g(\mathbf{n} \cdot j)$  is undefined. If  $j = 0$ , then, since  $\mathcal{K}$  is total, there is an infinite path  $\pi = w_0 w_1 w_2 \dots$  starting from  $g(\mathbf{n})$ . For all  $i \geq 1$ , we set  $g(\mathbf{n} \cdot 0^i) \stackrel{\text{def}}{=} w_i$  and  $\mathbf{t}(\mathbf{n} \cdot 0^i) \stackrel{\text{def}}{=} \mathbf{v}(w_i)$ . So let  $j > 0$ . Let  $\phi'$  be the unique formula such that  $\iota(\phi') = j$ . If  $\mathcal{K}, g(\mathbf{n}) \models \phi'$ , then, since  $\mathcal{K}$  is total, there is an infinite path  $\pi = w_0 w_1 w_2 \dots$  starting from  $g(\mathbf{n})$ . We define  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} w_1$ ,  $\mathbf{t}(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \mathbf{v}(w_1)$  and for all  $i \geq 1$ , we set  $g(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} w_{i+1}$  and  $\mathbf{t}(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} \mathbf{v}(w_{i+1})$ . So suppose that  $\mathcal{K}, g(\mathbf{n}) \not\models \phi'$ . We distinguish the following cases.

**Case  $\phi' = E \Theta$ .** By  $\mathcal{K}, g(\mathbf{n}) \models E \Theta$ , there is an infinite path  $\pi = w_0 w_1 w_2 \dots$  starting from  $g(\mathbf{n})$  and such that  $\mathbb{Z} \models \Theta(\mathbf{v}(w_0), \mathbf{v}(w_1))$ . As above, we define  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} w_1$ ,  $\mathbf{t}(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \mathbf{v}(w_1)$  and for all  $i \geq 1$ , we set  $g(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} w_{i+1}$  and  $\mathbf{t}(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} \mathbf{v}(w_{i+1})$ .

**Case  $\phi' = \text{EXE}(\phi_1 U\phi_2)$ .** By  $\mathcal{K}, g(\mathbf{n}) \models \text{EXE}(\phi_1 U\phi_2)$ , there is an infinite path  $\pi = w_0 w_1 w_2 \dots$  starting from  $g(\mathbf{n})$  and  $k \geq 1$  such that  $\mathcal{K}, w_k \models \phi_2$  and for all  $i \in [1, k-1]$  we have  $\mathcal{K}, w_i \models \phi_1 \wedge \neg\phi_2$  ( $k$  is minimal along  $\pi$  to satisfy  $\phi_2$ ). For all  $i \in [1, k]$ , we set  $g(\mathbf{n} \cdot j^i) \stackrel{\text{def}}{=} w_i$  and  $\mathbf{t}(\mathbf{n} \cdot j^i) \stackrel{\text{def}}{=} \mathbf{v}(w_i)$ , and  $g(\mathbf{n} \cdot j^k \cdot 0^i) \stackrel{\text{def}}{=} w_{k+i}$  and  $\mathbf{t}(\mathbf{n} \cdot j^k \cdot 0^i) \stackrel{\text{def}}{=} \mathbf{v}(w_{k+i})$  for all  $i \geq 1$ .

**Case  $\phi' = \text{EXE}(\phi_1 R\phi_2)$ .** If there is an infinite path  $\pi = w_0 w_1 w_2 \dots$  starting from  $g(\mathbf{n})$  such that  $\mathcal{K}, w_i \models \neg\phi_1 \wedge \phi_2$  for all  $i \geq 1$ , then for all  $i > 0$ ,  $g(\mathbf{n} \cdot j^i) \stackrel{\text{def}}{=} w_i$  and  $\mathbf{t}(\mathbf{n} \cdot j^i) \stackrel{\text{def}}{=} \mathbf{v}(w_i)$ . Otherwise, by  $\mathcal{K}, g(\mathbf{n}) \models \text{EXE}(\phi_1 R\phi_2)$  there must exist an infinite path  $\pi = w_0 w_1 w_2 \dots$  starting from  $g(\mathbf{n})$  and  $k \geq 1$  such that  $\mathcal{K}, w_k \models \phi_1 \wedge \phi_2$  and for all  $i \in [0, k-1]$ , we have  $\mathcal{K}, w_i \models \neg\phi_1 \wedge \phi_2$  ( $k$  is minimal along  $\pi$  to satisfy  $\phi_1 \wedge \phi_2$ ), then for all  $i \in [1, k]$ , we set  $g(\mathbf{n} \cdot j^i) \stackrel{\text{def}}{=} w_i$ ,  $\mathbf{t}(\mathbf{n} \cdot j^i) \stackrel{\text{def}}{=} \mathbf{v}(w_i)$ , and  $g(\mathbf{n} \cdot j^k \cdot 0^i) \stackrel{\text{def}}{=} w_{k+i}$ ,  $\mathbf{t}(\mathbf{n} \cdot j^k \cdot 0^i) \stackrel{\text{def}}{=} \mathbf{v}(w_{k+i})$  for all  $i \geq 1$ .

**Case  $\phi' = \text{EX}\psi$  and  $\psi$  is neither an EU-formula nor an ER-formula.**

By  $\mathcal{K}, g(\mathbf{n}) \models \text{EX}\psi$ , there is an infinite path  $\pi = w_0 w_1 w_2 \dots$  starting from  $g(\mathbf{n})$  such that  $\mathcal{K}, w_1 \models \psi'$ .

We set  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} w_1$ ,  $\mathbf{t}(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \mathbf{v}(w_1)$  and for all  $i \geq 1$ ,  $g(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} w_{i+1}$  and  $\mathbf{t}(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} \mathbf{v}(w_{i+1})$ .

It remains to show that for all  $\mathbf{n} \in [0, D]^*$  and for all  $\phi' \in \text{sub}(\phi)$ , if  $\mathcal{K}, g(\mathbf{n}) \models \phi'$ , then  $\mathbf{t}, \mathbf{n} \models \phi'$ , and  $\mathbf{t}$  obeys  $\iota$ , that is,

- if  $\phi' = \text{EX}\phi_1$  and  $\mathbf{t}, \mathbf{n} \models \phi'$ , then  $\mathbf{t}, \mathbf{n} \cdot j \models \phi_1$  with  $j = \iota(\phi')$ ,
- if  $\phi' = \text{E}\phi_1 \cup \phi_2$  and  $\mathbf{t}, \mathbf{n} \models \phi'$ , then there exists some  $k \geq 0$  such that  $\mathbf{t}, \mathbf{n} \cdot j^i \models \phi_1$  for all  $0 \leq i < k$ , and  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_2$  with  $j = \iota(\phi')$ ,
- if  $\phi' = \text{E} \Theta$  and  $\mathbf{t}, \mathbf{n} \models \phi'$ , then  $\mathbb{Z} \models \Theta(\mathbf{t}(\mathbf{n}), \mathbf{t}(\mathbf{n} \cdot j))$  with  $j = \iota(\phi')$ .

The proof is by induction on the subformula relation.

- Suppose  $\mathcal{K}, g(\mathbf{n}) \models \text{E} \Theta$ , and let  $j = \iota(\text{E} \Theta)$  for some  $1 \leq j \leq D$ . By definition of  $\mathbf{t}$ ,  $g(\mathbf{n} \cdot j) = w_1$  and  $g(\mathbf{n} \cdot j \cdot 0^i) = w_{i+1}$ , where  $w_0, w_1, \dots$  is an infinite path in  $\mathcal{K}$  starting from  $g(\mathbf{n})$  satisfying  $\mathbb{Z} \models \Theta(\mathbf{v}(w_0), \mathbf{v}(w_1))$ . We also have  $\mathbf{t}(\mathbf{n}) = \mathbf{v}(w_0)$  and  $\mathbf{t}(\mathbf{n} \cdot j) = \mathbf{v}(w_1)$ , hence the result.
- Suppose  $\mathcal{K}, g(\mathbf{n}) \models \phi_1 \wedge \phi_2$ . Hence  $\mathcal{K}, g(\mathbf{n}) \models \phi_1$  and  $\mathcal{K}, g(\mathbf{n}) \models \phi_2$ , so that by the induction hypothesis  $\mathbf{t}, \mathbf{n} \models \phi_1$  and  $\mathbf{t}, \mathbf{n} \models \phi_2$ , and thus  $\mathbf{t}, \mathbf{n} \models \phi_1 \wedge \phi_2$ .
- Suppose  $\mathcal{K}, g(\mathbf{n}) \models \phi_1 \vee \phi_2$ . Hence  $\mathcal{K}, g(\mathbf{n}) \models \phi_1$  or  $\mathcal{K}, g(\mathbf{n}) \models \phi_2$ , so that by the induction hypothesis  $\mathbf{t}, \mathbf{n} \models \phi_1$  or  $\mathbf{t}, \mathbf{n} \models \phi_2$ , and thus  $\mathbf{t}, \mathbf{n} \models \phi_1 \vee \phi_2$ .
- Suppose  $\mathcal{K}, g(\mathbf{n}) \models \text{EX}\phi'$ . Suppose  $j = \iota(\text{EX}\phi')$  for some  $1 \leq j \leq D$ . We distinguish three cases (coming from the definition of  $\mathbf{t}$ ).
  1.  $\phi'$  is of the form  $\text{E}\phi_1 \cup \phi_2$ : By definition of  $\mathbf{t}$ , we have  $g(\mathbf{n} \cdot j^i) = w_i$  for all  $0 \leq i \leq k$ , where  $w_0 w_1 w_2 \dots$  is a path starting from  $g(\mathbf{n})$  satisfying  $\mathcal{K}, w_k \models \phi_2$  and  $\mathcal{K}, w_i \models \phi_1$  for all  $1 \leq i < k$ . By the induction hypothesis, we have  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_2$  and  $\mathcal{K}, \mathbf{n} \cdot j^i \models \phi_1$  for all  $1 \leq i < k$ . Hence  $\mathbf{t}, \mathbf{n} \models \text{EX}\phi'$ .
  2.  $\phi'$  is of the form  $\text{E}\phi_1 \text{R}\phi_2$ : By definition of  $\mathbf{t}$ , there are two cases.
    - Either  $g(\mathbf{n} \cdot j^i) = w_i$  for all  $i \geq 0$ , where  $w_0 w_1 w_2 \dots$  is a path starting from  $\mathbf{n}$  such that  $\mathcal{K}, w_i \models \neg\phi_1 \wedge \phi_2$  for all  $i \geq 1$ . By the induction hypothesis, we have  $\mathbf{t}, \mathbf{n} \cdot j^i \models \neg\phi_1 \wedge \phi_2$  for all  $i \geq 1$ , and hence  $\mathbf{t}, \mathbf{n} \models \text{EX}\phi'$ .
    - Or  $g(\mathbf{n} \cdot j^i) = w_i$  for all  $0 \leq i \leq k$ , where  $w_0 w_1 w_2 \dots$  is a path starting from  $\mathbf{n}$  such that  $\mathcal{K}, w_k \models \phi_1 \wedge \phi_2$  and  $\mathcal{K}, w_i \models \neg\phi_1 \wedge \phi_2$  for all  $1 \leq i < k$ . By the induction hypothesis, we have  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_1 \wedge \phi_2$  and  $\mathbf{t}, \mathbf{n} \cdot j^i \models \neg\phi_1 \wedge \phi_2$  for all  $1 \leq i < k$ . Hence  $\mathbf{t}, \mathbf{n} \models \text{EX}\phi'$ .
  3.  $\phi'$  is neither an EU-formula nor an ER-formula: then  $g(\mathbf{n} \cdot j) = w_1$  and  $g(\mathbf{n} \cdot j \cdot 0^i) = w_{i+1}$ , where  $w_0 w_1 w_2 \dots$  is a path from  $g(\mathbf{n})$  such that  $\mathcal{K}, w_1 \models \phi'$ . By the induction hypothesis, we have  $\mathbf{t}, \mathbf{n} \cdot j \models \phi'$ , and hence  $\mathbf{t}, \mathbf{n} \models \text{EX}\phi'$ .
- Suppose  $\mathcal{K}, g(\mathbf{n}) \models \text{E}\phi_1 \cup \phi_2$ . We distinguish two cases.
  - Suppose  $\mathcal{K}, g(\mathbf{n}) \models \phi_2$ . By the induction hypothesis,  $\mathbf{t}, \mathbf{n} \models \phi_2$  and hence  $\mathbf{t}, \mathbf{n} \models \text{E}\phi_1 \cup \phi_2$ .
  - Suppose  $\mathcal{K}, g(\mathbf{n}) \not\models \phi_2$ . Then  $\mathcal{K}, g(\mathbf{n}) \models \phi_1$  and  $\mathcal{K}, g(\mathbf{n}) \models \text{EXE}\phi_1 \cup \phi_2$ . By the induction hypothesis, we also have  $\mathbf{t}, \mathbf{n} \models \phi_1$ . Suppose  $\iota(\text{EXE}\phi_1 \cup \phi_2) = j$  for some  $1 \leq j \leq D$ . By definition of  $\mathbf{t}$ , there exists some (minimal)  $k \geq 1$  such that  $g(\mathbf{n} \cdot j^i) = w_i$  for all  $0 \leq i \leq k$ , where  $w_0 w_1 w_2 \dots$  is a path starting from  $g(\mathbf{n})$  satisfying  $\mathcal{K}, w_k \models \phi_2$  and  $\mathcal{K}, w_i \models \phi_1$  for all  $1 \leq i < k$ . By the induction hypothesis, we have  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_2$  and  $\mathbf{t}, \mathbf{n} \cdot j^i \models \phi_1$  for all  $1 \leq i < k$ . Recall that we also have  $\mathbf{t}, \mathbf{n} \models \phi_1$ , so that indeed  $\mathbf{t}, \mathbf{n} \models \text{E}\phi_1 \cup \phi_2$ .
- Suppose  $\mathcal{K}, g(\mathbf{n}) \models \text{E}\phi_1 \text{R}\phi_2$ . We distinguish two cases.
  - Suppose  $\mathcal{K}, g(\mathbf{n}) \models \phi_2$  and  $\mathcal{K}, g(\mathbf{n}) \models \phi_1$ . By the induction hypothesis,  $\mathbf{t}, \mathbf{n} \models \phi_2$  and  $\mathbf{t}, \mathbf{n} \models \phi_1$ , so that  $\mathbf{t}, \mathbf{n} \models \text{E}\phi_1 \text{R}\phi_2$ .

- Suppose  $\mathcal{K}, g(\mathbf{n}) \models \neg\phi_1 \wedge \phi_2$  and  $\mathcal{K}, g(\mathbf{n}) \models \text{EXE}\phi_1 R\phi_2$ . By the induction hypothesis,  $\mathbf{t}, \mathbf{n} \models \neg\phi_1 \wedge \phi_2$ . Suppose  $\iota(\text{EXE}\phi_1 R\phi_2) = j$  for some  $1 \leq j \leq D$ . We distinguish two more cases.
  - \* There is an infinite path  $w_0 w_1 w_2 \dots$  from  $g(\mathbf{n})$  such that  $\mathcal{K}, w_i \models \neg\phi_1 \wedge \phi_2$  for all  $i \geq 1$ . We then have  $g(\mathbf{n} \cdot j^i) = w_i$  for all  $i \geq 1$ . By the induction hypothesis, we have  $\mathbf{t}, \mathbf{n} \cdot j^i \models \neg\phi_1 \wedge \phi_2$  for all  $i \geq 1$ . Together with  $\mathbf{t}, \mathbf{n} \models \neg\phi_1 \wedge \phi_2$ , we obtain  $\mathbf{t}, \mathbf{n} \models \text{E}\phi_1 R\phi_2$ .
  - \* Otherwise, we have  $g(\mathbf{n} \cdot j^k) = w_k$  and  $g(\mathbf{n} \cdot j^i) = w_i$ , where  $w_0 w_1 w_2 \dots$  is a path from  $g(\mathbf{n})$  such that  $\mathcal{K}, w_k \models \phi_1 \wedge \phi_2$  and  $\mathcal{K}, w_i \models \neg\phi_1 \wedge \phi_2$  for all  $1 \leq i < k$ . By the induction hypothesis, we have  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_1 \wedge \phi_2$ , and  $\mathbf{t}, \mathbf{n} \cdot j^i \models \neg\phi_1 \wedge \phi_2$  for all  $1 \leq i < k$ . Together with  $\mathbf{t}, \mathbf{n} \models \neg\phi_1 \wedge \phi_2$ , we obtain  $\mathbf{t}, \mathbf{n} \models \text{E}\phi_1 R\phi_2$ .
- Suppose  $\mathcal{K}, g(\mathbf{n}) \models \text{A } \Theta$ . By construction of  $\mathbf{t}$ , for all infinite paths  $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \dots \in [0, D]^\omega$ ,  $g(\mathbf{n}) \cdot g(\mathbf{n} \cdot j_1) \cdot g(\mathbf{n} \cdot j_1 j_2) \dots$  is an infinite path from  $g(\mathbf{n})$ . Consequently, for all  $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \dots \in [0, D]^\omega$ , we have  $\mathbb{Z} \models \Theta(\mathbf{v}(g(\mathbf{n})), \mathbf{v}(g(\mathbf{n} \cdot j_1)))$  and therefore  $\mathbb{Z} \models \Theta(\mathbf{t}(\mathbf{n}), \mathbf{t}(\mathbf{n} \cdot j_1))$  because by definition, we have  $\mathbf{t}(\mathbf{n}) = \mathbf{v}(g(\mathbf{n}))$  and  $\mathbf{t}(\mathbf{n} \cdot j) = \mathbf{v}(g(\mathbf{n} \cdot j))$ . As a consequence,  $\mathbf{t}, \mathbf{n} \models \text{A } \Theta$ .
- Suppose  $\mathcal{K}, g(\mathbf{n}) \models \text{A } \Phi$ , where  $\Phi$  is a path formula such that  $\Phi$  is not a constraint. By construction of  $\mathbf{t}$ , for all infinite paths  $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \dots \in [0, D]^\omega$ ,  $g(\mathbf{n}) \cdot g(\mathbf{n} \cdot j_1) \cdot g(\mathbf{n} \cdot j_1 j_2) \dots$  is an infinite path from  $g(\mathbf{n})$ . By way of example, assume that  $\Phi = \phi_1 U \phi_2$  (the cases  $\Phi = \phi_1 R \phi_2$  and  $\Phi = X\phi_1$  are handled in the very same way). For all infinite paths  $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \dots \in [0, D]^\omega$  with  $\pi = g(\mathbf{n}) \cdot g(\mathbf{n} \cdot j_1) \cdot g(\mathbf{n} \cdot j_1 j_2) \dots$ , we have  $\mathcal{K}, \pi \models \phi_1 U \phi_2$  and therefore there is  $k \in \mathbb{N}$  such that  $\mathcal{K}, g(\mathbf{n} \cdot j_1 \dots j_k) \models \phi_2$  and for all  $0 \leq k' < k$ , we have  $\mathcal{K}, g(\mathbf{n} \cdot j_1 \dots j_{k'}) \models \phi_1$ . By the induction hypothesis, we have  $\mathbf{t}, \mathbf{n} \cdot j_1 \dots j_k \models \phi_2$  and  $\mathbf{t}, \mathbf{n} \cdot j_1 \dots j_{k'} \models \phi_1$  for all  $0 \leq k' < k$ . Consequently,  $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \dots \models \phi_1 U \phi_2$  and therefore  $\mathbf{t}, \mathbf{n} \models \text{A } \phi_1 U \phi_2$  because the above path from  $\mathbf{n}$  was arbitrary in  $\mathbf{t}$ .  $\square$

### 3 Tree Constraint Automata

#### 3.1 A simple and natural definition

In this section, we introduce the class of tree constraint automata that accept sets of trees of the form  $\mathbf{t} : [0, D-1]^* \rightarrow (\Sigma \times \mathbb{Z}^\beta)$  for some finite alphabet  $\Sigma$  and some  $\beta \geq 1$ . The transition relation of such automata states constraints between the  $\beta$  integer values at a node and the integer values at its children nodes. To do so, we write  $\text{TreeCons}(\beta)$  to denote the Boolean constraints built over the terms  $\mathbf{x}_1, \dots, \mathbf{x}_\beta, X\mathbf{x}_1, \dots, X\mathbf{x}_\beta$ . These constraints are used to define the transition relation of such automata. As usual, we also write  $\mathbf{x}'_i$  to denote the term  $X\mathbf{x}_i$ , and we shall use valuations  $\mathbf{v}$  with profile  $\{\mathbf{x}_i, \mathbf{x}'_i \mid i \in [1, \beta]\} \rightarrow \mathbb{Z}$ . In forthcoming Definition 1, the acceptance condition on infinite branches is a Büchi condition, but this can be easily extended to more general conditions (which we already consider by the end of this section). Moreover, Definition 1 is specific to the concrete domain  $\mathbb{Z}$  but it can be easily adapted to other concrete domains.

**Definition 1.** A tree constraint automaton (TCA, for short) is a tuple  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{in}, \delta, F)$ , where

- $Q$  is a finite set of locations;  $\Sigma$  is a finite alphabet,
- $D \geq 1$  is the (branching) degree of (the trees accepted by)  $\mathbb{A}$ ,
- $\beta \geq 1$  is the number of variables (a.k.a. registers),
- $Q_{in} \subseteq Q$  is the set of initial locations,
- $\delta$  is a finite subset of  $Q \times \Sigma \times (\text{TreeCons}(\beta) \times Q)^D$ , the transition relation. That is,  $\delta$  consists of tuples of the form  $(q, \mathbf{a}, (\Theta_0, q_0), \dots, (\Theta_{D-1}, q_{D-1}))$ , where  $q, q_0, \dots, q_{D-1} \in Q$ ,  $\mathbf{a} \in \Sigma$ , and  $\Theta_0, \dots, \Theta_{D-1}$  are constraints built over  $\mathbf{x}_1, \dots, \mathbf{x}_\beta, \mathbf{x}'_1, \dots, \mathbf{x}'_\beta$ .

- $F \subseteq Q$  encodes the Büchi acceptance condition.

Let  $\mathbf{t} : [0, D-1]^* \rightarrow (\Sigma \times \mathbb{Z}^\beta)$  be an infinite full  $D$ -ary tree over  $\Sigma \times \mathbb{Z}^\beta$ . A *run* of  $\mathbb{A}$  on  $\mathbf{t}$  is a mapping  $\rho : [0, D-1]^* \rightarrow Q$  satisfying the following conditions:

- $\rho(\varepsilon) \in Q_{\text{in}}$ ;
- for every  $\mathbf{n} \in [0, D-1]^*$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \mathbf{z})$  and  $\rho(\mathbf{n}) = q$ ,  $\mathbf{t}(\mathbf{n} \cdot i) = (\mathbf{a}_i, \mathbf{z}_i)$  and  $\rho(\mathbf{n} \cdot i) = q_i$  for all  $0 \leq i < D$ , there exists a transition

$$(q, \mathbf{a}, (\Theta_0, q_0), \dots, (\Theta_{D-1}, q_{D-1})) \in \delta$$

and  $\mathbb{Z} \models \Theta_i(\mathbf{z}, \mathbf{z}_i)$  for all  $0 \leq i < D$ .

Suppose  $\rho$  is a run of  $\mathbb{A}$  on  $\mathbf{t}$ . Given a path  $\pi = j_1 \cdot j_2 \cdot j_3 \dots$  in  $\rho$  starting from the root, we define  $\text{inf}(\rho, \pi)$  to be the set of control states that appear infinitely often in  $\rho(\varepsilon)\rho(j_1)\rho(j_1 \cdot j_2)\rho(j_1 \cdot j_2 \cdot j_3) \dots$ . A run  $\rho$  is *accepting* if for all paths  $\pi$  in  $\rho$  starting from  $\varepsilon$ , we have  $\text{inf}(\rho, \pi) \cap F \neq \emptyset$ . We write  $L(\mathbb{A})$  to denote the set of trees  $\mathbf{t}$  that admit an accepting run. As usual, the *nonemptiness problem for TCA*, written  $\text{NE}(\text{TCA})$ , is defined as follows.

**Input:** A tree constraint automaton  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$ ,

**Question:** Is  $L(\mathbb{A})$  non-empty ?

Unlike (plain) Büchi tree automata [VW86], the number of transitions in a tree constraint automaton is *a priori* unbounded ( $\text{TreeCons}(\beta)$  is infinite) and the maximal size of a constraint occurring in transitions is unbounded too. In particular, this means that  $\text{card}(\delta)$  is a priori unbounded, even if  $Q$  and  $\Sigma$  are fixed. We write  $\text{MaxConsSize}(\mathbb{A})$  to denote the maximal size of a constraint occurring in  $\mathbb{A}$ .

The complexity of the nonemptiness problem should therefore also take into account these parameters. However, though  $\text{TreeCons}(\beta)$  is infinite, for a fixed finite set of constants  $\mathfrak{d}_1, \dots, \mathfrak{d}_\alpha$  (we always assume  $\alpha \geq 1$ ), there is a finite set of logically non-equivalent constraints in  $\text{TreeCons}(\beta)$  built over  $\mathfrak{d}_1, \dots, \mathfrak{d}_\alpha$  (double exponential magnitude in  $\alpha + \beta$ ). We shall control the use of the constraints in  $\text{TreeCons}(\beta)$  by introducing the forthcoming sets  $\text{SatTypes}(\beta)$ , see Section 4.2. Note also that our tree automaton model differs from the Presburger Büchi tree automata from [SSM08,BF22] for which, in the runs, arithmetical expressions are related to constraints between the numbers of children labelled by different locations. Herein, the arithmetical expressions state constraints between data values (not necessarily at the same node).

*TCA on  $\mathbb{N}$ .* We can easily handle the model of tree constraint automata accepting trees of the form  $[0, D-1]^* \rightarrow \Sigma \times \mathbb{N}^\beta$  (i.e. by requiring that all the integer values are natural numbers). Indeed, consider a TCA  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$  accepting only trees of the form  $[0, D-1]^* \rightarrow (\Sigma \times \mathbb{N}^\beta)$  (by definition). Let  $\mathbb{A}' = (Q, \Sigma, D, \beta + 1, Q_{\text{in}}, \delta', F)$  be the TCA accepting trees of the form  $[0, D-1]^* \rightarrow (\Sigma \times \mathbb{Z}^{\beta+1})$  which is defined as  $\mathbb{A}$  except that all the constraints  $\Theta \in \text{TreeCons}(\beta)$  (built over the concrete domain  $\mathbb{N}$ ) occurring in  $\delta$ , are replaced by  $\Theta' \in \text{TreeCons}(\beta + 1)$  obtained from  $\Theta$  by adding the conjunct  $\mathbf{x}_{\beta+1} = 0 \wedge \bigwedge_{1 \leq j \leq \beta} (\mathbf{x}_j = \mathbf{x}_{\beta+1} \vee \mathbf{x}_j > \mathbf{x}_{\beta+1})$ .

Given a tree  $\mathbf{t} : [0, D-1]^* \rightarrow (\Sigma \times \mathbb{Z}^{\beta+1})$ , we write  $\mathbf{t}[1 \dots \beta]$  to denote the tree  $\mathbf{t}[1 \dots \beta] : [0, D-1]^* \rightarrow (\Sigma \times \mathbb{Z}^\beta)$  such that for all  $\mathbf{n} \in [0, D-1]^*$ , if  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, (\mathfrak{d}_1, \dots, \mathfrak{d}_{\beta+1}))$ , then  $\mathbf{t}[1 \dots \beta](\mathbf{n}) \stackrel{\text{def}}{=} (\mathbf{a}, (\mathfrak{d}_1, \dots, \mathfrak{d}_\beta))$ . Consequently, we have  $L(\mathbb{A}) = \{\mathbf{t}[1 \dots \beta] \mid \mathbf{t} \in L(\mathbb{A}')\}$ . Hence  $L(\mathbb{A}) \neq \emptyset$  iff  $L(\mathbb{A}') \neq \emptyset$ .

*Generalised TCA.* Below, we introduce the class of tree constraint automata with generalised Büchi conditions, which will be handy in the sequel. Unsurprisingly, as for generalised Büchi automata, generalised Büchi conditions can be encoded by Büchi conditions with a polynomial blow-up only.

A *generalised tree constraint automaton* (GTCA, for short) is a tuple  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, \mathcal{F})$  defined as TCA except that  $\mathcal{F} \subseteq \mathcal{P}(Q)$  is the generalised Büchi condition. Runs for GTCA are defined as above and a run  $\rho$  is *accepting* if for all paths  $\pi$  in  $\rho$  starting from  $\varepsilon$ , for all  $F \in \mathcal{F}$ , we have  $\inf(\rho, \pi) \cap F \neq \emptyset$ .

Given a GTCA  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, \mathcal{F})$  with  $\mathcal{F} = \{F_1, \dots, F_k\}$ , we write  $\mathbb{A}' = (Q', \Sigma, D, \beta, Q'_{\text{in}}, \delta', F)$  to denote the TCA below.

- $Q' \stackrel{\text{def}}{=} [0, k] \times Q$ ,  $Q'_{\text{in}} \stackrel{\text{def}}{=} \{0\} \times Q_{\text{in}}$  and  $F \stackrel{\text{def}}{=} \{0\} \times Q$ .
- Given  $i \in [0, k]$  and  $q \in Q$ , we write  $\text{next}(i, q)$  to denote the copy number in  $[0, k]$  such that  $\text{next}(0, q) = 1$  for all  $q \in Q$ ,  $\text{next}(i, q) = i$  if  $q \notin F_i$  and  $\text{next}(i, q) = i + 1 \bmod (k + 1)$  if  $q \in F_i$ . The transition relation  $\delta'$  is defined as follows:  $((i, q), \mathbf{a}, (\Theta_0, (i_0, q_0)), \dots, (\Theta_{D-1}, (i_{D-1}, q_{D-1}))) \in \delta' \stackrel{\text{def}}{\iff}$  there is  $(q, \mathbf{a}, (\Theta_0, q_0), \dots, (\Theta_{D-1}, q_{D-1})) \in \delta$  and for all  $j \in [0, D - 1]$ ,  $i_j = \text{next}(i, q_j)$ .

As for generalised Büchi automata, one can show that  $L(\mathbb{A}) \neq \emptyset$  iff  $L(\mathbb{A}') \neq \emptyset$ , and the size of  $\mathbb{A}'$  is quadratic in the size of  $\mathbb{A}$ . Therefore, in the sequel, using GTCA instead of TCA has no consequence on worst-case complexity results.

*Rabin pairs and Streett pairs.* Tree constraint automata can be also defined with the acceptance condition made of a finite set of Rabin pairs  $\{(L_1, U_1), \dots, (L_k, U_k)\}$ , see more details in Section 4.5. A run  $\rho$  is accepting if for some  $i \in [1, k]$ ,  $\inf(\rho, \pi) \cap L_i \neq \emptyset$  and  $\inf(\rho, \pi) \cap U_i = \emptyset$ . Similarly, tree constraint automata can be defined with the acceptance condition made of a finite set of Streett pairs  $\{(L_1, U_1), \dots, (L_k, U_k)\}$  (also called *completely mented pairs* in the literature). A run  $\rho$  is accepting if for all  $i \in [1, k]$ , if  $\inf(\rho, \pi) \cap L_i \neq \emptyset$ , then  $\inf(\rho, \pi) \cap U_i \neq \emptyset$ . Along the paper, we use the above notations to represent Rabin/Streett acceptance conditions.

### 3.2 From CTL( $\mathbb{Z}$ ) formulae to generalised tree constraint automata

Let  $\phi$  be a CTL( $\mathbb{Z}$ ) formula in simple form built over  $\mathbf{x}_1, \dots, \mathbf{x}_\beta, \mathbf{X}\mathbf{x}_1, \dots, \mathbf{X}\mathbf{x}_\beta$ . Let  $D \stackrel{\text{def}}{=} \text{card}(\text{sub}_{\text{EX}}(\phi)) + \text{card}(\text{sub}_{\text{E}}(\phi))$  and  $\iota : \text{sub}_{\text{EX}}(\phi) \cup \text{sub}_{\text{E}}(\phi) \rightarrow [1, D]$  be a direction map. We build a generalised tree constraint automaton  $\mathbb{A}_\phi = (Q, \Sigma, D+1, \beta, Q_{\text{in}}, \delta, \mathcal{F})$  such that  $\phi$  is satisfiable if, and only if,  $L(\mathbb{A}_\phi) \neq \emptyset$ . The automaton  $\mathbb{A}_\phi$  accepts infinite trees of the form  $\text{tree } \mathbf{t} : [0, D]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$ . Let us define  $\mathbb{A}_\phi$  formally.

- $\Sigma \stackrel{\text{def}}{=} \{\dagger\}$  ( $\dagger$  is an arbitrary letter here).
- $Q$  is the subset of  $[0, D] \times \mathcal{P}(\text{sub}(\phi))$  such that  $(i, X)$  belongs to  $Q$  only if  $X$  is propositionally consistent. The first argument records the direction, as indicated by  $\iota$ .
- $Q_{\text{in}} \stackrel{\text{def}}{=} \{(0, X) \in Q \mid \phi \in X\}$ .
- The transition relation  $\delta$  is made of tuples of the form

$$((i, X), \dagger, (\Theta_0, (0, X_0)), \dots, (\Theta_D, (D, X_D)))$$

verifying the conditions below.

1. For all  $\text{EX}\psi \in X$ , we have  $\psi \in X_{\iota(\text{EX}\psi)}$ .
2. For all  $\text{AX}\psi \in X$  and  $j \in [0, D]$ , we have  $\psi \in X_j$ .

3. For all  $j \in [0, D]$ , if there is  $\mathbf{E} \Theta \in X$  such that  $\iota(\mathbf{E} \Theta) = j$ , then

$$\Theta_j \stackrel{\text{def}}{=} \left( \bigwedge_{\Theta' \in X} \Theta' \right) \wedge \Theta \quad \text{otherwise,} \quad \Theta_j \stackrel{\text{def}}{=} \bigwedge_{\Theta' \in X} \Theta'.$$

- $\mathcal{F}$  is made of two types of sets, those parameterised by some element in  $\text{sub}_{\mathbf{EU}}(\phi)$  and those parameterised by some element in  $\text{sub}_{\mathbf{AU}}(\phi)$ . The subformulae whose outermost connective is either  $\mathbf{ER}$  or  $\mathbf{AR}$  do not impose additional acceptance conditions. For each  $\mathbf{E}\psi_1\mathbf{U}\psi_2 \in \text{sub}_{\mathbf{EU}}(\phi)$ , the set  $F_{\mathbf{E}\psi_1\mathbf{U}\psi_2}$  belongs to  $\mathcal{F}$ :

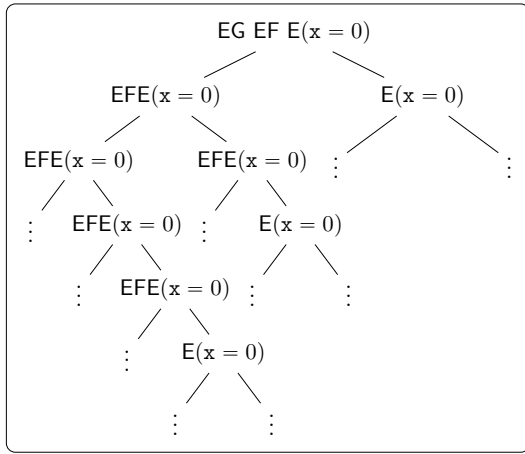
$$F_{\mathbf{E}\psi_1\mathbf{U}\psi_2} \stackrel{\text{def}}{=} \{(i, X) \in Q \mid i \neq \iota(\mathbf{E}\mathbf{X}\mathbf{E}\psi_1\mathbf{U}\psi_2) \text{ or } \psi_2 \in X \text{ or } \mathbf{E}\psi_1\mathbf{U}\psi_2 \notin X\}.$$

For each  $\mathbf{A}\psi_1\mathbf{U}\psi_2 \in \text{sub}_{\mathbf{AU}}(\phi)$ , the set  $F_{\mathbf{A}\psi_1\mathbf{U}\psi_2}$  belongs to  $\mathcal{F}$ :

$$F_{\mathbf{A}\psi_1\mathbf{U}\psi_2} \stackrel{\text{def}}{=} \{(i, X) \in Q \mid \psi_2 \in X \text{ or } \mathbf{A}\psi_1\mathbf{U}\psi_2 \notin X\}.$$

The construction of  $\mathbb{A}_\phi$  is mainly inspired from [VW08, page 702] for CTL formulae but a few essential differences need to be pointed out here. Obviously, our construction handles constraints, which is expected since  $\text{CTL}(\mathbb{Z})$  extends CTL by adding constraints between data values. More importantly, [VW08, section 5.2] uses another tree automaton model and  $F_{\mathbf{E}\psi_1\mathbf{U}\psi_2}$  and  $F_{\mathbf{A}\psi_1\mathbf{U}\psi_2}$  herein differ from [VW08, page 702] by the use of the direction map.

Let us provide below an example that illustrates the use of a direction map. Let  $\phi$  be the formula  $\mathbf{EG} \mathbf{EF} \mathbf{E}(\mathbf{x} = 0)$  that states the existence of a path  $\pi$  along which at every position  $i$  there is path  $\pi_i$  leading to a position  $k_i$  such that  $\mathbf{x}$  is equal to zero. Below, we present a tree-like infinite Kripke structure satisfying  $\phi$  on its root node. The path  $\pi$  is the leftmost branch,  $\pi_i$  is the rightmost branch from  $\pi(i)$  with  $k_i = i + 1$ . The Kripke structure can be also understood as a subtree extracted from an accepting run for  $\mathbb{A}_\phi$  such that we use a direction map  $\iota$  with  $\iota(\mathbf{E}\mathbf{X}\mathbf{E}\mathbf{F}\mathbf{E}(\mathbf{x} = 0))$  defined as the greatest direction (a lot of information is missing about the label of each node).



By definition of  $\mathbb{A}_\phi$ , on the path  $\pi$ , we do not require that infinitely often  $\mathbf{E}(\mathbf{x} = 0)$  or not  $\mathbf{EFE}(\mathbf{x} = 0)$  (indeed, none of these conditions hold on the leftmost path  $\pi$ ), but on every rightmost branch we do so, because this is the type of branch that we use to witness the satisfaction of  $\mathbf{EFE}(\mathbf{x} = 0)$ . This explains why the construction uses a direction map that is involved in the acceptance condition.

### 3.3 Correctness

We would like to show the following result.

**Lemma 1.**  $\phi$  is satisfiable if, and only if,  $L(\mathbb{A}_\phi) \neq \emptyset$ .

*Proof.* “if:” Suppose  $L(\mathbb{A}_\phi) \neq \emptyset$ . Then there exists a tree  $\mathbf{t} : [0, D]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$  in  $L(\mathbb{A}_\phi)$ . Let  $\rho : [0, D]^* \rightarrow [0, D] \times \mathcal{P}(\text{sub}(\phi))$  be an accepting run of  $\mathbb{A}_\phi$  on  $\mathbf{t}$ . We prove for all  $\phi' \in \text{sub}(\phi)$ , for all nodes  $\mathbf{n}$  in  $\mathbf{t}$  with  $\mathbf{t}(\mathbf{n}) = (\dagger, \mathbf{z})$ ,  $\rho(\mathbf{n}) = (d_{\mathbf{n}}, X_{\mathbf{n}})$  and  $\rho(\mathbf{n} \cdot i) = (d_{\mathbf{n} \cdot i}, X_{\mathbf{n} \cdot i})$  for all  $0 \leq i \leq D$ , if  $\phi' \in X_{\mathbf{n}}$ , then  $\mathbf{t}, \mathbf{n} \models \phi'$ . Here,  $\mathbf{t}$  is also understood as a tree Kripke model (in which we ignore the finite alphabet  $\Sigma$ ). Note that this indeed implies  $\mathbf{t}, \varepsilon \models \phi$ , as  $\rho$  starts with some initial location of the form  $(0, X_\varepsilon)$  such that  $\phi \in X_\varepsilon$ . Hence  $\phi$  is indeed satisfiable.

So let  $\mathbf{n}$  be a node in  $\mathbf{t}$ . Since  $\rho$  is a run of  $\mathbb{A}_\phi$ , we can infer that a transition of the form  $((d_{\mathbf{n}}, X_{\mathbf{n}}), \dagger, (\Theta_0, (0, X_0)), \dots, (\Theta_D, (D, X_D)))$  is in  $\delta$ , and

1. for all  $\text{EX}\psi \in X_{\mathbf{n}}$ , we have  $\psi \in X_{\mathbf{n} \cdot j}$  with  $j = \iota(\text{EX}\psi)$ ;
2. for all  $\text{AX}\psi \in X_{\mathbf{n}}$  and  $j \in [0, D]$ , we have  $\psi \in X_{\mathbf{n} \cdot j}$ ;
3. for all  $j \in [0, D]$ , if there is  $\text{E } \Theta \in X_{\mathbf{n}}$  such that  $\iota(\text{E } \Theta) = j$ , then

$$\Theta_j \stackrel{\text{def}}{=} \left( \bigwedge_{\text{A}\Theta' \in X_{\mathbf{n}}} \Theta' \right) \wedge \Theta \quad \text{otherwise,} \quad \Theta_j \stackrel{\text{def}}{=} \bigwedge_{\text{A}\Theta' \in X_{\mathbf{n}}} \Theta'.$$

The proof of the claim is by the induction on the formula with respect to the subformula ordering. So suppose  $\phi' \in X_{\mathbf{n}}$ .

- $\phi' = \text{E } \Theta$ , where  $\Theta$  is a Boolean combination of terms over  $\mathbf{x}_1, \dots, \mathbf{x}_\beta, \mathbf{x}'_1, \dots, \mathbf{x}'_\beta$ . Suppose  $\iota(\phi') = j$  for some  $1 \leq j \leq D$ . From condition 3. above, we obtain that  $\Theta_j \stackrel{\text{def}}{=} \left( \bigwedge_{\text{A}\Theta' \in X_{\mathbf{n}}} \Theta' \right) \wedge \Theta$ . Since  $\rho$  is a run, we have  $\mathbb{Z} \models \Theta_j(\mathbf{z}, \mathbf{z}_j)$ , where  $\mathbf{t}(\mathbf{n}) = (\dagger, \mathbf{z})$  and  $\mathbf{t}(\mathbf{n} \cdot j) = (\dagger, \mathbf{z}_j)$ , and hence also  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_j)$ . This yields  $\mathbf{t}, \mathbf{n} \models \text{E } \Theta$ .
- $\phi' = \text{A } \Theta$ , where  $\Theta$  is a Boolean combination of atomic constraints built over  $\mathbf{x}_1, \dots, \mathbf{x}_\beta, \mathbf{x}'_1, \dots, \mathbf{x}'_\beta$ . From condition 3. above, we know that  $\Theta$  appears in  $\Theta_j$  as a conjunct. Since  $\rho$  is a run, we have  $\mathbb{Z} \models \Theta_j(\mathbf{z}, \mathbf{z}_j)$  for all  $0 \leq j \leq D$ , where  $\mathbf{t}(\mathbf{n}) = (\dagger, \mathbf{z})$  and  $\mathbf{t}(\mathbf{n} \cdot j) = (\dagger, \mathbf{z}_j)$ , and hence also  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_j)$  for all  $0 \leq j \leq D$ . This yields  $\mathbf{t}, \mathbf{n} \models \text{A } \Theta$ .
- $\phi' = \phi_1 \wedge \phi_2$ . Since  $X_{\mathbf{n}}$  is propositionally consistent, we have  $\phi_1, \phi_2 \in X_{\mathbf{n}}$ . By the induction hypothesis,  $\mathbf{t}, \mathbf{n} \models \phi_1$  and  $\mathbf{t}, \mathbf{n} \models \phi_2$ . Hence  $\mathbf{t}, \mathbf{n} \models \phi'$ .
- Similarly for  $\phi' = \phi_1 \vee \phi_2$ .
- $\phi' = \text{EX}\phi_1$ . From condition 1. above we know that  $\phi_1 \in X_{\mathbf{n} \cdot i}$ , where  $i = \iota(\phi')$ . By the induction hypothesis,  $\mathbf{t}, \mathbf{n} \cdot i \models \phi_1$ . Hence  $\mathbf{t}, \mathbf{n} \models \phi'$ .
- $\phi' = \text{AX}\phi_1$ . From condition 2. above, we know that  $\phi_1 \in X_{\mathbf{n} \cdot i}$  for all  $0 \leq i \leq D$ . By the induction hypothesis,  $\mathbf{t}, \mathbf{n} \cdot i \models \phi_1$  for all  $0 \leq i \leq D$ . Hence  $\mathbf{t}, \mathbf{n} \models \phi'$ .
- $\phi' = \text{E}\phi_1 \cup \phi_2$ . Since  $X_{\mathbf{n}}$  is propositionally consistent, we have  $\phi_2 \in X_{\mathbf{n}}$ , or  $\phi_1, \text{EX}\phi' \in X_{\mathbf{n}}$ . In the first case, we have  $\mathbf{t}, \mathbf{n} \models \phi_2$  by the induction hypothesis, and hence  $\mathbf{t}, \mathbf{n} \models \phi'$ . In the second case, we have  $\mathbf{t}, \mathbf{n} \models \phi_1$  by the induction hypothesis, and we have  $\phi' \in X_{\mathbf{n} \cdot j}$ , where  $j = \iota(\text{EX}\phi')$ . Consider the infinite path  $\mathbf{n} \cdot j \cdot j \cdot j \dots$  starting from  $\mathbf{n}$ . Since  $\rho$  is an accepting run, states from  $F_{\phi'}$  must occur infinitely often along this path; that is, there must exist some  $k \geq 0$  such that  $\phi_2 \in X_{\mathbf{n} \cdot j^k}$  or  $\text{E}\phi_1 \cup \phi_2 \notin X_{\mathbf{n} \cdot j^k}$ . Let  $k \geq 0$  be minimal. We first show by induction that if  $\phi' \in X_{\mathbf{n} \cdot j^i}$ , then  $\mathbf{t}, \mathbf{n} \cdot j^i \models \phi_1$  and  $\phi' \in X_{\mathbf{n} \cdot j^{i+1}}$  for all  $0 \leq i < k$ . For  $i = 0$ , we have shown this above. So let  $0 < i < k$  and suppose  $\phi' \in X_{\mathbf{n} \cdot j^i}$ . Since  $\rho$  is a run,  $X_{\mathbf{n} \cdot j^i}$  is propositionally consistent and hence  $\phi_2 \in X_{\mathbf{n} \cdot j^i}$  or  $\phi_1, \text{EX}\phi' \in X_{\mathbf{n} \cdot j^i}$ . Note that  $\phi_2 \in X_{\mathbf{n} \cdot j^i}$  contradicts that  $k$  is minimal; hence  $\phi_1, \text{EX}\phi' \in X_{\mathbf{n} \cdot j^i}$ . By the induction hypothesis  $\mathbf{t}, \mathbf{n} \cdot j^i \models \phi_1$ , and  $\phi' \in X_{\mathbf{n} \cdot j^{i+1}}$ . We conclude that  $\mathbf{t}, \mathbf{n} \cdot j^i \models \phi_1$  for all  $0 \leq i < k$ . Next we prove that  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_2$ . Recall that  $\phi_2 \in X_{\mathbf{n} \cdot j^k}$  or  $\phi' \notin X_{\mathbf{n} \cdot j^k}$ . Note that  $\phi' \in X_{\mathbf{n} \cdot j^k}$  by what we just proved before. Hence  $\phi_2 \in X_{\mathbf{n} \cdot j^k}$ . By the induction hypothesis, we obtain  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_2$ . We hence have proved the existence of a path starting in  $\mathbf{n}$  and satisfying  $\phi_1 \cup \phi_2$ ; hence  $\mathbf{t}, \mathbf{n} \models \phi'$ .



- $\phi' = A\phi_1 U \phi_2$ . Since  $X_{\mathbf{n}}$  is propositionally consistent, we have  $\phi_2 \in X_{\mathbf{n}}$ , or  $\phi_1, AX\phi' \in X_{\mathbf{n}}$ . In the first case, we have  $\mathbf{t}, \mathbf{n} \models \phi_2$  by the induction hypothesis, and hence  $\mathbf{t}, \mathbf{n} \models \phi'$ . In the second case, we have  $\mathbf{t}, \mathbf{n} \models \phi_1$  by the induction hypothesis, and we have  $\phi' \in X_{\mathbf{n} \cdot i}$  for all  $0 \leq i \leq D$ . We will prove that every path that starts in  $\mathbf{n}$  satisfies  $\phi_1 U \phi_2$ . So consider an arbitrary infinite path  $\mathbf{n} \cdot j_1 \cdot j_2 \dots \in [0, D]^\omega$ . Since  $\rho$  is accepting, states from  $F_{\phi'}$  must occur infinitely often in  $\pi$ ; that is there exists some  $k \geq 0$  such that  $\phi_2 \in X_{\mathbf{n} \cdot j_1 \dots j_k}$  or  $\phi' \notin X_{\mathbf{n} \cdot j_1 \dots j_k}$ . Let  $k \geq 0$  be minimal. We first show by induction that if  $\phi' \in X_{\mathbf{n} \cdot j_1 \dots j_m}$ , then  $\mathbf{t}, \mathbf{n} \cdot j_1 \dots j_m \models \phi_1$ , and  $\phi' \in X_{\mathbf{n} \cdot j_1 \dots j_{m+1}}$  for all  $0 \leq m < k$ . We have proved this for  $m = 0$  above. So let  $0 \leq m < k$  and suppose  $\phi' \in X_{\mathbf{n} \cdot j_1 \dots j_m}$ . Since  $\rho$  is a run,  $X_{\mathbf{n} \cdot j_1 \dots j_m}$  is propositionally consistent and hence  $\phi_2 \in X_{\mathbf{n} \cdot j_1 \dots j_m}$  or  $\phi_1, AX\phi' \in X_{\mathbf{n} \cdot j_1 \dots j_m}$ . Note that  $\phi_2 \in X_{\mathbf{n} \cdot j_1 \dots j_m}$  contradicts that  $k$  is minimal; hence  $\phi_1, AX\phi' \in X_{\mathbf{n} \cdot j_1 \dots j_m}$ . By the induction hypothesis  $\mathbf{t}, \mathbf{n} \cdot j_1 \dots j_m \models \phi_1$ , and since  $\rho$  is a run, we also have  $\phi' \in X_{\mathbf{n} \cdot j_1 \dots j_{m+1}}$ . We conclude that  $\mathbf{t}, \mathbf{n} \cdot j_1 \dots j_m \models \phi_1$  for all  $0 \leq m < k$ .

Next we prove that  $\mathbf{t}, \mathbf{n} \cdot j_1 \dots j_k \models \phi_2$ . Recall that  $\phi_2 \in X_{\mathbf{n} \cdot j_1 \dots j_k}$  or  $\phi' \notin X_{\mathbf{n} \cdot j_1 \dots j_k}$ . But  $\phi' \in X_{\mathbf{n} \cdot j_1 \dots j_k}$  as we proved before. Hence  $\phi_2 \in X_{\mathbf{n} \cdot j_1 \dots j_k}$ . By the induction hypothesis, we obtain  $\mathbf{t}, \mathbf{n} \cdot j_1 \dots j_k \models \phi_2$ .

We hence have proved that an arbitrary chosen path starting in  $\mathbf{n}$  satisfies  $\phi_1 U \phi_2$ ; hence  $\mathbf{t}, \mathbf{n} \models \phi'$ .

- $\phi' = E\phi_1 R \phi_2$ . Since  $X_{\mathbf{n}}$  is consistent, we have  $\phi_2 \in X_{\mathbf{n}}$ , and  $\phi_1 \in X_{\mathbf{n}}$  or  $EX\phi' \in X_{\mathbf{n}}$ . In the first case, by the induction hypothesis we have  $\mathbf{t}, \mathbf{n} \models \phi_1$  and  $\mathbf{t}, \mathbf{n} \models \phi_2$ , and hence  $\mathbf{t}, \mathbf{n} \models \phi'$ . In the second case, we have  $\mathbf{t}, \mathbf{n} \models \phi_2$  and  $\phi' \in X_{\mathbf{n} \cdot j}$ , where  $j = \iota(EX\phi')$ . Consider the infinite path  $\mathbf{n} \cdot j \cdot j \cdot j \dots$  starting from  $\mathbf{n}$ . We prove that this path satisfies  $\phi_1 R \phi_2$ . We distinguish two cases.

- Suppose there exists some  $k \geq 0$  such that  $\phi_1 \in X_{\mathbf{n} \cdot j^k}$ . Let  $k$  be minimal. By the induction hypothesis,  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_1$ . We prove that if  $\phi' \in X_{\mathbf{n} \cdot j^i}$ , then  $\mathbf{t}, \mathbf{n} \cdot j^i \models \phi_2$  and  $\phi' \in X_{\mathbf{n} \cdot j^{i+1}}$  for all  $0 \leq i < k$ . For  $i = 0$ , we have proved this above. So let  $0 < i < k$  and suppose  $\phi' \in X_{\mathbf{n} \cdot j^i}$ . Since  $\rho$  is a run of  $\mathbb{A}_\phi$ ,  $X_{\mathbf{n} \cdot j^i}$  is propositionally consistent, and hence  $\phi_2 \in X_{\mathbf{n} \cdot j^i}$ , and  $\phi_1 \in X_{\mathbf{n} \cdot j^i}$  or  $EX\phi' \in X_{\mathbf{n} \cdot j^i}$ . Since  $\phi_1 \in X_{\mathbf{n} \cdot j^i}$  contradicts the minimality of  $k$ , we have  $EX\phi' \in X_{\mathbf{n} \cdot j^i}$ . By the induction hypothesis, we have  $\mathbf{t}, \mathbf{n} \cdot j^i \models \phi_2$ , and since  $\rho$  is a run, we have  $\phi' \in X_{\mathbf{n} \cdot j^{i+1}}$ . We can conclude that  $\mathbf{t}, \mathbf{n} \cdot j^i \models \phi_2$  for all  $0 \leq i \leq k$ .
- Suppose that  $\phi_1 \notin X_{\mathbf{n} \cdot j^k}$  for all  $k \geq 0$ . We prove that if  $\phi' \in X_{\mathbf{n} \cdot j^k}$ , then  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_2$  and  $\phi' \in X_{\mathbf{n} \cdot j^{k+1}}$  for all  $k \geq 0$ . For  $k = 0$  we have proved this above. So let  $k > 0$  and suppose  $\phi' \in X_{\mathbf{n} \cdot j^k}$ . Since  $\rho$  is a run, we know that  $X_{\mathbf{n} \cdot j^k}$  is propositionally consistent. Hence  $\phi_2 \in X_{\mathbf{n} \cdot j^k}$  (which, by the induction hypothesis, implies  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_2$ ) and  $\phi_1 \in X_{\mathbf{n} \cdot j^k}$  or  $EX\phi' \in X_{\mathbf{n} \cdot j^k}$ . Note that  $\phi_1 \in X_{\mathbf{n} \cdot j^k}$  cannot be by assumption, hence we have  $EX\phi' \in X_{\mathbf{n} \cdot j^k}$ . We conclude that  $\mathbf{t}, \mathbf{n} \cdot j^k \models \phi_2$  for all  $k \geq 0$ .

We hence have proved that the path above satisfies  $\phi_1 R \phi_2$ , and hence  $\mathbf{t}, \mathbf{n} \models \phi'$ .

- $\phi' = A\phi_1 R \phi_2$ . The proof is a combination of the proofs for  $A\phi_1 U \phi_2$  and  $E\phi_1 R \phi_2$ .

“only if:” Suppose  $\phi$  is satisfiable. By Proposition 3,  $\phi$  has a tree model  $\mathbf{t}$  with domain  $[0, D]^*$  that obeys the direction map  $\iota$ . We prove that  $\mathbf{t} \in L(\mathbb{A}_\phi)$ , that is, there exists some accepting run  $\rho : [0, D]^* \rightarrow [0, D] \times \mathcal{P}(\text{sub}(\phi))$  of  $\mathbb{A}_\phi$  on  $\mathbf{t}$ . Note that  $\mathbf{t}$  belongs to  $L(\mathbb{A}_\phi)$ , assuming that each node is labelled with the letter  $\dagger$  from the single-letter alphabet  $\Sigma$ . Below, we omit to mention the letter from this singleton alphabet and keep  $\mathbf{t}$  of the form  $\mathbf{t} : [0, D]^* \rightarrow \mathbb{Z}^\beta$  to stick to the Kripke structure.

For every node  $\mathbf{n}$  in  $\mathbf{t}$ , define  $X_{\mathbf{n}}$  to be the set of formulas  $\phi'$  in  $\text{sub}(\phi)$  such that  $\mathbf{t}, \mathbf{n} \models \phi'$ . Define  $\rho$  inductively as follows. First set  $\rho(\varepsilon) \stackrel{\text{def}}{=} (0, X_\varepsilon)$ . Assume that  $\rho(\mathbf{n})$  is defined for some node  $\mathbf{n}$ . Set  $\rho(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} (j, X_{\mathbf{n} \cdot j})$  for all  $0 \leq j \leq D$ . We prove that  $\rho$  is an accepting run of  $\mathbb{A}_\phi$  on  $\mathbf{t}$ .

- Let us first prove that  $X_{\mathbf{n}}$  is propositionally consistent for all nodes  $\mathbf{n}$  in  $\mathbf{t}$ .
  - Suppose  $\phi_1 \vee \phi_2 \in X_{\mathbf{n}}$ . That is  $\mathbf{t}, \mathbf{n} \models \phi_1$  or  $\mathbf{t}, \mathbf{n} \models \phi_2$ . But then also  $\phi_1 \in X_{\mathbf{n}}$  or  $\phi_2 \in X_{\mathbf{n}}$ , and hence  $\{\phi_1, \phi_2\} \cap X_{\mathbf{n}} \neq \emptyset$ .

- Suppose  $\phi_1 \wedge \phi_2 \in X_n$ . That is  $\mathbf{t}, \mathbf{n} \models \phi_1$  and  $\mathbf{t}, \mathbf{n} \models \phi_2$ . But then also  $\phi_1 \in X_n$  and  $\phi_2 \in X_n$ , and hence  $\{\phi_1, \phi_2\} \subseteq X_n$ .
  - Suppose  $E\phi_1 U\phi_2 \in X_n$ . We distinguish two cases. (i) Suppose  $\mathbf{t}, \mathbf{n} \models \phi_2$ . Then  $\phi_2 \in X_n$ . (ii) Suppose  $\mathbf{t}, \mathbf{n} \not\models \phi_2$ . By  $\mathbf{t}, \mathbf{n} \models E\phi_1 U\phi_2$ , we conclude that  $\mathbf{t}, \mathbf{n} \models \phi_1$  and  $\mathbf{t}, \mathbf{n} \models EXE\phi_1 U\phi_2$ . But then  $\{\phi_1, EXE\phi_1 U\phi_2\} \subseteq X_n$ .
  - Suppose  $A\phi_1 U\phi_2 \in X_n$ . We distinguish two cases. (i) Suppose  $\mathbf{t}, \mathbf{n} \models \phi_2$ . Then  $\phi_2 \in X_n$ . (ii) Suppose  $\mathbf{t}, \mathbf{n} \not\models \phi_2$ . By  $\mathbf{t}, \mathbf{n} \models A\phi_1 U\phi_2$ , we conclude that  $\mathbf{t}, \mathbf{n} \models \phi_1$  and  $\mathbf{t}, \mathbf{n} \models AXA\phi_1 U\phi_2$ . But then  $\{\phi_1, AXA\phi_1 U\phi_2\} \subseteq X_n$ .
  - Suppose  $E\phi_1 R\phi_2 \in X_n$ . That is,  $\mathbf{t}, \mathbf{n} \models E\phi_1 R\phi_2 \in X_n$  and hence  $\mathbf{t}, \mathbf{n} \models \phi_2$ , so that  $\phi_2 \in X_n$ , too. We distinguish two cases: (i) If  $\mathbf{t}, \mathbf{n} \models \phi_1$ , then  $\phi_1 \in X_n$ . (ii) If  $\mathbf{t}, \mathbf{n} \not\models \phi_1$ , then  $\mathbf{t}, \mathbf{n} \models EXE\phi_1 R\phi_2$ . But then  $EXE\phi_1 R\phi_2 \in X_n$ . Hence we can conclude that  $\{\phi_1, EXE\phi_1 R\phi_2\} \cap X_n \neq \emptyset$ .
  - The proof for  $A\phi_1 R\phi_2$  is analogous using the validity of  $A\phi_1 R\phi_2 \Leftrightarrow (\phi_2 \wedge (\phi_1 \vee AXA\phi_1 R\phi_2))$ .
- Since  $\mathbf{t} \models \phi$ , we must have  $\phi \in X_\varepsilon$ , so that  $(0, X_\varepsilon)$  is an initial state of  $\mathbb{A}_\phi$ .
- Next we prove that, for all nodes  $\mathbf{n}$  with  $\rho(\mathbf{n}) = (i, X_n)$ , there exists a transition

$$((i, X_n), \dagger, (\Theta_0, (0, X_{\mathbf{n} \cdot 0})), \dots, (\Theta_D, (D, X_{\mathbf{n} \cdot D}))) \in \delta$$

satisfying conditions 1., 2. and 3.

1. Let  $EX\phi_1 \in X_n$  and suppose  $\iota(EX\phi_1) = j$  for some  $1 \leq j \leq D$ . By definition of  $X_n$  we have  $\mathbf{t}, \mathbf{n} \models EX\phi_1$ . Since  $\mathbf{t}$  obeys  $\iota$ , we have  $\mathbf{t}, \mathbf{n} \cdot j \models \phi_1$ . Hence indeed  $\phi_1 \in X_{\mathbf{n} \cdot j}$  (and  $\text{sub}(\phi)$  is closed under subformulae).
  2. Let  $AX\phi_1 \in X_n$ . By definition of  $X_n$ , we have  $\mathbf{t}, \mathbf{n} \models AX\phi_1$ . By definition of  $\text{CTL}(\mathbb{Z})$ , we can conclude that  $\mathbf{t}, \mathbf{n} \cdot j \models \phi_1$  for all  $0 \leq j \leq D$ . Hence  $\phi_1 \in X_{\mathbf{n} \cdot j}$  for all  $0 \leq j \leq D$ .
  3. Let  $0 \leq j \leq D$ . Let  $A\Theta \in X_n$ . Hence  $\mathbf{t}, \mathbf{n} \models A\Theta$ . By definition of  $\text{CTL}(\mathbb{Z})$ ,  $\mathbf{t}, \mathbf{n} \cdot j \models \Theta$  for all  $0 \leq j \leq D$ . Hence  $\mathbb{Z} \models \Theta(z, z_j)$ , where  $z = \mathbf{t}(\mathbf{n})$  and  $z_j = \mathbf{t}(\mathbf{n} \cdot j)$ . We can conclude that  $\mathbb{Z} \models (\bigwedge_{A\Theta \in X_n} \Theta)(z, z_j)$ . If, additionally, there exists  $E\Theta' \in X_n$  such that  $\iota(E\Theta') = j$ , then, since  $\mathbf{t}$  obeys  $\iota$ , we have  $\mathbb{Z} \models \Theta'(z, z_j)$ . In that case we have  $\mathbb{Z} \models (\bigwedge_{A\Theta \in X_n} \Theta \wedge \Theta')(z, z_j)$ .
- Next we prove that  $\rho$  is accepting.

Suppose  $E\phi_1 U\phi_2$  is in  $\text{sub}_{EU}(\phi)$ , and let  $\iota(EXE\phi_1 U\phi_2) = j$ . This is the place where we use the index recording the direction. We show that for all branches  $j_1 j_2 \dots \in [0, D]^\omega$ , a location in  $F_{E\phi_1 U\phi_2}$  occurs infinitely often in  $\rho(j_1)\rho(j_2)\dots$ . If  $j_1 j_2 \dots$  is not of the form  $\mathbf{n} \cdot j^\omega$ ,  $\rho(j_1)\rho(j_2)\dots$  does not belong to  $Q^+ \{(j, X) \mid (j, X) \in Q\}^\omega$  and therefore a location  $(i, Y)$  with  $j \neq i$  in  $F_{E\phi_1 U\phi_2}$  occurs infinitely often in  $\rho(j_1)\rho(j_2)\dots$ . Now suppose that  $j_1 j_2 \dots$  is of the form  $\mathbf{n} \cdot j^\omega$ . *Ad absurdum*, assume that there exists some  $m \geq 0$  such that  $\rho(\mathbf{n} \cdot j^{m+k}) \notin F_{E\phi_1 U\phi_2}$  for all  $k \geq 0$ . By definition of  $F_{E\phi_1 U\phi_2}$ , we obtain  $\rho(\mathbf{n} \cdot j^{m+k}) = (j, X_{\mathbf{n} \cdot j^{m+k}})$ ,  $\phi_2 \notin X_{\mathbf{n} \cdot j^{m+k}}$ , and  $E\phi_1 U\phi_2 \in X_{\mathbf{n} \cdot j^{m+k}}$ , for all  $k \geq 0$ . By definition of  $X_{\mathbf{n} \cdot j^{m+k}}$ , we have  $\mathbf{t}, \mathbf{n} \cdot j^{m+k} \models E\phi_1 U\phi_2$ . Since  $\mathbf{t}$  obeys  $\iota$ , there must exist some  $p \geq 0$  such that  $\mathbf{t}, \mathbf{n} \cdot j^{m+p} \models \phi_2$ . But then  $\phi_2 \in X_{\mathbf{n} \cdot j^{m+p}}$ , contradiction.

Suppose  $A\phi_1 U\phi_2$  is in  $\text{sub}_{AU}(\phi)$ . We show that for all nodes  $\mathbf{n}$  in  $\rho$ , every path starting in  $\mathbf{n}$  visits  $F_{A\phi_1 U\phi_2}$  infinitely often. Towards contradiction, suppose that there exists a node  $\mathbf{n}$  and some infinite path  $\mathbf{n} \cdot j_1 \cdot j_2 \dots \in [0, D]^\omega$  that visits  $F_{A\phi_1 U\phi_2}$  only finitely often. That is, there exists some  $m \geq 0$  such that  $\rho(\mathbf{n}_k) \notin F_{A\phi_1 U\phi_2}$  for all  $k \geq m$ . By definition of  $F_{A\phi_1 U\phi_2}$ , and assuming  $\rho(\mathbf{n} \cdot j_1 \dots j_k) = (d_{\mathbf{n} \cdot j_1 \dots j_k}, X_{\mathbf{n} \cdot j_1 \dots j_k})$ , we have  $A\phi_1 U\phi_2 \in X_{\mathbf{n} \cdot j_1 \dots j_k}$  and  $\phi_2 \notin X_{\mathbf{n} \cdot j_1 \dots j_k}$  for all  $k \geq m$ . By definition of  $X_{\mathbf{n} \cdot j_1 \dots j_k}$ , we know that  $\mathbf{t}_{\mathbf{n} \cdot j_1 \dots j_m} \models A\phi_1 U\phi_2$ . But then there must exist some  $p \geq m$  such that  $\mathbf{t}_{\mathbf{n} \cdot j_1 \dots j_p} \models \phi_2$ . By definition,  $\phi_2 \in X_{\mathbf{n} \cdot j_1 \dots j_p}$ , contradiction.  $\square$

Let  $\mathbb{B}_\phi$  be the tree constraint automaton such that  $L(\mathbb{B}_\phi) = L(\mathbb{A}_\phi)$  updating generalised Büchi conditions to Büchi conditions (see Section 3.1). The size of  $\mathbb{B}_\phi$  is exponential in the size of  $\phi$  and in

Section 4, we shall show that the nonemptiness problem for TCA is EXPTIME-complete. In order to get EXPTIME-completeness of  $\text{SAT}(\text{CTL}(\mathbb{Z}))$  and therefore avoiding the *double* exponential blow-up, we need to refine the analysis of the size of  $\mathbb{B}_\phi$  by looking at the size of its different components as well as showing that nonemptiness is in EXPTIME but a few parameters are responsible for the exponential blow-up.

We can conclude the following result with quantitative properties about  $\mathbb{B}_\phi$ .

**Theorem 1.** *Let  $\phi$  be an  $\text{CTL}(\mathbb{Z})$  formula. There is a TCA  $\mathbb{B}_\phi$  such that  $\phi$  is satisfiable iff  $L(\mathbb{B}_\phi) \neq \emptyset$ , and satisfying the properties below.*

- $D$  is bounded above by  $\text{size}(\phi)$  and the alphabet  $\Sigma$  is a singleton,
- the number of locations is bounded by  $(D \times 2^{\text{size}(\phi)}) \times (\text{size}(\phi) + 1)$ ,
- the number of transitions is in  $\mathcal{O}(2^{p(\text{size}(\phi))})$  for some polynomial  $p(\cdot)$ ,
- the number of variables  $\beta$  is bounded by  $\text{size}(\phi)$  and the maximal size of a constraint in transitions is quadratic in  $\text{size}(\phi)$ .

## 4 Complexity of the Nonemptiness Problem

### 4.1 ExpTime-hardness

EXPTIME-hardness of the nonemptiness problem for TCA can be shown by reduction from the acceptance problem for alternating Turing machines running in polynomial space, see e.g. [CKS81]. Indeed, the polynomial-space tape using a finite alphabet  $\Sigma$  can be encoded by a polynomial amount of variables taking values in  $[1, \text{card}(\Sigma)]$ . The transition relation of the alternating Turing machine is encoded by the transition relation of the tree constraint automaton, and there are as many transitions as possible positions of the head. The reduction follows a standard pattern and we provide below a detailed description of the reduction to be self-contained. An *alternating Turing machine* (ATM) is a tuple  $\mathcal{M} = (Q, \Sigma, \delta, q_0, g)$  defined as follows.

- $Q$  is the finite set of control states.
- $\Sigma$  is the finite tape alphabet including the blank symbol  $\#$  and the left endmarker  $\triangleright$ .
- $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q \times \Sigma \times \{\leftarrow, \rightarrow\})$  is the transition function and each  $\delta(q, \mathbf{a})$  contains exactly two elements.
- $q_0 \in Q$  is the initial state.
- $g : Q \rightarrow \{\forall, \exists, \top, \perp\}$  specifies the type of each state. ( $\forall$  for universal states,  $\exists$  for existential states,  $\top$  for accepting states and  $\perp$  for rejecting states).

Without any loss of generality, we can assume that the machine accepts any configuration whenever the control state  $q$  is such that  $g(q) = \top$ , it never tries to go to the left of the first position marked by  $\triangleright$  and  $\triangleright$  is used only for this purpose, the initial state is universal and is never reached after the first step. These conditions can be easily encoded by requiring extra-conditions on  $\mathcal{M}$  and on some definitions below.

A *configuration* is a finite word in  $\Sigma^*(Q \times \Sigma)\Sigma^*$  and we write  $\text{Confs}(\mathcal{M})$  to denote the set of configurations of the ATM  $\mathcal{M}$ . As usually, a configuration  $C$  of the form  $\mathbf{w}(q, \mathbf{a})\mathbf{w}'$  encodes a word  $\mathbf{waw}'$  on the tape, with control state  $q$  and the head is on the  $|\mathbf{wa}|$ th tape cell where  $|\mathbf{wa}|$  denotes the length of the word  $\mathbf{wa}$ . We also say that  $C$  uses  $|\mathbf{waw}'|$  tape cells. The configuration  $C$  is *accepting* if  $g(q) = \top$ , and *rejecting* if  $g(q) = \perp$ . We write  $\vdash_{\mathcal{M}}$  to denote the derivation relation of the ATM  $\mathcal{M}$  defined as follows where  $C = \mathbf{w}(q, \mathbf{a})\mathbf{w}'$ .

- $C \vdash_{\mathcal{M}} C'$  with  $\mathbf{w} = \mathbf{w}''\mathbf{b}$ ,  $C' = \mathbf{w}''(q', \mathbf{b})\mathbf{c}\mathbf{w}'$  and  $(q', \mathbf{c}, \leftarrow) \in \delta(q, \mathbf{a})$ .
- $C \vdash_{\mathcal{M}} C'$  with  $\mathbf{w}' = \mathbf{b}\mathbf{w}''$ ,  $C' = \mathbf{w}\mathbf{c}(q', \mathbf{b})\mathbf{w}''$  and  $(q', \mathbf{c}, \rightarrow) \in \delta(q, \mathbf{a})$ .
- $C \vdash_{\mathcal{M}} C'$  with  $\mathbf{w}' = \varepsilon$ ,  $C' = \mathbf{w}\mathbf{c}(q', \#)$  and  $(q', \mathbf{c}, \rightarrow) \in \delta(q, \mathbf{a})$ .

Given a word  $\mathbf{w}$  in  $(\Sigma \setminus \{\triangleright, \#\})^*$ , its initial configuration  $C_{\text{in}}(\mathbf{w})$  is  $(q_0, \triangleright)\mathbf{w}$ . An *accepting run* for  $\mathbf{w}$  is a finite tree  $\mathbf{t} : \text{dom}(\mathbf{t}) \rightarrow \text{Configs}(\mathcal{M})$  with at most two children per node (but in  $\text{dom}(\mathbf{t})$ , some nodes may have a unique child) satisfying the following conditions:

- $\mathbf{t}(\varepsilon) = C_{\text{in}}(\mathbf{w})$  and all the leaves are labelled by accepting configurations,
- $\mathbf{t}$  does not contain a node labelled by a rejecting configuration,
- for all  $\mathbf{n} \in \text{dom}(\mathbf{t})$  with  $\mathbf{t}(\mathbf{n}) = \mathbf{w}(q, \mathbf{a})\mathbf{w}'$  and  $q$  is universal, the node  $\mathbf{n}$  has two children  $\mathbf{n} \cdot 0$  and  $\mathbf{n} \cdot 1$  that are labelled respectively by all the configurations  $C'$  such that  $C \vdash_{\mathcal{M}} C'$  (maybe the same for the two options).
- for all  $\mathbf{n} \in \text{dom}(\mathbf{t})$  with  $\mathbf{t}(\mathbf{n}) = \mathbf{w}(q, \mathbf{a})\mathbf{w}'$  and  $q$  is existential, the node  $\mathbf{n}$  has one child  $\mathbf{n} \cdot 0$  that is labelled by some configuration  $C'$  such that  $C \vdash_{\mathcal{M}} C'$ .

In that case, we say that  $\mathbf{w}$  is *accepted by the ATM*  $\mathcal{M}$ . The ATM  $\mathcal{M}$  is polynomially space-bounded if there is a polynomial  $p$  such that for all  $\mathbf{w} \in (\Sigma \setminus \{\triangleright, \#\})^*$ , if  $\mathbf{w}$  has an accepting run, then it has an accepting run such that all the configurations labelling the nodes use at most  $p(|\mathbf{w}|)$  tape cells. The problem of determining whether a polynomially space-bounded ATM  $\mathcal{M}$  accepts the input word  $\mathbf{w}$  is known to be EXPTIME-complete [CKS81, Corollary 3.6].

Let  $\mathcal{M} = (Q, \Sigma, \delta, q_0, g)$  be a polynomially space-bounded (with polynomial  $p(\mathbf{x}) \geq \mathbf{x}$ ) and  $\mathbf{w} \in \Sigma^*$  be an input word. Without any loss of generality, we assume that all the configurations considered below use exactly  $p(|\mathbf{w}|)$  tape cells, possibly by padding the suffix with copies of the letter  $\#$ . Let  $\beta = p(|\mathbf{w}|)$ .

We build a TCA  $\mathbb{A}_{\mathcal{M}, \mathbf{w}} = (Q', \Sigma', 2, \beta, Q'_{\text{in}}, \delta', F)$  such that  $\mathcal{M}$  accepts  $\mathbf{w}$  iff  $L(\mathbb{A}_{\mathcal{M}, \mathbf{w}}) \neq \emptyset$ . Actually, we design  $\mathbb{A}_{\mathcal{M}, \mathbf{w}}$  so that it accepts a representation of the accepting runs of  $\mathcal{M}$  from  $\mathbf{w}$ . To do so, we define a simple and natural correspondence between configurations from  $\Sigma^*(Q \times \Sigma)\Sigma^*$  using  $\beta$  tape cells and a subset of  $Q \times [1, \beta] \times [1, \text{card}(\Sigma)]^\beta$ . Elements of the set  $Q \times [1, \beta] \times [1, \text{card}(\Sigma)]^\beta$  are called *numerical configurations*. Let  $\mathbf{f} : \Sigma \rightarrow [1, \text{card}(\Sigma)]$  be an arbitrary one-to-one map. We say that  $\mathbf{w}(q, \mathbf{a})\mathbf{w}' \approx (q', i, \mathbf{d}_1, \dots, \mathbf{d}_\beta)$  iff  $q = q'$ ,  $i = |\mathbf{w}\mathbf{a}|$  and for all  $j \in [1, \beta]$ , we have  $\mathbf{d}_j = \mathbf{f}((\mathbf{w}\mathbf{a}\mathbf{w}')(j))$ . The tape of length  $p(|\mathbf{w}|)$  is encoded by the values of the variables  $\mathbf{x}_1, \dots, \mathbf{x}_{p(|\mathbf{w}|)}$ , i.e. the letter  $\mathbf{a}$  on the  $j$ th tape cell is encoded by the constraint  $\mathbf{x}_j = \mathbf{f}(\mathbf{a})$ .

Let us complete the definition of the TCA  $\mathbb{A}_{\mathcal{M}, \mathbf{w}} = (Q', \Sigma', 2, \beta, Q'_{\text{in}}, \delta', F)$  (we know the degree  $D$  is equal to two, and  $\beta$  variables are involved).

- $\Sigma' = \{\dagger\}$  ( $\Sigma'$  plays no essential role here).
- $Q' \stackrel{\text{def}}{=} Q \times [1, \beta] \uplus \{q^*\}$ . Each pair  $(q, i)$  encodes part of a configuration with control state  $q$  and the head is on the  $i$ th tape cell. The location  $q^*$  is a special accepting state.
- $Q'_{\text{in}} = \{(q_0, 1)\}$  and  $F \stackrel{\text{def}}{=} \{(q, i) \mid g(q) = \top\} \cup \{q^*\}$ .
- It remains to define the transition relation  $\delta'$ .
  - For all  $q \in Q$  such that either  $g(q) = \top$  or  $g(q) = \perp$  for all  $i \in [1, \beta]$ , the only transitions starting from  $(q, i)$  are  $((q, i), \dagger, ((\top, (q, i)), (\top, (q, i))))$ .
  - For all  $i \in [1, \beta]$ , the only transitions starting from  $q^*$  are of the form  $(q^*, \dagger, ((\top, q^*), (\top, q^*)))$ .
  - Given  $i \in [1, \beta]$ ,  $\mathbf{a}, \mathbf{a}' \in \Sigma$ , we write  $\Theta(i, \mathbf{a}, \mathbf{a}')$  to denote the constraint in  $\text{TreeCons}(\beta)$  below:

$$\left( \bigwedge_{j \in [1, \beta] \setminus \{i\}} \mathbf{X}\mathbf{x}_j = \mathbf{x}_j \right) \wedge \mathbf{x}_i = \mathbf{f}(\mathbf{a}) \wedge \mathbf{X}\mathbf{x}_i = \mathbf{f}(\mathbf{a}').$$

For all  $q \in Q$  such that  $q$  is existential, for all  $\mathbf{a} \in \Sigma$  with  $\delta(q, \mathbf{a})$  of the form  $\{(q_1, \mathbf{a}_1, d_1), (q_2, \mathbf{a}_2, d_2)\}$ , for all  $u \in \{1, 2\}$  and  $i \in [1, \beta]$ ,  $\delta'$  contains the transition

$$((q, i), \dagger, ((\Theta(i, \mathbf{a}, \mathbf{a}_u), (q_u, m(i, d_u))), (\top, q^*)))$$

with  $m(i, \rightarrow) = i + 1$  and  $m(i, \leftarrow) = i - 1$ .

- For all  $q \in Q$  such that  $q$  is universal and  $q \neq q_0$ ,  $\mathbf{a} \in \Sigma$  with  $\delta(q, \mathbf{a}) = \{(q_1, \mathbf{a}_1, d_1), (q_2, \mathbf{a}_2, d_2)\}$ , for all  $i \in [1, \beta]$ ,  $\delta'$  contains the transition

$$((q, i), \dagger, ((\Theta(i, \mathbf{a}, \mathbf{a}_1), (q_1, m(i, d_1))), (\Theta(i, \mathbf{a}, \mathbf{a}_2), (q_2, m(i, d_2)))))$$

- Assuming that  $\mathbf{w} = \mathbf{b}_1 \cdots \mathbf{b}_n$  and  $\mathbf{v} = \triangleright \mathbf{w} \sharp^{\beta'}$  with  $\beta' = \beta - (n + 1)$ , for all  $\mathbf{a} \in \Sigma$  with  $\delta(q_0, \mathbf{a}) = \{(q_1, \mathbf{a}_1, d_1), (q_2, \mathbf{a}_2, d_2)\}$ ,  $\delta'$  contains the transition

$$((q_0, 1), \dagger, ((\Theta'(1, \mathbf{a}, \mathbf{a}_1), (q_1, m(1, d_1))), (\Theta'(1, \mathbf{a}, \mathbf{a}_2), (q_2, m(1, d_2)))))$$

where  $\Theta'(1, \mathbf{a}, \mathbf{a}_D) \stackrel{\text{def}}{=} \Theta(1, \mathbf{a}, \mathbf{a}_D) \wedge \bigwedge_{1 \leq j \leq \beta} \mathbf{x}_j = \mathbf{f}(\mathbf{v}(j))$ . Observe that necessarily  $d_1 = d_2 \Rightarrow$  because the head cannot go to the left of the first position.  $q_0$  has a special status and the transtions from it encodes the input word  $\mathbf{w}$  (and that is why we never come back to it).

Correctness of the construction is stated below.

**Lemma 2.**  $\mathcal{M}$  accepts  $\mathbf{w}$  iff  $L(\mathbb{A}_{\mathcal{M}, \mathbf{w}}) \neq \emptyset$ .

*Proof.* Let us state a few simple properties that are used in the sequel (whose easy proofs are omitted herein).

- (I) For every configuration  $C$  using  $\beta$  tape cells, there is a unique numerical configuration  $(q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$  such that  $C \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ .
- (II) For every numerical configuration  $(q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ , there is a unique configuration such that  $C \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ .
- (III) For every configuration  $C$  with universal control state such that  $C \vdash_{\mathcal{M}} C_1$  and  $C \vdash_{\mathcal{M}} C_2$  using the transition function  $\delta$ , there are numerical configurations  $(q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ ,  $(q_1, i_1, \mathfrak{d}_1^1, \dots, \mathfrak{d}_\beta^1)$  and  $(q_2, i_2, \mathfrak{d}_1^2, \dots, \mathfrak{d}_\beta^2)$ , and a transition  $((q, i), \dagger, ((\Theta_1, (q_1, i_1)), (\Theta_2, (q_2, i_2)))) \in \delta'$  such that

$$C \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta), \quad C_1 \approx (q_1, i_1, \mathfrak{d}_1^1, \dots, \mathfrak{d}_\beta^1), \quad C_2 \approx (q_2, i_2, \mathfrak{d}_1^2, \dots, \mathfrak{d}_\beta^2)$$

$$\mathbb{Z} \models \Theta_1(\overline{\mathfrak{d}}, \overline{\mathfrak{d}^1}), \quad \mathbb{Z} \models \Theta_2(\overline{\mathfrak{d}}, \overline{\mathfrak{d}^2}).$$

- (IV) For every configuration  $C$  with existential control state such that  $C \vdash_{\mathcal{M}} C_1$ , there are numerical configurations  $(q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$  and  $(q_1, i_1, \mathfrak{d}_1^1, \dots, \mathfrak{d}_\beta^1)$ , and a transition  $((q, i), \dagger, ((\Theta_1, (q_1, i_1)), (\top, q^*)))$  such that

$$C \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta), \quad C_1 \approx (q_1, i_1, \mathfrak{d}_1^1, \dots, \mathfrak{d}_\beta^1), \quad \mathbb{Z} \models \Theta_1(\overline{\mathfrak{d}}, \overline{\mathfrak{d}^1}).$$

Conditions (III) and (IV) are proved using condition (I) and the way  $\delta'$  is defined. Moreover, this means that the relationships between nodes in an accepting run can be simulated by runs on trees for TCA.

(V) For all numerical configurations  $(q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ ,  $(q_1, i_1, \mathfrak{d}_1^1, \dots, \mathfrak{d}_\beta^1)$  and  $(q_2, i_2, \mathfrak{d}_1^2, \dots, \mathfrak{d}_\beta^2)$ , and transitions  $((q, i), \dagger, ((\Theta_1, (q_1, i_1)), (\Theta_2, (q_2, i_2))))$  such that  $\mathbb{Z} \models \Theta_1(\overline{\mathfrak{d}}, \overline{\mathfrak{d}^1})$  and  $\mathbb{Z} \models \Theta_2(\overline{\mathfrak{d}}, \overline{\mathfrak{d}^2})$ , there are configurations  $C$ ,  $C_1$  and  $C_2$  such that

$$C \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta), \quad C_1 \approx (q_1, i_1, \mathfrak{d}_1^1, \dots, \mathfrak{d}_\beta^1), \quad C_2 \approx (q_2, i_2, \mathfrak{d}_1^2, \dots, \mathfrak{d}_\beta^2),$$

$$C \vdash_{\mathcal{M}} C_1, \quad C \vdash_{\mathcal{M}} C_2.$$

(VI) For all numerical configurations  $(q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$  and  $(q_1, i_1, \mathfrak{d}_1^1, \dots, \mathfrak{d}_\beta^1)$  and transitions

$$((q, i), \dagger, ((\Theta_1, (q_1, i_1)), (\top, q^*)))$$

such that  $\mathbb{Z} \models \Theta_1(\overline{\mathfrak{d}}, \overline{\mathfrak{d}^1})$ , there are configurations  $C$  and  $C_1$  such that

$$C \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta), \quad C_1 \approx (q_1, i_1, \mathfrak{d}_1^1, \dots, \mathfrak{d}_\beta^1), \quad C \vdash_{\mathcal{M}} C_1.$$

Conditions (V) and (VI) are proved using condition (II) and the way  $\delta'$  is defined. Moreover, this means that the relationships between nodes in runs on trees for TCA can be simulated by accepting runs.

( $\Rightarrow$ ) Let  $\mathbf{t} : \text{dom}(\mathbf{t}) \rightarrow \text{Configs}(\mathcal{M})$  be an accepting run for  $\mathfrak{w}$ . Let  $\mathbf{t}^* : [0, 1]^* \rightarrow \mathbb{Z}^\beta$  be an infinite tree such that for all  $\mathbf{n} \in \text{dom}(\mathbf{t})$  with  $\mathbf{t}(\mathbf{n}) \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ , we have  $\mathbf{t}^*(\mathbf{n}) \stackrel{\text{def}}{=} (\mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ . Note that  $\mathbf{t}^*(\mathbf{n})$  has a unique value by Condition (I) when  $\mathbf{n} \in \text{dom}(\mathbf{t})$  and we have omitted to represent the unique possible letter on each node. Let  $\rho^* : [0, 1]^* \rightarrow Q'$  be the run defined as follows.

- For all  $\mathbf{n} \in \text{dom}(\mathbf{t})$  with  $\mathbf{t}(\mathbf{n}) \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ , we have  $\rho^*(\mathbf{n}) \stackrel{\text{def}}{=} (q, i)$ .
- For all  $\mathbf{n} \in ([0, 1]^* \setminus \text{dom}(\mathbf{t}))$  such that there is no strict prefix  $\mathbf{n}'$  of  $\mathbf{n}$  that is a leaf of  $\text{dom}(\mathbf{t})$ , we have  $\rho^*(\mathbf{n}) \stackrel{\text{def}}{=} q^*$ .
- For all  $\mathbf{n} \in ([0, 1]^* \setminus \text{dom}(\mathbf{t}))$  such that there is a strict prefix  $\mathbf{n}'$  of  $\mathbf{n}$  that is a leaf of  $\text{dom}(\mathbf{t})$  and  $\mathbf{t}(\mathbf{n}') \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ , we have  $\rho^*(\mathbf{n}) \stackrel{\text{def}}{=} (q, i)$ .

One can easily show that  $\rho^*$  is an accepting run on  $\mathbf{t}^*$  by using the conditions (III) and (IV), as well as the definition of the set  $F$  of accepting states in  $\mathbb{A}_{\mathcal{M}, \mathfrak{w}}$ .

( $\Leftarrow$ ) Let  $\mathbf{t} : [0, 1]^* \rightarrow \Sigma' \times \mathbb{Z}^\beta$  be an infinite tree and  $\rho : [0, 1]^* \rightarrow Q'$  be an accepting run on  $\mathbf{t}$ . By construction of  $\mathbb{A}_{\mathcal{M}, \mathfrak{w}}$ , along any infinite branch, once a location in  $F$  is visited, it is visited forever along that branch, and moreover any infinite branch always visits such a location in  $F$ . Let  $X$  be the finite subset of  $[0, 1]^*$  such that  $\mathbf{n} \in X$  iff either  $\rho(\mathbf{n}) \notin F$ , or  $\rho(\mathbf{n}) \in (F \setminus \{q^*\})$  and all its ancestors do not belong to  $F$ . One can check that  $X$  is a finite tree (use of König's Lemma here). Let  $\mathbf{t}^* : X \rightarrow \text{Configs}(\mathcal{M})$  be the map such that for all  $\mathbf{n} \in X$ , we have  $\mathbf{t}^*(\mathbf{n}) = C$  for the unique configuration such that  $C \approx (q, i, \mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$  with  $\rho(\mathbf{n}) = (q, i)$  and  $\mathbf{t}(\mathbf{n}) = (\mathfrak{d}_1, \dots, \mathfrak{d}_\beta)$ . One can easily show that  $\mathbf{t}^*$  is an accepting run for  $\mathfrak{w}$  by using the conditions (V) and (VI), as well as the definition of the set accepting states in  $\mathcal{M}$ .  $\square$

This concludes the EXPTIME-hardness proof for the nonemptiness problem for tree constraint automata. Observe that it is also easy to get EXPTIME-hardness without using any constant.

## 4.2 Symbolic trees

In Sections 4.2-4.4, we establish that NE(TCA) can be solved in exponential-time with a parameterised analysis that leads to an optimal complexity for SAT(CTL( $\mathbb{Z}$ )). The proof is actually divided in two parts. In order to determine whether  $L(\mathbb{A})$  is non-empty, we transform the existence of some tree  $\mathbf{t} \in L(\mathbb{A})$  into the existence of some regular symbolic tree (with constraints between nodes) that admits a concrete model. In the second stage, we characterise the complexity of determining the existence of such satisfiable regular symbolic trees. Indeed,  $L(\mathbb{A})$  is non-empty iff there are a tree  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$  and an accepting run  $\rho : [0, D-1]^* \rightarrow Q$  on  $\mathbf{t}$ , and therefore  $(\mathbf{t}, \rho)$  satisfies constraints based on the transition relation  $\delta$  (see Definition 1). We define a notion of regular symbolic tree that encodes the tree structure of  $(\mathbf{t}, \rho)$  with  $\mathbb{Z}$ -constraints satisfied by the different data values (but without tuples of data values in  $\mathbb{Z}^\beta$ ), and for which we require to check satisfiability.

From now on, we assume a fixed TCA  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$  with the constants  $\mathfrak{d}_1, \dots, \mathfrak{d}_\alpha$  occurring in  $\mathbb{A}$  such that  $\mathfrak{d}_1 < \dots < \mathfrak{d}_\alpha$  (we assume there is at least one constant). A *type* over the variables  $\mathbf{z}_1, \dots, \mathbf{z}_n$  is an expression of the form

$$(\bigwedge_i \Theta_i^{\text{CST}}) \wedge (\bigwedge_{i < j} \mathbf{z}_i \sim_{i,j} \mathbf{z}_j), \text{ where}$$

- for all  $i \in [1, n]$ , we have  $\Theta_i^{\text{CST}}$  is equal to either  $\mathbf{z}_i < \mathfrak{d}_1$ , or  $\mathbf{z}_i > \mathfrak{d}_\alpha$  or  $\mathbf{z}_i = \mathfrak{d}$  for some  $\mathfrak{d} \in [\mathfrak{d}_1, \mathfrak{d}_\alpha]$ . It goes a bit beyond the constraint language in  $\mathbb{Z}$  (because of expressions of the form  $\mathbf{z}_i < \mathfrak{d}_1$  and  $\mathbf{z}_i > \mathfrak{d}_\alpha$ ) but this is harmless in the sequel.
- $\sim_{i,j} \in \{>, =, <\}$  for all  $i < j$ .

What really matters in a type is the way the variables are compared to each other and to the constants (we may use alternative symbolic representations). Below,  $\mathbf{x} = \mathbf{y}$  and  $\mathbf{y} = \mathbf{x}$  are understood as identical, as well as  $\mathbf{x} < \mathbf{y}$  and  $\mathbf{y} > \mathbf{x}$ . Checking the satisfiability of a type can be done in polynomial-time (based on a standard cycle detection, see e.g. [Cer94, Lemma 5.5]). In the sequel, we are mainly interested in satisfiable types. The set of *satisfiable* types built over the variables  $\mathbf{x}_1, \dots, \mathbf{x}_\beta, \mathbf{x}'_1, \dots, \mathbf{x}'_\beta$  is written  $\text{SatTypes}(\beta)$ . Observe that  $\text{card}(\text{SatTypes}(\beta)) \leq ((\mathfrak{d}_\alpha - \mathfrak{d}_1) + 3)^{2\beta} \times 3^{\beta^2}$ . Strictly speaking, the set  $\text{SatTypes}(\beta)$  depends on the constants  $\mathfrak{d}_1, \dots, \mathfrak{d}_\alpha$  but we omit the reference to it in the notation  $\text{SatTypes}(\beta)$  to ease the reading.

The main properties we use about satisfiable types are stated below.

### Lemma 3.

- (I) Let  $\mathbf{v}, \mathbf{v}' \in \mathbb{Z}^\beta$ . There is a unique  $\Theta \in \text{SatTypes}(\beta)$  such that  $\mathbb{Z} \models \Theta(\mathbf{v}, \mathbf{v}')$ .
- (II) For every constraint  $\Theta$  built over the variables  $\mathbf{x}_1, \dots, \mathbf{x}_\beta, \mathbf{x}'_1, \dots, \mathbf{x}'_\beta$  and the constants  $\mathfrak{d}_1, \dots, \mathfrak{d}_\alpha$  there is a disjunction  $\Theta_1 \vee \dots \vee \Theta_\gamma$  logically equivalent to  $\Theta$  and each  $\Theta_i$  belongs to  $\text{SatTypes}(\beta)$  (empty disjunction stands for  $\perp$ ).
- (III) For all  $\Theta \neq \Theta' \in \text{SatTypes}(\beta)$ , the constraint  $\Theta \wedge \Theta'$  is not satisfiable.

The proof of Lemma 3 is by an easy verification and its statement justifies the term ‘type’ used in this context.

A *symbolic tree*  $\mathbf{t}$  is a map  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \text{SatTypes}(\beta)$  and such trees are intended to be abstractions of maps  $[0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$ . If  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$  for some  $\mathbf{n} \in [0, D-1]^*$ , the primed variables in  $\Theta$  refer to the  $\beta$  values at the node  $\mathbf{n}$  whereas the unprimed ones refer to the  $\beta$  values at the parent node, if any. We hope that this does not lead to confusions.

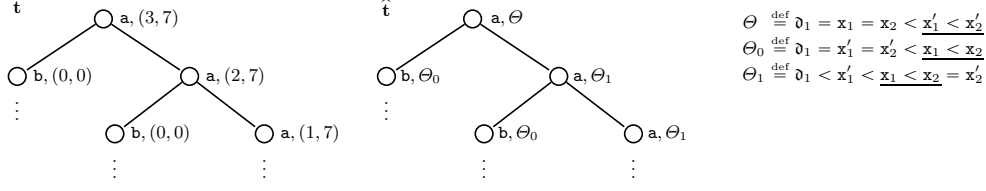


Fig. 3: A tree  $\mathbf{t}$  and its corresponding symbolic tree  $\hat{\mathbf{t}}$ .

Given a tree  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$ , its *abstraction*  $\hat{\mathbf{t}} : [0, D-1]^* \rightarrow \Sigma \times \text{SatTypes}(\beta)$  is such that for all  $\mathbf{n} \cdot i \in [0, D-1]^*$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \mathbf{z})$  and  $\mathbf{t}(\mathbf{n} \cdot i) = (\mathbf{a}_i, \mathbf{z}_i)$ ,  $\hat{\mathbf{t}}(\mathbf{n} \cdot i) \stackrel{\text{def}}{=} (\mathbf{a}_i, \Theta_i)$  for the unique  $\Theta_i \in \text{SatTypes}(\beta)$  (Lemma 3) such that  $\mathbb{Z} \models \Theta_i(\mathbf{z}, \mathbf{z}_i)$ . At the root  $\varepsilon$  with  $\mathbf{t}(\varepsilon) = (\mathbf{a}, \mathbf{z})$ , we have  $\hat{\mathbf{t}}(\varepsilon) \stackrel{\text{def}}{=} (\mathbf{a}, \Theta)$  for the unique  $\Theta \in \text{SatTypes}(\beta)$  such that  $\mathbb{Z} \models \Theta(\mathbf{0}, \mathbf{z})$  (the zero tuple  $\mathbf{0}$  is an arbitrary value here). A symbolic tree  $\hat{\mathbf{t}}$  is *satisfiable*  $\Leftrightarrow$  there is  $\check{\mathbf{t}} : [0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$  such that  $\hat{\mathbf{t}} = \check{\mathbf{t}}$ . We say that  $\check{\mathbf{t}}$  *witnesses the satisfaction of*  $\hat{\mathbf{t}}$ , also written  $\check{\mathbf{t}} \models \hat{\mathbf{t}}$ .

*Example 1.* In Figure 3, we show a tree  $\mathbf{t}$  with concrete values in  $\mathbb{Z}^\beta$  for  $\beta = 2$  (left) and its corresponding symbolic tree  $\hat{\mathbf{t}}$  (middle). We assume that  $\mathfrak{d}_1 = 0$  is the only constant; consequently,  $\hat{\mathbf{t}}$  uses constraints in  $\text{SatTypes}(\beta)$  (right) that are built with variables  $\mathbf{x}_1, \mathbf{x}_2$ , their primed variants  $\mathbf{x}'_1, \mathbf{x}'_2$ , and constant  $\mathfrak{d}_1$ . We underline constraints to illustrate the property of local consistency (see definition below).

In our quest to decide the nonemptiness of  $L(\mathbb{A})$ , in abstractions of trees from  $L(\mathbb{A})$ , local properties are required between a parent node and its child nodes. Moreover, such abstractions should be compatible with the transition relation in  $\mathbb{A}$ , which explains the two notions introduced below about symbolic trees.

A symbolic tree  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \text{SatTypes}(\beta)$  is *locally consistent*  $\Leftrightarrow$  for all  $\mathbf{n} \in [0, D-1]^*$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$ ,  $\mathbf{t}(\mathbf{n} \cdot 0) = (\mathbf{a}_0, \Theta_0), \dots, \mathbf{t}(\mathbf{n} \cdot (D-1)) = (\mathbf{a}_{D-1}, \Theta_{D-1})$ , we have

- (LC1)  $\Theta_0, \dots, \Theta_{D-1}$  agree on  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$  (same parent node),
- (LC2)  $\Theta$  restricted to  $\mathbf{x}'_1, \dots, \mathbf{x}'_\beta$  agrees with  $\Theta_0$  restricted to  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$  (any  $\Theta_j$  with  $j \in [0, D-1]$  would be fine, in view of the first point).

In order to define the second notion about symbolic trees, we need to introduce a richer type of runs, in which instead of having a single location per node, we record the transition starting from this location (by the way, this could have been our original definition for accepting runs too). An *enriched run*  $\rho^+$  of  $\mathbb{A}$  is a map  $\rho^+ : [0, D-1]^* \rightarrow \delta$  such that:

- (ER1)  $\rho^+(\varepsilon)$  is of the form  $(q_0, \dots)$  with  $q_0 \in Q_{\text{in}}$  (initial condition).
- (ER2) For all  $\mathbf{n} \in [0, D-1]^*$  with  $\rho^+(\mathbf{n}) = (q, \mathbf{a}, (\Theta_0, q_0), \dots, (\Theta_{D-1}, q_{D-1}))$  and for all  $i \in [0, D-1]$ ,  $\rho^+(\mathbf{n} \cdot i)$  is of the form  $(q_i, \dots)$ . (The transition  $\rho^+(\mathbf{n})$  matches the locations from the run.)
- (ER3) For all branches in  $\rho^+$ , one location in  $F$  occurs infinitely often as the first location of the transitions (acceptance condition).

An enriched run can be therefore viewed as a (standard) accepting run completed with the transitions from  $\delta$  that witness being a run.

A symbolic tree  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \text{SatTypes}(\beta)$  *respects*  $\mathbb{A}$   $\Leftrightarrow$  there is an enriched run  $\rho^+ : [0, D-1]^* \rightarrow \delta$  such that for all  $\mathbf{n} \in [0, D-1]^*$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$ ,  $\mathbf{t}(\mathbf{n} \cdot 0) = (\mathbf{a}_0, \Theta_0), \dots, \mathbf{t}(\mathbf{n} \cdot (D-1)) = (\mathbf{a}_{D-1}, \Theta_{D-1})$  and  $\rho^+(\mathbf{n}) = (q, \mathbf{a}, (\Theta'_0, q_0), \dots, (\Theta'_{D-1}, q_{D-1}))$ , we have for all  $i \in [0, D-1]$ ,  $\Theta_i \models \Theta'_i$ .



A symbolic tree  $\mathbf{t}$  is  $\mathbb{A}$ -consistent iff it is locally consistent and it respects  $\mathbb{A}$ .

**Lemma 4.** *Let  $\mathbf{t} \in L(\mathbb{A})$ . The symbolic tree  $\hat{\mathbf{t}}$  is  $\mathbb{A}$ -consistent.*

The proof of Lemma 4 is by an easy verification. Moreover, the class of  $\mathbb{A}$ -consistent symbolic trees can be defined with the help of a standard Büchi tree automaton (no constraints) [VW86], as done below with  $\mathbb{B}_{\text{cons}(\mathbb{A})}$ . Let  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  be the Büchi tree automaton  $\mathbb{B}_{\text{cons}(\mathbb{A})} \stackrel{\text{def}}{=} (Q', \Sigma \times \text{SatTypes}(\beta), D, Q'_{\text{in}}, \delta', F')$  defined as follows.

- $Q' \stackrel{\text{def}}{=} \text{SatTypes}(\beta) \times Q$ ;  $Q'_{\text{in}} \stackrel{\text{def}}{=} \{\Theta \in \text{SatTypes}(\beta) \mid \exists \mathbf{z} \in \mathbb{Z}^\beta \text{ s.t. } \mathbb{Z} \models \Theta(\mathbf{0}, \mathbf{z})\} \times Q_{\text{in}}$ ;  $F' \stackrel{\text{def}}{=} \text{SatTypes}(\beta) \times F$ .
- $((\Theta, q), (\mathbf{a}, \Theta), (\Theta_0, q_0), \dots, (\Theta_{D-1}, q_{D-1})) \in \delta' \stackrel{\text{def}}{\iff}$  there is a transition  $(q, \mathbf{a}, (\Theta'_0, q_0), \dots, (\Theta'_{D-1}, q_{D-1})) \in \delta$  such that
  - for all  $i \in [0, D-1]$ ,  $\Theta_i \models \Theta'_i$  (PTIME check because  $\Theta_i \in \text{SatTypes}(\beta)$ ),
  - $\Theta_0, \dots, \Theta_{D-1}$  agree on  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$  (same parent node),
  - $\Theta$  restricted to  $\mathbf{x}'_1, \dots, \mathbf{x}'_\beta$  agrees with  $\Theta_0$  restricted to  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$ .

Observe that the transition relation  $\delta'$  can be decided in polynomial-time in  $\text{card}(\delta) + \beta + D + \text{MaxConsSize}(\mathbb{A})$  (all the values in the sum are less than the size of  $\mathbb{A}$ ). A bit of explanations about the bound may be required here. In order to check whether  $((\Theta, q), (\mathbf{a}, \Theta), (\Theta_0, q_0), \dots, (\Theta_{D-1}, q_{D-1})) \in \delta'$  holds, we might go through all the transitions in  $\delta$  and verify the satisfaction of the three conditions above. The first condition requires polynomial-time in  $\beta + \text{MaxConsSize}(\mathbb{A})$  to check  $\Theta_i \models \Theta'_i$  and this is done  $D$  times. A similar time-complexity is required to check the satisfaction of the two remaining conditions. Moreover, obviously  $\text{card}(Q') = \text{card}(\text{SatTypes}(\beta)) \times \text{card}(Q)$  and  $\text{card}(\delta') = \text{card}(Q')^D \times \text{card}(\Sigma) \times \text{card}(\text{SatTypes}(\beta))$ .

One can check that  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  accepts all the  $\mathbb{A}$ -consistent symbolic trees since its definition reformulates the conditions (LC1), (LC2), (ER1), (ER2) and (ER3).

**Lemma 5.**  $L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  is equal to the set of  $\mathbb{A}$ -consistent symbolic trees.

More importantly, the crucial property to check is whether  $L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  contains a satisfiable symbolic tree.

*Example 2 (ctd.).* Consider again the symbolic tree  $\hat{\mathbf{t}}$  on the right of Figure 3. Note that if all the nodes in the rightmost branch of  $\mathbf{t}$  were labelled with the label  $(\mathbf{a}, \Theta_1)$ , then  $\hat{\mathbf{t}}$  would not be satisfiable: in order to satisfy  $\Theta_1$ 's conjunct  $\mathbf{x}'_2 < \mathbf{x}_2$ , the value of the variable  $\mathbf{x}_2$  must inevitably become finally equal to the value of the variable  $\mathbf{x}_1$ , violating the conjunct  $\mathbf{x}_1 < \mathbf{x}_2$ .

The result below is a variant of many similar results relating symbolic models and concrete models in logics for concrete domains, see e.g. [DD07, Corollary 4.1], [Gas09, Lemma 3.4], [CT16, Theorem 25], [LOS20, Theorem 11].

**Lemma 6.**  $L(\mathbb{A}) \neq \emptyset$  iff there is a satisfiable symbolic tree in  $L(\mathbb{B}_{\text{cons}(\mathbb{A})})$ .

*Proof.* “only if”: Suppose  $L(\mathbb{A}) \neq \emptyset$ . This means that there is  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$  in  $L(\mathbb{A})$ . Since  $L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  contains all the  $\mathbb{A}$ -consistent symbolic trees (Lemma 5) and  $\hat{\mathbf{t}}$  is  $\mathbb{A}$ -consistent by Lemma 4, we get that  $\hat{\mathbf{t}} \in L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  and  $\hat{\mathbf{t}}$  is clearly satisfiable (which is witnessed by  $\mathbf{t}$ ).

“if”: Suppose there is  $\mathbf{t} \in L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  such that  $\mathbf{t}$  is satisfiable. This entails the existence of  $\check{\mathbf{t}} : [0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$  such that  $\hat{\check{\mathbf{t}}} = \mathbf{t}$ . This means that for all  $\mathbf{n} \cdot i \in [0, D-1]^*$  with  $\check{\mathbf{t}}(\mathbf{n}) = (\mathbf{a}, \mathbf{z})$ ,

$\check{\mathbf{t}}(\mathbf{n} \cdot i) = (\mathbf{a}_i, \mathbf{z}_i)$  and  $\mathbf{t}(\mathbf{n} \cdot i) = (\mathbf{a}_i, \Theta_i)$ , we have  $\mathbb{Z} \models \Theta_i(\mathbf{z}, \mathbf{z}_i)$ . Moreover, if  $\check{\mathbf{t}}(\varepsilon) = (\mathbf{a}, \mathbf{z})$ , and  $\mathbf{t}(\varepsilon) = (\mathbf{a}, \Theta)$ , then  $\mathbb{Z} \models \Theta(\mathbf{0}, \mathbf{z})$ .

Since  $\mathbf{t}$  respects  $\mathbb{A}$  by Lemma 5, there is an enriched run  $\rho^+ : [0, D-1]^* \rightarrow \delta$  such that for all  $\mathbf{n} \in [0, D-1]^*$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$ ,  $\mathbf{t}(\mathbf{n} \cdot 0) = (\mathbf{a}_0, \Theta_0)$ ,  $\dots$ ,  $\mathbf{t}(\mathbf{n} \cdot (D-1)) = (\mathbf{a}_{D-1}, \Theta_{D-1})$  and  $\rho^+(\mathbf{n}) = (q, \mathbf{a}, (\Theta'_0, q_0), \dots, (\Theta'_{D-1}, q_{D-1}))$ , we have for all  $i \in [0, D-1]$ ,  $\Theta_i \models \Theta'_i$ . Let  $\rho : [0, D-1]^* \rightarrow Q$  be the map such that for all  $\mathbf{n} \in [0, D-1]^*$  with  $\rho^+(\mathbf{n}) = (q, \mathbf{a}, (\Theta'_0, q_0), \dots, (\Theta'_{D-1}, q_{D-1}))$ , we have  $\rho(\mathbf{n}) \stackrel{\text{def}}{=} q$ . Let us verify that  $\rho$  is a run of  $\mathbb{A}$  on  $\check{\mathbf{t}}$  and therefore  $L(\mathbb{A}) \neq \emptyset$ .

- By (ER1),  $\rho^+(\varepsilon)$  is of the form  $(q_0, \dots)$  with  $q_0$  being an initial location. So,  $\rho(\varepsilon) \in Q_{\text{in}}$ .
- Let  $\mathbf{n} \in [0, D-1]^*$  with  $\check{\mathbf{t}}(\mathbf{n}) = (\mathbf{a}, \mathbf{z})$ ,  $\rho(\mathbf{n}) = q$ ,  $\check{\mathbf{t}}(\mathbf{n} \cdot i) = (\mathbf{a}_i, \mathbf{z}_i)$  and  $\rho(\mathbf{n} \cdot i) = q_i$  for all  $0 \leq i < D$ . By (ER2), if  $\rho^+(\mathbf{n}) = (q, \mathbf{a}, (\Theta'_0, q_0), \dots, (\Theta'_{D-1}, q_{D-1}))$ , we have for all  $i \in [0, D-1]$ ,  $\mathbf{t}(\mathbf{n} \cdot i) \models \Theta'_i$ . Since  $\check{\mathbf{t}}$  witnesses the satisfaction of  $\mathbf{t}$ , we also have  $\mathbb{Z} \models \mathbf{t}(\mathbf{n} \cdot i)(\mathbf{z}, \mathbf{z}_i)$  and therefore  $\mathbb{Z} \models \Theta'_i(\mathbf{n} \cdot i)(\mathbf{z}, \mathbf{z}_i)$ , which allows us to conclude that  $\rho$  is indeed a run.
- To conclude that  $\rho$  is an accepting run, we take advantage of (ER3).  $\square$

### 4.3 Satisfiability for regular locally consistent symbolic trees

Now, we move to developments to determine when there are satisfiable symbolic trees in  $L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  while evaluating the computational complexity to check the existence of such satisfiable symbolic trees. Given a locally consistent symbolic tree  $\mathbf{t}$ , we introduce an auxiliary labelled graph  $G_{\mathbf{t}}^C$  that contains exactly the same constraints as in  $\mathbf{t}$  but expressed in a tree-like graph from which it is convenient to characterize satisfiability in terms of paths when  $\mathbf{t}$  is regular (a tree is regular if its set of subtrees is finite). Lemma 7 justifies why one can reason on  $G_{\mathbf{t}}^C$  afterwards. Note that similar symbolic structures are introduced in [Lut01, DD07, LOS20].

Let  $T(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$  be the set  $\{\mathbf{x}_1, \dots, \mathbf{x}_\beta\} \cup \{\mathfrak{d}_1, \mathfrak{d}_\alpha\}$  ( $\beta+2$  elements if  $\alpha > 1$ ) and we write  $\mathbf{x}, \mathbf{x}_1, \mathbf{x}_2, \dots$  to denote elements in  $T(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$ . Elements in  $T(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$  are either variables or constants (among  $\{\mathfrak{d}_1, \mathfrak{d}_\alpha\}$  only). As usual, the expression  $\mathbf{x}$  (possibly decorated) denotes a variable. Two nodes  $\mathbf{n}, \mathbf{n}' \in [0, D-1]^*$  are *neighbours*  $\stackrel{\text{def}}{\iff}$  either  $\mathbf{n} = \mathbf{n}'$ , or  $\mathbf{n} = \mathbf{n}' \cdot j$  or  $\mathbf{n}' = \mathbf{n} \cdot j$  for some  $j \in [0, D-1]$  (siblings are not neighbours). Two elements  $(\mathbf{n}, \mathbf{x})$  and  $(\mathbf{n}', \mathbf{x}')$  in  $[0, D-1]^* \times T(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$  are *neighbours*  $\stackrel{\text{def}}{\iff}$   $\mathbf{n}$  and  $\mathbf{n}'$  are neighbours. By construction, the edges in  $G_{\mathbf{t}}^C$  are possible only between neighbour elements.

The labelled graph  $G_{\mathbf{t}}^C$  is defined as the structure below.

$$G_{\mathbf{t}}^C = (\mathcal{D}_{\mathbf{t}}, \xrightarrow{=}, \xrightarrow{<}, \{U_i \mid i \in [0, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2]\}),$$

such that  $\mathcal{D}_{\mathbf{t}}^V \stackrel{\text{def}}{=} [0, D-1]^* \times \{\mathbf{x}_1, \dots, \mathbf{x}_\beta\}$ ,  $\mathcal{D}_{\mathbf{t}}^C \stackrel{\text{def}}{=} [0, D-1]^* \times \{\mathfrak{d}_1, \mathfrak{d}_\alpha\}$ ,  $\mathcal{D}_{\mathbf{t}} \stackrel{\text{def}}{=} \mathcal{D}_{\mathbf{t}}^V \uplus \mathcal{D}_{\mathbf{t}}^C$  (equal to  $[0, D-1]^* \times T(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$ ),  $\xrightarrow{=}$  and  $\xrightarrow{<}$  are binary relations on  $\mathcal{D}_{\mathbf{t}}$ ,  $\{U_i \mid i \in [0, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2]\}$  is a partition of  $\mathcal{D}_{\mathbf{t}}$  and the conditions below hold.

**(VAR)** For all  $(\mathbf{n}, \mathbf{x}_i), (\mathbf{n}', \mathbf{x}_{i'}) \in \mathcal{D}_{\mathbf{t}}^V$ , for all  $\sim \in \{<, =\}$ ,  $(\mathbf{n}, \mathbf{x}_i) \xrightarrow{\sim} (\mathbf{n}', \mathbf{x}_{i'}) \stackrel{\text{def}}{\iff}$  either  $\mathbf{n}' = \mathbf{n} \cdot j$  and  $\mathbf{x}_i \sim \mathbf{x}_{i'}$  in  $\Theta$  with  $\mathbf{t}(\mathbf{n}') = (\mathbf{a}, \Theta)$ , or  $\mathbf{n} = \mathbf{n}'$  and  $\mathbf{x}_i \sim \mathbf{x}_{i'}$  in  $\Theta$  with  $\mathbf{t}(\mathbf{n}') = (\mathbf{a}, \Theta)$ , or  $\mathbf{n} = \mathbf{n}' \cdot j$  and  $\mathbf{x}_{i'} \sim \mathbf{x}_i$  in  $\Theta$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$ .

**(PART1)** For all  $i \in [1, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1]$ , for all  $(\mathbf{n}, \mathbf{x}_j) \in \mathcal{D}_{\mathbf{t}}^V$ ,  $(\mathbf{n}, \mathbf{x}_j) \in U_i \stackrel{\text{def}}{\iff} \mathbf{x}'_j = \mathfrak{d}_1 + (i-1)$  in  $\Theta$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$ .

**(PART2)** For all  $(\mathbf{n}, \mathbf{x}_j) \in \mathcal{D}_{\mathbf{t}}^V$ ,  $(\mathbf{n}, \mathbf{x}_j) \in U_0 \stackrel{\text{def}}{\iff} \mathbf{x}'_j < \mathfrak{d}_1$  in  $\Theta$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$ .

**(PART3)** For all  $(\mathbf{n}, \mathbf{x}_j) \in \mathcal{D}_{\mathbf{t}}^V$ ,  $(\mathbf{n}, \mathbf{x}_j) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2} \stackrel{\text{def}}{\iff} \mathbf{x}'_j > \mathfrak{d}_\alpha$  in  $\Theta$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$ .

**(PART4)** For all  $\mathbf{n} \in [0, D-1]^*$ ,  $(\mathbf{n}, \mathfrak{d}_1) \in U_1$  and  $(\mathbf{n}, \mathfrak{d}_\alpha) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1}$ .

**(CONS)** For all  $((\mathbf{n}, \mathbf{x}\mathbf{d}), (\mathbf{n}', \mathbf{x}\mathbf{d}')) \in (\mathcal{D}_{\mathbf{t}} \times \mathcal{D}_{\mathbf{t}}) \setminus (\mathcal{D}_{\mathbf{t}}^V \times \mathcal{D}_{\mathbf{t}}^V)$  such that  $\mathbf{n}$  and  $\mathbf{n}'$  are neighbours, for all  $i, j \in [0, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2]$  such that  $(\mathbf{n}, \mathbf{x}\mathbf{d}) \in U_i$  and  $(\mathbf{n}', \mathbf{x}\mathbf{d}') \in U_j$ , for all  $\sim \in \{<, =\}$ ,  $(\mathbf{n}, \mathbf{x}\mathbf{d}) \xrightarrow{\sim} (\mathbf{n}', \mathbf{x}\mathbf{d}')$   
 $\stackrel{\text{def}}{\Leftrightarrow} i \sim j$ .

Below, we also write  $(\mathbf{n}, \mathbf{x}\mathbf{d}) \xrightarrow{\sim} (\mathbf{n}', \mathbf{x}\mathbf{d}')$  instead of  $(\mathbf{n}', \mathbf{x}\mathbf{d}') \xrightarrow{\sim} (\mathbf{n}, \mathbf{x}\mathbf{d})$ .

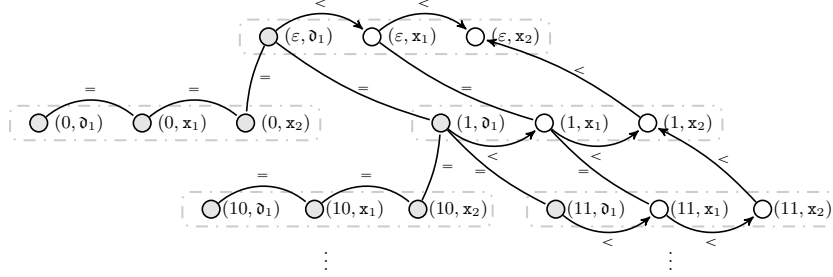


Fig. 4: The labelled graph  $G_{\mathbf{t}}^C$  for the symbolic tree  $\hat{\mathbf{t}}$  from Figure 3

*Example 3 (ctd.).* Consider again the symbolic tree  $\hat{\mathbf{t}}$  on the right of Figure 3. Figure 4 shows a part of the graph  $G_{\mathbf{t}}^C$ . We omitted edges if they can be inferred from the other edges. Grey nodes are in  $U_1$ , all other nodes are in  $U_2$  (no nodes in the figure are in  $U_0$ ).

The rationale behind the construction of  $G_{\mathbf{t}}^C$  is best illustrated below.

**Lemma 7.** *Let  $\mathbf{t}$  be a locally consistent symbolic tree and  $\mathbf{t}^* : [0, D-1]^* \rightarrow \mathbb{Z}^\beta$  be such that  $\mathbf{t}^*$  witnesses the satisfaction of  $\mathbf{t}$ . Then, if  $(\mathbf{n}, \mathbf{x}\mathbf{d}) \xrightarrow{\sim} (\mathbf{n}', \mathbf{x}\mathbf{d}')$  is an edge in  $G_{\mathbf{t}}^C$  for some  $\sim \in \{<, =\}$ , then  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}\mathbf{d}) \sim \mathbf{t}^*(\mathbf{n}')(\mathbf{x}\mathbf{d}')$ .*

By convention,  $\mathbf{t}^*(\mathbf{n})(\mathfrak{d}_1) \stackrel{\text{def}}{=} \mathfrak{d}_1$  and  $\mathbf{t}^*(\mathbf{n})(\mathfrak{d}_\alpha) \stackrel{\text{def}}{=} \mathfrak{d}_\alpha$ .

*Proof.* Suppose  $(\mathbf{n}, \mathbf{x}\mathbf{d}) \xrightarrow{\sim} (\mathbf{n}', \mathbf{x}\mathbf{d}')$  is an edge in  $G_{\mathbf{t}}^C$ . Let  $\mathbf{t}(\mathbf{n}) = (\cdot, \Theta_{\mathbf{n}})$  and  $\mathbf{t}(\mathbf{n}') = (\cdot, \Theta_{\mathbf{n}'})$ . We use  $\Theta'_{\mathbf{n}}$  to denote the restriction of  $\Theta_{\mathbf{n}}$  to  $\{\mathbf{x}'_1, \dots, \mathbf{x}'_\beta\}$ ; similarly for  $\Theta'_{\mathbf{n}'}$ . By (VAR), there are three cases possible.

- Suppose  $\mathbf{x}\mathbf{d} = \mathbf{x}_i$ ,  $\mathbf{x}\mathbf{d}' = \mathbf{x}_{i'}$  for some  $1 \leq i, i' \leq \beta$ .
  - Suppose  $\mathbf{n}' = \mathbf{n} \cdot j$  for some  $j \in [0, D-1]$ , and  $\mathbf{x}_i = \mathbf{x}'_{i'} \in \Theta_{\mathbf{n}'}$ . By  $\mathbb{Z} \models \Theta_{\mathbf{n}'}(\mathbf{t}^*(\mathbf{n}), \mathbf{t}^*(\mathbf{n}'))$  ( $\mathbf{t}^*$  witnesses the satisfaction of  $\mathbf{t}$ ) we can conclude  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i) = \mathbf{t}^*(\mathbf{n}')(\mathbf{x}_{i'})$ .
  - Suppose  $\mathbf{n}' = \mathbf{n}$  and  $\mathbf{x}'_i = \mathbf{x}'_{i'} \in \Theta_{\mathbf{n}'}$ . By  $\mathbb{Z} \models \Theta'_{\mathbf{n}'}(\mathbf{t}^*(\mathbf{n}'))$  ( $\mathbf{t}^*$  witnesses the satisfaction of  $\mathbf{t}$ ) we can conclude  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i) = \mathbf{t}^*(\mathbf{n}')(\mathbf{x}_{i'})$ .
  - Suppose  $\mathbf{n} = \mathbf{n}' \cdot j$  for some  $j \in [0, D-1]$ , and  $\mathbf{x}'_i = \mathbf{x}_{i'} \in \Theta_{\mathbf{n}}$ . By  $\mathbb{Z} \models \Theta_{\mathbf{n}}(\mathbf{t}^*(\mathbf{n}'), \mathbf{t}^*(\mathbf{n}))$  ( $\mathbf{t}^*$  witnesses the satisfaction of  $\mathbf{t}$ , and we omit this precision in the sequel) we can conclude  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i) = \mathbf{t}^*(\mathbf{n}')(\mathbf{x}_{i'})$ .

- Suppose  $\mathbf{x}\mathbf{d} = \mathbf{x}_i$  for some  $1 \leq i \leq \beta$  and  $\mathbf{x}\mathbf{d}' = \mathfrak{d}_1$ . By (PART4),  $(\mathbf{n}', \mathfrak{d}_1) \in U_1$ . By (CONS),  $(\mathbf{n}, \mathbf{x}_i) \in U_1$ . By (PART1),  $\mathbf{x}'_i = \mathfrak{d}_1 \in \Theta_{\mathbf{n}}$ . By  $\mathbb{Z} \models \Theta'_{\mathbf{n}}(\mathbf{t}^*(\mathbf{n}))$  we can conclude that  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i) = \mathfrak{d}_1$ . Hence  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i) = \mathbf{t}^*(\mathbf{n}')(\mathfrak{d}_1)$ .
- Suppose  $\mathbf{x}\mathbf{d} = \mathfrak{d}_1$  and  $\mathbf{x}\mathbf{d}' = \mathbf{x}_i$  for some  $1 \leq i \leq \beta$ . The proof is symmetric to the proof for the previous case.
- Suppose  $\mathbf{x}\mathbf{d} = \mathfrak{d}_\alpha$  or  $\mathbf{x}\mathbf{d}' = \mathfrak{d}_\alpha$ . The proof is very similar to the proof for the previous two cases.

Now suppose  $(\mathbf{n}, \mathbf{x}\mathbf{d}) \xrightarrow{\sim} (\mathbf{n}', \mathbf{x}\mathbf{d}')$  is an edge in  $G_{\mathbf{t}}^C$ .

- Suppose  $\mathbf{x}\mathbf{d} = \mathbf{x}_i$ ,  $\mathbf{x}\mathbf{d}' = \mathbf{x}_{i'}$  for some  $1 \leq i, i' \leq \beta$  by (VAR). The proof is analogous to the proof for the corresponding case for  $(\mathbf{n}, \mathbf{x}\mathbf{d}) \xrightarrow{=} (\mathbf{n}', \mathbf{x}\mathbf{d}')$ .
- Suppose  $\mathbf{x}\mathbf{d}' = \mathfrak{d}_1$ . Then  $\mathbf{x}\mathbf{d} = \mathbf{x}_i$  for some  $1 \leq i \leq \beta$  and  $(\mathbf{n}, \mathbf{x}_i) \in U_0$  by (CONS). Then  $\mathbf{x}'_i < \mathfrak{d}_1 \in \Theta_{\mathbf{n}}$  by (PART2). By  $\mathbb{Z} \models \Theta'_{\mathbf{n}}(\mathbf{t}^*(\mathbf{n}))$  we can conclude  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i) < \mathfrak{d}_1$ , and hence  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i) < \mathbf{t}^*(\mathbf{n}')(\mathfrak{d}_1)$ .
- Suppose  $\mathbf{x}\mathbf{d} = \mathfrak{d}_1$ . Then – by (CONS) – we have  $(\mathbf{n}', \mathbf{x}\mathbf{d}') \in U_j$  for some  $1 < j \leq (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2$ .
  - Suppose  $1 < j \leq (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1$  and  $\mathbf{x}\mathbf{d}' = \mathbf{x}_i$  for some  $i \in [1, \beta]$ . By (PART1), we have  $\mathbf{x}'_i = \mathfrak{d}_1 + (j - 1) \in \Theta_{\mathbf{n}'}$ . By  $\mathbb{Z} \models \Theta'_{\mathbf{n}'}(\mathbf{t}^*(\mathbf{n}'))$  we can conclude  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i) = \mathfrak{d}_1 + (j - 1)$ . Hence  $\mathbf{t}^*(\mathbf{n})(\mathfrak{d}_1) < \mathbf{t}^*(\mathbf{n}')(\mathbf{x}_i)$ .
  - Suppose  $j = (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1$  and  $\mathbf{x}\mathbf{d}' = \mathfrak{d}_\alpha$ . Then obviously  $\mathbf{t}^*(\mathbf{n})(\mathfrak{d}_1) < \mathbf{t}^*(\mathbf{n}')(\mathfrak{d}_\alpha)$ .
  - Suppose  $j = (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2$ . Then  $\mathbf{x}\mathbf{d}' = \mathbf{x}_i$  for some  $i \in [1, \beta]$ , and  $\mathbf{x}'_i > \mathfrak{d}_\alpha \in \Theta_{\mathbf{n}'}$  by (PART3). By  $\mathbb{Z} \models \Theta'_{\mathbf{n}'}(\mathbf{t}^*(\mathbf{n}'))$  we can conclude that  $\mathbf{t}^*(\mathbf{n}')(\mathbf{x}_i) > \mathfrak{d}_\alpha$ , so that  $\mathbf{t}^*(\mathbf{n})(\mathfrak{d}_1) < \mathbf{t}^*(\mathbf{n}')(\mathbf{x}_i)$ .
- Suppose  $\mathbf{x}\mathbf{d} = \mathfrak{d}_\alpha$ . Then  $\mathbf{x}\mathbf{d}' = \mathbf{x}_i$  for some  $i \in [1, \beta]$  and  $(\mathbf{n}', \mathbf{x}_i) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$ . Then  $\mathbf{x}'_i > \mathfrak{d}_\alpha \in \Theta'$ . By  $\mathbb{Z} \models \Theta'_{\mathbf{n}'}(\mathbf{t}^*(\mathbf{n}'))$ , we can conclude that  $\mathbf{t}^*(\mathbf{n}')(\mathbf{x}_i) > \mathfrak{d}_\alpha$ . Hence  $\mathbf{t}^*(\mathbf{n})(\mathfrak{d}_\alpha) < \mathbf{t}^*(\mathbf{n}')(\mathbf{x}_i)$ .
- Suppose  $\mathbf{x}\mathbf{d}' = \mathfrak{d}_\alpha$ . Then  $(\mathbf{n}, \mathbf{x}\mathbf{d}) \in U_j$  for some  $0 \leq j < (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1$ . The proof is very similar to the last but one case.  $\square$

A *path*  $\pi$  in  $G_{\mathbf{t}}^C$  is a (possibly infinite) sequence  $\pi = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim} (\mathbf{n}_1, \mathbf{x}\mathbf{d}_1) \cdots \xrightarrow{\sim} (\mathbf{n}_i, \mathbf{x}\mathbf{d}_i) \cdots$  such that  $\{\sim_1, \sim_2, \dots\} \subseteq \{=, <\}$ . Given a finite path  $\pi = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim} (\mathbf{n}_1, \mathbf{x}\mathbf{d}_1) \cdots \xrightarrow{\sim} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$ , its *strict length*, written  $\text{slen}(\pi)$ , is the number of edges labelled by ' $<$ ' in  $\pi$ , i.e.  $\text{card}(\{i \in [1, n] \mid \sim_i \text{ equal to } <\})$ . Given two nodes  $(\mathbf{n}, \mathbf{x}\mathbf{d})$  and  $(\mathbf{n}', \mathbf{x}\mathbf{d}')$ , the *strict length* from  $(\mathbf{n}, \mathbf{x}\mathbf{d})$  to  $(\mathbf{n}', \mathbf{x}\mathbf{d}')$ , written  $\text{slen}((\mathbf{n}, \mathbf{x}\mathbf{d}), (\mathbf{n}', \mathbf{x}\mathbf{d}'))$ , is the supremum of all the strict lengths of paths from  $(\mathbf{n}, \mathbf{x}\mathbf{d})$  to  $(\mathbf{n}', \mathbf{x}\mathbf{d}')$ . Though the strict length of any finite path is always finite,  $\text{slen}((\mathbf{n}, \mathbf{x}\mathbf{d}), (\mathbf{n}', \mathbf{x}\mathbf{d}'))$  may be infinite. Given  $(\mathbf{n}, \mathbf{x}) \in U_0$ , its *strict length*, written  $\text{slen}(\mathbf{n}, \mathbf{x})$ , is defined as  $\text{slen}((\mathbf{n}, \mathbf{x}), (\mathbf{n}, \mathfrak{d}_1))$ . Given  $(\mathbf{n}, \mathbf{x}) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$ , its *strict length*, written  $\text{slen}(\mathbf{n}, \mathbf{x})$ , is defined as  $\text{slen}((\mathbf{n}, \mathfrak{d}_\alpha), (\mathbf{n}, \mathbf{x}))$ .

A map  $p : \mathbb{N} \rightarrow [0, D - 1]^* \times \mathbf{T}(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$  is a *path map* in  $G_{\mathbf{t}}^C \stackrel{\text{def}}{\Leftrightarrow}$  for all  $i \in \mathbb{N}$ , either  $p(i) \xrightarrow{=} p(i + 1)$  or  $p(i) \xrightarrow{<} p(i + 1)$  in  $G_{\mathbf{t}}^C$ . Similarly,  $rp : \mathbb{N} \rightarrow [0, D - 1]^* \times \mathbf{T}(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$  is a *reverse path map* in  $G_{\mathbf{t}}^C \stackrel{\text{def}}{\Leftrightarrow}$  for all  $i \in \mathbb{N}$ , either  $rp(i) \xrightarrow{=} rp(i + 1)$  or  $rp(i + 1) \xrightarrow{<} rp(i)$ . A map  $p$  (resp.  $rp$ ) is *strict*  $\stackrel{\text{def}}{\Leftrightarrow} \{i \in \mathbb{N} \mid p(i) \xrightarrow{<} p(i + 1)\}$  (resp.  $\{i \in \mathbb{N} \mid rp(i + 1) \xrightarrow{<} rp(i)\}$ ) is infinite. These definitions make sense because between two elements there is at most one labelled edge (or maybe two but with the same equality sign).

An *infinite branch*  $\mathcal{B}$  is an element of  $[0, D - 1]^\omega$ . We write  $\mathcal{B}[i, j]$  with  $i \leq j$  to denote the subsequence  $\mathcal{B}(i) \cdot \mathcal{B}(i + 1) \cdots \mathcal{B}(j)$ . A path map  $p$  *from*  $(\mathbf{n}, \mathbf{x}\mathbf{d})$  *along*  $\mathcal{B}$  is such that  $p(0) = (\mathbf{n}, \mathbf{x}\mathbf{d})$  and for all  $i \geq 0$ ,  $p(i)$  is of the form  $(\mathbf{n} \cdot \mathcal{B}[0, i], \cdot)$ . A reverse path map  $rp$  *from*  $(\mathbf{n}, \mathbf{x}\mathbf{d})$  *along*  $\mathcal{B}$  admits a similar definition.

Below, we present the condition  $(\star^C)$  that is the central property for characterising regular trees in  $\mathbf{L}(\mathbb{B}_{\text{cons}(A)})$  that are satisfiable.

- $(\star^C)$  There are *no* elements  $(\mathbf{n}, \mathbf{x}\mathbf{d}), (\mathbf{n}, \mathbf{x}\mathbf{d}')$  in  $G_{\mathbf{t}}^C$  (same node  $\mathbf{n}$  from  $[0, D - 1]^*$ ) and no infinite branch  $\mathcal{B}$  such that

1. there exists a path map  $p$  from  $(\mathbf{n}, \mathbf{x}_d)$  along  $\mathcal{B}$ ,
2. there exists a reverse path map  $rp$  from  $(\mathbf{n}, \mathbf{x}_d')$  along  $\mathcal{B}$ ,
3.  $p$  or  $rp$  is strict; 4. for all  $i \in \mathbb{N}$ ,  $p(i) \prec rp(i)$ .

*Example 4.* Let the infinite branch  $\mathcal{B}$  be of the form  $1^\omega$ , and reconsider the graph  $G_{\mathbf{t}}^c$  in Figure 4. We identify the beginning of two maps  $rp$  and  $p$  in  $G_{\mathbf{t}}^c$  along  $\mathcal{B}$  that satisfy points 1.–4.:  $p$  is the path map from  $(\varepsilon, \mathbf{x}_1)$  along  $\mathcal{B}$ , and  $rp$  is the (strict) reverse path map from  $(\varepsilon, \mathbf{x}_2)$ .

To illustrate  $(\star^c)$ , we present below the beginning of two maps  $rp$  and  $p$  in  $G_{\mathbf{t}}^c$  (along the same branch) that satisfy the above points 1.–4. For instance, each node in  $\mathbf{t}$  from  $rp(i+1)$  (resp.  $p(i+1)$ ) is a child of the node from  $rp(i)$  (resp.  $p(i)$ ).

$$\begin{array}{ccccccc}
 (\mathbf{n}, \mathbf{x}_d') = & rp(0) & \xrightarrow{=} & rp(1) & \xrightarrow{>} & rp(2) & \xrightarrow{=} & rp(3) & \xrightarrow{>} & \dots \\
 & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \\
 & < & & < & & < & & < & \\
 (\mathbf{n}, \mathbf{x}_d) = & p(0) & \xrightarrow{<} & p(1) & \xrightarrow{=} & p(2) & \xrightarrow{=} & p(3) & \xrightarrow{<} & \dots
 \end{array}$$

Before going any further, we explain to the reader the relationships between  $G_{\mathbf{t}}^c$  with the condition  $(\star^c)$ , and the developments in [LOS20, Lab21]. To do so, we write  $G_{\mathbf{t}}$  to denote the labelled graph obtained from  $G_{\mathbf{t}}^c$  by removing all the elements in  $\mathcal{D}_{\mathbf{t}}^c = [0, D-1]^* \times \{\mathfrak{d}_1, \mathfrak{d}_\alpha\}$  as well as the edges involving those elements. Once we observe that the graphs  $G_{\mathbf{t}}^c$  correspond to so-called *framified constraint graphs* defined in [LOS20, Definition 14] obtained from consistent trees [LOS20, Definition 18], the condition  $(\star)$  on framified constraint graphs in [LOS20, Section 3.3] corresponds to our condition  $(\star^c)$  but applied to  $G_{\mathbf{t}}$  only (proving such a correspondence requires a tedious proof but with no real technical difficulties). Moreover, the condition  $(\star)$  in [LOS20, Section 3.3] generalises the condition  $C_{\mathbb{Z}}$  from [DD07, Section 6] (see also the condition  $\mathcal{C}$  in [DG08, Definition 2] and a similar condition in [EFK21, Section 2]). In short, the condition  $(\star)$  (only) requires the condition  $C_{\mathbb{Z}}$  on every branch.

It remains to explain why we introduced the graph  $G_{\mathbf{t}}^c$  and not its restriction  $G_{\mathbf{t}}$  following [LOS20, Lab21]. This change means for us that we have to prove all the results using  $G_{\mathbf{t}}^c$ , instead of, using  $G_{\mathbf{t}}$  and evoking the results in [LOS20, Lab21] (our initial intention by taking advantage of the results about framified constraint graphs). Let  $D = 1$ ,  $\beta = 1$ ,  $\alpha = 2$ ,  $\mathfrak{d}_1 = -2$ ,  $\mathfrak{d}_2 = 3$  and  $\Theta \in \text{SatTypes}(1)$  be the constraint  $\Theta = (\mathbf{x}_1 > \mathbf{x}_1') \wedge \mathbf{x}_1 > 3 \wedge \mathbf{x}_1' > 3$ . Obviously  $\Theta$  is satisfiable and  $\mathbf{w} = \Theta^\omega$  is a locally consistent symbolic word (symbolic tree with  $D = 1$  in  $L(\mathbb{B}_{\text{cons}(\Lambda)})$ ) that is also regular. Moreover,  $G_{\mathbf{w}}$  satisfies the condition  $(\star)$ . According to [LOS20, Lemma 22],  $\mathbf{w}$  should be satisfiable but actually this is not the case: at some point, the value of  $\mathbf{x}_1$  must become smaller than 3, so that the constraint  $\mathbf{x}_1 > 3$  is not satisfied anymore. In short, the way constant values are handled in the condition  $(\star)$  is not satisfactory and the simple patch we propose (once one accepts the correspondence with framified constraint graphs) is to add the elements in  $\mathcal{D}_{\mathbf{t}}^c$  while adapting  $(\star)$  to  $(\star^c)$ . In short,  $(\star^c)$  restricted to  $G_{\mathbf{t}}$  (which is not good enough) corresponds to  $(\star)$  on framified constraint graphs in [LOS20]. After scrutiny and to be exhaustive, the problematic part in [LOS20, Lab21] is due to [Lab21, Lemma 5.18] whose main argument takes advantage of [CKL16] (see also [DG08, Lemma 8]). Actually, as far as we can judge, the statement would hold if we add the elements related to constant values in the labelled graph. In this paper, we propose a proof to characterise satisfiability of symbolic trees that is independent of [CKL16] and simpler because tailored to our needs, see Lemma 24. Admittedly, our patch with  $(\star^c)$  is simple however, it generates some extra-work to prove the long chain of arguments to establish Proposition 4 (see Section 7 for a self-contained proof).

Before explaining why  $(\star^c)$  is essential for our needs, we would like to note that recently a condition similar to  $\mathcal{C}_{\mathbb{Z}}$  [DD07] and  $(\star)$  is introduced to decide the single-sided realizability problem for the logic  $\text{LTL}(\mathbb{Z}, <, =)$  (no constant values) in [BP22, Lemma 18]. The proof idea for [BP22, Lemma 18] is similar to the proof of [LOS20, Lemma 22] with the feature that witnessing non-satisfiability can be performed along a single branch. This is not really surprising in view of the intimate relationships between tree automata and games, see e.g. [GH82]. Our proof dedicated to the correctness of our condition  $(\star^c)$  goes therefore beyond [BP22, Lemma 18] and [LOS20, Lemma 22] (because we handle adequately the constants).

Now, let us show why  $(\star^c)$  is so crucial for our investigations. As usual,  $\mathbf{t} : [0, D - 1]^* \rightarrow \Sigma \times \text{SatTypes}(\beta)$  is *regular* whenever its set of infinite subtrees is *finite*. More precisely, we write  $\mathbf{t}_{|\mathbf{n}}$  to denote the subtree at the node  $\mathbf{n}$  in  $\mathbf{t}$  defined by: for all nodes  $\mathbf{n}' \in [0, D - 1]^*$ ,  $\mathbf{t}_{|\mathbf{n}}(\mathbf{n}') \stackrel{\text{def}}{=} \mathbf{t}(\mathbf{n} \cdot \mathbf{n}')$ . Consequently, a tree  $\mathbf{t}$  is regular whenever  $\{\mathbf{t}_{|\mathbf{n}} \mid \mathbf{n} \in [0, D - 1]^*\}$  is finite.

**Proposition 4.** *For every regular locally consistent symbolic tree  $\mathbf{t}$ ,  $G_{\mathbf{t}}^c$  satisfies  $(\star^c)$  iff  $\mathbf{t}$  is satisfiable.*

Proposition 4 is a variant of [LOS20, Lemma 22] and states a strong property: non-satisfaction of a regular locally consistent symbolic tree can be witnessed along a single branch by violation of  $(\star^c)$ . Hence, satisfiability of symbolic trees is not a regular property (see e.g. [DD07, LOS20]) but it can be overapproximated advantageously. Forthcoming Section 7 is fully dedicated to the lengthy proof of Proposition 4.

As shown below,  $(\star^c)$  is a regular property and therefore there is a Rabin tree automaton  $\mathbb{A}_{\star^c}$  that captures it. Before defining  $\mathbb{A}_{\star^c}$ , let us provide an essential consequence of its existence: the statements below are equivalent.

- (A<sub>1</sub>)  $L(\mathbb{A}) \neq \emptyset$ .
- (A<sub>2</sub>) There is a symbolic tree in  $L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  that is satisfiable.
- (A<sub>3</sub>) There is a symbolic tree in  $L(\mathbb{B}_{\text{cons}(\mathbb{A})}) \cap L(\mathbb{A}_{\star^c})$ .

The equivalence between (A<sub>1</sub>) and (A<sub>2</sub>) is by Lemma 6. The condition (A<sub>2</sub>) implies (A<sub>3</sub>) follows from the fact that every satisfiable symbolic tree necessarily satisfies the condition  $(\star^c)$  (Lemma 7),  $L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  contains all the satisfiable symbolic trees (Lemma 4 and Lemma 5) and  $L(\mathbb{A}_{\star^c})$  is equal to the set of the symbolic trees satisfying the condition  $(\star^c)$ . The condition (A<sub>3</sub>) implies (A<sub>2</sub>) follows from the fact that if  $L(\mathbb{B}_{\text{cons}(\mathbb{A})}) \cap L(\mathbb{A}_{\star^c})$  is non-empty, then  $L(\mathbb{B}_{\text{cons}(\mathbb{A})}) \cap L(\mathbb{A}_{\star^c})$  is regular and therefore contains a regular  $\mathbb{A}$ -consistent symbolic tree  $\mathbf{t}$  (see e.g. [Rab69] and [Tho90, Section 6.3] for the existence of regular trees) and by Proposition 4,  $\mathbf{t}$  is satisfiable.

To come back to the construction of  $\mathbb{A}_{\star^c}$ , first note that explanations about the construction of  $\mathbb{A}_{\star}$  (the counterpart of  $\mathbb{A}_{\star^c}$  on  $G_{\mathbf{t}}$ ) are provided in [LOS20]. Below, nevertheless, we propose a slight novelty by designing  $\mathbb{A}_{\star^c}$  without firstly constructing *a tree automaton for the complement language* (as done in [LOS20]) and then using results from [MS95] (elimination of alternation in tree automata). Our new approach shall be rewarding: not only we can better understand how to express the condition  $(\star^c)$  but also we fully control the size parameters of  $\mathbb{A}_{\star^c}$  involved in our forthcoming complexity analysis. Furthermore, this becomes central in case of a hypothetical future implementation of the decision procedure for solving the satisfiability problem for  $\text{CTL}(\mathbb{Z})$  (resp. for  $\text{CTL}^*(\mathbb{Z})$ ).

The construction of  $G_{\mathbf{t}}^c$  can be done even if  $\mathbf{t}$  is not locally consistent, and the condition  $(\star^c)$  can be defined on such labelled graphs too. The trees accepted by construction of  $\mathbb{A}_{\star^c}$  in Lemma 8 are not necessarily locally consistent, but this condition is taken care of by  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  later on.

**Lemma 8.** *There is a Rabin tree automaton  $\mathbb{A}_{\star^c}$  characterising the trees satisfying the condition  $(\star^c)$  such that*

- *the number of Rabin pairs is bounded above by  $8(\beta + 2)^2 + 3$ ,*
- *the number of locations is exponential in  $\beta$ ,*
- *the transition relation can be decided in polynomial-time in*

$$\max(\lceil \log(|\mathfrak{d}_1|) \rceil, \lceil \log(|\mathfrak{d}_\alpha|) \rceil) + \beta + \text{card}(\Sigma) + D.$$

As a consequence, the transition relation for  $\mathbb{A}_{\star^c}$  has  $\mathcal{O}(\text{card}(\Sigma \times \text{SatTypes}(\beta)) \times (2^{p^*(\beta)})^{(D+1)})$  transitions for some polynomial  $p^*$ , which is in  $\mathcal{O}(\text{card}(\Sigma) \times ((\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2)^{2\beta} 3^{\beta^2} \times 2^{p^*(\beta) \times (D+1)})$ .

Lemma 8 is similar to [LOS20, Proposition 26] but there is an essential difference: the number of Rabin pairs in Lemma 8 is not a constant but a value depending on  $\beta$ . It is important to know the number of Rabin pairs in  $\mathbb{A}_{\star^c}$  for our subsequent complexity analysis as checking nonemptiness of Rabin tree automata is *exponential* in the number of Rabin pairs [EJ00, Theorem 4.1].

In the proof of Lemma 8, we start defining a Büchi word automaton  $\mathbb{A}_B$  that captures the complement of  $(\star^c)$  when  $D = 1$ . We then take advantage of two classical results by Safra: using [Saf89, Theorem 1.1], we construct a *deterministic* Streett word automaton  $\mathbb{A}_S$  that captures the condition  $(\star^c)$ , and then we use [Saf89, Lemma 1.2] to define a deterministic Rabin word automaton  $\mathbb{A}_R$  accepting the same language. The final stage consists in computing the deterministic Rabin tree automaton  $\mathbb{A}_{\star^c}$  (thanks to the determinism of the word automaton  $\mathbb{A}_R$ ).

*Proof.* The first stage is defining a Büchi word automaton accepting words over  $\Sigma \times \text{SatTypes}(\beta)$  that contain a node  $\mathbf{n}$  that does not satisfy the condition  $(\star^c)$ , similarly to what is done in [DD07, Section 6] and [LOS20, Section 3.4]. Let  $\mathbb{A}_B = (Q_B, \Sigma \times \text{SatTypes}(\beta), Q_{B,\text{in}}, \delta_B, F_B)$  be the Büchi word automaton defined as follows. A *local thread* is a tuple  $(\mathbf{xd}_1, \mathbf{xd}_2, d, f)$ , where  $\mathbf{xd}_1, \mathbf{xd}_2 \in \mathbf{T}(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$ ,  $d \in \{p, rp\}$  and  $f \in \{<, =\}$ , where  $\mathbf{xd}_1$  is the current element in the path map,  $\mathbf{xd}_2$  is the current element in the reverse path map,  $d$  indicates whether the path map or the reverse path map is strict (formally,  $d = p$  iff the path map is strict), and the flag  $f$  indicates whether the automaton has just seen a strict edge for the map indicated by  $d$  (formally,  $f = <$  iff the last edge was strict). The automaton  $\mathbb{A}_B$  remembers local threads in its locations; the transition relation is defined to guarantee that a sequence of local threads forms an infinite branch along which a path map and a reverse path map do not satisfy condition  $(\star^c)$ .

- $Q_B = \{q_{\text{in}}\} \cup (\mathbf{T}(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)^2 \times \{p, rp\} \times \{<, =\})$ ;  $Q_{B,\text{in}} = \{q_{\text{in}}\}$ .
- $\delta_B$  is the union of the following sets.
  - $\{(q_{\text{in}}, (\mathbf{a}, \Theta), q_{\text{in}}) \mid (\mathbf{a}, \Theta) \in \Sigma \times \text{SatTypes}(\beta)\}$ .
  - $\{(q_{\text{in}}, (\mathbf{a}, \Theta), (\mathbf{xd}_1, \mathbf{xd}_2, d, f)) \mid (\mathbf{a}, \Theta) \in \Sigma \times \text{SatTypes}(\beta) \text{ such that } \Theta \models \mathbf{xd}'_1 < \mathbf{xd}'_2, d \in \{p, rp\}, f \in \{<, =\}\} \text{ (initialization of a violating thread)}.$
  - $\{((\mathbf{xd}_1, \mathbf{xd}_2, p, f), (\mathbf{a}, \Theta), (\mathbf{xd}_3, \mathbf{xd}_4, p, f_{\text{new}})) \mid (\mathbf{a}, \Theta) \in \Sigma \times \text{SatTypes}(\beta) \text{ such that } \Theta \models (\mathbf{xd}'_3 < \mathbf{xd}'_4) \wedge (\mathbf{xd}_1 \neq \mathbf{xd}'_3) \wedge ((\mathbf{xd}'_4 = \mathbf{xd}_2) \vee (\mathbf{xd}'_4 < \mathbf{xd}_2)), f, f_{\text{new}} \in \{<, =\}\}$ .
  - $\{((\mathbf{xd}_1, \mathbf{xd}_2, rp, f), (\mathbf{a}, \Theta), (\mathbf{xd}_3, \mathbf{xd}_4, rp, f_{\text{new}})) \mid (\mathbf{a}, \Theta) \in \Sigma \times \text{SatTypes}(\beta) \text{ such that } \Theta \models (\mathbf{xd}'_3 < \mathbf{xd}'_4) \wedge (\mathbf{xd}'_4 \neq \mathbf{xd}_2) \wedge ((\mathbf{xd}_1 = \mathbf{xd}'_3) \vee (\mathbf{xd}_1 < \mathbf{xd}'_3)), f, f_{\text{new}} \in \{<, =\}\}$ .
- $F_B \stackrel{\text{def}}{=} \{(\mathbf{xd}_1, \mathbf{xd}_2, d, <) \in Q_B \mid \mathbf{xd}_1, \mathbf{xd}_2 \in \mathbf{T}(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha), d \in \{p, rp\}\}$ .

Let us sketch the correctness of this definition, namely for all  $\mathbf{w} : \mathbb{N} \rightarrow \Sigma \times \text{SatTypes}(\beta)$ ,  $\mathbf{w} \in L(\mathbb{A}_B)$  iff  $\mathbf{w}$  does not satisfy the condition  $(\star^c)$ . So let  $\mathbf{w} \in L(\mathbb{A}_B)$ . Then there exists some run of the form  $q_0 \xrightarrow{\mathbf{w}(0)} q_1 \xrightarrow{\mathbf{w}(1)} \dots$  satisfying  $q_0 = q_{\text{in}}$ ,  $(q_i, \mathbf{w}(i), q_{i+1}) \in \delta_B$ , and for infinitely many positions  $i \geq 0$  we

have  $q_i \in F_B$ , that is,  $q_i = (\mathbf{xd}_1^i, \mathbf{xd}_2^i, d^i, <)$  for some  $\mathbf{xd}_1^i, \mathbf{xd}_2^i, d^i$ . By definition of  $\delta_B$  there must exist some position  $i \geq 1$  such that  $q_k = q_{\text{in}}$  for all  $0 \leq k < i$ , and  $q_k = (\mathbf{xd}_1^k, \mathbf{xd}_2^k, d^i, f^k)$  for all  $k \geq i$ . Note that  $d^k = d^i$  for the direction flag, because  $\delta$  does not allow to switch from  $(\cdot, \cdot, p, \cdot)$  to  $(\cdot, \cdot, rp, \cdot)$  or the other way round. It is now easy to identify a thread  $(p, rp)$  from the run starting at position  $i$ ; if  $d^i = p$ , the path map is strict; otherwise the reverse map is strict. Hence  $\mathbf{w}$  does not satisfy condition  $(\star^c)$ . It is also easy to see that  $\mathbf{w} \in L(\mathbb{A}_B)$  if  $\mathbf{w}$  does not satisfy condition  $(\star^c)$ .

Note that the number of locations in  $\mathbb{A}_B$  is bounded above by  $4(\beta + 2)^2 + 1$ . In the second stage, we use [Saf89, Theorem 1.1] to obtain a deterministic Rabin word automaton  $\mathbb{A}_{B \rightarrow R} = (Q_{B \rightarrow R}, \Sigma \times \text{SatTypes}(\beta), Q_{B \rightarrow R, \text{in}}, \delta_{B \rightarrow R}, \mathcal{F}_{B \rightarrow R})$  such that  $L(\mathbb{A}_{B \rightarrow R}) = L(\mathbb{A}_B)$ , see also Section 6.3 for an adaptation to constraint automata, which extends the case with finite alphabets. The cardinality of  $Q_{B \rightarrow R}$  is in  $2^{\mathcal{O}(\text{card}(Q_B) \log(\text{card}(Q_B)))}$ , i.e. exponential in  $\beta$ , and the number of acceptance pairs in  $\mathcal{F}_{B \rightarrow R}$  is equal to  $2 \times \text{card}(Q_B)$ , i.e. equal to  $8(\beta + 2)^2 + 2$ . Without any loss of generality, we can assume that  $\mathbb{A}_{B \rightarrow R}$  is also complete (apart from being deterministic). As a consequence, for any  $\mathbf{w} \in (\Sigma \times \text{SatTypes}(\beta))^\omega$ , there is a unique run  $\rho_{\mathbf{w}}$  on  $\mathbf{w}$  but  $\rho_{\mathbf{w}}$  is not necessarily accepting. We write  $\mathbb{A}_S = (Q_S, \Sigma \times \text{SatTypes}(\beta), Q_{S, \text{in}}, \delta_S, \mathcal{F}_S)$  to denote the Streett automaton accepting the complement language of  $L(\mathbb{A}_{B \rightarrow R})$ . All the components of  $\mathbb{A}_S$  are those from  $\mathbb{A}_{B \rightarrow R}$  but  $\mathcal{F}_S$  in  $\mathbb{A}_S$  is interpreted as a Streett condition (we recall that the negation of a Rabin condition is a Streett condition). Consequently, the deterministic Streett word automaton  $\mathbb{A}_S$  is syntactically equal to  $\mathbb{A}_{B \rightarrow R}$  and we have  $\mathbf{w} \in L(\mathbb{A}_S)$  iff  $\mathbf{w}$  satisfies the condition  $(\star^c)$ .

As a last stage, we define the Rabin word automaton  $\mathbb{A}_R$  such that  $L(\mathbb{A}_R) = L(\mathbb{A}_S)$ , based on the reduction from deterministic Streett word automata to deterministic Rabin word automata in [Saf89, Lemma 1.2]. We do not formally provide the correctness proof for the reduction but we refer to the proof of [Saf89, Lemma 1.2] for the general case.

We build the Rabin word automaton  $\mathbb{A}_R = (Q_R, \Sigma \times \text{SatTypes}(\beta), Q_{R, \text{in}}, \delta_R, \mathcal{F}_R)$  from  $\mathbb{A}_S$ . Before defining its components, we define a few notions. Moreover, we assume that  $\mathbb{A}_S$  has an additional  $(8(\beta + 2)^2 + 3)$ th Streett pair, namely  $(Q_S, Q_S)$ , which is technically helpful to design  $\mathbb{A}_R$  but does not change at all the language  $L(\mathbb{A}_S)$ . Hence, from now, we assume that  $\mathbb{A}_S$  has  $8(\beta + 2)^2 + 3$  Streett pairs, and  $\mathcal{F}_R$  shall contain too the same number of Rabin pairs (details will follow).

We write  $\mathbf{S}_{(8(\beta+2)^2+3)}$  to denote the set of permutations on  $[1, 8(\beta + 2)^2 + 3]$ , a permutation  $\sigma$  is a bijection  $\sigma : [1, 8(\beta + 2)^2 + 3] \rightarrow [1, 8(\beta + 2)^2 + 3]$ . It is well-known that  $\text{card}(\mathbf{S}_{(8(\beta+2)^2+3)}) = (8(\beta + 2)^2 + 3)!$ . Let us define the maps  $\mathbf{g}_1, \mathbf{g}_2 : Q_S \times \mathbf{S}_{(8(\beta+2)^2+3)} \rightarrow [1, 8(\beta + 2)^2 + 3]$  and  $\mathbf{g}_3 : Q_S \times \mathbf{S}_{(8(\beta+2)^2+3)} \rightarrow \mathbf{S}_{(8(\beta+2)^2+3)}$  that are instrumental to define the forthcoming transition relation  $\delta_R$ . Below,  $q \in Q_S$  and  $\sigma \in \mathbf{S}_{(8(\beta+2)^2+3)}$ .

- We set  $\mathbf{g}_1(q, \sigma) \stackrel{\text{def}}{=} \min\{i \in [1, 8(\beta + 2)^2 + 3] \mid q \in U_{\sigma(i)}\}$  ( $U_{\sigma(i)}$  is from the set of pairs  $\mathcal{F}_S$ ). A minimal value always exists thanks to the addition of the new Streett pair  $(Q_S, Q_S)$ .
- The value  $\mathbf{g}_2(q, \sigma)$  is defined using the sets  $L_i$ 's (from the set of pairs  $\mathcal{F}_S$ ):  $\mathbf{g}_2(q, \sigma) \stackrel{\text{def}}{=} \min\{i \in [1, 8(\beta + 2)^2 + 3] \mid q \in L_{\sigma(i)}\}$ .
- The permutation  $\mathbf{g}_3(q, \sigma)$  is obtained from  $\sigma$  by *moving*  $\sigma(\mathbf{g}_1(q, \sigma))$  to the rightmost position (so that  $\sigma(\mathbf{g}_1(q, \sigma))$  is at position  $8(\beta + 2)^2 + 3$  now). Formally

$$\mathbf{g}_3(q, \sigma)(i) \stackrel{\text{def}}{=} \begin{cases} \sigma(i) & \text{if } 1 \leq i < \mathbf{g}_1(q, \sigma) \\ \sigma(i + 1) & \text{if } \mathbf{g}_1(q, \sigma) \leq i \leq 8(\beta + 2)^2 + 2 \\ \sigma(\mathbf{g}_1(q, \sigma)) & \text{if } i = 8(\beta + 2)^2 + 3 \end{cases}$$

The automaton  $\mathbb{A}_R$  is defined as follows.



- $Q_R \stackrel{\text{def}}{=} Q_S \times \mathbf{S}_{(8(\beta+2)^2+3)} \times [1, 8(\beta+2)^2+3]^2$ , which is exponential in  $\beta$ .
- The transition relation  $\delta_R$  is defined as follows:  $(q', \sigma', e', f') \in \delta_S((q, \sigma, e, f), \mathbf{a})$  iff  $q' \in \delta_S(q, \mathbf{a})$ ,  $\sigma' = \mathbf{g}_3(q', \sigma)$ ,  $e' = \mathbf{g}_1(q', \sigma)$  and  $f' = \mathbf{g}_2(q', \sigma)$ . Since  $\mathbf{g}_1(q', \sigma)$ ,  $\mathbf{g}_2(q', \sigma)$  and  $\mathbf{g}_3(q', \sigma)$  can be computed in polynomial-time in  $\beta$ , we get that  $\delta_R$  can be computed in polynomial-time in  $\max(\lceil \log(|\mathfrak{D}_1|) \rceil, \lceil \log(|\mathfrak{D}_\alpha|) \rceil) + \beta + \text{card}(\Sigma)$ .
- $Q_{R,\text{in}}$  has a unique initial location  $(q_{\text{in}}, id, 8(\beta+2)^2+3, 8(\beta+2)^2+3)$ , with  $q_{\text{in}}$  being the only initial location in  $\mathbb{A}_S$  and identity permutation  $id$ . Since  $\delta_S$  is deterministic and  $Q_{R,\text{in}}$  is a singleton, we can conclude that  $\mathbb{A}_R$  is deterministic too.
- $\mathcal{F}_R$  is made of Rabin pairs  $(L'_i, U'_i)$  with  $i \in [1, 8(\beta+2)^2+3]$  such that

$$L'_i \stackrel{\text{def}}{=} \{(q, \sigma, e, f) \in Q_R \mid e = i\} \quad \text{and} \quad U'_i \stackrel{\text{def}}{=} \{(q, \sigma, e, f) \in Q_R \mid f < i\}.$$

We recall that the Rabin condition  $\mathcal{F}_R$  can be read as follows: along any accepting run, there is  $i \in [1, 8(\beta+2)^2+3]$  such that some location in  $L'_i$  occurs infinitely often and all the locations in  $U'_i$  occurs finitely.

By [Saf89, Lemma 1.2], we have  $L(\mathbb{A}_R) = L(\mathbb{A}_S)$ . By way of example, let us briefly explain why  $L(\mathbb{A}_S) \subseteq L(\mathbb{A}_R)$ . Given an accepting run  $\rho$  of  $\mathbb{A}_S$ , there is  $X_\rho \subseteq [1, 8(\beta+2)^2+3]$  such that  $k \in X_\rho$  iff the set  $U_{\sigma(k)}$  is visited infinitely often. Hence, for all  $k \in ([1, 8(\beta+2)^2+3] \setminus X_\rho)$ ,  $U_{\sigma(k)}$  and  $L_{\sigma(k)}$  are visited finitely along  $\rho$ . Moreover, from some position  $I \in \mathbb{N}$  in  $\rho$ , elements of  $([1, 8(\beta+2)^2+3] \setminus X_\rho)$  always occupy the leftmost position in the permutation  $\sigma$  and none of its values change its place. Let  $j = 8(\beta+2)^2+3 - \text{card}(X_\rho)$ . From the position  $I$ , we have  $f \geq j$  and one can show that  $e = j$  infinitely often along  $\rho$ . Since the run visits  $U_{\sigma(j)}$  infinitely often, whenever  $U_{\sigma(j)}$  is visited,  $e$  shall take the value  $j$  (minimal value among  $X_\rho$ ). Hence, the run  $\rho'$  of  $\mathbb{A}_R$  obtained from  $\rho$  by completing deterministically the three last components satisfies the Rabin pair  $(L'_j, U'_j)$ . Consequently, the word accepted by  $\rho$  is also accepted by  $\rho'$ .

The last (easy) stage consists in building the Rabin tree automaton

$$\mathbb{A}_{\star^c} \stackrel{\text{def}}{=} (Q_{\star^c}, \Sigma \times \text{SatTypes}(\beta), D, Q_{\star^c, \text{in}}, \delta_{\star^c}, \mathcal{F}_{\star^c})$$

as follows.

- $Q_{\star^c} \stackrel{\text{def}}{=} Q_R$ ,  $Q_{\star^c, \text{in}} \stackrel{\text{def}}{=} Q_{R, \text{in}}$  and  $\mathcal{F}_{\star^c} \stackrel{\text{def}}{=} \mathcal{F}_R$ .
- For all locations  $q, q_0, \dots, q_{D-1} \in Q_{\star^c}$  and  $(\mathbf{a}, \Theta) \in \Sigma \times \text{SatTypes}(\beta)$ , we have  $(q, (\mathbf{a}, \Theta), q_0, \dots, q_{D-1}) \in \delta_{\star^c}$  iff  $q_0 = \dots = q_{D-1}$  and  $q_0 \in \delta_R(q, (\mathbf{a}, \Theta))$ . Since  $\mathbb{A}_R$  is deterministic,  $\delta_R(q, (\mathbf{a}, \Theta))$  contains at most one location.

Obviously,  $\mathbb{A}_{\star^c}$  satisfies all the size conditions in Lemma 8. The final construction of  $\mathbb{A}_{\star^c}$  is quite standard as it depends on a deterministic word automaton. We have  $\mathbf{t} \in L(\mathbb{A}_{\star^c})$  iff all the branches of  $\mathbf{t}$  are in  $L(\mathbb{A}_R)$ , which means precisely that  $\mathbf{t}$  satisfies the condition  $(\star^c)$ .  $\square$

#### 4.4 ExpTime upper bound

We are interested in the nonemptiness of  $L(\mathbb{B}_{\text{cons}(\mathbb{A})}) \cap L(\mathbb{A}_{\star^c})$  and we know that regular tree languages are closed under intersection. However, assuming that a Rabin tree automaton  $\mathbb{B}$  satisfies  $L(\mathbb{B}) = L(\mathbb{B}_{\text{cons}(\mathbb{A})}) \cap L(\mathbb{A}_{\star^c})$ , we need to guarantee that the construction of  $\mathbb{B}$  does not lead to any complexity blow up. This is the purpose of Lemma 9 below. An exponential blow-up may have drastic consequences on forthcoming complexity analysis. In the proof of Lemma 9, we propose a construction that is not polynomial but it only performs an exponential blow-up for the number of locations, which shall be fine for our purpose.

**Lemma 9.** *There is a Rabin tree automaton  $\mathbb{B}$  such that  $L(\mathbb{B}) = L(\mathbb{B}_{\text{cons}(\mathbb{A})}) \cap L(\mathbb{A}_{\star^c})$  and verifying the conditions below.*

- The number of Rabin pairs is polynomial in  $\beta$ .
- The number of locations is in  $\mathcal{O}(\text{card}(\text{SatTypes}(\beta)) \times \text{card}(Q) \times 2^{p(\beta)})$  for some polynomial  $p(\cdot)$ .
- The transition relation can be decided in polynomial-time in

$$\text{card}(\delta) + \beta + \text{card}(\Sigma) + D + \text{MaxConsSize}(\mathbb{A}).$$

*Proof.* The proof is divided in two parts. In Part (I), we present a construction for the intersection of Rabin *tree* automata mainly based on ideas from the proof of [Bok18, Theorem 1] on Rabin *word* automata but for trees (the developments in the proof of [Lab21, Lemma 3.13] are not satisfactory, to our opinion). In Part (II), we apply the general construction to  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  and  $\mathbb{A}_{\star^c}$  and perform a quantitative analysis.

(I) For  $i = 1, 2$ , let  $\mathbb{A}_i = (Q_i, \Sigma, D, Q_{i,\text{in}}, \delta_i, \mathcal{F}_i)$  with  $\mathcal{F}_i = (L_i^j, U_i^j)_{j \in [1, N_i]}$  ( $N_i$  Rabin pairs) be a Rabin tree automaton. Let us build a Rabin tree automaton  $\mathbb{A} = (Q, \Sigma, D, Q_{\text{in}}, \delta, \mathcal{F})$  such that  $L(\mathbb{A}) = L(\mathbb{A}_1) \cap L(\mathbb{A}_2)$ .

- $Q \stackrel{\text{def}}{=} Q_1 \times Q_2 \times [0, 3]^{[1, N_1] \times [1, N_2]}$ . The elements in  $Q$  are of the form  $(q_1, q_2, \mathbf{f})$  with  $\mathbf{f} : [1, N_1] \times [1, N_2] \rightarrow [0, 3]$ .
- The tuple  $((q_1, q_2, \mathbf{f}), \mathbf{a}, (q_1^0, q_2^0, \mathbf{f}^0), \dots, (q_1^{D-1}, q_2^{D-1}, \mathbf{f}^{D-1}))$  belongs to  $\delta$  iff the conditions below hold.
  1. For  $i = 1, 2$ , we have  $(q_i, \mathbf{a}, q_i^0, \dots, q_i^{D-1}) \in \delta_i$ . The two first components in elements from  $Q$  behave as in  $\mathbb{A}_1$  and  $\mathbb{A}_2$ , respectively.
  2. For all  $(i, j) \in [1, N_1] \times [1, N_2]$ , the following conditions hold.
    - (a) If  $\mathbf{f}(i, j)$  is odd, then for all  $k \in [0, D-1]$ , we have  $\mathbf{f}^k(i, j) = (\mathbf{f}(i, j) + 1) \bmod 4$ . Odd values in  $[0, 3]$  are unstable and are replaced at the next step by the successor value.
    - (b) For all  $k \in [0, D-1]$ , if  $\mathbf{f}(i, j) = 0$  and  $q_1^k \in L_1^i$ , then  $\mathbf{f}^k(i, j) = 1$ . Hence, when the  $(i, j)$ th component of  $\mathbf{f}$  is equal to 0, it waits to visit a state in the set  $L_1^i$  to move to 1.
    - (c) For all  $k \in [0, D-1]$ , if  $\mathbf{f}(i, j) = 0$  and  $q_1^k \notin L_1^i$ , then  $\mathbf{f}^k(i, j) = 0$  (not yet the right moment to modify the  $(i, j)$ th component).
    - (d) For all  $k \in [0, D-1]$ , if  $\mathbf{f}(i, j) = 2$  and  $q_2^k \in L_2^j$ , then  $\mathbf{f}^k(i, j) = 3$ . Hence, when the  $(i, j)$ th component of  $\mathbf{f}$  is equal to 2, it waits to visit a state in the set  $L_2^j$  to move to 3.
    - (e) For all  $k \in [0, D-1]$ , if  $\mathbf{f}(i, j) = 2$  and  $q_2^k \notin L_2^j$ , then  $\mathbf{f}^k(i, j) = 2$  (not yet the right moment to modify the  $(i, j)$ th component).

At this stage, it is worth noting that each  $\mathbf{f}^k$  for  $k \in [0, D-1]$  takes a unique value, i.e. the update of the third component in  $Q$  is done deterministically.

The transition relation  $\delta$  can be decided in the sum of the time-complexity to decide  $\delta_1$  and  $\delta_2$  respectively, plus polynomial-time in  $N_1 \times N_2$ .

- $Q_{\text{in}} \stackrel{\text{def}}{=} Q_{1,\text{in}} \times Q_{2,\text{in}} \times \{\mathbf{f}_0\}$ , where  $\mathbf{f}_0$  is the unique map that takes always the value zero.
- The set of Rabin pairs in  $\mathcal{F}$  contains exactly the pairs  $(L, U)$  for which there is  $(i, j) \in [1, N_1] \times [1, N_2]$  such that

$$U \stackrel{\text{def}}{=} (U_1^i \times Q_2 \cup Q_1 \times U_2^j) \times [0, 3]^{[1, N_1] \times [1, N_2]} \quad L \stackrel{\text{def}}{=} L_1^i \times Q_2 \times \{\mathbf{f} \mid \mathbf{f}(i, j) = 1\}$$

Because the odd values are unstable, if a location in  $L$  is visited infinitely along a branch of a run for  $\mathbb{A}$  (and therefore a location in  $L_1^i$  is visited infinitely often on the first component), then a location in  $L_2^j$  is also visited infinitely often on the second component. Indeed, to revisit the value 1 on the

$(i, j)$ th component one needs to visit first the value 3, which witnesses that a location in  $L_2^j$  has been found.

If along a branch of a run for  $\mathbb{A}$  the triples in  $U$  are visited finitely, then a location in  $U_1^i$  is visited finitely on the first component and a location in  $U_2^j$  is visited finitely on the second component.

Consequently,  $\mathcal{F}$  contains at most  $N_1 \times N_2$  pairs.

We claim that  $L(\mathbb{A}) = L(\mathbb{A}_1) \cap L(\mathbb{A}_2)$ . We omit the proof here as the forthcoming proof of Lemma 21 generalises it.

(II) Let us analyse the size of the components in  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  and  $\mathbb{A}_{\star^c}$ , which provides bounds for  $\mathbb{B}$  such that  $L(\mathbb{B}) = L(\mathbb{A}_{\star^c}) \cap L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  following the above construction. The automaton  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  can be viewed as a Rabin tree automaton with a single pair, typically  $(F', \emptyset)$ , where  $F'$  is the set of accepting states of the Büchi tree automaton  $\mathbb{B}_{\text{cons}(\mathbb{A})}$ .

- $\mathbb{B}_{\text{cons}(\mathbb{A})}$  has a single Rabin pair,  $\mathbb{A}_{\star^c}$  has a number of Rabin pairs bounded by  $8(\beta + 2)^2 + 3$ , so  $\mathbb{B}$  has a number of Rabin pairs bounded by  $8(\beta + 2)^2 + 3$ .
- The locations in  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  are from  $\text{SatTypes}(\beta) \times Q$ , the number of locations in  $\mathbb{A}_{\star^c}$  is in  $\mathcal{O}(2^{p^*(\beta)})$ . Therefore, based on the above construction for intersection, the number of locations in  $\mathbb{B}$  is in

$$\mathcal{O}(\text{card}(\text{SatTypes}(\beta) \times Q) \times 2^{p^*(\beta)} \times 4^{8(\beta+2)^2+3}),$$

which is in  $\mathcal{O}(\text{card}(\text{SatTypes}(\beta)) \times \text{card}(Q) \times 2^{p(\beta)})$  for some polynomial  $p(\cdot)$ .

- The transition relation for  $\mathbb{A}_{\star^c}$  can be decided in polynomial-time in

$$\max(\lceil \log(|\mathfrak{d}_1|) \rceil, \lceil \log(|\mathfrak{d}_\alpha|) \rceil) + \beta + \text{card}(\Sigma) + D$$

(by Lemma 8), the transition relation for  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  can be decided in polynomial-time in  $\text{card}(\delta) + \beta + \text{card}(\Sigma) + D + \text{MaxConsSize}(\mathbb{A})$ . Moreover, the product of the number of Rabin pairs is polynomial in  $\beta$ . Therefore, the transition relation can be decided in polynomial-time in  $(\text{card}(\delta) + \beta + \text{card}(\Sigma) + D + \text{MaxConsSize}(\mathbb{A}))$ .  $\square$

Nonemptiness of Rabin tree automata is polynomial in the cardinality of the transition relation and exponential in the number of Rabin pairs, see e.g. [EJ00, Theorem 4.1]. More precisely, it is in time  $(m \times n)^{\mathcal{O}(n)}$ , where  $m$  is the number of locations and  $n$  is the number of Rabin pairs, see the statement [EJ00, Theorem 4.1]. However, this is not exactly what we need herein as the complexity expression above concerns binary trees only and it assumes that the transition relation  $\delta$  can be decided in constant time. When the degree is  $D \geq 1$  and deciding whether a tuple  $\mathfrak{t}$  belongs to  $\delta$  requires  $\gamma$  time units, by scrutiny of the proof of [EJ00, Theorem 4.1] (page 144 more precisely), the complexity for checking nonemptiness is actually in  $(\text{card}(\delta) \times \gamma \times n)^{\mathcal{O}(n)}$ . Here,  $\gamma$  may depend on parameters related to  $\mathbb{A}$  and in Lemma 10 below,  $\gamma$  takes the value

$$\text{card}(\delta) + \beta + \text{card}(\Sigma) + D + \text{MaxConsSize}(\mathbb{A}).$$

**Lemma 10.** *The nonemptiness problem for constraint tree automata can be solved in time in*

$$\mathcal{O}\left(q_1(\text{card}(Q) \times \text{card}(\delta) \times \text{MaxConsSize}(\mathbb{A}) \times \text{card}(\Sigma) \times q_2(\beta))^{q_2(\beta) \times q_3(D)}\right)$$

for some polynomials  $q_1$ ,  $q_2$  and  $q_3$ .

The above bound is roughly

- the size of the product alphabet  $\Sigma \times \text{SatTypes}(\beta)$  multiplied by  $\text{card}(Q^\dagger)^{D+1}$  (upper bound for the cardinality of the transition relation),
- by the number of Rabin pairs of  $\mathbb{B}$  (which is polynomial in  $\beta$  by Lemma 9),
- and by  $(\text{card}(\delta) + \beta + \text{card}(\Sigma) + D + \text{MaxConsSize}(\mathbb{A}))$  (time to decide whether a tuple belongs to the transition relation)

and then put to the power the number of Rabin pairs of  $\mathbb{B}$ . Herein,  $Q^\dagger$  is the set of locations of the product between  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  and  $\mathbb{A}_{\star^c}$  and  $\text{card}(Q^\dagger)$  is in  $\mathcal{O}(\text{card}(\text{SatTypes}(\beta)) \times \text{card}(Q) \times 2^{p(\beta)})$  (see Lemma 9). Furthermore, we recall that  $\text{card}(\text{SatTypes}(\beta)) \leq ((\mathfrak{d}_\alpha - \mathfrak{d}_1) + 3)^{2\beta} \times 3^{\beta^2}$ . Assuming that the size of the TCA  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$ , written  $\text{size}(\mathbb{A})$ , is polynomial in

$$\text{card}(Q) + \text{card}(\delta) + D + \beta + \text{MaxConsSize}(\mathbb{A})$$

(which makes sense for a reasonably succinct encoding), from the computation of the bound in Lemma 10, the nonemptiness of  $L(\mathbb{A})$  can be checked in time  $\mathcal{O}(q(\text{size}(\mathbb{A}))^{q'(\beta+D)})$  for some polynomials  $q$  and  $q'$ .

Theorem 2 is one of the main results of the paper.

**Theorem 2.** *The nonemptiness problem for tree constraint automata is EXPTIME-complete.*

The EXPTIME upper bound is a direct consequence of the above complexity expression and using the fact that the transitions in the product Rabin tree automaton between  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  and  $\mathbb{A}_{\star^c}$  can be decided in polynomial-time. We have seen that the EXPTIME-hardness holds as soon as  $D = 2$ .

**Theorem 3.** *For the fixed degree  $D = 1$ , the nonemptiness problem for word constraint automata is PSPACE-complete.*

*Proof.* (sketch) PSPACE-hardness is obtained similarly to what is done in Section 4.1 by reduction from the halting problem for deterministic Turing machines running in polynomial space. Concerning the PSPACE-easiness, we need to check the nonemptiness of  $L(\mathbb{A}_R) \cap L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  with the Rabin word automaton  $\mathbb{A}_R$  from the proof of Lemma 8 and  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  is already a Büchi automaton. Not only  $\mathbb{A}_R$  can be transformed into an equivalent Büchi automaton  $\mathbb{B}$  with a polynomial increase of the number of locations (because the number of Rabin pairs is bounded by  $8(\beta + 2)^2 + 3$ ), but the nonemptiness of the product Büchi automaton between  $\mathbb{B}$  and  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  can be performed in PSPACE. Indeed, the number of locations of the product is only exponential in the size of the input constraint automaton and the transition relation can be also decided in polynomial space.  $\square$

The Rabin word automaton  $\mathbb{A}_R$  captures therefore the condition  $C_{\mathbb{Z}}$  from [DD07, Section 6] (see also condition  $\mathcal{C}$  in [DG08, Definition 2]) and can be turned in polynomial-time into a nondeterministic Büchi automata, leading to the PSPACE upper bound for LTL( $\mathbb{Z}$ ) (the linear-time temporal logic LTL with arithmetical constraints from the concrete domain  $\mathbb{Z}$ , see Section 6.1), proposing therefore an alternative proof to [DG08, Theorem 1] and [ST11, Theorem 16] for the concrete domain  $\mathbb{Z}$ .

#### 4.5 Rabin Tree Constraint Automata

In this section, we introduce a variant of TCA defined in Section 3.1 by simply considering Rabin acceptance condition. We show that the nonemptiness problem is still in EXPTIME, and this is mainly useful to characterize the complexity of  $\text{SAT}(\text{CTL}^*(\mathbb{Z}))$ .

A *Rabin tree constraint automaton*  $\mathbb{A}$  (RTCA, for short) is a tuple of the form  $(Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, \mathcal{F})$  defined as for TCA (with Büchi acceptance condition) except that  $\mathcal{F}$  is a set of pairs of the form  $(L, U)$ , where  $L, U \subseteq Q$ . All the definitions about TCA apply except that a run  $\rho : [0, D-1]^* \rightarrow Q$  is *accepting* iff for all paths  $\pi$  in  $\rho$  starting from  $\varepsilon$ , there is some  $(L, U) \in \mathcal{F}$  such that  $\inf(\rho, \pi) \cap U = \emptyset$  and  $\inf(\rho, \pi) \cap L \neq \emptyset$ .

Given an RTCA  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, \mathcal{F})$ , the definition of symbolic trees respecting  $\mathbb{A}$  is updated so that it uses the acceptance condition  $\mathcal{F}$ . From the RTCA  $\mathbb{A}$ , we can define a Rabin tree automaton  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  (instead of a Büchi tree automaton with a TCA) such that the acceptance  $\mathcal{F}'$  is equal to

$$\{(\text{SatTypes}(\beta) \times U, \text{SatTypes}(\beta) \times L) \mid (L, U) \in \mathcal{F}\}.$$

Similarly to Lemma 6, one can show that  $L(\mathbb{A}) \neq \emptyset$  iff there is a symbolic tree  $\mathbf{t} \in L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  that is satisfiable. Moreover, we can take advantage of  $\mathbb{A}_{\star^c}$  (the same as in the proof of Lemma 8) so that  $L(\mathbb{A}) \neq \emptyset$  iff  $L(\mathbb{B}_{\text{cons}(\mathbb{A})}) \cap L(\mathbb{A}_{\star^c})$  is non-empty. It remains to determine how much it costs to test non-emptiness of  $L(\mathbb{B}_{\text{cons}(\mathbb{A})}) \cap L(\mathbb{A}_{\star^c})$ . We might expect a complexity jump compared to the case with TCA (but this is not the case). Indeed, the nonemptiness problem for Büchi tree automata is in PTIME [VW86, Theorem 2.2] whereas it is NP-complete for Rabin tree automata [EJ00, Theorem 4.10]. Here is the result that provide quantitative analysis about components of  $\mathbb{A}$ , which is a variant of Lemma 9, more particularly by considering the value  $\text{card}(\mathcal{F})$  in the analysis. For TCA with Büchi acceptance condition, this value was equal to one.

**Lemma 11.** *There is a Rabin tree automaton  $\mathbb{B}$  such that  $L(\mathbb{B}) = L(\mathbb{B}_{\text{cons}(\mathbb{A})}) \cap L(\mathbb{A}_{\star^c})$  and verifying the conditions below.*

- The number of Rabin pairs is polynomial in  $\beta + \text{card}(\mathcal{F})$ , where  $\text{card}(\mathcal{F})$  is the number of Rabin pairs in  $\mathbb{A}$ .
- The number of locations is in  $\mathcal{O}(\text{card}(\text{SatTypes}(\beta)) \times \text{card}(Q) \times 2^{p(\beta + \text{card}(\mathcal{F}))})$  for some polynomial  $p(\cdot)$ .
- the transition relation can be decided in polynomial-time in  $\text{card}(\delta) + \beta + \text{card}(\Sigma) + D + \text{MaxConsSize}(\mathbb{A})$ .

The proof of Lemma 11 is similar to the proof of Lemma 9. Moreover, as for Lemma 10, we can conclude that the nonemptiness problem for Rabin tree constraint automata can be solved in time in

$$\mathcal{O}\left(q_1(\text{card}(Q) \times \text{card}(\delta) \times \text{MaxConsSize}(\mathbb{A}) \times \text{card}(\Sigma) \times q_2(\beta + \text{card}(\mathcal{F})))^{q_2(\beta + \text{card}(\mathcal{F})) \times q_3(D)}\right)$$

for some polynomials  $q_1$ ,  $q_2$  and  $q_3$ . Theorem 4 is one of the main results of the paper.

**Theorem 4.** *The nonemptiness problem for Rabin tree constraint automata is EXPTIME-complete.*

## 5 Complexity for $\text{SAT}(\text{CTL}(\mathbb{Z}))$ and $\text{TSAT}(\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c))$

In this section, we provide new complexity characterisations about the satisfiability problem for  $\text{CTL}(\mathbb{Z})$  and related logics.

### 5.1 Satisfiability problem for $\text{CTL}(\mathbb{Z})$ and variants

Theorem 5 below is one of the main results of the paper. In particular, it witnesses that enriching the CTL models with numerical values interpreted in the concrete domain  $\mathbb{Z}$  does not cause a complexity blow-up.

**Theorem 5.** *The satisfiability problem for  $\text{CTL}(\mathbb{Z})$  is EXPTIME-complete.*

*Proof.* EXPTIME-hardness is inherited from CTL. Concerning the EXPTIME upper bound, let  $\phi$  be a  $\text{CTL}(\mathbb{Z})$  formula in simple form built over constants  $\mathfrak{d}_1 < \dots < \mathfrak{d}_\alpha$ . Let us show that the satisfiability status for  $\phi$  can be checked in EXPTIME, which is sufficient for our needs by Proposition 2.

From the developments in Section 3.2, there is a TCA  $\mathbb{A}_\phi$  such that  $\phi$  is satisfiable iff  $L(\mathbb{A}_\phi) \neq \emptyset$  and  $\mathbb{A}_\phi$  satisfies the following quantitative properties.

- The degree  $D$  is bounded above by  $\text{size}(\phi)$ .
- The number of locations is bounded by  $(D \times 2^{\text{size}(\phi)}) \times (\text{size}(\phi) + 1)$ .
- The number of transitions is in  $\mathcal{O}(2^{p(\text{size}(\phi))})$  for some polynomial  $p(\cdot)$ .
- The number of variables  $\beta$  is bounded by  $\text{size}(\phi)$ .
- The finite alphabet  $\Sigma$  in  $\mathbb{A}_\phi$  is unary.
- $\text{MaxConsSize}(\mathbb{A}_\phi)$  is quadratic in  $\text{size}(\phi)$ .

We have also seen that the nonemptiness problem for TCA can be solved in time

$$\mathcal{O}\left(q_1(\text{card}(Q) \times \text{card}(\delta) \times \text{MaxConsSize}(\mathbb{A}) \times \text{card}(\Sigma) \times q_2(\beta))^{q_2(\beta) \times q_3(D)}\right)$$

(Lemma 10). Since the transition relations of the automata  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  and  $\mathbb{A}_{\star^c}$  can be also built in polynomial-time, we get that nonemptiness of  $L(\mathbb{A}_\phi)$  can be solved in exponential-time.

Let  $\mathbb{N}$  be the concrete domain  $(\mathbb{N}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{N}})$  for which we explained in Section 3.1 that nonemptiness of TCA with constraints interpreted on  $\mathbb{N}$  has the same complexity as for TCA with constraints interpreted on  $\mathbb{Z}$ . Let  $\text{CTL}(\mathbb{N})$  be the variant of  $\text{CTL}(\mathbb{Z})$  with constraints interpreted on  $\mathbb{N}$ . We get the following corollary.

**Theorem 6.** *The satisfiability problem for  $\text{CTL}(\mathbb{N})$  is EXPTIME-complete.*

With the concrete domain  $(\mathbb{Q}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{Q}})$ , all the trees in  $L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  are satisfiable (no need to intersect  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  with a hypothetical  $\mathbb{A}_{\star^c}$ , a property already observed many times, see e.g. [Lut01, BC02, DD07, Gas09]), and therefore the satisfiability problem for  $\text{CTL}(\mathbb{Q})$  is also in EXPTIME.

### 5.2 Satisfiability for the description logic $\mathcal{ALCF}^P(\mathbb{Z}_c)$

In this section, we illustrate how the satisfiability problem w.r.t. a TBox for the description logic  $\mathcal{ALCF}^P(\mathbb{Z}_c)$  (written  $\text{TSAT}(\mathcal{ALCF}^P(\mathbb{Z}_c))$ ) can be reduced to the nonemptiness problem for tree constraint automata, along the lines of what is done for  $\text{CTL}(\mathbb{Z})$ . This leads us to the EXPTIME-easiness for  $\text{TSAT}(\mathcal{ALCF}^P(\mathbb{Z}_c))$ , a result already considered in [LOS20]. At this point, it is worth noting that the reduction from  $\text{TSAT}(\mathcal{ALCF}^P(\mathbb{Z}_c))$  to nonemptiness of Rabin tree automata in [LOS20] is done for tree automata over *finite* alphabets. As far as we know, the reduction below is the first one that involves tree automata with constraints to decide a description logic with concrete domain. Below, this does not

prevent us from using features from the automata-based technique for description logics, see e.g. [Baa09, Section 3.2], but tailored to the semantics of  $\mathcal{ALCF}^P(\mathbb{Z}_c)$  and to the presence of constraints in TCA.

Formal definitions about the well-known description logic  $\mathcal{ALCF}^P(\mathbb{Z}_c)$  can be found in [LOS20, Lab21]. Below, we briefly present  $\mathcal{ALCF}^P(\mathbb{Z}_c)$  in the style of a temporal logic because we do not wish to introduce too many new definitions and notations, but we believe the correspondence between the two presentations can be done, along the lines of the seminal work [Sch91].

Given a set  $\mathbf{N}_R = \{r, s, \dots\}$  of *role names* and a set  $\text{PROP}$  of *propositional variables*, the notion of Kripke structure from Section 2 is generalised to structures of the form  $(\mathcal{W}, (\mathcal{R}_r)_{r \in \mathbf{N}_R}, l, \mathbf{v})$ , where  $l : \mathcal{W} \rightarrow \mathcal{P}(\text{PROP})$ . Instead of having a single accessibility relation, the Kripke structures admit a family of accessibility relations indexed by role names  $r \in \mathbf{N}_R$ . We assume that  $\mathbf{N}_R$  contains a subfamily  $\mathbf{N}_F \subseteq \mathbf{N}_R$  such that for all  $f \in \mathbf{N}_F$ , we require that  $\mathcal{R}_f$  in Kripke structures is deterministic ( $((u, v) \in \mathcal{R}_f$  and  $(u, v') \in \mathcal{R}_f$  imply  $v = v'$ ). Elements in  $\mathbf{N}_F$  are called *functional role names*.

A *role path*  $P = r_1 \dots r_n$  is a (possibly empty) word in  $\mathbf{N}_R^*$ . The set of  $\mathcal{ALCF}^P(\mathbb{Z}_c)$ -formulae is defined as follows.

$$\phi ::= p \mid EP \Theta \mid AP \Theta \mid \neg \phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid EX_r \phi \mid AX_r \phi,$$

where  $p \in \text{PROP}$ ,  $P$  is a role path,  $\Theta$  is a constraint in  $\mathbb{Z}$  built over terms of the form  $X^j \mathbf{x}$ . Moreover, if  $\mathbf{t}_i = X^j \mathbf{x}$  occurs in  $EP \Theta$  or  $AP \Theta$ , then the length of  $P$  is at least  $j$ . Here is an example of an  $\mathcal{ALCF}^P(\mathbb{Z}_c)$ -formula:

$$E r_1 r_2 r_3 (\mathbf{x}_1 < X \mathbf{x}_2) \wedge \neg (XX \mathbf{x}_1 = 3).$$

The satisfaction relation is defined as follows (obvious clauses are omitted). Given a finite path  $\pi = w_0 \mathcal{R}_{r_1} w_1 \mathcal{R}_{r_2} \dots \mathcal{R}_{r_n} w_n$ , let us define  $\mathbf{v}(\pi, X^j \mathbf{x}) \stackrel{\text{def}}{=} \mathbf{v}(w_j, \mathbf{x})$ .

- $\mathcal{K}, w \models p \stackrel{\text{def}}{\iff} p \in l(w)$ ,
- $\mathcal{K}, w \models EX_r \phi \stackrel{\text{def}}{\iff}$  there is  $w' \in \mathcal{R}_r(w)$  such that  $\mathcal{K}, w' \models \phi$ ,
- $\mathcal{K}, w \models AX_r \phi \stackrel{\text{def}}{\iff}$  for all  $w' \in \mathcal{R}_r(w)$ , we have  $\mathcal{K}, w' \models \phi$ ,
- $\mathcal{K}, w \models E r_1 \dots r_n \Theta(\mathbf{t}_1, \dots, \mathbf{t}_d) \stackrel{\text{def}}{\iff}$  there is a path  $w_0 \mathcal{R}_{r_1} \dots \mathcal{R}_{r_n} w_n$  with  $w = w_0$  such that  $\mathbb{Z} \models \Theta(\mathbf{v}(\pi, \mathbf{t}_1), \dots, \mathbf{v}(\pi, \mathbf{t}_d))$ .
- $\mathcal{K}, w \models A r_1 \dots r_n \Theta(\mathbf{t}_1, \dots, \mathbf{t}_d) \stackrel{\text{def}}{\iff}$  for all finite paths with  $w_0 \mathcal{R}_{r_1} \dots \mathcal{R}_{r_n} w_n$   $w = w_0$ , we have  $\mathbb{Z} \models \Theta(\mathbf{v}(\pi, \mathbf{t}_1), \dots, \mathbf{v}(\pi, \mathbf{t}_d))$ .

An *axiom* is an expression of the form  $\phi \sqsubseteq \psi$ , where  $\phi, \psi$  are  $\mathcal{ALCF}^P(\mathbb{Z}_c)$  formulae. A *terminological box (TBox, for short)* is a finite set of axioms. We say that  $\mathcal{K}$  satisfies  $\phi \sqsubseteq \psi$  (written  $\mathcal{K} \models \phi \sqsubseteq \psi$ ) iff for every  $w \in \mathcal{W}$ ,  $\mathcal{K}, w \models \phi$  implies  $\mathcal{K}, w \models \psi$ ; this generalises to TBoxes in the expected way. The *satisfiability problem w.r.t. a TBox*, written  $\text{TSAT}(\mathcal{ALCF}^P(\mathbb{Z}_c))$ , is defined as follows.

**Input:** An  $\mathcal{ALCF}^P(\mathbb{Z}_c)$ -formula  $\phi$  and a finite TBox  $T$ ,

**Question:** Are there a Kripke structure  $\mathcal{K}$  and  $w \in \mathcal{W}$  such that  $\mathcal{K}, w \models \phi$  and  $\mathcal{K} \models T$ ?

Below, we reduce  $\text{TSAT}(\mathcal{ALCF}^P(\mathbb{Z}_c))$  to the nonemptiness problem for TCA assuming that all the input formulae are in negation normal form and terms are restricted to those in  $\text{T}_{\text{VAR}}^{\leq 1}$ . This means that the role paths are restricted to single role names  $r$  and to  $\varepsilon$ . Note that  $A\varepsilon \Theta$  is logically equivalent to  $E\varepsilon \Theta$ , and  $\Theta$  contains only variables, which states constraints only for the current world. This definitely simplifies the reduction but it is known that there is a polynomial-time reduction from  $\text{TSAT}(\mathcal{ALCF}^P(\mathbb{Z}_c))$  to this subproblem [LOS20, Lemma 5]. This is reminiscent to the formulae in simple form for  $\text{CTL}(\mathbb{Z})$ , see e.g. Proposition 2.

Unlike  $\text{CTL}(\mathbb{Z})$ , the models admit several accessibility relations and this needs to be taken into account for the reduction to TCA. We use a standard trick and reserve directions in  $[0, D-1]$  for each role name  $r$  occurring in the instance of  $\text{TSAT}(\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c))$ . This is needed because in the tree models  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$ , the edges are not labelled. (Another way to proceed would be to add a role name to each location of the TCA in order to remember how the node in the tree  $[0, D-1]^*$  is accessed to, which is a technique used in [Baa09, Section 3.2] in the presentation of the main features of the automata-based technique for description logics (without data values).) Another difference with  $\text{CTL}(\mathbb{Z})$  is related to the determinism of the relations  $\mathcal{R}_r$  with  $r \in \mathbf{N}_{\mathbf{F}}$ . To handle this new feature, we have to relax the notion of direction map, in particular to require  $\iota(\text{EX}_r \psi_1) = \iota(\text{EX}_r \psi_2)$  with  $r \in \mathbf{N}_{\mathbf{F}}$ , which modifies the definition of a direction map  $\iota$ . The details follow below. Finally, the models in  $\text{CTL}(\mathbb{Z})$  have total transition relations whereas the relations  $\mathcal{R}_r$  in  $\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c)$  models are not necessarily so. We take care of this point by adding a “dead-end” location  $\perp$  in the forthcoming constructed TCA, similarly to what is done in [Baa09, Section 3.2] (with the location  $\emptyset$ ).

Let  $T$  be a TBox  $\{\phi_1 \sqsubseteq \psi_1, \dots, \phi_\ell \sqsubseteq \psi_\ell\}$  and  $\phi_0$  be an  $\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c)$ -formula as an instance of  $\text{TSAT}(\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c))$  but with the above-mentioned syntactic restrictions. We write  $\text{sub}(T, \phi_0)$  to denote the set of subformulae obtained from formulae in  $T, \phi_0$ . Given  $X \subseteq \text{sub}(T, \phi_0)$ , we say that  $X$  is *propositionally  $T$ -consistent* iff the conditions below hold.

- there is no propositional variable  $p$  such that  $\{p, \neg p\} \subseteq X$ ,
- If  $\varphi_1 \vee \varphi_2 \in X$ , then  $\{\varphi_1, \varphi_2\} \cap X \neq \emptyset$ ; if  $\varphi_1 \wedge \varphi_2 \in X$ , then  $\{\varphi_1, \varphi_2\} \subseteq X$ .
- For all  $k \in [1, \ell]$ , if  $\phi_k \in X$ , then  $\psi_k \in X$ .

Propositionally  $T$ -consistent sets correspond to Hintikka sets from [Baa09, Section 3.2] and their introduction are common for developing an automata-based approach for description logics.

Given  $r \in \mathbf{N}_{\mathbf{R}}$ , we write  $\text{sub}_{\text{EX}_r}(T, \phi_0)$  (resp.  $\text{sub}_{\text{AX}_r}(T, \phi_0)$ ) to denote the set of formulae in  $\text{sub}(T, \phi_0)$  of the form  $\text{EX}_r \psi$  (resp.  $\text{AX}_r \psi$ ). Finally, we write  $\text{sub}_{\text{Er}}(T, \phi_0)$  to denote the set of formulae of the form  $\text{Er } \Theta$ . We introduce the set  $\text{sub}_{\text{Ar}}(T, \phi_0)$  with a similar definition.

We write  $\mathbf{N}_{\mathbf{F}}(T, \phi_0)$  to denote the set of functional role names  $r$  such that  $\text{sub}_{\text{EX}_r}(T, \phi_0) \cup \text{sub}_{\text{Er}}(T, \phi_0)$  is non-empty. Moreover, we write  $\mathbf{E}_{\mathbf{F}}(T, \phi_0)$  to denote the set below

$$\{\text{Er } \Theta \in \text{sub}(T, \phi_0) \mid r \notin \mathbf{N}_{\mathbf{F}}(T, \phi_0)\} \cup \{\text{EX}_r \psi \in \text{sub}(T, \phi_0) \mid r \notin \mathbf{N}_{\mathbf{F}}(T, \phi_0)\}.$$

Set  $D = \text{card}(\mathbf{N}_{\mathbf{F}}(T, \phi_0)) + \text{card}(\mathbf{E}_{\mathbf{F}}(T, \phi_0))$  and  $\iota$  be a bijection  $\iota : (\mathbf{N}_{\mathbf{F}}(T, \phi_0) \cup \mathbf{E}_{\mathbf{F}}(T, \phi_0)) \rightarrow [1, D]$ . We write  $r \triangleright j$  whenever  $\iota^{-1}(j) = r$  or  $\iota^{-1}(j)$  is of the form either  $\text{EX}_r \psi$  or  $\text{Er } \Theta$  (same role name  $r$ ).

We build a TCA  $\mathbb{A} = (Q, \Sigma, D+1, \beta, Q_{\text{in}}, \delta, F)$  such that  $T, \phi_0$  is a positive instance iff  $L(\mathbb{A}) \neq \emptyset$ . The automaton  $\mathbb{A}$  accepts infinite trees of the form  $\mathbf{t} : [0, D]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$  where  $\Sigma = \mathcal{P}(\{p_1, \dots, p_M\})$ ,  $\{p_1, \dots, p_M\}$  being the set of propositional variables occurring in  $T, \phi_0$  (here, we really take advantage of having a finite alphabet  $\Sigma$ ). Let us define  $\mathbb{A}$  formally.

- $Q$  is the set of propositionally  $T$ -consistent subsets of  $\text{sub}(T, \phi_0)$  plus the “dead-end” location  $\perp$  (and never  $\psi \in \perp$ , for some formula  $\psi$ ).
- $Q_{\text{in}} \stackrel{\text{def}}{=} \{Y \in Q \mid \phi_0 \in Y\}$ ,  $F \stackrel{\text{def}}{=} Q$ .
- The transition relation  $\delta$  is made of tuples  $(Y, X, (\Theta_0, Y_0), \dots, (\Theta_D, Y_D))$  verifying the conditions below.
  1. For all  $p \in Y$ , we have  $p \in X$  and for all  $\neg p \in Y$ , we have  $p \notin X$ .
  2. If  $Y = \perp$ , then  $Y_0 = \dots = Y_D = \perp$ .
  3. For all  $j \in [1, D]$  such that  $Y_j = \perp$ , (a) if  $\iota^{-1}(j) = r$  for some  $r \in \mathbf{N}_{\mathbf{F}}$ , then  $Y$  has no formulae of the form either  $\text{EX}_r \psi$  or  $\text{Er } \Theta$  and (b) if  $\iota^{-1}(j) \notin \mathbf{N}_{\mathbf{F}}$  then  $\iota^{-1}(j) \notin Y$ .



4. For all  $\text{EX}_r\psi \in Y$ , we have either  $(r \in \mathbf{N}_{\mathbf{F}}$  and  $\psi \in Y_{\iota(r)})$  or  $(r \notin \mathbf{N}_{\mathbf{F}}$  and  $\psi \in Y_{\iota(\text{EX}_r\psi)})$ .
5. For all  $\text{AX}_r\psi \in Y$  and  $j \in [1, D]$  such that  $Y_j \neq \perp$  and  $r \triangleright j$ , we have  $\psi \in Y_j$ . In this case, the direction  $j$  is reserved for the role name  $r$  and for obligations related to the satisfaction of  $\text{AX}_r\psi$ . The satisfaction of  $\text{AX}_r\psi$  implies the satisfaction of  $\psi$  for all the  $\mathcal{R}_r$ -successors, if any.
6. For all  $j \in [0, D]$ , the constraint  $\Theta_j$  is defined as follows.
  - (a) If  $Y = \perp$ , then  $\Theta_j = \top$ .
  - (b) Otherwise,  $j = 0$  or  $Y_j = \perp$ , then  $\Theta_j \stackrel{\text{def}}{=} (\bigwedge_{\text{E}\varepsilon \ \Theta', \text{A}\varepsilon \ \Theta' \in Y} \Theta') \wedge (\bigwedge_{\text{A}r \ \Theta' \in Y} \Theta')$ .
  - (c) Otherwise, if  $((\text{Er} \ \Theta \notin Y \text{ and } \iota(\text{Er} \ \Theta) = j) \text{ or } \iota(\text{EX}_r\psi) = j \text{ for some } \text{Er} \ \Theta, \text{EX}_r \ \psi \in \mathbf{E}_{\mathbf{F}}(T, \phi_0))$  and  $Y_j \neq \perp$  (necessarily  $r \notin \mathbf{N}_{\mathbf{F}}$ ), then

$$\Theta_j \stackrel{\text{def}}{=} (\bigwedge_{\text{E}\varepsilon \ \Theta', \text{A}\varepsilon \ \Theta' \in Y} \Theta') \wedge (\bigwedge_{\text{A}r \ \Theta' \in Y} \Theta').$$

- (d) Otherwise, if there is  $\text{Er} \ \Theta \in Y$  such that  $\iota(\text{Er} \ \Theta) = j$  (necessarily  $r \notin \mathbf{N}_{\mathbf{F}}$ ),

$$\Theta_j \stackrel{\text{def}}{=} (\bigwedge_{\text{E}\varepsilon \ \Theta', \text{A}\varepsilon \ \Theta' \in Y} \Theta') \wedge (\bigwedge_{\text{A}r \ \Theta' \in Y} \Theta') \wedge \Theta.$$

Note that by (3.) above,  $\text{Er} \ \Theta \in Y$ ,  $\iota(\text{Er} \ \Theta) = j$  and  $r \notin \mathbf{N}_{\mathbf{F}}$  imply that  $Y_j \neq \perp$ .

In short, this means that the direction  $j$  is reserved for the role name  $r$  and for obligations related to the satisfaction of  $\text{Er} \ \Theta$ . The current case occurs when there are obligations to satisfy  $\text{Er} \ \Theta$  whereas the case 6(b) gives more freedom because the satisfaction of  $\text{Er} \ \Theta$  is not imposed from  $Y$ .

- (e) Otherwise, i.e.  $r = \iota^{-1}(j) \in \mathbf{N}_{\mathbf{F}}$  and  $Y_j \neq \perp$ ,

$$\Theta_j \stackrel{\text{def}}{=} (\bigwedge_{\text{E}\varepsilon \ \Theta', \text{A}\varepsilon \ \Theta' \in Y} \Theta') \wedge (\bigwedge_{\text{A}r \ \Theta', \text{Er} \ \Theta' \in Y} \Theta').$$

The difference with the previous cases is due to the fact that, if  $r \in \mathbf{N}_{\mathbf{F}}$ , then  $\text{Ar} \ \Theta'$  and  $\text{Er} \ \Theta'$  are logically equivalent, assuming that there is one  $\mathcal{R}_r$ -successor.

As shown in Section 3 for  $\text{CTL}(\mathbb{Z})$ , we can show that our construction is correct.

**Lemma 12.**  $T, \phi_0$  is a positive instance of  $\text{TSAT}(\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c))$  iff  $\mathbf{L}(\mathbb{A}) \neq \emptyset$ .

*Proof.* “if”: Suppose  $\mathbf{L}(\mathbb{A}) \neq \emptyset$ . Then there exists some tree  $\mathbf{t} : [0, D]^* \rightarrow (\Sigma \times \mathbb{Z}^{\beta})$  in  $\mathbf{L}(\mathbb{A})$ . Let  $\rho : [0, D]^* \rightarrow Q$  be an accepting run of  $\mathbb{A}$  on  $\mathbf{t}$ .

From the tree  $\mathbf{t}$  and its accepting run  $\rho$ , we build an  $\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c)$ -model  $\mathcal{K} = (\mathcal{W}, (\mathcal{R}_r)_{r \in \mathbf{N}_{\mathbf{R}}}, l, \mathbf{v})$  that preserves the tree structure of  $\mathbf{t}$  while defining the relations  $\mathcal{R}_r$  using the map  $\iota$ . Let us formally define  $\mathcal{K}$ .

- (a)  $\mathcal{W} \stackrel{\text{def}}{=} [0, D]^*$ .
- (b) For all  $r \in \mathbf{N}_{\mathbf{F}}(T, \phi_0)$ , for all  $\mathbf{n}, \mathbf{n}' \in [0, D]^*$ ,  $(\mathbf{n}, \mathbf{n}') \in \mathcal{R}_r \stackrel{\text{def}}{\iff} \mathbf{n}' = \mathbf{n} \cdot \iota(r)$  and both  $\rho(\mathbf{n})$  and  $\rho(\mathbf{n}')$  are distinct from  $\perp$ .
- (c) For all non-functional  $r$ , such that  $\text{Er} \ \Theta$  occurs in  $\mathbf{E}_{\mathbf{F}}(T, \phi_0)$  or  $\text{EX}_r \ \psi$  occurs in  $\mathbf{E}_{\mathbf{F}}(T, \phi_0)$ , for all  $\mathbf{n}, \mathbf{n}' \in [0, D]^*$ ,  $(\mathbf{n}, \mathbf{n}') \in \mathcal{R}_r \stackrel{\text{def}}{\iff}$  for some  $j$ ,  $\mathbf{n}' = \mathbf{n} \cdot j$ ,  $r \triangleright j$  and both  $\rho(\mathbf{n})$  and  $\rho(\mathbf{n}')$  are distinct from  $\perp$ .
- (d) For all other role names  $r$ ,  $\mathcal{R}_r \stackrel{\text{def}}{=} \emptyset$ .

- (e) For all  $\mathbf{n} \in [0, D]^*$ , for all  $i \in [1, \beta]$ ,  $\mathbf{v}(\mathbf{n}, \mathbf{x}_i) \stackrel{\text{def}}{=} \mathbf{z}(i)$  with  $\mathbf{t}(\mathbf{n}) = (\cdot, \mathbf{z})$ .
- (f) For all  $\mathbf{n} \in [0, D]^*$ ,  $l(\mathbf{n}) \stackrel{\text{def}}{=} \rho(\mathbf{n}) \cap \text{PROP}$ .

For all  $r \in \mathbf{N}_{\mathbf{F}}$ , we guarantee that  $\mathcal{R}_r$  is deterministic. Note also that it would be possible to define  $\mathcal{K}$  a bit differently by defining  $\mathcal{W}$  as the subset of  $[0, D]^*$  made of nodes  $\mathbf{n}$  such that  $\rho(\mathbf{n})$  is different from  $\perp$ . In the model defined above, the nodes  $\mathbf{n}$  such that  $\rho(\mathbf{n}) = \perp$  are simply not reachable.

We prove that for all nodes  $\mathbf{n} \in [0, D]^*$ , if  $\rho(\mathbf{n}) = Y_{\mathbf{n}}$  is a propositionally  $T$ -consistent set, then  $\mathcal{K}, \mathbf{n} \models \phi$  for all  $\phi \in Y_{\mathbf{n}}$ . Note that  $\rho(\varepsilon) = Y_{\varepsilon}$  satisfies  $\phi_0 \in Y_{\varepsilon}$ , and hence the claim above implies  $\mathcal{K}, \varepsilon \models \phi_0$ . By definition of propositionally  $T$ -consistent sets, we also have, for all nodes  $\mathbf{n} \in [0, D]^*$  and all  $1 \leq i \leq \ell$ ,  $\phi_i \in Y_{\mathbf{n}}$  implies  $\psi_i \in Y_{\mathbf{n}}$ , and hence – by the claim above –  $\mathcal{K}, \mathbf{n} \models \phi_i$  implies  $\mathcal{K}, \mathbf{n} \models \psi_i$ . Hence if the claim holds, we obtain that  $(T, \phi_0)$  is a positive instance.

So let  $\mathbf{n} \in [0, D]^*$  and assume  $\phi' \in Y_{\mathbf{n}}$ . Let  $\mathbf{t}(\mathbf{n}) = (X_{\mathbf{n}}, \mathbf{z})$  and  $\mathbf{t}(\mathbf{n} \cdot i) = (\cdot, \mathbf{z}_i)$  for all  $i \in [0, D]$ . Since  $\rho$  is a run of  $\mathbb{A}$  on  $\mathbf{t}$ , we know that the transition  $(Y_{\mathbf{n}}, X_{\mathbf{n}}, (\Theta_0, \rho(\mathbf{n} \cdot 0)), \dots, (\Theta_D, \rho(\mathbf{n} \cdot D))) \in \delta$  satisfies (1.) to (6.) from the definition of  $\delta$ , where  $\mathbf{t}(\mathbf{n}) = (X_{\mathbf{n}}, \mathbf{z})$ .

- Suppose  $\phi' = p$  for some  $p \in \text{PROP}$ . By 1. in  $\delta$ , we know that  $p \in X_{\mathbf{n}}$ . Hence  $\mathcal{K}, \mathbf{n} \models p$ .
- Suppose  $\phi' = \neg p$  for some  $p \in \text{PROP}$ . By 1. in  $\delta$  and by propositional  $T$ -consistency of  $\rho(\mathbf{n})$ , we know that  $p \notin X_{\mathbf{n}}$ . Hence  $\mathcal{K}, \mathbf{n} \models \neg p$ .
- Suppose  $\phi' = E\varepsilon \Theta$ . By 6(b) in  $\delta$ ,  $\Theta$  is a conjunct in  $\Theta_0$ . By the fact that  $\rho$  is a run, we know that  $\mathbb{Z} \models \Theta_0(\mathbf{z}, \mathbf{z}_0)$ , and hence also  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_0)$ . Hence  $\mathcal{K}, \mathbf{n} \models E\varepsilon \Theta$ .
- Suppose  $\phi' = A\varepsilon \Theta$ . By 6(b) in  $\delta$ ,  $\Theta$  is a conjunct in  $\Theta_0$ . By the fact that  $\rho$  is a run, we know that  $\mathbb{Z} \models \Theta_0(\mathbf{z}, \mathbf{z}_0)$  and hence  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_0)$ . Recall that  $\Theta$  only uses variables stating properties about the current node, so that  $\Theta$  only states properties about the current node. We can infer  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_i)$  for all  $i \in [0, D]$ . Hence  $\mathcal{K}, \mathbf{n} \models A\varepsilon \Theta$ .
- Suppose  $\phi' = Er \Theta$ . We distinguish two cases.
  - Let  $r \notin \mathbf{N}_{\mathbf{F}}$ . Then there must exist some  $j \in [1, D]$  such that  $\iota(\phi') = j$ . By 6(d) in  $\delta$ ,  $\Theta$  is a conjunct in  $\Theta_j$ .
  - Let  $r \in \mathbf{N}_{\mathbf{F}}$ . Then there must exist some  $j \in [1, D]$  such that  $\iota(r) = j$ . By 3. in  $\delta$ ,  $\rho(\mathbf{n} \cdot j) \neq \perp$ . By 6(e) in  $\delta$ ,  $\Theta$  is a conjunct in  $\Theta_j$ .

In both cases, since  $\rho$  is a run,  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_j)$ . By definition of  $\mathcal{K}$ , this implies that  $\mathcal{K}, \mathbf{n} \models Er \Theta$ .

- Suppose  $\phi' = Ar \Theta$ . We distinguish two cases.
  - Let  $r \notin \mathbf{N}_{\mathbf{F}}$  and  $(\mathbf{n}, \mathbf{n}') \in \mathcal{R}_r$  (so  $\rho(\mathbf{n})$  and  $\rho(\mathbf{n}')$  are distinct from  $\perp$ ). We distinguish two cases.
    - Case 1:**  $\mathbf{n}' = \mathbf{n} \cdot j$ ,  $\iota(Er \Theta') = j$  for some  $\Theta'$ .  
By 6(c,d) in  $\delta$ ,  $\Theta_j$  contains  $\Theta$  and therefore  $\mathbb{Z} \models \Theta_j(\mathbf{z}, \mathbf{z}_j)$ .
    - Case 2:**  $\mathbf{n}' = \mathbf{n} \cdot j$ ,  $\iota(EX_r \psi) = j$  for some  $\psi$ .  
By 6(c) in  $\delta$ ,  $\Theta_j$  contains also  $\Theta$  and therefore  $\mathbb{Z} \models \Theta_j(\mathbf{z}, \mathbf{z}_j)$  too.
  - Let  $r \in \mathbf{N}_{\mathbf{F}}$  and  $(\mathbf{n}, \mathbf{n}') \in \mathcal{R}_r$ . Consequently, by definition of  $\mathcal{R}_r$ ,  $\mathbf{n}' = \mathbf{n} \cdot \iota(r)$ , and both  $\rho(\mathbf{n})$  and  $\rho(\mathbf{n}')$  are distinct from  $\perp$ . By 6(e) in  $\delta$ ,  $\Theta$  is a conjunct of  $\Theta_j$ . Again  $\mathbb{Z} \models \Theta_j(\mathbf{z}, \mathbf{z}_j)$ . This allows us to conclude that  $\mathcal{K}, \mathbf{n} \models Ar \Theta$  (this time  $\mathbf{n}$  has at most one  $\mathcal{R}_r$ -successor, namely  $\mathbf{n} \cdot \iota(r)$ ).
- Suppose  $\phi' = EX_r \psi$ . Again, we distinguish two cases.
  - Case 1:**  $r \notin \mathbf{N}_{\mathbf{F}}$ .  
By 4. in  $\delta$ ,  $\psi \in Y_{\mathbf{n} \cdot j}$  with  $j = \iota(EX_r \psi)$  (hence  $Y_{\mathbf{n} \cdot j} \neq \perp$ ). By the induction hypothesis, we have  $\mathcal{K}, \mathbf{n} \cdot j \models \psi$ . By definition of  $\mathcal{R}_r$ ,  $(\mathbf{n}, \mathbf{n} \cdot j) \in \mathcal{R}_r$ . By the definition of  $\models$ , we get  $\mathcal{K}, \mathbf{n} \models EX_r \psi$ .
  - Case 2:**  $r \in \mathbf{N}_{\mathbf{F}}$ . By 4. in  $\delta$ ,  $\psi \in Y_{\mathbf{n} \cdot j}$  with  $j = \iota(r)$ . By the induction hypothesis, we have  $\mathcal{K}, \mathbf{n} \cdot j \models \psi$ . Similarly, we conclude  $\mathcal{K}, \mathbf{n} \models EX_r \psi$ .

- Suppose  $\phi' = \mathbf{AX}_r \psi$ . First, observe that by construction of  $\mathcal{R}_r$ , if  $(\mathbf{n}, \mathbf{n}') \in \mathcal{R}_r$ , then for some  $j$ ,  $\mathbf{n} = \mathbf{n} \cdot j$ ,  $r \triangleright j$  and  $\rho(\mathbf{n}') \neq \perp$ .  
Let  $\mathbf{n}'$  such that  $(\mathbf{n}, \mathbf{n}') \in \mathcal{R}_r$ , say  $\mathbf{n}' = \mathbf{n} \cdot j$ . By 5. in  $\delta$  and by the above observation,  $\psi \in Y_{\mathbf{n}'}$ . By the induction hypothesis, we have  $\mathcal{K}, \mathbf{n}' \models \psi$ . Since this holds for all the nodes in  $\mathcal{R}_r(\mathbf{n})$ , we can conclude that  $\mathcal{K}, \mathbf{n} \models \mathbf{AX}_r \psi$ .

“only if” Suppose  $T, \phi_0$  is a positive instance, that is,  $\mathcal{K}, w_0 \models \phi_0$  and for all  $1 \leq i \leq \ell$  and all  $w \in \mathcal{W}$  we have  $\mathcal{K}, w \models \phi_i$  implies  $\mathcal{K}, w \models \psi_i$ .

Below, we shall define a tree  $\mathbf{t} : [0, D]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$ , a map  $\rho : [0, D]^* \rightarrow \mathcal{Q}$  and an auxiliary map  $g : [0, D]^* \rightarrow \mathcal{W} \uplus \{\perp\}$  such that

- ( $\Delta$ )  $\rho$  is an accepting run on  $\mathbf{t}$  (and therefore  $\mathbf{t} \in \mathbf{L}(\mathbb{A})$ ).
- ( $\Delta\Delta$ ) For all  $\mathbf{n} \in [0, D]^*$ ,  $g(\mathbf{n}) = \perp$  iff  $\rho(\mathbf{n}) = \perp$ . (by construction)
- ( $\Delta\Delta\Delta$ ) For all  $\mathbf{n} \in [0, D]^*$  with  $g(\mathbf{n}) \neq \perp$ ,  $\rho(\mathbf{n}) = \{\psi \in \text{sub}(T, \phi_0) \mid \mathcal{K}, g(\mathbf{n}) \models \psi\}$  and if  $\mathbf{t}(\mathbf{n}) = (X, \mathbf{z})$ , then  $X = \rho(\mathbf{n}) \cap \text{PROP}$  and for all  $i \in [1, \beta]$ ,  $\mathbf{v}(g(\mathbf{n}), \mathbf{x}_i) = \mathbf{z}(i)$ . (by construction)

Before going any further, let us recall that for all  $w \in \mathcal{W}$ , the set  $Y = \{\psi \in \text{sub}(T, \phi_0) \mid \mathcal{K}, w \models \psi\}$  (also written  $Y_w$  in the sequel) is indeed propositionally  $T$ -consistent.

- Towards contradiction, suppose  $\{p, \neg p\} \subseteq Y$  for some  $p \in \text{PROP}$ . By definition of  $Y$ , we have  $\mathcal{K}, w \models p$  and  $\mathcal{K}, w \models \neg p$ , contradiction.
- If  $\phi'_1 \vee \phi'_2 \in Y$ , then  $\mathcal{K}, w \models \phi'_1$  or  $\mathcal{K}, w \models \phi'_2$ , hence  $\phi'_1 \in Y$  or  $\phi'_2 \in Y$ , so that indeed  $\{\phi'_1, \phi'_2\} \cap Y \neq \emptyset$ .
- If  $\phi'_1 \wedge \phi'_2 \in Y$ , then  $\mathcal{K}, w \models \phi'_1$  and  $\mathcal{K}, w \models \phi'_2$ , hence  $\phi'_1 \in Y$  and  $\phi'_2 \in Y$ , so that indeed  $\{\phi'_1, \phi'_2\} \subseteq Y$ .
- Suppose  $\phi_i \in Y$  for some  $1 \leq i \leq \ell$ . Then  $\mathcal{K}, w \models \phi_i$ . By assumption ( $\mathcal{K}$  is a model of  $(T, \phi_0)$ ), we also have  $\mathcal{K}, w \models \psi_i$ . Hence  $\psi_i \in Y$ .

Let us define  $\mathbf{t}$ ,  $\rho$  and  $g$  inductively. For the base case,  $g(\varepsilon) \stackrel{\text{def}}{=} w_0$ ,  $\rho(\varepsilon) \stackrel{\text{def}}{=} Y_{w_0}$  and  $\mathbf{t}(\varepsilon) \stackrel{\text{def}}{=} (X, \mathbf{z})$  such that  $X \stackrel{\text{def}}{=} Y_{w_0} \cap \text{PROP}$  and for all  $i \in [1, \beta]$ ,  $\mathbf{z}(i) \stackrel{\text{def}}{=} \mathbf{v}(w_0, \mathbf{x}_i)$ . From now on, the definitions are completed by picking the smallest element  $\mathbf{n} \cdot j \in [0, D]^*$  with respect to the lexicographical ordering such that  $g(\mathbf{n})$  is defined and  $g(\mathbf{n} \cdot j)$  is undefined. The values  $g(\mathbf{n} \cdot j)$ ,  $\rho(\mathbf{n} \cdot j)$  and  $\mathbf{t}(\mathbf{n} \cdot j)$  are defined as follows.

- $g(\mathbf{n} \cdot j)$  is defined with a case analysis.
  - (A) If  $g(\mathbf{n}) = \perp$  or  $j = 0$ , then  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \perp$ .
  - (B) Otherwise, suppose that  $\iota(\mathbf{Er} \Theta) = j$  (so  $r \notin \mathbf{N}_{\mathbf{F}}$ ). If  $\mathbf{Er} \Theta \notin Y_{g(\mathbf{n})}$ , then  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \perp$ . Otherwise,  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} w'$  for some world  $w' \in \mathcal{R}_r(g(\mathbf{n}))$  verifying  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}')$ ,  $\mathbf{v}(g(\mathbf{n})) = \mathbf{z}$  and  $\mathbf{v}(w') = \mathbf{z}'$ . Observe that by satisfaction of  $\mathcal{K}, g(\mathbf{n}) \models \mathbf{Er} \Theta$ , such a world  $w'$  exists.
  - (C) Otherwise, suppose that  $\iota(\mathbf{EX}_r \psi) = j$  (so  $r \notin \mathbf{N}_{\mathbf{F}}$ ). If  $\mathbf{EX}_r \psi \notin Y_{g(\mathbf{n})}$ , then  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \perp$ . Otherwise,  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} w'$  with  $w' \in \mathcal{R}_r(g(\mathbf{n}))$  and  $\mathcal{K}, w' \models \psi$ . By satisfaction of  $\mathcal{K}, g(\mathbf{n}) \models \mathbf{EX}_r \psi$ , such a world  $w'$  exists.
  - (D) Otherwise, suppose that  $\iota(r) = j$  (so  $r \in \mathbf{N}_{\mathbf{F}}$ ). If  $(\text{sub}_{\mathbf{EX}_r}(T, \phi_0) \cup \text{sub}_{\mathbf{Er}}(T, \phi_0)) \cap Y_{g(\mathbf{n})} = \emptyset$ , then  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \perp$ . Otherwise,  $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} w'$  for the unique world  $w' \in \mathcal{R}_r(g(\mathbf{n}))$ . Again,  $w'$  necessarily exists.
- If  $g(\mathbf{n} \cdot j) = \perp$ , then  $\rho(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \perp$ . Otherwise,  $\rho(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} Y_{g(\mathbf{n} \cdot j)}$ .
- If  $g(\mathbf{n} \cdot j) \neq \perp$ , then  $\mathbf{t}(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} (X, \mathbf{z})$  such that  $X \stackrel{\text{def}}{=} Y_{g(\mathbf{n} \cdot j)} \cap \text{PROP}$  and for all  $i \in [1, \beta]$ ,  $\mathbf{z}(i) \stackrel{\text{def}}{=} \mathbf{v}(g(\mathbf{n} \cdot j), \mathbf{x}_i)$ . Otherwise,  $\mathbf{t}(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} (\emptyset, \mathbf{0})$  (arbitrary value).

One can check that the conditions  $(\triangle\triangle)$  and  $(\triangle\triangle\triangle)$  hold by construction. It remains to check the satisfaction of  $(\triangle)$ , namely that  $\rho$  is an accepting run on  $\mathbf{t}$ .

To do so, we would like first to observe that given  $Y, Y_0, \dots, Y_D \in Q$  and  $X \in \Sigma$  satisfying the conditions (1.)–(5.) in the definition of  $\delta$ , there are unique  $\Theta_0, \dots, \Theta_D$  such that  $(Y, X, (\Theta_0, Y_0), \dots, (\Theta_D, Y_D))$  belongs to  $\delta$ . Indeed, the five cases 6(a)–6(e) cover all the possibilities, are pairwise disjoint and each case defines  $\Theta_j$  in a unique way.

Secondly, for all  $\mathbf{n} \in [0, D]^*$ , one can check that  $\rho(\mathbf{n}), \rho(\mathbf{n} \cdot 0), \dots, \rho(\mathbf{n} \cdot D)$  and  $\rho(\mathbf{n}) \cap \text{PROP}$  satisfy the conditions (1.)–(5.) in the definition of  $\delta$ . By convention,  $\perp \cap \text{PROP} = \emptyset$ . This is a consequence of the fact that in (A)–(D) above, the definition of the map  $g$  is guided by the model  $\mathcal{K}$  and the satisfaction relation. By way of example, let us check the satisfaction of the conditions (4.) and (5.).

4. Let  $\text{EX}_r \psi \in \rho(\mathbf{n})$ . If  $r \in \mathbf{N}_{\mathbf{F}}$ , then  $\iota(r) = j$  for some  $j \in [1, D]$ . By (D),  $\psi \in \rho(\mathbf{n} \cdot j)$  as there is a unique  $w' \in \mathcal{R}_r(g(\mathbf{n}))$  such that  $\rho(\mathbf{n}) = Y_{g(\mathbf{n})}$  and  $\mathcal{K}, w' \models \psi$ . Otherwise  $\iota(\text{EX}_r \psi) = j$ . By (C),  $\psi \in \rho(\mathbf{n} \cdot j)$  as there is  $w' \in \mathcal{R}_r(g(\mathbf{n}))$  such that  $\rho(\mathbf{n}) = Y_{g(\mathbf{n})}$  and  $\mathcal{K}, w' \models \psi$ .
5. Let  $\text{AX}_r \psi \in \rho(\mathbf{n})$ , and  $j \in [1, D]$  such that  $\rho(\mathbf{n} \cdot j) \neq \perp$  and  $r \triangleright j$ . By (C) and (D), we conclude that  $\psi \in \rho(\mathbf{n} \cdot j)$  since  $\mathcal{K}, g(\mathbf{n}) \models \text{AX}_r \psi$  (equivalent to  $\text{AX}_r \psi \in \rho(\mathbf{n})$ ) implies for all  $w' \in \mathcal{R}_r(g(\mathbf{n}))$ , we have  $\mathcal{K}, w' \models \psi$ , in particular for  $w' = g(\mathbf{n} \cdot j)$ , thus  $\psi \in \rho(\mathbf{n} \cdot j)$ .

Consequently, we write  $\Theta_0^{\mathbf{n} \cdot 0}, \dots, \Theta_D^{\mathbf{n} \cdot D}$  to denote the unique tuple such that

$$(\rho(\mathbf{n}), \rho(\mathbf{n}) \cap \text{PROP}, (\Theta_0^{\mathbf{n} \cdot 0}, \rho(\mathbf{n} \cdot 0)), \dots, (\Theta_D^{\mathbf{n} \cdot D}, \rho(\mathbf{n} \cdot D)))$$

belongs to  $\delta$ . This guarantees that  $\rho$  is a run,  $\rho(\varepsilon)$  is initial because  $\phi_0 \in \rho(\varepsilon)$ , and it is accepting because  $F = Q$ . It remains to verify that  $\rho$  is a run on  $\mathbf{t}$ .

Let  $\mathbf{n} \in [0, D]^*$ ,  $j \in [0, D]$  with  $\mathbf{t}(\mathbf{n}) = (X, \mathbf{z})$ ,  $\mathbf{t}(\mathbf{n} \cdot 0) = (X_0, \mathbf{z}_0)$ ,  $\dots$ ,  $\mathbf{t}(\mathbf{n} \cdot D) = (X_D, \mathbf{z}_D)$ . Let us show that  $\mathbb{Z} \models \Theta_j^{\mathbf{n} \cdot j}(\mathbf{z}, \mathbf{z}_j)$ . We make a case analysis depending on how  $\Theta_j^{\mathbf{n} \cdot j}$  is computed. If  $\rho(\mathbf{n}) = \perp$ , then by 6(a) we have  $\Theta_j^{\mathbf{n} \cdot j} = \top$  and therefore  $\mathbb{Z} \models \Theta_j^{\mathbf{n} \cdot j}(\mathbf{z}, \mathbf{z}_j)$  holds. Otherwise  $\rho(\mathbf{n}) \neq \perp$ , and for all the cases 6(b)–(e),  $\Theta_\varepsilon = (\bigwedge_{E\varepsilon} \Theta', A\varepsilon \in \rho(\mathbf{n}) \Theta')$  is a conjunct of  $\Theta_j^{\mathbf{n} \cdot j}$ . Recall that each constraint  $\Theta'$ 's in  $\Theta_\varepsilon$  only contains variables (among  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$ ) expressing constraints on the data values of the current node (or equivalently of the world  $g(\mathbf{n})$ ). Hence, by

$$\mathcal{K}, g(\mathbf{n}) \models (\bigwedge_{E\varepsilon} \Theta') \wedge (\bigwedge_{A\varepsilon} \Theta'),$$

we get  $\mathbb{Z} \models \Theta_\varepsilon(\mathbf{z}, \mathbf{z}_j)$  (actually, the values in  $\mathbf{z}_j$  are useless to check the satisfaction). Let us perform a final case analysis.

- (b) If  $j = 0$  or  $\rho(\mathbf{n} \cdot j) = \perp$ , then  $\Theta_j^{\mathbf{n} \cdot j}$  is equal to  $\Theta_\varepsilon$  and therefore we are done.
- (c) Otherwise, if  $((\text{Er } \Theta \notin \rho(\mathbf{n}) \text{ and } \iota(\text{Er } \Theta) = j) \text{ or } \iota(\text{EX}_r \psi) = j \text{ for some } \text{Er } \Theta, \text{EX}_r \psi \in \mathbf{E}_{\overline{\mathbf{F}}}(T, \phi_0))$ , then  $\Theta_j^{\mathbf{n} \cdot j} = \Theta_\varepsilon \wedge (\bigwedge_{A_r \Theta' \in \rho(\mathbf{n})} \Theta')$ . Let  $A_r \Theta' \in \rho(\mathbf{n})$  and let us show that  $\mathbb{Z} \models \Theta'(\mathbf{z}, \mathbf{z}_j)$ . By definition of  $g$  and  $\rho$ , we have  $\mathcal{K}, g(\mathbf{n}) \models A_r \Theta'$  and  $(g(\mathbf{n}), g(\mathbf{n} \cdot j)) \in \mathcal{R}_r$  (by (B) and (C)). By definition of  $\models$ ,  $\mathbb{Z} \models \Theta'(\mathbf{v}(g(\mathbf{n})), \mathbf{v}(g(\mathbf{n} \cdot j)))$  and by  $(\triangle\triangle\triangle)$ , we can conclude  $\mathbb{Z} \models \Theta'(\mathbf{z}, \mathbf{z}_j)$ .
- (d) Otherwise, if there is  $\text{Er } \Theta \in Y$  such that  $\iota(\text{Er } \Theta) = j$ ,

$$\Theta_j^{\mathbf{n} \cdot j} = \Theta_\varepsilon \wedge (\bigwedge_{A_r \Theta' \in \rho(\mathbf{n})} \Theta') \wedge \Theta.$$

We have already handled above most of the constraints. Let us check that  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_j)$ . Again, by definition of  $g$  and  $\rho$ , we have  $\mathcal{K}, g(\mathbf{n}) \models \text{Er } \Theta$ ,  $(g(\mathbf{n}), g(\mathbf{n} \cdot j)) \in \mathcal{R}_r$  and  $\mathbb{Z} \models \Theta(\mathbf{v}(g(\mathbf{n})), \mathbf{v}(g(\mathbf{n} \cdot j)))$  (by (B)). By  $(\triangle\triangle\triangle)$ , we can conclude  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_j)$ .

(e) Otherwise, i.e.  $r = \iota^{-1}(j) \in \mathbf{N}_{\mathbf{F}}$  and  $\rho(\mathbf{n} \cdot j) \neq \perp$ ,

$$\Theta_j = \Theta_\varepsilon \wedge \left( \bigwedge_{Ar \ \Theta', Er \ \Theta' \in \rho(\mathbf{n})} \Theta' \right).$$

By (D),  $g(\mathbf{n} \cdot j)$  is the unique world in  $\mathcal{R}_r(g(\mathbf{n}))$ . By definition of  $\rho$ , we have

$$\mathcal{K}, g(\mathbf{n}) \models \left( \bigwedge_{Ar \ \Theta' \in \rho(\mathbf{n})} Ar \ \Theta' \right) \wedge \left( \bigwedge_{Er \ \Theta' \in \rho(\mathbf{n})} Er \ \Theta' \right).$$

As for (c) above, we can show that for all  $Ar \ \Theta' \in \rho(\mathbf{n})$ , we have  $\mathbb{Z} \models \Theta'(z, z_j)$ . As for (d) above, we can also show that for all  $Er \ \Theta' \in \rho(\mathbf{n})$ , we have  $\mathbb{Z} \models \Theta'(z, z_j)$ . In conclusion,  $\mathbb{Z} \models \Theta_j^{\mathbf{n} \cdot j}(z, z_j)$ .  $\square$

The complexity analysis for  $\text{CTL}(\mathbb{Z})$  in Section 5.1 can be adequately adapted to the above construction. This allows us to get the following by-product.

**Proposition 5 ([CT16, LOS20]).**  *$\text{TSAT}(\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z}_c))$  is decidable and in  $\text{EXPTIME}$ .*

## 6 SAT( $\text{CTL}^*(\mathbb{Z})$ ) is also 2ExpTime-complete

In this section, we show that  $\text{SAT}(\text{CTL}^*(\mathbb{Z}))$  can be solved in  $2\text{EXPTIME}$  by using Rabin tree constraint automata (see e.g. Section 4.5). We follow the automata-based approach for  $\text{CTL}^*$ , see e.g. [ES84, EJ00], but adapted to tree automata with constraints. Not only we had to check that the essential steps for  $\text{CTL}^*$  can be lifted to  $\text{CTL}^*(\mathbb{Z})$  but also that computationally we are in a position to provide an optimal complexity upper bound.

This section is structured as follows. In Section 6.1, we establish a special form for  $\text{CTL}^*(\mathbb{Z})$  formulae from which tree automata are defined, adapting the developments from [ES84]. In particular, the only terms that can occur in formulae in special form are from  $\text{T}_{\text{VAR}}^{\leq 1}$ . In Section 6.2, we explain how  $\text{LTL}(\mathbb{Z})$  path formulae can be encoded by nondeterministic word constraint automata, extending the reduction from  $\text{LTL}$  formulae into Büchi automata, see e.g. [VW94]. Determinisation of nondeterministic word constraint automata with Rabin word constraint automata is proved in Section 6.3 following developments from Safra's PhD thesis [Saf89, Chapter 1] but carefully adapted to the context of constraint automata. In Section 6.4, we take advantage of previous subsections to define Rabin tree constraint automata that accept trees that satisfy formulae of the form  $\text{AGE } \Phi$  and  $\text{A } \Phi$ , respectively where  $\Phi$  is an  $\text{LTL}(\mathbb{Z})$  formula in simple form, as it occurs in  $\text{CTL}^*(\mathbb{Z})$  formulae in special form. The determinisation construction from Section 6.3 is essential to design RTCA for the formulae of the form  $\text{A } \Phi$ . Section 6.5 wraps up everything to get the final  $2\text{EXPTIME}$  upper bound. It contains Lemma 21 about intersection of RTCA that is instrumental for the ultimate complexity analysis.

### 6.1 $\text{CTL}^*(\mathbb{Z})$ formulae in special form

*State formulae*  $\phi$  and *path formulae*  $\Phi$  of  $\text{CTL}^*(\mathbb{Z})$  are defined according to the grammars below.

$$\phi := \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \text{E}\Phi \mid \text{A}\Phi \quad \Phi := \phi \mid \Theta \mid \neg\Phi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \text{X}\Phi \mid \Phi\text{U}\Phi \mid \Phi\text{R}\Phi,$$

where  $\Theta$  is a constraint. Formulae of the logic  $\text{LTL}(\mathbb{Z})$  are defined from path formulae for  $\text{CTL}^*(\mathbb{Z})$  according to the grammar below.

$$\Phi := \Theta \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \text{X}\Phi \mid \Phi\text{U}\Phi \mid \Phi\text{R}\Phi,$$

where  $\Theta$  is a constraint. Note that negation occurs only in constraints since the LTL logical connectives have their dual in  $\text{LTL}(\mathbb{Z})$ . An  $\text{LTL}(\mathbb{Z})$  formula is in *simple form* iff its terms are from  $\text{T}_{\text{VAR}}^{\leq 1}$ . More generally, an  $\text{CTL}^*(\mathbb{Z})$  state formula is in *simple form* iff its terms are from  $\text{T}_{\text{VAR}}^{\leq 1}$ .

Following ideas from [ES84] for  $\text{CTL}^*$ , we prove that we can translate every  $\text{CTL}^*(\mathbb{Z})$  state formula  $\phi$  into a certain normal form, called *special*. A  $\text{CTL}^*(\mathbb{Z})$  state formula  $\phi$  is in *special form* if it has the form below

$$\text{E} (\mathbf{x} = 0) \wedge \left( \bigwedge_{i \in [1, D-1]} \text{AGE } \Phi_i \right) \wedge \left( \bigwedge_{j \in [1, D']} \text{A } \Phi'_j \right),$$

where the  $\Phi_i$ 's and the  $\Phi'_j$ 's are  $\text{LTL}(\mathbb{Z})$  formulae in simple form, for some  $D \geq 1$ ,  $D' \geq 0$ . First, we need to show that the proof of Proposition 2 can be adapted to  $\text{CTL}^*(\mathbb{Z})$ , leading to the result below.

**Proposition 6.** *For every  $\text{CTL}^*(\mathbb{Z})$  formula  $\phi$ , one can construct in polynomial-time in the size of  $\phi$  a  $\text{CTL}^*(\mathbb{Z})$  formula  $\phi'$  in simple form such that  $\phi$  is satisfiable iff  $\phi'$  is satisfiable.*

*Proof.* The proof is a slight variant of the proof of Proposition 2, we keep the same notations whenever possible. First, we can establish that  $\text{CTL}^*(\mathbb{Z})$  has the tree model property, exactly as done in the proof of Proposition 2 for  $\text{CTL}(\mathbb{Z})$  (use of unfoldings for Kripke structures) and therefore we omit the details herein.

Now, we move to the construction of  $\phi'$  in simple form. We use the notion of forward degree introduced in the proof of Proposition 2 (for  $\text{CTL}^*(\mathbb{Z})$  state formulae  $\psi$  and for constraints  $\Theta$ ). Let  $\phi$  be a state formula in  $\text{CTL}^*(\mathbb{Z})$  such that  $\text{fd}(\phi) = N$  and the variables occurring in  $\phi$  are among  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$ . Below, we build a formula  $\phi'$  over the variables  $\mathbf{x}_1^{-N}, \dots, \mathbf{x}_1^0, \dots, \mathbf{x}_\beta^{-N}, \dots, \mathbf{x}_\beta^0$  with  $\text{fd}(\phi') \leq 1$  such that  $\phi$  is satisfiable in a tree Kripke structure iff  $\phi'$  is satisfiable in a tree Kripke structure and  $\phi'$  can be computed in polynomial-time in the size of  $\phi$ .

Given  $\Theta$  occurring in  $\phi$  with  $\text{fd}(\Theta) = M$  (and therefore  $M \leq N$ ), we write  $\text{jump}(\Theta, M)$  to denote the constraints as done in the proof of Proposition 2.

Let  $\mathbf{t}$  be the translation map that is homomorphic for Boolean connectives, temporal connectives and path quantifiers such that  $\mathbf{t}(\Theta) \stackrel{\text{def}}{=} \mathbf{X}^M \text{jump}(\Theta, M)$ , where  $\text{fd}(\Theta) = M$  for all maximal constraints  $\Theta$  occurring in  $\phi$ . This is the most essential change with respect to the proof of Proposition 2. As  $\Theta$  is always in the scope of a path quantifier, the path formula  $\mathbf{X}^M \text{jump}(\Theta, M)$  is well-defined. Let  $\phi'$  be defined as follows:

$$\mathbf{t}(\phi) \wedge \text{AG A} \left( \bigwedge_{j \in [1, \beta], k \in [0, N-1]} \mathbf{x}_j^{-k} = \mathbf{X} \mathbf{x}_j^{-k-1} \right).$$

One can show that  $\phi$  is satisfiable in a tree Kripke structure iff  $\phi'$  is satisfiable in a tree Kripke structure.  $\square$

Below, we establish the property that allows us to restrict ourselves to  $\text{CTL}^*(\mathbb{Z})$  state formulae in special form only.

**Proposition 7.** *For every  $\text{CTL}^*(\mathbb{Z})$  formula  $\phi$ , one can construct in polynomial time in the size of  $\phi$  a  $\text{CTL}^*(\mathbb{Z})$  formula  $\phi'$  in special form such that  $\phi$  is satisfiable iff  $\phi'$  is satisfiable.*

*Proof.* By Proposition 6, we can assume that  $\phi$  is in simple form, and  $\phi$  is built over the variables  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$  and  $\mathbf{x}'_1, \dots, \mathbf{x}'_\beta$ . Moreover, we assume that  $\phi$  is in negation normal form.

We use a simple property that illustrates well the renaming technique [Sco62] used below. Let  $\psi$  be a  $\text{CTL}^*(\mathbb{Z})$  state formula with state subformula  $\psi'$  and  $y$  be a variable *not occurring* in  $\psi$ . Then,  $\psi$  is

satisfiable iff  $\psi[\psi' \leftarrow E(y = 0)] \wedge AG(E(y = 0) \Leftrightarrow \psi')$  is satisfiable, where  $\psi[\psi' \leftarrow E(y = 0)]$  denotes the state formula obtained from  $\psi$  by replacing every occurrence of  $\psi'$  by  $E(y = 0)$ . The idea is to replace  $\psi'$  by the atomic constraint  $(y = 0)$ , where  $y$  is a fresh variable not occurring before. The new constraint  $(y = 0)$  is tied to the original subformula  $\psi'$  via the conjunct  $AG(E(y = 0) \Leftrightarrow \psi')$ . Note that strictly speaking,  $y = 0$  is not a state formula but morally it is because its satisfaction depends only on the current state. That is why we use  $E(y = 0)$  instead of the more natural constraint  $y = 0$ . In the transformations below, we need sometimes to remove  $E$  in  $E(y = 0)$  when the occurrence of  $y = 0$  is already in the scope of a path quantifier. (Alternatively, we could slightly redefine  $CTL^*(\mathbb{Z})$  to accept also as atomic formulae constraints  $\Theta$  in which all terms are some variables  $x_i$  (no prefix with  $X$ ). In that slight extension,  $y = 0$  would be authorised as a state formula.)

In order to compute  $\phi'$ , we perform on  $\phi$  several transformations of the above form. We write  $\psi$  to denote current state formulae on which the transformations are performed and initially  $\psi$  takes the value  $\phi$  (a  $CTL^*(\mathbb{Z})$  state formula in simple form). Through the sequence of transformations,  $\psi$  is maintained in the following shape:

$$\varphi \wedge \bigwedge_i AG(E(y_i = 0) \Leftrightarrow Q_i \Phi_i),$$

where each  $Q_i \in \{E, A\}$ , each  $\Phi_i$  is an  $LTL(\mathbb{Z})$  path formula in simple form and  $\varphi$  is a  $CTL^*(\mathbb{Z})$  formula in simple form. In order to compute the new value for  $\psi$ , suppose that  $\varphi$  contains a state subformula  $\psi'$  of the form  $Q \Phi$ , where the only path quantifiers occurring in  $\Phi$  occur in state formulae of the form  $Q' z = 0$  (no need to perform a renaming on  $Q' z = 0$ ). We write  $\Phi^\dagger$  to denote the path formula (in simple form) obtained from  $\Phi$  by replacing every occurrence of  $Q' z = 0$  by  $z = 0$ . Obviously,  $Q \Phi$  is logically equivalent to  $Q \Phi^\dagger$ . The new value for  $\psi$  is defined below ( $y$  is a fresh variable):

$$\varphi[Q \Phi \leftarrow E(y = 0)] \wedge AG(E(y = 0) \Leftrightarrow Q \Phi^\dagger) \wedge \bigwedge_i AG(E(y_i = 0) \Leftrightarrow Q_i \Phi_i),$$

One can show that the transformation preserves satisfiability and moreover, repeating this procedure can be done only a finite amount of times, guaranteeing termination. At the end of all these transformations, the resulting formula  $\psi$  is now of the form

$$\varphi \wedge \bigwedge_i AG(E(y_i = 0) \Leftrightarrow Q_i \Phi_i),$$

where  $\varphi$  is a Boolean combination of state formulae of the form  $Q x = 0$  and the  $y_i$ 's are distinct and new variables not occurring in the original formula  $\phi$ . We write  $\varphi^\dagger$  to denote the path formula obtained from  $\varphi$  by removing all the path quantifiers. Again,  $Q \varphi$  is logically equivalent to  $Q \varphi^\dagger$  for all  $Q \in \{E, A\}$ . The intermediate state formula  $\phi^*$  takes the value below:

$$E(z = 0) \wedge AG(\overline{z = 0 \Leftrightarrow \varphi^\dagger}) \wedge \bigwedge_i AG(E(x_i = 0) \Leftrightarrow Q_i \Phi_i),$$

where  $z$  is again a fresh variable and  $\overline{z = 0 \Leftrightarrow \varphi^\dagger}$  is in negation normal form and logically equivalent to  $z = 0 \Leftrightarrow \varphi^\dagger$ . The formulae  $\phi$  and  $\phi^*$  are equi-satisfiable.

It remains to explain how to transform each  $AG(E(x_i = 0) \Leftrightarrow Q_i \Phi_i)$  so that we get the final formula  $\phi'$  in special form from  $\phi^*$ . For instance,  $AG(E(y_i = 0) \Leftrightarrow A \Phi_i)$  shall be replaced by

$$AG(\neg(y_i = 0) \vee \Phi_i) \wedge AGE(y_i = 0 \vee \overline{\neg \Phi_i}),$$

where  $\overline{\neg\Phi_i}$  is logically equivalent to  $\neg\Phi_i$  but in negation normal form. Note that  $\neg(y_i = 0) \vee \Phi_i$  and  $y_i = 0 \vee \neg\Phi_i$  are LTL( $\mathbb{Z}$ ) formulae in simple form, exactly what is needed for the final CTL\*( $\mathbb{Z}$ ) state formula  $\phi'$  in special form.

Below, we list the logical equivalences (hinted above) we take advantage of.

- $\text{AG}(\text{E}(y_i = 0) \Rightarrow \text{A } \Phi_i) \Leftrightarrow \text{AG}(\neg(y_i = 0) \vee \Phi_i).$
- $\text{AG}(\text{E}(y_i = 0) \Rightarrow \text{E } \Phi_i) \Leftrightarrow \text{AGE}(\neg(y_i = 0) \vee \Phi_i).$
- $\text{AG}(\neg\text{E}(y_i = 0) \Rightarrow \neg\text{A } \Phi_i) \Leftrightarrow \text{AGE}(y_i = 0 \vee \overline{\neg\Phi_i}).$
- $\text{AG}(\neg\text{E}(y_i = 0) \Rightarrow \neg\text{E } \Phi_i) \Leftrightarrow \text{AG}((y_i = 0) \vee \neg\Phi_i).$

The formula  $\phi'$  is obtained from  $\phi^*$  by replacing each element of the generalised conjunction in  $\phi^*$  by two formulae based on these equivalences.  $\square$

*Example 5.* To illustrate the construction from the proof of Proposition 7, we consider the formula  $\phi$  below.

$$\phi = \text{E}((x'_1 < x_1) \text{U } \text{AX}(x_2 = x'_2)) \wedge \text{EG}(x_1 < x_2).$$

Below, we present the formulae  $\psi_0 = \phi$ ,  $\psi_1, \dots, \psi_3$ ,  $\phi^*$  obtained by application of the different renaming steps.

$$\begin{aligned} \psi_1 &= \overbrace{\text{AG}(\text{E}(y_1 = 0) \Leftrightarrow (\text{AX}(x_2 = x'_2)))}^{= \psi'_1} \wedge \text{E}((x'_1 < x_1) \text{U } \text{E}(y_1 = 0)) \wedge \text{EG}(x_1 < x_2). \\ \psi_2 &= \overbrace{\text{AG}(\text{E}(y_2 = 0) \Leftrightarrow (\text{E}(x'_1 < x_1) \text{U } y_1 = 0))}^{= \psi'_2} \wedge \psi'_1 \wedge \text{E}(y_2 = 0) \wedge \text{EG}(x_1 < x_2). \end{aligned}$$

Note that ‘E’ is removed from  $\text{E}(y_1 = 0)$  in  $\psi'_2$ .

$$\begin{aligned} \psi_3 &= \overbrace{\text{AG}(\text{E}(y_3 = 0) \Leftrightarrow \text{EG}(x_1 < x_2))}^{= \psi'_3} \wedge \psi'_2 \wedge \psi'_1 \wedge \text{E}(y_2 = 0) \wedge \text{E}(y_3 = 0). \\ \phi^* &= \text{AG}((y_4 = 0) \Leftrightarrow (y_2 = 0 \wedge y_3 = 0)) \wedge \psi'_3 \wedge \psi'_2 \wedge \psi'_1 \wedge \text{E}(y_4 = 0). \end{aligned}$$

Each state formula  $\psi'_i$  in  $\phi^*$  is then also replaced by a conjunction of two state formulae in order to compute  $\phi'$ . By way of example, we present below the conjunction replacing  $\psi'_2$ .

$$\text{AGE}(\neg(y_2 = 0) \vee (x'_1 < x_1) \text{U } (y_1 = 0)) \wedge \text{AG}((y_2 = 0) \vee (\neg(x'_1 < x_1)) \text{R } \neg(y_1 = 0))$$

## 6.2 Nondeterministic word constraint automata for LTL( $\mathbb{Z}$ )-formulae

Models of LTL( $\mathbb{Z}$ ) are of the form  $\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta$ , assuming that the formulae evaluated on  $\mathbf{w}$  contains variables among  $x_1, \dots, x_\beta$  only. The models  $\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta$  are understood as paths from Kripke structures and therefore the satisfaction relation for LTL( $\mathbb{Z}$ ) is directly designed from the one involving paths for CTL\*( $\mathbb{Z}$ ). Moreover, models of the form  $\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta$  can be viewed as infinite branches in trees  $\mathbf{t}$  of the form  $\mathbf{t} : [0, D - 1]^* \rightarrow \mathbb{Z}^\beta$ . In short, LTL( $\mathbb{Z}$ ) is the fragment of CTL\*( $\mathbb{Z}$ ) in the same way LTL is a fragment CTL\*.

Since  $\omega$ -sequences  $\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta$  are trees of degree one, TCA with  $D = 1$  accepts LTL( $\mathbb{Z}$ ) models, ignoring the letters from the finite alphabet  $\Sigma$ . In the rest of this section, we assume that  $\Sigma = \{\dagger\}$  and we indifferently identify  $\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta$  with  $\mathbf{w} : \mathbb{N} \rightarrow \{\dagger\} \times \mathbb{Z}^\beta$ .



A tree constraint automaton  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$  accepts infinite words over  $\Sigma \times \mathbb{Z}^\beta$  if  $D = 1$ . In that case, we simply write  $\mathbb{A} = (Q, \Sigma, \beta, Q_{\text{in}}, \delta, F)$ , where  $\delta \subseteq (Q \times \Sigma \times \text{TreeCons}(\beta) \times Q)$ .

A constraint automaton  $\mathbb{A} = (Q, \Sigma, \beta, Q_{\text{in}}, \delta, F)$  is *deterministic* if for all transitions  $(q, \mathbf{a}, \Theta_1, q_1)$  and  $(q, \mathbf{a}, \Theta_2, q_2) \in \delta$ , if  $(q, \mathbf{a}, \Theta_1, q_1) \neq (q, \mathbf{a}, \Theta_2, q_2)$  implies that  $\Theta_1 \wedge \Theta_2$  is not satisfiable. That is, for all  $q \in Q$ ,  $\mathbf{a} \in \Sigma$  and  $(\mathbf{z}, \mathbf{z}') \in \mathbb{Z}^{2\beta}$ , there exists at most a single transition  $(q, \mathbf{a}, \Theta, q') \in \delta$  such that  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}')$ .

A run  $\rho$  of  $\mathbb{A}$  on an infinite word  $\mathbf{w} : \mathbb{N} \rightarrow (\Sigma \times \mathbb{Z}^\beta)$  is a mapping  $\rho : \mathbb{N} \rightarrow Q$  satisfying  $\rho(0) \in Q_{\text{in}}$ , and for every  $n \geq 0$  with  $\mathbf{w}(n) = (\mathbf{a}, \mathbf{z})$ ,  $\rho(n) = q$ ,  $\mathbf{w}(n+1) = (\mathbf{a}', \mathbf{z}')$ , and  $\rho(n+1) = q'$ , there exists a transition  $(q, \mathbf{a}, \Theta, q') \in \delta$  and  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}')$ . Below, we also need the notion of a *finite* run: by convention, a single location  $q \in Q$  is a run on every single-letter word  $(\mathbf{a}, \mathbf{z}) \in (\Sigma \times \mathbb{Z}^\beta)$ . Given a finite non-empty word  $\mathbf{w} = (\mathbf{a}_1, \mathbf{z}_1)(\mathbf{a}_2, \mathbf{z}_2) \dots (\mathbf{a}_k, \mathbf{z}_k)$ , a run of  $\mathbb{A}$  on  $\mathbf{w}$  is of the form  $q_1, q_2 \dots q_k$  such that for every  $1 \leq i < k$  there exists a transition  $(q_i, \mathbf{a}_i, \Theta_i, q_{i+1}) \in \delta$  such that  $\mathbb{Z} \models \Theta_i(\mathbf{z}_i, \mathbf{z}_{i+1})$ . Since the constraints in  $\Theta_i$ 's are about two tuples of  $\beta$  data values, the pair  $(\mathbf{a}_k, \mathbf{z}_k)$  is only there for the condition  $\mathbb{Z} \models \Theta_{k-1}(\mathbf{z}_{k-1}, \mathbf{z}_k)$ . In particular, by definition,  $q$  is always a run of  $(\mathbf{a}, \mathbf{z})$ . This notion of finite run is mainly used in Section 6.3.

Using the standard automata-based approach for LTL [VW94], we can show the following proposition.

**Proposition 8.** *Let  $\Phi$  be an LTL( $\mathbb{Z}$ ) formulae in simple form. There exists a constraint Büchi automaton  $\mathbb{A}_\Phi$  such that  $\{\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta \mid \mathbf{w} \models \Phi\} = L(\mathbb{A}_\Phi)$ , and the following conditions hold.*

- (I) *The number of locations in  $\mathbb{A}_\Phi$  is bounded by  $\text{size}(\Phi) \times 2^{2 \times \text{size}(\Phi)}$ .*
- (II) *The cardinality of  $\delta$  in  $\mathbb{A}_\Phi$  is exponential in  $\text{size}(\Phi)$ .*
- (III) *The maximal size of a constraint in  $\mathbb{A}_\Phi$  is quadratic in  $\text{size}(\Phi)$ .*

In the proof of Proposition 8, the construction of  $\mathbb{A}_\Phi$  is similar to the construction from CTL( $\mathbb{Z}$ ) formulae by imposing  $D = 1$  and by disqualifying the notion of direction map because there is a single direction.

*Proof.* We use the standard automata-based approach for LTL [VW94], except that we have to deal with constraints. Let  $\Phi$  be an LTL( $\mathbb{Z}$ ) in simple form. We provide below usual notations to define the automaton  $\mathbb{A}_\Phi$ . We write  $\text{sub}(\Phi)$  to denote the smallest set such that

- $\Phi \in \text{sub}(\Phi)$ ;  $\text{sub}(\Phi)$  is closed under subformulae,
- for all  $\text{Op} \in \{\cup, \cap, \mathbf{X}\}$ , if  $\Phi_1 \text{ Op } \Phi_2 \in \text{sub}(\Phi)$ , then  $\mathbf{X}(\Phi_1 \text{ Op } \Phi_2) \in \text{sub}(\Phi)$ .

The cardinality of  $\text{sub}(\Phi)$  is at most twice the number of subformulae of  $\Phi$ . Given  $X \subseteq \text{sub}(\Phi)$ ,  $X$  is *propositionally consistent*  $\stackrel{\text{def}}{\iff}$  the conditions below hold.

- If  $\Phi_1 \vee \Phi_2 \in X$ , then  $\{\Phi_1, \Phi_2\} \cap X \neq \emptyset$ ; if  $\Phi_1 \wedge \Phi_2 \in X$ , then  $\{\Phi_1, \Phi_2\} \subseteq X$ .
- If  $\Phi_1 \cup \Phi_2 \in X$ , then  $\Phi_2 \in X$  or  $\{\Phi_1, \mathbf{X}(\Phi_1 \cup \Phi_2)\} \subseteq X$ .
- If  $\Phi_1 \cap \Phi_2 \in X$ , then  $\Phi_2 \in X$  and  $\{\Phi_1, \mathbf{X}(\Phi_1 \cap \Phi_2)\} \cap X \neq \emptyset$ .

We write  $\text{sub}_\mathbf{X}(\Phi)$  to denote the set of formulae in  $\text{sub}(\Phi)$  of the form  $\mathbf{X}\Phi'$ . Similarly, we write  $\text{sub}_\cup(\Phi)$  to denote the set of formulae in  $\text{sub}(\Phi)$  of the form  $\Phi_1 \cup \Phi_2$ . Finally, we write  $\text{sub}_{\text{cons}}(\Phi)$  to denote the set of formulae of the form  $\Theta$  in  $\text{sub}(\Phi)$ .

We build a generalised word constraint automaton  $\mathbb{B}_\Phi = (Q, \Sigma, \beta, Q_{\text{in}}, \delta, \mathcal{F})$  such that  $\{\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta \mid \mathbf{w} \models \Phi\} = L(\mathbb{A}_\Phi)$ . The automaton  $\mathbb{B}_\Phi$  accepts infinite words  $\mathbf{w} : \mathbb{N} \rightarrow \Sigma \times \mathbb{Z}^\beta$  with  $\Sigma = \{\dagger\}$ . Let us define  $\mathbb{B}_\Phi$  formally.

- $\Sigma \stackrel{\text{def}}{=} \{\dagger\}$ ;  $Q \subseteq \mathcal{P}(\text{sub}(\Phi))$  contains the propositionally consistent sets.
- $Q_{\text{in}} \stackrel{\text{def}}{=} \{X \in Q \mid \Phi \in X\}$ .
- The transition relation  $\delta$  is made of tuples of the form  $(X, \dagger, \Theta, X')$ , verifying the conditions below.
  1. For all  $X\Phi' \in X$ , we have  $\Phi' \in X'$ .
  2.  $\Theta$  is equal to  $(\bigwedge_{\Theta' \in \text{sub}_{\text{cons}}(\Phi) \cap X} \Theta')$ .
- $\mathcal{F}$  is made of the sets  $F_{\Psi_1 \cup \Psi_2}$  with  $\Psi_1 \cup \Psi_2 \in \text{sub}_{\cup}(\Phi)$  with  $F_{\Psi_1 \cup \Psi_2} \stackrel{\text{def}}{=} \{X \mid \Psi_2 \in X \text{ or } \Psi_1 \cup \Psi_2 \notin X\}$ .

Transforming generalised Büchi conditions to standard Büchi conditions leads to a set of locations multiplied by the factor  $\text{card}(\mathcal{F})+1$  (which is bounded by  $\text{size}(\Phi)$ ) and to the word constraint automaton  $\mathbb{A}_{\Phi}$ . Moreover, one can check that the number of transitions in  $\mathbb{A}_{\Phi}$  is exponential in  $\text{size}(\Phi)$ . We omit the standard proof for correctness.  $\square$

The automaton  $\mathbb{A}_{\Phi}$  may not be deterministic; for instance, for  $\Phi$  equal to  $\text{FG } \mathbf{x} = \mathbf{x}'$  there cannot exist a *deterministic* constraint Büchi automaton such that  $\mathbf{w} \models \Phi$  if, and only if,  $\mathbf{w} \in L(\mathbb{A}_{\Phi})$ . In Section 6.3, we prove that by appropriately adapting Safra's construction [Saf89, Chapter 1] we can construct, for every nondeterministic constraint Büchi automaton  $\mathbb{A}$  a deterministic constraint Rabin automaton  $\mathbb{A}'$  such that  $L(\mathbb{A}) = L(\mathbb{A}')$ . This will be key in Section 6.4, where we prove how to construct from a  $\text{CTL}^*(\mathbb{Z})$  formula  $\phi$  a tree constraint (Rabin) automaton  $\mathbb{A}_{\phi}$  such that  $\mathbf{t} \models \phi$  if, and only if,  $L(\mathbb{A}_{\phi}) \neq \emptyset$ .

### 6.3 Determinisation of word constraint automata

Formulae in special form (see Section 6.1) may contain conjuncts of the form  $\mathbf{A} \Phi$ , where  $\Phi$  is an  $\text{LTL}(\mathbb{Z})$  formula in simple form. In order to define a tree constraint automaton accepting trees such that all branches satisfy  $\Phi$ , the usual approach consists in designing *deterministic* word automata accepting  $\omega$ -sequences satisfying  $\Phi$  and using the transitions of the word automata to define the transitions of the tree automata. A well-known construction to transform nondeterministic Büchi automata to equivalent deterministic Rabin automata is due to S. Safra [Saf89, Theorem 1.1], which is a building block to decide the satisfiability status of  $\text{CTL}^*$  formulae with optimal worst-case computational complexity. In this section, we show that it is possible to lift this construction to word constraint automata so that eventually, the Büchi word constraint automata designed from  $\text{LTL}(\mathbb{Z})$  formulae in Section 6.2 are transformed into equivalent deterministic Rabin word constraint automata. This allows us to design Rabin tree constraint automata for  $\mathbf{A} \Phi$  (see Section 6.4). The developments below are adapted from [Saf89, Chapter 1] to handle constraints in the transitions. A special attention is given to the cardinality of the transition relation and to the size of the constraints in transitions, as these two parameters are, *a priori*, unbounded in constraint automata but are essential to perform a forthcoming complexity analysis.

In the sequel,  $\mathbb{A} = (Q, \Sigma, \beta, Q_{\text{in}}, \delta, F)$  is a (nondeterministic) Büchi word constraint automaton with  $F \subseteq Q$ . We are going to define a deterministic Rabin word constraint automaton  $\mathbb{A}'$  such that  $L(\mathbb{A}) = L(\mathbb{A}')$ .

**Safra Trees** A *Safra tree* over  $Q$  is a finite tree  $\mathbf{s}$ , satisfying the following conditions.

1.  $\mathbf{s}$  is ordered, that is, if a node in the tree has children nodes, then there is a first child node, a second child node etc. In other words, given two sibling nodes, it is uniquely determined which of the two nodes is younger than the other.
2. Every node in  $\mathbf{s}$  has a unique name from  $[1, 2 \cdot \text{card}(Q)]$ , no two nodes have the same name.

3. Every node is labelled with some element in  $\mathcal{P}(Q) \setminus \{\emptyset\}$ . We use  $\text{Lab}(\mathbf{s}, J)$  to denote the set of labels of a node with name  $J$  in  $\mathbf{s}$ .
4. The label of a node is a proper superset of the union of the labels of its children nodes.
5. Two nodes with the same parent node have disjoint labels.
6. Every node is either marked or unmarked.

**Lemma 13.** *A Safra tree over  $Q$  has at most  $\text{card}(Q)$  nodes.*

*Proof.* Let  $\mathbf{s}$  be a Safra tree over  $Q$ . If  $\mathbf{s}$  has no nodes, the claim is of course true. So let us assume that  $\mathbf{s}$  contains at least one node. We prove the claim by induction on the height  $H$  of  $\mathbf{s}$ . For the induction base, let  $H = 1$ . The tree then has exactly one node, namely the root node, and the claim is trivially true. So suppose the claim holds for  $H \geq 1$ . We prove the claim for  $H + 1$ . Suppose the root node of  $\mathbf{s}$  has  $k$  children, denoted by  $\mathbf{n}_1, \dots, \mathbf{n}_k$ . For every  $1 \leq i \leq k$ , let  $Q_i \subseteq Q$  denote the label of  $\mathbf{n}_i$ . For every  $1 \leq i \leq k$ , the subtree of  $\mathbf{n}_i$  is a Safra tree over  $Q_i$  with depth at most  $H$ . By the induction hypothesis, such a subtree has at most  $\text{card}(Q_i)$  nodes. By condition 5 of Safra trees, the sets  $Q_1, \dots, Q_k$  are pairwise disjoint. By condition 4, the union  $\bigcup_{1 \leq i \leq k} Q_i$  is a proper subset of  $Q$ . Hence  $\sum_{i=1}^k \text{card}(Q_i) < \text{card}(Q)$ . Altogether, the number of nodes in  $\mathbf{s}$  is at most  $1 + \sum_{i=1}^k \text{card}(Q_i) < 1 + \text{card}(Q) \leq \text{card}(Q)$ .  $\square$

**Safra's determinisation construction** Define the deterministic Rabin word constraint automaton  $\mathbb{A}' = (Q', \Sigma, \beta, Q'_{\text{in}}, \delta', \mathcal{F}')$  as follows.

- $Q'$  is the set of all Safra trees over  $Q$ ;
- $Q'_{\text{in}} = \mathbf{s}_{Q_{\text{in}}}$ , where  $\mathbf{s}_{Q_{\text{in}}}$  is the Safra tree with a single unmarked node  $\mathbf{n}$  with name 1 and  $\text{Lab}(\mathbf{s}_{Q_{\text{in}}}, 1) = Q_{\text{in}}$ ;
- The finite transition relation  $\delta' \subseteq (Q' \times \Sigma \times \text{TreeCons}(\beta) \times Q')$  is defined as follows. Recall that  $\text{SatTypes}(\beta)$  denotes the set of all satisfiable complete constraints over  $\mathbf{x}_1, \dots, \mathbf{x}_\beta, \mathbf{x}'_1, \dots, \mathbf{x}'_\beta$  and the transition relation  $\delta'$  is defined over this strict subset of  $\text{TreeCons}(\beta)$ . Let  $\mathbf{s}$  be a Safra tree,  $\mathbf{a} \in \Sigma$ , and  $\Theta \in \text{SatTypes}(\beta)$ . We set  $(\mathbf{s}, \mathbf{a}, \Theta, \mathbf{s}') \in \delta'$ , where  $\mathbf{s}'$  is obtained from  $\mathbf{s}$ ,  $\mathbf{a}$  and  $\Theta$  by applying the following steps.
  1. Unmark all nodes in  $\mathbf{s}$ . Let us use  $\mathbf{s}^{(1)}$  to denote the resulting Safra tree.
  2. For every node with label  $P \subseteq Q$  such that  $P \cap F \neq \emptyset$ , create a new youngest child node with label  $P \cap F$ . The name of this node is picked from the names  $[1, 2 \cdot \text{card}(Q)]$  that are not assigned to any of the other nodes in  $\mathbf{s}$  yet (say, the smallest one to guarantee determinism). Let us use  $\mathbf{s}^{(2)}$  to denote the resulting Safra tree.
  3. Apply the powerset construction to every node of the resulting tree, that is, for every node with label  $P \subseteq Q$ , replace  $P$  by a new label defined by  $\bigcup_{q \in P} \{q' \in Q \mid \text{there is } (q, \mathbf{a}, \Theta', q') \in \delta \text{ such that } \Theta \models \Theta'\}$ . Let us use  $\mathbf{s}^{(3)}$  to denote the resulting Safra tree. Note that  $\Theta \models \Theta'$  can be checked in polynomial-time because  $\Theta \in \text{SatTypes}(\beta)$ . This is the part that differs the most with the construction with finite alphabets.
  4. (Horizontal Merge) For every two nodes with the same parent node and such that  $q \in Q$  is contained in the labels of both nodes, remove  $q$  from the labels of the younger node and all its descendants. Let us use  $\mathbf{s}^{(4)}$  to denote the resulting Safra tree.
  5. Remove all nodes with empty label, except the root node. Let us use  $\mathbf{s}^{(5)}$  to denote the resulting Safra tree.
  6. (Vertical Merge) For every node whose label equals the union of the labels of its children nodes, remove all descendants of this node and mark it. The resulting Safra tree is  $\mathbf{s}'$ .

- Define  $\mathcal{F}' = \{(L_1, U_1), \dots, (L_{2 \cdot \text{card}(Q)}, U_{2 \cdot \text{card}(Q)})\}$ , where for all  $1 \leq J \leq 2 \cdot \text{card}(Q)$ ,
  - $L_J = \{\mathbf{s} \in Q' \mid \mathbf{s} \text{ contains a node with name } J \text{ marked}\}$ ,
  - $U_J = \{\mathbf{s} \in Q' \mid \mathbf{s} \text{ does not contain a node with name } J\}$ .

Before we prove the correctness of this construction (Lemma 17 and Lemma 18), let us prove some simple properties expressed in Lemmas 14–16.

**Lemma 14.** *Let  $(\mathbf{s}, \mathbf{a}, \Theta, \mathbf{s}')$  be a transition in  $\delta'$  such that  $\mathbf{s}'$  contains a node with name  $K$  for some  $1 \leq K \leq 2 \cdot \text{card}(Q)$ . Then this node  $K$  has no children nodes in  $\mathbf{s}'$  if one of the following two conditions hold:*

- $\mathbf{s}$  does not contain a node with name  $K$ , or
- $\mathbf{s}'$  contains the node with name  $K$  marked.

*Proof.* If  $\mathbf{s}$  does not contain a node with the name  $K$ , then the node  $K$  was freshly created in step 2 from  $\mathbf{s}$ . None of the other steps introduces a child node for  $K$ .

If  $\mathbf{s}'$  contains  $K$  marked, i.e., step (6) of  $\delta'$  was applied. That is, in  $\mathbf{s}^{(5)}$  the label of the node  $K$  equals the union of the labels of its children, and  $\mathbf{s}'$  is then obtained from  $\mathbf{s}^{(5)}$  by removing all descendants from  $K$ , hence  $K$  has no children.  $\square$

**Lemma 15.** *Let  $(\mathbf{s}, \mathbf{a}, \Theta, \mathbf{s}')$  be a transition in  $\delta'$  such that there is some name  $1 \leq K \leq 2 \cdot \text{card}(Q)$  and  $\mathbf{s}$  does not contain a node with name  $K$  but  $\mathbf{s}'$  contains a node with name  $K$ . Then there exists some  $1 \leq J \leq 2 \cdot \text{card}(Q)$  such that for all  $q' \in \text{Lab}(\mathbf{s}', K)$  there is some location  $q \in \text{Lab}(\mathbf{s}, J) \cap F$  such that  $(q, \mathbf{a}, \Theta', q')$  be a transition in  $\delta$  with  $\Theta \models \Theta'$  and  $J$  is the parent node of  $K$  in  $\mathbf{s}'$ .*

*Proof.* If  $\mathbf{s}'$  contains a node with the name  $K$  and  $\mathbf{s}$  does not, then the node with name  $K$  is obtained from  $\mathbf{s}$  by step 2. Suppose  $q' \in \text{Lab}(\mathbf{s}', K)$ . Then  $q' \in \text{Lab}(\mathbf{s}^{(3)}, K)$  because  $\text{Lab}(\mathbf{s}', K) \subseteq \text{Lab}(\mathbf{s}^{(3)}, K)$ . By step 3 of  $\delta'$ , there exists  $q \in \text{Lab}(\mathbf{s}^{(2)}, K)$  such that  $(q, \mathbf{a}, \Theta', q')$  be a transition in  $\delta$  with  $\Theta \models \Theta'$ . But the node  $K$  must be the youngest child node of a node with some name  $J$  satisfying  $\text{Lab}(\mathbf{s}, J) \cap F \neq \emptyset$  and  $\text{Lab}(\mathbf{s}, J) \cap F = \text{Lab}(\mathbf{s}^{(2)}, K)$ .  $\square$

**Lemma 16.** *For all infinite runs  $\rho' = \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3 \dots$  of  $\mathbb{A}'$  on  $(\mathbf{a}_1, \mathbf{z}_1)(\mathbf{a}_2, \mathbf{z}_2) \dots$ , for every  $1 \leq J \leq 2 \cdot \text{card}(Q)$  and for every  $1 \leq j \leq j'$ , if  $\mathbf{s}_k$  contains a node with name  $J$  for all  $j \leq k \leq j'$ , then for every  $q' \in \text{Lab}(\mathbf{s}_{j'}, J)$  there exists some  $q \in \text{Lab}(\mathbf{s}_j, J)$  and some run  $q_j, q_{j+1} \dots q_{j'}$  of  $\mathbb{A}$  on  $(\mathbf{a}_j, \mathbf{z}_j)(\mathbf{a}_{j+1}, \mathbf{z}_{j+1}) \dots (\mathbf{a}_{j'}, \mathbf{z}_{j'})$  with  $q_j = q$  and  $q_{j'} = q'$ .*

*Proof.* The proof is by induction on  $\Delta = j' - j$ . For the induction base, let  $\Delta = 0$ , that is,  $j' = j$ . Suppose  $q' \in \text{Lab}(\mathbf{s}_{j'}, J) = \text{Lab}(\mathbf{s}_j, J)$ . By definition of finite runs,  $q'$  is a finite run of  $\mathbb{A}$  on  $(\mathbf{a}_j, \mathbf{z}_j)$ . Suppose  $\Delta \geq 0$  and the claim holds for  $\Delta$ . We prove it for  $\Delta + 1$ . So suppose  $q' \in \text{Lab}(\mathbf{s}_{j'}, J)$ . Let  $(\mathbf{s}_{j'-1}, \mathbf{a}_{j'-1}, \Theta_{j'-1}, \mathbf{s}_{j'}) \in \delta'$  be the transition used in the infinite run  $\rho'$ , where  $\Theta_{j'-1} \in \text{SatTypes}(\beta)$ . By step 3 of the definition of  $\delta'$ , there exists some  $q'' \in \text{Lab}(\mathbf{s}_{j'-1}, J)$  and some transition  $(q'', \mathbf{a}_{j'-1}, \Theta'_{j'-1}, q') \in \delta$  such that  $\Theta_{j'-1} \models \Theta'_{j'-1}$ . From  $\rho'$  being a run, we obtain that  $\mathbb{Z} \models \Theta_{j'-1}(\mathbf{z}_{j'-1}, \mathbf{z}_{j'})$ . We can infer that  $\mathbb{Z} \models \Theta'_{j'-1}(\mathbf{z}_{j'-1}, \mathbf{z}_{j'})$ , too. By the induction hypothesis, there exists some  $q \in \text{Lab}(\mathbf{s}_j, J)$  and some run  $q_j, \dots, q_{j'-1}$  of  $\mathbb{A}$  on  $(\mathbf{a}_j, \mathbf{z}_j) \dots (\mathbf{a}_{j'-1}, \mathbf{z}_{j'-1})$  such that  $q_j = q$  and  $q_{j'-1} = q''$ . Altogether, we obtain a run  $q_j, \dots, q_{j'-1}, q_{j'}$  of  $\mathbb{A}$  on  $(\mathbf{a}_j, \mathbf{z}_j) \dots (\mathbf{a}_{j'}, \mathbf{z}_{j'})$  such that  $q_j = q$  and  $q_{j'} = q'$ .  $\square$

For the correctness of the construction of  $\mathbb{A}'$ , we first prove the result below.

**Lemma 17.**  $L(\mathbb{A}) \subseteq L(\mathbb{A}')$ .

*Proof.* Let  $\mathbf{w} = (\mathbf{a}_1, \mathbf{z}_1)(\mathbf{a}_2, \mathbf{z}_2)(\mathbf{a}_2, \mathbf{z}_2) \dots$  be an infinite word over  $\Sigma \times \mathbb{Z}^\beta$  such that  $\mathbf{w} \in L(\mathbb{A})$ . Then there exists some initialized Büchi accepting run  $\rho$  of the form  $q_1 \xrightarrow{(\mathbf{a}_1, \mathbf{z}_1)} q_2 \xrightarrow{(\mathbf{a}_2, \mathbf{z}_2)} \dots$  of  $\mathbb{A}$  on  $\mathbf{w}$ . That is,  $q_1 \in Q_{\text{in}}$ , for every  $i \geq 1$  there exists a transition  $(q_i, \mathbf{a}_i, \Theta'_i, q_{i+1}) \in \delta$  such that  $\mathbb{Z} \models \Theta'_i(\mathbf{z}_i, \mathbf{z}_{i+1})$ , and there exists some  $q \in F$  that appears infinitely often in  $\rho$ . Let us fix such an accepting state and denote it by  $q_{\text{acc}}$ .

We are going to construct an infinite Rabin-accepting run of  $\mathbb{A}'$  on  $\mathbf{w}$ . Set  $\mathbf{s}_1$  to be the Safra tree  $\mathbf{s}_{Q_{\text{in}}}$ , that is, the Safra tree with a single node with name 1 and  $\text{Lab}(\mathbf{s}_{Q_{\text{in}}}, 1) = Q_{\text{in}}$ . We prove that for all  $i \geq 1$ , if  $\mathbf{s}_i$  has root node with name 1 and  $q_i \in \text{Lab}(\mathbf{s}_i, 1)$ , then there exists a unique constraint  $\Theta_i \in \text{SatTypes}(\beta)$  and a unique Safra tree  $\mathbf{s}_{i+1}$  such that

- $(\mathbf{s}_i, \mathbf{a}_i, \Theta_i, \mathbf{s}_{i+1}) \in \delta'$
- $\mathbf{s}_{i+1}$  has root node with name 1 and  $q_{i+1} \in \text{Lab}(\mathbf{s}_{i+1}, 1)$ , and
- $\mathbb{Z} \models \Theta_i(\mathbf{z}_i, \mathbf{z}_{i+1})$ .

So let  $i \geq 1$ . Let  $\Theta_i$  be the unique constraint in  $\text{SatTypes}(\beta)$  such that  $\mathbb{Z} \models \Theta_i(\mathbf{z}_i, \mathbf{z}_{i+1})$  (uniqueness and existence guaranteed by Lemma 3). By definition of  $\delta'$ , there exists a unique Safra tree  $\mathbf{s}_{i+1}$  such that  $(\mathbf{s}_i, \mathbf{a}_i, \Theta_i, \mathbf{s}_{i+1}) \in \delta'$ . We prove that the second condition holds for  $\mathbf{s}_{i+1}$ . So recall how  $\mathbf{s}_{i+1}$  is obtained from  $\mathbf{s}_i$ . By assumption,  $\mathbf{s}_i$  has root node with name 1 and  $q_i \in \text{Lab}(\mathbf{s}_i, 1)$ . Note that  $q_i \in \text{Lab}(\mathbf{s}_i^{(2)}, 1)$ . We prove  $q_{i+1} \in Y$ , where  $Y = \{q' \in Q \mid (q_i, \mathbf{a}_i, \Theta'_i, q') \in \delta, \Theta_i \models \Theta'_i\}$ . Recall that  $(q_i, \mathbf{a}_i, \Theta'_i, q_{i+1}) \in \delta$  and  $\mathbb{Z} \models \Theta'_i(\mathbf{z}_i, \mathbf{z}_{i+1})$ . Hence  $\Theta_i \models \Theta'_i$  (by Lemma 3) and  $q_{i+1} \in Y$ . But then also  $q_{i+1} \in \text{Lab}(\mathbf{s}_i^{(3)}, 1)$ . Further,  $q_{i+1} \in \text{Lab}(\mathbf{s}_i^{(4)}, 1)$  because the root node 1 has no siblings, so step 4 will not remove the root node;  $q_{i+1} \in \text{Lab}(\mathbf{s}_i^{(5)}, 1)$  as the label set of the root node is non-empty, so step 5 will not remove the root node; finally  $q_{i+1} \in \text{Lab}(\mathbf{s}_{i+1}, 1)$  because step 6 may only remove descendants of the root node, but not the root node itself.

We can conclude that there exists indeed a (unique) run  $\rho'$  of the form  $\mathbf{s}_1 \xrightarrow{\mathbf{a}_1, \mathbf{z}_1} \mathbf{s}_2 \xrightarrow{\mathbf{a}_2, \mathbf{z}_2} \dots$  of  $\mathbb{A}'$  on  $\mathbf{w}$ . Note that  $\mathbf{s}_1 \in Q'_{\text{in}}$ . In order to prove that  $\rho'$  is Rabin accepting, we prove that there exists some  $1 \leq J \leq 2 \cdot \text{card}(Q)$  such that  $\rho'$  contains infinitely many Safra trees that contain a node with name  $J$  marked, and only finitely many Safra trees that do not contain some node with name  $J$ . For this, we prove the following.

For every position  $i \geq 1$  and every name  $1 \leq J \leq 2 \cdot \text{card}(Q)$ , if for all  $k \geq i$ , the Safra tree  $\mathbf{s}_k$  contains a node with name  $J$  and  $q_k \in \text{Lab}(\mathbf{s}_k, J)$ , then one of the following two properties holds:

1. There exist infinitely many  $k \geq i$  such that  $\mathbf{s}_k$  contains the node with name  $J$  marked.
2. There exists some position  $i' \geq i$  and some name  $1 \leq J' \leq 2 \cdot \text{card}(Q)$  with  $J \neq J'$  such that for all  $k \geq i$ , the node with name  $J$  has a child node with name  $J'$  and  $q_k \in \text{Lab}(\mathbf{s}_k, J')$ .

For the proof, suppose  $i \geq 1$ ,  $1 \leq J \leq 2 \cdot \text{card}(Q)$ , and for all  $k \geq i$ ,  $\mathbf{s}_k$  contains a node with name  $J$  and  $q_k \in \text{Lab}(\mathbf{s}_k, J)$ . If the first property holds, we are done. Otherwise, there exists some position  $m \geq i$  such that

- for all  $k \geq m$ ,  $\mathbf{s}_k$  contains the node with name  $J$  unmarked, and
- $q_m = q_{\text{acc}}$ , and hence  $q_{\text{acc}} \in \text{Lab}(\mathbf{s}_m, J)$ .

Using the definition of  $\delta'$ , it is easy to prove that in  $\mathbf{s}_{m+1}$ , the node with name  $J$  has a child node with name  $M \neq J$  such that  $q_{m+1} \in \text{Lab}(\mathbf{s}_{m+1}, M)$ . If for all  $k \geq m+1$ ,  $\mathbf{s}_k$  contains a node with name  $M$  and  $q_k \in \text{Lab}(\mathbf{s}_k, M)$ , we are done. Otherwise, there must exist some position  $p > m+1$  such that

- $\mathbf{s}_k$  contains a node with name  $M$  and  $q_k \in \text{Lab}(\mathbf{s}_k, M)$  for all  $m+1 \leq k < p$ , and

- $\mathbf{s}_p$  does not contain a node with name  $M$ , or  $q_p \notin \text{Lab}(\mathbf{s}_p, M)$ .

By definition of  $\delta'$ , there are three cases: during the construction of  $\mathbf{s}_p$  out of  $\mathbf{s}_{p-1}$ ,

- (a) the state  $q_p$  is removed from the label of the node with name  $M$ , because there exists some younger sibling of  $M$  (that is, an older child of  $J$ ) whose label set contains  $q_p$ .
- (b) the node with name  $M$  has been removed from the Safra tree during step 5. But for this, the node with the name  $M$  must have an empty label set, contradicting  $q_{p-1} \in \text{Lab}(\mathbf{s}_{p-1}, M)$  and there is  $(q_{p-1}, \mathbf{a}_{p-1}, \Theta'_{p-1}, q_p) \in \delta$  such that  $\mathbb{Z} \models \Theta'_{p-1}(z_{p-1}, z_p)$  – so this case cannot occur. In the sequel, the satisfaction of  $(q_{p-1}, \mathbf{a}_{p-1}, \Theta'_{p-1}, q_p) \in \delta$  and  $\mathbb{Z} \models \Theta'_{p-1}(z_{p-1}, z_p)$  is denoted by the expression  $q_{p-1} \xrightarrow{\mathbf{a}_{p-1}, z_{p-1}} q_p$ . We adopt a similar with the automaton  $\mathbb{A}'$ .
- (c) the node with name  $M$  has been removed from the Safra tree during step 6. But for this the parent node with name  $J$  must be marked, contradiction – so this case cannot occur.

Note that case (a) can only occur at most  $\text{card}(Q) - 1$  times, as by Lemma 13, the node with name  $J$  can have at most  $\text{card}(Q) - 1$  children nodes. We can conclude that there must exist some position  $i' \geq i$  and some name  $1 \leq J' \leq 2 \cdot \text{card}(Q)$  with  $J \neq J'$  such that for all  $k \geq i'$ ,  $\mathbf{s}_k$  contains a node with name  $J'$  and  $q_k \in \text{Lab}(\mathbf{s}_k, J')$ .

Finally, recall that for every  $k \geq 0$ , the Safra tree  $\mathbf{s}_k$  contains a node with name 1 and  $q_k \in \text{Lab}(\mathbf{s}_k, 1)$ . Set  $i = 1$  and  $J = 1$ . We distinguish the following two cases.

**Case 1** There exist infinitely many  $k \geq i$  such that  $\mathbf{s}_k$  contains node with name  $J$  marked, that is,  $\mathbf{s}_k \in L_J$ . This implies that for only finitely many  $k \geq 0$  the Safra tree  $\mathbf{s}_k$  does not contain the node with name  $J$ , that is,  $\mathbf{s}_k \in U_J$ . Hence, the run  $\rho'$  is Rabin accepting.

**Case 2** Otherwise, by what we proved above, there exists some  $i' \geq i$  and some  $1 \leq J' \leq 2 \cdot \text{card}(Q)$  with  $J \neq J'$  such that for all  $k \geq i'$ , the node with name  $J$  has a child with name  $J'$ , and  $q_k \in \text{Lab}(\mathbf{s}_k, J')$ . We can now repeat the same case distinction, this time for  $i = i'$  and  $J = J'$ .

Note that after at most  $\text{card}(Q)$  steps, **Case 1** must necessarily be true, as by Lemma 13, every Safra tree has at most  $\text{card}(Q)$  nodes.  $\square$

The second part for correctness is proved below.

**Lemma 18.**  $L(\mathbb{A}') \subseteq L(\mathbb{A})$ .

*Proof.* Let  $\mathbf{w} = (\mathbf{a}_1, \mathbf{z}_1)(\mathbf{a}_2, \mathbf{z}_2)(\mathbf{a}_3, \mathbf{z}_3) \dots$  be an infinite word over  $\Sigma \times \mathbb{Z}^\beta$  such that  $\mathbf{w} \in L(\mathbb{A}')$ . Then there exists some initialized Rabin accepting run  $\rho'$  of the form  $\mathbf{s}_1 \xrightarrow{(\mathbf{a}_1, \mathbf{z}_1)} \mathbf{s}_2 \xrightarrow{(\mathbf{a}_2, \mathbf{z}_2)} \dots$  of  $\mathbb{A}'$  on  $\mathbf{w}$ . That is,  $\mathbf{s}_1 = \mathbf{s}_{Q_{\text{in}}}$ , there exists a unique  $\Theta_i \in \text{SatTypes}(\beta)$  such that  $(\mathbf{s}_i, \mathbf{a}_i, \Theta_i, \mathbf{s}_{i+1}) \in \delta'$  and  $\mathbb{Z} \models \Theta_i(z_i, z_{i+1})$  for all  $i \geq 1$ , and there exists some  $1 \leq J \leq 2 \cdot \text{card}(Q)$  such that  $\mathbf{s}_i \in L_J$  for infinitely many  $i \geq 0$  and  $\mathbf{s}_i \in U_J$  for only finitely many  $i \geq 0$ . Fix such a name  $1 \leq J \leq 2 \cdot \text{card}(Q)$ . Then, there must exist some position  $i \geq 0$  such that  $\mathbf{s}_k$  contains a node with name  $J$ , for all  $k \geq i$ . We let  $i_0$  be the minimal such position. Note that  $J$  in  $\mathbf{s}_{i_0}$  cannot be marked. Let  $i_0 < i_1 < i_2 < i_3 \dots$  be the infinitely many positions greater than  $i_0$  such that  $\mathbf{s}_{i_k}$  contains the node with name  $J$  marked, for all  $k \geq 1$ .

1. For every  $j \geq 1$  and for every  $q \in \text{Lab}(\mathbf{s}_{i_j}, J)$ , we define the set  $\text{Acc}(q, j)$  by

$$\text{Acc}(q, j) \stackrel{\text{def}}{=} \{q' \in F \mid \exists k \in [i_{j-1}, i_j - 1] \text{ such that } q' \in \text{Lab}(\mathbf{s}_k, J) \text{ and} \\ \exists \text{ run of } \mathbb{A} \text{ from } q' \text{ to } q \text{ on } (\mathbf{a}_k, \mathbf{z}_k)(\mathbf{a}_{k+1}, \mathbf{z}_{k+1}) \dots (\mathbf{a}_{i_j}, \mathbf{z}_{i_j})\}.$$

We prove that  $\text{Acc}(q, j) \neq \emptyset$ . We distinguish two cases.

- (a) First suppose there exists  $q' \in \text{Lab}(\mathbf{s}_{i_j-1}, J) \cap F$  such that  $q' \xrightarrow{\mathbf{a}_{i_j-1}, \mathbf{z}_{i_j-1}} q$ . Then we are done.
- (b) Second, suppose there does not exist  $q' \in \text{Lab}(\mathbf{s}_{i_j-1}, J) \cap F$  such that  $q' \xrightarrow{\mathbf{a}_{i_j-1}, \mathbf{z}_{i_j-1}} q$ . That is, for all  $q' \in \text{Lab}(\mathbf{s}_{i_j-1}, J)$  with  $q' \xrightarrow{\mathbf{a}_{i_j-1}, \mathbf{z}_{i_j-1}} q$  we have  $q' \notin F$ .

By definition of  $\delta'$ ,  $\text{Lab}(\mathbf{s}_{i_j}, J) = \text{Lab}((\mathbf{s}_{i_j-1})^{(5)}, J)$ . Since  $J$  is marked in  $\mathbf{s}_{i_j}$ , we also know that  $J$  must have children in  $(\mathbf{s}_{i_j-1})^{(5)}$ , and the labels of  $J$  equal the union of the labels of its children nodes. Thus,  $q \in \text{Lab}((\mathbf{s}_{i_j-1})^{(5)}, K)$ , where  $K$  is the name of a child node of  $J$  in  $(\mathbf{s}_{i_j-1})^{(5)}$ .

Then there exists  $q' \in \text{Lab}((\mathbf{s}_{i_j-1})^{(2)}, K)$  such that  $q' \xrightarrow{\mathbf{a}_{i_j-1}, \mathbf{z}_{i_j-1}} q$ . By property 4 of Safra trees, we know that  $q' \in \text{Lab}((\mathbf{s}_{i_j-1})^{(2)}, J)$ . This also implies  $q' \in \text{Lab}(\mathbf{s}_{i_j-1}, J)$ , and by assumption,  $q' \notin F$ . This implies that  $K$  is already a node in  $\mathbf{s}_{i_j-1}$  (it was not created as a youngest child of  $J$  by step 2 of  $\delta'$ ).

Let  $i_{j-1} < k < i_j - 1$  be the minimal position such that node  $J$  has no child node with name  $K$  in  $\mathbf{s}_{k-1}$ , and node  $J$  has a child node with name  $K$  in  $\mathbf{s}_{k'}$  for all  $k \leq k' \leq i_j - 1$ . By Lemma 14, such a position necessarily exists, as node  $J$  in  $\mathbf{s}_{i_j-1}$  has no children nodes and hence cannot contain the node with name  $K$ . By Lemma 16, there exists a state  $q_k \in \text{Lab}(\mathbf{s}_k, K)$  and some run  $q_k, q_{k+1}, \dots, q_{i_j-1}$  on  $(\mathbf{a}_k, \mathbf{z}_k) \dots (\mathbf{a}_{i_j-1}, \mathbf{z}_{i_j-1})$ , where  $q_{i_j-1} = q'$ . By Lemma 15, there exists  $q_{k-1} \in \text{Lab}(\mathbf{s}_{k-1}, J) \cap F$  such that  $q_{k-1} \xrightarrow{\mathbf{a}_{k-1}, \mathbf{z}_{k-1}} q_k$ . Done.

2. For every  $j \geq 2$  and for every  $q \in F$  such that  $q \in \text{Lab}(\mathbf{s}_k, J)$  for some  $i_{j-1} \leq k \leq i_j - 1$ , we use  $\text{Pre}(q, j)$  to denote the set defined by

$$\text{Pre}(q, j) \stackrel{\text{def}}{=} \{q' \in \text{Lab}(\mathbf{s}_{i_{j-1}}, J) \mid \exists \text{ run from } q' \text{ to } q \text{ of } \mathbb{A} \text{ on } (\mathbf{a}_{i_{j-1}}, \mathbf{z}_{i_{j-1}}) \dots (\mathbf{a}_{i_j-1}, \mathbf{z}_{i_j-1})\}.$$

It follows from Lemma 16 that  $\text{Pre}(q, j) \neq \emptyset$ .

We define an infinite tree  $\mathbf{t}$  with identifiers in  $\{Q_{\text{in}}\} \cup \{(q, \text{acc}, j) \mid q \in Q, j \geq 1\} \cup \{(q, j) \mid q \in Q, j \geq 1\}$ . (No two nodes have the same identifiers.)

1. The root node of  $\mathbf{t}$  is labelled with  $Q_{\text{in}}$ .
2. The root node is parent of a node iff the identifier of this node is  $(q, \text{acc}, 1)$  for some  $q \in \text{Lab}(\mathbf{s}_k, J) \cap F$ , where  $i_0 \leq k < i_1$ .
3. For all  $j \geq 1$  and for all  $q' \in \text{Lab}(\mathbf{s}_k, J) \cap F$ , where  $i_{j-1} \leq k < i_j$ , the node with identifier  $(q', \text{acc}, j)$  is parent of a node iff the identifier of this node is  $(q, j)$  for some  $q \in \text{Lab}(\mathbf{s}_{i_j}, J)$  satisfying  $q' \in \text{Acc}(q, j)$ .
4. For all  $j \geq 2$  and for all  $q' \in \text{Lab}(\mathbf{s}_{i_j}, J)$ , the node with identifier  $(q', j)$  is parent of a node iff the identifier of this node is  $(q, \text{acc}, j)$  for some  $q \in \text{Lab}(\mathbf{s}_k, J) \cap F$ , where  $i_{j-1} \leq k < i_j$ , satisfying  $q' \in \text{Pre}(q, j)$ .

Note that every node has at least one child (we have proved above that the sets  $\text{Pre}$  and  $\text{Acc}$  are non-empty), and every node has at most  $\text{card}(Q)$  children. In particular, this infinite tree is finite branching. By König's Lemma, there must be some infinite path from the root node of the tree, let us say, of the form  $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2 \dots$ . By construction, we have

- $\mathbf{n}_0 = Q_{\text{in}}$ ,
- $\mathbf{n}_1 = (p_1, \text{acc}, 1)$  for some  $p_1 \in \text{Lab}(\mathbf{s}_k, J) \cap F$ ,  $i_0 \leq k < i_1$ ,
- $\mathbf{n}_2 = (p_2, 1)$  for some  $p_2 \in \text{Lab}(\mathbf{s}_{i_1}, J)$  such that  $p_1 \in \text{Acc}(p_2, 1)$ ,
- $\mathbf{n}_3 = (p_3, \text{acc}, 2)$  for some  $p_3 \in \text{Lab}(\mathbf{s}_k, J) \cap F$ ,  $i_1 \leq k < i_2$ , such that  $p_2 \in \text{Pre}(p_3, 2)$ ,

- $\mathbf{n}_4 = (p_4, 2)$  for some  $q_4 \in \text{Lab}(\mathbf{s}_{i_2}, J)$  such that  $p_3 \in \text{Acc}(p_4, 2)$ ,
- etc.

We prove that this yields an infinite run of  $\mathbb{A}$  on  $\mathbf{w}$ . This partially follows from the definitions of  $\text{Acc}$  and  $\text{Pre}$ . For instance,  $p_1 \in \text{Acc}(p_2, 1)$  implies that there exists a run from  $p_1$  to  $p_2$  of  $\mathbb{A}$  on  $(\mathbf{a}_k, \mathbf{z}_k) \dots (\mathbf{a}_{i_1}, \mathbf{z}_{i_1})$ . And from  $p_2 \in \text{Pre}(p_3, 2)$  we can infer that there exists a run from  $p_2$  to  $p_3$  of  $\mathbb{A}$  on  $(\mathbf{a}_{i_1}, \mathbf{z}_{i_1}) \dots (\mathbf{a}_{i_2}, \mathbf{z}_{i_2})$ . Putting the pieces together, we obtain an infinite run starting from  $p_1$ . For obtaining the prefix run from  $q_{\text{in}} \in Q_{\text{in}}$  to  $p_1$  of  $\mathbb{A}$  on  $(\mathbf{a}_1, \mathbf{z}_1) \dots (\mathbf{a}_{i_1-1}, \mathbf{z}_{i_1-1})$ , we can use Lemma 16. Also note that the run is Büchi-accepting, as  $p_i \in F$  for every odd  $i \geq 1$ . Hence  $\mathbf{w} \in L(\mathbb{A})$ .  $\square$

Based on the above developments, we get the following result extending [Saf89, Theorem 1].

**Theorem 7.** *Let  $\mathbb{A} = (Q, \Sigma, \beta, Q_{\text{in}}, \delta, F)$  be a Büchi word constraint automaton involving the constants  $\mathfrak{d}_1, \dots, \mathfrak{d}_\alpha$ . There is a deterministic Rabin word constraint automaton  $\mathbb{A}' = (Q', \Sigma, \beta, Q'_{\text{in}}, \delta', F)$  such that  $L(\mathbb{A}) = L(\mathbb{A}')$  verifying the following quantitative properties.*

- (I) *card( $Q'$ ) is exponential in card( $Q$ ) and the number of Rabin pairs in  $\mathbb{A}'$  is bounded by  $2 \cdot \text{card}(Q)$  (same bounds as in [Saf89, Theorem 1.1]).*
- (II) *The constraints in the transitions are from  $\text{SatTypes}(\beta)$ , hence of size cubic in  $\beta + \max(\lceil \log(|\mathfrak{d}_1|) \rceil, \lceil \log(|\mathfrak{d}_\alpha|) \rceil)$ .*  
Moreover,

$$\text{card}(\delta') \leq \text{card}(Q')^2 \times \text{card}(\Sigma) \times ((\mathfrak{d}_\alpha - \mathfrak{d}_1) + 3)^{2\beta} \times 3^{\beta^2}.$$

We recall that  $\text{card}(\text{SatTypes}(\beta)) \leq ((\mathfrak{d}_\alpha - \mathfrak{d}_1) + 3)^{2\beta} \times 3^{\beta^2}$ . To define  $\mathbb{A}'$ , in the constraints, we need to know how the variables are compared to the constants, but not necessarily to determine the equality with a value when the variable is strictly between  $\mathfrak{d}_i$  and  $\mathfrak{d}_{i+1}$  for some  $i$ . So the above bound can be certainly improved but it is handful to use the set  $\text{SatTypes}(\beta)$  already defined in this document. We get the following corollary based on Proposition 8 and Theorem 7.

**Corollary 1.** *Let  $\Phi$  be an  $\text{LTL}(\mathbb{Z})$  formulae in simple form built over the variables  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$  and the constants  $\mathfrak{d}_1, \dots, \mathfrak{d}_\alpha$ . There exists a deterministic Rabin word constraint automaton  $\mathbb{A}_\Phi$  such that  $\{\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta \mid \mathbf{w} \models \Phi\} = L(\mathbb{A}_\Phi)$ , and the following conditions hold.*

- (I) *The number of locations in  $\mathbb{A}_\Phi$  is bounded by  $2^{2^{p^R(\text{size}(\Phi))}}$  for some polynomial  $p^R(\cdot)$ .*
- (II) *The number Rabin pairs is bounded by  $2 \times \text{size}(\Phi) \times 2^{2 \times \text{size}(\Phi)}$ .*
- (III) *The cardinality of  $\delta$  in  $\mathbb{A}_\Phi$  is bounded by  $\text{card}(\text{SatTypes}(\beta)) \times 2^{2^{p^R(\text{size}(\Phi))} + 1}$ .*
- (IV) *The maximal size of a constraint is cubic in  $\beta + \max(\lceil \log(|\mathfrak{d}_1|) \rceil, \lceil \log(|\mathfrak{d}_\alpha|) \rceil)$ , i.e. polynomial in  $\text{size}(\Phi)$ .*

## 6.4 Rabin tree constraint automata for specific formulae

We recall that  $\text{CTL}^*(\mathbb{Z})$  formulae  $\phi$  in special form are of the form below:

$$E(\mathbf{x} = 0) \wedge \left( \bigwedge_{i \in [1, D-1]} \text{AGE } \Phi_i \right) \wedge \left( \bigwedge_{j \in [1, D']} A \Phi'_j \right),$$

where the  $\Phi_i$ 's and the  $\Phi'_j$ 's are  $\text{LTL}(\mathbb{Z})$  formulae in simple form ( $D \geq 1$ ,  $D' \geq 0$ ). Below, we take advantage of Proposition 8 and Corollary 1 to define tree constraint automata that accept trees that satisfy  $\text{AGE } \Phi_i$  and  $A \Phi'_j$ , respectively.



Given a tree  $\mathbf{t} : [0, D-1] \rightarrow \mathbb{Z}^\beta$  with  $\mathbf{t} \models \text{AGE } \Phi$ , we say that  $\mathbf{t}$  *satisfies AGE  $\Phi$  via the direction  $i$* , for some  $i \in [1, D-1]$ , iff for all nodes  $\mathbf{n} \in [0, D-1]^*$ , we have  $\mathbf{w} \models \Phi$ , where  $\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta$  is defined by  $\mathbf{w}(0) \stackrel{\text{def}}{=} \mathbf{t}(\mathbf{n})$  and  $\mathbf{w}(j) \stackrel{\text{def}}{=} \mathbf{t}(\mathbf{n} \cdot i \cdot 0^{j-1})$  for all  $j \geq 1$ . Proposition 9 below is a counterpart of [ES84, Theorem 3.2] but for  $\text{CTL}^*(\mathbb{Z})$  instead of  $\text{CTL}^*$ , see also the variant [Gas09, Lemma 3.3].

**Proposition 9.** *Let  $\phi$  be a  $\text{CTL}^*(\mathbb{Z})$  formula in special form built over the variables  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$ .  $\phi$  is satisfiable iff there is a tree  $\mathbf{t} : [0, D-1] \rightarrow \mathbb{Z}^\beta$  such that  $\mathbf{t} \models \phi$  and  $\mathbf{t}$  satisfies AGE  $\Phi_i$  via  $i$ , for each  $i \in [1, D-1]$ .*

*Proof.* Assume that  $\phi$  has the form below:

$$\mathbf{E}(\mathbf{x} = 0) \wedge \left( \bigwedge_{i \in [1, D-1]} \text{AGE } \Phi_i \right) \wedge \left( \bigwedge_{j \in [1, D']} \mathbf{A} \Phi'_j \right).$$

The direction from right to left is trivial. So let us prove the direction from left to right: suppose that  $\phi$  is satisfiable. Let  $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathbf{v})$  be a Kripke structure,  $w_{\text{in}} \in \mathcal{W}$  be a world in  $\mathcal{K}$  such that  $\mathcal{K}, w_{\text{in}} \models \phi$ . Since  $\phi$  contains only the variables in  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$ , the map  $\mathbf{v}$  can be restricted to the variables among  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$ . Furthermore, below, we can represent  $\mathbf{v}$  as a map  $\mathcal{W} \rightarrow \mathbb{Z}^\beta$  such that  $\mathbf{v}(w)(i)$  for some  $i \in [1, \beta]$  is understood as the value of the variable  $\mathbf{x}_i$  on  $w$ . We construct a tree  $\mathbf{t} : [0, D-1]^* \rightarrow \mathbb{Z}^\beta$  such that  $\mathbf{t} \models \phi$  and  $\mathbf{t}$  satisfies AGE  $\Phi_i$  via  $i$ , for each  $i \in [1, D-1]$ .

We introduce an auxiliary map  $g : [0, D-1]^* \rightarrow \mathcal{W}$  such that  $g(\varepsilon) \stackrel{\text{def}}{=} w_{\text{in}}$ ,  $\mathbf{t}(\varepsilon) \stackrel{\text{def}}{=} \mathbf{v}(g(\varepsilon))$ . More generally, we require that for all nodes  $\mathbf{n} \in [0, D-1]^*$ , we have  $\mathbf{t}(\mathbf{n}) = \mathbf{v}(g(\mathbf{n}))$ , and if  $\mathbf{n} = j_1 \cdots j_k$ , then there exists a finite path  $g(\varepsilon)g(j_1)g(j_1j_2) \dots g(\mathbf{n})$  in  $\mathcal{K}$ . (Note that this is satisfied by  $\varepsilon$ , too.) The definition of  $g$  is performed by picking the smallest node  $\mathbf{n} \cdot j \in [0, D-1]^*$  with respect to the lexicographical ordering such that  $g(\mathbf{n})$  is defined and  $g(\mathbf{n} \cdot j)$  is undefined. So let  $\mathbf{n} \cdot j$  be the smallest node  $\mathbf{n} \cdot j \in [0, D-1]^*$  such that  $g(\mathbf{n})$  is defined,  $g(\mathbf{n} \cdot j)$  is undefined, and if  $\mathbf{n} = j_1 \cdots j_k$ , then there exists a finite path  $g(\varepsilon)g(j_1)g(j_1j_2) \dots g(\mathbf{n})$  in  $\mathcal{K}$ .

If  $j = 0$ , then since  $\mathcal{K}$  is total, there is an infinite path  $\pi = w_0w_1w_2 \dots$  starting from  $g(\mathbf{n})$ . For all  $k \geq 1$ , we set  $g(\mathbf{n} \cdot 0^k) \stackrel{\text{def}}{=} w_k$  and  $\mathbf{t}(\mathbf{n} \cdot 0^k) \stackrel{\text{def}}{=} \mathbf{v}(w_k)$ .

Otherwise ( $j \neq 0$ ), since  $g(\mathbf{n})$  is a world on a path starting from  $w_{\text{in}}$ , we obtain  $\mathcal{K}, g(\mathbf{n}) \models \mathbf{E}\Phi_j$  by assumption. So there exists some infinite path  $\pi = w_0w_1w_2 \dots$  starting from  $g(\mathbf{n})$  such that  $\mathcal{K}, \pi \models \Phi_j$ . Define  $g(\mathbf{n} \cdot j \cdot 0^k) \stackrel{\text{def}}{=} w_{k+1}$ , and  $\mathbf{t}(\mathbf{n} \cdot j \cdot 0^k) \stackrel{\text{def}}{=} \mathbf{v}(w_{k+1})$  for all  $k \geq 0$ . Note that this implies that  $\mathbf{t}(\mathbf{n})\mathbf{t}(\mathbf{n} \cdot j)\mathbf{t}(\mathbf{n} \cdot j \cdot 0)\mathbf{t}(\mathbf{n} \cdot j \cdot 0^2) \dots$  satisfies  $\Phi_j$  and therefore  $\mathbf{t}, \mathbf{n} \models \mathbf{E} \Phi_j$ . By construction, we get  $\mathbf{t} \models \text{AGE } \Phi_j$  for all  $j \in [1, D-1]$ . Moreover, by construction of  $\mathbf{t}$ , for all infinite paths  $j_1j_2 \dots \in [0, D-1]^\omega$ ,  $g(\varepsilon)g(j_1)g(j_1j_2) \dots$  is an infinite path from  $g(\varepsilon)$ , consequently  $\mathbf{t} \models \mathbf{E}(\mathbf{x} = 0) \wedge \bigwedge_{j \in [1, D']} \mathbf{A} \Phi'_j$  too. Hence,  $\mathbf{t} \models \phi$  and  $\mathbf{t}$  satisfies AGE  $\Phi_i$  via  $i$ , for each  $i \in [1, D-1]$ .  $\square$

Let  $\Phi$  be an  $\text{LTL}(\mathbb{Z})$  formula in simple form built over the variables  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$  and the constants  $\mathfrak{d}_1, \dots, \mathfrak{d}_\alpha$ , and let  $D \geq 1$ . By Corollary 1, there is a Rabin deterministic constraint automaton  $\mathbb{A} = (Q, \Sigma, \beta, Q_{\text{in}}, \delta, \mathcal{F})$  such that  $\{\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta \mid \mathbf{w} \models \Phi\} = \text{L}(\mathbb{A})$ . Below, we construct a Rabin tree constraint automaton  $\mathbb{A}' = (Q', \Sigma, D, \beta, Q'_{\text{in}}, \delta', \mathcal{F}')$  such that

$$\{\mathbf{t} : [0, D-1]^* \rightarrow \mathbb{Z}^\beta \mid \mathbf{t} \models \mathbf{A} \Phi\} = \text{L}(\mathbb{A}').$$

Let us define  $\mathbb{A}'$  formally.

$$- Q' \stackrel{\text{def}}{=} Q; Q'_{\text{in}} \stackrel{\text{def}}{=} Q_{\text{in}}; \mathcal{F}' \stackrel{\text{def}}{=} \mathcal{F}.$$

- $\delta'$  is made of tuples of the form  $(q, \mathbf{a}, (\Theta_0, q_0), \dots, (\Theta_{D-1}, q_{D-1}))$ , where  $(q, \mathbf{a}, \Theta_i, q_i) \in \delta$  for all  $0 \leq i < D$ .

**Lemma 19.**  $L(\mathbb{A}')$  is equal to the set of full  $D$ -ary trees satisfying the  $CTL^*(\mathbb{Z})$  formula  $\mathbb{A} \Phi$ .

The proof is by an easy verification thanks to the determinism of  $\mathbb{A}$ , which is essential here. Typically, Proposition 8 is not sufficient because determinism of the word automaton is required. For example, for any  $\mathbf{t} : [0, D-1]^* \rightarrow \mathbb{Z}^\beta$  and any  $\rho$  on  $\mathbf{t}$ , an infinite branch satisfies the acceptance condition  $\mathcal{F}'$  iff the word  $\mathbf{w}$  from the branch of  $\mathbf{t}$  satisfies  $\Phi$  by definition of  $\mathbb{A}$ . That is why we put so much efforts in developing material in Section 6.3.

$CTL^*(\mathbb{Z})$  formulae in special form are essentially made of conjunctions with arguments of the form  $\mathbb{A} \Phi$  and  $\text{AGE } \Phi$ , where  $\Phi$  is an  $LTL(\mathbb{Z})$  formula in simple form. Above, we have seen how to handle the formula  $\mathbb{A} \Phi$ . Below, let us explain how to build tree constraint automata capturing a relevant subclass of tree models for  $\text{AGE } \Phi$ . Let  $\phi = \text{AGE } \Phi$  with  $\Phi$  built over the variables  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$  and the constants  $\mathbf{d}_1, \dots, \mathbf{d}_\alpha$  and  $D \geq 2$ .

By Proposition 8, there is a word constraint automaton  $\mathbb{A} = (Q, \Sigma, \beta, Q_{\text{in}}, \delta, F)$  such that  $\{\mathbf{w} : \mathbb{N} \rightarrow \mathbb{Z}^\beta \mid \mathbf{w} \models \Phi\} = L(\mathbb{A})$ . Note that  $\mathbb{A}$  is not necessarily deterministic and the acceptance condition is a Büchi condition. Below, we construct a tree constraint automaton  $\mathbb{A}' = (Q', \Sigma, D, \beta, Q'_{\text{in}}, \delta', F')$  such that

$$\{\mathbf{t} : [0, D-1]^* \rightarrow \mathbb{Z}^\beta \mid \mathbf{t} \models \text{AGE } \Phi \text{ and } \mathbf{t} \text{ satisfies AGE } \Phi \text{ via } i\} = L(\mathbb{A}').$$

Let us define  $\mathbb{A}'$  formally:

- $Q' \stackrel{\text{def}}{=} [0, D-1] \times (Q \cup \{\perp\})$ , where  $\perp \notin Q$  ( $\Sigma = \{\dagger\}$ );
- $Q'_{\text{in}} \stackrel{\text{def}}{=} \{(0, \perp)\}$ ;  $F' \stackrel{\text{def}}{=} \{(j, q) \mid j \neq 0 \text{ or } q \in F\}$ ,
- The transition relation  $\delta'$  is made of tuples of the form

$$((i, q), \dagger, (\Theta_0, (0, q_0)), \dots, (\Theta_{D-1}, (D-1, q_{D-1})))$$

verifying the conditions below.

1.  $(q_{\text{in}}, \dagger, \Theta_i, q_i) \in \delta$  for some  $q_{\text{in}} \in Q_{\text{in}}$  (starting off  $\mathbb{A}$  in the  $i$ th child).
2.  $q_0 = \perp$  and  $\Theta_0 = \top$  if  $q = \perp$  and  $(q, \dagger, \Theta_0, q_0) \in \delta$  if  $q \in Q$  (continuing a run from  $q$  of  $\mathbb{A}$  in the 0th child).
3.  $q_k = \perp$  and  $\Theta_k = \top$  for all  $k \in ([0, D-1] \setminus \{0, i\})$ .

**Lemma 20.**  $\{\mathbf{t} : [0, D-1]^* \rightarrow \mathbb{Z}^\beta \mid \mathbf{t} \models \phi \text{ and } \mathbf{t} \text{ satisfies AGE } \Phi \text{ via } i\} = L(\mathbb{A}')$ .

*Proof.* Let  $\phi = \text{AGE } \Phi$  and  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$  be a tree such that  $\mathbf{t} \in L(\mathbb{A}')$ . Below, we prove that for all nodes  $\mathbf{n} \in [0, D-1]^*$ , the word  $\mathbf{t}(\mathbf{n}) \mathbf{t}(\mathbf{n} \cdot i) \mathbf{t}(\mathbf{n} \cdot i \cdot 0) \mathbf{t}(\mathbf{n} \cdot i \cdot 0^2) \dots$  satisfies  $\Phi$ . This implies  $\mathbf{t} \models \phi$  and  $\mathbf{t}$  satisfies  $\phi$  via  $i$ .

So let  $\mathbf{n} \in [0, D-1]^*$  be an arbitrary node in  $\mathbf{t}$ . Consider the infinite path  $\mathbf{n} \cdot i \cdot 0^\omega \in [0, D-1]^\omega$ . Let  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \mathbf{z})$  and  $\mathbf{t}(\mathbf{n} \cdot i \cdot 0^k) = (\mathbf{a}_k, \mathbf{z}_k)$  for all  $k \geq 0$ . Let  $\rho : [0, D-1]^* \rightarrow Q'$  be an accepting run of  $\mathbb{A}'$  on  $\mathbf{t}$ . By definition of  $\delta'$ ,  $\rho(\mathbf{n} \cdot i) = (i, q_0)$  for some transition  $(q_{\text{in}}, \mathbf{a}, \Theta, q_0) \in \delta$  with  $q_{\text{in}} \in Q_{\text{in}}$  and  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_0)$ . Again by definition of  $\delta'$ , we have, for all  $k \geq 1$ ,  $\rho(\mathbf{n} \cdot i \cdot 0^k) = (0, q_k)$  for some location  $q_k \in Q$  such that there exists some transition  $(q_{k-1}, \mathbf{a}_{k-1}, \Theta_{k-1}, q_k) \in \delta$ , and we have  $\mathbb{Z} \models \Theta_{k-1}(\mathbf{z}_{k-1}, \mathbf{z}_k)$ . Since  $\rho$  is accepting, we know that there are infinitely many positions  $\ell \geq 1$  such that  $q_\ell \in F$ . Hence the run  $q \xrightarrow{(\mathbf{a}, \mathbf{z})} q_0 \xrightarrow{(\mathbf{a}_0, \mathbf{z}_0)} q_1 \dots$  is an accepting run of  $\mathbb{A}$  on  $(\mathbf{a}, \mathbf{z})(\mathbf{a}_0, \mathbf{z}_0) \dots$ . But then also  $(\mathbf{a}, \mathbf{z})(\mathbf{a}_0, \mathbf{z}_0) \dots$  satisfies  $\Phi$ . Since  $\mathbf{n}$  is arbitrary, we have  $\mathbf{t} \models \phi$  and  $\mathbf{t}$  satisfies  $\phi$  via  $i$ .

For the other direction, suppose that  $\mathbf{t} \models \phi$  and  $\mathbf{t}$  satisfies  $\phi$  via  $i$ . We prove that  $\mathbf{t} \in L(\mathbb{A}')$ , that is, there exists some accepting run  $\rho : [0, D-1]^* \rightarrow Q'$  of  $\mathbb{A}'$  on  $\mathbf{t}$ . We prove that we can construct a run  $\rho$  such that, for every node  $\mathbf{n} \in [0, D-1]^*$ , the path  $\rho(\mathbf{n})\rho(\mathbf{n} \cdot i)\rho(\mathbf{n} \cdot i \cdot 0)\rho(\mathbf{n} \cdot i \cdot 0^2) \dots$  corresponds to some accepting run of  $\mathbb{A}$ . This implies that  $\rho$  is an accepting run of  $\mathbb{A}$  on  $\mathbf{t}$  (all the other paths are non-critical).

Let  $\mathbf{n} \in [0, D-1]^*$  be an arbitrary node in  $\mathbf{t}$ . Let us assume  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \mathbf{z})$  and  $\mathbf{t}(\mathbf{n} \cdot i \cdot 0^k) = (\mathbf{a}_k, \mathbf{z}_k)$  for all  $k \geq 0$ , and we use  $\mathbf{w}_{\mathbf{n}}$  to denote the corresponding infinite word  $(\mathbf{a}, \mathbf{z})(\mathbf{a}_0, \mathbf{z}_0)(\mathbf{a}_1, \mathbf{z}_1) \dots$ . Since  $\mathbf{t}$  satisfies  $\phi$  via  $i$ , we know that  $\mathbf{w}_{\mathbf{n}} \models \Phi$ . But then also  $\mathbf{w}_{\mathbf{n}} \in L(\mathbb{A})$ . So there must exist some accepting run of  $\mathbb{A}$  on  $\mathbf{w}_{\mathbf{n}}$ , say,  $q_{\text{in}}, q_0, q_1, q_2 \dots$  such that  $q_{\text{in}} \in Q_{\text{in}}$ , there exist transitions  $(q_{\text{in}}, \mathbf{a}, \Theta, q_0) \in \delta$  and  $(q_k, \mathbf{a}_k, \Theta_k, q_{k+1}) \in \delta$  for all  $k \geq 0$ , and  $\mathbb{Z} \models \Theta(\mathbf{z}, \mathbf{z}_0)$  and  $\mathbb{Z} \models \Theta_k(\mathbf{z}_k, \mathbf{z}_{k+1})$  for all  $k \geq 0$ . The definition of  $\delta'$  now allows us to define  $\rho(\mathbf{n} \cdot i) = (i, q_0)$ , and  $\rho(\mathbf{n} \cdot i \cdot 0^k) = (0, q_k)$  for every  $k \geq 1$ .  $\square$

## 6.5 The final step of the complexity analysis

Let  $\phi$  be a  $\text{CTL}^*(\mathbb{Z})$  formula in special form defined as follows.

$$\phi = \mathbf{E}(\mathbf{x}_1 = 0) \wedge \left( \bigwedge_{i \in [1, D-1]} \text{AGE } \Phi_i \right) \wedge \left( \bigwedge_{i \in [1, D']} \mathbf{A} \Phi'_i \right),$$

where  $D \geq 1$ ,  $D' \geq 0$  and, the  $\Phi_i$ 's and the  $\Phi'_i$ 's are  $\text{LTL}(\mathbb{Z})$  formulae in simple form (empty conjunctions are understood as true). We assume that the variables in  $\phi$  are among  $\mathbf{x}_1, \dots, \mathbf{x}_\beta$ .

Let us show that the satisfiability status of  $\phi$  can be checked in double-exponential time. By Proposition 7, this is sufficient to establish that the satisfiability problem for  $\text{CTL}^*(\mathbb{Z})$  is in  $2\text{EXPTIME}$ . Moreover, by Proposition 9,  $\phi$  is satisfiable iff there is a tree  $\mathbf{t} : [0, D-1] \rightarrow \mathbb{Z}^\beta$  such that  $\mathbf{t} \models \phi$  and  $\mathbf{t}$  satisfies  $\text{AGE } \Phi_i$  via  $i$ , for each  $i \in [1, D-1]$ . Let us recapitulate a few properties about the tree constraint automata we can build from  $\phi$ .

- There is a two-location tree constraint automaton  $\mathbb{A}_0$  such that  $L(\mathbb{A}_0)$  is the set of trees  $\mathbf{t} : [0, D-1]^* \rightarrow \mathbb{Z}^\beta$  such that  $\mathbf{t}(\varepsilon)(1) = 0$ . This is to handle  $\mathbf{E}(\mathbf{x}_1 = 0)$  in  $\text{CTL}^*(\mathbb{Z})$  formulae in special form.
- By Lemma 19, for each  $i \in [1, D']$ , there is Rabin tree constraint automaton  $\mathbb{B}_i$  such that  $L(\mathbb{B}_i)$  is equal to the set of full  $D$ -ary trees satisfying the  $\text{CTL}^*(\mathbb{Z})$  formula  $\mathbf{A} \Phi'_i$ .
- By Lemma 20, for each  $i \in [1, D-1]$ , there is (Büchi) tree constraint automaton  $\mathbb{A}_i$  that  $L(\mathbb{A}_i)$  is equal to the set of full  $D$ -ary trees  $\mathbf{t}$  of degree  $D$  such that  $\mathbf{t} \models \text{AGE } \Phi_i$  and  $\mathbf{t}$  satisfies  $\text{AGE } \Phi_i$  via  $i$ .

Consequently, by Proposition 9,  $\phi$  is satisfiable iff the intersection below is non-empty:

$$L(\mathbb{A}_0) \bigcap_{i \in [1, D-1]} L(\mathbb{A}_i) \bigcap_{i \in [1, D']} L(\mathbb{B}_i).$$

The tree constraint automata  $\mathbb{A}_i$ 's can be viewed as Rabin tree constraint automata with a single Rabin pair. By contrast, the  $\mathbb{B}_i$ 's are already Rabin automata with an exponential number of Rabin pairs in  $\text{size}(\phi)$  (see Corollary 1(II)). In order to define a Rabin tree constraint automaton  $\mathbb{A}$  such that

$$L(\mathbb{A}) = L(\mathbb{A}_0) \bigcap_{i \in [1, D-1]} L(\mathbb{A}_i) \bigcap_{i \in [1, D']} L(\mathbb{B}_i),$$

and then use the complexity bounds from Section 4.5, we need the result below that generalises the first part of the proof of Lemma 9. The generalisation is performed in two ways: we intersect an arbitrary number of tree automata (instead of two in the proof of Lemma 9) and we have to handle constraints.

**Lemma 21.** Let  $(\mathbb{A}_i)_{1 \leq i \leq n}$  be a family of arbitrary Rabin tree constraint automata such that  $\mathbb{A}_i = (Q_i, \Sigma, D, \beta, Q_{i,\text{in}}, \delta_i, \mathcal{F}_i)$ ,  $\text{card}(\mathcal{F}_i) = N_i$  and  $N = \prod_i N_i$ . There is a Rabin tree constraint automaton  $\mathbb{A}$  such that  $L(\mathbb{A}) = \bigcap_i L(\mathbb{A}_i)$  and verifying the conditions below.

- The number of Rabin pairs is equal to  $N$ .
- The number of locations is bounded by  $(\prod_i \text{card}(Q_i)) \times (2n)^N$ .
- The number of transitions is bounded by  $\prod_i \text{card}(\delta_i)$ .
- $\text{MaxConsSize}(\mathbb{A}) \leq \sum_i (1 + \text{MaxConsSize}(\mathbb{A}_i))$ .

*Proof.* We present a construction for the intersection of an arbitrary number of Rabin tree constraint automata mainly based on ideas from the proof of [Bok18, Theorem 1] on Rabin word automata over finite alphabets.

For each  $i \in [1, n]$ , let  $\mathbb{A}_i = (Q_i, \Sigma, D, \beta, Q_{i,\text{in}}, \delta_i, \mathcal{F}_i)$  with  $\mathcal{F}_i = (L_i^j, U_i^j)_{j \in [1, N_i]}$  be a RTCA. Let us build a RTCA  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, \mathcal{F})$  such that  $L(\mathbb{A}) = \bigcap_i L(\mathbb{A}_i)$ .

- $Q \stackrel{\text{def}}{=} Q_1 \times \dots \times Q_n \times [0, 2n-1]^{[1, N_1] \times \dots \times [1, N_n]}$ . The elements in  $Q$  are of the form  $(q_1, \dots, q_n, \mathbf{f})$  with  $\mathbf{f}: [1, N_1] \times \dots \times [1, N_n] \rightarrow [0, 2n-1]$ .
- The tuple

$$((q_1, \dots, q_n, \mathbf{f}), \mathbf{a}, (\Theta_0, (q_1^0, \dots, q_n^0, \mathbf{f}^0)), \dots, (\Theta_{D-1}, (q_1^{D-1}, \dots, q_n^{D-1}, \mathbf{f}^{D-1})))$$

belongs to  $\delta$  iff the conditions below hold.

1. For each  $k \in [1, n]$ , there is  $(q_k, \mathbf{a}, ((\Theta_k^0, q_k^0), \dots, (\Theta_k^{D-1}, q_k^{D-1}))) \in \delta_k$  and for each  $j \in [0, D-1]$ ,

$$\Theta_j \stackrel{\text{def}}{=} \bigwedge_{k=1}^n \Theta_k^j.$$

2. For all  $(i_1, \dots, i_n) \in [1, N_1] \times \dots \times [1, N_n]$ , the following conditions hold.
  - (a) If  $\mathbf{f}(i_1, \dots, i_n)$  is odd, then for all  $j \in [0, D-1]$ , we have  $\mathbf{f}^j(i_1, \dots, i_n) = (\mathbf{f}(i_1, \dots, i_n) + 1) \bmod 2n$ . Odd values in  $[0, 2n-1]$  are unstable and are replaced at the next step by the successor value (modulo  $2n$ ).
  - (b) For all  $k \in [1, n]$ , for all  $j \in [0, D-1]$ ,
    - if  $\mathbf{f}(i_1, \dots, i_n) = 2k-2$  and  $q_k^j \in L_k^{i_k}$ , then  $\mathbf{f}^j(i_1, \dots, i_n) = 2k-1$  (increment),
    - if  $\mathbf{f}(i_1, \dots, i_n) = 2k-2$  and  $q_k^j \notin L_k^{i_k}$ , then  $\mathbf{f}^j(i_1, \dots, i_n) = 2k-2$  (no change).

When  $n = 2$ , the above conditions are identical to those in the proof of Lemma 9 and therefore the above developments generalise what is done in that proof. Observe that the “ $\mathbf{f}$ -part” is updated deterministically. We write  $\mathbf{f}, (q_k^j)_{0 \leq j \leq D-1, 1 \leq k \leq n} \mapsto (\mathbf{f}^0, \dots, \mathbf{f}^{D-1})$  if the tuple  $(\mathbf{f}^0, \dots, \mathbf{f}^{D-1})$  is obtained from  $\mathbf{f}$  and  $(q_k^j)_{0 \leq j \leq D-1, 1 \leq k \leq n}$ .

- $Q_{\text{in}} \stackrel{\text{def}}{=} Q_{1,\text{in}} \times \dots \times Q_{n,\text{in}} \times \{\mathbf{f}_0\}$ , where  $\mathbf{f}_0$  is the unique map that takes always the value zero (this value is arbitrary and any value will do the job).
- The set of Rabin pairs in  $\mathcal{F}$  contains exactly the pairs  $(L, U)$  for which there is  $(i_1, \dots, i_n) \in [1, N_1] \times \dots \times [1, N_n]$  such that

$$U \stackrel{\text{def}}{=} \left( \bigcup_{k=1}^n Q_1 \times \dots \times Q_{k-1} \times U_k^{i_k} \times Q_{k+1} \times \dots \times Q_n \right) \times [0, 2n-1]^{[1, N_1] \times \dots \times [1, N_n]}$$

$$L \stackrel{\text{def}}{=} (L_1^{i_1} \times Q_2 \times \cdots \times Q_n) \times \{f \mid f(i_1, \dots, i_n) = 1\}.$$

Again, when  $n = 2$ , the above construction is identical to the one in the proof of Lemma 9. Strictly speaking,  $(L, U)$  above is indexed by a tuple  $(i_1, \dots, i_n)$  but we omit such decorations as it will not lead to any confusion.

By way of example, if along a branch of a run for  $\mathbb{A}$  the tuples in  $U$  are visited finitely, then a location in  $U_1^{i_1}$  is visited finitely on the first component, and a location in  $U_2^{i_2}$  is visited finitely on the second component, and more generally, a location in  $U_k^{i_k}$  is visited finitely on the  $k$ th component,

Finally,  $\mathcal{F}$  contains at most  $N = N_1 \times \cdots \times N_n$  pairs.

Let us sketch the proof of  $L(\mathbb{A}) = \bigcap_k L(\mathbb{A}_k)$ .

First, assume that  $\mathbf{t} \in \bigcap_k L(\mathbb{A}_k)$ . For each  $k \in [1, n]$ , there is an enriched run  $\rho_k^+ : [0, D-1]^* \rightarrow \delta_k$  on  $\mathbf{t}$ . We also write  $\rho_k$  to denote the accepting run on  $\mathbf{t}$  produced from  $\rho_k^+$ , typically if  $\rho_k^+(\mathbf{n}) = (q_k, \mathbf{a}, ((\theta_k^0, q_k^0), \dots, (\theta_k^{D-1}, q_k^{D-1})))$  (with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \mathbf{z})$ ), then  $\rho_k(\mathbf{n}) = q_k$ , for all  $\mathbf{n} \in [0, D-1]^*$ .

Let us define a map  $\rho^+ : [0, D-1]^* \rightarrow \delta$  that we can show to be an enriched run. The satisfaction of (ER1) (see Section 4.2) is by definition of  $Q_{\text{in}}$  where the satisfaction of (ER2) follows easily from the definition of  $\delta$ . As far as the satisfaction of (ER3) is concerned, it is a consequence of the definition of  $\mathcal{F}$  but we need here to provide a bit more arguments. The final step is to check that the run induced from  $\rho^+$  is accepted on the tree  $\mathbf{t}$  and this can be done thanks to the way constraints are defined in  $\delta$  by taking conjunction of constraints the transitions of the  $\mathbb{A}_k$ 's.

We start by  $\rho(\varepsilon) \stackrel{\text{def}}{=} (\rho_1(\varepsilon), \dots, \rho_n(\varepsilon), f_0)$  and for all  $\mathbf{n} \in [0, D-1]^*$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \mathbf{z})$ ,  $\rho^+(\mathbf{n})$  is equal to

$$((q_1, \dots, q_n, f), \mathbf{a}, (\theta_0, (q_1^0, \dots, q_n^0, f^0)), \dots, (\theta_{D-1}, (q_1^{D-1}, \dots, q_n^{D-1}, f^{D-1}))),$$

with for each  $j \in [0, D-1]$ ,  $\theta_j \stackrel{\text{def}}{=} \bigwedge_{k=1}^n \theta_k^j$  and  $f, (q_k^j) \mapsto (f^0, \dots, f^{D-1})$ . As announced above, (ER1) is by definition of  $Q_{\text{in}}$  and satisfaction of (ER2) is by definition of  $\delta$ . Moreover, since  $\delta$  is defined by taking conjunctions of constraints from the  $\mathbb{A}_k$ 's,  $\rho$  induced by  $\rho^+$  is actually a run on  $\mathbf{t}$ . To conclude that  $\mathbf{t} \in L(\mathbb{A})$ , we need to show that  $\rho$  is accepting.

Let  $j_1 j_2 \cdots \in [0, D-1]^\omega$  be an arbitrary branch. Since the  $\rho_k$ 's are accepting runs, for each  $k \in [1, n]$ , there is  $i_k$  such that a location in  $L_k^{i_k}$  occurs infinitely in  $\rho_k(j_1), \rho_k(j_1 j_2), \rho_k(j_1 j_2 j_3) \dots$  and all the locations in  $U_k^{i_k}$  occur finitely in  $\rho_k(j_1), \rho_k(j_1 j_2), \rho_k(j_1 j_2 j_3) \dots$ . Let us pick  $(L, U)$  in  $\mathcal{F}$  such that

$$U = \left( \bigcup_{k=1}^n Q_1 \times \cdots \times Q_{k-1} \times U_k^{i_k} \times Q_{k+1} \times \cdots \times Q_n \right) \times [0, 2n-1]^{[1, N_1] \times \cdots \times [1, N_n]}$$

$$L = (L_1^{i_1} \times Q_2 \times \cdots \times Q_n) \times \{f \mid f(i_1, \dots, i_n) = 1\}.$$

Since all the  $Q_k$ 's are finite sets and  $[0, 2n-1]^{[1, N_1] \times \cdots \times [1, N_n]}$  is finite too, all the locations in  $U$  occur finitely in  $\rho(j_1), \rho(j_1 j_2), \rho(j_1 j_2 j_3) \dots$ . In order to verify that some location in  $L$  occurs infinitely in  $\rho(j_1), \rho(j_1 j_2), \rho(j_1 j_2 j_3) \dots$ , note that the odd values are unstable in  $f$ . If a location in  $L$  is visited infinitely along a branch of a run for  $\mathbb{A}$  (and therefore a location in  $L_1^{i_1}$  is visited infinitely often on the first component), then a location in  $L_k^{i_k}$  ( $k \in [2, n]$ ) is also visited infinitely often on the  $k$ th component. Indeed, to revisit the value 1 on the  $(i_1, \dots, i_n)$ th component one needs to visit first the value  $2k-1$ , which witnesses that a location in  $L_k^{i_k}$  has been found.

The proof in the other direction is similar and is omitted herein. For instance, assuming that  $\mathbf{t} \in L(\mathbb{A})$ , there is an enriched run  $\rho^+$  on  $\mathbf{t}$ . For each  $k \in [1, n]$ , by projection of the values from the transitions  $\rho^+(\mathbf{n})$ 's, we get maps  $\rho_k^+ : [0, D-1]^* \rightarrow \delta_k$ . Similarly to the argument for the first direction,  $\rho_k^+$  satisfies (ER1) and (ER2) as a direct consequence of the respective definitions of  $Q_{\text{in}}$  and  $\delta$ . Again, as  $\delta$  is

defined by taking conjunctions of constraints from the  $\mathbb{A}_k$ 's, each  $\rho_k$  (run induced from  $\rho_k^+$ ) is a run on  $\mathbf{t}$ . It remains to check it is accepting. Given an arbitrary infinite branch  $j_1 j_2 \dots \in [0, D-1]^\omega$ , there is  $(L, U) \in \mathcal{F}$  such that the Rabin pair holds on the branch and this pair is defined from some tuple  $(i_1, \dots, i_n)$ . For analogous reasons to what is stated above, one can show that  $(L_k^{i_k}, U_k^{i_k})$  holds on the branch  $j_1 j_2 \dots \in [0, D-1]^\omega$  in  $\rho_k$ .

□

In Section 4.5, we have seen that nonemptiness of the language  $L(\mathbb{A})$  for some Rabin tree constraint automaton  $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, \mathcal{F})$  can be solved in time in

$$\mathcal{O}\left(q_1(\text{card}(Q) \times \text{card}(\delta) \times \text{MaxConsSize}(\mathbb{A}) \times \text{card}(\Sigma) \times q_2(\beta + \text{card}(\mathcal{F})))^{q_2(\beta + \text{card}(\mathcal{F})) \times q_3(D)}\right)$$

Let us evaluate the size of the components for  $\mathbb{A}$  such that

$$L(\mathbb{A}) = L(\mathbb{A}_0) \bigcap_{i \in [1, D-1]} L(\mathbb{B}_i) \bigcap_{i \in [1, D']} L(\mathbb{A}_i),$$

based on the previous developments. First observe that  $\beta, D + D' \leq \text{size}(\phi)$  and  $\Sigma$  is a singleton.

- The  $\mathbb{A}_i$ 's have a single Rabin pair, and the  $\mathbb{B}_i$ 's have a number of Rabin pairs exponential in  $\text{size}(\phi)$  (by Lemma 20). By Lemma 21 and  $D' \leq \text{size}(\phi)$ , the number of Rabin pairs in  $\mathbb{A}$  is therefore exponential in  $\text{size}(\phi)$ .
- The number of locations in the  $\mathbb{A}_i$ 's is exponential in  $\text{size}(\phi)$  and the number of locations in the  $\mathbb{B}_i$ 's is double-exponential in  $\text{size}(\phi)$ . Since the number of Rabin pairs in  $\mathbb{A}$  is only exponential in  $\text{size}(\phi)$ , by Lemma 21 and  $D + D' \leq \text{size}(\phi)$ , the number of locations in  $\mathbb{A}$  is double-exponential in  $\text{size}(\phi)$ . A similar analysis can be performed for the number of transitions, leading to a double-exponential number of transitions.
- The maximal size of a constraint appearing in a transition from any  $\mathbb{A}_i$ 's and any  $\mathbb{B}_j$ 's is polynomial in  $\text{size}(\phi)$ . The maximal size of a constraint in the product automaton  $\mathbb{A}$  is therefore polynomial in  $\phi$  too ( $D + D' \leq \text{size}(\phi)$ ).

Putting all results together, the nonemptiness of  $L(\mathbb{A})$  can be checked in double-exponential time in  $\text{size}(\phi)$ , leading to Theorem 8 below. It answers open questions from [BG06, CKL16, CT16, LOS20] and it is the main result of the paper.

**Theorem 8.**  $\text{SAT}(\text{CTL}^*(\mathbb{Z}))$  is 2EXPTIME-complete.

2EXPTIME-hardness is from  $\text{SAT}(\text{CTL}^*)$  [VS85, Theorem 5.2]. As a corollary,  $\text{SAT}(\text{CTL}^*(\mathbb{N}))$  is also 2EXPTIME-complete. Furthermore, assuming that  $<_{\text{pre}}$  is the prefix relation on the set of finite strings  $\{0, 1\}^*$ , we can use the reduction from [DD16, Section 4.2] to get the following result.

**Corollary 2.**  $\text{SAT}(\text{CTL}^*(\{0, 1\}^*, <_{\text{pre}}))$  is 2EXPTIME-complete.

Moreover, as observed earlier, when the concrete domain is  $(\mathbb{Q}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{Q}})$ , all the trees in  $L(\mathbb{B}_{\text{cons}(\mathbb{A})})$  are satisfiable (no need to intersect  $\mathbb{B}_{\text{cons}(\mathbb{A})}$  with a hypothetical  $\mathbb{A}_{\star^C}$ ), and therefore  $\text{SAT}(\text{CTL}^*(\mathbb{Q}))$  is also in 2EXPTIME, which is a result already known from [Gas09, Theorem 4.3].

## 7 Proving the correctness of the condition $(\star^c)$

This section is dedicated to the proof of Proposition 4 (see Section 4.3). This is all the more important as the condition  $(\star^c)$  is central in our paper. Below, we assume that  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \text{SatTypes}(\beta)$  is locally consistent.

### 7.1 A simple characterisation for satisfiability

First, we establish a few auxiliary results about  $G_{\mathbf{t}}^c$  that are helpful in the sequel and that take advantage of the fact that every type in  $\text{SatTypes}(\beta)$  is satisfiable.

**Lemma 22.** *Let  $\pi = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim_1} \dots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$  be a path in  $G_{\mathbf{t}}^c$  such that  $\mathbf{n}_0$  and  $\mathbf{n}_n$  are neighbours. If  $\sim_1 = \dots = \sim_n$  equal to  $'='$ , then  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{=} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$ , otherwise  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{<} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$ .*

*Proof.* Let  $\pi = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim_1} \dots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$  be a path in  $G_{\mathbf{t}}^c$  such that  $\mathbf{n}_0$  and  $\mathbf{n}_n$  are neighbours.

(Property 1) Firstly, if  $n \geq 2$ , then one can show that there is  $0 < h < n$  such that  $\mathbf{n}_0$ ,  $\mathbf{n}_h$  and  $\mathbf{n}_n$  are pairwise neighbours. Let us explain briefly how  $h$  is computed.

- Case  $|\mathbf{n}_0| = |\mathbf{n}_n|$  ( $\mathbf{n}_0 = \mathbf{n}_n$ ).  $h \stackrel{\text{def}}{=} n - 1$ .
- Case  $|\mathbf{n}_0| = |\mathbf{n}_n| + 1$  ( $\mathbf{n}_0$  is a child of  $\mathbf{n}_n$ ) and for all  $i \in [1, n-1]$ ,  $|\mathbf{n}_i| \geq |\mathbf{n}_0|$ .  $h \stackrel{\text{def}}{=} n - 1$ .
- Case  $|\mathbf{n}_0| = |\mathbf{n}_n| + 1$  and there is  $i \in [1, n-1]$ ,  $|\mathbf{n}_i| < |\mathbf{n}_0|$ .  $h \stackrel{\text{def}}{=} \min\{i \in [1, n-1] \mid |\mathbf{n}_i| = |\mathbf{n}_n|\}$ . Note that actually  $\mathbf{n}_h = \mathbf{n}_n$ .
- Case  $|\mathbf{n}_n| = |\mathbf{n}_0| + 1$  ( $\mathbf{n}_n$  is a child of  $\mathbf{n}_0$ ) and for all  $i \in [1, n-1]$ ,  $|\mathbf{n}_i| \geq |\mathbf{n}_n|$ .  $h \stackrel{\text{def}}{=} 1$ . Note that actually  $\mathbf{n}_h = \mathbf{n}_n$ .
- Case  $|\mathbf{n}_n| = |\mathbf{n}_0| + 1$  and there is  $i \in [1, n-1]$ ,  $|\mathbf{n}_i| < |\mathbf{n}_n|$ .  $h \stackrel{\text{def}}{=} \min\{i \in [1, n-1] \mid |\mathbf{n}_i| = |\mathbf{n}_0|\}$ .

Observe that  $h \leq n-1$ ,  $n-h \leq n-1$ , and that  $\mathbf{n}_0$  and  $\mathbf{n}_h$  are neighbours, and  $\mathbf{n}_n$  and  $\mathbf{n}_h$  are neighbours.

(Property 2) Secondly, by construction of  $G_{\mathbf{t}}^c$  from  $\mathbf{t}$  built over satisfiable constraints in  $\text{SatTypes}(\beta)$ , we can show the property below (this requires a lengthy case analysis). Let  $(\mathbf{m}_1, \mathbf{x}\mathbf{d}_1)$ ,  $(\mathbf{m}_2, \mathbf{x}\mathbf{d}_2)$  and  $(\mathbf{m}_3, \mathbf{x}\mathbf{d}_3)$  be nodes in the graph  $G_{\mathbf{t}}^c$  that are pairwise neighbours such that  $(\mathbf{m}_1, \mathbf{x}\mathbf{d}_1) \xrightarrow{\sim_1} (\mathbf{m}_2, \mathbf{x}\mathbf{d}_2)$  and  $(\mathbf{m}_2, \mathbf{x}\mathbf{d}_2) \xrightarrow{\sim_2} (\mathbf{m}_3, \mathbf{x}\mathbf{d}_3)$  with  $\sim_1, \sim_2 \in \{<, =\}$ . If  $< \in \{\sim_1, \sim_2\}$ , then  $(\mathbf{m}_1, \mathbf{x}\mathbf{d}_1) \xrightarrow{<} (\mathbf{m}_3, \mathbf{x}\mathbf{d}_3)$  otherwise  $(\mathbf{m}_1, \mathbf{x}\mathbf{d}_1) \xrightarrow{=} (\mathbf{m}_3, \mathbf{x}\mathbf{d}_3)$ .

Now, we can prove the lemma. If  $n = 0$  or  $n = 1$ , we are done. Otherwise, the induction hypothesis assumes that the property holds for  $n \leq K$  and let  $\pi = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim_1} \dots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$  be a path in  $G_{\mathbf{t}}^c$  such that  $\mathbf{n}_0$  and  $\mathbf{n}_n$  are neighbours with  $n = K + 1$ . By (Property 1), there is  $0 < h < n$  such that  $\mathbf{n}_0$ ,  $\mathbf{n}_h$  and  $\mathbf{n}_n$  are pairwise neighbours,  $h \leq K$  and  $n - h \leq K$ . By the induction hypothesis, we have  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{<} (\mathbf{n}_h, \mathbf{x}\mathbf{d}_h)$  if  $< \in \{\sim_1, \dots, \sim_h\}$ , otherwise  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{=} (\mathbf{n}_h, \mathbf{x}\mathbf{d}_h)$ . Similarly, we have  $(\mathbf{n}_h, \mathbf{x}\mathbf{d}_h) \xrightarrow{<} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$  if  $< \in \{\sim_{h+1}, \dots, \sim_n\}$ , otherwise  $(\mathbf{n}_h, \mathbf{x}\mathbf{d}_h) \xrightarrow{=} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$ . By (Property 2), we get  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{<} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$  if  $< \in \{\sim_1, \dots, \sim_n\}$ , otherwise  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{=} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$ .  $\square$

As a corollary, we get the following lemma.

**Lemma 23.** *Let  $\pi = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim_1} \dots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$  be a path in  $G_{\mathbf{t}}^c$  such that  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) = (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$ . All  $\sim_i$ 's are equal to  $'='$ .*

Below, we provide a simple characterisation for a locally consistent symbolic tree to be satisfiable. Note that this is different from [Lab21, Lemma 5.18] because we have “constant elements” of the form  $(\mathbf{n}, \mathbf{d}_1)$  and  $(\mathbf{n}, \mathbf{d}_\alpha)$  in  $G_{\mathbf{t}}^c$ .

**Lemma 24.** Let  $\mathbf{t} : [0, D-1]^* \rightarrow \Sigma \times \text{SatTypes}(\beta)$  be a locally consistent symbolic tree. The statements below are equivalent.

- (I)  $\mathbf{t}$  is satisfiable.
- (II) For all  $(\mathbf{n}, \mathbf{x})$  in  $U_0 \cup U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$  in  $G_{\mathbf{t}}^C$ ,  $\text{slen}(\mathbf{n}, \mathbf{x}) < \omega$ .

The proof is similar to the proof of [DD07, Lemma 7.1] and does not require the generality of [CKL16, Lemma 34] as evoked to prove [Lab21, Lemma 5.18].

*Proof.* (I)  $\Rightarrow$  (II) Suppose  $\mathbf{t}$  is satisfiable. Then there exists a tree  $\mathbf{t}^* : [0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$  such that for all  $\mathbf{n} \cdot i \in [0, D-1]^+$  with  $\mathbf{t}(\mathbf{n} \cdot i) = (\mathbf{a}, \Theta)$ , we have  $\mathbb{Z} \models \Theta(\mathbf{t}(\mathbf{n}), \mathbf{t}(\mathbf{n} \cdot i))$ . Moreover, if  $\mathbf{t}^*(\varepsilon) = (\mathbf{a}, \Theta)$  and  $\mathbf{t}(\varepsilon) = (\mathbf{a}, z)$ , then  $\mathbb{Z} \models \Theta(\mathbf{0}, z)$ .

Given  $\mathbf{n} \in [0, D-1]^*$  and  $\mathbf{x}_i \in \{\mathbf{x}_1, \dots, \mathbf{x}_\beta\}$ , in the following, let us use  $\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i)$  to denote the data value  $z_i$  if  $\mathbf{t}^*(\mathbf{n}) = (z_1, \dots, z_i, \dots, z_\beta)$ . Similarly, let us use  $\mathbf{t}^*(\mathbf{n})(\mathfrak{d}_1)$  to denote  $\mathfrak{d}_1$  and  $\mathbf{t}^*(\mathbf{n})(\mathfrak{d}_\alpha)$  to denote  $\mathfrak{d}_\alpha$ .

*Ad absurdum*, suppose there exists  $(\mathbf{n}, \mathbf{x}_i) \in U_0 \cup U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$  in  $G_{\mathbf{t}}^C$  such that  $\text{slen}(\mathbf{n}, \mathbf{x}_i) = \omega$ . We prove the claim for the case  $(\mathbf{n}, \mathbf{x}_i) \in U_0$ ; the proof for  $(\mathbf{n}, \mathbf{x}) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$  is analogous. Recall that, by definition,  $\text{slen}(\mathbf{n}, \mathbf{x}_i) = \text{slen}((\mathbf{n}, \mathbf{x}_i), (\mathbf{n}, \mathfrak{d}_1))$ . Define  $\Delta = \mathfrak{d}_1 - \mathbf{t}^*(\mathbf{n})(\mathbf{x}_i)$ . Let  $\pi$  be a path  $(\mathbf{n}_0, \mathbf{x}\mathfrak{d}_0) \xrightarrow{\sim} (\mathbf{n}_1, \mathbf{x}\mathfrak{d}_1) \xrightarrow{\sim} \dots \xrightarrow{\sim} (\mathbf{n}_k, \mathbf{x}\mathfrak{d}_k)$  such that  $(\mathbf{n}_0, \mathbf{x}\mathfrak{d}_0) = (\mathbf{n}, \mathbf{x}_i)$ ,  $(\mathbf{n}_k, \mathbf{x}\mathfrak{d}_k) = (\mathbf{n}, \mathfrak{d}_1)$ , and  $\text{slen}(\pi) > \Delta$ . (Such a path must exist by assumption.) By Lemma 7, we have  $\mathbf{t}^*(\mathbf{n}_{i-1})(\mathbf{x}\mathfrak{d}_{i-1}) \sim_i \mathbf{t}^*(\mathbf{n}_i)(\mathbf{x}\mathfrak{d}_i)$  for all  $1 \leq i \leq k$ . But this implies that there are more than  $\Delta$  different data values in the interval  $[\mathbf{t}^*(\mathbf{n}_0)(\mathbf{x}\mathfrak{d}_0), \mathbf{t}^*(\mathbf{n}_k)(\mathbf{x}\mathfrak{d}_k)] = [\mathbf{t}^*(\mathbf{n})(\mathbf{x}_i), \mathbf{t}^*(\mathbf{n})(\mathfrak{d}_1)]$ , contradiction.

(II)  $\Rightarrow$  (I) Suppose that for all  $(\mathbf{n}, \mathbf{x})$  in  $(U_0 \cup U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2})$  in  $G_{\mathbf{t}}^C$  we have  $\text{slen}(\mathbf{n}, \mathbf{x}) < \omega$ . We define the mapping  $g : [0, D-1]^* \times \{\mathbf{x}_1, \dots, \mathbf{x}_\beta\} \rightarrow \mathbb{Z}$  as follows:

- $g(\mathbf{n}, \mathbf{x}) \stackrel{\text{def}}{=} \mathfrak{d}_1 + (i-1) \Leftrightarrow (\mathbf{n}, \mathbf{x}) \in U_i$  for some  $i \in [1, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1]$ ,
- $g(\mathbf{n}, \mathbf{x}) \stackrel{\text{def}}{=} \mathfrak{d}_1 - \text{slen}(\mathbf{n}, \mathbf{x}) \Leftrightarrow (\mathbf{n}, \mathbf{x}) \in U_0$ , and
- $g(\mathbf{n}, \mathbf{x}) \stackrel{\text{def}}{=} \mathfrak{d}_\alpha + \text{slen}(\mathbf{n}, \mathbf{x}) \Leftrightarrow (\mathbf{n}, \mathbf{x}) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$ .

Recall that  $\{U_i \mid i \in [0, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2]\}$  is a partition of  $[0, D-1]^* \times \mathbb{T}(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$  so that  $g$  is indeed well-defined. Now define  $\mathbf{t}' : [0, D-1]^* \rightarrow \Sigma \times \mathbb{Z}^\beta$  by  $\mathbf{t}'(\mathbf{n}) \stackrel{\text{def}}{=} (\mathbf{a}, (g(\mathbf{n}, \mathbf{x}_1), \dots, g(\mathbf{n}, \mathbf{x}_\beta)))$  for all  $\mathbf{n} \in [0, D-1]^*$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \cdot)$ . We prove that  $\mathbf{t}'$  witnesses the satisfaction of  $\mathbf{t}$ . For this, let  $\mathbf{n} \in [0, D-1]^*$  and suppose  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$ . Below, we verify that  $\mathbb{Z} \models \Theta'(\mathbf{t}'(\mathbf{n}))$ , where  $\Theta'$  is  $\Theta$  restricted to  $\mathbf{x}'_1, \dots, \mathbf{x}'_\beta$ . In a similar way, one can verify that for all  $\mathbf{n} \cdot j \in [0, D-1]^+$  with  $\mathbf{t}(\mathbf{n} \cdot j) = (\mathbf{a}, \Theta)$ , we have  $\mathbb{Z} \models \Theta'(\mathbf{t}'(\mathbf{n}), \mathbf{t}'(\mathbf{n} \cdot j))$  (plus the special condition at the root  $\varepsilon$ ). We omit this general case as it is very similar to the verification provided below, only more cases need to be considered.

- Suppose  $\mathbf{x}' = \mathfrak{d}_1 + (i-1) \in \Theta$  for some  $i \in [1, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1]$ . By definition of  $G_{\mathbf{t}}^C$ , we have  $(\mathbf{n}, \mathbf{x}) \in U_i$ . By definition,  $g(\mathbf{n}, \mathbf{x}) = \mathfrak{d}_1 + (i-1)$ .
- Suppose  $\mathbf{x}' < \mathfrak{d}_1 \in \Theta$ . By definition of  $G_{\mathbf{t}}^C$ , we have  $(\mathbf{n}, \mathbf{x}) \in U_0$ . By definition,  $g(\mathbf{n}, \mathbf{x}) = \mathfrak{d}_1 - \text{slen}(\mathbf{n}, \mathbf{x})$ . By definition of  $G_{\mathbf{t}}^C$ , we also have  $(\mathbf{n}, \mathfrak{d}_1) \in U_1$  and hence  $(\mathbf{n}, \mathbf{x}) \xrightarrow{\sim} (\mathbf{n}, \mathfrak{d}_1)$ . Hence  $\text{slen}(\mathbf{n}, \mathbf{x}) \geq 1$ , so that indeed  $g(\mathbf{n}, \mathbf{x}) < \mathfrak{d}_1$ .
- Suppose  $\mathbf{x}' > \mathfrak{d}_\alpha \in \Theta$ . By definition of  $G_{\mathbf{t}}^C$ , we have  $(\mathbf{n}, \mathbf{x}) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$ . By definition,  $g(\mathbf{n}, \mathbf{x}) = \mathfrak{d}_\alpha + \text{slen}(\mathbf{n}, \mathbf{x})$ . We also have  $(\mathbf{n}, \mathfrak{d}_\alpha) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1}$  and hence  $(\mathbf{n}, \mathfrak{d}_\alpha) \xrightarrow{\sim} (\mathbf{n}, \mathbf{x})$ . Hence  $\text{slen}(\mathbf{n}, \mathbf{x}) \geq 1$ , so that indeed  $g(\mathbf{n}, \mathbf{x}) > \mathfrak{d}_\alpha$ .
- Suppose  $\mathbf{x}' < \mathbf{y}' \in \Theta$ . We distinguish the following cases.



- Suppose  $(\mathbf{n}, \mathbf{x}) \in U_i$  and  $(\mathbf{n}, \mathbf{y}) \in U_{i'}$  for some  $i, j \in [1, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1]$ . This also implies  $\mathbf{x}' = \mathfrak{d}_i, \mathbf{y}' = \mathfrak{d}_{i'} \in \Theta$ . Recall that  $\Theta$  is satisfiable, hence  $i < i'$  must hold. By definition,  $g(\mathbf{n}, \mathbf{x}) = \mathfrak{d}_1 + (i - 1)$  and  $g(\mathbf{n}, \mathbf{y}) = \mathfrak{d}_1 + (i' - 1)$ , and hence clearly  $g(\mathbf{n}, \mathbf{x}) < g(\mathbf{n}, \mathbf{y})$ .
  - Suppose  $(\mathbf{n}, \mathbf{x}) \in U_0$  and  $(\mathbf{n}, \mathbf{y}) \in U_j$  for some  $j \in [1, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1]$ . This also implies  $\mathbf{x}' < \mathfrak{d}_1 \in \Theta$ . We have proved above that  $g(\mathbf{n}, \mathbf{x}) < \mathfrak{d}_1$ . By definition,  $g(\mathbf{n}, \mathbf{y}) = \mathfrak{d}_1 + (j - 1) \geq \mathfrak{d}_1$ . Hence  $g(\mathbf{n}, \mathbf{x}) < g(\mathbf{n}, \mathbf{y})$ .
  - Suppose  $(\mathbf{n}, \mathbf{x}) \in U_0$  and  $(\mathbf{n}, \mathbf{y}) \in U_0$ . By definition,  $g(\mathbf{n}, \mathbf{x}) = \mathfrak{d}_1 - \text{slen}(\mathbf{n}, \mathbf{x})$  and  $g(\mathbf{n}, \mathbf{y}) = \mathfrak{d}_1 - \text{slen}(\mathbf{n}, \mathbf{y})$ . By assumption and definition of  $G_{\mathbf{t}}^c$ , we have  $(\mathbf{n}, \mathbf{x}) \prec (\mathbf{n}, \mathbf{y})$ . Recall that  $\text{slen}(\mathbf{n}, \mathbf{x}) = \text{slen}((\mathbf{n}, \mathbf{x}), (\mathbf{n}, \mathfrak{d}_1))$  and  $\text{slen}(\mathbf{n}, \mathbf{y}) = \text{slen}((\mathbf{n}, \mathbf{y}), (\mathbf{n}, \mathfrak{d}_1))$ . By construction of  $G_{\mathbf{t}}^c$ , we have  $(\mathbf{n}, \mathfrak{d}_1) \in U_1$ , and hence  $(\mathbf{n}, \mathbf{x}) \prec (\mathbf{n}, \mathfrak{d}_1)$  and  $(\mathbf{n}, \mathbf{y}) \prec (\mathbf{n}, \mathfrak{d}_1)$ . This clearly yields  $\text{slen}(\mathbf{n}, \mathbf{x}) \geq \text{slen}(\mathbf{n}, \mathbf{y}) + 1$ . Hence  $\text{slen}(\mathbf{n}, \mathbf{x}) > \text{slen}(\mathbf{n}, \mathbf{y})$ , so that indeed  $g(\mathbf{n}, \mathbf{x}) < g(\mathbf{n}, \mathbf{y})$ .
  - Suppose  $(\mathbf{n}, \mathbf{x}) \in U_0$  and  $(\mathbf{n}, \mathbf{y}) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$ . This implies  $\mathbf{x}' < \mathfrak{d}_1, \mathbf{y}' > \mathfrak{d}_\alpha \in \Theta$ . We have proved above that  $g(\mathbf{n}, \mathbf{x}) < \mathfrak{d}_1$  and  $g(\mathbf{n}, \mathbf{y}) > \mathfrak{d}_\alpha$ . Hence  $g(\mathbf{n}, \mathbf{x}) < g(\mathbf{n}, \mathbf{y})$ .
  - Suppose  $(\mathbf{n}, \mathbf{x}) \in U_i$  for some  $i \in [1, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 1]$  and  $(\mathbf{n}, \mathbf{y}) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$ . This also implies  $\mathbf{y}' > \mathfrak{d}_\alpha \in \Theta$ . We have proved above that  $g(\mathbf{n}, \mathbf{y}) > \mathfrak{d}_\alpha$ . By definition,  $g(\mathbf{n}, \mathbf{x}) = \mathfrak{d}_1 + (i - 1) \leq \mathfrak{d}_\alpha$ . Hence  $g(\mathbf{n}, \mathbf{x}) < g(\mathbf{n}, \mathbf{y})$ .
  - Suppose  $(\mathbf{n}, \mathbf{x}), (\mathbf{n}, \mathbf{y}) \in U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$ . By definition,  $g(\mathbf{n}, \mathbf{x}) = \mathfrak{d}_\alpha + \text{slen}(\mathbf{n}, \mathbf{x})$  and  $g(\mathbf{n}, \mathbf{y}) = \mathfrak{d}_\alpha + \text{slen}(\mathbf{n}, \mathbf{y})$ . By assumption and definition of  $G_{\mathbf{t}}^c$ , we have  $(\mathbf{n}, \mathbf{x}) \prec (\mathbf{n}, \mathbf{y})$ . Recall that  $\text{slen}(\mathbf{n}, \mathbf{x}) = \text{slen}((\mathbf{n}, \mathfrak{d}_\alpha), (\mathbf{n}, \mathbf{x}))$  and  $\text{slen}(\mathbf{n}, \mathbf{y}) = \text{slen}((\mathbf{n}, \mathfrak{d}_\alpha), (\mathbf{n}, \mathbf{y}))$ . By construction of  $G_{\mathbf{t}}^c$ , we have  $(\mathbf{n}, \mathfrak{d}_\alpha) \in U_{(\mathfrak{d}_\alpha \mathfrak{d}_1) + 1}$ , and hence  $(\mathbf{n}, \mathfrak{d}_\alpha) \prec (\mathbf{n}, \mathbf{x})$  and  $(\mathbf{n}, \mathfrak{d}_\alpha) \prec (\mathbf{n}, \mathbf{y})$ . This clearly yields  $\text{slen}(\mathbf{n}, \mathbf{y}) \geq \text{slen}(\mathbf{n}, \mathbf{x}) + 1$  because  $(\mathbf{n}, \mathbf{x}) \prec (\mathbf{n}, \mathbf{y})$ . Hence  $\text{slen}(\mathbf{n}, \mathbf{x}) < \text{slen}(\mathbf{n}, \mathbf{y})$ , so that indeed  $g(\mathbf{n}, \mathbf{x}) < g(\mathbf{n}, \mathbf{y})$ .
- Suppose  $\mathbf{x}' = \mathbf{y}' \in \Theta$  with  $\mathbf{t}(\mathbf{n}) = (\mathbf{a}, \Theta)$ . Since  $\Theta$  is satisfiable, for some  $i \in [0, (\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2]$ , we have  $(\mathbf{n}, \mathbf{x}), (\mathbf{n}, \mathbf{y}) \in U_i$ . If  $i$  is different from 0 and  $(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2$ , necessarily  $g(\mathbf{n}, \mathbf{x}) = g(\mathbf{n}, \mathbf{y})$ . Otherwise, since  $(\mathbf{n}, \mathbf{x}) \xrightarrow{=} (\mathbf{n}, \mathbf{y})$  in  $G_{\mathbf{t}}^c$ , we have  $\text{slen}(\mathbf{n}, \mathbf{x}) = \text{slen}(\mathbf{n}, \mathbf{y})$ . Consequently,  $g(\mathbf{n}, \mathbf{x}) = g(\mathbf{n}, \mathbf{y})$  too.  $\square$

## 7.2 Final steps in the proof of Proposition 4

When  $\mathbf{t}$  is regular and  $G_{\mathbf{t}}^c$  has an element with an infinite strict length, the proof of Proposition 4 firstly consists in showing that the existence of paths with infinitely increasing strict lengths can be further constrained so that the paths are without detours and with a strict discipline to visit the tree structure underlying  $G_{\mathbf{t}}^c$ . That is why, below, we introduce several restrictions on paths followed by forthcoming Lemma 25 stating the main properties we expect. Actually, this approach is borrowed from the proof sketch for [LOS20, Lemma 22] as well as from the detailed developments in Labai's PhD thesis [Lab21, Section 5.2]. Hence, the main ideas in the developments below are due to [LOS20, Lab21], sometimes adapted and completed to meet our needs (for instance, we introduce a simple and explicit taxonomy on paths).

A path  $\pi = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim^1} \dots \xrightarrow{\sim^n} (\mathbf{n}_n, \mathbf{x}\mathbf{d}_n)$  is *direct* iff (1.)–(3.) hold:

1. for all  $i \in [1, n - 1]$ ,  $\mathbf{n}_i \neq \mathbf{n}_{i+1}$  (if  $n = 1$ , then we authorise  $\mathbf{n}_0 = \mathbf{n}_1$ ),
2. for all  $j \neq i$ ,  $(\mathbf{n}_i, \mathbf{x}\mathbf{d}_i) \neq (\mathbf{n}_j, \mathbf{x}\mathbf{d}_j)$  (no element is visited twice),
3. for all  $i < j$ , if  $\mathbf{n}_i = \mathbf{n}_j$  and  $n > 1$ , then  $<$  belongs to  $\{\sim_{i+1}, \dots, \sim_j\}$  (revisiting a node implies some progress in the strict length).

We write  $\text{slen}^d((\mathbf{n}, \mathbf{xd}), (\mathbf{n}', \mathbf{xd}'))$  to denote the strict length between  $(\mathbf{n}, \mathbf{xd})$  and  $(\mathbf{n}', \mathbf{xd}')$  based on *direct* paths only. Obviously,

$$\text{slen}^d((\mathbf{n}, \mathbf{xd}), (\mathbf{n}', \mathbf{xd}')) \leq \text{slen}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}', \mathbf{xd}')).$$

Similarly, a path  $\pi = (\mathbf{n}_0, \mathbf{xd}_0) \xrightarrow{\sim^1} \dots \xrightarrow{\sim^n} (\mathbf{n}_n, \mathbf{xd}_n)$  is *rooted* iff all the  $\mathbf{n}_i$ 's are either descendants of the initial node  $\mathbf{n}_0$ , or equal to it. We write  $\text{slen}^{dr}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}, \mathbf{xd}'))$  (same node  $\mathbf{n}$  for both elements) to denote the strict length between  $(\mathbf{n}, \mathbf{xd})$  and  $(\mathbf{n}, \mathbf{xd}')$  based on *direct and rooted* paths only. This definition extends to  $\text{slen}^{dr}(\mathbf{n}, \mathbf{xd})$  with  $(\mathbf{n}, \mathbf{xd}) \in (U_0 \cup U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2})$ . Obviously,

$$\text{slen}^{dr}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}, \mathbf{xd}')) \leq \text{slen}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}, \mathbf{xd}')) \quad \text{slen}^{dr}(\mathbf{n}, \mathbf{xd}) \leq \text{slen}(\mathbf{n}, \mathbf{xd}).$$

The last restriction we consider on direct and rooted paths is to be made of a unique descending part followed by a unique ascending part, the terms ‘descending’ and ‘ascending’ refer to the underlying tree structure for  $[0, D-1]^*$ . A path  $\pi = (\mathbf{n}_0, \mathbf{xd}_0) \xrightarrow{\sim^1} \dots \xrightarrow{\sim^n} (\mathbf{n}_n, \mathbf{xd}_n)$  is  $\mathcal{U}$ -*structured* iff there is  $i \in [0, n]$  such that  $\mathbf{n}_0 <_{\text{pre}} \mathbf{n}_1 <_{\text{pre}} \dots <_{\text{pre}} \mathbf{n}_i$  and  $\mathbf{n}_n <_{\text{pre}} \mathbf{n}_{n-1} <_{\text{pre}} \dots <_{\text{pre}} \mathbf{n}_i$ , where  $<_{\text{pre}}$  denotes the (strict) prefix relation (in this context, we get the parent-child relation in the tree  $[0, D-1]^*$ ). We write  $\text{slen}^{\mathcal{U}}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}, \mathbf{xd}'))$  (same node  $\mathbf{n}$  for both elements) to denote the strict length between  $(\mathbf{n}, \mathbf{xd})$  and  $(\mathbf{n}, \mathbf{xd}')$  based on *direct, rooted and  $\mathcal{U}$ -structured* paths only. This definition extends to  $\text{slen}^{\mathcal{U}}(\mathbf{n}, \mathbf{xd})$  with  $(\mathbf{n}, \mathbf{xd}) \in (U_0 \cup U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2})$ . Again,  $\text{slen}^{\mathcal{U}}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}, \mathbf{xd}')) \leq \text{slen}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}, \mathbf{xd}'))$  and  $\text{slen}^{\mathcal{U}}(\mathbf{n}, \mathbf{xd}) \leq \text{slen}(\mathbf{n}, \mathbf{xd})$ .

In Figure 5, we illustrate different kinds of paths on a subgraph of  $G_t^C$  with  $\beta = 2$ ,  $\alpha = 1$  and  $\mathfrak{d}_1 = 7$ .

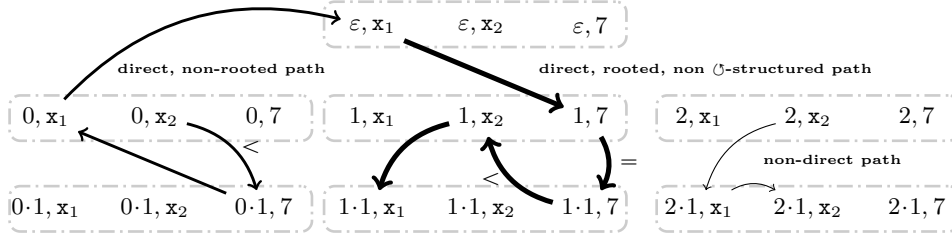


Fig. 5: Different kinds of paths in  $G_t^C$

The equivalence between (II)(1.) and (II)(4.) in Lemma 25 below, is the main property to prove Proposition 4. The proof for the equivalence between (II)(1.) and (II)(3.) (resp. between (II)(3.) and (II)(4.)) is inspired from the proof sketch of [Lab21, Lemma 5.20] (resp. from the proof of [Lab21, Lemma 5.27]). In both cases, we provide several substantial adjustments.

**Lemma 25.**

(I) If  $\text{slen}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}', \mathbf{xd}')) < \omega$ , then

$$\frac{\text{slen}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}', \mathbf{xd}'))}{\beta + 2} \leq \text{slen}^d((\mathbf{n}, \mathbf{xd}), (\mathbf{n}', \mathbf{xd}')) \leq \text{slen}((\mathbf{n}, \mathbf{xd}), (\mathbf{n}', \mathbf{xd}'))$$

(II) The statements below are equivalent.

1. There is  $(\mathbf{n}, \mathbf{x})$  such that  $\text{slen}(\mathbf{n}, \mathbf{x}) = \omega$ .
2. There is  $(\mathbf{n}, \mathbf{x})$  such that  $\text{slen}^d(\mathbf{n}, \mathbf{x}) = \omega$ .
3. There is  $(\mathbf{n}, \mathbf{x})$  such that  $\text{slen}^{dr}(\mathbf{n}, \mathbf{x}) = \omega$ .
4. There is  $(\mathbf{n}, \mathbf{x})$  such that  $\text{slen}^j(\mathbf{n}, \mathbf{x}) = \omega$ .

*Proof.* (I) Since direct paths are paths and the strict length between two elements is computed by taking the supremum, we get

$$\text{slen}^d((\mathbf{n}, \mathbf{x}), (\mathbf{n}', \mathbf{x}')) \leq \text{slen}((\mathbf{n}, \mathbf{x}), (\mathbf{n}', \mathbf{x}')).$$

In order to show that

$$\frac{\text{slen}((\mathbf{n}, \mathbf{x}), (\mathbf{n}', \mathbf{x}'))}{\beta + 2} \leq \text{slen}^d((\mathbf{n}, \mathbf{x}), (\mathbf{n}', \mathbf{x}')),$$

we establish that for any path  $\pi = (\mathbf{n}_0, \mathbf{x}_{d_0}) \xrightarrow{\sim 1} \dots \xrightarrow{\sim n} (\mathbf{n}_n, \mathbf{x}_{d_n})$ , there is a direct path  $\pi'$  from  $(\mathbf{n}_0, \mathbf{x}_{d_0})$  to  $(\mathbf{n}_n, \mathbf{x}_{d_n})$  with  $\text{slen}(\pi') \geq \frac{\text{slen}(\pi)}{\beta + 2}$ .

Given a path  $\pi$  of the above form, we transform it a finite amount of times leading to a final path  $\pi'$ . If there are  $i < j$  such that  $\mathbf{n}_i = \mathbf{n}_{i+1} = \dots = \mathbf{n}_j$  (and no way to extend the interval of indices  $[i, j]$  while satisfying the sequence of equalities), we replace the subpath  $(\mathbf{n}_i, \mathbf{x}_{d_i}) \dots (\mathbf{n}_j, \mathbf{x}_{d_j})$  in  $\pi$  by a shortcut. Several cases need to be distinguished.

- If  $i > 0$ , then  $(\mathbf{n}_{i-1}, \mathbf{x}_{d_{i-1}})$  and  $(\mathbf{n}_j, \mathbf{x}_{d_j})$  are neighbours and therefore by Lemma 22, we can safely replace  $(\mathbf{n}_{i-1}, \mathbf{x}_{d_{i-1}}) \dots (\mathbf{n}_j, \mathbf{x}_{d_j})$  by  $(\mathbf{n}_{i-1}, \mathbf{x}_{d_{i-1}}) \xrightarrow{\sim} (\mathbf{n}_j, \mathbf{x}_{d_j})$  for some  $\sim \in \{=, <\}$  leading to a new value for  $\pi$ . The label  $\sim$  on the edge depends whether  $<$  occurs in the subpath from  $(\mathbf{n}_{i-1}, \mathbf{x}_{d_{i-1}})$  to  $(\mathbf{n}_j, \mathbf{x}_{d_j})$ . Moreover, the strict length of the new path is decreased by at most  $\beta + 1$  (and we do not meet again an element with the node  $\mathbf{n}_i$ ).
- If  $i = 0$  and  $j < n$ , then  $(\mathbf{n}_0, \mathbf{x}_{d_0})$  and  $(\mathbf{n}_{j+1}, \mathbf{x}_{d_{j+1}})$  are neighbours and therefore by Lemma 22, we can safely replace  $(\mathbf{n}_0, \mathbf{x}_{d_0}) \dots (\mathbf{n}_{j+1}, \mathbf{x}_{d_{j+1}})$  by  $(\mathbf{n}_0, \mathbf{x}_{d_0}) \xrightarrow{\sim} (\mathbf{n}_{j+1}, \mathbf{x}_{d_{j+1}})$  for some  $\sim \in \{=, <\}$  leading to a new value for  $\pi$ . Again, the strict length of the new path is decreased by at most  $\beta + 1$ .
- Finally, if  $i = 0$  and  $j = n$ , then by Lemma 22, we replace  $(\mathbf{n}_0, \mathbf{x}_{d_0}) \dots (\mathbf{n}_n, \mathbf{x}_{d_n})$  by  $(\mathbf{n}_0, \mathbf{x}_{d_0}) \xrightarrow{\sim} (\mathbf{n}_n, \mathbf{x}_{d_n})$  for some  $\sim \in \{=, <\}$  leading to a new value for  $\pi$ . Again, the strict length of the new path is decreased by at most  $\beta + 1$ .

In the second step, we proceed as follows. If there are  $i + 1 < j$  such that  $(\mathbf{n}_i, \mathbf{x}_{d_i}) = (\mathbf{n}_j, \mathbf{x}_{d_j})$  ( $j - i > 1$  because otherwise we could apply the previous transformation), then by Lemma 23, the subpath  $(\mathbf{n}_i, \mathbf{x}_{d_i}) \dots (\mathbf{n}_j, \mathbf{x}_{d_j})$  in  $\pi$  contains no edge labelled by  $<$  and therefore by Lemma 22, we can safely replace  $(\mathbf{n}_i, \mathbf{x}_{d_i}) \dots (\mathbf{n}_j, \mathbf{x}_{d_j})$  by  $(\mathbf{n}_i, \mathbf{x}_{d_i})$ , leading to a new value for  $\pi$ . Note that the strict length of the new path is unchanged. Similarly, if there are  $i + 1 < j$  such that  $\mathbf{n}_i = \mathbf{n}_j$  and  $\sim_{i+1} = \dots = \sim_j$  is equal to  $'=''$ , we can safely remove the subpath  $(\mathbf{n}_i, \mathbf{x}_{d_i}) \dots (\mathbf{n}_j, \mathbf{x}_{d_j})$  by Lemma 22 along the lines of the previous transformations (details are omitted). We proceed as many times as necessary until the path  $\pi$  becomes direct. As a consequence, the strict length of the final direct path is at least equal to the strict length of the initial value for  $\pi$  divided by  $\beta + 2$ .

(II) Obviously, (4) implies (3), (3) implies (2) and (2) implies (1). It remains to prove that (1) implies (2), (2) implies (3) and (3) implies (4). By the proof of (I), we get that (1) implies (2).

Let us show that (2) implies (3). Let  $(\mathbf{n}, \mathbf{x})$  be an element of  $G_{\mathbf{t}}^C$  such that  $\text{slen}^d(\mathbf{n}, \mathbf{x}) = \omega$ . The element  $(\mathbf{n}, \mathbf{x})$  belongs to  $U_0 \cup U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$  in  $G_{\mathbf{t}}^C$ . Suppose that  $(\mathbf{n}, \mathbf{x})$  is in  $U_0$  (we omit the other case as it admits a similar analysis). By definition of  $\text{slen}^d(\mathbf{n}, \mathbf{x})$ , there is a family of direct paths  $(\pi_i)_{i \in \mathbb{N}}$  from  $(\mathbf{n}, \mathbf{x})$  to  $(\mathbf{n}, \mathfrak{d}_1)$  such that  $\text{slen}(\pi_i) \geq i$ .

For each path in the family  $(\pi_i)_{i \in \mathbb{N}}$ , below, we define its *maximal type* as an element of the set  $Types \stackrel{\text{def}}{=} [-1, D-1] \times \mathbf{T}(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$ . Since  $Types$  is finite, there is  $(j, \mathbf{x}\mathbf{d}, \mathbf{x}\mathbf{d}') \in Types$  such that for infinitely many  $i$ ,  $\pi_i$  has maximal type  $(j, \mathbf{x}\mathbf{d}, \mathbf{x}\mathbf{d}')$ , which shall allow us to conclude easily. Firstly, let us provide a few definitions.

Let  $\pi = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \cdots (\mathbf{n}_m, \mathbf{x}\mathbf{d}_m)$  be a path in  $(\pi_i)_{i \in \mathbb{N}}$  of strict length at least  $3(D+1)\gamma$  with  $\gamma \stackrel{\text{def}}{=} \lfloor \frac{\beta+2}{2} \rfloor$ . This means that  $(\mathbf{n}, \mathbf{x}) = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0)$  and  $(\mathbf{n}, \mathfrak{d}_1) = (\mathbf{n}_m, \mathbf{x}\mathbf{d}_m)$ . Since  $\pi$  is direct, there are at most  $\beta+2$  positions in  $\pi$  visiting an element on the node  $\mathbf{n}$  (called an *n-element* later on). Such positions are written  $i_0 < i_1 < \cdots < i_s$  with  $i_0 = 0$  and  $i_s = m$ . For each  $h \in [0, s-1]$ , we write  $\pi_h^\dagger$  to denote the subpath of  $\pi$  below:

$$\pi_h^\dagger \stackrel{\text{def}}{=} (\mathbf{n}_{i_h+1}, \mathbf{x}\mathbf{d}_{i_h+1}) \cdots (\mathbf{n}_{i_{h+1}-1}, \mathbf{x}\mathbf{d}_{i_{h+1}-1}).$$

The *type* of  $\pi_h^\dagger$ , written  $t(\pi_h^\dagger)$ , is the triple  $(j, \mathbf{x}\mathbf{d}, \mathbf{x}\mathbf{d}') \in Types$  defined as follows:

- $\mathbf{x}\mathbf{d} \stackrel{\text{def}}{=} \mathbf{x}\mathbf{d}_{i_h+1}$  (entrance term) and  $\mathbf{x}\mathbf{d}' \stackrel{\text{def}}{=} \mathbf{x}\mathbf{d}_{i_{h+1}-1}$  (exit term).
- If  $\mathbf{n}_{i_h+1} = \mathbf{n} \cdot j'$  for some  $j' \in [0, D-1]$ , then  $j \stackrel{\text{def}}{=} j'$ . Otherwise  $j \stackrel{\text{def}}{=} -1$ . ( $j$  is a direction, where  $-1$  means “to the parent node” and  $j \in [0, D-1]$  “to the  $j$ th child node”)

A few useful properties are worth being stated.

- $\mathbf{n}_{i_h+1} = \mathbf{n}_{i_{h+1}-1}$ ,  $\pi_h^\dagger$  does not contain an *n-element* and  $\pi_h^\dagger$  is rooted.
- By Lemma 22,  $(\mathbf{n}, \mathbf{x}) \rightsquigarrow (\mathbf{n}_{i_h+1}, \mathbf{x}\mathbf{d}_{i_h+1})$  and  $(\mathbf{n}_{i_{h+1}-1}, \mathbf{x}\mathbf{d}_{i_{h+1}-1}) \rightsquigarrow' (\mathbf{n}, \mathfrak{d}_1)$  for some  $\sim, \sim' \in \{<, =\}$ .
- For all  $h \neq h'$ , if  $t(\pi_h^\dagger) = (j_1, \mathbf{x}\mathbf{d}_1, \mathbf{x}\mathbf{d}'_1)$ ,  $t(\pi_{h'}^\dagger) = (j_2, \mathbf{x}\mathbf{d}_2, \mathbf{x}\mathbf{d}'_2)$  and  $j_1 = j_2$ , then  $\mathbf{x}\mathbf{d}_1 \neq \mathbf{x}\mathbf{d}'_1$  and  $\mathbf{x}\mathbf{d}_2 \neq \mathbf{x}\mathbf{d}'_2$  ( $\pi$  is direct). As a consequence,  $s \leq (D+1)\gamma$  and there are no  $\pi_h^\dagger$  and  $\pi_{h'}^\dagger$  with  $h \neq h'$  having the same type.

As a conclusion, there is  $h$  such that the strict length of  $\pi_h^\dagger$  is at least

$$\lceil \frac{\text{slen}(\pi) - 2(D+1)\gamma}{(D+1)\gamma} \rceil$$

When  $\text{slen}(\pi) \geq 3(D+1)\gamma$ , the above value is at least one. Indeed, we subtract  $2(D+1)\gamma$  from  $\text{slen}(\pi)$  to take into account the edges from some *n-element* to the elements  $(\mathbf{n}_{i_h+1}, \mathbf{x}\mathbf{d}_{i_h+1})$ , and from the elements  $(\mathbf{n}_{i_{h+1}-1}, \mathbf{x}\mathbf{d}_{i_{h+1}-1})$  to some *n-element*. The maximal type of  $\pi$  is defined as a type  $t(\pi_h^\dagger)$  such that  $\text{slen}(\pi_h^\dagger)$  is maximal (we can also fix an arbitrary linear ordering on  $Types$  in case maximality of the (direct) strict lengths is witnessed strictly more than once). As announced earlier, there is  $(j, \mathbf{x}\mathbf{d}, \mathbf{x}\mathbf{d}') \in Types$  such that for infinitely many  $i$  (say belonging to the infinite set  $X \subseteq [3(D+1)\gamma, +\infty)$ ),  $\pi_i$  has maximal type  $(j, \mathbf{x}\mathbf{d}, \mathbf{x}\mathbf{d}')$ .

(a) If  $j \in [0, D-1]$ , we have  $\text{slen}^{dr}((\mathbf{n} \cdot j, \mathbf{x}\mathbf{d}), (\mathbf{n} \cdot j, \mathbf{x}\mathbf{d}')) = \omega$  because

$$\text{slen}^{dr}((\mathbf{n} \cdot j, \mathbf{x}\mathbf{d}), (\mathbf{n} \cdot j, \mathbf{x}\mathbf{d}')) \geq \sup_{i \in X} \lceil \frac{i - 2(D+1)\gamma}{(D+1)\gamma} \rceil$$

Since  $(\mathbf{n}, \mathbf{x}) \rightsquigarrow (\mathbf{n} \cdot j, \mathbf{x}\mathbf{d})$  and  $(\mathbf{n} \cdot j, \mathbf{x}\mathbf{d}') \rightsquigarrow' (\mathbf{n}, \mathfrak{d}_1)$  for some  $\sim, \sim' \in \{<, =\}$ , we get that  $\text{slen}^{dr}(\mathbf{n}, \mathbf{x}) = \omega$  too.

- (b) If  $j = -1$ , then let  $\mathbf{n}'$  be the parent of  $\mathbf{n}$ . We have  $\text{slen}^d((\mathbf{n}', \mathbf{x}\mathbf{d}), (\mathbf{n}', \mathbf{x}\mathbf{d}')) = \omega$  and  $(\mathbf{n}', \mathbf{x}\mathbf{d}') \xrightarrow{\sim} (\mathbf{n}, \mathbf{d}_1)$  for some  $\sim \in \{<, =\}$ . Since  $\mathbf{n}$  and  $\mathbf{n}'$  are neighbour nodes, we also get that  $(\mathbf{n}', \mathbf{x}\mathbf{d}') \xrightarrow{\sim} (\mathbf{n}', \mathbf{d}_1)$  by construction of  $G_{\mathbf{t}}^C$ . Moreover, by Lemma 23,  $\mathbf{x}\mathbf{d}$  is necessarily a variable. Consequently,  $\text{slen}^d(\mathbf{n}', \mathbf{x}\mathbf{d}) = \omega$  and we can apply the above construction but  $|\mathbf{n}'| < |\mathbf{n}|$ . This means that at some point, we must meet the case (a), which allows us eventually to identify some element  $(\mathbf{m}, \mathbf{y})$  such that  $\text{slen}^{dr}(\mathbf{m}, \mathbf{y}) = \omega$ . (Another way to proceed is to provide a proof by induction on  $|\mathbf{n}|$ . The base case  $|\mathbf{n}| = 0$  corresponds to  $\mathbf{n} = \varepsilon$  for which only case (a) can hold.)

Let us show that (3) implies (4). Let  $(\mathbf{n}, \mathbf{x})$  be an element of  $G_{\mathbf{t}}^C$  such that  $\text{slen}^{dr}(\mathbf{n}, \mathbf{x}) = \omega$ . The element  $(\mathbf{n}, \mathbf{x})$  belongs to  $U_0 \cup U_{(\mathbf{d}_\alpha - \mathbf{d}_1) + 2}$  in  $G_{\mathbf{t}}^C$ . Suppose that  $(\mathbf{n}, \mathbf{x})$  is in  $U_{(\mathbf{d}_\alpha - \mathbf{d}_1) + 2}$  (we omit the other case as it admits a similar analysis). By definition of  $\text{slen}^{dr}(\mathbf{n}, \mathbf{x})$ , there is a family of direct and rooted paths  $(\pi_i)_{i \in \mathbb{N}}$  from  $(\mathbf{n}, \mathbf{d}_\alpha)$  to  $(\mathbf{n}, \mathbf{x})$  such that  $\text{slen}(\pi_i) \geq i$ . Below, we show that  $\text{slen}^{\mathcal{J}}(\mathbf{n}, \mathbf{x}) = \omega$  (and therefore no need to witness infinity on another element as it may happen to prove that (2) implies (3)). To do so, for every  $j$ , we explain how to construct a directed, rooted and  $\mathcal{J}$ -structured path from  $(\mathbf{n}, \mathbf{d}_\alpha)$  to  $(\mathbf{n}, \mathbf{x})$  of strict length at least  $j$ . We use the auxiliary family  $(L_{i,j})_{i,j \in \mathbb{N}}$  of integers defined recursively.

- For all  $i \in \mathbb{N}$ ,  $L_{i,0} \stackrel{\text{def}}{=} i$  and for all  $j \in \mathbb{N}$ ,  $L_{i,j+1} \stackrel{\text{def}}{=} \lceil \frac{L_{i,j} - 2D\gamma}{D\gamma} \rceil$ .

By induction on  $j$ , it is easy to show that for each fixed  $j$ ,  $\sup\{L_{i,j} \mid i \in \mathbb{N}\} = \omega$ . Indeed, for the base case  $\sup\{L_{i,0} \mid i \in \mathbb{N}\} = \sup\{i \mid i \in \mathbb{N}\} = \omega$ . Suppose the property true for  $j$ , i.e.  $\sup\{L_{i,j} \mid i \in \mathbb{N}\} = \omega$ . Subtracting a constant and dividing by a constant preserves the limit behavior, consequently  $\sup\{\frac{L_{i,j} - 2D\gamma}{D\gamma} \mid i \in \mathbb{N}\} = \omega$  and therefore  $\sup\{L_{i,j+1} \mid i \in \mathbb{N}\} = \omega$ . As a consequence, for all  $j \in \mathbb{N}$ , there is  $N_j$  such that  $L_{N_j,j} \geq 1$ . For instance,  $N_0$  can take the value 1,  $N_1$  the value  $3D\gamma$  and  $N_2$  the value  $3(D\gamma)^2 + 2D\gamma$ . There are a few other properties that we can use (or just require) about  $(L_{i,j})_{i,j \in \mathbb{N}}$ .

- (P1) For all  $j$ ,  $\lceil \frac{N_{j+1} - 2D\gamma}{D\gamma} \rceil \geq N_j$  and  $N_{j+1} \geq N_j$ .
- (P2) For all  $i < i'$  and  $j$ ,  $L_{i,j} \leq L_{i',j}$ .
- (P3) For all  $i$  and  $j < j'$ ,  $L_{i,j} \geq L_{i,j'}$ .
- (P4) For all  $M, j \in \mathbb{N}$ ,  $M \geq N_j$  implies (if  $M, j \geq 1$ , then  $M - 1 \geq N_{j-1}$  and if  $M, j \geq 2$ , then  $M - 2 \geq N_{j-2}$ ).

Let  $\pi = (\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim^1} \dots \xrightarrow{\sim^m} (\mathbf{n}_m, \mathbf{x}\mathbf{d}_m)$  be a direct and rooted path from  $(\mathbf{n}, \mathbf{d}_\alpha)$  to  $(\mathbf{n}, \mathbf{x})$  of strict length at least  $N_j$ . For instance,  $\pi$  can take the value  $\pi_{N_j}$  from the family  $(\pi_i)_{i \in \mathbb{N}}$ . In the developments below, we build a path  $\pi'$  from  $(\mathbf{n}, \mathbf{d}_\alpha)$  to  $(\mathbf{n}, \mathbf{x})$  that is of strict length at least  $j$  and that is direct, rooted and  $\mathcal{J}$ -structured. To do so, we maintain three auxiliary paths while guaranteeing the satisfaction of an invariant.

- $\pi_{des}$  is a descending and direct path from  $(\mathbf{n}, \mathbf{d}_\alpha)$ . Its initial value is  $(\mathbf{n}, \mathbf{d}_\alpha)$ .
- $\pi_{asc}$  is an ascending and direct path to  $(\mathbf{n}, \mathbf{x})$ . Its initial value is  $(\mathbf{n}, \mathbf{x})$ .
- $\pi^\dagger$  is a direct and rooted path from the last element of  $\pi_{des}$  (say  $(\mathbf{n}^\dagger, \mathbf{x}\mathbf{d}_1^\dagger)$ ) to the first element of  $\pi_{asc}$  (say  $(\mathbf{n}^\dagger, \mathbf{x}\mathbf{d}_2^\dagger)$ ) and is a subpath of the initial path  $\pi$ . The first and last elements belong therefore to the same node  $\mathbf{n}^\dagger$ . The initial value for  $\pi^\dagger$  is  $\pi$ . By slight abuse, below we assume that  $\pi^\dagger$  can be also written  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim^1} \dots \xrightarrow{\sim^m} (\mathbf{n}_m, \mathbf{x}\mathbf{d}_m)$ .
- Let  $C$  be equal to  $\text{slen}(\pi_{des}) + \text{slen}(\pi_{asc})$ . The invariant we maintain is the following: if  $C \leq j$  then  $\text{slen}(\pi^\dagger) \geq N_{j-C}$ . Consequently, if  $\pi^\dagger$  is made of elements from the same node, then  $\text{slen}(\pi^\dagger) \leq 1$  and therefore  $C \geq j$ .

We transform these three paths with a process that terminates because the length of  $\pi^\dagger$  decreases strictly after each step. Moreover, we shall verify that the invariant holds after each step (sometimes after a step,  $\text{slen}(\pi^\dagger)$  and  $C$  are unchanged). Initially, we have  $\text{slen}(\pi^\dagger) \geq N_{j-C}$  because  $C = 0$ ,  $\pi^\dagger = \pi$  and  $\text{slen}(\pi) \geq N_j$ .

Let us define the *maximal type* of  $\pi^\dagger$  similarly to what is done earlier. Here,  $Types \stackrel{\text{def}}{=} [0, D-1] \times T(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$  (no value  $-1$  because  $\pi^\dagger$  is rooted). Since  $\pi^\dagger$  is direct, there is at most  $\beta + 2$  positions in  $\pi^\dagger$  visiting an element on the node  $\mathbf{n}^\dagger$ . Such positions are written  $i_0 < i_1 < \dots < i_s$  with  $i_0 = 0$ ,  $i_s = m$  (and  $s \leq D\gamma$ ). For each  $h \in [0, s-1]$ , we write  $\pi_h^{\dagger\dagger}$  to denote the subpath of  $\pi^\dagger$  below:

$$\pi_h^{\dagger\dagger} \stackrel{\text{def}}{=} (\mathbf{n}_{i_h+1}, \mathbf{xd}_{i_h+1}) \cdots (\mathbf{n}_{i_{h+1}-1}, \mathbf{xd}_{i_{h+1}-1}).$$

The *type* of  $\pi_h^{\dagger\dagger}$ , written  $t(\pi_h^{\dagger\dagger})$ , is the triple  $(j, \mathbf{xd}, \mathbf{xd}') \in Types$  defined as follows:  $\mathbf{xd} \stackrel{\text{def}}{=} \mathbf{xd}_{i_h+1}$ ,  $\mathbf{xd}' \stackrel{\text{def}}{=} \mathbf{xd}_{i_{h+1}-1}$  and  $\mathbf{n}_{i_h+1} = \mathbf{n} \cdot j$  for some  $j \in [0, D-1]$ . A few useful properties are worth being stated.

- $\mathbf{n}_{i_h+1} = \mathbf{n}_{i_{h+1}-1}$ ,  $\pi_h^{\dagger\dagger}$  does not contain an  $\mathbf{n}^\dagger$ -element and  $\pi_h^{\dagger\dagger}$  is rooted.
- By Lemma 22,  $(\mathbf{n}^\dagger, \mathbf{xd}_1^\dagger) \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_{i_h+1}, \mathbf{xd}_{i_h+1})$  for some  $\sim_{1,h}^\dagger \in \{<, =\}$  and  $(\mathbf{n}_{i_{h+1}-1}, \mathbf{xd}_{i_{h+1}-1}) \xrightarrow{\sim_{2,h}^\dagger} (\mathbf{n}^\dagger, \mathbf{xd}_2^\dagger)$  for some  $\sim_{2,h}^\dagger \in \{<, =\}$ .

There is  $h$  such that the strict length of  $\pi_h^{\dagger\dagger}$  is at least  $\lceil \frac{\text{slen}(\pi^\dagger) - 2D\gamma}{D\gamma} \rceil$ .

- If  $s = 1$ , then  $\mathbf{n}_1 = \mathbf{n}_{m-1}$ , there is a single  $\pi_0^{\dagger\dagger}$  and the paths are updated as follows.

- $\pi_{des}$  becomes  $\pi_{des} \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_1, \mathbf{xd}_1)$ ,  $\pi_{asc}$  becomes  $(\mathbf{n}_{m-1}, \mathbf{xd}_{m-1}) \xrightarrow{\sim_{2,h}^\dagger} \pi_{asc}$ .
- $\pi^\dagger$  takes the value  $\pi_0^{\dagger\dagger}$  if  $\pi_0^{\dagger\dagger}$  is made of elements from distinct nodes.

Note that the strict length of  $\pi^\dagger$  decreases by at most two (can be zero if  $\sim_{1,h}^\dagger$  and  $\sim_{2,h}^\dagger$  are both the equality) and thanks to (P4), the invariant is maintained. If  $\pi_0^{\dagger\dagger}$  is made of elements from the same node,  $\text{slen}(\pi_0^{\dagger\dagger}) \leq 1$  because  $\pi$  is direct and  $C \geq j$ . The final direct, rooted and  $\mathcal{U}$ -structured path is  $\pi_{des} \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_1, \mathbf{xd}_1) \xrightarrow{\sim} (\mathbf{n}_{m-1}, \mathbf{xd}_{m-1}) \xrightarrow{\sim_{2,h}^\dagger} \pi_{asc}$  and its strict length is at least  $j$ .

- Otherwise, i.e.  $s > 1$ . Pick  $h \in [0, s-1]$  such that

$$\text{slen}(\pi_h^{\dagger\dagger}) \geq \lceil \frac{\text{slen}(\pi^\dagger) - 2D\gamma}{D\gamma} \rceil$$

Note that  $(\mathbf{n}^\dagger, \mathbf{xd}_1^\dagger) \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_{i_h+1}, \mathbf{xd}_{i_h+1})$  and  $(\mathbf{n}_{i_{h+1}-1}, \mathbf{xd}_{i_{h+1}-1}) \xrightarrow{\sim_{2,h}^\dagger} (\mathbf{n}^\dagger, \mathbf{xd}_2^\dagger)$  and  $< \in \{\sim_{1,h}^\dagger, \sim_{2,h}^\dagger\}$  because  $h < s-1$  or  $0 < h$ , and  $\pi^\dagger$  is direct.

The paths are updated as follows.  $\pi_{des}$  becomes  $\pi_{des} \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_{i_h+1}, \mathbf{xd}_{i_h+1})$ ,  $\pi_{asc}$  becomes  $(\mathbf{n}_{i_{h+1}-1}, \mathbf{xd}_{i_{h+1}-1}) \xrightarrow{\sim_{2,h}^\dagger} \pi_{asc}$  and  $\pi^\dagger$  becomes  $\pi_h^{\dagger\dagger}$ . From  $\text{slen}(\pi^\dagger) \geq N_{j-C}$ , we get

$$\text{slen}(\pi_h^{\dagger\dagger}) \geq \lceil \frac{\text{slen}(\pi^\dagger) - 2D\gamma}{D\gamma} \rceil \geq \underbrace{N_{j-C-1} \geq N_{j-C-2}}_{\text{by (P1)}}$$

by the invariant, (P1) and (P2) (assuming that  $j - C - 2 \geq 0$ , otherwise we remove the appropriate expressions). Indeed, by the satisfaction of the invariant, we have  $\text{slen}(\pi^\dagger) \geq N_{j-C}$  and

$$\overbrace{\left\lceil \frac{\text{slen}(\pi^\dagger) - 2D\gamma}{D\gamma} \right\rceil}^{\text{by (P2)}} \geq \left\lceil \frac{N_{j-C} - 2D\gamma}{D\gamma} \right\rceil \geq N_{j-C-1}$$

Therefore the invariant is maintained. The process terminates and at termination, we have seen that this implies that we have built a direct, rooted and  $\mathcal{U}$ -structured path from  $(\mathbf{n}, \mathfrak{d}_\alpha)$  to  $(\mathbf{n}, \mathbf{x})$  that is of strict length at least  $j$ .  $\square$

Here is the final step to prove Proposition 4. Since violation of  $(\star^c)$  is witnessed on a single branch, the proof is analogous to the proof of [DD07, Lemma 6.2] and reformulates the final part of the proof of [Lab21, Lemma 5.16].

*Proof.* (Proposition 4) Let  $\mathbf{t}$  be a regular locally consistent symbolic tree. We write  $N$  to denote the number of distinct subtrees in  $\mathbf{t}$  ( $N$  exists because  $\mathbf{t}$  is regular).

It is easy to show that satisfiability of  $\mathbf{t}$  implies that  $G_{\mathbf{t}}^c$  satisfies  $(\star^c)$ . Indeed, if  $G_{\mathbf{t}}^c$  does not satisfy  $(\star^c)$ , then the existence of the witness path map  $p$  and the reverse path map  $rp$  forbids the possibility to interpret the variables so that  $\mathbf{t}$  is satisfiable (by Lemma 7). The main part of the proof consists in showing that if  $\mathbf{t}$  is not satisfiable, then  $G_{\mathbf{t}}^c$  does not satisfy  $(\star^c)$ . By Lemma 24, there is  $(\mathbf{n}, \mathbf{x})$  in  $U_0 \cup U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$  in  $G_{\mathbf{t}}^c$ , such that  $\text{slen}(\mathbf{n}, \mathbf{x}) = \omega$ . By Lemma 25, there is  $(\mathbf{n}, \mathbf{x})$  in  $U_0 \cup U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$  in  $G_{\mathbf{t}}^c$ , such that  $\text{slen}^{\mathcal{U}}(\mathbf{n}, \mathbf{x}) = \omega$ . Suppose that  $(\mathbf{n}, \mathbf{x})$  is in  $U_{(\mathfrak{d}_\alpha - \mathfrak{d}_1) + 2}$  (the other case is similar, and is therefore omitted as usual).

Let  $M = 2((\beta + 2)^2 N + 1)$  and  $\pi$  be a direct, rooted and  $\mathcal{U}$ -structured path of strict length at least  $M$  from  $(\mathbf{n}, \mathfrak{d}_\alpha)$  to  $(\mathbf{n}, \mathbf{x})$ . The path  $\pi$  is of the form below

$$(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \xrightarrow{\sim^1} (\mathbf{n}_1, \mathbf{x}\mathbf{d}_1) \cdots \xrightarrow{\sim^\ell} (\mathbf{n}_\ell, \mathbf{x}\mathbf{d}_\ell) = (\mathbf{n}_\ell, \mathbf{x}\mathbf{d}'_\ell) \xrightarrow{\sim^\ell} (\mathbf{n}_{\ell-1}, \mathbf{x}\mathbf{d}'_{\ell-1}) \cdots \xrightarrow{\sim^1} (\mathbf{n}_0, \mathbf{x}\mathbf{d}'_0),$$

where the following conditions hold for some  $j_1 \cdots j_\ell \in [0, D - 1]^*$ :

- $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) = (\mathbf{n}, \mathfrak{d}_\alpha)$  and  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}'_0) = (\mathbf{n}, \mathbf{x})$ .
- For all  $k \in [1, \ell]$ , we have  $\mathbf{n}_k = \mathbf{n} \cdot j_1 \cdots j_k$ .

Since  $\text{slen}(\pi) \geq 2((\beta + 2)^2 N + 1)$ , one of the paths among  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \cdots (\mathbf{n}_\ell, \mathbf{x}\mathbf{d}_\ell)$  and  $(\mathbf{n}_\ell, \mathbf{x}\mathbf{d}'_\ell) \cdots (\mathbf{n}_0, \mathbf{x}\mathbf{d}'_0)$  has strict length at least  $(\beta + 2)^2 N + 1$ . Below, we assume that the strict length of  $(\mathbf{n}_0, \mathbf{x}\mathbf{d}_0) \cdots (\mathbf{n}_\ell, \mathbf{x}\mathbf{d}_\ell)$  is at least  $(\beta + 2)^2 N + 1$  (the other case is similar, and is omitted herein). By the Pigeonhole Principle, there are  $K < K' \in [0, \ell]$  such that the subtree rooted at  $\mathbf{n}_K$  is equal to the subtree rooted at  $\mathbf{n}_{K'}$ ,  $\mathbf{x}\mathbf{d}_K = \mathbf{x}\mathbf{d}_{K'}$  and  $\mathbf{x}\mathbf{d}'_K = \mathbf{x}\mathbf{d}'_{K'}$ , and the subpath  $(\mathbf{n}_K, \mathbf{x}\mathbf{d}_K) \cdots (\mathbf{n}_{K'}, \mathbf{x}\mathbf{d}_{K'})$  has strict length at least one. Observe that by Lemma 22, for every  $k \in [K, K']$ , we have  $(\mathbf{n}_k, \mathbf{x}\mathbf{d}_k) \xrightarrow{\sim} (\mathbf{n}_k, \mathbf{x}\mathbf{d}'_k)$ . Moreover, since  $\mathbf{t}_{|\mathbf{n}_K} = \mathbf{t}_{|\mathbf{n}_{K'}}$ , for all  $(\mathbf{m}_1, \mathbf{x}\mathbf{d}_1^*), (\mathbf{m}_2, \mathbf{x}\mathbf{d}_2^*) \in \{\mathbf{n}_K, \dots, \mathbf{n}_{K'-1}\} \times \mathbf{T}(\beta, \mathfrak{d}_1, \mathfrak{d}_\alpha)$ , for all  $i \in \mathbb{N}$ ,  $m \in [0, K' - K - 1]$  and  $\sim \in \{<, =\}$ ,

$$(\mathbf{m}_1, \mathbf{x}\mathbf{d}_1^*) \xrightarrow{\sim} (\mathbf{m}_2, \mathbf{x}\mathbf{d}_2^*) \text{ iff}$$

$$(\mathbf{m}_1 \cdot (j_{K+1} \cdots j_{K'})^i (j_{K+1} \cdots j_{K+m}), \mathbf{x}\mathbf{d}_1) \xrightarrow{\sim} (\mathbf{m}_2 \cdot (j_{K+1} \cdots j_{K'})^i (j_{K+1} \cdots j_{K+m}), \mathbf{x}\mathbf{d}_2).$$

By convention, if  $m = 0$ , then  $(j_{K+1} \cdots j_{K+m}) = \varepsilon$ . Let us define  $(\mathbf{n}, \mathbf{x}\mathbf{d})$ ,  $(\mathbf{n}, \mathbf{x}\mathbf{d}')$ ,  $\mathcal{B}$ ,  $p$  and  $rp$  that witness that the condition  $(\star^c)$  is not satisfied.

- $(\mathbf{n}, \mathbf{xd}) \stackrel{\text{def}}{=} (\mathbf{n}_K, \mathbf{xd}_K)$ ,  $(\mathbf{n}, \mathbf{xd}') \stackrel{\text{def}}{=} (\mathbf{n}_K, \mathbf{xd}'_K)$  and  $\mathcal{B} \stackrel{\text{def}}{=} (j_{K+1} \cdots j_{K'})^\omega$ .
- for all  $i \in \mathbb{N}$ ,  $m \in [0, K' - K - 1]$ ,
  - $p(i(K' - K) + m) \stackrel{\text{def}}{=} (\mathbf{n}_K(j_{K+1} \cdots j_{K'})^i(j_{K+1} \cdots j_{K+m}), \mathbf{xd}_{K+m})$  and,
  - $rp(i(K' - K) + m) \stackrel{\text{def}}{=} (\mathbf{n}_K(j_{K+1} \cdots j_{K'})^i(j_{K+1} \cdots j_{K+m}), \mathbf{xd}'_{K+m})$ .

One can check that all the properties hold to violate  $(\star^c)$ , in particular,  $p$  is strict and for all  $i$ ,  $p(i) \leq rp(i)$ .  $\square$

## 8 Concluding Remarks

In this document, we developed an automata-based approach to solve the satisfiability problem for the logics  $\text{CTL}(\mathbb{Z})$  and  $\text{CTL}^*(\mathbb{Z})$  respectively, by introducing tree constraint automata that accept infinite data trees with data domain  $\mathbb{Z}$ . In general, our contributions stem from the cross-fertilisation of automata-based techniques for temporal logics and reasoning about (infinite) structures made of symbolic  $\mathbb{Z}$ -constraints, while trying to do our best to be self-contained. Our main results are listed below (all the results hold also for the concrete domains  $\mathbb{N}$  and  $\mathbb{Q}$ ).

- The nonemptiness problem for tree constraint automata with Büchi acceptance conditions (resp. with Rabin pairs) is EXPTIME-complete, see Theorem 2 (resp. Theorem 4). The difficult part consists in proving the EXPTIME-easiness for which we show how to substantially adapt the material in [Lab21, Section 5.2] that guided us to design the correctness proof of  $(\star^c)$ . Furthermore, we proposed a compact presentation of the main arguments (see Section 7) and we established other fine-tuned results about automata with Rabin acceptance conditions (including word and tree automata), see e.g. Lemma 8.
- The satisfiability problem for  $\text{CTL}(\mathbb{Z})$  is EXPTIME-complete (Theorem 5), whereas the satisfiability problem for  $\text{CTL}^*(\mathbb{Z})$  is 2EXPTIME-complete (Theorem 8). The only decidability proof for  $\text{SAT}(\text{CTL}^*(\mathbb{Z}))$  done so far, see [CKL16, Theorem 32], is by reduction to a decidable second-order logic with the unbounding quantifier  $U$  [Boj04, BC06], for which no complexity upper bound was known. Our complexity characterisation for  $\text{SAT}(\text{CTL}^*(\mathbb{Z}))$  provides an answer to several open problems related to  $\text{CTL}^*(\mathbb{Z})$  fragments, see e.g. [BG06, Gas09, CKL16, CT16, LOS20]. Note also that if we enrich the concrete domain with predicates  $\mathbf{x} \sim \mathfrak{d}$  ( $\sim \in \{<, >\}$ ), the results still hold thanks to our use of the sets  $\text{SatTypes}(\beta)$ 's that already encode such constraints.
- As tree constraint automata become a pivot formalism that can be reused for logics other than  $\text{CTL}(\mathbb{Z})$  and  $\text{CTL}^*(\mathbb{Z})$ , as a by-product, we get that  $\text{TSAT}(\mathcal{ALCF}^P(\mathbb{Z}_c))$  for the description logic  $\mathcal{ALCF}^P(\mathbb{Z}_c)$  is in EXPTIME, a result established for the first time in [LOS20]. We believe that our results on TCA can help to establish decidability/complexity results for other logics (see also Corollary 2 about a domain for strings and [EFK22, Section 4] to handle more concrete domains).

## References

- ADG20. E.G. Amparore, S. Donatelli, and F. Gallà. A  $\text{CTL}^*$  model checker for Petri nets. In *PETRI'20*, volume 12152 of *LNCS*, pages 403–413. Springer, 2020.
- Baa09. F. Baader. Description Logics. In *Reasoning Web. Semantic Technologies for Information Systems, 5th International Summer School 2009, Tutorial Lectures*, volume 5689 of *LNCS*, pages 1–39. Springer, 2009.



- 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.
- BC06. M. Bojańczyk and Th. Colcombet. Bounds in w-Regularity. In *LICS'06*, pages 285–296. IEEE Computer Society, 2006.
- BF22. B. Bednarczyk and O. Fiuk. Presburger Büchi tree automata with applications to logics with expressive counting. In *WoLLIC'22*, volume 13468 of *LNCS*, pages 295–308. Springer, 2022.
- BG06. L. Bozzelli and R. Gascon. Branching-time temporal logic extended with Presburger constraints. In *LPAR'06*, volume 4246 of *LNCS*, pages 197–211. Springer, 2006.
- BH91. F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *IJCAI'91*, pages 452–457, 1991.
- Boj04. M. Bojańczyk. A bounding quantifier. In *CSL'04*, volume 3210 of *LNCS*, pages 41–55. Springer, 2004.
- Bok18. U. Boker. Why these automata types? In *LPAR'18*, volume 57 of *EPiC Series in Computing*, pages 143–163. EasyChair, 2018.
- BP22. A. Bhaskar and M. Praveen. Realizability problem for constraint LTL. In *TIME'22*, volume 247 of *LIPICs*, pages 8:1–8:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- BR22. F. Baader and J. Rydval. Using model theory to find decidable and tractable description logics with concrete domains. *Journal of Automated Reasoning*, 66(3):357–407, 2022.
- BT12. M. Bojańczyk and S. Toruńczyk. Weak MSO+U over infinite trees. In *STACS'12*, *LIPICs*, pages 648–660, 2012.
- Büc62. J.R. Büchi. On a decision method in restricted second-order arithmetic. In *International Congress on Logic, Method and Philosophical Science'60*, pages 1–11, 1962.
- Car15. C. Carapelle. *On the satisfiability of temporal logics with concrete domains*. PhD thesis, Leipzig University, 2015.
- Čer94. K. Čerāns. Deciding properties of integral relational automata. In *ICALP'94*, volume 820 of *LNCS*, pages 35–46. Springer, 1994.
- CKL16. C. Carapelle, A. Kartzow, and M. Lohrey. Satisfiability of ECTL\* with constraints. *Journal of Computer and System Sciences*, 82(5):826–855, 2016.
- CKP15. B. Cook, H. Khlaaf, and N. Piterman. On automation of CTL\* verification for infinite-state systems. In *CAV'15*, volume 9206 of *LNCS*, pages 13–29. Springer, 2015.
- CKS81. A. Chandra, D. Kozen, and L. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, 1981.
- CT16. C. Carapelle and A.-Y. Turhan. Description Logics Reasoning w.r.t. General TBoxes is Decidable for Concrete Domains with the EHD-property. In *ECAI'16*, volume 285, pages 1440–1448. IOS Press, 2016.
- DD07. S. Demri and D. D'Souza. An automata-theoretic approach to constraint LTL. *Information and Computation*, 205(3):380–415, 2007.
- DD16. S. Demri and M. Deters. Temporal logics on strings with prefix relation. *Journal of Logic and Computation*, 26:989–1017, 2016.
- DDS18. S. Demri, A.K. Dhar, and A. Sangnier. Equivalence between model-checking flat counter systems and Presburger arithmetic. *TCS*, 735:2–23, 2018.
- DG08. S. Demri and R. Gascon. Verification of qualitative  $\mathbb{Z}$  constraints. *TCS*, 409(1):24–40, 2008.
- DHV14. A. Deutsch, R. Hull, and V. Vianu. Automatic verification of database-centric system. *SIGMOD Record*, 43(3):5–17, 2014.
- DL09. S. Demri and R. Lazić. LTL with the freeze quantifier and register automata. *ACM ToCL*, 10(3), 2009.
- DQ21. S. Demri and K. Quaas. Concrete domains in logics: a survey. *ACM SIGLOG News*, 8(3):6–29, 2021.
- EFK21. L. Exibard, E. Filiot, and A. Khalimov. Church synthesis on register automata over linearly ordered data domains. In *STACS'21*, volume 187 of *LIPICs*, pages 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

- EFK22. L. Exibard, E. Filiot, and A. Khalimov. A generic solution to register-bounded synthesis with an application to discrete orders. In *ICALP'22*, volume 229 of *LIPICs*, pages 122:1–122:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- EH86. E.A. Emerson and J. Halpern. ‘sometimes’ and ‘not never’ revisited: on branching versus linear time temporal logic. *Journal of the Association for Computing Machinery*, 33:151–178, 1986.
- EJ00. E.A. Emerson and C.S. Jutla. The complexity of tree automata and logics of programs. *SIAM Journal of Computing*, 29(1):132–158, 2000.
- ES84. E.A. Emerson and P. Sistla. Deciding full branching time logic. *Information and Control*, 61:175–201, 1984.
- FHP22. B. Finkbeiner, Ph. Heim, and N. Passing. Temporal stream logic modulo theories. In *FoSSaCS'22*, volume 13242 of *LNCS*, pages 325–346. Springer, 2022.
- Fig10. D. Figueira. *Reasoning on words and trees with data*. PhD thesis, ENS Cachan, December 2010.
- Fig12. D. Figueira. Decidability of Downward XPath. *ACM ToCL*, 13(4):1–40, 2012.
- FMW22a. P. Felli, M. Montali, and S. Winkler. CTL\* model checking for data-aware dynamic systems with arithmetic. In *IJCAR'22*, volume 13385 of *LNCS*, pages 36–56. Springer, 2022.
- FMW22b. P. Felli, M. Montali, and S. Winkler. Linear-time verification of data-aware dynamic systems with arithmetic. In *AAAI'22*, pages 5642–5650. AAAI Press, 2022.
- Gas09. R. Gascon. An automata-based approach for CTL\* with constraints. *Electronic Notes in Theoretical Computer Science*, 239:193–211, 2009.
- GGG22. L. Geatti, A. Gianola, and N. Gigante. Linear temporal logic modulo theories over finite traces. In *IJCAI'22*, pages 2641–2647. ijcai.org, 2022.
- GH82. Y. Gurevich and L. Harrington. Trees, automata and games. In *STOC'82*, pages 60–65. ACM, 1982.
- KVW00. O. Kupferman, M. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the Association for Computing Machinery*, 47(2):312–360, 2000.
- KW15. A. Kartzow and Th. Weidner. Model checking constraint LTL over trees. *CoRR*, abs/1504.06105, 2015.
- Lab21. N. Labai. *Automata-based reasoning for decidable logics with data values*. PhD thesis, TU Wien, May 2021.
- LM07. C. Lutz and M. Milčić. A Tableau Algorithm for Description Logics with Concrete Domains and General Tboxes. *Journal of Automated Reasoning*, 38(1-3):227–259, 2007.
- LOS20. N. Labai, M. Ortiz, and M. Simkus. An Exptime Upper Bound for  $\mathcal{ALC}$  with integers. In *KR'20*, pages 425–436. Morgan Kaufman, 2020.
- Lut01. C. Lutz. Interval-based temporal reasoning with general TBoxes. In *IJCAI'01*, pages 89–94. Morgan-Kaufmann, 2001.
- Lut02. C. Lutz. *The Complexity of Description Logics with Concrete Domains*. PhD thesis, RWTH, Aachen, 2002.
- Lut03. C. Lutz. Description logics with concrete domains—a survey. In *Advances in Modal Logics Volume 4*, pages 265–296. King’s College Publications, 2003.
- Lut04a. C. Lutz. Combining interval-based temporal reasoning with general TBoxes. *Artificial Intelligence*, 152(2):235–274, 2004.
- Lut04b. C. Lutz. NEXPTIME-complete description logics with concrete domains. *ACM ToCL*, 5(4):669–705, 2004.
- LW08. S. Lasota and I. Walukiewicz. Alternating timed automata. *ACM ToCL*, 9(2):10:1–10:27, 2008.
- MS95. D. Muller and E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. *TCS*, 141(1–2):69–107, 1995.
- NSV04. F. Neven, T. Schwentick, and V. Vianu. Finite state machines for strings over infinite alphabets. *ACM ToCL*, 5(3):403–435, 2004.
- PQ22. D. Peteler and K. Quaas. Deciding Emptiness for Constraint Automata on Strings with the Prefix and Suffix Order. In *MFCS'22*, volume 241 of *LIPICs*, pages 76:1–76:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

- Rab69. M.O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the AMS*, 141:1–35, 1969.
- Rev02. P. Revesz. *Introduction to Constraint Databases*. Springer, New York, 2002.
- Saf89. S. Safra. *Complexity of Automata on Infinite Objects*. PhD thesis, The Weizmann Institute of Science, Rehovot, 1989.
- Sch91. K. Schild. A correspondence theory for terminological logics: preliminary report. In *IJCAI-12*, pages 466–471, 1991.
- Sco62. D. Scott. A decision method for validity of sentences in two variables. *The Journal of Symbolic Logic*, 27:377, 1962.
- SSM08. H. Seidl, Th. Schwentick, and A. Muscholl. Counting in trees. In *Logic and Automata: History and Perspectives*, volume 2 of *Texts in Logic and Games*, pages 575–612. Amsterdam University Press, 2008.
- ST11. L. Segoufin and S. Toruńczyk. Automata based verification over linearly ordered data domains. In *STACS’11*, pages 81–92, 2011.
- Tho90. W. Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science, Volume B, Formal models and semantics*, pages 133–191. Elsevier, 1990.
- VS85. M. Vardi and L. Stockmeyer. Improved upper and lower bounds for modal logics of programs. In *STOC’85*, pages 240–251. ACM, 1985.
- VW86. M. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences*, 32:183–221, 1986.
- VW94. M. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115:1–37, 1994.
- VW08. M. Vardi and Th. Wilke. Automata: from logics to algorithms. In *Logic and Automata: History and Perspectives*, number 2 in *Texts in Logic and Games*, pages 629–736. Amsterdam University Press, 2008.
- Wol87. P. Wolper. On the relation of programs and computations to models of temporal logic. In *Temporal Logic in Specifications*, volume 398 of *LNCS*, pages 75–123. Springer, 1987.