Logical Methods in Computer Science
Volume 21, Issue 2, 2025, pp. 16:1–16:79
https://lmcs.episciences.org/

Submitted Feb. 02, 2024
Published May 30, 2025

CONSTRAINT AUTOMATA ON INFINITE DATA TREES: FROM CTL($\mathbb{Z}$)/CTL*($\mathbb{Z}$) TO DECISION PROCEDURES

STÉPHANE DEMRI [a] AND KARIN QUAAS [b]

[a] Université Paris-Saclay, ENS Paris-Saclay, CNRS, Laboratoire Méthodes Formelles, 91190, Gif-sur-Yvette, France

[b] Universität Leipzig, Fakultät für Mathematik und Informatik, Germany
e-mail address: quaas@informatik.uni-leipzig.de

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 its nonemptiness problem is in ExpTime. Using an automata-based approach, we establish that the satisfiability problem for CTL($\mathbb{Z}$) (CTL with constraints in $\mathbb{Z}$) is ExpTime-complete, and the satisfiability problem for CTL*($\mathbb{Z}$) is 2ExpTime-complete (only decidability was known so far). By-product results with other concrete domains and other logics are also briefly discussed.

1. Introduction

In this paper, we study the satisfiability problem for the branching-time temporal logics CTL($\mathbb{Z}$) and CTL*($\mathbb{Z}$), extending the classical temporal logics CTL and CTL* in that atomic formulae express constraints about the relational structure $(\mathbb{Z}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{Z}})$. Formulae in these logics are interpreted over Kripke structures that are annotated with values in $\mathbb{Z}$, see for instance the tree $\mathbb{t}$ in Figure 3 (page 11). A typical CTL*($\mathbb{Z}$) formula is the expression $\mathsf{AGF}(\mathtt{x} < \mathsf{X}\mathtt{x})$ stating that on all paths infinitely often the value of the variable $\mathtt{x}$ at the current position is strictly smaller than the value of $\mathtt{x}$ at the next position. Formalisms defined over relational structures, also known as *concrete domains*, are considered in many works, including works on temporal logics [GM98, Car15, MT16, LMO+18, FMP20, CDOT21, FMW22b, FP24], description logics [Lut02, Lut03, Lut04b, CT16, Lab21, BR22, ABB+23, BBK24], and automata [Gas09, ST11, KW15, Wei16, TZ22, PQ22]. 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 data-aware systems, see e.g. [DHV14, FMW22a]. A brief survey can be found in [DQ21].

Decidability results for concrete domains handled in [LM07, Gas09, BR22, BB24] exclude the ubiquitous concrete domain $(\mathbb{Z}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{Z}})$. By contrast, decidability results for logics with concrete domain $\mathbb{Z}$ require dedicated proof techniques, see e.g. [BG06, DD07,

Key words and phrases: Constraints, Constraint Automata, Temporal Logics, Infinite Data Trees .
Karin Quaas is supported by the Deutsche Forschungsgemeinschaft (DFG), project 504343613.

DOI:10.46298/LMCS-21(2:16)2025

ST11, LOS20, BP24]. In particular, *fragments* of CTL*($\mathbb{Z}$) are shown decidable in [BG06] using integral relational automata from [Čer94], and the satisfiability problem for the existential and the universal fragment of CTL* with gap-order constraints (more general than the ones in this paper) can be solved in PSpace [BP14, Theorem 14]. Another important breakthrough came with the decidability of CTL*($\mathbb{Z}$) [CKL16, Theorem 32] (see also [CKL13]) by designing a reduction to a decidable second-order logic, whose formulae are made of Boolean combinations of formulae from MSO and from Weak MSO+U [BT12], where U is the unbounding second-order quantifier, see e.g. [Boj04, BC06] (in Weak MSO, second-order quantification is over finite sets). 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 inferred. More recently, the condition $C_{\mathbb{Z}}$ [DD07] to approximate the set of satisfiable symbolic models of a given LTL($\mathbb{Z}$) formula (in a problematic way, not necessarily an ω-regular language) is extended to the branching case in [LOS20] leading to the ExpTime-membership of the concept satisfiability problem w.r.t. general TBoxes for the description logic $\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z})$. However, no elementary complexity upper bounds for the satisfiability problem for CTL($\mathbb{Z}$) nor CTL*($\mathbb{Z}$) were known since their decidability was established in [CKL13, Car15, CKL16].

In this paper, we prove that the satisfiability problem for CTL($\mathbb{Z}$) is ExpTime-complete, and the satisfiability problem for CTL*($\mathbb{Z}$) is 2ExpTime-complete. We pursue the *automata-based approach* for solving decision problems for temporal logics, following seminal works 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, see e.g. [ST11, KW15, PQ22] and also in [Fig12] related to automata accepting finite data trees. 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 difficulty rests on the design of decision procedures for checking nonemptiness of constraint automata (but 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, Gas09, LOS20]. In this case, the main effort is focused on the design of the translation (based on abstractions for tuples of data values) since the decision procedures for the target automata are usually already well-studied.

It is well-known that decision procedures for CTL* are difficult to design, and the combination with the concrete domain $\mathbb{Z}$ is definitely challenging. Moreover, we aim at proposing a general framework: we investigate a new class of *tree constraint automata*, understood as a target formalism in the pure tradition of the automata-based approach, and easy to reuse. The structures accepted by such tree constraint automata are *infinite* trees in which nodes are labelled by a letter from a finite alphabet and a tuple in $\mathbb{Z}^{\beta}$ for some $\beta \geqslant 1$ (this excludes 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, IX19], and therefore we advocate the introduction of *nondeterministic* tree constraint automata without alternation. Our definition of tree constraint automata naturally extends the definition of constraint automata for words (see e.g. [Čer94, Rev02, ST11, KW15, PQ22]) and as far as we know, the extension to infinite trees in the way done herein has not been considered earlier in the literature. Note 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).

As a key result, we show that the nonemptiness problem for tree constraint automata over $(\mathbb{Z}, <, =, (=_{\mathfrak{d}})_{\mathfrak{d} \in \mathbb{Z}})$ is ExpTime-complete. In order to obtain the 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 (see Lemma 3.1 and Lemma 4.14). As a corollary, we establish that the satisfiability problem for CTL($\mathbb{Z}$) is ExpTime-complete (Theorem 5.6), which is one of the main results of the paper. As a by-product, it also allows us to conclude that the concept satisfiability problem w.r.t. general TBoxes for the description logic $\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z})$ is in ExpTime, a result known since [LOS20]. The details can be found in [DQ23b, Section 5.2]. By lack of space, we do not provide the details herein.

Our main contribution is the characterisation of the complexity for CTL*($\mathbb{Z}$) satisfiability, which is an open problem evoked in [CKL16, Section 9] and [LOS20, Section 5] (decidability was established ten years ago in [CKL13]). 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. In Section 6, we show that the satisfiability problem for CTL*($\mathbb{Z}$) is in 2ExpTime by using Rabin tree constraint automata (introduced herein). We have to check that the essential steps for CTL* can be lifted to CTL*($\mathbb{Z}$) to get the optimal upper bound while guaranteeing that computationally we are in a position to provide an optimal upper bound. In Section 6.1, we establish a special form for CTL*($\mathbb{Z}$) formulae from which tree constraint 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 7 following developments from [Saf89, Chapter 1] but carefully adapted to the context of constraint automata. Observe that we can get away with nondeterminism, but using alternation with registers/variables would more problematic because, in some way, this would require to know how to handle an unbounded number of data values due to alternation.
This paper is a revised and completed version of the conference paper [DQ23a]. *It can be seen also as a revised and more compact version of the arXiv report* [DQ23b]. *In order to limit the length of the body of this paper, several proofs are placed in the technical appendix (see Appendices A–F).*

2. Temporal Logics with Numerical Domains

2.1. **Constraints and Kripke Structures.** Let $V = \{\mathtt{x}, \mathtt{y}, \ldots\}$ be a countably infinite set of variables. A *term* $\mathtt{t}$ *over* V is an expression of the form $\mathsf{X}^i\mathtt{x}$, where $\mathtt{x} \in V$ and X^i is a (possibly empty) sequence of i symbols 'X'. A term $\mathsf{X}^i\mathtt{x}$ should be understood as a variable (that needs to be interpreted) but, later on, we will see that the prefix X^i will have a temporal interpretation. We write T_V to denote the set of all terms over V. For all $i \in \mathbb{N}$, we write $\mathrm{T}_V^{\leqslant i}$ to denote the subset of terms of the form $\mathsf{X}^j\mathtt{x}$, where $j \leqslant i$. For instance, $\mathrm{T}_V^{\leqslant 0} = V$ and $\mathrm{T}_V^{\leqslant 1} = V \cup \{\mathsf{X}\mathtt{x} \mid \mathtt{x} \in V\}$. A *valuation* $\mathfrak{v} : \mathrm{T}_V \to \mathbb{Z}$ is a function that maps terms in T_V to elements in $\mathbb{Z}$. To be precise, and a bit more general, we assume that the set of valuations is polymorphic: as a common feature it admits as argument a term and returns an integer. Quite often, valuations $\mathfrak{v}$ are of the form $\{\mathtt{x}_1, \ldots, \mathtt{x}_\beta\} \to \mathbb{Z}$ when we are only interested in the values for the variables in $\{\mathtt{x}_1, \ldots, \mathtt{x}_\beta\}$. Possibly, additional arguments are considered to provide some context to the interpretation, such as a world or a path in a Kripke structure; details will follow but should not lead to any confusion.

We consider the concrete domain $(\mathbb{Z}, <, =, (=_{\mathbb{d}})_{\mathbb{d} \in \mathbb{Z}})$ (also written $\mathbb{Z}$), where $=_{\mathbb{d}}$ is a unary predicate stating the equality with the constant $\mathbb{d}$, $<$ is the usual order on $\mathbb{Z}$, and $=$ denotes equality. An *atomic constraint* θ *over* T_V is an expression of one of the forms below:

$$\mathtt{t} < \mathtt{t}' \qquad \mathtt{t} = \mathtt{t}' \qquad =_{\mathbb{d}} (\mathtt{t}) \text{ (also written } \mathtt{t} = \mathbb{d}),$$

where $\mathbb{d} \in \mathbb{Z}$ and $\mathtt{t}, \mathtt{t}' \in \mathrm{T}_V$. A *constraint* Θ is defined as a Boolean combination of atomic constraints. Constraints are interpreted on valuations $\mathfrak{v} : \mathrm{T}_V \to \mathbb{Z}$: a valuation $\mathfrak{v}$ *satisfies* θ, written $\mathfrak{v} \models \theta$ iff the interpretation of the terms in θ makes θ true in $\mathbb{Z}$ in the usual way. The Boolean connectives are interpreted as usual. A constraint Θ is *satisfiable* iff there is a valuation $\mathfrak{v} : \mathrm{T}_V \to \mathbb{Z}$ such that $\mathfrak{v} \models \Theta$. Similarly, a constraint Θ_1 *entails* a constraint Θ_2 (written $\Theta_1 \models \Theta_2$) iff for all valuations $\mathfrak{v}$, we have $\mathfrak{v} \models \Theta_1$ implies $\mathfrak{v} \models \Theta_2$. The satisfiability problem restricted to finite conjunctions of atomic constraints can be solved in PTime (see e.g. [Čer94, Lemma 5.5]) and entailment is in coNP. The PTime upper bound can be refined to a cubic bound by performing a linear amount of calls to Bellman-Ford algorithm [CLRS22] that computes shortest paths in weighted directed graphs and detects negative cycles, see also [DMP91, CLO+16].

Kripke structures. In order to define logics with the concrete domain $\mathbb{Z}$, the semantical structures for such logics 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 $(\mathcal{W}, \mathcal{R}, \mathfrak{v})$, where $\mathcal{W}$ is a non-empty set of *worlds*, $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$ is the accessibility relation and $\mathfrak{v} : \mathcal{W} \times V \to \mathbb{Z}$ is a valuation. A Kripke structure $\mathcal{K}$ is *total* if for all $w \in \mathcal{W}$ there is $w' \in \mathcal{W}$ such that $(w, w') \in \mathcal{R}$. Given a Kripke structure $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathfrak{v})$ and a world $w \in \mathcal{W}$, an *infinite path* π from w is an ω-sequence $w_0, w_1 \ldots w_n, \ldots$ 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.

Labelled trees. Along the paper, the expression $[n, m]$ with $n, m \in \mathbb{Z}$ denotes the set $\{k \in \mathbb{Z} \mid n \leqslant k \leqslant m\}$. Given $D \geqslant 1$, a *labelled tree* of degree D is a map $\mathbb{t} : dom(\mathbb{t}) \to \Sigma$, where Σ is some (potentially infinite) alphabet and $dom(\mathbb{t})$ is an infinite subset of $[0, D-1]^*$ such that $\mathbf{n} \in dom(\mathbb{t})$, and $\mathbf{n} \cdot i \in dom(\mathbb{t})$ for all $0 \leqslant i < j$ whenever $\mathbf{n} \cdot j \in dom(\mathbb{t})$ for some $\mathbf{n} \in [0, D-1]^*$ and $j \in [0, D-1]$. The elements of $dom(\mathbb{t})$ are called *nodes*. The empty word ε is the *root node* of $\mathbb{t}$. For every $\mathbf{n} \in dom(\mathbb{t})$, the elements $\mathbf{n} \cdot i \in dom(\mathbb{t})$ with $i \in [0, D-1]$ are called the *children nodes of* $\mathbf{n}$, and $\mathbf{n}$ is called the *parent node of* $\mathbf{n} \cdot i$. Nodes with the same parent node are called *sibling nodes*; sibling nodes are implicitly ordered, though this feature is seldom used in the document. We say that the tree $\mathbb{t}$ is a *full D-ary tree* if every node $\mathbf{n}$ has exactly D children $\mathbf{n} \cdot 0, \ldots, \mathbf{n} \cdot (D-1)$ (equivalently, $dom(\mathbb{t}) = [0, D-1]^*$). Given a tree $\mathbb{t}$ and a node $\mathbf{n}$ in $dom(\mathbb{t})$, an infinite *path* in $\mathbb{t}$ starting from $\mathbf{n}$ is an infinite sequence $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \ldots$, where $j_i \in [0, D-1]$ and $\mathbf{n} \cdot j_1 \ldots j_i \in dom(\mathbb{t})$ for all $i \geqslant 1$.

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

2.2. **The Logic CTL*($\mathbb{Z}$).** We introduce the logic CTL*($\mathbb{Z}$), which extends the branching-time temporal logic CTL* from [EH86] with constraints over $\mathbb{Z}$. *State formulae* ϕ and *path formulae* Φ of CTL*($\mathbb{Z}$) are defined by

$$\phi := \neg\phi \mid \phi \wedge \phi \mid \mathsf{E}\Phi \qquad \Phi := \phi \mid \mathtt{t} = \mathbb{d} \mid \mathtt{t}_1 = \mathtt{t}_2 \mid \mathtt{t}_1 < \mathtt{t}_2 \mid \neg\Phi \mid \Phi \wedge \Phi \mid \mathsf{X}\Phi \mid \Phi\mathsf{U}\Phi,$$

where $\mathtt{t}, \mathtt{t}_1, \mathtt{t}_2 \in \mathrm{T}_V$. State formulae respectively path formulae are interpreted on worlds, respectively on infinite paths of a Kripke structure. Let $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathfrak{v})$ is a total Kripke structure, and $w \in \mathcal{W}$. We define the satisfaction relation (omitting the clauses for Boolean connectives) for state formulae by

- $\mathcal{K}, w \models \mathsf{E}\Phi \overset{\text{def}}{\Leftrightarrow}$ there is an infinite path π from w such that $\mathcal{K}, \pi \models \Phi$.

Let $\pi = w_0, w_1, \ldots$ be an infinite path of $\mathcal{K}$. Let us define $\mathfrak{v}(\pi, \mathsf{X}^j\mathtt{x}) \overset{\text{def}}{=} \mathfrak{v}(w_j, \mathtt{x})$, for all terms of the form $\mathsf{X}^j\mathtt{x}$. Hence, the term $\mathsf{X}^j\mathtt{x}$ refers to the value of the variable $\mathtt{x}$ exactly j steps ahead along a path. For every n, $\pi[n, +\infty)$ is the suffix of π truncated by the n first worlds. We define the satisfaction relation for path formula by

- $\mathcal{K}, \pi \models \mathtt{t} = \mathbb{d} \overset{\text{def}}{\Leftrightarrow} \mathfrak{v}(\pi, \mathtt{t}) = \mathbb{d}$;
- $\mathcal{K}, \pi \models \mathtt{t}_1 \sim \mathtt{t}_2 \overset{\text{def}}{\Leftrightarrow} \mathfrak{v}(\pi, \mathtt{t}_1) \sim \mathfrak{v}(\pi, \mathtt{t}_2)$ for all $\sim \in \{<, =\}$;
- $\mathcal{K}, \pi \models \Phi\mathsf{U}\Psi \overset{\text{def}}{\Leftrightarrow}$ there is $j \geqslant 0$ such that $\mathcal{K}, \pi[j, +\infty) \models \Psi$ and for all $j' \in [0, j-1]$, we have $\mathcal{K}, \pi[j', +\infty) \models \Phi$;
- $\mathcal{K}, \pi \models \mathsf{X}\Phi \overset{\text{def}}{\Leftrightarrow} \mathcal{K}, \pi[1, +\infty) \models \Phi$.

The *size of a formula* is understood as its number of symbols with integers encoded in binary. As usual, we also use disjunction $\vee$, the universal path quantifier A, defined by $\mathsf{A}\Phi \overset{\text{def}}{=} \neg\mathsf{E}\neg\Phi$, and the standard temporal connectives R and G, defined by $\Phi_1\mathsf{R}\Phi_2 \overset{\text{def}}{=} \neg(\neg\Phi_1\mathsf{U}\neg\Phi_2)$ and $\mathsf{G}\Phi \overset{\text{def}}{=} \mathsf{E}(\mathtt{x} < \mathtt{x})\,\mathsf{R}\,\Phi$. Propositional variables p can easily be encoded with an atomic formula $\mathsf{E}(\mathtt{x}_p = 0)$. A formula in CTL*($\mathbb{Z}$) is in *simple form* if it is in negation normal form (using $\vee$, A, and R as primitives) and all terms occurring in the formula are from $\mathrm{T}_V^{\leqslant 1}$. We define two fragments of CTL*($\mathbb{Z}$): formulae in the logic CTL($\mathbb{Z}$) are state formulae of the form

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

where Θ is a constraint. By way of example, $\mathsf{AX}(\mathsf{E}(\mathsf{X}\mathtt{x} = 3) \vee \mathsf{A}(\neg(\mathsf{X}\mathtt{x} = 3)))$ is a CTL($\mathbb{Z}$) formula. Formulae in the logic LTL($\mathbb{Z}$) are defined from path formulae for CTL*($\mathbb{Z}$) according to $\Phi := \Theta \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \mathsf{X}\Phi \mid \Phi\mathsf{U}\Phi \mid \Phi\mathsf{R}\Phi$, where Θ is a constraint. Negation occurs only in constraints since the LTL logical connectives have their dual in LTL($\mathbb{Z}$). In contrast to CTL*($\mathbb{Z}$) and CTL($\mathbb{Z}$), LTL($\mathbb{Z}$) formulae are evaluated over infinite paths of valuations $\mathfrak{v} : V \to \mathbb{Z}$ (no branching involved).

The *satisfiability problem for* CTL*($\mathbb{Z}$), written SAT(CTL*($\mathbb{Z}$)), is defined as follows.

Input: A CTL*($\mathbb{Z}$) state formula ϕ.
Question: Is there a total Kripke structure $\mathcal{K}$ and a world w such that $\mathcal{K}, w \models \phi$?

The satisfiability problem SAT(CTL($\mathbb{Z}$)) for CTL($\mathbb{Z}$) is defined analogously; for LTL($\mathbb{Z}$), SAT(LTL($\mathbb{Z}$)) is the problem to decide whether there exists an infinite sequence of valuations $\mathfrak{v} : V \to \mathbb{Z}$ such that $\mathfrak{v} \models \Phi$ for a given LTL($\mathbb{Z}$) formula Φ.

Decidability, and more precisely, PSPACE-completeness of SAT(LTL($\mathbb{Z}$)) is shown in [DG08]. For some strict fragments of CTL*($\mathbb{Z}$), decidability is shown in [BG06, BP14].

It is only recently in [CKL13, Car15, CKL16], that decidability is established for the full logic using a translation into a decidable second-order logic:

Proposition 2.1 [CKL13, Car15, CKL16]**.** SAT(CTL*($\mathbb{Z}$)) *is decidable.*

The proof in [CKL13, Car15, CKL16] does not provide a complexity upper bound as the target second-order logic admits an automata-based decision procedure with open complexity [BT12, Boj04, BC06]. Moreover, the target logic uses the standard weak monadic theory of one successor (WS1S) with a non-elementary complexity, which disqualifies this approach to obtain a direct optimal complexity upper bound.

Let us shortly explain why the satisfiability problem is challenging. First of all, observe that CTL*($\mathbb{Z}$) has atomic formulae in which integer values at the current and successor states are compared. This prevents us from using a simple translation from CTL*($\mathbb{Z}$) to CTL* with new propositions. Models of CTL*($\mathbb{Z}$) formulae can be viewed as an infinite network of constraints on $\mathbb{Z}$; even if a formula contains only a finite set of constants, a model may contain an infinite set of values, as it is the case for, e.g., the formula $\mathsf{EG}(\mathtt{x} < \mathsf{X}\mathtt{x})$. Hence a direct Boolean abstraction does not work; by contrast, when variables cannot be compared at different positions, Boolean abstraction can be an option, see recent developments in [RS23, RS24]. On the other hand, CTL*($\mathbb{Z}$) has no freeze quantifier and no data variable quantification, and hence no way to directly compare values at unbounded distance (this can only be done by propagating local constraints), unlike e.g. the formalisms in [DHLT14, SW16, BL17, AFF17]. Hence, the lower bounds from [JL11] cannot apply either. Related work about the model-checking problem can be found in Section 9.1.

In this paper, we prove the precise worst-case computational complexity of the problems SAT(CTL*($\mathbb{Z}$)) and SAT(CTL($\mathbb{Z}$)), respectively. We follow the automata-based approach, that is, we translate formulae in our logics into equivalent automata – Büchi tree constraint automata for CTL($\mathbb{Z}$), and Rabin tree constraint automata for CTL*($\mathbb{Z}$) – so that we can reduce the satisfiability problem for the logics to the nonemptiness problem for the corresponding automata.

3. Tree Constraint Automata

In this section, we introduce the class of tree constraint automata that accept sets of infinite trees of the form $\mathbb{t} : [0, D-1]^* \to (\Sigma \times \mathbb{Z}^\beta)$ for some finite alphabet Σ and some $\beta \geqslant 1$. The transition relation of such automata puts constraints between the β integer values at a node and the integer values at its children nodes. The automaton is equipped with an acceptance condition (Büchi, Rabin, Streett) that will be defined later on. The forthcoming definition is specific to the concrete domain $\mathbb{Z}$, but it can be easily adapted to other concrete domains. Formally, a *tree constraint automaton* (TCA, for short) is a tuple $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\rm in}, \delta, F)$, where

- Q is a finite set of locations; Σ is a finite alphabet,
- $D \geqslant 1$ is the branching degree of the trees processed by $\mathbb{A}$, or the degree of $\mathbb{A}$ for short,
- $\beta \geqslant 1$ is the number of variables,
- $Q_{\rm in} \subseteq Q$ is the set of initial locations,
- δ is a *finite* subset of $Q \times \Sigma \times (\mathrm{TreeCons}(\beta) \times Q)^D$, the transition relation. Here, TreeCons(β) denotes the Boolean combinations of atomic constraints built over the terms $\mathtt{x}_1, \ldots, \mathtt{x}_\beta, \mathtt{x}'_1, \ldots, \mathtt{x}'_\beta$, where $\mathtt{x}'_i$ stands for the term $\mathsf{X}\mathtt{x}_i$. δ consists of tuples

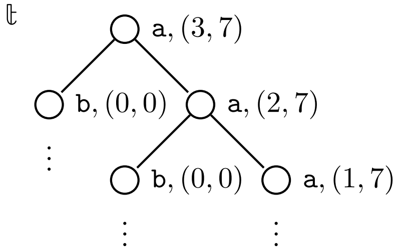
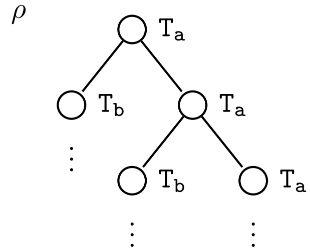


FIGURE 1. A tree $\mathbb{t}$



$\mathtt{T_a} = (q, \mathtt{a}, (\Theta'_0, q), (\Theta'_1, q))$
$\Theta'_0 \;=\; (\mathtt{x}'_1 = 0)$
$\Theta'_1 \;=\; (\mathtt{x}'_1 < \mathtt{x}_1 < \mathtt{x}_2 = \mathtt{x}'_2)$

$\mathtt{T_b} = (q, \mathtt{b}, (\Theta'_2, q), (\Theta'_2, q))$
$\Theta'_2 \;=\; (\mathtt{x}_1 = \mathtt{x}_2 = 0)$

FIGURE 2. A run ρ of some TCA on $\mathbb{t}$

$(q, \mathtt{a}, (\Theta_0, q_0), \ldots, (\Theta_{D-1}, q_{D-1}))$, where $q \in Q$ is called the *source location*, $\mathtt{a} \in \Sigma$, $q_0, \ldots, q_{D-1} \in Q$, and $\Theta_0, \ldots, \Theta_{D-1}$ are constraints in TreeCons(β).
- F encodes the acceptance condition, defined below.

Runs. Let $\mathbb{t} : [0, D-1]^* \to (\Sigma \times \mathbb{Z}^\beta)$ be an infinite full D-ary tree over $\Sigma \times \mathbb{Z}^\beta$. A *run* of $\mathbb{A}$ on $\mathbb{t}$ is a mapping $\rho : [0, D-1]^* \to \delta$ satisfying the following condition: for every $\mathbf{n} \in [0, D-1]^*$ with $\mathbb{t}(\mathbf{n}) = (\mathtt{a}, \boldsymbol{z})$ and $\mathbb{t}(\mathbf{n} \cdot i) = (\mathtt{a}_i, \boldsymbol{z}_i)$ for all $0 \leqslant i < D$, if $\rho(\mathbf{n}) = (q, \mathtt{a}, (\Theta_0, q_0), \ldots, (\Theta_{D-1}, q_{D-1}))$, then for all $0 \leqslant i < D$, we have

(i) the source location of $\rho(\mathbf{n} \cdot i)$ is q_i, and
(ii) $\mathbb{Z} \models \Theta_i(\boldsymbol{z}, \boldsymbol{z}_i)$, where $\mathbb{Z} \models \Theta_i(\boldsymbol{z}, \boldsymbol{z}_i)$ is a shortcut for $[\vec{\mathtt{x}} \leftarrow \boldsymbol{z}, \vec{\mathtt{x}'} \leftarrow \boldsymbol{z}_i] \models \Theta_i$ where $[\vec{\mathtt{x}} \leftarrow \boldsymbol{z}, \vec{\mathtt{x}'} \leftarrow \boldsymbol{z}_i]$ is a valuation $\mathfrak{v}$ on the variables $\{\mathtt{x}_j, \mathtt{x}'_j \mid j \in [1, \beta]\}$ with $\mathfrak{v}(\mathtt{x}_j) = \boldsymbol{z}(j)$ and $\mathfrak{v}(\mathtt{x}'_j) = \boldsymbol{z}_i(j)$ for all $j \in [1, \beta]$.

We show an example of a run ρ in Figure 2.

A run ρ is *initialized* if the source location of $\rho(\varepsilon)$ is in Q_{in}. Given a path $\pi = j_1 \cdot j_2 \cdot j_3 \ldots$ in ρ starting from ε, we define $\inf(\rho, \pi)$ to be the set of locations that appear infinitely often as the source locations of the transitions in $\rho(\varepsilon)\rho(j_1)\rho(j_1 \cdot j_2)\rho(j_1 \cdot j_2 \cdot j_3) \ldots$.

Acceptance Conditions. Analogously to classical ω-tree automata, TCA are equipped with an acceptance condition. We will mainly study TCA with the Büchi acceptance condition: a TCA $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$ is a *Büchi TCA* if $F \subseteq Q$ and a run ρ is *accepting* if for all paths π in ρ starting from ε, we have $\inf(\rho, \pi) \cap F \neq \varnothing$. We write $\mathrm{L}(\mathbb{A})$ to denote the set of trees $\mathbb{t}$ for which there exists some initialized and accepting run of $\mathbb{A}$ on $\mathbb{t}$. In the paper, whenever we do not further specify the acceptance condition of a TCA, we assume that the TCA is equipped with the Büchi acceptance condition.

For dealing with the logics defined in the previous sections, we will also use TCA with generalised Büchi acceptance, Rabin acceptance, and Streett acceptance conditions. In a *generalised Büchi TCA*, F is a set $\{F_1, \ldots, F_k\} \subseteq \mathcal{P}(Q)$ of states, and a run ρ is accepting if for all paths π in ρ starting from ε, for all $F_i \in F$, we have $\inf(\rho, \pi) \cap F_i \neq \varnothing$. Unsurprisingly, as for generalised Büchi automata, for every generalised Büchi TCA $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$ with $F = \{F_1, \ldots, F_k\}$ there exists a TCA $\mathbb{A}' = (Q', \Sigma, D, \beta, Q'_{\text{in}}, \delta', F')$ such that $\mathrm{L}(\mathbb{A}) = \mathrm{L}(\mathbb{A}')$ and the size of $\mathbb{A}'$ is quadratic in the size of $\mathbb{A}$:

- $Q' \stackrel{\text{def}}{=} [0, k] \times Q$, $Q'_{\text{in}} \stackrel{\text{def}}{=} \{0\} \times Q_{\text{in}}$ and $F' \stackrel{\text{def}}{=} \{0\} \times Q$.
- Given $i \in [0, k]$ and $q \in Q$, we write $\mathrm{nxt}(i, q)$ to denote the copy number in $[0, k]$ such that $\mathrm{nxt}(0, q) = 1$ for all $q \in Q$, $\mathrm{nxt}(i, q) = i$ if $q \notin F_i$ and $\mathrm{nxt}(i, q) = i + 1 \mod (k + 1)$ if $q \in F_i$.

The transition relation δ' is defined by $((i,q), \mathtt{a}, (\Theta_0, (i_0, q_0)), \ldots, (\Theta_{D-1}, (i_{D-1}, q_{D-1}))) \in \delta' \overset{\text{def}}{\Leftrightarrow}$ there is $(q, \mathtt{a}, (\Theta_0, q_0), \ldots, (\Theta_{D-1}, q_{D-1})) \in \delta$ and for all $j \in [0, D-1]$, $i_j = \mathrm{nxt}(i, q_j)$. Therefore, in the sequel, using generalised Büchi TCA instead of TCA has no consequence on worst-case complexity results.

A TCA $\mathbb{A}$ is a *Rabin TCA* if F is a set of pairs of the form (L, U), where $L, U \subseteq Q$, and a run ρ is accepting if for all paths π in ρ starting from ε, there is some pair $(L, U) \in F$ such that $\inf(\rho, \pi) \cap L \neq \emptyset$ and $\inf(\rho, \pi) \cap U = \emptyset$. Note that every TCA with set F of accepting locations can be encoded as a Rabin TCA with a single Rabin pair $(F, \emptyset)$. Hence Büchi TCA can be seen as a special case of Rabin TCA. Finally, $\mathbb{A}$ is a *Streett TCA* if F has the same form as for Rabin TCA, and a run is accepting if for all pairs $(L, U) \in F$ (also called *completemented pairs* in the literature) if $\inf(\rho, \pi) \cap L \neq \emptyset$, then $\inf(\rho, \pi) \cap U \neq \emptyset$.

Nonemptiness Problem. In this paper, we study the *nonemptiness problem for TCA*, which asks whether a given TCA $\mathbb{A}$ satisfies $\mathrm{L}(\mathbb{A}) \neq \emptyset$. For determining the computational complexity, we need to consider the size of $\mathbb{A}$, which depends on several parameters. Note that $\mathrm{TreeCons}(\beta)$ is infinite, so that, in contrast to plain Büchi tree automata [VW86], the number of transitions in a TCA is *a priori* unbounded. In particular, this means that $\mathrm{card}(\delta)$ is a priori unbounded, even if Q, Σ and D are fixed. The maximal size of a constraint occurring in transitions is unbounded, too. We write $\mathtt{MCS}(\mathbb{A})$ to denote the maximal size of a constraint occurring in $\mathbb{A}$ (with binary encoding of the integers).

Closure Properties and Expressiveness. In order to conclude this section, let us drop a few lines about the expressive power of TCA, even though this is not really in the intended scope of this paper. Obviously, the languages accepted by TCA (for any kind of acceptance condition) are closed under unions. Closure under complementations of tree languages accepted by TCA is to the best of our knowledge open. For closure under intersection, we present the following lemma, which is also key for obtaining precise complexity bounds in Section 6.

Lemma 3.1. *Let $(\mathbb{A}_k)_{1 \leqslant k \leqslant n}$ be a family of Rabin TCA s.t. $\mathbb{A}_k = (Q_k, \Sigma, D, \beta, Q_{k,in}, \delta_k, F_k)$, $\mathrm{card}(F_k) = N_k$ and $N = \prod_k N_k$. There is a Rabin TCA $\mathbb{A}$ such that $\mathrm{L}(\mathbb{A}) = \bigcap_k \mathrm{L}(\mathbb{A}_k)$ and verifying the conditions below.*

- *The number of Rabin pairs is equal to N.*
- $\mathtt{MCS}(\mathbb{A}) \leqslant n + \mathtt{MCS}(\mathbb{A}_1) + \cdots + \mathtt{MCS}(\mathbb{A}_n)$.
- *The number of locations is bounded by $\left(\prod_k \mathrm{card}(Q_k)\right) \times (2n)^N$.*
- *The number of transitions is bounded by $\prod_k \mathrm{card}(\delta_k)$.*

The proof can be found in Appendix A.1. The proof contains 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. Our contribution is related to the extension to trees (in particular to handle the acceptance conditions), to the use of constraints and to master the combinatorial explosion of the product automaton so that this is fine to get the final 2ExpTime upper bound for $\mathrm{SAT}(\mathrm{CTL}^*(\mathbb{Z}))$. By the way, the first part of the proof of the forthcoming Lemma 4.14 uses a particular case of Lemma 3.1 with $n = 2$ Rabin TCA and no use of constraints.

Regarding the expressive power of TCA, there is no TCA that accepts the set of all full D-trees over $\Sigma \times \mathbb{Z}$ such that all the data values are distinct, which is easier to capture with registers and alternation. Indeed, *ad absurdum* suppose there is a TCA $\mathbb{A}$ such that $L(\mathbb{A})$ is the set of full binary trees such that all data values are distinct, say $\beta = 1$. Let $\mathbb{t}$ be the tree such that all data values are strictly greater than $\mathbb{d}_\alpha$ (greatest constant occurring in $\mathbb{A}$, if any; otherwise zero), the root is labelled by $\mathbb{d}_\alpha + 1$ and then the data values are defined in a breadth-first fashion by incrementing by one the data value. Hence, the "left" child of the root has value $\mathbb{d}_\alpha + 2$, the "right" child of the root has value $\mathbb{d}_\alpha + 3$, etc. By definition, $\mathbb{t} \in L(\mathbb{A})$. Let $\mathbb{t}'$ be the tree obtained from $\mathbb{t}$ by replacing the right subtree of $\mathbb{t}$'s root by its left subtree. In particular, both the "left" and the "right" children of the root of $\mathbb{t}'$ have data value $\mathbb{d}_\alpha + 2$. However, note that the same constraints are satisfied between a node and its parent node in $\mathbb{t}$ and in $\mathbb{t}'$. Consequently, $\mathbb{t}' \in L(\mathbb{A})$, which leads to a contradiction.

4. Complexity of the Nonemptiness Problem for TCA

This section is dedicated to prove the ExpTime-completeness of the nonemptiness problem for Büchi TCA (Theorem 4.16) and Rabin TCA (Theorem 4.19). We make a distinction between Büchi TCA and Rabin TCA because the complexity bounds differ slightly, see Lemma 4.15 and Lemma 4.18. We mainly focus on the ExpTime-membership, which will be used for establishing upper bounds for SAT(CTL($\mathbb{Z}$)) and SAT(CTL*($\mathbb{Z}$)).

But first, let us drop a few words on the proof of ExpTime-hardness. The proof is by reduction from the acceptance problem for alternating Turing machines running in polynomial space, see e.g. [CKS81, Corollary 3.6]. The proof is presented for Büchi TCA in Appendix B.1. The key idea follows standard patterns: we use the variables of the TCA to store the content (that is a letter from a finite alphabet) of each of the polynomial cells of the alternating Turing machine. A bit more detailed, the TCA uses one variable for each tape cell. The tree structure of the runs of the alternating Turing machine can be enoded into trees that can be accepted by TCA. ExpTime-hardness for Rabin TCA follows immediately as Büchi automata are a special case of Rabin TCA.

The proof of the ExpTime-membership of the nonemptiness problem for TCA is divided into two parts. In order to determine whether $L(\mathbb{A})$ is nonempty for a given TCA $\mathbb{A}$, we first reduce the existence of some tree $\mathbb{t} \in L(\mathbb{A})$ to the existence of some *regular symbolic tree* that is *satisfiable*, that is, it admits a concrete model (Sections 4.1 and 4.2). Second, we characterise the complexity of determining the existence of such satisfiable regular symbolic trees (Section 4.3). The result for Rabin TCA is presented in Section 4.4.

From now on, we assume a fixed TCA $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$ with the constants $\mathbb{d}_1, \ldots, \mathbb{d}_\alpha$ occurring in $\mathbb{A}$ such that $\mathbb{d}_1 < \cdots < \mathbb{d}_\alpha$ (we assume there is at least one constant).

4.1. Symbolic Trees.

A *type* over the variables $\mathtt{z}_1, \ldots, \mathtt{z}_n$ is an expression of the form

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

- for all $i \in [1, n]$, Θ_i^{CST} is equal to either $\mathtt{z}_i < \mathbb{d}_1$, or $\mathtt{z}_i > \mathbb{d}_\alpha$ or $\mathtt{z}_i = \mathbb{d}$ for some $\mathbb{d} \in [\mathbb{d}_1, \mathbb{d}_\alpha]$. This definition goes a bit beyond the constraint language in $\mathbb{Z}$ (because of expressions of the form $\mathtt{z}_i < \mathbb{d}_1$ and $\mathtt{z}_i > \mathbb{d}_\alpha$), but this is harmless in the sequel. What really matters in a type is the way the variables are compared to each other and to the constants.
- $\sim_{i,j} \in \{>, =, <\}$ for all $i < j$.

Below, $\mathtt{x} = \mathtt{y}$ and $\mathtt{y} = \mathtt{x}$ are understood as identical, as well as $\mathtt{x} < \mathtt{y}$ and $\mathtt{y} > \mathtt{x}$. Checking the satisfiability of a type can be done in polynomial-time, based on a standard cycle detection, see e.g. [Čer94, Lemma 5.5]. The set of *satisfiable types* built over the terms

$\mathtt{x}_1, \ldots, \mathtt{x}_\beta, \mathtt{x}'_1, \ldots, \mathtt{x}'_\beta$ is written $\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ (n above is equal here to 2β). By way of example, we provide below a satisfiable type in $\mathrm{STypes}(2, \mathbb{d}_1, \mathbb{d}_\alpha)$ with $\alpha = 1$:

$$\mathtt{x}_1 > \mathbb{d}_1 \wedge \mathtt{x}_2 = \mathbb{d}_1 \wedge \mathtt{x}'_1 = \mathbb{d}_1 \wedge \mathtt{x}'_2 < \mathbb{d}_1 \wedge \mathtt{x}_1 > \mathtt{x}_2 \wedge \mathtt{x}_1 > \mathtt{x}'_1 \wedge \mathtt{x}_1 > \mathtt{x}'_2 \wedge \mathtt{x}_2 = \mathtt{x}'_1 \wedge \mathtt{x}_2 > \mathtt{x}'_2 \wedge \mathtt{x}'_2 < \mathtt{x}'_1$$

Lemma 4.1. $\mathrm{card}(\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)) \leqslant ((\mathbb{d}_\alpha - \mathbb{d}_1) + 3)^{2\beta} 3^{2\beta^2}$.

Indeed, in a type $(\bigwedge_i \Theta_i^{\mathrm{CST}}) \wedge (\bigwedge_{i<j} \mathtt{z}_i \sim_{i,j} \mathtt{z}_j)$, each Θ_i^{CST} can take at most $(\mathbb{d}_\alpha - \mathbb{d}_1) + 3$ values and the generalized conjunction has $(2\beta - 1)\beta \leqslant 2\beta^2$ conjuncts, and each conjunct can take at most three possible values.

The *restriction* of the type Θ to some set of variables $X \subseteq \{\mathtt{x}_i, \mathtt{x}'_i \mid i \in [1, \beta]\}$ is made of all the conjuncts in which only variables in X occur. The type Θ_1 restricted to $\{\mathtt{x}'_i \mid i \in [1, \beta]\}$ *agrees* with the type Θ_2 restricted to $\{\mathtt{x}_i \mid i \in [1, \beta]\}$ iff Θ_1 and Θ_2 are logically equivalent modulo the renaming for which $\mathtt{x}_i$ and $\mathtt{x}'_i$ are substituted, for all $i \in [1, \beta]$. For instance, in Figure 3, Θ restricted to $\{\mathtt{x}'_1, \mathtt{x}'_2\}$ agrees with Θ_0 restricted to $\{\mathtt{x}_1, \mathtt{x}_2\}$. The main properties of satisfiable types we use below are stated in the next lemma.

Lemma 4.2.

(I) : *Let* $\boldsymbol{z}, \boldsymbol{z}' \in \mathbb{Z}^\beta$. *There is a unique satisfiable type* $\Theta \in \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ *such that* $\mathbb{Z} \models \Theta(\boldsymbol{z}, \boldsymbol{z}')$.

(II): *For every constraint* Θ *built over the terms* $\mathtt{x}_1, \ldots, \mathtt{x}_\beta, \mathtt{x}'_1, \ldots, \mathtt{x}'_\beta$ *and the constants* $\mathbb{d}_1, \ldots, \mathbb{d}_\alpha$ *there is a disjunction* $\Theta_1 \vee \cdots \vee \Theta_\gamma$ *logically equivalent to* Θ *and each* Θ_i *belongs to* $\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ *(empty disjunction stands for* $\perp$*).*

(III): *For all* $\Theta \neq \Theta' \in \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$, *the constraint* $\Theta \wedge \Theta'$ *is not satisfiable.*

The proof of Lemma 4.2 is by an easy verification and its statement justifies the term 'type' used in this context.

Abstraction with types. A *symbolic tree* t is a map $t : [0, D-1]^* \to \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$. Symbolic trees are intended to be abstractions of trees labelled with concrete values in $\mathbb{Z}$, defined as follows.

Given a tree $\mathbb{t} : [0, D-1]^* \to \Sigma \times \mathbb{Z}^\beta$, its *abstraction* is the symbolic tree $t_{\mathbb{t}} : [0, D-1]^* \to \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ such that for all $\mathbf{n} \cdot i \in [0, D-1]^*$ with $\mathbb{t}(\mathbf{n}) = (\mathtt{a}, \boldsymbol{z})$ and $\mathbb{t}(\mathbf{n} \cdot i) = (\mathtt{a}_i, \boldsymbol{z}_i)$, $t_{\mathbb{t}}(\mathbf{n} \cdot i) \stackrel{\text{def}}{=} (\mathtt{a}_i, \Theta_i)$ for the unique $\Theta_i \in \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ such that $\mathbb{Z} \models \Theta_i(\boldsymbol{z}, \boldsymbol{z}_i)$. Note that the primed variables in Θ_i refer to the β values at the node $\mathbf{n} \cdot i$, whereas the unprimed ones refer to the β values at the parent node $\mathbf{n}$. For the root node ε with $\mathbb{t}(\varepsilon) = (\mathtt{a}, \boldsymbol{z})$, which has no parent node, we fix some arbitrary $\mathbf{0} \in \mathbb{Z}^\beta$, and set $t_{\mathbb{t}}(\varepsilon) \stackrel{\text{def}}{=} (\mathtt{a}, \Theta)$ for the unique $\Theta \in \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ such that $\mathbb{Z} \models \Theta(\mathbf{0}, \boldsymbol{z})$. A symbolic tree t is *satisfiable* $\stackrel{\text{def}}{\Leftrightarrow}$ there is $\mathbb{t} : [0, D-1]^* \to \Sigma \times \mathbb{Z}^\beta$ such that $t_{\mathbb{t}} = t$. We say that $\mathbb{t}$ *witnesses the satisfaction of* t, also written $\mathbb{t} \models t$. A symbolic tree t is *regular* if its set of subtrees is finite.

$\mathbb{A}$-consistency. In our quest to decide whether $\mathrm{L}(\mathbb{A}) \neq \emptyset$, we are interested in symbolic trees that satisfy certain properties that we subsume under the name $\mathbb{A}$-*consistent*. A symbolic tree $t : [0, D-1]^* \to \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ is $\mathbb{A}$-*consistent* if the following conditions are satisfied:

- t is *locally consistent*: for every node $\mathbf{n}$, the type Θ labelling $\mathbf{n}$ restricted to $\mathtt{x}'_1, \ldots, \mathtt{x}'_\beta$ agrees with all types Θ_i labelling its children nodes $\mathbf{n} \cdot i$ restricted to $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$, and

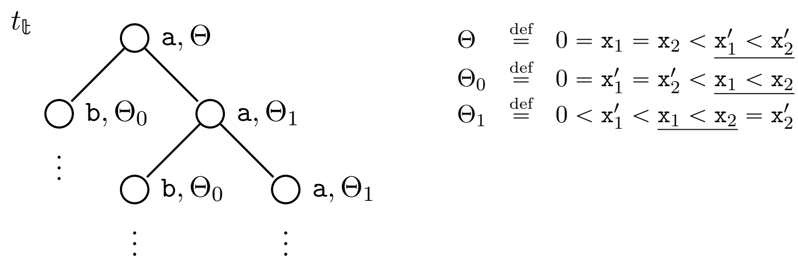


FIGURE 3. The symbolic tree $t_{\mathbb{t}}$ for $\mathbb{t}$ from Figure 1

- there exists an initialized and accepting tree $\rho : [0, D-1]^* \to \delta$ that satisfies condition (i) of the definition of runs (see page 7), and for all $\mathbf{n} \in [0, D-1]^*$ with $t(\mathbf{n}) = (\mathtt{a}, \Theta)$, $t(\mathbf{n} \cdot i) = (\mathtt{a}_i, \Theta_i)$ for all $i \in [0, D-1]$, and $\rho(\mathbf{n}) = (q, \mathtt{a}, (\Theta'_0, q_0) \dots (\Theta'_{D-1}, q_{D-1}))$, we have $\Theta_i \models \Theta'_i$ for all $i \in [0, D-1]$.

Example 4.3. In Figure 3, we show the abstraction $t_{\mathbb{t}}$ of the tree $\mathbb{t}$ from Figure 1. We assume that $\mathbb{d}_1 = \mathbb{d}_\alpha = 0$ is the only constant; consequently, $t_{\mathbb{t}}$ uses constraints in $\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ that are built with variables $\mathtt{x}_1, \mathtt{x}_2$, their primed variants $\mathtt{x}'_1, \mathtt{x}'_2$, and the constant $\mathbb{d}_1$. We underline constraints to illustrate the property of local consistency.

Lemma 4.4. *Let $\mathbb{t} \in \mathrm{L}(\mathbb{A})$. The symbolic tree $t_{\mathbb{t}}$ is $\mathbb{A}$-consistent.*

It is routine to show Lemma 4.4. Next we show that the set of all $\mathbb{A}$-consistent symbolic trees is ω-regular, that is, it can be accepted by a classical tree automaton without constraints. In the following, we use the standard letter A to distinguish automata *without constraints* from TCA (in the same way, we use t to denote symbolic trees whereas trees with data values are denoted by $\mathbb{t}$ by default).

Lemma 4.5. *There exists a Büchi tree automaton (without constraints) $A_{cons(\mathbb{A})}$ such that $\mathrm{L}(A_{cons(\mathbb{A})}) = \{t \mid t$ is an $\mathbb{A}$-consistent symbolic tree$\}$, the number of locations is bounded by $\mathrm{card}(\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)) \times \mathrm{card}(Q)$, and the transition relation can be decided in polynomial-time in $\mathrm{card}(\Sigma) + \mathrm{card}(\delta) + D + \beta + \mathtt{MCS}(\mathbb{A})$.*

See also Lemma 4.1 for an upper bound on $\mathrm{card}(\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha))$ that is exponential in β for fixed $\mathbb{d}_1, \mathbb{d}_\alpha$.

Proof. Let $A_{\mathrm{cons}(\mathbb{A})}$ be the Büchi tree automaton

$$A_{\mathrm{cons}(\mathbb{A})} \stackrel{\text{def}}{=} (Q', \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha), D, Q'_{\mathrm{in}}, \delta', F'),$$

where the components are defined as follows.

- $Q' \stackrel{\text{def}}{=} \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha) \times Q$,
- $Q'_{\mathrm{in}} \stackrel{\text{def}}{=} \{\Theta \in \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha) \mid \text{there exists } \boldsymbol{z} \in \mathbb{Z}^\beta \text{ such that } \mathbb{Z} \models \Theta(\mathbf{0}, \boldsymbol{z})\} \times Q_{\mathrm{in}}$,
- $F' \stackrel{\text{def}}{=} \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha) \times F$.
- $((\Theta, q), (\mathtt{a}, \Theta), (\Theta_0, q_0), \dots, (\Theta_{D-1}, q_{D-1})) \in \delta' \stackrel{\text{def}}{\Leftrightarrow}$ there exists a transition
$$(q, \mathtt{a}, (\Theta'_0, q_0), \dots, (\Theta'_{D-1}, q_{D-1})) \in \delta$$
such that
  - for all $i \in [0, D-1]$, $\Theta_i \models \Theta'_i$ (PTIME check because $\Theta_i \in \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$),
  - $\Theta_0, \dots, \Theta_{D-1}$ agree on $\mathtt{x}_1, \dots, \mathtt{x}_\beta$ (same parent node),

– Θ restricted to $\mathtt{x}'_1, \ldots, \mathtt{x}'_\beta$ agrees with Θ_0 restricted to $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$.

It is not hard to check that $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})}) = \{t \mid t \text{ is an } \mathbb{A}\text{-consistent symbolic tree}\}$. The transition relation δ' can be decided in polynomial-time in $\mathrm{card}(\Sigma) + \mathrm{card}(\delta) + D + \beta + \mathtt{MCS}(\mathbb{A})$ (all the values in this sum are less than the size of $\mathbb{A}$): in order to check whether $((\Theta, q), (\mathtt{a}, \Theta), (\Theta_0, q_0), \ldots, (\Theta_{D-1}, q_{D-1})) \in \delta'$, we might go through all the transitions in δ and verify the satisfaction of the three conditions above. For the first condition, we need to check D times whether $\Theta_i \models \Theta'_i$, which requires polynomial-time in $\beta + \mathtt{MCS}(\mathbb{A})$. A similar time-complexity is required to check the satisfaction of the two other conditions. Moreover, $\mathrm{card}(Q') = \mathrm{card}(\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)) \times \mathrm{card}(Q)$ and $\mathrm{card}(\delta') \leqslant \mathrm{card}(Q')^{D+1} \times \mathrm{card}(\Sigma) \times \mathrm{card}(\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha))$. □

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] and [LOS20, Theorem 11].

Lemma 4.6. $\mathrm{L}(\mathbb{A}) \neq \varnothing$ *iff there is a* satisfiable *symbolic tree in* $\mathrm{L}(A_{cons(\mathbb{A})})$.

Proof. "only if": Suppose $\mathrm{L}(\mathbb{A}) \neq \varnothing$. Then there exists some $\mathbb{t} : [0, D-1]^* \to \Sigma \times \mathbb{Z}^\beta$ in $\mathrm{L}(\mathbb{A})$. The abstraction $t_{\mathbb{t}}$ of $\mathbb{t}$ is clearly satisfiable. By Lemma 4.4, $t_{\mathbb{t}}$ is $\mathbb{A}$-consistent. By Lemma 4.5, $t_{\mathbb{t}} \in \mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$.

"if": Suppose there exists some symbolic tree $t \in \mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$ such that t is satisfiable. Satisfiability of t entails the existence of $\mathbb{t} : [0, D-1]^* \to \Sigma \times \mathbb{Z}^\beta$ such that $t_{\mathbb{t}} = t$, that is, for all $\mathbf{n} \cdot i \in [0, D-1]^*$ with $\mathbb{t}(\mathbf{n}) = (\mathtt{a}, \boldsymbol{z})$, $\mathbb{t}(\mathbf{n} \cdot i) = (\mathtt{a}_i, \boldsymbol{z}_i)$ and $t(\mathbf{n} \cdot i) = (\mathtt{a}_i, \Theta_i)$, we have $\mathbb{Z} \models \Theta_i(\boldsymbol{z}, \boldsymbol{z}_i)$. Moreover, if $\mathbb{t}(\varepsilon) = (\mathtt{a}, \boldsymbol{z})$, and $t(\varepsilon) = (\mathtt{a}, \Theta)$, then $\mathbb{Z} \models \Theta(\mathbf{0}, \boldsymbol{z})$. By $t \in \mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$ and Lemma 4.5, t is $\mathbb{A}$-consistent. Hence there exists an initialized accepting tree $\rho : [0, D-1]^* \to \delta$ that satisfies condition (i) of the definition of runs, and for all $\mathbf{n} \in [0, D-1]^*$ with $t(\mathbf{n}) = (\mathtt{a}, \Theta)$, $t(\mathbf{n} \cdot 0) = (\mathtt{a}_0, \Theta_0)$, $\ldots$, $t(\mathbf{n} \cdot (D-1)) = (\mathtt{a}_{D-1}, \Theta_{D-1})$ and $\rho(\mathbf{n}) = (q, \mathtt{a}, (\Theta'_0, q_0), \ldots, (\Theta'_{D-1}, q_{D-1}))$, we have $\Theta_i \models \Theta'_i$ for all $i \in [0, D-1]$ (by definition of δ' in $A_{\mathrm{cons}(\mathbb{A})}$). Let us briefly verify that ρ is a run of $\mathbb{A}$ on $\mathbb{t}$, and therefore $\mathrm{L}(\mathbb{A}) \neq \varnothing$. For this, it suffices to prove that condition (ii) of the definition of runs is satisfied. Let $\mathbf{n} \in [0, D-1]^*$ with $\mathbb{t}(\mathbf{n}) = (\mathtt{a}, \boldsymbol{z})$ and $\mathbb{t}(\mathbf{n} \cdot i) = (\mathtt{a}_i, \boldsymbol{z}_i)$ for all $0 \leqslant i < D$, and let $\rho(\mathbf{n}) = (q, \mathtt{a}, (\Theta'_0, q_0), \ldots, (\Theta'_{D-1}, q_{D-1}))$. By the transition relation of $A_{\mathrm{cons}(\mathbb{A})}$, for all $i \in [0, D-1]$, $\Theta_i \models \Theta'_i$ with $t(\mathbf{n} \cdot i) = (\mathtt{a}_i, \Theta_i)$. But, as proved above, we also have $\mathbb{Z} \models \Theta_i(\boldsymbol{z}, \boldsymbol{z}_i)$ and therefore $\mathbb{Z} \models \Theta'_i(\mathbf{n} \cdot i)(\boldsymbol{z}, \boldsymbol{z}_i)$. □

By virtue of Lemma 4.6, the nonemptiness problem for TCA can be reduced to deciding whether $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$ contains a satisfiable symbolic tree t. However, not every symbolic tree is satisfiable, as the following example shows.

Example 4.7. Assume that every node along the rightmost branch in the symbolic tree $t_{\mathbb{t}}$ in Figure 3 is labelled with $(\mathtt{a}, \Theta_1)$. Then $t_{\mathbb{t}}$ is not satisfiable: in order to satisfy the constraint $\mathtt{x}'_1 < \mathtt{x}_1$, the value of $\mathtt{x}_1$ must finally become smaller than $\mathbb{d}_1$, violating the constraint $\mathbb{d}_1 < \mathtt{x}_1$.

Thus, the most important property is to check whether $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$ contains some *satisfiable* symbolic tree. This is the subject of the next two subsections.

4.2. Satisfiability for Regular Locally Consistent Symbolic Trees.

In this subsection, we focus on deciding when $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$ contains a *satisfiable* symbolic tree, while evaluating the computational complexity to check its existence. For this, we define a property on symbolic trees, denoted by ($\bigstar^{\mathrm{C}}$), such that if a symbolic tree t satisfies ($\bigstar^{\mathrm{C}}$) and is regular, then t is satisfiable. We will prove that there exists a Rabin tree automaton that accepts all symbolic trees that satisfy ($\bigstar^{\mathrm{C}}$). As a result, we will be able to reduce the nonemptiness problem to the nonemptiness problem for the product automaton of this automaton and $A_{\mathrm{cons}(\mathbb{A})}$. The definition of ($\bigstar^{\mathrm{C}}$) is based on an infinite tree-like graph inferred by a given symbolic tree t, denoted by G_t^{C}. Similar symbolic structures are introduced in [Lut01, DD07, CKL13, LOS20].

So let us start with defining G_t^{C}. Let $t : [0, D-1]^* \to \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ be a symbolic tree. The graph G_t^{C} is equal to the structure

$$G_t^{\mathrm{C}} = (V_t, \xrightarrow{=}, \xrightarrow{<}, U_{<\mathbb{d}_1}, (U_{\mathbb{d}})_{\mathbb{d} \in [\mathbb{d}_1, \mathbb{d}_\alpha]}, U_{>\mathbb{d}_\alpha}),$$

where

- $V_t = [0, D-1]^* \times \mathtt{T}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ with $\mathtt{T}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha) \stackrel{\text{def}}{=} \{\mathtt{x}_1, \ldots, \mathtt{x}_\beta\} \cup \{\mathbb{d}_1, \mathbb{d}_\alpha\}$,
- $\xrightarrow{=}$ and $\xrightarrow{<}$ are two binary relations over V_t, and
- $\{U_{<\mathbb{d}_1}, U_{\mathbb{d}_1}, U_{\mathbb{d}_1+1}, \ldots, U_{\mathbb{d}_\alpha}, U_{>\mathbb{d}_\alpha}\}$ is a partition of V_t.

Elements in $\{\mathtt{x}_1, \ldots, \mathtt{x}_\beta\} \cup \{\mathbb{d}_1, \mathbb{d}_\alpha\}$ are denoted by $\mathtt{xd}, \mathtt{xd}_1, \mathtt{xd}_2, \ldots$ (variables or constants) and $V_t^{var} \stackrel{\text{def}}{=} [0, D-1]^* \times \{\mathtt{x}_1, \ldots, \mathtt{x}_\beta\}$.

Two nodes $\mathbf{n}, \mathbf{n}' \in [0, D-1]^*$ in t are *neighbours* $\stackrel{\text{def}}{\Leftrightarrow}$ 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 $(\mathbf{n}, \mathtt{xd})$ and $(\mathbf{n}', \mathtt{xd}')$ in $[0, D-1]^* \times \mathtt{T}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ are *neighbours* $\stackrel{\text{def}}{\Leftrightarrow}$ $\mathbf{n}$ and $\mathbf{n}'$ are neighbours. By construction, the edges in G_t^{C} are possible only between neighbour elements.

The rationale behind the construction of G_t^{C} is to reflect the constraints between parent and children nodes as well as the constraints regarding constants, in such a way that, if $\mathbb{t}$ witnesses the satisfaction of t, then, e.g., $\mathbb{t}(\mathbf{n})(\mathtt{xd}) < \mathbb{t}(\mathbf{n}')(\mathtt{xd}')$ if $(\mathbf{n}, \mathtt{xd}) \xrightarrow{<} (\mathbf{n}', \mathtt{xd}')$, and $\mathbb{t}(\mathbf{n})(\mathtt{xd}) = \mathbb{d}_1$ if $(\mathbf{n}, \mathtt{xd}) \in U_{\mathbb{d}_1}$ (see also Lemma 4.9). Here are all conditions for building G_t^{C}.

(VAR): For all $(\mathbf{n}, \mathtt{x}_i), (\mathbf{n}', \mathtt{x}_{i'}) \in V_t^{var}$, for all $\sim \in \{<, =\}$, $(\mathbf{n}, \mathtt{x}_i) \xrightarrow{\sim} (\mathbf{n}', \mathtt{x}_{i'})$ iff one of the conditions below holds:
- either $\mathbf{n}' = \mathbf{n} \cdot j$ and $\mathtt{x}_i \sim \mathtt{x}'_{i'}$ in Θ with $t(\mathbf{n}') = (\mathtt{a}, \Theta)$,
- or $\mathbf{n} = \mathbf{n}'$ and $\mathtt{x}'_i \sim \mathtt{x}'_{i'}$ in Θ with $t(\mathbf{n}') = (\mathtt{a}, \Theta)$,
- or $\mathbf{n} = \mathbf{n}' \cdot j$ and $\mathtt{x}'_i \sim \mathtt{x}_{i'}$ in Θ with $t(\mathbf{n}) = (\mathtt{a}, \Theta)$.

(P1): For all $\mathbb{d} \in [\mathbb{d}_1, \mathbb{d}_\alpha]$ and $(\mathbf{n}, \mathtt{x}_j) \in V_t^{var}$, $(\mathbf{n}, \mathtt{x}_j) \in U_{\mathbb{d}}$ iff $\mathtt{x}'_j = \mathbb{d}$ in Θ with $t(\mathbf{n}) = (\mathtt{a}, \Theta)$.

(P2): For all $(\mathbf{n}, \mathtt{x}_j) \in V_t^{var}$, $(\mathbf{n}, \mathtt{x}_j) \in U_{<\mathbb{d}_1}$ iff $\mathtt{x}'_j < \mathbb{d}_1$ in Θ with $t(\mathbf{n}) = (\mathtt{a}, \Theta)$.

(P3): For all $(\mathbf{n}, \mathtt{x}_j) \in V_t^{var}$, $(\mathbf{n}, \mathtt{x}_j) \in U_{>\mathbb{d}_\alpha}$ iff $\mathtt{x}'_j > \mathbb{d}_\alpha$ in Θ with $t(\mathbf{n}) = (\mathtt{a}, \Theta)$.

(P4): For all $\mathbf{n} \in [0, D-1]^*$, $(\mathbf{n}, \mathbb{d}_1) \in U_{\mathbb{d}_1}$ and $(\mathbf{n}, \mathbb{d}_\alpha) \in U_{\mathbb{d}_\alpha}$.

(CONS): This condition is about elements of V_t labelled by constants and how the edge labels reflect the relationships between the constants. Formally, for all $(\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}') \in (V_t \backslash V_t^{var})$ such that $\mathbf{n}$ and $\mathbf{n}'$ are neighbours, for all $\mathbb{d}^\dagger, \mathbb{d}^{\dagger\dagger}$ in '$< \mathbb{d}_1$', $\mathbb{d}_1$, ..., $\mathbb{d}_\alpha$, '$> \mathbb{d}_\alpha$' such that $(\mathbf{n}, \mathtt{xd}) \in U_{\mathbb{d}^\dagger}$ and $(\mathbf{n}', \mathtt{xd}') \in U_{\mathbb{d}^{\dagger\dagger}}$, for all $\sim \in \{<, =\}$, there is an edge $(\mathbf{n}, \mathtt{xd}) \xrightarrow{\sim} (\mathbf{n}', \mathtt{xd}')$ iff
- either $\mathbb{d}^\dagger, \mathbb{d}^{\dagger\dagger} \in [\mathbb{d}_1, \mathbb{d}_\alpha]$ and $\mathbb{d}^\dagger \sim \mathbb{d}^{\dagger\dagger}$,
- or $\mathbb{d}^\dagger =$ '$< \mathbb{d}_1$', $\mathbb{d}^{\dagger\dagger} \neq$ '$< \mathbb{d}_1$' and $\sim$ is equal to $<$,
- or $\mathbb{d}^\dagger \neq$ '$> \mathbb{d}_\alpha$', $\mathbb{d}^{\dagger\dagger} =$ '$> \mathbb{d}_\alpha$' and $\sim$ is equal to $<$.

Here, $\mathbb{d}^\dagger$ and $\mathbb{d}^{\dagger\dagger}$ are subscripts that are not associated to data values when it takes the values '$< \mathbb{d}_1$' or '$> \mathbb{d}_\alpha$'.

Below, we also write $(\mathbf{n}, \mathtt{xd}) \xrightarrow{>} (\mathbf{n}', \mathtt{xd}')$ instead of $(\mathbf{n}', \mathtt{xd}') \xrightarrow{<} (\mathbf{n}, \mathtt{xd})$. Observe that in the condition (CONS), it is also possible to remove the requirement that $\mathbf{n}$ and $\mathbf{n}'$ are

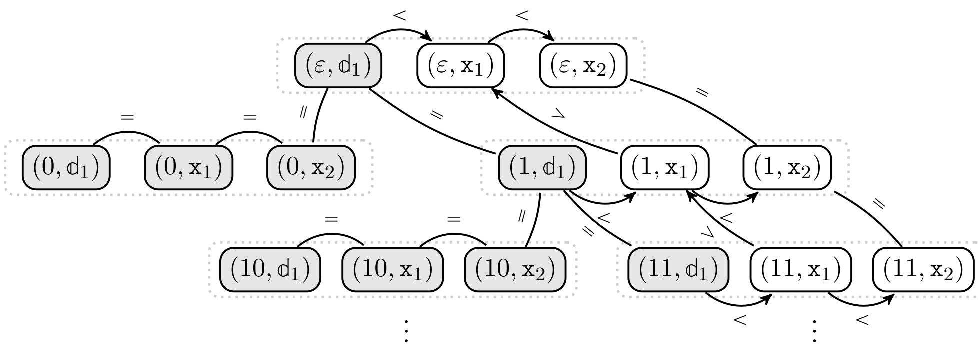


Figure 4. The labelled graph $G^{\mathrm{C}}_{t_{\mathbb{t}}}$ for the symbolic tree $t_{\mathbb{t}}$ from Figure 3 (page 11).

neighbours. This would still be fine but we added this condition so that all over G^{C}_t, an edge exists between two elements only if the underlying nodes are neighbours (but other edges could be easily inferred). The forthcoming condition ($\bigstar^{\mathrm{C}}$) shall be defined on such labelled graphs, see also the symbolic trees accepted by construction of $A_{\bigstar^{\mathrm{C}}}$ in Lemma 4.12. The construction of G^{C}_t is done from any symbolic tree t, even if it is not locally consistent; local consistency is taken care of by $A_{\mathrm{cons}(\mathbb{A})}$.

Example 4.8. In Figure 4, we illustrate the definition of the graph $G^{\mathrm{C}}_{t_{\mathbb{t}}}$ for the symbolic tree $t_{\mathbb{t}}$ in Figure 3. The edges labelled with $=$ or $<$ reflect the constraints (we omit edges if they can be inferred from the other edges). For instance, $(1, \mathtt{x}_1) \xrightarrow{<} (\varepsilon, \mathtt{x}_1)$ corresponds to the constraint $\mathtt{x}'_1 < \mathtt{x}_1$ (by application of (VAR), third item). Grey nodes are in $U_{\mathbb{d}_1}$, all other nodes are in $U_{>\mathbb{d}_1}$ (no nodes in $U_{<\mathbb{d}_1}$).

The rationale behind the construction of G^{C}_t is best illustrated with the next lemma.

Lemma 4.9. *Let t be a symbolic tree and $\mathbb{t} : [0, D-1]^* \to \mathbb{Z}^\beta$ be such that $\mathbb{t}$ witnesses the satisfaction of t. Then, if $(\mathbf{n}, \mathtt{xd}) \xrightarrow{\sim} (\mathbf{n}', \mathtt{xd}')$ is an edge in G^C_t for some $\sim \in \{<, =\}$, then $\mathbb{t}(\mathbf{n})(\mathtt{xd}) \sim \mathbb{t}(\mathbf{n}')(\mathtt{xd}')$.*

By convention, $\mathbb{t}(\mathbf{n})(\mathbb{d}_1) \stackrel{\text{def}}{=} \mathbb{d}_1$ and $\mathbb{t}(\mathbf{n})(\mathbb{d}_\alpha) \stackrel{\text{def}}{=} \mathbb{d}_\alpha$. Moreover, whenever $\mathbb{t}(\mathbf{n}) = (z_1, \ldots, z_\beta)$, we write $\mathbb{t}(\mathbf{n})(\mathtt{x}_i)$ to denote z_i.

Proof. Suppose $(\mathbf{n}, \mathtt{xd}) \xrightarrow{=} (\mathbf{n}', \mathtt{xd}')$ is an edge in G^{C}_t. Let $t(\mathbf{n}) = (\cdot, \Theta_{\mathbf{n}})$ and $t(\mathbf{n}') = (\cdot, \Theta_{\mathbf{n}'})$. We use $\Theta'_{\mathbf{n}}$ to denote the restriction of $\Theta_{\mathbf{n}}$ to $\{\mathtt{x}'_1, \ldots, \mathtt{x}'_\beta\}$; similarly for $\Theta'_{\mathbf{n}'}$. By (VAR) and (CONS), there are four possible cases.

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

- Suppose $\mathtt{xd} = \mathtt{x}_i$ for some $1 \leqslant i \leqslant \beta$ and $\mathtt{xd}' = \mathbb{d}_1$. By (P4), $(\mathbf{n}', \mathbb{d}_1) \in U_{\mathbb{d}_1}$. By (CONS), $(\mathbf{n}, \mathtt{x}_i) \in U_{\mathbb{d}_1}$. By (P1), $\mathtt{x}'_i = \mathbb{d}_1 \in \Theta_{\mathbf{n}}$. By $\mathbb{Z} \models \Theta'_{\mathbf{n}}(\mathbb{t}(\mathbf{n}))$ we can conclude that $\mathbb{t}(\mathbf{n})(\mathtt{x}_i) = \mathbb{d}_1$. Hence $\mathbb{t}(\mathbf{n})(\mathtt{x}_i) = \mathbb{t}(\mathbf{n}')(\mathbb{d}_1)$.
- Suppose $\mathtt{xd} = \mathbb{d}_1$ and $\mathtt{xd}' = \mathtt{x}_i$ for some $1 \leqslant i \leqslant \beta$. The proof is symmetric to the proof for the previous case.
- Suppose $\mathtt{xd} = \mathbb{d}_\alpha$ or $\mathtt{xd}' = \mathbb{d}_\alpha$. The proof is very similar to the proof for the previous two cases.

Now suppose $(\mathbf{n}, \mathtt{xd}) \xrightarrow{<} (\mathbf{n}', \mathtt{xd}')$ is an edge in G^{C}_t.

- Suppose $\mathtt{xd} = \mathtt{x}_i$, $\mathtt{xd}' = \mathtt{x}_{i'}$ for some $1 \leqslant i, i' \leqslant \beta$ by (VAR). The proof is analogous to the proof for the corresponding case for $(\mathbf{n}, \mathtt{xd}) \xrightarrow{=} (\mathbf{n}', \mathtt{xd}')$.
- Suppose $\mathtt{xd}' = \mathbb{d}_1$. Then $\mathtt{xd} = \mathtt{x}_i$ for some $1 \leqslant i \leqslant \beta$ and $(\mathbf{n}, \mathtt{x}_i) \in U_{<\mathbb{d}_1}$ by (CONS). Then $\mathtt{x}'_i < \mathbb{d}_1 \in \Theta_{\mathbf{n}}$ by (P2). By $\mathbb{Z} \models \Theta'_{\mathbf{n}}(\mathbb{t}(\mathbf{n}))$ we can conclude $\mathbb{t}(\mathbf{n})(\mathtt{x}_i) < \mathbb{d}_1$, and hence $\mathbb{t}(\mathbf{n})(\mathtt{x}_i) < \mathbb{t}(\mathbf{n}')(\mathbb{d}_1)$.
- Suppose $\mathtt{xd} = \mathbb{d}_1$. Then – by (CONS) – we have $(\mathbf{n}', \mathtt{xd}') \in U_{\mathbb{d}^\dagger}$ for some $\mathbb{d}^\dagger \in [\mathbb{d}_1 + 1, \mathbb{d}_\alpha] \cup \{\text{‘}> \mathbb{d}_\alpha\text{’}\}$.
  - Suppose $\mathbb{d}^\dagger \in [\mathbb{d}_1 + 1, \mathbb{d}_\alpha]$ and $\mathtt{xd}' = \mathtt{x}_i$ for some $i \in [1, \beta]$. By (P1), we have $\mathtt{x}'_i = \mathbb{d}^\dagger \in \Theta_{\mathbf{n}'}$. By $\mathbb{Z} \models \Theta'_{\mathbf{n}'}(\mathbb{t}(\mathbf{n}))$ we can conclude $\mathbb{t}(\mathbf{n})(\mathtt{x}_i) = \mathbb{d}^\dagger$. Hence $\mathbb{t}(\mathbf{n})(\mathbb{d}_1) < \mathbb{t}(\mathbf{n}')(\mathtt{x}_i)$.
  - Suppose $\mathbb{d}^\dagger = \mathbb{d}_\alpha$ and $\mathtt{xd}' = \mathbb{d}_\alpha$. Then obviously $\mathbb{t}(\mathbf{n})(\mathbb{d}_1) < \mathbb{t}(\mathbf{n}')(\mathbb{d}_\alpha)$.
  - Suppose $\mathbb{d}^\dagger = \text{‘}> \mathbb{d}_\alpha\text{’}$. Then $\mathtt{xd}' = \mathtt{x}_i$ for some $i \in [1, \beta]$, and $\mathtt{x}'_i > \mathbb{d}_\alpha \in \Theta_{\mathbf{n}'}$ by (P3). By $\mathbb{Z} \models \Theta'_{\mathbf{n}'}(\mathbb{t}(\mathbf{n}'))$ we can conclude that $\mathbb{t}(\mathbf{n}')(\mathtt{x}_i) > \mathbb{d}_\alpha$, so that $\mathbb{t}(\mathbf{n})(\mathbb{d}_1) < \mathbb{t}(\mathbf{n}')(\mathtt{x}_i)$.
- Suppose $\mathtt{xd} = \mathbb{d}_\alpha$. Then $\mathtt{xd}' = \mathtt{x}_i$ for some $i \in [1, \beta]$ and $(\mathbf{n}', \mathtt{x}_i) \in U_{>\mathbb{d}_\alpha}$. Then $\mathtt{x}'_i > \mathbb{d}_\alpha \in \Theta'_{\mathbf{n}'}$. By $\mathbb{Z} \models \Theta'_{\mathbf{n}'}(\mathbb{t}(\mathbf{n}'))$, we can conclude that $\mathbb{t}(\mathbf{n}')(\mathtt{x}_i) > \mathbb{d}_\alpha$. Hence $\mathbb{t}(\mathbf{n})(\mathbb{d}_\alpha) < \mathbb{t}(\mathbf{n}')(\mathtt{x}_i)$.
- Suppose $\mathtt{xd}' = \mathbb{d}_\alpha$. Then $(\mathbf{n}, \mathtt{xd}) \in U_{\mathbb{d}^\dagger}$ for some $\mathbb{d}^\dagger \in \{\text{‘}< \mathbb{d}_1\text{’}\} \cup [\mathbb{d}_1, \mathbb{d}_\alpha - 1]$. The proof is very similar to the last but one case. □

A map $p : \mathbb{N} \to V_t$ is a *path map* in G^{C}_t iff for all $i \in \mathbb{N}$, either $p(i) \xrightarrow{=} p(i+1)$ or $p(i) \xrightarrow{<} p(i+1)$ in G^{C}_t. Similarly, $rp : \mathbb{N} \to V_t$ is a *reverse path map* in G^{C}_t iff for all $i \in \mathbb{N}$, either $rp(i+1) \xrightarrow{=} rp(i)$ or $rp(i+1) \xrightarrow{<} rp(i)$. These definitions make sense because between two elements there is at most one labelled edge (or maybe two but with the same equality sign). A path map p (resp. reverse path map rp) is *strict* iff $\{i \in \mathbb{N} \mid p(i) \xrightarrow{<} p(i+1)\}$ (resp. $\{i \in \mathbb{N} \mid rp(i+1) \xrightarrow{<} rp(i)\}$) is infinite. An *infinite branch* $\mathcal{B}$ is an element of $[0, D-1]^\omega$. We write $\mathcal{B}[i, j]$ with $i \leqslant j$ to denote the subsequence $\mathcal{B}(i) \cdot \mathcal{B}(i+1) \cdots \mathcal{B}(j)$. Given $(\mathbf{n}, \mathtt{xd}) \in V_t$, a path map p *from* $(\mathbf{n}, \mathtt{xd})$ *along* $\mathcal{B}$ is such that $p(0) = (\mathbf{n}, \mathtt{xd})$ and for all $i \geqslant 0$, $p(i)$ is of the form $(\mathbf{n} \cdot \mathcal{B}[0, i], \cdot)$. A reverse path map rp from $(\mathbf{n}, \mathtt{xd})$ along $\mathcal{B}$ admits a similar definition.

We say that a symbolic tree t satisfies $(\bigstar^{\mathrm{C}})$ if in G^{C}_t there are *no* elements $(\mathbf{n}, \mathtt{xd})$, $(\mathbf{n}, \mathtt{xd}')$ (same node $\mathbf{n}$ from $[0, D-1]^*$) and no infinite branch $\mathcal{B}$ such that

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

The following proposition states a key property: non-satisfaction of a regular locally consistent symbolic tree can be witnessed along a *single* branch by violation of $(\bigstar^{\mathrm{C}})$, following the remarkable result established in [LOS20, Lemma 22]. The approach developed in Section 8 takes advantage of the techniques to prove [LOS20, Lemma 22].

Proposition 4.10. *For every regular locally consistent symbolic tree t, t satisfies $(\bigstar^C)$ iff t is satisfiable.*

A proof can be found in Section 8. A comparison between the condition $(\bigstar^{\mathrm{C}})$ and the condition $(\bigstar)$ from [LOS20, Lab21] is postponed to Section 9.2. We recall that there are *nonregular* locally consistent symbolic trees t such that G_t^{C} satisfies $(\bigstar^{\mathrm{C}})$ but t is not satisfiable (see e.g. the proof of [DD07, Corollary 6.5], and [LOS20]); indeed, satisfiability of symbolic trees is not an ω-regular property. Observe also that t is satisfiable (not necessarily regular) entails that G_t^{C} satisfies $(\bigstar^{\mathrm{C}})$ (by Lemma 4.9).

Example 4.11. Consider the non-satisfiable symbolic tree $t_{\mathbb{t}}$ from Example 4.7, for which we depict $G_{t_{\mathbb{t}}}^{\mathrm{C}}$ in Figure 4. Note that $t_{\mathbb{t}}$ does not satisfy $(\bigstar^{\mathrm{C}})$: for the infinite branch $\mathcal{B} = 1^\omega$, there exists a path map p from $(\varepsilon, \mathbb{d}_1)$ along $\mathcal{B}$, there exists a strict reverse path map rp from $(\varepsilon, \mathtt{x}_1)$ along $\mathcal{B}$, and for all $i \in \mathbb{N}$ we have $p(i) \xrightarrow{\leq} rp(i)$.

The next result states that $(\bigstar^{\mathrm{C}})$ is ω-regular (Lemma 4.12): there is a Rabin tree automaton $A_{\bigstar^{\mathrm{C}}}$ that captures it, so that satisfiability of symbolic trees can be overapproximated advantageously, see Lemma 4.13. Observe that local consistency could be easily defined in the conditions for G_t^{C}. One reason for keeping separate $A_{\mathrm{cons}(\mathbb{A})}$ dealing with local consistency and the automaton for dealing with G_t^{C} (forthcoming $A_{\bigstar^{\mathrm{C}}}$) is that for certain concrete domains (including $(\mathbb{Q}, <)$), local consistency is sufficient to guarantee the existence of a satisfiable model, so that the construction of $A_{\bigstar^{\mathrm{C}}}$ is not even necessary. We hence prefer this modular approach.

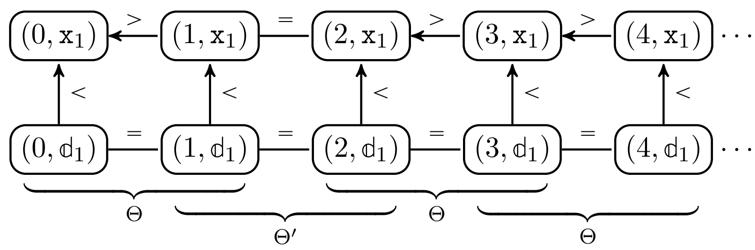
Lemma 4.12. *There is a Rabin tree automaton $A_{\bigstar^C}$ such that $\mathrm{L}(A_{\bigstar^C}) = \{t \mid t \text{ sat. } (\bigstar^C)\}$, the number of Rabin pairs is bounded above by $8(\beta+2)^2+3$, the number of locations is exponential in β, the transition relation can be decided in polynomial-time in*

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

As a consequence, the transition relation for $A_{\bigstar^{\mathrm{C}}}$ has $\mathrm{card}(\Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)) \times (2^{\mathcal{O}(P^\star(\beta))})^{(D+1)}$ transitions for some polynomial $P^\star$, which is in $\mathrm{card}(\Sigma) \times ((\mathbb{d}_\alpha - \mathbb{d}_1) + 3)^{2\beta} 3^{2\beta^2} \times 2^{\mathcal{O}(P^\star(\beta) \times (D+1))}$.

Proof. The proof of Lemma 4.12 is structured as follows.

(1) We construct a Büchi word automaton A_B accepting the complement of $(\bigstar^{\mathrm{C}})$ for $D = 1$.
(2) We determinize A_B and obtain a deterministic Rabin word automaton $A_{B\to R}$ such that $\mathrm{L}(A_B) = \mathrm{L}(A_{B\to R})$ (using the classical determinisation construction from [Saf89, Theorem 1.1]).
(3) From $A_{B\to R}$, we obtain a deterministic Street word automaton A_S accepting the complement of $\mathrm{L}(A_{B\to R})$; it accepts words that satisfy $(\bigstar^{\mathrm{C}})$ for $D = 1$. Here, we use the well-known fact that the negation of a Rabin acceptance condition is a Street acceptance condition.
(4) From A_S, we construct a deterministic Rabin word automaton A_R such that $\mathrm{L}(A_S) = \mathrm{L}(A_R)$ (using [Saf89, Lemma 1.2]). Observe that both $A_{B\to R}$ and A_R are Rabin word automata but $A_{B\to R}$ handles the complement of the condition $(\bigstar^{\mathrm{C}})$ whereas A_R handles $(\bigstar^{\mathrm{C}})$ itself.
(5) We construct a Rabin tree automaton $A_{\bigstar^{\mathrm{C}}}$, by "letting run A_R" along every branch of a run of $A_{\bigstar^{\mathrm{C}}}$, which is possible thanks to the determinism of A_R. Since $(\bigstar^{\mathrm{C}})$ states a property on every branch of the trees, we are done.



$$q_{\text{in}} \xrightarrow{\Theta} (\mathbb{d}_1, \mathtt{x}_1, rp, <) \xrightarrow{\Theta'} (\mathbb{d}_1, \mathtt{x}_1, rp, =) \xrightarrow{\Theta} (\mathbb{d}_1, \mathtt{x}_1, rp, <) \xrightarrow{\Theta} (\mathbb{d}_1, \mathtt{x}_1, rp, <) \ldots$$

FIGURE 5. A symbolic word w representing an infinite branch along which a path map from $(0, \mathbb{d}_1)$ and a strict reverse path map from $(0, \mathtt{x}_1)$ do not satisfy the condition $(\bigstar^{\mathrm{C}})$; below, a run of A_B on w.

So let us explain each of these steps.

(1) We construct a Büchi word automaton accepting the complement of $(\bigstar^{\mathrm{C}})$ for $D = 1$, similarly to what is done in [DD07, Section 6] and [LOS20, Section 3.4]. Let us first explain the idea of the construction. If a word w over $\Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ does not satisfy $(\bigstar^{\mathrm{C}})$, then the graph G^{C}_w contains a node $\mathbf{n}$ such that, for some $\mathtt{xd}_1$ and $\mathtt{xd}_2$, there exists a path map from $(\mathbf{n}, \mathtt{xd}_1)$ and there exists a reverse path map from $(\mathbf{n}, \mathtt{xd}_2)$ that do not satisfy the condition $(\bigstar^{\mathrm{C}})$, cf. Figure 5, where there exists a path map from $(0, \mathbb{d}_1)$ and a reverse path map from $(0, \mathtt{x}_1)$. The automaton nondeterministically guesses such nodes, and checks whether it initializes a violation of $(\bigstar^{\mathrm{C}})$. For this, the automaton remembers in its locations

- the current variable/constant $\mathtt{xd}_1$ of the path map p,
- the current variable/constant $\mathtt{xd}_2$ of the reverse path map rp,
- whether the path map p or the reverse path map rp is strict (p or rp), and
- whether in the strict (reverse) path map, it has just seen $<$ or $=$.

We call a 4-tuple $(\mathtt{xd}_1, \mathtt{xd}_2, d, \bowtie)$ a *local thread*, if $\mathtt{xd}_1, \mathtt{xd}_2 \in \mathtt{T}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$, $d \in \{p, rp\}$ and $\bowtie \in \{<, =\}$, with the intended meaning as explained above. The transition relation of the automaton 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 the condition $(\bigstar^{\mathrm{C}})$. Formally, let $A_B = (Q_B, \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha), Q_{B,\text{in}}, \delta_B, F_B)$ be the Büchi word automaton defined as follows.

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

It is not hard to prove that for all $w : \mathbb{N} \to \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$, $w \in \mathrm{L}(A_B)$ iff w does not satisfy the condition $(\bigstar^{\mathrm{C}})$. Note that the number of locations in A_B is bounded above by $4(\beta + 2)^2 + 1$.

(2) Using [Saf89, Theorem 1.1], from A_B we obtain a deterministic Rabin word automaton $A_{B\to R} = (Q_{B\to R}, \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha), Q_{B\to R,\mathrm{in}}, \delta_{B\to R}, F_{B\to R})$ such that $\mathrm{L}(A_{B\to R}) = \mathrm{L}(A_B)$. The cardinality of $Q_{B\to R}$ is in $2^{\mathcal{O}(\mathrm{card}(Q_B)\ log(\mathrm{card}(Q_B)))}$, i.e. exponential in β, and the number of acceptance pairs in $F_{B\to R}$ is equal to $2 \times \mathrm{card}(Q_B)$, i.e. equal to $8(\beta+2)^2+2$ (see also Section 7 where the proof of [Saf89, Theorem 1.1] is generalised to Büchi word constraint automata). Without any loss of generality, we can assume that $A_{B\to R}$ is also complete. As a consequence, for any $w \in (\Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha))^\omega$, there is a unique (not necessarily accepting) run ρ_w on w.

(3) We write $A_S = (Q_S, \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha), Q_{S,\mathrm{in}}, \delta_S, F_S)$ to denote the Streett automaton accepting the complement language of $\mathrm{L}(A_{B\to R})$. All the components of A_S are those from $A_{B\to R}$ but F_S in A_S is interpreted as a Streett condition (recall that the negation of a Rabin condition is a Streett condition). Consequently, the deterministic Streett word automaton A_S is syntactically equal to $A_{B\to R}$ and we have $w \in \mathrm{L}(A_S)$ iff w satisfies the condition $(\bigstar^{\mathrm{C}})$.

(4) Using [Saf89, Lemma 1.2], from A_S we define the *deterministic* Rabin word automaton $A_R = (Q_R, \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha), Q_{R,\mathrm{in}}, \delta_R, F_R)$ such that $\mathrm{L}(A_R) = \mathrm{L}(A_S)$. Before defining its components, we define a few notions. First of all, we assume that A_S has an additional $(8(\beta+2)^2+3)^{\mathrm{th}}$ Streett pair, namely (Q_S, Q_S), which is technically helpful to design A_R but does not change the language $\mathrm{L}(A_S)$. We write $\mathbf{S}_{(8(\beta+2)^2+3)}$ to denote the set of permutations on $[1, 8(\beta+2)^2+3]$, where a permutation σ is a bijection $\sigma : [1, 8(\beta+2)^2+3] \to [1, 8(\beta+2)^2+3]$. It is well-known that $\mathrm{card}(\mathbf{S}_{(8(\beta+2)^2+3)}) = (8(\beta+2)^2+3)!$. Let us define the maps $\mathfrak{g}_1, \mathfrak{g}_2 : Q_S \times \mathbf{S}_{(8(\beta+2)^2+3)} \to [1, 8(\beta+2)^2+3]$ and $\mathfrak{g}_3 : Q_S \times \mathbf{S}_{(8(\beta+2)^2+3)} \to \mathbf{S}_{(8(\beta+2)^2+3)}$ that are instrumental to define the forthcoming transition relation δ_R. Below, $q \in Q_S$ and $\sigma \in \mathbf{S}_{(8(\beta+2)^2+3)}$.

- We set $\mathfrak{g}_1(q,\sigma) \stackrel{\text{def}}{=} \min\{i \in [1, 8(\beta+2)^2+3] \mid q \in U_{\sigma(i)}\}$ where $U_{\sigma(i)}$ is from the set of pairs F_S. A minimal value always exists thanks to the addition of the new Streett pair (Q_S, Q_S).
- The value $\mathfrak{g}_2(q,\sigma)$ is defined using the L_i sets from the set of pairs F_S:

$$\mathfrak{g}_2(q,\sigma) \stackrel{\text{def}}{=} \min\{i \in [1, 8(\beta+2)^2+3] \mid q \in L_{\sigma(i)}\}.$$

- The permutation $\mathfrak{g}_3(q,\sigma)$ is obtained from σ by *moving* $\sigma(\mathfrak{g}_1(q,\sigma))$ to the rightmost position, so that $\sigma(\mathfrak{g}_1(q,\sigma))$ is at position $8(\beta+2)^2+3$ now. Formally

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

By way of example, if $\sigma(1) = 2$, $\sigma(2) = 4$, $\sigma(3) = 1$, $\sigma(4) = 3$, $\cdots$ and, 3 is the minimal value i such that $q \in U_{\sigma(i)}$ (so $q \notin U_2 \cup U_4$), then $\mathfrak{g}_3(q,\sigma)(1) = 2$, $\mathfrak{g}_3(q,\sigma)(2) = 4$, $\mathfrak{g}_3(q,\sigma)(3) = 3$ $\cdots$ and $\mathfrak{g}_3(q,\sigma)(8(\beta+2)^2+3) = 3$.

Let us now define the components of A_R.

- $Q_R \stackrel{\text{def}}{=} Q_S \times \mathbf{S}_{(8(\beta+2)^2+3)} \times [1, 8(\beta+2)^2+3]^2$.
- The transition relation δ_R is defined as follows: $(q', \sigma', e', f') \in \delta_R((q, \sigma, e, f), \mathtt{a})$ iff $q' \in \delta_S(q, \mathtt{a})$, $\sigma' = \mathfrak{g}_3(q', \sigma)$, $e' = \mathfrak{g}_1(q', \sigma)$ and $f' = \mathfrak{g}_2(q', \sigma)$. Since $\mathfrak{g}_1(q', \sigma)$, $\mathfrak{g}_2(q', \sigma)$ and

$\mathfrak{g}_3(q', \sigma)$ can be computed in polynomial-time in β and F_S has $8(\beta+2)^2+3$ pairs, we get that δ_R can be decided in polynomial-time in $\max(\lceil log(|\mathbb{d}_1|)\rceil, \lceil log(|\mathbb{d}_\alpha|)\rceil) + \beta + \mathrm{card}(\Sigma)$. Indeed, this amounts to determine the complexity of deciding δ_S. However, this question amounts to determine the complexity of deciding δ_B and $\delta_{B\to R}$. Both relations can be decided in in polynomial-time in $\max(\lceil log(|\mathbb{d}_1|)\rceil, \lceil log(|\mathbb{d}_\alpha|)\rceil) + \beta + \mathrm{card}(\Sigma)$.

- $Q_{R,\mathrm{in}}$ has a unique initial location $(q_{\mathrm{in}}, id, 8(\beta+2)^2+3, 8(\beta+2)^2+3)$, with q_{in} being the only initial location in A_S and id being the identity permutation. Since δ_S is a function and $Q_{R,\mathrm{in}}$ is a singleton, we can conclude that A_R is deterministic too.
- 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 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 $\mathrm{L}(A_R) = \mathrm{L}(A_S)$. By way of example, let us briefly explain why $\mathrm{L}(A_S) \subseteq \mathrm{L}(A_R)$. Given an accepting run ρ of A_S, we define $X_\rho \subseteq [1, 8(\beta+2)^2+3]$ such that $k \in X_\rho$ iff the set $U_{\sigma(k)}$ is visited infinitely often. Hence, for all $k \in ([1, 8(\beta+2)^2+3] \backslash X_\rho)$, $U_{\sigma(k)}$ and $L_{\sigma(k)}$ are visited finitely along ρ. Moreover, from some position $I \in \mathbb{N}$ in ρ, elements of $([1, 8(\beta+2)^2+3] \backslash X_\rho)$ always occupy the leftmost position in the permutation σ and none of its values change its place. Let $j = 8(\beta+2)^2+3-\mathrm{card}(X_\rho)$. From the position I, we have $f \geqslant j$ and one can show that $e = j$ infinitely often along ρ. 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_ρ). Hence, the run ρ' of A_R obtained from ρ by completing deterministically the three last components satisfies the Rabin pair (L'_j, U'_j). Consequently, the word accepted by ρ is also accepted by ρ'.

(5) Finally, the last (easy) stage consists in building the Rabin tree automaton

$$A_{\bigstar^{\mathrm{C}}} \stackrel{\text{def}}{=} (Q_{\bigstar^{\mathrm{C}}}, \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha), D, Q_{\bigstar^{\mathrm{C}},\mathrm{in}}, \delta_{\bigstar^{\mathrm{C}}}, F_{\bigstar^{\mathrm{C}}})$$

as follows.

- $Q_{\bigstar^{\mathrm{C}}} \stackrel{\text{def}}{=} Q_R$, $Q_{\bigstar^{\mathrm{C}},\mathrm{in}} \stackrel{\text{def}}{=} Q_{R,\mathrm{in}}$ and $F_{\bigstar^{\mathrm{C}}} \stackrel{\text{def}}{=} F_R$.
- For all $q, q_0, \ldots, q_{D-1} \in Q_{\bigstar^{\mathrm{C}}}$ and $(\mathtt{a}, \Theta) \in \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$, we have

$$(q, (\mathtt{a}, \Theta), q_0, \ldots, q_{D-1}) \in \delta_{\bigstar^{\mathrm{C}}} \stackrel{\text{def}}{\Leftrightarrow} q_0 = \cdots = q_{D-1} \text{ and } q_0 \in \delta_R(q, (\mathtt{a}, \Theta)).$$

Since A_R is deterministic, $\delta_R(q, (\mathtt{a}, \Theta))$ contains at most one location.

We have $t \in \mathrm{L}(A_{\bigstar^{\mathrm{C}}})$ iff all the branches of t are in $\mathrm{L}(A_R)$, which means precisely that t satisfies the condition $(\bigstar^{\mathrm{C}})$. Further, $A_{\bigstar^{\mathrm{C}}}$ satisfies all the size conditions in Lemma 4.12. Indeed,

- $F_{\bigstar^{\mathrm{C}}}$ has exactly $8(\beta+2)^2+3$ Rabin pairs,
- the number of locations in $Q_{\bigstar^{\mathrm{C}}}$ is in

$$2^{\mathcal{O}((4(\beta+2)^2+1)\cdot log(4(\beta+2)^2+1))} \times (8(\beta+2)^2+3)! \times (8(\beta+2)^2+3)^2,$$

  i.e. exponential in β,
- as explained earlier, δ_S can be decided in polynomial-time in $\max(\lceil log(|\mathbb{d}_1|)\rceil, \lceil log(|\mathbb{d}_\alpha|)\rceil) + \beta$ and therefore δ_R can be decided in polynomial-time in

$$\max(\lceil log(|\mathbb{d}_1|)\rceil, \lceil log(|\mathbb{d}_\alpha|)\rceil) + \beta + \mathrm{card}(\Sigma) + D. \qquad \square$$

Summarizing the developments so far, we can conclude this subsection as follows:

Lemma 4.13. $\mathrm{L}(\mathbb{A}) \neq \emptyset$ *iff* $\mathrm{L}(A_{cons(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar C}) \neq \emptyset$.

Proof. One can show that the statements below are equivalent.

(I) : $\mathrm{L}(\mathbb{A}) \neq \emptyset$.

(II): There is a symbolic tree $t : [0, D-1]^* \to \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ in $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$ that is satisfiable.

(III): There is a symbolic tree $t : [0, D-1]^* \to \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ in $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar \mathrm{C}})$.

The equivalence between (I) and (II) is by Lemma 4.6. That condition (II) implies (III) follows from the fact that for every satisfiable symbolic tree t (not necessarily regular), t satisfies the condition ($\bigstar^{\mathrm{C}}$) (Lemma 4.9), $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$ contains all the satisfiable symbolic trees (Lemma 4.4 and Lemma 4.5) and $\mathrm{L}(A_{\bigstar \mathrm{C}})$ is equal to the set of symbolic trees satisfying the condition ($\bigstar^{\mathrm{C}}$). Hence, $t \in \mathrm{L}(A_{\mathrm{cons}(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar \mathrm{C}})$. That condition (III) implies (II) follows from the fact that if $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar \mathrm{C}})$ is non-empty, then $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar \mathrm{C}})$ is regular and therefore contains a regular $\mathbb{A}$-consistent symbolic tree t (see e.g. [Rab69] and [Tho90, Section 6.3] for the existence of regular trees) and by Proposition 4.10, t is satisfiable. By Lemma 4.6, since t is satisfiable and $t \in \mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$, we get $\mathrm{L}(\mathbb{A}) \neq \emptyset$. □

4.3. **ExpTime Upper Bound for TCAs.** Lemma 4.13 justifies why deciding the nonemptiness of $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar \mathrm{C}})$ is crucial. Fortunately, regular tree languages are closed under intersection. However, assuming that a Rabin tree automaton A satisfies $\mathrm{L}(A) = \mathrm{L}(A_{\mathrm{cons}(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar \mathrm{C}})$, we need to guarantee that the construction of A does not lead to any complexity blow up. This is the purpose of Lemma 4.14 below. An exponential blow-up may have drastic consequences on the forthcoming complexity analysis. In the proof of Lemma 4.14, we propose a construction that is not polynomial but it only performs an exponential blow-up on the number of locations, which shall be fine for our purpose.

Lemma 4.14. *There is a Rabin tree automaton A such that*

(I) : $\mathrm{L}(A) = \mathrm{L}(A_{cons(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar C})$,

(II): *the number of Rabin pairs is polynomial in β,*

(III): *the number of locations is in* $(\mathbb{d}_\alpha - \mathbb{d}_1)^{2\beta} \times \mathrm{card}(Q) \times 2^{\mathcal{O}(P(\beta))}$ *for some polynomial P,*

(IV): *the cardinality of the transition relation is in* $((\mathbb{d}_\alpha - \mathbb{d}_1)^{2\beta} \times \mathrm{card}(Q) \times 2^{\mathcal{O}(P'(\beta))})^{D+2} \times \mathrm{card}(\Sigma)$ *for some polynomial P',*

(V): *the transition relation can be decided in polynomial-time in* $\mathrm{card}(\delta) + \beta + \mathrm{card}(\Sigma) + D + \mathtt{MCS}(\mathbb{A})$.

Proof. The proof is divided into 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). The construction is a particular case of the one for the proof of Lemma 3.1, but here with only two input Rabin tree automata and no constraints. In Part (II), we apply the general construction from Part (I) to $A_{\mathrm{cons}(\mathbb{A})}$ and $A_{\bigstar \mathrm{C}}$ and perform a quantitative analysis.

(I) For $i = 1, 2$, let $A_i = (Q_i, \Sigma, D, Q_{i,\mathrm{in}}, \delta_i, F_i)$ with $F_i = (L_i^j, U_i^j)_{j \in [1, N_i]}$, that is N_i Rabin pairs, be a Rabin tree automaton. Let us build a Rabin tree automaton $A = (Q, \Sigma, D, Q_{\mathrm{in}}, \delta, F)$ such that $\mathrm{L}(A) = \mathrm{L}(A_1) \cap \mathrm{L}(A_2)$.

- $Q \stackrel{\text{def}}{=} Q_1 \times Q_2 \times [0,3]^{[1,N_1]\times[1,N_2]}$. The elements in Q are of the form $(q_1, q_2, \mathfrak{f})$ with $\mathfrak{f} : [1, N_1] \times [1, N_2] \to [0, 3]$.
- The tuple $((q_1, q_2, \mathfrak{f}), \mathtt{a}, (q_1^0, q_2^0, \mathfrak{f}^0), \ldots, (q_1^{D-1}, q_2^{D-1}, \mathfrak{f}^{D-1}))$ belongs to δ iff the conditions below hold.
  (1) For $i = 1, 2$, we have $(q_i, \mathtt{a}, q_i^0, \ldots, q_i^{D-1}) \in \delta_i$. The two first components in elements from Q behave as in A_1 and A_2, respectively.
  (2) For all $(i, j) \in [1, N_1] \times [1, N_2]$, the following conditions hold.
    (a) If $\mathfrak{f}(i, j)$ is odd, then for all $k \in [0, D-1]$, we have $\mathfrak{f}^k(i, j) = (\mathfrak{f}(i, j) + 1) \mod 4$. Odd values in $[0, 3]$ are unstable and are replaced at the next step by the successor value (modulo 4).
    (b) For all $k \in [0, D-1]$, if $\mathfrak{f}(i, j) = 0$ and $q_1^k \in L_1^i$, then $\mathfrak{f}^k(i, j) = 1$. Hence, when the $(i, j)^{\text{th}}$ component of $\mathfrak{f}$ 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 $\mathfrak{f}(i, j) = 0$ and $q_1^k \notin L_1^i$, then $\mathfrak{f}^k(i, j) = 0$ (not yet the right moment to modify the $(i, j)^{\text{th}}$ component).
    (d) For all $k \in [0, D-1]$, if $\mathfrak{f}(i, j) = 2$ and $q_2^k \in L_2^j$, then $\mathfrak{f}^k(i, j) = 3$. Hence, when the $(i, j)^{\text{th}}$ component of $\mathfrak{f}$ 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 $\mathfrak{f}(i, j) = 2$ and $q_2^k \notin L_2^j$, then $\mathfrak{f}^k(i, j) = 2$ (not yet the right moment to modify the $(i, j)^{\text{th}}$ component).

  At this stage, it is worth noting that each $\mathfrak{f}^k$ 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 δ can be decided in the sum of the time-complexity to decide δ_1 and δ_2 respectively, plus polynomial-time in $N_1 \times N_2$.
- $Q_{\text{in}} \stackrel{\text{def}}{=} Q_{1,\text{in}} \times Q_{2,\text{in}} \times \{\mathfrak{f}_0\}$, where $\mathfrak{f}_0$ is the unique map that takes always the value zero.
- The set of Rabin pairs in F contains exactly the pairs (L, U) for which there is $(i, j) \in [1, N_1] \times [1, N_2]$ such that

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

  Because the odd values are unstable, if a location in L is visited infinitely along a branch of a run for 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 $(i, j)^{\text{th}}$ 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 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, F contains at most $N_1 \times N_2$ pairs.

We claim that $\mathrm{L}(A) = \mathrm{L}(A_1) \cap \mathrm{L}(A_2)$. We omit the proof here as the proof of Lemma 3.1 generalises it.
(II) Let us analyse the size of the components in $A_{\text{cons}(\mathbb{A})}$ and $A_{\bigstar \mathrm{C}}$, which provides bounds for A such that $\mathrm{L}(A) = \mathrm{L}(A_{\bigstar \mathrm{C}}) \cap \mathrm{L}(A_{\text{cons}(\mathbb{A})})$ following the above construction. The automaton $A_{\text{cons}(\mathbb{A})}$ can be viewed as a Rabin tree automaton with a single pair, typically $(F', \varnothing)$, where F' is the set of accepting states of the Büchi tree automaton $A_{\text{cons}(\mathbb{A})}$.

- $A_{\mathrm{cons}(\mathbb{A})}$ has a single Rabin pair, $A_{\bigstar\mathrm{C}}$ has a number of Rabin pairs bounded by $8(\beta+2)^2+3$, so A has a number of Rabin pairs bounded by $8(\beta+2)^2+3$.
- The locations in $A_{\mathrm{cons}(\mathbb{A})}$ are from $\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha) \times Q$, the number of locations in $A_{\bigstar\mathrm{C}}$ is in $\mathcal{O}(2^{P^\dagger(\beta)})$ for some polynomial $P^\dagger$ (see Lemma 4.12). Therefore, based on the above construction for intersection, the number of locations in A is in

$$\mathrm{card}(\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha) \times Q) \times 2^{\mathcal{O}(P^\dagger(\beta))} \times 4^{8(\beta+2)^2+3},$$

  which is in $(\mathbb{d}_\alpha - \mathbb{d}_1)^{2\beta} \times \mathrm{card}(Q) \times 2^{\mathcal{O}(P(\beta))}$ for some polynomial $P(\cdot)$.
- The transition relation for $A_{\bigstar\mathrm{C}}$ can be decided in polynomial-time in

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

  (by Lemma 4.12), the transition relation for $A_{\mathrm{cons}(\mathbb{A})}$ can be decided in polynomial-time in $\mathrm{card}(\delta) + \beta + \mathrm{card}(\Sigma) + D + \mathtt{MCS}(\mathbb{A})$. Note also that $\mathrm{card}(\delta^\dagger)$ (where $\delta^\dagger$ is the transition relation of A) is in

$$(\mathrm{card}(\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)) \times \mathrm{card}(Q) \times 2^{\mathcal{O}(P(\beta))})^{D+1} \times \mathrm{card}(\Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)),$$

  since the finite alphabet of A is $\Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$. Consequently, $\mathrm{card}(\delta^\dagger)$ is in

$$\big((\mathbb{d}_\alpha - \mathbb{d}_1)^{2\beta} \times \mathrm{card}(Q) \times 2^{\mathcal{O}(P'(\beta))}\big)^{D+2} \times \mathrm{card}(\Sigma),$$

  for some polynomial P'. Moreover, the product of the number of Rabin pairs is polynomial in β. Therefore, the transition relation can be decided in polynomial-time in $\big(\mathrm{card}(\delta) + \beta + \mathrm{card}(\Sigma) + D + \mathtt{MCS}(\mathbb{A})\big)$. □

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 can be solved in time $(\mathrm{card}(\delta) \times \gamma \times N)^{\mathcal{O}(N)}$ (by scrutiny of the proof of [EJ00, Theorem 4.1], page 144) where N is the number of Rabin pairs, δ is the transition relation and γ is the time to decide δ (this depends on how the locations and the transitions are encoded). For instance, in our case, γ may depend on parameters related to $\mathbb{A}$ and in Lemma 4.15 below, γ takes the value $\mathrm{card}(\delta) + \beta + \mathrm{card}(\Sigma) + D + \mathtt{MCS}(\mathbb{A})$ (by Lemma 4.14). When a Rabin tree automaton is provided in extension, $\mathrm{card}(\delta) \times \gamma$ is polynomial in its size and usually γ is omitted. Hence the following result.

Lemma 4.15. *The nonemptiness problem for TCA can be solved in time in*

$$R_1\big(\mathrm{card}(Q) \times \mathrm{card}(\delta) \times \mathtt{MCS}(\mathbb{A}) \times \mathrm{card}(\Sigma) \times R_2(\beta)\big)^{\mathcal{O}(R_2(\beta) \times R_3(D))},$$

for some polynomials R_1, R_2 and R_3.

Proof. (Sketch) In the above expression $(\mathrm{card}(\delta) \times \gamma \times N)^{\mathcal{O}(N)}$, let us see how this is instantiated for A from Lemma 4.14.

- N is in $R'(\beta)$ for some polynomial R' (Lemma 4.14(II)).
- $\mathrm{card}(\delta)$ is in $((\mathbb{d}_\alpha - \mathbb{d}_1)^{2\beta} \times \mathrm{card}(Q) \times 2^{\mathcal{O}(P'(\beta))})^{D+2} \times \mathrm{card}(\Sigma)$ for some polynomial P' (Lemma 4.14(IV)).
- γ is in $R''(\mathrm{card}(\delta) + \beta + \mathrm{card}(\Sigma) + D + \mathtt{MCS}(\mathbb{A}))$ for some polynomial R'' (Lemma 4.14(V)).

This allows to get the bound from the statement. It is essential in the calculation that the exponent is only polynomial in the size of the input constraint automaton $\mathbb{A}$, which is the case as the exponent is polynomial in $\beta + D$. □

Assuming that the size of the TCA $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$, written $\mathtt{size}(\mathbb{A})$, is polynomial in $\text{card}(Q) + \text{card}(\delta) + D + \beta + \mathtt{MCS}(\mathbb{A})$ (which makes sense for a reasonably succinct encoding), from the computation of the bound in Lemma 4.15, the nonemptiness of $\text{L}(\mathbb{A})$ can be checked in time $R(\mathtt{size}(\mathbb{A}))^{\mathcal{O}(R'(\beta+D))}$ for some polynomials R and R'. The ExpTime upper bound of the nonemptiness problem for TCA is now a consequence of the above complexity expression and using the fact that the transitions in the product Rabin tree automaton between $A_{\text{cons}(\mathbb{A})}$ and $A_{\bigstar\text{C}}$ can be decided in polynomial-time.

Theorem 4.16. *Nonemptiness problem for tree constraint automata is* ExpTime*-complete.*

We have seen that the ExpTime-hardness holds as soon as $D = 2$. The case $D = 1$ differs slightly.

Theorem 4.17. *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 Appendix B.1 by reduction from the halting problem for deterministic Turing machines running in polynomial space. Actually, PSpace-hardness is a corollary of the ExpTime-hardness proof from Appendix B.1, since the construction also works for non-deterministic Turing machines and then the runs are sequences instead of trees. Concerning the PSpace-membership, we need to check the nonemptiness of $\text{L}(A_R) \cap \text{L}(A_{\text{cons}(\mathbb{A})})$ with the Rabin word automaton A_R from the proof of Lemma 4.12 and $A_{\text{cons}(\mathbb{A})}$ is already a Büchi automaton. Not only A_R can be transformed into an equivalent Büchi automaton A 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 A and $A_{\text{cons}(\mathbb{A})}$ can be performed in PSpace. Indeed, the number of locations of the product is only exponential in the size of the input word constraint automaton and the transition relation can be also decided in polynomial space. □

The Rabin word automaton A_R captures therefore the condition $C_{\mathbb{Z}}$ from [DD07, Section 6] (see also the 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 with constraints from the concrete domain $\mathbb{Z}$, see Section 2.2 and Section 6.1), proposing therefore an alternative proof to [DG08, Theorem 1] and [ST11, Theorem 16] for the concrete domain $\mathbb{Z}$.

4.4. **Rabin Tree Constraint Automata.** In this section, we show that the nonemptiness problem for Rabin tree constraint automata is also in ExpTime. This will be key to characterize the complexity of SAT(CTL*($\mathbb{Z}$)). Given an Rabin TCA $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\text{in}}, \delta, F)$, the definition of symbolic trees respecting $\mathbb{A}$ is updated so that its uses the acceptance condition F. From the Rabin TCA $\mathbb{A}$, we can define a Rabin tree automaton $A'_{\text{cons}(\mathbb{A})}$ (instead of a Büchi tree automaton with a Büchi TCA) such that the acceptance F' is equal to

$$\{(\text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha) \times L, \text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha) \times U) \mid (L, U) \in F\}.$$

Similarly to Lemma 4.6, one can show that $\text{L}(\mathbb{A}) \neq \varnothing$ iff there is a symbolic tree $t \in \text{L}(A'_{\text{cons}(\mathbb{A})})$ that is satisfiable. Moreover, we can take advantage of $A_{\bigstar\text{C}}$ (the same as in the proof of Lemma 4.12) so that $\text{L}(\mathbb{A}) \neq \varnothing$ iff $\text{L}(A'_{\text{cons}(\mathbb{A})}) \cap \text{L}(A_{\bigstar\text{C}})$ is non-empty (same arguments as for the proof of Lemma 4.13). It remains to determine how much it costs to test nonemptiness of

$\mathrm{L}(A'_{\mathrm{cons}(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar \mathrm{C}})$. We might expect a complexity jump compared to the case with Büchi TCA (but this is not the case), because 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 provides quantitative analysis about components of $\mathbb{A}$, which is a variant of Lemma 4.14, more particularly by considering the value $\mathrm{card}(F)$ in the analysis. For TCA with Büchi acceptance condition, the value for $\mathrm{card}(F)$ is equal to one.

Lemma 4.18. *There is a Rabin tree automaton A such that*

(I) : $\mathrm{L}(A) = \mathrm{L}(A'_{cons(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar C})$,

(II): *the number of Rabin pairs is polynomial in $\beta + \mathrm{card}(F)$, where $\mathrm{card}(F)$ is the number of Rabin pairs in $\mathbb{A}$,*

(III): *the number of locations is in $(\mathbb{d}_\alpha - \mathbb{d}_1)^{2\beta} \times \mathrm{card}(Q) \times 2^{\mathcal{O}(P(\beta+\mathrm{card}(F)))}$ for some polynomial P,*

(IV): *the cardinality of the transition relation is in*

$$((\mathbb{d}_\alpha - \mathbb{d}_1)^{2\beta} \times \mathrm{card}(Q) \times 2^{\mathcal{O}(P'(\beta+\mathrm{card}(F)))})^{D+2} \times \mathrm{card}(\Sigma)$$

for some polynomial P',

(V): *the transition relation can be decided in polynomial-time in $\mathrm{card}(\delta) + \beta + \mathrm{card}(\Sigma) + D + \mathtt{MCS}(\mathbb{A})$.*

The proof of Lemma 4.18 is similar to the proof of Lemma 4.14. Moreover, as for Lemma 4.15, we can conclude that the nonemptiness problem for Rabin tree constraint automata can be solved in time in

$$R_1\big(\mathrm{card}(Q) \times \mathrm{card}(\delta) \times \mathtt{MCS}(\mathbb{A}) \times \mathrm{card}(\Sigma) \times R_2(\beta + \mathrm{card}(F))\big)^{\mathcal{O}(R_2(\beta+\mathrm{card}(F))\times R_3(D))}$$

for some polynomials R_1, R_2 and R_3. Theorem 4.19 is one of the main results of the paper.

Theorem 4.19. *The nonemptiness problem for Rabin tree constraint automata is* ExpTime-*complete.*

5. Tree Constraint Automata for CTL($\mathbb{Z}$)

Below, we harvest the first result from what is achieved in the previous section: SAT(CTL($\mathbb{Z}$)) is in ExpTime. We follow the automata-based approach and – after proving a refined version of the tree model property for CTL($\mathbb{Z}$) – the key step is to translate CTL($\mathbb{Z}$) formulae into equivalent TCA (Theorem 5.3). As usual, the tree model property means that we can restrict ourselves to tree Kripke structures to determine the satisfiability status of CTL($\mathbb{Z}$) formulae. In preparation, we first show how CTL($\mathbb{Z}$) formulae can be put into simple form (Proposition 5.1), and that the tree model property for CTL($\mathbb{Z}$) can follow a strict discipline (Proposition 5.2).

A CTL($\mathbb{Z}$) formula is in *simple form* iff it is in negation normal form and terms are restricted to those in $\mathrm{T}_V^{\leqslant 1}$. Preprocessing CTL($\mathbb{Z}$) formulae to obtain simple formulae is computationally harmless, but will simplify the translation of CTL($\mathbb{Z}$) formulae to tree constraint automata.

Proposition 5.1. *For every CTL($\mathbb{Z}$) formula ϕ, one can construct in polynomial-time in the size of ϕ a CTL($\mathbb{Z}$) formula ϕ' in simple form such that ϕ is satisfiable iff ϕ' is satisfiable.*

The proof of Proposition 5.1 can be found in Appendix C.1 and it is made of two standard arguments. First, it establishes a tree model property (in the standard way using unfoldings of Kripke models). Then it uses a renaming technique (see e.g. [Sco62]) to flatten the constraints that introduces additional variables for values i steps ahead.

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

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

The cardinality of sub(ϕ) is at most twice the number of subformulae of ϕ. Given $X \subseteq \mathrm{sub}(\phi)$, 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 $\mathsf{E}\phi_1\mathsf{U}\phi_2 \in X$, then $\phi_2 \in X$ or $\{\phi_1, \mathsf{EXE}\phi_1\mathsf{U}\phi_2\} \subseteq X$.
- If $\mathsf{A}\phi_1\mathsf{U}\phi_2 \in X$, then $\phi_2 \in X$ or $\{\phi_1, \mathsf{AXA}\phi_1\mathsf{U}\phi_2\} \subseteq X$.
- If $\mathsf{E}\phi_1\mathsf{R}\phi_2 \in X$, then $\phi_2 \in X$ and $\{\phi_1, \mathsf{EXE}\phi_1\mathsf{R}\phi_2\} \cap X \neq \emptyset$.
- If $\mathsf{A}\phi_1\mathsf{R}\phi_2 \in X$, then $\phi_2 \in X$ and $\{\phi_1, \mathsf{AXA}\phi_1\mathsf{R}\phi_2\} \cap X \neq \emptyset$.

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

Let ϕ be a CTL($\mathbb{Z}$) formula in simple form built over the variables $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$ for some $\beta \geqslant 1$, and set $D = \mathrm{card}(\mathrm{sub}_{\mathsf{EX}}(\phi)) + \mathrm{card}(\mathrm{sub}_{\mathsf{E}}(\phi))$. A *direction map* ι *for* ϕ is a bijection

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

We say that a tree model $\mathbb{t} : [0, D]^* \to \mathbb{Z}^\beta$ of ϕ (i.e. $\mathbb{t}, \varepsilon \models \phi$) *obeys a direction map* ι if for all nodes $\mathbf{n} \in [0, D]^*$, the following three conditions hold.

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

(2) For every $\mathsf{E}\phi_1\mathsf{U}\phi_2 \in \mathrm{sub}_{\mathsf{EU}}(\phi)$, if $\mathbb{t}, \mathbf{n} \models \mathsf{E}\phi_1\mathsf{U}\phi_2$, then there exists some $k \geqslant 0$ such that $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_2$ and $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_1$ for all $0 \leqslant i < k$, where $j = \iota(\mathsf{EXE}\phi_1\mathsf{U}\phi_2)$. In other words, the path $\mathbf{n}, \mathbf{n} \cdot j, \mathbf{n} \cdot j^2 \ldots \mathbf{n} \cdot j^k$ satisfies $\phi_1\mathsf{U}\phi_2$.

(3) For every $\mathsf{E}\ \Theta \in \mathrm{sub}_{\mathsf{E}}(\phi)$, if $\mathbb{t}, \mathbf{n} \models \mathsf{E}\ \Theta$, then $\mathbb{Z} \models \Theta(\mathbb{t}(\mathbf{n}), \mathbb{t}(\mathbf{n} \cdot j))$ with $\iota(\mathsf{E}\ \Theta) = j$.

Again, here, $\mathbb{Z} \models \Theta(\boldsymbol{z}, \boldsymbol{z}')$ with $\boldsymbol{z}, \boldsymbol{z}'$ is a shortcut for $[\vec{\mathtt{x}} \leftarrow \boldsymbol{z}, \vec{\mathtt{x}'} \leftarrow \boldsymbol{z}'] \models \Theta$ where $[\vec{\mathtt{x}} \leftarrow \boldsymbol{z}, \vec{\mathtt{x}'} \leftarrow \boldsymbol{z}']$ is a valuation $\mathfrak{v}$ on the variables $\{\mathtt{x}_j, \mathtt{x}'_j \mid j \in [1, \beta]\}$ with $\mathfrak{v}(\mathtt{x}_j) = \boldsymbol{z}(j)$ and $\mathfrak{v}(\mathtt{x}'_j) = \boldsymbol{z}'(j)$ for all $j \in [1, \beta]$. A tree model $\mathbb{t}$ obeying a direction map ι follows a discipline to verify the satisfaction of formulae involving existential quantifications over paths, namely dedicated directions are reserved for specific formulae.

Proposition 5.2. *Let ϕ be a* CTL($\mathbb{Z}$)*-formula in simple form and ι be a direction map for ϕ. Then, ϕ is satisfiable iff ϕ has a tree model with branching width equal to* $\mathrm{card}(sub_{\mathsf{EX}}(\phi)) + \mathrm{card}(sub_{\mathsf{E}}(\phi)) + 1$ *and that obeys ι.*

The proof of Proposition 5.2 can be found in Appendix C.2. It amounts to structure the paths from a state satisfying ϕ in a tree-like fashion while obeying the direction map ι. Let us now turn to the key step and explain how to obtain a TCA $\mathbb{B}_\phi$ from a CTL($\mathbb{Z}$) formula ϕ in simple form. Observe that we take very good care of the quantitative properties of $\mathbb{B}_\phi$: in general, the size of $\mathbb{B}_\phi$ is exponential in the size of the input formula ϕ. In Section 4, we proved that the nonemptiness problem for TCA is ExpTime-complete. In order to get ExpTime-completeness of SAT(CTL($\mathbb{Z}$)) and therefore avoiding the *double* exponential blow-up, we refine the analysis of the size of $\mathbb{B}_\phi$ by looking at the size of its different components, revealing which components are responsible for the exponential blow-up.

Theorem 5.3. *Let ϕ be a CTL($\mathbb{Z}$) formula in simple form. There exists a TCA $\mathbb{B}_\phi$ such that ϕ is satisfiable iff $\mathrm{L}(\mathbb{B}_\phi) \neq \emptyset$, and satisfying the properties below.*

(I) : *The degree D and the number of variables β is bounded above by $\mathtt{size}(\phi)$.*
(II): *The alphabet Σ is a singleton.*
(III): *The number of locations is bounded by $(D \times 2^{\mathtt{size}(\phi)}) \times (\mathtt{size}(\phi) + 1)$.*
(IV): *The number of transitions is in $2^{\mathcal{O}(P(\mathtt{size}(\phi)))}$ for some polynomial $P(\cdot)$.*
(V): *The maximal size of a constraint in transitions is quadratic in $\mathtt{size}(\phi)$.*

Proof. Let ϕ be a CTL($\mathbb{Z}$) formula in simple form built over $\mathtt{x}_1, \ldots, \mathtt{x}_\beta, \mathsf{X}\mathtt{x}_1, \ldots, \mathsf{X}\mathtt{x}_\beta$. Let $D = \mathrm{card}(\mathrm{sub}_{\mathsf{EX}}(\phi)) + \mathrm{card}(\mathrm{sub}_{\mathsf{E}}(\phi))$ and $\iota : \mathrm{sub}_{\mathsf{EX}}(\phi) \cup \mathrm{sub}_{\mathsf{E}}(\phi) \to [1, D]$ be a direction map. We start building a *generalised Büchi* TCA (see Section 3) $\mathbb{A}_\phi = (Q, \Sigma, D+1, \beta, Q_{\mathrm{in}}, \delta, F)$ such that ϕ is satisfiable iff $\mathrm{L}(\mathbb{A}_\phi) \neq \emptyset$. The automaton $\mathbb{A}_\phi$ accepts infinite trees of the form $\mathbb{t} : [0, D]^* \to \Sigma \times \mathbb{Z}^\beta$. Let us define $\mathbb{A}_\phi$ formally.

- $\Sigma \stackrel{\text{def}}{=} \{\dagger\}$ ($\dagger$ is an arbitrary letter).
- Q is the subset of $[0, D] \times \mathcal{P}(\mathrm{sub}(\phi))$ such that (i, X) belongs to Q only if X is propositionally consistent. The first argument records the direction (from which the nodes are reached), as indicated by ι.
- $Q_{\mathrm{in}} \stackrel{\text{def}}{=} \{(0, X) \in Q \mid \phi \in X\}$.
- The transition relation δ is made of tuples of the form
$$((i, X), \dagger, (\Theta_0, (0, X_0)), \ldots, (\Theta_D, (D, X_D)))$$
verifying the conditions below.
  (1) For all $\mathsf{EX}\psi \in X$, we have $\psi \in X_{\iota(\mathsf{EX}\psi)}$.
  (2) For all $\mathsf{AX}\psi \in X$ and $j \in [0, D]$, we have $\psi \in X_j$.
  (3) For all $j \in [0, D]$, if there is $\mathsf{E}\ \Theta \in X$ such that $\iota(\mathsf{E}\ \Theta) = j$, then
$$\Theta_j \stackrel{\text{def}}{=} (\bigwedge_{\mathsf{A}\ \Theta' \in X} \Theta') \wedge \Theta \quad \text{otherwise,} \quad \Theta_j \stackrel{\text{def}}{=} \bigwedge_{\mathsf{A}\ \Theta' \in X} \Theta'.$$
- F is made of two types of sets, those parameterised by some element in $\mathrm{sub}_{\mathsf{EU}}(\phi)$ and those parameterised by some element in $\mathrm{sub}_{\mathsf{AU}}(\phi)$. Recall that F is a generalised Büchi condition made of subsets of Q: for all branches of the accepted trees, for all sets F in F, some location in F must occur infinitely often. The subformulae whose outermost connective is either ER or AR do not impose additional acceptance conditions. For each $\mathsf{E}\psi_1\mathsf{U}\psi_2 \in \mathrm{sub}_{\mathsf{EU}}(\phi)$, the set $F_{\mathsf{E}\psi_1\mathsf{U}\psi_2}$ defined below belongs to F:
$$F_{\mathsf{E}\psi_1\mathsf{U}\psi_2} \stackrel{\text{def}}{=} \{(i, X) \in Q \mid i \neq \iota(\mathsf{EXE}\psi_1\mathsf{U}\psi_2) \text{ or } \psi_2 \in X \text{ or } \mathsf{E}\psi_1\mathsf{U}\psi_2 \notin X\}.$$
Hence, if $i = \iota(\mathsf{EXE}\psi_1\mathsf{U}\psi_2)$, then along a branch $\mathbf{n} \cdot i^\omega$, the satisfaction of $\psi_1\mathsf{U}\psi_2$ cannot be postponed forever. This is a standard encoding to translate LTL formulae into Büchi automata, see e.g. [VW94]. Moreover, for each $\mathsf{A}\psi_1\mathsf{U}\psi_2 \in \mathrm{sub}_{\mathsf{AU}}(\phi)$, the set $F_{\mathsf{A}\psi_1\mathsf{U}\psi_2}$ belongs to F:
$$F_{\mathsf{A}\psi_1\mathsf{U}\psi_2} \stackrel{\text{def}}{=} \{(i, X) \in Q \mid \psi_2 \in X \text{ or } \mathsf{A}\psi_1\mathsf{U}\psi_2 \notin X\}.$$
By contrast to the behaviours induced by the sets $F_{\mathsf{E}\psi_1\mathsf{U}\psi_2}$'s, on any branch from a node $\mathbf{n}$ satisfying $\mathsf{A}\psi_1\mathsf{U}\psi_2$, the satisfaction of $\psi_1\mathsf{U}\psi_2$ cannot be postponed forever.

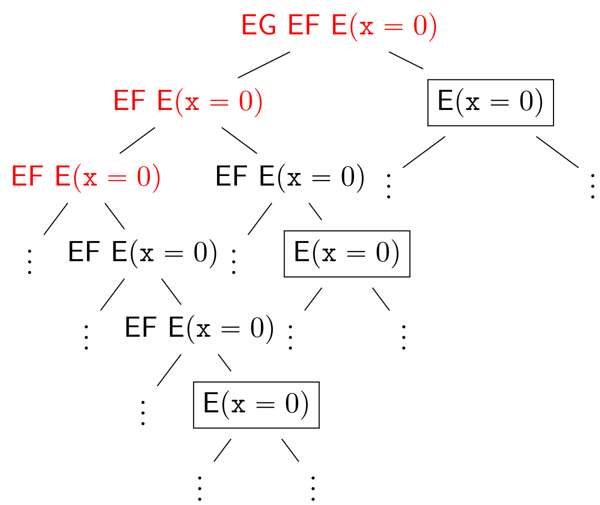


Figure 6. Tree-like Kripke structure for $\mathsf{EG}\ \mathsf{EF}\ \mathsf{E}(\mathtt{x} = 0)$

The correctness of $\mathbb{A}_\phi$ is stated below.

Lemma 5.4. *ϕ is satisfiable iff $\mathrm{L}(\mathbb{A}_\phi) \neq \varnothing$.*

The proof of Lemma 5.4 can be found in Appendix C.3. It follows a standard pattern but we need to handle the constraints on data values, and we take advantage of the direction map ι in order to guide the satisfaction of the qualitative constraints related to the temporal connective U.

Finally, let $\mathbb{B}_\phi$ be the TCA obtained from $\mathbb{A}_\phi$ as explained in Section 3, and recall that the size of $\mathbb{B}_\phi$ is polynomial in the size of $\mathbb{A}_\phi$. It is now easy to check that all quantitative properties given in the claim of the theorem are satisfied. □

The construction of the generalised Büchi TCA $\mathbb{A}_\phi$ 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 CTL($\mathbb{Z}$) extends CTL by adding constraints between data values. More importantly, [VW08, Section 5.2] uses another tree automaton model and $F_{\mathsf{E}\psi_1 \mathsf{U} \psi_2}$ and $F_{\mathsf{A}\psi_1 \mathsf{U} \psi_2}$ herein differ from [VW08, page 702] by the use of the direction map that we believe necessary here.

Example 5.5. Herein, we illustrate the use of direction maps. First, observe that direction maps are helpful for the satisfaction of EX-formulae in the definition of $\mathbb{A}_\phi$ (see the proof of Theorem 5.3) and each set $F_{\mathsf{E}\psi_1 \mathsf{U} \psi_2}$ of accepting locations defined by $\{(i, X) \in Q \mid i \neq \iota(\mathsf{EXE}\psi_1 \mathsf{U} \psi_2)$ or $\psi_2 \in X$ or $\mathsf{E}\psi_1 \mathsf{U} \psi_2 \notin X\}$ involves the direction map ι.

Let us consider the formula $\phi = \mathsf{EG}\ \mathsf{EF}\ \mathsf{E}(\mathtt{x} = 0)$ that states the existence of a path π along which at every position i there is a path π_i leading to a position k_i such that $\mathtt{x}$ is equal to zero. Note that $\mathsf{EF}\ \mathsf{E}(\mathtt{x} = 0)$ is a formula of the form $\mathsf{E}\psi_1 \mathsf{U} \psi_2$ and therefore would generate a set $F_{\mathsf{EF}\ \mathsf{E}(\mathtt{x}=0)}$ of accepting locations. Let us explain why it is not adequate to define $F_{\mathsf{EF}\ \mathsf{E}(\mathtt{x}=0)}$ by $\{(i, X) \in Q \mid \mathsf{E}(\mathtt{x} = 0) \in X$ or $\mathsf{EF}\ \mathsf{E}(\mathtt{x} = 0) \notin X\}$, i.e. without the use of the direction map ι.

In Figure 6, we present a tree-like infinite Kripke structure satisfying ϕ on its root node. Each node is labelled by a formula that holds on it, witnessing the satisfaction of ϕ

at the root. Framed nodes are the only ones satisfying the subformula $\mathsf{E}(\mathtt{x} = 0)$. For the satisfaction of ϕ at the root, the above mentioned path π can be the leftmost branch, and each π_i is the rightmost branch from $\pi(i)$ with $k_i = i + 1$. Observe that for all nodes along π (in red in a coloured version of the document), the formula $\mathsf{E}(\mathtt{x} = 0)$ does not hold and the formula $\mathsf{EF}\ \mathsf{E}(\mathtt{x} = 0)$ holds. However, infinitely often on the path π, a node on π is the leftmost child of its parent node (and therefore it is not the rightmost child of its parent node). This is fine as soon as the satisfaction of $\mathsf{EF}\ \mathsf{E}(\mathtt{x} = 0)$ is witnessed along a branch (which is not the leftmost branch, actually the rightmost branch in this example).

Hence, considering the structure in Figure 6 as a run for $\mathbb{A}_\phi$ in which instead of labelling the nodes by pairs (i, X), we have just represented a distinguished subformula, we can check that this leads to an accepting run assuming that the direction map ι is such that $\iota(\mathsf{EX}\ \mathsf{EF}\ \mathsf{E}(\mathtt{x} = 0))$ defined as the greatest direction, usually represented as the rightmost direction. Along the leftmost branch, no element in $\{(i, X) \in Q \mid \mathsf{E}(\mathtt{x} = 0) \in X$ or $\mathsf{EF}\ \mathsf{E}(\mathtt{x} = 0) \notin X\}$ is visited infinitely often, whereas it does for $F_{\mathsf{EF}\ \mathsf{E}(\mathtt{x}=0)}$ involving the direction map ι. Indeed, all the nodes along the leftmost branch have indices different from $\iota(\mathsf{EX}\ \mathsf{EF}\ \mathsf{E}(\mathtt{x} = 0))$.

The following theorem is one of our main results.

Theorem 5.6. *The satisfiability problem for* CTL($\mathbb{Z}$) *is* ExpTime*-complete.*

Proof. ExpTime-hardness is inherited from CTL. For ExpTime-membership, let ψ be a CTL($\mathbb{Z}$) formula. First, we construct in polynomial-time from ψ a formula ϕ in *simple form* such that ψ is satisfiable iff ϕ is satisfiable (see Proposition 5.1). Second, using Theorem 5.3, we construct from ϕ the TCA $\mathbb{B}_\phi = (Q, \Sigma, D, \beta, Q_{\rm in}, \delta, F)$ such that ϕ is satisfiable iff $\mathrm{L}(\mathbb{B}_\phi) \neq \varnothing$ and $\mathbb{B}_\phi$ satisfying the following quantitative properties:

- the degree D and the number of variables β are bounded by $\mathtt{size}(\phi)$,
- $\mathrm{card}(Q)$ is bounded by $(D \times 2^{\mathtt{size}(\phi)}) \times (\mathtt{size}(\phi) + 1)$,
- $\mathrm{card}(\delta)$ is in $2^{\mathcal{O}(P(\mathtt{size}(\phi)))}$ for some polynomial $P(\cdot)$,
- $\mathrm{card}(\Sigma) = 1$, and $\mathtt{MCS}(\mathbb{B}_\phi)$ is quadratic in $\mathtt{size}(\phi)$.

By Lemma 4.15, the nonemptiness problem for TCA can be solved in time

$$R_1\big(\mathrm{card}(Q) \times \mathrm{card}(\delta) \times \mathtt{MCS}(\mathbb{B}_\phi) \times \mathrm{card}(\Sigma) \times R_2(\beta)\big)^{\mathcal{O}(R_2(\beta) \times R_3(D))}.$$

Since the transition relations of the automata $A_{\mathrm{cons}(\mathbb{B}_\phi)}$ and $A_{\bigstar\mathrm{C}}$ can be built in polynomial-time, we get that nonemptiness of $\mathrm{L}(\mathbb{B}_\phi)$ can be solved in exponential-time. □

Results for other domains. Let $\mathbb{N}$ be the concrete domain $(\mathbb{N}, <, =, (=_{\mathbbm{d}})_{\mathbbm{d} \in \mathbb{N}})$ for which we can also show that nonemptiness of TCA with constraints interpreted on $\mathbb{N}$ has the same complexity as for TCA with constraints interpreted on $\mathbb{Z}$. Indeed, the nonemptiness problem for TCA on $\mathbb{N}$ can be easily reduced in polynomial-time to the nonemptiness problem for TCA on $\mathbb{Z}$. Let CTL($\mathbb{N}$) be the variant of CTL($\mathbb{Z}$) with constraints interpreted on $\mathbb{N}$. As a corollary, SAT(CTL($\mathbb{N}$)) is ExpTime-complete. With the concrete domain $(\mathbb{Q}, <, =, (=_{\mathbbm{d}})_{\mathbbm{d} \in \mathbb{Q}})$, all the trees in $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})})$ are satisfiable (no need to intersect $A_{\mathrm{cons}(\mathbb{A})}$ with a hypothetical $A_{\bigstar\mathrm{C}}$, see e.g. [Lut01, BC02, DD07, Gas09]), and therefore SAT(CTL($\mathbb{Q}$)) is in ExpTime too. TCA can be also used to show that the concept satisfiability w.r.t. general TBoxes for the description logic $\mathcal{ALCF}^{\mathcal{P}}(\mathbb{Z})$ is in ExpTime [LOS20, Lab21], see the details in [DQ23b, Section 5.2].

6. Complexity of the Satisfiability Problem for the Logic CTL*(ℤ)

In this section, we show that SAT(CTL*(ℤ)) can be solved in 2ExpTime by using Rabin TCA (see e.g. Section 4.4). We follow the automata-based approach for CTL*, see e.g. [ES84, EJ00]. Besides checking that the essential steps for CTL* can be lifted to CTL*(ℤ), we also prove that computationally we are in a position to provide an optimal complexity upper bound.

6.1. CTL*(ℤ) formulae in special form.

We start by establishing a special form for CTL*(ℤ) formulae from which Rabin TCA will be defined, following ideas from [ES84] for CTL*. A CTL*(ℤ) state formula ϕ is in *special form* if it has the form below

$$\mathsf{E}\ (\mathtt{x}_1 = 0)\ \wedge\ \Big(\bigwedge_{i\in[1,D-1]} \mathsf{AGE}\ \Phi_i\Big)\ \wedge\ \Big(\bigwedge_{j\in[1,D']} \mathsf{A}\ \Phi'_j\Big), \tag{SF}$$

where the Φ_i's and the Φ'_j's are LTL(ℤ) formulae in simple form (see Section 2), for some $D \geqslant 1$, $D' \geqslant 0$. We prove below that we can restrict ourselves to CTL*(ℤ) state formulae in special form. Indeed, formulae in simple form are in negation normal form and its terms are only from $\mathrm{T}_V^{\leqslant 1}$. Formulae in special form are not only in simple form but also the subformulae are normalised as described in (SF). In order to restrict ourselves to formulae in special form, we first adapt the proof of Proposition 5.1 to CTL*(ℤ), leading to Proposition 6.1 and this is to handle only simple forms. Then, Proposition 6.2 is dedicated to formulae in special form. The proof of Proposition 6.1 below can be found in Appendix D.1.

Proposition 6.1. *For every* CTL*(ℤ) *state formula ϕ, one can construct in polynomial-time in the size of ϕ a* CTL*(ℤ) *formula ϕ' in simple form such that ϕ is satisfiable iff ϕ' is satisfiable.*

Hence the size of ϕ' is polynomial in the size of ϕ. Using this, we establish the property that allows us to restrict ourselves to CTL*(ℤ) state formulae in special form only.

Proposition 6.2. *For every* CTL*(ℤ) *state formula ϕ, one can construct in polynomial time in the size of ϕ a* CTL*(ℤ) *formula ϕ' in special form such that ϕ is satisfiable iff ϕ' is satisfiable.*

So ϕ' is also of polynomial size in the size of ϕ. The proof of Proposition 6.2 can be found in Appendix D.2. It is similar to the proof for CTL* except that we need to handle constraints.

Let us now state a tree model property of formulae in special form, with a strict discipline on the witness paths. Given a tree $\mathbb{t} : [0, D-1] \to \mathbb{Z}^\beta$ with $\mathbb{t} \models \mathsf{AGE}\ \Phi$, we say that $\mathbb{t}$ *satisfies* $\mathsf{AGE}\ \Phi$ *via the direction* i, for some $i \in [1, D-1]$, iff for all nodes $\mathbf{n} \in [0, D-1]^*$, we have $w \models \Phi$, where $w : \mathbb{N} \to \mathbb{Z}^\beta$ is defined by $w(0) \stackrel{\text{def}}{=} \mathbb{t}(\mathbf{n})$ and $w(j) \stackrel{\text{def}}{=} \mathbb{t}(\mathbf{n} \cdot i \cdot 0^{j-1})$ for all $j \geqslant 1$. Proposition 6.3 below is a counterpart of [ES84, Theorem 3.2] but for CTL*(ℤ) instead of CTL*, see also the variant [Gas09, Lemma 3.3].

Proposition 6.3. *Let ϕ be a* CTL*(ℤ) *state formula in special form (with the notations of equation* (SF) *on page 29) built over the variables $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$. The formula ϕ is satisfiable iff there is a tree $\mathbb{t} : [0, D-1] \to \mathbb{Z}^\beta$ such that $\mathbb{t}, \varepsilon \models \phi$ and for each $i \in [1, D-1]$, $\mathbb{t}$ satisfies $\mathsf{AGE}\ \Phi_i$ via the direction i, that is, if $\mathbb{t}, \mathbf{n} \models \mathsf{E}\ \Phi_i$, then Φ_i is satisfied on the path $\mathbf{n} \cdot i \cdot 0^\omega$.*

Proof. Assume that ϕ has the form below:

$$\mathsf{E}\ (\mathtt{x}_1 = 0)\ \wedge\ \Big(\bigwedge_{i \in [1,D-1]} \mathsf{AGE}\ \Phi_i\Big)\ \wedge\ \Big(\bigwedge_{j \in [1,D']} \mathsf{A}\ \Phi'_j\Big).$$

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

We introduce an auxiliary map $g : [0, D-1]^* \to \mathcal{W}$ such that $g(\varepsilon) \stackrel{\text{def}}{=} w_{\text{in}}$, $\mathbb{t}(\varepsilon) \stackrel{\text{def}}{=} \mathfrak{v}(g(\varepsilon))$. More generally, we require that for all nodes $\mathbf{n} \in [0, D-1]^*$, we have $\mathbb{t}(\mathbf{n}) = \mathfrak{v}(g(\mathbf{n}))$, and if $\mathbf{n} = j_1 \cdot \cdots \cdot j_k$, then there exists a finite path $g(\varepsilon)g(j_1)g(j_1 j_2) \ldots g(\mathbf{n})$ in $\mathcal{K}$. Note that this is satisfied by ε, 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 $\mathbf{n} = j_1 \cdot \cdots \cdot j_k$, then there exists a finite path $g(\varepsilon)g(j_1)g(j_1 j_2) \ldots g(\mathbf{n})$ in $\mathcal{K}$.

If $j = 0$, then since $\mathcal{K}$ is total, there is an infinite path $\pi = w_0 w_1 w_2 \cdots$ starting from $g(\mathbf{n})$. For all $k \geqslant 1$, we set $g(\mathbf{n} \cdot 0^k) \stackrel{\text{def}}{=} w_k$ and $\mathbb{t}(\mathbf{n} \cdot 0^k) \stackrel{\text{def}}{=} \mathfrak{v}(w_k)$.

Otherwise ($j \neq 0$), since $g(\mathbf{n})$ is a world on a path starting from w_{in}, we obtain $\mathcal{K}, g(\mathbf{n}) \models \mathsf{E}\Phi_j$ by assumption. So there exists some infinite path $\pi = w_0 w_1 w_2 \ldots$ 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 $\mathbb{t}(\mathbf{n} \cdot j \cdot 0^k) \stackrel{\text{def}}{=} \mathfrak{v}(w_{k+1})$ for all $k \geqslant 0$. Note that this implies that $\mathbb{t}(\mathbf{n})\mathbb{t}(\mathbf{n} \cdot j)\mathbb{t}(\mathbf{n} \cdot j \cdot 0)\mathbb{t}(\mathbf{n} \cdot j \cdot 0^2) \ldots$ satisfies Φ_j and therefore $\mathbb{t}, \mathbf{n} \models \mathsf{E}\ \Phi_j$. By construction, we get $\mathbb{t} \models \mathsf{AGE}\ \Phi_j$ for all $j \in [1, D-1]$. Moreover, by construction of $\mathbb{t}$, for all infinite paths $j_1 j_2 \cdots \in [0, D-1]^\omega$, $g(\varepsilon)g(j_1)g(j_1 j_2) \cdots$ is an infinite path from $g(\varepsilon)$, consequently $\mathbb{t} \models \mathsf{E}\ (\mathtt{x}_1 = 0) \wedge \bigwedge_{j \in [1,D']} \mathsf{A}\ \Phi'_j$ too. Hence, $\mathbb{t} \models \phi$ and $\mathbb{t}$ satisfies $\mathsf{AGE}\ \Phi_i$ via the direction i, for each $i \in [1, D-1]$. □

Proposition 6.3 justifies our restriction to infinite trees and to TCA in the rest of this section. Next, we will show how to translate CTL*($\mathbb{Z}$) formulae in special form as in (SF) into TCA accepting their corresponding infinite tree models. We will start by giving the construction of *word* constraint automata for the underlying LTL($\mathbb{Z}$) formulas.

6.2. Constructing Automata for LTL($\mathbb{Z}$) Formulas.

In this section, we translate LTL($\mathbb{Z}$) formulae in simple form into equivalent *word* constraint automata (TCA of degree 1). Adapting the standard automata-based approach for LTL [VW94], we can show the following proposition.

Proposition 6.4. *Let Φ be an* LTL($\mathbb{Z}$) *formula in simple form. There is a word constraint automaton $\mathbb{A}_\Phi$ such that* $\mathrm{L}(\mathbb{A}_\Phi) = \{w : \mathbb{N} \to \mathbb{Z}^\beta \mid w \models \Phi\}$*, and the following conditions hold.*

(I) : *The number of locations in $\mathbb{A}_\Phi$ is bounded by* $\mathtt{size}(\Phi) \times 2^{2 \times \mathtt{size}(\Phi)}$.
(II): *The cardinality of δ in $\mathbb{A}_\Phi$ is in $2^{\mathcal{O}(P(\mathtt{size}(\Phi)))}$ for some polynomial $P(\cdot)$.*
(III): *The maximal size of a constraint in $\mathbb{A}_\Phi$ is quadratic in* $\mathtt{size}(\Phi)$.

In the proof of Proposition 6.4 (see Appendix D.3), 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. In Proposition 6.4 and later in several places, we consider that the constraint automata accept trees of the form $[0, D-1]^* \to \mathbb{Z}^\beta$ for some $D, \beta \geqslant 1$ (i.e., by dropping the finite alphabet), assuming implicitly a singleton alphabet. We remark that the automaton $\mathbb{A}_\Phi$ may not be deterministic; for instance, for Φ equal to $\mathsf{FG}\ (\mathtt{x} = \mathtt{x}')$ there does not exist any *deterministic* word constraint automaton with Büchi acceptance condition such that $w \models \Phi$ iff $w \in \mathrm{L}(\mathbb{A}_\Phi)$. However, for handling CTL*($\mathbb{Z}$) formulae of the form $\mathsf{A}\ \Phi$, we need the automaton accepting the word models of Φ to be deterministic. The forthcoming Theorem 7.2 dedicated to the determinisation of word constraint automata (in Section 7) and Proposition 6.4 imply the following result.

Corollary 6.5. *Let Φ be an LTL($\mathbb{Z}$) formula in simple form built over the variables $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$ and the constants $\mathbb{d}_1, \ldots, \mathbb{d}_\alpha$. There exists a deterministic Rabin word constraint automaton $\mathbb{A}_\Phi$ such that $\mathrm{L}(\mathbb{A}_\Phi) = \{w : \mathbb{N} \to \mathbb{Z}^\beta \mid w \models \Phi\}$, and the following conditions hold.*

(I) : *The number of locations in $\mathbb{A}_\Phi$ is bounded by $2^{2^{\mathcal{O}(P^\dagger(\mathtt{size}(\Phi)))}}$ for some polynomial $P^\dagger(\cdot)$.*
(II): *The number of Rabin pairs is bounded by $2 \times \mathtt{size}(\Phi) \times 2^{2 \times \mathtt{size}(\Phi)}$.*
(III): *The cardinality of δ in $\mathbb{A}_\Phi$ is bounded by $\mathrm{card}(\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)) \times 2^{2^{\mathcal{O}(P^\dagger(\mathtt{size}(\Phi)))+1}}$.*
(IV): $\mathtt{MCS}(\mathbb{A}_\Phi)$ *is cubic in $\beta + \max(\lceil log(|\mathbb{d}_1|)\rceil, \lceil log(|\mathbb{d}_\alpha|)\rceil)$, i.e. polynomial in $\mathtt{size}(\Phi)$.*

6.3. **Constructing Automata for CTL*($\mathbb{Z}$) Formulas.** We are now ready to give the construction of the Rabin TCA accepting precisely the tree models of a CTL*($\mathbb{Z}$) formula in special form. For the rest of this section, suppose ϕ is a CTL*($\mathbb{Z}$) formula over the variables $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$ and of the form

$$\mathsf{E}\ (\mathtt{x}_1 = 0) \ \wedge \ \Big(\bigwedge_{i \in [1, D-1]} \mathsf{AGE}\ \Phi_i\Big) \ \wedge \ \Big(\bigwedge_{j \in [1, D']} \mathsf{A}\ \Phi'_j\Big),$$

where the Φ_i's and the Φ'_j's are LTL($\mathbb{Z}$) formulae in simple form. Our approach is modular, that is, we construct for each conjunct a corresponding (Rabin) TCA of degree D with β variables. Calling Lemma 3.1 then yields the desired Rabin TCA. Let us remark that the alphabet Σ does not play any role here, and we set Σ to the dummy singleton alphabet $\{\dagger\}$ for the rest of this section.

Handling $\mathsf{E}(\mathtt{x}_1 = 0)$. Define the TCA $\mathbb{A}_0 = (Q, \Sigma, D, \beta, Q_{\mathrm{in}}, \delta, F)$ where
- $Q = \{q, q'\}, Q_{\mathrm{in}} = \{q_0\}, F = \{q'\}$,
- $\delta = \{(q, \dagger, (\mathtt{x}_1 = 0, q'), (\top, q'), \ldots, (\top, q')), (q', \dagger, (\top, q'), \ldots, \ldots, (\top, q'))\}$.

Clearly, $\mathrm{L}(\mathbb{A}_0) = \{\mathbb{t} : [0, D-1]^* \to \mathbb{Z}^\beta \mid \mathbb{t} \models \mathsf{E}\ (\mathtt{x}_1 = 0)\}$.

Handling $\mathsf{AGE}\ \Phi_i$. Let us construct, for every $i \in [0, D-1]$, a (Büchi) TCA $\mathbb{A}_i$ of degree D such that $\mathrm{L}(\mathbb{A}_i) = \{\mathbb{t} : [0, D-1]^* \to \mathbb{Z}^\beta \mid \mathbb{t} \models \mathsf{AGE}\ \Phi_i$ and $\mathbb{t}$ satisfies $\mathsf{AGE}\ \Phi_i$ via direction $i\}$. The idea is to construct $\mathbb{A}_i$ so that it starts off the word constraint automaton dedicated to the LTL($\mathbb{Z}$) formula Φ_i at each node $\mathbf{n}$ of the tree and runs it down the designated path $\mathbf{n} \cdot i\ \cdot 0^\omega$ to check whether Φ_i actually holds along this path. Let $\mathbb{A} = (Q, \Sigma, \beta, Q_{\mathrm{in}}, \delta, F)$ be the Büchi word constraint automaton such that $\mathrm{L}(\mathbb{A}) = \{w : \mathbb{N} \to \mathbb{Z}^\beta \mid w \models \Phi_i\}$ (see Proposition 6.4). Let us define $\mathbb{A}_i = (Q', \Sigma, D, \beta, Q'_{\mathrm{in}}, \delta', F')$, where
- $Q' \stackrel{\text{def}}{=} [0, D-1] \times (Q \cup \{\bot\})$, where $\bot \notin Q$,

- $Q'_{\rm in} \stackrel{\rm def}{=} \{(0, \perp)\}$, $F' \stackrel{\rm def}{=} \{(j, q) \mid j \neq 0 \text{ or } q \in F\}$,
- The transition relation δ' is made of tuples of the form

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

verifying the conditions below.
(1) $(q_{\rm in}, \dagger, \Theta_i, q_i) \in \delta$ for some $q_{\rm in} \in Q_{\rm in}$ (starting off $\mathbb{A}$ in the $i^{\rm th}$ 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 $0^{\rm th}$ child).
(3) $q_j = \perp$ and $\Theta_j = \top$ for all $j \in ([0, D-1] \backslash \{0, i\})$.

Lemma 6.6. $\mathrm{L}(\mathbb{A}_i) = \{\mathbb{t} : [0, D-1]^* \to \mathbb{Z}^\beta \mid \mathbb{t} \models \mathsf{AGE}\ \Phi_i$ *and* $\mathbb{t}$ *satisfies* $\mathsf{AGE}\ \Phi_i$ *via* $i\}$, *and*

(I) : *the number of locations is bounded exponential in* $\mathtt{size}(\phi)$*, and*
(II): $\mathtt{MCS}(\mathbb{A}_i)$ *is bounded polynomial in* $\mathtt{size}(\phi)$*.*

The proof of Lemma 6.6 can be found in Appendix D.4. Recall that a Büchi TCA can be seen as a Rabin TCA with a single Rabin pair.

Handling $\mathsf{A}\ \Phi'_j$. Let us construct, for every $j \in [0, D']$, a Rabin TCA $\mathbb{A}'_j$ of degree D such that $\mathrm{L}(\mathbb{A}'_j) = \{\mathbb{t} : [0, D-1]^* \to \mathbb{Z}^\beta \mid \mathbb{t} \models \mathsf{A}\ \Phi'_j\}$, and the number of Rabin acceptance pairs is exponential in $\mathtt{size}(\Phi'_j)$. Let Φ'_j be an LTL($\mathbb{Z}$) formula in simple form built over the variables $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$ and the constants $\mathbb{d}_1, \ldots, \mathbb{d}_\alpha$. By Corollary 6.5, there exists a deterministic Rabin word constraint automaton $\mathbb{A} = (Q, \Sigma, \beta, Q_{\rm in}, \delta, F)$ such that $\mathrm{L}(\mathbb{A}) = \{w : \mathbb{N} \to \mathbb{Z}^\beta \mid w \models \Phi'_j\}$. Define the Rabin TCA $\mathbb{A}'_j = (Q', \Sigma, D, \beta, Q'_{\rm in}, \delta', \mathcal{F}')$, where

- $Q' \stackrel{\rm def}{=} Q$; $Q'_{\rm in} \stackrel{\rm def}{=} Q_{\rm in}$; $\mathcal{F}' \stackrel{\rm def}{=} F$.
- δ' is made of tuples of the form $(q, \mathtt{a}, (\Theta_0, q_0), \ldots, (\Theta_{D-1}, q_{D-1}))$, where $(q, \mathtt{a}, \Theta_i, q_i) \in \delta$ for all $0 \leqslant i < D$.

Lemma 6.7. $\mathrm{L}(\mathbb{A}'_j) = \{\mathbb{t} : [0, D-1]^* \to \mathbb{Z}^\beta \mid \mathbb{t} \models \mathsf{A}\ \Phi'_j\}$*, and*

(I) : *the number of locations is double exponential in* $\mathtt{size}(\phi)$*,*
(II): *the number of Rabin acceptance pairs is exponential in* $\mathtt{size}(\phi)$*,*
(III): $\mathtt{MCS}(\mathbb{A}'_j)$ *is bounded polynomial in* $\mathtt{size}(\phi)$*.*

The proof is by an easy verification thanks to the determinism of $\mathbb{A}$, which is essential here. Typically, Proposition 6.4 is not sufficient because determinism of the word automaton is required. For example, assuming that all the branches of $\mathbb{t} : [0, D-1]^* \to \mathbb{Z}^\beta$ satisfy the formula Φ'_j, if $\mathrm{L}(\mathbb{A})$ is equal to all words satisfying Φ'_j and $\mathbb{A}$ were nondeterministic, then we cannot guarantee that each nondeterministic step in $\mathbb{A}$ can lead to acceptance for all possible future branches. As a matter of fact, the determinisation construction presented in Section 7 is key to design Rabin TCA for formulae of the form $\mathsf{A}\ \Phi'_j$, which is why we put so much efforts in developing the corresponding material. Moreover, Lemma 6.7(I) is a direct consequence of Corollary 6.5(I), Lemma 6.7(II) is a direct consequence of Corollary 6.5(II) and Lemma 6.7(III) is a consequence of Corollary 6.5(IV). Indeed, $\mathtt{MCS}(\mathbb{A})$ is polynomial in $\mathtt{size}(\Phi'_j)$ (and therefore polynomial in $\mathtt{size}(\phi)$), the constraints in $\mathbb{A}'_j$ are those from $\mathbb{A}$ and therefore $\mathtt{MCS}(\mathbb{A}'_j)$ is polynomial in $\mathtt{size}(\phi)$.

6.4. **The final step.** We are now ready to give the final Rabin TCA $\mathbb{A}$ such that $\mathrm{L}(\mathbb{A})$ is precisely the set of tree models of ϕ. By Lemma 3.1, there exists a Rabin TCA $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\mathrm{in}}, \delta, F)$ such that

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

In Section 4.4, we have seen that nonemptiness of the language $\mathrm{L}(\mathbb{A})$ for $\mathbb{A}$ can be solved in time in

$$R_1\big(\mathrm{card}(Q) \times \mathrm{card}(\delta) \times \mathtt{MCS}(\mathbb{A}) \times \mathrm{card}(\Sigma) \times R_2(\beta + \mathrm{card}(F))\big)^{\mathcal{O}(R_2(\beta + \mathrm{card}(F)) \times R_3(D))}$$

Let us thus evaluate the size of the respective components for $\mathbb{A}$. First of all, observe that the involved Büchi TCA can be seen as Rabin TCA with a single Rabin acceptance pair. Further, $\beta, D + D' \leqslant \mathtt{size}(\phi)$, and Σ is a singleton.

- By Lemmas 6.7 and 3.1 and $D \leqslant \mathtt{size}(\phi)$, the number of Rabin pairs in $\mathbb{A}$ is exponential in $\mathtt{size}(\phi)$.
- By Lemmas 6.6 and 6.7, the number of locations in each involved automaton is at most double-exponential in $\mathtt{size}(\phi)$; and the number of Rabin pairs is only exponential in $\mathtt{size}(\phi)$. By Lemma 3.1 and $D + D' \leqslant \mathtt{size}(\phi)$, the number of locations in $\mathbb{A}$ is double-exponential in $\mathtt{size}(\phi)$. 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 involved automaton is polynomial in $\mathtt{size}(\phi)$. The maximal size of a constraint in the product automaton $\mathbb{A}$ is therefore polynomial in ϕ too ($D + D' \leqslant \mathtt{size}(\phi)$).

Putting all results together, the nonemptiness of $\mathrm{L}(\mathbb{A})$ can be checked in double-exponential time in $\mathtt{size}(\phi)$, leading to Theorem 6.8 below. It answers open questions from [BG06, CKL16, CT16, LOS20], and it is the main result of the paper.

Theorem 6.8. SAT(CTL*($\mathbb{Z}$)) *is* 2EXPTIME-*complete.*

2EXPTIME-hardness is inherited from SAT(CTL*) [VS85, Theorem 5.2]. As a corollary, SAT(CTL*($\mathbb{N}$)) is also 2EXPTIME-complete. Indeed, SAT(CTL*($\mathbb{N}$)) can be easily reduced to SAT(CTL*($\mathbb{Z}$)) in polynomial-time (for instance, one could introduce a fresh variable $\mathtt{x}$ enforced to be equal to zero –use of $\mathsf{AG}(\mathtt{x} = 0)$– and further enforce that all the original variables $\mathtt{x}_i$ are such that $\mathtt{x}_i \geqslant \mathtt{x}$). Furthermore, assuming that $<_{\mathrm{pre}}$ 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 6.9. SAT(CTL*($\{0, 1\}^*, <_{\mathrm{pre}}$)) *is* 2EXPTIME-*complete.*

Actually, the 2EXPTIME-membership also holds for SAT(CTL*($\mathbb{D}^*, <_{\mathrm{pre}}$)), where $\mathbb{D}$ is an infinite domain, see e.g. [DD16, Section 4.2].

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

7. Determinisation of Büchi Word Constraint Automata

In this section, we present a construction to transform nondeterministic Büchi word constraint automata to equivalent deterministic Rabin word constraint automata (Theorem 7.2 below, used to establish Corollary 6.5 in Section 6.3). The construction is based on Safra's well known construction for Büchi automata [Saf89, Theorem 1.1]. In our adaptation, we give a special attention 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 they are essential to perform the complexity analysis in Section 6.3.

A constraint word automaton $\mathbb{A}$ is *deterministic* whenever for all locations q, letters $\mathtt{a}$ and pairs of valuations $(\boldsymbol{z}, \boldsymbol{z}') \in \mathbb{Z}^{2\beta}$, there exists in $\mathbb{A}$ *at most* one transition $(q, \mathtt{a}, \Theta, q')$ such that $\mathbb{Z} \models \Theta(\boldsymbol{z}, \boldsymbol{z}')$.

We start with defining the key notion, namely *Safra trees*. A *Safra tree over Q* is a finite tree $\mathbf{s}$, satisfying the following conditions. Note that we use the symbol '$\mathbf{s}$' for trees because later on, such trees shall be locations (a.k.a. states) of the forthcoming (deterministic) Rabin word constraint automaton.

(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 (the rightmost sibling is understood as the youngest one).
(2) Every node in $\mathbf{s}$ has a unique *name* from the interval $[1, 2 \cdot \mathrm{card}(Q)]$, no two nodes have the same name.
(3) Every node has a *label* from $\mathcal{P}(Q)\backslash\{\varnothing\}$. We use $\mathrm{Lab}(\mathbf{s}, J) \subseteq Q$ 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*.

The proof of the following lemma can be found in Appendix E.1.

Lemma 7.1. *A Safra tree over Q has at most* $\mathrm{card}(Q)$ *nodes.*

Let us prove that for every (nondeterministic) Büchi word constraint automaton one can construct a deterministic Rabin word constraint automaton accepting the same language. The following result generalizes [Saf89, Theorem 1.1] to constraint automata.

Theorem 7.2. *Let $\mathbb{A} = (Q, \Sigma, \beta, Q_{in}, \delta, F)$ be a Büchi word constraint automaton involving the constants $\mathbb{d}_1, \ldots, \mathbb{d}_\alpha$. There is a deterministic Rabin word constraint automaton $\mathbb{A}' = (Q', \Sigma, \beta, Q'_{in}, \delta', F)$ such that* $\mathrm{L}(\mathbb{A}) = \mathrm{L}(\mathbb{A}')$ *verifying the following quantitative properties.*

(I) : $\mathrm{card}(Q')$ *is exponential in* $\mathrm{card}(Q)$ *and the number of Rabin pairs in $\mathbb{A}'$ is bounded by* $2 \cdot \mathrm{card}(Q)$ *(same bounds as in* [Saf89, Theorem 1.1]*).*

(II): *The constraints in the transitions are from* $\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$*, are of size cubic in* $\beta + \max(\lceil log(|\mathbb{d}_1|)\rceil, \lceil log(|\mathbb{d}_\alpha|)\rceil)$ *and* $\mathrm{card}(\delta') \leqslant \mathrm{card}(Q')^2 \times \mathrm{card}(\Sigma) \times ((\mathbb{d}_\alpha - \mathbb{d}_1) + 3)^{2\beta} 3^{2\beta^2}$.

Proof. We define the det. Rabin word constraint automaton $\mathbb{A}' = (Q', \Sigma, \beta, Q'_{\mathrm{in}}, \delta', \mathcal{F}')$ as follows. Essential differences with the construction to prove [Saf89, Theorem 1.1] can be found in the definition of δ' below, condition (3), as constraints need to be taken into account.

- Q' is the set of all Safra trees over Q.
- $Q'_{\mathrm{in}} \stackrel{\text{def}}{=} \{\mathbf{s}_{Q_{\mathrm{in}}}\}$, where $\mathbf{s}_{Q_{\mathrm{in}}}$ is the Safra tree with a single unmarked node $\mathbf{n}$ with name 1 and $\mathrm{Lab}(\mathbf{s}_{Q_{\mathrm{in}}}, 1) \stackrel{\text{def}}{=} Q_{\mathrm{in}}$.

- The finite transition relation $\delta' \subseteq (Q' \times \Sigma \times \text{TreeCons}(\beta) \times Q')$ is defined as follows. Recall that $\text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ denotes the set of all satisfiable complete constraints over $\mathtt{x}_1, \ldots, \mathtt{x}_\beta, \mathtt{x}'_1, \ldots, \mathtt{x}'_\beta$. The transition relation δ' is defined over this strict subset of $\text{TreeCons}(\beta)$. Let $\mathbf{s}$ be a Safra tree, $\mathtt{a} \in \Sigma$, and $\Theta \in \text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$. We set $(\mathbf{s}, \mathtt{a}, \Theta, \mathbf{s}') \in \delta'$, where $\mathbf{s}'$ is obtained from $\mathbf{s}$, $\mathtt{a}$ and Θ 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 in $\mathbf{s}^{(1)}$ 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 the smallest number in $[1, 2 \cdot \text{card}(Q)]$ that is not assigned to any of the other nodes in $\mathbf{s}^{(1)}$ yet. Let us use $\mathbf{s}^{(2)}$ to denote the resulting Safra tree.
  (3) Apply the powerset construction to every node in $\mathbf{s}^{(2)}$, that is, for every node with label $P \subseteq Q$, replace P by
  $$\bigcup_{q \in P} \{q' \in Q \mid \text{ there exists } (q, \mathtt{a}, \Theta', q') \in \delta \text{ such that } \Theta \models \Theta'\}.$$
  Let us use $\mathbf{s}^{(3)}$ to denote the resulting tree. Note that $\Theta \models \Theta'$ can be checked in polynomial-time because $\Theta \in \text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$. The tree $\mathbf{s}^{(3)}$ may not satisfy the condition (5) of Safra trees, but this is only provisionally.
  (4) (Horizontal Merge) For every two nodes in $\mathbf{s}^{(3)}$ 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*, yielding $\mathbf{s}^{(5)}$.
  (6) (Vertical Merge) For every node in $\mathbf{s}^{(5)}$ whose label equals the union of the labels of its children nodes (if there are any), remove *all descendants of this node* and mark it. The resulting Safra tree is $\mathbf{s}'$.
- The set of Rabin pairs $\mathcal{F}'$ is equal to $\mathcal{F}' = \{(L_1, U_1), \ldots, (L_{2 \cdot \text{card}(Q)}, U_{2 \cdot \text{card}(Q)})\}$, where for all $1 \leqslant J \leqslant 2 \cdot \text{card}(Q)$,
  – $L_J = \{\mathbf{s} \in Q' \mid \mathbf{s} \text{ contains a node with name } J \text{ marked}\}$,
  – $U_J = \{\mathbf{s} \in Q' \mid \mathbf{s} \text{ does not contain a node with name } J\}$.

It is worth observing that there is a slight abuse of notation here: $\text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ is not strictly a subset of $\text{TreeCons}(\beta)$ because of the constraints of the form $\mathtt{x} < \mathbb{d}_1$ and $\mathtt{x} > \mathbb{d}_\alpha$. However, this can be easily simulated by adding two new variables $\mathtt{x}_{\mathbb{d}_1}$ and $\mathtt{x}_{\mathbb{d}_\alpha}$ and to perform the following changes in the automata: replace $\mathtt{x} < \mathbb{d}_1$ by $\mathtt{x} < \mathtt{x}_{\mathbb{d}_1}$, replace $\mathtt{x} > \mathbb{d}_\alpha$ by $\mathtt{x} > \mathtt{x}_{\mathbb{d}_\alpha}$ and add to every constraint the conjunct $\mathtt{x}_{\mathbb{d}_1} = \mathbb{d}_1 \wedge \mathtt{x}_{\mathbb{d}_\alpha} = \mathbb{d}_\alpha$. This only adds a constant to the maximal constraint size as well as two to the number of variables. Therefore, this is harmless for all the complexity results established in the paper. For the sake of readability, we keep below β instead of $\beta + 2$ for the number of variables, but the reader should keep in mind that $\text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ are constraints in $\mathbb{A}'$ at the cost of adding two auxiliary variables and a constant-size constraint to every constraint.

Let us prove the correctness of the construction of $\mathbb{A}'$. For proving $\text{L}(\mathbb{A}) \subseteq \text{L}(\mathbb{A}')$, let $w = (\mathtt{a}_1, \boldsymbol{z}_1)(\mathtt{a}_2, \boldsymbol{z}_2)(\mathtt{a}_2, \boldsymbol{z}_2) \ldots$ be an infinite word over $\Sigma \times \mathbb{Z}^\beta$ such that $w \in \text{L}(\mathbb{A})$. Then there is some initialized Büchi accepting run $\rho : \mathbb{N} \to \delta$ of $\mathbb{A}$ on w, say, of the form
$$(q_1, \mathtt{a}_1, \Theta'_1, q_2)(q_2, \mathtt{a}_2, \Theta'_2, q_3)(q_3, \mathtt{a}_3, \Theta'_3, q_4) \ldots$$
satisfying $q_1 \in Q_{\text{in}}$, $\mathbb{Z} \models \Theta'_i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1})$, and there exists some $q \in F$ that appears infinitely often in ρ. Let us fix such an accepting location and denote it by q_{acc}. We are going to

construct an initialized Rabin accepting run of $\mathbb{A}'$ on 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 $\text{Lab}(\mathbf{s}_{Q_{\text{in}}}, 1) = Q_{\text{in}}$ – the only initial location of $\mathbb{A}'$. We prove that for all $i \geqslant 1$, 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 $\Theta_i \in \text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ and a unique Safra tree $\mathbf{s}_{i+1}$ such that

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

So let $i \geqslant 1$. Let Θ_i be the unique constraint in $\text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ such that $\mathbb{Z} \models \Theta_i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1})$ (unicity and existence guaranteed by Lemma 4.2). By definition of δ', there exists a unique Safra tree $\mathbf{s}_{i+1}$ such that $(\mathbf{s}_i, \mathtt{a}_i, \Theta_i, \mathbf{s}_{i+1}) \in \delta'$. For proving the third condition, recall how $\mathbf{s}_{i+1}$ is obtained from $\mathbf{s}_i$:

- By assumption, $\mathbf{s}_i$ has root node with name 1 and $q_i \in \text{Lab}(\mathbf{s}_i, 1)$.
- By definition, $q_i \in \text{Lab}(\mathbf{s}_i^{(2)}, 1)$.
- Recall that $(q_i, \mathtt{a}_i, \Theta'_i, q_{i+1}) \in \delta$ and $\mathbb{Z} \models \Theta'_i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1})$. Hence $\Theta_i \models \Theta'_i$ (by Lemma 4.2) and $q_{i+1} \in \{q' \in Q \mid \text{there exists } (q_i, \mathtt{a}_i, \Theta', q') \in \delta,\ \Theta_i \models \Theta'\}$. Then also $q_{i+1} \in \text{Lab}(\mathbf{s}_i^{(3)}, 1)$.
- $q_{i+1} \in \text{Lab}(\mathbf{s}_i^{(4)}, 1)$ because the root node 1 has no siblings.
- $q_{i+1} \in \text{Lab}(\mathbf{s}_i^{(5)}, 1)$ as the label set of the root node is nonempty.
- Finally $q_{i+1} \in \text{Lab}(\mathbf{s}_{i+1}, 1)$ because the root node cannot be removed.

Since $\mathbf{s}_1$'s root node has the name 1 and $q_1 \in \text{Lab}(\mathbf{s}_1, 1)$, we just have proved that for all $i \geqslant 1$, $\mathbf{s}_i$'s root node has name 1, $q_i \in \text{Lab}(\mathbf{s}_i, 1)$, and there exist a unique constraint Θ_i and a unique Safra tree $\mathbf{s}_{i+1}$ such that $(\mathbf{s}_i, a_i, \Theta_i, \mathbf{s}_{i+1}) \in \delta'$ and $\mathbb{Z} \models \Theta_i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1})$. Hence ρ' of the form

$$(\mathbf{s}_1, \mathtt{a}_1, \Theta_1, \mathbf{s}_2)(\mathbf{s}_2, \mathtt{a}_2, \Theta_2, \mathbf{s}_3)(\mathbf{s}_3, \mathtt{a}_3, \Theta_3, \mathbf{s}_4)\ldots$$

initialized (unique) run of $\mathbb{A}'$ on w. For proving that ρ' is Rabin accepting, we use the following lemma, whose proof can be found in Appendix E.2.

Lemma 7.3. *For every position $i \geqslant 1$ and every name $1 \leqslant J_i \leqslant 2 \cdot \text{card}(Q)$, if $\mathbf{s}_k$ contains a node with name J_i and $q_k \in Lab(\mathbf{s}_k, J_i)$ for all $k \geqslant i$, then*

(1) *there exist infinitely many $k \geqslant i$ such that the node with name J_i is marked in $\mathbf{s}_k$; or*
(2) *there exists some position $i' \geqslant i$ and some name $1 \leqslant J_{i'} \leqslant 2 \cdot \text{card}(Q)$ with $J_i \neq J_{i'}$ such that for all $k \geqslant i'$, the node with name J_i has a child node with name $J_{i'}$ and $q_k \in Lab(\mathbf{s}_k, J_{i'})$.*

Finally, recall that for every $k \geqslant 0$, 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 $k \geqslant i$ 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 $k \geqslant 0$ the Safra tree $\mathbf{s}_k$ does not contain the node with name J, that is, $\mathbf{s}_k \in U_J$ (indeed, by assumption, for all $k \geqslant 1$, there is in $\mathbf{s}_k$ a node with name J). Hence, the run ρ' is Rabin accepting.

Case 2: Otherwise, by Lemma 7.3, there exists some $i' \geqslant i$ and some $1 \leqslant J' \leqslant 2 \cdot \text{card}(Q)$ with $J \neq J'$ such that for all $k \geqslant i'$, 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 7.1, every Safra tree has at most $\text{card}(Q)$ nodes.

For proving $\mathrm{L}(\mathbb{A}') \subseteq \mathrm{L}(\mathbb{A})$, let $w = (\mathtt{a}_1, \boldsymbol{z}_1)(\mathtt{a}_2, \boldsymbol{z}_2)(\mathtt{a}_2, \boldsymbol{z}_2)\ldots$ be an infinite word over $\Sigma \times \mathbb{Z}^\beta$ such that $w \in \mathrm{L}(\mathbb{A}')$. Then there exists some (unique) initialized Rabin accepting run $\rho' : \mathbb{N} \to \delta'$ of $\mathbb{A}'$ on w, say

$$(\mathbf{s}_1, \mathtt{a}_1, \Theta_1, \mathbf{s}_2)(\mathbf{s}_2, \mathtt{a}_2, \Theta_2, \mathbf{s}_3)(\mathbf{s}_3, \mathtt{a}_3, \Theta, \mathbf{s}_4)\ldots.$$

That is, $\mathbf{s}_1 = \mathbf{s}_{Q_{\mathrm{in}}}$, $\mathbb{Z} \models \Theta_i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1})$ for all $i \geqslant 1$, and there exists some $1 \leqslant J \leqslant 2 \cdot \mathrm{card}(Q)$ such that $\mathbf{s}_i \in L_J$ for infinitely many $i \geqslant 0$ and $\mathbf{s}_i \in U_J$ for only finitely many $i \geqslant 0$. Fix such a name $1 \leqslant J \leqslant 2 \cdot \mathrm{card}(Q)$. Then, there must exist some position $i \geqslant 0$ such that $\mathbf{s}_k$ contains a node with name J, for all $k \geqslant i$. We let i_0 be the minimal such position, and let $i_0 < i_1 < i_2 < i_3 \ldots$ be the infinitely many positions greater than i_0 such that the node with name J is marked in $\mathbf{s}_{i_k}$, for all $k \geqslant 1$.

Let us define, for every $j \geqslant 1$ and for every $q \in \mathrm{Lab}(\mathbf{s}_{i_j}, J)$, the set $\mathrm{Acc}(q, j)$ by

$$\begin{aligned}\mathrm{Acc}(q, j) \stackrel{\text{def}}{=} \{q' \in F \mid{} & \text{there exists } i_{j-1} \leqslant k < i_j \text{ such that } q' \in \mathrm{Lab}(\mathbf{s}_k, J) \\ & \text{and } (q_k, \mathtt{a}_k, \Theta'_k, q_{k+1}) \ldots (q_{i_j - 1}, \mathtt{a}_{i_j - 1}, \Theta'_{i_j - 1}, q_{i_j}) \\ & \text{with } q_k = q', q_{i_j} = q \text{ is a finite run of } \mathbb{A} \text{ on } (\mathtt{a}_k, \boldsymbol{z}_k) \ldots (\mathtt{a}_{i_j}, \boldsymbol{z}_{i_j})\};\end{aligned}$$

and for every $j \geqslant 2$, every $i_{j-1} \leqslant k < i_j$ and every $q \in \mathrm{Lab}(\mathbf{s}_k, J) \cap F$, we define

$$\begin{aligned}\mathrm{Pre}(q, j, k) \stackrel{\text{def}}{=} \{q' \in \mathrm{Lab}(\mathbf{s}_{i_{j-1}}, J) \mid{} & \text{there exists some finite run } (q_{i_{j-1}}, \mathtt{a}_{i_{j-1}}, \Theta'_{i_{j-1}}, q_{i_{j-1}+1}) \\ & \ldots (q_{k-1}, \mathtt{a}_{k-1}, \Theta_{k-1}, q_k) \text{ with } q' = q_{i_{j-1}}, q = q_k \text{ of } \mathbb{A} \text{ on} \\ & (a_{i_{j-1}}, \boldsymbol{z}_{i_{j-1}}) \ldots (\mathtt{a}_k, \boldsymbol{z}_k)\}.\end{aligned}$$

In Appendix E.3, we prove that these sets $\mathrm{Acc}(q, j)$ and $\mathrm{Pre}(q, j, k)$ are nonempty. Next we define an infinite tree from which we will derive an initialized Büchi accepting run of $\mathbb{A}$ on w. Each node in this tree has a unique identifier in $\{Q_{\mathrm{in}}\} \cup \bigcup_{j \geqslant 1} I_{\mathrm{acc},j} \cup \bigcup_{j \geqslant 1} I_{\mathrm{mrkd},j}$, where

- $I_{\mathrm{acc},j} \stackrel{\text{def}}{=} \{(q, j) \mid q \in \mathrm{Acc}(q', j) \text{ for some } q' \in \mathrm{Lab}(\mathbf{s}_{i_j}, J)\}$,
- $I_{\mathrm{mrkd},j} \stackrel{\text{def}}{=} \{(q, q', j) \mid q \in \mathrm{Lab}(\mathbf{s}_{i_j}, J), q' \in \mathrm{Acc}(q, j)\}$.

The root node of the tree has identifier Q_{in}, and the root node is parent of a node $\mathbf{n}$ iff the identifier of the $\mathbf{n}$ is $(q_1, 1)$ for some $(q_1, 1) \in I_{\mathrm{acc},1}$. For all $j \geqslant 1$, node with identifier $(q_j, j) \in I_{\mathrm{acc},j}$ is parent of every node $\mathbf{n}$ iff the identifier of $\mathbf{n}$ is (q'_j, q_j, j) for some $(q'_j, q_j, j) \in I_{\mathrm{mrkd},j}$. Finally, for all $j \geqslant 1$, a node with identifier $(q'_j, q_j, j) \in I_{\mathrm{mrkd},j}$ is parent of a node $\mathbf{n}$ iff the identifier of $\mathbf{n}$ is $(q_{j+1}, j+1)$ for some $(q_{j+1}, j+1) \in I_{\mathrm{acc},j+1}$ such that $q'_j \in \mathrm{Pre}(q_{j+1}, j+1, k)$ for some $i_j \leqslant k < i_{j+1}$. Since the sets $\mathrm{Acc}(q, j)$ and $\mathrm{Pre}(q, j, k)$ are never empty, every node has at least one child, and every node has at most $\mathrm{card}(Q)^2$ child nodes. 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 $\mathbf{n}_0, \mathbf{n}_1, \mathbf{n}_2 \ldots$ By construction, we have

- $\mathbf{n}_0 = Q_{\mathrm{in}}$,
- $\mathbf{n}_1 = (p_1, 1)$ for some $p_1 \in \mathrm{Lab}(\mathbf{s}_{k_1}, J) \cap F$, $i_0 \leqslant k_1 < i_1$,
- $\mathbf{n}_2 = (p_2, p_1, 1)$ for some $p_2 \in \mathrm{Lab}(\mathbf{s}_{i_1}, J)$ such that $p_1 \in \mathrm{Acc}(p_2, 1)$,
- $\mathbf{n}_3 = (p_3, 2)$ for some $p_3 \in \mathrm{Lab}(\mathbf{s}_{k_2}, J) \cap F$, $i_1 \leqslant k_2 < i_2$, such that $p_2 \in \mathrm{Pre}(p_3, 2, k_2)$,
- $\mathbf{n}_4 = (p_4, p_3, 2)$ for some $p_4 \in \mathrm{Lab}(\mathbf{s}_{i_2}, J)$ such that $p_3 \in \mathrm{Acc}(p_4, 2)$,
- etc.

Let us argue that this yields an initialized Büchi-accepting run of $\mathbb{A}$ on w. One can show that there exists some initialized finite run from $q_{\mathrm{in}} \in Q_{\mathrm{in}}$ to p_1 of $\mathbb{A}$ on $(\mathtt{a}_1, \boldsymbol{z}_1) \ldots (\mathtt{a}_{k_1}, \boldsymbol{z}_{k_1})$

(as a consequence of Lemma E.1 in Appendix E.3). From $p_1 \in \mathrm{Acc}(p_2, 1)$, we obtain a run from p_1 to p_2 of $\mathbb{A}$ on $(\mathtt{a}_{k_1}, \boldsymbol{z}_{k_1}) \ldots (\mathtt{a}_{i_1}, \boldsymbol{z}_{i_1})$, etc. Putting the pieces together, we obtain an infinite initialized run of $\mathbb{A}$ on w. Since $p_1, p_3, \ldots$ are in F, the run is Büchi accepting.

We recall that $\mathrm{card}(\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)) \leqslant ((\mathbb{d}_\alpha - \mathbb{d}_1) + 3)^{2\beta} 3^{2\beta^2}$. 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 $\mathbb{d}_i$ and $\mathbb{d}_{i+1}$ for some i. So the above bound can be certainly improved but it is handful to use the set $\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ already defined in this document. □

As observed by one anonymous reviewer, an alternative approach for determinisation consists in viewing word constraint automata as Büchi automata (or Rabin automata) over types (made of constraints) and then determinize it using Safra's construction as a blackbox. Most probably, this approach would work by introducing the alphabet of satisfiable types; the semantics of Rabin word automata would then take care of which infinite sequences of satisfiable types is satisfiable. Instead, in this section, we worked directly with constraint word automata and we lifted arguments from Safra's thesis, apart from our handling of constraints. On the downside, this may not be optimal size-wise, but we prefer to follow our approach to work directly with constraint automata and to control most of the technical developments involved in the work.

8. Proving the correctness of the condition ($\bigstar^{\mathrm{C}}$)

This section is dedicated to the proof of Proposition 4.10 (see Section 4.2): for every regular locally consistent symbolic tree t, t satisfies ($\bigstar^{\mathrm{C}}$) iff t is satisfiable. This is all the more important as the condition ($\bigstar^{\mathrm{C}}$) is central in our paper. Below, we assume that $t : [0, D-1]^* \to \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ is a locally consistent symbolic tree.

Given a finite path $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} (\mathbf{n}_1, \mathtt{xd}_1) \cdots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathtt{xd}_n)$ in G^{C}_t, its *strict length*, written $\mathrm{slen}(\pi)$, is the number of edges labelled by '$<$' in π, i.e. $\mathrm{card}(\{i \in [1, n] \mid \sim_i$ equal to $<\})$. Given two nodes $(\mathbf{n}, \mathtt{xd})$ and $(\mathbf{n}', \mathtt{xd}')$, the *strict length* from $(\mathbf{n}, \mathtt{xd})$ to $(\mathbf{n}', \mathtt{xd}')$, written $\mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}'))$, is the supremum of all the strict lengths of paths from $(\mathbf{n}, \mathtt{xd})$ to $(\mathbf{n}', \mathtt{xd}')$. Though the strict length of any finite path is always finite, $\mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}'))$ may be infinite. Given $(\mathbf{n}, \mathtt{x}) \in U_{<\mathbb{d}_1}$, its *strict length*, written $\mathrm{slen}(\mathbf{n}, \mathtt{x})$, is defined as $\mathrm{slen}((\mathbf{n}, \mathtt{x}), (\mathbf{n}, \mathbb{d}_1))$. Given $(\mathbf{n}, \mathtt{x}) \in U_{>\mathbb{d}_\alpha}$, its *strict length*, written $\mathrm{slen}(\mathbf{n}, \mathtt{x})$, is defined as $\mathrm{slen}((\mathbf{n}, \mathbb{d}_\alpha), (\mathbf{n}, \mathtt{x}))$.

8.1. A simple characterisation for satisfiability.

First, we establish a few auxiliary results about G^{C}_t that are helpful in the sequel and that take advantage of the fact that every type in $\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ is satisfiable.

Lemma 8.1. *Let $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathtt{xd}_n)$ be a path in G^C_t such that $\mathbf{n}_0$ and $\mathbf{n}_n$ are neighbours. If $\sim_1 = \cdots = \sim_n$ is equal to '$=$', then $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{=} (\mathbf{n}_n, \mathtt{xd}_n)$, otherwise $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{<} (\mathbf{n}_n, \mathtt{xd}_n)$.*

See the proof in Appendix F.1. As a corollary, we get the following lemma.

Lemma 8.2. *Let $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathtt{xd}_n)$ be a path in G^C_t such that $(\mathbf{n}_0, \mathtt{xd}_0) = (\mathbf{n}_n, \mathtt{xd}_n)$. 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 $(\mathbf{n}, \mathbb{d}_1)$ and $(\mathbf{n}, \mathbb{d}_\alpha)$ in G_t^{C}.

Lemma 8.3. *Let* $t : [0, D-1]^* \to \Sigma \times \mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ *be a locally consistent symbolic tree. The statements below are equivalent.*

(I) : *t is satisfiable.*
(II): *For all* $(\mathbf{n}, \mathtt{x})$ *in* $U_{<\mathbb{d}_1} \cup U_{>\mathbb{d}_\alpha}$ *in* G_t^C, $\mathrm{slen}(\mathbf{n}, \mathtt{x}) < \omega$.

The proof can be found in Appendix F.2. We provide an elementary self-contained proof that is similar to the proof of [DD07, Lemma 7.1]. We remark that our proof does not use [CKL16, Lemma 34] as was done in the proof of [Lab21, Lemma 5.18]. Observe that G_t from [Lab21] does not contain constant nodes, which is problematic to apply [CKL16, Lemma 34]. Our proof of Lemma 8.3 is direct, although it would be possible to use [CKL16, Lemma 34] on our labelled graph G_t^{C}.

8.2. **Final steps in the proof of Proposition 4.10.** When t is regular and G_t^{C} has an element with an infinite strict length, the proof of Proposition 4.10 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_t^{C}. That is why, below, we introduce several restrictions on paths followed by forthcoming Lemma 8.4 stating the main properties we aim for. 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 $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathtt{xd}_n)$ is *direct* iff (1)–(3) below 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 $j \neq i$, $(\mathbf{n}_i, \mathtt{xd}_i) \neq (\mathbf{n}_j, \mathtt{xd}_j)$ (no element is visited twice, see Lemma 8.2),
(3) for all $i < j$, if $\mathbf{n}_i = \mathbf{n}_j$ and $n > 1$, then $<$ belongs to $\{\sim_{i+1}, \ldots, \sim_j\}$ (revisiting a node implies some progress in the strict length).

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

$$\mathrm{slen}^d((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}')) \leqslant \mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}')).$$

Similarly, a path $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathtt{xd}_n)$ is *rooted* iff all the $\mathbf{n}_i$'s are either descendants of the initial node $\mathbf{n}_0$, or equal to it (equivalently, the unique parent node of $\mathbf{n}_0$, if any, is never visited). We write $\mathrm{slen}^{dr}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}, \mathtt{xd}'))$ (same node $\mathbf{n}$ for both elements) to denote the strict length between $(\mathbf{n}, \mathtt{xd})$ and $(\mathbf{n}, \mathtt{xd}')$ based on *direct and rooted* paths only. This definition extends to $\mathrm{slen}^{dr}(\mathbf{n}, \mathtt{xd})$ with $(\mathbf{n}, \mathtt{xd}) \in (U_{<\mathbb{d}_1} \cup U_{>\mathbb{d}_\alpha})$. Obviously,

$$\mathrm{slen}^{dr}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}, \mathtt{xd}')) \leqslant \mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}, \mathtt{xd}')), \qquad \mathrm{slen}^{dr}(\mathbf{n}, \mathtt{xd}) \leqslant \mathrm{slen}(\mathbf{n}, \mathtt{xd}).$$

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 $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathtt{xd}_n)$ is ↺*-structured* iff there is $i \in [0, n]$ such that $\mathbf{n}_0 <_{\mathrm{pre}} \mathbf{n}_1 <_{\mathrm{pre}} \cdots <_{\mathrm{pre}} \mathbf{n}_i$ and $\mathbf{n}_n <_{\mathrm{pre}} \mathbf{n}_{n-1} <_{\mathrm{pre}} \cdots <_{\mathrm{pre}} \mathbf{n}_i$, where $<_{\mathrm{pre}}$ denotes the (strict) prefix relation (in this context,

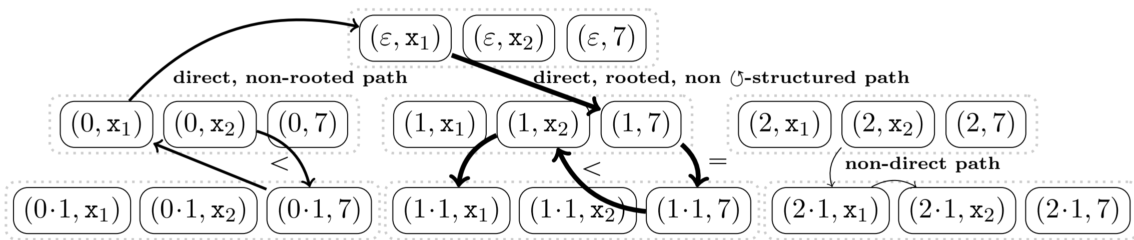


Figure 7. Different kinds of paths in G_t^{C}

we get the parent-child relation in the tree $[0, D-1]^*$). We write $\mathrm{slen}^{\circlearrowleft}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}, \mathtt{xd}'))$ (same node $\mathbf{n}$ for both elements) to denote the strict length between $(\mathbf{n}, \mathtt{xd})$ and $(\mathbf{n}, \mathtt{xd}')$ based on *direct, rooted and* $\circlearrowleft$*-structured* paths only. This definition extends to $\mathrm{slen}^{\circlearrowleft}(\mathbf{n}, \mathtt{xd})$ with $(\mathbf{n}, \mathtt{xd}) \in (U_{<\mathbb{d}_1} \cup U_{>\mathbb{d}_\alpha})$. Again, $\mathrm{slen}^{\circlearrowleft}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}, \mathtt{xd}')) \leqslant \mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}, \mathtt{xd}'))$ and $\mathrm{slen}^{\circlearrowleft}(\mathbf{n}, \mathtt{xd}) \leqslant \mathrm{slen}(\mathbf{n}, \mathtt{xd})$. In Figure 7, we illustrate the different kinds of paths on a subgraph of G_t^{C} with $\beta = 2$, $\alpha = 1$ and $\mathbb{d}_1 = 7$. The equivalence between (II)(1) and (II)(4) in Lemma 8.4 below, is the main property to prove Proposition 4.10. 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 8.4.

(I) : *If* $\mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}')) < \omega$*, then*

$$\frac{\mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}'))}{\beta + 2} \leqslant \mathrm{slen}^{d}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}')) \leqslant \mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}'))$$

(II): *The statements below are equivalent.*

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

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

$$\mathrm{slen}^{d}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}')) \leqslant \mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}')).$$

In order to show that

$$\frac{\mathrm{slen}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}'))}{\beta + 2} \leqslant \mathrm{slen}^{d}((\mathbf{n}, \mathtt{xd}), (\mathbf{n}', \mathtt{xd}')),$$

we establish that for any path $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathtt{xd}_n)$, there is a direct path π' from $(\mathbf{n}_0, \mathtt{xd}_0)$ to $(\mathbf{n}_n, \mathtt{xd}_n)$ with $\mathrm{slen}(\pi') \geqslant \frac{\mathrm{slen}(\pi)}{\beta+2}$.

Given a path π of the above form, we transform it a finite amount of times leading to a final path π'. If there are $i < j$ such that $\mathbf{n}_i = \mathbf{n}_{i+1} = \cdots = \mathbf{n}_j$ (and no way to extend the interval of indices $[i, j]$ while satisfying the sequence of equalities), we replace the subpath $(\mathbf{n}_i, \mathtt{xd}_i) \cdots (\mathbf{n}_j, \mathtt{xd}_j)$ in π by a shortcut. Note that $j - i \leqslant \beta + 1$ because $V_t = [0, D-1]^* \times \{\mathtt{x}_1, \ldots, \mathtt{x}_\beta\} \cup \{\mathbb{d}_1, \mathbb{d}_\alpha\}$ and $\beta + 2 = \mathrm{card}(\{\mathtt{x}_1, \ldots, \mathtt{x}_\beta\} \cup \{\mathbb{d}_1, \mathbb{d}_\alpha\})$. Several cases need to be distinguished.

- If $i > 0$, then $(\mathbf{n}_{i-1}, \mathtt{xd}_{i-1})$ and $(\mathbf{n}_j, \mathtt{xd}_j)$ are neighbours and therefore by Lemma 8.1, we can safely replace $(\mathbf{n}_{i-1}, \mathtt{xd}_{i-1}) \cdots (\mathbf{n}_j, \mathtt{xd}_j)$ by $(\mathbf{n}_{i-1}, \mathtt{xd}_{i-1}) \xrightarrow{\sim} (\mathbf{n}_j, \mathtt{xd}_j)$ for some $\sim \in \{=, <\}$ leading to a new value for π. The label $\sim$ on the edge depends whether $<$ occurs in the subpath from $(\mathbf{n}_{i-1}, \mathtt{xd}_{i-1})$ to $(\mathbf{n}_j, \mathtt{xd}_j)$. 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 $\mathbf{n}_i$) and divided by at most $\beta + 2$.
- If $i = 0$ and $j < n$, then $(\mathbf{n}_0, \mathtt{xd}_0)$ and $(\mathbf{n}_{j+1}, \mathtt{xd}_{j+1})$ are neighbours and therefore by Lemma 8.1, we can safely replace $(\mathbf{n}_0, \mathtt{xd}_0) \cdots (\mathbf{n}_{j+1}, \mathtt{xd}_{j+1})$ by $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim} (\mathbf{n}_{j+1}, \mathtt{xd}_{j+1})$ for some $\sim \in \{=, <\}$ leading to a new value for π. Again, the strict length of the new path is decreased by at most $\beta + 1$ and divided by at most $\beta + 2$.
- Finally, if $i = 0$ and $j = n$, then by Lemma 8.1, we replace $(\mathbf{n}_0, \mathtt{xd}_0) \cdots (\mathbf{n}_n, \mathtt{xd}_n)$ by $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim} (\mathbf{n}_n, \mathtt{xd}_n)$ for some $\sim \in \{=, <\}$ leading to a new value for π. Again, the strict length of the new path is decreased by at most $\beta + 1$ and divided by at most $\beta + 2$.

In the second step, we proceed as follows. If there are $i + 1 < j$ such that $(\mathbf{n}_i, \mathtt{xd}_i) = (\mathbf{n}_j, \mathtt{xd}_j)$ ($j - i > 1$ because otherwise we could apply the previous transformation), then by Lemma 8.2, the subpath $(\mathbf{n}_i, \mathtt{xd}_i) \cdots (\mathbf{n}_j, \mathtt{xd}_j)$ in π contains no edge labelled by $<$ and therefore by Lemma 8.1, we can safely replace $(\mathbf{n}_i, \mathtt{xd}_i) \cdots (\mathbf{n}_j, \mathtt{xd}_j)$ by $(\mathbf{n}_i, \mathtt{xd}_i)$, leading to a new value for π. 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 $\sim_{i+1} = \cdots = \sim_j$ is equal to '=', we can safely remove the subpath $(\mathbf{n}_i, \mathtt{xd}_i) \cdots (\mathbf{n}_j, \mathtt{xd}_j)$ by Lemma 8.1 along the lines of the previous transformations (easy details are omitted). We proceed as many times as necessary until the path π 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 π divided by $\beta + 2$. Indeed, in the first round of transformations, a subpath with $\beta + 2$ strict edges may be replaced by a path with only one strict edge.

(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 $(\mathbf{n}, \mathtt{x})$ be an element of G_t^{C} such that $\mathrm{slen}^d(\mathbf{n}, \mathtt{x}) = \omega$. The element $(\mathbf{n}, \mathtt{x})$ belongs to $U_{<\mathbb{d}_1} \cup U_{>\mathbb{d}_\alpha}$ in G_t^{C}. Suppose that $(\mathbf{n}, \mathtt{x})$ is in $U_{<\mathbb{d}_1}$ (we omit the other case as it admits a similar analysis). By definition of $\mathrm{slen}^d(\mathbf{n}, \mathtt{x})$, there is a family of direct paths $(\pi_i)_{i \in \mathbb{N}}$ from $(\mathbf{n}, \mathtt{x})$ to $(\mathbf{n}, \mathbb{d}_1)$ such that $\mathrm{slen}(\pi_i) \geqslant i$.

For each path in the family $(\pi_i)_{i \in \mathbb{N}}$, below, we define its *maximal entrance signature* as an element of the set $ES \stackrel{\text{def}}{=} [-1, D - 1] \times \mathtt{T}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)^2$. Since the set ES is finite, there is $(j, \mathtt{xd}, \mathtt{xd}') \in ES$ such that for infinitely many i, π_i has maximal entrance signature $(j, \mathtt{xd}, \mathtt{xd}')$, which shall allow us to conclude easily. Firstly, let us provide a few definitions.

Let $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \cdots (\mathbf{n}_m, \mathtt{xd}_m)$ be a path in $(\pi_i)_{i \in \mathbb{N}}$ of strict length at least $3(\beta + 1)$. This means that $(\mathbf{n}, \mathtt{x}) = (\mathbf{n}_0, \mathtt{xd}_0)$ and $(\mathbf{n}, \mathbb{d}_1) = (\mathbf{n}_m, \mathtt{xd}_m)$. Since π is direct, there are at most $\beta + 2$ positions in π visiting an element on the node $\mathbf{n}$ (called an $\mathbf{n}$-*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 π below:

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

The *entrance signature* of $\pi_h^\dagger$, written $es(\pi_h^\dagger)$, is the triple $(j, \mathtt{xd}, \mathtt{xd}') \in ES$ defined as follows:

- $\mathtt{xd} \stackrel{\text{def}}{=} \mathtt{xd}_{i_h+1}$ (entrance term) and $\mathtt{xd}' \stackrel{\text{def}}{=} \mathtt{xd}_{i_{h+1}-1}$ (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 $j \stackrel{\text{def}}{=} -1$. The value j is a direction, where -1 is intended to mean "to the parent node" and $j \in [0, D-1]$ "to the j^{th} child node". Note also that in the degenerate case $i_h + 1 = i_{h+1} - 1$, the path $\pi^\dagger_h$ is made of a single element and $\mathtt{xd} = \mathtt{xd}'$.

A few useful properties are worth being stated.

- $\mathbf{n}_{i_h+1} = \mathbf{n}_{i_{h+1}-1}$, $\pi^\dagger_h$ does not contain an $\mathbf{n}$-element and $\pi^\dagger_h$ is rooted.
- By Lemma 8.1, $(\mathbf{n}, \mathtt{x}) \xrightarrow{\sim} (\mathbf{n}_{i_h+1}, \mathtt{xd}_{i_h+1})$ and $(\mathbf{n}_{i_{h+1}-1}, \mathtt{xd}_{i_{h+1}-1}) \xrightarrow{\sim'} (\mathbf{n}, \mathbb{d}_1)$ for some $\sim, \sim' \in \{<, =\}$.
- For all $h \neq h'$, if $es(\pi^\dagger_h) = (j_1, \mathtt{xd}_1, \mathtt{xd}'_1)$, $es(\pi^\dagger_{h'}) = (j_2, \mathtt{xd}_2, \mathtt{xd}'_2)$ and $j_1 = j_2$, then $\mathtt{xd}_1 \neq \mathtt{xd}'_1$ and $\mathtt{xd}_2 \neq \mathtt{xd}'_2$ (π is direct). As a consequence, $s \leqslant \beta + 1$ and there are no $\pi^\dagger_h$ and $\pi^\dagger_{h'}$ with $h \neq h'$ having the same entrance signature.

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

$$\left\lceil \frac{\mathrm{slen}(\pi) - 2(\beta+1)}{(\beta+1)} \right\rceil$$

When $\mathrm{slen}(\pi) \geqslant 3(\beta+1)$, the above value is at least one. Indeed, we subtract $2(\beta+1)$ from $\mathrm{slen}(\pi)$ to take into account the edges from some $\mathbf{n}$-element to the elements $(\mathbf{n}_{i_h+1}, \mathtt{xd}_{i_h+1})$, and from the elements $(\mathbf{n}_{i_{h+1}-1}, \mathtt{xd}_{i_{h+1}-1})$ to some $\mathbf{n}$-element. The maximal entrance of π is defined as an entrance signature $es(\pi^\dagger_h)$ such that $\mathrm{slen}(\pi^\dagger_h)$ is maximal (we can also fix an arbitrary linear ordering on ES in case maximality of the – direct – strict lengths is witnessed strictly more than once). As announced earlier, there is $(j, \mathtt{xd}, \mathtt{xd}') \in ES$ such that for infinitely many i (say belonging to the infinite set $X \subseteq [3(\beta+1), +\infty)$), π_i has maximal type $(j, \mathtt{xd}, \mathtt{xd}')$.

(a) If $j \in [0, D-1]$, we have $\mathrm{slen}^{dr}((\mathbf{n} \cdot j, \mathtt{xd}), (\mathbf{n} \cdot j, \mathtt{xd}')) = \omega$ because
$$\mathrm{slen}^{dr}((\mathbf{n} \cdot j, \mathtt{xd}), (\mathbf{n} \cdot j, \mathtt{xd}')) \geqslant \sup_{i \in X} \left\lceil \frac{i - 2(\beta+1)}{(\beta+1)} \right\rceil$$
Since $(\mathbf{n}, \mathtt{x}) \xrightarrow{\sim} (\mathbf{n} \cdot j, \mathtt{xd})$ and $(\mathbf{n} \cdot j, \mathtt{xd}') \xrightarrow{\sim'} (\mathbf{n}, \mathbb{d}_1)$ for some $\sim, \sim' \in \{<, =\}$, we get that $\mathrm{slen}^{dr}(\mathbf{n}, \mathtt{x}) = \omega$ too.

(b) If $j = -1$, then let $\mathbf{n}'$ be the unique parent of $\mathbf{n}$. We have $\mathrm{slen}^{d}((\mathbf{n}', \mathtt{xd}), (\mathbf{n}', \mathtt{xd}')) = \omega$ (the maximal entrance signature is $(j, \mathtt{xd}, \mathtt{xd}')$) and $(\mathbf{n}', \mathtt{xd}') \xrightarrow{\sim} (\mathbf{n}, \mathbb{d}_1)$ for some $\sim \in \{<, =\}$. Since $\mathbf{n}$ and $\mathbf{n}'$ are neighbour nodes, we also get that $(\mathbf{n}', \mathtt{xd}') \xrightarrow{\sim} (\mathbf{n}', \mathbb{d}_1)$ by construction of G^{C}_t. Moreover, $\mathtt{xd}$ is necessarily a variable. Indeed, by Lemma 8.1, $\mathtt{xd}$ is distinct from $\mathbb{d}_1$ and by satisfiability of the elements in $\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$, $\mathtt{xd}$ is distinct from $\mathbb{d}_\alpha$. Consequently, $\mathrm{slen}^{d}(\mathbf{n}', \mathtt{xd}) = \omega$ 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 $(\mathbf{m}, \mathtt{y})$ such that $\mathrm{slen}^{dr}(\mathbf{m}, \mathtt{y}) = \omega$. Another way to proceed would be 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 $(\mathbf{n}, \mathtt{x})$ be an element of G^{C}_t such that $\mathrm{slen}^{dr}(\mathbf{n}, \mathtt{x}) = \omega$. The element $(\mathbf{n}, \mathtt{x})$ belongs to $U_{<\mathbb{d}_1} \cup U_{>\mathbb{d}_\alpha}$ in G^{C}_t. Suppose that $(\mathbf{n}, \mathtt{x})$ is in $U_{>\mathbb{d}_\alpha}$ (we omit the other case as it admits a similar analysis). By definition of $\mathrm{slen}^{dr}(\mathbf{n}, \mathtt{x})$, there is a family of direct and rooted paths $(\pi_i)_{i \in \mathbb{N}}$ from $(\mathbf{n}, \mathbb{d}_\alpha)$ to $(\mathbf{n}, \mathtt{x})$ such that $\mathrm{slen}(\pi_i) \geqslant i$. Below, we show that $\mathrm{slen}^{\circlearrowleft}(\mathbf{n}, \mathtt{x}) = \omega$ (and therefore no need to witness unbounded strict length 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 $\circlearrowleft$-structured path from $(\mathbf{n}, \mathbb{d}_\alpha)$ to $(\mathbf{n}, \mathtt{x})$ 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 $j \in \mathbb{N}$, $L_{i,j+1} \stackrel{\text{def}}{=} \left\lceil \frac{L_{i,j} - 2(\beta+1)}{(\beta+1)} \right\rceil$.

Here are properties that can be easily shown.

(PL1): For all $i < i'$ and j, $L_{i,j} \leqslant L_{i',j}$.
(PL2): For all i and $j < j'$, $L_{i,j} \geqslant L_{i,j'}$.

Moreover, partly based on (PL1), 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 is true for j, i.e. $\sup\{L_{i,j} \mid i \in \mathbb{N}\} = \omega$. Substracting a constant and dividing by a constant preserves the limit behavior, consequently $\sup\{\frac{L_{i,j} - 2(\beta+1)}{(\beta+1)} \mid i \in \mathbb{N}\} = \omega$ too 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 $L_{N_j,j} \geqslant 1$. For instance, N_0 can take the value 1, N_1 the value $3(\beta+1)$ and N_2 the value $3(\beta+1)^2 + 2(\beta+1)$. However, we shall require a bit more properties about the N_j's using simple properties about $(L_{i,j})_{i,j \in \mathbb{N}}$.

(PL3): For all j, $N_{j+1} > N_j \geqslant 2$ and $\left\lceil \frac{N_{j+1} - 2(\beta+1)}{(\beta+1)} \right\rceil \geqslant N_j$.
(PL4): For all $M, j \in \mathbb{N}$, $M \geqslant N_j$ implies (if $M, j \geqslant 1$, then $M - 1 \geqslant N_{j-1}$ and if $M, j \geqslant 2$, then $M - 2 \geqslant N_{j-2}$).

Thanks to (PL1), we can choose the N_j's as large as possible, which allows us to establish (PL3). By contrast, the satisfaction of (PL4) is a consequence of (PL3). By way of example, $M, j \geqslant 1$ and $M \geqslant N_j$ imply $M \geqslant N_j \geqslant N_{j-1} + 1$ and therefore $M - 1 \geqslant N_{j-1}$.

Let $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_m} (\mathbf{n}_m, \mathtt{xd}_m)$ be a direct and rooted path from $(\mathbf{n}, \mathbb{d}_\alpha)$ to $(\mathbf{n}, \mathtt{x})$ of strict length at least N_j for some $j \geqslant 1$. By (PL3), we have $\left\lceil \frac{N_j - 2(\beta+1)}{(\beta+1)} \right\rceil \geqslant N_{j-1} \geqslant 2$. For instance, π can take the value π_{N_j} from the family $(\pi_i)_{i \in \mathbb{N}}$. In the developments below, we build a path π' from $(\mathbf{n}, \mathbb{d}_\alpha)$ to $(\mathbf{n}, \mathtt{x})$ that is of strict length at least j and that is direct, rooted and $\circlearrowleft$-structured. To do so, we maintain three auxiliary paths while guaranteeing the satisfaction of an invariant.

- π_{des} is a descending and direct path from $(\mathbf{n}, \mathbb{d}_\alpha)$. Its initial value is $(\mathbf{n}, \mathbb{d}_\alpha)$.
- π_{asc} is an ascending and direct path to $(\mathbf{n}, \mathtt{x})$. Its initial value is $(\mathbf{n}, \mathtt{x})$.
- $\pi^\dagger$ is a direct and rooted path from the last element of π_{des} (say $(\mathbf{n}^\dagger, \mathtt{xd}_1^\dagger)$) to the first element of π_{asc} (say $(\mathbf{n}^\dagger, \mathtt{xd}_2^\dagger)$) and is a (consecutive) subpath of the initial path π. The first and last elements belong therefore to the same node $\mathbf{n}^\dagger$. The initial value for $\pi^\dagger$ is π. By slight abuse, below we assume that $\pi^\dagger$ can be also written $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_m} (\mathbf{n}_m, \mathtt{xd}_m)$.
- Let C be equal to $\mathrm{slen}(\pi_{des}) + \mathrm{slen}(\pi_{asc})$. If $C \geqslant j$, then there is a direct, rooted and $\circlearrowleft$-structured path from $(\mathbf{n}, \mathbb{d}_\alpha)$ to $(\mathbf{n}, \mathtt{x})$ passing via $(\mathbf{n}^\dagger, \mathtt{xd}_1^\dagger)$. It uses the edges from π_{des} and π_{asc}. Otherwise ($C < j$), we maintain $\mathrm{slen}(\pi^\dagger) \geqslant N_{j-C}$. Consequently, if $\pi^\dagger$ is made of elements from the same node, then $\mathrm{slen}(\pi^\dagger) \leqslant 1$ and therefore $C \geqslant j$.

We transform these three paths π_{des}, π_{asc} and $\pi^\dagger$ 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, $\mathrm{slen}(\pi^\dagger)$ and C are unchanged). Initially, we have $\mathrm{slen}(\pi^\dagger) \geqslant N_{j-C}$ because $C = 0 < j$, $\pi^\dagger = \pi$ and $\mathrm{slen}(\pi) \geqslant N_j$.

Let us define the *maximal entrance signature* of $\pi^\dagger$ similarly to what is done earlier. Here, $ES \stackrel{\text{def}}{=} [0, D-1] \times \mathtt{T}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)^2$ (no value -1 in the interval 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 $i_0 < i_1 < \cdots < i_s$ with $i_0 = 0$, $i_s = m$ (and $s \leqslant (\beta+1)$). For each $h \in [0, s-1]$, we write $\pi_h^{\dagger\dagger}$ to denote the subpath of $\pi^\dagger$ below:

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

The *entrance signature* of $\pi_h^{\dagger\dagger}$, written $es(\pi_h^\dagger)$, is the triple $(j, \mathtt{xd}, \mathtt{xd}') \in ES$ defined as follows: $\mathtt{xd} \stackrel{\text{def}}{=} \mathtt{xd}_{i_h+1}$, $\mathtt{xd}' \stackrel{\text{def}}{=} \mathtt{xd}_{i_{h+1}-1}$ 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 8.1, $(\mathbf{n}^\dagger, \mathtt{xd}_1^\dagger) \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_{i_h+1}, \mathtt{xd}_{i_q+1})$ for some $\sim_{1,h}^\dagger \in \{<, =\}$ and

  $(\mathbf{n}_{i_{h+1}-1}, \mathtt{xd}_{i_{h+1}-1}) \xrightarrow{\sim_{2,h}^\dagger} (\mathbf{n}^\dagger, \mathtt{xd}_2^\dagger)$ for some $\sim_{2,h}^\dagger \in \{<, =\}$.

There is h such that the strict length of $\pi_h^{\dagger\dagger}$ is at least $\lceil \frac{\mathrm{slen}(\pi^\dagger) - 2(\beta+1)}{(\beta+1)} \rceil$. Indeed, $s \leqslant \beta + 1$ and there are at most $\beta + 1$ paths of the form $\pi_h^{\dagger\dagger}$ with at most $2(\beta + 1)$ connecting edges.

- 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.
  - π_{des} becomes $\pi_{des} \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_1, \mathtt{xd}_1)$, π_{asc} becomes $(\mathbf{n}_{m-1}, \mathtt{xd}_{m-1}) \xrightarrow{\sim_{2,h}^\dagger} \pi_{asc}$.
  - $\pi^\dagger$ 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 (PL4), the invariant is maintained. If $\pi_0^{\dagger\dagger}$ is made of elements from the same node, $\mathrm{slen}(\pi_0^{\dagger\dagger}) \leqslant 1$ because π is direct and $C \geqslant j$. The final direct, rooted and $\circlearrowleft$-structured path is $\pi_{des} \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_1, \mathtt{xd}_1) \xrightarrow{\sim} (\mathbf{n}_{m-1}, \mathtt{xd}_{m-1}) \xrightarrow{\sim_{2,h}^\dagger} \pi_{asc}$ and its strict length is at least j.
- Otherwise, i.e. $s > 1$. Pick $h \in [0, s-1]$ such that

$$\mathrm{slen}(\pi_h^{\dagger\dagger}) \geqslant \left\lceil \frac{\mathrm{slen}(\pi^\dagger) - 2(\beta+1)}{(\beta+1)} \right\rceil$$

  Note that $(\mathbf{n}^\dagger, \mathtt{xd}_1^\dagger) \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_{i_h+1}, \mathtt{xd}_{i_h+1})$ and $(\mathbf{n}_{i_{h+1}-1}, \mathtt{xd}_{i_{h+1}-1}) \xrightarrow{\sim_{2,h}^\dagger} (\mathbf{n}^\dagger, \mathtt{xd}_2^\dagger)$ 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. π_{des} becomes $\pi_{des} \xrightarrow{\sim_{1,h}^\dagger} (\mathbf{n}_{i_h+1}, \mathtt{xd}_{i_h+1})$, π_{asc} becomes $(\mathbf{n}_{i_{h+1}-1}, \mathtt{xd}_{i_{h+1}-1}) \xrightarrow{\sim_{2,h}^\dagger} \pi_{asc}$ and $\pi^\dagger$ becomes $\pi_h^{\dagger\dagger}$. From $\mathrm{slen}(\pi^\dagger) \geqslant N_{j-C}$, we get

$$\mathrm{slen}(\pi_h^{\dagger\dagger}) \geqslant \left\lceil \frac{\mathrm{slen}(\pi^\dagger) - 2(\beta+1)}{(\beta+1)} \right\rceil \geqslant \underbrace{N_{j-C-1} > N_{j-C-2}}_{\text{by (PL3)}}$$

by the invariant, (PL3) and (PL1) (assuming that $j - C - 2 \geqslant 0$, otherwise we remove the appropriate expressions). Indeed, by the satisfaction of the invariant, we have $\mathrm{slen}(\pi^\dagger) \geqslant$

N_{j-C} and

$$\overbrace{\left\lceil \frac{\mathrm{slen}(\pi^\dagger) - 2(\beta+1)}{(\beta+1)} \right\rceil \geqslant \left\lceil \frac{N_{j-C} - 2(\beta+1)}{(\beta+1)} \right\rceil}^{\text{by (PL1)}} \geqslant N_{j-C-1}$$

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 $\circlearrowleft$-structured path from $(\mathbf{n}, \mathbb{d}_\alpha)$ to $(\mathbf{n}, \mathtt{x})$ that is of strict length at least j. □

Here is the final step to prove Proposition 4.10. Since the violation of ($\bigstar^{\mathrm{C}}$) 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 4.10) Let t be a regular locally consistent symbolic tree. We write N to denote the number of distinct subtrees in t (N exists because t is regular). It is easy to show that the satisfiability of t implies that t satisfies ($\bigstar^{\mathrm{C}}$). Indeed, if t does not satisfy ($\bigstar^{\mathrm{C}}$), then the existence of the witness path map p and the reverse path map rp forbids the possibility to interpret the variables so that t is satisfiable (by Lemma 4.9). The main part of the proof consists in showing that if t is not satisfiable, then G^{C}_t does not satisfy ($\bigstar^{\mathrm{C}}$). By Lemma 8.3, there is $(\mathbf{n}, \mathtt{x})$ in $U_{<\mathbb{d}_1} \cup U_{>\mathbb{d}_\alpha}$ in G^{C}_t, such that $\mathrm{slen}(\mathbf{n}, \mathtt{x}) = \omega$. By Lemma 8.4, there is $(\mathbf{n}, \mathtt{x})$ in $U_{<\mathbb{d}_1} \cup U_{>\mathbb{d}_\alpha}$ in G^{C}_t, such that $\mathrm{slen}^{\circlearrowleft}(\mathbf{n}, \mathtt{x}) = \omega$. Suppose that $(\mathbf{n}, \mathtt{x})$ belongs to $U_{>\mathbb{d}_\alpha}$ in G^{C}_t (the other case is very similar, and is omitted).

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

$$(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} (\mathbf{n}_1, \mathtt{xd}_1) \cdots \xrightarrow{\sim_\ell} (\mathbf{n}_\ell, \mathtt{xd}_\ell) = (\mathbf{n}_\ell, \mathtt{xd}'_\ell) \xrightarrow{\sim'_\ell} (\mathbf{n}_{\ell-1}, \mathtt{xd}'_{\ell-1}) \cdots \xrightarrow{\sim'_1} (\mathbf{n}_0, \mathtt{xd}'_0),$$

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

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

Since $\mathrm{slen}(\pi) \geqslant 2((\beta+2)^2 N + 1)$, one of the paths among $(\mathbf{n}_0, \mathtt{xd}_0) \cdots (\mathbf{n}_\ell, \mathtt{xd}_\ell)$ and $(\mathbf{n}_\ell, \mathtt{xd}'_\ell) \cdots (\mathbf{n}_0, \mathtt{xd}'_0)$ has strict length at least $(\beta+2)^2 N + 1$. Below, we assume that the strict length of $(\mathbf{n}_0, \mathtt{xd}_0) \cdots (\mathbf{n}_\ell, \mathtt{xd}_\ell)$ is at least $(\beta+2)^2 N + 1$ (the other case is very 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 $\mathbf{n}_{K'}$, $\mathtt{xd}_K = \mathtt{xd}_{K'}$ and $\mathtt{xd}'_K = \mathtt{xd}'_{K'}$ and the (consecutive) subpath $(\mathbf{n}_K, \mathtt{xd}_K) \cdots (\mathbf{n}_{K'}, \mathtt{xd}_{K'})$ has strict length at least one. Let us provide a bit more details about this claim. The strict length of the descending path $\pi^\dagger = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} (\mathbf{n}_1, \mathtt{xd}_1) \cdots \xrightarrow{\sim_\ell} (\mathbf{n}_\ell, \mathtt{xd}_\ell)$ is at least $L = (\beta+2)^2 N + 1$. Both $(\mathbf{n}_K, \mathtt{xd}_K)$ and $(\mathbf{n}_{K'}, \mathtt{xd}_{K'})$ are claimed to belong to this descending path. Let us see why exactly. The path $\pi^\dagger$ can be decomposed as a sequence of consecutive subpaths such that $\pi^\dagger = \cdots \pi^\dagger_1 \cdots \pi^\dagger_L \cdots$ where each $\pi^\dagger_i$ contains a single strict edge (i.e., labelled by '$<$'), say its first edge. So, let us say that $\pi^\dagger_i$ starts by the elements $(\mathbf{n}_{s_i}, \mathtt{xd}_{s_i})$. Hence, if $i < i'$, then there is a strict path from $(\mathbf{n}_{s_i}, \mathtt{xd}_{s_i})$ to $(\mathbf{n}_{s_{i'}}, \mathtt{xd}_{s_{i'}})$. Let us define that the subtree signature of $(\mathbf{n}_{s_i}, \mathtt{xd}_{s_i})$ is a triple made of the subtree at $\mathbf{n}_{s_i}$ from the regular tree (a finite amount of possibilities), $\mathtt{xd}_{s_i}$ and $\mathtt{xd}'_{s_i}$ on the return ascending path at the node $\mathbf{n}_{s_i}$ (unique visit in the ascending part). By the Pigeonhole Principle, there are $K < K'$ such that $(\mathbf{n}_{s_K}, \mathtt{xd}_{s_K})$ and $(\mathbf{n}_{s'_K}, \mathtt{xd}_{s'_K})$ have the same subtree signature.

Observe that by Lemma 8.1, for every $k \in [K, K']$, we have $(\mathbf{n}_k, \mathtt{xd}_k) \xrightarrow{\leq} (\mathbf{n}_k, \mathtt{xd}'_k)$. Moreover, since $t_{|\mathbf{n}_K} = t_{|\mathbf{n}_{K'}}$ (same subtree), for all $(\mathbf{m}_1, \mathtt{xd}^*_1), (\mathbf{m}_2, \mathtt{xd}^*_2) \in \{\mathbf{n}_K, \ldots, \mathbf{n}_{K'-1}\} \times \mathtt{T}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$, for all $i \in \mathbb{N}$ and $\sim \in \{<, =\}$,

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

$$(\mathbf{m}_1 \cdot (j_{K+1} \cdots j_{K'})^i (j_{K+1} \cdots j_{K+m_1}), \mathtt{xd}^*_1) \xrightarrow{\sim} (\mathbf{m}_2 \cdot (j_{K+1} \cdots j_{K'})^i (j_{K+1} \cdots j_{K+m_2}), \mathtt{xd}^*_2),$$

with $\mathbf{m}_1 = \mathbf{n}_K \cdot j_{K+1} \cdots j_{K+m_1}$, $\mathbf{m}_2 = \mathbf{n}_K \cdot j_{K+1} \cdots j_{K+m_2}$ and by convention if $m = 0$, then $j_{K+1} \cdots j_{K+m} = \varepsilon$.

Let us define $(\mathbf{n}, \mathtt{xd})$, $(\mathbf{n}, \mathtt{xd}')$, $\mathcal{B}$, p and rp that witness the violation of the condition ($\bigstar^{\mathrm{C}}$) (see Section 4.2).

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

One can check that all the properties hold to violate ($\bigstar^{\mathrm{C}}$), in particular, p is strict and for all i, $p(i) \xrightarrow{\leq} rp(i)$. Indeed, let us explain a bit more why the conditions for violating ($\bigstar^{\mathrm{C}}$) are meet.

- The path map p can be written as follows:

$$(\mathbf{n}_K, \mathtt{xd}_K) \xrightarrow{\sim_{K+1}} \cdots \xrightarrow{\sim_{K'}} (\mathbf{n}_K (j_{K+1} \cdots j_{K'}), \mathtt{xd}_K) \xrightarrow{\sim_{K+1}} (\mathbf{n}_K (j_{K+1} \cdots j_{K'}) \cdot j_{K+1}, \mathtt{xd}_{K+1}) \cdots \xrightarrow{\sim_{K'}}$$

$$(\mathbf{n}_K \cdot (j_{K+1} \cdots j_{K'})^2, \mathtt{xd}_K) \xrightarrow{\sim_{K+1}} (\mathbf{n}_K \cdot (j_{K+1} \cdots j_{K'})^2 \cdot j_{K+1}, \mathtt{xd}_{K+1}) \cdots \xrightarrow{\sim_{K'}} (\mathbf{n}_K \cdot (j_{K+1} \cdots j_{K'})^3, \mathtt{xd}_K) \cdots$$

  Note that this is a valid path map because $\mathbf{n}_K$ and $\mathbf{n}'_K$ have the same subtree. Moreover p is strict, i.e. edges of the form $\xrightarrow{\leq}$ occur infinitely often because the strict length of the path $(\mathbf{n}_K, \mathtt{xd}_K) \xrightarrow{\sim_{K+1}} \cdots \xrightarrow{\sim_{K'}} (\mathbf{n}_{K'}, \mathtt{xd}_{K'})$ is at least one.
- Similarly, the reverse path map rp can be written as follows:

$$(\mathbf{n}_K, \mathtt{xd}'_K) \xrightarrow{\overline{\sim'_{K+1}}} \cdots \xrightarrow{\overline{\sim'_{K'}}} (\mathbf{n}_K (j_{K+1} \cdots j_{K'}), \mathtt{xd}'_K) \xrightarrow{\overline{\sim'_{K+1}}} (\mathbf{n}_K (j_{K+1} \cdots j_{K'}) \cdot j_{K+1}, \mathtt{xd}'_{K+1}) \cdots \xrightarrow{\overline{\sim'_{K'}}}$$

$$(\mathbf{n}_K \cdot (j_{K+1} \cdots j_{K'})^2, \mathtt{xd}'_K) \xrightarrow{\overline{\sim'_{K+1}}} (\mathbf{n}_K (j_{K+1} \cdots j_{K'})^2 \cdot j_{K+1}, \mathtt{xd}'_{K+1}) \cdots \xrightarrow{\overline{\sim'_{K'}}} (\mathbf{n}_K \cdot (j_{K+1} \cdots j_{K'})^3, \mathtt{xd}'_K) \cdots$$

  where $\overline{=}$ ("reverse of $=$") is $=$ and $\overline{<}$ ("reverse of $<$") is $>$. Note that this is a valid reverse path map ($\mathbf{n}_K$ and $\mathbf{n}'_K$ have the same subtree).
- Concerning the fourth condition for violating ($\bigstar^{\mathrm{C}}$), namely for all i, $p(i) \xrightarrow{\leq} rp(i)$, we have already observed that by Lemma 8.1, for every $k \in [K, K']$, we have $(\mathbf{n}_k, \mathtt{xd}_k) \xrightarrow{\leq} (\mathbf{n}_k, \mathtt{xd}'_k)$. Since $\mathbf{n}_K$ and $\mathbf{n}'_K$ have the same subtree, for all $i \in \mathbb{N}$, for all $k \in [K, K']$, $(\mathbf{n}_K \cdot (j_{K+1} \cdots j_{K'})^i \cdot (j_{K+1} \cdots j_k), \mathtt{xd}_k) \xrightarrow{\leq} (\mathbf{n}_K (j_{K+1} \cdots j_{K'})^i \cdot (j_{K+1} \cdots j_k), \mathtt{xd}'_k)$. □

9. Related Work

This section is dedicated to related work and more specifically to results about temporal logics with numerical values (Section 9.1) and some technical differences with [LOS20, Lab21] (Section 9.2 and Section 9.3). It goes a bit further than what can be found in the rest of the document but, of course, related work is also discussed in other sections.

9.1. **Model-checking problem.** A problem related to satisfiability is the *model-checking problem* (bibliographical references about satisfiability can be mainly found in Sections 1-2). Fragments of the model-checking problem involving a temporal logic similar to CTL*($\mathbb{Z}$) are investigated in [Čer94, BG06, BP14, FMW22a] (see also [GHOW12, CKP15, Ves15, ADG20]). However, model-checking problems with CTL*($\mathbb{Z}$)-like languages are easily undecidable, see e.g. [Čer94, Theorem 1] and [MT16, Theorem 4.1] (more general constraints are used in [MT16] but the undecidability proof uses only the constraints involved herein). The difference between model-checking and satisfiability is subtle and underlines that the decidability/complexity of CTL($\mathbb{Z}$)/CTL*($\mathbb{Z}$) satisfiability is not immediate. For instance, a reduction following the undecidability proof in [MT13] fails: tree models satisfying a similar formula as in [MT13] can still "cheat" as one cannot express that *it is not possible* to move to a state detecting cheating.

9.2. **($\star^{\mathrm{C}}$): variant of ($\star$).** Proposition 4.10 is a variant of [LOS20, Lemma 22]. Let us explain the improvement of our developments compared to what is done in [LOS20, Lab21]. The *framified constraint graphs* defined in [LOS20, Definition 14] correspond to the graph G_t^{C} without the elements in $[0, D-1]^* \times \{\mathbb{d}_1, \mathbb{d}_\alpha\}$ and corresponding edges. However, Example 4.11 illustrates the importance of taking into account these elements when deciding satisfiability (without $\mathbb{d}_1$, the graph would satisfy ($\star^{\mathrm{C}}$)). Actually, Example 4.11 invalidates ($\star$) as used in [LOS20, Lab21] because the constants are missing to apply properly results from [CKL16]. The problematic part in [LOS20, Lab21] is due to the proof of [Lab21, Lemma 5.18] whose main argument takes advantage of [CKL16] but without the elements related to constant values (see also [DG08, Lemma 8]). Note also that 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]). A condition similar to ($\star$) is also introduced recently in [BP24] (conference version in [BP22]) to decide the single-sided realizability problem based on LTL($\mathbb{Z}, <, =$) (without constant values). Though the technical developments are presented quite differently and the settings and purposes are distinct, one can establish interesting connections. For instance, there are correspondences between [BP24, Lemma 23] and Lemma 8.3, between [BP24, Lemma 26] and Lemma 4.12, and between the construction of tree automata in [BP24, Section 7] and what is done in the paper to build Rabin tree automata in order to analyze the computational complexity of the decision problems. It remains open how to use the best of the two papers for further analysis, for instance how to take advantage of our results on constraint automata? It is also possible to find relationships between techniques developed for lock-sharing systems in [MMW23] and what we developed herein.

Besides, our proof of Lemma 4.12 also proposes a slight novelty compared to the construction in [LOS20]: we design $\mathbb{A}_{\star^{\mathrm{C}}}$ without firstly constructing *a tree automaton for the complement language* (as done in [LOS20, Section 3.4]) and then using results from [MS95] (elimination of alternation in tree automata). Our new path happens to be rewarding: not only we can better understand how to express the condition ($\star^{\mathrm{C}}$), but also we control the size parameters of $\mathbb{A}_{\star^{\mathrm{C}}}$ involved in our complexity analysis. This may prove useful when implementing the decision procedure for solving the satisfiability problem for CTL($\mathbb{Z}$) (resp. for CTL*($\mathbb{Z}$)).

9.3. **Differences with [LOS20] about the number of Rabin pairs.** To conclude this section about related works, we would like to draw the attention of the reader that Lemma 4.12 is similar to [LOS20, Proposition 26] but there is an essential difference: the number of Rabin pairs in Lemma 4.12 is not a constant but a value depending on β, an outcome of our investigations. Note also that the question on the number of Rabin pairs discussed herein is independent from the question discussed above about having the elements in $[0, D-1]^* \times \{\mathbb{d}_1, \mathbb{d}_\alpha\}$ within G_t^{C}. It is important to know the number of Rabin pairs in $A_{\bigstar^{\mathrm{C}}}$ for our complexity analysis as checking nonemptiness of Rabin tree automata is *exponential* in the number of Rabin pairs [EJ00, Theorem 4.1]. It 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, for binary trees, 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 branching degree $D \geqslant 1$ is arbitrary in our investigations. That is why, for the proof of Lemma 4.15 we used that nonemptiness can be checked in time $(\mathrm{card}(\delta) \times \gamma \times N)^{\mathcal{O}(N)}$ (D is taken into account in $\mathrm{card}(\delta)$). This makes a difference with the argument used to establish [LOS20, Proposition 27].

As a conclusion, our analysis about the number of Rabin pairs, and the complexity measure to test nonemptiness of Rabin tree automata differs slightly from what is used in [LOS20]. This lead us to establish the complexity measures sometimes quite pedantically. However, this was for the sake of guaranteing that, indeed, nonemptiness of $\mathrm{L}(A_{\mathrm{cons}(\mathbb{A})}) \cap \mathrm{L}(A_{\bigstar^{\mathrm{C}}})$ can be tested in exponential-time, as stated in Lemma 4.15.

10. Concluding Remarks

We developed an automata-based approach to solve SAT(CTL($\mathbb{Z}$)) and SAT(CTL*($\mathbb{Z}$)), by introducing tree constraint automata that accept infinite data trees with data domain $\mathbb{Z}$. The nonemptiness problem for tree constraint automata with Büchi acceptance conditions (resp. with Rabin pairs) is ExpTime-complete, see Theorem 4.16 (resp. Theorem 4.19). The difficult part consists in proving the ExpTime-membership for which we show how to substantially adapt the material in [Lab21, Section 5.2] that guided us to design the correctness proof of ($\bigstar^{\mathrm{C}}$). The work [LOS20] was indeed a great inspiration but we adjusted a few statements from there. We recall that ($\bigstar$) in [LOS20] is not fully correct (see Section 9.2) as we need to add constants (leading to the variant condition ($\bigstar^{\mathrm{C}}$)). Moreover, our construction of the automaton in Lemma 4.12 does depend on the number of variables unlike [LOS20, Proposition 26] (again, see Section 9.2). This is crucial for complexity, as it is related to the number of Rabin pairs. We also use [EJ00] more precisely than [LOS20, p.621] as we handle non-binary trees. In short, we introduced TCA for which we characterise complexity of the nonemptiness problem (providing a few improvements to [LOS20]). We left aside the question of the expressiveness of TCA, which is interesting but out of the scope of this paper.

This lead us to show that SAT(CTL($\mathbb{Z}$)) is ExpTime-complete (Theorem 5.6), and SAT(CTL*($\mathbb{Z}$)) is 2ExpTime-complete (Theorem 6.8). The only decidability proof for SAT(CTL*($\mathbb{Z}$)) done so far, see [CKL16, Theorem 32], is by reduction to a decidable second-order logic. Our complexity characterisation for SAT(CTL*($\mathbb{Z}$)) provides an answer to several open problems related to CTL*($\mathbb{Z}$) fragments, see e.g. [BG06, Gas09, CKL16, CT16, LOS20].

We believe that our results on TCA can help to establish complexity results for other logics (see also Section 6 about a domain for strings, [EFK22, Section 4] to handle more concrete domains and a recent result for some description logic with a concrete domain made of finite strings [DQ23c]).

Acknowledgement

We would like to warmly thank the anonymous reviewers for all their comments and suggestions that help us a lot to improve the quality of this document. Thanks also to the reviewers of the conference version of this work.

References

[ABB+23] Ch. Alrabbaa, F. Baader, S. Borgwardt, P. Koopmann, and A. Kovtunova. Combining proofs for description logic and concrete domain reasoning. In *RuleML+RR'23*, volume 14244 of *LNCS*, pages 54–69. Springer, 2023.

[ADG20] E.G. Amparore, S. Donatelli, and F. Gallà. A CTL* model checker for Petri nets. In *Petri Nets'20*, volume 12152 of *LNCS*, pages 403–413. Springer, 2020.

[AFF17] S. Abriola, D. Figueira, and S. Figueira. Logics of repeating values on data trees and branching counter systems. In *FoSSaCS'17*, volume 10203 of *LNCS*, pages 196–212, 2017.

[BB24] F. Baader and F. De Bortoli. The abstract expressive power of first-order and description logics with concrete domains. In *SAC'24*, pages 754–761. ACM, 2024.

[BBK24] S. Borgwardt, F. De Bortoli, and P. Koopmann. The precise complexity of reasoning in $\mathcal{ALC}$ with ω-admissible concrete domains. In *DL'24*, volume 3739 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2024.

[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 ω-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.

[BL17] K. Bartek and M. Lelyk. Modal mu-calculus with atoms. In *CSL'17*, pages 30:1–30:21. Leibniz-Zentrum für Informatik, LIPIcs, 2017.

[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.

[BP14] L. Bozzelli and S. Pinchinat. Verification of gap-order constraint abstractions of counter systems. *Theoretical Computer Science*, 523:1–36, 2014.

[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.

[BP24] A. Bhaskar and M. Praveen. Realizability problem for constraint LTL. *Information & Computation*, 296:105126, 2024.

[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 Sz. Toruńczyk. Weak MSO+U over infinite trees. In *STACS'12*, LIPIcs, pages 648–660, 2012.

[Car15] C. Carapelle. *On the satisfiability of temporal logics with concrete domains*. PhD thesis, Leipzig University, 2015.

[CDOT21] R. Condurache, C. Dima, Y. Oualhdaj, and N. Troquard. Rational Synthesis in the Commons with Careless and Careful Agents. In *AAMAS'21*, pages 368–376, 2021.

[Čer94] K. Čerāns. Deciding properties of integral relational automata. In *ICALP'94*, volume 820 of *LNCS*, pages 35–46. Springer, 1994.

[CKL13] C. Carapelle, A. Kartzow, and M. Lohrey. Satisfiability of CTL* with constraints. In *CONCUR'13*, LNCS, pages 455–463. Springer, 2013.

[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 ACM*, 28(1):114–133, 1981.

[CLO+16] L. Candeago, D. Larraz, A. Oliveras, E. Rodríguez-Carbonell, and A. Rubio. Speeding up the constraint-based method in difference logic. In *SAT'16*, volume 9710 of *LNCS*, pages 284–301. Springer, 2016.

[CLRS22] Th. Cormen, Ch. Leiserson, Ronald L. Rivest, and C. Stein. *Introduction to Algorithms, 4th Edition*. MIT Press, 2022.

[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 & 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.

[DG08] S. Demri and R. Gascon. Verification of qualitative $\mathbb{Z}$ constraints. *Theoretical Computer Science*, 409(1):24–40, 2008.

[DHLT14] N. Decker, P. Habermehl, M. Leucker, and D. Thoma. Ordered navigation on multi-attributed data words. In *CONCUR'14*, volume 8704 of *LNCS*, pages 497–511, 2014.

[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 Transitions on Computational Logic*, 10(3), 2009.

[DMP91] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[DQ21] S. Demri and K. Quaas. Concrete domains in logics: a survey. *ACM SIGLOG News*, 8(3):6–29, 2021.

[DQ23a] S. Demri and K. Quaas. Constraint automata on infinite data trees: from CTL*($\mathbb{Z}$)/CTL($\mathbb{Z}$) to decision procedures. In *CONCUR'23*, volume 279 of *LIPIcs*, pages 29:1–29:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[DQ23b] S. Demri and K. Quaas. Constraint automata on infinite data trees: From CTL(Z)/CTL*(Z) to decision procedures. CoRR, arXiv:2302.05327 (version 2), May 2023.

[DQ23c] S. Demri and K. Quaas. First steps towards taming description logics with strings. In *JELIA'23*, volume 14281 of *LNCS*, pages 322–337. Springer, 2023.

[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 ACM*, 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.

[Fig10] D. Figueira. *Reasoning on words and trees with data*. PhD thesis, ENS Cachan, 2010.

[Fig12] D. Figueira. Decidability of Downward XPath. *ACM Transitions on Computational Logic*, 13(4):1–40, 2012.

[FMP20] D. Figueira, A. Majumdar, and M. Praveen. Playing with repetitions in data words using energy games. *Logical Methods in Computer Science*, 16(3), 2020.

[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.

[FP24] M. Faella and G. Parlato. A unified automata-theoretic approach to LTL_f modulo theories. In *ECAI'24*, pages 1254–1261, 2024.

[Gas09] R. Gascon. An automata-based approach for CTL* with constraints. *Electronic Notes in Theoretical Computer Science*, 239:193–211, 2009.

[GHOW12] S. Göller, Ch. Haase, J. Ouaknine, and J. Worrell. Branching-time model checking of parametric one-counter automata. In *FoSSaCS'12*, volume 7213 of *LNCS*, pages 406–420. Springer, 2012.

[GM98] J.F. Groote and R. Mastescu. Verification of temporal properties of processes in a setting with data. In *AMAST'98*, volume 1548 of *LNCS*, pages 74–90, 1998.

[IX19] R. Iosif and X. Xu. Alternating automata modulo first order theories. In *CAV'19*, volume 11562 of *LNCS*, pages 46–63, 2019.

[JL11] M. Jurdziński and R. Lazić. Alternating automata on data trees and XPath satisfiability. *ACM Transitions on Computational Logic*, 12(3):19:1–19:21, 2011.

[KVW00] O. Kupferman, M. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 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. Milicić. A Tableau Algorithm for Description Logics with Concrete Domains and General Tboxes. *Journal of Automated Reasoning*, 38(1-3):227–259, 2007.

[LMO$^+$18] A. Lechner, R. Mayr, J. Ouaknine, A. Pouly, and J. Worrell. Model checking flat freeze LTL on one-counter automata. *Logical Methods in Computer Science*, 14(4), 2018.

[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 Transitions on Computational Logic*, 5(4):669–705, 2004.

[LW08] S. Lasota and I. Walukiewicz. Alternating timed automata. *ACM Transitions on Computational Logic*, 9(2):10:1–10:27, 2008.

[MMW23] C. Mascle, A. Muscholl, and I. Walukiewicz. Model-checking parametric lock-sharing systems against regular constraints. In *CONCUR'23*, volume 279 of *LIPIcs*, pages 24:1–24:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[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. *Theoretical Computer Science*, 141(1–2):69–107, 1995.

[MT13] R. Mayr and R. Totzke. Branching-time model checking gap-order constraint systems. In *RP'13*, volume 8169 of *LNCS*, pages 171–182. Springer, 2013.

[MT16] R. Mayr and P. Totzke. Branching-time model checking gap-order constraint systems. *Fundamenta Informaticae*, 143(3–4):339–353, 2016.

[NSV04] F. Neven, T. Schwentick, and V. Vianu. Finite state machines for strings over infinite alphabets. *ACM Transitions on Computational Logic*, 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.

[RS23] A. Rodríguez and C. Sánchez. Boolean abstractions for realizability modulo theories. In *CAV'23*, volume 13966 of *LNCS*, pages 305–328. Springer, 2023.

[RS24] A. Rodríguez and C. Sánchez. Realizability modulo theories. *Journal of Logical and Algebraic Methods in Programming*, 140:100971, 2024.

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

[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 Sz. Toruńczyk. Automata based verification over linearly ordered data domains. In *STACS'11*, pages 81–92, 2011.

[SW16] F. Song and Z. Wu. On temporal logics with data variable quantifications: decidability and complexity. *Information & Computation*, 251:104–139, 2016.

[Tho90] W. Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science, Volume B, Formal models and semantics*, pages 133–191. Elsevier, 1990.

[TZ22] Sz. Toruńczyk and Th. Zeume. Register automata with extrema constraints, and an application to two-variable logic. *Logical Methods in Computer Science*, 18(1), 2022.

[Ves15] S. Vester. On the complexity of model-checking branching and alternating-time temporal logics in one-counter systems. In *ATVA'15*, volume 9364 of *LNCS*, pages 361–377. Springer, 2015.

[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 & 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.

[Wei16] Th. Weidner. *Probabilistic Logic, Probabilistic Regular Expressions, and Constraint Temporal Logic*. PhD thesis, University of Leipzig, 2016.

Appendix A. Proof for Section 3

A.1. Proof of Lemma 3.1.

Proof. For each $k \in [1,n]$, let $\mathbb{A}_k = (Q_k, \Sigma, D, \beta, Q_{k,\mathrm{in}}, \delta_k, F_k)$ with $F_k = (L_k^\gamma, U_k^\gamma)_{\gamma \in [1,N_k]}$ be a Rabin TCA. We define a Rabin TCA $\mathbb{A} = (Q, \Sigma, D, \beta, Q_{\mathrm{in}}, \delta, F)$ such that $\mathrm{L}(\mathbb{A}) = \bigcap_{1 \leqslant k \leqslant n} \mathrm{L}(\mathbb{A}_k)$. The Rabin TCA $\mathbb{A}$ is designed as a product automaton, where the locations are of the form $(q_1, \ldots, q_n, \mathfrak{f})$ with $(q_1, \ldots, q_n) \in Q_1 \times \cdots \times Q_n$ and $\mathfrak{f}$ is a map $\mathfrak{f} : [1, N_1] \times \cdots \times [1, N_n] \to [0, 2n-1]$ that can be viewed as a finite memory related to the visit of locations in the sets L_k^γ's. We write $\mathbf{F}(N_1, \ldots, N_n)$ to denote the set of maps of the form $\mathfrak{f} : [1, N_1] \times \cdots \times [1, N_n] \to [0, 2n-1]$. Let us explain how we use its elements to encode the conjunction of the Rabin acceptance conditions $F_1, \ldots, F_n$ into a single Rabin acceptance condition in the product tree automaton, with a single exponential blow-up, which happens to be of reasonable magnitude. Suppose that along a path of a run for $\mathbb{A}$ (yet to be defined), for the sequence of locations below

$$(q_{1,0}, \ldots, q_{n,0}, \mathfrak{f}^0), (q_{1,1}, \ldots, q_{n,1}, \mathfrak{f}^1), (q_{1,2}, \ldots, q_{n,2}, \mathfrak{f}^2) \ldots,$$

we wish to guarantee that it satisfies the conjunction of the Rabin acceptance conditions $F_1, \ldots, F_n$. Since the tuples $(q_{1,i}, \ldots, q_{n,i})$'s are updated synchronously according to the transition relations of the Rabin TCA $\mathbb{A}_k$'s, we have to enforce that for some $(\gamma_1, \ldots, \gamma_n) \in [1, N_1] \times \cdots \times [1, N_n]$, the following properties hold.

(a) For all $k \in [1,n]$, $L_k^{\gamma_k}$ is visited infinitely often in $q_{k,0}, q_{k,1}, q_{k,2}, \ldots$.
(b) For all $k \in [1,n]$, $U_k^{\gamma_k}$ is visited finitely in $q_{k,0}, q_{k,1}, q_{k,2}, \ldots$.

In the product automaton $\mathbb{A}$, we shall associate a unique Rabin pair (L, U) to each tuple $(\gamma_1, \ldots, \gamma_n)$ in order to enforce the satisfaction of (a) and (b) above. Since the Q_k's and $\mathbf{F}(N_1, \ldots, N_n)$ are finite sets, U can be defined as the union of sets of the form $Q_1 \times \cdots \times Q_{k-1} \times U_k^{\gamma_k} \times Q_{k+1} \times \cdots \times Q_n \times \mathbf{F}(N_1, \ldots, N_n)$, $k \in [1,n]$. In order to take care of (a), we use the maps from the set $\mathbf{F}(N_1, \ldots, N_n)$. The idea is to enforce that $L_1^{\gamma_1}$ is visited, then $L_2^{\gamma_2}$, then ... then $L_n^{\gamma_n}$, and this is repeated infinitely along the sequence of locations labelling the path of the run. For each map $\mathfrak{f} \in \mathbf{F}(N_1, \ldots, N_n)$, the intentions for the value $\mathfrak{f}(\gamma_1, \ldots, \gamma_n)$ are the following ones. If $\mathfrak{f}(\gamma_1, \ldots, \gamma_n) = 2k-2$ for some $k \in [1,n]$, then along the path we are waiting to meet a forthcoming location $(q_{1,i}, \ldots, q_{n,i}, \mathfrak{f}_i)$ such that $q_{k,i}$ belongs to $L_k^{\gamma_k}$. Once it is done, $\mathfrak{f}_i(\gamma_1, \ldots, \gamma_n)$ takes the odd value $2k-1$. By construction, at the next step $\mathfrak{f}_{i+1}(\gamma_1, \ldots, \gamma_n) = 2k \mod 2n$. So, $\mathfrak{f}_{i+1}(\gamma_1, \ldots, \gamma_n)$ is equal to $2k'$ for some $k' \in [0, n-1]$, and now along the path we are waiting to meet a forthcoming location $(q_{1,i'}, \ldots, q_{n,i'}, \mathfrak{f}_{i'})$ such that now $q_{k'+1,i'}$ belongs to $L_{k'+1}^{\gamma_{k'+1}}$. By requiring that $\mathfrak{f}_i(\gamma_1, \ldots, \gamma_n) = 1$ infinitely often along the sequence of locations labelling the path of the run, we shall guarantee the satisfaction of (a). As a consequence, L simply contains all the locations $(q_1, \ldots, q_n, \mathfrak{f})$ such that $\mathfrak{f}(\gamma_1, \ldots, \gamma_n) = 1$.

Before providing the formal definition for $\mathbb{A}$, we define below the function

$$\mathcal{U} : \mathbf{F}(N_1, \ldots, N_n) \times (Q_1 \times \cdots \times Q_n) \to \mathbf{F}(N_1, \ldots, N_n)$$

that is instrumental to update the maps in $\mathbf{F}(N_1, \ldots, N_n)$ along the paths of $\mathbb{A}$'s runs: define $\mathcal{U}(\mathfrak{f}, (q_1, \ldots, q_n)) = \mathfrak{f}'$, where, for all $(\gamma_1, \ldots, \gamma_n) \in [1, N_1] \times \cdots \times [1, N_n]$,

$$\mathfrak{f}'(\gamma_1, \ldots, \gamma_n) \stackrel{\text{def}}{=} \begin{cases} 2k \bmod 2n & \text{if } \mathfrak{f}(\gamma_1, \ldots, \gamma_n) = 2k-1 \text{ for some } k \in [1,n], \\ 2k-1 & \text{if } \mathfrak{f}(\gamma_1, \ldots, \gamma_n) = 2k-2 \text{ for some } k \in [1,n], \text{ and } q_k \in L_k^{\gamma_k}, \\ 2k-2 & \text{if } \mathfrak{f}(\gamma_1, \ldots, \gamma_n) = 2k-2 \text{ for some } k \in [1,n], \text{ and } q_k \notin L_k^{\gamma_k}. \end{cases}$$

Next, we state several simple properties. Let us consider the ω-sequence below in $(Q_1 \times \cdots \times Q_n \times \mathbf{F}(N_1, \ldots, N_n))^\omega$

$$(q_{1,0}, \ldots, q_{n,0}, \mathfrak{f}_0), (q_{1,1}, \ldots, q_{n,1}, \mathfrak{f}_1), (q_{1,2}, \ldots, q_{n,2}, \mathfrak{f}_2) \ldots,$$

such that for all $i \in \mathbb{N}$, $\mathfrak{f}_{i+1} = \mathcal{U}(\mathfrak{f}_i, (q_{1,i+1}, \ldots, q_{n,i+1}))$ and $\mathfrak{f}_0$ is the unique map that takes always the value zero. Such a sequence may correspond to the source locations of the transitions along a path in a run of $\mathbb{A}$, see below. One can easily show the truth of the following two claims.

Claim 1 For all $(\gamma_1, \ldots, \gamma_n) \in [1, N_1] \times \cdots \times [1, N_n]$, the following three statements are equivalent.

(1) For infinitely many $i \geqslant 0$, $\mathfrak{f}_i(\gamma_1, \ldots, \gamma_n) = 1$.
(2) For all $k \in [0, 2n-1]$, for infinitely many $i \geqslant 0$, $\mathfrak{f}_i(\gamma_1, \ldots, \gamma_n) = k$.
(3) For all $k \in [1, n]$, $L_k^{\gamma_k}$ is visited infinitely often in $q_{k,0}, q_{k,1}, q_{k,2} \ldots$.

Proof of Claim 1. The direction from (2) to (1) is trivial. For the direction from (1) to (2), suppose $\mathfrak{f}_i(\gamma_1, \ldots, \gamma_n) = 1$ for infinitely many $i \geqslant 0$. Let i be the first index such that $\mathfrak{f}_i(\gamma_1, \ldots, \gamma_n) = 1$. Recall that $\mathfrak{f}_0(\gamma_1, \ldots, \gamma_n) = 0 = 2k-2$ for $k = 1$. By definition of $\mathcal{U}$, we have $\mathfrak{f}_j(\gamma_1, \ldots, \gamma_n) = 0$ for $0 \leqslant j < i$, and $q_{1,i} \in L_1^{\gamma_1}$. But then, also by definition of $\mathcal{U}$, $\mathfrak{f}_{i+1}(\gamma_1, \ldots, \gamma_n) = 2 \neq 1$. Using similar arguments, one can show that the successive values for $\mathfrak{f}_0(\gamma_1, \ldots, \gamma_n), \mathfrak{f}_1(\gamma_1, \ldots, \gamma_n), \mathfrak{f}_2(\gamma_1, \ldots, \gamma_n) \ldots$ modulo stuttering are $(0 \cdot 1 \cdots (2n-2) \cdot (2n-1))^\omega$. For the direction from (2) to (3), the satisfaction of (2) implies that for all $k \in [1, n]$, for infinitely many i, $\mathfrak{f}_i(\gamma_1, \ldots, \gamma_n) = 2k-1$. Therefore $L_k^{\gamma_k}$ is visited infinitely often. Conversely, if (3) holds, then again, the successive values for $\mathfrak{f}_0(\gamma_1, \ldots, \gamma_n), \mathfrak{f}_1(\gamma_1, \ldots, \gamma_n), \mathfrak{f}_2(\gamma_1, \ldots, \gamma_n) \ldots$ modulo stuttering are $(0 \cdot 1 \cdots (2n-2) \cdot (2n-1))^\omega$ and therefore (2) holds. □

Claim 2 For all $(\gamma_1, \ldots, \gamma_n) \in [1, N_1] \times \cdots \times [1, N_n]$, the following two statements are equivalent.

(1) For all $k \in [1, n]$, $U_k^{\gamma_k}$ is visited only finitely in $q_{k,0}, q_{k,1}, q_{k,2} \ldots$.
(2) U is visited only finitely in $(q_{1,0}, \ldots, q_{n,0}, \mathfrak{f}_0), (q_{1,1}, \ldots, q_{n,1}, \mathfrak{f}_1), (q_{1,2}, \ldots, q_{n,2}, \mathfrak{f}_2) \ldots$, where

$$U = \Big(\bigcup_{k=1}^{n} Q_1 \times \cdots \times Q_{k-1} \times U_k^{\gamma_k} \times Q_{k+1} \times \cdots \times Q_n\Big) \times \mathbf{F}(N_1, \ldots, N_n).$$

Proof of Claim 2. Suppose that (1) holds. Since all Q_k's are finite, it follows that U cannot be visited infinitely often, and hence (2) holds. The direction from (2) to (1) is trivial. □

Below, we provide the formal definition for $\mathbb{A}$ according to the above informal description and then we prove the correctness of the construction based on the properties established above.

- $Q \stackrel{\text{def}}{=} Q_1 \times \cdots \times Q_n \times \mathbf{F}(N_1, \ldots, N_n)$.

- The tuple $((q_1,\ldots,q_n,\mathfrak{f}),\mathtt{a},(\Theta_0,(q^1_0,\ldots,q^n_0,\mathfrak{f}^0)),\ldots,(\Theta_{D-1},(q^1_{D-1},\ldots,q^n_{D-1},\mathfrak{f}^{D-1})))$ belongs to δ iff the conditions below hold.
  (1) For each $k\in[1,n]$, there is $(q_k,\mathtt{a},((\Theta^k_0,q^k_0),\ldots,(\Theta^k_{D-1},q^k_{D-1})))\in\delta_k$ and for each $j\in[0,D-1]$,
  $$\Theta_j \stackrel{\text{def}}{=} \bigwedge_{k=1}^{n}\Theta^k_j.$$
  Observe that the size of Θ_j is bounded above by $\sum_k (1+\mathtt{MCS}(\mathbb{A}_k))$.
  (2) For all $j\in[0,D-1]$, $\mathfrak{f}^j \stackrel{\text{def}}{=} \mathcal{U}(\mathfrak{f},(q^1_j,\ldots,q^n_j))$ (deterministic update).
  When $n=2$, the above conditions are identical to those in the proof of Lemma 4.14 and therefore the above developments generalise what is done in that proof.
- $Q_{\rm in}\stackrel{\text{def}}{=}Q_{1,\rm in}\times\cdots\times Q_{n,\rm in}\times\{\mathfrak{f}_0\}$, where $\mathfrak{f}_0$ 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 F contains exactly the pairs (L,U) for which there is $(\gamma_1,\ldots,\gamma_n)\in[1,N_1]\times\cdots\times[1,N_n]$ such that
  $$U\stackrel{\text{def}}{=}\big(\bigcup_{k=1}^{n}Q_1\times\cdots\times Q_{k-1}\times U_k^{\gamma_k}\times Q_{k+1}\times\cdots\times Q_n\big)\times\mathbf{F}(N_1,\ldots,N_n)$$
  $$L\stackrel{\text{def}}{=}\big(Q_1\times Q_2\times\cdots\times Q_n\big)\times\{\mathfrak{f}\mid\mathfrak{f}(\gamma_1,\ldots,\gamma_n)=1\}.$$
  Again, when $n=2$, the above construction is identical to the one in the proof of Lemma 4.14. Strictly speaking, (L,U) above is indexed by a tuple $(\gamma_1,\ldots,\gamma_n)$ but we omit such decorations as it will not lead to any confusion.

Finally, by construction F contains at most $N=\prod_k N_k$ pairs as required in the statement.

Let us prove that $\mathrm{L}(\mathbb{A})=\bigcap_{1\leqslant k\leqslant n}\mathrm{L}(\mathbb{A}_k)$. For the first inclusion, assume that $\mathbb{t}\in\bigcap_{1\leqslant k\leqslant n}\mathrm{L}(\mathbb{A}_k)$. For each $1\leqslant k\leqslant n$, there exists some initialized accepting run $\rho_k:[0,D-1]^*\to\delta_k$ of $\mathbb{A}_k$ on $\mathbb{t}$. That is, for every node $\mathbf{n}\in[0,D-1]^*$ with $\mathbb{t}(\mathbf{n})=(\mathtt{a},\boldsymbol{z})$ and $\mathbb{t}(\mathbf{n}\cdot i)=(\mathtt{a}_i,\boldsymbol{z}_i)$ for all $i\in[0,D-1]$, if $\rho_k(\mathbf{n})=(q^k,\mathtt{a},(\Theta^k_0,q^k_0),\ldots,(\Theta^k_{D-1},q^k_{D-1}))$, then, for all $i\in[0,D-1]$

(1) $\rho_k(\mathbf{n}\cdot i)$'s source location is q^k_i,
(2) $\mathbb{Z}\models\Theta^k_i(\boldsymbol{z},\boldsymbol{z}_i)$. Further, since ρ_k is initialized and accepting, we have
(3) $\rho_k(\varepsilon)$'s source location is in $Q_{k,\rm in}$, and
(4) for all paths π in ρ starting from the root node ε, there exists some $(L,U)\in F_k$ such that $\inf(\rho_k,\pi)\cap L\neq\varnothing$ and $\inf(\rho_k,\pi)\cap U=\varnothing$.

From these runs ρ_k, we define the map $\rho:[0,D-1]^*\to\delta$ as follows: for all $\mathbf{n}\in[0,D-1]^*$ with $\mathbb{t}(\mathbf{n})=(\mathtt{a},\boldsymbol{z})$, define

$$\rho(\mathbf{n})\stackrel{\text{def}}{=}((q_1,\ldots,q_n,\mathfrak{f}),\mathtt{a},(\Theta_0,(q^1_0,\ldots,q^n_0,\mathfrak{f}^0)),\ldots,(\Theta_{D-1},(q^1_{D-1},\ldots,q^n_{D-1},\mathfrak{f}^{D-1}))),$$

where $\Theta_j\stackrel{\text{def}}{=}\bigwedge_{k=1}^{n}\Theta^j_k$ and $\mathfrak{f}^j\stackrel{\text{def}}{=}\mathcal{U}(\mathfrak{f},(q^1_j,\ldots,q^n_j))$ for each $j\in[0,D-1]$. For this definition to make sense, we additionally require the source location of $\rho(\varepsilon)$ to be equal to $(\rho_1(\varepsilon),\ldots,\rho_n(\varepsilon),\mathfrak{f}_0)$, where $\mathfrak{f}_0$ is the function mapping everything to 0. Note that the corresponding transition is indeed in δ. Let us prove that ρ is an initialized accepting run of $\mathbb{A}$ on $\mathbb{t}$. Let $\mathbf{n}\in[0,D-1]^*$ and $j\in[0,D-1]$. Suppose $\mathbb{t}(\mathbf{n})=(\mathtt{a},\boldsymbol{z})$, $\mathbb{t}(\mathbf{n}\cdot j)=(\mathtt{a}_j,\boldsymbol{z}_j)$, and $\rho(\mathbf{n})=((q_1,\ldots,q_n,\mathfrak{f}),\mathtt{a},(\Theta_0,(q^1_0,\ldots,q^n_0,\mathfrak{f}^0)),\ldots,(\Theta_{D-1},(q^1_{D-1},\ldots,q^n_{D-1},\mathfrak{f}^{D-1})))$.

- That $\rho(\mathbf{n} \cdot j)$'s source location is $(q_j^1, \ldots, q_j^n, \mathfrak{f}^j)$ follows from (1) above, and by the deterministic update of the function.
- That $\mathbb{Z} \models \Theta_j(\boldsymbol{z}, \boldsymbol{z}_j)$ follows from (2) above.
- That $\rho(\varepsilon)$'s source location $(\rho_1(\varepsilon), \ldots, \rho_n(\varepsilon), \mathfrak{f}_0)$ is in Q_{in} follows from the definition of Q_{in} and (3) above.
- For proving that ρ is accepting, let $\pi = j_1 j_2 \cdots \in [0, D-1]^\omega$ be an infinite path in ρ starting from ε. Let $s = (q_{1,0}, q_{2,0}, \ldots, q_{n,0}, \mathfrak{f}^0), (q_{1,1}, q_{2,1}, \ldots, q_{n,1}, \mathfrak{f}^1), (q_{1,2}, q_{2,2}, \ldots, q_{n,2}, \mathfrak{f}^2) \ldots$ be the corresponding sequence of source locations. By (4) above, for each $k \in [1, n]$, there is an index γ_k such that some location in $L_k^{\gamma_k}$ occurs infinitely often in $q_{k,0}, q_{k,1}, q_{k,2} \ldots$. and all the locations in $U_k^{\gamma_k}$ occur only finitely often in $q_{k,0}, q_{k,1}, q_{k,2} \ldots$. Let us pick (L, U) in F such that
$$U = \big(\bigcup_{k=1}^{n} Q_1 \times \cdots \times Q_{k-1} \times U_k^{\gamma_k} \times Q_{k+1} \times \cdots \times Q_n\big) \times \mathbf{F}(N_1, \ldots, N_n)$$
$$L = \big(Q_1 \times Q_2 \times \cdots \times Q_n\big) \times \{\mathfrak{f} \mid \mathfrak{f}(\gamma_1, \ldots, \gamma_n) = 1\}.$$
By Claim 2, all the locations in U occur only finitely often in the sequence s. Hence $\inf(\rho, \pi) \cap U = \varnothing$. By Claim 1 and the definition of L for the tuple $(\gamma_1, \ldots, \gamma_n)$, there is some location in L occurring infinitely often in the sequence s. Hence $\inf(\rho, \pi) \cap L \neq \varnothing$, which finishes the proof of the first direction.

For the other inclusion, assume that $\mathbb{t} \in \mathrm{L}(\mathbb{A})$. Then there exists some initialized accepting run $\rho : [0, D-1]^* \to \delta$. That is, for every node $\mathbf{n} \in [0, D-1]^*$ with $\mathbb{t}(\mathbf{n}) = (\mathsf{a}, \boldsymbol{z})$ and $\mathbb{t}(\mathbf{n} \cdot i) = (\mathsf{a}_i, \boldsymbol{z}_i)$ for all $i \in [0, D-1]$, if
$$\rho(\mathbf{n}) = ((q_1, \ldots, q_n, \mathfrak{f}), \mathsf{a}, (\Theta_0, (q_0^1, \ldots, q_0^n, \mathfrak{f}^0)), \ldots, (\Theta_{D-1}, (q_{D-1}^1, \ldots, q_{D-1}^n, \mathfrak{f}^{D-1}))$$
then, for all $i \in [0, D-1]$

(1) $\rho(\mathbf{n} \cdot i)$'s source location is $(q_i^1, \ldots, q_i^n, \mathfrak{f}^i)$,
(2) $\mathbb{Z} \models \Theta_i(\boldsymbol{z}, \boldsymbol{z}_i)$.
(3) By definition of δ, we can also infer that $(q_k, \mathsf{a}, ((\Theta_0^k, q_0^k), \ldots, (\Theta_{D-1}^k, q_{D-1}^k))) \in \delta_k$, and $\Theta_i \overset{\text{def}}{=} \bigwedge_{k=1}^n \Theta_k^i$ for all $k \in [1, n]$. Further, since ρ is initialized and accepting, we have
(4) $\rho(\varepsilon)$'s source location is in Q_{in}, and
(5) for all paths π in ρ starting from ε, there exists some $(L, U) \in F$ such that $\inf(\rho, \pi) \cap L \neq \varnothing$ and $\inf(\rho, \pi) \cap U = \varnothing$.

For all $k \in [1, n]$, we define the map $\rho_k : [0, D-1]^* \to \delta_k$ by
$$\rho_k(\mathbf{n}) \overset{\text{def}}{=} (q_k, \mathsf{a}, ((\Theta_0^k, q_0^k), \ldots, (\Theta_{D-1}^k, q_{D-1}^k))),$$
for each $\mathbf{n} \in [0, D-1]^*$. Let us prove that ρ_k is an initialized accepting run of $\mathbb{A}_k$ on $\mathbb{t}$. Let $\mathbf{n} \in [0, D-1]^*$ and $j \in [0, D-1]$. Suppose $\mathbb{t}(\mathbf{n}) = (\mathsf{a}, \boldsymbol{z})$, $\mathbb{t}(\mathbf{n} \cdot j) = (\mathsf{a}_j, \boldsymbol{z}_j)$, and $\rho_k(\mathbf{n}) = (q^k, \mathsf{a}, (\Theta_0^k, q_0^k), \ldots, (\Theta_{D-1}^k, q_{D-1}^k))$.

- That $\rho_k(\mathbf{n} \cdot j)$'s source location is q_k^1 follows from (1) above.
- That $\mathbb{Z} \models \Theta_j^k(\boldsymbol{z}, \boldsymbol{z}_j)$ follows from (2) and (3) above (Θ_j^k is a conjunct of Θ_j).
- That $\rho_k(\varepsilon)$'s source location is in $Q_{k,\text{in}}$ follows from (4) above and the definition of Q_{in}.
- For proving that ρ_k is accepting, let $\pi = j_1 j_2 \cdots \in [0, D-1]^\omega$ be an infinite path of ρ from ε. By (5) above, there exists some $(L, U) \in F$ such that $\inf(\rho, \pi) \cap L \neq \varnothing$ and $\inf(\rho, \pi) \cap U = \varnothing$. By definition of F, (L, U) is defined for some tuple of indices $(\gamma_1, \ldots, \gamma_n) \in [1, N_1] \times \cdots \times [1, N_n]$. This implies that, assuming that $s =$

$(q_{1,0}, q_{2,0}, \ldots, q_{n,0}, \mathfrak{f}^0), (q_{1,1}, q_{2,1}, \ldots, q_{n,1}, \mathfrak{f}^1), (q_{1,2}, q_{2,2}, \ldots, q_{n,2}, \mathfrak{f}^2) \ldots$ be the sequence of source locations in $\mathbb{A}$ corresponding to π, that all elements in

$$\big(\bigcup_{k=1}^{n} Q_1 \times \cdots \times Q_{k-1} \times U_k^{\gamma_k} \times Q_{k+1} \times \cdots \times Q_n\big) \times \mathbf{F}(N_1, \ldots, N_n)$$

occur only finitely in s. Moreover, some location in s

$$\big(Q_1 \times Q_2 \times \cdots \times Q_n\big) \times \{\mathfrak{f} \mid \mathfrak{f}(\gamma_1, \ldots, \gamma_n) = 1\}$$

occurs infinitely often in the sequence s. By projecting on the locations in Q_k, this means that all the elements in $U_k^{\gamma_k}$ occur only finitely in $q_{k,0}, q_{k,1}, q_{k,2} \ldots$ and, by Claim 1, some location in $L_k^{\gamma_k}$ occurs infinitely often in $q_{k,0}, q_{k,1}, q_{k,2} \ldots$. This concludes that for all $k \in [1, n]$, ρ_k is an accepting run on $\mathbb{t}$ and therefore $\mathbb{t} \in \bigcap_k \mathrm{L}(\mathbb{A}_k)$. □

Appendix B. Proof for Section 4

B.1. Proof of ExpTime-hardness for the nonemptiness problem.

Proof. 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]. Let us first give a formal definition of alternating Turing machines. An *alternating Turing machine* (ATM) is a tuple $\mathcal{M} = (Q, \Sigma, \delta, q_0, q_{\mathrm{acc}}, q_{\mathrm{rej}}, g)$ defined as follows.

- Q is the finite set of control states.
- Σ is the finite tape alphabet including the blank symbol $\sharp$ and the left endmarker $\triangleright$.
- $\delta : Q \backslash \{q_{\mathrm{acc}}, q_{\mathrm{rej}}\} \times \Sigma \to \mathcal{P}(Q \backslash \{q_0\} \times \Sigma \times \{\leftarrow, \rightarrow\})$ is the transition function and each $\delta(q, \mathtt{a})$ contains exactly two elements.
- $q_0 \in Q$ is the initial state, $q_{\mathrm{acc}} \in Q$ is the accepting state, and $q_{\mathrm{rej}} \in Q$ is the rejecting state.
- $g : Q \backslash \{q_{\mathrm{acc}}, q_{\mathrm{rej}}\} \to \{\forall, \exists\}$ specifies the type of a state ($\forall$ for universal states, $\exists$ for existential states); without loss of generality, we assume $g(q_0) = \forall$.

A *configuration* is a finite word in $\Sigma^*(Q \times \Sigma)\Sigma^*$, and we write Configs$(\mathcal{M})$ to denote the set of all configurations of the ATM $\mathcal{M}$. As usual, a configuration C of the form $w(q, \mathtt{a})w'$ encodes a word $w\mathtt{a}w'$ on the tape, with control state q and the head is on the $|w\mathtt{a}|^{\mathrm{th}}$ tape cell where $|w\mathtt{a}|$ denotes the length of the word $w\mathtt{a}$. We also say that C uses $|w\mathtt{a}w'|$ tape cells. We write $\vdash_{\mathcal{M}}$ to denote the derivation relation of the ATM $\mathcal{M}$, defined as follows, and where $C = w(q, \mathtt{a})w'$.

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

Given a word w in $(\Sigma \backslash \{\triangleright, \sharp\})^*$, its initial configuration $C_{\mathrm{in}}(w)$ is $(q_0, \triangleright)w$. The configuration C is *accepting* if $q = q_{\mathrm{acc}}$, and *rejecting* if $q = q_{\mathrm{rej}}$. An *accepting run* for w is a finite tree $\mathbb{t} : \mathtt{dom}(\mathbb{t}) \to$ Configs$(\mathcal{M})$ with at most two children per node (but in $\mathtt{dom}(\mathbb{t})$, some nodes may have a unique child) satisfying the following conditions:

- We have $\mathbb{t}(\varepsilon) = C_{\mathrm{in}}(w)$ and all the leaves are labelled by accepting configurations.
- The tree $\mathbb{t}$ does not contain any node labelled by a rejecting configuration.

- For all $\mathbf{n} \in \mathtt{dom}(\mathbb{t})$ with $\mathbb{t}(\mathbf{n}) = w(q, \mathtt{a})w'$ 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 the configurations C' such that $w(q, \mathtt{a})w' \vdash_{\mathcal{M}} C'$ (maybe the same for the two options).
- For all $\mathbf{n} \in \mathtt{dom}(\mathbb{t})$ with $\mathbb{t}(\mathbf{n}) = w(q, \mathtt{a})w'$ 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 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 $w \in (\Sigma \backslash \{\rhd, \sharp\})^*$, if w has an accepting run, then it has an accepting run such that all the configurations labelling the nodes use at most $P(|w|)$ tape cells. The problem of determining whether a polynomially space-bounded ATM $\mathcal{M}$ accepts the input word w is known to be ExpTime-complete [CKS81, Corollary 3.6].

Let us now turn to the reduction proof. Let $\mathcal{M} = (Q, \Sigma, \delta, q_0, g)$ be a polynomially space-bounded ATM (with polynomial $P(n) \geqslant n$) and $w \in \Sigma^*$ be an input word. We define a Büchi TCA $\mathbb{A}_{\mathcal{M},w} = (Q', \Sigma', 2, \beta, Q'_{\text{in}}, \delta', F)$ such that $\mathcal{M}$ accepts w iff $\mathrm{L}(\mathbb{A}_{\mathcal{M},w}) \neq \emptyset$.

The idea is to design $\mathbb{A}_{\mathcal{M},w}$ so that it accepts a representation of the accepting runs of $\mathcal{M}$ from w. Let us first explain how we can represent a configuration of $\mathcal{M}$. Without any loss of generality, we assume that all the configurations use exactly $P(|w|)$ tape cells, possibly by padding the suffix with copies of the letter $\sharp$. The content of each of the $P(|w|)$ tape cells is encoded by the value of its own variable in the TCA; hence we set the number of variables $\beta = P(|w|)$. For the encoding, let $\mathfrak{f} : \Sigma \cup \{\sharp, \rhd\} \to [1, \mathrm{card}(\Sigma) + 2]$ be an arbitrary one-to-one map. The tape of length $P(|w|)$ is encoded by the values of the variables $\mathtt{x}_1, \ldots, \mathtt{x}_{P(|w|)}$, i.e. the letter $\mathtt{b}$ on the j^{th} tape cell is encoded by the constraint $\mathtt{x}_j = \mathfrak{f}(\mathtt{b})$. Let us define a simple and natural correspondence between configurations from $\Sigma^*(Q \times \Sigma)\Sigma^*$ using β tape cells and a subset of $Q \times [1, \beta] \times [1, \mathrm{card}(\Sigma)]^{\beta}$. Elements of the set $Q \times [1, \beta] \times [1, \mathrm{card}(\Sigma)]^{\beta}$ are called *numerical configurations*. We say that $w(q, \mathtt{a})w' \approx (q', i, \mathbb{d}_1, \ldots, \mathbb{d}_{\beta})$ iff $q = q'$, $i = |w\mathtt{a}|$ and for all $j \in [1, \beta]$, we have $\mathbb{d}_j = \mathfrak{f}((w\mathtt{a}w')(j))$.

Let us complete the definition of the TCA $\mathbb{A}_{\mathcal{M},w} = (Q', \Sigma', 2, \beta, Q'_{\text{in}}, \delta', F')$ by

- $\Sigma' = \{\dagger\}$ (Σ' plays no essential role).
- $Q' \stackrel{\text{def}}{=} Q \times [1, \beta] \uplus \{q^{\star}\}$. 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 $q^{\star}$ is a special accepting state.
- $Q'_{\text{in}} = \{(q_0, 1)\}$ and $F' \stackrel{\text{def}}{=} \{(q_{\text{acc}}, i) \mid i \in [1, \beta]\} \cup \{q^{\star}\}$.
- It remains to define the transition relation δ'.
  - For all $q \in \{q_{\text{acc}}, q_{\text{rej}}\}$, and all $i \in [1, \beta]$, the only transitions starting from (q, i) are $((q, i), \dagger, ((\top, (q, i)), (\top, (q, i))))$.
  - The only transition starting from $q^{\star}$ is $(q^{\star}, \dagger, ((\top, q^{\star}), (\top, q^{\star})))$.
  - Given $i \in [1, \beta]$, $\mathtt{a}, \mathtt{a}' \in \Sigma$, we write $\Theta(i, \mathtt{a}, \mathtt{a}')$ to denote the constraint in $\mathrm{TreeCons}(\beta)$ below:

$$\big(\bigwedge_{j \in [1,\beta] \backslash \{i\}} \mathsf{X}\mathtt{x}_j = \mathtt{x}_j\big) \wedge \mathtt{x}_i = \mathfrak{f}(\mathtt{a}) \wedge \mathsf{X}\mathtt{x}_i = \mathfrak{f}(\mathtt{a}').$$

  For all $q \in Q$ s.t. $g(q) = \exists$, for all $\mathtt{a} \in \Sigma$ with $\delta(q, \mathtt{a})$ of the form $\{(q_1, \mathtt{a}_1, d_1), (q_2, \mathtt{a}_2, d_2)\}$, for all $u \in \{1, 2\}$ and $i \in [1, \beta]$, δ' contains the transition

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

  with the expression $m(i, d_u)$ just above defined from $m(i, \rightarrow) \stackrel{\text{def}}{=} i{+}1$ and $m(i, \leftarrow) \stackrel{\text{def}}{=} i{-}1$.

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

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

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

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

where $\Theta'(1, \mathtt{a}, \mathtt{a}_u) \stackrel{\text{def}}{=} \Theta(1, \mathtt{a}, \mathtt{a}_u) \wedge \bigwedge_{1 \leqslant j \leqslant \beta} \mathtt{x}_j = \mathfrak{f}(v(j))$ $(u \in \{1, 2\})$. Observe that necessarily $d_1 = d_2 = \rightarrow$ because the head cannot go to the left of the first position. The location q_0 has a special status and the transitions from it encodes the input word w (and that is why we never come back to it).

Correctness of the construction is stated below.

Lemma B.1. *$\mathcal{M}$ accepts w iff* $\mathrm{L}(\mathbb{A}_{\mathcal{M},w}) \neq \emptyset$.

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 β tape cells, there is a unique numerical configuration $(q, i, \mathbb{d}_1, \ldots, \mathbb{d}_\beta)$ such that $C \approx (q, i, \mathbb{d}_1, \ldots, \mathbb{d}_\beta)$.

(II): For every numerical configuration $(q, i, \mathbb{d}_1, \ldots, \mathbb{d}_\beta)$, there is a unique configuration such that $C \approx (q, i, \mathbb{d}_1, \ldots, \mathbb{d}_\beta)$.

(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 δ, there are numerical configurations

$$(q, i, \mathbb{d}_1, \ldots, \mathbb{d}_\beta),\ (q_1, i_1, \mathbb{d}^1_1, \ldots, \mathbb{d}^1_\beta) \text{ and } (q_2, i_2, \mathbb{d}^2_1, \ldots, \mathbb{d}^2_\beta),$$

and a transition $((q, i), \dagger, ((\Theta_1, (q_1, i_1)), (\Theta_2, (q_2, i_2)))) \in \delta'$ such that

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

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

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

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

Conditions (III) and (IV) are proved using condition (I) and the way δ' 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 $(q, i, \mathbb{d}_1, \ldots, \mathbb{d}_\beta)$, $(q_1, i_1, \mathbb{d}^1_1, \ldots, \mathbb{d}^1_\beta)$ and $(q_2, i_2, \mathbb{d}^2_1, \ldots, \mathbb{d}^2_\beta)$, and transitions $((q, i), \dagger, ((\Theta_1, (q_1, i_1)), (\Theta_2, (q_2, i_2))))$ such that $\mathbb{Z} \models \Theta_1(\overline{\mathbb{d}}, \overline{\mathbb{d}^1})$ and $\mathbb{Z} \models \Theta_2(\overline{\mathbb{d}}, \overline{\mathbb{d}^2})$, there are configurations C, C_1 and C_2 such that

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

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

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

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

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

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

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

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

- For all $\mathbf{n} \in \mathtt{dom}(\mathbb{t})$ with $\mathbb{t}(\mathbf{n}) \approx (q, i, \mathbb{d}_1, \ldots, \mathbb{d}_\beta)$, we have $\rho^\star(\mathbf{n}) \stackrel{\text{def}}{=} ((q,i), \cdot, \cdot, \cdot)$, that is, a transition with source location (q,i). By way of example, assuming that $\delta(q, \mathfrak{f}^{-1}(\mathbb{d}_i)) = \{(q_1, \mathtt{a}_1, d_1), (q_2, \mathtt{a}_2, d_2)\}$, then more precisely,

$$\rho^\star(\mathbf{n}) \stackrel{\text{def}}{=} ((q,i), \dagger, ((\Theta(i, \mathfrak{f}^{-1}(\mathbb{d}_i), \mathtt{a}_1), (q_1, m(i, d_1))), (\Theta(i, \mathfrak{f}^{-1}(\mathbb{d}_i), \mathtt{a}_2), (q_2, m(i, d_2)))).$$

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

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

($\Leftarrow$) Let $\mathbb{t} : [0,1]^* \to \Sigma' \times \mathbb{Z}^\beta$ be an infinite tree and $\rho : [0,1]^* \to \delta'$ be an accepting run on $\mathbb{t}$. By construction of $\mathbb{A}_{\mathcal{M},w}$, 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 $\rho(\mathbf{n})$'s source locations is not in F, or the source location of $\rho(\mathbf{n})$ is in $F \backslash \{q^\star\}$ and the source locations of all its ancestors do not belong to F. One can check that X is a finite tree (use of König's Lemma here). Let $\mathbb{t}^\star : X \to \mathrm{Configs}(\mathcal{M})$ be the map such that for all $\mathbf{n} \in X$, we have $\mathbb{t}^\star(\mathbf{n}) = C$ for the unique configuration such that $C \approx (q, i, \mathbb{d}_1, \ldots, \mathbb{d}_\beta)$ with the source location of $\rho(\mathbf{n})$ being (q,i) and $\mathbb{t}(\mathbf{n}) = (\mathbb{d}_1, \ldots, \mathbb{d}_\beta)$. One can easily show that $\mathbb{t}^\star$ is an accepting run for w by using the conditions (V) and (VI), as well as the definition of the set of accepting states in $\mathcal{M}$. □

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. Indeed, N distinct constants can be simulated by N variables whose values remain unchanged along the accepted tree. The initial constraints between these variables at the root correspond to the constraints between the corresponding constants. Furthermore, note that the ordering $<$ is not needed for the reduction (the constraints use only the equality), the ExpTime-hardness holds without constants and $<$. □

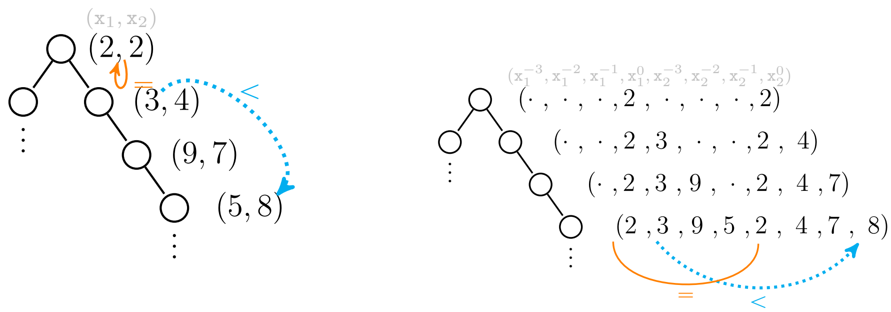


FIGURE 8. A tree over two variables $\mathtt{x}_1$ and $\mathtt{x}_2$ (left) and the corresponding tree over eight variables $\mathtt{x}_1^{-3}, \mathtt{x}_1^{-2}, \mathtt{x}_1^{-1}, \mathtt{x}_1^{0}, \mathtt{x}_2^{-3}, \mathtt{x}_2^{-2}, \mathtt{x}_2^{-1}, \mathtt{x}_2^{0}$ (right) for illustrating the translation of the formula $\mathsf{EX}\ \mathsf{E}((\mathsf{X}\mathtt{x}_1 < \mathsf{XXX}\mathtt{x}_2) \wedge (\mathtt{x}_1 = \mathtt{x}_2))$ into its simple form.

APPENDIX C. PROOFS FOR SECTION 5

C.1. **Proof of Proposition 5.1.**

Proof. First, we establish that 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 $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathfrak{v})$ be a total Kripke structure and $w \in \mathcal{W}$. We write $\hat{\mathcal{K}}_w$ to denote the (total) Kripke structure $(\hat{\mathcal{W}}, \hat{\mathcal{R}}, \hat{\mathfrak{v}})$ 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 (π, π') such that π is of the form $w_1 \cdots w_n$ and π' 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 $\mathtt{x} \in V$, we have $\hat{\mathfrak{v}}(\pi, \mathtt{x}) \stackrel{\text{def}}{=} \mathfrak{v}(w_n, \mathtt{x})$.

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 ϕ in 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 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 ϕ' in simple form. Let us introduce below the natural notion of forward degree. Given a term $\mathtt{t} = \mathsf{X}^i\mathtt{x}$ for some $i \in \mathbb{N}$, the *forward degree* of $\mathtt{t}$, written fd($\mathtt{t}$), is equal to i. Given a constraint Θ, we write fd(Θ) to denote the *forward degree* of Θ defined as the maximal forward degree of any term occurring in Θ. For instance, $\mathrm{fd}((\mathsf{X}\mathtt{x}_1 < \mathsf{XXX}\mathtt{x}_2) \wedge (\mathtt{x}_1 = \mathtt{x}_2)) = 3$. By extension, given a CTL($\mathbb{Z}$) formula ϕ, we write fd(ϕ) to denote the *forward degree* of ϕ defined as the maximal forward degree of any constraint occurring in ϕ. For instance, $\mathrm{fd}(\mathsf{EX}\ \mathsf{E}((\mathsf{X}\mathtt{x}_1 < \mathsf{XXX}\mathtt{x}_2) \wedge (\mathtt{x}_1 = \mathtt{x}_2))) = 3$ and $\mathrm{fd}(\mathsf{E}\ ((\mathsf{XX}\mathtt{x}_1 < \mathsf{XXXX}\mathtt{x}_2) \wedge (\mathsf{X}\mathtt{x}_1 = \mathsf{X}\mathtt{x}_2))) = 4$.

Let ϕ be a CTL($\mathbb{Z}$) formula in negation normal form such that $\mathrm{fd}(\phi) = N \geqslant 2$ and the variables occurring in ϕ are among $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$. Below, we build a formula ϕ' over

$\mathtt{x}_1^{-N}, \ldots, \mathtt{x}_1^0, \ldots, \mathtt{x}_\beta^{-N}, \ldots, \mathtt{x}_\beta^0$ with $\mathrm{fd}(\phi') \leqslant 1$ such that ϕ is satisfiable in a tree Kripke structure iff ϕ' is satisfiable in a tree Kripke structure and ϕ' can be computed in polynomial-time in the size of ϕ. 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 $\mathtt{x}_j^{-k}$ on a node should be understood as the value for $\mathtt{x}_j$ exactly k nodes behind along the branch leading to the node. Thanks to the tree structure, the variable $\mathtt{x}_j^{-k}$ takes a unique value. Note that we could not work with forward values (say with $\mathtt{x}_j^{+k}$ to keep the same syntactic rule for naming) because the structures are branching and therefore k steps ahead does not lead necessarily to a unique world.

Given $\mathsf{E}\ \Theta$ or $\mathsf{A}\ \Theta$ occurring in ϕ with $\mathrm{fd}(\Theta) = M$ (so $M \leqslant N$), we write $\mathtt{jump}(\Theta, M)$ to denote the constraints built over $\mathtt{x}_1^{-N}, \ldots, \mathtt{x}_1^0, \ldots, \mathtt{x}_\beta^{-N}, \ldots, \mathtt{x}_\beta^0$ such that

- $\mathtt{jump}(\cdot, M)$ is homomorphic for Boolean connectives,
- $\mathtt{jump}(\mathtt{t}_1 < \mathtt{t}_2, M) \stackrel{\text{def}}{=} \mathtt{jump}(\mathtt{t}_1, M) < \mathtt{jump}(\mathtt{t}_2, M)$,
- $\mathtt{jump}(\mathtt{t}_1 = \mathtt{t}_2, M) \stackrel{\text{def}}{=} \mathtt{jump}(\mathtt{t}_1, M) = \mathtt{jump}(\mathtt{t}_2, M)$,
- $\mathtt{jump}(\mathtt{t} = \mathbb{d}, M) \stackrel{\text{def}}{=} \mathtt{jump}(\mathtt{t}, M) = \mathbb{d}$ with $\mathtt{jump}(\mathsf{X}^i \mathtt{x}_j, M) \stackrel{\text{def}}{=} \mathtt{x}_j^{i-M}$. The value $i - M$ is greater than $-N$ because $\mathrm{fd}(\Theta) \leqslant N$.

For instance, $\mathtt{jump}((\mathsf{X}\mathtt{x}_1 < \mathsf{XXX}\mathtt{x}_2) \wedge (\mathtt{x}_1 = \mathtt{x}_2), 3) = (\mathtt{x}_1^{-2} < \mathtt{x}_2^0) \wedge (\mathtt{x}_1^{-3} = \mathtt{x}_2^{-3})$. In short, $\mathtt{jump}(\Theta, M)$ corresponds to the constraint equivalent to Θ if we evaluate it exactly M steps ahead. To do so, we therefore need to evoke in $\mathtt{jump}(\Theta, M)$ variables capturing previous local values along the branch. Let $\mathfrak{t}$ be the translation map that is homomorphic for Boolean and temporal connectives such that

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

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

$$\mathfrak{t}(\phi) \wedge \mathsf{AG}\ \mathsf{A}\big(\bigwedge_{j \in [1,\beta],\ k \in [0,N-1]} \mathtt{x}_j^{-k} = \mathsf{X}\mathtt{x}_j^{-k-1}\big).$$

The second conjunct in the definition of ϕ' corresponds to the renaming part. Recall that AG is the CTL temporal connective that enforces a property for all the reachable nodes from the root node (i.e. to all the worlds of the tree model). Let us provide hints to understand why ϕ is satisfiable in a tree Kripke structure iff ϕ' is satisfiable in a tree Kripke structure.

First, suppose that $\mathcal{K}, w \models \phi$, where $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathfrak{v})$ is a tree Kripke structure with root w. Let $\mathcal{K}' = (\mathcal{W}, \mathcal{R}, \mathfrak{v}')$ 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 $n \geqslant 0$, for all $k \in [0, N]$ and $j \in [1, \beta]$, we require $\mathfrak{v}'(w', \mathtt{x}_j^{-k}) \stackrel{\text{def}}{=} \mathfrak{v}(w_{n-k}, \mathtt{x}_j)$ if $n - k \geqslant 0$, otherwise $\mathfrak{v}'(w', \mathtt{x}_j^{-k}) \stackrel{\text{def}}{=} 0$ (arbitrary value). Figure 8 illustrates how $\mathfrak{v}'$ is defined. 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).

- $\mathcal{K}', w \models \mathsf{AG}\ \mathsf{A}\big(\bigwedge_{j \in [1,\beta],\ k \in [0,N-1]} \mathtt{x}_j^{-k} = \mathsf{X}\mathtt{x}_j^{-k-1}\big)$. This holds thanks to the definition of $\mathfrak{v}'$ and the tree structure of $\mathcal{K}$.
- If $\mathsf{E}\ \Theta$ occurs in ϕ and $\mathcal{K}, w' \models \mathsf{E}\ \Theta$, then $\mathcal{K}', w' \models (\mathsf{EX})^M\ \mathtt{jump}(\Theta, M)$, where $\mathrm{fd}(\Theta) = M$.
- If $\mathsf{A}\ \Theta$ occurs in ϕ and $\mathcal{K}, w' \models \mathsf{A}\ \Theta$, then $\mathcal{K}', w' \models (\mathsf{AX})^M\ \mathtt{jump}(\Theta, M)$, where $\mathrm{fd}(\Theta) = M$.

For the other direction, suppose that $\mathcal{K}, w \models \phi'$ and $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathfrak{v})$ is a tree Kripke structure with root w (recall the tree model property holds). Let $\mathcal{K}' = (\mathcal{W}, \mathcal{R}, \mathfrak{v}')$ 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 $\mathfrak{v}'(w', \mathtt{x}_j) \stackrel{\text{def}}{=} \mathfrak{v}(w', \mathtt{x}^0_j)$. By structural induction, one can show that $\mathcal{K}, w \models \mathfrak{t}(\phi)$ implies $\mathcal{K}', w \models \phi$. □

C.2. **Proof of Proposition 5.2.**

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

Let $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathfrak{v})$ 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 ϕ contains only the variables in $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$, the map $\mathfrak{v}$ can be restricted to the variables among $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$. Furthermore, below, we can represent $\mathfrak{v}$ as a map $\mathcal{W} \to \mathbb{Z}^\beta$ such that $\mathfrak{v}(w)(i)$ for some $i \in [1, \beta]$ is understood as the value of the variable $\mathtt{x}_i$ on w. Below, we construct a tree $\mathbb{t} : [0, D]^* \to \mathbb{Z}^\beta$ such that $\mathbb{t}$ obeys ι and $\mathbb{t}, \varepsilon \models \phi$.

We introduce an auxiliary map $g : [0, D]^* \to \mathcal{W}$ such that $g(\varepsilon) \stackrel{\text{def}}{=} w_{\text{in}}$, $\mathbb{t}(\varepsilon) \stackrel{\text{def}}{=} \mathfrak{v}(g(\varepsilon))$ and more generally, we require that for all $\mathbf{n} \in [0, D]^*$, we have $\mathbb{t}(\mathbf{n}) \stackrel{\text{def}}{=} \mathfrak{v}(g(\mathbf{n}))$. 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 $\pi = w_0 w_1 w_2 \cdots$ starting from $g(\mathbf{n})$. For all $i \geqslant 1$, we set $g(\mathbf{n} \cdot 0^i) \stackrel{\text{def}}{=} w_i$ and $\mathbb{t}(\mathbf{n} \cdot 0^i) \stackrel{\text{def}}{=} \mathfrak{v}(w_i)$. So let $j > 0$ and ϕ' be the unique subformula of ϕ such that $\iota(\phi') = j$. If $\mathcal{K}, g(\mathbf{n}) \not\models \phi'$, then, since $\mathcal{K}$ is total, there is an infinite path $\pi = w_0 w_1 w_2 \cdots$ starting from $g(\mathbf{n})$. We define $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} w_1$, $\mathbb{t}(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \mathfrak{v}(w_1)$ and for all $i \geqslant 1$, we set $g(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} w_{i+1}$ and $\mathbb{t}(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} \mathfrak{v}(w_{i+1})$. If $\mathcal{K}, g(\mathbf{n}) \models \phi'$, then we distinguish the following cases.

Case $\phi' = \mathsf{E}\ \Theta$: By $\mathcal{K}, g(\mathbf{n}) \models \mathsf{E}\ \Theta$, there is an infinite path $\pi = w_0 w_1 w_2 \cdots$ starting from $g(\mathbf{n})$ and such that $\mathbb{Z} \models \Theta(\mathfrak{v}(w_0), \mathfrak{v}(w_1))$. Recall that all the terms in Θ are in $\mathrm{T}_V^{\leqslant 1}$. As above, we define $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} w_1$, $\mathbb{t}(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \mathfrak{v}(w_1)$ and for all $i \geqslant 1$, we set $g(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} w_{i+1}$ and $\mathbb{t}(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} \mathfrak{v}(w_{i+1})$.

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

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

Case $\phi' = \mathsf{EX}\psi$ and ψ is neither an EU-formula nor an ER-formula:
By $\mathcal{K}, g(\mathbf{n}) \models \mathsf{EX}\psi$, there is an infinite path $\pi = w_0 w_1 w_2 \cdots$ starting from $g(\mathbf{n})$

such that $\mathcal{K}, w_1 \models \psi$. We set $g(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} w_1$, $\mathbb{t}(\mathbf{n} \cdot j) \stackrel{\text{def}}{=} \mathfrak{v}(w_1)$ and for all $i \geqslant 1$, $g(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} w_{i+1}$ and $\mathbb{t}(\mathbf{n} \cdot j \cdot 0^i) \stackrel{\text{def}}{=} \mathfrak{v}(w_{i+1})$.

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 $\mathbb{t}, \mathbf{n} \models \phi'$, and $\mathbb{t}$ obeys ι, that is,

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

The proof is by induction on the subformula relation.

- Suppose $\mathcal{K}, g(\mathbf{n}) \models \mathsf{E}\ \Theta$, and let $j = \iota(\mathsf{E}\ \Theta)$ for some $1 \leqslant j \leqslant D$. By definition of $\mathbb{t}$, $g(\mathbf{n} \cdot j) = w_1$ and $g(\mathbf{n} \cdot j \cdot 0^i) = w_{i+1}$, where $w_0, w_1, \ldots$ is an infinite path in $\mathcal{K}$ starting from $g(\mathbf{n})$ satisfying $\mathbb{Z} \models \Theta(\mathfrak{v}(w_0), \mathfrak{v}(w_1))$. We also have $\mathbb{t}(\mathbf{n}) = \mathfrak{v}(w_0)$ and $\mathbb{t}(\mathbf{n} \cdot j) = \mathfrak{v}(w_1)$, 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 $\mathbb{t}, \mathbf{n} \models \phi_1$ and $\mathbb{t}, \mathbf{n} \models \phi_2$, and thus $\mathbb{t}, \mathbf{n} \models \phi_1 \wedge \phi_2$. The case with $\phi_1 \vee \phi_2$ is similar.
- Suppose $\mathcal{K}, g(\mathbf{n}) \models \mathsf{EX}\phi'$. Suppose $j = \iota(\mathsf{EX}\phi')$ for some $1 \leqslant j \leqslant D$. We distinguish three cases (coming from the definition of $\mathbb{t}$).
  1. ϕ' is of the form $\mathsf{E}\phi_1\mathsf{U}\phi_2$: By definition of $\mathbb{t}$, we have $g(\mathbf{n} \cdot j^i) = w_i$ for all $0 \leqslant i \leqslant k$, where $w_0 w_1 w_2 \ldots$ 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 $1 \leqslant i < k$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_2$ and $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_1$ for all $1 \leqslant i < k$. Hence $\mathbb{t}, \mathbf{n} \models \mathsf{EX}\phi'$.
  2. ϕ' is of the form $\mathsf{E}\phi_1\mathsf{R}\phi_2$: By definition of $\mathbb{t}$, there are two cases.
     - Either $g(\mathbf{n} \cdot j^i) = w_i$ for all $i \geqslant 0$, where $w_0 w_1 w_2 \ldots$ is a path starting from $\mathbf{n}$ such that $\mathcal{K}, w_i \models \phi_2$ for all $i \geqslant 1$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_2$ for all $i \geqslant 1$, and hence $\mathbb{t}, \mathbf{n} \models \mathsf{EX}\phi'$.
     - Or $g(\mathbf{n} \cdot j^i) = w_i$ for all $0 \leqslant i \leqslant k$, where $w_0 w_1 w_2 \ldots$ is a path starting from $\mathbf{n}$ such that $\mathcal{K}, w_k \models \phi_1 \wedge \phi_2$ and $\mathcal{K}, w_i \models \phi_2$ for all $1 \leqslant i < k$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_1 \wedge \phi_2$ and $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_2$ for all $1 \leqslant i < k$. Hence $\mathbb{t}, \mathbf{n} \models \mathsf{EX}\phi'$.
  3. ϕ' 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 $w_0 w_1 w_2 \ldots$ is a path from $g(\mathbf{n})$ such that $\mathcal{K}, w_1 \models \phi'$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j \models \phi'$, and hence $\mathbb{t}, \mathbf{n} \models \mathsf{EX}\phi'$.
- Suppose $\mathcal{K}, g(\mathbf{n}) \models \mathsf{E}\phi_1\mathsf{U}\phi_2$. We distinguish two cases.
  - Suppose $\mathcal{K}, g(\mathbf{n}) \models \phi_2$. By the induction hypothesis, $\mathbb{t}, \mathbf{n} \models \phi_2$ and hence $\mathbb{t}, \mathbf{n} \models \mathsf{E}\phi_1\mathsf{U}\phi_2$.
  - Suppose $\mathcal{K}, g(\mathbf{n}) \not\models \phi_2$. Then $\mathcal{K}, g(\mathbf{n}) \models \phi_1$ and $\mathcal{K}, g(\mathbf{n}) \models \mathsf{EXE}\phi_1\mathsf{U}\phi_2$. By the induction hypothesis, we also have $\mathbb{t}, \mathbf{n} \models \phi_1$. Suppose $\iota(\mathsf{EXE}\phi_1\mathsf{U}\phi_2) = j$ for some $1 \leqslant j \leqslant D$. By definition of $\mathbb{t}$, there exists some (minimal) $k \geqslant 1$ such that $g(\mathbf{n} \cdot j^i) = w_i$ for all $0 \leqslant i \leqslant k$, where $w_0 w_1 w_2 \ldots$ 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 $1 \leqslant i < k$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_2$ and $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_1$ for all $1 \leqslant i < k$. Recall that we also have $\mathbb{t}, \mathbf{n} \models \phi_1$, so that indeed $\mathbb{t}, \mathbf{n} \models \mathsf{E}\phi_1\mathsf{U}\phi_2$.
- Suppose $\mathcal{K}, g(\mathbf{n}) \models \mathsf{E}\phi_1\mathsf{R}\phi_2$. 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, $\mathbb{t}, \mathbf{n} \models \phi_2$ and $\mathbb{t}, \mathbf{n} \models \phi_1$, so that $\mathbb{t}, \mathbf{n} \models \mathsf{E}\phi_1\mathsf{R}\phi_2$.

– Suppose $\mathcal{K}, g(\mathbf{n}) \models \neg\phi_1 \wedge \phi_2$ and $\mathcal{K}, g(\mathbf{n}) \models \mathsf{EXE}\phi_1\mathsf{R}\phi_2$. By the induction hypothesis, $\mathbb{t}, \mathbf{n} \models \phi_2$. Suppose $\iota(\mathsf{EXE}\phi_1\mathsf{R}\phi_2) = j$ for some $1 \leqslant j \leqslant D$. We distinguish two more cases.
  * There is an infinite path $w_0 w_1 w_2 \ldots$ from $g(\mathbf{n})$ such that $\mathcal{K}, w_i \models \phi_2$ for all $i \geqslant 1$. We then have $g(\mathbf{n} \cdot j^i) = w_i$ for all $i \geqslant 1$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_2$ for all $i \geqslant 1$. Together with $\mathbb{t}, \mathbf{n} \models \phi_2$, we obtain $\mathbb{t}, \mathbf{n} \models \mathsf{E}\phi_1\mathsf{R}\phi_2$.
  * Otherwise, we have $g(\mathbf{n} \cdot j^k) = w_k$ and $g(\mathbf{n} \cdot j^i) = w_i$, where $w_0 w_1 w_2 \ldots$ is a path from $g(\mathbf{n})$ such that $\mathcal{K}, w_k \models \phi_1 \wedge \phi_2$ and $\mathcal{K}, w_i \models \phi_2$ for all $1 \leqslant i < k$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_1 \wedge \phi_2$, and $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_2$ for all $1 \leqslant i < k$. Together with $\mathbb{t}, \mathbf{n} \models \phi_2$, we obtain $\mathbb{t}, \mathbf{n} \models \mathsf{E}\phi_1\mathsf{R}\phi_2$.

- Suppose $\mathcal{K}, g(\mathbf{n}) \models \mathsf{A}\ \Theta$. By construction of $\mathbb{t}$, for all infinite paths $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \cdots \in [0, D]^\omega$, $g(\mathbf{n}) \cdot g(\mathbf{n} \cdot j_1) \cdot g(\mathbf{n} \cdot j_1 j_2) \cdots$ is an infinite path from $g(\mathbf{n})$. Consequently, for all $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \cdots \in [0, D]^\omega$, we have $\mathbb{Z} \models \Theta(\mathfrak{v}(g(\mathbf{n})), \mathfrak{v}(g(\mathbf{n} \cdot j_1)))$ and therefore $\mathbb{Z} \models \Theta(\mathbb{t}(\mathbf{n}), \mathbb{t}(\mathbf{n} \cdot j_1))$ because by definition, we have $\mathbb{t}(\mathbf{n}) = \mathfrak{v}(g(\mathbf{n}))$ and $\mathbb{t}(\mathbf{n} \cdot j) = \mathfrak{v}(g(\mathbf{n} \cdot j))$. As a consequence, $\mathbb{t}, \mathbf{n} \models \mathsf{A}\ \Theta$.
- Suppose $\mathcal{K}, g(\mathbf{n}) \models \mathsf{A}\ \Phi$, where Φ is a path formula such that Φ is not a constraint. By construction of $\mathbb{t}$, for all infinite paths $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \cdots \in [0, D]^\omega$, $g(\mathbf{n}) \cdot g(\mathbf{n} \cdot j_1) \cdot g(\mathbf{n} \cdot j_1 j_2) \cdots$ is an infinite path from $g(\mathbf{n})$. By way of example, assume that $\Phi = \phi_1\mathsf{U}\phi_2$ (the cases $\Phi = \phi_1\mathsf{R}\phi_2$ and $\Phi = \mathsf{X}\phi_1$ are handled in the very same way). For all infinite paths $\mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \cdots \in [0, D]^\omega$ with $\pi = g(\mathbf{n}) \cdot g(\mathbf{n} \cdot j_1) \cdot g(\mathbf{n} \cdot j_1 j_2) \cdots$, we have $\mathcal{K}, \pi \models \phi_1\mathsf{U}\phi_2$ and therefore there is $k \in \mathbb{N}$ such that $\mathcal{K}, g(\mathbf{n} \cdot j_1 \cdots j_k) \models \phi_2$ and for all $0 \leqslant k' < k$, we have $\mathcal{K}, g(\mathbf{n} \cdot j_1 \cdots j_{k'}) \models \phi_1$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j_1 \cdots j_k \models \phi_2$ and $\mathbb{t}, \mathbf{n} \cdot j_1 \cdots j_{k'} \models \phi_1$ for all $0 \leqslant k' < k$. Consequently, $\mathbb{t}, \mathbf{n} \cdot j_1 \cdot j_2 \cdot j_3 \cdots \models \phi_1\mathsf{U}\phi_2$ and therefore $\mathbb{t}, \mathbf{n} \models \mathsf{A}\ \phi_1\mathsf{U}\phi_2$ because the above path from $\mathbf{n}$ was arbitrary in $\mathbb{t}$. ☐

C.3. Proof of Lemma 5.4.

Proof. "if:" Suppose $\mathrm{L}(\mathbb{A}_\phi) \neq \varnothing$. Then there exists a tree $\mathbb{t} : [0, D]^* \to \Sigma \times \mathbb{Z}^\beta$ in $\mathrm{L}(\mathbb{A}_\phi)$. Let $\rho : [0, D]^* \to \delta$ be an accepting run of $\mathbb{A}_\phi$ on $\mathbb{t}$. We prove for all $\phi' \in \mathrm{sub}(\phi)$, for all nodes $\mathbf{n}$ in $\mathbb{t}$ with $\mathbb{t}(\mathbf{n}) = (\dagger, \boldsymbol{z})$ and $\rho(\mathbf{n}) = ((d_{\mathbf{n}}, X_{\mathbf{n}}), \dagger, (\Theta_0, (0, X_{\mathbf{n}\cdot 0})), \ldots, (\Theta_D, (D, X_{\mathbf{n}\cdot D})))$, if $\phi' \in X_{\mathbf{n}}$, then $\mathbb{t}, \mathbf{n} \models \phi'$ (this is not an equivalence, recall that ϕ is in negation normal form). Here, $\mathbb{t}$ is also understood as a tree Kripke model (in which we may ignore the finite alphabet Σ). Note that this indeed implies $\mathbb{t}, \varepsilon \models \phi$, as $\rho(\varepsilon)$ starts with some initial location of the form $(0, X_\varepsilon)$ such that $\phi \in X_\varepsilon$. Hence ϕ is indeed satisfiable.

So let $\mathbf{n}$ be a node in $\mathbb{t}$ with

$$\rho(\mathbf{n}) = ((d_{\mathbf{n}}, X_{\mathbf{n}}), \dagger, (\Theta_0, (0, X_{\mathbf{n}\cdot 0})), \ldots, (\Theta_D, (D, X_{\mathbf{n}\cdot D})))$$

and such that

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

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

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

- $\phi' = \mathsf{E}\ \Theta$, where Θ is a Boolean combination of terms among $\mathtt{x}_1, \ldots, \mathtt{x}_\beta, \mathtt{x}'_1, \ldots, \mathtt{x}'_\beta$. Suppose $\iota(\phi') = j$ for some $1 \leqslant j \leqslant D$. From the condition (3) above, we obtain that $\Theta_j \stackrel{\text{def}}{=} (\bigwedge_{\mathsf{A}\Theta' \in X_{\mathbf{n}}} \Theta') \wedge \Theta$. Since ρ is a run, we have $\mathbb{Z} \models \Theta_j(\boldsymbol{z}, \boldsymbol{z}_j)$, where $\mathbb{t}(\mathbf{n}) = (\dagger, \boldsymbol{z})$ and $\mathbb{t}(\mathbf{n} \cdot j) = (\dagger, \boldsymbol{z}_j)$, and hence also $\mathbb{Z} \models \Theta(\boldsymbol{z}, \boldsymbol{z}_j)$. This yields $\mathbb{t}, \mathbf{n} \models \mathsf{E}\ \Theta$.
- $\phi' = \mathsf{A}\ \Theta$, where Θ is a Boolean combination of atomic constraints built over $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$ and $\mathtt{x}'_1, \ldots, \mathtt{x}'_\beta$. From the condition (3) above, we know that Θ appears in Θ_j as a conjunct. Since ρ is a run, we have $\mathbb{Z} \models \Theta_j(\boldsymbol{z}, \boldsymbol{z}_j)$ for all $0 \leqslant j \leqslant D$, where $\mathbb{t}(\mathbf{n}) = (\dagger, \boldsymbol{z})$ and $\mathbb{t}(\mathbf{n} \cdot j) = (\dagger, \boldsymbol{z}_j)$, and hence also $\mathbb{Z} \models \Theta(\boldsymbol{z}, \boldsymbol{z}_j)$ for all $0 \leqslant j \leqslant D$. This yields $\mathbb{t}, \mathbf{n} \models \mathsf{A}\ \Theta$.
- $\phi' = \phi_1 \wedge \phi_2$. Since $X_{\mathbf{n}}$ is propositionally consistent, we have $\phi_1, \phi_2 \in X_{\mathbf{n}}$. By the induction hypothesis, $\mathbb{t}, \mathbf{n} \models \phi_1$ and $\mathbb{t}, \mathbf{n} \models \phi_2$. Hence $\mathbb{t}, \mathbf{n} \models \phi'$.
- Similarly for $\phi' = \phi_1 \vee \phi_2$.
- $\phi' = \mathsf{EX}\phi_1$. From condition (1) above we know that $\phi_1 \in X_{\mathbf{n} \cdot i}$, where $i = \iota(\phi')$. By the induction hypothesis, $\mathbb{t}, \mathbf{n} \cdot i \models \phi_1$. Hence $\mathbb{t}, \mathbf{n} \models \phi'$.
- $\phi' = \mathsf{AX}\phi_1$. From condition (2) above, we know that $\phi_1 \in X_{\mathbf{n} \cdot i}$ for all $0 \leqslant i \leqslant D$. By the induction hypothesis, $\mathbb{t}, \mathbf{n} \cdot i \models \phi_1$ for all $0 \leqslant i \leqslant D$. Hence $\mathbb{t}, \mathbf{n} \models \phi'$.
- $\phi' = \mathsf{E}\phi_1\mathsf{U}\phi_2$. Since $X_{\mathbf{n}}$ is propositionally consistent, we have $\phi_2 \in X_{\mathbf{n}}$, or $\phi_1, \mathsf{EX}\phi' \in X_{\mathbf{n}}$. In the first case, we have $\mathbb{t}, \mathbf{n} \models \phi_2$ by the induction hypothesis, and hence $\mathbb{t}, \mathbf{n} \models \phi'$. In the second case, we have $\mathbb{t}, \mathbf{n} \models \phi_1$ by the induction hypothesis, and we have $\phi' \in X_{\mathbf{n} \cdot j}$, where $j = \iota(\mathsf{EX}\phi')$. Consider the infinite path $\mathbf{n} \cdot j \cdot j \cdot j \ldots$ starting from $\mathbf{n}$. Since ρ is an accepting run, states from $F_{\phi'}$ must occur infinitely often along this path; that is, there must exist some $k \geqslant 0$ such that $\phi_2 \in X_{\mathbf{n} \cdot j^k}$ or $\mathsf{E}\phi_1\mathsf{U}\phi_2 \notin X_{\mathbf{n} \cdot j^k}$. Recall that

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

  Indeed, the first component i in the location (i, X) in Q records the direction i from which the node is reached. Let $k \geqslant 0$ be minimal. We first show by induction that if $\phi' \in X_{\mathbf{n} \cdot j^i}$, then $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_1$ and $\phi' \in X_{\mathbf{n} \cdot j^{i+1}}$ for all $0 \leqslant i < k$. For $i = 0$, we have shown this above. So let $0 < i < k$ and suppose $\phi' \in X_{\mathbf{n} \cdot j^i}$. Since ρ is a run, $X_{\mathbf{n} \cdot j^i}$ is propositionally consistent and hence $\phi_2 \in X_{\mathbf{n} \cdot j^i}$ or $\phi_1, \mathsf{EX}\phi' \in X_{\mathbf{n} \cdot j^i}$. Note that $\phi_2 \in X_{\mathbf{n} \cdot j^i}$ contradicts that k is minimal; hence $\phi_1, \mathsf{EX}\phi' \in X_{\mathbf{n} \cdot j^i}$. By the induction hypothesis $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_1$, and $\phi' \in X_{\mathbf{n} \cdot j^{i+1}}$. We conclude that $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_1$ for all $0 \leqslant i < k$.

  Next we prove that $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_2$. 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 $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_2$. Hence, we have proved the existence of a path starting in $\mathbf{n}$ and satisfying $\phi_1\mathsf{U}\phi_2$, which leads to $\mathbb{t}, \mathbf{n} \models \phi'$.
- $\phi' = \mathsf{A}\phi_1\mathsf{U}\phi_2$. Since $X_{\mathbf{n}}$ is propositionally consistent, we have $\phi_2 \in X_{\mathbf{n}}$, or $\phi_1, \mathsf{AX}\phi' \in X_{\mathbf{n}}$. In the first case, we have $\mathbb{t}, \mathbf{n} \models \phi_2$ by the induction hypothesis, and hence $\mathbb{t}, \mathbf{n} \models \phi'$. In the second case, we have $\mathbb{t}, \mathbf{n} \models \phi_1$ by the induction hypothesis, and we have $\phi' \in X_{\mathbf{n} \cdot i}$ for all $0 \leqslant i \leqslant D$. We will prove that every path that starts in $\mathbf{n}$ satisfies $\phi_1\mathsf{U}\phi_2$. So consider an arbitrary infinite path $\mathbf{n} \cdot j_1 \cdot j_2 \cdot \cdots \in [0, D]^\omega$. Since ρ is accepting, states from $F_{\phi'}$ must occur infinitely often in π; that is there exists some $k \geqslant 0$ such that $\phi_2 \in X_{\mathbf{n} \cdot j_1 \cdot \cdots \cdot j_k}$ or $\phi' \notin X_{\mathbf{n} \cdot j_1 \cdot \cdots \cdot j_k}$. Let $k \geqslant 0$ be minimal. We first show by induction that if $\phi' \in X_{\mathbf{n} \cdot j_1 \cdot \cdots \cdot j_m}$, then $\mathbb{t}, \mathbf{n} \cdot j_1 \ldots j_m \models \phi_1$, and $\phi' \in X_{\mathbf{n} \cdot j_1 \cdot \cdots \cdot j_{m+1}}$ for all $0 \leqslant m < k$. We have proved this for $m = 0$ above. So let $0 \leqslant m < k$ and suppose $\phi' \in X_{\mathbf{n} \cdot j_1 \cdot \cdots \cdot j_m}$. Since ρ is a run, $X_{\mathbf{n} \cdot j_1 \cdot \cdots \cdot j_m}$ is propositionally consistent and hence $\phi_2 \in X_{\mathbf{n} \cdot j_1 \ldots j_m}$ or $\phi_1, \mathsf{AX}\phi' \in X_{\mathbf{n} \cdot j_1 \ldots j_m}$. Note that $\phi_2 \in X_{\mathbf{n} \cdot j_1 \ldots j_m}$ contradicts that k is minimal; hence $\phi_1, \mathsf{AX}\phi' \in X_{\mathbf{n} \cdot j_1 \ldots j_m}$. By the induction

hypothesis $\mathbb{t}, \mathbf{n} \cdot j_1 \ldots j_m \models \phi_1$, and since ρ is a run, we also have $\phi' \in X_{\mathbf{n} \cdot j_1 \ldots j_{m+1}}$. We conclude that $\mathbb{t}, \mathbf{n} \cdot j_1 \ldots j_m \models \phi_1$ for all $0 \leqslant m < k$.

Next we prove that $\mathbb{t}, \mathbf{n} \cdot j_1 \ldots j_k \models \phi_2$. Recall that $\phi_2 \in X_{\mathbf{n} \cdot j_1 \cdots j_k}$ or $\phi' \notin X_{\mathbf{n} \cdot j_1 \ldots j_k}$. But $\phi' \in X_{\mathbf{n} \cdot j_1 \ldots j_k}$ as we proved before. Hence $\phi_2 \in X_{\mathbf{n} \cdot j_1 \ldots j_k}$. By the induction hypothesis, we obtain $\mathbb{t}, \mathbf{n} \cdot j_1 \cdots j_k \models \phi_2$. Hence, we have proved that an arbitrary chosen path starting in $\mathbf{n}$ satisfies $\phi_1 \mathsf{U} \phi_2$, leading to $\mathbb{t}, \mathbf{n} \models \phi'$.

- $\phi' = \mathsf{E}\phi_1 \mathsf{R} \phi_2$. Since $X_{\mathbf{n}}$ is consistent, we have $\phi_2 \in X_{\mathbf{n}}$, and $\phi_1 \in X_{\mathbf{n}}$ or $\mathsf{EX}\phi' \in X_{\mathbf{n}}$. In the first case, by the induction hypothesis we have $\mathbb{t}, \mathbf{n} \models \phi_1$ and $\mathbb{t}, \mathbf{n} \models \phi_2$, and hence $\mathbb{t}, \mathbf{n} \models \phi'$. In the second case, we have $\mathbb{t}, \mathbf{n} \models \phi_2$ and $\phi' \in X_{\mathbf{n} \cdot j}$, where $j = \iota(\mathsf{EX}\phi')$. Consider the infinite path $\mathbf{n} \cdot j \cdot j \cdot j \ldots$ starting from $\mathbf{n}$. We prove that this path satisfies $\phi_1 \mathsf{R} \phi_2$. We distinguish two cases.
  - Suppose there exists some $k \geqslant 0$ such that $\phi_1 \in X_{\mathbf{n} \cdot j^k}$. Let k be minimal. By the induction hypothesis, $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_1$. We prove that if $\phi' \in X_{\mathbf{n} \cdot j^i}$, then $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_2$ and $\phi' \in X_{\mathbf{n} \cdot j^{i+1}}$ for all $0 \leqslant i < k$. For $i = 0$, we have proved this above. So let $0 < i < k$ and suppose $\phi' \in X_{\mathbf{n} \cdot j^i}$. Since ρ 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 $\mathsf{EX}\phi' \in X_{\mathbf{n} \cdot j^i}$. Since $\phi_1 \in X_{\mathbf{n} \cdot j^i}$ contradicts the minimality of k, we have $\mathsf{EX}\phi' \in X_{\mathbf{n} \cdot j^i}$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_2$, and since ρ is a run, we have $\phi' \in X_{\mathbf{n} \cdot j^{i+1}}$. We can conclude that $\mathbb{t}, \mathbf{n} \cdot j^i \models \phi_2$ for all $0 \leqslant i \leqslant k$.
  - Suppose that $\phi_1 \notin X_{\mathbf{n} \cdot j^k}$ for all $k \geqslant 0$. We prove that if $\phi' \in X_{\mathbf{n} \cdot j^k}$, then $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_2$ and $\phi' \in X_{\mathbf{n} \cdot j^{k+1}}$ for all $k \geqslant 0$. For $k = 0$ we have proved this above. So let $k > 0$ and suppose $\phi' \in X_{\mathbf{n} \cdot j^k}$. Since ρ 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 $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_2$) and $\phi_1 \in X_{\mathbf{n} \cdot j^k}$ or $\mathsf{EX}\phi' \in X_{\mathbf{n} \cdot j^k}$. Note that $\phi_1 \in X_{\mathbf{n} \cdot j^k}$ cannot be by assumption, hence we have $\mathsf{EX}\phi' \in X_{\mathbf{n} \cdot j^k}$. We conclude that $\mathbb{t}, \mathbf{n} \cdot j^k \models \phi_2$ for all $k \geqslant 0$.

  Hence, we have proved that the path above satisfies $\phi_1 \mathsf{R} \phi_2$, and therefore $\mathbb{t}, \mathbf{n} \models \phi'$.
- $\phi' = \mathsf{A}\phi_1 \mathsf{R} \phi_2$. The proof is a combination of the proofs for $\mathsf{A}\phi_1 \mathsf{U} \phi_2$ and $\mathsf{E}\phi_1 \mathsf{R} \phi_2$. Since $X_{\mathbf{n}}$ is consistent, we have $\phi_2 \in X_{\mathbf{n}}$, and $\phi_1 \in X_{\mathbf{n}}$ or $\mathsf{AX}\phi' \in X_{\mathbf{n}}$. In the first case, by the induction hypothesis we have $\mathbb{t}, \mathbf{n} \models \phi_1$ and $\mathbb{t}, \mathbf{n} \models \phi_2$, and hence $\mathbb{t}, \mathbf{n} \models \phi'$. In the second case, we have $\mathbb{t}, \mathbf{n} \models \phi_1$ by the induction hypothesis, and we have $\phi' \in X_{\mathbf{n} \cdot i}$ for all $0 \leqslant i \leqslant D$. We will prove that every path that starts in $\mathbf{n}$ satisfies $\phi_1 \mathsf{R} \phi_2$. So consider an arbitrary infinite path $\mathbf{n} \cdot j_1 \cdot j_2 \cdots \in [0, D]^\omega$. We distinguish two cases.
  - Suppose there exists some $k \geqslant 1$ such that $\phi_1 \in X_{\mathbf{n} \cdot j_1 \cdots j_k}$. Let k be minimal. By the induction hypothesis, $\mathbb{t}, \mathbf{n} \cdot j_1 \cdots j_k \models \phi_1$. We prove that if $\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_i}$, then $\mathbb{t}, \mathbf{n} \cdot j_1 \cdots j_i \models \phi_2$ and $\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_{i+1}}$ for all $0 \leqslant i < k$. For $i = 0$, we have proved this above. So let $0 < i < k$ and suppose $\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_i}$. Since ρ is a run of $\mathbb{A}_\phi$, $X_{\mathbf{n} \cdot j_1 \cdots j_i}$ is propositionally consistent, and hence $\phi_2 \in X_{\mathbf{n} \cdot j_1 \cdots j_i}$, and $\phi_1 \in X_{\mathbf{n} \cdot j_1 \cdots j_i}$ or $\mathsf{AX}\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_i}$. Since $\phi_1 \in X_{\mathbf{n} \cdot j_1 \cdots j_i}$ contradicts the minimality of k, we have $\mathsf{AX}\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_i}$. By the induction hypothesis, we have $\mathbb{t}, \mathbf{n} \cdot j_1 \cdots j_i \models \phi_2$, and since ρ is a run, we have $\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_{i+1}}$. We can conclude that $\mathbb{t}, \mathbf{n} \cdot j_1 \cdots j_i \models \phi_2$ for all $0 \leqslant i \leqslant k$.
  - Suppose that $\phi_1 \notin X_{\mathbf{n} \cdot j_1 \cdots j_k}$ for all $k \geqslant 0$. We prove that if $\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_k}$, then $\mathbb{t}, \mathbf{n} \cdot j_1 \cdots j_k \models \phi_2$ and $\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_{k+1}}$ for all $k \geqslant 0$. For $k = 0$ we have proved this above. So let $k > 0$ and suppose $\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_k}$. Since ρ is a run, we know that $X_{\mathbf{n} \cdot j_1 \cdots j_k}$ is propositionally consistent. Hence $\phi_2 \in X_{\mathbf{n} \cdot j_1 \cdots j_k}$ (which, by the induction hypothesis, implies $\mathbb{t}, \mathbf{n} \cdot j_1 \cdots j_k \models \phi_2$) and $\phi_1 \in X_{\mathbf{n} \cdot j_1 \cdots j_k}$ or $\mathsf{AX}\phi' \in X_{\mathbf{n} \cdot j_1 \cdots j_k}$. Note that

$\phi_1 \in X_{\mathbf{n}\cdot j_1\cdots j_k}$ cannot be by assumption, hence we have $\mathsf{AX}\phi' \in X_{\mathbf{n}\cdot j_1\cdots j_k}$. We conclude that $\mathbb{t}, \mathbf{n}\cdot j_1\cdots j_k \models \phi_2$ for all $k \geqslant 0$.

"only if:" Suppose ϕ is satisfiable. By Proposition 5.2, ϕ has a tree model $\mathbb{t}$ with domain $[0,D]^*$ that obeys the direction map ι. We prove that $\mathbb{t} \in \mathrm{L}(\mathbb{A}_\phi)$, that is, there exists some accepting run $\rho : [0,D]^* \to \delta$ of $\mathbb{A}_\phi$ on $\mathbb{t}$. Note that $\mathbb{t}$ belongs to $\mathrm{L}(\mathbb{A}_\phi)$, assuming that each node is labelled with the letter † from the single-letter alphabet Σ. Below, we omit to mention the letter from this singleton alphabet and keep $\mathbb{t}$ of the form $\mathbb{t} : [0,D]^* \to \mathbb{Z}^\beta$ to stick to the Kripke structure.

For every node $\mathbf{n}$ in $\mathbb{t}$, define $X_\mathbf{n}$ to be the set of formulas ϕ' in $\mathrm{sub}(\phi)$ such that $\mathbb{t}, \mathbf{n} \models \phi'$. Define ρ inductively as follows. First set

$$\rho(\varepsilon) \stackrel{\text{def}}{=} ((0, X_\varepsilon), \dagger, (\Theta_0, (0, X_0)), \ldots, (\Theta_D, (D, X_D)))$$

where for all $j \in [0,D]$, if there is $\mathsf{E}\ \Theta \in X_\varepsilon$ such that $\iota(\mathsf{E}\ \Theta) = j$, then

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

More generally, $\rho(\mathbf{n}\cdot j)$ is defined by

$$\rho(\mathbf{n}\cdot j) \stackrel{\text{def}}{=} ((j, X_{\mathbf{n}\cdot j}), \dagger, (\Theta_0, (0, X_{\mathbf{n}\cdot j\cdot 0})), \ldots, (\Theta_D, (D, X_{\mathbf{n}\cdot j\cdot D})))$$

where for all $j \in [0,D]$, if there is $\mathsf{E}\ \Theta \in X_{\mathbf{n}\cdot j}$ such that $\iota(\mathsf{E}\ \Theta) = j$, then

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

We prove that ρ is an accepting run of $\mathbb{A}_\phi$ on $\mathbb{t}$.

- Let us first prove a well-known property: $X_\mathbf{n}$ is propositionally consistent for all nodes $\mathbf{n} \in [0,D]^*$.
  - Suppose $\phi_1 \vee \phi_2 \in X_\mathbf{n}$. That is $\mathbb{t}, \mathbf{n} \models \phi_1$ or $\mathbb{t}, \mathbf{n} \models \phi_2$. But then also $\phi_1 \in X_\mathbf{n}$ or $\phi_2 \in X_\mathbf{n}$, and hence $\{\phi_1, \phi_2\} \cap X_\mathbf{n} \neq \varnothing$.
  - Suppose $\phi_1 \wedge \phi_2 \in X_\mathbf{n}$. That is $\mathbb{t}, \mathbf{n} \models \phi_1$ and $\mathbb{t}, \mathbf{n} \models \phi_2$. But then also $\phi_1 \in X_\mathbf{n}$ and $\phi_2 \in X_\mathbf{n}$, and hence $\{\phi_1, \phi_2\} \subseteq X_\mathbf{n}$.
  - Suppose $\mathsf{E}\phi_1\mathsf{U}\phi_2 \in X_\mathbf{n}$. We distinguish two cases. (i) Suppose $\mathbb{t}, \mathbf{n} \models \phi_2$. Then $\phi_2 \in X_\mathbf{n}$. (ii) Suppose $\mathbb{t}, \mathbf{n} \not\models \phi_2$. By $\mathbb{t}, \mathbf{n} \models \mathsf{E}\phi_1\mathsf{U}\phi_2$, we conclude that $\mathbb{t}, \mathbf{n} \models \phi_1$ and $\mathbb{t}, \mathbf{n} \models \mathsf{EXE}\phi_1\mathsf{U}\phi_2$. But then $\{\phi_1, \mathsf{EXE}\phi_1\mathsf{U}\phi_2\} \subseteq X_\mathbf{n}$.
  - Suppose $\mathsf{A}\phi_1\mathsf{U}\phi_2 \in X_\mathbf{n}$. We distinguish two cases. (i) Suppose $\mathbb{t}, \mathbf{n} \models \phi_2$. Then $\phi_2 \in X_\mathbf{n}$. (ii) Suppose $\mathbb{t}, \mathbf{n} \not\models \phi_2$. By $\mathbb{t}, \mathbf{n} \models \mathsf{A}\phi_1\mathsf{U}\phi_2$, we conclude that $\mathbb{t}, \mathbf{n} \models \phi_1$ and $\mathbb{t}, \mathbf{n} \models \mathsf{AXA}\phi_1\mathsf{U}\phi_2$. But then $\{\phi_1, \mathsf{AXA}\phi_1\mathsf{U}\phi_2\} \subseteq X_\mathbf{n}$.
  - Suppose $\mathsf{E}\phi_1\mathsf{R}\phi_2 \in X_\mathbf{n}$. That is, $\mathbb{t}, \mathbf{n} \models \mathsf{E}\phi_1\mathsf{R}\phi_2 \in X_\mathbf{n}$ and hence $\mathbb{t}, \mathbf{n} \models \phi_2$, so that $\phi_2 \in X_\mathbf{n}$, too. We distinguish two cases: (i) If $\mathbb{t}, \mathbf{n} \models \phi_1$, then $\phi_1 \in X_\mathbf{n}$. (ii) If $\mathbb{t}, \mathbf{n} \not\models \phi_1$, then $\mathbb{t}, \mathbf{n} \models \mathsf{EXE}\phi_1\mathsf{R}\phi_2$. But then $\mathsf{EXE}\phi_1\mathsf{R}\phi_2 \in X_\mathbf{n}$. Hence we can conclude that $\{\phi_1, \mathsf{EXE}\phi_1\mathsf{R}\phi_2\} \cap X_\mathbf{n} \neq \varnothing$.
  - The proof for $\mathsf{A}\phi_1\mathsf{R}\phi_2$ is analogous using the validity of $\mathsf{A}\phi_1\mathsf{R}\phi_2 \Leftrightarrow \big(\phi_2 \wedge (\phi_1 \vee \mathsf{AXA}\phi_1\mathsf{R}\phi_2)\big)$.
- Since $\mathbb{t} \models \phi$, we must have $\phi \in X_\varepsilon$, so that $(0, X_\varepsilon)$ is an initial location of $\mathbb{A}_\phi$.
- Next we prove that, for all nodes $\mathbf{n}$ with the source location of $\rho(\mathbf{n})$ being $(i, X_\mathbf{n})$, there exists a transition

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

  satisfying the conditions (1)–(3) below.

(1) Let $\mathsf{EX}\phi_1 \in X_{\mathbf{n}}$ and suppose $\iota(\mathsf{EX}\phi_1) = j$ for some $1 \leqslant j \leqslant D$. By definition of $X_{\mathbf{n}}$ we have $\mathbb{t}, \mathbf{n} \models \mathsf{EX}\phi_1$. Since $\mathbb{t}$ obeys ι, we have $\mathbb{t}, \mathbf{n} \cdot j \models \phi_1$. Hence indeed $\phi_1 \in X_{\mathbf{n}\cdot j}$ (and $\mathrm{sub}(\phi)$ is closed under subformulae).
(2) Let $\mathsf{AX}\phi_1 \in X_{\mathbf{n}}$. By definition of $X_{\mathbf{n}}$, we have $\mathbb{t}, \mathbf{n} \models \mathsf{AX}\phi_1$. By definition of the satisfaction relation for CTL($\mathbb{Z}$), we can conclude that $\mathbb{t}, \mathbf{n} \cdot j \models \phi_1$ for all $0 \leqslant j \leqslant D$. Hence $\phi_1 \in X_{\mathbf{n}\cdot j}$ for all $0 \leqslant j \leqslant D$.
(3) Let $0 \leqslant j \leqslant D$. Let $\mathsf{A}\ \Theta \in X_{\mathbf{n}}$. Hence $\mathbb{t}, \mathbf{n} \models \mathsf{A}\ \Theta$. By definition of the satisfaction relation for CTL($\mathbb{Z}$), $\mathbb{Z} \models \Theta(\boldsymbol{z}, \boldsymbol{z}_j)$, where $\boldsymbol{z} = \mathbb{t}(\mathbf{n})$ and $\boldsymbol{z}_j = \mathbb{t}(\mathbf{n} \cdot j)$. We can conclude that $\mathbb{Z} \models \left(\bigwedge_{\mathsf{A}\Theta \in X_{\mathbf{n}}} \Theta\right)(\boldsymbol{z}, \boldsymbol{z}_j)$. If, additionally, there exists $\mathsf{E}\Theta' \in X_{\mathbf{n}}$ such that $\iota(\mathsf{E}\ \Theta') = j$, then, since $\mathbb{t}$ obeys ι, we have $\mathbb{Z} \models \Theta'(\boldsymbol{z}, \boldsymbol{z}_j)$. In that case we have $\mathbb{Z} \models \left(\bigwedge_{\mathsf{A}\Theta \in X_{\mathbf{n}}} \Theta \wedge \Theta'\right)(\boldsymbol{z}, \boldsymbol{z}_j)$.

- Next we prove that ρ is accepting.

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

Suppose $\mathsf{A}\phi_1\mathsf{U}\phi_2$ is in $\mathrm{sub}_{\mathsf{AU}}(\phi)$. We show that for all nodes $\mathbf{n}$ in ρ, every path starting in $\mathbf{n}$ visits $F_{\mathsf{A}\phi_1\mathsf{U}\phi_2}$ infinitely often. Towards contradiction, suppose that there exists a node $\mathbf{n}$ and some infinite path $\mathbf{n} \cdot j_1 \cdot j_2 \cdot \cdots \in [0, D]^\omega$ that visits $F_{\mathsf{A}\phi_1\mathsf{U}\phi_2}$ only finitely. That is, there exists some $m \geqslant 0$ such that the source location in $\rho(\mathbf{n}_k)$ does not belong to $F_{\mathsf{A}\phi_1\mathsf{U}\phi_2}$ for all $k \geqslant m$. By definition of $F_{\mathsf{A}\phi_1\mathsf{U}\phi_2}$, and assuming $\rho(\mathbf{n} \cdot j_1 \ldots j_k) = (d_{\mathbf{n}\cdot j_1 \ldots j_k}, X_{\mathbf{n}\cdot j_1 \ldots j_k})$, we have $\mathsf{A}\phi_1\mathsf{U}\phi_2 \in X_{\mathbf{n}\cdot j_1 \ldots j_k}$ and $\phi_2 \notin X_{\mathbf{n}\cdot j_1 \ldots j_k}$ for all $k \geqslant m$. By definition of $X_{\mathbf{n}\cdot j_1 \ldots j_k}$, we know that $\mathbb{t}_{\mathbf{n}\cdot j_1 \ldots j_m} \models \mathsf{A}\phi_1\mathsf{U}\phi_2$. But then there must exist some $p \geqslant m$ such that $\mathbb{t}_{\mathbf{n}\cdot j_1 \ldots j_p} \models \phi_2$. By definition, $\phi_2 \in X_{\mathbf{n}\cdot j_1 \ldots j_p}$, which leads to a contradiction. □

Appendix D. Proofs for Section 6

D.1. Proof of Proposition 6.1.

Proof. The proof is a slight variant of the proof of Proposition 5.1, we keep the same notations whenever possible. First, we can establish that CTL*($\mathbb{Z}$) has the tree model property, exactly as done in the proof of Proposition 5.1 for CTL($\mathbb{Z}$). We use unfoldings for Kripke structures and we omit the details herein.

Now, we move to the construction of ϕ' in simple form. We use the notion of forward degree introduced in the proof of Proposition 5.1 that applies to CTL*($\mathbb{Z}$) state formulae and to constraints Θ. Let ϕ be a state formula in CTL*($\mathbb{Z}$) such that $\mathrm{fd}(\phi) = N$ (see page 61 the definition of fd()) and the variables occurring in ϕ are among $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$. Below, we build a formula ϕ' over the variables $\mathtt{x}_1^{-N}, \ldots, \mathtt{x}_1^0, \ldots, \mathtt{x}_\beta^{-N}, \ldots, \mathtt{x}_\beta^0$ with $\mathrm{fd}(\phi') \leqslant 1$ such that

ϕ is satisfiable in a tree Kripke structure iff ϕ' is satisfiable in a tree Kripke structure and ϕ' can be computed in polynomial-time in the size of ϕ.

Given Θ occurring in ϕ with $\mathrm{fd}(\Theta) = M$ (and therefore $M \leqslant N$), we write $\mathtt{jump}(\Theta, M)$ to denote the constraints as done in the proof of Proposition 5.1 (page 61).

Let $\mathfrak{t}$ be the translation map that is homomorphic for Boolean and temporal connectives and path quantifiers such that $\mathfrak{t}(\Theta) \stackrel{\text{def}}{=} \mathsf{X}^M\ \mathtt{jump}(\Theta, M)$, where $\mathrm{fd}(\Theta) = M$ for all maximal constraints Θ occurring in ϕ. This is the most significant change with respect to the proof of Proposition 5.1 (we have replaced $(\mathsf{EX})^M$ from the proof of Proposition 5.1 by X^M). As Θ is always in the scope of a path quantifier, the path formula $\mathsf{X}^M\ \mathtt{jump}(\Theta, M)$ is well-defined. Let ϕ' be defined as follows:

$$\mathfrak{t}(\phi) \wedge \mathsf{AG}\ \mathsf{A}\Big(\bigwedge_{j \in [1,\beta],\ k \in [0,N-1]} \mathtt{x}_j^{-k} = \mathsf{X}\mathtt{x}_j^{-k-1} \Big).$$

Observe that the second conjunct of ϕ' is identical to the case for CTL($\mathbb{Z}$) in the proof of Proposition 5.1. One can show that ϕ is satisfiable in a tree Kripke structure iff ϕ' is satisfiable in a tree Kripke structure. The main argument can be provided along the lines of the proof of Propositionn 5.1. Below, we briefly provide the essential steps.

First, suppose that $\mathcal{K}, w \models \phi$, where $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathfrak{v})$ is a *tree Kripke structure* with root w. Let $\mathcal{K}' = (\mathcal{W}, \mathcal{R}, \mathfrak{v}')$ 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 $n \geqslant 0$, for all $k \in [0, N]$ and $j \in [1, \beta]$, we require $\mathfrak{v}'(w', \mathtt{x}_j^{-k}) \stackrel{\text{def}}{=} \mathfrak{v}(w_{n-k}, \mathtt{x}_j)$ if $n - k \geqslant 0$, otherwise $\mathfrak{v}'(w', \mathtt{x}_j^{-k}) \stackrel{\text{def}}{=} 0$ (arbitrary value). To establish $\mathcal{K}', w \models \phi'$, it boils down to check the properties below.

- $\mathcal{K}', w \models \mathsf{AG}\ \mathsf{A}\big(\bigwedge_{j \in [1,\beta],\ k \in [0,N-1]} \mathtt{x}_j^{-k} = \mathsf{X}\mathtt{x}_j^{-k-1} \big)$. This holds thanks to the definition of $\mathfrak{v}'$ and the tree structure of $\mathcal{K}$.
- For every infinite path π, $\mathcal{K}, \pi \models \Theta$ implies $\mathcal{K}', \pi \models \mathsf{X}^M\ \mathtt{jump}(\Theta, M)$, where $\mathrm{fd}(\Theta) = M$. This can be lifted to all path formulae and to all state formulae (in negation normal form) by structural induction.

For the other direction, suppose that $\mathcal{K}, w \models \phi'$ and $\mathcal{K} = (\mathcal{W}, \mathcal{R}, \mathfrak{v})$ is a *tree Kripke structure* with root w. Let $\mathcal{K}' = (\mathcal{W}, \mathcal{R}, \mathfrak{v}')$ 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 $\mathfrak{v}'(w', \mathtt{x}_j) \stackrel{\text{def}}{=} \mathfrak{v}(w', \mathtt{x}_j^0)$. By structural induction, one can show that $\mathcal{K}, w \models \mathfrak{t}(\phi)$ implies $\mathcal{K}', w \models \phi$. □

D.2. **Proof of Proposition 6.2.**

Proof. By Proposition 6.1, we can assume that ϕ is in simple form, and ϕ is built over the terms $\mathtt{x}_1, \ldots, \mathtt{x}_\beta$ and $\mathtt{x}'_1, \ldots, \mathtt{x}'_\beta$. Consequently, we can assume that ϕ is in negation normal form.

We use a standard property that illustrates the renaming technique [Sco62] used below. Let ψ be a CTL*($\mathbb{Z}$) state formula with state subformula ψ' and $\mathtt{y}$ be a (fresh) variable *not occurring* in ψ. Then, ψ is satisfiable iff $\psi[\psi' \leftarrow \mathsf{E}(\mathtt{y} = 0)] \wedge \mathsf{AG}(\mathsf{E}(\mathtt{y} = 0) \Leftrightarrow \psi')$ is satisfiable, where $\psi[\psi' \leftarrow \mathsf{E}(\mathtt{y} = 0)]$ denotes the state formula obtained from ψ by replacing every occurrence of ψ' by $\mathsf{E}(\mathtt{y} = 0)$. The idea is to replace ψ' by the atomic constraint $(\mathtt{y} = 0)$, where $\mathtt{y}$ is a fresh variable not occurring before. The new constraint $(\mathtt{y} = 0)$ is tied to the original subformula ψ' via the conjunct $\mathsf{AG}(\mathsf{E}(\mathtt{y} = 0) \Leftrightarrow \psi')$. Note that strictly speaking, $\mathtt{y} = 0$ is not a state formula but morally it is because its satisfaction depends only on the current state. That is why we use $\mathsf{E}(\mathtt{y} = 0)$ instead of the more natural constraint $\mathtt{y} = 0$. In the transformations below, we need sometimes to remove E in $\mathsf{E}(\mathtt{y} = 0)$ when the occurrence of $\mathtt{y} = 0$ is already in the scope of a path quantifier. Observe that alternatively, we could slightly redefine CTL*($\mathbb{Z}$) to accept also as atomic formulae constraints Θ in which all terms are some variable $\mathtt{x}_i$ (no prefix with X). In that slight extension, $\mathtt{y} = 0$ would be authorised as a state formula.

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

$$\varphi \wedge \bigwedge_i \mathsf{AG}(\mathsf{E}(\mathtt{y}_i = 0) \Leftrightarrow \mathcal{Q}_i\ \Phi_i),$$

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

$$\overbrace{\varphi[\mathcal{Q}\ \Phi \leftarrow \mathsf{E}(\mathtt{y} = 0)]}^{\text{renaming of } \mathcal{Q}\ \Phi} \wedge \underbrace{\mathsf{AG}(\mathsf{E}(\mathtt{y} = 0) \Leftrightarrow \mathcal{Q}\ \Phi^{\dagger})}_{\text{new equivalence}} \wedge \overbrace{\bigwedge_i \mathsf{AG}(\mathsf{E}(\mathtt{y}_i = 0) \Leftrightarrow \mathcal{Q}_i\ \Phi_i)}^{\text{conjunction already in } \psi},$$

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

$$\varphi \wedge \bigwedge_i \mathsf{AG}(\mathsf{E}(\mathtt{y}_i = 0) \Leftrightarrow \mathcal{Q}_i\ \Phi_i),$$

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

$$\mathsf{E}\ (\mathtt{z} = 0) \wedge \mathsf{AG}(\overline{\mathtt{z} = 0 \Leftrightarrow \varphi^{\dagger}}) \wedge \bigwedge_i \mathsf{AG}(\mathsf{E}(\mathtt{x}_i = 0) \Leftrightarrow \mathcal{Q}_i\ \Phi_i),$$

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

It remains to explain how to transform each $\mathsf{AG}(\mathsf{E}(\mathtt{x}_i = 0) \Leftrightarrow \mathcal{Q}_i\ \Phi_i)$ so that we get the final formula ϕ' in special form from $\phi^{\star}$. For instance, $\mathsf{AG}(\mathsf{E}(\mathtt{y}_i = 0) \Leftrightarrow \mathsf{A}\ \Phi_i)$ shall be replaced by

$$\mathsf{AG}(\neg(\mathtt{y}_i = 0) \vee \Phi_i) \quad \wedge \quad \mathsf{AGE}\ (\mathtt{y}_i = 0 \vee \overline{\neg\Phi_i}),$$

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

Below, we list the logical equivalences (hinted above) we take advantage of and that are slight variants of equivalences used for CTL* in the proof of [ES84, Theorem 3.1].

- $\mathsf{AG}(\mathsf{E}(\mathtt{y}_i = 0) \Rightarrow \mathsf{A}\ \Phi_i) \ \Leftrightarrow \mathsf{AG}(\neg(\mathtt{y}_i = 0) \vee \Phi_i)$.
- $\mathsf{AG}(\mathsf{E}(\mathtt{y}_i = 0) \Rightarrow \mathsf{E}\ \Phi_i) \ \Leftrightarrow \mathsf{AGE}(\neg(\mathtt{y}_i = 0) \vee \Phi_i)$.
- $\mathsf{AG}(\neg\mathsf{E}(\mathtt{y}_i = 0) \Rightarrow \neg\mathsf{A}\ \Phi_i) \ \Leftrightarrow \mathsf{AGE}\ (\mathtt{y}_i = 0 \vee \overline{\neg\Phi_i})$.
- $\mathsf{AG}(\neg\mathsf{E}(\mathtt{y}_i = 0) \Rightarrow \neg\mathsf{E}\ \Phi_i) \ \Leftrightarrow \mathsf{AG}((\mathtt{y}_i = 0) \vee \overline{\neg\Phi_i})$.

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

Example D.1. To illustrate the construction from the above proof of Proposition 6.2, we consider the formula ϕ below.

$$\phi = \mathsf{E}((\mathtt{x}_1' < \mathtt{x}_1)\mathsf{U}\ \mathsf{AX}(\mathtt{x}_2 = \mathtt{x}_2'))\ \wedge\ \mathsf{EG}(\mathtt{x}_1 < \mathtt{x}_2).$$

Below, we present the formulae $\psi_0 = \phi$, ψ_1, ψ_2, ψ_3, $\phi^\star$ obtained by application of the different renaming steps.

$$\psi_1 = \overbrace{\mathsf{AG}\big(\mathsf{E}(\mathtt{y}_1 = 0) \Leftrightarrow (\mathsf{AX}(\mathtt{x}_2 = \mathtt{x}_2'))\big)}^{=\ \psi_1'} \wedge \mathsf{E}((\mathtt{x}_1' < \mathtt{x}_1)\mathsf{U}\ \mathsf{E}(\mathtt{y}_1 = 0))\ \wedge\ \mathsf{EG}(\mathtt{x}_1 < \mathtt{x}_2).$$

$$\psi_2 = \overbrace{\mathsf{AG}\big(\mathsf{E}(\mathtt{y}_2 = 0) \Leftrightarrow (\mathsf{E}\ (\mathtt{x}_1' < \mathtt{x}_1)\mathsf{U}\ \mathtt{y}_1 = 0)\big)}^{=\ \psi_2'} \wedge \psi_1'\ \wedge\ \mathsf{E}(\mathtt{y}_2 = 0)\ \wedge\ \mathsf{EG}(\mathtt{x}_1 < \mathtt{x}_2).$$

Note that ‘E’ is removed from $\mathsf{E}(\mathtt{y}_1 = 0)$ in ψ_2'.

$$\psi_3 = \overbrace{\mathsf{AG}\big(\mathsf{E}(\mathtt{y}_3 = 0) \Leftrightarrow \mathsf{EG}(\mathtt{x}_1 < \mathtt{x}_2)\big)}^{=\ \psi_3'} \wedge \psi_2'\ \wedge\ \psi_1'\ \wedge\ \mathsf{E}(\mathtt{y}_2 = 0)\ \wedge\ \mathsf{E}(\mathtt{y}_3 = 0).$$

$$\phi^\star = \mathsf{AG}\big((\mathtt{y}_4 = 0) \Leftrightarrow (\mathtt{y}_2 = 0 \wedge \mathtt{y}_3 = 0)\big)\ \wedge\ \psi_3'\ \wedge\ \psi_2'\ \wedge\ \psi_1'\ \wedge\ \mathsf{E}(\mathtt{y}_4 = 0).$$

Each state formula ψ_i' in $\phi^\star$ is then also replaced by a conjunction of two state formulae in order to compute ϕ'. By way of example, we present below the conjunction replacing ψ_2'.

$$\mathsf{AGE}\big(\neg(\mathtt{y}_2 = 0) \vee (\mathtt{x}_1' < \mathtt{x}_1)\mathsf{U}\ (\mathtt{y}_1 = 0)\big)\ \wedge\ \mathsf{AG}\big((\mathtt{y}_2 = 0) \vee (\neg(\mathtt{x}_1' < \mathtt{x}_1))\mathsf{R}\ \neg(\mathtt{y}_1 = 0)\big)$$

D.3. **Proof of Proposition 6.4.**

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

- $\Phi \in \mathrm{sub}(\Phi)$; $\mathrm{sub}(\Phi)$ is closed under subformulae,
- for all $\mathsf{Op} \in \{\mathsf{U}, \mathsf{R}\}$, if $\Phi_1\ \mathsf{Op}\ \Phi_2 \in \mathrm{sub}(\Phi)$, then $\mathsf{X}(\Phi_1\ \mathsf{Op}\ \Phi_2) \in \mathrm{sub}(\Phi)$.

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

- If $\Phi_1 \vee \Phi_2 \in X$, then $\{\Phi_1, \Phi_2\} \cap X \neq \varnothing$; if $\Phi_1 \wedge \Phi_2 \in X$, then $\{\Phi_1, \Phi_2\} \subseteq X$.
- If $\Phi_1\mathsf{U}\Phi_2 \in X$, then $\Phi_2 \in X$ or $\{\Phi_1, \mathsf{X}(\Phi_1\mathsf{U}\Phi_2)\} \subseteq X$.
- If $\Phi_1\mathsf{R}\Phi_2 \in X$, then $\Phi_2 \in X$ and $\{\Phi_1, \mathsf{X}(\Phi_1\mathsf{R}\Phi_2)\} \cap X \neq \varnothing$.

We write $\mathrm{sub}_{\mathsf{X}}(\Phi)$ to denote the set of formulae in $\mathrm{sub}(\Phi)$ of the form $\mathsf{X}\Phi'$. Similarly, we write $\mathrm{sub}_{\mathsf{U}}(\Phi)$ to denote the set of formulae in $\mathrm{sub}(\Phi)$ of the form $\Phi_1\mathsf{U}\Phi_2$. Finally, we write $\mathrm{sub}_{\mathrm{cons}}(\Phi)$ to denote the set of formulae of the form Θ in $\mathrm{sub}(\Phi)$.

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

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

Transforming generalised Büchi conditions to standard Büchi conditions leads to a set of locations multiplied by the factor $\mathrm{card}(F) + 1$ (which is bounded by $\texttt{size}(\Phi)$) 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 $\texttt{size}(\Phi)$. We omit the standard proof for correctness (similar to the proof of Lemma 5.4). □

D.4. **Proof of Lemma 6.6.**

Proof. Let us start checking that (I) and (II) hold true. We have $Q' = [0, D-1] \times (Q \cup \{\perp\})$ with $\mathrm{card}(Q)$ exponential in $\texttt{size}(\Phi_i)$ (and therefore exponential in $\texttt{size}(\phi)$) by Lemma 6.4(I). Moreover, D is linear in $\texttt{size}(\phi)$, whence $\mathrm{card}(Q')$ is exponential in $\texttt{size}(\phi)$. By Lemma 6.4(II), $\texttt{MCS}(\mathbb{A})$ is quadratic in $\texttt{size}(\Phi_i)$ (and therefore quadratic in $\texttt{size}(\phi)$). The constraints in $\mathbb{A}_i$ are those from $\mathbb{A}$ (maybe except $\top$). Consequently, $\texttt{MCS}(\mathbb{A}_i)$ is quadratic in $\texttt{size}(\phi)$. This concludes the proof of (I) and (II).

Let $\phi = \mathsf{AGE}\ \Phi_i$ and $\mathbb{t} : [0, D-1]^* \to \Sigma \times \mathbb{Z}^\beta$ be a tree such that $\mathbb{t} \in \mathrm{L}(\mathbb{A}_i)$. Below, we prove that for all nodes $\mathbf{n} \in [0, D-1]^*$, the word $\mathbb{t}(\mathbf{n})\ \mathbb{t}(\mathbf{n} \cdot i)\ \mathbb{t}(\mathbf{n} \cdot i \cdot 0)\ \mathbb{t}(\mathbf{n} \cdot i \cdot 0^2) \ldots$ satisfies Φ_i. This implies $\mathbb{t} \models \phi$ and $\mathbb{t}$ satisfies ϕ via i.

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

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

Let $\mathbf{n} \in [0, D-1]^*$ be an arbitrary node in $\mathbb{t}$. Let us assume $\mathbb{t}(\mathbf{n}) = (\mathsf{a}, \boldsymbol{z})$ and $\mathbb{t}(\mathbf{n} \cdot i \cdot 0^k) = (\mathsf{a}_k, \boldsymbol{z}_k)$ for all $k \geqslant 0$, and we use $w_\mathbf{n}$ to denote the corresponding infinite word $(\mathsf{a}, \boldsymbol{z})(\mathsf{a}_0, \boldsymbol{z}_0)(\mathsf{a}_1, \boldsymbol{z}_1) \ldots$. Since $\mathbb{t}$ satisfies ϕ via i, we know that $w_\mathbf{n} \models \Phi_i$. But then also $w_\mathbf{n} \in \mathrm{L}(\mathbb{A})$. So there must exist some accepting run of $\mathbb{A}$ on $w_\mathbf{n}$, say with projection on the set of locations equal to $q^{\mathbf{n}}_{\text{in}}, q^{\mathbf{n}}_0, q^{\mathbf{n}}_1, q^{\mathbf{n}}_2 \ldots$ such that

- $q^{\mathbf{n}}_{\text{in}} \in Q_{\text{in}}$,
- there exist transitions $(q^{\mathbf{n}}_{\text{in}}, \mathsf{a}, \Theta, q^{\mathbf{n}}_0) \in \delta$ and $(q^{\mathbf{n}}_k, \mathsf{a}_k, \Theta_k, q^{\mathbf{n}}_{k+1}) \in \delta$ for all $k \geqslant 0$, with $\mathbb{Z} \models \Theta(\boldsymbol{z}, \boldsymbol{z}_0)$ and $\mathbb{Z} \models \Theta_k(\boldsymbol{z}_k, \boldsymbol{z}_{k+1})$ for all $k \geqslant 0$.

The definition of δ' now allows us to define $\rho(\mathbf{n} \cdot i) = ((i, q^{\mathbf{n}}_0), \dagger, \cdots, (i, q^{\mathbf{n} \cdot i}_0), \cdots)$, and $\rho(\mathbf{n} \cdot i \cdot 0^k) = ((0, q^{\mathbf{n}}_k), \dagger, \cdots, (i, q^{\mathbf{n} \cdot i \cdot 0^k}_0), \cdots)$ for every $k \geqslant 1$. ☐

Appendix E. Proofs for Section 7

E.1. **Proof of Lemma 7.1.**

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 $H \geqslant 1$. We prove the claim for $H + 1$. Suppose the root node of $\mathbf{s}$ has k children, denoted by $\mathbf{n}_1, \ldots, \mathbf{n}_k$. For every $1 \leqslant i \leqslant k$, let $Q_i \subseteq Q$ denote the label of $\mathbf{n}_i$. For every $1 \leqslant i \leqslant k$, 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 $\mathrm{card}(Q_i)$ nodes. By condition 5 of Safra trees, the sets $Q_1, \ldots, Q_k$ are pairwise disjoint. By condition 4, the union $\bigcup_{1 \leqslant i \leqslant k} Q_i$ is a proper subset of Q. Hence $\sum_{i=1}^{k} \mathrm{card}(Q_i) < \mathrm{card}(Q)$. Altogether, the number of nodes in $\mathbf{s}$ is at most $1 + \sum_{i=1}^{k} \mathrm{card}(Q_i) < 1 + \mathrm{card}(Q) \leqslant \mathrm{card}(Q)$. ☐

E.2. **Proof of Lemma 7.3.**

Proof. Suppose $i \geqslant 1$, $1 \leqslant J_i \leqslant 2 \cdot \mathrm{card}(Q)$, and for all $k \geqslant i$, $\mathbf{s}_k$ contains a node with name J_i and $q_k \in \mathrm{Lab}(\mathbf{s}_k, J_i)$. If the first property holds, we are done. Otherwise, there exists some position $m \geqslant i$ such that

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

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

- $\mathbf{s}_k$ contains a node with name J_m and $q_k \in \mathrm{Lab}(\mathbf{s}_k, J_m)$ for all $m + 1 \leqslant k < p$, and
- $\mathbf{s}_p$ does not contain a node with name J_m, or $q_p \notin \mathrm{Lab}(\mathbf{s}_p, J_m)$.

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

(a) The location q_p is removed from the label of the node with name J_m, because there exists some younger sibling of the node named J_m (that is, an older child of J_i) whose label set contains q_p.

(b) The node with name J_m has been removed from the Safra tree during step (5). But for this, the node with the name J_m must have an empty label set, contradicting $q_{p-1} \in \text{Lab}(\mathbf{s}_{p-1}, J_m)$ and there is $(q_{p-1}, \mathtt{a}_{p-1}, \Theta'_{p-1}, q_p) \in \delta$ such that $\mathbb{Z} \models \Theta'_{p-1}(\boldsymbol{z}_{p-1}, \boldsymbol{z}_p)$– so this case cannot occur.

(c) The node with name J_m has been removed from the Safra tree during step (6). But for this the parent node with name J_i 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 7.1, the node with name J_i can have at most $\text{card}(Q) - 1$ children nodes. We can conclude that there must exist some position $i' \geqslant i$ and some name $1 \leqslant J_{i'} \leqslant 2 \cdot \text{card}(Q)$ with $J_i \neq J_{i'}$ such that for all $k \geqslant i'$, $\mathbf{s}_k$ contains a node with name $J_{i'}$ and $q_k \in \text{Lab}(\mathbf{s}_k, J_{i'})$. ☐

E.3. **Proof of Nonemptiness of Acc**(q, j) **and Pre**(q, j, k)**.** In this section, we prove that the sets $\text{Acc}(q, j)$ and $\text{Pre}(q, j, k)$ are nonempty as stated in forthcoming Lemma E.2. Before, we prove one helpful lemma, also related to the correctness of the Safra construction.

Lemma E.1. *For all infinite runs ρ' of the form*

$$(\mathbf{s}_1, \mathtt{a}_1, \Theta_1, \mathbf{s}_2)(\mathbf{s}_2, \mathtt{a}_2, \Theta_2, \mathbf{s}_3)(\mathbf{s}_3, \mathtt{a}_3, \Theta_3, \mathbf{s}_4)\ldots$$

of $\mathbb{A}'$ on $(\mathtt{a}_1, \boldsymbol{z}_1)(\mathtt{a}_2, \boldsymbol{z}_2)(\mathtt{a}_3, \boldsymbol{z}_3)\ldots$, for every $1 \leqslant J \leqslant 2 \cdot \text{card}(Q)$ and for every $1 \leqslant j \leqslant j'$, if $\mathbf{s}_k$ contains a node with name J for all $j \leqslant k \leqslant j'$, then for every $q' \in Lab(\mathbf{s}_{j'}, J)$ there exist some $q \in Lab(\mathbf{s}_j, J)$ and some finite run

$$(q_j, \mathtt{a}_j, \Theta'_j, q_{j+1})\ldots(q_{j'-1}, \mathtt{a}_{j'-1}, \Theta'_{j'-1}, q_{j'})$$

of $\mathbb{A}$ on $(\mathtt{a}_j, \boldsymbol{z}_j)(\mathtt{a}_{j+1}, \boldsymbol{z}_{j+1})\ldots(\mathtt{a}_{j'}, \boldsymbol{z}_{j'})$ with $q_j = q$ and $q_{j'} = q'$.

Proof. The proof is by induction on the difference $\Delta = j' - j$. For the induction base, let $\Delta = 0$. By convention, q' is a finite run on $(\mathtt{a}_{j'}, \boldsymbol{z}_{j'})$ for every $q' \in \text{Lab}(\mathbf{s}_{j'}, J) = \text{Lab}(\mathbf{s}_j, J)$. Suppose the claim holds for $\Delta \geqslant 0$; we prove it for $\Delta+1$. So suppose $q' \in \text{Lab}(\mathbf{s}_{j'}, J)$. Consider the transition step $(\mathbf{s}_{j'-1}, \mathtt{a}_{j'-1}, \Theta_{j'-1}, \mathbf{s}_{j'}) \in \delta'$ used in ρ'. By step (3) of the definition of δ', there exists some $q'' \in \text{Lab}(\mathbf{s}_{j'-1}, J)$ and some transition $(q'', \mathtt{a}_{j'-1}, \Theta'_{j'-1}, q') \in \delta$ such that $\Theta_{j'-1} \models \Theta'_{j'-1}$. From ρ' being a run, we obtain that $\mathbb{Z} \models \Theta_{j'-1}(\boldsymbol{z}_{j'-1}, \boldsymbol{z}_{j'})$, hence also $\mathbb{Z} \models \Theta'_{j'-1}(\boldsymbol{z}_{j'-1}, \boldsymbol{z}_{j'})$. Hence $\rho_2 = (q'', \mathtt{a}_{j'-1}, \Theta'_{j'-1}, q')$ is a finite run of $\mathbb{A}$ on $(\mathtt{a}_{j'-1}, \boldsymbol{z}_{j'-1})(\mathtt{a}_{j'}, \boldsymbol{z}_{j'})$. By the induction hypothesis, there exist $q \in \text{Lab}(\mathbf{s}_j, J)$ and some finite run

$$\rho_1 = (q_j, \mathtt{a}_j, \Theta'_j, q_{j+1})\ldots(q_{j'-2}, \mathtt{a}_{j'-2}, \Theta'_{j'-2}, q_{j'-1})$$

of $\mathbb{A}$ on $(\mathtt{a}_j, \boldsymbol{z}_j)(\mathtt{a}_{j+1}, \boldsymbol{z}_{j+1})\ldots(\mathtt{a}_{j'-1}, \boldsymbol{z}_{j'-1})$ with $q_j = q$ and $q_{j'} = q''$. The final run is obtained by composing ρ_1 and ρ_2. ☐

Lemma E.2. *For every $j \geqslant 1$ and for every $q \in Lab(\mathbf{s}_{i_j}, J)$, $\text{Acc}(q, j) \neq \varnothing$, and for every $j \geqslant 2$, every $i_{j-1} \leqslant k < i_j$ and every $q \in Lab(\mathbf{s}_k, J) \cap F$, $Pre(q, j, k) \neq \varnothing$.*

Proof. Let $j \geqslant 1$ and $q \in \text{Lab}(\mathbf{s}_{i_j}, J)$. We prove that $\text{Acc}(q, j) \neq \varnothing$. By definition of δ', $\text{Lab}(\mathbf{s}_{i_j}, J) = \text{Lab}((\mathbf{s}_{i_j-1})^{(5)}, J)$, hence $q \in \text{Lab}((\mathbf{s}_{i_j-1})^{(5)}, J)$. Since the node with name J is marked in $\mathbf{s}_{i_j}$ (step (6)), it must have some child node in $(\mathbf{s}_{i_j-1})^{(5)}$, say with name $K \neq J$, such that $q \in \text{Lab}((\mathbf{s}_{i_j-1})^{(5)}, K)$. Then there exists $q' \in \text{Lab}((\mathbf{s}_{i_j-1})^{(2)}, K)$ such that $(q', \mathtt{a}_{i_j-1}, \Theta'_{i_j-1}, q)$ is a finite run of $\mathbb{A}$ on $(\mathtt{a}_{i_j-1}, \boldsymbol{z}_{i_j-1})(\mathtt{a}_{i_j}, \boldsymbol{z}_{i_j})$. By property (4) of

Safra trees, we know that $q' \in \text{Lab}((\mathbf{s}_{i_j-1})^{(2)}, J)$. If $q' \in F$, then we are done. Otherwise, K is a node in $\mathbf{s}_{i_j-1}$ (it was not created as a youngest child of J by step (2) of δ'). Let $i_{j-1} < k < i_j - 1$ be the minimal position such that node named J has no child node with name K in $\mathbf{s}_{k-1}$, and node named J has a child node with name K in $\mathbf{s}_{k'}$ for all $k \leqslant k' \leqslant i_j - 1$. Such a position necessarily exists; indeed, for all $m \geqslant 0$, node J in $\mathbf{s}_{i_m}$ has no children nodes: for $\mathbf{s}_{i_0}$, this is because J is freshly introduced by step (2), and for all $m > 1$, this is because the node with name J in $\mathbf{s}_{i_m}$ is marked and marked nodes (step (6)) do not have children nodes. By Lemma E.1, there exists $q_k \in \text{Lab}(\mathbf{s}_k, K)$ and some finite run $(q_k, \mathsf{a}_k, \Theta'_k, q_{k+1}) \ldots (q_{i_j-2}, \mathsf{a}_{i_j-2}, \Theta'_{i_j-2}, q_{i_j-1})$ of $\mathbb{A}$ on $(\mathsf{a}_k, \boldsymbol{z}_k) \ldots (\mathsf{a}_{i_j-1}, \boldsymbol{z}_{i_j-1})$, where $q_{i_j-1} = q'$. Finally, using the fact that $\mathbf{s}_k$ contains the node with name K, whereas $\mathbf{s}_{i_j-1}$ does not, it is not hard to prove that there exists $q_{k-1} \in \text{Lab}(\mathbf{s}_{k-1}, J) \cap F$ such that $(q_{k-1}, \mathsf{a}_{k-1}, \Theta'_{k-1}, q_k)$ is a finite run of $\mathbb{A}$ on $(\mathsf{a}_{k-1}, \boldsymbol{z}_{k-1})(\mathsf{a}_k, \boldsymbol{z}_k)$. Hence $q_{k-1} \in \text{Acc}(q, j)$, which finishes the proof.

Let $j \geqslant 2$, $i_{j-1} \leqslant k < i_j$ and $q \in \text{Lab}(\mathbf{s}_k, J) \cap F$. For proving that $\text{Pre}(q, j, k) \neq \varnothing$, we can apply Lemma E.1. □

Appendix F. Proofs for Section 8

F.1. **Proof of Lemma 8.1.**

Proof. Let $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathtt{xd}_n)$ be a path in G_t^{C} such that $\mathbf{n}_0$ and $\mathbf{n}_n$ are neighbours.

(Property 1) Firstly, if $n \geqslant 2$, 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 $i \in [1, n-1]$, $|\mathbf{n}_i| \geqslant |\mathbf{n}_0|$. $h \stackrel{\text{def}}{=} n - 1$.
- 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 $i \in [1, n-1]$, $|\mathbf{n}_i| \geqslant |\mathbf{n}_n|$. $h \stackrel{\text{def}}{=} 1$. 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 $h \leqslant n - 1$, $n - h \leqslant n - 1$, and that $\mathbf{n}_0$ and $\mathbf{n}_h$ are neighbours, and $\mathbf{n}_n$ and $\mathbf{n}_h$ are neighbours.

(Property 2) Second, by construction of G_t^{C} from t built over satisfiable constraints in $\text{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$, we can show the property below (this requires a lengthy case analysis). Let $(\mathbf{m}_1, \mathtt{xd}_1)$, $(\mathbf{m}_2, \mathtt{xd}_2)$ and $(\mathbf{m}_3, \mathtt{xd}_3)$ be nodes in the graph G_t^{C} that are pairwise neighbours such that $(\mathbf{m}_1, \mathtt{xd}_1) \xrightarrow{\sim_1} (\mathbf{m}_2, \mathtt{xd}_2)$ and $(\mathbf{m}_2, \mathtt{xd}_2) \xrightarrow{\sim_2} (\mathbf{m}_3, \mathtt{xd}_3)$ with $\sim_1, \sim_2 \in \{<, =\}$. If $< \in \{\sim_1, \sim_2\}$, then $(\mathbf{m}_1, \mathtt{xd}_1) \xrightarrow{<} (\mathbf{m}_3, \mathtt{xd}_3)$ otherwise $(\mathbf{m}_1, \mathtt{xd}_1) \xrightarrow{=} (\mathbf{m}_3, \mathtt{xd}_3)$ (this uses the local consistency of t).

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 $n \leqslant K$ and let $\pi = (\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} \cdots \xrightarrow{\sim_n} (\mathbf{n}_n, \mathtt{xd}_n)$ be a path in G_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, $h \leqslant K$ and

$n - h \leqslant K$. By the induction hypothesis, we have $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{<} (\mathbf{n}_h, \mathtt{xd}_h)$ if $< \in \{\sim_1, \ldots, \sim_h\}$, otherwise $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{=} (\mathbf{n}_h, \mathtt{xd}_h)$. Similarly, we have $(\mathbf{n}_h, \mathtt{xd}_h) \xrightarrow{<} (\mathbf{n}_n, \mathtt{xd}_n)$ if $< \in \{\sim_{h+1}, \ldots, \sim_n\}$, otherwise $(\mathbf{n}_h, \mathtt{xd}_h) \xrightarrow{=} (\mathbf{n}_n, \mathtt{xd}_n)$. By (Property 2), we get $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{<} (\mathbf{n}_n, \mathtt{xd}_n)$ if $< \in \{\sim_1, \ldots, \sim_n\}$, otherwise $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{=} (\mathbf{n}_n, \mathtt{xd}_n)$. □

F.2. **Proof of Lemma 8.3.**

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

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

Ad absurdum, suppose there exists $(\mathbf{n}, \mathtt{x}_i) \in U_{<\mathbb{d}_1} \cup U_{>\mathbb{d}_\alpha}$ in G^{C}_t such that $\mathrm{slen}(\mathbf{n}, \mathtt{x}_i) = \omega$. We prove the claim for the case $(\mathbf{n}, \mathtt{x}_i) \in U_{<\mathbb{d}_1}$; the proof for $(\mathbf{n}, \mathtt{x}) \in U_{>\mathbb{d}_\alpha}$ is analogous. Recall that, by definition, $\mathrm{slen}(\mathbf{n}, \mathtt{x}_i) = \mathrm{slen}((\mathbf{n}, \mathtt{x}_i), (\mathbf{n}, \mathbb{d}_1))$. Define $\Delta = \mathbb{d}_1 - \mathbb{t}(\mathbf{n})(\mathtt{x}_i)$. Let π be a path $(\mathbf{n}_0, \mathtt{xd}_0) \xrightarrow{\sim_1} (\mathbf{n}_1, \mathtt{xd}_1) \xrightarrow{\sim_2} \ldots \xrightarrow{\sim_k} (\mathbf{n}_k, \mathtt{xd}_k)$ such that $(\mathbf{n}_0, \mathtt{xd}_0) = (\mathbf{n}, \mathtt{x}_i)$, $(\mathbf{n}_k, \mathtt{xd}_k) = (\mathbf{n}, \mathbb{d}_1)$, and $\mathrm{slen}(\pi) > \Delta$. Such a path must exist by assumption because $\mathrm{slen}(\mathbf{n}, \mathtt{x}_i) = \omega$. By Lemma 4.9, we have $\mathbb{t}(\mathbf{n}_{i-1})(\mathtt{xd}_{i-1}) \sim_i \mathbb{t}(\mathbf{n}_i)(\mathtt{xd}_i)$ for all $1 \leqslant i \leqslant k$. But this implies that there are more than Δ different data values in the interval $[\mathbb{t}(\mathbf{n}_0)(\mathtt{xd}_0), \mathbb{t}(\mathbf{n}_k)(\mathtt{xd}_k)] = [\mathbb{t}(\mathbf{n})(\mathtt{x}_i), \mathbb{t}(\mathbf{n})(\mathbb{d}_1)]$, which leads to a contradiction.

(II) ⇒ (I) Suppose that for all $(\mathbf{n}, \mathtt{x})$ in $(U_{<\mathbb{d}_1} \cup U_{>\mathbb{d}_\alpha})$ in G^{C}_t we have $\mathrm{slen}(\mathbf{n}, \mathtt{x}) < \omega$. We define the mapping $g : [0, D-1]^* \times \{\mathtt{x}_1, \ldots, \mathtt{x}_\beta\} \to \mathbb{Z}$ as follows:

- $g(\mathbf{n}, \mathtt{x}) \stackrel{\text{def}}{=} \mathbb{d}$ if $(\mathbf{n}, \mathtt{x}) \in U_{\mathbb{d}}$ for some $\mathbb{d} \in [\mathbb{d}_1, \mathbb{d}_\alpha]$,
- $g(\mathbf{n}, \mathtt{x}) \stackrel{\text{def}}{=} \mathbb{d}_1 - \mathrm{slen}(\mathbf{n}, \mathtt{x})$ if $(\mathbf{n}, \mathtt{x}) \in U_{<\mathbb{d}_1}$, and
- $g(\mathbf{n}, \mathtt{x}) \stackrel{\text{def}}{=} \mathbb{d}_\alpha + \mathrm{slen}(\mathbf{n}, \mathtt{x})$ if $(\mathbf{n}, \mathtt{x}) \in U_{>\mathbb{d}_\alpha}$.

Recall that $\{U_{\mathbb{d}} \mid \mathbb{d} \in [\mathbb{d}_1, \mathbb{d}_\alpha]\} \cup \{U_{<\mathbb{d}_1}, U_{>\mathbb{d}_\alpha}\}$ is a partition of $[0, D-1]^* \times \mathtt{T}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$ so that g is indeed well-defined. Now define $\mathbb{t}' : [0, D-1]^* \to \Sigma \times \mathbb{Z}^\beta$ by $\mathbb{t}'(\mathbf{n}) \stackrel{\text{def}}{=} (\mathtt{a}, (g(\mathbf{n}, \mathtt{x}_1), \ldots, g(\mathbf{n}, \mathtt{x}_\beta)))$ for all $\mathbf{n} \in [0, D-1]^*$ with $t(\mathbf{n}) = (\mathtt{a}, \cdot)$. We prove that $\mathbb{t}'$ witnesses the satisfaction of t. For this, we verify that for all $\mathbf{n} \cdot j \in [0, D-1]^+$ with $t(\mathbf{n} \cdot j) = (\mathtt{a}, \Theta)$, we have $\mathbb{Z} \models \Theta(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$. Moreover, concerning the case with the root ε, one can show that $\mathbb{Z} \models \Theta(\mathbf{0}, \mathbb{t}'(\varepsilon))$ with $t(\varepsilon) = (\mathtt{a}, \Theta)$ but we omit it below as it is very similar to the general case.

- Suppose $\mathtt{x}' = \mathbb{d} \in \Theta$ for some $\mathbb{d} \in [\mathbb{d}_1, \mathbb{d}_\alpha]$. By definition of G^{C}_t, we have $(\mathbf{n} \cdot j, \mathtt{x}) \in U_{\mathbb{d}}$. By definition, $g(\mathbf{n} \cdot j, \mathtt{x}) = \mathbb{d}$ and therefore $\mathbb{Z} \models (\mathtt{x}' = \mathbb{d})(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
- Suppose $\mathtt{x} = \mathbb{d} \in \Theta$ for some $\mathbb{d} \in [\mathbb{d}_1, \mathbb{d}_\alpha]$. By definition of G^{C}_t, and as t is locally consistent, we have $(\mathbf{n}, \mathtt{x}) \in U_{\mathbb{d}}$. By definition, $g(\mathbf{n}, \mathtt{x}) = \mathbb{d}$ and therefore $\mathbb{Z} \models (\mathtt{x} = \mathbb{d})(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
- Suppose $\mathtt{x}' < \mathbb{d}_1 \in \Theta$. By definition of G^{C}_t, we have $(\mathbf{n} \cdot j, \mathtt{x}) \in U_{<\mathbb{d}_1}$. By definition, $g(\mathbf{n} \cdot j, \mathtt{x}) = \mathbb{d}_1 - \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x})$. By definition of G^{C}_t, we also have $(\mathbf{n} \cdot j, \mathbb{d}_1) \in U_{\mathbb{d}_1}$ and hence $(\mathbf{n} \cdot j, \mathtt{x}) \xrightarrow{<} (\mathbf{n} \cdot j, \mathbb{d}_1)$. Hence $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x}) \geqslant 1$, so that indeed $g(\mathbf{n} \cdot j, \mathtt{x}) < \mathbb{d}_1$ and $\mathbb{Z} \models (\mathtt{x}' < \mathbb{d}_1)(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
- Suppose $\mathtt{x} < \mathbb{d}_1 \in \Theta$. By definition of G^{C}_t, and as t is locally consistent, we have $(\mathbf{n}, \mathtt{x}) \in U_{<\mathbb{d}_1}$. By definition, $g(\mathbf{n}, \mathtt{x}) = \mathbb{d}_1 - \mathrm{slen}(\mathbf{n}, \mathtt{x})$. By definition of G^{C}_t, we also have

$(\mathbf{n}, \mathbb{d}_1) \in U_{\mathbb{d}_1}$ and hence $(\mathbf{n}, \mathtt{x}) \xrightarrow{\leq} (\mathbf{n}, \mathbb{d}_1)$. Hence $\mathrm{slen}(\mathbf{n}, \mathtt{x}) \geqslant 1$, so that indeed $g(\mathbf{n}, \mathtt{x}) < \mathbb{d}_1$ and $\mathbb{Z} \models (\mathtt{x} < \mathbb{d}_1)(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.

- Suppose $\mathtt{x}' > \mathbb{d}_\alpha \in \Theta$. By definition of G_t^{C}, we have $(\mathbf{n} \cdot j, \mathtt{x}) \in U_{>\mathbb{d}_\alpha}$. By definition, $g(\mathbf{n} \cdot j, \mathtt{x}) = \mathbb{d}_\alpha + \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x})$. We also have $(\mathbf{n} \cdot j, \mathbb{d}_\alpha) \in U_{\mathbb{d}_\alpha}$ and hence $(\mathbf{n} \cdot j, \mathbb{d}_\alpha) \xrightarrow{\leq} (\mathbf{n} \cdot j, \mathtt{x})$. Hence $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x}) \geqslant 1$, so that indeed $g(\mathbf{n} \cdot j, \mathtt{x}) > \mathbb{d}_\alpha$ and therefore $\mathbb{Z} \models (\mathtt{x}' > \mathbb{d}_\alpha)(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
- Suppose $\mathtt{x} > \mathbb{d}_\alpha \in \Theta$. By definition of G_t^{C} and as t is locally consistent, we have $(\mathbf{n}, \mathtt{x}) \in U_{>\mathbb{d}_\alpha}$. By definition, $g(\mathbf{n}, \mathtt{x}) = \mathbb{d}_\alpha + \mathrm{slen}(\mathbf{n}, \mathtt{x})$. We also have $(\mathbf{n}, \mathbb{d}_\alpha) \in U_{\mathbb{d}_\alpha}$ and hence $(\mathbf{n}, \mathbb{d}_\alpha) \xrightarrow{\leq} (\mathbf{n}, \mathtt{x})$. Hence $\mathrm{slen}(\mathbf{n}, \mathtt{x}) \geqslant 1$, so that indeed $g(\mathbf{n}, \mathtt{x}) > \mathbb{d}_\alpha$ and therefore $\mathbb{Z} \models (\mathtt{x} > \mathbb{d}_\alpha)(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
- Suppose $\mathtt{x}' < \mathtt{y}' \in \Theta$. We distinguish the following cases.
  - Suppose $(\mathbf{n} \cdot j, \mathtt{x}) \in U_{\mathbb{d}}$ and $(\mathbf{n} \cdot j, \mathtt{y}) \in U_{\mathbb{d}'}$ for some $\mathbb{d}, \mathbb{d}' \in [\mathbb{d}_1, \mathbb{d}_\alpha]$. This also implies $\mathtt{x}' = \mathbb{d}, \mathtt{y}' = \mathbb{d}' \in \Theta$. Recall that Θ is satisfiable, hence $\mathbb{d} < \mathbb{d}'$ must hold. By definition, $g(\mathbf{n} \cdot j, \mathtt{x}) = \mathbb{d}$ and $g(\mathbf{n} \cdot j, \mathtt{y}) = \mathbb{d}'$, and hence clearly $g(\mathbf{n} \cdot j, \mathtt{x}) < g(\mathbf{n} \cdot j, \mathtt{y})$ and therefore $\mathbb{Z} \models (\mathtt{x}' < \mathtt{y}')(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
  - Suppose $(\mathbf{n} \cdot j, \mathtt{x}) \in U_{<\mathbb{d}_1}$ and $(\mathbf{n} \cdot j, \mathtt{y}) \in U_{\mathbb{d}}$ for some $\mathbb{d} \in [\mathbb{d}_1, \mathbb{d}_\alpha]$. This also implies $\mathtt{x}' < \mathbb{d}_1 \in \Theta$. We have proved above that $g(\mathbf{n} \cdot j, \mathtt{x}) < \mathbb{d}_1$. By definition, $g(\mathbf{n} \cdot j, \mathtt{y}) = \mathbb{d} \geqslant \mathbb{d}_1$. Hence $g(\mathbf{n} \cdot j, \mathtt{x}) < g(\mathbf{n} \cdot j, \mathtt{y})$ and therefore $\mathbb{Z} \models (\mathtt{x}' < \mathtt{y}')(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
  - Suppose $(\mathbf{n} \cdot j, \mathtt{x}) \in U_{<\mathbb{d}_1}$ and $(\mathbf{n} \cdot j, \mathtt{y}) \in U_{<\mathbb{d}_1}$. By definition, $g(\mathbf{n} \cdot j, \mathtt{x}) = \mathbb{d}_1 - \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x})$ and $g(\mathbf{n} \cdot j, \mathtt{y}) = \mathbb{d}_1 - \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{y})$. By assumption and definition of G_t^{C}, we have $(\mathbf{n} \cdot j, \mathtt{x}) \xrightarrow{\leq} (\mathbf{n} \cdot j, \mathtt{y})$. Recall that $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x}) = \mathrm{slen}((\mathbf{n} \cdot j, \mathtt{x}), (\mathbf{n} \cdot j, \mathbb{d}_1))$ and $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{y}) = \mathrm{slen}((\mathbf{n} \cdot j, \mathtt{y}), (\mathbf{n} \cdot j, \mathbb{d}_1))$. By construction of G_t^{C}, we have $(\mathbf{n} \cdot j, \mathbb{d}_1) \in U_{\mathbb{d}_1}$, and hence $(\mathbf{n} \cdot j, \mathtt{x}) \xrightarrow{\leq} (\mathbf{n} \cdot j, \mathbb{d}_1)$ and $(\mathbf{n} \cdot j, \mathtt{y}) \xrightarrow{\leq} (\mathbf{n} \cdot j, \mathbb{d}_1)$. This clearly yields $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x}) \geqslant \mathrm{slen}(\mathbf{n} \cdot, \mathtt{y}) + 1$. Hence $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x}) > \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{y})$, so that indeed $g(\mathbf{n} \cdot j, \mathtt{x}) < g(\mathbf{n} \cdot j, \mathtt{y})$ and $\mathbb{Z} \models (\mathtt{x}' < \mathtt{y}')(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
  - Suppose $(\mathbf{n} \cdot j, \mathtt{x}) \in U_{<\mathbb{d}_1}$ and $(\mathbf{n} \cdot j, \mathtt{y}) \in U_{>\mathbb{d}_\alpha}$. This implies $\mathtt{x}' < \mathbb{d}_1, \mathtt{y}' > \mathbb{d}_\alpha \in \Theta$. We have proved above that $g(\mathbf{n} \cdot j, \mathtt{x}) < \mathbb{d}_1$ and $g(\mathbf{n} \cdot j, \mathtt{y}) > \mathbb{d}_\alpha$. Hence $g(\mathbf{n} \cdot j, \mathtt{x}) < g(\mathbf{n} \cdot j, \mathtt{y})$ and $\mathbb{Z} \models (\mathtt{x}' < \mathtt{y}')(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
  - Suppose $(\mathbf{n} \cdot j, \mathtt{x}) \in U_{\mathbb{d}}$ for some $\mathbb{d} \in [\mathbb{d}_1, \mathbb{d}_\alpha]$ and $(\mathbf{n} \cdot j, \mathtt{y}) \in U_{>\mathbb{d}_\alpha}$. This also implies $\mathtt{y}' > \mathbb{d}_\alpha \in \Theta$. We have proved above that $g(\mathbf{n} \cdot j, \mathtt{y}) > \mathbb{d}_\alpha$. By definition, $g(\mathbf{n} \cdot j, \mathtt{x}) = \mathbb{d} \leqslant \mathbb{d}_\alpha$. Hence $g(\mathbf{n} \cdot j, \mathtt{x}) < g(\mathbf{n} \cdot j, \mathtt{y})$ and $\mathbb{Z} \models (\mathtt{x}' < \mathtt{y}')(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
  - Suppose $(\mathbf{n} \cdot j, \mathtt{x}), (\mathbf{n} \cdot j, \mathtt{y}) \in U_{>\mathbb{d}_\alpha}$. By definition, $g(\mathbf{n} \cdot j, \mathtt{x}) = \mathbb{d}_\alpha + \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x})$ and $g(\mathbf{n} \cdot j, \mathtt{y}) = \mathbb{d}_\alpha + \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{y})$. By assumption and definition of G_t^{C}, we have $(\mathbf{n} \cdot j, \mathtt{x}) \xrightarrow{\leq} (\mathbf{n} \cdot j, \mathtt{y})$. Recall that $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x}) = \mathrm{slen}((\mathbf{n} \cdot j, \mathbb{d}_\alpha), (\mathbf{n} \cdot j, \mathtt{x}))$ and $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{y}) = \mathrm{slen}((\mathbf{n} \cdot j, \mathbb{d}_\alpha), (\mathbf{n} \cdot j, \mathtt{y}))$. By construction of G_t^{C}, we have $(\mathbf{n} \cdot j, \mathbb{d}_\alpha) \in U_{\mathbb{d}_\alpha}$, and hence $(\mathbf{n} \cdot j, \mathbb{d}_\alpha) \xrightarrow{\leq} (\mathbf{n} \cdot j, \mathtt{x})$ and $(\mathbf{n} \cdot j, \mathbb{d}_\alpha) \xrightarrow{\leq} (\mathbf{n} \cdot j, \mathtt{y})$. This clearly yields $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{y}) \geqslant \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x}) + 1$ because $(\mathbf{n} \cdot j, \mathtt{x}) \xrightarrow{\leq} (\mathbf{n} \cdot j, \mathtt{y})$. Hence $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x}) < \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{y})$, so that indeed $g(\mathbf{n} \cdot j, \mathtt{x}) < g(\mathbf{n} \cdot j, \mathtt{y})$ and $\mathbb{Z} \models (\mathtt{x}' < \mathtt{y}')(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.

  The other cases cannot happen thanks to local consistency. For instance, $\mathtt{y}' < \mathbb{d}_1, \mathtt{x}' > \mathbb{d}_\alpha$ and $\mathtt{x}' < \mathtt{y}'$ in Θ cannot happen due to consistency for elements in $\mathrm{STypes}(\beta, \mathbb{d}_1, \mathbb{d}_\alpha)$.
- The cases $\mathtt{x}' < \mathtt{y} \in \Theta$, $\mathtt{x} < \mathtt{y}' \in \Theta$ and $\mathtt{x} < \mathtt{y} \in \Theta$ are similar to the previous case and are omitted herein.
- Suppose $\mathtt{x}' = \mathtt{y}' \in \Theta$ with $t(\mathbf{n} \cdot j) = (\mathtt{a}, \Theta)$. Since Θ is satisfiable, for some $\mathbb{d}^\dagger$ in '$< \mathbb{d}_1$', $\mathbb{d}_1, \ldots, \mathbb{d}_\alpha$, '$> \mathbb{d}_\alpha$', we have $(\mathbf{n} \cdot j, \mathtt{x}), (\mathbf{n} \cdot j, \mathtt{y}) \in U_{\mathbb{d}^\dagger}$. If $\mathbb{d}^\dagger$ is different from '$> \mathbb{d}_1$' and '$> \mathbb{d}_\alpha$', necessarily $g(\mathbf{n} \cdot j, \mathtt{x}) = g(\mathbf{n} \cdot j, \mathtt{y})$. Otherwise, since $(\mathbf{n} \cdot j, \mathtt{x}) \xrightarrow{=} (\mathbf{n} \cdot j, \mathtt{y})$ in G_t^{C}, we have $\mathrm{slen}(\mathbf{n} \cdot j, \mathtt{x}) = \mathrm{slen}(\mathbf{n} \cdot j, \mathtt{y})$. Consequently, $g(\mathbf{n} \cdot j, \mathtt{x}) = g(\mathbf{n} \cdot j, \mathtt{y})$ too and $\mathbb{Z} \models (\mathtt{x}' = \mathtt{y}')(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.

- Suppose $\mathtt{x} = \mathtt{y} \in \Theta$ with $t(\mathbf{n} \cdot j) = (\mathtt{a}, \Theta)$. Since Θ is satisfiable and t is locally consistent, for some $\mathbb{d}^{\dagger}$ in '$< \mathbb{d}_1$', $\mathbb{d}_1, \ldots, \mathbb{d}_{\alpha}$, '$> \mathbb{d}_{\alpha}$', we have $(\mathbf{n}, \mathtt{x}), (\mathbf{n}, \mathtt{y}) \in U_{\mathbb{d}^{\dagger}}$. If $\mathbb{d}^{\dagger}$ is different from '$> \mathbb{d}_1$' and '$> \mathbb{d}_{\alpha}$', necessarily $g(\mathbf{n}, \mathtt{x}) = g(\mathbf{n}, \mathtt{y})$. Otherwise, since $(\mathbf{n}, \mathtt{x}) \xrightarrow{=} (\mathbf{n}, \mathtt{y})$ in G_t^{C}, we have $\mathrm{slen}(\mathbf{n}, \mathtt{x}) = \mathrm{slen}(\mathbf{n}, \mathtt{y})$. Consequently, $g(\mathbf{n}, \mathtt{x}) = g(\mathbf{n}, \mathtt{y})$ too and $\mathbb{Z} \models (\mathtt{x} = \mathtt{y})(\mathbb{t}'(\mathbf{n}), \mathbb{t}'(\mathbf{n} \cdot j))$.
- The case $\mathtt{x}' = \mathtt{y} \in \Theta$ with $t(\mathbf{n} \cdot j) = (\mathtt{a}, \Theta)$ is similar and it is omitted below. □