Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 239 (2009) 193–211
www.elsevier.com/locate/entcs

An Automata-based Approach for CTL* With Constraints

Régis Gascon

INRIA Sophia-Antipolis, CNRS, I3S, UNSA.
2004 Route des Lucioles, BP93, F-06902 Sophia-Antipolis Cedex, France
Email: regis.gascon@sophia.inria.fr

Abstract

We introduce a general definition for a family of branching-time logics that extend CTL* by allowing constraints between variables at the atomic level. These constraints allow to compare values of variables at different states of the model. We define an automata-theoretic approach to solve verification problems for such extensions. Our method is based on a finite abstraction of the infinite state space and a symbolic representation of the models that generalizes several approaches used for extensions of the linear-time logic LTL with constraints. We extend and combine several constructions involving alternating tree automata. We apply this approach to prove decidability and optimal complexity results for particular instances of CTL* extensions whenever an abstraction of the models verifying a "nice" property can be computed. These theoretical results generalize several results on LTL with constraints where such nice abstractions are used.

Keywords: Temporal-logic, Model-checking, Automata approach.

1 Introduction

In model-checking, temporal logics are commonly used to specify properties on symbolic representations of computing systems. The atomic formulas of these logics are usually propositional variables whose truth value depends on the states of the symbolic system. This allows to specify properties on these states but not on the data that can be used in some models s.t. clocks, counters, strings... To overcome this problem, several extensions of the linear-time logic LTL with constraints on the data have been considered in the literature [19,2,14]. However, less results are known about their branching-time extensions.

We consider in this paper a general definition for a family of branching-time temporal logics that extend CTL* by adding constraints on the data. In these extensions, the atomic formulas are refined to relations between terms of the form $\mathsf{X}\ldots\mathsf{X}x$, representing a future value of the variable x. So, values of the variables at different states of an execution can be compared: for instance, the formula $A(x =$

1571-0661/
doi:10.1016/j.entcs.2009.05.040

$\mathsf{XX}y$) means that for every execution the current value of x is equal to the value of y two states further. Similar extensions of LTL with constraints have already been studied where often PSPACE-completeness results are shown (see e.g., [19,2,14]). In [7], the authors have extended the automata approach of [18] to establish this complexity bound for satisfiability and model-checking problems. We also have adapted this technique in previous works [9,10]. Thus, a natural question raised by these results would be: can we generalize this automata approach to branching-time?

We answer to this question in this paper by defining a general automata approach for satisfiability of CTL* extended with constraints and model-checking of this logic over a class of automata whose transitions are labeled by atomic constraints of the logic. This construction relies on a symbolic abstraction of the concrete models. Unlike [15], we do not introduce new classes of automata but our construction requires non-trivial adaptations and combinations of classical constructions. Indeed, we must deal with the capability of the atomic constraints to compare values at different states of the model. Another difficulty is that the class of constraint automata we consider for model-checking can induce infinitely branching configuration graphs. We show how to tackle this problem by defining an abstraction of the automaton behaviour.

This construction can be used to refine several results about LTL extended with constraints (for instance [2,7,3]). Indeed, we can take advantage of the fact that the models of the logics concerned can be abstracted in such a way that the correspondence between concrete and symbolic models can be verified easier. We prove that satisfiability and model-checking problems are in 2EXPTIME for the CTL* extensions of such logics. Our method directly extends the different constructions for the corresponding LTL fragments cited above, as well as the symbolic representations of the models used in these constructions. As a consequence, the optimal complexity results for satisfiability of linear-time formulas are preserved. As expected, the construction of the automaton for the branching-time extensions is much more complicated than in the linear case and the complexity bound is quite high (it is already the case for CTL*). So, these results are mainly theoretical and we do not claim that the technique would be very efficient in practice. The theoretical significance of this work comes also from the lack of results about this kind of extensions of branching-time logic compared to their linear-time fragments. Some existing work concerns for instance the verification of constraints between counters [4,8] or on stack contents [13,12]. However, [4] restricts the use of the temporal quantifiers of the logic and [8] the structure of automata that can be verified. Finally, in [13] and [12] the constraints are checked by adding states in the model but this technique can only be used when the relations considered form regular languages.

2 A branching-time logic extended with constraints

Concrete domains. Let VAR $= \{x_0, x_1, \dots\}$ be a countably infinite set of variables. A *concrete domain* is a pair $\mathcal{D} = \langle D, \mathcal{R} \rangle$ where D is an interpretation domain

for the variables and $\mathcal{R}$ is a countable set of relations on the elements of D. We call $\mathcal{D}$-*constraint* an expression of the form $\mathsf{R}(x_1, \ldots, x_k)$ where $\mathsf{R} \in \mathcal{R}$ is a relation of the domain whose arity is k and $x_1, \ldots, x_k \in \mathrm{VAR}$. A D-*valuation* is a function $v : \mathrm{VAR} \to D$ assigning a value in D to every variable and the satisfaction relation is defined by $v \models \mathsf{R}(x_1, \ldots, x_n)$ iff $(v(x_1), \ldots, v(x_n)) \in R$ where R is the relation associated to the symbol R.

CTL* over concrete domains. Given a concrete domain $\mathcal{D} = \langle D, \mathcal{R} \rangle$, the extension $\mathrm{CTL}^*(\mathcal{D})$ of the branching time logic CTL^* over the concrete domain $\mathcal{D}$ is defined by

$$\phi ::= \top \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathrm{E}\psi \mid \mathrm{A}\psi$$
$$\psi ::= \phi \mid \neg\psi \mid \psi \wedge \psi \mid \mathsf{R}(\mathsf{X}^{i_1} x_{j_1}, \ldots, \mathsf{X}^{i_k} x_{j_k}) \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi$$

where $x_1, \ldots, x_k \in \mathrm{VAR}$ and $\mathsf{R} \in \mathcal{R}$. As usual, we distinguish in $\mathrm{CTL}^*(\mathcal{D})$ state formulas (ϕ) and path formulas (ψ) and we have $\mathrm{A}\psi \equiv \neg\mathrm{E}\neg\psi$. The atomic constraints $\mathsf{R}(\mathsf{X}^{i_1} x_{j_1}, \ldots, \mathsf{X}^{i_k} x_{j_k})$ do not involve variables but expressions called terms. A *term* is made of a variable x prefixed by $i \in \mathbb{N}$ symbols X (we shortly write $\mathsf{X}^i x$) and represents the value of the variable x at the i^{th} next state in the model. Because of terms, atomic constraints must be in the scope of a path quantifier like the other temporal subformulas that refer to the future. We define the next length $|\phi|_{\mathsf{X}}$ of a formula ϕ to be the maximal integer i s.t. a term of the form $\mathsf{X}^i x$ occurs in ϕ. This value corresponds to the depth we need to explore in the model to evaluate every atomic constraint in ϕ.

The models of $\mathrm{CTL}^*(\mathcal{D})$ are infinite trees of the form $T = \langle N, n_0, \to, \Gamma \rangle$ where N is a set of nodes, $n_0 \in N$ is the root, $\to \subseteq N \times N$ is an edge relation and $\Gamma : N \to (\mathrm{VAR} \to D)$ is a map that associates a valuation to each node. Moreover, the edge relation of T is s.t. (i) each node except the root as exactly one incoming edge (tree shape) and (ii) every node has at least one successor. A *path* in T is a sequence of nodes $\pi = n_0 \cdot n_1 \cdots$ s.t. $n_i \to n_{i+1}$ for all $1 \leq i < |\pi|$. We denote the i^{th} suffix of a path π by $\pi^i = n_i \cdot n_{i+1} \cdots$, and the i^{th} node of π by $\pi(i)$. By property (ii), every maximal path of a $\mathrm{CTL}^*(\mathcal{D})$ model, also called *branch*, is infinite.

We introduce additional definitions making these models easier to handle in the following. The branching degree of a graph G is the least upper bound of the number of outgoing edges of every node (possibly ω) and a d-graph is a graph s.t. all nodes have exactly d outgoing edges. We arbitrarily order the successors of every node n of G whose degree is d by labeling the edges with elements of $\{0, \ldots, d-1\}$. For every $a \in \{0, \ldots, d-1\}$, a node n' is the a-successor of n iff $n \xrightarrow{a} n'$ and every node has at most one a-successor. So, the labeling allows to describe a path by giving the sequence of labels on the edges visited. For example, the path from the root n_0 following $w = 1 \cdot 0 \cdot 0 \cdots$ is $n_0 \xrightarrow{1} n_1 \xrightarrow{0} n_2 \xrightarrow{0} \cdots$ (see the bold path in Fig. 1). Given a path π in a graph, we note w_π the sequence made of the successive edge labels encountered when following π, i.e. for every $0 \leq i \leq |\pi| - 1$ we have $\pi(i) \xrightarrow{w(i)} \pi(i+1)$.

Let ϕ be a $\mathrm{CTL}^*(\mathcal{D})$ formula. Given a model T, a node n and a branch π of T, the state satisfaction relation $\langle T, n \rangle \models \phi$ and the path satisfaction relation

$\langle T, \pi\rangle \models \phi$ are defined as following (boolean cases are standard and omitted):

$\langle T, n\rangle \models \top$ for every $n \in N$,

$\langle T, n\rangle \models \mathrm{E}\ \psi$ iff there is a branch $\pi = n_0 \cdot n_1 \cdots$ s.t. $n_0 = n$ and $\langle T, \pi\rangle \models \psi$,

$\langle T, \pi\rangle \models \mathsf{R}(\mathsf{X}^{i_1} x_{j_1}, \ldots, \mathsf{X}^{i_k} x_{j_k})$ iff $(\Gamma(\pi(i_1))(x_{j_1}), \ldots, \Gamma(\pi(i_k))(x_{j_k})) \in R$,

$\langle T, \pi\rangle \models \mathsf{X}\psi$ iff $\langle T, \pi^1\rangle \models \psi$,

$\langle T, \pi\rangle \models \psi \mathsf{U} \psi'$ iff $\exists i \in \mathbb{N}$ s.t. $\langle T, \pi^i\rangle \models \psi'$ and for every $0 \leq j < i$, $\langle T, \pi^j\rangle \models \psi$.

The formula ϕ is *satisfiable* iff there is a model $T = \langle N, n_0, \rightarrow, \Gamma\rangle$ s.t. $\langle T, n_0\rangle \models \phi$.

Example 2.1 Let IPC^{++} [6] be the language defined by the following grammar:

$$\alpha ::= x \equiv_k y + [c, c'] \mid x \equiv_k [c, c'] \mid x = y \mid x = c \mid x < c \mid x > c \mid \alpha \wedge \alpha \mid \neg\alpha \mid \exists x.\, \alpha$$

where $x, y \in \mathrm{VAR}$, $c, c' \in \mathbb{Z}$ and $k \in \mathbb{N}$. We also use the abbreviations $x \equiv_k y + c$ and $x \equiv_k c$ for $x \equiv_k y + [c, c]$ and $x \equiv_k [c, c]$. The models for this language are valuations of the form $v : \mathrm{VAR} \rightarrow \mathbb{Z}$ and the satisfaction relation $\models$ is naturally defined for equality and inequality constraints. Concerning periodicity constraints, we have $v \models x \equiv_k y + [c_1, c_2]$ iff $\exists d \in \mathbb{Z}$ s.t. $v(x) - v(y) = c + dk$ for some $c_1 \leq c \leq c_2$.

We consider the concrete domain induced by the language IPC^{++}, i.e. the concrete domain whose interpretation domain is $\mathbb{Z}$ and whose set of relations is induced by the language IPC^{++}. In the following, we identify this domain with the language IPC^{++}. So, the set of CTL*(IPC^{++}) atomic formulas are IPC^{++} constraints over terms. For instance, the formula $\mathrm{E}\big((\mathsf{X}x > x) \wedge (\mathsf{X}x \equiv_2 x + 1))\ \mathsf{U}\ (\mathrm{A}(\mathsf{X}x = 0))\big)$ is a CTL*(IPC^{++}) formula satisfied by the model represented in Fig. 1 (the values of x are given in the nodes).

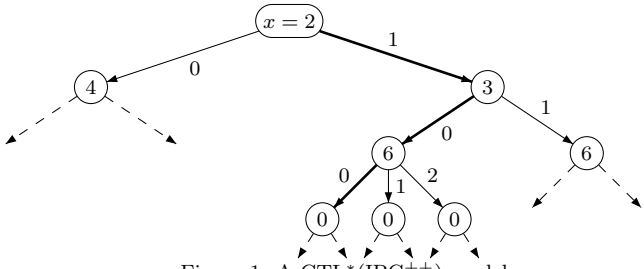

Figure 1. A CTL*(IPC^{++}) model

Constraint automata. We consider the model-checking problem for the logic CTL$^*(\mathcal{D})$ over a particular class of automata called $\mathcal{D}$-*constraint automata*. The transitions of such automata are labeled by constraints from $\mathcal{D}$ between terms representing the values of the variables at the current and the next state. Formally, a $\mathcal{D}$-constraint automaton is a structure $\mathcal{A} = \langle Q, I, F, \delta\rangle$ where Q is a finite set of locations, I and F are respectively the sets of initial and final locations, and the transition relation δ is a subset of $Q \times \Sigma \times Q$ where Σ is the set of $\mathcal{D}$-constraints built on terms of the form x and $\mathsf{X}x$ for $x \in \mathrm{VAR}$.

A state of the automaton is a pair $\langle q, v\rangle$ s.t. $q \in Q$ and v is a valuation. We note $\langle q, v\rangle \xrightarrow{\alpha} \langle q', v'\rangle$ iff there is a transition $\langle q, \alpha, q'\rangle \in \delta$ and the valuation assigning the value $v(x)$ to x and $v'(x)$ to $\mathsf{X}x$ for every $x \in \mathrm{VAR}$ satisfies α. A path in the

automaton $\mathcal{A}$ is a sequence of the form $\langle q_0, v_0\rangle \xrightarrow{\alpha_0} \langle q_1, v_1\rangle \xrightarrow{\alpha_1} \cdots$ and a run is a labeled tree $\langle N, n_0, \rightarrow, \Gamma\rangle$ s.t. (i) $\Gamma(n_0) = \langle q_0, v_0\rangle$ where $q_0 \in I$ and v_0 is an initial valuation and (ii) if $\Gamma(n) = \langle q, v\rangle$ then for every $\langle q', v'\rangle$ s.t. $\langle q, v\rangle \xrightarrow{\alpha} \langle q', v'\rangle$ holds in $\mathcal{A}$ there is a successor of n labeled by $\langle q', v'\rangle$. Condition (ii) allows to have infinitely many successors when the interpretation domain of $\mathcal{D}$ is infinite. A run is accepting iff every infinite branch has an infinite number of nodes labeled with a final state (Büchi acceptance condition). A $\mathcal{D}$-constraint automaton $\mathcal{A}$ satisfies a CTL$^*(\mathcal{D})$ formula ϕ (denoted $\mathcal{A} \models \phi$) iff there exists a run T of $\mathcal{A}$ s.t. $\langle T_v, n_0\rangle \models \phi$ where T_v is the restriction of T with the map restricted to valuations.

This definition of constraint automata subsumes several well-known models s.t. counter automata, pushdown automata or lossy channel systems, to quote few examples. Indeed, one can handle all the different objects of these formalisms by considering the right interpretation domain (integers or strings in these examples) and set of relations (standard arithmetic, prefix or subword relations). Note also that one can express tests/guards s.t. $x = 0$ as well as updates like $\mathsf{X}x \equiv_k x + 1$ on the same transition using conjunctions.

Example 2.2 Using IPC^{++}-constraint automata, we can abstract the behaviour of counter automata by performing operations modulo some integer (see [16]). We present in Fig. 2 the abstraction of a pay-phone controller with two counters x and y (see [5, Ex. 1]). The increments of a counter x are abstracted by $x < 2^{32} \wedge \mathsf{X}x \equiv_{2^{32}} x + 1$ which corresponds to the encoding of integers in standard programming languages. The formula $\phi_=$ stands for $\mathsf{X}x = x \wedge \mathsf{X}y = y$ which means that no counter is modified. Messages are omitted because they are irrelevant here.

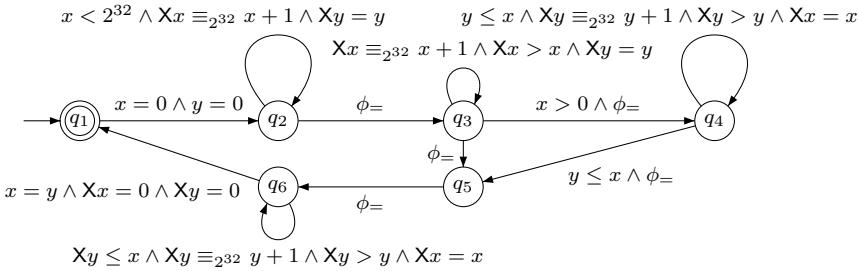

Figure 2. An IPC^{++}-constraint automaton

The *model-checking problem* takes as input a CTL$^*(\mathcal{D})$ formula ϕ and a $\mathcal{D}$-constraint automaton $\mathcal{A}$ and consists in checking whether $\mathcal{A} \models \phi$. Unlike for its LTL fragment restriction (see e.g, [7]), this problem cannot be reduced to the satisfiability problem because there is no way in the logic to express that a state have infinitely many successors.

3 From concrete to symbolic models

We consider concrete domains of the form $\mathcal{D} = \langle D, \mathcal{R}\rangle$ s.t. D is infinite (otherwise CTL$^*(\mathcal{D})$ can easily be reduced to CTL*) and the set $\mathcal{R}$ is not trivial, i.e. it contains at least one relation that is neither the empty nor the universal set. Let ϕ be a

CTL*($\mathcal{D}$) formula s.t. $V \subseteq \mathrm{VAR}$ is the finite set of variables used in ϕ and $|\phi|_{\mathsf{X}} = l$. We denote the set of terms used in ϕ by $\mathrm{Terms} = \{\mathsf{X}^i x \mid x \in V \text{ and } i \in \{0, \ldots, l\}\}$.

Symbolic valuations. The automata-based approach we propose relies on an abstraction of the models w.r.t. ϕ. The first step is to abstract valuations of the form $\mathrm{Terms} \to D$. We call *symbolic valuation abstraction* w.r.t. ϕ a pair composed of a set $\mathrm{SV}(\phi)$ and a surjective function of the form $symb : (\mathrm{Terms} \to D) \to \mathrm{SV}(\phi)$ associating a unique element of $\mathrm{SV}(\phi)$ to every valuation $v : \mathrm{Terms} \to D$, which verifies the following condition:

(SV) for every pair of valuations v, v' s.t. $symb(v) = symb(v')$ and every atomic constraint α of ϕ, we have $v \models \alpha$ iff $v' \models \alpha$.

The above condition implies that each element of $\mathrm{SV}(\phi)$, called *symbolic valuation*, is an equivalence class of valuations. Thus, we can define a symbolic satisfaction relation as following: for every atomic constraint α of ϕ and symbolic valuation $sv \in \mathrm{SV}(\phi)$, $sv \models_{\mathrm{symb}} \alpha$ iff for every valuation v s.t. $symb(v) = sv$ we have $v \models \alpha$.

Example 3.1 There are different methods to define such symbolic valuation abstractions. A famous one is the region abstraction of timed automata [1]. Several works about extensions of LTL over concrete domains also relies on symbolic valuation abstractions [2,7,3]. In most of them the symbolic valuations are described by sets of constraints.

For instance, given a CTL*(IPC^{++}) formula ϕ we denote respectively by Terms and CONS the sets of terms and constants used in ϕ, and K the least common multiple of the set of integers k s.t. a relation $\equiv_k$ occurs in ϕ. We define $\mathrm{SV}(\phi)$ to be the set of triples of the form $\langle X_{\mathrm{eq}}, X_{\mathrm{cons}}, X_{\mathrm{mod}} \rangle$ s.t.

• X_{eq} is a maximal consistent set of equality relations between the elements of Terms. Maximal consistency means that X_{eq} is consistent, i.e. there is a valuation that satisfies all the constraints, and that no proper extension of this set is consistent.

• X_{cons} is a maximal consistent set of comparisons between the elements of Terms and CONS. For every $t \in \mathrm{Terms}$ and $c \in \mathrm{CONS}$ there is a constraint $t \sim c \in X_{\mathrm{cons}}$ where $\sim \in \{<, =, >\}$.

• X_{mod} is a maximal consistent set of periodicity relations of the form $t \equiv_K c$ s.t. $t \in \mathrm{Terms}$ and $c \in \{0, \ldots K\}$. So, for every $t \in \mathrm{Terms}$ there is exactly one constraint of the form $t \equiv_K c$ in X_{mod}.

Simple additional properties that we do not develop here (transitivity, reflexivity, consistence of the modulos and comparisons...) have to be checked in order to ensure the consistency of $X_{\mathrm{eq}} \uplus X_{\mathrm{cons}} \uplus X_{\mathrm{mod}}$. The map *symb* associates to every valuation $v : \mathrm{Terms} \to \mathbb{Z}$ the element of $\mathrm{SV}(\phi)$ s.t. v satisfies every constraint in the triple. This element is unique since the definition of $\mathrm{SV}(\phi)$ induces a partition of $\mathbb{Z}^{\mathrm{Terms}}$. Then, the property **(SV)** can be shown by structural induction on the constraint in the same way that [6, Lemma 1].

For the sake of simplicity, we will consider in the following that symbolic valuations are sets of relations between terms. The definitions in the remaining can be rewritten with more general notations but it would make them less clear. More-

over, all the results that we want to generalize in Sect. 4 use such a representation of symbolic valuations.

Frames. Symbolic valuations only give information on a linear path whereas the models of CTL*($\mathcal{D}$) are trees. So we have to extend this symbolic representation to a branching structure giving information on the different paths. In order to consider only a finite set of such branching structures, we first establish a bound property on the branching degree of the models that have to be considered.

A CTL*($\mathcal{D}$) formula is in *positive normal form* iff negations appear only in atomic constraints. The set of positive formulas can be defined by

$$\phi ::= \top \mid \mathrm{E}\psi \mid \mathrm{A}\psi \mid \phi \wedge \phi \mid \phi \vee \phi$$
$$\psi ::= \phi \mid \mathrm{R}(\mathsf{X}^{i_1}x_1, ..., \mathsf{X}^{i_k}x_k) \mid \neg\mathrm{R}(\mathsf{X}^{i_1}x_1, ..., \mathsf{X}^{i_k}x_k) \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathsf{X}\psi \mid \psi \mathsf{U} \psi \mid \psi \tilde{\mathsf{U}} \psi$$

where $\tilde{\mathsf{U}}$ is the dual operator of U, i.e. $\phi \tilde{\mathsf{U}} \phi' \equiv \neg(\neg\phi \mathsf{U} \neg\phi')$. One can easily build from any CTL*($\mathcal{D}$) formula an equivalent positive formula by using the duality of CTL*($\mathcal{D}$) logical operators and quantifiers.

Lemma 3.2 *For every* CTL*($\mathcal{D}$) *formula* ϕ *there is a formula* ϕ' *in positive normal form computable in linear time s.t.* ϕ *is satisfiable iff* ϕ' *is satisfiable.*

For every formula ϕ, we denote by $\mathrm{E}_\sharp(\phi)$ the number of existential path quantifiers in the positive normal form of ϕ. The following Lemma is the same kind of small model property shown in [11] for CTL*.

Lemma 3.3 *A* CTL*($\mathcal{D}$) *formula* ϕ *is satisfiable iff there exists a* $(\mathrm{E}_\sharp(\phi)+1)$*-tree that satisfies* ϕ*.*

Basically, given a model satisfying ϕ we can build another model enjoying this property by unfolding at each step one path satisfying each existential subformula of the positive normal form of ϕ which is satisfied in the initial model. By definition, there are at most $(\mathrm{E}_\sharp(\phi)+1)$ such subformulas. Then we complete to reach $(\mathrm{E}_\sharp(\phi)+1)$ outgoing paths for every node (see details in Appendix A).

Let us set $d = \mathrm{E}_\sharp(\phi)$. The result of Lemma 3.3 is very important since it allows us to restrict the search of a model satisfying ϕ to the subclass of $(d+1)$-trees. Given a symbolic valuation abstraction $\langle \mathrm{SV}(\phi), symb \rangle$, we define a *frame* fr as a mapping $\{0, \ldots, d\}^l \to \mathrm{SV}(\phi)$ associating to every word of length l (corresponding to paths) a symbolic valuation. As we said before, we suppose that the elements of $\mathrm{SV}(\phi)$ are sets of constraints (see Ex. 3.1). The mapping fr must coincide on the common prefixes, i.e. constraints that refer to the same terms in different symbolic valuations must be the identical:

(FR) for every $w_1, w_2 \in \{1, \ldots, d\}^l$ and w their longest common prefix, we have $\alpha \in fr(w_1)$ iff $\alpha \in fr(w_2)$ for every atomic constraint α of the form $\mathrm{R}(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k)$ s.t. $0 \leq i_1, \ldots, i_k \leq |w|$.

We denote by Frame(ϕ) the set of frames defined w.r.t. ϕ.

Symbolic models. We use frames to define a symbolic representation of CTL*($\mathcal{D}$) models. We say that a pair of frames $\langle fr, fr' \rangle$ is *a-step consistent* iff for every

$i_1, \ldots, i_k > 0$, we have $\mathsf{R}(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k) \in fr(a \cdot w) \Leftrightarrow \mathsf{R}(\mathsf{X}^{i_1-1}x_1, \ldots, \mathsf{X}^{i_k-1}x_k) \in fr'(w \cdot b)$ for every $b \in \{1, \ldots, d\}$. This means that all the symbolic valuations associated with path moving in the direction of a are consistent with all the possible next ones (similarly to **(FR)**).

A *symbolic model* $T_{\text{symb}} = \langle N, \{n_0\}, \rightarrow, \Gamma_{\text{symb}} \rangle$ is a labeled d-tree associating to each node a frame $(\Gamma_{\text{symb}} : N \rightarrow \text{Frame}(\phi))$ s.t. every node n of T_{symb} is *one-step consistent*: for every $a \in \{1, \ldots, d\}$ the a-successor n' of n is s.t. $\langle \Gamma_{\text{symb}}(n), \Gamma_{\text{symb}}(n') \rangle$ is a-step consistent. The symbolic satisfaction relation defined for symbolic valuations can be naturally extended using the satisfaction relation of CTL$^*(\mathcal{D})$. The only difference is for the atomic constraints: for any branch π of T_{symb} we have $\langle T_{\text{symb}}, \pi \rangle \models_{\text{symb}} \alpha$ iff $\Gamma_{\text{symb}}(n)(w_\pi) \models_{\text{symb}} \alpha$ where n is the initial node of π.

We also define a satisfaction relation between concrete models and symbolic models. Given a graph $G = \langle N, \{n_0\}, \rightarrow, \Gamma \rangle$, a node $n \in N$ and a path $\pi \in N^l$, v_π is the valuation s.t. for every $x \in \text{VAR}$ and $i \in \{0, \ldots, l\}$, $v_\pi(\mathsf{X}^i x) = \Gamma(\pi(i))(x)$. We note $\langle G, n \rangle \models fr$ iff for every $w \in \{1, \ldots, d\}^l$ the path π starting at n and following w is defined in G and $symb(v_\pi) = fr(w_\pi)$. A graph G satisfies a symbolic model T_{symb} iff for every node n_{symb} reachable in T_{symb} from the initial state by following a word $w \in \{1, \ldots, d\}^*$ the node n reachable from the root by following the same path in G is s.t. $\langle G, n \rangle \models \Gamma_{\text{symb}}(n_{\text{symb}})$. This relation express that T_{symb} is an abstraction of G but it also implies a correspondence between the nodes of G and T_{symb}. Indeed, G must be a d-graph (otherwise some required path can be undefined) and when following the same path in both graph the nodes reached verify the satisfaction relation. The result below, which is the base of the automaton construction, can be proved by unfolding the definitions of this section.

Lemma 3.4 *A* CTL$^*(\mathcal{D})$ *formula ϕ is satisfiable iff there exist a symbolic model T_{symb} and a d-tree T s.t. $T_{\text{symb}} \models_{\text{symb}} \phi$ and T satisfies T_{symb}.*

4 A decidable model-checking problem

Nice abstractions. Several works have studied extensions of LTL with constraints from a concrete domain $\mathcal{D}$, which is the linear-time fragment of CTL$^*(\mathcal{D})$, where decidability proofs often rely on the kind of abstraction described in Sect. 3. We have observed from these works that, in many cases, checking the existence of a concrete model satisfying a given symbolic model can be done by performing tests only on a finite number of consecutive symbolic valuations. Often, checking one-step consistence is enough. This is the case for instance for extensions with concrete domains of the form $\langle D, <, = \rangle$ which are dense and open (see [7]) or domains verifying the global consistency of [2] or the ω-admissibility of [17]. In artificial intelligence, the domain RCC8 allowing to specify topological relations on the real plane $\mathbb{R}^2$ or the Allen relations on rational numbers used to represent temporal interval knowledge are some examples of such concrete domains among many others (see a more exhaustive list in [2, Sect.2]). These domains can also be used in temporal logic extensions to express interesting properties about space and time representations

of the knowledge. According to our motivations, we are interested in extending the results about extensions of LTL with the domains $\langle \mathbb{Q}, <, = \rangle$ and $\langle \mathbb{R}, <, = \rangle$ [7], IPC^{++} [6] (see Ex. 4.2) or the qualitative spatial reasoning of [2]. Finally, the results for the LTL extension of [3] with constraints built from the separation logic to verify memory allocation of programs also use an abstraction technique verifying this property. For all these LTL extensions, the satisfiability and model-checking problems have been shown to be PSPACE-complete. However, there are no corresponding results concerning their branching-time extension. Herein, we refine these results by extending the automata based approach introduced in [7].

First, we want to introduce a general property to subsume all the abstractions used in the above mentioned works. A symbolic valuation abstraction $\langle \mathrm{SV}(\phi), \mathit{symb} \rangle$ is said "*nice*" iff for every symbolic valuation $sv \in \mathrm{SV}(\phi)$ any partial assignment of the terms v' which satisfies all the constraints of sv involving only terms in the definition domains of v' can be extended to a valuation satisfying all the constraints of sv. In other words, any partial assignment that does not contradict any constraint can be extended to a valuation satisfying all the constraints. This property directly implies the following result and simplifies the forthcoming developments.

Lemma 4.1 *For every nice symbolic abstraction, every one-step consistent symbolic model is satisfiable.*

Nice symbolic abstractions also satisfy the property that any frame is satisfiable. This is not the case in general even if the symbolic valuations associated to each path are satisfiable.

Example 4.2 The abstraction we have defined for CTL*(IPC^{++}) models can easily be proved to be nice. Informally, suppose that we are given a symbolic valuation and we know a partial valuation of the terms satisfying the hypothesis. Let t be a term we want to assign a value. If there is a constraint $t = t'$ in X_{eq} s.t. t' has already a value assigned or $t = c$ in X_{cons} then we are done. Otherwise one can assign a value satisfying the constraint $t \equiv_K c$ in X_{mod} in the interval defined by the constraints of X_{cons} since symbolic valuations are supposed to be consistent. By repeating this procedure, we obtain a valuation satisfying the whole symbolic valuation.

Consider a formula ϕ of CTL$^*(\mathcal{D})$ with a nice symbolic abstraction $\langle \mathrm{SV}(\phi), \mathit{symb} \rangle$ and a $\mathcal{D}$-automaton $\mathcal{A} = \langle Q, I, F, \delta \rangle$. Wlog, we suppose that ϕ is in positive form (thanks to Lemma 3.2). We note $V \subseteq \mathrm{VAR}$ the variables used in ϕ and $\mathcal{A}$, $|\phi|_{\mathsf{X}} = l$ and $\mathrm{E}_\sharp(\phi) = d$.

Satisfiability. First, let us say few words about checking whether ϕ is satisfiable. By Lemma 3.4, we can solve this problem by defining an automaton $\mathcal{A}^\phi$ accepting d-trees which is the intersection of two automata: $\mathcal{A}^\phi_{\mathrm{symb}}$ checking the symbolic satisfaction of the formula and $\mathcal{A}^\phi_{\mathrm{sat}}$ checking whether the input corresponds to a satisfiable symbolic model. In a nutshell, the automaton $\mathcal{A}^\phi_{\mathrm{symb}}$ can be defined by adapting the automaton construction for CTL* using the same ideas as in [15]. The

main technical difficulty is to take into account the fact that atomic constraints refer to future values and paths from the current state to these values may need to be stored. This can be done easily by storing informations in the control states of the automata because the number of possible paths is finite (details in Appendix D). Since we have a nice abstraction, $\mathcal{A}^{\phi}_{\text{sat}}$ just have to check one-step consistency. This automaton can be built by taking as transition relation the elements s.t. the next frame is consistent with the current one w.r.t. the move in the tree. So the construction requires an exponential number of tests.

By construction, the language accepted by $\mathcal{A}_{\phi}$ is non-empty iff ϕ is satisfiable (consequence of Lemma 3.4). Since the emptiness problem for alternating tree automata is decidable (EXPTIME-complete), we have all the elements to establish decidability. Moreover, a simple complexity analysis allows to prove that the construction of $\mathcal{A}^{\phi}_{\text{symb}}$ can be done in exponential time whenever the symbolic satisfaction of an atomic constraint can be checked in exponential time. This is the case of the abstractions used for every linear-time logic cited at the beginning of this section (the symbolic satisfaction relation in $\text{CTL}^*(\text{IPC}^{++})$ can be tested in PTIME). So the complexity of the whole procedure for these logics is 2EXPTIME. The 2EXPTIME-hardness is easy to get since CTL^* is subsumed by $\text{CTL}^*(\mathcal{D})$ when $\mathcal{R}$ is not trivial (propositional variables can be simulated).

Theorem 4.3 $\text{CTL}^*(\mathcal{D})$ *satisfiability problem is* 2EXPTIME*-complete if for every formula there is a nice abstraction s.t. symbolic satisfaction relation can be tested in* EXPTIME*.*

Model-checking. We now consider the model-checking problem for $\text{CTL}^*(\mathcal{D})$. The main difficulty we have to handle is that $\mathcal{D}$-constraint automata can induce infinitely branching configuration graphs. So we start by defining an abstraction that bounds the branching degree and then we will extend the approach used for satisfiability. The difference with the method we sketched for satisfiability is that we have to recognize only the set of symbolic models that are satisfied by a run of $\mathcal{A}$.

We extend slightly the definition of the abstraction we consider. We note $\text{SV}(\phi, \mathcal{A})$ the set of symbolic valuations that now have to be built w.r.t. the atomic constraints of the formula ϕ and the constraints on the transitions of the automaton $\mathcal{A}$. This abstraction still have to verify the condition **(SV)** so that we can also define an associated symbolic satisfaction relation as in Sect. 3.

Lemma 4.4 *For every* $\text{CTL}^*(\mathcal{D})$ *formula* ϕ *and* $\mathcal{D}$*-automaton* $\mathcal{A}$*, one can build an automaton* $\mathcal{A}^{r}_{\text{abs}}$ *such that* $\mathcal{A} \models \phi$ *iff there is a symbolic model* T_{symb} *accepted by* $\mathcal{A}^{r}_{\text{abs}}$ *and verifying* $T_{\text{symb}} \models_{\text{symb}} \phi$*.*

We explain in few words how this automaton can be built. The whole construction is given in Appendix E. We introduce the projection of $\text{SV}(\phi, \mathcal{A})$ on the set of variables denoted by $\text{SV}_0(\phi, \mathcal{A})$. This set is obtained by keeping the relations referring to variables of the current state only which means that $X \in \text{SV}_0(\phi, \mathcal{A})$ iff there exists $sv \in \text{SV}(\phi, \mathcal{A})$ s.t. for every $x_1, \ldots, x_n \in \text{VAR}$ we have $\text{R}(x_1, \ldots, x_n) \in X$ iff

$\mathrm{R}(x_1, \ldots, x_n) \in sv$. The elements of $\mathrm{SV}_0(\phi, \mathcal{A})$ are then used to abstract the states of $\mathcal{A}$: the state $\langle q, v\rangle$ of $\mathcal{A}$ corresponds to $\langle q, X\rangle$ in an intermediate automaton $\mathcal{A}_{\mathrm{abs}}$ s.t. $X \in \mathrm{SV}_0(\phi, \mathcal{A})$ and v satisfies all the constraints in X. We build the automaton $\mathcal{A}_{\mathrm{abs}}$ in such a way that each path in $\mathcal{A}$ corresponds to a one-step consistent sequence in $\mathcal{A}_{\mathrm{abs}}$. Moreover, $\mathcal{A}_{\mathrm{abs}}$ generates finite branching runs only. From this automaton we can then define the automaton $\mathcal{A}^r_{\mathrm{abs}}$ recognizing the runs generated by $\mathcal{A}_{\mathrm{abs}}$ and establish the result of Lemma 4.4.

So we modify the construction for the satisfiability problem by intersecting the automaton $\mathcal{A}^\phi$ with the alternating tree automaton $\mathcal{A}^r_{\mathrm{abs}}$ recognizing the symbolic models corresponding to $\mathcal{A}_{\mathrm{abs}}$. We can easily deduce from the result of Lemma 3.4 and the construction of $\mathcal{A}^r_{\mathrm{abs}}$ that $\mathcal{A} \models \phi$ iff the language accepted by $\mathcal{A}^\phi \cap \mathcal{A}^r_{\mathrm{abs}}$ is non-empty. Thus we obtain a decidability procedure.

Theorem 4.5 $\mathrm{CTL}^*(\mathcal{D})$ *model-checking problem is decidable whenever there exists a nice symbolic valuation abstraction.*

The complexity of this extended construction does not increase if the symbolic satisfaction relation can be checked in EXPTIME.

Theorem 4.6 $\mathrm{CTL}^*(\mathcal{D})$ *model-checking problem is in* 2EXPTIME *if for every formula there is a nice abstraction s.t. symbolic satisfaction relation can be tested in* EXPTIME.

Concluding remarks. The gap with the complexity of CTL^* comes from the difference of expressiveness and conciseness between $\mathcal{D}$-automata and Kripke structures. Though being difficult to handle in practise, these theoretical results generalize several works on LTL extensions and improve the knowledge about temporal logics extended with concrete domains. However, it remains other concrete domains remains like $\langle \mathbb{Z}, <, =\rangle$ for which one need to check conditions on the whole symbolic model of CTL^* extention. We hope that this general approach approach could be extended to such cases.

References

[1] R. Alur and D. Dill. A theory of timed automata. *TCS*, 126:183–235, 1994.

[2] P. Balbiani and J. 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.

[3] R. Brochenin, S. Demri, and É. Lozes. Reasoning about sequences of memory states. In *Proceedings of LFCS'07*, volume 4514 of *LNCS*, pages 100–114. Springer, 2007.

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

[5] H. Comon and V. Cortier. Flatness is not a weakness. In *CSL*, volume 1862 of *Lecture Notes in Computer Science*, pages 262–276. Springer, 2000.

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

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

[8] S. Demri, A. Finkel, V. Goranko, and G. van Drimmelen. Towards a model-checker for counter systems. In *ATVA'06*, volume 4218 of *LNCS*, pages 493–507. Springer, 2006.

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

[10] S. Demri and R. Gascon. The effects of bounding syntactic resources on Presburger LTL (extended abstract). In *TIME'07*, pages 94–104. IEEE Computer Society Press, 2007.

[11] E. A. Emerson and A. P. Sistla. Deciding full branching time logic. *I&C*, 61(3):175–201, 1984.

[12] J. Esparza, A. Kučera, and S. Schwoon. Model checking LTL with regular valuations for pushdown systems. *I&C*, 186(2):355–376, 2003.

[13] A. Finkel, B. Willems, and P. Wolper. A direct symbolic approach to model checking pushdown systems (extended abstract). In F. Moller, editor, *INFINITY'97*, volume 9 of *ENTCS*. Elsevier Science Publishers, 1997.

[14] D. Gabelaia, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyaschev. On the computational complexity of spatio-temporal logics. In *FLAIRS'03*, pages 460–464, 2003.

[15] O. Kupferman, M. Y. Vardi, and P. Wolper. An automata-theoretic approach to branching-time model checking. *JACM*, 47(2):312–360, 2000.

[16] G. Logothetis and K. Schneider. Symbolic model checking of real-time systems. In *TIME*, pages 214–223, 2001.

[17] C. Lutz and M. Milicic. A tableau algorithm for description logics with concrete domains and gcis. In *TABLEAUX 2005*, LNAI, 2005.

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

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

A Proof of Lemma 3.2

Let ϕ be a CTL$^*(\mathcal{D})$ formula. We define a normalization of the formula that push the negations at the atomic level. The map f is defined by induction on the structure of ϕ.

- $f(\neg\neg\phi) = f(\phi)$,
- $f(\neg(\phi \wedge \phi')) = f(\neg\phi) \vee f(\neg\phi')$,
- $f(\neg(\phi \vee \phi')) = f(\neg\phi) \wedge f(\neg\phi')$,
- $f(\neg \mathrm{E}\phi) = \mathrm{A} f(\neg\phi)$,
- $f(\neg \mathrm{A}\phi) = \mathrm{E} f(\neg\phi)$,
- $f(\neg \mathsf{X}\phi) = \mathsf{X} f(\neg\phi)$,
- $f(\neg(\phi \mathsf{U} \phi')) = f(\neg\phi) \tilde{\mathsf{U}} f(\neg\phi')$,
- $f(\neg(\phi \tilde{\mathsf{U}} \phi')) = f(\neg\phi) \mathsf{U} f(\neg\phi')$,
- for the remaining cases, f coincides with the identity.

For every ϕ, the computation of $f(\phi)$ terminates since the recursive calls are on strict subformulas. By construction, the only subformulas that can be negated are the atomic formulas. So, $f(\phi)$ is in positive form. Finally, it is easy to prove that ϕ is satisfiable iff $f(\phi)$ is also satisfiable since the definition of f respects logical equivalences. □

B Proof of Lemma 3.3

Let ϕ be a CTL$^*(\mathcal{D})$ formula and $d = \mathrm{E}_\sharp(\phi) + 1$. Wlog, we can suppose that ϕ is already in positive form since it does not change its set of models. We denote by $\{\mathrm{E}\psi_1, \ldots, \mathrm{E}\psi_d\}$ the set of existential subformulas in ϕ.

Suppose that a graph $G = \langle N, \{n_0\}, \rightarrow, \Gamma\rangle$ satisfies ϕ. It is well known that this model can be unfolded into a tree-like model. We adapt this unfolding so that each existential formula satisfied at a node of G is satisfied along a designated path in our construction.

The tree $T = \langle N', \{n'_0\}, \rightarrow', \Gamma'\rangle$ where $N' \subseteq N \times \mathbb{N}$ and $n'_0 = \langle n_0, 0\rangle$ is defined as following. We define the map Γ' such that $\Gamma'(\langle n, i\rangle) = \Gamma(n)$ for every $\langle n, i\rangle \in N'$. We describe the construction of T by induction on the distance from the root of T.

Now, suppose that we have already built a path from n'_0 to $n' = \langle n, i\rangle$ in T and let $\{\mathrm{E}\psi_1^n, \dots, \mathrm{E}\psi_k^n\}$ be the set of existential formulas, satisfied by the node n in G. For every formula $\mathrm{E}\psi_j$ of this set, there exists a path $\pi = n_0^j \cdot n_1^j \cdot n_2^j \cdots$ in G such that $n_0^j = n$ satisfying $\mathrm{E}\psi_j$. We add a copy of this path T by adding the edges $\langle n, i\rangle \xrightarrow{j} \langle n_1^j, i+1\rangle$ and $\langle n_l^j, i+l\rangle \xrightarrow{0} \langle n_{l+1}^j, i+l+1\rangle$ for every $l > 0$. This method ensure that two paths that we copy never overlap. Obviously, the number of paths added by this operation is bounded by d. We can then add additional copies of paths so that $\langle n, i\rangle$ has exactly d successors (this does not change the set of existential formulas satisfied).

It is easy to prove that $\langle T, n'_0\rangle \models \phi$ by induction on the structure of the formula . □

C Proof of Lemma 3.4

Let ϕ be a satisfiable CTL$^*(\mathcal{D})$ formula. By Lemma 3.3 there is a d-tree $T = \langle N, \{n_0\}, \rightarrow, \Gamma\rangle$ such that $\langle T, n_0\rangle \models \phi$ (where n_0 is the root of T). We can define inductively a symbolic model $T_{\text{symb}} = \langle N', \{n'_0\}, \rightarrow, \Gamma_{\text{symb}}\rangle$ satisfied by T. We use a bijection between the nodes of T and T': we denote by $\{n_0, n_1, \dots\}$ the set of nodes of T and we prime the corresponding nodes of T_{symb}. For every $n \in N$ and sequence w, we define $v_{n,w}$ the valuation s.t. for every $x \in \mathrm{VAR}$ we have $v_{n,w}(\mathsf{X}^i x) = \Gamma(\pi(i))(x)$ where π is defined by $\pi(0) = n$ and $\pi(i) \xrightarrow{w(i)} \pi(i+1)$ for every $i \in \mathbb{N}$.

- The root of T_{symb} is the node n'_0 and Γ_{symb} is such that $\Gamma_{\text{symb}}(n'_0)(w) = \mathit{symb}(v_{n_0,w})$ for every $w \in \{1, \dots, d\}^l$.
- For each node n' of T_{symb} we define the next level as following. For every $a \in \{1, \dots, d\}$, we add an a-successor n'_a of n' in T_{symb}. The map Γ_{symb} is extended on the new node by defining $\Gamma(n'_a)(w) = \mathit{symb}(v_{n_a,w})$ for every $w \in \{1, \dots, d\}^l$, where n_a is the a-successor of n in T.

The one-step consistency of T_{symb} and the correctness of the frames w.r.t. property **(FR)** can be deduced from the fact that the successive symbolic valuations are built from paths that overlaps and shares common valuations. In this construction we clearly have a one-one correspondence between the nodes of T and the nodes of T_{symb}. We also have $T \models T_{\text{symb}}$ by definition of the successive frames (for every node n of T, we have $\langle T, n\rangle \models \Gamma_{\text{symb}}(n')$).

Let α be an atomic constraint occurring in ϕ. Using property (SV) of symbolic valuations, for every node n_i of T, $w \in \{1,\dots,d\}^l$ and for any valuation v' satisfying $symb(v_{n_i,w}) = symb(v')$ we have $v_{n_i,w} \models \alpha$ iff $v' \models \alpha$. This implies that for every n_i in T and $w \in \{1,\dots,d\}^l$, if $v_{n_i,w} \models \alpha$ then $\Gamma_{\text{symb}}(n_i')(w) \models_{\text{symb}} \alpha$. Thus, for every path $\pi = n_{i_0} \cdot n_{i_1} \cdots$ in T, $\langle T, \pi\rangle \models \alpha$ implies $\langle T_{\text{symb}}, \pi'\rangle \models_{\text{symb}} \alpha$ where $\pi' = n'_{i_0} \cdot n'_{i_1} \cdots$.

Then it is easy to prove that $T_{\text{symb}} \models_{\text{symb}} \phi$ since the symbolic satisfaction relation differs from CTL$^*(\mathcal{D})$ satisfaction relation only for atomic constraints and we have a one-one correspondence between the nodes of T and those T_{symb}.

Conversely suppose that $T_{\text{symb}} \models_{\text{symb}} \phi$ and $T \models T_{\text{symb}}$. We recall that by definition of $T \models T_{\text{symb}}$ there must exist a bijection between the nodes of T_{symb} and T.

We consider a constraint α of ϕ, a branch π of T_{symb} and we pose w to be the prefix of length l of w_π. If we have $\langle T_{\text{symb}}, \pi\rangle \models_{\text{symb}} \alpha$ for an atomic constraint α then by definition of the symbolic valuation $\Gamma_{\text{symb}}(\pi(0))(w)$ satisfies α. Since $T \models T_{\text{symb}}$ then the node n corresponding to $\pi(0)$ in T is such that $\langle T, n\rangle \models \Gamma_{\text{symb}}(\pi(0))$ which implies that $v_{n,w} \models \Gamma_{\text{symb}}(\pi(0))(w)$. By definition of the symbolic satisfaction relation we have $v_{n,w} \models \alpha$. Thus we have a corresponding branch π' in T such that $\langle T, \pi'\rangle \models \alpha$ where π' is the path starting at the node n and following the same directions than π.

Since the symbolic satisfaction relation only differs from CTL$^*(\mathcal{D})$ satisfaction relation at the atomic level we can then prove that $\langle T, n_0\rangle \models \phi$ using the correspondence between the nodes of T and T_{symb}. □

D Construction of $\mathcal{A}^{\phi}_{\text{symb}}$

The automaton $\mathcal{A}^{\phi}_{\text{symb}}$ is an alternating parity tree automaton. We use the following notations for alternating tree automata: an alternating d-tree automaton $\mathcal{A}$ is denoted by a tuple $\langle Q, Q_0, \Sigma, \delta, \text{Cond}\rangle$ where Q is a set of states, Σ is the input alphabet, $Q_0 \subseteq Q$ is a set of initial states, Cond specifies the acceptance condition and the transition relation δ is a subset of $\subseteq Q \times \Sigma \times \text{PBF}(\{1,\dots,d\} \times Q)$. The set PBF$(X)$ of positive boolean formula is defined by:

$$\theta ::= \top \mid \bot \mid p \mid \theta \wedge \theta \mid \theta \vee \theta$$

where p is an element of X. We say that a set $Y \subseteq X$ satisfies a formula $\theta \in \text{PBF}(X)$ iff the valuation assigning true to the elements of Y and false to the other elements of X saisfies θ. The transition relation associates to a pair $\langle q, \alpha\rangle \in Q \times \Sigma$ a positive boolean formula over $\{1,\dots,d\} \times Q$ which stands for the direction we move in the tree and the state we move in the automaton. We note $\delta(q,\alpha) = \theta$ whenever $\langle q, \alpha, \theta\rangle \in \delta$. For instance, $\delta(q,\alpha) = \langle 0, q_1\rangle \vee (\langle 1, q_2\rangle \wedge \langle 1, q_3\rangle)$ means that if we are in the state q and we read the letter α then either we move in the 0-succesor of the current node of the tree and the automaton moves in the state q_1 or we send

two a copy of the automaton moving in the 0-succesor of the current node, the first one entering in state q_2 and the other in state q_3. A run of $\mathcal{A}$ on a d-tree $T = \langle N, \{n_0\}, \delta, \Gamma\rangle$ is another tree $T_r = \langle N_r, \{n_{0,r}\}, \rightarrow_r, \Gamma_r\rangle$ such that the root is labeled by a state $q_0 \in Q_0$ and the other nodes by an element of $N \times Q$. A node of T_r labeled by $\langle q, n\rangle$ corresponds to a copy of the automaton in state q and reading the node n. Each node of T_r and its successors have to satisfy the transition property: for every node n_r of T_r such that $\Gamma_r(n_r) = \langle q, n\rangle$ and $\delta(q, \Gamma(n)) = \theta$, the (possibly empty) set $S \subseteq (\{1, \ldots, d\} \times Q)$ defined by $\langle a, q'\rangle \in S$ iff there is a successor labeled by $\langle n', q'\rangle \in S$ where n' is the a-successor of n in T satisfies θ.

We describe the construction of $\mathcal{A}^{\phi}_{\text{symb}}$ by induction on the structure of the formula ϕ. The basis case for $\phi \equiv \top$ is obvious.

We say that a CTL$^*(\mathcal{D})$ formula ϕ' is maximal in ϕ iff ϕ' is a strict subformula of ϕ and there is no strict subformula ϕ'' of ϕ such that ϕ' is a strict subformula of ϕ''. We denote by $\max(\phi) = \{\phi_1, \ldots, \phi_n\}$ the set of strict subformula of ϕ. With each maximal subformula ϕ_i of ϕ we suppose by induction hypothesis that the automaton $\mathcal{A}^{\phi_i}_{\text{symb}} = \langle Q^i, \{q^i_0\}, \text{Frame}(\phi), \delta^i, \text{Rank}^i\rangle$ and its complement, denoted by $\tilde{\mathcal{A}}^{\phi_i}_{\text{symb}} = \langle \tilde{Q}^i, \{\tilde{q}^i_0\}, \text{Frame}(\phi), \tilde{\delta}^i, \tilde{\text{Rank}}^i\rangle$, are defined. We assume wlog that the states of these automata are disjoint.

If $\phi \equiv \phi_1 \wedge \phi_2$ then $\mathcal{A}^{\phi}_{\text{symb}} = \langle Q^1 \cup Q^2 \cup \{q_0\}, \{q_0\}, \text{Frame}(\phi), \delta, \text{Rank}\rangle$ is defined as following. Intuitively we add a new initial state and send copies to $\mathcal{A}^{\phi_1}_{\text{symb}}$ and $\mathcal{A}^{\phi_2}_{\text{symb}}$. So the functions δ and Rank agrees with δ^1 and Rank^1 (resp. δ^2 and Rank^2) on the states of Q^1 (resp. Q^2). For the state q_0 we pose $\text{Rank}(q_0) = 1$ and $\delta(q_0, fr) = \delta(q^1_0, fr) \wedge \delta(q^1_0, fr)$. The case $\phi \equiv \phi_1 \vee \phi_2$ is similar with $\delta(q_0, fr) = \delta(q^1_0, fr) \vee \delta(q^1_0, fr)$.

If $\phi \equiv E\psi$ we adapt the linear construction for ψ by verifying a single branch of the tree. The adaptation is not straightforward because at every position we have to store the l^{th} next step to evaluate the atomic constraints. We describe below the whole construction. We define $cl(\phi)$ the closure of ϕ as usual with the only difference that maximal subformulas are considered as propositional variables. An atom of ϕ is a maximally consistent subset of $cl(\phi)$ and the set of atoms of ϕ is denoted by $\text{Atom}(\phi)$. Let $\mathcal{A}_1 = \langle Q, \{q_0\}, \text{Frame}(\phi) \times 2^{\max(\phi)}, \delta, F\rangle$ be the generalized Büchi alternating tree automaton s.t.

- $Q = \langle \text{Atom}(\phi), \{0, \ldots, d\}^l\rangle \cup \{q_0\}$,
- $\delta(q_0, \langle fr, X\rangle) = \bigvee_{\pi \in \sigma^l} \delta(\langle \text{At}, \pi\rangle, \langle fr, \text{X},)\rangle$ s.t. $\phi \in \text{At}$,
- $\delta(\langle \text{At}, a \cdot \pi\rangle, \langle fr, X\rangle) = \bigvee_{b \in \{0,\ldots,d\}} (a, \langle \text{At}', \pi \cdot b\rangle)$ iff
  - $fr(a \cdot \pi) \models \alpha$ for every $\alpha \in \text{At}$,
  - $\phi_i \in X$ for every $\phi_i \in \text{At}$,
  - $\mathsf{X}\phi \in \text{At}$ iff $\phi \in \text{At}'$.
- let $\{\phi'_1 \mathsf{U} \phi''_1, \ldots, \phi'_r \mathsf{U} \phi''_r\}$ be the set of until formulae in $cl(\phi)$. We pose $F = \{F_1, \ldots, F_r\}$ where for every $i \in \{1, \ldots, r\}$, $F_i = \{At \in Q : \phi'_i \mathsf{U} \phi''_i \notin \text{At} \text{ or } \phi''_i \in \text{At}\}$.

The idea is that at every position we know the l next moves to make in the tree and when we fire a transition in the automaton we follow the path that has been already guessed and guess the $(l+1)^{\text{th}}$th next step.

We can easily transform $\mathcal{A}_1$ into an equivalent automaton $\mathcal{A}_2 = \langle Q', \{q'_0\}, \text{Frame}(\phi) \times 2^{\max(\phi)}, \delta', \text{Rank}' \rangle$ with a parity acceptance condition instead of the generalized Büchi condition. To build $\mathcal{A}^{\phi}_{\text{symb}}$ we need to complete this construction in order to handle the maximal subformulas supposed to be true at each step. This can be done by applying the following rules to the transition relation of $\mathcal{A}_2$

$$\delta(q, fr) = \bigvee_{\langle fr, X\rangle \in \text{Frame}(\phi)\times 2^{\max(\phi)}} \big(\delta'(q, \langle fr, X\rangle) \wedge \bigwedge_{\phi_i \in X} \delta^i(q^i_0, fr) \wedge \bigwedge_{\phi_i \notin X} \tilde{\delta}^i(\tilde{q}^i_0, fr)\big)$$

For the case $\phi = A\psi$ we build the automaton for $E\neg\phi$ and then complement it.

Now we prove the correctness of this construction. We develop the only non trivial case which is $\phi \equiv E\psi$. If a symbolic model T_{symb} satisfies ϕ then there is a path π such that $\langle T_{\text{symb}}, \pi\rangle \models_{\text{symb}} \psi$. By construction a run of $\mathcal{A}^{\phi}_{\text{symb}}$ that proceeds along π must satisfy ϕ. Conversely, if a symbolic model T_{symb} is accepted by a run T_r of $\mathcal{A}^{\phi}_{\text{symb}}$ then there is a path π in T_r such that if we proceed along this path in $\mathcal{A}_1$, T_{symb} is accepted. Since $\mathcal{A}_1$ is just a slight adaptation of the word automaton recognizing the symbolic models of a linear time formula, we can easily conclude that $\langle T_{\text{symb}}, \pi\rangle \models_{\text{symb}} \psi$ and $T_{\text{symb}} \models_{\text{symb}} \phi$ by using the correspondence between the maximal subformulas. □

E Construction of $\mathcal{A}^r_{\text{abs}}$

Let $\text{SV}_0(\phi, \mathcal{A})$ be the projection of $\text{SV}(\phi, \mathcal{A})$ on the set of variables obtained by keeping the relations referring to variables of the current state only: $X \in \text{SV}_0(\phi, \mathcal{A})$ iff there exists $sv \in \text{SV}(\phi, \mathcal{A})$ s.t. for every $x_1, \ldots, x_n \in \text{VAR}$ we have $\text{R}(x_1, \ldots, x_n) \in X$ iff $\text{R}(x_1, \ldots, x_n) \in sv$. As a consequence of this definition, $\langle \text{SV}_0(\phi, \mathcal{A}), symb_0\rangle$ verifies **(SV)** where $symb_0 : (\text{VAR} \to D) \to \text{SV}(\phi)$ is defined by $symb_0(v) = X$ iff v satisfies all the constraints of X.

We use the elements of $\text{SV}_0(\phi, \mathcal{A})$ to abstract the current state of the automaton $\mathcal{A}$. We build from $\mathcal{A}$ a standard Büchi automaton $\mathcal{A}_{\text{abs}} = \langle Q', I', F', \delta'\rangle$ over the alphabet $\text{SV}(\phi, \mathcal{A})$ s.t. $Q' = Q \times \text{SV}_0(\phi, \mathcal{A})$ and the transition relation is defined by $\langle q, X\rangle \xrightarrow{sv} \langle q', X'\rangle$ iff:

- X is the projection of sv w.r.t V, i.e. $\text{R}(x_1, \ldots, x_n) \in X$ iff $\text{R}(x_1, \ldots, x_n) \in sv$.
- $X'[x \leftarrow \mathsf{X}x \mid x \in \text{VAR}]$ is the projection of sv' w.r.t. the terms of the form $\mathsf{X}x$ where $X'[x \leftarrow \mathsf{X}x \mid x \in \text{VAR}]$ is obtained from X' by substituting every occurrence of x by $\mathsf{X}x$ for every $x \in V$. This means that $\text{R}(x_1, \ldots, x_n) \in X$ iff $\text{R}(\mathsf{X}x_1, \ldots, \mathsf{X}x_n) \in sv$.
- there exists a transition $q \xrightarrow{\alpha} q'$ in $\mathcal{A}$ such that $sv \models_{\text{symb}} \alpha$.

The set of initial states is $I' = \{\langle q_0, symb_0(v_0)\rangle \mid q_0\}$ where v_0 is the initial valuation

and the set of final states $F' = F \times \mathrm{SV}_0(\phi, \mathcal{A})$. This construction can be done in exponential time if the satisfaction relation can be checked in exponential time. The construction implies a bisimulation between a state $\langle q, v\rangle$ of $\mathcal{A}$ and $\langle q, X\rangle$ of $\mathcal{A}_{\mathrm{abs}}$ such that $symb_0(v) = X$. We can establish a stronger result in order to have a correspondence between paths in the automata $\mathcal{A}$ and $\mathcal{A}_{\mathrm{abs}}$. According to the definition of Sect. 3, one-step consistency in the case of symbolic valuations, which is a particular case of frame, is given by: for every $i_1, \ldots, i_k > 0$, $\mathsf{R}(\mathsf{X}^{i_1}x_1, \ldots, \mathsf{X}^{i_k}x_k) \in sv_1 \Leftrightarrow \mathsf{R}(\mathsf{X}^{i_1-1}x_1, \ldots, \mathsf{X}^{i_k-1}x_k) \in sv_2$. This definition is naturally extended for one-step consistent sequences of symbolic valuations.

Lemma E.1 *There is a path $\langle q_0, v_0\rangle \xrightarrow{\alpha_0} \langle q_1, v_1\rangle \xrightarrow{\alpha_1} \ldots$ in $\mathcal{A}$ iff there exists a path $\langle q_0, X_0\rangle \xrightarrow{sv_0} \langle q_1, X_1\rangle \xrightarrow{sv_1} \ldots$ in $\mathcal{A}_{\mathrm{abs}}$ such that $\sigma = v_0 \cdot v_1 \cdots$ and $\rho = sv_0 \cdots sv_1 \cdots$ verify*

- *ρ is one-step consistent,*
- *$\sigma \models \rho$, i.e. for every $i \in \mathbb{N}$ we have $\sigma, i \models \rho(i)$. In particular for every $i \in \mathbb{N}$ we have $v_i \models X_i$.*

Proof Let $\langle q_0, v_0\rangle \xrightarrow{\alpha_0} \langle q_1, v_1\rangle \xrightarrow{\alpha_1} \ldots$ be a path in $\mathcal{A}$. By construction of the abstraction, there exists for each subpath $\langle q_i, v_i\rangle \xrightarrow{\alpha_i} \cdots \xrightarrow{\alpha_{i+l-1}} \langle q_{i+l}, v_{i+l}\rangle$ of length l a transition in $\mathcal{A}_{\mathrm{abs}}$ of the form $\langle q_i, X_i\rangle \xrightarrow{sv(v_i^l)} \langle q_{i+1}, X_{i+1}\rangle$ where v_i^l is the valuation defined by $v_i^l(\mathsf{X}^j x) = v_{i+j}(x)$ for all $j \in \{0, \ldots, l\}$. Indeed, by the conditions **(SV)** on symbolic valuations and the definition of the symbolic satisfaction relation we have

- if v_i satisfies all the constraints of X_i then it is also the case for v_i^l and so $sv(v_i^l)$ symbolically satisfies all the constraints of X_i,
- similarly, if v_{i+1} satisfies all the constraints of X_{i+1} then $sv(v_i^l)$ symbolically satisfies the constraints of $X'[x \leftarrow \mathsf{X}x \mid x \in \mathrm{VAR}]$,
- for every $j \in \{0, \ldots, l-1\}$, if $v_i \models \alpha_i$ then $symb(v_i^l) \models_{\mathrm{symb}} \mathsf{X}^i\alpha_i$.

We pose $sv_i = symb(v_i^l)$ for every $i \in \mathbb{N}$. Since the symbolic valuations sv_i and sv_{i+1} are defined w.r.t. paths that overlap, it is obvious that $\langle sv_i, sv_{i+1}\rangle$ is one-step consistent. Thus, the run $\langle q_0, X_0\rangle \xrightarrow{sv_0} \langle q_1, X_1\rangle \xrightarrow{sv_1} \langle q_2, X_2\rangle \xrightarrow{sv_2} \cdots$ is such that $\rho = sv_0 \cdot sv_1 \cdots$ is one-step consistent. Moreover, since for every $i \in \mathbb{N}$ we have $sv_i = sv(v_i^l)$, the sequence $\sigma = v_0 \cdot v_1 \cdots$ satisfies ρ.

Conversely, suppose that there is an infinite path $\langle q_0, X_0\rangle \xrightarrow{sv_0} \langle q_1, X_1\rangle \xrightarrow{sv_1} \ldots$ in $\mathcal{A}_{\mathrm{abs}}$ such that the sequence $\rho = sv_0 \cdot sv_1 \cdots$ is one-step consistent. One can build from this path a path in $\mathcal{A}$ satisfying the requirements. We proceed by induction on the position in the path.

By definition of the transition relation of $\mathcal{A}_{\mathrm{abs}}$, there is a path $q_0 \xrightarrow{\alpha_0} q_1 \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_{l-1}} q_l$ in $\mathcal{A}$ such that for every $i \in \{0, \ldots, l-1\}$ we have $sv_0 \models_{\mathrm{symb}} \mathsf{X}^i\alpha_i$. Since $symb$ is a surjective function, there exists a valuation v such that $symb(v) = sv_0$. By definition of the symbolic satisfaction relation the finite path $\langle q_0, v_0\rangle \xrightarrow{\alpha_0} \langle q_1, v_1\rangle \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_{l-1}} \langle q_l, v_l\rangle$ such that for every $j \in \{0, \ldots, l\}$ and $x \in \mathrm{VAR}$ we have $v_j(x) = v(\mathsf{X}^j x)$ is a

valid path in $\mathcal{A}$. Indeed, the pair $\langle v_j, v_{j+1}\rangle$ satisfies α_j for every $j \in \{0,\dots,l-1\}$.

Now suppose that there exists a path of the form $\langle q_0, v_0\rangle \xrightarrow{\alpha_0} \langle q_1, v_1\rangle \xrightarrow{\alpha_1} \cdots \xrightarrow{\alpha_{i-2+l}} \langle q_{i-1+l}, v_{i-1+l}\rangle$ in $\mathcal{A}$ that satisfies all the symbolic valuations from ρ up to the position $i-1$. Since the sequence ρ is one-step consistent, if $sv_{i-1} \models_{\text{symb}} \mathsf{X}^j\alpha_{i-1+j}$ for every $j \in \{0,\dots,l-1\}$ then $sv_i \models_{\text{symb}} \mathsf{X}^j\alpha_{i+j}$ for every $j \in \{0,\dots,l-2\}$. Moreover, the valuation v_i^{l-1} defined by $v_i^{l-1}(\mathsf{X}^j x) = v_{i+j}(x)$ for all $j \in \{0,\dots,l-1\}$ partially satisfy sv_i.

We can claim that there exists a transition of the form $q_{i+l-1} \xrightarrow{\alpha_{i+l-1}} q_{i+l}$ in $\mathcal{A}$ such that $sv_i \models_{\text{symb}} \mathsf{X}^l\alpha_{i+l-1}$. Otherwise, since the sequence ρ is one-step consistent there is no transition $q_{i+l-1} \xrightarrow{\alpha_{i+l-1}} q_{i+l}$ such that $sv_{i+l-1} \models_{\text{symb}} \alpha_{i+l-1}$ and so the transition $\langle q_{i+l-1}, X_{i+l-1}\rangle \xrightarrow{sv_{i+l-1}} \langle q_{i+l}, X_{i+l}\rangle$ does not exist in $\mathcal{A}_{\text{abs}}$. Since the domain satisfy the completion property, there is a valuation v_{i+l} extending v_i^{l-1} into v_i^l such that $v_i^l(\mathsf{X}^l x) = v_{i+l}(x)$ and $v_i^l \models sv_i$. By definition of the symbolic satisfaction relation, we have $v_i^l \models \alpha_{i+l-1}$ and so the path can be extended by using the transition $\langle q_{i+l-1}, v_{i+l-1}\rangle \xrightarrow{\alpha_{i+l-1}} \langle q_{i+l}, v_{i+l}\rangle$. □

We now pose d to be the maximum of $\mathrm{E}_\sharp(\phi)+1$ and the maximal number of outgoing transition of states of $\mathcal{A}$. We denote by $\mathrm{Frame}(\phi,\mathcal{A})$ the set of frames of the form $\{1,\dots,d\}^l \to \mathrm{SV}(\phi,\mathcal{A})$. A symbolic run of $\mathcal{A}_{\text{abs}}$ is a symbolic model $T_{\text{symb}} = \langle N, \{n_0\}, \rightarrow, \Gamma_{\text{symb}}\rangle$ s.t.

- $N \subseteq Q \times \mathrm{SV}_0(\phi,\mathcal{A}) \times \{1,\dots,d\}^*$ where the last component describes the path from the root and distinguishes the copies of a same state of $\mathcal{A}_{\text{abs}}$,
- $\Gamma_{\text{symb}} : N \to \mathrm{Frame}(\phi,\mathcal{A})$,
- n_0 is of the form $\langle q_0, X_0, \epsilon\rangle$ s.t. $\langle q_0, X_0\rangle \in I'$ and $\Gamma_{\text{symb}}(n_0) = fr_0$ is s.t. X_0 is the projection of $fr_0(w)$ for all $w \in \{1,\dots,d\}^l$,
- Let $n = \langle q, X, w_n\rangle \in N$ be a node s.t. $\Gamma_{\text{symb}}(n) = fr$ and $\{n_1,\dots,n_d\}$ its set of successors. We pose $n_j = \langle q_j, X_j, w_n \cdot j\rangle$ for every $j \in \{1,\dots,d\}$.

(A1) for all $j \in \{1,\dots,d\}$ there is a transition $\langle q, X\rangle \xrightarrow{sv_j} \langle q_j, X_j\rangle$ in $\mathcal{A}_{\text{abs}}$ and there exists $w \in \{1,\dots,d\}^{l-1}$ s.t. $fr(j \cdot w) = sv_j$,

(A2) for all $\langle q, X\rangle \xrightarrow{sv} \langle q', X'\rangle$ in $\mathcal{A}_{\text{abs}}$ there exist $j \in \{1,\dots,d\}$ s.t. $\langle q', X'\rangle = \langle q_j, X_j\rangle$ and $w \in \{1,\dots,d\}^{l-1}$ s.t. $fr(j \cdot w) = sv_j$.

(A3) for every $j \in \{1,\dots,d\}$, the pair $\langle fr, \Gamma_{\text{symb}}(n_j)\rangle$ is j-step consistent.

By construction we have the following property.

Lemma E.2 *For every path $\langle q_0, X_0, w_0\rangle \to \langle q_1, X_1, w_1\rangle \to \cdots$ in T_{symb}, there exists a run $\langle q_0, X_0\rangle \xrightarrow{sv_0} \langle q_1, X_1\rangle \xrightarrow{sv_1} \dots$ of $\mathcal{A}_{\text{abs}}$ s.t. the sequence $sv_0 \cdot sv_1 \cdots$ is one-step consistent.*

Proof Consider a path $\pi = \langle q_0, X_0, w_0\rangle \to \langle q_1, X_1, w_1\rangle \to \cdots$ in T_{symb}. By definition of $T_{\text{symb}} = \langle N, \{n_0\}, \rightarrow, \Gamma_{\text{symb}}\rangle$ there exists a transition $\langle q_i, X_i\rangle \xrightarrow{sv_i} \langle q_{i+1}, X_{i+1}\rangle$ in $\mathcal{A}^T_{\text{abs}}$ for every $i \in \mathbb{N}$, and so there is a path $q_i \xrightarrow{\alpha_i} q_{i+1} \xrightarrow{\alpha^i_{i+1}} q^i_{i+2} \xrightarrow{\alpha^i_{i+2}} \cdots \xrightarrow{\alpha^i_{i+l-1}} q^i_{i+l}$ in $\mathcal{A}$ such that

- X_i is the projection of sv_i w.r.t V,
- $X_{i+1}[x \leftarrow \mathsf{X}x \mid x \in V]$ is the projection of sv_i w.r.t. the terms of the form $\mathsf{X}x$,
- $sv_i \models_{\text{symb}} \alpha_i$,

and a sequence $w_i'' \in \{1,\dots,d\}^l$ s.t. $fr_i(w_i'') = sv_i$ where $fr_i = \Gamma_{\text{symb}}(\langle q_i, X_i, i\rangle)$.

We pose $w_i' = w_\pi(i)\cdots w_\pi(i+l)$ for all $i \in \mathbb{N}$. By definition of frames, the symbolic valuation $fr_i(w_i')$ coincide with sv_i on the terms of the form x and $\mathsf{X}x$. This implies that X_i is the projection of $fr_i(w_i')$, $X_{i+1}[x \leftarrow \mathsf{X}x \mid x \in \text{VAR}]$ is the projection of $fr_i(w_i')$ w.r.t. the terms of the form $\mathsf{X}x$, and $fr_i(w_i') \models_{\text{symb}} \alpha_i$. By iterating these arguments, we can prove that $fr_{i+j}(w_{i+j}') \models_{\text{symb}} \alpha_{i+j}$ for every $j \in \{1,\dots,l-1\}$ and since by definition of the symbolic models the sequence $fr_i(w_i')\cdot fr_{i+1}(w_{i+1}')\cdots fr_{i+l-1}(w_{i+l-1}')$ is one-step consistent we have $fr_i(w_i') \models \mathsf{X}^j\alpha_{i+j}$ for all $j \in \{1,\dots,l-1\}$. So we have all the elements to show that there exists a transition of the form $\langle q_i, X_i\rangle \xrightarrow{fr_i(w_i)} \langle q_{i+1}, X_{i+1}\rangle$ in $\mathcal{A}^T_{\text{abs}}$. By induction, one can build a path $\langle q_0, X_0\rangle \xrightarrow{fr_0(w_0')} \langle q_1, X_1\rangle \xrightarrow{fr_1(w_1')} \cdots$ such that $fr_0(w_0')\cdot fr_1(w_1')\cdots$ is one-step consistent. □

So we build an alternating tree automaton $\mathcal{A}^r_{\text{abs}}$ recognizing the symbolic models corresponding to $\mathcal{A}_{\text{abs}}$. The set of locations of $\mathcal{A}^r_{\text{abs}}$ is the same than the set of locations of $\mathcal{A}_{\text{abs}}$, as well as the sets of initial and final sates. The alphabet of $\mathcal{A}^r_{\text{abs}}$ is of course $\text{Frame}(\phi,\mathcal{A})$ and the transition relation is the translation of the conditions defined for the construction of symbolic runs

$$\delta(\langle q, X\rangle, fr) = \bigwedge_{\langle q,X\rangle \xrightarrow{sv} \langle q',X'\rangle} \bigvee_{i:\exists w, fr(iw)=sv} (i, \langle q', X'\rangle)$$

$$\wedge \bigwedge_{i\in\{0,\dots,d\}} \bigvee_{\langle q,X\rangle \xrightarrow{sv} \langle q',X'\rangle : \exists w, fr(iw)=sv} (i, \langle q', X'\rangle).$$

The first part express that every transition of $\mathcal{A}_{\text{abs}}$ corresponds to a successor and the second part that every successor correspond to a transition of $\mathcal{A}_{\text{abs}}$ (in the case where $\text{E}_\sharp(\phi)+1$ is greater than the maximal number of outgoing transition of states of $\mathcal{A}$). Since this automaton will be intersected with $\mathcal{A}^\phi$ recognizing the set of symbolic models satisfying ϕ, we do not need to check one-step consistency (A3).