An Auxiliary Logic on Trees: on the Tower-Hardness of Logics Featuring Reachability and Submodel Reasoning

Alessio Mansutti

Department of Computer Science, University of Oxford, UK

Abstract

We describe a set of simple features that are sufficient in order to make the satisfiability problem of logics interpreted on trees TOWER-hard. We exhibit these features through an *Auxiliary Logic on Trees* (ALT), a modal logic that essentially deals with reachability of a fixed node inside a forest and features modalities from sabotage modal logic to reason on submodels. After showing that ALT admits a TOWER-complete satisfiability problem, we prove that this logic is captured by four other logics that were independently found to be TOWER-complete: two-variables separation logic, quantified computation tree logic, modal logic of heaps and modal separation logic. As a by-product of establishing these connections, we discover strict fragments of these logics that are still non-elementary.

1. The Hardness of Reachability and Submodel Reasoning

In mathematical logic there is a well-known trade-off between expressive power and complexity, where weaker languages cannot capture interesting properties of complex systems, whereas finding solutions of a given problem is infeasible for richer languages. For instance, many verification tasks, such as reachability and homomorphisms queries, happen to be expressible in monadic second-order logic (MSO) [17]. This logic is however not usable in practice, as its satisfiability problem SAT(MSO) is undecidable in general and was famously proved by Rabin [39] to be decidable but non-elementary when the logic is interpreted on trees or on one unary function. A more recent analysis that uses the hierarchy of non-elementary ranking functions [41] classifies SAT(MSO) on these two structures as TOWER-complete, i.e. complete for the class of problems of time complexity bounded by a tower of exponentials, whose height is an elementary function of the input.

In order to bypass these problems, a general approach is to design restrictions of MSO that can solve complex reasoning tasks while being more appealing complexity-wise. An example of this is given by the framework of temporal logics, formalisms that describe the evolution of reactive systems [25]. Among the various temporal logics, from the classical linear temporal logic (LTL) [42]

* This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 852769, ARiAT).

Email address: alessio.mansutti@cs.ox.ac.uk (Alessio Mansutti)
URL: alessiomansutti.github.io (Alessio Mansutti)

and computation tree logic (CTL) [14], as well as their fragments [2, 35], to the more recently developed interval temporal logics [7, 8], the main common feature of this framework is perhaps the ability to check whether the system can evolve to a certain configuration, i.e. a *reachability* query. In this context, we recall the landmark result on the satisfiability of CTL, shown ExpTime-complete by Fisher and Ladner [24]. Another possibility to deal with the complexity of MSO is to restrict the second-order quantifications to specific *submodels*. This is the idea behind ambient logic [18], separation logic [40] and more generally bunched logics [38] and graphs logics [1]. These logics provide primitives for reasoning about resource composition, mainly by adding a *spatial conjunction* $\varphi * \psi$ which requires to split a model into two disjoint pieces, one satisfying φ and the other satisfying ψ. Similar ideas are developed in sabotage modal logics, where the formula $\blacklozenge\varphi$, headed by the *sabotage* modality $\blacklozenge$, states that φ must hold in a graph obtained by removing one edge from the current model [4, 23]. Within these logics, we highlight the quantifier-free fragment of separation logic restricted to the operator $*$, denoted here with SL($*$) and whose satisfiability problem is proved to be PSpace-complete in [13].

Once a framework provides a solid foundation for reasoning tasks, a natural step is to extend its expressiveness while keeping its complexity in check. Sometimes the additional capabilities do not change the complexity of the logic, as for example SL($*$) extended with reachability predicates, whose satisfiability problem is still PSpace-complete [22]. However, it often happens that the new features make the problem jump to higher complexity classes and, sometimes, reach MSO. We pinpoint two instances of this:

- SL($*$) enriched with first-order quantifiers, albeit less expressive than MSO interpreted on one unary function, has a Tower-complete satisfiability problem [10].
- CTL enriched with propositional quantifiers has an undecidable satisfiability problem on general models. On trees (i.e. QCTL^t), the problem is Tower-complete [29].

Consequently, it is natural to ask ourselves why the additional features made the problem harder. Answering this question requires to study the interplays between the various operators of the logic, searching for a sufficient set of conditions explaining its complexity.

1.1. Our motivation

Second-order features often lead to logics with Tower-hard satisfiability problems, as illustrated above for first-order SL($*$) and QCTL^t. A good amount of research has been done independently on these logics [5, 10, 19, 29], culminating with the Tower-hardness of SL($*$) with two quantified variables [19] and the Tower-hardness of QCTL^t with just one temporal operator between *exists-finally* EF and *exists-next* EX [5] (see Section 5 for the definitions). Connections between these two formalisms have not been explicitly developed so far, perhaps because of the quite different logics: QCTL is built on top of propositional calculus and it is interpreted on infinite trees, whereas SL($*$) does not feature propositional symbols and it is essentially interpreted on finite structures. Nevertheless, we argue that these and other logics are related not only as they are fragments of MSO, but also as they share a form of reachability and an ability of reasoning on submodels which is sufficient to obtain Tower-hard logics.

1.2. Our contribution

We explicit these common features that lead to Tower-hard logics by relying on an *Auxiliary Logic on Trees* (ALT), introduced in Section 2. ALT reasons about reachability of a fixed *target node* inside a finite forest and features modalities from sabotage logics to reason on submodels. Here, reachability should be understood as the ability to reach the target node in at least one step,

starting from a "current" node which can be updated thanks to the existential modality *somewhere* $\langle \mathrm{U} \rangle$ [27]. In Section 3, we familiarise with the logic and take a look at the expressive power of ALT. In Section 4 we show that SAT(ALT) is TOWER-hard. The proof goes by reduction from the satisfiability problem of interval temporal logic (PITL) interpreted under locality principle [37]. In Section 5, we then display how ALT is captured by QCTL and the two-variables fragment of the first-order separation logic SL($*$), as well as modal logic of heaps (MLH) and modal separation logics (MSL), two other logics introduced in [19] and [20], respectively. In this context, beside exposing that all these logics are TOWER-hard because of the way they reason about reachability and submodels, we discover interesting sublogics that are still TOWER-complete:

- $\mathsf{SL}([\exists]_1, *, \mathtt{x} \hookrightarrow \_, \hookrightarrow^+)$, i.e. the extension of SL($*$) studied in [32], featuring Boolean connectives, (only) one quantified variable name, the separating conjunction $*$, and the well-known predicates *alloc* $\mathtt{x} \hookrightarrow \_$ and *reach-plus* $\hookrightarrow^+$,
- $\mathsf{SL}(*, \mathrel{-\!\!\circledast}^{[1]}, \mathtt{ls})$, i.e. the quantifier-free extension of SL($*$) featuring the *bounded separating implication* $\mathrel{-\!\!\circledast}^{[1]}$ from [10] and the *list-segment* predicate $\mathtt{ls}$,
- QCTL restricted to $\mathsf{E}(\varphi \,\mathsf{U}\, \psi)$ modalities, where φ, ψ are Boolean combinations of atomic propositions, or to the EF modality, which can be nested at most once,
- the common fragment of MLH and MSL having Boolean connectives and the modalities $\Diamond$, $\langle \mathrm{U} \rangle$ and $*$. Notice that this logic does not have propositional symbols.

This paper is an extended and completed version of [33], together with the first part of [32].

2. An Auxiliary Logic on Trees: Syntax and Semantics

Notation. The symbol $\mathbb{N}$ denotes the set of natural numbers. Given a partial function $\mathfrak{f} : D \rightharpoonup C$, we write $\mathsf{dom}(\mathfrak{f})$ for its domain, i.e. $\{d \in D \mid \mathfrak{f}(d) \text{ is defined}\}$, and $\mathsf{ran}(\mathfrak{f})$ for its image, i.e. $\{c \in C \mid \text{there is } d \in D, \mathfrak{f}(d) = c\}$. When $C = D$, given $\delta \in \mathbb{N}$ we write $\mathfrak{f}^\delta$ for the δth composition of $\mathfrak{f}$, where $\mathfrak{f}^0$ is the identity function on D, and $\mathfrak{f}^{\delta+1}(d) \stackrel{\text{def}}{=} \mathfrak{f}(\mathfrak{f}^\delta(d))$. We often see $\mathfrak{f}$ as a binary relation, and write $\mathfrak{f}^*$ and $\mathfrak{f}^+$ for the relations obtained by applying the Kleene closure and Kleene plus on $\mathfrak{f}$, respectively. Similarly, $\mathfrak{f}^{-1}$ stands for the inverse relation of $\mathfrak{f}$.

Syntax. The formulae φ of the *Auxiliary Logic on Trees* (ALT) are built from the grammars below:

$$
\begin{array}{llll}
\pi := & \top & & \text{(true)} \\
\mid & \mathtt{Hit} & & \text{(hit predicate)} \\
\mid & \mathtt{Miss} & & \text{(miss predicate)}
\end{array}
\qquad
\begin{array}{lll}
\varphi := & \pi & \text{(atomic formulae)} \\
\mid & \varphi \wedge \varphi \;\mid\; \neg\varphi & \text{(Boolean connectives)} \\
\mid & \blacklozenge\, \varphi & \text{(sabotage modality)} \\
\mid & \blacklozenge^{*}\, \varphi & \text{(repeated sabotage modality)} \\
\mid & \langle \mathrm{U} \rangle\, \varphi & \text{(somewhere modality)}
\end{array}
$$

The symbol $\langle \mathrm{U} \rangle$ corresponds to the univeral modality from [27]. Readers who are familiar with sabotage modal logics will recognise in $\blacklozenge$ the sabotage modality [4], and in $\blacklozenge^*$ its Kleene closure (i.e. the operator $\blacklozenge$ applied an arbitrary number of times). These two operators modify the model during the evaluation of a formula, making ALT a *relation-changing* modal logic (following the terminology used in [3]). However, contrary to most modal logics, ALT does not feature classical propositional symbols. Instead, this logic only features two interpreted atomic propositions Hit and Miss. Roughly speaking, Hit stands for "the target node is reachable" whereas Miss stands

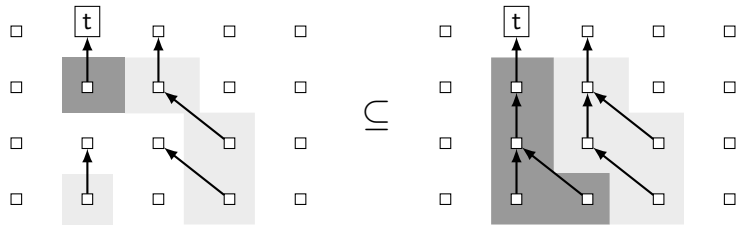


Figure 1: Subforest relation.

for "the target node is not reachable". The formal definitions will be given soon in order to clarify these two sentences.

Finite forests. Let $\mathcal{N}$ be a countably infinite set of *nodes*. We introduce the class of finite forests.

Definition 1 (Forest)**.** A *(finite) forest* $\mathcal{F} : \mathcal{N} \rightharpoonup_{\mathsf{fin}} \mathcal{N}$ is a partial function that has finite domain and is acyclic, i.e. $\mathcal{F}^{\delta}(\mathsf{n}) \neq \mathsf{n}$ for all $\mathsf{n} \in \mathsf{dom}(\mathcal{F})$ and $\delta \geq 1$. Pairs in $\mathcal{F}$ are called *edges*.

Albeit non-standard, our definition of finite forests over an infinite set of nodes simplifies the forthcoming definitions. Besides, in Section 3.2 (see Lemma 17(III)) we show how restricting $\mathcal{N}$ to a finite set does not change the expressive power nor the complexity of ALT. We recall the standard notions of ancestors and parent of a node.

Definition 2 (Ancestors and Parents)**.** Let n, n' be two nodes, and let $\mathcal{F}$ be a forest. n' is an $\mathcal{F}$-*ancestor* of n if there is a path in the forest going from n to n', i.e. $\mathcal{F}^{\delta}(\mathsf{n}) = \mathsf{n}'$ for some $\delta \geq 1$. If $\delta = 1$ then n' is the $\mathcal{F}$-*parent* of n.

Notice that, with this classification, $\mathcal{F}$ encodes the parent relation. We drop the prefix $\mathcal{F}$- from $\mathcal{F}$-ancestor and $\mathcal{F}$-parent when the forest is clear from the context. As usual, if n' is an ancestor of n, then we can alternatively say that n is a *descendant* of n'. Similarly, n is a *child* of n' whenever n' is the parent of n. Given two forests $\mathcal{F}, \mathcal{F}'$, we say that $\mathcal{F}'$ is a *subforest* of $\mathcal{F}$, written $\mathcal{F}' \subseteq \mathcal{F}$, whenever $\mathsf{dom}(\mathcal{F}') \subseteq \mathsf{dom}(\mathcal{F})$ and for every $\mathsf{n} \in \mathsf{dom}(\mathcal{F}')$, $\mathcal{F}'(\mathsf{n}) = \mathcal{F}(\mathsf{n})$. Figure 1 intuitively represents two forests, the one on the left being a subforest of the one on the right. Nodes of $\mathcal{N}$ are denoted by small boxes (□), and arrows represent the forest.

Semantics. ALT is interpreted on *pointed forests* $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, where $\mathcal{F}$ is a forest and $\mathsf{t}, \mathsf{n} \in \mathcal{N}$ are two nodes. The node t is called the *target node*. The node n is the *current (evaluation) node*. The satisfaction relation $\models$ for the formulae of ALT is given in Figure 2. The semantics of `Hit` and `Miss` is pretty straightforward. `Hit` holds if there is a non-empty path in the forest going from the current node to the target node. Instead, `Miss` holds if the current node is in the domain of the forest, but such a path does not exist. Given a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, n is called a *hit node* whenever $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \texttt{Hit}$. Instead, if $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \texttt{Miss}$ then n is a *miss node*. As a visual aid, the hit nodes of the forest in Figure 1 are the ones in the darker area, whereas the ones in the lighter (not white) area are miss nodes. It is worth noting that `Miss` is not exactly the negation of `Hit`, as it requires the current evaluation node to be in the domain of the forest. On the other hand, let us define the formula $\texttt{inDom} \stackrel{\text{def}}{=} \texttt{Hit} \vee \texttt{Miss}$, which is satisfied by $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ if and only if $\mathsf{n} \in \mathsf{dom}(\mathcal{F})$. Any two of the three formulae `Hit`, `Miss` and `inDom` suffice to define the third one. In particular:

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \top$ always,

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi \wedge \psi$ iff $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$ and $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \psi$,

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \neg\varphi$ iff not $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$,

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{Hit}$ iff n is a $\mathcal{F}$-descendant of t,

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{Miss}$ iff $\mathsf{n} \in \mathsf{dom}(\mathcal{F})$ and n is not a $\mathcal{F}$-descendant of t,

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge\, \varphi$ iff $\mathsf{card}(\mathsf{dom}(\mathcal{F}'))+1 = \mathsf{card}(\mathsf{dom}(\mathcal{F}))$ and $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \varphi$, for some $\mathcal{F}' \subseteq \mathcal{F}$

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge^*\, \varphi$ iff there is $\mathcal{F}'$ such that $\mathcal{F}' \subseteq \mathcal{F}$ and $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \varphi$,

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathrm{U} \rangle\, \varphi$ iff there is $\mathsf{n}' \in \mathcal{N}$ such that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \varphi$.

Figure 2: Satisfaction relation for ALT, with respect to a pointed forest state $(\mathcal{F}, \mathsf{t}, \mathsf{n})$.

$$\mathtt{Hit} \equiv \mathtt{inDom} \wedge \neg\mathtt{Miss}, \qquad\qquad \mathtt{Miss} \equiv \mathtt{inDom} \wedge \neg\mathtt{Hit},$$

where $\equiv$ stands for logical equivalence.

Let us continue the analysis on the features of ALT. As stated before, the semantics given to $\langle \mathrm{U} \rangle\, \varphi$ is the one of the existential modality *somewhere* [27], stating that there is a way to change the current evaluation node so that φ becomes true. This operator can be seen as a restricted form of first-order quantification, where the reassignment only occurs on the current evaluation node. Its dual operator $[\mathrm{U}]\varphi \stackrel{\text{def}}{=} \neg \langle \mathrm{U} \rangle \neg\varphi$ is the universal modality *everywhere*, stating that φ holds on every node in $\mathcal{N}$. The semantics given to $\blacklozenge\, \varphi$ is the one of the *sabotage* modality from [4], which requires to find one edge of the forest that, when removed, makes the model satisfy φ. Its dual operator $\blacksquare\varphi \stackrel{\text{def}}{=} \neg \blacklozenge \neg\varphi$ states that φ holds on every subforest obtained from the current forest by removing just one edge. Lastly, the modality $\blacklozenge^*$, here called *repeated sabotage*, can be seen as the operator obtained by applying $\blacklozenge$ an arbitrary number of times. Indeed, by inductively defining $\blacklozenge^k \varphi$ ($k \in \mathbb{N}$) as the formula φ for $k = 0$ and otherwise ($k \geq 1$) as $\blacklozenge \blacklozenge^{k-1} \varphi$, it is easy to see that

$$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge^*\, \varphi \text{ if and only if } (\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge^k\, \varphi \text{ for some } k \in \mathbb{N}.$$

Given a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, we write $\mathcal{F}[\mathtt{Miss}]_{\mathsf{t}}$ to denote the set of its miss nodes, i.e. $\mathcal{F}[\mathtt{Miss}]_{\mathsf{t}} \stackrel{\text{def}}{=} \{\mathsf{n}' \in \mathcal{N} \mid (\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \mathtt{Miss}\}$. We omit the subscript t from $\mathcal{F}[\mathtt{Miss}]_{\mathsf{t}}$ when it is clear from the context. We augment the standard precedence rules of propositional logic so that the modalities $\langle \mathrm{U} \rangle$, $\blacklozenge$ and $\blacklozenge^*$ have the same precedence as the negation $\neg$. For instance, the formula $\langle \mathrm{U} \rangle\, \mathtt{Hit} \wedge \mathtt{Miss}$ should be read as $(\langle \mathrm{U} \rangle\, \mathtt{Hit}) \wedge \mathtt{Miss}$.

Satisfiability. As usual, given a logic $\mathcal{L}$ interpreted on a class of structures S, the satisfiability problem for $\mathcal{L}$ takes in input a formula $\varphi \in \mathcal{L}$ and asks whether there is a structure $s \in S$ such that $s \models \varphi$. In this work, we are mainly interested in the satisfiability problem of ALT (SAT(ALT) in short) with respect to the interpretation on pointed forests.

The complexity class Tower. Let $\mathfrak{t}$ be the tetration function inductively defined as $\mathfrak{t}(0, n) = n$ and $\mathfrak{t}(k, n) = 2^{\mathfrak{t}(k-1,n)}$. Intuitively, $\mathfrak{t}(k, n)$ defines a tower of exponentials of height k. The complexity class Tower is the class of all problems decidable with a Turing machine running in time $\mathfrak{t}(g(n), f(n))$ for some polynomial f and elementary function g, on each input of length n [41].

Note that considering deterministic, non-deterministic or alternating Turing machines does not change the set of problems in TOWER. The main result of this paper is showing that SAT(ALT) is TOWER-complete.

3. On the Expressive Power of ALT

The two atomic propositions `Hit` and `Miss` make rather obscure what properties can be expressed in ALT. To become more familiar with the features of this logic, in this section we start playing with it. As we will soon find out, the ability to reason about submodels given by the combination of the two operators $\blacklozenge$ and $\blacklozenge^*$ greatly increases the expressive power of ALT. In particular, we show that ALT is able to characterise finite words. Encoding finite words in ALT is also the first step we need to show that this logic admits a TOWER-complete satisfiability problem. The proof of TOWER-hardness, addressed in Section 4, is by reduction from the satisfiability problem of Moszkowski's propositional interval temporal logic under locality condition (defined in Section 4.1). As we will see, this reduction is somewhat non-intuitive and can perhaps appear needlessly cumbersome. The reason for this is that we need to get around the fact that, given a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, ALT cannot deduce any property of the portion of the model corresponding to the set $\mathcal{F}[\texttt{Miss}]$, other than bounds on the size of this set and whether n belongs to it or not. This is shown formally at the end of this section, after providing a notion of Ehrenfeucht-Fraïssé games for ALT.

3.1. Towards TOWER-hardness: how to encode finite words in ALT

As a first step, we define a correspondence between finite words and specific pointed forests. As usual, the set of *finite words* on a *finite alphabet* Σ is defined as the closure of Σ under Kleene star, i.e. Σ^*. To ease our modelling, we suppose $\Sigma \stackrel{\text{def}}{=} [1, n]$ to be the alphabet of natural numbers between 1 and n. Let $\mathfrak{w} = \mathsf{a}_1 \dots \mathsf{a}_k$ be a k-symbols word in Σ^*. Let us explain how to encode $\mathfrak{w}$ as a finite forest. Every *symbol* a_j ($j \in [1, k]$), is encoded using a node n_j and $\mathsf{a}_j + 1$ additional nodes that are children of n_j. So, for example the symbol 3 is represented by a node having four children. All the nodes in $\{\mathsf{n}_j \mid j \in [1, k]\}$ are then connected in a path going from n_1 to the target node t, so that for every $j \in [1, k-1]$ n_j is a child of n_{j+1}, and n_k is a child of t. Notice that, with the exception of n_1, for every $j \in [2, k]$ this increases the number of children of n_j by one. Let us now formalise this encoding.

Definition 3 (Word encoding)**.** Let $\mathfrak{w} = \mathsf{a}_1 \dots \mathsf{a}_k$ be in Σ^*, where $\Sigma = [1, n]$ for some $n \geq 1$. We say that a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ *encodes* $\mathfrak{w}$ if and only if there is a tuple $\mathbb{M} = (\mathsf{n}_1, \dots, \mathsf{n}_k)$ of k nodes and tuple $\mathbb{C} = (\mathsf{N}_1, \dots, \mathsf{N}_k)$ of k sets of nodes such that

1. $\{\mathsf{n}_1, \dots, \mathsf{n}_k\}, \mathsf{N}_1, \dots, \mathsf{N}_k, \mathcal{F}[\texttt{Miss}]_{\mathsf{t}}$ are pairwise disjoint sets, i.e. they do not share any node,
2. $\mathbb{M}$ and $\mathbb{C}$ are all the descendants of t, i.e. $(\mathcal{F}^{-1})^+(\mathsf{t}) = \{\mathsf{n}_1, \dots, \mathsf{n}_k\} \cup \bigcup_{j \in [1,k]} \mathsf{N}_j$,
3. n_k is the only child of t and for every $j \in [1, k-1]$ $\mathcal{F}(\mathsf{n}_j) = \mathsf{n}_{j+1}$,
4. for every $j \in [1, k]$, $\mathsf{card}(\mathsf{N}_j) = \mathsf{a}_j + 1$ and for every $\mathsf{n}' \in \mathsf{N}_j$, $\mathcal{F}(\mathsf{n}') = \mathsf{n}_j$.

Notice that given a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ encoding $\mathfrak{w}$, the tuples $\mathbb{M}$ and $\mathbb{C}$ are uniquely defined. With respect to the elements in Definition 3, the k nodes in $\mathbb{M}$ are called *main nodes*, whereas the nodes in N_j ($j \in [1, k]$) are called *character nodes*. Main nodes and character nodes partition the set of $\mathcal{F}$-descendants of t. Thanks to the condition (3), main nodes form a path in the forest $\mathcal{F}$,

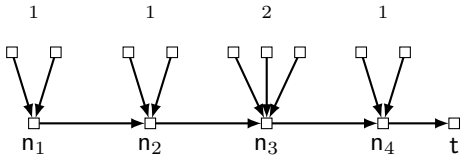


Figure 3: A forest encoding the word 1121.

going from n_1 to n_k. We call this path the *main path* of $\mathcal{F}$ (notice that we do not include the target node t). The following proposition stresses four important properties of our encoding that follow directly from its definition.

Proposition 4. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be an encoding of $\mathfrak{w} = \mathsf{a}_1 \dots \mathsf{a}_k$, with main nodes $\mathbb{M} = (\mathsf{n}_1, \dots, \mathsf{n}_k)$.

(I) $\mathsf{n}' \in \mathcal{N}$ is a main node if and only if it is a descendant of t and has at least one child.

(II) n_1 is the only main node having the same number of descendants and children.

(III) Given $j \in [2, k]$, n_j has exactly one child that is a main node.

(IV) Given $j \in [1, k]$, n_j has exactly $\mathsf{a}_j + 1$ children that are character nodes.

We say that a node $\mathsf{n} \in \mathsf{dom}(\mathcal{F})$ *encodes* the symbol $\mathsf{a} \in \Sigma$ if it has exactly $\mathsf{a} + 1$ children that are not in $\mathbb{M}$. Then, main nodes are the only ones encoding symbols, where n_j encodes a_j for every $j \in [1, k]$ (by property (IV)).

Example 5. Figure 3 shows a pointed forest encoding the word 1121. The main nodes of the encoding are $\mathbb{M} = (\mathsf{n}_1, \mathsf{n}_2, \mathsf{n}_3, \mathsf{n}_4)$, and its main path is given as $\{(\mathsf{n}_1, \mathsf{n}_2), (\mathsf{n}_2, \mathsf{n}_3), (\mathsf{n}_3, \mathsf{n}_4)\}$. Supposing that the tuple of character nodes is $\mathbb{C} = (\mathsf{N}_1, \mathsf{N}_2, \mathsf{N}_3, \mathsf{N}_4)$, the set N_j ($j \in [1, 4]$) contains the children of n_j that are not in $\mathbb{M}$, so that $\mathsf{card}(\mathsf{N}_1) = \mathsf{card}(\mathsf{N}_2) = \mathsf{card}(\mathsf{N}_4) = 2$, whereas $\mathsf{card}(\mathsf{N}_3) = 3$. Albeit the forest depicted here does not have miss nodes, in general encodings of words can have an arbitrary number of them.

Descendants and Children. We are now interested in characterising the class of pointed forests encoding finite words. In order to do so, we start by defining some easy formulae, which also serves as a way of familiarising with the logic. Looking at the properties in Proposition 4, we notice that they mainly rely on counting the number of descendants and children of a given node. Therefore, for now we focus on defining two formulae, $\#\mathtt{desc} \geq \beta$ and $\#\mathtt{child} \geq \beta$, that given a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ bound from below the number of descendants and children of the current evaluation node n, provided that n is a descendant of the target node t.

Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest. Given $\beta \in \mathbb{N}$, we start by defining the formula $\mathtt{size}(\mathtt{Miss}) \geq \beta$ stating that $\mathcal{F}$ contains at least β miss nodes, that is:

$$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{size}(\mathtt{Miss}) \geq \beta \;\text{ if and only if }\; \mathsf{card}(\mathcal{F}[\mathtt{Miss}]) \geq \beta.$$

This formula is inductively defined below:

$$\begin{aligned} \mathtt{size}(\mathtt{Miss}) \geq 0 &\stackrel{\text{def}}{=} \top, \\ \mathtt{size}(\mathtt{Miss}) \geq \beta{+}1 &\stackrel{\text{def}}{=} \langle \mathrm{U} \rangle \big(\mathtt{Miss} \wedge \underbrace{\blacklozenge(\neg \mathtt{inDom} \wedge \mathtt{size}(\mathtt{Miss}) \geq \beta)}_{\text{by excluding a miss node, at least other } \beta \text{ miss nodes can be found}} \big). \end{aligned}$$

Let us consider for a moment the definition of $\texttt{size}(\texttt{Miss}) \geq \beta{+}1$. Informally, this formula is satisfied if it is possible to find a node in $\mathcal{F}[\texttt{Miss}]$ (as expressed by the "$\langle \mathrm{U} \rangle(\texttt{Miss} \wedge \ldots$" part of the formula), removing it from the model (as done by the "$\blacklozenge(\neg\texttt{inDom}\ldots$" part), and then find other β elements of $\mathcal{F}[\texttt{Miss}]$. This formula essentially works because the set of miss nodes monotonically decreases when considering subforests, i.e. given $\mathcal{F}' \subseteq \mathcal{F}$ we have $\mathcal{F}'[\texttt{Miss}] \subseteq \mathcal{F}[\texttt{Miss}]$. Hence, finding a miss node in the subforest $\mathcal{F}'$ implies finding a miss node in the original forest $\mathcal{F}$. This idea is generalisable to similar monotonous properties. Let us extend our notation and, given a formula φ of ALT and a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, write $\mathcal{F}[\varphi]_{\mathsf{t}}$ for the set $\{\mathsf{n}' \in \mathcal{N} \mid (\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \varphi\}$. Moreover, given $\beta \in \mathbb{N}$ we inductively define the formula $\texttt{size}(\varphi) \geq \beta$ that bounds from below the amount of nodes satisfying φ, provided that φ satisfies some monotonic property formally defined below in Lemma 6. $\texttt{size}(\varphi) \geq \beta$ is defined by simply replacing $\texttt{Miss}$ by φ in $\texttt{size}(\texttt{Miss}) \geq \beta$:

$$\begin{aligned} \texttt{size}(\varphi) \geq 0 &\stackrel{\text{def}}{=} \top, \\ \texttt{size}(\varphi) \geq \beta{+}1 &\stackrel{\text{def}}{=} \langle \mathrm{U} \rangle \big(\varphi \wedge \underbrace{\blacklozenge(\neg\texttt{inDom} \wedge \texttt{size}(\varphi) \geq \beta)}_{\text{by excluding a node in } \mathcal{F}[\varphi], \text{ at least other } \beta \text{ such nodes can be found}}\big). \end{aligned}$$

The semantics of $\texttt{size}(\varphi) \geq \beta$ is characterised by the lemma below.

Lemma 6. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest. Let φ be a formula with the following properties:

1. $\mathcal{F}[\varphi]_{\mathsf{t}} \subseteq \mathsf{dom}(\mathcal{F})$, i.e. the set of nodes satisfying φ is a subset of the domain of the forest,
2. for every $\mathcal{F}' \subseteq \mathcal{F}$, if $\mathsf{dom}(\mathcal{F}) \setminus \mathsf{dom}(\mathcal{F}') \subseteq \mathcal{F}[\varphi]_{\mathsf{t}}$ then $\mathcal{F}[\varphi]_{\mathsf{t}} \cap \mathsf{dom}(\mathcal{F}') = \mathcal{F}'[\varphi]_{\mathsf{t}}$.

Given $\beta \in \mathbb{N}$, we have $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \texttt{size}(\varphi) \geq \beta$ if and only if $\mathsf{card}(\mathcal{F}[\varphi]_{\mathsf{t}}) \geq \beta$.

Before proving this lemma, let us look at the property (2) of φ. Consider a partition $\{S, T\}$ of the nodes in $\mathcal{F}[\varphi]_{\mathsf{t}}$, and the subforest $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F}) \setminus S$. Property (2) states that then $\mathcal{F}'[\varphi]_{\mathsf{t}} = T$. Informally, this means that removing nodes in $\mathcal{F}[\varphi]_{\mathsf{t}}$ does not change the set of nodes in the domain of the forest that still satisfy φ. This property, as well as property (1), holds for the case of $\varphi = \texttt{Miss}$, so that Lemma 6 implies the correctness of $\texttt{size}(\texttt{Miss}) \geq \beta$.

Proof. The proof is by induction on β, over the domain of natural numbers. The base case $\beta = 0$ is direct, so let us consider the inductive case for $\beta = \beta' + 1$ where $\beta' \in \mathbb{N}$.
($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \texttt{size}(\varphi) \geq \beta'{+}1$, and therefore there is a node $\mathsf{n}' \in \mathcal{N}$ such that

$$\text{A. } (\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \varphi, \qquad \text{B. } (\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \blacklozenge(\neg\texttt{inDom} \wedge \texttt{size}(\varphi) \geq \beta').$$

From (B), there is a forest $\mathcal{F}' \subseteq \mathcal{F}$ such that

$$\text{C. } \mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1, \qquad \text{D. } (\mathcal{F}', \mathsf{t}, \mathsf{n}') \models \neg\texttt{inDom}, \qquad \text{E. } (\mathcal{F}', \mathsf{t}, \mathsf{n}') \models \texttt{size}(\varphi) \geq \beta'.$$

First, let us prove that $\mathcal{F}[\varphi]_{\mathsf{t}} = \mathcal{F}'[\varphi]_{\mathsf{t}} \cup \{\mathsf{n}'\}$ and $\mathsf{n}' \notin \mathcal{F}'[\varphi]_{\mathsf{t}}$. From (A) and the property (1) on φ, we have $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$. From (C) and (D), this means that $\mathsf{dom}(\mathcal{F}) \setminus \mathsf{dom}(\mathcal{F}') = \{\mathsf{n}'\}$, which allows us to conclude that $\mathcal{F}[\varphi]_{\mathsf{t}} \cap \mathsf{dom}(\mathcal{F}') = \mathcal{F}'[\varphi]_{\mathsf{t}}$, directly from the property (2) of φ. This implies that $\mathcal{F}[\varphi]_{\mathsf{t}} = \mathcal{F}'[\varphi]_{\mathsf{t}} \cup \{\mathsf{n}'\}$ and $\mathsf{n}' \notin \mathcal{F}'[\varphi]_{\mathsf{t}}$. We now use this fact to show that the properties (1) and (2) of φ hold with respect to the forest $\mathcal{F}'$, so that we can then apply the induction hypothesis directly by (E). The property (1), i.e. $\mathcal{F}'[\varphi]_{\mathsf{t}} \subseteq \mathsf{dom}(\mathcal{F}')$, follows directly from $\mathcal{F}[\varphi]_{\mathsf{t}} \cap \mathsf{dom}(\mathcal{F}') = \mathcal{F}'[\varphi]_{\mathsf{t}}$. For the property (2), let $\mathcal{F}''$ be a forest such that $\mathcal{F}'' \subseteq \mathcal{F}'$ and $\mathsf{dom}(\mathcal{F}') \setminus \mathsf{dom}(\mathcal{F}'') \subseteq \mathcal{F}'[\varphi]_{\mathsf{t}}$. Let us prove that $\mathcal{F}'[\varphi]_{\mathsf{t}} \cap \mathsf{dom}(\mathcal{F}'') = \mathcal{F}''[\varphi]_{\mathsf{t}}$. From $\mathcal{F}' \subseteq \mathcal{F}$ it holds that $\mathcal{F}'' \subseteq \mathcal{F}$. Moreover,

$$\begin{array}{lcll} \mathsf{dom}(\mathcal{F}) \setminus \mathsf{dom}(\mathcal{F}'') & = & (\mathsf{dom}(\mathcal{F}') \cup \{\mathsf{n}'\}) \setminus \mathsf{dom}(\mathcal{F}'') & (\textit{by } \mathsf{dom}(\mathcal{F}) = \mathsf{dom}(\mathcal{F}') \cup \{\mathsf{n}'\}) \\ & = & (\mathsf{dom}(\mathcal{F}') \setminus \mathsf{dom}(\mathcal{F}'')) \cup \{\mathsf{n}'\} & (\textit{by } \mathsf{n}' \notin \mathsf{dom}(\mathcal{F}') \textit{ and } \mathcal{F}'' \subseteq \mathcal{F}') \\ & \subseteq & \mathcal{F}'[\varphi]_{\mathsf{t}} \cup \{\mathsf{n}'\} & (\textit{by } \mathsf{dom}(\mathcal{F}') \setminus \mathsf{dom}(\mathcal{F}'') \subseteq \mathcal{F}'[\varphi]_{\mathsf{t}}) \\ & = & \mathcal{F}[\varphi]_{\mathsf{t}} & (\textit{by } \mathcal{F}[\varphi]_{\mathsf{t}} = \mathcal{F}'[\varphi]_{\mathsf{t}} \cup \{\mathsf{n}'\}) \end{array}$$

Therefore, by property (2) (w.r.t. $\mathcal{F}$), $\mathcal{F}[\varphi]_{\mathsf{t}} \cap \mathsf{dom}(\mathcal{F}'') = \mathcal{F}''[\varphi]_{\mathsf{t}}$ holds, which is equivalent to $(\mathcal{F}'[\varphi]_{\mathsf{t}} \cup \{\mathsf{n}'\}) \cap \mathsf{dom}(\mathcal{F}'') = \mathcal{F}''[\varphi]_{\mathsf{t}}$. Lastly, from $\mathsf{n}' \notin \mathsf{dom}(\mathcal{F}')$ and $\mathcal{F}'' \subseteq \mathcal{F}'$, we conclude that $\mathcal{F}'[\varphi]_{\mathsf{t}} \cap \mathsf{dom}(\mathcal{F}'') = \mathcal{F}''[\varphi]_{\mathsf{t}}$, completing the proof of property (2) w.r.t. $\mathcal{F}'$. This allows us to use the induction hypothesis and conclude from (E) that $\mathsf{card}(\mathcal{F}'[\varphi]_{\mathsf{t}}) \geq \beta'$. This is sufficient to also conclude that $\mathsf{card}(\mathcal{F}[\varphi]_{\mathsf{t}}) \geq \beta'+1$, as we have already shown that $\mathcal{F}[\varphi]_{\mathsf{t}} = \mathcal{F}'[\varphi]_{\mathsf{t}} \cup \{\mathsf{n}'\}$ and $\mathsf{n}' \notin \mathcal{F}'[\varphi]_{\mathsf{t}}$. We leave the proof of the other direction to the reader. □

The formula $\mathtt{size}(\varphi) \geq \beta$ is not only useful as it can be quickly instantiated to define various interesting formulae in ALT, but also because it shows a suitable way of reasoning in ALT. Roughly speaking, we often use the somewhere modality $\langle \mathsf{U} \rangle$ to find a node in the domain of the forest that satisfies a certain property. Afterwards, we remove it with the sabotage operator $\blacklozenge$, in order to check if the resulting subforest satisfies a second property.

We can already make use of the formula $\mathtt{size}(\varphi) \geq \beta$ in order to define a formula that checks whether the number of children of the target node is at least β. It is sufficient to notice that such a child can be characterised with the formula $\mathtt{tchild} \stackrel{\text{def}}{=} \mathtt{Hit} \wedge \neg \blacklozenge \mathtt{Miss}$, and that this formula satisfies both the properties (1) and (2) of Lemma 6. This leads to the following result.

Lemma 7. $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{size}(\mathtt{tchild}) \geq \beta$ if and only if t has at least β children.

Proof. In order to prove this result it is sufficient to show that $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{tchild}$ holds if and only if n is a child of t, and that $\mathtt{tchild}$ satisfies the properties (1) and (2) of Lemma 6. For the correctness of $\mathtt{tchild}$, simply notice that if the current evaluation node n is a child of the target node t, then the same holds in every subforest $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{n} \in \mathsf{dom}(\mathcal{F}')$. Thus, $(\mathcal{F}', \mathsf{t}, \mathsf{n})$ cannot satisfy $\mathtt{Miss}$. Otherwise, if n is a descendant of t but not one of its children, removing $(\mathcal{F}(\mathsf{n}), \mathcal{F}(\mathcal{F}(\mathsf{n})))$ from the forest makes n a miss node, hence $\mathtt{tchild}$ is not satisfied.

The property (1), i.e. $\mathcal{F}[\mathtt{tchild}]_{\mathsf{t}} \subseteq \mathsf{dom}(\mathcal{F})$, holds from the tautology $\models \mathtt{Hit} \Rightarrow \mathtt{inDom}$. To prove the property (2), it is sufficient to see that the following (stronger) statement holds:

$$\text{for every } \mathcal{F}' \subseteq \mathcal{F},\ \mathcal{F}[\mathtt{tchild}]_{\mathsf{t}} \cap \mathsf{dom}(\mathcal{F}') = \mathcal{F}'[\mathtt{tchild}]_{\mathsf{t}}.$$

Showing this statement is straightforward, as a $\mathcal{F}$-child of t that is in the domain of $\mathcal{F}'$ is by definition a $\mathcal{F}'$-child of t, and vice versa. □

Let us now move to the definition of $\#\mathtt{desc} \geq \beta$, the formula stating that the current evaluation node is a hit node with at least β descendants. It is defined as follows:

$$\#\mathtt{desc} \geq \beta \ \stackrel{\text{def}}{=}\ \blacklozenge^* \big(\underbrace{[\mathsf{U}]\, \neg \mathtt{Miss}}_{\mathcal{F}[\mathtt{Miss}] \text{ is empty}} \wedge \mathtt{Hit} \wedge \underbrace{\blacklozenge(\neg \mathtt{inDom} \wedge \mathtt{size}(\mathtt{Miss}) \geq \beta)}_{\text{removing } \mathsf{n} \text{ lead to at least } \beta \text{ miss nodes}} \big).$$

The proof of correctness of this formula is given in Lemma 8. Intuitively, given a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ where n is a descendant of t, this formula uses the fact that removing $(\mathsf{n}, \mathcal{F}(\mathsf{n}))$ from the forest $\mathcal{F}$ makes all its descendants miss nodes. The repeated sabotage $\blacklozenge^*$ is used to remove the miss nodes before removing $(\mathsf{n}, \mathcal{F}(\mathsf{n}))$, so that then the formula $\mathtt{size}(\mathtt{Miss}) \geq \beta$ can be used to correctly count the descendants of n in $\mathcal{F}$.

Thanks to the formula $\#\mathtt{desc} \geq \beta$ we are able to define the formula $\#\mathtt{child} \geq \beta$ that checks the number of children of the current evaluation node n (assuming that n is a hit node):

$$\begin{array}{lcl} \#\mathtt{child} \geq 0 & \stackrel{\text{def}}{=} & \mathtt{Hit}, \\ \#\mathtt{child} \geq \beta{+}1 & \stackrel{\text{def}}{=} & \#\mathtt{desc} \geq \beta{+}1 \wedge \underbrace{\blacksquare^{\beta}(\mathtt{Hit} \Rightarrow \#\mathtt{desc} \geq 1)}_{\text{whenever } \beta \text{ nodes of } \mathsf{dom}(\mathcal{F}) \text{ are removed, if } \mathsf{n} \text{ still reaches } \mathsf{t} \text{ then it has at least one descendant}}. \end{array}$$

Informally, for a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, this formula expresses that n has $\beta \geq 1$ children by stating that the removal of $\beta{-}1$ edges from $\mathcal{F}$ cannot lead to a subforest where n has no descendants. The following lemma establishes the correctness of $\#\mathtt{desc} \geq \beta$ and $\#\mathtt{child} \geq \beta$.

Lemma 8. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest. Then,

(I) $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \#\mathtt{desc} \geq \beta$ iff n has at least β descendants and it is a descendant of t.

(II) $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \#\mathtt{child} \geq \beta$ iff n has at least β children and it is a descendant of t.

Proof of (I). ($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \#\mathtt{desc} \geq \beta$, and so there is a forest $\mathcal{F}' \subseteq \mathcal{F}$ such that

A. $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models [\mathrm{U}]\, \neg\mathtt{Miss}$. So, $\mathcal{F}'[\mathtt{Miss}]_{\mathsf{t}} = \varnothing$, and every $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}')$ is an $\mathcal{F}'$-descendant of t,

B. $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \mathtt{Hit}$. So, n is an $\mathcal{F}'$-descendant of t (thus, n is also an $\mathcal{F}$-descendant of t),

C. $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \blacklozenge(\neg\mathtt{inDom} \wedge \mathtt{size}(\mathtt{Miss}) \geq \beta)$.

From (C), there is a finite forest $\mathcal{F}''$ such that $\mathcal{F}'' \subseteq \mathcal{F}'$ and

D. $\mathsf{card}(\mathsf{dom}(\mathcal{F}'')) = \mathsf{card}(\mathsf{dom}(\mathcal{F}'))-1$,

E. $(\mathcal{F}'', \mathsf{t}, \mathsf{n}) \models \neg\mathtt{inDom}$,

F. $(\mathcal{F}'', \mathsf{t}, \mathsf{n}) \models \mathtt{size}(\mathtt{Miss}) \geq \beta$.

By (B), we obtain that $\mathsf{n} \in \mathsf{dom}(\mathcal{F}')$, whereas (E) implies $\mathsf{n} \notin \mathsf{dom}(\mathcal{F}'')$. Therefore, by (D) it holds that $\mathsf{dom}(\mathcal{F}'') = \mathsf{dom}(\mathcal{F}') \setminus \{\mathsf{n}\}$. We now consider the set $\mathcal{F}''[\mathtt{Miss}]_{\mathsf{t}}$, which by (F) has at least β elements. We prove that every $\mathsf{n}' \in \mathcal{F}''[\mathtt{Miss}]_{\mathsf{t}}$ is an $\mathcal{F}'$-descendant of n. *Ad absurdum*, suppose that there is $\mathsf{n}' \in \mathcal{F}''[\mathtt{Miss}]_{\mathsf{t}}$ that is not an $\mathcal{F}'$-descendant of n. By definition of $\mathcal{F}''[\mathtt{Miss}]_{\mathsf{t}}$,

G. n' is not an $\mathcal{F}''$-descendant of t,

H. $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}'')$ and therefore by $\mathcal{F}'' \subseteq \mathcal{F}'$ it holds that $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}')$.

From (A) and (H), n' is an $\mathcal{F}'$-descendant of t. Thus, from (G), there must be $\delta \geq 1$ such that the node $\mathcal{F}'^{\delta}(\mathsf{n}')$ is an $\mathcal{F}'$-descendant of t, but is not in the domain of $\mathcal{F}''$. However, as we already established that $\mathsf{dom}(\mathcal{F}'') = \mathsf{dom}(\mathcal{F}') \setminus \{\mathsf{n}\}$, this implies that $\mathcal{F}'^{\delta}(\mathsf{n}') = \mathsf{n}$ and therefore n' is an $\mathcal{F}'$-descendant of n: a contradiction. Therefore, every element in $\mathcal{F}''[\mathtt{Miss}]_{\mathsf{t}}$ is an $\mathcal{F}'$-descendant of n. Together with (B) and $\mathcal{F}' \subseteq \mathcal{F}$, we conclude that n has at least β $\mathcal{F}$-descendants and it is an $\mathcal{F}$-descendant of t.
($\Leftarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ to be a finite forest such that n has at least β $\mathcal{F}$-descendants and it is a $\mathcal{F}$-descendant of t. We then consider the two subforests $\mathcal{F}''$ and $\mathcal{F}'$ of $\mathcal{F}$ characterised as

$$\begin{aligned}\mathsf{dom}(\mathcal{F}'') =& \{\mathsf{n}' \in \mathcal{N} \mid \mathsf{n}' \text{ is an } \mathcal{F}\text{-descendant of } \mathsf{n}\} \cup \\ & \{\mathsf{n}' \in \mathcal{N} \mid \mathsf{n}' \text{ is an } \mathcal{F}\text{-ancestor of } \mathsf{n} \text{ and an } \mathcal{F}\text{-descendant of } \mathsf{t}\}. \\ \mathcal{F}' =& \mathcal{F}'' \cup \{(\mathsf{n}, \mathcal{F}(\mathsf{n}))\}\end{aligned}$$

Notice that $\mathsf{n} \notin \mathsf{dom}(\mathcal{F}'')$. From their characterisation, it is easy to see that

U. $\mathcal{F}'' \subseteq \mathcal{F}' \subseteq \mathcal{F}$,

V. $\{\mathsf{n}' \in \mathcal{N} \mid \mathsf{n}' \text{ is an } \mathcal{F}\text{-descendant of } \mathsf{n}\} \subseteq \mathcal{F}''[\mathtt{Miss}]_{\mathsf{t}}$ and $\mathsf{card}(\mathcal{F}''[\mathtt{Miss}]_{\mathsf{t}}) \geq \beta$.

This property holds because n is not an $\mathcal{F}''$-descendant of t, and all $\mathcal{F}$-descendants of n are also $\mathcal{F}''$-descendants of n. Thus, every $\mathcal{F}$-descendant of n is in $\mathcal{F}''[\mathtt{Miss}]_{\mathsf{t}}$. By hypothesis, n has at least β descendants,

W. $\mathsf{n} \in \mathsf{dom}(\mathcal{F}')$ and $\mathcal{F}'[\mathtt{Miss}]_{\mathsf{t}} = \varnothing$ (as n is an $\mathcal{F}'$-descendant of t).

This property holds because $\mathcal{F}'$ contains all the $\mathcal{F}$-descendants of n plus a path going from n to t, and nothing else. So, every element in $\mathsf{dom}(\mathcal{F}')$ is a hit node for $\mathcal{F}'$.

Then, we conclude that:

X. from (V), $(\mathcal{F}'', \mathsf{t}, \mathsf{n}) \models \neg\mathtt{inDom} \wedge \mathtt{size}(\mathtt{Miss}) \geq \beta$,

Y. from (W). $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models [\mathrm{U}]\, \neg\mathtt{Miss} \wedge \mathtt{Hit}$,

Z. from $\mathcal{F}' = \mathcal{F}'' \cup \{(\mathsf{n}, \mathcal{F}(\mathsf{n}))\}$ and (X), $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \blacklozenge(\neg\mathtt{inDom} \wedge \mathtt{size}(\mathtt{Miss}) \geq \beta)$.

Lastly, by (U), (Y) and (Z), $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge^*\big([\mathrm{U}]\, \neg\mathtt{Miss} \wedge \mathtt{Hit} \wedge \blacklozenge(\neg\mathtt{inDom} \wedge \mathtt{size}(\mathtt{Miss}) \geq \beta)\big)$. □

Proof of (II). The lemma is trivial for $\#\mathtt{child} \geq 0$, so we consider $\#\mathtt{child} \geq \beta{+}1$ where $\beta \in \mathbb{N}$.
($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \#\mathtt{child} \geq \beta{+}1$. From the first conjunct of $\#\mathtt{child} \geq \beta{+}1$:

A. n has at least $\beta{+}1$ $\mathcal{F}$-descendants;

B. n is an $\mathcal{F}$-descendant of t.

Ad absurdum, suppose that n has $k < \beta + 1$ $\mathcal{F}$-children $\{\mathsf{n}_1, \ldots, \mathsf{n}_k\}$. Let us consider a subset S of β descendants of n, such that $\{\mathsf{n}_1, \ldots, \mathsf{n}_k\} \subseteq S$. From (A), S exists. Let $\mathcal{F}'$ be the finite forest such that $\mathcal{F}' \subseteq \mathcal{F}$ and $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F}) \setminus S$. Since $\mathcal{F}'$ is constructed from $\mathcal{F}$ by only removing descendants of n, and in particular removing all its children, we have:

C. By (B), $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \mathtt{Hit}$,

D. $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \not\models \#\mathtt{desc} \geq 1$,

E. $\mathsf{card}(\mathsf{dom}(\mathcal{F}')) = \mathsf{card}(\mathsf{dom}(\mathcal{F})) - \beta$. Indeed, $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F}) \setminus S$ and $\mathsf{card}(S) = \beta$.

However, from the semantics of $\blacklozenge^\beta$ and (C)–(E), $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge^\beta(\mathtt{Hit} \wedge \neg\#\mathtt{desc} \geq 1)$, in contradiction with the second conjunct of $\#\mathtt{child} \geq \beta{+}1$. Thus, n has at least $\beta + 1$ $\mathcal{F}$-children and, from (B), it is an $\mathcal{F}$-descendant of t.
($\Leftarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ to be a forest such that n has at least $\beta + 1$ children and it is a descendant of t. Trivially, as every $\mathcal{F}$-child is an $\mathcal{F}$-descendant, we obtain $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \#\mathtt{desc} \geq \beta + 1$. We now show that $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ also satisfies $\blacksquare^\beta(\mathtt{Hit} \Rightarrow \#\mathtt{desc} \geq 1)$. Let $\mathcal{F}' \subseteq \mathcal{F}$ be the forest such that

Y. $\mathtt{n}$ is an $\mathcal{F}'$-descendant of $\mathtt{t}$;

Z. there is a set $S \subseteq \mathsf{dom}(\mathcal{F})$ such that $\mathsf{card}(S) = \beta$ and $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F}) \setminus S$.

To prove that $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \blacksquare^{\beta}(\mathtt{Hit} \Rightarrow \#\mathtt{desc} \geq 1)$ it is sufficient to establish $(\mathcal{F}', \mathtt{t}, \mathtt{n}) \models \#\mathtt{desc} \geq 1$. This is quite straightforward. From (Y), we conclude that $(\mathcal{F}', \mathtt{t}, \mathtt{n}) \models \mathtt{Hit}$. From (Z), since S has cardinality β and $\mathtt{n}$ has β+1 $\mathcal{F}$-children, $\mathtt{n}$ has at least one $\mathcal{F}'$-child (thus, an $\mathcal{F}'$-descendant). □

Given a syntactical element $\mathtt{S} \in \{\mathtt{size}(\varphi),\ \#\mathtt{desc},\ \#\mathtt{child}\}$, we write $\mathtt{S} = \beta$ for the formula $\mathtt{S} \geq \beta \wedge \neg \mathtt{S} \geq \beta{+}1$. For instance, $\#\mathtt{child} = \beta$ is the formula stating that $\mathtt{n}$ has exactly β children and it is a descendant of $\mathtt{t}$. We can now conclude the encoding of finite words.

Characterising words in ALT. We now move to the definition of the formula $\mathtt{word}_{\Sigma}$ that characterises the class of forests encoding words in Σ^*, where $\Sigma = [1, n]$. Recall that we assume Σ to be the alphabet of natural numbers in $[1, n]$, for some $n \geq 1$. Let $(\mathcal{F}, \mathtt{t}, \mathtt{n})$ be a pointed forest encoding the word $\mathfrak{w} = \mathtt{a}_1 \dots \mathtt{a}_k$, and let $\mathbb{M} = (\mathtt{n}_1, \dots, \mathtt{n}_k)$ be the set of its main nodes. Let us recall two of the properties of our encoding, expressed in Proposition 4, and introduce suitable formulae to express these properties. First, a node $\mathtt{n}$ encodes a symbol of $\mathfrak{w}$ (i.e. it is a main node) if it is a hit node with at least one child (Property (I)). To better reflect this property, we write $\mathtt{symb}$ for the formula $\#\mathtt{child} \geq 1$, so that $\mathtt{n}$ encodes a symbol of $\mathfrak{w}$ if and only if $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \mathtt{symb}$. Among the main nodes, $\mathtt{n}_1$ is the only one having the same number of descendants and children (Property (II)). For this property, given $S \subseteq \Sigma$, we introduce the formula $\mathtt{1st}_S$ that checks if the current evaluation node $\mathtt{n}$ corresponds to $\mathtt{n}_1$ and encodes a symbol in S:

$$\mathtt{1st}_S \stackrel{\text{def}}{=} \bigvee_{\beta \in S} (\#\mathtt{desc} = \beta + 1 \wedge \#\mathtt{child} = \beta + 1).$$

The formula $\mathtt{word}_{\Sigma}$ is defined as follows, and it is proved correct in Lemma 9,

$$\mathtt{word}_{\Sigma} \stackrel{\text{def}}{=} \neg\mathtt{size}(\mathtt{tchild}) \geq 2 \wedge \overbrace{(\langle \mathrm{U} \rangle \mathtt{Hit} \Rightarrow \langle \mathrm{U} \rangle \mathtt{symb})}^{\text{The target node has no descendants, or has a descendant that encodes a symbol.}}$$
$$\wedge\ [\mathrm{U}](\mathtt{symb} \Rightarrow \mathtt{1st}_{\Sigma} \vee \underbrace{(\neg \mathtt{1st}_{\{n+1\}} \wedge \blacklozenge\, \mathtt{1st}_{\Sigma})}_{\text{the current node encodes a symbol in } [1,n] \text{ and exactly one of its children encodes a symbol.}}).$$

The first two conjuncts of the formula $\mathtt{word}_{\Sigma}$ are quite self-explanatory. First, the target node $\mathtt{t}$ has at most one child. Second, if $\mathfrak{w}$ is the empty word then the forest does not contain hit nodes (alternatively, $\mathtt{t}$ does not have children), and otherwise there is a hit node encoding a symbol. The last conjunct is more complex, and subsumes the four properties in Proposition 4. Let $\mathtt{n}'$ be a node such that $(\mathcal{F}, \mathtt{t}, \mathtt{n}') \models \mathtt{symb}$. From the property (I), this means that $\mathtt{n}'$ is a main node. If it is the first node in the main path, then from the property (II) it must have the same number of descendants and children, and it must have $\mathtt{a} + 1$ children for some $\mathtt{a} \in \Sigma$ (Property (IV)). Basically, $\mathtt{n}'$ must satisfy $\mathtt{1st}_{\Sigma}$. Otherwise, suppose that $\mathtt{n}'$ encodes a node in the main path that is different from the first one. From the property (III), exactly one of its children, say $\mathtt{n}''$, must encode a symbol, whereas the other children are $\mathtt{a} + 1$ character nodes, for some $\mathtt{a} \in \Sigma$ (again, from the property (IV)). This means that removing $(\mathtt{n}'', \mathtt{n}')$ from $\mathcal{F}$ makes the node $\mathtt{n}'$ be the first node in the main path, according to property (II). So, $\mathtt{n}'$ satisfies $\neg \mathtt{1st}_{\{n+1\}} \wedge \blacklozenge\, \mathtt{1st}_{\Sigma}$. We prove that $\mathtt{word}_{\Sigma}$ characterise the class of forests encoding words in Σ^*.

Lemma 9. A pointed forest $(\mathcal{F}, \mathtt{t}, \mathtt{n})$ is an encoding of a word in Σ^* iff $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \mathtt{word}_{\Sigma}$.

Proof. ($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest encoding the word $\mathfrak{w} = \mathsf{a}_1 \dots \mathsf{a}_k \in [1, n]^*$, where $n \geq 1$. Let $\mathbb{M} = (\mathsf{n}_1, \dots, \mathsf{n}_k)$ and $\mathbb{C} = (\mathsf{N}_1, \dots, \mathsf{N}_k)$ be the main nodes and character nodes of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, respectively. Recapitulating Definition 3:

1. $\{\mathsf{n}_1, \dots, \mathsf{n}_k\}, \mathsf{N}_1, \dots, \mathsf{N}_k, \mathcal{F}[\mathtt{Miss}]_{\mathsf{t}}$ are pairwise disjoint sets, i.e. they do not share any node,
2. $\mathbb{M}$ and $\mathbb{C}$ are all the descendants of t, i.e. $(\mathcal{F}^{-1})^+(\mathsf{t}) = \{\mathsf{n}_1, \dots, \mathsf{n}_k\} \cup \bigcup_{j \in [1,k]} \mathsf{N}_j$,
3. n_k is the only child of t and for every $j \in [1, k-1]$ $\mathcal{F}(\mathsf{n}_j) = \mathsf{n}_{j+1}$,
4. for every $j \in [1, k]$, $\mathsf{card}(\mathsf{N}_j) = \mathsf{a}_j + 1$ and for every $\mathsf{n}' \in \mathsf{N}_j$, $\mathcal{F}(\mathsf{n}') = \mathsf{n}_j$.

Notice that (2) implies that if $\mathfrak{w}$ is the empty word, then $\mathcal{F}$ does not have hit nodes. If this is the case, then $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models [\mathrm{U}] \neg \mathtt{Hit}$, which implies that $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathrm{U} \rangle \mathtt{Hit} \Rightarrow \langle \mathrm{U} \rangle \mathtt{symb}$. Otherwise, the set of main nodes is non-empty, and by (3) and (4) we conclude that each main node is a descendant of t and has at least one child (as stated in Proposition 4). Again, this implies the satisfaction of $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathrm{U} \rangle \mathtt{Hit} \Rightarrow \langle \mathrm{U} \rangle \mathtt{symb}$. From (3) we have $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \neg \mathtt{size}(\mathtt{tchild}) \geq 2$, leaving us with only the last conjunct of $\mathtt{word}_{\Sigma}$ being open. Let us consider a node n' such that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \mathtt{symb}$. In particular, this implies that a main node exists and so $\mathfrak{w}$ is not empty. By (3) and (4), n' is a main node and so there is $j \in [1, k]$ such that $\mathsf{n}' = \mathsf{n}_j$. If $j = 1$, we prove that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \mathtt{1st}_{\Sigma}$. In this case, every child of n' is a character node from N_1 (and vice versa), which in turn does not have any children, so that n' has the same number of descendants and children (as stated in Proposition 4). Moreover, (4) implies that $\mathsf{card}(\mathsf{N}_1) = \mathsf{a}_1 + 1$. Thus, $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \#\mathtt{desc} = \mathsf{a}_1 + 1 \wedge \#\mathtt{child} = \mathsf{a}_1 + 1$, i.e one of the disjuncts of $\mathtt{1st}_{\Sigma}$. Otherwise, consider the case where $j \neq 1$. Let us prove that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \neg \mathtt{1st}_{\{n+1\}} \wedge \blacklozenge \mathtt{1st}_{\Sigma}$. Exactly one child of n' is a main node (i.e. n_{j-1}), whereas all other children are character nodes. As n_{j-1} is a main node, it has at least one child. So, $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \neg \mathtt{1st}_{\{n+1\}}$. Let $\mathcal{F}' \subseteq \mathcal{F}$ be the subforest such that $\mathcal{F}' = \mathcal{F} \setminus \{(\mathsf{n}_{j-1}, \mathsf{n}')\}$. On this subforest, all the $\mathcal{F}'$-children of n' are character nodes, more specifically (4) states that these children are the $\mathsf{a}_j + 1$ nodes from N_j. As in the case of $j = 1$, this implies that $(\mathcal{F}', \mathsf{t}, \mathsf{n}') \models \#\mathtt{desc} = \mathsf{a}_j + 1 \wedge \#\mathtt{child} = \mathsf{a}_j + 1$, i.e a disjunct of $\mathtt{1st}_{\Sigma}$. Thus, $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \blacklozenge \mathtt{1st}_{\Sigma}$.
($\Leftarrow$): Conversely, suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{word}_{\Sigma}$. From the first two conjuncts of $\mathtt{word}_{\Sigma}$ we have:

A. t has at most one child (from $\neg \mathtt{size}(\mathtt{tchild}) \geq 2$),
B. If $\mathcal{F}$ has a hit node, one descendant of t has a child (from $\langle \mathrm{U} \rangle \mathtt{Hit} \Rightarrow \langle \mathrm{U} \rangle \mathtt{symb}$).

Notice that if t does not have descendants then it trivially encodes the empty word. So, let us assume that t has at least one descendant. From (A), t has exactly one child, say $\overline{\mathsf{n}}$. Together with (B), this means that $(\mathcal{F}, \mathsf{t}, \overline{\mathsf{n}})$ must have a child (every other descendant of t is a descendant of $\overline{\mathsf{n}}$). We define the following subsets of the descendants of t:

- $\mathsf{M} \stackrel{\text{def}}{=} \{\mathsf{n}' \in (\mathcal{F}^{-1})^+(\mathsf{t}) \mid \mathcal{F}(\mathsf{n}'') = \mathsf{n}' \text{ for some } \mathsf{n}'' \in \mathcal{N}\}$, i.e. the *non-leaf* descendants of t,
- for $\mathsf{n}' \in \mathsf{M}$, $\mathsf{N}_{\mathsf{n}'} \stackrel{\text{def}}{=} \{\mathsf{n}'' \notin \mathsf{M} \mid \mathcal{F}(\mathsf{n}'') = \mathsf{n}'\}$, i.e. the *leafs* descendants of t that are children of n'.

Notice that $\overline{\mathsf{n}}$ belongs to M. Besides, the nodes in M are the only ones that satisfy $\mathtt{symb}$ (as they have a child and are descendants of t). To prove that $\mathcal{F}$ encodes a word in $[1, n]^+$ we show:

I. for each node $\mathsf{n}' \in \mathsf{M}$ there is at most one node $\mathsf{n}'' \in \mathsf{M}$ such that $\mathcal{F}(\mathsf{n}'') = \mathsf{n}'$. This shows the existence of a main path in the tree, made by the elements in M;

II. for every $\mathsf{n}' \in \mathsf{M}$, $\mathsf{card}(\mathsf{N}_{\mathsf{n}'}) \in [2, n+1]$. This shows that nodes of M encode symbols in $[1, n]$.

To prove (I) and (II), we use the fact that $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ satisfies the last conjunct of $\mathtt{word}_{\Sigma}$, i.e.

$$[\mathrm{U}](\mathtt{symb} \Rightarrow \mathtt{1st}_{\Sigma} \vee (\neg \mathtt{1st}_{\{n+1\}} \wedge \blacklozenge \mathtt{1st}_{\Sigma})).$$

Let us consider $\mathsf{n}' \in \mathsf{M}$. As it satisfies $\mathtt{symb}$, we have $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \mathtt{1st}_{\Sigma} \vee (\neg \mathtt{1st}_{\{n+1\}} \wedge \blacklozenge \mathtt{1st}_{\Sigma})$. If $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \mathtt{1st}_{\Sigma}$, as $\Sigma = [1, n]$ we conclude that the number of children and descendants of n are equal, and take a value in $[2, n+1]$. This implies that there is no $\mathsf{n}'' \in \mathsf{M}$ such that $\mathcal{F}(\mathsf{n}'') = \mathsf{n}'$ and $\mathsf{card}(\mathsf{N}_{\mathsf{n}}) \in [2, n+1]$. In this case, both (I) and (II) are verified. Otherwise, let us suppose that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \neg \mathtt{1st}_{\{n+1\}} \wedge \blacklozenge \mathtt{1st}_{\Sigma}$. In order to show that both (I) and (II) hold (concluding the proof), we reason by contradiction. First, suppose *ad absurdum* that (I) does not hold and so there are two distinct nodes $\mathsf{n}'', \mathsf{n}''' \in \mathsf{M}$ s.t. $\mathcal{F}(\mathsf{n}'') = \mathsf{n}' = \mathcal{F}(\mathsf{n}''')$. As n'' and n''' are both in M, they both have at least one child. However, this implies that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \not\models \blacklozenge \mathtt{1st}_{\Sigma}$, leading to a contradiction. Indeed, $\blacklozenge \mathtt{1st}_{\Sigma}$ is satisfied if it is possible to remove a single edge from the forest $\mathcal{F}$, leading to a finite forest $\mathcal{F}'$, so that n' is a $\mathcal{F}'$-descendant of t and the number of its children coincide with the number of its descendants. So, (I) holds. Lastly, suppose *ad absurdum* that (II) does not hold, and so $\mathsf{card}(\mathsf{N}_{\mathsf{n}'}) \notin [2, n+1]$. It is helpful to realise that the formula

$$\blacklozenge \mathtt{1st}_{\Sigma} \Rightarrow (\#\mathtt{child} \geq 2 \wedge \#\mathtt{child} \leq n+2),$$

is valid (recall that $\Sigma = [1, n]$). Indeed, consider a model $(\mathcal{F}_\star, \mathsf{t}_\star, \mathsf{n}_\star)$ that satisfies $\blacklozenge \mathtt{1st}_{\Sigma}$. By definition, there is an edge $e \in \mathcal{F}_\star$ such that $\mathcal{F}'_\star \stackrel{\text{def}}{=} \mathcal{F}_\star \setminus \{e\}$ enjoys $(\mathcal{F}'_\star, \mathsf{t}_\star, \mathsf{n}_\star) \models \mathtt{1st}_{\Sigma}$. This implies $(\mathcal{F}'_\star, \mathsf{t}_\star, \mathsf{n}_\star) \models \#\mathtt{child} = \beta$ for some $\beta \in [2, n+1]$, which in turn means that in $\mathcal{F}_\star$, $\mathsf{n}_\star$ can only have between 2 and $n+2$ $\mathcal{F}_\star$-children.
From this tautology we conclude that n' has between 2 and $n+2$ $\mathcal{F}$-children. If one of these children belongs to M, then of course $\mathsf{card}(\mathsf{N}_{\mathsf{n}'}) \in [2, n+1]$, leading to a contradiction, and thus proving (II). If instead every child of n' belongs to $\mathsf{N}_{\mathsf{n}'}$, then n' has the same number of descendants and children. Moreover, from the assumption $\mathsf{card}(\mathsf{N}_{\mathsf{n}'}) \notin [2, n+1]$, we derive $\mathsf{card}(\mathsf{N}_{\mathsf{n}'}) = n+2$. However, this is contradictory with the fact that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \neg \mathtt{1st}_{\{n+1\}}$. Thus, (II) holds. □

3.2. Intermezzo: inexpressibility results via Ehrenfeucht-Fraïssé games

Now that we are more familiar with ALT, before moving to the TOWER-hardness proof of its satisfiability problem, it is helpful to see some of the properties that the logic cannot express. Notably, these properties give us insights on what we should do (or rather, what we should avoid) in order to build very expressive queries in ALT in a concise way, needed to reach TOWER-hardness. In particular, we show that the expressive power of ALT is very weak when it comes to expressing properties of miss nodes. On these nodes, ALT can essentially only state the properties captured by Boolean combinations of the formulae $\mathtt{Miss}$ and $\mathtt{size}(\mathtt{Miss}) \geq \beta$. Therefore, the expressive power of ALT is almost entirely concerned with hit nodes. Another reason to look at inexpressibility results is that these results effectively reduce the set of forests that must be considered in order to solve the satisfiability problem. This in turn makes reductions from this problem to the satisfiability of other logics more immediate, as we show throughout Section 5.

Various mathematical tools from model theory are suited to prove inexpressibility results, as for example compactness theorems, Löwenheim-Skolem theorem and Ehrenfeucht-Fraïssé games. However, when dealing with logics interpreted on finite structures, Ehrenfeucht-Fraïssé games are the only major tool available. This is the case for ALT. Without extensively discussing the other tools and why they fail on finite structures (a clear presentation is given in [30]), let us briefly recall the *compactness* property.

Definition 10 (Compactness)**.** A logic $\mathcal{L}$ interpreted on the class of structures $\mathcal{M}$ is said to have the *compactness property* whenever, for every set S of formulae in $\mathcal{L}$, S has a model from $\mathcal{M}$ (i.e. S is consistent w.r.t. $\mathcal{M}$) if and only if every finite subset of S has a model from $\mathcal{M}$.

In the case that this property holds for ALT, it can be used to prove that a certain subclass C of pointed forests is not definable in the logic as follows. We first assume that C is characterised by a formula φ_C. We construct an infinite set of formulae S such that every finite subset of $S \cup \{\varphi_C\}$ is consistent, whereas the full set $S \cup \{\varphi_C\}$ is inconsistent. However, by compactness, this contradicts the correctness of φ_C. Therefore, C cannot be characterised in ALT. Unfortunately, we cannot rely on this technique, as the compactness property does not hold for ALT. Indeed, consider the infinite set of formulae $S \stackrel{\text{def}}{=} \{\blacklozenge^k \top \mid k \in \mathbb{N}\}$. It is clear that every finite subset $T \subseteq_{\mathsf{fin}} S$ is satisfied by every pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ such that $\mathsf{card}(\mathsf{dom}(\mathcal{F})) \geq \max\{k \mid \blacklozenge^k \top \in T\}$. However, S can only be satisfied by an infinite forest, and is therefore inconsistent with respect to the class of pointed forests. This invalidates Definition 10.

EF-games. As compactness fails, to prove inexpressibility results for ALT we adapt the notion of *Ehrenfeucht-Fraïssé games* (EF-games, in short) of first-order logic [30]. This has already be done for other relation-changing logics such as context logic for trees [11] and ambient logic [18]. EF-games are two-player games. One player is called the *spoiler* and the other is called the *duplicator*. In the case of ALT, a game is played on a *state* that is represented by a triple $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1), (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2), \mathsf{rk})$ made of two pointed forests $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$, and a rank rk. The rank, to be formally defined below, roughly represents the numbers of turns in the game. At each turn, the spoiler performs a *move* in one of the two pointed forests, which must be countered by the duplicator with a move on the other pointed forest. These moves are related to ALT, as they capture the semantics of the three modalities $\langle \mathrm{U} \rangle$, $\blacklozenge$ and $\blacklozenge^*$. The goal of the spoiler is to show that the two structures are different. The goal of the duplicator is to show that the two structures are similar. The notion of *being different* also traces back to the semantics of ALT: two pointed forests are different if and only if there is a formula of ALT that it is satisfied by only one of the two. The exact correspondence between the games and ALT is formalised with an adequacy result (Theorem 16, below).

In order to introduce the games, we need to define the rank of a formula φ in ALT.

Definition 11 (Rank)**.** The *rank* of φ a triple $(\mathsf{m}, \mathsf{s}, \mathsf{k}) \in \mathbb{N}^3$ where the *modal rank* m is the greatest nesting depth of the modal operator $\langle \mathrm{U} \rangle$ in φ. The *sabotage rank* s (resp. *repeated sabotage rank* k) is the greatest nesting depth of the operator $\blacklozenge$ (resp. $\blacklozenge^*$) in φ.

We write $\mathsf{ALT}_{\mathsf{rk}}$ for the set of formulae with rank $\mathsf{rk} \in \mathbb{N}^3$. We define the rank order $<_{rk}$ on $\mathbb{N}^3$.

Definition 12 (Rank order)**.** The *rank order* $<_{rk} \subseteq \mathbb{N}^3 \times \mathbb{N}^3$ is the relation defined as

$$(\mathsf{m}, \mathsf{s}, \mathsf{k}) <_{rk} (\mathsf{m}', \mathsf{s}', \mathsf{k}') \text{ iff } \mathsf{m} \leq \mathsf{m}', \mathsf{s} \leq \mathsf{s}', \mathsf{k} \leq \mathsf{k}' \text{ and } (\mathsf{m} < \mathsf{m}' \text{ or } \mathsf{s} < \mathsf{s}' \text{ or } \mathsf{k} < \mathsf{k}').$$

Notice that $<_{rk}$ is a well-founded strict order.

The EF-games for ALT are played with respect to a rank $\mathsf{rk} \in \mathbb{N}^3$, and they are formally defined in Figure 4. As we can see, at the beginning of the turn, we check whether the two atomic formulae `Hit` and `Miss` are satisfied by only one of the two pointed forests $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$. If this is the case, the spoiler wins. Otherwise, it must choose one move, among three possibilities which essentially capture the semantics of the modalities of ALT.

As usual, we say that a player has a *winning strategy* if it can play in a way that guarantees it the victory, regardless what the other player does. We write $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$ whenever the

EF-Game played on the state $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1), (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2), (\mathsf{m}, \mathsf{s}, \mathsf{k}))$

if there is $\pi \in \{\mathtt{Miss}, \mathtt{Hit}\}$ such that $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \pi$ iff $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \models \pi)$ does not hold
then the spoiler wins,
else the spoiler chooses $i \in \{1, 2\}$ and plays on $(\mathcal{F}_i, \mathsf{t}_i, \mathsf{n}_i)$.

The duplicator replies on $(\mathcal{F}_j, \mathsf{t}_j, \mathsf{n}_j)$ where $j \in \{1, 2\} \backslash \{i\}$.
The spoiler **must** choose one of the following moves (otherwise the duplicator wins).

$\langle \mathrm{U} \rangle$ **move**: if $\mathsf{m} \geq 1$ then the spoiler **can** choose to play a $\langle \mathrm{U} \rangle$ move. If he does so,

1. The spoiler selects a node $\mathsf{n}'_i \in \mathcal{N}$.
2. The duplicator **must** select a node $\mathsf{n}'_j \in \mathcal{N}$ (otherwise the spoiler wins).
3. The game continues on $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}'_1), (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}'_2), (\mathsf{m}{-}1, \mathsf{s}, \mathsf{k}))$.

$\blacklozenge$ **move**: if $\mathsf{s} \geq 1$ and $\mathsf{dom}(\mathcal{F}_i) \neq \varnothing$ then the spoiler **can** choose to play a $\blacklozenge$ move.

1. The spoiler selects a finite forest $\mathcal{F}'_i \subseteq \mathcal{F}_i$ such that $\mathsf{card}(\mathcal{F}'_i) = \mathsf{card}(\mathcal{F}_i) - 1$.
2. The duplicator **must** reply with a forest $\mathcal{F}'_j \subseteq \mathcal{F}_j$ s.t. $\mathsf{card}(\mathsf{dom}(\mathcal{F}'_j)) = \mathsf{card}(\mathsf{dom}(\mathcal{F}_j)) - 1$.
3. The game continues on $((\mathcal{F}'_1, \mathsf{t}_1, \mathsf{n}_1), (\mathcal{F}'_2, \mathsf{t}_2, \mathsf{n}_2), (\mathsf{m}, \mathsf{s}{-}1, \mathsf{k}))$.

$\blacklozenge^*$ **move**: if $\mathsf{k} \geq 1$ then the spoiler **can** choose to play a $\blacklozenge^*$ move.

1. The spoiler selects a finite forest $\mathcal{F}'_i \subseteq \mathcal{F}_i$.
2. The duplicator **must** reply with a finite forest $\mathcal{F}'_j \subseteq \mathcal{F}_j$.
3. The game continues on $((\mathcal{F}'_1, \mathsf{t}_1, \mathsf{n}_1), (\mathcal{F}'_2, \mathsf{t}_2, \mathsf{n}_2), (\mathsf{m}, \mathsf{s}, \mathsf{k}{-}1))$.

Figure 4: Ehrenfeucht-Fraïssé games for ALT.

duplicator has a winning strategy for the game $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1), (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2), \mathsf{rk})$. As our games are finite, by Zermelo's Theorem [43] (or Martin's Theorem [34]), they are determined: if the duplicator does not have a winning strategy then spoiler has one, and vice versa. We write $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \not\sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$ to state that the spoiler has a winning strategy.

Proposition 13. For every state of the game, one of the two players has a winning strategy.

We now aim at connecting the EF-games with ALT by proving that they are adequate with respect to the satisfaction relation $\models$ of ALT. This result, which is formalised in Theorem 16, requires first to state some properties of ranks. The first property is that $\mathsf{ALT}_{\mathsf{rk}}$ is a finite set of formulae, up to logical equivalence.

Lemma 14. For each rank $\mathsf{rk} \in \mathbb{N}^3$, $\mathsf{ALT}_{\mathsf{rk}}$ is finite up to logical equivalence.

The proof of this lemma, which follows a standard inductive argument on the rank rk, is given in Appendix A. Lemma 14 implies that given a rank rk, every pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ has a (finite) *characteristic formula* $\Gamma_{\mathsf{rk}}(\mathcal{F}, \mathsf{t}, \mathsf{n}) \in \mathsf{ALT}_{\mathsf{rk}}$ that is logically equivalent to the infinite conjunction $\bigwedge\{\varphi \in \mathsf{ALT}_{\mathsf{rk}} \mid (\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi\}$. Moreover, the formula $\Gamma_{\mathsf{rk}}(\mathcal{F}, \mathsf{t}, \mathsf{n})$ enjoys the following properties.

Lemma 15. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest and let $\mathsf{rk} \in \mathbb{N}^3$. (I) $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \Gamma_{\mathsf{rk}}(\mathcal{F}, \mathsf{t}, \mathsf{n})$, and (II) given a second pointed forest $(\mathcal{F}', \mathsf{t}', \mathsf{n}')$, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \Gamma_{\mathsf{rk}}(\mathcal{F}', \mathsf{t}', \mathsf{n}')$ iff $(\mathcal{F}', \mathsf{t}', \mathsf{n}') \models \Gamma_{\mathsf{rk}}(\mathcal{F}, \mathsf{t}, \mathsf{n})$.

Proof. The statement (I) follows directly by the definition of characteristic formula. For the statement (II), by symmetry we just need to show one direction. Assume $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \Gamma_{\mathsf{rk}}(\mathcal{F}', \mathsf{t}', \mathsf{n}')$. Let

$\psi \in \mathsf{ALT}_{\mathsf{rk}}$ and suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \psi$. To prove the result it is sufficient to show that $(\mathcal{F}', \mathsf{t}', \mathsf{n}') \models \psi$. *Ad absurdum*, suppose that $(\mathcal{F}', \mathsf{t}', \mathsf{n}') \not\models \psi$. Then by definition $(\mathcal{F}', \mathsf{t}', \mathsf{n}') \models \neg\psi$, and by definition of rank, we have that $\neg\psi \in \mathsf{ALT}_{\mathsf{rk}}$. Therefore, from the equivalence

$$\Gamma_{(\mathsf{m},\mathsf{s},\mathsf{k})}(\mathcal{F}', \mathsf{t}', \mathsf{n}') \overset{\text{by def}}{\equiv} \bigwedge\{\varphi \in \mathsf{ALT}_{\mathsf{m},\mathsf{s},\mathsf{k}} \mid (\mathcal{F}', \mathsf{t}', \mathsf{n}') \models \varphi\},$$

we derive $\models \Gamma_{(\mathsf{m},\mathsf{s},\mathsf{k})}(\mathcal{F}', \mathsf{t}', \mathsf{n}') \Rightarrow \neg\psi$. As $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \Gamma_{(\mathsf{m},\mathsf{s},\mathsf{k})}(\mathcal{F}', \mathsf{t}', \mathsf{n}')$, this implies $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \neg\psi$, in contradiction with the hypothesis $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \psi$. Hence, $(\mathcal{F}', \mathsf{t}', \mathsf{n}') \models \psi$. □

We are now ready to prove the adequacy of the games.

Theorem 16. Let $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$ be two pointed forests. Let $\mathsf{rk} \in \mathbb{N}^3$.

(I) If $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \varphi$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \not\models \varphi$ for some φ in $\mathsf{ALT}_{\mathsf{rk}}$, then $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \not\sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$.

(II) If for every φ in $\mathsf{ALT}_{\mathsf{rk}}$ $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \varphi$ iff $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \models \varphi)$, then $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$.

The statement (I) of Theorem 16, also called the *soundness* of the games, is proved by structural induction on φ. The *completeness* of the games, i.e. the statement (II), is proven by showing the contrapositive by induction on the rank and by cases on the first move that the spoiler makes in his winning strategy, which exists by Proposition 13.

Proof of Theorem 16(I). The proof by structural induction on φ is similar to the one in [11].

base case: $\varphi \in \{\mathtt{Hit}, \mathtt{Miss}\}$**.** From the hypothesis $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \varphi$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \not\models \varphi$ we conclude that spoiler wins the game (from the 1st and 2nd line of Figure 4).

For the induction steps, we omit the straigthforward cases of Boolean connectives. The three cases for $\varphi = \langle \mathrm{U} \rangle \psi$, $\varphi = \blacklozenge \psi$ and $\varphi = \blacklozenge^* \psi$ are all very similar. Below, we develop the case of $\varphi = \blacklozenge \psi$ and leave the other two to the reader.

induction step: $\varphi = \blacklozenge \psi$**.** By hypothesis $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \blacklozenge \psi$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \not\models \blacklozenge \psi$. There is a finite forest $\mathcal{F}_1' \subseteq \mathcal{F}_1$ such that $\mathsf{card}(\mathcal{F}_1') = \mathsf{card}(\mathcal{F}_1) - 1$ and $(\mathcal{F}_1', \mathsf{t}_1, \mathsf{n}_1) \models \psi$. Hence, $\mathsf{dom}(\mathcal{F}_1) \neq \varnothing$ and moreover by definition the sabotage rank of $\blacklozenge \psi$ is at least 1. Therefore, the spoiler can play a $\blacklozenge$ move. Suppose that the spoiler selects the structure $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1)$ and chooses exactly $\mathcal{F}_1'$. According to the game, the duplicator must choose a finite forest $\mathcal{F}_2' \subseteq \mathcal{F}_2$ such that $\mathsf{card}(\mathcal{F}_2') = \mathsf{card}(\mathcal{F}_2) - 1$. Since $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \not\models \blacklozenge \psi$, it holds that $(\mathcal{F}_2', \mathsf{t}_2, \mathsf{n}_2) \not\models \psi$. By induction hypothesis, the spoiler has a winning strategy for $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1'), (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2'), (\mathsf{m}, \mathsf{s} - 1, \mathsf{k}))$. So, by chosing $\mathcal{F}_1'$ the spoiler built a winning strategy for $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1), (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2), (\mathsf{m}, \mathsf{s}, \mathsf{k}))$. □

Proof of Theorem 16(II). We follow again the schema of the proof in [11]. We consider the contrapositive statement and thus prove that if $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$ does not hold, then there is φ in $\mathsf{ALT}_{\mathsf{rk}}$ s.t. $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \varphi$ iff $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \models \varphi)$ does not hold. Since the games are determined (Proposition 13) and ALT is closed under negation, we can alternatively show that

If $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \not\sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$ then there is φ in $\mathsf{ALT}_{\mathsf{rk}}$ s.t. $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \varphi$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \not\models \varphi$.

As already stated, the result is shown by induction on the rank rk, with respect to the order $<_{rk}$, and by cases on the first move that the spoiler makes in his winning strategy for the game $((\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1), (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2), \mathsf{rk})$. Below, we reserve the symbol φ for the formula that distinguishes the two models, as in the statement above (i.e. $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \varphi$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \not\models \varphi$).

base case: $\mathsf{rk} = (0,0,0)$**.** Since spoiler has a winning strategy, in particular it wins the game of rank $(0,0,0)$. By definition of the game, spoiler does not play any move, and from the 1st and 2nd line of Figure 4 one of the following must hold:

- $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \mathtt{Hit}$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \not\models \mathtt{Hit}$. Hence, $\varphi = \mathtt{Hit}$.
- $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \not\models \mathtt{Hit}$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \models \mathtt{Hit}$. Hence, $\varphi = \neg\mathtt{Hit}$.
- $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \mathtt{Miss}$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \not\models \mathtt{Miss}$. Hence, $\varphi = \mathtt{Miss}$.
- $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \not\models \mathtt{Miss}$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \models \mathtt{Miss}$. Hence, $\varphi = \neg\mathtt{Miss}$.

This case also holds for games on arbitrary rank $(\mathsf{m}, \mathsf{s}, \mathsf{k})$ where the spoiler wins simply from the conditions of the game that are imposed at the beginning of each round (1st and 2nd line of Figure 4), before playing any move.

In the induction steps, let us assume $\mathsf{rk} = (\mathsf{m}, \mathsf{s}, \mathsf{k})$. There are three cases to consider, depending on whether the spoiler plays a $\langle \mathrm{U} \rangle$, $\blacklozenge$ or $\blacklozenge^*$ move. As in the proof of Theorem 16(I), these three cases are all quite similar. Below we develop the one for $\blacklozenge$.

induction step: the spoiler plays a $\blacklozenge$ move. Assume that, by following its strategy, the spoiler chooses $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1)$ and plays a $\blacklozenge$ move. This implies $\mathsf{s} \geq 1$ and $\mathsf{dom}(\mathcal{F}_1) \neq \varnothing$. Let $\mathcal{F}'_1$ be the finite forest chosen by the spoiler, so $\mathcal{F}'_1 \subseteq \mathcal{F}_1$ and $\mathsf{card}(\mathcal{F}'_1) = \mathsf{card}(\mathcal{F}_1) - 1$. By Lemma 15(I), $(\mathcal{F}'_1, \mathsf{t}_1, \mathsf{n}_1) \models \Gamma_{(\mathsf{m},\mathsf{s}-1,\mathsf{k})}(\mathcal{F}'_1, \mathsf{t}_1, \mathsf{n}_1)$. Let φ be defined as the formula $\blacklozenge \Gamma_{(\mathsf{m},\mathsf{s}-1,\mathsf{k})}(\mathcal{F}'_1, \mathsf{t}_1, \mathsf{n}_1)$. By definition, $\varphi \in \mathsf{ALT}_{\mathsf{rk}}$ and this formula is satisfied by $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1)$. *Ad absurdum*, suppose that $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \models \varphi$. There is $\mathcal{F}'_2$ such that $\mathcal{F}'_2 \subseteq \mathcal{F}_2$, $\mathsf{card}(\mathcal{F}'_2) = \mathsf{dom}(\mathcal{F}_2) - 1$ and $(\mathcal{F}'_2, \mathsf{t}_2, \mathsf{n}_2) \models \Gamma_{(\mathsf{m},\mathsf{s}-1,\mathsf{k})}(\mathcal{F}'_1, \mathsf{t}_1, \mathsf{n}_1)$. By Lemma 15(II) together with the definition of characteristic formula, there is no formula in $\mathsf{ALT}_{(\mathsf{m},\mathsf{s}-1,\mathsf{k})}$ that can discriminate between $(\mathcal{F}'_1, \mathsf{t}_1, \mathsf{n}_1)$ and $(\mathcal{F}'_2, \mathsf{t}_2, \mathsf{n}_2)$. As our games are determined, by induction hypothesis this implies that the duplicator has a winning strategy for the game $((\mathcal{F}'_1, \mathsf{t}_1, \mathsf{n}_1), (\mathcal{F}'_2, \mathsf{t}_2, \mathsf{n}_2), (\mathsf{m}, \mathsf{s}-1, \mathsf{k}))$. However, this is contradictory as by hypothesis the spoiler has a winning strategy and the move it played is part of this strategy. Therefore, $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \models \varphi$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \not\models \varphi$.

Again, the proof is analogous for the case where the spoiler chooses $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$ and a finite tree $\mathcal{F}'_2 \subseteq \mathcal{F}_2$ such that $\mathsf{card}(\mathcal{F}'_2) = \mathsf{card}(\mathcal{F}_2) - 1$. In this case we obtain $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \not\models \psi$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2) \models \psi$ where $\psi = \blacklozenge \Gamma_{(\mathsf{m},\mathsf{s}-1,\mathsf{k})}(\mathcal{F}'_2, \mathsf{t}_2, \mathsf{n}_2)$. Hence, $\varphi \stackrel{\text{def}}{=} \neg\psi$ proves the result. □

Inexpressibility results for ALT. Thanks to Theorem 16, we can now use the EF-games for ALT to derive three easy inexpressibility results. Notably, these results are later helpful as they reduce the set of pointed forests needed in order to conclude that a formula φ of ALT is satisfiable.

Lemma 17. Let φ be a formula.

(I) φ is satisfiable iff it is satisfiable by a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ where $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$.

(II) Given a forest $\mathcal{F}$ and nodes $\mathsf{t} \in \mathcal{N}$ and $\mathsf{n}, \mathsf{n}' \notin \mathsf{dom}(\mathcal{F})$, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$ iff $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \varphi$.

(III) If $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$ then the duplicator has a winning strategy where it always replies to $\langle \mathrm{U} \rangle$ moves by selecting nodes in $\mathsf{dom}(\mathcal{F}_i) \cup \mathsf{ran}(\mathcal{F}_i)$, for some $i \in \{1, 2\}$.

As these results are quite straightforward, we just sketch their proof so that we can focus on how the EF-games are used without getting lost in technical details.

Proof (sketch). Consider the left-to-right direction of (I) (the other direction is obvious), and so let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest such that $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$. We modify $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ so that the target node is not in the domain of the forest. In particular, we consider a node $\mathsf{t}' \notin \mathsf{dom}(\mathcal{F}) \cup \mathsf{ran}(\mathcal{F})$ and define the forest $\mathcal{F}'(\mathsf{n}') \stackrel{\text{def}}{=} \textbf{if } \mathcal{F}(\mathsf{n}') = \mathsf{t} \textbf{ then } \mathsf{t}' \textbf{ else } \mathcal{F}(\mathsf{n}')$. Notice that $\mathsf{t}' \notin \mathsf{dom}(\mathcal{F}')$. We show that for every $\mathsf{rk} \in \mathbb{N}^3$, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \sim_{\mathsf{rk}} (\mathcal{F}', \mathsf{t}', \mathsf{n})$ with an easy induction on rk, leading to (I) directly by Theorem 16. The proof of (II) is even simpler, as we just need to prove that for all $\mathsf{rk} \in \mathbb{N}^3$, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \sim_{\mathsf{rk}} (\mathcal{F}, \mathsf{t}, \mathsf{n}')$, again by induction on the rank. (III) is a consequence of (II). □

Interestingly enough, Lemma 17(III) fundamentally implies that changing the definition of the set of nodes $\mathcal{N}$ to be finite, instead of infinite as we do throughout this work, does not change the expressive power nor the complexity of ALT.

The proof of Lemma 17(I) shows us how the games can be used in order to conclude an inexpressibility result. In general, we consider a property that we want to show to be not expressible in the logic, as for example the fact that the target node is in the domain of the forest (as in Lemma 17(I)). Then, for every rank $\mathsf{rk} \in \mathbb{N}^3$, we construct two pointed forests $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$ such that only one of the two has the wanted property. We show that $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$, which allows us to conclude that the property cannot be expressed, by Theorem 16. When this property is very simple, as it is the case for Lemma 17(I), it is possible to construct a single pair of finite forests $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$ so that for every $\mathsf{rk} \in \mathbb{N}^3$ we can prove $(\mathcal{F}_1, \mathsf{t}_1, \mathsf{n}_1) \sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}_2, \mathsf{n}_2)$.

Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest. We now show that ALT has a very limited expressive power with respect to the miss nodes. In particular, it can only check whether the current evaluation node n is a member of $\mathcal{F}[\texttt{Miss}]$ (with the formula $\texttt{Miss}$), and for the size of $\mathcal{F}[\texttt{Miss}]$ (with the formula $\texttt{size}(\texttt{Miss}) \geq \beta$). We formalise this inexpressibility result with the following lemma.

Lemma 18. Let $\mathsf{rk} = (\mathsf{m}, \mathsf{s}, \mathsf{k})$. Let $\mathcal{F}_1, \mathcal{F}_2$ be two forests, and $\mathsf{n}_1, \mathsf{n}_2, \mathsf{t} \in \mathcal{N}$. Suppose that

1. $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$ agree on the set of descendants of t, i.e. for every $\mathcal{F}_1$-descendant or $\mathcal{F}_2$-descendant n of t, $\mathcal{F}_1(\mathsf{n}) = \mathcal{F}_2(\mathsf{n})$, and if n_1 or n_2 are descendants of t, then $\mathsf{n}_1 = \mathsf{n}_2$,
2. $\mathsf{n}_1 \in \mathcal{F}_1[\texttt{Miss}]_{\mathsf{t}}$ if and only if $\mathsf{n}_2 \in \mathcal{F}_2[\texttt{Miss}]_{\mathsf{t}}$,
3. $\min(\mathsf{card}(\mathcal{F}_1[\texttt{Miss}]_{\mathsf{t}}), \mathsf{m} + \mathsf{s} + \mathsf{k}) = \min(\mathsf{card}(\mathcal{F}_2[\texttt{Miss}]_{\mathsf{t}}), \mathsf{m} + \mathsf{s} + \mathsf{k})$.

Then $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1) \sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$.

The proof, given in full details in Appendix A, is by induction on the rank rk, with respect to the strict order $<_{rk}$ and by cases on the move made by the spoiler in the game.

Lemma 18 shows that every property of miss nodes that is expressible in ALT is equivalent to a Boolean combination of the formulae $\texttt{Miss}$ and $\texttt{size}(\texttt{Miss}) \geq \beta$. Let us informally explain why. For example, let us suppose (ad absurdum) that there is a formula φ that characterises the set of pointed forests having a miss node with at least two children. Let us consider a rank $\mathsf{rk} = (\mathsf{m}, \mathsf{s}, \mathsf{k})$ and a pointed forest $(\mathcal{F}_1, \mathsf{t}, \mathsf{n})$ that satisfies the formula φ. We consider the subforest $\mathcal{F} \subseteq \mathcal{F}_1$ whose domain corresponds to the set of $\mathcal{F}_1$-descendants of t. We extend $\mathcal{F}$ to a forest $\mathcal{F}_2$ by (re)defining it on the nodes in $\mathcal{F}_1[\texttt{Miss}]_{\mathsf{t}}$ so that $\mathcal{F}_2[\texttt{Miss}]_{\mathsf{t}} = \mathcal{F}_1[\texttt{Miss}]_{\mathsf{t}}$ and none of these nodes has more than one $\mathcal{F}_2$-child (this construction can always be done). Notice that $\mathcal{F}_2$ is defined in a way that $(\mathcal{F}_1, \mathsf{t}, \mathsf{n})$ and $(\mathcal{F}_2, \mathsf{t}, \mathsf{n})$ satisfy the three properties (1), (2) and (3). We apply Lemma 18 to conclude $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}) \sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}, \mathsf{n})$, which in turn shows that $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}) \models \varphi$ by Theorem 16. However, $(\mathcal{F}_2, \mathsf{t}, \mathsf{n})$ is defined so that every node in $\mathcal{F}_2[\texttt{Miss}]_{\mathsf{t}}$ has at most one child. Thus, φ cannot characterise the set of models having a miss node with at least two children.

$\mathtt{a}_1 \dots \mathtt{a}_k \models \mathtt{1}$ iff $k = 1$ (i.e. the word $\mathtt{a}_1 \dots \mathtt{a}_k$ is a symbol of Σ),

$\mathtt{a}_1 \dots \mathtt{a}_k \models \mathtt{a}$ iff $\mathtt{a}_1 = \mathtt{a}$ (i.e the word is headed by the symbol $\mathtt{a}$),

$\mathtt{a}_1 \dots \mathtt{a}_k \models \varphi | \psi$ iff there is $j \in [1, k]$ such that $\mathtt{a}_1 \dots \mathtt{a}_j \models \varphi$ and $\mathtt{a}_j \dots \mathtt{a}_k \models \psi$.

Figure 5: Satisfaction relation for PITL, under locality principle.

As we discuss in the next section, the inexpressibility result shown in Lemma 18 plays a central role in the development of the reduction that leads to the TOWER-hardness of the satisfiability problem for ALT. In particular, most of the difficulties of this reduction stem from the fact that we need to get around the limited expressiveness that ALT has with respect to miss nodes.

4. The Complexity of ALT

We are now ready to show that the satisfiability problem for ALT is TOWER-complete. The hardness proof is by reduction from the satisfiability problem of Propositional Interval Temporal Logic under locality principle [37, 26]. A TOWER upper bound for SAT(ALT) (formally shown in Section 5) can be derived directly from the satisfiability problem of monadic second-order logic interpreted on tree-like structures [39].

4.1. Propositional Interval Temporal Logic

Propositional Interval Temporal Logic (PITL) is a logic that was introduced by B. Moszkowski in [37] for the verification of hardware components. It is interpreted on non-empty finite words over a finite alphabet of unary symbols Σ. Its formulae φ are from the grammar below ($\mathtt{a} \in \Sigma$):

$$\begin{array}{rlll} \pi ::= & \top & \textit{(true)} & \\ \mid & \mathtt{1} & \textit{(single predicate)} & \\ \mid & \mathtt{a} & \textit{(head predicate)} & \end{array} \qquad \begin{array}{rlll} \varphi ::= & \pi & & \textit{(atomic formulae)} \\ \mid & \varphi \wedge \varphi \mid \neg\varphi & & \textit{(Boolean connectives)} \\ \mid & \varphi \mid \varphi & & \textit{(composition operator)} \end{array}$$

The satisfaction relation $\models$ for the formulae of PITL is defined in Figure 5, with respect to a non-empty word $\mathtt{a}_1 \dots \mathtt{a}_k \in \Sigma^+$. Standard cases for $\top$ and Boolean connectives are omitted. The interpretation considered here is often called the *locality principle* interpretation of PITL. This name highlights the fact that the satisfaction of the predicate $\mathtt{a}$ only depends on the first symbol (i.e. the head) of the word. The main feature of this logic is its *composition operator* $|$. Intuitively, $\varphi|\psi$ is satisfied by words that can be "chopped" into a prefix and a suffix, so that the prefix satisfies φ and the suffix satisfies ψ. It is important to notice that these prefix and suffix overlap: the last symbol of the prefix is the first symbol of the suffix.

The satisfiability problem of PITL under locality principle is TOWER-complete. The fact that it is non-elementary decidable was proven by B. Moszkowski [37], by reduction from the non-emptiness problem of star-free regular languages previously studied by A. R. Meyer and L. J. Stockmeyer [36]. TOWER-completeness is then established from [41].

Proposition 19 (From [37, 41])**.** The satisfiabiilty problem of PITL is TOWER-complete.

As we have already shown that finite words can be encoded in ALT (Section 3.1), a promising route to prove that the satisfiability problem of ALT is TOWER-hard is by reduction from the

satisfiability problem of PITL. However, because of the limited expressive power that ALT has on miss nodes (see Lemma 18) we already know that the composition operator $\mid$ cannot be easily translated. Let us consider a pointed forest $(\mathcal{F}, \mathsf{n}, \mathsf{t})$ encoding a non-empty word $\mathfrak{w} = \mathsf{a}_1 \dots \mathsf{a}_k$. Moreover, let us assume that the set of main nodes of this encoding is $\mathbb{M} = (\mathsf{n}_1, \dots, \mathsf{n}_k)$. Chopping $\mathfrak{w}$ into two pieces means splitting in some way the main path $\mathsf{n}_1, \dots, \mathsf{n}_k$ of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ to then check that the word encoded by $\mathsf{n}_1, \dots, \mathsf{n}_i$ satisfies a certain formula φ, whereas the one encoded by $\mathsf{n}_i, \dots, \mathsf{n}_k$ satisfies a formula ψ. Neglecting the fact that the two structures share the node n_i, the main problem in doing this is that after the split the nodes $\mathsf{n}_1, \dots, \mathsf{n}_i$ stop being descendants of the target node, and so they become miss nodes. As a consequence of Lemma 18, we know that ALT cannot check in any way what is the word encoded by these nodes. Trivial translations from PITL to ALT seem therefore impossible.

4.2. PITL *on marked words*

To solve the issue of capturing the composition operator of PITL in ALT, we consider an alternative interpretation of PITL where, instead of chopping a word, the operator $\mid$ marks the symbol where the cut should have taken place. As we will see, the alternative interpretation is equivalent to the one under locality principle given above, so that the TOWER-completeness result of Proposition 19 still hold. We start by introducing the notions of marking of a symbol, an alphabet and a word, as well as a notion of decomposition for marked words. For simplicity, throughout the section we fix a (non-empty) finite alphabet Σ.

Definition 20 (Markings)**.** Let $\overline{\Sigma}$ be an alphabet disjoint from Σ and such that $\mathsf{card}(\overline{\Sigma}) = \mathsf{card}(\Sigma)$. A *marking* for Σ is a bijection $\overline{(.)} : \Sigma \to \overline{\Sigma}$, relating a symbol $\mathsf{a} \in \Sigma$ to its *marked variant* $\overline{\mathsf{a}} \in \overline{\Sigma}$.

We fix $\Sigma_\bullet$ to be the alphabet $\Sigma \cup \overline{\Sigma}$. A word of $\Sigma_\bullet$ is *marked* if it has some symbol from $\overline{\Sigma}$.

Definition 21 (Marked word decomposition)**.** Given a marked word $\mathfrak{w} \in \Sigma_\bullet^+$, we write $\Delta(\mathfrak{w})$ for the *decomposition* $(\mathfrak{w}', \overline{\mathsf{a}}, \mathfrak{w}'')$ where $\mathfrak{w}' \in \Sigma^*$ is not marked, $\overline{\mathsf{a}} \in \overline{\Sigma}$ is marked, and $\mathfrak{w} = \mathfrak{w}' \, \overline{\mathsf{a}} \, \mathfrak{w}''$.

Notice that the decomposition $\Delta(\mathfrak{w}) = (\mathfrak{w}', \overline{\mathsf{a}}, \mathfrak{w}'')$ of a marked word $\mathfrak{w}$ is uniquely defined, as the word $\mathfrak{w}' \, \overline{\mathsf{a}}$ is the (only) prefix of $\mathfrak{w}$ ending with its first marked symbol. As we will see, the notion of satisfiability we are about to define only depends on these prefixes.

We interpret PITL on marked words. Given a marked word $\mathfrak{w} \in \Sigma_\bullet^+$, the new satisfaction relation $\models_\bullet$ for the formulae of PITL is given in Figure 6, again omitting standard cases for $\top$ and Boolean connectives. The semantics of the predicates $\mathtt{1}$ and $\mathtt{a}$ is quite simple, and reflects the fact that the satisfaction of a formula depends on the only prefix of a marked word $\mathfrak{w}$ that ends with its first marked symbol. For the predicate $\mathtt{1}$ to be satisfied, the word $\mathfrak{w}$ must begin with a marked symbol, so in the decomposition $\Delta(\mathfrak{w}) = (\mathfrak{w}', \overline{\mathsf{a}}, \mathfrak{w}'')$ the word $\mathfrak{w}'$ is the *empty word* ϵ. For the predicate $\mathtt{a}$, we simply check if $\mathfrak{w}$ is headed by the symbol $\mathtt{a}$ or its marked variant $\overline{\mathtt{a}}$. The definition of $\varphi \mid \psi$ is more involved. Let us consider the prefix $\mathsf{a}_1 \dots \mathsf{a}_{k-1} \, \overline{\mathsf{a}_k}$ of $\mathfrak{w}$ that ends with the first marked symbol. In order for $\mathfrak{w} \models_\bullet \varphi \mid \psi$ to hold, we must find a position $j \in [1, k]$ inside this prefix so that φ is satisfied by the word obtained from $\mathfrak{w}$ by marking the j-th symbol (if it is not already marked), whereas ψ is satisfied by the suffix of $\mathfrak{w}$ starting in j. In the formal definition given in Figure 6, this idea is split into four cases (a)–(d), depending on truthiness of $j = 1$ and $j = k$. For example, (a) correspond to the case where $j = k = 1$. This split is done as it better reflects the encoding of PITL in ALT.

$\mathfrak{w} \models_\bullet 1$ iff $\Delta(\mathfrak{w}) = (\mathfrak{w}', \overline{\mathtt{a}}, \mathfrak{w}'')$ and $\mathfrak{w}' = \epsilon$ (i.e. $\mathfrak{w}$ is headed by a marked symbol),

$\mathfrak{w} \models_\bullet \mathtt{a}$ iff $\mathfrak{w}$ is headed by the symbol $\mathtt{a}$ or the symbol $\overline{\mathtt{a}}$,

$\mathfrak{w} \models_\bullet \varphi \mid \psi$ iff $\Delta(\mathfrak{w}) = (\mathfrak{w}', \overline{\mathtt{a}}, \mathfrak{w}'')$ and there is a symbol $\mathtt{b} \in \Sigma$ such that

(a) $\mathfrak{w}' = \epsilon$ and $\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi \wedge \psi$

or (b) $\mathfrak{w}' = \mathtt{b}\mathfrak{w}_2$ and $\overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi$ and $\mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \psi$, for some $\mathfrak{w}_2 \in \Sigma^*$

or (c) $\mathfrak{w}' \neq \epsilon$ and $\mathfrak{w}'\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi$ and $\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \psi$

or (d) $\mathfrak{w}' = \mathfrak{w}_1\mathtt{b}\mathfrak{w}_2$ and $\mathfrak{w}_1\,\overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi$ and $\mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \psi$, for some $\mathfrak{w}_1 \in \Sigma^+$ and $\mathfrak{w}_2 \in \Sigma^*$.

Figure 6: Satisfaction relation for PITL on marked words.

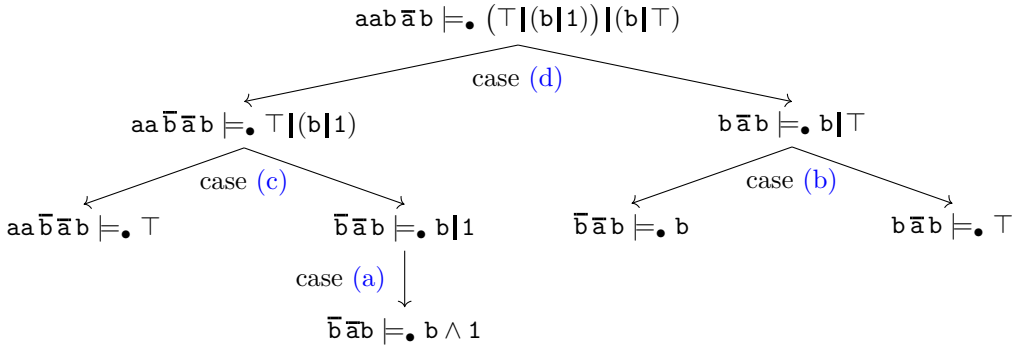


Figure 7: Example of the satisfaction of a formula on marked words.

Example 22. Consider the alphabets $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ and $\overline{\Sigma} = \{\overline{\mathtt{a}}, \overline{\mathtt{b}}\}$. The schema in Figure 7 certifies that the marked word $\mathtt{aab}\,\overline{\mathtt{a}}\,\mathtt{b}$ satisfies the formula $\big(\top \mid (\mathtt{b} \mid 1)\big) \mid (\mathtt{b} \mid \top)$. At each step we highlight which of the four cases in the definition of $\varphi \mid \psi$ is used. For instance, let us pick the first step of the schema above, in which the case (d) in the definition of $\varphi \mid \psi$ is used. According to this case $\mathtt{aab}\,\overline{\mathtt{a}}\,\mathtt{b} \models_\bullet \big(\top \mid (\mathtt{b} \mid 1)\big) \mid (\mathtt{b} \mid \top)$ holds as the word obtained by marking the third symbol, i.e. $\mathtt{aa}\,\overline{\mathtt{b}}\,\overline{\mathtt{a}}\,\mathtt{b}$, satisfies $\top \mid (\mathtt{b} \mid 1)$, whereas the suffix of the word starting on this symbol, i.e. $\mathtt{b}\,\overline{\mathtt{a}}\,\mathtt{b}$, satisfies $\mathtt{b} \mid \top$. As we will show in a moment, marking symbols and considering suffixes of words are operations that can be simulated in ALT. Let us consider the decomposition $\Delta(\mathtt{aab}\,\overline{\mathtt{a}}\,\mathtt{b}) = (\mathtt{aab}, \overline{\mathtt{a}}, \mathtt{b})$. One can check that the word $\mathtt{aaba}$, obtained by concatenating the prefix $\mathtt{aab}$ of the decomposition with the non-marked symbol that corresponds to $\overline{\mathtt{a}}$ in the decomposition, satisfies the formula $\big(\top \mid (\mathtt{b} \mid 1)\big) \mid (\mathtt{b} \mid \top)$ in the standard semantics of PITL. Lemma 23 (below) shows that this is always the case.

The semantics on marked words is related to the standard semantics of PITL as follows.

Lemma 23. Let $\mathfrak{w}' \in \Sigma^*$, $\mathtt{a} \in \Sigma$ and $\mathfrak{w}'' \in \Sigma^*_\bullet$. Let φ be a formula in PITL. We have,

$$\mathfrak{w}'\mathtt{a} \models \varphi \text{ if and only if } \mathfrak{w}'\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi.$$

Proof. The proof is by structural induction on φ (with the natural induction hypothesis stating that the lemma holds for strict subformulae of φ). Let us write $\mathfrak{w}$ for $\mathfrak{w}'\mathtt{a}$, and $\overline{\mathfrak{w}}$ for the marked word $\mathfrak{w}'\,\overline{\mathtt{a}}\,\mathfrak{w}''$. The base case with the atomic formulæ $\mathtt{1}$ and $\mathtt{b} \in \Sigma$ is by easy verification.

base case: $\varphi = \mathtt{1}$**.** The following double implications show the result:

$$\begin{array}{lll} \mathfrak{w} \models \mathtt{1} \quad \text{if and only if} \quad \mathfrak{w}' = \epsilon & (\textit{by definition of} \models \textit{and}\ \mathfrak{w} = \mathfrak{w}'\mathtt{a}) \\ \qquad\qquad \text{if and only if} \quad \overline{\mathfrak{w}} = \overline{\mathtt{a}}\,\mathfrak{w}'' & (\textit{from}\ \overline{\mathfrak{w}} = \mathfrak{w}'\,\overline{\mathtt{a}}\,\mathfrak{w}'') \\ \qquad\qquad \text{if and only if} \quad \overline{\mathfrak{w}} \models_\bullet \mathtt{1}. & (\textit{by definition of} \models_\bullet) \end{array}$$

base case: $\varphi = \mathtt{b}$**, where** $\mathtt{b} \in \Sigma$**.** The following double implication shows the result:

$$\begin{array}{lll} \mathfrak{w} \models \mathtt{b} \quad \text{if and only if} \quad \mathfrak{w}'\mathtt{a} \text{ is headed by } \mathtt{b} & (\textit{by definition of} \models \textit{and}\ \mathfrak{w} = \mathfrak{w}'\mathtt{a}) \\ \qquad\qquad \text{if and only if} \quad \overline{\mathfrak{w}} \text{ is headed by } \mathtt{b} \text{ or } \overline{\mathtt{b}} & (\textit{from}\ \overline{\mathfrak{w}} = \mathfrak{w}'\,\overline{\mathtt{a}}\,\mathfrak{w}'') \\ \qquad\qquad \text{if and only if} \quad \overline{\mathfrak{w}} \models_\bullet \mathtt{b}. & (\textit{by definition of} \models_\bullet) \end{array}$$

The cases for Boolean connectives are obvious. We prove the result for the composition operator.

induction step: $\varphi = \varphi_1 \mid \varphi_2$**.** Again, the result holds following a series of double implications:

$\mathfrak{w} \models \varphi_1 \mid \varphi_2$

$\Leftrightarrow$ there are $\mathtt{b} \in \Sigma$ and $\mathfrak{w}_1, \mathfrak{w}_2 \in \Sigma^*$ s.t. $\mathfrak{w} = \mathfrak{w}_1\mathtt{b}\mathfrak{w}_2$, $\mathfrak{w}_1\mathtt{b} \models \varphi_1$ and $\mathtt{b}\mathfrak{w}_2 \models \varphi_2$
(*by definition of* $\models$)

$\Leftrightarrow$ there are $\mathtt{b} \in \Sigma$ and $\mathfrak{w}_1, \mathfrak{w}_2 \in \Sigma^*$ s.t. $\mathfrak{w} = \mathfrak{w}_1\mathtt{b}\mathfrak{w}_2$ and

$$\begin{array}{llll} & \text{(a)} & \mathfrak{w}_1 = \epsilon,\ \mathfrak{w}_2 = \epsilon,\ \mathtt{b} \models \varphi_1 \text{ and } \mathtt{b} \models \varphi_2, & \text{(in this case, } \mathtt{b} = \mathtt{a}) \\ \text{or} & \text{(b)} & \mathfrak{w}_1 = \epsilon,\ \mathfrak{w}_2 \neq \epsilon,\ \mathtt{b} \models \varphi_1 \text{ and } \mathtt{b}\mathfrak{w}_2 \models \varphi_2, & \text{(in this case, } \exists \mathfrak{w}_2'\ \mathfrak{w}_2 = \mathfrak{w}_2'\mathtt{a}) \\ \text{or} & \text{(c)} & \mathfrak{w}_1 \neq \epsilon,\ \mathfrak{w}_2 = \epsilon,\ \mathfrak{w}_1\mathtt{b} \models \varphi_1 \text{ and } \mathtt{b} \models \varphi_2, & \text{(in this case, } \mathtt{b} = \mathtt{a},\ \mathfrak{w}_1 = \mathfrak{w}') \\ \text{or} & \text{(d)} & \mathfrak{w}_1 \neq \epsilon,\ \mathfrak{w}_2 \neq \epsilon,\ \mathfrak{w}_1\mathtt{b} \models \varphi_1 \text{ and } \mathtt{b}\mathfrak{w}_2 \models \varphi_2. & \text{(in this case, } \exists \mathfrak{w}_2'\ \mathfrak{w}_2 = \mathfrak{w}_2'\mathtt{a}) \end{array}$$

(*by case distinction, on the truthiness of* $\mathfrak{w}_1 = \epsilon$ *and* $\mathfrak{w}_2 = \epsilon$)

$\Leftrightarrow$ there are $\mathtt{b} \in \Sigma$ and $\mathfrak{w}_1, \mathfrak{w}_2 \in \Sigma^*$ s.t. $\mathfrak{w} = \mathfrak{w}_1\mathtt{b}\mathfrak{w}_2$ and

$$\begin{array}{llll} & \text{(a)} & \mathfrak{w}_1 = \epsilon,\ \mathfrak{w}_2 = \epsilon,\ \mathtt{b} = \mathtt{a},\ \overline{\mathtt{b}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \overline{\mathtt{b}}\,\mathfrak{w}'' \models_\bullet \varphi_2, \\ \text{or} & \text{(b)} & \mathfrak{w}_1 = \epsilon,\ \exists \mathfrak{w}_2' \in \Sigma^* \text{ s.t. } \mathfrak{w}_2 = \mathfrak{w}_2'\mathtt{a},\ \overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_2, \\ \text{or} & \text{(c)} & \mathfrak{w}_1 \neq \epsilon,\ \mathfrak{w}_2 = \epsilon,\ \mathtt{b} = \mathtt{a},\ \mathfrak{w}'\,\overline{\mathtt{b}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \overline{\mathtt{b}}\,\mathfrak{w}'' \models_\bullet \varphi_2, \\ \text{or} & \text{(d)} & \mathfrak{w}_1 \neq \epsilon,\ \exists \mathfrak{w}_2' \in \Sigma^* \text{ s.t. } \mathfrak{w}_2 = \mathfrak{w}_2'\mathtt{a},\ \mathfrak{w}_1\,\overline{\mathtt{b}}\,\mathfrak{w}_2'\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \mathtt{b}\mathfrak{w}_2'\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_2. \end{array}$$

(*by induction hypothesis, on all four cases*)

$\Leftrightarrow$ there is a symbol $\mathtt{b} \in \Sigma$ such that

$$\begin{array}{llll} & \text{(a)} & \mathfrak{w}' = \epsilon \text{ and } \overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \wedge \varphi_2 \\ \text{or} & \text{(b)} & \mathfrak{w}' = \mathtt{b}\mathfrak{w}_2 \text{ and } \overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_2, \text{ for some } \mathfrak{w}_2 \in \Sigma^* \\ \text{or} & \text{(c)} & \mathfrak{w}' \neq \epsilon \text{ and } \mathfrak{w}'\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_2 \\ \text{or} & \text{(d)} & \mathfrak{w}' = \mathfrak{w}_1\mathtt{b}\mathfrak{w}_2 \text{ and } \mathfrak{w}_1\,\overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_2, \\ & & \text{for some } \mathfrak{w}_1 \in \Sigma^+ \text{ and } \mathfrak{w}_2 \in \Sigma^*, \end{array}$$

(*by easy manipulation of the formula*)

$\Leftrightarrow \overline{\mathfrak{w}} \models_\bullet \varphi_1 \mathbf{|} \varphi_2$. (*by definition of* $\models_\bullet$) □

4.3. Reducing PITL *to* ALT

The alternative interpretation of PITL allows us to reduce the satisfiability problem of PITL to the satisfiability problem of ALT in a rather neat way. Once again, let us consider the alphabets $\Sigma, \overline{\Sigma}$ and $\Sigma_\bullet = \Sigma \cup \overline{\Sigma}$ of the previous section, and let us assume $\Sigma = [1, n]$ for some natural number $n \geq 1$. We consider the bijection $\mathfrak{f} : \Sigma_\bullet \to [1, 2n]$ defined as $\mathfrak{f}(\mathtt{a}) \stackrel{\text{def}}{=} 2\mathtt{a}$ for every symbol $\mathtt{a} \in \Sigma$, and defined as $\mathfrak{f}(\overline{\mathtt{a}}) \stackrel{\text{def}}{=} 2\mathtt{a} - 1$ for every marked symbol $\overline{\mathtt{a}} \in \overline{\Sigma}$. We write $\mathfrak{f}(\mathtt{a}_1 \dots \mathtt{a}_k)$ to denote the word $\mathfrak{f}(\mathtt{a}_1) \dots \mathfrak{f}(\mathtt{a}_k)$. Based on these definitions, $\mathfrak{f}$ maps $\Sigma_\bullet$ into the alphabet $[1, 2n]$, whose words can be encoded into trees (as in Section 3.1). In these trees, each symbol $\mathtt{a} \in \Sigma$ corresponds to a main node having $2\mathtt{a} + 1$ children that are character nodes (recall that we use $\mathtt{a} + 1$ children to encode the symbol $\mathtt{a}$). Similarly, each marked symbol $\overline{\mathtt{a}} \in \overline{\Sigma}$ corresponds to a main node having $2\mathtt{a}$ children that are character nodes. Let us consider a node n encoding a symbol in Σ. Because of the above distribution of non-marked and marked symbols, removing exactly one child of n that is a character node is equivalent to marking the symbol that n encodes. Let us now see how to capture this encoding in ALT.

Let us fix a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, which we suppose encodes a marked word $\mathfrak{w} \in \Sigma_\bullet$. We can check if the current node n encodes a marked symbol from $\overline{\Sigma}$ with the following formula:

$$\mathtt{mark}_\Sigma \stackrel{\text{def}}{=} \bigvee_{\mathtt{a} \in \Sigma} \big((\#\mathtt{child} = 2\mathtt{a} \wedge \mathtt{1st}_{[1,2n]}) \vee (\#\mathtt{child} = 2\mathtt{a} + 1 \wedge \neg \mathtt{1st}_{[1,2n]}) \big)$$

We recall that marked symbols correspond to nodes with $2\mathtt{a}$ children that are character nodes, for some $\mathtt{a} \in \Sigma$. In order to capture this notion, the formula $\overline{\Sigma}$ distinguishes the case where the current node n is the first node in the main path (whose only children are character nodes) from the case where n is not the first node in the main path (hence, it has one child in the main path).

As already stated, $\mathfrak{w} \models_\bullet \varphi$ examines the prefix of $\mathfrak{w}$ that ends with the first marked symbol. To correctly reduce PITL to ALT we need to be able to find the part of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ that corresponds to this prefix. We can do so by noticing that this part is the only subtree whose root encodes a marked symbol and is a $\mathcal{F}$-descendant of every other node encoding marked symbols. This characterisation requires us to track the number of nodes encoding marked symbols in $(\mathcal{F}, \mathsf{t}, \mathsf{n})$. To do so, first define a formula $\mathtt{marks}_\Sigma \geq \beta$ stating that the forest has at least $\beta \in \mathbb{N}$ nodes encoding marked symbols. Luckily, we can rely on the formula $\mathtt{size}(\varphi) \geq \beta$ defined in Section 3.1, and define $\mathtt{marks}_\Sigma \geq \beta$ simply as $\mathtt{size}(\mathtt{mark}_\Sigma) \geq \beta$. Unfortunately, we cannot rely exactly on Lemma 6 to prove that this formula is correct, as the formula $\mathtt{mark}_\Sigma$ does not enjoy the property (2) required by this lemma. Similarly, we introduce the formula $\#\mathtt{markAnc}_\Sigma \geq \beta$ which states that the current evaluation node encodes a symbol and has at least β ancestors that encode marked symbols. It is defined as follows:

$$\#\mathtt{markAnc}_\Sigma \geq \beta \stackrel{\text{def}}{=} \mathtt{symb} \wedge \blacklozenge(\neg \mathtt{inDom} \wedge \mathtt{marks}_\Sigma \geq \beta).$$

The following lemma assures that the three formulae $\mathtt{mark}_\Sigma$, $\mathtt{marks}_\Sigma \geq \beta$ and $\#\mathtt{markAnc}_\Sigma \geq \beta$ are correct, and highlights their semantics.

Lemma 24. Let $\mathfrak{w} \in \Sigma_\bullet^+$ and let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest encoding the word $\mathfrak{f}(\mathfrak{w}) \in [1, 2n]^+$.

(I) $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{mark}_\Sigma$ iff n encodes a marked symbol of $\Sigma_\bullet$.

(II) $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{marks}_\Sigma \geq \beta$ iff $\mathcal{F}$ contains at least β nodes encoding marked symbols of $\Sigma_\bullet$.

$$
\begin{aligned}
\tau_\beta(\top) &\stackrel{\text{def}}{\equiv} \top,\\
\tau_\beta(\mathtt{1}) &\stackrel{\text{def}}{\equiv} \langle \mathrm{U}\rangle(\mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma),\\
\tau_\beta(\mathtt{a}) &\stackrel{\text{def}}{\equiv} \langle \mathrm{U}\rangle\, \mathtt{1st}_{[2\mathtt{a}-1,2\mathtt{a}]},\\
\tau_\beta(\neg\psi) &\stackrel{\text{def}}{\equiv} \neg\tau_\beta(\psi),\\
\tau_\beta(\psi_1 \wedge \psi_2) &\stackrel{\text{def}}{\equiv} \tau_\beta(\psi_1) \wedge \tau_\beta(\psi_2),\\
\tau_\beta(\psi_1 \mid \psi_2) &\stackrel{\text{def}}{\equiv} \langle \mathrm{U}\rangle\Big(\mathtt{symb} \wedge \Big((\mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma \wedge \tau_\beta(\psi_1) \wedge \tau_\beta(\psi_2))\\
&\qquad \vee (\mathtt{1st}_{[1,2n]} \wedge \neg\mathtt{mark}_\Sigma \wedge \blacklozenge(\mathtt{mark}_\Sigma \wedge \tau_{\beta+1}(\psi_1)) \wedge \tau_\beta(\psi_2))\\
&\qquad \vee (\neg\mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma \wedge \#\mathtt{markAnc}_\Sigma \geq \beta - 1 \wedge \tau_\beta(\psi_1) \wedge \blacklozenge(\mathtt{1st}_{[1,2n]} \wedge \tau_\beta(\psi_2)))\\
&\qquad \vee (\neg\mathtt{1st}_{[1,2n]} \wedge \neg\mathtt{mark}_\Sigma \wedge \#\mathtt{markAnc}_\Sigma \geq \beta \wedge \blacklozenge(\mathtt{mark}_\Sigma \wedge \tau_{\beta+1}(\psi_1))\\
&\qquad\qquad \wedge \blacklozenge(\mathtt{1st}_{[1,2n]} \wedge \tau_\beta(\psi_2)))\Big)\Big).
\end{aligned}
$$

Figure 8: Translation from PITL to ALT.

(III) $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \#\mathtt{markAnc}_\Sigma \geq \beta$ iff $\mathtt{n}$ has at least β ancestors encoding marked symbols of $\Sigma_\bullet$.

The proof of this lemma is given in Appendix B. Whereas statements (I) and (III) are shown by simply unrolling the definitions, the proof of (II) follows by a straightforward induction on β.

We now show how to translate formulae of PITL into formulae of ALT. Given a formula φ in PITL with symbols from $\Sigma = [1, n]$, we introduce its translation $\tau_\beta(\varphi)$ in ALT, where the index β is a positive natural number that we use to track the number of nodes encoding marked symbols. The translation is defined in Figure 8. It is homomorphic for $\top$ and Boolean connectives. For the predicates $\mathtt{1}$ and $\mathtt{a}$, the translation faithfully represents the relation $\models_\bullet$. In the case of $\mathtt{1}$, it requires the first node in the main path to correspond to a marked node. For the predicate $\mathtt{a}$, it checks whether this node encodes the symbols $2\mathtt{a}-1$ or $2\mathtt{a}$ which, by definition of $\mathfrak{f}$, correspond to $\overline{\mathtt{a}} \in \overline{\Sigma}$ and $\mathtt{a} \in \Sigma$, respectively. Lastly, the formula $\tau_\beta(\varphi \mid \psi)$ follows very closely the definition of the relation $\models_\bullet$: after the prefix "$\langle \mathrm{U}\rangle(\mathtt{symb} \wedge \dots$", the formula splits into four disjuncts, one for each of the cases in the definition of $\varphi \mid \psi$. For instance, let us consider a word $\mathfrak{w}$ such that $\Delta(\mathfrak{w}) = (\mathfrak{w}', \overline{\mathtt{a}}, \mathfrak{w}'')$. The second disjunct of $\tau_\beta(\varphi \mid \psi)$ encodes the case (b) in the definition of $\mathfrak{w} \models_\bullet \varphi \mid \psi$, as schematised below:

PITL	there is $\mathtt{b} \in \Sigma$...	$\exists \mathfrak{w}_2 \in \Sigma^*$ s.t. $\mathfrak{w}' = \mathtt{b}\mathfrak{w}_2$	and	$\overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models \varphi$	and	$\mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models \psi$
ALT	$\langle \mathrm{U}\rangle(\mathtt{symb}$...	$\mathtt{1st}_{[1,2n]} \wedge \neg\mathtt{mark}_\Sigma$	$\wedge$	$\blacklozenge(\mathtt{mark}_\Sigma \wedge \tau_{\beta+1}(\varphi))$	$\wedge$	$\tau_\beta(\psi)$

The lemma below ensures that the translation matches the semantics of the formula in PITL under the interpretation on marked words.

Lemma 25. Let $\mathfrak{w} \in \Sigma_\bullet^+$ ba a marked word with $\beta \geq 1$ marked symbols. Let $(\mathcal{F}, \mathtt{t}, \mathtt{n})$ be an encoding of $\mathfrak{f}(\mathfrak{w})$. For every φ in PITL, $\mathfrak{w} \models_\bullet \varphi$ if and only if $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \tau_\beta(\varphi)$.

Proof. By induction on the structure of φ, with the standard induction hypothesis stating that the lemma holds for every strict subformula of φ. Let $\mathfrak{w} = \mathtt{a}_1 \dots \mathtt{a}_k \in \Sigma_\bullet^+$ with β marked symbols. According to Definition 3, let $\mathbb{M} = (\mathtt{n}_1, \dots, \mathtt{n}_k)$ be the tuple of main nodes of $(\mathcal{F}, \mathtt{t}, \mathtt{n})$.

base case: $\varphi = \mathtt{a}$. Then,

$$
\begin{array}{lll}
\mathfrak{w} \models_\bullet \mathtt{a} \quad \text{iff} \quad \mathtt{a}_1 = \mathtt{a} \text{ or } \mathtt{a}_1 = \overline{\mathtt{a}}, & & (\textit{by definition of } \models_\bullet) \\
\text{iff} \quad \mathtt{n}_1 \text{ encodes } 2\mathtt{a} \text{ or } 2\mathtt{a}-1, & & (\textit{by definition of } \mathcal{F} \textit{ and } \mathfrak{f}) \\
\text{iff} \quad (\mathcal{F}, \mathtt{t}, \mathtt{n}_1) \models \mathtt{1st}_{[2\mathtt{a}-1,2\mathtt{a}]}, & & (\textit{by definition of } \mathtt{1st}_{[2\mathtt{a}-1,2\mathtt{a}]}) \\
\text{iff} \quad (\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \langle \mathrm{U} \rangle \, \mathtt{1st}_{[2\mathtt{a}-1,2\mathtt{a}]}, & & (\textit{by semantics of } \langle \mathrm{U} \rangle \textit{ and def. of } \mathtt{n}_1) \\
\text{iff} \quad (\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \tau_\beta(\mathtt{a}). & & (\textit{by definition of } \tau_\beta)
\end{array}
$$

In the second to last step, we need to rely on the definition of encoding in order to perform the backward direction, i.e. derive $(\mathcal{F}, \mathtt{t}, \mathtt{n}_1) \models \mathtt{1st}_{[2\mathtt{a}-1,2\mathtt{a}]}$ from $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \langle \mathrm{U} \rangle \, \mathtt{1st}_{[2\mathtt{a}-1,2\mathtt{a}]}$. Indeed, recall that the first node in the main path, i.e. $\mathtt{n}_1$, is the only one satisfying $\mathtt{1st}_{[i,j]}$ (for some $1 \leq i < j \leq 2n$). As above, we write "*def. of* $\mathtt{n}_1$" when this property is used.

base case: $\varphi = \mathtt{1}$**.** Then,

$$
\begin{array}{lll}
\mathfrak{w} \models_\bullet \mathtt{1} \quad \text{iff} \quad \overline{\mathtt{a}} \in \overline{\Sigma} \text{ such that } \mathtt{a}_1 = \overline{\mathtt{a}}, & & (\textit{by definition of } \models_\bullet) \\
\text{iff} \quad \mathtt{n}_1 \text{ encodes } 2\mathtt{a}-1 \text{ for some } \mathtt{a} \in \Sigma, & & (\textit{by definition of } \mathcal{F} \textit{ and } \mathfrak{f}) \\
\text{iff} \quad (\mathcal{F}, \mathtt{t}, \mathtt{n}_1) \models \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma, & & (\textit{by definition of } \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma) \\
\text{iff} \quad (\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \langle \mathrm{U} \rangle (\mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma), & & (\textit{by semantics of } \langle \mathrm{U} \rangle \textit{ and def. of } \mathtt{n}_1) \\
\text{iff} \quad (\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \tau_\beta(\mathtt{1}). & & (\textit{by definition of } \tau_\beta)
\end{array}
$$

As in Lemma 23, the cases for Boolean connectives are obvious, so let us focus on $\varphi = \varphi_1 | \varphi_2$. Let $\mathfrak{w}' \in \Sigma^*$, $\overline{\mathtt{a}} \in \overline{\Sigma}$ and $\mathfrak{w}'' \in \Sigma_\bullet^*$ be such that $\Delta(\mathfrak{w}) = (\mathfrak{w}', \overline{\mathtt{a}}, \mathfrak{w}'')$. As $\overline{\mathtt{a}}$ is the leftmost marked symbol in $\mathfrak{w}$, notice that $\mathfrak{w}''$ contains $\beta - 1$ marked symbols.

induction step: $\varphi = \varphi_1 | \varphi_2$**.** We recall that $\mathfrak{w} \models_\bullet \varphi_1 | \varphi_2$ if and only if there is $\mathtt{b} \in \Sigma$ such that

$$
\begin{array}{lll}
 & \text{(a)} & \mathfrak{w}' = \epsilon, \, \mathtt{b} = \mathtt{a} \text{ and } \overline{\mathtt{a}} \, \mathfrak{w}'' \models_\bullet \varphi_1 \wedge \varphi_2 \\
\text{or} & \text{(b)} & \mathfrak{w}' = \mathtt{b}\mathfrak{w}_2 \text{ and } \overline{\mathtt{b}} \, \mathfrak{w}_2 \, \overline{\mathtt{a}} \, \mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \mathtt{b}\mathfrak{w}_2 \, \overline{\mathtt{a}} \, \mathfrak{w}'' \models_\bullet \varphi_2, \text{ for some } \mathfrak{w}_2 \in \Sigma^* \\
\text{or} & \text{(c)} & \mathfrak{w}' \neq \epsilon \text{ and } \mathtt{b} = \mathtt{a} \text{ and } \mathfrak{w}' \, \overline{\mathtt{a}} \, \mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \overline{\mathtt{a}} \, \mathfrak{w}'' \models_\bullet \varphi_2 \\
\text{or} & \text{(d)} & \mathfrak{w}' = \mathfrak{w}_1 \mathtt{b} \mathfrak{w}_2 \text{ and } \mathfrak{w}_1 \, \overline{\mathtt{b}} \, \mathfrak{w}_2 \, \overline{\mathtt{a}} \, \mathfrak{w}'' \models_\bullet \varphi_1 \text{ and } \mathtt{b}\mathfrak{w}_2 \, \overline{\mathtt{a}} \, \mathfrak{w}'' \models_\bullet \varphi_2, \\
 & & \text{for some } \mathfrak{w}_1 \in \Sigma^+ \text{ and } \mathfrak{w}_2 \in \Sigma^*.
\end{array}
$$

We split the proof into four double implications, making a correspondence between the four cases (a)–(d) in the semantics of $\mathfrak{w} \models_\bullet \varphi_1 | \varphi_2$ and the four disjuncts in the definition of $\tau_\beta(\varphi_1 | \varphi_2)$. More precisely, we prove:

A. there is $\mathtt{b} \in \Sigma$ such that $\mathfrak{w}' = \epsilon$, $\mathtt{b} = \mathtt{a}$ and $\overline{\mathtt{a}} \, \mathfrak{w}'' \models_\bullet \varphi_1 \wedge \varphi_2$, if and only if

$$(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \langle \mathrm{U} \rangle \big(\mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma \wedge \tau_\beta(\varphi_1) \wedge \tau_\beta(\varphi_2) \big),$$

B. there are $\mathtt{b} \in \Sigma$ and $\mathfrak{w}_2 \in \Sigma^*$ s.t. $\mathfrak{w}' = \mathtt{b}\mathfrak{w}_2$, $\overline{\mathtt{b}} \, \mathfrak{w}_2 \, \overline{\mathtt{a}} \, \mathfrak{w}'' \models \varphi_1$ and $\mathtt{b}\mathfrak{w}_2 \, \overline{\mathtt{a}} \, \mathfrak{w}'' \models \varphi_2$, iff

$$(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \langle \mathrm{U} \rangle \big(\mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \neg \mathtt{mark}_\Sigma \wedge \blacklozenge (\mathtt{mark}_\Sigma \wedge \tau_{\beta+1}(\varphi_1)) \wedge \tau_\beta(\varphi_2) \big),$$

C. there is $\mathtt{b} \in \Sigma$ such that $\mathfrak{w}' \neq \epsilon$, $\mathtt{b} = \mathtt{a}$, $\mathfrak{w}' \, \overline{\mathtt{a}} \, \mathfrak{w}'' \models \varphi_1$ and $\overline{\mathtt{a}} \, \mathfrak{w}'' \models \varphi_2$, if and only if

$$\begin{array}{c}(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \langle \mathrm{U} \rangle \big(\mathtt{symb} \wedge \neg \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma \wedge \#\mathtt{markAnc}_\Sigma \geq \beta - 1 \\ \wedge \, \tau_\beta(\varphi_1) \wedge \blacklozenge (\mathtt{1st}_{[1,2n]} \wedge \tau_\beta(\varphi_2)) \big),\end{array}$$

D. there are $\mathtt{b} \in \Sigma$, $\mathfrak{w}_1 \in \Sigma^+$ and $\mathfrak{w}_2 \in \Sigma^*$ such that $\mathfrak{w}' = \mathfrak{w}_1\mathtt{b}\mathfrak{w}_2$, $\mathfrak{w}_1\,\overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models \varphi_1$ and $\mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models \varphi_2$, if and only if

$$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathrm{U} \rangle \big(\mathtt{symb} \wedge \neg \mathtt{1st}_{[1,2n]} \wedge \neg \mathtt{mark}_{\Sigma} \wedge \#\mathtt{markAnc}_{\Sigma} \geq \beta \\ \wedge \blacklozenge(\mathtt{mark}_{\Sigma} \wedge \tau_{\beta+1}(\varphi_1)) \wedge \blacklozenge(\mathtt{1st}_{[1,2n]} \wedge \tau_{\beta}(\varphi_2))\big).$$

Proving these four correspondences suffices to prove the lemma. Indeed, the disjunction of the four ALT formulae in these four cases is equivalent to $\tau_\beta(\varphi_1 \,|\, \varphi_2)$ since the somewhere modality distributes over disjunction, i.e. $\langle \mathrm{U} \rangle(\varphi \vee \psi) \Leftrightarrow \langle \mathrm{U} \rangle \varphi \vee \langle \mathrm{U} \rangle \psi$, and conjunction distributes over disjunction. In the proofs below, we often use the fact that if $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ encodes $\mathfrak{f}(\mathsf{w})$, then for every node n', $(\mathcal{F}, \mathsf{t}, \mathsf{n}')$ encodes $\mathfrak{f}(\mathsf{w})$. This follows directly from the definition of encoding, which does not depend on the current world n. Below, we formalise the proof of (D). The (all quite similar) proofs of the other cases are relegated to Appendix B.

proof of (D): Lastly, for the fourth double implication,

there are $\mathtt{b} \in \Sigma$, $\mathfrak{w}_1 \in \Sigma^+$ and $\mathfrak{w}_2 \in \Sigma^*$ s.t. $\mathfrak{w}' = \mathfrak{w}_1\mathtt{b}\mathfrak{w}_2$, $\mathfrak{w}_1\,\overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models \varphi_1$ and $\mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models \varphi_2$,

$\Leftrightarrow$ there is $j \in [2, k]$ (recall $\mathfrak{w} = \mathtt{a}_1 \dots \mathtt{a}_k$ and $\Delta(\mathfrak{w}) = (\mathfrak{w}', \overline{\mathtt{a}}, \mathfrak{w}'')$) such that

1. the main node n_j encodes a non marked symbol and has exactly β ancestors in $\{\mathsf{n}_{j+1}, \dots, \mathsf{n}_k\}$ that encode marked symbols (i.e. it encodes a symbol of $\mathfrak{w}$ that strictly precedes all the β marked symbols in $\mathfrak{w}$). Equivalently, $(\mathcal{F}, \mathsf{t}, \mathsf{n}_j)$ satisfies $\mathtt{symb} \wedge \neg \mathtt{1st}_{[1,2n]} \wedge \neg \mathtt{mark}_{\Sigma} \wedge \#\mathtt{markAnc}_{\Sigma} \geq \beta$,
   (*by definition of $\mathcal{F}$, Lemma 24 and as $j > 1$*)
2. There is $\mathcal{F}' \subseteq \mathcal{F}$ s.t. $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$, $(\mathcal{F}', \mathsf{t}, \mathsf{n}_j) \models \mathtt{mark}_{\Sigma} \wedge \tau_{\beta+1}(\varphi_1)$.
   For this step, consider the finite forest $\mathcal{F}' \subseteq \mathcal{F}$ obtained from $\mathcal{F}$ by removing one character node of n_j. As n_j encodes the non marked symbol $\mathtt{b}$ w.r.t. $\mathcal{F}$, by definition it encodes the marked symbol $\overline{\mathtt{b}}$ w.r.t. $\mathcal{F}'$. Every other node has the same number of $\mathcal{F}'$-children as in $\mathcal{F}$. In other words, $\mathcal{F}'$ encodes the word $\mathfrak{w}_1\,\overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}''$, with $\beta + 1$ marked symbols, obtained from $\mathfrak{w}$ by marking the j-th symbol (which belongs to $\mathfrak{w}'$). By definition of $\mathcal{F}'$, $(\mathcal{F}', \mathsf{t}, \mathsf{n}_j) \models \mathtt{mark}_{\Sigma}$. By induction hypothesis, $(\mathcal{F}', \mathsf{t}, \mathsf{n}_j) \models \tau_{\beta+1}(\varphi_1)$.
3. There is $\mathcal{F}'' \subseteq \mathcal{F}$ s.t. $\mathsf{card}(\mathcal{F}'') = \mathsf{card}(\mathcal{F}) - 1$, $(\mathcal{F}'', \mathsf{t}, \mathsf{n}_j) \models \mathtt{1st}_{[1,2n]} \wedge \tau_{\beta}(\varphi_2)$.
   For this step, consider the subforest $\mathcal{F}'' \subseteq \mathcal{F}$ obtained from $\mathcal{F}$ by removing n_{j-1}, i.e. the only main node such that $\mathcal{F}(\mathsf{n}_{j-1}) = \mathsf{n}_j$ (which exists as $j > 1$). So, $\mathcal{F}''$ encodes the word $\mathtt{a}_j\mathtt{a}_{j+1} \dots \mathtt{a}_k = \mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}''$. By definition, $(\mathcal{F}'', \mathsf{t}, \mathsf{n}_j) \models \mathtt{1st}_{[1,2n]}$. By induction hypothesis, $(\mathcal{F}'', \mathsf{t}, \mathsf{n}_j) \models \tau_{\beta}(\varphi_2)$,

$\Leftrightarrow$ there is a main node n_j in the main path of $\mathcal{F}$ such that

$$(\mathcal{F}, \mathsf{t}, \mathsf{n}_j) \models \mathtt{symb} \wedge \neg \mathtt{1st}_{[1,2n]} \wedge \neg \mathtt{mark}_{\Sigma} \wedge \#\mathtt{markAnc}_{\Sigma} \geq \beta \\ \wedge \blacklozenge(\mathtt{mark}_{\Sigma} \wedge \tau_{\beta+1}(\varphi_1)) \wedge \blacklozenge(\mathtt{1st}_{[1,2n]} \wedge \tau_{\beta}(\varphi_2)).$$

(*by definition of $\mathcal{F}' \subseteq \mathcal{F}$, $\mathcal{F}'' \subseteq \mathcal{F}$ and $\blacklozenge$*)

$\Leftrightarrow$ $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathrm{U} \rangle \big(\mathtt{symb} \wedge \neg \mathtt{1st}_{[1,2n]} \wedge \neg \mathtt{mark}_{\Sigma} \wedge \#\mathtt{markAnc}_{\Sigma} \geq \beta \\ \wedge \blacklozenge(\mathtt{mark}_{\Sigma} \wedge \tau_{\beta+1}(\varphi_1)) \wedge \blacklozenge(\mathtt{1st}_{[1,2n]} \wedge \tau_{\beta}(\varphi_2))\big)$.

(*by semantics of $\langle \mathrm{U} \rangle$*) □

The reduction from the satisfiability problem of PITL on standard semantics follows as we are able to characterise the set of pointed forests encoding words in $\Sigma^* \overline{\Sigma}$ (first three conjuncts in the formula in the lemma below). To conclude, we simply apply Lemma 23 and Lemma 25.

Lemma 26. Every φ in PITL written with symbols from $\Sigma = [1, n]$ is satisfiable under the standard interpretation of PITL if and only if the following formula in ALT is satisfiable

$$\underbrace{\texttt{word}_{[1,2n]} \land \langle \mathrm{U} \rangle \texttt{Hit} \land [\mathrm{U}](\texttt{mark}_\Sigma \Leftrightarrow \texttt{Hit} \land \neg \blacklozenge(\texttt{Miss}))}_{\text{The forest encodes a non-empty word. The only child of the target node is the only node encoding a marked symbol.}} \land \mathcal{T}_1(\varphi).$$

Proof. ($\Rightarrow$): Suppose that φ is satisfiable, and let $\mathfrak{w} = \mathtt{a}_1 \dots \mathtt{a}_k \in \Sigma^+$ be a word satisfying it. By Lemma 23, the marked word $\overline{\mathfrak{w}} = \mathtt{a}_1 \dots \overline{\mathtt{a}_k} \in \Sigma^* \overline{\Sigma}$ satisfies φ with respect to the satisfaction relation $\models_\bullet$. Notice that $\overline{\mathfrak{w}}$ contains only one marked symbol. Let $(\mathcal{F}, \mathtt{t}, \mathtt{n})$ be a pointed forest encoding $\mathfrak{f}(\overline{\mathfrak{w}})$. By Lemma 25, $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \mathcal{T}_1(\varphi)$. Moreover, as $\mathfrak{w}$ is not empty we derive that $(\mathcal{F}, \mathtt{t}, \mathtt{n})$ satisfies $\langle \mathrm{U} \rangle\, \texttt{Hit}$, and by Lemma 9 it satisfies $\texttt{word}_{[1,2n]}$. As shown in Lemma 7, the formula $\texttt{Hit} \land \neg \blacklozenge(\texttt{Miss})$ is only satisfied when the current evaluation node corresponds to a child of $\mathtt{t}$. Instead, the formula $\texttt{mark}_\Sigma$ is only satisfied if the current node encodes a marked symbol (Lemma 24(I)). We then conclude that $(\mathcal{F}, \mathtt{t}, \mathtt{n})$ also satisfies $[\mathrm{U}](\texttt{mark}_\Sigma \Leftrightarrow (\texttt{Hit} \land \neg \blacklozenge(\texttt{Miss})))$. Indeed, $\overline{\mathtt{a}_k}$ is the only marked symbol of $\overline{\mathfrak{w}}$ and, by definition of encoding (Definition 3), it is encoded by the only $\mathcal{F}$-child of $\mathtt{t}$.
($\Leftarrow$): Suppose $\texttt{word}_{[1,2n]} \land \langle \mathrm{U} \rangle\, \texttt{Hit} \land [\mathrm{U}] \big(\texttt{mark}_\Sigma \Leftrightarrow (\texttt{Hit} \land \neg \blacklozenge(\texttt{Miss}))\big) \land \mathcal{T}_1(\varphi)$ satisfiable, and let $(\mathcal{F}, \mathtt{t}, \mathtt{n})$ be a pointed forest satisfying it. From the satisfaction of $\texttt{word}_{[1,2n]}$ and $\langle \mathrm{U} \rangle\, \texttt{Hit}$, by Lemma 9, $(\mathcal{F}, \mathtt{t}, \mathtt{n})$ is an encoding of a non-empty word in $[1, 2n]^+$. Let $\mathtt{b}_1 \dots \mathtt{b}_k$ be this word and let $\mathtt{n}_k$ be the node corresponding to $\mathtt{b}_k$. By definition of encoding, $\mathtt{n}_k$ is the only child of $\mathtt{t}$. Thus, from $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models [\mathrm{U}] \big(\texttt{mark}_\Sigma \Leftrightarrow (\texttt{Hit} \land \neg \blacklozenge(\texttt{Miss}))\big)$, together with Lemma 24(I) and Lemma 7, we conclude that $\mathtt{n}_k$ is the only node of $\mathsf{dom}(\mathcal{F})$ encoding a marked symbol. This means that $\mathtt{b}_1 \dots \mathtt{b}_k$ is of the form $\mathtt{a}_1 \dots \overline{\mathtt{a}_k} \in \Sigma^* \overline{\Sigma}$. From $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \mathcal{T}_1(\varphi)$ and by Lemma 25 $\mathtt{a}_1 \dots \overline{\mathtt{a}_k} \models_\bullet \varphi$. Lastly, $\mathtt{a}_1 \dots \mathtt{a}_k \models \varphi$ by Lemma 23. ◻

Because of the four disjuncts appearing in the formula $\mathcal{T}_\beta(\varphi \,|\, \psi)$, the translation is exponential in the number of symbols used to write the PITL formula. Since the satisfiability problem of PITL is Tower-hard (Proposition 19), any elementary translation suffices in order to conclude that the satisfiability problem of ALT is also Tower-hard, directly by Lemma 26. Decidability in Tower stems directly from the satisfiability problem of monadic second-order logic on tree-like structures [39], and it is formally reproved multiple times in the next section.

Theorem 27. The satisfiability problem of ALT is Tower-complete.

5. Revisiting Tower-hard Logics with ALT

Strong of Theorem 27, we now display the usefulness of ALT as a tool for proving the Tower-hardness of logics interpreted on tree-like structures. In particular, we provide semantically faithful reductions from SAT(ALT) to the satisfiability problem of four logics that were independently found to be Tower-complete: the two-variables fragment of the first-order separation logic $\mathsf{SL}(\exists, *)$ and its extension featuring the bounded separating implication $-\!\!\circledast_{[1]}$ [10, 9, 19], quantified CTL on trees [29], modal logic of heaps [19] and modal separation logic [20]. Thanks to the simplicity of ALT, our reductions only use strict fragments of these formalisms, allowing us to refine their non-elementary

$$
\begin{array}{lll}
(s,h) \models \mathtt{emp} & \text{iff} & \mathsf{dom}(h) = \varnothing \text{ (i.e. the heap is empty)}, \\
(s,h) \models \mathtt{x} = \mathtt{y} & \text{iff} & s(\mathtt{x}) = s(\mathtt{y}), \\
(s,h) \models \mathtt{x} \hookrightarrow \mathtt{y} & \text{iff} & h(s(\mathtt{x})) = s(\mathtt{y}), \\
(s,h) \models \varphi * \psi & \text{iff} & \text{there are } h_1 \text{ and } h_2 \text{ s.t. } h_1 + h_2 = h,\ (s,h_1) \models \varphi \text{ and } (s,h_2) \models \psi, \\
(s,h) \models \exists \mathtt{z}\, \varphi & \text{iff} & \text{there is } \ell \in \mathsf{LOC} \text{ such that } (s[\mathtt{z} \leftarrow \ell], h) \models \varphi.
\end{array}
$$

Figure 9: Satisfaction relation for $\mathsf{SL}(\exists, *, -\!\!*)$, with respect to a memory state (s,h).

boundaries. The technical machinery required to establish these reductions is very modest compared to the original Tower-hardness proofs of the aforementioned logics, as most of the heavy lifting has already been done when proving Theorem 27. Most notably, this section shed a new light on why all these logics are Tower-hard: it is because they fundamentally provide the reachability and submodel reasoning given by ALT.

5.1. From ALT *to fragments of First-Order Separation Logic*

Separation logic (SL) [40] is an assertion language used in state-of-the-art tools [6, 12] for Hoare-style verification of heap-manipulating programs. In this section, we briefly introduce the first-order separation logic $\mathsf{SL}(\exists, *)$ shown Tower-complete in [10], and rely on ALT to study some of its fragments and variants.

We write VAR and LOC for two countably infinite sets of program variables and (memory) locations, respectively. The formulae φ of the *first-order separation logic* $\mathsf{SL}(\exists, *)$ are built from the grammar below (where $\mathtt{x}, \mathtt{y}, \mathtt{z} \in \mathsf{VAR}$):

$$
\begin{array}{rlll}
\pi \coloneqq & \top & \text{(true)} \\
\mid & \mathtt{emp} & \text{(empty predicate)} \\
\mid & \mathtt{x} = \mathtt{y} & \text{(equality predicate)} \\
\mid & \mathtt{x} \hookrightarrow \mathtt{y} & \text{(points-to predicate)}
\end{array}
\qquad
\begin{array}{rlll}
\varphi \coloneqq & \pi & & \text{(atomic formulae)} \\
\mid & \varphi \wedge \varphi \mid \neg\varphi & & \text{(Boolean connectives)} \\
\mid & \varphi * \varphi & & \text{(separating conjunction)} \\
\mid & \exists \mathtt{z}\, \varphi & & \text{(first-order quantification)}
\end{array}
$$

Separation logic is interpreted on memory states.

Definition 28 (Memory State)**.** A *memory state* is a pair (s,h) consisting of a function (the *store*) $s : \mathsf{VAR} \to \mathsf{LOC}$ and a partial function with finite domain (the *heap*) $h : \mathsf{LOC} \to_{\mathsf{fin}} \mathsf{LOC}$.

Since $\mathcal{N}$ and LOC are both countably infinite sets, we assume w.l.o.g. that $\mathsf{LOC} = \mathcal{N}$. Given a program variable $\mathtt{z}$, a location $\ell \in \mathsf{LOC}$ and a store s, we write $s[\mathtt{z} \leftarrow \ell]$ for the store obtained from s by only changing the evaluation of $\mathtt{z}$ from $s(\mathtt{z})$ to ℓ. Two heaps h_1 and h_2 are said to be disjoint, written $h_1 \perp h_2$, whenever $\mathsf{dom}(h_1) \cap \mathsf{dom}(h_2) = \varnothing$, and when this holds the union $h_1 + h_2$ of h_1 and h_2 is defined as the sum of functions:

$$
(h_1 + h_2)(\ell) \stackrel{\text{def}}{=} \textbf{if } \ell \in \mathsf{dom}(h_1) \textbf{ then } h_1(\ell) \textbf{ else } h_2(\ell).
$$

Given a memory state (s,h), the satisfaction relation $\models$ for the formulae of $\mathsf{SL}(*)$ is given in Figure 9 (standard cases for $\top$ and Boolean connectives are omitted). As we can see, the predicate $\mathtt{emp}$ tests whether the heap h is empty. The formula $\mathtt{x} = \mathtt{y}$ simply states that, with respect to s, the

same location is assigned to both $\mathtt{x}$ and $\mathtt{y}$. The formula $\mathtt{x} \hookrightarrow \mathtt{y}$ goes one step further, and states that the location assigned to the variable $\mathtt{x}$ points to the location assigned to $\mathtt{y}$. The formula $\exists \mathtt{z}\, \varphi$ asks whether it is possible to update the location assigned to $\mathtt{z}$ in a way that satisfies the formula φ. Lastly, the formula $\varphi * \psi$ states that it is possible to split the heap h into two disjoint heaps h_1 and h_2 so that the memory state (s, h_1) satisfies φ, whereas the memory state (s, h_2) satisfies ψ. The *separating conjunction* $*$ is the main ingredient of separation logic, and it is powerful enough to capture the operators $\blacklozenge$ and $\blacklozenge^*$. We define $\blacklozenge_{\mathsf{SL}} \varphi \stackrel{\text{def}}{=} (\neg \mathtt{emp} \wedge \neg(\neg \mathtt{emp} * \neg \mathtt{emp})) * \varphi$ and $\blacklozenge^*_{\mathsf{SL}}\, \varphi \stackrel{\text{def}}{=} \top * \varphi$. Since the formula $\neg \mathtt{emp} \wedge \neg(\neg \mathtt{emp} * \neg \mathtt{emp})$ is satisfied by a memory state (s, h) whenever $\mathsf{card}(\mathsf{dom}(h)) = 1$, the semantics of $\blacklozenge_{\mathsf{SL}}$ and $\blacklozenge^*_{\mathsf{SL}}$ is related to the analogous operators of ALT as follows:

$$(s,h) \models \blacklozenge_{\mathsf{SL}} \varphi \quad \Leftrightarrow \quad \exists h_1, h_2 \text{ s.t. } h_1 \perp h_2,\ h_1 + h_2 = h,\ \mathsf{card}(\mathsf{dom}(h_1)) = 1 \text{ and } (s, h_2) \models \varphi.$$

$$(s,h) \models \blacklozenge^*_{\mathsf{SL}} \varphi \quad \Leftrightarrow \quad \exists h_1, h_2 \text{ s.t. } h_1 \perp h_2,\ h_1 + h_2 = h \text{ and } (s, h_2) \models \varphi.$$

We use several well-known abbreviations from the separation logic literature, such as the *alloc* predicate $\mathtt{x} \hookrightarrow \_$, the *reach-plus* predicate $\mathtt{x} \hookrightarrow^+ \mathtt{y}$ and the *list-segment* predicate $\mathtt{ls}(\mathtt{x}, \mathtt{y})$. The table below summarises these abbreviations. The correctness of these formulae with respect to their semantics can be traced back to [40] and [19].

Formula:	Definition:	Semantics w.r.t. (s, h):
$\mathtt{x} \hookrightarrow \_$	$\exists \mathtt{y}\, \mathtt{x} \hookrightarrow \mathtt{y}$	$s(\mathtt{x}) \in \mathsf{dom}(h)$
$\_ \hookrightarrow \mathtt{x}$	$\exists \mathtt{y}\, \mathtt{y} \hookrightarrow \mathtt{x}$	$s(\mathtt{x}) \in \mathsf{ran}(h)$
$\mathtt{x} \hookrightarrow^+ \mathtt{y}$	$\blacklozenge^*_{\mathsf{SL}} \Big(\mathtt{x} \hookrightarrow \_ \wedge (\_ \hookrightarrow \mathtt{x} \Rightarrow \mathtt{x} = \mathtt{y})$ $\wedge \forall \mathtt{x}' \forall \mathtt{y}' \left(\mathtt{y}' \hookrightarrow \mathtt{x}' \Rightarrow \neg \blacklozenge_{\mathsf{SL}} (\neg \mathtt{y}' \hookrightarrow \mathtt{x}' \wedge \_ \hookrightarrow \mathtt{x}') \right)$ $\wedge \forall \mathtt{x}' \left(\mathtt{x}' \neq \mathtt{y} \wedge \_ \hookrightarrow \mathtt{x}' \Rightarrow \mathtt{x}' \hookrightarrow \_ \right) \Big)$	$h^\delta(s(\mathtt{x})) = s(\mathtt{y})$ for some $\delta \geq 1$
$\mathtt{x} \hookrightarrow^* \mathtt{y}$	$\mathtt{x} = \mathtt{y} \vee \mathtt{x} \hookrightarrow^+ \mathtt{y}$	$h^\delta(s(\mathtt{x})) = s(\mathtt{y})$ for some $\delta \geq 0$
$\mathtt{ls}(\mathtt{x}, \mathtt{y})$	$\mathtt{x} \hookrightarrow^* \mathtt{y} \wedge \neg \blacklozenge_{\mathsf{SL}} (\mathtt{x} \hookrightarrow^* \mathtt{y})$	$h^\delta(s(\mathtt{x})) = s(\mathtt{y})$ iff $\delta = \mathsf{card}(\mathsf{dom}(h))$

In the definition of $\mathtt{x} \hookrightarrow^+ \mathtt{y}$, notice that the bounded variables $\mathtt{x}'$ and $\mathtt{y}'$ can be safely replaced by $\mathtt{x}$ and $\mathtt{y}$, respectively, obtaining an equivalent formula that only uses two variable names.

5.1.1. From ALT *to* $\mathsf{SL}([\exists]_1, *, \mathtt{x} \hookrightarrow \_, \hookrightarrow^+)$.

The two-variables fragment of $\mathsf{SL}(\exists, *)$, i.e. the sublogic of $\mathsf{SL}(\exists, *)$ where formulae are restricted to two variable names, has been shown TOWER-complete in [19]. We consider the fragment of this logic, denoted by $\mathsf{SL}([\exists]_1, *, \mathtt{x} \hookrightarrow \_, \hookrightarrow^+)$, featuring in particular only one free variable name $\mathtt{v}$ and one quantified variable $\mathtt{u}$. Formally, formulae φ of $\mathsf{SL}([\exists]_1, *, \mathtt{x} \hookrightarrow \_, \hookrightarrow^+)$ are given by the grammar below, where $\{\mathtt{x}, \mathtt{y}\} \subseteq \{\mathtt{u}, \mathtt{v}\} \subseteq \mathsf{VAR}$:

$$\varphi := \top \mid \varphi \wedge \varphi \mid \neg \varphi \mid \mathtt{emp} \mid \mathtt{x} = \mathtt{y} \mid \mathtt{x} \hookrightarrow \mathtt{y} \mid \mathtt{x} \hookrightarrow \_ \mid \mathtt{x} \hookrightarrow^+ \mathtt{y} \mid \varphi * \varphi \mid \exists \mathtt{u}\, \varphi.$$

$$\tau_{\mathtt{v}}(\mathtt{Hit}) \overset{\text{def}}{=} \mathtt{u}\hookrightarrow^{+}\mathtt{v}, \qquad \tau_{\mathtt{v}}(\blacklozenge\,\varphi) \overset{\text{def}}{=} \blacklozenge_{\text{SL}}\tau_{\mathtt{v}}(\varphi),$$
$$\tau_{\mathtt{v}}(\mathtt{Miss}) \overset{\text{def}}{=} \mathtt{u}\hookrightarrow\_\wedge\neg\tau_{\mathtt{v}}(\mathtt{Hit}), \qquad \tau_{\mathtt{v}}(\blacklozenge^{*}\,\varphi) \overset{\text{def}}{=} \blacklozenge^{*}_{\text{SL}}\,\tau_{\mathtt{v}}(\varphi),$$
$$\tau_{\mathtt{v}}(\langle \mathrm{U}\rangle\,\varphi) \overset{\text{def}}{=} \exists\mathtt{u}\ \tau_{\mathtt{v}}(\varphi),$$

Figure 10: Translation from ALT to $\mathsf{SL}([\exists]_1, *, \mathtt{x}\hookrightarrow\_, \hookrightarrow^{+})$.

As explained in [32], $\mathsf{SL}([\exists]_1, *, \mathtt{x}\hookrightarrow\_, \hookrightarrow^{+})$ is interesting from a verification point of view as it allows to express several properties of memory states such as acyclicity and garbage freedom. However, through ALT we show that this logic is already TOWER-complete.

The reduction from $\mathsf{SAT}(\mathsf{ALT})$ to $\mathsf{SAT}(\mathsf{SL}([\exists]_1, *, \mathtt{x}\hookrightarrow\_, \hookrightarrow^{+}))$ is straightforward. We rely on the location corresponding to $\mathtt{v}$ to encode the target node of a pointed forest, and on the location corresponding to $\mathtt{u}$ to encode the current node, which can be updated thanks to the first-order quantifier $\exists\mathtt{u}$. Given a formula φ in ALT, the resulting translation $\tau_{\mathtt{v}}(\varphi)$ is given in Figure 10. The figure omits the cases for $\top$ and Boolean connectives, which are defined homomorphically (e.g. $\tau_{\mathtt{v}}(\neg\varphi) \overset{\text{def}}{=} \neg\tau_{\mathtt{v}}(\varphi)$, as in the translation τ_{β} from PITL to ALT). Intuitively, the formula $\mathtt{Hit}$ is translated directly into $\mathtt{u}\hookrightarrow^{+}\mathtt{v}$, which encodes the fact that the current node represented by $\mathtt{u}$ reaches the target node in at least one step. Similarly, $\tau_{\mathtt{v}}(\mathtt{Miss})$ checks whether $\mathtt{u}$ belongs to the domain of the heap, but does not reach the target node. The correctness of the translation follows with a straightforward structural induction on φ (the proof can be found in Appendix C).

Lemma 29. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest, and let s be a store such that $s(\mathtt{v}) = \mathsf{t}$ and $s(\mathtt{u}) = \mathsf{n}$. $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$ if and only if $(s, \mathcal{F}) \models \tau_{\mathtt{v}}(\varphi)$.

Lemma 29, together with the fact that the formula $\forall\mathtt{u}\,\neg(\mathtt{u}\hookrightarrow^{+}\mathtt{u})$ characterises the class of acyclic heaps (which correspond to the finite forests of ALT), directly implies the following result.

Lemma 30. φ in ALT and $\tau_{\mathtt{v}}(\varphi) \wedge \forall\mathtt{u}\ \neg(\mathtt{u}\hookrightarrow^{+}\mathtt{u})$ in $\mathsf{SL}([\exists]_1, *, \mathtt{x}\hookrightarrow\_, \hookrightarrow^{+})$ are equisatisfiable.

This lemma reproves that the two-variables fragment of $\mathsf{SL}(\exists, *)$ is TOWER-hard. As this logic is TOWER-complete [19], it also shows that ALT is decidable in TOWER, as anticipated in Theorem 27. Moreover, the translation does not use $\mathtt{emp}$ and only uses the separating conjunction $*$ in order to express $\blacklozenge_{\text{SL}}$ and $\blacklozenge^{*}_{\text{SL}}$, which allows us to conclude the following result.

Theorem 31. $\mathsf{SAT}(\mathsf{SL}([\exists]_1, *, \mathtt{x}\hookrightarrow\_, \hookrightarrow^{+}))$ and the satisfiability problem of the two-variables fragment of $\mathsf{SL}(\exists, *)$ are both TOWER-complete, even when $\mathtt{emp}$ and $*$ are replaced by $\blacklozenge_{\text{SL}}$ and $\blacklozenge^{*}_{\text{SL}}$.

5.1.2. From ALT *to quantifier-free* SL(∗) *with bounded separating implication*

In [10] it is shown that $\mathsf{SL}(\exists, *)$ is still TOWER-complete when enriched with the so-called *bounded separating implication* $\varphi \mathrel{-\!\!\circledast^{[n]}} \psi$, where $n \in \mathbb{N}$, having the following semantics:

$$(s, h) \models \varphi \mathrel{-\!\!\circledast^{[n]}} \psi \text{ iff for every heap } h', \text{ if } h' \perp h,\ \mathsf{card}(h') \leq n \text{ and } (s, h') \models \varphi, \text{ then } (s, h + h') \models \psi.$$

Essentially, $\varphi \mathrel{-\!\!\circledast^{[n]}} \psi$ is satisfied whenever adding to the current heap h a heap h' of size at most n and such that $(s, h') \models \varphi$, we obtain a memory state $(s, h + h')$ satisfying ψ. Since we have shown that ALT is a fragment of the separation logic $\mathsf{SL}([\exists]_1, *, \mathtt{x}\hookrightarrow\_, \hookrightarrow^{+})$, and from [22] it is known that the separating implication can sometimes be used to mimic first-order quantifications, it is quite

natural to ask ourselves whether we can modify the TOWER-hardness of $\mathsf{SL}([\exists]_1, *, \mathtt{x} \hookrightarrow \_, \hookrightarrow^+)$ so that it uses the bounded separating implication instead of the first-order quantification. We answer this question by showing the following result.

Theorem 32. Satisfiability of the two-variable fragment of $\mathsf{SL}(*, \multimap[1], \mathtt{ls})$ is TOWER-complete.

The grammar of the formulae φ in $\mathsf{SL}(*, \multimap[1], \mathtt{ls})$ is given below:

$$\varphi \coloneqq \top \mid \mathtt{emp} \mid \mathtt{x} = \mathtt{y} \mid \mathtt{x} \hookrightarrow \mathtt{y} \mid \mathtt{ls}(\mathtt{x}, \mathtt{y}) \mid \varphi \wedge \varphi \mid \neg\varphi \mid \varphi * \varphi \mid \varphi \multimap[1]\, \varphi.$$

This logic features the *list-segment* predicate $\mathtt{ls}(\mathtt{x}, \mathtt{y})$ from [40, 16] (see the table of abbreviations in page 29, where it is defined in terms of $\mathtt{x} \hookrightarrow^+ \mathtt{y}$), which states that the heap is a linear structure going from the location corresponding to $\mathtt{x}$ to the one corresponding to $\mathtt{y}$. Moreover, notice that the logic is quantifier-free, and the operator $\varphi \multimap[n]\, \psi$ is restricted to $n = 1$, so that

$$(s,h) \models \varphi \multimap[1]\, \psi \text{ iff for every heap } h', \text{ if } h' \perp h, \mathsf{card}(h') \le 1 \text{ and } (s,h') \models \varphi, \text{ then } (s, h+h') \models \psi.$$

Besides, the logic can still express the *alloc* predicate $\mathtt{x} \hookrightarrow \_$, defined as $\mathtt{x} \hookrightarrow \mathtt{x} \multimap[1] \perp$, as well as the *reach* predicate $\mathtt{x} \hookrightarrow^* \mathtt{y}$, defined as $\mathtt{x} = \mathtt{y} \vee (\top * \mathtt{ls}(\mathtt{x}, \mathtt{y}))$. We introduce the formula $\mathtt{size} \geq \beta$ that is satisfied whenever the domain of the heap contains at least β locations. We have $\mathtt{size} \geq 0 \stackrel{\mathsf{def}}{=} \top$ and $\mathtt{size} \geq \beta + 1 \stackrel{\mathsf{def}}{=} \blacklozenge_{\text{SL}} \mathtt{size} \geq \beta$. We write $\mathtt{size} = \beta$ for $\mathtt{size} \geq \beta \wedge \neg \mathtt{size} \geq \beta + 1$. Lastly, we define the *bounded septraction* $\multimap\langle 1 \rangle$ as the right dual of the bounded separating implication $\multimap[1]$, i.e. $\varphi \multimap\langle 1 \rangle\, \psi \stackrel{\mathsf{def}}{=} \neg(\varphi \multimap[1]\, \neg\psi)$. Its semantics is as follows:

$$(s,h) \models \varphi \multimap\langle 1 \rangle\, \psi \text{ iff there is a heap } h' \text{ such that } h' \perp h, \mathsf{card}(h') \le 1, (s,h') \models \varphi \text{ and } (s, h+h') \models \psi.$$

Let us discuss how to encode a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ as a memory state (s,h). We use the location assigned to a fixed variable $\mathtt{x}$ in order to mimic the first-order quantification of $\langle \mathrm{U} \rangle\, \varphi$, by modifying the location pointed by $s(\mathtt{x})$. So, in the encoding, the location $h(s(\mathtt{x}))$ corresponds to the current node n. In order to mimic the quantification correctly, this location must be different from $s(\mathtt{x})$. Similarly to the reduction of the previous section, to encode the target node we rely on the location assigned to a second program variable $\mathtt{y}$, and require that both $s(\mathtt{y}) = \mathsf{t} \neq s(\mathtt{x})$ and $s(\mathtt{y}) \notin \mathsf{dom}(h)$ hold. This last condition is without loss of generality, as we can assume that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$ by Lemma 17(I). This leads to a very simple translation from ALT to $\mathsf{SL}(*, \multimap, \mathtt{ls})$, where for instance $\mathsf{n} = h(s(\mathtt{x}))$ is a hit node whenever the formula $\mathtt{x} \hookrightarrow^* \mathtt{y} \wedge \neg \mathtt{x} \hookrightarrow \mathtt{y}$ is satsisfied.

The encoding of a forest as a heap we just sketched is formalised as follows.

Definition 33 (Forests as heaps)**.** (s,h) is an $(\mathtt{x}, \mathtt{y})$-*encoding* of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, where $\mathtt{x}, \mathtt{y} \in \mathsf{VAR}$, iff

1. $h = \mathcal{F} + \{(s(\mathtt{x}), \mathsf{n})\}$ (seeing $\mathcal{F}$ as a heap), 2. $\mathsf{n} \neq s(\mathtt{x})$, 3. $s(\mathtt{y}) = \mathsf{t} \notin \mathsf{dom}(h)$.

With respect to memory states that are $(\mathtt{x}, \mathtt{y})$-encodings of some pointed forest, given a formula φ in ALT we translate it into a formula $\tau_{\mathtt{x},\mathtt{y}}(\varphi)$ in $\mathsf{SL}(*, \multimap[1], \mathtt{ls})$ following the definition in Figure 11. The figure omits the homomorphic cases for $\top$ and Boolean connectives. The translation of $\blacklozenge$ and $\blacklozenge^*$ uses the analogous formulae $\blacklozenge_{\text{SL}}$ and $\blacklozenge^*_{\text{SL}}$, while taking care that the location assigned to $\mathtt{x}$ is not discharged from the domain of the heap. The translation of $\langle \mathrm{U} \rangle\, \varphi$ essentially encodes a first-order quantification by modifying the location pointed by $s(\mathtt{x})$: the pair $(s(\mathtt{x}), h(s(\mathtt{x})))$ in the heap h is replaced with some $(s(\mathtt{x}), \ell)$, leading to the satisfaction of $\tau_{\mathtt{x},\mathtt{y}}(\varphi)$. While the translation of $\mathtt{Hit}$ is also quite straightforward, the translation of $\mathtt{Miss}$ requires some work. In particular, this predicate must be checked on $h(s(\mathtt{x}))$ and should hold only if this location is in the domain of the heap but

$$\mathcal{T}_{\mathtt{x},\mathtt{y}}(\mathtt{Hit}) \stackrel{\text{def}}{=} \mathtt{x} \hookrightarrow^* \mathtt{y} \wedge \neg \mathtt{x} \hookrightarrow \mathtt{y},$$

$$\mathcal{T}_{\mathtt{x},\mathtt{y}}(\mathtt{Miss}) \stackrel{\text{def}}{=} \neg\mathcal{T}_{\mathtt{x},\mathtt{y}}(\mathtt{Hit}) \wedge \neg \mathtt{x} \hookrightarrow \mathtt{y} \wedge \big(\mathtt{size} = 1 \mathrel{-\!\!*}_{[1]} \neg(\top * (\mathtt{ls}(\mathtt{x},\mathtt{y}) \wedge \mathtt{size} = 2))\big),$$

$$\mathcal{T}_{\mathtt{x},\mathtt{y}}(\blacklozenge\, \varphi) \stackrel{\text{def}}{=} \blacklozenge_{\text{SL}}(\mathtt{x} \hookrightarrow \_ \wedge \mathcal{T}_{\mathtt{x},\mathtt{y}}(\varphi)),$$

$$\mathcal{T}_{\mathtt{x},\mathtt{y}}(\blacklozenge^*\, \varphi) \stackrel{\text{def}}{=} \blacklozenge^*_{\text{SL}}(\mathtt{x} \hookrightarrow \_ \wedge \mathcal{T}_{\mathtt{x},\mathtt{y}}(\varphi)),$$

$$\mathcal{T}_{\mathtt{x},\mathtt{y}}(\langle \mathrm{U} \rangle\, \varphi) \stackrel{\text{def}}{=} (\mathtt{size} = 1 \wedge \mathtt{x} \hookrightarrow \_) * \big(\mathtt{size} = 1 \mathrel{-\!\!\circledast}_{\langle 1 \rangle} (\mathtt{x} \hookrightarrow \_ \wedge \neg \mathtt{x} \hookrightarrow \mathtt{x} \wedge \mathcal{T}_{\mathtt{x},\mathtt{y}}(\varphi))\big).$$

Figure 11: Translation from ALT to SL($*, -\!\!*, \mathtt{ls}$) with bounded magic wand.

does not reach $s(\mathtt{y})$ in at least one step. The formula $\mathcal{T}_{\mathtt{x},\mathtt{y}}(\mathtt{Miss})$ achieves this by stating that it is not possible to add one arrow to the heap in order to construct a path of length two going from $s(\mathtt{x})$ to $s(\mathtt{y})$. Indeed, under the hypothesis that $h(s(\mathtt{x}))$ is not in the domain of the heap, such a path can always be constructed by adding $\{(h(s(\mathtt{x})), s(\mathtt{y}))\}$ to the heap. So, $h(s(\mathtt{x}))$ must be in $\mathsf{dom}(h)$ which, together with $\neg\mathcal{T}_{\mathtt{x},\mathtt{y}}(\mathtt{Hit})$, effectively captures the semantics of $\mathtt{Miss}$. The correctness of the translation is formalised below.

Lemma 34. Let (s, h) be an $(\mathtt{x}, \mathtt{y})$-encoding of a pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$. Let φ be a formula in ALT. We have, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$ if and only if $(s, h) \models \mathcal{T}_{\mathtt{x},\mathtt{y}}(\varphi)$.

Proof. The proof is by structural induction on φ. The base case for $\varphi = \top$ is obvious.

base case: $\varphi = \mathtt{Hit}$.

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{Hit}$,

$\Leftrightarrow$ there is $\delta \geq 1$ such that $\mathcal{F}^\delta(\mathsf{n}) = \mathsf{t}$,

(*by definition of* $\models$)

$\Leftrightarrow$ there is $\delta \geq 2$ such that $h^\delta(s(\mathtt{x})) = \mathsf{t} = s(\mathtt{y})$ and for $\delta' < \delta$, $h^{\delta'}(s(\mathtt{x})) \neq s(\mathtt{y})$,

($\mathsf{n} \neq \mathsf{t}$ *and by definition of encoding, i.e.* $h(s(\mathtt{x})) = \mathsf{n}$, $s(\mathtt{y}) = \mathsf{t}$, $s(\mathtt{x}) \neq s(\mathtt{y})$)

$\Leftrightarrow$ there is $\delta \geq 0$ such that $h^\delta(s(\mathtt{x})) = s(\mathtt{y})$ and $h(s(\mathtt{x})) \neq s(\mathtt{y})$,

(*left-to-right direction: weakening. right-to-left direction: definition of encoding*)

$\Leftrightarrow$ $(s, h) \models \mathtt{x} \hookrightarrow^* \mathtt{y} \wedge \neg \mathtt{x} \hookrightarrow \mathtt{y}$.

(*definition of* $\models$)

Before treating the base case for $\varphi = \mathtt{Miss}$, let us prove the following intermediate result:

NDom. Let (s, h) be a memory state s.t. $s(\mathtt{x}) \in \mathsf{dom}(h)$ and for all $\delta \geq 0$, $h^\delta(s(\mathtt{x})) \neq s(\mathtt{y})$. $h(s(\mathtt{x})) \notin \mathsf{dom}(h)$ if and only if $(s, h) \models \mathtt{size} = 1 \mathrel{-\!\!\circledast}_{\langle 1 \rangle} (\top * (\mathtt{ls}(\mathtt{x},\mathtt{y}) \wedge \mathtt{size} = 2))$.

Proof of (NDom). ($\Rightarrow$): Suppose $h(s(\mathtt{x})) \notin \mathsf{dom}(h)$. Let $h' = \{(h(s(\mathtt{x})), s(\mathtt{y}))\}$. So, $h' \perp h$ and (s, h') satisfies $\mathtt{size} = 1$. From the two hypotheses $s(\mathtt{x}) \in \mathsf{dom}(h)$ and for all $\delta \geq 0$, $h^\delta(s(\mathtt{x})) \neq s(\mathtt{y})$, we conclude that there is a location ℓ such that $h'' \stackrel{\text{def}}{=} \{(s(\mathtt{x}), \ell), (\ell, s(\mathtt{y}))\} \subseteq h + h'$, where $s(\mathtt{x}) \neq s(\mathtt{y})$ and $\ell \neq s(\mathtt{y})$. Therefore, $(s, h'') \models \mathtt{ls}(\mathtt{x},\mathtt{y}) \wedge \mathtt{size} = 2$, which in turn implies that $(s, h + h')$ satisfies $\top * (\mathtt{ls}(\mathtt{x},\mathtt{y}) \wedge \mathtt{size} = 2)$. By definition of h', $(s, h) \models \mathtt{size} = 1 \mathrel{-\!\!\circledast}_{\langle 1 \rangle} (\top * (\mathtt{ls}(\mathtt{x},\mathtt{y}) \wedge \mathtt{size} = 2))$.

($\Leftarrow$): Suppose $(s,h) \models \mathtt{size} = 1 \mathrel{-\!\!\langle 1 \rangle} (\top * (\mathtt{ls}(\mathtt{x},\mathtt{y}) \wedge \mathtt{size} = 2))$. There is a heap h' with $\mathsf{card}(h') = 1$ and a heap $h'' \subseteq h + h'$ such that $(s,h'') \models \mathtt{ls}(\mathtt{x},\mathtt{y}) \wedge \mathtt{size} = 2$. So, $h'' = \{(s(\mathtt{x}),\ell),(\ell,s(\mathtt{y}))\}$ for some location $\ell \neq s(\mathtt{y})$. From the hypothesis $s(\mathtt{x}) \in \mathsf{dom}(h)$ and for all $\delta \geq 0$, $h^{\delta}(s(\mathtt{x})) \neq s(\mathtt{y})$, we derive $\ell \in \mathsf{dom}(h')$ and $h'(\ell) = s(\mathtt{y})$, and thus $h(s(\mathtt{x})) \notin \mathsf{dom}(h)$. (*End of Proof of* (NDom))

base case: $\varphi = \mathtt{Miss}$**.** ($\Rightarrow$): Suppose $(\mathcal{F},\mathtt{t},\mathtt{n}) \models \mathtt{Miss}$. So, $\mathtt{n} \in \mathsf{dom}(\mathcal{F})$ and $\mathtt{n}$ is not a $\mathcal{F}$-descendant of $\mathtt{t}$. By definition of encoding, $h(s(\mathtt{x})) = \mathtt{n} \neq s(\mathtt{x})$ and for all $\delta \geq 0$, $h^{\delta}(s(\mathtt{x})) \neq s(\mathtt{y}) = \mathtt{t}$. Therefore, $(s,h) \not\models \mathtt{x} \hookrightarrow \mathtt{y}$ and $h(s(\mathtt{x})) \in \mathsf{dom}(\mathcal{F})$. From (NDom), this implies $(s,h) \not\models \mathtt{size} = 1 \mathrel{-\!\!\langle 1 \rangle} (\top * (\mathtt{ls}(\mathtt{x},\mathtt{y}) \wedge \mathtt{size} = 2))$, whereas from the previous base case we conclude that $(s,h) \not\models \tau_{\mathtt{x},\mathtt{y}}(\mathtt{Hit})$.

($\Leftarrow$): Suppose $(s,h) \models \neg\tau_{\mathtt{x},\mathtt{y}}(\mathtt{Hit}) \wedge \neg\mathtt{x} \hookrightarrow \mathtt{y} \wedge (\mathtt{size} = 1 \mathrel{-\!\!\langle 1 \rangle} \neg(\top * (\mathtt{ls}(\mathtt{x},\mathtt{y}) \wedge \mathtt{size} = 2)))$. Recalling that $\tau_{\mathtt{x},\mathtt{y}}(\mathtt{Hit}) = \mathtt{x} \hookrightarrow^{*} \mathtt{y} \wedge \neg\mathtt{x} \hookrightarrow \mathtt{y}$, the satisfaction of $\neg\tau_{\mathtt{x},\mathtt{y}}(\mathtt{Hit}) \wedge \neg\mathtt{x} \hookrightarrow \mathtt{y}$ implies that for every $\delta \geq 0$, $h^{\delta}(s(\mathtt{x})) = s(\mathtt{y})$. Since by definition of encoding $s(\mathtt{x}) \in \mathsf{dom}(h)$, by (NDom) we conclude that $h(s(\mathtt{x})) \in \mathsf{dom}(h)$. So, $\mathtt{n} = h(s(\mathtt{x})) \in \mathsf{dom}(\mathcal{F})$. From the previous base case, $(\mathcal{F},\mathtt{t},\mathtt{n}) \not\models \mathtt{Hit}$. Thus, $(\mathcal{F},\mathtt{t},\mathtt{n}) \models \mathtt{Miss}$.

The induction steps for Boolean connectives are obvious. The cases for $\varphi = \blacklozenge\,\psi$ and $\varphi = \blacklozenge^{*}\,\psi$ are very similar. So, we just explicit the case for $\varphi = \blacklozenge\,\psi$.

induction step: $\varphi = \blacklozenge\,\psi$**.** ($\Rightarrow$): Suppose $(\mathcal{F},\mathtt{t},\mathtt{n}) \models \blacklozenge\,\psi$, and so there is a subforest $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$ and $(\mathcal{F}',\mathtt{t},\mathtt{n}) \models \psi$. Consider the heap $h' \stackrel{\text{def}}{=} \mathcal{F}' + \{s(\mathtt{x}) \mapsto \mathtt{n}\}$. Since $\mathtt{n} \neq s(\mathtt{x})$ and $s(\mathtt{y}) = \mathtt{t} \notin \mathsf{dom}(h)$ (from the fact that (s,h) is a $(\mathtt{x},\mathtt{y})$-encoding of $(\mathcal{F},\mathtt{t},\mathtt{n})$), we conlude that (s,h') is a $(\mathtt{x},\mathtt{y})$-encoding of $(\mathcal{F}',\mathtt{t},\mathtt{n})$. By induction hypothesis $(s,h') \models \tau_{\mathtt{x},\mathtt{y}}(\psi)$. By definition of h', $(s,h') \models \mathtt{x} \hookrightarrow \_$, $h' \subseteq h$ and $\mathsf{card}(h') = \mathsf{card}(h) - 1$. Thus, $(s,h) \models \tau_{\mathtt{x},\mathtt{y}}(\blacklozenge\,\psi)$.

($\Leftarrow$): Suppose $(s,h) \models \blacklozenge_{\text{SL}}(\mathtt{x} \hookrightarrow \_ \wedge \tau_{\mathtt{x},\mathtt{y}}(\psi))$. There is a subheap $h' \subseteq h$ such that $\mathsf{card}(h') = \mathsf{card}(h) - 1$ and $(s,h') \models \mathtt{x} \hookrightarrow \_ \wedge \tau_{\mathtt{x},\mathtt{y}}(\psi)$. Thus, together with the fact that $h = \mathcal{F} + \{s(\mathtt{x}) \mapsto \mathtt{n}\}$, we conclude that there is $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$ and $h' = \mathcal{F}' + \{s(\mathtt{x}) \mapsto \mathtt{n}\}$. So, (s,h') is a $(\mathtt{x},\mathtt{y})$-encoding of $(\mathcal{F}',\mathtt{t},\mathtt{n})$. By induction hypothesis, $(\mathcal{F}',\mathtt{t},\mathtt{n}) \models \psi$. From the semantics of the modality $\blacklozenge$, $(\mathcal{F},\mathtt{t},\mathtt{n}) \models \blacklozenge\,\psi$.

induction step: $\varphi = \langle \mathrm{U} \rangle\,\psi$**.** ($\Rightarrow$): Recall that (s,h) is a $(\mathtt{x},\mathtt{y})$-encoding of $(\mathcal{F},\mathtt{t},\mathtt{n})$, and so in particular $s(\mathtt{x}) \notin \mathsf{dom}(\mathcal{F})$ and $s(\mathtt{y}) = \mathtt{t} \notin \mathsf{dom}(h)$. Suppose $(\mathcal{F},\mathtt{t},\mathtt{n}) \models \langle \mathrm{U} \rangle\,\psi$. This implies that there is $\mathtt{n}' \in \mathcal{N}$ such that $(\mathcal{F},\mathtt{t},\mathtt{n}') \models \psi$. By Lemma 17(II) together with the fact that $s(\mathtt{x}) \notin \mathsf{dom}(\mathcal{F})$, $\mathtt{n}'$ can always be picked to be different from $s(\mathtt{x})$. Consider the heap $h' \stackrel{\text{def}}{=} \mathcal{F} + \{s(\mathtt{x}) \mapsto \mathtt{n}'\}$. As $\mathtt{n}' \neq s(\mathtt{x})$ and $s(\mathtt{y}) = \mathtt{t} \notin \mathsf{dom}(h)$, it holds that (s,h') is a $(\mathtt{x},\mathtt{y})$-encoding of $(\mathcal{F},\mathtt{t},\mathtt{n}')$. Clearly, $(s,h') \models \mathtt{x} \hookrightarrow \_ \wedge \neg\mathtt{x} \hookrightarrow \mathtt{x}$, and by induction hypothesis $(s,h') \models \tau_{\mathtt{x},\mathtt{y}}(\psi)$. By definition of h' and the semantics of $\mathrel{-\!\!\langle 1 \rangle}$, $(s,\mathcal{F}) \models \mathtt{size} = 1 \mathrel{-\!\!\langle 1 \rangle} (\mathtt{x} \hookrightarrow \_ \wedge \neg\mathtt{x} \hookrightarrow \mathtt{x} \wedge \tau_{\mathtt{x},\mathtt{y}}(\psi))$. Lastly, from $h = \mathcal{F} + \{(s(\mathtt{x}),\mathtt{n})\}$, we conclude that $(s,h) \models \tau_{\mathtt{x},\mathtt{y}}(\psi)$.

($\Leftarrow$): Suppose $(s,h) \models \tau_{\mathtt{x},\mathtt{y}}(\psi)$. So, there are two disjoint heaps h_1 and h_2 s.t. $h = h_1 + h_2$, $(s,h_1) \models \mathtt{size} = 1 \wedge \mathtt{x} \hookrightarrow \_$ and $(s,h_2) \models \mathtt{size} = 1 \mathrel{-\!\!\langle 1 \rangle} (\mathtt{x} \hookrightarrow \_ \wedge \neg\mathtt{x} \hookrightarrow \mathtt{x} \wedge \tau_{\mathtt{x},\mathtt{y}}(\psi))$. By definition of encoding $h = \mathcal{F} + \{(s(\mathtt{x}),\mathtt{n})\}$, which allows us to conclude that $h_1 = \{(s(\mathtt{x}),\mathtt{n})\}$ and $h_2 = \mathcal{F}$. So, there is a heap h' disjoint from $\mathcal{F}$ and such that $(s,h') \models \mathtt{size} = 1$ and $(s,\mathcal{F} + h') \models \mathtt{x} \hookrightarrow \_ \wedge \neg\mathtt{x} \hookrightarrow \mathtt{x} \wedge \tau_{\mathtt{x},\mathtt{y}}(\psi)$. As $(s,\mathcal{F} + h') \models \mathtt{x} \hookrightarrow \_ \wedge \neg\mathtt{x} \hookrightarrow \mathtt{x}$ and $s(\mathtt{x}) \notin \mathsf{dom}(\mathcal{F})$, there must be a node $\mathtt{n}' \neq s(\mathtt{x})$ such that $h' = \{s(\mathtt{x}) \mapsto \mathtt{n}'\}$. By definition of encoding, $(s,\mathcal{F} + h')$ is an $(\mathtt{x},\mathtt{y})$-encoding of $(\mathcal{F},\mathtt{t},\mathtt{n}')$. By induction hypothesis, $(\mathcal{F},\mathtt{t},\mathtt{n}') \models \psi$, which implies $(\mathcal{F},\mathtt{t},\mathtt{n}) \models \langle \mathrm{U} \rangle\,\psi$ by semantics of the modality $\langle \mathrm{U} \rangle$. □

The formula $\mathtt{x} \hookrightarrow \_ \wedge \neg \mathtt{x} \hookrightarrow \mathtt{x} \wedge \neg \mathtt{y} \hookrightarrow \_$ characterises the set of memory states encoding pointed forests, which allows us to show the lemma below and conclude the reduction. For the proof of this last lemma we also rely on Lemma 17(I) in order to only consider pointed forests that admit an encoding. Lemma 35 implies Theorem 32.

Lemma 35. φ in ALT is satisfiable iff so is $\mathtt{x} \hookrightarrow \_ \wedge \neg \mathtt{x} \hookrightarrow \mathtt{x} \wedge \neg \mathtt{y} \hookrightarrow \_ \wedge \mathcal{T}_{\mathtt{x},\mathtt{y}}(\varphi)$ in $\mathsf{SL}(*, \multimap^{[1]}, \mathtt{ls})$.

Proof. ($\Rightarrow$): Suppose φ satisfiable, and let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest satisfying it. Thanks to Lemma 17(I), without loss of generality we can assume that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$. Let us consider a memory state (s, h) such that $h = \mathcal{F} + \{s(\mathtt{x}) \mapsto \mathsf{n}\}$, $\mathsf{n} \neq s(\mathtt{x})$ and $s(\mathtt{y}) = \mathsf{t} \notin \mathsf{dom}(h)$. By Definition 33, this memory state is a $(\mathtt{x}, \mathtt{y})$-encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$. The three (dis)equalities characterising (s, h) directly imply that $(s, h) \models \mathtt{x} \hookrightarrow \_ \wedge \neg \mathtt{x} \hookrightarrow \mathtt{x} \wedge \neg \mathtt{y} \hookrightarrow \_$. By Lemma 34, $(s, h) \models \mathcal{T}_{\mathtt{x},\mathtt{y}}(\varphi)$.
($\Leftarrow$): Suppose $\mathtt{x} \hookrightarrow \_ \wedge \neg \mathtt{x} \hookrightarrow \mathtt{x} \wedge \neg \mathtt{y} \hookrightarrow \_ \wedge \mathcal{T}_{\mathtt{x},\mathtt{y}}(\varphi)$ satisfiable, and let (s, h) be a memory state satisfying it. From the satisfaction of $\mathtt{x} \hookrightarrow \_$, there is a location ℓ such that $h(s(\mathtt{x})) = \ell$. We consider the pointed forest $(\mathcal{F}, s(\mathtt{y}), \ell)$ where $\mathcal{F} = h \setminus \{s(\mathtt{x}) \mapsto h(s(\mathtt{x}))\}$. From $(s, h) \not\models \mathtt{x} \hookrightarrow \mathtt{x}$ we have $s(\mathtt{x}) \neq \ell$. From $(s, h) \not\models \mathtt{y} \hookrightarrow \_$ we have $s(\mathtt{y}) \notin \mathsf{dom}(h)$. Thus, by Definition 33 (s, h) is a $(\mathtt{x}, \mathtt{y})$-encoding of $(\mathcal{F}, s(\mathtt{y}), \ell)$. By Lemma 34, $(\mathcal{F}, s(\mathtt{y}), \ell) \models \varphi$. ☐

Interestingly, Theorem 32 already holds when the bounded separating implication is limited to formulae of the form $\mathtt{size} = 1 \multimap^{[1]} \varphi$. Indeed, the alloc formula $\mathtt{x} \hookrightarrow \_$ can be substituted with the equivalent formula $\mathtt{x} \hookrightarrow \mathtt{x} \vee (\mathtt{size} = 1 \multimap^{[1]} \neg \mathtt{x} \hookrightarrow \mathtt{x})$, so that the translation only uses the bounded separating implication in this way.

Corollary 36. $\mathsf{SL}(*, \multimap^{[1]}, \mathtt{ls})$ where $\multimap^{[1]}$ is restricted to $\mathtt{size} = 1 \multimap^{[1]} \varphi$ is TOWER-complete.

5.2. From ALT *to Quantified Computation Tree Logic*

We now consider *Computation Tree Logic* (CTL), a well-known logic for branching time model checking [15, 14]. Among its extensions, in [29] the addition of propositional quantifiers is considered. The resulting logic, called *Quantified Computation Tree Logic* (QCTL) is undecidable on Kripke structures, and TOWER-complete on trees (QCTL^t). This non-elementary boundary has been recently refined in [5]: even when considering just one operator among *exists-next* EX or *exists-finally* EF (the definitions are below), QCTL^t still admits a TOWER-complete satisfiability problem. Here, we reprove the result for EF by first tackling the TOWER-hardness of the logic with the *exists-until* $\mathsf{E}(\varphi \,\mathsf{U}\, \psi)$, to then show that this operator can be defined using EF. Differently from [5] and thanks to the properties of ALT, our reduction does not imbricate until operators, showing that this extension of CTL remains TOWER-hard even when $\mathsf{E}(\varphi \,\mathsf{U}\, \psi)$ is restricted so that φ and ψ are Boolean combinations of propositional symbols.

Let us start by recalling the syntax of QCTL, as defined in [29]. We use AP to denote the countable set of *propositional symbols* $\{p, q, \ldots\}$. The formulae φ of QCTL are built from the following grammar (where $p \in \mathsf{AP}$):

$\pi :=$	$\top$	*(true)*	$\varphi :=$	π	*(atomic formulae)*
$\mid$	p	*(propositional symbol)*	$\mid$	$\varphi \wedge \varphi \mid \neg\varphi$	*(Boolean connectives)*
			$\mid$	$\mathsf{EX}\,\varphi$	*(exists-next modality)*
			$\mid$	$\mathsf{E}(\varphi \,\mathsf{U}\, \varphi)$	*(exists-until modality)*
			$\mid$	$\mathsf{A}(\varphi \,\mathsf{U}\, \varphi)$	*(all-until modality)*
			$\mid$	$\exists p\, \varphi$	*(propositional quantification)*

$$
\begin{array}{lll}
(\mathcal{K},\mathsf{w}) \models p & \text{iff} & \mathsf{w} \in \mathcal{V}(p), \\
(\mathcal{K},\mathsf{w}) \models \mathsf{EX}\,\varphi & \text{iff} & \exists \mathsf{w}' \in R(\mathsf{w}) \text{ such that } (\mathcal{K},\mathsf{w}') \models \varphi, \\
(\mathcal{K},\mathsf{w}) \models \mathsf{E}(\varphi\,\mathsf{U}\,\psi) & \text{iff} & \text{there are } (\mathsf{w}_0,\mathsf{w}_1,\dots) \in \Pi_R(\mathsf{w}) \text{ and } j \in \mathbb{N} \text{ such that} \\
 & & (\mathcal{K},\mathsf{w}_j) \models \psi \text{ and for every } i < j,\ (\mathcal{K},\mathsf{w}_i) \models \varphi, \\
(\mathcal{K},\mathsf{w}) \models \mathsf{A}(\varphi\,\mathsf{U}\,\psi) & \text{iff} & \text{for all } (\mathsf{w}_0,\mathsf{w}_1,\dots) \in \Pi_R(\mathsf{w}), \text{ there is } j \in \mathbb{N} \text{ such that} \\
 & & (\mathcal{K},\mathsf{w}_j) \models \psi \text{ and for every } i < j,\ (\mathcal{K},\mathsf{w}_i) \models \varphi, \\
(\mathcal{K},\mathsf{w}) \models \exists p\ \varphi & \text{iff} & \text{there is } \mathcal{W}' \subseteq \mathcal{W} \text{ such that } (\mathcal{W},R,\mathcal{V}[p{\leftarrow}\mathcal{W}']) \models \varphi.
\end{array}
$$

Figure 12: Satisfaction relation for QCTL.

QCTL is interpreted on standard Kripke structures [28].

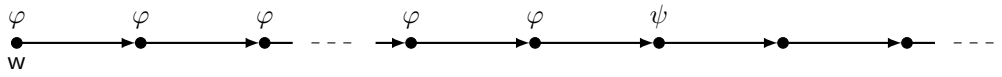
Definition 37 (Kripke structure)**.** A *Kripke structure* is a triple $(\mathcal{W},R,\mathcal{V})$ where $\mathcal{W}$ is a countable set of *worlds*, $R \subseteq \mathcal{W}\times\mathcal{W}$ is a left-total[1] *accessibility relation* and $\mathcal{V} : \mathsf{AP} \to 2^{\mathcal{W}}$ is a *labelling function* which, given a propositional symbol p, returns the set of worlds satisfying p.

The satisfaction of the exists-until and all-until modalities depends on the paths in the structure.

Definition 38 (Path)**.** Let $R \subseteq \mathcal{W}\times\mathcal{W}$ be a binary relation on worlds (possibly left-total). A *path* ρ starting in w is a (possibly finite) sequence of worlds $(\mathsf{w}_0,\mathsf{w}_1,\dots)$ such that $\mathsf{w}_0 = \mathsf{w}$ and $(\mathsf{w}_i,\mathsf{w}_{i+1}) \in R$ for every two successive elements $\mathsf{w}_i,\mathsf{w}_{i+1}$ of the sequence.
(*Maximal Path*) The path ρ is said to be *maximal* whenever it is not a strict prefix of any other path. We denote with $\Pi_R(\mathsf{w})$ the set of *maximal paths* starting in w.

Notice that if R is left-total then $\Pi_R(\mathsf{w})$ is the set of all infinite paths starting in w. Given a world $\mathsf{w} \in \mathcal{W}$, we write $R(\mathsf{w})$ for the set $\{\mathsf{w}' \in \mathcal{W} \mid (\mathsf{w},\mathsf{w}') \in R\}$, i.e. the set of worlds that are *accessible* from w. Therefore $R^*(\mathsf{w})$ (where R^* is the Kleene closure of R) denotes the set of worlds reachable from w, i.e. the worlds belonging to a path in $\Pi_R(\mathsf{w})$.

Let $\mathcal{K} = (\mathcal{W},R,\mathcal{V})$ be a Kripke structure and consider $\mathsf{w} \in \mathcal{W}$. The pair $(\mathcal{K},\mathsf{w})$ denotes a *pointed Kripke structure*, where the world w is the *current world*. Given $(\mathcal{K},\mathsf{w})$, the satisfaction relation $\models$ for formulae in QCTL is defined in Figure 12 (as usual, omitting standard cases for $\top$ and Boolean connectives). The atomic formula p simply asks whether the propositional symbol p is satisfied by w. Using the exists-next modality $\mathsf{EX}\,\varphi$, we can check whether φ holds in a world that is accessible from w. The two *temporal modalities* are more sophisticated. The formula $\mathsf{E}(\varphi\,\mathsf{U}\,\psi)$ checks whether there is a maximal path $(\mathsf{w}_0,\mathsf{w}_1,\dots)$ starting in w for which there is a finite prefix $(\mathsf{w}_0,\dots,\mathsf{w}_{j-1})$ of worlds satisfying φ, followed by the world $\mathsf{w}_j \in R(\mathsf{w}_{j-1})$ that satisfies ψ. This path can be schematised as follows (arrows represent R, every "•" is a world).



[1] left-total means that for each world $\mathsf{w} \in \mathcal{W}$ there is $\mathsf{w}' \in \mathcal{W}$ such that $(\mathsf{w},\mathsf{w}') \in R$.

The formula $\mathsf{A}(\varphi\,\mathsf{U}\,\psi)$ asks the above property to hold for every maximal path, instead of at least one: given a maximal path $(\mathsf{w}_0, \mathsf{w}_1, \dots)$, there must be a finite prefix $(\mathsf{w}_0, \dots, \mathsf{w}_{j-1})$ of worlds satisfying φ, followed by the world $\mathsf{w}_j \in R(\mathsf{w}_{j-1})$ that satisfies ψ. Lastly, the propositional existential quantification $\exists p\,\varphi$ is quite similar to the second-order quantification of second-order logic. Essentially, this formula is satisfied if it is possible to update the satisfaction of the propositional symbol p to a new subset $\mathcal{W}'$ of w, so that then φ holds. In the formal definition given in Figure 12, this update is written as $\mathcal{V}[p \leftarrow \mathcal{W}']$. Similarly to the store update $s[\mathtt{u} \leftarrow \ell']$ of the existential quantification of $\mathsf{SL}(\exists, *, -\!\!*)$, this notation stands for the function obtained from $\mathcal{V}$ by changing the evaluation of p from $\mathcal{V}(p)$ to $\mathcal{W}'$.

The universal quantification $\forall p$ and the Boolean connectives $\Rightarrow$ and $\vee$ are defined as usual. So are the classical temporal operators of CTL, from [15]:

$$\begin{array}{rcll}
\mathsf{EF}\ \varphi & \stackrel{\text{def}}{\equiv} & \mathsf{E}(\top\,\mathsf{U}\,\varphi) & \textit{(exists-finally)} \\
\mathsf{AG}\ \varphi & \stackrel{\text{def}}{\equiv} & \neg\mathsf{EF}\,\neg\varphi & \textit{(all-generally)} \\
\mathsf{AF}\ \varphi & \stackrel{\text{def}}{\equiv} & \mathsf{A}(\top\,\mathsf{U}\,\varphi) & \textit{(all-finally)} \\
\mathsf{EG}\ \varphi & \stackrel{\text{def}}{\equiv} & \neg\mathsf{AF}\,\neg\varphi & \textit{(exists-generally)} \\
\mathsf{E}(\varphi\,\mathsf{M}\,\psi) & \stackrel{\text{def}}{\equiv} & \mathsf{E}(\varphi\,\mathsf{U}\,\varphi \wedge \psi) & \textit{(exists-strong-release)}
\end{array}$$

From [29], the satisfiability problem of QCTL is known to be undecidable on arbitrary Kripke structures, whereas it becomes Tower-complete when the interpretation is restricted to the class of Kripke trees. We denote this restriction with QCTL^t.

Definition 39 (Kripke tree)**.** A Kripke structure $(\mathcal{W},R,\mathcal{V})$ is a *(finitely-branching) Kripke tree* if

1. R^{-1} is functional and acyclic,
2. for every world $\mathsf{w} \in \mathcal{W}$, $R(\mathsf{w})$ is finite,
3. it has a *root*, i.e. $R^*(\mathsf{r}) = \mathcal{W}$ for some $\mathsf{r} \in \mathcal{W}$.

Given $\mathsf{w} \in \mathcal{W}$, the worlds in $R^*(\mathsf{w}) \setminus \{\mathsf{w}\}$ are said to be *descendants* of w. As Kripke structures are left-total, Kripke trees can be seen as finitely-branching infinite trees. This correspondence leads to the satisfiability problem of QCTL^t being in Tower by reduction to monadic second-order logic on trees [29].

From ALT *to* QCTL^t*.* In the following, we only consider Kripke trees instead of arbitrary Kripke structures (and write *pointed Kripke tree* instead of pointed Kripke structure), and we aim at reducing the satisfiability problem of ALT to the satisfiability problem of QCTL^t. The semantics of the formula $\exists p\ \varphi$ should already give a good clue on how to perform such a reduction. Informally speaking, we can represent the nodes of a finite forest as the set of worlds satisfying a propositional symbol $\mathtt{D}$. Then, for instance, the modality $\blacklozenge^*$ can be encoded by using an existential $\exists \mathtt{E}$ that changes the evaluation of a propositional symbol $\mathtt{E}$ so that it only holds on a subset of the worlds satisfying $\mathtt{D}$ (as in the semantics of the repeated sabotage modality). If we are able to check whether only one world satisfies $\mathtt{D}$ but not $\mathtt{E}$, we can then also capture the semantics of the sabotage modality $\blacklozenge$. Similarly, the propositional quantification can be used to encode the universal modality $\langle \mathrm{U} \rangle$, whereas for the reachability predicates `Hit` and `Miss` we can rely on the exists-until modality.

Let us discuss this encoding a little bit further. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest that we want to encode as a pointed Kripke structure $(\mathcal{K}, \mathsf{w})$, where $\mathcal{K} = (\mathcal{W}, R, \mathcal{V})$ is a Kripke tree. We use w

to play the role of the target node t. To encode the forest $\mathcal{F}$ and the current evaluation node n we use the worlds appearing in $R^*(\mathsf{w})$ and three propositional symbols: D, end and n. The intended use of D is to state which elements of $R^*(\mathsf{w})$ encode nodes in $\mathsf{dom}(\mathcal{F})$. We need to be careful here, as $R^*(\mathsf{w})$ is an infinite set whereas $\mathsf{dom}(\mathcal{F})$ is finite. We use the propositional symbol end to solve this inconsistency: we constraint $\mathcal{K}$ to satisfy the formula $\mathsf{AF}(\mathit{end})$ stating that every maximal path $(\mathsf{w}_0, \mathsf{w}_1, \dots) \in \Pi_R(\mathsf{w})$ has a finite prefix $(\mathsf{w}_0, \dots, \mathsf{w}_{j-1})$ ($j \in \mathbb{N}$) of worlds not satisfying end, whereas $\mathsf{w}_j \in \mathcal{V}(\mathit{end})$. Then, a world in $\mathcal{W}$ encodes an element in $\mathsf{dom}(\mathcal{F})$ whenever it satisfies D and it belongs to one of these prefixes. We use the propositional symbol n to encode the current evaluation node. During the translation, we require n to be satisfied by exactly one descendant of w, so that the modality $\langle \mathrm{U} \rangle$ roughly becomes a quantification over n. For technical reasons, we treat in a similar way the world w, which encodes the target node, and require it to be the only world (among the ones in $R^*(\mathsf{w})$) satisfying the auxiliary propositional symbol t. Lastly, we use an additional propositional symbol E in order to encode subforests and deal with the encoding of $\blacklozenge$ and $\blacklozenge^*$ (as already stated above). Notice that we can use the following formula from [29] to check if a formula φ holds in exactly one descendant of w:

$$\mathtt{uniq}(\varphi) \stackrel{\text{def}}{=} \mathsf{EF}(\varphi) \wedge \forall p\ (\mathsf{EF}(\varphi \wedge p) \Rightarrow \mathsf{AG}(\varphi \Rightarrow p)),$$

where $p \in \mathsf{AP}$ is a propositional symbol that does not appear in φ.

Proposition 40 (From [29])**.** Let $(\mathcal{K}, \mathsf{w})$ be a pointed Kripke structure, where $\mathcal{K} = (\mathcal{W}, R, \mathcal{V})$ is a Kripke tree. $(\mathcal{K}, \mathsf{w}) \models \mathtt{uniq}(\varphi)$ iff there is exactly one $\mathsf{w}' \in R^*(\mathsf{w})$ such that $(\mathcal{K}, \mathsf{w}') \models \varphi$.

For the rest of the section, we fix a tuple $S \stackrel{\text{def}}{=} (\mathit{end}, \mathit{n}, \mathit{t})$ of three different propositional symbols, and two (distinct) additional symbols D and E not in S. We also restrict ourselves to pointed forests $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ such that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$. From Lemma 17(I), this restriction is without loss of generality. We formally define the encoding of pointed forests into pointed Kripke trees.

Definition 41 (QCTL^t - Pointed forests encoding)**.** A pointed Kripke tree $(\mathcal{K} = (\mathcal{W}, R, \mathcal{V}), \mathsf{w})$ is an (S, D)-*encoding* of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ if there is a bijection $\mathfrak{f} : \mathcal{N} \to R^*(\mathsf{w})$ such that

1. $\mathfrak{f}(\mathsf{t}) \stackrel{\text{def}}{=} \mathsf{w}$ is the only world in $\mathsf{ran}(\mathfrak{f}) \cap \mathcal{V}(\mathit{t})$, and $\mathfrak{f}(\mathsf{n})$ is the only world in $\mathsf{ran}(\mathfrak{f}) \cap \mathcal{V}(\mathit{n})$,
2. for every $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$, it holds that $(\mathfrak{f}(\mathcal{F}(\mathsf{n}')), \mathfrak{f}(\mathsf{n}')) \in R$,
3. for every infinite path $(\mathsf{w}_0, \mathsf{w}_1 \dots) \in \Pi_R(\mathsf{w})$ there is $i \in \mathbb{N}$ such that
   a. $\mathsf{w}_i \in \mathcal{V}(\mathit{end})$ and for every $j \in [0, i-1]$ we have $\mathsf{w}_j \notin \mathcal{V}(\mathit{end})$,
   b. for every $j \in \mathbb{N}$, ($\mathsf{w}_j \in \mathcal{V}(\mathit{D})$ and $j < i$) if and only if $\mathfrak{f}(\mathsf{n}') = \mathsf{w}_j$ for some $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$.

We simply write *encoding* when (S, D) is clear from the context.

For instance, Figure 13 shows a possible encoding of a pointed forest into a pointed Kripke tree. Informally, the property (1) states that w encodes t and is the only world in $R^*(\mathsf{w})$ satisfying t. Similarly, the world $\mathfrak{f}(\mathsf{n})$ encoding n is the only world in $R^*(\mathsf{w})$ that satisfies n. The property (2) states that the forest must be correctly encoded in the Kripke structure. In particular, notice that the parent relation of the finite forest is inverted so that it becomes the child relation in the Kripke structure (as shown in Figure 13). As $\mathfrak{f}$ is a bijection, the encoding does not merge together subforests that are disconnected in $\mathcal{F}$. Lastly, the property (3) of $\mathfrak{f}$ states that the elements in $\mathsf{dom}(\mathcal{F})$ must be encoded by nodes in $R^*(\mathsf{w})$ that precede every world satisfying end. Moreover,

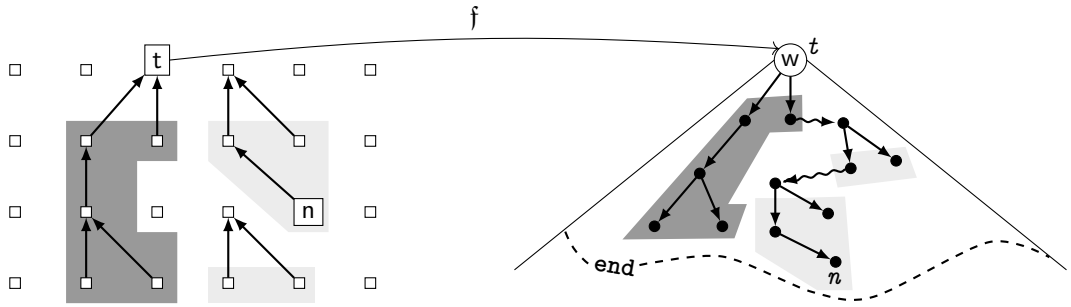


Figure 13: A pointed forest (left) and one of its encoding as a Kripke tree (right).

$$
\begin{aligned}
\tau_u(\mathtt{Hit}) &\stackrel{\text{def}}{\equiv} \mathsf{E}(((u \vee t) \wedge \neg \mathit{end})\,\mathsf{M}\,(u \wedge n)),\\
\tau_u(\mathtt{Miss}) &\stackrel{\text{def}}{\equiv} \mathsf{E}(\neg \mathit{end}\,\mathsf{M}\,(u \wedge n)) \wedge \neg \tau_u(\mathtt{Hit}),\\
\tau_u(\langle \mathrm{U} \rangle \varphi) &\stackrel{\text{def}}{\equiv} \exists n\ (\mathtt{uniq}(n) \wedge \tau_u(\varphi)),\\
\tau_u(\blacklozenge^* \varphi) &\stackrel{\text{def}}{\equiv} \exists \overline{u}\ (\mathsf{AG}\,(\overline{u} \Rightarrow u) \wedge \tau_{\overline{u}}(\varphi)),\\
\tau_u(\blacklozenge \varphi) &\stackrel{\text{def}}{\equiv} \exists \overline{u}\ \big(\mathsf{AG}\,(\overline{u} \Rightarrow u) \wedge \mathtt{uniq}(u \wedge \neg\, \overline{u}) \wedge \mathsf{E}(\neg \mathit{end}\,\mathsf{M}\,(u \wedge \neg\, \overline{u})) \wedge \tau_{\overline{u}}(\varphi)\big).
\end{aligned}
$$

Figure 14: Translation from ALT to QCTL.

among all the descendants of w preceding end, the worlds encoding $\mathsf{dom}(\mathcal{F})$ are the only ones satisfying D. As $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$, t does not satisfy D. It is quite easy to see that every pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ such that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$ admits an encoding.

We now formalise the translation of a formula in ALT into a formula in QCTL^t. During the translation, we alternate between D and E in order to keep track of the domain of the forest, following a $\blacklozenge$ or $\blacklozenge^*$ operator. To facilitate this alternation, we define $\overline{D} \stackrel{\text{def}}{\equiv} E$ and $\overline{E} \stackrel{\text{def}}{\equiv} D$. The translation $\tau_u(\varphi)$ in QCTL^t of a formula φ in ALT is parametrised by $u \in \{D, E\}$ and, implicitly, by S. It is homomorphic for $\top$ and Boolean connectives, and otherwise it is defined in Figure 14. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest such that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$ and let $((\mathcal{W}, R, \mathcal{V}), \mathsf{w})$ be one of its (S, u)-encodings. Let f be the injection certifying that the encoding holds (as in Definition 41). For instance, $\tau_u(\mathtt{Hit})$ requires that there is a path $(\mathsf{w}, \mathsf{w}_1, \ldots, \mathsf{w}_j)$ starting in $\mathsf{f}(\mathsf{t}) = \mathsf{w}$ and whose worlds do not satisfy end and must satisfy u or t. Moreover, the last world w_j must satisfy u and n. From the property (1) of the definition of f, the only world satisfying t is w, which does not satisfy u. Moreover, n is only satisfied by $\mathsf{f}(\mathsf{n})$. Lastly, from the property (3) of f, worlds satisfying u and preceding the ones that satisfy end must encode nodes in the domain of the forest $\mathcal{F}$. Therefore, the path $(\mathsf{w}, \mathsf{w}_1, \ldots, \mathsf{w}_j)$ corresponds to a path in the pointed forest, going from the current evaluation node n (which is encoded by the only world satisfying n) to the target node t. The correctness of the translation follows from the lemma below.

Lemma 42. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest such that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$, and let $(\mathcal{K}, \mathsf{w})$ be a (S,u)-encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$. Given a formula φ in ALT, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$ if and only if $(\mathcal{K}, \mathsf{w}) \models \tau_u(\varphi)$.

The proof of this lemma is shown with a structural induction on the formula φ that, despite

being quite technical, does not pose any serious challenge and it is thus relegated to Appendix C.

In order to conclude the reduction we just need to characterise the set of pointed Kripke trees encoding pointed forests. This can be done with $\mathtt{enc} \stackrel{\text{def}}{=} \neg \mathit{D} \wedge \mathit{t} \wedge \mathtt{uniq}(\mathit{t}) \wedge \mathtt{uniq}(\mathit{n}) \wedge \mathsf{AF}(\mathit{end})$.

Lemma 43. A formula φ in ALT is satisfiable iff so is $\mathtt{enc} \wedge \tau_{\mathit{D}}(\varphi)$ in QCTL^{t}.

Proof. ($\Rightarrow$): Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest satisfying φ. From Lemma 17(I), we can assume without loss of generality that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$. This auxiliary property allows us to construct an (S, D)-encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$. Let $(\mathcal{K}, \mathsf{w})$ be such an encoding. By Lemma 42, $(\mathcal{K}, \mathsf{w}) \models \tau_{\mathit{D}}(\varphi)$. From the property (1) of the encoding (Definition 41), $(\mathcal{K}, \mathsf{w}) \models \mathit{t} \wedge \mathtt{uniq}(\mathit{t}) \wedge \mathtt{uniq}(\mathit{n})$. From the property (3)(a) of the encoding, $(\mathcal{K}, \mathsf{w}) \models \mathsf{AF}(\mathit{end})$. Lastly, property (3)(b) of the encoding, $(\mathcal{K}, \mathsf{w}) \models \neg \mathit{D}$, as t is not in the domain of $\mathcal{F}$.
($\Leftarrow$): Let $(\mathcal{K}, \mathsf{w})$, with $\mathcal{K} = (\mathcal{W}, R, \mathcal{V})$, be a pointed Kripke tree satisfying $\mathtt{enc} \wedge \tau_{\mathit{D}}(\varphi)$. We show that $(\mathcal{K}, \mathsf{w})$ is an (S, D)-encoding of some pointed forest. From $(\mathcal{K}, \mathsf{w}) \models \mathsf{AF}(\mathit{end})$, in every infinite path $(\mathsf{w}_0, \mathsf{w}_1 \dots) \in \Pi_R(\mathsf{w})$ there is $i \in \mathbb{N}$ such that $\mathsf{w}_i \in \mathcal{V}(\mathit{end})$ and for every $j \in [0, i-1]$ we have $\mathsf{w}_j \notin \mathcal{V}(\mathit{end})$ (notice that this corresponds to property (a) of the encoding). Let us consider the set $U \stackrel{\text{def}}{=} \{\mathsf{w}_k \mid k \in \mathbb{N}, (\mathsf{w}_0, \mathsf{w}_1, \dots, \mathsf{w}_k, \dots) \in \Pi_R(\mathsf{w}),$ for every $j \in [0, k], \mathsf{w}_j \notin \mathcal{V}(\mathit{end})\}$. Informally, U is the set of those worlds that can be reached from w without passing through or ending in a world satisfying end. Thanks to $\mathsf{AF}(\mathit{end})$ and the fact that Kripke trees are finitely-branching, U is finite. W.l.o.g. let us assume $\mathcal{N} = \mathcal{W}$ (both sets are countably infinite). Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be the pointed forest characterised as follows:

A. $\mathsf{t} = \mathsf{w}$ and $\{\mathsf{n}\} = \mathcal{V}(\mathit{n})$,

B. $\mathcal{F}(\mathsf{w}) = \mathsf{w}'$ if and only if $\mathsf{w} \in U$, $(\mathcal{K}, \mathsf{w}) \models \mathit{D}$ and $\{\mathsf{w}'\} = R^{-1}(\mathsf{w})$.

Notice that this pointed forest is well-defined and unique. Indeed, from $(\mathcal{K}, \mathsf{w}) \models \mathtt{uniq}(\mathit{n})$, $\mathsf{card}(\mathcal{V}(\mathit{n})) = 1$, and since R encodes the children relation of a tree, R^{-1} is functional. Moreover, from $(\mathcal{K}, \mathsf{w}) \models \neg \mathit{D}$, we conclude that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$. In order to conclude the proof, it is sufficient to show that $(\mathcal{K}, \mathsf{w})$ is an (S, D)-encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$, as we can then derive $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$ by Lemma 42. As a witness of the encoding, let us take any injection $\mathfrak{f} : \mathcal{N} \to R^*(\mathsf{w})$ such that for every $\mathsf{n} \in \mathsf{dom}(\mathcal{F})$, $\mathfrak{f}(\mathsf{n}) = \mathsf{n}$. We show the following properties:

$1_{\mathfrak{f}}$. $\mathfrak{f}(\mathsf{t}) \stackrel{\text{def}}{=} \mathsf{w}$ is the only world in $\mathsf{ran}(\mathfrak{f}) \cap \mathcal{V}(\mathit{t})$, and $\mathfrak{f}(\mathsf{n})$ is the only world in $\mathsf{ran}(\mathfrak{f}) \cap \mathcal{V}(\mathit{n})$,

$2_{\mathfrak{f}}$. for every $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$, it holds that $(\mathfrak{f}(\mathcal{F}(\mathsf{n}')), \mathfrak{f}(\mathsf{n}')) \in R$,

$3_{\mathfrak{f}}$. for every infinite path $(\mathsf{w}_0, \mathsf{w}_1 \dots) \in \Pi_R(\mathsf{w})$ there is $i \in \mathbb{N}$ such that

  a. $\mathsf{w}_i \in \mathcal{V}(\mathit{end})$ and for every $j \in [0, i-1]$ we have $\mathsf{w}_j \notin \mathcal{V}(\mathit{end})$,

  b. for every $j \in \mathbb{N}$, ($\mathsf{w}_j \in \mathcal{V}(\mathit{D})$ and $j < i$) if and only if $\mathfrak{f}(\mathsf{n}') = \mathsf{w}_j$ for some $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$.

The property $(1_{\mathfrak{f}})$ holds directly from (A) and $(\mathcal{K}, \mathsf{w}) \models \mathtt{uniq}(\mathit{t})$. The property $(2_{\mathfrak{f}})$ holds directly from (B). We have already shown property $(3_{\mathfrak{f}})$(a) when considering the formula $\mathsf{AF}(\mathit{end})$. Lastly, $(3_{\mathfrak{f}})$(b) is also quite direct. Let us consider some infinite path $(\mathsf{w}_0, \mathsf{w}_1 \dots) \in \Pi_R(\mathsf{w})$ and some $i \in \mathbb{N}$ such that $\mathsf{w}_i \in \mathcal{V}(\mathit{end})$ and for every $j \in [0, i-1]$ we have $\mathsf{w}_j \notin \mathcal{V}(\mathit{end})$. For the left-to-right direction of $(3_{\mathfrak{f}})$(b), we consider a world $\mathsf{w}_j \in \mathcal{V}(\mathit{D})$ where $j < i$. By definition of U, $\mathsf{w}_j \in U$. As $\mathsf{w}_j \in \mathcal{V}(\mathit{D})$, $\mathsf{w}_j \neq \mathsf{w}$ and so $j > 1$, and moreover $(\mathcal{K}, \mathsf{w}_j) \models \mathit{D}$. By (B), we have $\mathfrak{f}(\mathsf{w}_j) = \mathsf{w}_j \in \mathsf{dom}(\mathcal{F})$. Conversely, suppose that $\mathfrak{f}(\mathsf{w}_j) = \mathsf{w}_j \in \mathsf{dom}(\mathcal{F})$. From (B), $\mathsf{w}_j \in U$ and $(\mathcal{K}, \mathsf{w}_j) \models \mathit{D}$, which in turn implies that $\mathsf{w}_j \in \mathcal{V}(\mathit{u})$ and $j < i$, completing the proof. □

TOWER-hard fragments of QCTL^t. We now take a closer look at the translation. Given a temporal modality $\mathcal{T}$ (e.g. EF) and $k \in \mathbb{N} \cup \{\omega\}$, we write $\mathsf{QCTL}^t(\mathcal{T}^k)$ to denote the fragment of QCTL^t restricted to formulae where the only temporal modality allowed is $\mathcal{T}$, which can be nested at most k times (ω stands for an arbitrary number of imbrications). For instance, $\mathsf{QCTL}^t(\mathsf{EF}^k)$ denotes the set of formulae restricted to the operator EF, which can be nested at most k times. This logic is shown to be k-NEXPTIME-hard in [5], which directly leads to the TOWER-hardness of $\mathsf{QCTL}^t(\mathsf{EF}^\omega)$ and $\mathsf{QCTL}^t(\mathsf{EU}^\omega)$. By analysing our translation it is easy to show that $\mathsf{QCTL}^t(\mathsf{EU}^0)$, i.e. QCTL restricted to the only modality $\mathsf{E}(\varphi\,\mathsf{U}\,\psi)$ where φ and ψ are Boolean combination of propositional symbols, and $\mathsf{QCTL}^t(\mathsf{EF}^1)$ are already TOWER-hard.

Theorem 44. The satisfiability problems of $\mathsf{QCTL}^t(\mathsf{EU}^0)$ and $\mathsf{QCTL}^t(\mathsf{EF}^1)$ are TOWER-complete.

Let us first informally discuss this result. Let us fix a pointed Kripke tree $((\mathcal{W}, R, \mathcal{V}), \mathsf{w})$. First of all, an exists-until modality $\mathsf{E}(\varphi\,\mathsf{U}\,\psi)$ in $\mathsf{QCTL}^t(\mathsf{EU}^0)$ can be shown equivalent to the formula $\chi_{\mathsf{EU}}(\varphi, \psi)$, in $\mathsf{QCTL}^t(\mathsf{EF}^1)$, defined below:

$$\chi_{\mathsf{EU}}(\varphi, \psi) \stackrel{\text{def}}{=} \exists p\big(\mathsf{AG}\,(\neg\varphi \wedge \neg\psi \Rightarrow p) \wedge \mathsf{AG}\,(p \Rightarrow \mathsf{AG}\,p) \wedge \mathsf{EF}\,(\psi \wedge \neg p)\big),$$

where p is a propositional symbol that does not appear in φ or ψ. The idea behind this formula is quite simple. The formula $\chi_{\mathsf{EU}}(\varphi, \psi)$ states that it is possible to change the evaluation of the symbol p so that, for every path starting from the current world w, p holds whenever $\neg\varphi \wedge \neg\psi$ holds (first conjunct of the formula), and if p holds in a world, then it holds on every world reachable from it (second conjunct of the formula). Lastly, the third conjunct states that it is possible to find a world $\mathsf{w}' \in R^*(\mathsf{w})$ satisfying $\psi \wedge \neg p$. This means that the path going from w to w' cannot witness worlds satisfying $\neg\varphi \wedge \neg\psi$, which in turn implies that $\mathsf{E}(\varphi\,\mathsf{U}\,\psi)$ is satisfied. Thanks to $\chi_{\mathsf{EU}}(\varphi, \psi)$, we just need to prove Theorem 44 for $\mathsf{QCTL}^t(\mathsf{EU}^0)$.

Clearly, the translation τ_u is defined so that the resulting formula is already in $\mathsf{QCTL}^t(\mathsf{EU}^0)$. However, we need to deal with the occurrence of $\mathsf{AF}\,(\textit{end})$ used inside the formula $\mathtt{enc}$. Let us first consider the formula $\mathsf{AG}\,(\varphi \Rightarrow \mathsf{AG}\,\psi)$ which is satisfied by models where once φ is found to hold in a certain world w, then ψ is satisfied in every world of $R^*(\mathsf{w})$. Despite not being in $\mathsf{QCTL}^t(\mathsf{EU}^0)$, the formula $\mathsf{AG}\,(\varphi \Rightarrow \mathsf{AG}\,\psi)$ is equivalent to the formula $\chi_{\mathsf{AG}\,\mathsf{AG}}\,(\varphi, \psi)$ below:

$$\chi_{\mathsf{AG}\,\mathsf{AG}}\,(\varphi, \psi) \stackrel{\text{def}}{=} \forall p \forall q\ \big(\mathtt{uniq}(p) \wedge \mathtt{uniq}(q) \wedge \mathsf{EF}\,(p \wedge \varphi) \wedge \mathsf{EF}\,(q \wedge \neg\psi) \Rightarrow \mathsf{E}(\neg p\,\mathsf{M}\,q)\big),$$

where p and q do not appear in φ or ψ. This formula characterises $\mathsf{AG}\,(\varphi \Rightarrow \mathsf{AG}\,\psi)$ by stating that whenever we pick two worlds $\mathsf{w}_1, \mathsf{w}_2 \in R^*(\mathsf{w})$, if w_1 satisfies φ and w_2 does not satisfy ψ, then it is possible to find a path going from the current world w to w_2 that does not include w_1.

Lastly, we define a formula $\chi_{\mathsf{EG}}\,(\varphi)$ that only uses EF modalities and is equivalent to $\mathsf{EG}\,\varphi$, so that then $\neg\chi_{\mathsf{EG}}\,(\neg\varphi)$ is equivalent to $\mathsf{AF}\,\varphi$:

$$\begin{aligned}\chi_{\mathsf{EG}}\,(\varphi) \stackrel{\text{def}}{=} \exists p\big(&\neg p \wedge \mathsf{AG}\,(\neg\varphi \Rightarrow p) \wedge \mathsf{AG}\,(p \Rightarrow \mathsf{AG}\,p) \wedge \\ &\forall q\big(\mathtt{uniq}(q) \wedge \mathsf{EF}\,(q \wedge \neg p) \Rightarrow \mathsf{EF}\,(q \wedge \mathsf{EF}\,(\neg q \wedge \neg p))\big)\big),\end{aligned}$$

where p does not appear in φ. This formula is expressible in $\mathsf{QCTL}^t(\mathsf{EU}^0)$, as every subformula that is not in this fragment is an instance of $\mathsf{AG}\,(\varphi \Rightarrow \mathsf{AG}\,\psi)$. From the correctness of this formula, proved below, we conclude that $\mathsf{AF}\,(\textit{end})$ is expressible in $\mathsf{QCTL}^t(\mathsf{EU}^0)$, leading to Theorem 44. Differently from the formulae $\chi_{\mathsf{EU}}(\varphi, \psi)$ and $\chi_{\mathsf{AG}\,\mathsf{AG}}\,(\varphi, \psi)$, understanding why $\chi_{\mathsf{EG}}\,(\varphi)$ captures $\mathsf{EG}\,\varphi$ is not immediate. Instead of giving just an informal explanation, we directly show the formal proof.

Proof of $\mathsf{EG}\,\varphi \equiv \chi_{\mathsf{EG}}(\varphi)$. Below, we consider a pointed Kripke tree $(\mathcal{K}, \mathsf{w})$ where $\mathcal{K} = (\mathcal{W}, R, \mathcal{V})$.
($\Rightarrow$): Suppose $(\mathcal{K}, \mathsf{w}) \models \mathsf{EG}\,\varphi$, and therefore that there is an infinite path $\rho \in \Pi_R(\mathsf{w})$ where for every $i \geq 0$ the i-th world w_i of the path ρ is such that $(\mathcal{K}, \mathsf{w}_i) \models \varphi$. We write $\widehat{\mathcal{W}}$ for the set of worlds in ρ. Let us consider the model $\mathcal{K}' = (\mathcal{W}, R, \mathcal{V}[p \leftarrow \mathcal{W} \setminus \widehat{\mathcal{W}}])$ obtained from $\mathcal{K}$ by changing the evaluation of p to the set of worlds that are not in $\widehat{\mathcal{W}}$. Since w belongs to ρ, we have $(\mathcal{K}', \mathsf{w}) \models \neg p$. Moreover, as every world in $\widehat{\mathcal{W}}$ satisfies φ whereas every world in $\mathcal{W} \setminus \widehat{\mathcal{W}}$ satisfies p, we conclude that $(\mathcal{K}', \mathsf{w}) \models \mathsf{AG}\,(\neg\varphi \Rightarrow p)$. Similarly, $(\mathcal{K}', \mathsf{w})$ satisfies $\mathsf{AG}\,(p \Rightarrow \mathsf{AG}\,p)$. Indeed, let us consider a world $\mathsf{w}' \in R^*(\mathsf{w})$ such that $\mathsf{w}' \in \mathcal{V}(p)$, and show that for every $\mathsf{w}'' \in R^*(\mathsf{w}')$, $\mathsf{w}'' \in \mathcal{V}(p)$ (as required by this formula). By definition, $\mathsf{w}' \notin \widehat{\mathcal{W}}$. As $\mathcal{K}'$ is a Kripke tree (in particular, it is an acyclic structure), every world w'' reachable from w' does not belong to ρ. So, by definition of $\mathcal{K}'$, $\mathsf{w}'' \in \mathcal{V}(p)$. Lastly, let us focus on the subformula

$$\forall q\big(\mathtt{uniq}(q) \wedge \mathsf{EF}\,(q \wedge \neg p) \Rightarrow \mathsf{EF}\,(q \wedge \mathsf{EF}\,(\neg q \wedge \neg p))\big).$$

We consider a Kripke tree $\mathcal{K}'' = (\mathcal{W}, R, \mathcal{V}[p \leftarrow \mathcal{W} \setminus \widehat{\mathcal{W}}][q \leftarrow \mathcal{W}''])$ obtained from $\mathcal{K}'$ by updating the evaluation of q, and such that $(\mathcal{K}'', \mathsf{w}) \models \mathtt{uniq}(q) \wedge \mathsf{EF}\,(q \wedge \neg p)$. By definition of $\mathtt{uniq}(p)$, there is a world $\widehat{\mathsf{w}}$ such that $\mathcal{W}'' = \{\widehat{\mathsf{w}}\}$. Together with $\mathsf{EF}\,(q \wedge \neg p)$, this implies that $\widehat{\mathsf{w}}$ belongs to the path ρ. Let us say that $\widehat{\mathsf{w}} = \mathsf{w}_i$, i.e. $\widehat{\mathsf{w}}$ is the i-th world in ρ. Let us consider its successor w_{i+1} in the path. Clearly, $(\mathcal{K}'', \mathsf{w}_{i+1}) \models \neg q \wedge \neg p$ and thus $(\mathcal{K}'', \mathsf{w}_i) \models \mathsf{EF}\,(\neg q \wedge \neg p)$, which in turn leads to $(\mathcal{K}'', \mathsf{w}) \models \mathsf{EF}\,(q \wedge \mathsf{EF}\,(\neg q \wedge \neg p))$. From the semantics of the propositional quantification, $(\mathcal{K}', \mathsf{w}) \models \forall q\big(\mathtt{uniq}(q) \wedge \mathsf{EF}\,(q \wedge \neg p) \Rightarrow \mathsf{EF}\,(q \wedge \mathsf{EF}\,(\neg q \wedge \neg p))\big)$ and $(\mathcal{K}, \mathsf{w}) \models \chi_{\mathsf{EG}}(\varphi)$.
($\Leftarrow$): We take the contrapositive and show that if $(\mathcal{K}, \mathsf{w}) \not\models \mathsf{EG}\,\varphi$ then $(\mathcal{K}, \mathsf{w}) \models \neg\chi_{\mathsf{EG}}(\varphi)$. Notice that $\neg\chi_{\mathsf{EG}}(\varphi)$ can be rewritten as

$$\forall p\big(\neg p \wedge \mathsf{AG}\,(\neg\varphi \Rightarrow p) \wedge \mathsf{AG}\,(p \Rightarrow \mathsf{AG}\,p) \Rightarrow \exists q\big(\mathtt{uniq}(q) \wedge \mathsf{EF}\,(q \wedge \neg p) \wedge \mathsf{AG}\,(q \Rightarrow \mathsf{AG}\,(q \vee p))\big)\big)$$

Suppose that $(\mathcal{K}, \mathsf{w}) \not\models \mathsf{EG}\,\varphi$, and thus every path $(\mathsf{w}_0, \mathsf{w}_1, \dots) \in \Pi_R(\mathsf{w})$ must contain a world w_i ($i \geq 0$) s.t. $(\mathcal{K}, \mathsf{w}_i) \models \neg\varphi$. Equivalently, one of the following holds:

A. $(\mathcal{K}, \mathsf{w}) \models \neg\varphi$, or

B. for every path $(\mathsf{w}_0, \mathsf{w}_1, \dots) \in \Pi_R(\mathsf{w})$ there is $j \geq 0$ such that for every $i \leq j$ $(\mathcal{K}, \mathsf{w}_i) \models \varphi$ whereas every $\mathsf{w}' \in R(\mathsf{w}_j)$ is such that $(\mathcal{K}, \mathsf{w}') \models \neg\varphi$.

Let us now consider a Kripke tree $\mathcal{K}' = (\mathcal{W}, R, \mathcal{V}[p \leftarrow \mathcal{W}'])$ obtained from $\mathcal{K}$ by updating the evaluation of p with respect to a set $\mathcal{W}'$ such that $(\mathcal{K}', \mathsf{w})$ satisfies $\neg p \wedge \mathsf{AG}\,(\neg\varphi \Rightarrow p) \wedge \mathsf{AG}\,(p \Rightarrow \mathsf{AG}\,p)$. From the first two conjuncts we conclude that (A) does not hold, and so $(\mathcal{K}, \mathsf{w}) \models \varphi$ and (B) hold. Notice that, in particular, $(\mathcal{K}', \mathsf{w}) \models \varphi \wedge \neg p$. Then, from $\Pi_R(\mathsf{w}) \neq \varnothing$ (Kripke trees are left-total), (B) and $(\mathcal{K}', \mathsf{w}) \models \mathsf{AG}\,(\neg\varphi \Rightarrow p) \wedge \mathsf{AG}\,(p \Rightarrow \mathsf{AG}\,p)$ we derive

C. there is a path $(\mathsf{w}_0, \mathsf{w}_1, \dots) \in \Pi_R(\mathsf{w})$ and a $j \geq 0$ such that $(\mathcal{K}', \mathsf{w}_j) \models \varphi \wedge \neg p$ and for every $\mathsf{w}' \in R^*(\mathsf{w}_j) \setminus \{\mathsf{w}_j\}$ it holds that $(\mathcal{K}', \mathsf{w}') \models p$.

Let us consider the world w_j in (C) and define $\mathcal{K}'' = (\mathcal{W}, R, \mathcal{V}[p \leftarrow \mathcal{W}'][q \leftarrow \{\mathsf{w}_j\}])$ to be the Kripke tree obtained by updating $\mathcal{K}'$ so that q evaluates to $\{\mathsf{w}_j\}$. By definition of $\mathcal{K}''$ and (C), $(\mathcal{K}'', \mathsf{w}) \models \mathtt{uniq}(q) \wedge \mathsf{EF}\,(q \wedge \neg p) \wedge \mathsf{AG}\,(q \Rightarrow \mathsf{AG}\,(q \vee p))$. By semantics of $\exists p$ we have

$$(\mathcal{K}', \mathsf{w}) \models \neg p \wedge \mathsf{AG}\,(\neg\varphi \Rightarrow p) \wedge \mathsf{AG}\,(p \Rightarrow \mathsf{AG}\,p) \Rightarrow \exists q\big(\mathtt{uniq}(q) \wedge \mathsf{EF}\,(q \wedge \neg p) \wedge \mathsf{AG}\,(q \Rightarrow \mathsf{AG}\,(q \vee p))\big)$$

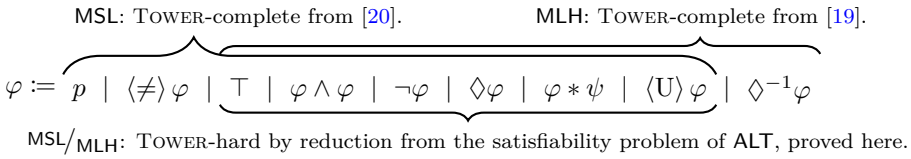
Again from the semantics of $\exists p$, together with the definition of $\mathcal{K}'$, we get $(\mathcal{K}, \mathsf{w}) \models \neg\chi_{\mathsf{EG}}(\varphi)$. $\square$

$$
\begin{array}{lll}
(\mathcal{K},\mathsf{w}) \models \Diamond\varphi & \text{iff} & \text{there is } \mathsf{w}' \in R(\mathsf{w}) \text{ such that } (\mathcal{K},\mathsf{w}) \models \varphi, \\
(\mathcal{K},\mathsf{w}) \models \langle \mathrm{U} \rangle \varphi & \text{iff} & \text{there is } \mathsf{w}' \in \mathcal{W} \text{ such that } (\mathcal{K},\mathsf{w}') \models \varphi, \\
(\mathcal{K},\mathsf{w}) \models \varphi * \psi & \text{iff} & (\mathcal{K}_1,\mathsf{w}) \models \varphi \text{ and } (\mathcal{K}_2,\mathsf{w}) \models \psi \text{ for some } \mathcal{K}_1, \mathcal{K}_2 \text{ s.t. } \mathcal{K}_1 + \mathcal{K}_2 = \mathcal{K}.
\end{array}
$$

Figure 15: Satisfaction relation for $\mathsf{MSL}/_{\mathsf{MLH}}$.

5.3. From ALT *to Modal Separation Logic*

In [19] and later in [20] two families of logics are presented, respectively called *modal logic of heaps* (MLH) and *modal separation logic* (MSL). At their core, both logics can be seen as modal logics extended with separating connectives, hence mixing separation logic (Section 5.1) with the modality EX of CTL (Section 5.2). As we already showed how ALT is captured by these two logics, it is natural to ask ourselves if the same holds for MLH and MSL. In this section, we show that this is indeed the case and, as in the previous two sections, ALT allows us to refine the analysis on these logics. Both MLH and MSL are interpreted on finite Kripke functions. A *finite Kripke function* is a Kripke structure $(\mathcal{W}, R, \mathcal{V})$ (see Definition 37) where $\mathcal{W}$ is infinite and R, instead of being left-total, is finite and weakly functional, i.e. $\mathsf{card}(R) \in \mathbb{N}$ and for every $\mathsf{w}, \mathsf{w}', \mathsf{w}'' \in \mathcal{W}$, if $(\mathsf{w}, \mathsf{w}') \in R$ and $(\mathsf{w}, \mathsf{w}'') \in R$ then $\mathsf{w}' = \mathsf{w}''$. As $\mathcal{N}$ and $\mathcal{W}$ are both countably infinite sets, without loss of generality we assume $\mathcal{W} = \mathcal{N}$. Two Kripke structures $\mathcal{K}_1 = (\mathcal{W}, R_1, \mathcal{V})$ and $\mathcal{K}_2 = (\mathcal{W}, R_2, \mathcal{V})$ are disjoint if $R_1 \cap R_2 = \varnothing$. When this holds, $\mathcal{K}_1 + \mathcal{K}_2$ denotes the model $(\mathcal{W}, R_1 \cup R_2, \mathcal{V})$. To shorten the presentation, in the following diagram we introduce a language having the operators from MSL and MLH, and summarise known and new results on these logics (where $p \in \mathsf{AP}$):



The operator $\Diamond$ is the standard alethic modality from modal logic, $\Diamond^{-1}$ is its converse, and $\langle \neq \rangle$ is the elsewhere modality that capture the somewhere modality $\langle \mathrm{U} \rangle$ with the formula $\langle \mathrm{U} \rangle \varphi = \varphi \vee \langle \neq \rangle \varphi$. Given a pointed finite function $(\mathcal{K}, \mathsf{w})$, where $\mathcal{K} = (\mathcal{W}, R, \mathcal{V})$ is a Kripke-style finite function and $\mathsf{w} \in \mathcal{W}$, Figure 15 recalls the satisfaction relation $\models$ for the fragment $\mathsf{MSL}/_{\mathsf{MLH}}$ of MSL and MLH we show to be Tower-complete (omitting $\top$ and Boolean connectives). By looking at the syntax of $\mathsf{MSL}/_{\mathsf{MLH}}$, compared to the work in [20], ALT allows us to show that propositional symbols and the elsewhere modality can be removed from MSL without changing the complexity status of its satisfiability problem (notice that this logic features the somewhere modality). Similarly, ALT allows us to refine the analysis on the complexity of MLH done in [19] by showing that the $\Diamond^{-1}$ modality is not needed in order to achieve non-elementary complexities.

From ALT *to* $\mathsf{MSL}/_{\mathsf{MLH}}$. As $\mathcal{W}$ and $\mathcal{N}$ are both countably infinite sets, without loss of generality we assume $\mathcal{W} = \mathcal{N}$. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest and let $(\mathcal{K}, \mathsf{w})$ be a pointed finite function where $\mathcal{K} = (\mathcal{N}, R, \mathcal{V})$. As done in Section 5.1 in order to relate ALT to first-order separation logic, we start by introducing the sabotage and repeated sabotage modalities in $\mathsf{MSL}/_{\mathsf{MLH}}$. We define the

$$\begin{array}{lcl@{\qquad}lcl}
\tau(\texttt{Hit}) & \stackrel{\text{def}}{=} & \blacklozenge^*_{\text{ML}}\big(\Diamond\top \wedge [\text{U}](\Diamond\top \Rightarrow \Diamond\Diamond\top)\big), & \tau(\blacklozenge\varphi) & \stackrel{\text{def}}{=} & \blacklozenge_{\text{ML}}(\tau(\varphi) \wedge \langle\text{U}\rangle\, \texttt{selfloop}), \\
\tau(\texttt{Miss}) & \stackrel{\text{def}}{=} & \Diamond\top \wedge \neg\tau(\texttt{Hit}), & \tau(\blacklozenge^*\varphi) & \stackrel{\text{def}}{=} & \blacklozenge^*_{\text{ML}}(\tau(\varphi) \wedge \langle\text{U}\rangle\, \texttt{selfloop}). \\
\tau(\langle\text{U}\rangle\,\varphi) & \stackrel{\text{def}}{=} & \langle\text{U}\rangle(\neg\texttt{selfloop} \wedge \tau(\varphi)), & & &
\end{array}$$

Figure 16: Translation from ALT to $\text{MSL}/_{\text{MLH}}$.

formula $\texttt{size}=1 \stackrel{\text{def}}{=} \langle\text{U}\rangle\,\Diamond\top \wedge \neg(\langle\text{U}\rangle\,\Diamond\top * \langle\text{U}\rangle\,\Diamond\top)$, that is satisfied whenever $\mathsf{card}(R)$=1. Then, the modalities $\blacklozenge$ and $\blacklozenge^*$ are defined in $\text{MSL}/_{\text{MLH}}$ as $\blacklozenge_{\text{ML}}\varphi \stackrel{\text{def}}{=} (\texttt{size}=1) * \varphi$ and $\blacklozenge^*_{\text{ML}}\varphi \stackrel{\text{def}}{=} \top * \varphi$.

For the reduction, we use w to encode the current node n. Encoding t is not so immediate, as $\text{MSL}/_{\text{MLH}}$ does not have propositional symbols. A possible solution is to encode it as a self-loop, so that the formula Hit is translated to a query stating that w reaches the self-loop. So, we introduce the formula selfloop that is satisfied by $(\mathcal{K}, \mathsf{w}')$ if $(\mathsf{w}', \mathsf{w}') \in R$:

$$\texttt{selfloop} \stackrel{\text{def}}{=} \blacklozenge^*_{\text{ML}}(\Diamond\Diamond\top \wedge \neg\blacklozenge_{\text{ML}}\blacklozenge_{\text{ML}}\top).$$

Informally, this formula characterises a self-loop by stating that it is possible to find a structure $\mathcal{K}' \subseteq \mathcal{K}$ that has an accessibility relation of cardinality one and satisfies $\Diamond\Diamond\top$. Suppose for a moment that we are able to use this formula to characterise the class of every Kripke-style finite function having exactly one cycle, and where this cycle is a self-loop, say on a world w_t. On these finite functions, we use w_t to encode the target node t of a finite forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ while being careful that the $\blacklozenge$ and $\blacklozenge^*$ operators of ALT are translated in such a way that the self-loop on w_t is preserved. Because of the specific treatment of w_t, it is convenient to assume that the current evaluation node n is encoded by a world different from w_t, which reflects on the translation of $\langle\text{U}\rangle$. As it was the case for the translation to QCTL^t, the admissibility of this assumption follows by Lemma 17. We formalise the encoding of a pointed forest as a pointed finite function.

Definition 45 ($\text{MSL}/_{\text{MLH}}$ - Pointed forest encoding)**.** Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest such that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$ and $\mathsf{n} \neq \mathsf{t}$. The pointed finite function $((\mathcal{N}, R, \mathcal{V}), \mathsf{n})$ is an *encoding* of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ if and only if for every $\mathsf{n}', \mathsf{n}'' \in \mathcal{N}$ we have $(\mathsf{n}', \mathsf{n}'') \in R \Leftrightarrow (\mathcal{F}(\mathsf{n}') = \mathsf{n}''$ or $\mathsf{n}' = \mathsf{n}'' = \mathsf{t})$.

Notice how R is essentially defined from $\mathcal{F}$ by adding the self-loop (t, t). The translation $\tau(\varphi)$ in MLH of a formula φ in ALT is homomorphic for $\top$ and Boolean connectives, and otherwise it is defined as in Figure 16. We highlight two points of this translation. First, $\tau(\texttt{Hit})$ essentially asks to find a submodel where every path reaches the self-loop and the current evaluation node is in one of these paths. Second, notice how the translation of $\blacklozenge$ and $\blacklozenge^*$ checks that the model is updated so that the self-loop is not lost, as required by our encoding. The following lemma shows the correctness of our translation.

Lemma 46. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed model s.t. $\mathsf{n} \neq \mathsf{t}$ and $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$. Let $(\mathcal{K}, \mathsf{n})$ be an encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$. Given a formula φ in ALT, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$ iff $(\mathcal{K}, \mathsf{n}) \models \tau(\varphi)$.

Proof. Recall that we assume $\mathcal{W} = \mathcal{N}$, and so in what follows we write directly $\mathcal{N}$ and $\mathsf{n}, \mathsf{n}', \mathsf{n}'' \ldots$ instead of $\mathcal{W}$ and $\mathsf{w}, \mathsf{w}', \mathsf{w}'' \ldots$. Then, as in the statement, let $\mathcal{K} = (\mathcal{N}, R, \mathcal{V})$ be a Kripke-style finite function so that $(\mathcal{K}, \mathsf{n})$ is the encoding of a finite forest $(\mathcal{F}, \mathsf{n}, \mathsf{t})$ where $\mathsf{n} \neq \mathsf{t} \notin \mathsf{dom}(\mathcal{F})$. In particular, this means that $R = \mathcal{F} \cup \{(\mathsf{t}, \mathsf{t})\}$ (see Definition 45).

Similarly to Lemma 42, the proof is by structural induction on φ.

base case: $\varphi = \texttt{Hit}$**.**

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{Hit}$,

$\Leftrightarrow$ there is $\delta \geq 1$ such that $\mathcal{F}^{\delta}(\mathsf{n}) = \mathsf{t}$ (*by definition of* $\models$),

$\Leftrightarrow$ $\mathsf{t} \in R^{+}(\mathsf{n})$ (*as* $R = \mathcal{F} \cup \{(\mathsf{t}, \mathsf{t})\}$),

$\Leftrightarrow$ there is a subset $R_1 \subseteq R$ such that

1. for every $\mathsf{n}' \in \mathcal{N}$, if $R_1(\mathsf{n}') \neq \varnothing$ then $\mathsf{t} \in R_1^{+}(\mathsf{n}')$,
2. $R_1(\mathsf{n}) \neq \varnothing$ and $(\mathsf{t}, \mathsf{t}) \in R_1$,

(*again from the definition of* R. R_1 *simply removes worlds that do not reach* t)

$\Leftrightarrow$ there is $R_1 \subseteq R$ such that $R_1(\mathsf{n}) \neq \varnothing$ and for every $\mathsf{n}' \in \mathcal{N}$, if $R_1(\mathsf{n}') \neq \varnothing$ then there is $\mathsf{n}'' \in R_1(\mathsf{n}')$ such that $R_1(\mathsf{n}'') \neq \varnothing$,

(*as* R *is finite and its only cycle is the self-loop on* t)

$\Leftrightarrow$ there is $R_1 \subseteq R$ such that $((\mathcal{N}, R_1, \mathcal{V}), \mathsf{n}) \models \Diamond\top \wedge [\mathrm{U}](\Diamond\top \Rightarrow \Diamond\Diamond\top)$ (*by def. of* $\models$),

$\Leftrightarrow$ $(\mathcal{K}, \mathsf{n}) \models \blacklozenge^{*}(\Diamond\top \wedge [\mathrm{U}](\Diamond\top \Rightarrow \Diamond\Diamond\top))$ (*by definition of* $\blacklozenge^{*}$ *and* $\models$).

base case: $\varphi = \mathtt{Miss}$**.**

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{Miss}$,

$\Leftrightarrow$ $\mathsf{n} \in \mathsf{dom}(\mathcal{F})$ and $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \not\models \mathtt{Hit}$ (*by definition of* $\models$),

$\Leftrightarrow$ $R(\mathsf{n}) \neq \varnothing$ and $(\mathcal{K}, \mathsf{n}) \not\models \tau(\mathtt{Hit})$ (*by def. of the encoding and the previous base case*),

$\Leftrightarrow$ $(\mathcal{K}, \mathsf{n}) \models \Diamond\top \wedge \neg\tau(\mathtt{Hit})$ (*by definition of* $\models$).

We omit the obvious cases for $\top$ and Boolean connectives.

induction step: $\varphi = \langle \mathrm{U} \rangle \psi$**.** By relying on Lemma 17(II), $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$ and $R = \mathcal{F} \cup \{(\mathsf{t}, \mathsf{t})\}$, we have the following set of equivalences:

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathrm{U} \rangle \psi$,

$\Leftrightarrow$ there is $\mathsf{n}' \in \mathcal{N}$ such that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \psi$ (*by def. of* $\models$),

$\Leftrightarrow$ there is $\mathsf{n}' \in \mathcal{N}$ such that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \psi$ and $\mathsf{n}' \neq \mathsf{t}$,

(*by Lemma 17(II) and* $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$)

$\Leftrightarrow$ there is $\mathsf{n}' \in \mathcal{N}$ such that $(\mathcal{K}, \mathsf{n}') \models \tau(\psi)$ and $\mathsf{n}' \neq \mathsf{t}$,

(*by induction hypothesis, as obviously* $(\mathcal{K}, \mathsf{n}')$ *encodes* $(\mathcal{F}, \mathsf{t}, \mathsf{n}')$)

$\Leftrightarrow$ there is $\mathsf{n}' \in \mathcal{N}$ such that $(\mathcal{K}, \mathsf{n}') \models \tau(\psi)$ and $(\mathsf{n}', \mathsf{n}') \notin R$,

(*by* $R = \mathcal{F} \cup \{(\mathsf{t}, \mathsf{t})\}$, *we have* $(\mathsf{n}', \mathsf{n}') \in R$ *if and only if* $\mathsf{n}' = \mathsf{t}$)

$\Leftrightarrow$ there is $\mathsf{n}' \in \mathcal{N}$ such that $(\mathcal{K}, \mathsf{n}') \models \neg\mathtt{selfloop} \wedge \tau(\psi)$ (*by def. of* $\models$),

$\Leftrightarrow$ $(\mathcal{K}, \mathsf{n}) \models \langle \mathrm{U} \rangle(\neg\mathtt{selfloop} \wedge \tau(\psi))$ (*from the semantics of* $\langle \mathrm{U} \rangle$).

induction step: $\varphi = \blacklozenge\psi$**.** ($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge\psi$, and so there is a subforest $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$ and $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \psi$. Let n' be the node removed from $\mathsf{dom}(\mathcal{F})$ in order to obtain $\mathcal{F}'$, i.e. $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F}) \setminus \{\mathsf{n}'\}$. We consider two Kripke-style finite functions $\mathcal{K}_1 = (\mathcal{N}, R_1, \mathcal{V})$ and $\mathcal{K}_2 = (\mathcal{N}, R_2, \mathcal{V})$ such that

A. $R_1 \stackrel{\text{def}}{=} \{(\mathsf{n}', \mathcal{F}(\mathsf{n}'))\}$, B. $R_2 = \mathcal{F}' \cup \{(\mathsf{t}, \mathsf{t})\}$.

From (A), we conclude that $(\mathcal{K}_1, \mathsf{n}) \models \mathtt{size} = 1$. From (B), and by definition of encoding it holds that $(\mathcal{K}_2, \mathsf{n})$ is an encoding of $(\mathcal{F}', \mathsf{t}, \mathsf{n})$. As $(\mathsf{t}, \mathsf{t}) \in R_2$, $(\mathcal{K}_2, \mathsf{n}) \models \langle \mathrm{U} \rangle \, \mathtt{selfloop}$. By induction hypothesis, $(\mathcal{K}_2, \mathsf{n}) \models \tau(\psi)$. From $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F}) \setminus \{\mathsf{n}'\}$, $R_1 \cap R_2 = \varnothing$. From $R = \mathcal{F} \cup \{(\mathsf{t}, \mathsf{t})\}$, (A) and (B), we conclude that $\mathcal{K}_1 + \mathcal{K}_2 = \mathcal{K}$. Thus, $(\mathcal{K}, \mathsf{n})$ satisfies $\mathtt{size} = 1 * (\tau(\psi) \wedge \langle \mathrm{U} \rangle \, \mathtt{selfloop})$, i.e. $\tau(\blacklozenge \, \psi)$.

($\Leftarrow$): Suppose that $(\mathcal{K}, \mathsf{n}) \models \mathtt{size} = 1 * (\tau(\psi) \wedge \langle \mathrm{U} \rangle \, \mathtt{selfloop})$, and so there are two finite functions $\mathcal{K}_1 = (\mathcal{N}, R_1, \mathcal{V})$ and $\mathcal{K}_2 = (\mathcal{N}, R_2, \mathcal{V})$ s.t. $\mathcal{K}_1 + \mathcal{K}_2 = \mathcal{K}$, $\mathsf{card}(R_1) = 1$, $(\mathcal{K}_2, \mathsf{n}) \models \tau(\psi)$ and $(\mathcal{K}_2, \mathsf{n}) \models \langle \mathrm{U} \rangle \, \mathtt{selfloop}$. As $R = \mathcal{F} \cup \{(\mathsf{t}, \mathsf{t})\}$ and $\mathcal{F}$ is acyclic, we have $(\mathsf{t}, \mathsf{t}) \in R_2$ and $R_1 \subseteq \mathcal{F}$. Consider the forest $\mathcal{F}' \stackrel{\text{def}}{=} \mathcal{F} \setminus R_1$. We have $R_2 = \mathcal{F}' \cup \{(\mathsf{t}, \mathsf{t})\}$, and so $(\mathcal{K}_2, \mathsf{n})$ encodes $(\mathcal{F}', \mathsf{t}, \mathsf{n})$. $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \psi$ follows by induction hypothesis. By definition of $\mathcal{F}' \subseteq \mathcal{F}$ we have $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$, and so $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge \, \psi$.

induction step: $\varphi = \blacklozenge^* \, \psi$**.** This case is very similar to the previous one, the only difference being that $\mathcal{F}'$ is not constrained to be such that $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$. □

To conclude the reduction we show that we can characterise the class of models encoding pointed forests, i.e. the pointed finite functions with exactly one cycle, which is a self-loop that does not involve the current evaluation node. We first define a formula that checks whether a Kripke-style finite function has at least one cycle:

$$\mathtt{hascycles} \stackrel{\text{def}}{=} \blacklozenge^*_{\text{ML}} \big(\langle \mathrm{U} \rangle \Diamond \top \wedge [\mathrm{U}](\Diamond \top \Rightarrow \Diamond \Diamond \top) \big).$$

Informally, this formula characterises the presence of a cycle by stating that there is a structure $(\mathcal{W}, R', \mathcal{V}) \subseteq \mathcal{K}$ such that R' is not empty and every world w' that has a successor, i.e. $R'(\mathsf{w}') \neq \varnothing$, also reaches a world in two steps $R'^2(\mathsf{w}) \neq \varnothing$. The cyclicity of $\mathcal{K}$ then follows from the fact that the accessibility relation is finite. Afterwards, the desired property of having exactly one cycle that is a self-loop can be defined by stating that there is a self-loop which, whenever removed, leads to an acyclic Kripke-style finite function. The following formula does the job:

$$\exists 1\mathtt{selfloop} \stackrel{\text{def}}{=} \langle \mathrm{U} \rangle \big(\mathtt{selfloop} \wedge \neg \blacklozenge_{\text{ML}} (\Box \bot \wedge \mathtt{hascycles}) \big).$$

Lemma 47. Every formula φ in ALT is equisatisfiable with $\tau(\varphi) \wedge \exists 1\mathtt{selfloop} \wedge \neg \mathtt{selfloop}$.

Proof. ($\Rightarrow$): Suppose φ to be satisfiable, and let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest satisfying it. From Lemma 17((I) and (II)), we can assume w.l.o.g. that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$ and $\mathsf{n} \neq \mathsf{t}$. Let $(\mathcal{K}, \mathsf{n})$ be the pointed Kripke-style finite function where $\mathcal{K} = (\mathcal{N}, R, \mathcal{V})$ and $R = \mathcal{F} + \{(\mathsf{t}, \mathsf{t})\}$. By Definition 45, $(\mathcal{K}, \mathsf{n})$ is an encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$. We have $(\mathcal{K}, \mathsf{n}) \models \exists 1\mathtt{selfloop}$. By $\mathsf{n} \neq \mathsf{t}$ and the definition of R, $(\mathcal{K}, \mathsf{n}) \models \neg \mathtt{selfloop}$. By Lemma 46, $(\mathcal{K}, \mathsf{n}) \models \tau(\varphi)$.
($\Leftarrow$): Suppose that $\tau(\varphi) \wedge \exists 1\mathtt{selfloop} \wedge \neg \mathtt{selfloop}$ is satisfiable, and let us consider a pointed Kripke-style finite function $(\mathcal{K}, \mathsf{n})$ satisfying it, where $\mathcal{K} = (\mathcal{N}, R, \mathcal{V})$. By $(\mathcal{K}, \mathsf{n}) \models \exists 1\mathtt{selfloop}$, there is a world t such that $(\mathsf{t}, \mathsf{t}) \in R$ and $R \setminus \{(\mathsf{t}, \mathsf{t})\}$ is acylic (and so it is a finite forest). From $(\mathcal{K}, \mathsf{n}) \models \neg \mathtt{selfloop}$ we have $\mathsf{n} \neq \mathsf{t}$. Let us consider the pointed forest $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ where $\mathcal{F} = R \setminus \{(\mathsf{t}, \mathsf{t})\}$. According to Definition 45, $(\mathcal{K}, \mathsf{n})$ is an encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$. By Lemma 46, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$. □

Theorem 48. The fragment of MLH and MSL with the $*$ (alternatively, $\blacklozenge_{\text{ML}}$ and $\blacklozenge^*_{\text{ML}}$), $\top$, Boolean connectives, $\Diamond$ and $\langle \mathrm{U} \rangle$ modalities, has a TOWER-complete satisfiability problem.

6. Conclusions

We studied an *Auxiliary Logic on Trees* (ALT), a quite simple logic that admits a TOWER-complete satisfiability problem. ALT is shown to be easily captured by various non-elementary logics: first-order separation logic, quantified CTL, modal logic of heaps and modal separation logic. Through ALT, we were not only able to connect these logics, but also to refine their analysis and find strict fragments that are still TOWER-hard. Most importantly, with ALT we hope to have shown a set of simple and concrete properties, centered around reachability and submodel reasoning, that when put together lead to logics having a non-elementary satisfiability problem.

This work leaves a few questions open. First, the fragments of ALT where $\blacklozenge$ or $\blacklozenge^*$ are removed from the logic have not been studied yet. The logic without $\blacklozenge^*$ is of particular interest, as it is connected with the sabotage logics from [4]. Second, the analysis done on first-order separation logic and on modal logic of heaps (Sections 5.1 and 5.3) reveals that the complexity of these logics does not change when the operator $*$ and the predicate `emp` are replaced with the less general operators $\blacklozenge$ and $\blacklozenge^*$. We find this point interesting, as from an overview of the literature, it seems that this result also holds for the separation logics considered in [10, 19, 21, 31, 32]. Moreover, for the logics whose expressiveness is known, i.e. the ones in [21, 31], it seems that also the expressive power remains unchanged. However, we struggle to see how to uniformly express the operator $*$ with $\blacklozenge$ and $\blacklozenge^*$, as the resulting logics reason on the model in a different way.

Lastly, this work illustrates the potential of ALT as a tool for proving the TOWER-hardness of logics interpreted on tree-like structures. As the operators of our logic are simple, we hope ALT to be useful in the future to study logics with unknown complexities.

References

[1] T. Antonopoulos and A. Dawar. Separating graph logic from MSO. In *Foundations of Software Science and Computational Structures*, volume 5504 of *LNCS*, pages 63–77. Springer, 2009.

[2] A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. The complexity of clausal fragments of LTL. In *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 8312 of *LNCS*, pages 35–52. Springer, 2013.

[3] G. Aucher, P. Balbiani, L. Fariñas del Cerro, and A. Herzig. Global and local graph modifiers. *Electronic Notes in Theoretical Computer Science*, 231:293–307, 2009.

[4] G. Aucher, J. van Benthem, and D. Grossi. Sabotage modal logic: Some model and proof theoretic aspects. In *Logic, Rationality, and Interaction*, volume 9394 of *LNCS*, pages 1–13. Springer, 2015.

[5] B. Bednarczyk and S. Demri. Why propositional quantification makes modal logics on trees robustly hard? In *Logic in Computer Science*, pages 1–13. IEEE, 2019.

[6] J. Berdine, B. Cook, and S. Ishtiaq. Slayer: Memory safety for systems-level code. In *Computer-Aided Verification*, volume 6806 of *LNCS*, pages 178–183. Springer, 2011.

[7] L. Bozzelli, A. Molinari, A. Montanari, and A. Peron. On the complexity of model checking for syntactically maximal fragments of the interval temporal logic HS with regular expressions. In *Games, Automata, Logics, and Formal Verification*, volume 256 of *EPTCS*, pages 31–45, 2017.

[8] L. Bozzelli, A. Molinari, A. Montanari, A. Peron, and P. Sala. Interval vs. point temporal logic model checking: an expressiveness comparison. In *Foundations of Software Technology and Theoretical Computer Science*, volume 65 of *LIPIcs*, pages 26:1–26:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

[9] R. Brochenin. *Separation logic : expressiveness, complexity, temporal extension.* PhD thesis, 2013.

[10] R. Brochenin, S. Demri, and E. Lozes. On the almighty wand. *Information and Computation*, 211:106–137, 2012.

[11] C. Calcagno, T. Dinsdale-Young, and P. Gardner. Adjunct elimination in context logic for trees. *Information and Computation*, 208:474–499, 2010.

[12] C. Calcagno, D. Distefano, J. Dubreil, D. Gabi, P. Hooimeijer, M. Luca, P. W. O'Hearn, I. Papakonstantinou, J. Purbrick, and D. Rodriguez. Moving fast with software verification. In *Nasa Formal Methods*, volume 9058 of *LNCS*, pages 3–11. Springer, 2015.

[13] C. Calcagno, H. Yang, and P. W. O'Hearn. Computability and complexity results for a spatial assertion language for data structures. In *Foundations of Software Technology and Theoretical Computer Science*, volume 2245 of *LNCS*, pages 108–119. Springer, 2001.

[14] E. M. Clarke. The birth of model checking. In *25 Years of Model Checking: History, Achievements, Perspectives*, pages 1–26. Springer, 2008.

[15] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Logics of Programs*, volume 131 of *LNCS*, pages 52–71. Springer, 1982.

[16] B. Cook, C. Haase, J. Ouaknine, M. J. Parkinson, and J. Worrell. Tractable reasoning in a fragment of separation logic. In *Concurrency Theory*, volume 6901 of *LNCS*, pages 235–249. Springer, 2011.

[17] B. Courcelle. Graph structure and monadic second-order logic: Language theoretical aspects. In *Automata, Languages and Programming*, volume 5125 of *LNCS*, pages 1–13. Springer, 2008.

[18] A. Dawar, P. Gardner, and G. Ghelli. Adjunct elimination through games in static ambient logic. In *Foundations of Software Technology and Theoretical Computer Science*, volume 3328 of *LNCS*, pages 211–223. Springer, 2004.

[19] S. Demri and M. Deters. Two-variable separation logic and its inner circle. *Transactions on Computational Logic*, 16:15:1–15:36, 2015.

[20] S. Demri and R. Fervari. On the complexity of modal separation logics. In *Advances in Modal Logic*, pages 179–198. College Publications, 2018.

[21] S. Demri, D. Galmiche, D. Larchey-Wendling, and D. Méry. Separation logic with one quantified variable. *Theory of Computing Systems*, 61:371–461, 2017.

[22] S. Demri, E. Lozes, and A. Mansutti. The effects of adding reachability predicates in propositional separation logic. In *Foundations of Software Science and Computational Structures*, volume 10803 of *LNCS*, pages 476–493. Springer, 2018.

[23] R. Fervari. *Relation-Changing Modal Logics*. PhD thesis, 2014.

[24] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18:194 – 211, 1979.

[25] V. Goranko. Temporal logics of computations. Lecture Notes from ESSLLI'00, 2000.

[26] V. Goranko, A. Montanari, and G. Sciavicco. A road map of interval temporal logics and duration calculi. *Journal of Applied Non-Classical Logics*, 14:9–54, 2004.

[27] V. Goranko and S. Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.

[28] S. A. Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[29] F. Laroussinie and N. Markey. Quantified CTL: expressiveness and complexity. *Logical Methods in Computer Science*, 10(4), 2014.

[30] L. Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[31] E. Lozes. Adjuncts elimination in the static ambient logic. *Electronic Notes in Theoretical Computer Science*, 96:51–72, 2004.

[32] A. Mansutti. Extending propositional separation logic for robustness properties. In *Foundations of Software Technology and Theoretical Computer Science*, volume 122 of *LIPIcs*, pages 42:1–42:23. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

[33] A. Mansutti. An auxiliary logic on trees: on the TOWER-hardness of logics featuring reachability and submodel reasoning. In *Foundations of Software Science and Computational Structures*, volume 12077 of *LNCS*, pages 462–481. Springer, 2020.

[34] D. A. Martin. Borel determinacy. *Annals of Mathematics*, 102:363–371, 1975.

[35] A. Meier, M. Mundhenk, M. Thomas, and H. Vollmer. The complexity of satisfiability for fragments of CTL and CTL*. *Electronic Notes in Theoretical Computer Science*, 223:201–213, 2008.

[36] A. R. Meyer and L. J. Stockmeyer. Word problems requiring exponential time: Preliminary report. In *Symposium on Theory of Computing*, pages 1–9. ACM, 1973.

[37] B. C. Moszkowski. *Reasoning About Digital Circuits*. PhD thesis, 1983.

[38] P. W. O'Hearn and D. J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5:215–244, 1999.

[39] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 41:1–35, 1969.

[40] J. C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Logic in Computer Science*, pages 55–74. IEEE, 2002.

[41] S. Schmitz. Complexity hierarchies beyond elementary. *Transactions on Computation Theory*, 8:3:1–3:36, 2016.

[42] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the Association for Computing Machinery*, 32:733–749, 1985.

[43] E. Zermelo. über eine anwendung der mengenlehre auf die theorie des schachspiels. In *International Congress of Mathematicians*, volume 2, pages 501–504. Cambridge University, 1913.

Appendix A. Missing proofs from Section 3

Lemma 14. For each rank $\mathsf{rk} \in \mathbb{N}^3$, $\mathsf{ALT}_{\mathsf{rk}}$ is finite up to logical equivalence.

Proof. The proof of this lemma is rather standard, and similar ones can be found in [30, 11, 18]. We rely on the analogous result from propositional logic [30]:

> ($\star$) given a set of formulae S that is finite up to logical equivalence, there are only finitely many Boolean combinations of formulae from S, up to logical equivalence.

We reason by induction on $\mathsf{rk} \in \mathbb{N}^3$ with respect to the order $<_{rk}$.

base case: $\mathsf{rk} = (0,0,0)$. Every formula of rank rk is a Boolean combination of formulae from $\{\mathtt{Hit}, \mathtt{Miss}\}$. By ($\star$) the set of formulae of rank $(0,0,0)$ is finite up to logical equivalence.

For the inductive case, we partition the set of formulae of rank $\mathsf{rk} = (\mathsf{m}, \mathsf{s}, \mathsf{k})$ in two disjoint sets and show that both of them are finite up to logical equivalence:

induction step: formulae dominated by $\langle \mathrm{U} \rangle$, $\blacklozenge$ or $\blacklozenge^*$. We consider the set of formulae dominated by the $\langle \mathrm{U} \rangle$ operator, i.e. the set of every formula φ of the form $\langle \mathrm{U} \rangle \psi$ for some $\psi \in \mathsf{ALT}_{(\mathsf{m}-1,\mathsf{s},\mathsf{k})}$. By induction hypothesis, there are only finitely many such ψ up to logical equivalence. As $\psi \equiv \chi$ implies $\langle \mathrm{U} \rangle \psi \equiv \langle \mathrm{U} \rangle \chi$, the set of formulae dominated by $\langle \mathrm{U} \rangle$ is finite up to logical equivalence. The same reasoning holds for formulae dominated by $\blacklozenge$ or $\blacklozenge^*$.

induction step: formulae that are not dominated by $\langle \mathrm{U} \rangle$, $\blacklozenge$ or $\blacklozenge^*$. We consider the set of formulae belonging to $\mathsf{ALT}_{\mathsf{m},\mathsf{s},\mathsf{k}}$ and that are not dominated by $\langle \mathrm{U} \rangle$, $\blacklozenge$ or $\blacklozenge^*$ operators. Each formula φ of this set is therefore a Boolean combination of formulae $\varphi_1, \ldots, \varphi_n$, syntactically different from φ, that have rank at most $(\mathsf{m}, \mathsf{s}, \mathsf{k})$ and that are equal to $\mathtt{Hit}$ or $\mathtt{Miss}$, or are dominated by $\langle \mathrm{U} \rangle$, $\blacklozenge$ or $\blacklozenge^*$ operators. From the previous case as well as the base case, the set of such formulae $\varphi_1, \ldots, \varphi_n$ is finite up to logical equivalence. Then, by ($\star$) we conclude that the set of formulae of $\mathsf{ALT}_{\mathsf{rk}}$ that are not dominated by $\langle \mathrm{U} \rangle$, $\blacklozenge$ or $\blacklozenge^*$ operators is also finite up to logical equivalence. □

Lemma 18. Let $\mathsf{rk} = (\mathsf{m}, \mathsf{s}, \mathsf{k})$. Let $\mathcal{F}_1, \mathcal{F}_2$ be two forests, and $\mathsf{n}_1, \mathsf{n}_2, \mathsf{t} \in \mathcal{N}$. Suppose that

1. $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$ agree on the set of descendants of t, i.e. for every $\mathcal{F}_1$-descendant or $\mathcal{F}_2$-descendant n of t, $\mathcal{F}_1(\mathsf{n}) = \mathcal{F}_2(\mathsf{n})$, and if n_1 or n_2 are descendants of t, then $\mathsf{n}_1 = \mathsf{n}_2$,
2. $\mathsf{n}_1 \in \mathcal{F}_1[\mathtt{Miss}]_{\mathsf{t}}$ if and only if $\mathsf{n}_2 \in \mathcal{F}_2[\mathtt{Miss}]_{\mathsf{t}}$,
3. $\min(\mathsf{card}(\mathcal{F}_1[\mathtt{Miss}]_{\mathsf{t}}), \mathsf{m} + \mathsf{s} + \mathsf{k}) = \min(\mathsf{card}(\mathcal{F}_2[\mathtt{Miss}]_{\mathsf{t}}), \mathsf{m} + \mathsf{s} + \mathsf{k})$.

Then $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1) \sim_{\mathsf{rk}} (\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$.

Proof. The proof is by induction on the rank rk, with respect to the strict order $<_{rk}$ and by cases on the move made by the spoiler in the game. As the statement is symmetrical with respect to the two pointed forests $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$, we assume w.l.o.g. that the spoiler chooses and plays on the structure $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ and hence define below the strategy of duplicator on $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$. The strategy of the duplicator for the cases where it must reply on $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ can be described from the one below by simply swapping the two structures. Below, the indices (1), (2) and (3) refer to the homonymous properties in the statement of the lemma. We refer to them as *hypothesis* whenever we consider $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$. Instead, we call them *properties* when we are proving them for subforests of $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ and $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$.

base case: $\mathsf{rk} = (0,0,0)$**.** The hypothesis (1) and (2) imply that for every $\pi \in \{\mathtt{Miss}, \mathtt{Hit}\}$, the double implication $((\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1) \models \pi$ iff $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2) \models \pi)$ holds. Since the spoiler cannot play any move, the duplicator wins the game.

For the induction step, we assume $\mathsf{rk} = (\mathsf{m}, \mathsf{s}, \mathsf{k}) \neq (0,0,0)$ and that the lemma holds for every $\mathsf{rk}' <_{rk} \mathsf{rk}$. We divide the proof following the move of the spoiler.

induction step: the spoiler plays a $\langle \mathrm{U} \rangle$ move. This implies $\mathsf{m} \geq 1$. Let $\mathsf{n}'_1 \in \mathcal{N}$ be the node choosen by the spoiler. Let us consider the following procedure for the duplicator:

> **if** n'_1 is a hit node of $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ **then** the duplicator selects n'_1
> **else if** $\mathsf{n}'_1 \in \mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}$ **then** the duplicator selects a node $\mathsf{n}'_2 \in \mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}$
> **else** the duplicator selects a node $\mathsf{n}'_2 \notin \mathsf{dom}(\mathcal{F}_2)$.

Notice the this procedure is well-defined. In particular, if $\mathsf{n}'_1 \in \mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}$ then from $\mathsf{m} \geq 1$ and the hypothesis (3), we conclude that $\mathsf{card}(\mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}) \geq 1$, so that the duplicator can effectively select a node $\mathsf{n}'_2 \in \mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}$. Moreover, the hypothesis (1) insures that if n'_1 is a hit node of $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ then it is also a hit node of $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$. Lastly, if the duplicator selects a node $\mathsf{n}'_2 \notin \mathsf{dom}(\mathcal{F}_2)$, it means that n'_1 is not a hit or miss node, hence $\mathsf{n}'_1 \notin \mathsf{dom}(\mathcal{F}_1)$. The EF-game continues on the state $((\mathcal{F}_1, \mathsf{t}, \mathsf{n}'_1), (\mathcal{F}_2, \mathsf{t}, \mathsf{n}'_2), (\mathsf{m}-1, \mathsf{s}, \mathsf{k}))$. By definition of n'_2, we can check that $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}'_1)$ and $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}'_2)$ satisfy the three properties (1), (2) and (3), w.r.t. the rank $(\mathsf{m}-1, \mathsf{s}, \mathsf{k})$. By induction hypothesis, we conclude $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}'_1) \sim_{(\mathsf{m}-1,\mathsf{s},\mathsf{k})} (\mathcal{F}_2, \mathsf{t}, \mathsf{n}'_2)$. This implies that, by relying on the procedure above, the duplicator can build a winning strategy for the game $((\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1), (\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2), \mathsf{rk})$.

induction step: the spoiler plays a $\blacklozenge$ move. This implies $\mathsf{s} \geq 1$. Let $\mathcal{F}'_1 \subseteq \mathcal{F}_1$ be the subforest chosen by the spoiler. We have $\mathsf{card}(\mathcal{F}'_1) = \mathsf{card}(\mathcal{F}_1) - 1$. Let $\overline{\mathsf{n}}$ be the only node in $\mathsf{dom}(\mathcal{F}_1) \setminus \mathsf{dom}(\mathcal{F}'_1)$. Let us consider the following procedure for the duplicator:

> **if** $\overline{\mathsf{n}}$ is a hit node of $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$ **then** the duplicator selects the forest $\mathcal{F}_2 \setminus \{(\overline{\mathsf{n}}, \mathcal{F}_2(\overline{\mathsf{n}}))\}$
> **else** the duplicator selects a forest $\mathcal{F}_2 \setminus \{(\mathsf{n}', \mathcal{F}_2(\mathsf{n}'))\}$
> where $\mathsf{n}' \in \mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}$, and $\mathsf{n}' = \mathsf{n}_2 \Leftrightarrow \overline{\mathsf{n}} = \mathsf{n}_1$.

Notice that this procedure is well-defined. In particular, if $\overline{\mathsf{n}}$ is a hit node of $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$, from the hypothesis (1) $\overline{\mathsf{n}}$ is a hit node of $(\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2)$, and so $(\overline{\mathsf{n}}, \mathcal{F}_2(\overline{\mathsf{n}}))$ is defined. Moreover, if $\overline{\mathsf{n}}$ is not a hit node, then from $\overline{\mathsf{n}} \in \mathsf{dom}(\mathcal{F}_1)$ we conclude that it is a miss node. By $\mathsf{s} \geq 1$ and thanks to the hypothesis (3), $\mathsf{card}(\mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}) \geq 1$. With the hypothesis (2), this implies that the duplicator can effectively select the forest in the else branch of the procedure. Let $\mathcal{F}'_2$ be the forest selected by the duplicator, using the procedure above. We show that $(\mathcal{F}'_1, \mathsf{t}, \mathsf{n}_1)$ and $(\mathcal{F}'_2, \mathsf{t}, \mathsf{n}_2)$ satisfy the properties (1), (2) and (3) w.r.t. the rank $(\mathsf{m}, \mathsf{s}-1, \mathsf{k})$. We divide the proof into two cases, depending on whether or not $\overline{\mathsf{n}} \in \mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}$.

case: $\overline{\mathsf{n}} \in \mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}$**.** Let n' be the node such that $\{\mathsf{n}'\} = \mathsf{dom}(\mathcal{F}_2) \setminus \mathsf{dom}(\mathcal{F}'_2)$. From the definition of the procedure above, $\mathsf{n}' \in \mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}$, and $\mathsf{n}' = \mathsf{n}_1$ if and only if $\overline{\mathsf{n}} = \mathsf{n}_2$. This implies the satisfaction of the property (2). Moreover, the property (1) is also satisfied. Indeed, since $\overline{\mathsf{n}}$ is a miss node, every $\mathcal{F}'_1$-descendant of t is also a $\mathcal{F}_1$-descendant of t, and vice versa. Similarly, as n' is a miss node, every $\mathcal{F}'_2$-descendant of t is also a $\mathcal{F}_2$-descendant of t, and vice versa. Thus, the property (1) is implied by the hypothesis (1).

In order to conclude this case, we prove the satisfaction of property (3). First, since $\overline{\mathsf{n}}$ is a miss node, $\mathcal{F}_1'[\mathtt{Miss}]_\mathsf{t} \cup \{\overline{\mathsf{n}}\} = \mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}$. Similarly, $\mathcal{F}_2'[\mathtt{Miss}]_\mathsf{t} \cup \{\mathsf{n}'\} = \mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}$. As $\overline{\mathsf{n}} \notin \mathsf{dom}(\mathcal{F}_1')$ and $\mathsf{n}' \notin \mathsf{dom}(\mathcal{F}_2')$, we conclude that

$$\mathsf{card}(\mathcal{F}_1'[\mathtt{Miss}]_\mathsf{t}) + 1 = \mathsf{card}(\mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}) \text{ and } \mathsf{card}(\mathcal{F}_2'[\mathtt{Miss}]_\mathsf{t}) + 1 = \mathsf{card}(\mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}).$$

Thanks to the hypothesis (3), i.e.

$$\min(\mathsf{card}(\mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}), \mathsf{m}+\mathsf{s}+\mathsf{k}) = \min(\mathsf{card}(\mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}), \mathsf{m}+\mathsf{s}+\mathsf{k}),$$

we show property (3) with the following equivalences:

$$\begin{aligned}
\min(\mathsf{card}(\mathcal{F}_1'[\mathtt{Miss}]_\mathsf{t}), \mathsf{m}+(\mathsf{s}-1)+\mathsf{k}) &= \min(\mathsf{card}(\mathcal{F}_1'[\mathtt{Miss}]_\mathsf{t})+1, \mathsf{m}+\mathsf{s}+\mathsf{k}) - 1 \\
&= \min(\mathsf{card}(\mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}), \mathsf{m}+\mathsf{s}+\mathsf{k}) - 1 \\
&= \min(\mathsf{card}(\mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}), \mathsf{m}+\mathsf{s}+\mathsf{k}) - 1 \\
&= \min(\mathsf{card}(\mathcal{F}_2'[\mathtt{Miss}]_\mathsf{t})+1, \mathsf{m}+\mathsf{s}+\mathsf{k}) - 1 \\
&= \min(\mathsf{card}(\mathcal{F}_2'[\mathtt{Miss}]_\mathsf{t}), \mathsf{m}+(\mathsf{s}-1)+\mathsf{k}).
\end{aligned}$$

Here, we use the equivalence $\min(x+1, y+1) = \min(x, y) + 1$ (x, y arbitrary numbers).

case: $\overline{\mathsf{n}} \notin \mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}$**.** This implies that $\overline{\mathsf{n}}$ is a hit node of $(\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1)$, and from the procedure followed by the duplicator, $\mathcal{F}_2' = \mathcal{F}_2 \setminus \{(\overline{\mathsf{n}}, \mathcal{F}_2(\overline{\mathsf{n}}))\}$. First of, since $\overline{\mathsf{n}}$ is a hit node and $\mathsf{dom}(\mathcal{F}_1) \setminus \mathsf{dom}(\mathcal{F}_1') = \{\overline{\mathsf{n}}\}$, we can show that

$$\mathcal{F}_1'[\mathtt{Miss}]_\mathsf{t} = \mathcal{F}_1[\mathtt{Miss}]_\mathsf{t} \cup \{\mathsf{n}' \in \mathcal{N} \mid \mathsf{n}' \text{ is a } \mathcal{F}_1\text{-descendant of } \overline{\mathsf{n}}\}.$$

Indeed, when $(\overline{\mathsf{n}}, \mathcal{F}(\overline{\mathsf{n}}))$ is removed from $\mathcal{F}_1$, all its descendants become miss nodes, whereas every other hit node of $\mathcal{F}_1$ ($\overline{\mathsf{n}}$ excluded) is still a hit node of $\mathcal{F}_1'$. The same holds true for $\mathcal{F}_2$, so that the following equality holds:

$$\mathcal{F}_2'[\mathtt{Miss}]_\mathsf{t} = \mathcal{F}_2[\mathtt{Miss}]_\mathsf{t} \cup \{\mathsf{n}' \in \mathcal{N} \mid \mathsf{n}' \text{ is a } \mathcal{F}_2\text{-descendant of } \overline{\mathsf{n}}\}.$$

By hypothesis (1), removing $(\overline{\mathsf{n}}, \mathcal{F}_2(\overline{\mathsf{n}}))$ from both $\mathcal{F}_1$ and $\mathcal{F}_2$ leads to two pointed forests that agree on the set of descendants of t. Thus, property (1) is satisfied. Moreover, again from the hypothesis (1), the set of $\mathcal{F}_1$-descendants of $\overline{\mathsf{n}}$ is also the set of $\mathcal{F}_2$-descendants of $\overline{\mathsf{n}}$, i.e.

$$\{\mathsf{n}' \in \mathcal{N} \mid \mathsf{n}' \text{ is a } \mathcal{F}_1\text{-descendant of } \overline{\mathsf{n}}\} = \{\mathsf{n}' \in \mathcal{N} \mid \mathsf{n}' \text{ is a } \mathcal{F}_2\text{-descendant of } \overline{\mathsf{n}}\}.$$

This implies two things. First, from the characterisation of $\mathcal{F}_1'[\mathtt{Miss}]_\mathsf{t}$ and $\mathcal{F}_2'[\mathtt{Miss}]_\mathsf{t}$ (above), together with the hypothesis (2), $(\mathcal{F}_1', \mathsf{t}, \mathsf{n}_1)$ and $(\mathcal{F}_2', \mathsf{t}, \mathsf{n}_2)$ satisfy property (2). Second, thanks to the hypothesis (3), i.e.

$$\min(\mathsf{card}(\mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}), \mathsf{m}+\mathsf{s}+\mathsf{k}) = \min(\mathsf{card}(\mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}), \mathsf{m}+\mathsf{s}+\mathsf{k}),$$

we show property (3) with the following equivalences, where $\beta = \mathsf{m}+(\mathsf{s}-1)+\mathsf{k}$:

$$\begin{aligned}
\min(\mathsf{card}(\mathcal{F}_1'[\mathtt{Miss}]_\mathsf{t}), \beta) &= \min(\mathsf{card}(\mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}) + \mathsf{card}((\mathcal{F}_1^{-1})^+(\overline{\mathsf{n}})), \beta) \\
&= \min(\mathsf{card}(\mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}) + \mathsf{card}((\mathcal{F}_2^{-1})^+(\overline{\mathsf{n}})), \beta) \\
&= \min(\mathsf{card}(\mathcal{F}_2'[\mathtt{Miss}]_\mathsf{t}), \beta)
\end{aligned}$$

where, given $j \in \{1, 2\}$, $(\mathcal{F}_j^{-1})^+(\overline{\mathsf{n}})$ is the set of $\mathcal{F}_j$-descendants of $\overline{\mathsf{n}}$.

In both cases, since we have shown that $(\mathcal{F}_1', \mathsf{t}, \mathsf{n}_1)$ and $(\mathcal{F}_2', \mathsf{t}, \mathsf{n}_2)$ satisfy the properties (1), (2) and (3) w.r.t. the rank $(\mathsf{m}, \mathsf{s}-1, \mathsf{k})$, we can apply the induction hypothesis and conclude that

$(\mathcal{F}'_1, \mathsf{t}, \mathsf{n}_1) \sim_{(\mathsf{m},\mathsf{s}-1,\mathsf{k})} (\mathcal{F}'_2, \mathsf{t}, \mathsf{n}_2)$. This implies that, by relying on the procedure above, the duplicator can build a winning strategy for the game $((\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1), (\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2), \mathsf{rk})$.

induction step: the spoiler plays a $\blacklozenge^*$ move. This implies $\mathsf{k} \geq 1$. Let $\mathcal{F}'_1 \subseteq \mathcal{F}_1$ be the forest choosen by the spoiler. Let us partition $\mathcal{F}'_1$ into the two subforests H and M_1 s.t.

$$H \stackrel{\text{def}}{=} \{(\mathsf{n}, \mathcal{F}_1(\mathsf{n})) \in \mathcal{F}'_1 \mid \mathsf{n} \text{ is a } \mathcal{F}_1\text{-descendant of } \mathsf{t}\},$$
$$M_1 \stackrel{\text{def}}{=} \{(\mathsf{n}, \mathcal{F}_1(\mathsf{n})) \in \mathcal{F}'_1 \mid \mathsf{n} \in \mathcal{F}_1[\mathtt{Miss}]_\mathsf{t}\}.$$

By hypothesis (1), H is a subforest of $\mathcal{F}_2$. Let us consider a subforest M_2 of $\mathcal{F}_2$ such that

A. M_2 contains only miss nodes of $\mathcal{F}_2$, i.e. $\mathsf{dom}(M_2) \subseteq \mathcal{F}_2[\mathtt{Miss}]_\mathsf{t}$,

B. $\mathsf{n}_2 \in M_2$ if and only if $\mathsf{n}_1 \in M_1$,

C. $\mathsf{card}(M_2) = \min(M_1, \mathsf{m} + \mathsf{s} + (\mathsf{k} - 1))$.

From the hypothesis (2) and (3), the subforest M_2 can always be defined. Moreover, M_2 is disjoint from H. Let us show that the duplicator has a winning strategy in which it replies to $\mathcal{F}'_1$ with the subforest $\mathcal{F}'_2 \stackrel{\text{def}}{=} H \cup M_2$ of $\mathcal{F}_2$. We show that $(\mathcal{F}'_1, \mathsf{t}, \mathsf{n}_1)$ and $(\mathcal{F}'_2, \mathsf{t}, \mathsf{n}_2)$ satisfy the properties (1), (2) and (3) w.r.t. the rank $(\mathsf{m}, \mathsf{s}, \mathsf{k} - 1)$. Property (1) holds directly from the definition of H together with hypothesis (1). For the properties (2) and (3), we first notice that $\mathcal{F}'_1[\mathtt{Miss}]_\mathsf{t} = H[\mathtt{Miss}]_\mathsf{t} \cup \mathsf{dom}(M_2)$ and that $\mathcal{F}'_1[\mathtt{Miss}]_\mathsf{t} = H[\mathtt{Miss}]_\mathsf{t} \cup \mathsf{dom}(M_1)$. Then, property (2) stems from (B), whereas property (3) stems from (C). This allows us to apply the induction hypothesis to conclude that $(\mathcal{F}'_1, \mathsf{t}, \mathsf{n}_1) \sim_{(\mathsf{m},\mathsf{s},\mathsf{k}-1)} (\mathcal{F}'_2, \mathsf{t}, \mathsf{n}_2)$. Therefore, the duplicator can build a winning strategy for the game $((\mathcal{F}_1, \mathsf{t}, \mathsf{n}_1), (\mathcal{F}_2, \mathsf{t}, \mathsf{n}_2), \mathsf{rk})$. □

Appendix B. Missing proofs from Section 4

Lemma 24. Let $\mathfrak{w} \in \Sigma_\bullet^+$ and let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest encoding the word $\mathfrak{f}(\mathfrak{w}) \in [1, 2n]^+$.

(I) $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{mark}_\Sigma$ iff n encodes a marked symbol of $\Sigma_\bullet$.

(II) $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{marks}_\Sigma \geq \beta$ iff $\mathcal{F}$ contains at least β nodes encoding marked symbols of $\Sigma_\bullet$.

(III) $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \#\mathtt{markAnc}_\Sigma \geq \beta$ iff n has at least β ancestors encoding marked symbols of $\Sigma_\bullet$.

The proofs of (I) and (III) are done by simply unrolling the definitions. Of these two statements, we just show the left-to-right directions. The right-to-left direction is quite straightforward. The proof of (II) is by induction on β.

Proof of (I). ($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{mark}_\Sigma$, so there must be a symbol $\mathsf{a} \in \Sigma$ such that

$$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models (\#\mathtt{child} = 2\mathsf{a} \wedge \mathtt{1st}_{[1,2n]}) \vee (\#\mathtt{child} = 2\mathsf{a}{+}1 \wedge \neg\mathtt{1st}_{[1,2n]}).$$

First, suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \#\mathtt{child} = 2\mathsf{a} \wedge \mathtt{1st}_{[1,2n]}$. From Lemma 8(II), n has exactly $2\mathsf{a}$ children and it is a descendant of t. Moreover, from $\mathtt{1st}_{[1,2n]}$, n is the first node in the main path of $\mathcal{F}$. Then, by definition of encoding of a word (Definition 3), all the children of n are character nodes, and so n encodes a marked symbol. Otherwise, consider the case where the second disjunct holds, i.e. $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \#\mathtt{child} = 2\mathsf{a}{+}1 \wedge \neg\mathtt{1st}_{[1,2n]}$. From Lemma 8(II), n has exactly $2\mathsf{a} + 1$ $\mathcal{F}$-children and it is a descendant of t. Again, from Definition 3, n is a node in the main path of $\mathcal{F}$. From

$\neg\mathtt{1st}_{[1,2n]}$, n cannot be the first node in the main path of $\mathcal{F}$. So, one of its children is a main node, whereas all its other children are character nodes (by Definition 3). Thus, n has exactly $2\mathsf{a}$ character nodes, meaning that it encodes a marked symbol. □

Proof of (II). It should be noted that we cannot apply Lemma 6 to prove this statement, as its hypothesis (2) is not satisfied. The proof is by induction on β. The base case for $\beta = 0$ is trivial. Let us look at the *induction step*. Assume $\beta = k + 1$ for some $k \in \mathbb{N}$.
($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{marks}_{\Sigma} \geq k{+}1$, and therefore there is a node $\mathsf{n}' \in \mathcal{N}$ such that

A. $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \mathtt{mark}_{\Sigma}$. From Lemma 24(I), n' encodes a marked symbol in $\Sigma_\bullet$,

B. $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \blacklozenge(\neg\mathtt{inDom} \wedge \mathtt{marks}_{\Sigma} \geq k)$.

From (B), there is a forest $\mathcal{F}'$ such that

C. $\mathcal{F}' \subseteq \mathcal{F}$ and $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$;

D. $(\mathcal{F}', \mathsf{t}, \mathsf{n}') \models \neg\mathtt{inDom} \wedge \mathtt{marks}_{\Sigma} \geq k$.

From (D):

E. by semantics of $\mathtt{inDom}$, the node n' does not belong to $\mathsf{dom}(\mathcal{F}')$;

F. by induction hypothesis, $\mathcal{F}'$ contains at least k nodes encoding marked symbols of $\Sigma_\bullet$.

(A) implies that n' is a $\mathcal{F}$-descendant of t that is a main node. Together with (C) and (E), this implies that $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F}) \setminus \{\mathsf{n}'\}$. Because of this (by (F)) the k nodes encoding marked symbols w.r.t. $\mathcal{F}'$ must be elements of the main path of $\mathcal{F}$ that are ancestors of n'. Indeed, n' and all its $\mathcal{F}$-descendants are not $\mathcal{F}'$-descendants of t. Recall that a symbol is encoded by a main node by using the number of its children that are character nodes, and that main nodes are not character nodes. As $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F}) \setminus \{\mathsf{n}'\}$ and n' is a main node, every node encoding a marked symbol in $\mathcal{F}'$ also encodes a marked symbol in $\mathcal{F}$. Thus, by (F) and (A), $\mathcal{F}$ contains at least $k + 1$ nodes encoding marked symbols in $\Sigma_\bullet$.
($\Leftarrow$): Let $\{(\mathsf{n}_1, \mathsf{n}_2) \dots (\mathsf{n}_{m-1}, \mathsf{n}_m)\}$ be the main path of $\mathcal{F}$, so that, for every $i \in [1, m]$, n_i corresponds to the main node encoding the i-th character of $\mathfrak{f}(\mathfrak{w})$. Suppose that $\mathcal{F}$ contains at least $k+1$ nodes encoding marked symbols of $\Sigma_\bullet$, or equivalently that there are $k + 1$ positions $\{i_1, \dots, i_{k+1}\}$ such that, for every $j \in [1, k + 1]$, n_{i_j} encodes a marked symbol. Consider n_{i_1}, the node encoding the first marked symbol of $\mathfrak{f}(\mathfrak{w})$. Furthermore, consider the subforest $\mathcal{F}' = \mathcal{F} \setminus \{(\mathsf{n}_{i_1}, \mathsf{n}_{i_1+1})\}$. We have:

W. as n_{i_1} encodes a marked symbol, from Lemma 24(I), $(\mathcal{F}, \mathsf{t}, \mathsf{n}_{i_1}) \models \mathtt{mark}_{\Sigma}$,

X. $\mathsf{n}_{i_1} \notin \mathsf{dom}(\mathcal{F}')$. So, $(\mathcal{F}', \mathsf{n}_{i_1}, \mathsf{t}) \models \neg\mathtt{inDom}$,

Y. $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$.

Z. For every node n'' of the k nodes in $\{\mathsf{n}_{i_2}, \dots, \mathsf{n}_{i_{k+1}}\}$, the number of $\mathcal{F}'$-children of n'' that are character nodes is the same as the number of $\mathcal{F}$-children of n'' that are character nodes. Moreover, n'' is a $\mathcal{F}'$-descendant of t. This property holds as n_{i_1} is a descendant of every node in $\{\mathsf{n}_{i_2}, \dots, \mathsf{n}_{i_{k+1}}\}$.

From (Z), $\mathcal{F}'$ contains at least k nodes encoding marked symbols of $\Sigma_\bullet$. By induction hypothesis, $(\mathcal{F}', \mathsf{t}, \mathsf{n}_{i_1}) \models \mathtt{marks}_{\Sigma} \geq k$. With (X) and (Y), $(\mathcal{F}, \mathsf{t}, \mathsf{n}_{i_1}) \models \blacklozenge(\neg\mathtt{inDom} \wedge \mathtt{marks}_{\Sigma} \geq k)$. Lastly, by (W) and the definition of the modality $\langle \mathrm{U} \rangle$, we conclude: $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{marks}_{\Sigma} \geq k{+}1$. □

Proof of (III). Let $\mathsf{n}_1\mathsf{n}_2\dots\mathsf{n}_m$ be the nodes in the main path of $\mathcal{F}$, so that, for every $j \in [1,m]$, n_j corresponds to the node encoding the j-th character of $\mathfrak{f}(\mathfrak{w})$.
($\Rightarrow$): Suppose $(\mathcal{F},\mathsf{t},\mathsf{n}) \models \#\mathtt{markAnc}_\Sigma \geq \beta$. From the first conjunct of $\#\mathtt{markAnc}_\Sigma \geq \beta$, n is a main node. Equivalently, there is $j \in [1,m]$ such that $\mathsf{n}_j = \mathsf{n}$. From the second conjunct of $\#\mathtt{markAnc}_\Sigma \geq \beta$, we conclude that there is a finite forest $\mathcal{F}' \subseteq \mathcal{F}$ such that

A. $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F})\backslash\{\mathsf{n}\}$. Indeed, n is a $\mathcal{F}$-descendant of t (as it is a main node), but $(\mathcal{F}',\mathsf{t},\mathsf{n}) \not\models \mathtt{inDom}$ implies $\mathsf{n} \notin \mathsf{dom}(\mathcal{F}')$.

B. $(\mathcal{F}',\mathsf{t},\mathsf{n}) \models \mathtt{marks}_\Sigma \geq \beta$. Thus, by Lemma 24(II) $\mathcal{F}'$ contains at least β main nodes encoding marked symbols of $\Sigma_\bullet$.

By (A), since $\mathcal{F}$ encodes a finite word $\mathfrak{f}(\mathfrak{w})$ and n is the j-th element of its main path, we derive that the pointed forest $(\mathcal{F}',\mathsf{t},\mathsf{n})$ encodes the suffix of the word $\mathfrak{f}(\mathfrak{w})$ corresponding to the nodes $\mathsf{n}_{j+1}\dots\mathsf{n}_m$ (which characterises the main path of $\mathcal{F}'$). These nodes are ancestors of n and, from (B), at least β of them encode marked symbols (w.r.t. both $\mathcal{F}'$ and $\mathcal{F}$, as the former structure encodes a suffix of the word encoded by $\mathcal{F}$). So, n has at least β ancestors encoding marked symbols of $\Sigma_\bullet$. □

Lemma 25. Let $\mathfrak{w} \in \Sigma_\bullet^+$ ba a marked word with $\beta \geq 1$ marked symbols. Let $(\mathcal{F},\mathsf{t},\mathsf{n})$ be an encoding of $\mathfrak{f}(\mathfrak{w})$. For every φ in PITL, $\mathfrak{w} \models_\bullet \varphi$ if and only if $(\mathcal{F},\mathsf{t},\mathsf{n}) \models \tau_\beta(\varphi)$.

Proof. We conclude the proof of Lemma 25 we began in the body or the paper. Recall that we are working under induction hypothesis stating that the statement in the lemma holds for every strict subformula of φ. Let $\mathfrak{w} = \mathsf{a}_1\dots\mathsf{a}_k \in \Sigma_\bullet^+$ with β marked symbols. According to Definition 3, let $\mathbb{M} = (\mathsf{n}_1,\dots,\mathsf{n}_k)$ be the tuple of main nodes of $(\mathcal{F},\mathsf{t},\mathsf{n})$.

Consider the inductive step where $\varphi = \varphi_1 \mid \varphi_2$. In order to conclude the proof, it suffices to show the following three statements:

A. there is $\mathsf{b} \in \Sigma$ such that $\mathfrak{w}' = \epsilon$, $\mathsf{b} = \mathsf{a}$ and $\overline{\mathsf{a}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \wedge \varphi_2$, if and only if

$$(\mathcal{F},\mathsf{t},\mathsf{n}) \models \langle \mathrm{U}\rangle \big(\mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma \wedge \tau_\beta(\varphi_1) \wedge \tau_\beta(\varphi_2)\big),$$

B. there are $\mathsf{b} \in \Sigma$ and $\mathfrak{w}_2 \in \Sigma^*$ s.t. $\mathfrak{w}' = \mathsf{b}\mathfrak{w}_2$, $\overline{\mathsf{b}}\,\mathfrak{w}_2\,\overline{\mathsf{a}}\,\mathfrak{w}'' \models \varphi_1$ and $\mathsf{b}\mathfrak{w}_2\,\overline{\mathsf{a}}\,\mathfrak{w}'' \models \varphi_2$, iff

$$(\mathcal{F},\mathsf{t},\mathsf{n}) \models \langle \mathrm{U}\rangle \big(\mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \neg\mathtt{mark}_\Sigma \wedge \blacklozenge(\mathtt{mark}_\Sigma \wedge \tau_{\beta+1}(\varphi_1)) \wedge \tau_\beta(\varphi_2)\big),$$

C. there is $\mathsf{b} \in \Sigma$ such that $\mathfrak{w}' \neq \epsilon$, $\mathsf{b} = \mathsf{a}$, $\mathfrak{w}'\,\overline{\mathsf{a}}\,\mathfrak{w}'' \models \varphi_1$ and $\overline{\mathsf{a}}\,\mathfrak{w}'' \models \varphi_2$, if and only if

$$(\mathcal{F},\mathsf{t},\mathsf{n}) \models \langle \mathrm{U}\rangle \big(\mathtt{symb} \wedge \neg\mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma \wedge \#\mathtt{markAnc}_\Sigma \geq \beta{-}1 \\ \wedge \tau_\beta(\varphi_1) \wedge \blacklozenge(\mathtt{1st}_{[1,2n]} \wedge \tau_\beta(\varphi_2))\big).$$

proof of (A)**:** Let us consider the first double implication.

there is $\mathsf{b} \in \Sigma$ such that $\mathfrak{w}' = \epsilon$ and $\overline{\mathsf{a}}\,\mathfrak{w}'' \models_\bullet \varphi_1 \wedge \varphi_2$,
$\Leftrightarrow$ $\mathfrak{w}$ is headed by the marked symbol $\overline{\mathsf{a}}$ and $\mathfrak{w} \models_\bullet \varphi_1 \wedge \varphi_2$,
$\Leftrightarrow$ 1. n_1 encodes a marked symbol, i.e. $(\mathcal{F},\mathsf{t},\mathsf{n}_1) \models \mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma$,
(*by definition of* $\mathcal{F}$ *and Lemma 24. Note:* n_1 *always satisfies* $\mathtt{symb} \wedge \mathtt{1st}_{[1,2n]}$)

2. $(\mathcal{F}, \mathtt{t}, \mathtt{n}_1) \models \tau_\beta(\varphi_1) \wedge \tau_\beta(\varphi_2)$,
   (*by induction hypothesis*)

$\Leftrightarrow$ $(\mathcal{F}, \mathtt{t}, \mathtt{n}_1) \models \mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma \wedge \tau_\beta(\varphi_1) \wedge \tau_\beta(\varphi_2)$
(*by definition of* $\models$)

$\Leftrightarrow$ $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \langle \mathrm{U} \rangle \big(\mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma \wedge \tau_\beta(\varphi_1) \wedge \tau_\beta(\varphi_2)\big)$.
(*by semantics of* $\langle \mathrm{U} \rangle$ *and def. of* $\mathtt{n}_1$)

proof of (B): For the second double implication,

there are $\mathtt{b} \in \Sigma$ and $\mathfrak{w}_2 \in \Sigma^*$ s.t. $\mathfrak{w}' = \mathtt{b}\mathfrak{w}_2$, $\overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_1$ and $\mathtt{b}\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models_\bullet \varphi_2$,

$\Leftrightarrow$ $\mathfrak{w}$ is headed by a (non marked) symbol in Σ, $\mathfrak{w} \models_\bullet \varphi_2$ and the word obtained from $\mathfrak{w}$ by marking the first symbol satisfies φ_1,

$\Leftrightarrow$
1. $\mathtt{n}_1$ does not encode a marked symbol, i.e. $(\mathcal{F}, \mathtt{t}, \mathtt{n}_1) \models \mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \neg \mathtt{mark}_\Sigma$,
   (*by definition of* $\mathcal{F}$ *and Lemma 24. Note:* $\mathtt{n}_1$ *satisfies* $\mathtt{symb} \wedge \mathtt{1st}_{[1,2n]}$)
2. $(\mathcal{F}, \mathtt{t}, \mathtt{n}_1) \models \tau_\beta(\varphi_2)$,
   (*by induction hypothesis form* $\mathfrak{w} \models \varphi_2$)
3. There is $\mathcal{F}' \subseteq \mathcal{F}$ s.t. $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$, $(\mathcal{F}', \mathtt{t}, \mathtt{n}_1) \models \mathtt{mark}_\Sigma \wedge \tau_{\beta+1}(\varphi_1)$.
   For this step, consider the finite forest $\mathcal{F}' \subseteq \mathcal{F}$ obtained from $\mathcal{F}$ by removing one child of $\mathtt{n}_1$. As $\mathtt{n}_1$ encodes the non marked symbol $\mathtt{b}$ w.r.t. $\mathcal{F}$, by definition it encodes the marked symbol $\overline{\mathtt{b}}$ w.r.t. $\mathcal{F}'$. Every other node has the same number of $\mathcal{F}'$-children as in $\mathcal{F}$. In other words, $\mathcal{F}'$ encodes the word $\overline{\mathtt{b}}\,\mathfrak{w}_2\,\overline{\mathtt{a}}\,\mathfrak{w}''$, with $\beta+1$ marked symbols, obtained from $\mathfrak{w}$ by marking the first (non marked) symbol. By definition of $\mathcal{F}'$, $(\mathcal{F}', \mathtt{t}, \mathtt{n}_1) \models \mathtt{mark}_\Sigma$. By induction hypothesis, $(\mathcal{F}', \mathtt{t}, \mathtt{n}_1) \models \tau_{\beta+1}(\varphi_1)$.

$\Leftrightarrow$ $(\mathcal{F}, \mathtt{t}, \mathtt{n}_1) \models \mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \neg \mathtt{mark}_\Sigma \wedge \blacklozenge(\mathtt{mark}_\Sigma \wedge \tau_{\beta+1}(\varphi_1)) \wedge \tau_\beta(\varphi_2)$,
(*by definition of* $\mathcal{F}' \subseteq \mathcal{F}$ *and* $\blacklozenge$)

$\Leftrightarrow$ $(\mathcal{F}, \mathtt{t}, \mathtt{n}) \models \langle \mathrm{U} \rangle \big(\mathtt{symb} \wedge \mathtt{1st}_{[1,2n]} \wedge \neg \mathtt{mark}_\Sigma \wedge \blacklozenge(\mathtt{mark}_\Sigma \wedge \tau_{\beta+1}(\varphi_1)) \wedge \tau_\beta(\varphi_2)\big)$.
(*by semantics of* $\langle \mathrm{U} \rangle$ *and def. of* $\mathtt{n}_1$)

proof of (C): For the third double implication,

there is $\mathtt{b} \in \Sigma$ such that $\mathfrak{w}' \neq \epsilon$, $\mathfrak{w}'\,\overline{\mathtt{a}}\,\mathfrak{w}'' \models \varphi_1$ and $\overline{\mathtt{a}}\,\mathfrak{w}'' \models \varphi_2$,

$\Leftrightarrow$ there is an index $j \in [2, k]$ (recall $\mathfrak{w} = \mathtt{a}_1 \dots \mathtt{a}_k$ and $\Delta(\mathfrak{w}) = (\mathfrak{w}', \overline{\mathtt{a}}, \mathfrak{w}'')$) s.t.
1. the main node $\mathtt{n}_j$ encodes a marked symbol, and among the main nodes in the set $\{\mathtt{n}_{j+1}, \dots, \mathtt{n}_k\}$ (ancestors of $\mathtt{n}_j$), exactly $\beta-1$ encode marked symbols. Equivalently, $(\mathcal{F}, \mathtt{t}, \mathtt{n}_j) \models \mathtt{symb} \wedge \neg \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_\Sigma \wedge \#\mathtt{markAnc}_\Sigma \geq \beta-1$,
   (*by definition of* $\mathcal{F}$, *Lemma 24 and as* $j > 1$)
2. $(\mathcal{F}, \mathtt{t}, \mathtt{n}_j) \models \tau_\beta(\varphi_1)$,
   (*by induction hypothesis from* $\mathfrak{w} \models \varphi_1$)
3. There is $\mathcal{F}' \subseteq \mathcal{F}$ s.t. $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$, $(\mathcal{F}', \mathtt{t}, \mathtt{n}_j) \models \mathtt{1st}_{[1,2n]} \wedge \tau_\beta(\varphi_2)$.
   For this step, consider the finite forest $\mathcal{F}' \subseteq \mathcal{F}$ obtained from $\mathcal{F}$ by removing $\mathtt{n}_{j-1}$, i.e. the only main node such that $\mathcal{F}(\mathtt{n}_{j-1}) = \mathtt{n}_j$ (which exists as $j > 1$). It is quite straightforward to see that $\mathcal{F}'$ encodes the word $\mathtt{a}_j\mathtt{a}_{j+1} \dots \mathtt{a}_k = \overline{\mathtt{a}}\,\mathfrak{w}''$. By definition, $(\mathcal{F}', \mathtt{t}, \mathtt{n}_j) \models \mathtt{1st}_{[1,2n]}$. By induction hypothesis, $(\mathcal{F}', \mathtt{t}, \mathtt{n}_j) \models \tau_\beta(\varphi_2)$,

$\Leftrightarrow$ there is a main node n_j in the main path of $\mathcal{F}$ such that

$$(\mathcal{F}, \mathsf{t}, \mathsf{n}_j) \models \mathtt{symb} \wedge \neg \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_{\Sigma} \wedge \#\mathtt{markAnc}_{\Sigma} \geq \beta - 1 \\ \wedge \tau_\beta(\varphi_1) \wedge \blacklozenge(\mathtt{1st}_{[1,2n]} \wedge \tau_\beta(\varphi_2))$$

(*by definition of* $\mathcal{F}' \subseteq \mathcal{F}$ *and* $\blacklozenge$)

$\Leftrightarrow$ $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathrm{U} \rangle \big(\mathtt{symb} \wedge \neg \mathtt{1st}_{[1,2n]} \wedge \mathtt{mark}_{\Sigma} \wedge \#\mathtt{markAnc}_{\Sigma} \geq \beta - 1 \\ \wedge \tau_\beta(\varphi_1) \wedge \blacklozenge(\mathtt{1st}_{[1,2n]} \wedge \tau_\beta(\varphi_2))\big)$.

(*by semantics of* $\langle \mathrm{U} \rangle$) □

Appendix C. Missing proofs from Section 5

Lemma 29. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest, and let s be a store such that $s(\mathtt{v}) = \mathsf{t}$ and $s(\mathtt{u}) = \mathsf{n}$. $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$ if and only if $(s, \mathcal{F}) \models \tau_{\mathtt{v}}(\varphi)$.

Proof. By structural induction on φ.

base case: $\varphi = \mathtt{Hit}$**.**

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{Hit}$,

$\Leftrightarrow$ there is $\delta \geq 1$ such that $\mathcal{F}^\delta(\mathsf{n}) = \mathsf{t}$,

(*by definition of* $\models$)

$\Leftrightarrow$ there is $\delta \geq 1$ such that $\mathcal{F}^\delta(s(\mathtt{u})) = s(\mathtt{v})$,

(*by definition of* s)

$\Leftrightarrow$ $(s, \mathcal{F}) \models \mathtt{u} \hookrightarrow^+ \mathtt{v}$.

(*definition of* $\models$)

base case: $\varphi = \mathtt{Miss}$**.**

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \mathtt{Miss}$,

$\Leftrightarrow$ $\mathsf{n} \in \mathsf{dom}(\mathcal{F})$ and $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \not\models \mathtt{Hit}$,

(*by definition of* $\models$)

$\Leftrightarrow$ $s(\mathtt{u}) \in \mathsf{dom}(\mathcal{F})$ and $(s, \mathcal{F}) \not\models \tau_{\mathtt{v}}(\mathtt{Hit})$,

(*by definition of* s *and from the previous base case*)

$\Leftrightarrow$ $(s, \mathcal{F}) \models \mathtt{u} \hookrightarrow \_ \wedge \neg \tau_{\mathtt{v}}(\mathtt{Hit})$.

(*definition of* $\mathtt{u} \hookrightarrow \_$ *and* $\models$)

The cases for $\top$ and Boolean connectives are trivial, and hence omitted.

induction step: $\varphi = \langle \mathrm{U} \rangle \, \psi$**.**

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathrm{U} \rangle \, \psi$,

$\Leftrightarrow$ there is $\mathsf{n}' \in \mathcal{N}$ such that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \psi$,

(*by definition of* $\models$)

$\Leftrightarrow$ there is $\mathsf{n}' \in \mathcal{N}$ such that $(s[\mathtt{u} \leftarrow \mathsf{n}'], \mathcal{F}) \models \tau_{\mathtt{v}}(\psi)$,
(*by definition of s and by induction hypothesis*)

$\Leftrightarrow$ $(s, \mathcal{F}) \models \exists \mathtt{u}\, \tau_{\mathtt{v}}(\psi)$.
(*definition of $\models$*)

induction step: $\varphi = \blacklozenge\, \psi$**.**

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge\, \psi$,

$\Leftrightarrow$ there is $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{card}(\mathsf{dom}(\mathcal{F}')) + 1 = \mathsf{card}(\mathsf{dom}(\mathcal{F}))$ and $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \psi$,
(*by definition of $\models$*)

$\Leftrightarrow$ there is $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{card}(\mathsf{dom}(\mathcal{F}')) + 1 = \mathsf{card}(\mathsf{dom}(\mathcal{F}))$ and $(s, \mathcal{F}') \models \tau_{\mathtt{v}}(\psi)$,
(*by induction hypothesis*)

$\Leftrightarrow$ there are $\mathcal{F}_1$ and $\mathcal{F}_2$ s.t. $\mathcal{F}_1 \bot \mathcal{F}_2$, $\mathcal{F}_1 + \mathcal{F}_2 = \mathcal{F}$, $\mathsf{card}(\mathsf{dom}(\mathcal{F}_1)) = 1$ and $(s, \mathcal{F}_2) \models \tau_{\mathtt{v}}(\psi)$,
(*by definition of $+$*)

$\Leftrightarrow$ $(s, \mathcal{F}) \models \blacklozenge_{\mathsf{SL}} \tau_{\mathtt{v}}(\psi)$.
(*definition of $\models$*)

induction step: $\varphi = \blacklozenge^*\, \psi$**.**

$(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge^*\, \psi$,

$\Leftrightarrow$ there is $\mathcal{F}' \subseteq \mathcal{F}$ such that $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \psi$,
(*by definition of $\models$*)

$\Leftrightarrow$ there is $\mathcal{F}' \subseteq \mathcal{F}$ such that $(s, \mathcal{F}') \models \tau_{\mathtt{v}}(\psi)$,
(*by induction hypothesis*)

$\Leftrightarrow$ there are $\mathcal{F}_1$ and $\mathcal{F}_2$ such that $\mathcal{F}_1 \bot \mathcal{F}_2$, $\mathcal{F}_1 + \mathcal{F}_2 = \mathcal{F}$ and $(s, \mathcal{F}_2) \models \tau_{\mathtt{v}}(\psi)$,
(*by definition of $+$*)

$\Leftrightarrow$ $(s, \mathcal{F}) \models \blacklozenge^*_{\mathsf{SL}}\, \tau_{\mathtt{v}}(\psi)$.
(*definition of $\models$*) □

Lemma 42. Let $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ be a pointed forest such that $\mathsf{t} \notin \mathsf{dom}(\mathcal{F})$, and let $(\mathcal{K}, \mathsf{w})$ be a $(S, \boldsymbol{u})$-encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n})$. Given a formula φ in ALT, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \varphi$ if and only if $(\mathcal{K}, \mathsf{w}) \models \tau_{\boldsymbol{u}}(\varphi)$.

Proof. Let $\mathcal{K} = (\mathcal{W}, R, \mathcal{V})$. Let $\mathfrak{f} : \mathcal{N} \to R^*(\mathsf{w})$ be a bijection witnessing that $(\mathcal{K}, \mathsf{w})$ is a $(S, \boldsymbol{u})$ encoding of $(\mathcal{F}, \mathsf{n}, \mathsf{t})$ ($\boldsymbol{u} \in \{\boldsymbol{D}, \boldsymbol{E}\}$). We recall that this means that:

$1_{\mathfrak{f}}$. $\mathfrak{f}(\mathsf{t}) \stackrel{\mathsf{def}}{=} \mathsf{w}$ is the only world in $\mathsf{ran}(\mathfrak{f}) \cap \mathcal{V}(\boldsymbol{t})$, and $\mathfrak{f}(\mathsf{n})$ is the only world in $\mathsf{ran}(\mathfrak{f}) \cap \mathcal{V}(\boldsymbol{n})$,

$2_{\mathfrak{f}}$. for every $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$ it holds that $(\mathfrak{f}(\mathcal{F}(\mathsf{n}')), \mathfrak{f}(\mathsf{n}')) \in R$,

$3_{\mathfrak{f}}$. for every infinite path $(\mathsf{w}_0, \mathsf{w}_1 \dots) \in \Pi_R(\mathsf{w})$ there is $i \in \mathbb{N}$ such that

a. $\mathsf{w}_i \in \mathcal{V}(\boldsymbol{end})$ and for every $j \in [0, i-1]$ we have $\mathsf{w}_j \notin \mathcal{V}(\boldsymbol{end})$,

b. for every $j \in \mathbb{N}$, ($\mathsf{w}_j \in \mathcal{V}(\boldsymbol{u})$ and $j < i$) if and only if there is $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$ $\mathfrak{f}(\mathsf{n}') = \mathsf{w}_j$.

Below, we call the three properties $(1_\mathfrak{f})$–$(3_\mathfrak{f})$ *hypotheses* whenever they refer to $(\mathcal{K}, \mathsf{w})$ and $(\mathcal{F}, \mathsf{n}, \mathsf{t})$. Instead, we call them *properties* when referring to their analogue on other two structures, for which we want to prove their satisfaction. The proof is by structural induction on φ.

base case: $\varphi = \mathtt{Hit}$**.**

$(\mathcal{F}, \mathsf{n}, \mathsf{t}) \models \mathtt{Hit}$

$\Leftrightarrow$ there is $k \geq 1$ and there are $k+1$ different nodes $\mathsf{n}_0, \mathsf{n}_1, \ldots, \mathsf{n}_k$ such that $\mathsf{n}_k = \mathsf{n}$, $\mathsf{n}_0 = \mathsf{t}$ and for every $i \in [0, k-1]$, $\mathcal{F}(\mathsf{n}_{i+1}) = \mathsf{n}_i$,

(*by definition of* $\models$)

$\Leftrightarrow$ there are $k \geq 1$ and $k+1$ different worlds $\mathsf{w}_0 = \mathfrak{f}(\mathsf{n}_0), \mathsf{w}_1 = \mathfrak{f}(\mathsf{n}_1), \ldots, \mathsf{w}_k = \mathfrak{f}(\mathsf{n}_k)$ s.t.

1. from hypothesis $(1_\mathfrak{f})$, $\mathsf{w}_k = \mathfrak{f}(\mathsf{n})$, $\{\mathsf{w}_k\} = \mathcal{V}(\boldsymbol{n})$, and $\mathsf{w}_0 = \mathfrak{f}(\mathsf{t})$, $\{\mathsf{w}_0\} = \mathcal{V}(\boldsymbol{t})$,
2. from hypothesis $(2_\mathfrak{f})$, for every $j \in [0, k-1]$, $(\mathsf{w}_j, \mathsf{w}_{j+1}) \in R$,
3. from hypothesis $(3_\mathfrak{f})$, $\mathsf{w}_0 \notin \mathcal{V}(\boldsymbol{u})$, $\{\mathsf{w}_1, \ldots, \mathsf{w}_k\} \subseteq \mathcal{V}(\boldsymbol{u})$, $\{\mathsf{w}_0, \ldots, \mathsf{w}_k\} \cap \mathcal{V}(\boldsymbol{end}) = \varnothing$.

$\Leftrightarrow$ there is a path $(\mathsf{w}_0, \mathsf{w}_1, \ldots, \mathsf{w}_k)$ in $\mathcal{K}$ such that

1. $\{\mathsf{w}_k\} = \mathcal{V}(\boldsymbol{n})$, $\mathsf{w}_k \in \mathcal{V}(\boldsymbol{u})$,
2. $\mathsf{w} = \mathsf{w}_0 \notin \mathcal{V}(\boldsymbol{u})$,
3. for every $j \in [0, k-1]$, ($\mathsf{w}_j \in \mathcal{V}(\boldsymbol{u})$ or $\mathsf{w}_j \in \mathcal{V}(\boldsymbol{t})$) and $\mathsf{w}_j \notin \mathcal{V}(\boldsymbol{end})$,

(*manipulation from last step, and the hypothesis of* $\mathfrak{f}$*)*

$\Leftrightarrow$ $(\mathcal{K}, \mathsf{w}) \models \mathsf{E}((\boldsymbol{u} \vee \boldsymbol{t}) \wedge \neg \boldsymbol{end}\ \mathsf{M}\ \boldsymbol{u} \wedge \boldsymbol{n})$

(*by definition of* $\models$).

base case: $\varphi = \mathtt{Miss}$**.**

$(\mathcal{F}, \mathsf{n}, \mathsf{t}) \models \mathtt{Miss}$

$\Leftrightarrow$ $\mathsf{n} \in \mathsf{dom}(\mathsf{t})$ and $(\mathcal{F}, \mathsf{n}, \mathsf{t}) \not\models \mathtt{Miss}$ (by definition of $\models$)

$\Leftrightarrow$ there is a path $(\mathsf{w}_0, \mathsf{w}_1, \ldots, \mathsf{w}_k)$ in $\mathcal{K}$ (*by hypothesis* $(2_\mathfrak{f})$) such that

1. from hypothesis $(1_\mathfrak{f})$ and $(3_\mathfrak{f})$ $\mathsf{w}_0 = \mathsf{w}$, $\mathsf{w}_k \in \mathcal{V}(\boldsymbol{n})$, $\mathsf{w}_k \notin \mathcal{V}(\boldsymbol{end})$ and $\mathsf{w}_k \in \mathcal{V}(\boldsymbol{u})$,
2. from hypothesis $(3_\mathfrak{f})$, for every $j \in [0, k-1]$ $\mathsf{w}_j \notin \mathcal{V}(\boldsymbol{end})$,
3. $(\mathcal{K}, \mathsf{w}) \not\models \tau_{\boldsymbol{u}}(\mathtt{Hit})$, (*from the previous base case*)

$\Leftrightarrow$ $(\mathcal{K}, \mathsf{w}) \models \mathsf{E}(\neg \boldsymbol{end}\ \mathsf{M}\ \boldsymbol{u} \wedge \boldsymbol{n}) \wedge \neg \tau_{\boldsymbol{u}}(\mathtt{Hit})$.

(*by definition of* $\models$)

We omit the obvious cases for $\top$ and Boolean connectives.

induction step: $\varphi = \langle \mathsf{U} \rangle\, \psi$**.** ($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathsf{U} \rangle\, \psi$, and so there is $\mathsf{n}' \in \mathcal{N}$ such that $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \psi$. Let us consider the Kripke tree $\mathcal{K}' = (\mathcal{W}, R, \mathcal{V}[\boldsymbol{n} \leftarrow \{\mathfrak{f}(\mathsf{n}')\}])$ obtained from $\mathcal{K}$ by updating the evaluation of $\boldsymbol{n}$ from $\{\mathfrak{f}(\mathsf{n})\}$ to $\{\mathfrak{f}(\mathsf{n}')\}$ (see hypothesis $(1_\mathfrak{f})$ for $\mathcal{V}(\boldsymbol{n}) = \{\mathfrak{f}(\mathsf{n})\}$). It is easy to verify that $(\mathcal{K}', \mathsf{w})$ is a $(S, \boldsymbol{u})$-encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n}')$. Indeed, it is quite obvious that the same bijection $\mathfrak{f} : \mathcal{N} \to R^*(\mathsf{w})$ considered for the two structures $(\mathcal{F}, \mathsf{t}, \mathsf{n})$ and $(\mathcal{K}, \mathsf{w})$ satisfy the properties $(1_\mathfrak{f})$–$(3_\mathfrak{f})$ also with respect to the two structures $(\mathcal{K}', \mathsf{w})$ and $(\mathcal{F}, \mathsf{t}, \mathsf{n}')$. By definition of $\mathcal{K}'$, clearly $(\mathcal{K}', \mathsf{w}) \models \mathtt{uniq}(\boldsymbol{n})$. By induction hypothesis, $(\mathcal{K}', \mathsf{w}) \models \tau_{\boldsymbol{u}}(\psi)$. Lastly, from the semantics of $\exists \boldsymbol{n}$, we conclude that $(\mathcal{K}, \mathsf{w}) \models \exists \boldsymbol{n}\ (\mathtt{uniq}(\boldsymbol{n}) \wedge \tau_{\boldsymbol{u}}(\psi))$.

($\Leftarrow$): The other direction follows with similar arguments (backwards). Briefly, suppose $(\mathcal{K}, \mathsf{w}) \models \exists \boldsymbol{n}\ (\mathtt{uniq}(\boldsymbol{n}) \wedge \tau_{\boldsymbol{u}}(\psi))$. Then there is $\mathcal{K}' = (\mathcal{W}, R, \mathcal{V}[\boldsymbol{n} \leftarrow \mathcal{W}'])$, for some $\mathcal{W}' \subseteq \mathcal{W}$ such that $(\mathcal{K}', \mathsf{w}) \models \mathtt{uniq}(\boldsymbol{n}) \wedge \tau_{\boldsymbol{u}}(\varphi)$. From $\mathtt{uniq}(\boldsymbol{n})$ we conclude that there is a world $\mathsf{w}' \in R^*(\mathsf{w})$ such that $\mathcal{V}(\boldsymbol{n}) = \{\mathsf{w}'\}$. Let n' be the node such that $\mathfrak{f}(\mathsf{n}') = \mathsf{w}'$ (recall that $\mathfrak{f}$ is bijective). It is quite easy to see that $(\mathcal{K}', \mathsf{w})$ is an $(S, \boldsymbol{u})$-encoding of $(\mathcal{F}, \mathsf{t}, \mathsf{n}')$. Again, $\mathfrak{f}$ is a witness of this encoding. By induction hypothesis $(\mathcal{F}, \mathsf{t}, \mathsf{n}') \models \psi$. Thus, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \langle \mathrm{U} \rangle\, \psi$.

induction step: $\varphi = \blacklozenge\, \psi$**.** ($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge\, \psi$, and so there is a subforest $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$ and $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \psi$. Let $\widehat{\mathsf{n}}$ be the (only) node such that $\mathsf{dom}(\mathcal{F}) = \mathsf{dom}(\mathcal{F}') \cup \{\widehat{\mathsf{n}}\}$. Notice that $\widehat{\mathsf{n}} \in \mathsf{dom}(\mathcal{F})$ and hence, by hypothesis ($2_{\mathfrak{f}}$) and ($3_{\mathfrak{f}}$):

A. $\mathfrak{f}(\widehat{\mathsf{n}}) \in \mathcal{V}(\boldsymbol{u})$;

B. There is a path $(\mathsf{w}_0, \ldots, \mathsf{w}_k)$ in $(\mathcal{K}, \mathsf{w})$ going from $\mathsf{w}_0 = \mathsf{w}$ to $\mathsf{w}_k = \mathfrak{f}(\widehat{\mathsf{n}})$. Moreover, for every $j \in [0, k]$ we have $\mathsf{w}_j \notin \mathcal{V}(\boldsymbol{end})$.

Let us consider the Kripke tree $\mathcal{K}' = (\mathcal{W}, R, \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'])$ where $\mathcal{W}' \stackrel{\text{def}}{=} \mathcal{V}(\boldsymbol{u}) \setminus \{\mathfrak{f}(\widehat{\mathsf{n}})\}$. Notice that then $\mathcal{W}' \subseteq \mathcal{V}(\boldsymbol{u})$ and from (A) we have $\mathsf{card}(\mathcal{W}') = \mathsf{card}(\mathcal{V}(\boldsymbol{u})) - 1$. Thus, $(\mathcal{K}', \mathsf{w}) \models \mathsf{AG}\,(\overline{\boldsymbol{u}} \Rightarrow \boldsymbol{u}) \wedge \mathtt{uniq}(\boldsymbol{u} \wedge \neg\, \overline{\boldsymbol{u}})$. Furthermore, from (B) we conclude that $(\mathcal{K}', \mathsf{w})$ also satisfies $\mathsf{E}(\neg \boldsymbol{end}\ \mathsf{M}\ \boldsymbol{u} \wedge \neg\, \overline{\boldsymbol{u}})$. It remains to show that $(\mathcal{K}', \mathsf{w}) \models \tau_{\overline{\boldsymbol{u}}}(\psi)$, which follows by induction hypothesis, as we show that $(\mathcal{K}', \mathsf{w})$ is a $(S, \overline{\boldsymbol{u}})$-encoding of $(\mathcal{F}', \mathsf{t}, \mathsf{n})$ (notice that now the encoding uses $\overline{\boldsymbol{u}}$ instead of $\boldsymbol{u}$). More precisely, it is sufficient to check that the bijection $\mathfrak{f} : \mathcal{N} \to R^*(\mathsf{w})$ satisfies the properties ($1_{\mathfrak{f}}$)–($3_{\mathfrak{f}}$) for the two structures $(\mathcal{K}', \mathsf{w})$ and $(\mathcal{F}', \mathsf{t}, \mathsf{n})$. We show the three properties separately.

Proof of property ($1_{\mathfrak{f}}$)**.** We show that

- $\mathfrak{f}(\mathsf{t}) \stackrel{\text{def}}{=} \mathsf{w}$ is the only element in $\mathsf{ran}(\mathfrak{f})$ such that $\mathsf{w} \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\boldsymbol{t}\,)$, and
- $\mathfrak{f}(\mathsf{n})$ is the only element in $\mathsf{ran}(\mathfrak{f})$ such that $\mathfrak{f}(\mathsf{n}) \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\boldsymbol{n}\,)$.

Clearly, both statements hold directly from hypothesis ($1_{\mathfrak{f}}$). Indeed, $\mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}']$ only updates the evaluation of $\overline{\boldsymbol{u}}$, so $\mathcal{V}(\boldsymbol{n}\,) = \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\boldsymbol{n}\,)$ and $\mathcal{V}(\boldsymbol{t}\,) = \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\boldsymbol{t}\,)$.

Proof of property ($2_{\mathfrak{f}}$)**.** We show that $(\mathfrak{f}(\mathcal{F}(\mathsf{n}')), \mathfrak{f}(\mathsf{n}')) \in R$ holds for every $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}')$. This holds directly from $\mathcal{F}' \subseteq \mathcal{F}$ and hypothesis ($2_{\mathfrak{f}}$).

Proof of property ($3_{\mathfrak{f}}$)**.** We show that for every infinite path $(\mathsf{w}_0, \mathsf{w}_1 \ldots) \in \Pi_R(\mathsf{w})$ there is $i \in \mathbb{N}$ such that

1. $\mathsf{w}_i \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\boldsymbol{end}\,)$ and for every $j \in [0, i-1]$ we have $\mathsf{w}_j \notin \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\boldsymbol{end}\,)$,
2. for all $j \in \mathbb{N}$, ($\mathsf{w}_j \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\overline{\boldsymbol{u}})$ and $j < i$) iff $\mathfrak{f}(\mathsf{n}') = \mathsf{w}_j$ for some $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}')$.

From $\mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\boldsymbol{end}\,) = \mathcal{V}(\boldsymbol{end}\,)$ together with the hypothesis ($3_{\mathfrak{f}}$)(a) we derive (1). To prove (2), let $(\mathsf{w}_0, \mathsf{w}_1 \ldots) \in \Pi_R(\mathsf{w})$ and $i \in \mathbb{N}$ so that (1) is satisfied. Let $j \in \mathbb{N}$. First, consider $\mathsf{w}_j \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\overline{\boldsymbol{u}})$ and $j < i$. As $\mathcal{W}' = \mathcal{V}(\boldsymbol{u}) \setminus \{\mathfrak{f}(\widehat{\mathsf{n}})\}$, we conclude that $\mathsf{w}_j \in \mathcal{V}(\boldsymbol{u})$ and $\mathsf{w}_j \neq \mathfrak{f}(\widehat{\mathsf{n}})$. From hypothesis ($3_{\mathfrak{f}}$)(b), there is $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$ such that $\mathfrak{f}(\mathsf{n}') = \mathsf{w}_j$. As $\mathsf{w}_j \neq \mathfrak{f}(\widehat{\mathsf{n}})$, $\mathsf{n}' \neq \widehat{\mathsf{n}}$ and so $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}')$. Conversely, suppose that there is $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}')$ such that $\mathfrak{f}(\mathsf{n}') = \mathsf{w}_j$. In particular, $\mathsf{n}' \neq \widehat{\mathsf{n}}$. From ($3_{\mathfrak{f}}$)(b), $\mathsf{w}_j \in \mathcal{V}(\boldsymbol{u})$ and $j < i$. As $\mathcal{W}' = \mathcal{V}(\boldsymbol{u}) \setminus \{\mathfrak{f}(\widehat{\mathsf{n}})\}$, we conclude $\mathsf{w}_j \in \mathcal{W}'$. So, $\mathsf{w}_j \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\overline{\boldsymbol{u}})$.

This concludes the proof that $(\mathcal{K}', \mathsf{w})$ is an $(S, \overline{\boldsymbol{u}})$-encoding of $(\mathcal{F}', \mathsf{t}, \mathsf{n})$, which allows us to conclude that $(\mathcal{K}, \mathsf{w}) \models \tau_{\boldsymbol{u}}(\blacklozenge\, \psi)$ from the semantics of the modality $\blacklozenge$.

($\Leftarrow$): For the converse direction, let us assume that

$$(\mathcal{K}, \mathsf{w}) \models \exists\, \overline{\boldsymbol{u}}\ (\mathsf{AG}\,(\overline{\boldsymbol{u}} \Rightarrow \boldsymbol{u}) \wedge \mathtt{uniq}(\boldsymbol{u} \wedge \neg\, \overline{\boldsymbol{u}}) \wedge \mathsf{E}(\neg \mathit{end}\ \mathsf{M}\ \boldsymbol{u} \wedge \neg\, \overline{\boldsymbol{u}}) \wedge \tau_{\overline{\boldsymbol{u}}}(\psi)).$$

There is $\mathcal{W}' \subseteq \mathcal{W}$ and $\mathcal{K}' = (\mathcal{W}, R, \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'])$ such that

$$(\mathcal{K}', \mathsf{w}) \models \mathsf{AG}\,(\overline{\boldsymbol{u}} \Rightarrow \boldsymbol{u}) \wedge \mathtt{uniq}(\boldsymbol{u} \wedge \neg\, \overline{\boldsymbol{u}}) \wedge \mathsf{E}(\neg \mathit{end}\ \mathsf{M}\ \boldsymbol{u} \wedge \neg\, \overline{\boldsymbol{u}}) \wedge \tau_{\overline{\boldsymbol{u}}}(\psi).$$

By $(\mathcal{K}', \mathsf{w}) \models \mathsf{AG}\,(\overline{\boldsymbol{u}} \Rightarrow \boldsymbol{u}) \wedge \mathtt{uniq}(\boldsymbol{u} \wedge \neg\, \overline{\boldsymbol{u}})$, there is a world $\widehat{\mathsf{w}} \in R^*(\mathsf{w}) \cap \mathcal{V}(\boldsymbol{u})$ such that

$$(\dagger) \qquad R^*(\mathsf{w}) \cap \mathcal{W}' = R^*(\mathsf{w}) \cap \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\overline{\boldsymbol{u}}) = (R^*(\mathsf{w}) \cap \mathcal{V}(\boldsymbol{u})) \setminus \{\widehat{\mathsf{w}}\}.$$

Moreover, from $(\mathcal{K}', \mathsf{w}) \models \mathsf{E}(\neg \mathit{end}\ \mathsf{M}\ \boldsymbol{u} \wedge \neg\, \overline{\boldsymbol{u}})$, we conclude that the only path $(\mathsf{w}_0, \mathsf{w}_1 \ldots, \mathsf{w}_k)$ going from $\mathsf{w}_0 = \mathsf{w}$ to $\mathsf{w}_k = \widehat{\mathsf{w}}$ is such that for all $i \in [0, k]$, $\mathsf{w}_i \notin \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\mathit{end}) = \mathcal{V}(\mathit{end})$. From hypothesis ($3_\mathfrak{f}$) and $\widehat{\mathsf{w}} \in \mathcal{V}(\boldsymbol{u})$, we conclude that there is a node $\widehat{\mathsf{n}} \in \mathsf{dom}(\mathcal{F})$ such that $\mathfrak{f}(\widehat{\mathsf{n}}) = \widehat{\mathsf{w}}$. Let us then consider the finite forest $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{dom}(\mathcal{F}') = \mathsf{dom}(\mathcal{F}) \setminus \{\widehat{\mathsf{n}}\}$. From ($\dagger$) and the hypothesis ($1_\mathfrak{f}$)–($3_\mathfrak{f}$) we can show that $(\mathcal{K}', \mathsf{w})$ is an $(S, \overline{\boldsymbol{u}})$-encoding of $(\mathcal{F}', \mathsf{t}, \mathsf{n})$. More precisely, $\mathfrak{f}$ is a witness of the encoding between these two structures. Details are omitted, as the proof is analogous to the left-to-right direction. By induction hypothesis, $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \psi$. As $\mathsf{card}(\mathcal{F}') = \mathsf{card}(\mathcal{F}) - 1$ and $\mathcal{F}' \subseteq \mathcal{F}$, $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge\, \psi$.

induction step: $\varphi = \blacklozenge^*\, \psi$**.** This case is very similar to the case for $\varphi = \blacklozenge\, \psi$.

($\Rightarrow$): Suppose $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge^*\, \psi$, and so there is $\mathcal{F}' \subseteq \mathcal{F}$ such that $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \psi$. Let us consider the set $\mathcal{W}' = \{\mathsf{w}' \in \mathcal{W} \mid \text{ there is } \mathsf{n}' \in \mathsf{dom}(\mathcal{F}') \text{ s.t. } \mathfrak{f}(\mathsf{n}') = \mathsf{w}'\}$. Informally, this is the set of worlds that corresponds to nodes in $\mathsf{dom}(\mathcal{F}')$. Consider the Kripke tree $\mathcal{K}' = (\mathcal{W}, R, \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'])$. By hypothesis ($3_\mathfrak{f}$), for every $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$ we have $\mathfrak{f}(\mathsf{n}') \in \mathcal{V}(\boldsymbol{u})$, which in turn implies $\mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\overline{\boldsymbol{u}}) = \mathcal{W}' \subseteq \mathcal{V}(\boldsymbol{u}) = \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\boldsymbol{u})$ (as $\mathcal{F}' \subseteq \mathcal{F}$). This implies that $(\mathcal{K}', \mathsf{w})$ satisfies $\mathsf{AG}\,(\overline{\boldsymbol{u}} \Rightarrow \boldsymbol{u})$. It remains then to show that $(\mathcal{K}', \mathsf{w}) \models \tau_{\overline{\boldsymbol{u}}}(\psi)$, which follows by induction hypothesis, as we show that $(\mathcal{K}', \mathsf{w})$ is a $(S, \overline{\boldsymbol{u}})$-encoding of $(\mathcal{F}', \mathsf{t}, \mathsf{n})$. As in the induction step dealing with the modality $\blacklozenge$, it is sufficient to check that the bijection $\mathfrak{f} : \mathcal{N} \to R^*(\mathsf{w})$ satisfies properties ($1_\mathfrak{f}$)–($3_\mathfrak{f}$) for the two structures $(\mathcal{K}', \mathsf{w})$ and $(\mathcal{F}', \mathsf{t}, \mathsf{n})$. The proof of the properties ($1_\mathfrak{f}$) and ($2_\mathfrak{f}$) is analogous to the one given for the induction step $\varphi = \blacklozenge\, \psi$. Therefore, let us focus on the proof of property ($3_\mathfrak{f}$):

Proof of property ($3_\mathfrak{f}$)**.** We show that for every infinite path $(\mathsf{w}_0, \mathsf{w}_1 \ldots) \in \Pi_R(\mathsf{w})$ there is $i \in \mathbb{N}$ such that

1. $\mathsf{w}_i \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\mathit{end})$ and for every $j \in [0, i-1]$ we have $\mathsf{w}_j \notin \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\mathit{end})$,
2. for all $j \in \mathbb{N}$, $(\mathsf{w}_j \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\overline{\boldsymbol{u}})$ and $j < i)$ iff $\mathfrak{f}(\mathsf{n}') = \mathsf{w}_j$ for some $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}')$.

From $\mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\mathit{end}) = \mathcal{V}(\mathit{end})$ together with the hypothesis ($3_\mathfrak{f}$)(a) we derive (1). To prove (2), let $(\mathsf{w}_0, \mathsf{w}_1 \ldots) \in \Pi_R(\mathsf{w})$ and $i \in \mathbb{N}$ so that (1) is satisfied. Let $j \in \mathbb{N}$. The left-to-right direction of (2) is obvious. Indeed, given $\mathsf{w}_j \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\overline{\boldsymbol{u}}) = \mathcal{W}'$, by definition of $\mathcal{W}'$, there is a node $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}')$ such that $\mathfrak{f}(\mathsf{n}') = \mathsf{w}_j$. For the right-to-left direction, suppose that there is a node $\mathsf{n}' \in \mathsf{dom}(\mathcal{F}')$ such that $\mathfrak{f}(\mathsf{n}') = \mathsf{w}_j$. Clearly, from the definition of $\mathcal{W}'$, $\mathsf{w}_j \in \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\overline{\boldsymbol{u}}) = \mathcal{W}'$. Moreover, $j < i$ directly from the fact that $\mathsf{n}' \in \mathsf{dom}(\mathcal{F})$ (by $\mathcal{F}' \subseteq \mathcal{F}$) and from the hypothesis ($3_\mathfrak{f}$)(b).

This concludes the proof that $(\mathcal{K}', \mathsf{w})$ is an $(S, \overline{\boldsymbol{u}})$-encoding of $(\mathcal{F}', \mathsf{t}, \mathsf{n})$, which allows us to conclude that $(\mathcal{K}, \mathsf{w}) \models \tau_{\boldsymbol{u}}(\blacklozenge^* \psi)$ from the semantics of the modality $\blacklozenge^*$.

($\Leftarrow$): Suppose $(\mathcal{K}, \mathsf{w}) \models \exists\, \overline{\boldsymbol{u}}\, (\mathsf{AG}\,(\overline{\boldsymbol{u}} \Rightarrow \boldsymbol{u}) \wedge \tau_{\overline{\boldsymbol{u}}}(\psi))$, and so there is a $\mathcal{W}' \subseteq \mathcal{W}$ and a Kripke tree $\mathcal{K}' = (\mathcal{W}, R, \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'])$ such that $(\mathcal{K}', \mathsf{w}) \models \mathsf{AG}\,(\overline{\boldsymbol{u}} \Rightarrow \boldsymbol{u}) \wedge \tau_{\overline{\boldsymbol{u}}}(\psi)$. We define $\mathcal{F}' \subseteq \mathcal{F}$ such that $\mathsf{dom}(\mathcal{F}') = \{\mathsf{n} \in \mathsf{dom}(\mathcal{F}) \mid \mathfrak{f}(\mathsf{n}) \in \mathcal{W}'\}$. From $(\mathcal{K}', \mathsf{w}) \models \mathsf{AG}\,(\overline{\boldsymbol{u}} \Rightarrow \boldsymbol{u})$ we derive that $\mathcal{W}' = \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\overline{\boldsymbol{u}}) \subseteq \mathcal{V}[\overline{\boldsymbol{u}} \leftarrow \mathcal{W}'](\boldsymbol{u}) = \mathcal{V}(\boldsymbol{u})$. By hypothesis ($1_\mathfrak{f}$)–($3_\mathfrak{f}$) we can show that $(\mathcal{K}', \mathsf{w})$ is a $(S, \overline{\boldsymbol{u}})$-encoding of $(\mathcal{F}', \mathsf{t}, \mathsf{n})$. Details are omitted, as the proof is analogous to the left-to-right direction. By induction hypothesis, $(\mathcal{F}', \mathsf{t}, \mathsf{n}) \models \psi$ holds. From $\mathcal{F}' \subseteq \mathcal{F}$, we conclude: $(\mathcal{F}, \mathsf{t}, \mathsf{n}) \models \blacklozenge^* \psi$. □