Decidability of Context Logic

Cristiano Calcagno, Thomas Dinsdale-Young, and Philippa Gardner

Department of Computing, Imperial College London, UK
{ccris,td202,pg}@doc.ic.ac.uk

Abstract. We consider the problem of decidability for Context Logic for sequences, ranked trees and unranked trees. We show how to translate quantifier-free formulae into finite automata that accept just the sequences or trees which satisfy the formulae. Satisfiability is thereby reduced to the language emptiness problem for finite automata, which is simply a question of reachability. This reduction shows that Context Logic formulae define languages that are regular; indeed, we show that for sequences they are exactly the star-free regular languages. We also show that satisfiability is still decidable when quantification over context hole labels is added to the logic, by reducing the problem to the quantifier-free case.

Keywords: Context Logic, decidability, trees, automata

1 Introduction

Context Logic (*CL*) [1] was introduced by Calcagno, Gardner and Zarfaty to reason about structured data (*e.g.* trees), in contrast with O'Hearn and Pym's Bunched Logic (*BL*) [2] which reasons about unstructured resource (*e.g.* heaps). Using *CL*, it is possible to provide local Hoare reasoning about tree update, such as specifying and reasoning about the XML update library DOM [3]. This work was inspired by O'Hearn, Reynolds and Yang's previous work on local Hoare reasoning about heap update [4–6], using Separation Logic (*SL*) based on *BL* for heaps. This type of reasoning is not possible using Cardelli and Gordon's Ambient Logic (*AL*) [7], an earlier logic for reasoning about trees. It is made possible by *CL*'s context composition connective [8], which describes a tree in terms of an arbitrary subtree and its surrounding context. Composition has two right adjoints, which describe a tree in terms of the result of inserting it into a certain context and a context in terms of the result of inserting a tree into it.

We are interested in the decidability of model checking, satisfiability and validity for Context Logic for sequences and trees. A model-checking procedure verifies assertions about data: *e.g.*, that an XML document satisfies a schema. A satisfiability procedure checks if any data satisfy an assertion. A validity procedure checks that an assertion is satisfied by all data and can, for instance, play an important part in automatically checking specifications for programs that manipulate tree data. Validity may be expressed in terms of satisfiability: a formula is valid exactly when its negation is unsatisfiable.

With others, Calcagno has already proved the decidability of model checking and satisfiability for *SL* [9] and *AL* [10] without quantification (*i.e.* a constant-only fragment), based on a size argument about heaps and trees. Such a size argument does not work for *CL*, since context composition can be used to explore arbitrarily deeply within a data structure. Dal Zilio *et al* used an approach based on a class of automata to show decidability for *AL* [11]. We also use automata for our results: we show how the connectives of *CL* — in particular, context composition and its adjoints — can be implemented by automaton constructions. These constructed automata accept exactly the sequences or trees which satisfy the corresponding formulae. Model checking a formula against a tree can be decided by simply running the corresponding automaton on that tree. Satisfiability can be decided by determining whether there is a path to an accepting state in the automaton; such a path provides a witness to satisfiability. We present our results for the multi-holed *CL* that we previously studied in [12]; this logic subsumes the original single-holed logic.

Our results also link the expressive power of *CL* with classes of regular languages. Regular languages are exactly those definable by finite automata and so our results immediately imply that *CL* formulae define regular languages. We demonstrate *CL* for sequences corresponds to the star-free regular languages, by appealing to Schützenberger's characterisation of the star-free regular languages as the aperiodic languages [13]. McNaughton and Papert showed that these are exactly the languages definable in First-order Logic (*FOL*) over words [14] (a proof may also be found in [15]). *CL* for ranked trees (via an adjunct-elimination result along the lines of [12]) corresponds to Heuter's special star-free regular languages, which are exactly the tree languages definable in *FOL* over ranked trees [16]. The unranked tree case is less clear. Bojańczyk has shown that the unranked tree languages definable in *FOL* correspond to a form of regular expressions that are equivalent to single-holed *CL* without the structural adjoints [17]. Although we showed in [12] that the structural adjoints can be eliminated from multi-holed *CL*, at present we can only conjecture that single- and multi-holed *CL* are equally expressive without adjoints. This would imply that *CL* corresponds to *FOL* in definability of unranked trees.

The final results of this paper show the decidability of *CL* with quantification over the hole labels, which may only occur linearly; decidability does not hold for quantification over arbitrary node labels, which do not have to be unique. Our key observation is that existential quantification can be encoded using the freshness quantification of Gabbay and Pitts. It is then possible to convert formulae with freshness quantification into a prenex normal form, as in [18], for which decidability reduces to the quantifier-free case.

2 Sequences

In this section, we consider decidability of multi-holed Context Logic for sequences, without quantification, abbreviated CL^{m}_{Seq}. The sequence model for Context Logic considers finite sequences of symbols from alphabet Σ, taken to be finite. In the framework of multi-holed Context Logic, we consider the alphabet to be partitioned into two sets: Υ the set of labels, which is ranged over by a, b, and Ω the set of hole labels, ranged over by x, y. The distinction between the two is that, while labels may be repeated within a sequence, hole labels may occur at most once (linearly) and are treated as context holes into which another sequence may be placed. We distinguish between the words over Σ, in which each hole label may occur arbitrarily often, and sequences, in which the hole labels occur linearly. We are primarily interested in sequences, but automata are defined in terms of words.

Definition 1 (Words). *The set of* words *over $\Sigma = \Upsilon \uplus \Omega$, S_Σ, ranged over by w, is defined inductively as*

$$w ::= \varepsilon \;\big|\; a \;\big|\; x \;\big|\; w_1 \cdot w_2 \qquad\qquad (a \in \Upsilon, x \in \Omega)$$

modulo structural equivalences given by the '$\cdot$' operator being associative with identity ε (the empty word). In other words, S_Σ is Σ^*, the free monoid over Σ.

Definition 2 (Multi-holed Sequence Contexts). *A* multi-holed sequence context *(or simply* sequence*) is a word from S_Σ in which each $x \in \Omega$ occurs* at most once*. The set of multi-holed sequence contexts over Υ and Ω is denoted by $S_{\Upsilon,\Omega}$, and s, s_1, s_2 denote elements of this set. The set of hole labels that occur in s is denoted by $fn(s)$.*

Definition 3 (Context Composition). Context composition *is a set of partial functions indexed by hole labels, $cp_x : S_{\Upsilon,\Omega} \times S_{\Upsilon,\Omega} \rightharpoonup S_{\Upsilon,\Omega}$, defined by*

$$cp_x(s_1, s_2) = \begin{cases} s_1[s_2/x] & \textit{if } x \in fn(s_1) \textit{ and } fn(s_1) \cap fn(s_2) \subseteq \{x\} \\ \text{undefined} & \textit{otherwise.} \end{cases}$$

The notation $s_1 \; ⓧ \; s_2$ is used as an abbreviation of $cp_x(s_1, s_2)$.

We define quantifier-free CL^{m}_{Seq}. The formulae of Context Logic make use of variables which range over hole labels. The variable names are taken from an infinite set of atoms, the set of *hole variables* Θ, ranged over by α, β.

Definition 4 (Formulae). *The set of formulae of CL^{m}_{Seq}, denoted $\mathcal{F}_{Seq}$ and ranged over by P, P_1, P_2, is defined by:*

$$\begin{array}{llll} P ::= & 0 \;\big|\; a \;\big|\; P_1 \cdot P_2 & (a \in \Upsilon) & \textit{sequence formulae} \\ & \alpha \;\big|\; P_1 \circ_\alpha P_2 \;\big|\; P_1 \circ\!\!-^{\exists}_{\alpha} P_2 \;\big|\; P_1 -\!\!\circ^{\exists}_{\alpha} P_2 & (\alpha \in \Theta) & \textit{structural formulae} \\ & \textit{False} \;\big|\; P_1 \Rightarrow P_2 & & \textit{Boolean formulae.} \end{array}$$

The satisfaction relation for CL^{m}_{Seq} describes the satisfaction of a formula by a sequence with respect to an environment that assigns hole labels to the hole variables that occur in the formula.

Definition 5 (Environment). *An* environment *is a finite partial function $\sigma : \Theta \rightharpoonup_{fin} \Omega$ which assigns hole labels to hole variables.*

Definition 6 (Satisfaction Relation). *The satisfaction relation for CL^{m}_{Seq} is defined inductively on the structure of the formulae as follows, where $x = \sigma(\alpha)$:*

$$
\begin{array}{lcl}
s, \sigma \models 0 & \iff & s = \varepsilon \\
s, \sigma \models a & \iff & s = a \\
s, \sigma \models P_1 \cdot P_2 & \iff & \exists s_1, s_2 .\, s = s_1 \cdot s_2 \wedge s_1, \sigma \models P_1 \wedge s_2, \sigma \models P_2 \\
s, \sigma \models \alpha & \iff & s = x \\
s, \sigma \models P_1 \circ_\alpha P_2 & \iff & \exists s_1, s_2 .\, s = s_1 \, ⓧ \, s_2 \wedge s_1, \sigma \models P_1 \wedge s_2, \sigma \models P_2 \\
s, \sigma \models P_1 \circ\!\!-^{\exists}_{\alpha} P_2 & \iff & \exists s_1, s_2 .\, s_2 = s_1 \, ⓧ \, s \wedge s_1, \sigma \models P_1 \wedge s_2, \sigma \models P_2 \\
s, \sigma \models P_1 -\!\!\circ^{\exists}_{\alpha} P_2 & \iff & \exists s_1, s_2 .\, s_2 = s \, ⓧ \, s_1 \wedge s_1, \sigma \models P_1 \wedge s_2, \sigma \models P_2 \\
s, \sigma \not\models \mathit{False} & & \\
s, \sigma \models P_1 \Rightarrow P_2 & \iff & s, \sigma \models P_1 \implies s, \sigma \models P_2 .
\end{array}
$$

The standard Boolean connectives ($\wedge, \vee, \neg, \mathit{True}$) are all derivable from the minimal set provided in our formulation. In other work, we have used the right adjoints of '$\circ$', denoted '$\circ\!\!-$' and '$-\!\!\circ$', as the primitive connectives, and derived the existential variants '$\circ\!\!-^{\exists}$' and '$-\!\!\circ^{\exists}$' in terms of these. Here, we take the opposite approach, defining $P_1 \circ\!\!-_\alpha P_2 \triangleq \neg(P_1 \circ\!\!-^{\exists}_{\alpha} \neg P_2)$ and $P_1 -\!\!\circ_\alpha P_2 \triangleq \neg(P_1 -\!\!\circ^{\exists}_{\alpha} P_2)$. Our choice is motivated by the fact that existential quantification sits more naturally in the framework of non-deterministic automata than universal quantification. The same choice was made in [8] in order to give a normal modal presentation of Context Logic.

2.1 Automata

Given formula P and environment σ, we are interested in the problems of model checking and satisfiability. In terms of $L_{P,\sigma} = \{s \mid s, \sigma \models P\}$, model checking asks whether $s \in L_{P,\sigma}$ and satisfiability asks whether $L_{P,\sigma} \neq \emptyset$.

We shall show that each $L_{P,\sigma}$ is a regular language; that is, a set of words that is recognised by a finite automaton. We show this by constructing automata that correspond to CL^{m}_{Seq}-formulae. That is, for a given P and σ, we define an automaton $\mathcal{A}_{P,\sigma}$ that accepts exactly the language $L_{P,\sigma}$.

First, we define finite automata. We are mainly concerned with non-deterministic finite automata with ε-transitions since their flexibility leads to a more concise presentation of our constructions than other variations would allow. Although our primary interest is in sequences, there is no natural, structural definition of automata for sequences, and so we deal with automata for words.

Definition 7 (ε-NFA). *A* non-deterministic finite automaton with ε-transitions*, abbreviated ε-NFA, is a tuple $\mathcal{A} = (Q, e, \{f^l\}_{l \in \Sigma \uplus \{\varepsilon\}}, A)$ where: Q is the set of states, a finite set; $e \in Q$ is the initial state; for every $l \in \Sigma$, $f^l \subseteq Q \times Q$ is the state transition relation for l; $f^\varepsilon \subseteq Q \times Q$ is the non-consuming state transition relation; and $A \subseteq Q$ is the set of accepting states.*

For a given $q \in Q$, the notation $f^l(q)$ is used for the set $\{q' \mid (q, q') \in f^l\}$. An ε-NFA having $f^\varepsilon = \emptyset$ is a non-deterministic finite automaton*, abbreviated*

NFA. An NFA for which f^l is a partial function for all $l \in \Sigma$ is a deterministic finite automaton*, abbreviated DFA. A* pre-automaton *is an automaton without a set of accepting states,* i.e. $\hat{\mathcal{A}} = (Q, e, \{f^l\}_{l \in \Sigma \uplus \{\varepsilon\}})$.

To formally define the language recognised by an automaton, we make some auxiliary definitions. The ε-$\mathsf{closure}$ of a state is the set of states reachable by any number of ε-transitions. That is, $\varepsilon\text{-}\mathsf{closure} \subseteq Q \times Q$ is the reflexive-transitive closure of f^ε: $\varepsilon\text{-}\mathsf{closure} = (f^\varepsilon)^*$. Each automaton $\mathcal{A}$ induces a function $[\![-]\!]_\mathcal{A} : S_\Sigma \to \mathcal{P}(Q)$ that maps each word to a set of states as follows:[1]

$$
\begin{aligned}
[\![\varepsilon]\!]_\mathcal{A} &= \varepsilon\text{-}\mathsf{closure}(e) \\
[\![w \cdot l]\!]_\mathcal{A} &= \{q \in Q \mid \exists q' \in [\![w]\!]_\mathcal{A}.\, q \in (\varepsilon\text{-}\mathsf{closure} \circ f^l)(q')\}.
\end{aligned}
$$

A word w is said to be *accepted* by $\mathcal{A}$ if $[\![w]\!]_\mathcal{A} \cap A \neq \emptyset$. The language $L_\mathcal{A}$ defined by $\mathcal{A}$ is the set $\{w \in S_\Sigma \mid [\![w]\!]_\mathcal{A} \cap A \neq \emptyset\}$. An automaton $\mathcal{A}$ also induces a function $(\!|-|\!)_\mathcal{A} : S_\Sigma \to \mathcal{P}(Q \times Q)$ that maps each word to a state transition relation as follows:

$$
\begin{aligned}
(\!|\varepsilon|\!)_\mathcal{A} &= \varepsilon\text{-}\mathsf{closure} \\
(\!|w \cdot l|\!)_\mathcal{A} &= \varepsilon\text{-}\mathsf{closure} \circ f^l \circ (\!|w|\!)_\mathcal{A}.
\end{aligned}
$$

The relation $(\!|w|\!)_\mathcal{A}$ captures the behaviour of the automaton $\mathcal{A}$ on reading the word w from any state. This describes the way the automaton interprets the word in any context, and so is useful for constructing automata for CL^{m}_{Seq}-formulae.

With finite automata, language membership and emptiness are decidable. For membership, it is sufficient to consider the runs of an automaton on a given word. For emptiness, it is sufficient to determine whether an accepting state is reachable from the initial state by any combination of transitions. This is effectively an instance of the reachability problem for finite directed graphs.

The class of languages definable by automata is the class of *regular languages*. A full exposition of regular languages and automata may be found in [19].[2] For our purposes, it is important that regular languages include the empty language, all single-element languages and $S_{\Upsilon,\Omega}$, and are closed under union, intersection, complementation with respect to S_Σ, and language concatenation.

We require automaton constructions to implement the structural connectives '$\circ$', '$\circ\!-^{\exists}$' and '$-\!\circ^{\exists}$'. To this end, we generalise context composition to words with non-linear holes. Two possible generalisations given by substituting each x by either the same or different words from some language. The first is non-regular and would lead to undecidability, while the second does not have two right adjoints. Hence, we use non-deterministic linear substitution.

Definition 8 (Non-deterministic Linear Substitution). *Given words w_1, $w_2 \in S_\Sigma$, define the non-deterministic linear substitution $w_1 \oslash_x w_2$ to be the set of words obtained by replacing exactly one occurrence of x in w_1 by the word*

[1] Here, $\circ$ is used to denote relational composition.
[2] Note that in [19], λ is used where we have used ε.

w_2. For languages $L_1, L_2 \subseteq S_\Sigma$, non-deterministic linear substitution and the existential variants of its two right adjoints are defined as follows:

$$L_1 \otimes_x L_2 = \bigcup_{w_1 \in L_1, w_2 \in L_2} w_1 \otimes_x w_2$$
$$L_1 \multimap\!\!\otimes_x^{\exists} L_2 = \{w \in S_\Sigma \mid \exists w' \in L_1.\, (w \otimes_x w') \cap L_2 \neq \emptyset\}$$
$$L_1 \otimes\!\!-_x^{\exists} L_2 = \{w \in S_\Sigma \mid \exists w' \in L_1.\, (w' \otimes_x w) \cap L_2 \neq \emptyset\}.$$

Automaton Constructions We now present automaton constructions for the operation '$\otimes$', and the related operations '$\multimap\!\!\otimes^{\exists}$' and '$\otimes\!\!-^{\exists}$'.

Definition 9 ('$\otimes$' Construction). *Given $x \in \Omega$ and ε-NFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^l\}, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^l\}, A_2)$ accepting languages L_1 and L_2 respectively, define the ε-NFA $\mathcal{A}_1 \otimes_x \mathcal{A}_2 = (Q, e, \{f^l\}, A)$ by:*
- *$Q = Q_1 \times (Q_2 \uplus \{0, 1\})$;*
- *$e = (e_1, 0)$;*
- *for $l \in \Sigma$, f^l is the smallest relation satisfying: $(q_1', n) \in f^l((q_1, n))$ whenever $q_1' \in f_1^l(q_1)$, and $(q_1, q_2') \in f^l((q_1, q_2))$ whenever $q_2' \in f_2^l(q_2)$;*
- *f^ε is the smallest relation satisfying: $(q_1', n) \in f^\varepsilon((q_1, n))$ whenever $q_1' \in f_1^\varepsilon(q_1)$, $(q_1, q_2') \in f^\varepsilon((q_1, q_2))$ whenever $q_2' \in f_2^\varepsilon(q_2)$, $(q_1, e_2) \in f^\varepsilon((q_1, 0))$, and $(q_1', 1) \in f^\varepsilon((q_1, q_2))$ whenever $q_1' \in f_1^x(q_1)$ and $q_2 \in A_2$; and*
- *$A = A_1 \times \{1\}$.*

Proposition 1. *The automaton $\mathcal{A} = \mathcal{A}_1 \otimes_x \mathcal{A}_2$ accepts the language $L_1 \otimes_x L_2$.*

When this automaton is run on a word, it initially behaves like $\mathcal{A}_1$; the state has the form $(q_1, 0)$. At some point in the run, the automaton may switch to behave like $\mathcal{A}_2$ from its initial state by making a ε-transition and keeping a record of the state of $\mathcal{A}_1$ it was previously in; the state then has the form (q_1, q_2). If the automaton eventually reaches an accepting state of $\mathcal{A}_2$, the automaton may switch back to behave like $\mathcal{A}_1$ as if it had just read x instead of the word from L_2 it actually read; the state then has the form $(q_1, 1)$. Once the run is completed, if the automaton is in an accepting state, it has read a word of the form $w_1' \cdot w_2 \cdot w_1''$ where $w_1' \cdot x \cdot w_1'' \in L_1$ and $w_2 \in L_2$.

Definition 10 ('$\multimap\!\!\otimes^{\exists}$' Construction). *Given $x \in \Omega$ and ε-NFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^l\}, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^l\}, A_2)$ accepting languages L_1 and L_2 respectively, define the ε-NFA $\mathcal{A}_1 \multimap\!\!\otimes_x^{\exists} \mathcal{A}_2 = (Q, e, \{f^l\}, A)$ by:*
- *$Q = Q_2 \times \{0, 1\}$;*
- *$e = (e_2, 0)$;*
- *$\{f^l\}$ is the set of the smallest relations satisfying: for $l \in \Sigma \uplus \{\varepsilon\}$, $(q_2', n) \in f^l((q_2, n))$ if and only if $q_2' \in f_2^l(q_2)$, and $(q_2', 1) \in f^x((q_2, 0))$ if and only if, for some $w \in L_1$, $q_2' \in (\!| w_1 |\!)_{\mathcal{A}_2}(q_2)$; and*
- *$A = A_2 \times \{1\}$.*

Proposition 2. *The automaton $\mathcal{A} = \mathcal{A}_1 \multimap\!\!\otimes_x^{\exists} \mathcal{A}_2$ accepts the language $L_1 \multimap\!\!\otimes_x^{\exists} L_2$.*

When this automaton is run on a word, it starts in state $(e_2, 0)$ and proceeds to read the word as $\mathcal{A}_2$ would. Eventually, it may be in state $(q_2, 0)$ having so far read w_2', say, and about to read the symbol x; we know that $q_2 \in [\![w_2']\!]_{\mathcal{A}_2}$. On reading the x, the automaton may transition to the state $(q_2', 1)$ if there is some $w_1 \in L_1$ with $q_2' \in (\!|w_1|\!)_{\mathcal{A}_2}(q_2)$. At this point, the automaton has consumed $w_2' \cdot x$ and is in state $(q_2', 1)$ where $q_2' \in [\![w_2' \cdot w_1]\!]_{\mathcal{A}_2}$ for some $w_1 \in L_1$. The automaton then proceeds to read the remainder of the word, call it w_2'', as $\mathcal{A}_2$ would, eventually reaching a state $(q_2'', 1)$, say, where $q_2'' \in [\![w_2' \cdot w_1 \cdot w_2'']\!]_{\mathcal{A}_2}$ for some $w_1 \in L_1$. If this is an accepting state, that signifies that the automaton has read $w_2' \cdot x \cdot w_2''$ for some w_2', w_2'' with $w_2' \cdot w_1 \cdot w_2'' \in L_2$ for some $w_1 \in L_1$.

In order for the construction of $\mathcal{A}_1 \multimap\!\!\oslash_x^{\exists} \mathcal{A}_2$ to be effective, we must be able to determine if there is some $w_1 \in L_1$ with $q_2' \in (\!|w_1|\!)_{\mathcal{A}_2}(q_2)$ for any given $q_2, q_2' \in Q_2$. This may be done by considering the product pre-automaton $\hat{\mathcal{A}}_1 \times \hat{\mathcal{A}}_2 = (Q_1 \times Q_2, (e_1, e_2), \{f_1^l \times f_2^l\})$. This pre-automaton behaves like $\mathcal{A}_1$ and $\mathcal{A}_2$ run in parallel, and so there is a path in $\hat{\mathcal{A}}_1 \times \hat{\mathcal{A}}_2$ from state (e_1, q_2) to state (q_1', q_2'), for some accepting $q_1' \in A_1$, if and only if there is some $w_1 \in L_1$ with $q_2' \in (\!|w_1|\!)_{\mathcal{A}_2}(q_2)$.

Definition 11 ('$\oslash\!\!-^{\exists}$' Construction). *Given $x \in \Omega$ and ε-NFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^l\}, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^l\}, A_2)$ accepting languages L_1 and L_2 respectively, define the ε-NFA $\mathcal{A}_1 \oslash\!\!-_x^{\exists} \mathcal{A}_2 = (Q, e, \{f^l\}, A)$ by:*

- *$Q = \mathcal{P}(Q_2 \times Q_2)$;*
- *$e = \varepsilon\text{-}\mathsf{closure}_2$;*
- *for $l \in \Sigma$, $f^l(q) = \{\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^l \circ q\}$;*
- *$f^\varepsilon = \emptyset$; and*
- *$q \in A$ if and only if $\exists w.\, [\![w]\!]_{\hat{\mathcal{A}}_1 \times \hat{\mathcal{A}}_q} \cap (A_1 \times A_q) \neq \emptyset$ where*
  - *$\mathcal{A}_q = (Q_2 \times \{0,1\}, (e_2, 0), \{f_q^l\}, A_q)$,*
  - *for $l \neq x$, $f_q^l = \{((q_2, n), (q_2', n)) \mid (q_2, q_2') \in f_2^l, n \in \{0,1\}\}$,*
  - *$f_q^x = \{((q_2, n), (q_2', n)) \mid (q_2, q_2') \in f_2^x, n \in \{0,1\}\} \cup \{((q_2, 0), (q_2', 1)) \mid (q_2, q_2') \in q\}$,*
  - *$A_q = A_2 \times \{1\}$.*

Proposition 3. *The automaton $\mathcal{A} = \mathcal{A}_1 \oslash\!\!-_x^{\exists} \mathcal{A}_2$ accepts the language $L_1 \oslash\!\!-_x^{\exists} L_2$.*

The automaton $\mathcal{A}$ is deterministic, and the state on reading a word w is a relation expressing the effect of reading w in $\mathcal{A}_2$ from any given state. That is, $[\![w]\!]_{\mathcal{A}} = \{(\!|w|\!)_{\mathcal{A}_2}\}$. The automaton $\mathcal{A}_q$ where $q = (\!|w|\!)_{\mathcal{A}_2}$ accepts a word w' if and only if $(w' \oslash_x w) \cap L_2 \neq \emptyset$. This is since to reach a state of the form $(q_2, 1)$, the automaton must at some point read x and make a transition in the first component of the state equivalent to reading the word w. The condition $[\![w']\!]_{\hat{\mathcal{A}}_1 \times \hat{\mathcal{A}}_q} \cap (A_1 \times A_q) \neq \emptyset$ is then equivalent to $w' \in L_1 \wedge (w' \oslash_x w) \cap L_2 \neq \emptyset$.

Decidability Given automata $\mathcal{A}_\emptyset$, $\mathcal{A}_{S_{\Upsilon,\Omega}}$ and $\mathcal{A}_{\{w\}}$ recognising the languages $\emptyset$, $S_{\Upsilon,\Omega}$ and $\{w\}$ respectively, together with the operators for constructing automata corresponding to concatenation, union, complementation with respect to S_Σ, and those defined above, we can define an encoding of a formula P and environment σ as an automaton $\mathcal{A}_{P,\sigma}$ as follows, where $x = \sigma(\alpha)$:

$$\begin{aligned}
\mathcal{A}_{P_1 \cdot P_2,\sigma} &= (\mathcal{A}_{P_1,\sigma} \cdot \mathcal{A}_{P_2,\sigma}) \cap \mathcal{A}_{S_{\Upsilon,\Omega}} & \mathcal{A}_{0,\sigma} &= \mathcal{A}_{\{\varepsilon\}} \\
\mathcal{A}_{P_1 \circ_\alpha P_2,\sigma} &= (\mathcal{A}_{P_1,\sigma} \otimes_x \mathcal{A}_{P_2,\sigma}) \cap \mathcal{A}_{S_{\Upsilon,\Omega}} & \mathcal{A}_{a,\sigma} &= \mathcal{A}_{\{a\}} \\
\mathcal{A}_{P_1 \circ\!-^{\exists}_{\alpha} P_2,\sigma} &= (\mathcal{A}_{P_1,\sigma} \otimes\!-^{\exists}_{x} \mathcal{A}_{P_2,\sigma}) \cap \mathcal{A}_{S_{\Upsilon,\Omega}} & \mathcal{A}_{\alpha,\sigma} &= \mathcal{A}_{\{x\}} \\
\mathcal{A}_{P_1 -\!\circ^{\exists}_{\alpha} P_2,\sigma} &= (\mathcal{A}_{P_1,\sigma} -\!\otimes^{\exists}_{x} \mathcal{A}_{P_2,\sigma}) \cap \mathcal{A}_{S_{\Upsilon,\Omega}} & \mathcal{A}_{False,\sigma} &= \mathcal{A}_{\emptyset} \\
\mathcal{A}_{P_1 \Rightarrow P_2,\sigma} &= (\overline{\mathcal{A}_{P_1,\sigma}} \cap \mathcal{A}_{S_{\Upsilon,\Omega}}) \cup \mathcal{A}_{P_2,\sigma}.
\end{aligned}$$

Theorem 1. *Given formula $P \in \mathcal{F}_{Seq}$, environment $\sigma : \Theta \rightharpoonup_{fin} \Omega$, and sequence $s \in S_{\Upsilon,\Omega}$, it is decidable whether $s, \sigma \models P$.*

Theorem 2. *Given formula $P \in \mathcal{F}_{Seq}$ and environment $\sigma : \Theta \rightharpoonup_{fin} \Omega$, it is decidable whether there exists a sequence $s \in S_{\Upsilon,\Omega}$ such that $s, \sigma \models P$.*

Since each formula mentions only a finite number of variables, it is only necessary to consider a finite number of environments in order to determine whether a sequence satisfies the formula in any possible environment.

Corollary 1. *Given formula $P \in \mathcal{F}_{Seq}$, it is decidable whether there exist an environment $\sigma : \Theta \rightharpoonup_{fin} \Omega$ and sequence $s \in S_{\Upsilon,\Omega}$ such that $s, \sigma \models P$.*

2.2 Expressivity

By embedding CL^{m}_{Seq} into regular languages, we can immediately infer that regular languages are at least as expressive as CL^{m}_{Seq}. In fact, we can refine this result somewhat to show that CL^{m}_{Seq}-definable sequences are exactly the star-free regular languages: the smallest class of languages containing the empty language, all single-element languages, and closed under Boolean operations and language concatenation. As previously mentioned, the star-free regular languages are also the aperiodic languages and the languages definable by formulae of First-order Logic interpreted over words.

We observe that CL^{m}_{Seq}, even without the connectives '$\circ$', '$\circ\!-^{\exists}$' and '$-\!\circ^{\exists}$', is able to express the empty language, single-element languages, Boolean operations and language concatenation. Thus, CL^{m}_{Seq}-formulae can express any star-free regular language. (The only caveat is that linearity of elements of Ω must be preserved, which may be assured by always choosing Ω so that it contains no elements of Σ that occur non-linearly in the language under consideration (which is over S_Σ).) Conversely, it can be shown that, when L_1 and L_2 are star-free regular languages and hence aperiodic, the languages $L_1 \otimes_x L_2$, $L_1 \otimes\!-^{\exists}_{x} L_2$ and $L_1 -\!\otimes^{\exists}_{x} L_2$ are themselves aperiodic and hence star-free. This gives us the following two results.

Theorem 3. *The languages definable by formulae of CL^{m}_{Seq} are exactly the star-free regular languages.*

Corollary 2. *The connectives '$\circ$', '$\circ\!-^{\exists}$' and '$-\!\circ^{\exists}$' do not contribute to the expressivity of CL^{m}_{Seq}.*

3 Trees

The decidability results for CL^{m}_{Seq} also hold for Context Logic for ranked trees (or terms), abbreviated CL^{m}_{Term}. Again, the key step is constructing automata corresponding to the connectives '$\circ$', '$\circ\!-^{\exists}$' and '$-\!\circ^{\exists}$'; we do not give the constructions here. Automata for ranked trees are well known in the literature; for a comprehensive treatment, see [20, 21].

For the remainder this section, we consider decidability of multi-holed Context Logic for unranked trees (or forests), abbreviated CL^{m}_{Tree}. The forest model for Context Logic consists of forests with nodes labelled by a finite unranked alphabet, Σ. A forest may have zero or more root nodes, and each node may have zero or more children. As before, we consider the alphabet to be partitioned into two sets: Υ the set of labels, which is ranged over by a, b, and Ω the set of hole labels, ranged over by x, y. Only nodes without children may be labelled from Ω.

Definition 12 (Unranked Trees). *The set of* unranked trees *(or* forests*) over $\Sigma = \Upsilon \uplus \Omega$, T_Σ, ranged over by t, t_1, t_2, is defined inductively as*

$$t ::= \varepsilon \;\big|\; a[t] \;\big|\; x \;\big|\; t_1 \mid t_2 \qquad\qquad (a \in \Upsilon, x \in \Omega)$$

modulo structural equivalences given by the '$|$' operator being associative with identity ε (the empty tree).

Definition 13 (Multi-holed Unranked Tree Contexts). *A* multi-holed unranked tree context *(or simply* forest context*) is a forest from T_Σ in which each $x \in \Omega$ occurs* at most once. *The set of forest contexts over Υ and Ω is denoted by $T_{\Upsilon,\Omega}$, and t, t_1, t_2 denote elements of this set.*

Context composition is defined for forest contexts as it is for sequences. We omit the definition here. We define quantifier-free CL^{m}_{Tree}. As before, the names of variables ranging over holes are taken from an infinite set of atoms, the set of *hole variables* Θ, ranged over by α, β. The satisfaction relation for CL^{m}_{Tree} is defined with respect to an environment that assigns hole labels to the hole variables that appear in the formula in question.

Definition 14 (Formulae). *The set of formulae of CL^{m}_{Tree}, denoted $\mathcal{F}_{Tree}$ and ranged over by P, P_1, P_2, is defined by:*

$$\begin{array}{rlll} P ::= & 0 \;\big|\; a[P] \;\big|\; P_1 \mid P_2 & (a \in \Upsilon) & \textit{tree formulae} \\ & \alpha \;\big|\; P_1 \circ_\alpha P_2 \;\big|\; P_1 \circ\!-^{\exists}_{\alpha} P_2 \;\big|\; P_1 -\!\circ^{\exists}_{\alpha} P_2 & (\alpha \in \Theta) & \textit{structural formulae} \\ & \mathit{False} \;\big|\; P_1 \Rightarrow P_2 & & \textit{Boolean formulae.} \end{array}$$

Definition 15 (Satisfaction Relation). *The satisfaction relation for CL^{m}_{Tree} is defined inductively on the structure of the formulae. The definitions for the* *structural and Boolean formulae are identical to the sequence case, and so we only present the definitions for tree formulae.*

$$\begin{aligned} t,\sigma \models 0 &\iff t = \varepsilon \\ t,\sigma \models a[P] &\iff \exists t'.\, t = a[t'] \,\wedge\, t',\sigma \models P \\ t,\sigma \models P_1 \mid P_2 &\iff \exists t_1, t_2.\, t = t_1 \mid t_2 \,\wedge\, t_1,\sigma \models P_1 \,\wedge\, t_2,\sigma \models P_2. \end{aligned}$$

3.1 Automata

We again consider automata in order to decide model checking and satisfiability. Our definition of automata for unranked trees does not follow that of [21], but can be seen to be equivalent by the first-child-next-sibling encoding of forests into binary trees described *ibidem*. The definition here can be seen as a natural extension of that for sequences by considering '|' to be analogous to '$\cdot$'; the transition relation for each label considers not only the state after consuming the tree to the left of the current node (as in sequences), but also the state given by running the automaton on the subtree beneath the current node.

Definition 16 (ε-NFFA). *A* non-deterministic finite forest automaton with ε-transitions*, abbreviated ε-NFFA, is a tuple $\mathcal{A} = (Q, e, \{f^a\}_{a\in\Upsilon}, \{f^x\}_{x\in\Omega}, f^\varepsilon, A)$ where: Q is the set of states, a finite set; $e \in Q$ is the initial state; for every $a \in \Upsilon$, $f^a \subseteq Q \times Q \times Q$ is the state transition relation for a; for every $x \in \Omega$, $f^x \subseteq Q\times Q$ is the state transition relation for x; $f^\varepsilon \subseteq Q\times Q$ is the non-consuming state transition relation; and $A \subseteq Q$ is the set of accepting states.*

Given $q_1, q_2 \in Q$, the notation $f^a(q_1, q_2)$ is used for the set $\{q' \mid (q_1, q_2, q') \in f^a\}$, and, given $q \in Q$, $f^l(q)$ is used for the set $\{q' \mid (q, q') \in f^l\}$ where $l = \varepsilon$ or $l \in \Omega$.

As for sequences, the ε-$\mathsf{closure}$ of a state is defined to be the set of states reachable by any number of ε-transitions: ε-$\mathsf{closure}$ is the reflexive-transitive closure of f^ε. Each automaton $\mathcal{A}$ induces a function $[\![-]\!]_\mathcal{A} : T_\Sigma \to \mathcal{P}(Q)$ that maps each forest to a set of states according to the following definition:

$$\begin{aligned} [\![\varepsilon]\!]_\mathcal{A} &= \varepsilon\text{-}\mathsf{closure}(e) \\ [\![t \mid x]\!]_\mathcal{A} &= \{q \in Q \mid \exists q' \in [\![t]\!]_\mathcal{A}.\, q \in (\varepsilon\text{-}\mathsf{closure} \circ f^x)(q')\} \\ [\![t_1 \mid a[t_2]]\!]_\mathcal{A} &= \left\{ q \in Q \;\middle|\; \begin{array}{c} \exists q_1 \in [\![t_1]\!]_\mathcal{A},\ q_2 \in [\![t_2]\!]_\mathcal{A}. \\ q \in (\varepsilon\text{-}\mathsf{closure} \circ f^a)(q_1, q_2) \end{array} \right\}. \end{aligned}$$

A forest t is said to be *accepted* by $\mathcal{A}$ if $[\![t]\!]_\mathcal{A} \cap A \neq \emptyset$. The (forest) language $L_\mathcal{A}$ defined by $\mathcal{A}$ is the set $\{t \in T_\Sigma \mid [\![t]\!]_\mathcal{A} \cap A \neq \emptyset\}$. An automaton $\mathcal{A}$ also induces a function $(\!|-|\!)_\mathcal{A} : T_\Sigma \to \mathcal{P}(Q \times Q)$ that maps each forest to a state transition relation as follows:

$$\begin{aligned} (\!|\varepsilon|\!)_\mathcal{A} &= \varepsilon\text{-}\mathsf{closure} \\ (\!|t \mid x|\!)_\mathcal{A} &= \varepsilon\text{-}\mathsf{closure} \circ f^x \circ (\!|t|\!)_\mathcal{A} \\ (\!|t_1 \mid a[t_2]|\!)_\mathcal{A} &= \left\{ (q, q') \in Q \times Q \;\middle|\; \begin{array}{c} \exists q_1 \in (\!|t_1|\!)_\mathcal{A}(q),\ q_2 \in [\![t_2]\!]_\mathcal{A}. \\ q' \in (\varepsilon\text{-}\mathsf{closure} \circ f^a)(q_1, q_2) \end{array} \right\}. \end{aligned}$$

The language membership and emptiness problems for forest automata are decidable in a similar fashion to the equivalent problems for automata on words. The class of languages definable by forest automata is the class of *regular forest languages*. This class includes the empty language, all single-element languages, T_Σ and $T_{\Upsilon,\Omega}$, and is closed under union, intersection, complementation (with respect to T_Σ) and concatenation.

Automaton Constructions As in the case of words, we can define non-deterministic linear substitution, '$\varobslash$', and its related operations, '$-\!\varobslash^\exists$' and '$\varobslash\!-^\exists$', for forests. We give automata constructions for each of these which are analogous to those for words.

Definition 17 ('$\varobslash$' Construction). *Given $x \in \Omega$ and ε-NFFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^a\}_{a\in\Upsilon}, \{f_1^x\}_{x\in\Omega}, f_1^\varepsilon, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^a\}_{a\in\Upsilon}, \{f_2^x\}_{x\in\Omega}, f_2^\varepsilon, A_2)$ accepting languages L_1 and L_2 respectively, define the ε-NFFA $\mathcal{A}_1 \varobslash_x \mathcal{A}_2 = (Q, e, \{f^a\}_{a\in\Upsilon}, \{f^x\}_{x\in\Omega}, f^\varepsilon, A)$ by:*
- *$Q = (Q_1 \times (Q_2 \uplus \{0,1\})) \uplus Q_2 \uplus \{e\}$;*
- *e is fresh;*
- *for $a \in \Upsilon$, f^a is the smallest relation satisfying: $(q_1'', n'') \in f^a((q_1, n), (q_1', n'))$ whenever $q_1'' \in f_1^a(q_1, q_1')$ and $n'' = n + n'$ with $n, n', n'' \in \{0,1\}$, $(q_1, q_2'') \in f^a((q_1, q_2), q_2')$ whenever $q_2'' \in f_2^a(q_2, q_2')$, and $f_2^a \subseteq f^a$;*
- *for $y \in \Omega$, f^y is the smallest relation satisfying: $(q_1', n) \in f^y((q_1, n))$ whenever $q_1' \in f_1^y(q_1)$ and for $n \in \{0,1\}$, $(q_1, q_2') \in f^y((q_1, q_2))$ whenever $q_2' \in f_2^y(q_2)$, and $f_2^y \subseteq f^y$;*
- *f^ε is the smallest relation satisfying: $(q_1', n) \in f^\varepsilon((q_1, n))$ whenever $q_1' \in f_1^\varepsilon(q_1)$ and for $n \in \{0,1\}$, $(q_1, q_2') \in f^\varepsilon((q_1, q_2))$ whenever $q_2' \in f_2^\varepsilon(q_2)$, $f_2^\varepsilon \subseteq f^\varepsilon$, $(e_1, 0), e_2 \in f^\varepsilon(e)$, $(q_1, e_2) \in f^\varepsilon((q_1, 0))$, and $(q_1', 1) \in f^\varepsilon((q_1, q_2))$ whenever $q_1' \in f_1^x(q_1)$ and $q_2 \in A_2$; and*
- *$A = A_1 \times \{1\}$.*

Proposition 4. *The automaton $\mathcal{A} = \mathcal{A}_1 \varobslash_x \mathcal{A}_2$ accepts the language $L_1 \varobslash_x L_2$.*

For a tree, t, the each state $q \in [\![t]\!]_\mathcal{A}$ is of one of five types. If $q = e$ then $t = \varepsilon$. If $q \in Q_2$ then $q \in [\![t]\!]_{\mathcal{A}_2}$. If $q = (q_1, 0)$ then $q_1 \in [\![t]\!]_{\mathcal{A}_1}$. If $q = (q_1, q_2)$ then $t = t_1 \mid t_2$ such that $q_1 \in [\![t_1]\!]_{\mathcal{A}_1}$ and $q_2 \in [\![t_2]\!]_{\mathcal{A}_2}$. If $q = (q_1, 1)$ then $t = t_1 \varobslash_x t_2$ such that $q_1 \in [\![t_2]\!]_{\mathcal{A}_1}$ and $t_2 \in L_2$. The set of states $[\![t]\!]_\mathcal{A}$ is the most general set satisfying these requirements, so we can be sure that there is a $q \in [\![t]\!]_\mathcal{A} \cap A$ if and only if $t \in L_1 \varobslash_x L_2$.

Definition 18 ('$-\!\varobslash^\exists$' Construction). *Given $x \in \Omega$ and ε-NFFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^a\}_{a\in\Upsilon}, \{f_1^x\}_{x\in\Omega}, f_1^\varepsilon, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^a\}_{a\in\Upsilon}, \{f_2^x\}_{x\in\Omega}, f_2^\varepsilon, A_2)$ accepting languages L_1 and L_2 respectively, define the ε-NFFA $\mathcal{A}_1 -\!\varobslash_x^\exists \mathcal{A}_2 = (Q, e, \{f^a\}_{a\in\Upsilon}, \{f^x\}_{x\in\Omega}, f^\varepsilon, A)$ by:*
- *$Q = Q_2 \times \{0,1\}$;*
- *$e = (e_2, 0)$;*
- *for $a \in \Upsilon$, f^a is the smallest relation such that for $n, n', n'' \in \{0,1\}$, $(q_2'', n'') \in f^a((q_2, n), (q_2', n'))$ if and only if $n'' = n + n'$ and $q_2'' \in f_2^a(q_2, q_2')$;*

– *for* $l \in \Omega \uplus \{\varepsilon\}$, f^l *is the smallest relation such that* $(q_2', n) \in f^l((q_2, n))$ *if and only if* $q_2' \in f_2^l(q_2, q_2')$ *and if* $l = x$ *then* $(q_2', 1) \in f^l((q_2, 0))$ *if and only if there is some* $t_1 \in L_1$ *such that* $q_2' \in (\!|t_1|\!)_{\mathcal{A}_2}(q_2)$*; and*
– $A = A_2 \times \{1\}$.

Proposition 5. *The automaton* $\mathcal{A} = \mathcal{A}_1 \multimap\!\!\oslash_x^{\exists} \mathcal{A}_2$ *accepts the language* $L_1 \multimap\!\!\oslash_x^{\exists} L_2$.

As in the sequence case, a state of the form $(q_2, 0)$ records that the consumed tree has state q_2 in the automaton $\mathcal{A}_2$, while a state of the form $(q_2, 1)$ records that the consumed tree, after one instance of x is replaced by a particular tree from L_1, has state q_2 in the automaton $\mathcal{A}_2$. In order for this construction to be effective we must be able to determine if there is some $t_1 \in L_1$ with $q_2' \in (\!|t_1|\!)_{\mathcal{A}_2}(q_2)$. This can be done in the same manner as in the sequence case.

Definition 19 (‘$\oslash\!\!-^{\exists}$’ Construction). *Given* $x \in \Omega$ *and* ε*-NFFA* $\mathcal{A}_1 = (Q_1, e_1, \{f_1^a\}_{a \in \Upsilon}, \{f_1^x\}_{x \in \Omega}, f_1^{\varepsilon}, A_1)$ *and* $\mathcal{A}_2 = (Q_2, e_2, \{f_2^a\}_{a \in \Upsilon}, \{f_2^x\}_{x \in \Omega}, f_2^{\varepsilon}, A_2)$ *accepting languages* L_1 *and* L_2 *respectively, define the* ε*-NFFA* $\mathcal{A}_1 \oslash\!\!-_x^{\exists} \mathcal{A}_2 = (Q, e, \{f^a\}_{a \in \Upsilon}, \{f^x\}_{x \in \Omega}, f^{\varepsilon}, A)$ *by:*
– $Q = \mathcal{P}(Q_2 \times Q_2)$*;*
– $e = \varepsilon\text{-}\mathsf{closure}_2$*;*
– *for* $a \in \Upsilon$, $f^a(q, q') = \{\varepsilon\text{-}\mathsf{closure}_2 \circ \{(q_2, q_2') \mid \exists q_2'' \in q(q_2), q_2''' \in q'(e_2).\, q_2' \in f_2^a(q_2'', q_2''')\}\}$*;*
– *for* $y \in \Omega$, $f^y(q) = \{\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y \circ q\}$*;*
– $f^{\varepsilon} = \emptyset$*; and*
– $q \in A$ *if and only if* $\exists t.\, [\![t]\!]_{\hat{\mathcal{A}}_1 \times \hat{\mathcal{A}}_q} \in A_1 \times A_q$ *where*
  • $\mathcal{A}_q = (Q_2 \times \{0,1\}, (e_2, 0), \{f_q^a\}_{a \in \Upsilon}, \{f_q^y\}_{y \in \Omega}, f_q^{\varepsilon}, A_q)$,
  • $f_q^a = \{((q_2, n), (q_2', n'), (q_2'', n'')) \mid (q_2, q_2', q_2'') \in f_2^a, n + n' = n''\}$,
  • *for* $y \neq x$, $f_q^y = \{((q_2, n), (q_2', n)) \mid (q_2, q_2') \in f_2^y, n \in \{0,1\}\}$,
  • $f_q^x = \{((q_2, n), (q_2', n)) \mid (q_2, q_2') \in f_2^x, n \in \{0,1\}\} \cup \{((q_2, 0), (q_2', 1)) \mid (q_2, q_2') \in q\}$,
  • $f_q^{\varepsilon} = \{((q_2, n), (q_2', n)) \mid (q_2, q_2') \in f_2^{\varepsilon}, n \in \{0,1\}\}$*, and*
  • $A_q = A_2 \times \{1\}$.

Proposition 6. *The automaton* $\mathcal{A} = \mathcal{A}_1 \oslash\!\!-_x^{\exists} \mathcal{A}_2$ *accepts the language* $L_1 \oslash\!\!-_x^{\exists} L_2$.

A state of $\mathcal{A}$ is a relation describing how the consumed tree would behave when appended to any other tree in the automaton $\mathcal{A}_2$. The principle of this construction is the same as in the sequence case.

Decidability The automaton constructions we have described, together with the standard closure properties of regular forest languages allow us to translate all formulae of CL^{m}_{Tree} into automata, as we did for CL^{m}_{Seq}. Hence we get the following decidability results.

Theorem 4. *Given formula* $P \in \mathcal{F}_{Tree}$*, environment* $\sigma : \Theta \rightharpoonup_{fin} \Omega$ *and forest context* $t \in T_{\Upsilon,\Omega}$*, it is decidable whether* $t, \sigma \models P$.

Theorem 5. *Given formula* $P \in \mathcal{F}_{Tree}$ *and environment* $\sigma : \Theta \rightharpoonup_{fin} \Omega$*, it is decidable whether there exists a forest context* $t \in T_{\Upsilon,\Omega}$ *such that* $t, \sigma \models P$.

Corollary 3. *Given formula* $P \in \mathcal{F}_{Tree}$*, it is decidable whether there exist an environment* $\sigma : \Theta \rightharpoonup_{fin} \Omega$ *and forest context* $t \in T_{\Upsilon,\Omega}$ *such that* $t, \sigma \models P$.

4 Decidability with Quantifiers

We extend our results to Context Logic with quantification over hole labels. In this setting, we naturally assume the set of hole labels, Ω to be infinite — the finite case is degenerate and may be solved by replacing quantified formulae by disjunctions or conjunctions.[3] For the most part, our results are also independent of the model of Context Logic used. We take C_Ω to be the set of contexts of an arbitrary model, ranged over by c, c_1, c_2.

Definition 20 (Freshness). *For hole label x and context c, if $x \notin fn(c)$ then we write $x \sharp c$. For hole label x and environment σ, if $x \notin range(\sigma)$ then we write $x \sharp \sigma$. We write $x \sharp c, \sigma$ to indicate that $x \sharp c$ and $x \sharp \sigma$.*

We extend the formulae of Context Logic with additional connectives to the set $\mathcal{F}_{\exists,\text{И}}$, as follows.

$$P ::= \ \cdots \ \big| \ \Diamond P \ \big| \ \mapsto \ \big| \ \exists\alpha.\, P \ \big| \ \text{И}\alpha.\, P$$

The satisfaction relation is extended correspondingly.

$$\begin{array}{lcl} c, \sigma \models \Diamond P & \iff & \exists x \in \Omega, c_1, c_2 \in C_\Omega.\, c = c_1 \circledx c_2 \wedge c_2, \sigma \models P \\ c, \sigma \models \mapsto & \iff & \exists x \in \Omega.\, c = x \\ c, \sigma \models \exists\alpha.\, P & \iff & \exists x \in \Omega.\, c, \sigma[\alpha \mapsto x] \models P \\ c, \sigma \models \text{И}\alpha.\, P & \iff & \exists x \in \Omega.\, x \sharp c, \sigma \wedge c, \sigma[\alpha \mapsto x] \models P \end{array}$$

Our automata techniques can be extended to decide satisfiability for formulae with containing '$\Diamond$' and '$\mapsto$'. We show how to decide satisfiability for formulae containing classical quantification ($\exists$) and Gabbay-Pitts fresh quantification (И).

Two formulae, P_1 and P_2, are *logically equivalent*, denoted $P_1 \equiv P_2$, if they are satisfied by exactly the same contexts and environments.

Lemma 1 (Encoding Existential with Freshness). *For all P,*

$$\exists\alpha.\, P \ \equiv \ \text{И}\alpha.\, P \circ_\alpha \left(\mapsto \wedge \neg \bigvee_{\beta \in fv(P) \setminus \{\alpha\}} \beta \right) \vee P \vee \bigvee_{\beta \in fv(P) \setminus \{\alpha\}} P[\beta/\alpha].$$

Consequently, every formula can be rewritten to an equivalent formula that contains no existential quantifiers.

Lemma 2 (Prenex Normalisation). *Every $\exists$-free formula is equivalent to a formula in which all quantifiers appear at the head of the formula — the* prenex normal form.

[3] Without quantification, extending our automata techniques to handle infinite alphabets requires some technical manipulation which we do not describe here.

Lemma 3 (Deciding Satisfiability). *For all environments σ, formulae P, and hole variables α with $\alpha \notin dom(\sigma)$,*

$$\begin{aligned} &\exists c \in C_{\Omega}.\, c, \sigma \models \reflectbox{N}\alpha.\, P \\ \iff &\exists y \in \Omega.\, y \sharp \sigma \;\wedge\; \exists c \in C_{\Omega}.\, c, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha \\ \iff &\forall y \in \Omega.\, y \sharp \sigma \implies \exists c \in C_{\Omega}.\, c, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha. \end{aligned}$$

By rewriting a formula with Lemmata 1 and 2 and applying Lemma 2, we can reduce the problem of deciding an arbitrary formula of Context Logic to the problem of deciding a quantifier-free formula, which we have already shown to be decidable. We sum up these results in the following theorems.

Theorem 6. *Given formula $P \in \mathcal{F}_{\exists,\reflectbox{N}}$, environment $\sigma : \Theta \rightharpoonup_{fin} \Omega$ and context $c \in C_{\Omega}$, it is decidable whether $c, \sigma \models P$.*

Theorem 7. *Given formula $P \in \mathcal{F}_{\exists,\reflectbox{N}}$ and environment $\sigma : \Theta \rightharpoonup_{fin} \Omega$, it is decidable whether there exists a context $c \in C_{\Omega}$ such that $c, \sigma \models P$.*

Corollary 4. *Given formula $P \in \mathcal{F}_{\exists,\reflectbox{N}}$, it is decidable whether there exist an environment $\sigma : \Theta \rightharpoonup_{fin} \Omega$ and unranked tree context $c \in C_{\Omega}$ such that $c, \sigma \models P$.*

5 Conclusions

We have shown how to decide model-checking and satisfiability for multi-holed Context Logic for sequences and trees. We have shown how classical quantification over the linear hole labels ($\forall, \exists$) can be re-expressed with Gabbay-Pitts fresh quantification ($\reflectbox{N}$), which can be extruded so that decidability reduces to the quantifier-free case. For this case, we have shown how to construct automata corresponding to formulae, which may be used to decide model-checking and satisfiability. These constructions embed Context Logic into the regular languages, and we have shown how the logic relates to other known classes of languages — in particular, the close correspondence with first-order definable languages.

Although our results settle the question of decidability for Context Logic, they do leave open several avenues for further work. Firstly, our decision procedure gives a poor upper bound on the complexity of satisfiability and model-checking. In particular, the construction for $\mathcal{A}_1 \oslash\!\!-^{\exists}_{x} \mathcal{A}_2$ has 2^{n^2} states if $\mathcal{A}_2$ has n states. Since this state space must be explored to determine satisfiability, this gives a complexity upper bound that is NONELEMENTARY in the number of connectives in a formula. Our experiments in implementing the procedures presented here have not so far achieved practically useful complexity for reasonable examples.[4] Hence, the questions of what the true complexity of decidability is and whether good performance can be achieved in practise remain open.

Decision procedures based on automata can produce witnesses to satisfiability — the path to an accepting state corresponds to a sequence that the

[4] Our prototype Haskell implementation can be found at: `http://www.doc.ic.ac.uk/~td202/automaton3.hs`

automaton accepts. Validity of a formula, on the other hand, is essentially decided by eliminating all counterexamples and so our procedure does not directly produce a proof. The search for a practical proof procedure for Context Logic is therefore another interesting avenue for further work.

References

1. Calcagno, C., Gardner, P., Zarfaty, U.: Context Logic and tree update. In: POPL '05. (2005) 271–282
2. O'Hearn, P., Pym, D.: Logic of bunched implications. Bulletin of Symbolic Logic **5**(2) (1999) 215–244
3. Gardner, P.A., Smith, G.D., Wheelhouse, M.J., Zarfaty, U.D.: Local hoare reasoning about dom. In: PODS '08, New York, NY, USA, ACM (2008) 261–270
4. Ishtiaq, S.S., O'Hearn, P.W.: BI as an assertion language for mutable data structures. In: POPL '01. (2001)
5. Reynolds, J.C.: Separation Logic: A logic for shared mutable data structures. In: LICS '02. (2002) 55–74
6. Yang, H., O'Hearn, P.W.: A semantic basis for local reasoning. In: ETAPS '02 and FOSSACS '02. (2002)
7. Cardelli, L., Gordon, A.D.: Anytime, anywhere: modal logics for mobile ambients. In: POPL '00. (2000) 365–377
8. Calcagno, C., Gardner, P., Zarfaty, U.: Context Logic as modal logic: completeness and parametric inexpressivity. In: POPL '07. (2007) 123–134
9. Calcagno, C., Yang, H., O'Hearn, P.W.: Computability and complexity results for a spatial assertion language for data structures. In: FSTTCS '01. (2001) 108–119
10. Calcagno, C., Cardelli, L., Gordon, A.D.: Deciding validity in a spatial logic for trees. SIGPLAN Not. **38**(3) (2003) 62–73
11. Dal Zilio, S., Lugiez, D., Meyssonnier, C.: A logic you can count on. In: POPL '04. (2004) 135–146
12. Calcagno, C., Dinsdale-Young, T., Gardner, P.: Adjunct elimination in Context Logic for trees. In: APLAS '07. (2007)
13. Schützenberger, M.P.: On finite monoids having only trivial subgroups. Information and Control **8** (April 1965) 190–194
14. McNaughton, R., Papert, S.A.: Counter-Free Automata (M.I.T. research monograph no. 65). The MIT Press (1971)
15. Thomas, W.: Languages, automata, and logic. In: Handbook of formal languages, vol. 3: beyond words. Springer, New York (1997) 389–455
16. Heuter, U.: First-order properties of trees, star-free expressions, and aperiodicity. Informatique théorique et applications **25**(2) (1991) 125–145
17. Bojańczyk, M.: Forest expressions. In: CSL '07. (2007)
18. Conforti, G., Ghelli, G.: Decidability of freshness, undecidability of revelation. In: FOSSACS '04. (2004) 105–120
19. Yu, S.: Regular languages. In: Handbook of formal languages, vol. 1: word, language, grammar. Springer, New York (1997) 41–110
20. Gécseg, F., Steinby, M.: Tree languages. In: Handbook of formal languages, vol. 3: beyond words. Springer, New York (1997) 1–68
21. Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications. Available on: `http://www.grappa.univ-lille3.fr/tata` (2007) release October, 12th 2007.

A Proofs

This appendix (not for publication) contains proof details not given in the body of the paper, for the benefit of referees.

A.1 Sequence Constructions

Correctness of '$\oslash$' Construction Given ε-NFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^l\}_{l \in \Sigma \cup \{\varepsilon\}}, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^l\}_{l \in \Sigma \cup \{\varepsilon\}}, A_2)$ accepting languages L_1 and L_2 respectively, let $\mathcal{A} = \mathcal{A}_1 \oslash_x \mathcal{A}_2 = (Q, e, \{f^l\}_{l \in \Sigma \cup \{\varepsilon\}}, A)$ as per Def. 9.

Lemma 4. *For all $w \in S_\Sigma$, $q \in Q_1$,*

$$(q_1, 0) \in [\![w]\!]_{\mathcal{A}} \iff q_1 \in [\![w]\!]_{\mathcal{A}_1}.$$

Proof. Both directions: by induction on the structure of w.

$\Rightarrow$:

Base case: $w = \varepsilon$ so $(q_1, 0) \in \varepsilon\text{-}\mathsf{closure}((e_1, 0))$, and hence $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q_1)$, and $q_1 \in [\![\varepsilon]\!]_{\mathcal{A}_1}$.

Inductive case: $w = w' \cdot l$ (for $l \in \Sigma$). In this case $(q_1, 0) \in (\varepsilon\text{-}\mathsf{closure} \circ f^l)(q')$ for some $q' \in [\![w']\!]_{\mathcal{A}}$. That is, $(q_1, 0) \in ((f^\varepsilon)^n \circ f^l)(q')$ for some n. This implies that $q' = (q_1', 0)$ and that $q_1 \in ((f_1^\varepsilon)^n \circ f_1^l)(q_1')$, and hence $q_1 \in [\![w]\!]_{\mathcal{A}_1}$.

$\Leftarrow$:

Base case: $w = \varepsilon$ so $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(e_1)$, and hence $(q_1, 0) \in \varepsilon\text{-}\mathsf{closure}(e)$.

Inductive case: $w = w' \cdot l$ (for $l \in \Sigma$). In this case $q_1 \in (\varepsilon\text{-}\mathsf{closure}_1 \circ f_1^l)(q_1')$ for some $q_1' \in [\![w']\!]_{\mathcal{A}_1}$. That is, $q_1 \in ((f_1^\varepsilon)^n \circ f^l)(q_1')$ for some n. Since by the inductive hypothesis $(q_1', 0) \in [\![w']\!]_{\mathcal{A}}$, this implies that $(q_1, 0) \in [\![w]\!]_{\mathcal{A}}$. □

Lemma 5. *For all $w \in S_\Sigma$, $q_1 \in Q_1$, $q_2 \in Q_2$,*

$$(q_1, q_2) \in [\![w]\!]_{\mathcal{A}} \iff \exists w_1, w_2.\, w = w_1 \cdot w_2 \wedge q_1 \in [\![w_1]\!]_{\mathcal{A}_1} \wedge q_2 \in [\![w_2]\!]_{\mathcal{A}_2}.$$

Proof. Both directions: by induction on the structure of w.

$\Rightarrow$:

Base case: $w = \varepsilon$ so $(q_1, q_2) \in \varepsilon\text{-}\mathsf{closure}((e_1, 0))$. Thus, by the definition of $\mathcal{A}$, it must be the case that:

- $(q_1, q_2) \in \varepsilon\text{-}\mathsf{closure}((q_1, e_2))$, and hence $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(e_2)$, and $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$;
- $(q_1, e_2) \in f^\varepsilon((q_1, 0))$; and
- $(q_1, 0) \in [\![\varepsilon]\!]_{\mathcal{A}}$ and so $q_1 \in [\![\varepsilon]\!]_{\mathcal{A}_1}$ (by Lem. 4).

Inductive case: $w = w' \cdot l$ (for $l \in \Sigma$). In this case $(q_1, q_2) \in (\varepsilon\text{-}\mathsf{closure} \circ f^l)(q')$. Either $q' = (q_1, q_2')$ or $q' = (q_1', 0)$.

In the former case, $q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^l)(q_2')$, by the definition of $\mathcal{A}$. By the inductive hypothesis, there are w_1, w_2' with $w = w_1 \cdot w_2' \cdot l$, $q_1 \in [\![w_1]\!]_{\mathcal{A}_1}$ and $q_2' \in [\![w_2']\!]_{\mathcal{A}_2}$. Hence $q_2 \in [\![w_2' \cdot l]\!]_{\mathcal{A}_2}$, and so the choice of w_1 and $w_2 = w_2' \cdot l$ fulfills the requirements.

In the latter case, it must be that:

- $(q_1, q_2) \in \varepsilon\text{-}\mathsf{closure}((q_1, e_2))$, and hence $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(e_2)$, and $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$;
- $(q_1, e_2) \in f^{\varepsilon}((q_1, 0))$; and
- $(q_1, 0) \in [\![w]\!]_{\mathcal{A}}$ and so $q_1 \in [\![w]\!]_{\mathcal{A}_1}$ (by Lem. 4).

Therefore, the choice of $w_1 = w$ and $w_2 = \varepsilon$ fulfills the requirements.

$\Leftarrow$:

Base case: $w = \varepsilon$. In this case, $w_1 = \varepsilon$ and $w_2 = \varepsilon$. By Lem. 4, $(q_1, 0) \in [\![\varepsilon]\!]_{\mathcal{A}}$, and so, since $(q_1, e_2) \in f^{\varepsilon}((q_1, 0))$, $(q_1, e_2) \in [\![\varepsilon]\!]_{\mathcal{A}}$. Now, it must be the case that $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(e_2)$ and hence $(q_1, q_2) \in [\![\varepsilon]\!]_{\mathcal{A}}$ as required.

Inductive case: $w = w' \cdot l$. Here, either $w_2 = w_2' \cdot l$, or $w_2 = \varepsilon$ and $w_1 = w = w' \cdot l$.

In the former case, $q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^l)(q_2')$ for some q_2' with $q_2' \in [\![w_2']\!]_{\mathcal{A}_2}$. Hence, by the inductive hypothesis, $(q_1, q_2') \in [\![w_1 \cdot w_2']\!]_{\mathcal{A}}$. By the definition of $\mathcal{A}$, it follows that $(q_1, q_2) \in [\![w_1 \cdot w_2]\!]_{\mathcal{A}}$, as required.

In the latter case, $q_1 \in [\![w]\!]_{\mathcal{A}_1}$ and so, by Lem. 4, $(q_1, 0) \in [\![w]\!]_{\mathcal{A}}$. It follows then that $(q_1, e_2) \in [\![w]\!]_{\mathcal{A}}$. Further, since $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2} = \varepsilon\text{-}\mathsf{closure}_2(e_2)$, it follows that $(q_1, q_2) \in [\![w]\!]_{\mathcal{A}}$, as required. $\square$

Lemma 6. *For all $w \in S_{\Sigma}$, $q_1 \in Q_1$,*

$$(q_1, 1) \in [\![w]\!]_{\mathcal{A}} \iff \exists w_1, w_2.\, w \in w_1 \otimes_x w_2 \wedge q_1 \in [\![w_1]\!]_{\mathcal{A}_1} \wedge w_2 \in L_2$$

Proof. Both directions: by induction on the structure of w.

$\Rightarrow$:

Base case: $w = \varepsilon$. It must be the case that there are some q_1', q_1'', q_2 with:

- $(q_1'', q_2) \in [\![\varepsilon]\!]_{\mathcal{A}}$, and hence $q_1'' \in [\![\varepsilon]\!]_{\mathcal{A}_1}$ and $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$;
- $(q_1', 1) \in f^{\varepsilon}((q_1'', q_2))$, and hence $q_1' \in f_1^x(q_1'')$ and $q_2 \in A_2$, so $q_1' \in [\![x]\!]_{\mathcal{A}_2}$ and $\varepsilon \in L_2$; and
- $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}((q_1', 1))$, and hence $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q_1')$, so $q_1 \in [\![x]\!]_{\mathcal{A}_2}$.

Thus, $w_1 = x$ and $w_2 = \varepsilon$ fit the requirements: $\varepsilon \in x \otimes_x \varepsilon$.

Inductive case: $w = w' \cdot l$ for some w' and some $l \in \Sigma$. There must be some q_1' with either:

- $(q_1, 1) \in (\varepsilon\text{-}\mathsf{closure} \circ f^l)((q_1', 1))$ and $(q_1', 1) \in [\![w']\!]_{\mathcal{A}}$; or
- $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}((q_1', 1))$ and $(q_1', 1) \in f^{\varepsilon}((q_1'', q_2))$ for some q_1'', q_2 with $(q_1'', q_2) \in [\![w]\!]_{\mathcal{A}}$.

In the former case, by the inductive hypothesis, there are w_1' and w_2 with $w' \in w_1' \otimes_x w_2$, $q_1' \in [\![w_1']\!]_{\mathcal{A}_1}$, and $w_2' \in L_2$. By the definition of $\mathcal{A}$, $q_1 \in (\varepsilon\text{-}\mathsf{closure}_1 \circ f_1^l)(q_1)$ and so $q_1 \in [\![w_1' \cdot l]\!]_{\mathcal{A}_1}$. Observing that $(w_1' \otimes_x w_2) \cdot \{l\} \subseteq (w_1' \cdot l) \otimes_x w_2$, the sequences $w_1 = w_1' \cdot l$ and w_2 fit the requirements.

In the latter case, by Lem. 5 there are w_1', w_2 with $w = w_1' \cdot w_2$, $q_1'' \in [\![w_1']\!]_{\mathcal{A}_1}$ and $q_2 \in [\![w_2]\!]_{\mathcal{A}_2}$. It follows, by the definition of f^{ε}, that $q_1' \in [\![w_1' \cdot x]\!]_{\mathcal{A}_1}$ and $q_2 \in A_2$. Thus $w_1 = w_1' \cdot x$ and w_2 fit the requirements: $w = w_1' \cdot w_2 \in (w_1' \cdot x) \otimes_x w_2$, $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q_1') \subseteq [\![w_1]\!]_{\mathcal{A}_1}$ and $w_2 \in L_2$.

$\Leftarrow$:

Base case: $w = \varepsilon$. In this case, $w_1 = x$ and $w_2 = \varepsilon$. Since $q_1 \in [\![w_1]\!]_{\mathcal{A}_1}$, it follows that $q_1 \in (\varepsilon\text{-closure}_1 \circ f_1^x)(q_1')$ for some $q_1' \in [\![\varepsilon]\!]_{\mathcal{A}_1}$. Hence, by Lem. 5, $(q_1', q_2) \in [\![\varepsilon]\!]_{\mathcal{A}}$. Thus, by the definition of f^ε, $(q_1, 1) \in \varepsilon\text{-closure}((q_1', q_2))$ and so $(q_1, 1) \in [\![w]\!]_{\mathcal{A}}$ as required.

Inductive case: $w = w' \cdot l$ for some w' and some $l \in \Sigma$. Either:

- $w = w_1'' \cdot w_2$ and $w_1 = w_1'' \cdot x$; or
- $w' \in w_1' \otimes_x w_2$ and $w_1 = w_1' \cdot l$.

In the former case, there is a $q_1'' \in [\![w_1'']\!]_{\mathcal{A}_1}$ such that $q_1 \in (\varepsilon\text{-closure}_1 \circ f_1^x)(q_1'')$, and a $q_2 \in [\![w_2]\!]_{\mathcal{A}_2} \cap A_2$. By Lem. 5, $(q_1'', q_2) \in [\![w_1'' \cdot w_2]\!]_{\mathcal{A}} = [\![w]\!]_{\mathcal{A}}$. It follows, using the definition of f^ε, that $(q_1, 1) \in \varepsilon\text{-closure}((q_1'', q_2))$, and hence $(q_1, 1) \in [\![w]\!]_{\mathcal{A}}$, as required.

In the latter case, there is a $q_1' \in [\![w_1']\!]_{\mathcal{A}_1}$ such that $q_1 \in (\varepsilon\text{-closure}_1 \circ f_1^l)(q_1')$. By the inductive hypothesis, $(q_1', 1) \in [\![w']\!]_{\mathcal{A}}$. By the definition of $\mathcal{A}$, $(q_1, 1) \in (\varepsilon\text{-closure} \circ f^l)((q_1', 1))$, and hence $(q_1, 1) \in [\![w]\!]_{\mathcal{A}}$ as required. □

Proposition 1. *The automaton defined in Def. 9 accepts the language $L_1 \otimes_x L_2$.*

Proof.

$$\begin{array}{ll} & w \in L_1 \otimes_x L_2 \\ \iff & \exists w_1, w_2.\, w \in w_1 \otimes_x w_2 \wedge w_1 \in L_1 \wedge w_2 \in L_2 \\ \iff & \exists q_1 \in A_1.\, (q_1, 1) \in [\![w]\!]_{\mathcal{A}} \quad \text{(by Lem. 6)} \\ \iff & A \cap [\![w]\!]_{\mathcal{A}} \neq \emptyset. \end{array}$$

□

Correctness of the '$-\!\!\otimes^{\exists}$' construction Given ε-NFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^l\}_{l \in \Sigma \cup \{\varepsilon\}}, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^l\}_{l \in \Sigma \cup \{\varepsilon\}}, A_2)$ accepting languages L_1 and L_2 respectively, let $\mathcal{A} = \mathcal{A}_1 -\!\!\otimes_x^{\exists} \mathcal{A}_2 = (Q, e, \{f^l\}_{l \in \Sigma \cup \{\varepsilon\}}, A)$ as per Def. 10.

Lemma 7. *For all words, w, and all $q_2 \in Q_2$*

$$(q_2, 1) \in [\![w]\!]_{\mathcal{A}} \iff \exists w_1, w_2.\, w_1 \in L_1 \wedge q_2 \in [\![w_2]\!]_{\mathcal{A}_2} \wedge w_2 \in w \otimes_x w_1.$$

Proof. $\Rightarrow$:
Supposing that $(q_2, 1) \in [\![w]\!]_{\mathcal{A}}$, it must be the case that there exist q_2', q_2'' and w', w'' such that

$$\begin{gathered} w = w'' \cdot x \cdot w' \\ (q_2, 1) \in (\!|w'|\!)_{\mathcal{A}}((q_2', 1)) \\ (q_2', 1) \in f^x((q_2'', 0)) \\ (q_2'', 0) \in [\![w'']\!]_{\mathcal{A}}. \end{gathered}$$

By the definition of $\mathcal{A}$, we see that $q_2'' \in [\![w'']\!]_{\mathcal{A}_2}$. Similarly, there exists some $w_1 \in L_1$ such that $q_2' \in (\!|w_1|\!)_{\mathcal{A}_2}$, and hence $q_1' \in [\![w'' \cdot w_1]\!]_{\mathcal{A}_2}$. Further, $q_2 \in (\!|w'|\!)_{\mathcal{A}_2}(q_2')$ and so $q_2 \in [\![w'' \cdot w_1 \cdot w']\!]_{\mathcal{A}_2}$. If we let $w_2 = w'' \cdot w_1 \cdot w'$, clearly $w_2 \in w \oslash_x w_1$. Thus, w_1 and w_2 fit the requirements.

$\Leftarrow$:
Supposing that there are w_1 and w_2 with $w_1 \in L_1$, $q_2 \in [\![w_2]\!]_{\mathcal{A}_2}$ and $w_2 \in w \oslash_x w_1$, it must be the case that there exist q_2', q_2'' and w', w'' such that

$$\begin{aligned} w &= w'' \cdot x \cdot w' \\ w_2 &= w'' \cdot w_1 \cdot w' \\ q_2 &\in (\!|w'|\!)_{\mathcal{A}_2}(q_2') \\ q_2' &\in (\!|w_1|\!)_{\mathcal{A}_2}(q_2'') \\ q_2'' &\in [\![w'']\!]_{\mathcal{A}_2}. \end{aligned}$$

It follows from the definition of $\mathcal{A}$ that $(q_2'', 0) \in [\![w'']\!]_{\mathcal{A}}$. Similarly, since $w_1 \in L_1$, $(q_2', 1) \in f^x((q_2'', 0))$ and so $(q_2', 1) \in [\![w'' \cdot x]\!]_{\mathcal{A}}$. Further, $(q_2, 1) \in [\![w'' \cdot x \cdot w']\!]_{\mathcal{A}} = [\![w]\!]_{\mathcal{A}}$ as required. □

Proposition 2. *The automaton defined in Def. 10 accepts the language $L_1 \multimap\!\!\oslash^{\exists}_x L_2$.*

Proof.

$$\begin{aligned} & w \in L_1 \multimap\!\!\oslash^{\exists}_x L_2 \\ \iff & \exists w_1, w_2.\, w_1 \in L_1 \ \wedge\ w_2 \in L_2 \ \wedge\ w_2 \in w \oslash_x w_1 \\ \iff & \exists q_2 \in A_2.\, \exists w_1, w_2.\, w_1 \in L_1 \ \wedge\ q_2 \in [\![w_2]\!]_{\mathcal{A}_2} \ \wedge\ w_2 \in w \oslash_x w_1 \\ \iff & \exists q_2 \in A_2.\, (q_2, 1) \in [\![w]\!]_{\mathcal{A}} \\ \iff & [\![w]\!]_{\mathcal{A}} \cap A \neq \emptyset. \end{aligned}$$

□

Correctness of the '$\oslash\!\!-^{\exists}$' construction Given ε-NFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^l\}_{l \in \Sigma \cup \{\varepsilon\}}, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^l\}_{l \in \Sigma \cup \{\varepsilon\}}, A_2)$ accepting languages L_1 and L_2 respectively, let $\mathcal{A} = \mathcal{A}_1 \oslash\!\!-^{\exists}_x \mathcal{A}_2 = (Q, e, \{f^l\}_{l \in \Sigma \cup \{\varepsilon\}}, A)$ as per Def. 11.

Lemma 8. *For all $w \in S_\Sigma$,*

$$[\![w]\!]_{\mathcal{A}} = \{(\!|w|\!)_{\mathcal{A}_2}\}.$$

Proof. By induction on the structure of w. Note that, since $f^\varepsilon = \emptyset$, ε-closure is the identity relation.

Base case: $w = \varepsilon$.

$$\begin{aligned}
[\![\varepsilon]\!]_{\mathcal{A}} &= \varepsilon\text{-}\mathsf{closure}(e) \\
&= \varepsilon\text{-}\mathsf{closure}(\varepsilon\text{-}\mathsf{closure}_2) \\
&= \{\varepsilon\text{-}\mathsf{closure}_2\} \\
&= \{(\![\varepsilon]\!)_{\mathcal{A}_2}\}.
\end{aligned}$$

Inductive case: $w = w' \cdot l$ for some $w' \in S_\Sigma$, $l \in \Sigma$.

$$\begin{aligned}
[\![w' \cdot l]\!]_{\mathcal{A}} &= \{q \mid q' \in [\![w']\!]_{\mathcal{A}}, q \in (\varepsilon\text{-}\mathsf{closure} \circ f^l)(q')\} \\
\text{(by IH)} \quad &= \{q \mid q \in (\varepsilon\text{-}\mathsf{closure} \circ f^l)((\![w']\!)_{\mathcal{A}_2})\} \\
&= f^l((\![w']\!)_{\mathcal{A}_2}) \\
&= \{\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^l \circ (\![w']\!)_{\mathcal{A}_2}\} \\
&= \{(\![w' \cdot l]\!)_{\mathcal{A}_2}\}.
\end{aligned}$$

□

For $r \in Q$, let $\mathcal{A}_r$ be as given in Def. 11.

Lemma 9. *Suppose that $r = (\![w]\!)_{\mathcal{A}_2}$ for some $w \in S_\Sigma$. Then for $w_1 \in S_\Sigma$, for all $q_2 \in Q_2$,*

$$(q_2, 1) \in [\![w_1]\!]_{\mathcal{A}_r} \iff \exists w_2 \in S_\Sigma.\, q_2 \in [\![w_2]\!]_{\mathcal{A}_2} \wedge w_2 \in w_1 \oslash_x w.$$

Proof. $\Rightarrow$:
Supposing $(q_2, 1) \in [\![w_1]\!]_{\mathcal{A}_r}$, it must be the case that there exist $q_2', q_2'' \in Q_2$ and $w_1', w_1'' \in S_\Sigma$ such that

$$\begin{aligned}
w_1 &= w_1'' \cdot x \cdot w_1' \\
(q_2, 1) &\in (\![w_1']\!)_{\mathcal{A}_r}((q_2', 1)) \\
(q_2', 1) &\in f_r^x((q_2'', 0)) \\
(q_2'', 0) &\in [\![w_1'']\!]_{\mathcal{A}_r}.
\end{aligned}$$

By the definition of $\mathcal{A}_r$, we see that $q_2'' \in [\![w_1'']\!]_{\mathcal{A}_2}$. Similarly, $q_2' \in r(q_2'') = (\![w]\!)_{\mathcal{A}_2}(q_2'')$ and so $q_2' \in [\![w_1'' \cdot w]\!]_{\mathcal{A}_2}$. Futhermore, $q_2 \in (\![w_1']\!)_{\mathcal{A}_2}(q_1')$ and so $q_2 \in [\![w_1'' \cdot w \cdot w_1']\!]_{\mathcal{A}_2}$. Taking $w_2 = w_1'' \cdot w \cdot w_1'$, we have $q_1 \in [\![w_2]\!]_{\mathcal{A}_2}$ and $w_2 \in w_1 \oslash_x w$, as required.

$\Leftarrow$:
Supposing that $w_2 \in S_\Sigma$ is some word such that $q_2 \in [\![w_2]\!]_{\mathcal{A}_2}$ and $w_2 \in w_1 \oslash_x w$, it must be the case that there exist $q_2', q_2'' \in Q_2$ and $w_1', w_1'' \in S_\Sigma$ such that

$$\begin{aligned}
w_1 &= w_1'' \cdot x \cdot w_1' \\
w_2 &= w_1'' \cdot w \cdot w_1' \\
q_2 &\in (\![w_1']\!)_{\mathcal{A}_2}(q_2') \\
q_2' &\in (\![w]\!)_{\mathcal{A}_2}(q_2'') \\
q_2'' &\in [\![w_1'']\!]_{\mathcal{A}_2}.
\end{aligned}$$

It follows from the definition of $\mathcal{A}_r$ that $(q_2'', 0) \in [\![w_1'']\!]_{\mathcal{A}_r}$. Similarly, $(q_2', 1) \in f_r^x$ and so $(q_2', 1) \in [\![w_1'' \cdot x]\!]_{\mathcal{A}_r}$. Furthermore, $(q_2, 1) \in (\!|w_1'|\!)_{\mathcal{A}_r}(q_2')$ and so $(q_2, 1)[\![w_1'' \cdot x \cdot w_1']\!]_{\mathcal{A}_r} = [\![w_1]\!]_{\mathcal{A}_r}$, as required. □

Proposition 3. *The automaton defined in Def. 11 accepts the language* $L_1 \oslash\!\!-^{\exists}_{x} L_2$.

Proof.

$$
\begin{aligned}
& w \in L_1 \oslash\!\!-^{\exists}_{x} L_2 \\
\iff & \exists w_1, w_2.\, w_1 \in L_1 \wedge w_2 \in L_2 \wedge w_2 \in w_1 \oslash_x w \\
\iff & \exists w_1, w_2.\, \exists q_2.\, w_1 \in L_1 \wedge q_2 \in [\![w_2]\!]_{\mathcal{A}_2} \wedge q_2 \in A_2 \wedge w_2 \in w_1 \oslash_x w \\
\iff & \exists r.\, r = (\!|w|\!)_{\mathcal{A}_2} \wedge \exists w_1, w_2.\, \exists q_2.\, w_1 \in L_1 \wedge q_2 \in [\![w_2]\!]_{\mathcal{A}_2} \wedge q_2 \in A_2 \wedge w_2 \in w_1 \oslash_x w \\
\iff & \exists r.\, r = (\!|w|\!)_{\mathcal{A}_2} \wedge \exists w_1.\, \exists q_2.\, w_1 \in L_1 \wedge (q_2, 1) \in [\![w_1]\!]_{\mathcal{A}_r} \wedge (q_2, 1) \in A_r \\
\iff & \exists r.\, r = (\!|w|\!)_{\mathcal{A}_2} \wedge \exists w_1.\, [\![w_1]\!]_{\mathcal{A}_1} \cap A_1 \neq \emptyset \wedge [\![w_1]\!]_{\mathcal{A}_r} \cap A_r \neq \emptyset \\
\iff & \exists r.\, r = (\!|w|\!)_{\mathcal{A}_2} \wedge r \in A \\
\iff & \exists r.\, r = [\![w]\!]_{\mathcal{A}} \wedge r \in A \\
\iff & [\![w]\!]_{\mathcal{A}} \cap A \neq \emptyset.
\end{aligned}
$$

□

A.2 Sequence Expressivity

Lemma 10. *The aperiodicity number of* $S_{\Upsilon,\Omega}$ *is* 2.

Proof. For w_1, w_2, w_3 with $w_1 \cdot w_2^2 \cdot w_3 \in S_{\Upsilon,\Omega}$, it must be the case that no $x \in \Omega$ occurs in w_2 and may only occur linearly in one of w_1 or w_3, and hence $w_1 \cdot w_2^3 \cdot w_3 \in S_{\Upsilon,\Omega}$. Likewise, for w_1, w_2, w_3 with $w_1 \cdot w_2^3 \cdot w_3 \in S_{\Upsilon,\Omega}$, it must be the case that no $x \in \Omega$ occurs in w_2 and may only occur linearly in one of w_1 or w_3, and hence $w_1 \cdot w_2^2 \cdot w_3 \in S_{\Upsilon,\Omega}$.

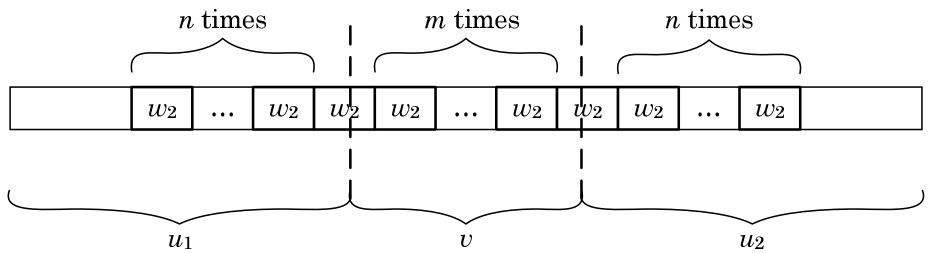

Fig. 1. A pathological splitting of $w = w_1 \cdot w_2^{(2n+m+2)} \cdot w_3$ as $u_1 \cdot v \cdot u_2$

Lemma 11. *Suppose that L_1 and L_2 are star-free regular languages with aperiodicity numbers n and m respectively. The aperiodicity number of $L_1 \oslash_x L_2$ is no greater than $2n + m + 2$.*

Proof. Suppose that $w = w_1 \cdot w_2^{(2n+m+2)} \cdot w_3 \in L_1 \oslash_x L_2$. Then $w = u_1 \cdot v \cdot u_2$ for some u_1, u_2, v with $u_1 \cdot x \cdot u_2 \in L_1$ and $v \in L_2$. One of the following must then be the case: $u_1 = w'_1 \cdot w_2^n \cdot w'_3$, $v = w'_1 \cdot w_2^m \cdot w'_3$ or $u_2 = w'_1 \cdot w_2^n \cdot w'_3$, such that the block of repeated w_2s are part of the $w_2^{(2n+m+2)}$ block of w. (To see this, consider Fig. 1. No alternative splitting can decrease the number of consecutive w_2s in all three blocks.) Correspondingly, by aperiodicity, either $w'_1 \cdot w_2^{(n+1)} \cdot w'_3 \cdot x \cdot u_2 \in L_1$, $w'_1 \cdot w_2^{(m+1)} \cdot w'_3 \in L_2$ or $u_1 \cdot x \cdot w'_1 \cdot w_2^{(n+1)} \cdot w'_3 \in L_1$. Hence, $w_1 \cdot w_2^{(2n+m+3)} \cdot w_3 \in L_1 \oslash_x L_2$.

All of the implications in the above can be reversed to prove the converse. In order to reverse the final implication, we require the bound on the aperiodicity number to be $2n + m + 2$, since a splitting such as in Fig. 1 could be possible otherwise.

Lemma 12. *Suppose that L_1 and L_2 are star-free regular languages with aperiodicity numbers n and m respectively. The aperiodicity number of $L_1 -\!\!\oslash_x^\exists L_2$ is no greater than $2m + 1$.*

Proof. Suppose that $w = w_1 \cdot w_2^{(2m+1)} \cdot w_3 \in L_1 -\!\!\oslash_x^\exists L_2$. Then $w = u_1 \cdot x \cdot u_2$ and there is some $v \in L_1$ such that $u_1 \cdot v \cdot u_2 \in L_2$. Either $u_1 = w'_1 \cdot w_2^m \cdot w'_3$ or $u_2 = w'_1 \cdot w_2^m \cdot w'_3$. Correspondingly, by aperiodicity of L_2, either $w'_1 \cdot w_2^{(m+1)} \cdot w'_3 \cdot v \cdot u_2 \in L_2$ or $u_1 \cdot v \cdot w'_1 \cdot w_2^{(m+1)} \cdot w'_3 \in L_2$. Hence $w_1 \cdot w_2^{(2m+2)} \cdot w_3 \in L_1 -\!\!\oslash_x^\exists L_2$.

Conversely, suppose that $w_1 \cdot w_2^{(2m+2)} \cdot w_3 \in L_1 -\!\!\oslash_x^\exists L_2$. Then $w = u_1 \cdot x \cdot u_2$ and there is some $v \in L_1$ such that $u_1 \cdot v \cdot u_2 \in L_2$. Either $u_1 = w'_1 \cdot w_2^{(m+1)} \cdot w'_3$ or $u_2 = w'_1 \cdot w_2^{(m+1)} \cdot w'_3$. Correspondingly, by aperiodicty of L_2, either $w'_1 \cdot w_2^m \cdot w'_3 \cdot v \cdot u_2 \in L_2$ or $u_1 \cdot v \cdot w'_1 \cdot w_2^m \cdot w'_3 \in L_2$. Hence $w_1 \cdot w_2^{(2m+1)} \cdot w_3 \in L_1 -\!\!\oslash_x^\exists L_2$.

Lemma 13. *Suppose that L_1 and L_2 are star-free regular languages with aperiodicity numbers n and m respectively. The aperiodicity number of $L_1 \oslash\!\!-_x^\exists L_2$ is no greater than m.*

Proof. Suppose that $w = w_1 \cdot w_2^m \cdot w_3 \in L_1 \oslash\!\!-_x^\exists L_2$. This is the case if and only if for some u_1, u_2, $u_1 \cdot w_1 \cdot w_2^m \cdot w_3 \cdot u_2 \in L_2$. By the aperiodicity of L_2, this is the case if and only if $u_1 \cdot w_1 \cdot w_2^{(m+1)} \cdot w_3 \cdot u_2 \in L_2$. This in turn is the case if and only if $w_1 \cdot w_2^{(m+1)} \cdot w_3 \in L_1 \oslash\!\!-_x^\exists L_2$.

A.3 Ranked Tree Constructions

Automata As with sequences, we are motivated to consider automata for ranked trees in order to decide model checking and satisfiability. Our definition of automata for ranked trees generalises the definition for words in a natural fashion. We deal only with frontier-to-root (or bottom-up) automata; for a comprehensive treatment of automata for ranked trees, see [20, 21].

Definition 21 (ε-NFTA). *A* non-deterministic finite (ranked) tree automaton with ε-transitions*, abbreviated ε-NFTA, is a tuple $\mathcal{A} = (Q, \{f^l\}_{l\in\Sigma\uplus\{\varepsilon\}}, A)$ where:*

- *Q is the set of states, a finite set;*
- *for every $l \in \Sigma$, $f^l \subseteq S^{n_l+1}$ is the state transition relation for l, which is an (n_l+1)-ary relation on S where n_l is the arity of l;*
- *$f^\varepsilon \subseteq S \times S$ is the non-consuming state transition relation, a binary relation;*
- *$A \subseteq S$ is the set of accepting states.*

Given $q_1, \dots q_{n_l} \in Q$, the notation $f^l(q_1, \dots, q_{n_l})$ is used for the set $\{q' \mid (q_1, \dots, q_{n_l}, q) \in f^l\}$. A pre-automaton *is an automaton without a set of accepting states,* i.e. *$\hat{\mathcal{A}} = (Q, \{f^l\}_{l\in\Sigma\uplus\{\varepsilon\}})$.*

To formally define the language recognised by an automaton, we make some auxilliary definitions. As for sequences, the ε-closure of a state is the set of states reachable by any number of ε-transitions: ε-closure is the reflexive-transitive closure of f^ε. Each automaton, $\mathcal{A}$, induces a function, $[\![-]\!]_\mathcal{A} : R_\Sigma \to \mathcal{P}(Q)$, that maps each ranked tree to a set of states according to the following definition:

$$[\![l(r_1, \dots, r_n)]\!]_\mathcal{A} = \left\{q \mid q_1 \in [\![r_1]\!]_\mathcal{A}, \dots, q_n \in [\![r_n]\!]_\mathcal{A}, q \in (\varepsilon\text{-closure} \circ f^l)(q_1, \dots, st_n)\right\}.$$

A tree r is said to be *accepted* by $\mathcal{A}$ if $[\![r]\!]_\mathcal{A} \cap A \neq \emptyset$. The (tree) language $L_\mathcal{A}$ defined by $\mathcal{A}$ is the set $\{r \in R_\Sigma \mid [\![r]\!]_\mathcal{A} \cap A \neq \emptyset\}$.

The language membership and emptiness problems for tree automata are decidable in a similar fashion to the equivalent problems for automata on words. The class of languages definable by tree automata is the class of *regular tree languages*. This class is closed under a number of operations; if $L, L', L_1, \dots, L_n$ are regular tree languages, then so are

- $\emptyset$, $R_{\Upsilon,\Omega}$, R_Σ,
- $L \cup L'$, $L \cap L'$, $\overline{L} \triangleq R_\Sigma \setminus L$, and
- $\{a(r_1, \cdots, r_{n_a}) \mid r_i \in L_i\}$ where $a \in \Upsilon$ has rank n_a.

For details, consult [20]. Other closure properties that are known in the literature could be used to implement the structural connectives of CL^{m}_{Term}. However, we present constructions analogous to those in the sequence case to implement these connectives.

Correctness of '$\oslash$' Construction

Definition 22 ('$\oslash$' Construction). *Given ε-NFTA $\mathcal{A}_1 = (Q_1, \{f_1^l\}, A_1)$ and $\mathcal{A}_2 = (Q_2, \{f_2^l\}, A_2)$ accepting languages L_1 and L_2 respectively, define the ε-NFTA $\mathcal{A} = \mathcal{A}_1 \oslash_x \mathcal{A}_2 = (Q, \{f^l\}, A)$ as follows.*

- *$Q = (Q_1 \times \{0,1\}) \uplus Q_2$;*
- *for $l \in \Sigma$ with rank m, f^l is the smallest relation satisfying:*
  - *$(q_1', n) \in f^l((q_{1,1}, n_1), \dots, (q_{1,m}, n_m))$ if $q_1' \in f_1^l(q_{1,1}, \dots, q_{1,m})$ and $n = \sum_{i=1}^m n_i$, and*

  - $q'_2 \in f^l(q_{2,1}, \ldots, q_{2,m})$ *if* $q'_2 \in f^l_2(q_{2,1}, \ldots, q_{2,m})$;
- f^ε *is the smallest relation satisfying:*
  - $(q'_1, n) \in f^\varepsilon((q_1, n))$ *whenever* $q'_1 \in f^\varepsilon_1((q_1, n))$,
  - $q'_2 \in f^\varepsilon(q_2)$ *whenever* $q'_2 \in f^\varepsilon_2(q_2)$, *and*
  - *for each* $q_1 \in f^x$ *and* $q_2 \in A_2$, $(q_1, 1) \in f^\varepsilon(q_2)$; *and*
- $A = A_1 \times \{1\}$.

Lemma 14. *For all* $r \in R_\Sigma$, $q_2 \in Q_2$,

$$q_2 \in [\![r]\!]_{\mathcal{A}} \iff q_2 \in [\![r]\!]_{\mathcal{A}_2}.$$

Proof. By induction on the structure of the tree r.

Base case: $r = l$ for some $l \in \Sigma$ with rank 0.

$$\begin{array}{rc}
 & q_2 \in [\![r]\!]_{\mathcal{A}} \\
\iff & \exists q'_2 \in Q_2.\, q'_2 \in f^l \wedge q_2 \in \varepsilon\text{-}\mathsf{closure}(q'_2) \\
\iff & \exists q'_2 \in Q_2.\, q'_2 \in f^l_2 \wedge q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q'_2) \\
\iff & q_2 \in [\![r]\!]_{\mathcal{A}_2}.
\end{array}$$

Inductive case: $r = l(r_1, \ldots, r_m)$ for some $l \in \Sigma$ with rank m and $r_1, \ldots, r_m \in R_\Sigma$.

$$\begin{array}{rrl}
 & & q_2 \in [\![r]\!]_{\mathcal{A}} \\
 & \iff & \exists q'_2, q_{2,1}, \ldots, q_{2,m} \in Q_2.\, q'_2 \in f^l(q_{2,1}, \ldots, q_{2,m}) \wedge \\
 & & \quad q_{2,1} \in [\![r_1]\!]_{\mathcal{A}} \wedge \cdots \wedge q_{2,m} \in [\![r_m]\!]_{\mathcal{A}} \wedge \\
 & & \quad q_2 \in \varepsilon\text{-}\mathsf{closure}(q'_2) \\
\text{(by IH)} & \iff & \exists q'_2, q_{2,1}, \ldots, q_{2,m} \in Q_2.\, q'_2 \in f^l(q_{2,1}, \ldots, q_{2,m}) \wedge \\
 & & \quad q_{2,1} \in [\![r_1]\!]_{\mathcal{A}_2} \wedge \cdots \wedge q_{2,m} \in [\![r_m]\!]_{\mathcal{A}_2} \wedge \\
 & & \quad q_2 \in \varepsilon\text{-}\mathsf{closure}(q'_2) \\
 & \iff & \exists q'_2, q_{2,1}, \ldots, q_{2,m} \in Q_2.\, q'_2 \in f^l_2(q_{2,1}, \ldots, q_{2,m}) \wedge \\
 & & \quad q_{2,1} \in [\![r_1]\!]_{\mathcal{A}_2} \wedge \cdots \wedge q_{2,m} \in [\![r_m]\!]_{\mathcal{A}_2} \wedge \\
 & & \quad q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q'_2) \\
 & \iff & q_2 \in [\![r]\!]_{\mathcal{A}_2}.
\end{array}$$

□

Lemma 15. *For all* $r \in R_\Sigma$, $q_1 \in Q_1$,

$$(q_1, 0) \in [\![r]\!]_{\mathcal{A}} \iff q_1 \in [\![r]\!]_{\mathcal{A}_1}.$$

Proof. By induction on the structure of the tree r.

Base case: $r = l$ for some $l \in \Sigma$ with rank 0.

$$\begin{array}{rl} & (q_1, 0) \in [\![r]\!]_{\mathcal{A}} \\ \iff & \exists q_1' \in Q_1.\, (q_1', 0) \in f^l \wedge (q_1, 0) \in \varepsilon\text{-}\mathsf{closure}((q_1', 0)) \\ \iff & \exists q_1' \in Q_1.\, q_1' \in f_1^l \wedge q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q_1') \\ \iff & q_1 \in [\![r]\!]_{\mathcal{A}_1}. \end{array}$$

Inductive case: $r = l(r_1, \ldots, r_m)$ for some $l \in \Sigma$ with rank m and $r_1, \ldots, r_m \in R_\Sigma$.

$$\begin{array}{rrl} & & (q_1, 0) \in [\![r]\!]_{\mathcal{A}} \\ & \iff & \exists q_1', q_{1,1}, \ldots, q_{1,m} \in Q_1.\, (q_1', 0) \in f^l((q_{1,1}, 0), \ldots, (q_{1,m}, 0)) \wedge \\ & & \quad (q_{1,1}, 0) \in [\![r_1]\!]_{\mathcal{A}} \wedge \cdots \wedge (q_{1,m}, 0) \in [\![r_m]\!]_{\mathcal{A}} \wedge \\ & & \quad (q_1, 0) \in \varepsilon\text{-}\mathsf{closure}(q_1') \\ \text{(by IH)} & \iff & \exists q_1', q_{1,1}, \ldots, q_{1,m} \in Q_1.\, (q_1', 0) \in f^l((q_{2,1}, 0), \ldots, (q_{2,m}, 0)) \wedge \\ & & \quad q_{1,1} \in [\![r_1]\!]_{\mathcal{A}_1} \wedge \cdots \wedge q_{1,m} \in [\![r_m]\!]_{\mathcal{A}_1} \wedge \\ & & \quad (q_1, 0) \in \varepsilon\text{-}\mathsf{closure}((q_1', 0)) \\ & \iff & \exists q_1', q_{1,1}, \ldots, q_{1,m} \in Q_1.\, q_1' \in f_1^l(q_{1,1}, \ldots, q_{1,m}) \wedge \\ & & \quad q_{1,1} \in [\![r_1]\!]_{\mathcal{A}_1} \wedge \cdots \wedge q_{1,m} \in [\![r_m]\!]_{\mathcal{A}_1} \wedge \\ & & \quad q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q_1') \\ & \iff & q_1 \in [\![r]\!]_{\mathcal{A}_1}. \end{array}$$

□

Lemma 16. *For all $r \in R_\Sigma$, $q_1 \in Q_1$,*

$$(q_1, 1) \in [\![r]\!]_{\mathcal{A}} \iff \exists r_1, r_2 \in R_\Sigma.\, r \in r_1 \oslash_x r_2 \wedge r_2 \in L_2 \wedge q_1 \in [\![r_1]\!]_{\mathcal{A}_1}.$$

Proof. By induction on the structure of the tree r.

Base case: $r = l$ for some $l \in \Sigma$ with rank 0.

$$\begin{array}{rl} & (q_1, 1) \in [\![l]\!]_{\mathcal{A}} \\ \iff & \exists q_1' \in Q_1, q_2, q_2' \in Q_2.\, (q_1, 1) \in \varepsilon\text{-}\mathsf{closure}((q_1', 1)) \wedge (q_1', 1) \in f^\varepsilon(q_2) \wedge \\ & \quad q_2 \in \varepsilon\text{-}\mathsf{closure}(q_2') \wedge q_2' \in f^l \\ \iff & \exists q_1' \in Q_1, q_2, q_2' \in Q_2.\, q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q_1') \wedge q_1' \in f_1^x \wedge q_2 \in A_2 \wedge \\ & \quad q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2') \wedge q_2' \in f^l \\ \iff & q_1 \in [\![x]\!]_{\mathcal{A}_1} \wedge [\![l]\!]_{\mathcal{A}_2} \cap A_2 \neq \emptyset \\ \iff & r_1, r_2 \in R_\Sigma.\, l \in r_1 \oslash_x r_2 \wedge r_2 \in L_2 \wedge q_1 \in [\![r_1]\!]_{\mathcal{A}_1}. \end{array}$$

Inductive case: $r = l(r^{(1)}, \dots, r^{(m)})$ for some $l \in \Sigma$ with rank m and $r^{(1)}, \dots, r^{(m)} \in R_\Sigma$. We consider the implications in each direction separately.

$\Rightarrow$:

By definition, $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}(q')$ where $q' \in f^l(q^{(1)}, \dots, q^{(m)})$ for some $q' \in Q$, $q^{(i)} \in [\![r^{(i)}]\!]_\mathcal{A}$. From the definition of f^ε, either $q' = (q'_1, 1)$ for some $q'_1 \in Q_1$, or $q' = q_2$ for some $q_2 \in Q_2$. We consider each case.

In the first case, it must be the case that for exactly one k, $q^{(k)} = (q_{1,k}, 1)$ and for all $i \neq k$, $q^{(i)} = (q_{1,i}, 0)$. By the inductive hypothesis, there are $r'_1, r_2 \in R_\Sigma$ with $r^{(k)} \in r_1^{(k)} \oslash_x r_2$, $r_2 \in L_2$ and $q_{1,k} \in [\![r_1^{(k)}]\!]_{\mathcal{A}_1}$. By Lem. 15, for $i \neq k$, $q_{1,i} \in [\![r^{(i)}]\!]_{\mathcal{A}_1}$. By definition, $q'_1 \in f_1^l(q_{1,1}, \dots, q_{1,m})$, and so

$$q'_1 \in [\![l(r^{(1)}, \dots, r^{(k-1)}, r_1^{(k)}, r^{(k+1)}, \dots, r^{(m)})]\!]_{\mathcal{A}_1}.$$

Let

$$r_1 = l(r^{(1)}, \dots, r^{(k-1)}, r_1^{(k)}, r^{(k+1)}, \dots, r^{(m)})$$

and observe that $r \in r_1 \oslash_x r_2$. Further, since $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}((q'_1, 1))$, $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q'_1)$. Hence, $q_1 \in [\![r_1]\!]_{\mathcal{A}_1}$, as required.

In the second case, there must be some $q'_1 \in Q_1$ and $q'_2 \in Q_2$ such that $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}((q'_1, 1))$, $(q'_1, 1) \in f^\varepsilon(q'_2)$, and $q'_2 \in \varepsilon\text{-}\mathsf{closure}(q_2)$. By the definition of f^ε, it follows that $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q'_1)$, $q'_1 \in f_1^x$, $q'_2 \in A_2$ and $q'_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2)$. Hence, $q_1 \in [\![x]\!]_{\mathcal{A}_1}$. Further, since $q_2 \in [\![r]\!]_{\mathcal{A}_2}$ by Lem. 14, $q'_2 \in [\![r]\!]_{\mathcal{A}_2} \cap A_2$, and so $r \in L_2$. Let $r_1 = x$ and $r_2 = r$, and observe that $r \in r_1 \oslash_x r_2$, as required.

$\Leftarrow$:

Either $r_1 = x$ or $r_1 \neq x$. We consider each case.

In the first case, $r = r_2 \in L_2$ and so there is some $q_2 \in [\![r]\!]_{\mathcal{A}_2} \cap A_2$. By Lem. 14, $q_2 \in [\![r]\!]_\mathcal{A}$. Also, since $q_1 \in [\![x]\!]_{\mathcal{A}_1}$, there is some $q'_1 \in Q_1$ with $q'_1 \in f_1^x$ and $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q'_1)$. Hence, by the definition of f^ε, $(q'_1, 1) \in f^\varepsilon(q_2)$ and $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}((q'_1, 1))$. Thus, $(q_1, 1) \in [\![r]\!]_\mathcal{A}$ as required.

In the second case, it must be that, for some $l \in \Sigma$ of rank m, $r_{1,1}, \dots, r_{1,m}, r' \in R_\Sigma$, k with $1 \leq k \leq m$

$$\begin{aligned} r_1 &= l(r_{1,1}, \dots, r_{1,m}) \\ r &= l(r_{1,1}, \dots, r_{1,k-1}, r', r_{1,k+1}, \dots, r_{1,m}) \\ r' &\in r_{1,k} \oslash_x r_2. \end{aligned}$$

Since $q_1 \in [\![r_1]\!]_{\mathcal{A}_1}$ it follows that there are $q'_1, q_{1,1}, \dots, q_{1,m} \in Q_1$ with

$$\begin{aligned} & q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q'_1) \\ & q'_1 \in f_1^l(q_{1,1}, \dots, q_{1,m}) \\ \text{for } 1 \leq i \leq m \quad & q_{1,i} \in [\![r_{1,i}]\!]_{\mathcal{A}_1}. \end{aligned}$$

Hence, by the inductive hypothesis, $(q_{1,k}, 1) \in [\![r']\!]_\mathcal{A}$. By Lem. 14, $(q_{1,i}, 0) \in [\![r_{1,i}]\!]_\mathcal{A}$ for $1 \leq i \leq m$. By definition,

$$(q'_1, 1) \in f^l((q_{1,1}, 0), \dots, (q_{1,k-1}, 0), (q_{1,k}, 1), (q_{1,k+1}, 0), \dots, (q_{1,m}, 0)).$$

Also, $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}(q'_1, 1)$. Hence, $(q_1, 1) \in [\![r]\!]_\mathcal{A}$, as required. $\square$

Proposition 7. *The automaton defined in Def. 22 accepts the language* $L_1 \oslash_x L_2$.

Proof.

$$
\begin{array}{lll}
 & & r \in L_1 \oslash_x L_2 \\
 & \iff & \exists r_1, r_2.\, r \in r_1 \oslash_x r_2 \wedge r_1 \in L_1 \wedge r_2 \in L_2 \\
\text{(by Lem. 16)} & \iff & \exists q_1 \in A_1.\, (q_1, 1) \in [\![r]\!]_{\mathcal{A}} \\
 & \iff & A \cap [\![r]\!]_{\mathcal{A}} \neq \emptyset.
\end{array}
$$

□

Correctness of '$-\!\oslash^{\exists}$' Construction

Definition 23 ('$-\!\oslash^{\exists}$' Construction). *Given* ε-*NFTA* $\mathcal{A}_1 = (Q_1, \{f_1^l\}, A_1)$ *and* $\mathcal{A}_2 = (Q_2, \{f_2^l\}, A_2)$ *accepting languages* L_1 *and* L_2 *respectively, define the* ε-*NFTA* $\mathcal{A} = \mathcal{A}_1 -\!\oslash_x^{\exists} \mathcal{A}_2 = (Q, \{f^l\}, A)$ *as follows.*

- $Q = Q_2 \times \{0, 1\}$;
- *for* $l \in \Sigma \uplus \{\varepsilon\}$, *where* l *has rank* m ($m = 1$ *in the case of* ε), f^l *is the smallest relation satisfying*
  - $(q_2', n) \in f^l((q_{2,1}, n_1), \ldots, (q_{2,m}, n_m))$ *if* $q_2' \in f_2^l(q_{2,1}, \ldots, q_{2,m})$ *and* $n = \sum_{i=1}^{m} n_i$, *and*
  - *if* $l = x$, *then* $(q_2', 1) \in f^l$ *for every* $q_2' \in [\![r]\!]_{\mathcal{A}_2}$ *for any* $r \in L_1$; *and*
- $A = A_2 \times \{1\}$.

Lemma 17. *For all* $r \in R_\Sigma$, $q_2 \in Q_2$,

$$(q_2, 0) \in [\![r]\!]_{\mathcal{A}} \iff q_2 \in [\![r]\!]_{\mathcal{A}_2}.$$

Proof. By induction on the structure of the tree r.

Base case: $r = l$ for some $l \in \Sigma$ with rank 0.

$$
\begin{array}{ll}
 & (q_2, 0) \in [\![r]\!]_{\mathcal{A}} \\
\iff & \exists q_2' \in Q_2.\, (q_2', 0) \in f^l \wedge (q_2, 0) \in \varepsilon\text{-}\mathsf{closure}((q_2', 0)) \\
\iff & \exists q_2' \in Q_2.\, q_2' \in f_2^l \wedge q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2') \\
\iff & q_2 \in [\![r]\!]_{\mathcal{A}_2}.
\end{array}
$$

Inductive case: $r = l(r_1, \ldots, r_m)$ for some $l \in \Sigma$ with rank m, and $r_1, \ldots, r_m \in R_\Sigma$.

$$
\begin{aligned}
& && (q_2, 0) \in [\![r]\!]_{\mathcal{A}} \\
& \iff && \exists q_2', q_{2,1}, \ldots, q_{2,m} \in Q_2.\, (q_2', 0) \in f^l((q_{2,1}, 0), \ldots, (q_{2,m}, 0)) \wedge \\
& && \qquad (q_{2,1}, 0) \in [\![r_1]\!]_{\mathcal{A}} \wedge \cdots \wedge (q_{2,m}, 0) \in [\![r_m]\!]_{\mathcal{A}} \wedge \\
& && \qquad (q_2, 0) \in \varepsilon\text{-}\mathsf{closure}(q_2', 0) \\
\text{(by IH)} & \iff && \exists q_2', q_{2,1}, \ldots, q_{2,m} \in Q_2.\, (q_2', 0) \in f^l((q_{2,1}, 0), \ldots, (q_{2,m}, 0)) \wedge \\
& && \qquad q_{2,1} \in [\![r_1]\!]_{\mathcal{A}_2} \wedge \cdots \wedge q_{2,m} \in [\![r_m]\!]_{\mathcal{A}_2} \wedge \\
& && \qquad (q_2, 0) \in \varepsilon\text{-}\mathsf{closure}(q_2', 0) \\
& \iff && \exists q_2', q_{2,1}, \ldots, q_{2,m} \in Q_2.\, q_2' \in f_2^l(q_{2,1}, \ldots, q_{2,m}) \wedge \\
& && \qquad q_{2,1} \in [\![r_1]\!]_{\mathcal{A}_2} \wedge \cdots \wedge q_{2,m} \in [\![r_m]\!]_{\mathcal{A}_2} \wedge \\
& && \qquad q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2') \\
& \iff && \qquad q_2 \in [\![r]\!]_{\mathcal{A}_2}.
\end{aligned}
$$

□

Lemma 18. *For all $r \in R_\Sigma$, $q_2 \in Q_2$,*

$$(q_2, 1) \in [\![r]\!]_{\mathcal{A}} \iff \exists r_1, r_2 \in R_\Sigma.\, r_2 \in r \oslash_x r_1 \wedge r_1 \in L_1 \wedge q_2 \in [\![r_2]\!]_{\mathcal{A}_2}.$$

Proof. By induction on the structure of tree r.

Base case: $r = l$ for some $l \in \Sigma$ with rank 0.

$$
\begin{aligned}
& && (q_2, 1) \in [\![l]\!]_{\mathcal{A}} \\
& \iff && \exists q_2' \in Q_2.\, (q_2', 1) \in f^l \wedge (q_2, 1) \in \varepsilon\text{-}\mathsf{closure}((q_2', 1)) \\
& \iff && r = x \wedge \exists q_2' \in Q_2, r_1 \in L_1.\, q_2' \in [\![r_1]\!]_{\mathcal{A}_2} \wedge q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2') \\
& \iff && r = x \wedge \exists r_1 \in L_1.\, q_2 \in [\![r_1]\!]_{\mathcal{A}_2} \\
& \iff && r = x \wedge \exists r_1, r_2 \in R_\Sigma.\, r_2 \in r \oslash_x r_1 \wedge r_1 \in L_1 \wedge r_1 = r_2 \wedge q_2 \in [\![r_2]\!]_{\mathcal{A}_2} \\
& \iff && \exists r_1, r_2 \in R_\Sigma.\, r_2 \in r \oslash_x r_1 = l \oslash_x r_1 \wedge r_1 \in [\![r_2]\!]_{\mathcal{A}_2}.
\end{aligned}
$$

Inductive case: $r = l(r^{(1)}, \ldots, r^{(m)})$ for some $l \in \Sigma$ with rank m and $r^{(1)}, \ldots, r^{(m)} \in R_\Sigma$. Note that we do not consider $l = x$, since this is the base case. We consider the implications in each direction separately.

$\Rightarrow$:

For some $q_2, q_{2,1}, \ldots, q_{2,m} \in Q_2$ and some $1 \leq k \leq m$, it must be the case that

$$
\begin{aligned}
\text{for } i \neq k \quad & (q_{2,i}, 0) \in [\![r^{(i)}]\!]_{\mathcal{A}} \\
& (q_{2,k}, 1) \in [\![r^{(k)}]\!]_{\mathcal{A}} \\
& (q_2', 1) \in f^l((q_{2,1}, 0), \ldots, (q_{2,k-1}, 0), (q_{2,k}, 1), (q_{2,k+1}, 0), \ldots, (q_{2,m}, 0)) \\
& (q_2, 1) \in \varepsilon\text{-}\mathsf{closure}((q_2', 1)).
\end{aligned}
$$

By Lem. 17,

$$\text{for } i \neq k q_{2,i} \in [\![r^{(i)}]\!]_{\mathcal{A}_2}.$$

By the inductive hypothesis, there exist $r'_2, r_1 \in R_\Sigma$ such that the following hold:

$$\begin{aligned} r'_2 &\in r^{(k)} \oslash_x r_1 \\ r_1 &\in L_1 \\ q_{2,k} &\in [\![r'_2]\!]_{\mathcal{A}_2}. \end{aligned}$$

Let

$$r_2 = l(r^{(1)}, \ldots, r^{(k-1)}, r'_2, r^{(k+1)}, \ldots, r^{(m)})$$

and observe that

$$r_2 \in r \oslash_x r_1.$$

By the definition of $\mathcal{A}$,

$$\begin{aligned} q'_2 &\in f_2^l(q_{2,1}, \ldots, q_{2,m}) \\ q_2 &\in \varepsilon\text{-}\mathsf{closure}_2(q'_2). \end{aligned}$$

Hence

$$q_2 \in [\![l(r^{(1)}, \ldots, r^{(k-1)}, r'_2, r^{(k+1)}, \ldots, r^{(m)})]\!]_{\mathcal{A}_2} = [\![r_2]\!]_{\mathcal{A}_2}.$$

We have therefore shown the existence of $r_1, r_2 \in R_\Sigma$ such that

$$r_2 \in r \oslash_x r_1 \ \wedge \ r_1 \in L_1 \ \wedge \ q_2 \in [\![r_2]\!]_{\mathcal{A}_2}$$

as required.

$\Leftarrow$:

For some $1 \leq k \leq m$, and some r'_2, the following must hold:

$$\begin{aligned} r_2 &= l(r^{(1)}, \ldots, r^{(k-1)}, r'_2, r^{(k+1)}, \ldots, r^{(m)}) \\ r'_2 &\in r^{(k)} \oslash_x r_1. \end{aligned}$$

Since $q_2 \in [\![r_2]\!]_{\mathcal{A}_2}$, it must be that for some $q'_2, q_{2,1}, \ldots, q_{2,m} \in Q_2$,

$$\begin{aligned} \text{for } i \neq k \qquad q_{2,i} &\in [\![r^{(i)}]\!]_{\mathcal{A}_2} \\ q_{2,k} &\in [\![r'_2]\!]_{\mathcal{A}_2} \\ q'_2 &\in f_2^l(q_{2,1}, \ldots, q_{2,k-1}, q_{2,k}, q_{2,k+1}, \ldots, q_{2,m}) \\ q_2 &\in \varepsilon\text{-}\mathsf{closure}_2(q'_2). \end{aligned}$$

By Lem. 17,

$$\text{for } i \neq k (q_{2,i}, 0) \in [\![r^{(i)}]\!]_{\mathcal{A}}.$$

By the inductive hypothesis,

$$(q_{2,k}, 1) \in [\![r^{(k)}]\!]_{\mathcal{A}}.$$

By the definition of $\mathcal{A}$,

$$(q_2', 1) \in f^l((q_{2,1}, 0), \ldots, (q_{2,k-1}, 0), (q_{2,k}, 1), (q_{2,k+1}, 0), \ldots, (q_{2,m}, 0))$$
$$(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}(q_2', 1).$$

Therefore,

$$(q_2, 1) \in [\![l(r^{(1)}, \ldots, r^{(m)})]\!]_{\mathcal{A}} = [\![r]\!]_{\mathcal{A}}$$

as required. □

Proposition 8. *The automaton defined in Def. 23 accepts the language $L_1 \multimap\!\!\!\oslash^{\exists}_x L_2$.*

Proof.

$$\begin{array}{rcl} & & r \in L_1 \multimap\!\!\!\oslash^{\exists}_x L_2 \\ & \iff & \exists r_1, r_2.\, r_2 \in r \oslash_x r_2 \wedge r_1 \in L_1 \wedge r_2 \in L_2 \\ \text{(by Lem. 18)} & \iff & \exists q_2 \in A_2.\, (q_2, 1) \in [\![r]\!]_{\mathcal{A}} \\ & \iff & A \cap [\![r]\!]_{\mathcal{A}} \neq \emptyset. \end{array}$$

Correctness of '$\oslash\!\!-^{\exists}$' Construction

Definition 24 ('$\oslash\!\!-^{\exists}$' Construction). *Given ε-NFTA $\mathcal{A}_1 = (Q_1, \{f_1^l\}, A_1)$ and $\mathcal{A}_2 = (Q_2, \{f_2^l\}, A_2)$ accepting languages L_1 and L_2 respectively, define the ε-NFTA $\mathcal{A} = \mathcal{A}_1 \oslash\!\!-^{\exists}_x \mathcal{A}_2 = (Q, \{f^l\}, A)$ as follows.*

- $Q = Q_2$
- $f^l = f_2^l$ *for all* $l \in \Sigma \uplus \{\varepsilon\}$*; and*
- $q \in A$ *if and only if* $\exists r \in R_\Sigma.\, [\![r]\!]_{\hat{\mathcal{A}}_1 \times \hat{\mathcal{A}}_q} \cap (A_1 \times A_q) \neq \emptyset$ *where*
  - $\mathcal{A}_q = (Q_2 \times \{0, 1\}, \{f_q^l\}, A_q)$,
  - *for* $l \in \Sigma \uplus \{\varepsilon\}$*, where l has rank m ($m = 1$ in the case of ε), f_q^l is the smallest relation satisfying*
    - $(q_2', n) \in f_q^l((q_{2,1}, n_1), \ldots, (q_{2,m}, n_m))$ *if* $q_2' \in f_2^l(q_{2,1}, \ldots, q_{2,m})$ *and* $n = \sum_{i=1}^m n_i$*, and*
    - *if $l = x$, then $(q, 1) \in f_q^l$, and*
  - $A = A_2 \times \{1\}$.

Lemma 19. *For all $r \in R_\Sigma$, $q_2 \in Q = Q_2$,*

$$q \in [\![r]\!]_{\mathcal{A}} \iff q \in [\![r]\!]_{\mathcal{A}_2}.$$

Proof. By definition. □

For $q \in Q$, let $\mathcal{A}_q$ be given as in Def. 24.

Lemma 20. *For all $q \in Q$, for all $r \in R_\Sigma$, $q_2 \in Q_2$,*

$$(q_2, 0) \in [\![r]\!]_{\mathcal{A}_q} \iff q_2 \in [\![r]\!]_{\mathcal{A}_2}.$$

Proof. By definition. □

Lemma 21. *For all $r, r_1 \in R_\Sigma$, $q_2 \in Q_2$,*

$$\exists q \in Q_2.\, q \in [\![r]\!]_{\mathcal{A}_2} \wedge (q_2, 1) \in [\![r_1]\!]_{\mathcal{A}_q}$$
$$\Longleftrightarrow$$
$$\exists r_2 \in R_\Sigma.\, r_2 \in r_1 \oslash_x r \wedge q_2 \in [\![r_2]\!]_{\mathcal{A}_2}.$$

Proof. We consider each direction of implication separately, in each case proceeding by induction on the structure of the tree r_1.

$\Rightarrow$:

Base case: $r_1 = l$.

In this case, there must be some $q_2' \in Q_2$ with

$$(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}_q((q_2', 1))$$
$$(q_2', 1) \in f_q^l.$$

The definition of $\mathcal{A}_q$ means that $r_1 = l = x$ and $q_2' = q$. Let $r_2 = r$. Clearly, $r_2 \in r_1 \oslash_x r$. Also, by definition,

$$q_2' = q \in [\![r]\!]_{\mathcal{A}_2} = [\![r_2]\!]_{\mathcal{A}_2}$$
$$q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2'),$$

and so $q_2 \in [\![r_2]\!]_{\mathcal{A}_2}$, as required.

Inductive case: $r_1 = l(r^{(1)}, \dots, r^{(m)})$ for some m, $l \in \Sigma$ of rank m and $r^{(1)}, \dots, r^{(m)} \in R_\Sigma$.

In this case, since $(q_2, 1) \in [\![r_1]\!]_{\mathcal{A}_q}$, there must be some $q_2', q_{2,1}, \dots, q_{2,m}$ and $1 \le k \le m$ with

$$\text{for } i \neq k \quad (q_{2,i}, 0) \in [\![r^{(i)}]\!]_{\mathcal{A}_q}$$
$$(q_{2,k}, 1) \in [\![r^{(k)}]\!]_{\mathcal{A}_q}$$
$$(q_2', 1) \in f_q^l((q_{2,1}, 0), \dots, (q_{2,k-1}, 0), (q_{2,k}, 1), (q_{2,k+1}, 0), \dots, (q_{2,m}, 0))$$
$$(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}_q((q_2', 1)).$$

By the inductive hypothesis, there is some $r_2' \in R_\Sigma$ such that $r_2' \in r^{(k)} \oslash_x r$ and $q_2' \in [\![r_2']\!]_{\mathcal{A}_2}$. Let

$$r_2 = l(r^{(1)}, \dots, r^{(k-1)}, r_2', r^{(k+1)}, \dots, r^{(m)})$$

and observe that $r_2 \in r_1 \oslash_x r$. By Lem. 20, for each $i \neq k$, $q_{2,i} \in [\![r^{(i)}]\!]_{\mathcal{A}_2}$. By the definition of $\mathcal{A}_q$,

$$q_2' \in f_2^l(q_{2,1}, \dots, q_{2,m})$$
$$q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2'),$$

and so

$$q_2 \in [\![r_2]\!]_{\mathcal{A}_2}$$

as required.

$\Leftarrow$:

Base case: $r_1 = l$.

In this case, $r_1 = x$, since $r_2 \in r_1 \oslash_x r$. This means that $r_2 = r$. Let

$$q = q_2 \in [\![r_2]\!]_{\mathcal{A}_2} = [\![r]\!]_{\mathcal{A}_2}.$$

By definition,

$$(q_2, 1) = (q, 1) \in f_q^x \subseteq [\![r_1]\!]_{\mathcal{A}_q},$$

as required.

Inductive case: $r_1 = l(r^{(1)}, \ldots, r^{(m)})$ for some m, $l \in \Sigma$ of rank m and $r^{(1)}, \ldots, r^{(m)} \in R_\Sigma$.

In this case, there must be some $1 \leq k \leq m$ and some $r_2' \in R_\Sigma$ such that the following hold:

$$\begin{aligned} &r_2' \in r^{(k)} \oslash_x r \\ &r_2 = l(r^{(1)}, \ldots, r^{(k-1)}, r_2', r^{(k+1)}, \ldots, r^{(m)}). \end{aligned}$$

Since $q_2 \in [\![r_2]\!]_{\mathcal{A}_2}$, it follows that there are some $q_2', q_{2,1}, \ldots, q_{2,m}$ with

$$\begin{aligned} \text{for } i \neq k \qquad\qquad & q_{2,i} \in [\![r^{(i)}]\!]_{\mathcal{A}_2} \\ & q_{2,k} \in [\![r_2']\!]_{\mathcal{A}_2} \\ & q_2' \in f_2^l(q_{2,1}, \ldots, q_{2,m}) \\ & q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2'). \end{aligned}$$

By the inductive hypothesis, there is some $q \in Q_2$ such that $q \in [\![r]\!]_{\mathcal{A}_2}$ and $(q_{2,k}, 1) \in [\![r^{(k)}]\!]_{\mathcal{A}_q}$. By Lem. 20, for each $i \neq k$, $(q_{2,i}, 0) \in [\![r^{(i)}]\!]_{\mathcal{A}_q}$. By the definition of $\mathcal{A}_q$,

$$\begin{aligned} &(q_2', 1) \in f_q^l((q_{2,1}, 0), \ldots, (q_{2,k-1}, 0), (q_{2,k}, 1), (q_{2,k+1}, 0), \ldots, (q_{2,m}, 0)) \\ &(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}_q((q_2', 1)), \end{aligned}$$

and so

$$(q_2, 1) \in [\![r_1]\!]_{\mathcal{A}_q},$$

as required. $\square$

Proposition 9. *The automaton defined in Def. 24 accepts the language* $L_1 \oslash\!\!-_x^{\exists} L_2$.

Proof.

$$
\begin{array}{lll}
 & & r \in L_1 \oslash\!\!-^{\exists}_{x} L_2 \\
 & \iff & \exists r_1, r_2 \in R_\Sigma .\, r_2 \in r_1 \oslash_x r \wedge r_1 \in L_1 \wedge r_2 \in L_2 \\
 & \iff & \exists r_1 \in L_1, q_2 \in A_2 .\, \exists r_2 \in R_\Sigma .\, r_2 \in r_2 \oslash_x r \wedge q_2 \in [\![r_2]\!]_{\mathcal{A}_2} \\
\text{(by Lem. 21)} & \iff & \exists r_1 \in L_1, q_2 \in A_2 .\, \exists q \in Q_2 .\, q \in [\![r]\!]_{\mathcal{A}_2} \wedge (q_2, 1) \in [\![r_1]\!]_{\mathcal{A}_q} \\
\text{(by Lem. 17)} & \iff & \exists r_1 \in L_1, q_2 \in A_2 .\, \exists q \in Q .\, q \in [\![r]\!] \wedge (q_2, 1) \in [\![r_1]\!]_{\mathcal{A}_q} \\
 & \iff & \exists q \in [\![r]\!]_{\mathcal{A}} .\, \exists r_1 \in R_\Sigma, q_1 \in A_1, q_2 \in A_2 .\, q_1 \in [\![r_1]\!]_{\mathcal{A}_1} \wedge (q_2, 1) \in [\![r_1]\!]_{\mathcal{A}_q} \\
 & \iff & \exists q \in [\![r]\!]_{\mathcal{A}} .\, \exists r_1 \in R_\Sigma .\, [\![r]\!]_{\hat{\mathcal{A}}_1 \times \hat{\mathcal{A}}_q} \cap (A_1 \times A_q) \neq \emptyset \\
 & \iff & A \cap [\![r]\!]_{\mathcal{A}} \neq \emptyset .
\end{array}
$$

□

A.4 Unranked Tree Constructions

Correctness of '$\oslash$' Construction Given ε-NFFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^a\}_{a \in \Upsilon}, \{f_1^x\}_{x \in \Omega}, f_1^\varepsilon, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^a\}_{a \in \Upsilon}, \{f_2^x\}_{x \in \Omega}, f_2^\varepsilon, A_2)$ accepting languages L_1 and L_2 respectively, let $\mathcal{A} = \mathcal{A}_1 \oslash_x \mathcal{A}_2 = (Q, e, \{f^a\}_{a \in \Upsilon}, \{f^x\}_{x \in \Omega}, f^\varepsilon, A)$ as per Def. 17.

Lemma 22. *For all $t \in T_\Sigma$,*

$$e \in [\![t]\!]_{\mathcal{A}} \iff t = \varepsilon .$$

Proof. By definition, $e \in [\![\varepsilon]\!]_{\mathcal{A}}$. If $t \neq \varepsilon$, then each $q \in [\![t]\!]_{\mathcal{A}}$ must be the result of a transition from some other state(s) and e is not the result of any such transition. □

Lemma 23. *For all $t \in T_\Sigma$, $q_2 \in Q_2$,*

$$q_2 \in [\![t]\!]_{\mathcal{A}} \iff q_2 \in [\![t]\!]_{\mathcal{A}_2} .$$

Proof. By induction on the structure of the tree t.

Base case: $t = \varepsilon$:

Observe that $e_2 \in [\![\varepsilon]\!]_{\mathcal{A}}$, that $f_2^\varepsilon \subseteq f^\varepsilon$ and that $q_2 \in f^\varepsilon(q')$ implies $q' \in Q_2$ and $q_2 \in f_2^\varepsilon(q')$. Consequently, $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}}$ if and only if $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$.

Inductive case: $t = t' \mid y$ for some $y \in \Omega$:

$$
\begin{aligned}
q_2 \in [\![t]\!]_{\mathcal{A}} &\iff \exists q_2' \in Q .\, q_2 \in (\varepsilon\text{-}\mathsf{closure} \circ f^y)(q_2') \wedge q_2' \in [\![t']\!]_{\mathcal{A}} \\
&\iff \exists q_2' \in Q_2 .\, q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q_2') \wedge q_2' \in [\![t']\!]_{\mathcal{A}} \\
&\iff \exists q_2' \in Q_2 .\, q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q_2') \wedge q_2' \in [\![t']\!]_{\mathcal{A}_2} \\
&\iff q_2 \in [\![t' \mid y]\!]_{\mathcal{A}_2} = [\![t]\!]_{\mathcal{A}_2} .
\end{aligned}
$$

Inductive case: $t = t' \mid a[t'']$ for some $a \in \Upsilon$:

$$\begin{aligned}
q_2 \in [\![t]\!]_{\mathcal{A}} \iff\ & \exists q_2', q_2'', q_2''' \in Q.\, q_2 \in \varepsilon\text{-}\mathsf{closure}(q_2''') \wedge q_2''' \in f^a(q_2', q_2'') \\
& \wedge q_2' \in [\![t']\!]_{\mathcal{A}} \wedge q_2'' \in [\![t'']\!]_{\mathcal{A}} \\
\iff\ & \exists q_2', q_2'', q_2''' \in Q_2.\, q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2''') \wedge q_2''' \in f_2^a(q_2', q_2'') \\
& \wedge q_2' \in [\![t']\!]_{\mathcal{A}} \wedge q_2'' \in [\![t'']\!]_{\mathcal{A}} \\
\iff\ & \exists q_2', q_2'', q_2''' \in Q_2.\, q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2''') \wedge q_2''' \in f_2^a(q_2', q_2'') \\
& \wedge q_2' \in [\![t']\!]_{\mathcal{A}}\mathcal{A}_2 \wedge q_2'' \in [\![t'']\!]_{\mathcal{A}_2} \\
\iff\ & q_2 \in [\![t' \mid a[t'']]\!]_{\mathcal{A}_2} = [\![t]\!]_{\mathcal{A}_2}
\end{aligned}$$

□

Lemma 24. *For all $t \in T_\Sigma$, $q_1 \in Q_1$,*

$$(q_1, 0) \in [\![t]\!]_{\mathcal{A}} \iff q_1 \in [\![t]\!]_{\mathcal{A}_1}.$$

Proof. By induction on the structure of the tree t.

Base case: $t = \varepsilon$:
Observe that $(e_1, 0) \in [\![\varepsilon]\!]_{\mathcal{A}}$, and that, for all $q', q'' \in Q$ with $q' \in f^\varepsilon(q'')$,

$$\begin{aligned}
\exists q_1' \in Q_1.\, q' = (q_1', 0) &\implies \exists q_1'' \in Q_1.\, q'' = (q_1'', 0) \wedge q_1' \in f_1^\varepsilon(q_1'') \\
\exists q_1'' \in Q_1.\, q'' = (q_1'', 0) &\implies \exists q_1' \in Q_1.\, q' = (q_1', 0) \wedge q_1' \in f_1^\varepsilon(q_1'').
\end{aligned}$$

Consequently, $(q_1, 0) \in [\![\varepsilon]\!]_{\mathcal{A}}$ if and only if $q_1 \in [\![\varepsilon]\!]_{\mathcal{A}_1}$.

Inductive case: $t = t' \mid y$ for some $y \in \Omega$:

$$\begin{aligned}
(q_1, 0) \in [\![t]\!]_{\mathcal{A}} \iff\ & \exists q' \in Q.\, (q_1, 0) \in (\varepsilon\text{-}\mathsf{closure} \circ f^y)(q') \wedge q' \in [\![t']\!]_{\mathcal{A}} \\
\iff\ & \exists q_1' \in Q_1.\, q_1 \in (\varepsilon\text{-}\mathsf{closure}_1 \circ f_1^y)(q_1') \wedge (q_1', 0) \in [\![t']\!]_{\mathcal{A}} \\
\iff\ & \exists q_1' \in Q_1.\, q_1 \in (\varepsilon\text{-}\mathsf{closure}_1 \circ f_1^y)(q_1') \wedge q_1' \in [\![t']\!]_{\mathcal{A}_1} \\
\iff\ & q_1 \in [\![t' \mid y]\!]_{\mathcal{A}_1} = [\![t]\!]_{\mathcal{A}_1}.
\end{aligned}$$

Inductive case: $t = t' \mid a[t'']$ for some $a \in \Upsilon$:

$$\begin{aligned}
(q_1, 0) \in [\![t]\!]_{\mathcal{A}} \iff\ & \exists q', q'', q''' \in Q.\, q_2 \in \varepsilon\text{-}\mathsf{closure}(q''') \wedge q''' \in f^a(q', q'') \\
& \wedge q' \in [\![t']\!]_{\mathcal{A}} \wedge q'' \in [\![t'']\!]_{\mathcal{A}} \\
\iff\ & \exists q_1', q_1'', q_1''' \in Q_1.\, q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q_1''') \wedge q_1''' \in f_1^a(q_1', q_1'') \\
& \wedge (q_1', 0) \in [\![t']\!]_{\mathcal{A}} \wedge (q_1'', 0) \in [\![t'']\!]_{\mathcal{A}} \\
\iff\ & \exists q_1', q_1'', q_1''' \in Q_1.\, q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q_1''') \wedge q_1''' \in f_1^a(q_1', q_1'') \\
& \wedge q_1' \in [\![t']\!]_{\mathcal{A}}\mathcal{A}_1 \wedge q_1'' \in [\![t'']\!]_{\mathcal{A}_1} \\
\iff\ & q_1 \in [\![t' \mid a[t'']]\!]_{\mathcal{A}_1} = [\![t]\!]_{\mathcal{A}_1}
\end{aligned}$$

□

Lemma 25. *For all* $t \in T_\Sigma$*,* $q_1 \in Q_1$*,* $q_2 \in Q_2$*,*

$$(q_1, q_2) \in [\![t]\!]_{\mathcal{A}} \iff \exists t_1, t_2.\, t = t_1 \mid t_2 \wedge q_1 \in [\![t_1]\!]_{\mathcal{A}_1} \wedge q_2 \in [\![t_2]\!]_{\mathcal{A}_2}.$$

Proof. Both directions: by induction on the structure of t.

$\Longrightarrow$:

Base case: $t = \varepsilon$ and so $(q_1, q_2) \in \varepsilon\text{-}\mathsf{closure}(e)$. By the definition of $\mathcal{A}$, it follows that

- $(q_1, q_2) \in \varepsilon\text{-}\mathsf{closure}((q_1, e_2))$, and hence $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(e_2)$ and $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$;
- $(q_1, e_2) \in f^\varepsilon((q_1, 0))$; and
- $(q_1, 0) \in [\![\varepsilon]\!]_{\mathcal{A}}$ and so $q_1 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$ (by Lem. 24).

Inductive case: $t = t' \mid y$ for some $y \in \Omega$. In this case, $(q_1, q_2) \in (\varepsilon\text{-}\mathsf{closure} \circ f^y)(q')$ for some $q' \in Q$. Either $q' = (q_1, q_2')$ for some $q_2' \in Q_2$, or $q' = (q_1', 0)$ for some $q_1' \in Q_1$.

If the former, it follows from the definition of $\mathcal{A}$ that $q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q_2')$. By the inductive hypothesis, there are t_1, t_2' with $t = t_1 \mid t_2' \mid y$, $q_1 \in [\![t_1]\!]_{\mathcal{A}_1}$ and $q_2' \in [\![t_2']\!]_{\mathcal{A}_2}$. Hence $q_2 \in [\![t_2' \mid y]\!]_{\mathcal{A}_2}$ and so the choice of t_1 and $t_2 = t_2' \mid y$ fulfills the requirements.

If the latter, it must be that:

- $(q_1, q_2) \in \varepsilon\text{-}\mathsf{closure}((q_1, e_2))$, and hence $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(e_2)$ and $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$;
- $(q_1, e_2) \in f^\varepsilon((q_1, 0))$; and
- $(q_1, 0) \in [\![t]\!]_{\mathcal{A}}$ and so $q_1 \in [\![t]\!]_{\mathcal{A}_1}$ (by Lem. 24).

Therefore the choice of $t_1 = t$ and $t_2 = \varepsilon$ fulfills the requirements.

Inductive case: $t = t'|a[t'']$ for some $a \in \Upsilon$. In this case, $(q_1, q_2) \in \varepsilon\text{-}\mathsf{closure}(q''')$ for some $q''' \in Q$ with $q''' \in f^a(q', q'')$ for some $q', q'' \in Q$ with $q' \in [\![t']\!]_{\mathcal{A}}$ and $q'' \in [\![t'']\!]_{\mathcal{A}}$. Either $q''' = (q_1, q_2')$ for some $q_2' \in Q_2$, or $q''' = (q_1', 0)$ for some $q_1' \in Q_1$.

If the former, it follows from the definition of $\mathcal{A}$ that $q' = (q_1, q_2')$ and $q'' = q_2''$ for some $q_2', q_2'' \in Q_2$ with $q_2' \in f_2^a(q_2', q_2'')$. By the inductive hypothesis, there are t_1, t_2' with $t' = t_1|t_2'$, $q_1 \in [\![t_1]\!]_{\mathcal{A}_1}$ and $q_2' \in [\![t_2']\!]_{\mathcal{A}_2}$. Furthermore, by Lem. 23, $q'' = q_2'' \in [\![t'']\!]_{\mathcal{A}_2}$ and so $q_2' \in [\![t_2' \mid a[t'']]\!]_{\mathcal{A}_2}$. It must also be that $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2')$, and so $q_2 \in [\![t_2' \mid a[t'']]\!]_{\mathcal{A}_2}$. Therefore choosing t_1 and $t_2' = t_2' \mid a[t'']$ fulfills the requirements.

If the latter, it must be that:

- $(q_1, q_2) \in \varepsilon\text{-}\mathsf{closure}((q_1, e_2))$, and hence $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(e_2)$ and $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$;
- $(q_1, e_2) \in f^\varepsilon((q_1, 0))$; and
- $(q_1, 0) \in [\![t]\!]_{\mathcal{A}}$ and so $q_1 \in [\![t]\!]_{\mathcal{A}_1}$ (by Lem. 24).

Therefore the choice of $t_1 = t$ and $t_2 = \varepsilon$ fulfills the requirements.

$\Longleftarrow$:

Base case: $t = \varepsilon$. In this case, $t_1 = \varepsilon$ and $t_2 = \varepsilon$. By Lem. 24, $(q_1, 0) \in [\![\varepsilon]\!]_{\mathcal{A}}$, and so, since $(q_1, e_2) \in f^\varepsilon((q_1, 0))$ by definition, $(q_1, e_2) \in [\![\varepsilon]\!]_{\mathcal{A}}$. Furthermore, it must be that $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(e_2)$ and hence $(q_1, q_2) \in [\![\varepsilon]\!]_{\mathcal{A}}$ as required.

Inductive case: $t = t' \mid y$ for some $y \in \Omega$. Here, either $t_2 = t'_2 \mid y$, or $t_2 = \varepsilon$ and $t_1 = t = t' \mid y$.

If the former, $q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q'_2)$ for some $q'_2 \in Q_2$ with $q'_2 \in [\![t'_2]\!]_{\mathcal{A}_2}$. By the inductive hypothesis, $(q_1, q'_2) \in [\![t_2 \mid t'_2]\!]_{\mathcal{A}}$. Therefore, by the definition of $\mathcal{A}$, $(q_1, q'_2) \in [\![t_1 \mid t'_2 \mid y]\!]_{\mathcal{A}} = [\![t]\!]_{\mathcal{A}}$ as required.

If the latter, $q_1 \in [\![t]\!]_{\mathcal{A}_1}$ and hence $(q_1, 0) \in [\![t]\!]_{\mathcal{A}}$ (by Lem. 24). It follows then that $(q_1, e_2) \in [\![t]\!]_{\mathcal{A}}$. Furthermore, since $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2} = \varepsilon\text{-}\mathsf{closure}_2(e_2)$, we can conclude that $(q_1, q_2) \in [\![t]\!]_{\mathcal{A}}$ by the definition of $\mathcal{A}$.

Inductive case: $t = t' \mid a[t'']$ for some $a \in \Upsilon$. Here, either $t_2 = t'_2 \mid a[t'']$, or $t_2 = \varepsilon$ and $t_1 = t = t' \mid a[t'']$.

If the former, $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q'''_2)$ for some $q'''_2 \in Q_2$ with $q'''_2 \in f_2^a(q'_2, q''_2)$ for some $q'_2, q''_2 \in Q_2$ with $q'_2 \in [\![t'_2]\!]_{\mathcal{A}_2}$ and $q''_2 \in [\![t''_2]\!]_{\mathcal{A}_2}$. By the inductive hypothesis, $(q_1, q'_2) \in [\![t_1 \mid t'_2]\!]_{\mathcal{A}} = [\![t']\!]_{\mathcal{A}}$. By Lem. 23, $q''_2 \in [\![t''_2]\!]_{\mathcal{A}}$. Hence, $(q_1, q'''_2) \in f^a((q_1, q'_2), q''_2) \subseteq [\![t' \mid a[t'']]\!]_{\mathcal{A}} = [\![t]\!]_{\mathcal{A}}$. Therefore, since by definition $(q_1, q_2) \in \varepsilon\text{-}\mathsf{closure}(q_1, q'''_2)$, we conclued that $(q_1, q_2) \in [\![t]\!]_{\mathcal{A}}$ as required.

If the latter, $q_1 \in [\![t]\!]_{\mathcal{A}_1}$ and hence $(q_1, 0) \in [\![t]\!]_{\mathcal{A}}$ (by Lem. 24). It follows then that $(q_1, e_2) \in [\![t]\!]_{\mathcal{A}}$. Furthermore, since $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2} = \varepsilon\text{-}\mathsf{closure}_2(e_2)$, we can conclude that $(q_1, q_2) \in [\![t]\!]_{\mathcal{A}}$ by the definition of $\mathcal{A}$. □

Lemma 26. *For all $t \in T_\Sigma$, $q_1 \in Q_1$,*

$$(q_1, 1) \in [\![t]\!]_{\mathcal{A}} \iff \exists t_1, t_2.\, t \in t_1 \oslash_x t_2 \wedge q_1 \in [\![t_1]\!]_{\mathcal{A}_1} \wedge t_2 \in L_2.$$

Proof. Both directions: by induction on the structure of t.

$\Longrightarrow$:

Base case: $t = \varepsilon$. It must be the case that there are some $q'_1, q''_1 \in Q_1$, $q_2 \in Q_1$ with:

- $(q''_1, q_2) \in [\![\varepsilon]\!]_{\mathcal{A}}$, and hence $q''_1 \in [\![\varepsilon]\!]_{\mathcal{A}_1}$ and $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$ (by Lem. 25);
- $(q'_1, 1) \in f^\varepsilon((q''_1, q_2))$, and hence $q'_1 \in f_1^x(q''_1)$ and $q_2 \in A_2$, so $q'_1 \in [\![x]\!]_{\mathcal{A}_2}$ and $\varepsilon \in L_2$; and
- $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}((q'_1, 1))$, and hence $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q'_1)$, so $q_1 \in [\![x]\!]_{\mathcal{A}_2}$.

Thus, $t_1 = x$ and $t_2 = \varepsilon$ fit the requirements: $\varepsilon \in x \oslash_x \varepsilon$.

Inductive case: $t = t' \mid y$ for some $y \in \Omega$. There must be some $q'_1 \in Q_1$ with:

- $(q_1, 1) \in (\varepsilon\text{-}\mathsf{closure} \circ f^y)((q'_1, 1))$ and $(q'_1, 1) \in [\![t]\!]_{\mathcal{A}}$; or
- $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}((q'_1, 1))$ and $(q'_1, 1) \in f^\varepsilon((q''_1, q_2))$ for some $q''_1 \in Q_1$, $q_2 \in Q_2$ with $(q''_1, q_2) \in [\![t]\!]_{\mathcal{A}}$.

If the former, then, by the inductive hypothesis, there are t'_1 and t_2 with $t' \in t'_1 \oslash_x t_2$, $q'_1 \in [\![t'_1]\!]_{\mathcal{A}_1}$ and $t_2 \in L_2$. By the definition of $\mathcal{A}$, $q_1 \in (\varepsilon\text{-}\mathsf{closure}_1 \circ f_1^y)(q'_1)$ and so $q_1 \in [\![t'_1 \mid y]\!]_{\mathcal{A}_1}$. Observing that $(t'_1 \oslash_x t_2) \mid y \subseteq (t'_1 \mid y) \oslash_x t_2$, we can see that the choice of $t_1 = t'_1 \mid y$ and t_2 fulfills the requirements.

If the latter, then, by Lem. 25, $t = t'_1 \mid t_2$ with $q''_1 \in [\![t'_1]\!]_{\mathcal{A}_1}$ and $q_2 \in [\![t_2]\!]_{\mathcal{A}_2}$. Furthermore, by the definition of $\mathcal{A}$, $q_2 \in A_2$ and so $t_2 \in L_2$. Also, $q'_1 \in f_1^x(q''_1)$, so $q'_1 \in [\![t'_1 \mid x]\!]_{\mathcal{A}_1}$. Moreover, $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q'_1)$, so $q_1 \in [\![t'_1 \mid x]\!]_{\mathcal{A}_1}$. Observing that $t'_1 \mid t_2 \in (t'_1 \mid x) \oslash_x t_2$, we can see that the choice of $t_1 = t'_1 \mid x$ and t_2 fulfills the requirements.

Inductive case: $t = t' \mid a[t'']$ for some $ain\Upsilon$. There must be some $q'''_1 \in Q_1$ with $(q_1, 1) \in \varepsilon\text{-closure}(q'''_1, 1)$ such that one of the following holds:

- $(q'''_1, 1) \in f^a((q'_1, 1), (q''_1, 0))$ for some $q'_1, q''_1 \in Q_1$ with $(q'_1, 1) \in [\![t']\!]_{\mathcal{A}}$ and $(q''_1, 0) \in [\![t'']\!]_{\mathcal{A}}$;
- $(q'''_1, 1) \in f^a((q'_1, 0), (q''_1, 1))$ for some $q'_1, q''_1 \in Q_1$ with $(q'_1, 0) \in [\![t']\!]_{\mathcal{A}}$ and $(q''_1, 1) \in [\![t'']\!]_{\mathcal{A}}$; or
- $(q'''_1, 1) \in f^\varepsilon((q'_1, q_2))$ for some $q'_1 \in Q_1$, $q_2 \in Q_2$ with $(q'_1, q_2) \in [\![t]\!]_{\mathcal{A}}$.

In the first case, by the inductive hypothesis, there are t'_1, t_2 with $t' \in t'_1 \oslash_x t_2$, $q'_1 \in [\![t'_1]\!]_{\mathcal{A}_1}$ and $t_2 \in L_2$. Furthermore, by Lem. 24, $q''_1 \in [\![t'']\!]_{\mathcal{A}_1}$. By the definition of $\mathcal{A}$, $q'''_1 \in f^a_1(q'_1, q''_1)$, and so $q'''_1 \in [\![t'_1 \mid a[t'']]\!]_{\mathcal{A}_1}$. Moreover, $q_1 \in \varepsilon\text{-closure}_1(q'''_1)$, so $q_1 \in [\![t'_1 \mid a[t'']]\!]_{\mathcal{A}_1}$. Observing that $(t'_1 \oslash_x t_2) \mid a[t''] \subseteq (t'_1 \mid a[t'']) \oslash_x t_2$, we can see that the choice of $t_1 = t'_1 \mid a[t'']$ and t_2 fulfills the requirements.

In the second case, by Lem. 24, $q'_1 \in [\![t']\!]_{\mathcal{A}_1}$. Furthermore, by the inductive hypothesis, there are t''_1, t_2 with $t'' \in t''_1 \oslash_x t_2$, $q''_1 \in [\![t''_1]\!]_{\mathcal{A}_1}$ and $t_2 \in L_2$. By the definition of $\mathcal{A}$, $q'''_1 \in f^a_1(q'_1, q''_1)$, and so $q'''_1 \in [\![t' \mid a[t''_1]]\!]_{\mathcal{A}_1}$. Moreover, $q_1 \in \varepsilon\text{-closure}_1(q'''_1)$, so $q_1 \in [\![t_1 | a[t''_1]]\!]_{\mathcal{A}_1}$. Observing that $t' | a[t''_1 \oslash_x t_2] \subseteq (t' | a[t''_1]) \oslash_x t_2$, we can see that the choice of $t_1 = t' \mid a[t''_1]$ and t_2 fulfills the requirements.

In the third case, by Lem. 25, $t = t'_1 \mid t_2$ with $q'_1 \in [\![t'_1]\!]_{\mathcal{A}_1}$ and $q_2 \in [\![t_2]\!]_{\mathcal{A}_2}$. Furthermore, by the definition of $\mathcal{A}$, $q_2 \in A_2$ and so $t_2 \in L_2$. Also, $q'''_1 \in f^x_1(q'_1)$, so $q'''_1 \in [\![t'_1 \mid x]\!]_{\mathcal{A}_1}$. Moreover, $q_1 \in \varepsilon\text{-closure}_1(q'''_1)$, so $q_1 \in [\![t'_1 \mid x]\!]_{\mathcal{A}_1}$. Observing that $t'_1 \mid t_2 \in (t'_1 \mid x) \oslash_x t_2$, we can see that the choice of $t_1 = t'_1 \mid x$ and t_2 fulfills the requirements.

$\Longleftarrow$:

Base case: $t = \varepsilon$. In this case, it must be that $t_1 = x$ and $t_2 = \varepsilon$. Since $q_1 \in [\![x]\!]_{\mathcal{A}_1}$ it follows that $q_1 \in \varepsilon\text{-closure}_1(q'_1)$ for some $q'_1 \in f^x_1(q''_1)$ for some $q''_1 \in [\![\varepsilon]\!]_{\mathcal{A}_1}$. Further, since $\varepsilon = t_2 \in L_2$, there must be some $q_2 \in A_2$ with $q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}$. By Lem. 25, $(q''_1, q_2) \in [\![\varepsilon]\!]_{\mathcal{A}}$. By the construction of $\mathcal{A}$, it follows that $(q'_1, 1) \in f^\varepsilon(q''_1, q_2)$ and so $(q'_1, 1) \in [\![\varepsilon]\!]_{\mathcal{A}}$. Also, $(q_1, 1) \in \varepsilon\text{-closure}_(q'_1, 1)$ and so $(q_1, 1) \in [\![\varepsilon]\!]_{\mathcal{A}}$ as required.

Inductive case: $t = t' \mid y$ for some $y \in \Omega$. One of the following must hold:

- $t = t''_1 \mid t_2$ and $t_1 = t''_1 \mid x$; or
- $t' \in t'_1 \oslash_x t_2$ and $t_1 = t'_1 \mid y$.

If the former, there is a $q''_1 \in [\![t''_1]\!]_{\mathcal{A}_1}$ such that $q_1 \in (\varepsilon\text{-closure}_1 \circ f^x_1)(q''_1)$, and $q_2 \in [\![t]\!]_{\mathcal{A}_2} \cap A_2$. By Lem. 25, $(q''_1, q_2) \in [\![t''_1 \mid t_2]\!]_{\mathcal{A}} = [\![t]\!]_{\mathcal{A}}$. From the definition of f^ε, it follows that $(q_1, 1) \in \varepsilon\text{-closure}((q''_1, q_2))$ and so $(q_1, 1) \in [\![t]\!]_{\mathcal{A}}$ as required.

If the latter, there is a $q'_1 \in [\![t'_1]\!]_{\mathcal{A}_1}$ such that $q_1 \in (\varepsilon\text{-closure}_1 \circ f^y_1)(q'_1)$. By the inductive hypothesis, $(q'_1, 1) \in [\![t']\!]_{\mathcal{A}}$. By the definition of $\mathcal{A}$, $(q_1, 1) \in (\varepsilon\text{-closure} \circ f^y)(q'_1, 1)$, and so $(q_1, 1) \in [\![t]\!]_{\mathcal{A}}$ as required.

Inductive case: $t = t' \mid a[t'']$ for some $a \in \Upsilon$. One of the following must hold:

- $t = t''_1 \mid t_2$ and $t_1 = t''_1 \mid y$;
- $t' \in t'_1 \oslash_x t_2$ and $t_1 = t'_1 \mid a[t'']$; or

– $t'' \in t''_1 \oslash_x t_2$ and $t_1 = t' \mid a[t''_1]$.

In the first case, there is a $q''_1 \in [\![t''_1]\!]_{\mathcal{A}_1}$ such that $q_1 \in (\varepsilon\text{-}\mathsf{closure}_1 \circ f^x_1)(q''_1)$, and $q_2 \in [\![t]\!]_{\mathcal{A}_2} \cap A_2$. By Lem. 25, $(q''_1, q_2) \in [\![t''_1 \mid t_2]\!]_{\mathcal{A}} = [\![t]\!]_{\mathcal{A}}$. From the definition of f^ε, it follows that $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}((q''_1, q_2))$ and so $(q_1, 1) \in [\![t]\!]_{\mathcal{A}}$ as required.

In the second case, $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q'_1)$ for some $q'_1 \in f^a_1(q''_1, q'''_1)$ for some $q''_1 \in [\![t'_1]\!]_{\mathcal{A}_1}$ and $q'''_1 \in [\![t'']\!]_{\mathcal{A}_1}$. By the inductive hypothesis, $(q''_1, 1) \in [\![t']\!]_{\mathcal{A}}$. By Lem. 24, $(q'''_1, 0) \in [\![t'']\!]_{\mathcal{A}}$. By the definition of $\mathcal{A}$, $(q'_1, 1) \in f^a((q''_1, 1), (q'''_1, 0))$, and so $(q'_1, 1) \in [\![t' \mid a[t'']]\!]_{\mathcal{A}} = [\![t]\!]_{\mathcal{A}}$. Furthermore, $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}(q'_1, 1)$, and so $(q_1, 1) \in [\![t]\!]_{\mathcal{A}}$ as required.

In the third case, $q_1 \in \varepsilon\text{-}\mathsf{closure}_1(q'_1)$ for some $q'_1 \in f^a_1(q''_1, q'''_1)$ for some $q''_1 \in [\![t']\!]_{\mathcal{A}_1}$ and $q'''_1 \in [\![t''_1]\!]_{\mathcal{A}_1}$. By Lem. 24, $(q''_1, 0) \in [\![t']\!]_{\mathcal{A}}$. By the inductive hypothesis, $(q'''_1, 1) \in [\![t'']\!]_{\mathcal{A}}$. By the definition of $\mathcal{A}$, $(q'_1, 1) \in f^a((q''_1, 0), (q'''_1, 1))$, and so $(q'_1, 1) \in [\![t' \mid a[t'']]\!]_{\mathcal{A}} = [\![t]\!]_{\mathcal{A}}$. Furthermose, $(q_1, 1) \in \varepsilon\text{-}\mathsf{closure}(q'_1, 1)$, and so $(q_1, 1) \in [\![t]\!]_{\mathcal{A}}$ as required. □

Proposition 4. *The automaton defined in Def. 17 accepts the language $L_1 \oslash_x L_2$.*

Proof.

$$
\begin{array}{ll}
 & t \in L_1 \oslash_x L_2 \\
\iff & \exists t_1, t_2.\, t \in t_1 \oslash_x t_2 \ \wedge\ t_1 \in L_1 \ \wedge\ t_2 \in L_2 \\
\iff & \exists q_1 \in A_1.\, (q_1, 1) \in [\![t]\!]_{\mathcal{A}} \quad \text{(by Lem. 26)} \\
\iff & A \cap [\![t]\!]_{\mathcal{A}} \neq \emptyset.
\end{array}
$$

□

Correctness of '$-\!\oslash^{\exists}$' Construction Given ε-NFFA $\mathcal{A}_1 = (Q_1, e_1, \{f^a_1\}_{a\in\Upsilon}, \{f^x_1\}_{x\in\Omega}, f^\varepsilon_1, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f^a_2\}_{a\in\Upsilon}, \{f^x_2\}_{x\in\Omega}, f^\varepsilon_2, A_2)$ accepting languages L_1 and L_2 respectively, let $\mathcal{A} = \mathcal{A}_1 -\!\oslash^{\exists}_x \mathcal{A}_2 = (Q, e, \{f^a\}_{a\in\Upsilon}, \{f^x\}_{x\in\Omega}, f^\varepsilon, A)$ as per Def. 18.

Lemma 27. *For all $t \in T_\Sigma$, $q_1 \in Q_1$,*

$$(q_2, 0) \in [\![t]\!]_{\mathcal{A}} \iff q_2 \in [\![t]\!]_{\mathcal{A}_2}.$$

Proof. By induction on the structure of the tree t.

Base case: $t = \varepsilon$:

$$
\begin{aligned}
(q_2, 0) \in [\![\varepsilon]\!]_{\mathcal{A}} &\iff (q_2, 0) \in \varepsilon\text{-}\mathsf{closure}((e_2, 0)) \\
&\iff q_2 \in \varepsilon\text{-}\mathsf{closure}_2(e_2) \\
&\iff q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}.
\end{aligned}
$$

Inductive case: $t = t' \mid y$ for some $y \in \Omega$:

$$\begin{aligned}(q_2, 0) \in [\![t' \mid y]\!]_{\mathcal{A}} &\iff \exists q_2' \in Q_2.\, (q_2, 0) \in (\varepsilon\text{-}\mathsf{closure} \circ f^y)((q_2', 0)) \wedge (q_2', 0) \in [\![t']\!]_{\mathcal{A}} \\ &\iff \exists q_2' \in Q_2.\, q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q_2') \wedge q_2' \in [\![t']\!]_{\mathcal{A}_2} \\ &\iff q_2 \in [\![t' \mid y]\!]_{\mathcal{A}_2}.\end{aligned}$$

Inductive case: $t = t' \mid a[t'']$ for some $a \in \Upsilon$:

$$\begin{aligned}(q_2, 0) \in [\![t' \mid a[t'']]\!]_{\mathcal{A}} &\iff \exists q_2', q_2'', q_2''' \in Q_2.\, (q_2, 0) \in \varepsilon\text{-}\mathsf{closure}((q_2''', 0)) \wedge \\ &\quad (q_2''', 0) \in f^a((q_2', 0), (q_2'', 0)) \wedge (q_2', 0) \in [\![t']\!]_{\mathcal{A}} \wedge (q_2'', 0) \in [\![t'']\!]_{\mathcal{A}} \\ &\iff \exists q_2', q_2'', q_2''' \in Q_2.\, q_1 \in \varepsilon\text{-}\mathsf{closure}_2(q_2''') \wedge \\ &\quad q_2''' \in f_2^a(q_2', q_2'') \wedge q_2' \in [\![t']\!]_{\mathcal{A}_2} \wedge [\![t'']\!]_{\mathcal{A}_2} \\ &\iff q_2 \in [\![t' \mid a[t'']]\!]_{\mathcal{A}_2}.\end{aligned}$$

□

Lemma 28. *For all $t \in T_\Sigma$, $q_2 \in Q_2$,*

$$(q_2, 1) \in [\![t]\!]_{\mathcal{A}} \iff \exists t_1, t_2.\, t_2 \in t \oslash_x t_1 \wedge t_1 \in L_1 \wedge q_2 \in [\![t_2]\!]_{\mathcal{A}_2}.$$

Proof. In both directions, the proof is by induction on the structure of the tree t.

$\Longrightarrow$:

Base case: $t = \varepsilon$. There are no $q_2', q_2'' \in Q_2$ such that $(q_2', 1) \in f^\varepsilon((q_2'', 0))$ and so it is not possible that $(q_2, 1) \in [\![t]\!]_{\mathcal{A}}$. Hence, the implication holds vacuuosly in this case.

Inductive case: $t = t' \mid y$ for some $y \in \Omega$. Assume $(q_2, 1)[\![t]\!]_{\mathcal{A}}$. One of the following must apply:

- $(q_2, 1) \in (\varepsilon\text{-}\mathsf{closure} \circ f^y)((q_2', 1))$ for some $q_2' \in Q_2$ with $(q_2', 1) \in [\![t']\!]_{\mathcal{A}}$; or
- $y = x$ and $(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}((q_2', 1))$ for some $q_2' \in Q_2$ with $(q_2', 1) \in f^x((q_2'', 0))$ for some $q_2'' \in Q_2$ with $(q_2'', 0) \in [\![t]\!]_{\mathcal{A}}$.

In the first case, by the inductive hypothesis, there are t_1, t_2' with $t_2' \in t' \oslash_x t_1$, $t_1 \in L_1$ and $q_2' \in [\![t_2']\!]_{\mathcal{A}_2}$. By the definition of $\mathcal{A}$, $q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q_2')$ and so $q_2 \in [\![t_2]\!]_{\mathcal{A}_2}$. Observe that $t_2' \mid y \in (t' \mid y) \oslash_x t_1$ and so the choice of t_1 and $t_2 = t_2' \mid y$ fulfills the requirements.

In the second case, Lem. 27, we know that $q_2'' \in [\![t']\!]_{\mathcal{A}_2}$. By the definition of f^x, $q_2' \in (\!|t_1|\!)_{\mathcal{A}_2}(q_2'')$ for some $t_1 \in L_1$. Hence, $t_2' \in [\![t' \mid t_1]\!]_{\mathcal{A}_2}$. Furthermore, $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2')$ by the definition of $\mathcal{A}$ and so $q_2 \in [\![t' | t_1]\!]_{\mathcal{A}_2}$. Let $t_2 = t' | t_1$ and observe that $t_2 \in (t' \,|\, x) \oslash_x t_1 = t \oslash_x t_1$. Hence t_1 and t_2 fulfill the requirements.

Inductive case: $t = t' \mid a[t'']$ for some $a \in \Upsilon$. Assume $(q_2, 1) \in [\![t]\!]_{\mathcal{A}}$. It follows that $(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}((q_2''', 1))$ for some $q_2''' \in Q_2$ with either:

- $(q_2''', 1) \in f^a((q_2', 1), (q_2'', 0))$ for some $q_2', q_2'' \in Q_2$ with $(q_2', 1) \in [\![t']\!]_{\mathcal{A}}$ and $(q_2'', 0) \in [\![t'']\!]_{\mathcal{A}}$; or

– $(q_2''', 1) \in f^a((q_2', 0), (q_2'', 1))$ for some $q_2', q_2'' \in Q_2$ with $(q_2', 0) \in [\![t']\!]_{\mathcal{A}}$ and $(q_2'', 1) \in [\![t'']\!]_{\mathcal{A}}$.

In the first case, by the inductive hypothesis, there are t_1, t_2' with $t_2' \in t' \oslash_x t_1$, $t_1 \in L_1$ and $q_2' \in [\![t_2']\!]_{\mathcal{A}_2}$. By Lem. 27, we know that $q_2'' \in [\![t'']\!]_{\mathcal{A}_2}$. Hence, by the definition of f^a, it follows that $q_2''' \in f_2^a(q_2', q_2'')$ and so $q_2''' \in [\![t_2' \mid a[t'']]\!]_{\mathcal{A}_2}$. Furthermore, $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2')$ and so $q_2 \in [\![t_2' \mid a[t'']]\!]_{\mathcal{A}_2} = [\![t_2]\!]_{\mathcal{A}_2}$. Let $t_2 = t_2' \,|\, a[t'']$ and observe that, since $t_2' \in t' \oslash_x t_1$, $t_2' \,|\, a[t''] \in (t' \,|\, a[t'']) \oslash_x t_1 = t \oslash_x t_1$. Thus, t_1 and t_2 fulfill the requirements.

In the second case, by Lem. 27, we know that $q_1' \in [\![t']\!]_{\mathcal{A}_2}$. By the inductive hypothesis, there are t_1, t_2'' with $t_2'' \in t'' \oslash_x t_1$, $t_1 \in L_1$ and $q_2'' \in [\![t_2'']\!]_{\mathcal{A}_2}$. Hence, by the definition of f^a, it follows that $q_2''' \in f_2^a(q_2', q_2'')$ and so $q_2''' \in [\![t' \,|\, a[\mathcal{A}_2'']]\!]_{\mathcal{A}_2}$. Furthermore, $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2')$ and so $q_2 \in [\![t' \mid a[t_2'']]\!]_{\mathcal{A}_2} = [\![t_2]\!]_{\mathcal{A}_2}$. Let $t_2 = t' \,|\, a[t_2'']$ and observe that, since $t_2'' \in t'' \oslash_y t_1$, $t' \,|\, a[t_2''] \in (t' \,|\, a[t'']) \oslash_x t_1 = t \oslash_x t_1$. Thus, t_1 and t_2 fulfill the requirements.

$\Longleftarrow$:

Base case: $t = \varepsilon$. There is no such t_1 and t_2 (since x does not appear in t).

Inductive case: $t = t' \mid y$ for some $y \in \Omega$. Assume there are t_1, t_2 such that $t_2 \in t \oslash_x t_1$, $t_1 \in L_1$ and $q_2 \in [\![t_2]\!]_{\mathcal{A}_2}$ Either:

– $t_2 = t_2' \,|\, y$ for some $t_2' \in t' \oslash_x t_1$; or
– $y = x$ and $t_2 = t' \mid t_1$.

In the first case, $q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q_2')$ for some $q_2' \in Q_2$ with $q_2' \in [\![t_2']\!]_{\mathcal{A}_2}$. By the inductive hypothesis, $(q_2', 1) \in [\![t']\!]_{\mathcal{A}}$. By the definition of $\mathcal{A}$, $(q_2, 1) \in (\varepsilon\text{-}\mathsf{closure} \circ f^y)((q_2', 1))$ and so $(q_2, 1) \in [\![t']\!]_{\mathcal{A}}$ as required.

In the second case, $q_2 \in (\!|t_1|\!)_{\mathcal{A}_2}(q_2')$ for some $q_2' \in Q_2$ with $q_2' \in [\![t']\!]_{\mathcal{A}_2}$. By Lem. 27, $(q_2', 0) \in [\![t']\!]_{\mathcal{A}}$. By definition, $(q_2, 1) \in f^x((q_2', 0))$ and so $(q_2, 1) \in [\![t' \mid x]\!]_{\mathcal{A}} = [\![t]\!]_{\mathcal{A}}$, as required.

Inductive case: $t = t' \,|\, a[t'']$ for some $a \in \Upsilon$. Assume there are t_1, t_2 such that $t_2 \in t \oslash_x t_1$, $t_1 \in L_1$ and $q_2 \in [\![t_2]\!]_{\mathcal{A}_2}$ Either:

– $t_2 = t_2' \,|\, a[t'']$ for some $t_2' \in t' \oslash_x t_1$; or
– $t_2 = t' \,|\, a[t_2'']$ for some $t_2'' \in t'' \oslash_x t_1$.

In the first case, $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2''')$ for some $q_2''' \in Q_2$ with $q_2''' \in f_(^a q_2', q_2'')$ for some $q_2', q_2'' \in Q_2$ with $q_2' \in [\![t_2']\!]_{\mathcal{A}_2}$, $q_2'' \in [\![t'']\!]_{\mathcal{A}_2}$. By the inductive hypothesis, $(q_2', 1) \in [\![t']\!]_{\mathcal{A}}$. By Lem. 27, $(q_2'', 0) \in [\![t'']\!]_{\mathcal{A}}$. By the definition of $\mathcal{A}$, $(q_2''', 1) \in f^a((q_2', 1), (q_2'', 0))$ and so $(q_2''', 1) \in [\![t' \mid a[t'']]\!]_{\mathcal{A}} = [\![t]\!]_{\mathcal{A}}$. Furthermore, $(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}((q_2''', 1))$ and so $(q_2, 1) \in [\![t]\!]_{\mathcal{A}}$ as required.

In the second case, $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2''')$ for some $q_2''' \in Q_2$ with $q_2''' \in f_(^a q_2', q_2'')$ for some $q_2', q_2'' \in Q_2$ with $q_2' \in [\![t']\!]_{\mathcal{A}_2}$, $q_2'' \in [\![t_2'']\!]_{\mathcal{A}_2}$. By Lem. 27, $(q_2', 0) \in [\![t']\!]_{\mathcal{A}}$. By the inductive hypothesis, $(q_2'', 1) \in [\![t'']\!]_{\mathcal{A}}$. By the definition of $\mathcal{A}$, $(q_2''', 1) \in f^a((q_2', 0), (q_2'', 1))$ and so $(q_2''', 1) \in [\![t' \mid a[t'']]\!]_{\mathcal{A}} = [\![t]\!]_{\mathcal{A}}$. Furthermore, $(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}((q_2''', 1))$ and so $(q_2, 1) \in [\![t]\!]_{\mathcal{A}}$ as required. $\square$

Proposition 5. *The automaton defined in Def. 18 accepts the language $L_1 \multimap\!\!\oslash^{\exists}_x L_2$.*

Proof.

$$\begin{aligned}
& t \in L_1 \multimap\!\!\oslash^{\exists}_x L_2 \\
\iff\quad & \exists t_1, t_2.\, t_1 \in L_1 \wedge t_2 \in L_2 \wedge t_2 \in t \oslash_x t_1 \\
\iff\quad & \exists q_2 \in A_2.\, \exists t_1, t_2.\, t_1 \in L_1 \wedge q_2 \in [\![t_2]\!]_{\mathcal{A}_2} \wedge t_2 \in t \oslash_x t_1 \\
\iff\quad & \exists q_2 \in A_2.\, (q_2, 1) \in [\![t]\!]_{\mathcal{A}} \\
\iff\quad & [\![t]\!]_{\mathcal{A}} \cap A \neq \emptyset.
\end{aligned}$$

□

Correctness of '$\oslash\!\!-^{\exists}$' Construction Given ε-NFFA $\mathcal{A}_1 = (Q_1, e_1, \{f_1^a\}_{a\in\Upsilon}, \{f_1^x\}_{x\in\Omega}, f_1^\varepsilon, A_1)$ and $\mathcal{A}_2 = (Q_2, e_2, \{f_2^a\}_{a\in\Upsilon}, \{f_2^x\}_{x\in\Omega}, f_2^\varepsilon, A_2)$ accepting languages L_1 and L_2 respectively, let $\mathcal{A} = \mathcal{A}_1 \oslash\!\!-^{\exists}_x \mathcal{A}_2 = (Q, e, \{f^a\}_{a\in\Upsilon}, \{f^x\}_{x\in\Omega}, f^\varepsilon, A)$ as per Def. 19.

Lemma 29. *For all $t \in T_\Sigma$,*

$$[\![t]\!]_{\mathcal{A}} = \{(\!|t|\!)_{\mathcal{A}_2}\}.$$

Proof. By induction on the structure of t. Note that, since $f^\varepsilon = \emptyset$, ε-closure is the identity relation on Q.

Base case: $t = \varepsilon$.

$$\begin{aligned}
[\![\varepsilon]\!]_{\mathcal{A}} &= \varepsilon\text{-closure}(e) \\
&= \varepsilon\text{-closure}(\varepsilon\text{-closure}_2) \\
&= \{\varepsilon\text{-closure}_2\} \\
&= \{(\!|\varepsilon|\!)_{\mathcal{A}_2}\}.
\end{aligned}$$

Inductive case: $t = t' \mid y$ for some $y \in \Omega$.

$$\begin{aligned}
[\![t' \mid y]\!]_{\mathcal{A}} &= \{q \mid q' \in [\![t']\!]_{\mathcal{A}}, q \in (\varepsilon\text{-closure} \circ f^y)(t')\} \\
&= \{q \mid q \in (\varepsilon\text{-closure} \circ f^y)((\!|t'|\!)_{\mathcal{A}_2})\} \qquad\qquad = f^y((\!|t'|\!)_{\mathcal{A}_2}) \\
&= \{\varepsilon\text{-closure}_2 \circ f_2^y \circ (\!|t'|\!)_{\mathcal{A}_2}\} \\
&= \{(\!|t' \mid y|\!)_{\mathcal{A}_2}\}.
\end{aligned}$$

Inductive case: $t = t' \mid a[t'']$ for some $a \in \Upsilon$.

$$\begin{aligned}
[\![t' \mid a[t'']]\!]_{\mathcal{A}} &= \{q \mid q' \in [\![t']\!]_{\mathcal{A}}, q'' \in [\![t'']\!]_{\mathcal{A}}, q \in (\varepsilon\text{-closure} \circ f^a)(q', q'')\} \\
&= \{q \mid q \in (\varepsilon\text{-closure} \circ f^a)((\!|t'|\!)_{\mathcal{A}_2}, (\!|t''|\!)_{\mathcal{A}_2})\} \\
&= f^a((\!|t'|\!)_{\mathcal{A}_2}, (\!|t''|\!)_{\mathcal{A}_2}) \\
&= \{\varepsilon\text{-closure}_2 \circ \{(q_2, q_2') \mid q_2 \in Q_2, q_2'' \in (\!|t'|\!)_{\mathcal{A}_2}(q_2), q_2''' \in (\!|t''|\!)_{\mathcal{A}_2}(e_2), q_2' \in f_2^a(q_2'', q_2''')\}\} \\
&= \{\{(q_2, q_2') \mid q_2 \in Q_2, q_2'' \in (\!|t'|\!)_{\mathcal{A}_2}(q_2), q_2''' \in [\![t'']\!]_{\mathcal{A}_2}, q_2' \in (\varepsilon\text{-closure}_2 \circ f_2^a)(q_2'', q_2''')\}\} \\
&= \{(\!|t' \mid a[t'']|\!)_{\mathcal{A}_2}\}.
\end{aligned}$$

□

For $q \in Q$, let $\mathcal{A}_q$ be as given in Def. 19.

Lemma 30. *For any $q \in Q$, $q_2 \in Q_2$, and $t \in T_\Sigma$,*

$$(q_2, 0) \in [\![t]\!]_{\mathcal{A}_q} \iff q_2 \in [\![t]\!]_{\mathcal{A}_2}.$$

Proof. By induction on the structure of the tree t.

Base case: $t = \varepsilon$:

$$\begin{aligned}(q_2, 0) \in [\![\varepsilon]\!]_{\mathcal{A}_q} &\iff (q_2, 0) \in \varepsilon\text{-}\mathsf{closure}((e_2, 0))\\ &\iff q_2 \in \varepsilon\text{-}\mathsf{closure}_2(e_2)\\ &\iff q_2 \in [\![\varepsilon]\!]_{\mathcal{A}_2}.\end{aligned}$$

Inductive case: $t = t' \mid y$ for some $y \in \Omega$:

$$\begin{aligned}(q_2, 0) \in [\![t' \mid y]\!]_{\mathcal{A}_q} &\iff \exists q_2' \in Q_2.\, (q_2, 0) \in (\varepsilon\text{-}\mathsf{closure} \circ f^y)((q_2', 0)) \wedge (q_2', 0) \in [\![t']\!]_{\mathcal{A}_q}\\ &\iff \exists q_2' \in Q_2.\, q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q_2') \wedge q_2' \in [\![t']\!]_{\mathcal{A}_2}\\ &\iff q_2 \in [\![t' \mid y]\!]_{\mathcal{A}_2}.\end{aligned}$$

Inductive case: $t = t' \mid a[t'']$ for some $a \in \Upsilon$:

$$\begin{aligned}(q_2, 0) \in [\![t' \mid a[t'']]\!]_{\mathcal{A}_q} &\iff \exists q_2', q_2'', q_2''' \in Q_2.\, (q_2, 0) \in \varepsilon\text{-}\mathsf{closure}((q_2''', 0)) \wedge\\ &\qquad (q_2''', 0) \in f^a((q_2', 0), (q_2'', 0)) \wedge (q_2', 0) \in [\![t']\!]_{\mathcal{A}_q} \wedge (q_2'', 0) \in [\![t'']\!]_{\mathcal{A}_q}\\ &\iff \exists q_2', q_2'', q_2''' \in Q_2.\, q_1 \in \varepsilon\text{-}\mathsf{closure}_2(q_2''') \wedge\\ &\qquad q_2''' \in f_2^a(q_2', q_2'') \wedge q_2' \in [\![t']\!]_{\mathcal{A}_2} \wedge [\![t'']\!]_{\mathcal{A}_2}\\ &\iff q_2 \in [\![t' \mid a[t'']]\!]_{\mathcal{A}_2}.\end{aligned}$$

□

Lemma 31. *Suppose that $q = (\!|t|\!)_{\mathcal{A}_2}$ for some $t \in T_\Sigma$. Then for all $t_1 \in T_\Sigma$, all $q_2 \in Q_2$,*

$$(q_2, 1) \in [\![t_1]\!]_{\mathcal{A}_q} \iff \exists t_2 \in T_\Sigma.\, q_2 \in [\![t_2]\!]_{\mathcal{A}_2} \wedge t_2 \in t_1 \oslash_x t.$$

Proof. Both directions are by induction on the structure of the tree t_1.

$\Longrightarrow$:

Base case: $t_1 = \varepsilon$.
In this case, it is not possible that $(q_2, 1) \in [\![\varepsilon]\!]_{\mathcal{A}_q}$ and so the implication holds trivially.

Inductive case: $t_1 = t_1' \mid y$ for some $y \in \Omega$.
In this case, $(t_2, 1) \in (\varepsilon\text{-}\mathsf{closure}_q \circ f_q^x)(q_q)$ for some $q_q \in Q_q$ with either:

- $q_q = (q_2', 1)$ for some $q_2' \in Q_2$; or
- $q_q = (q_2', 0)$ for some $q_2' \in Q_1$, $y = y$, and $(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}_q((q_2'', 1))$ for some $q_2'' \in Q_2$ with $(q_2'', 1) \in f_q^x((q_2', 0))$.

In the first case, since $(q_2', 1) \in [\![t_1']\!]_{\mathcal{A}_q}$, we have that $q_2' \in [\![t_2']\!]_{\mathcal{A}_2}$ for some $t_2' \in t_1' \oslash_x t$, by the inductive hypothesis. From the definition of $\mathcal{A}_q$, we can deduce that $q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q_2')$ and hence $q_2 \in [\![t_2' \mid y]\!]_{\mathcal{A}_2}$. Observe that $t_2'|y \in (t_1'|y) \oslash_x t = t_1 \oslash_x t$, and so the choice of $t_2 = t_2'|y$ fulfills the requirements.

In the second case, since $(q_2', 0) \in [\![t_1']\!]_{\mathcal{A}_q}$, we have that $q_2' \in [\![t_1']\!]_{\mathcal{A}_2}$ by Lem. 30. Since $(q_2'', 1) \in f_q^x((q_2', 0))$, by the definition of $\mathcal{A}_q$ we have $q_2'' \in q(q_2') = (\!|t|\!)_{\mathcal{A}_2} q_2'$ and so $q_2'' \in [\![t_1' \mid t]\!]_{\mathcal{A}_2}$. Furthermore, $q - 2' \in \varepsilon\text{-}\mathsf{closure}_2(q_2'')$ and so $q_2 \in [\![t_1' \mid t]\!]_{\mathcal{A}_2}$. Observe that $t_1' \mid t \in (q_1' \mid x) \oslash_x t = t_1 \oslash_x t$ and so the choice $t_2 = t_2' \mid y$ fulfills the requirements.

Inductive case: $t_1 = t_1' \mid a[t_1'']$ for some $a \in \Upsilon$.
In this case, $(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}_q((q_2''', 1))$ for some $q_2''' \in Q_2$ with either:

- $(q_2''', 1) \in f_q^a((q_2', 1), (q_2'', 0))$ for some $q_2', q_2'' \in Q_2$ with $(q_1', 1) \in [\![t_1']\!]_{\mathcal{A}_q}$ and $(q_2'', 0) \in [\![t_1'']\!]_{\mathcal{A}_q}$; or
- $(q_2''', 1) \in f_q^a((q_2', 0), (q_2'', 1))$ for some $q_2', q_2'' \in Q_2$ with $(q_1', 0) \in [\![t_1']\!]_{\mathcal{A}_q}$ and $(q_2'', 1) \in [\![t_1'']\!]_{\mathcal{A}_q}$.

In the first case, by the inductive hypothesis, $q_2' \in [\![t_2']\!]_{\mathcal{A}_2}$ for some $t_2' \in t_1' \oslash_x t$. By Lem. 30, $q_2'' \in [\![t_1'']\!]_{\mathcal{A}_2}$. By the definition of $\mathcal{A}_q$, $q_2''' \in f_2^a(q_2', q_2'')$ and so $q_2''' \in [\![t_2' \mid a[t_1'']]\!]_{\mathcal{A}_2}$. Furthermore, $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2''')$ and so $q_2 \in [\![t_2' \mid a[t_1'']]\!]_{\mathcal{A}_2}$. Observe that $t_2' \mid a[t_1''] \in (t_1' \oslash_x t) \mid a[t_1''] \subseteq (t_1 \mid a[t_1'']) \oslash_x t$ and so the choice $t_2 = t_2' \mid a[t_1'']$ fulfills the requirements.

In the second case, by Lem. 30, $q_2' \in [\![t_1']\!]_{\mathcal{A}_2}$. By the inductive hypothesis, $q_2'' \in [\![t_2'']\!]_{\mathcal{A}_2}$ for some $t_2'' \in t_1'' \oslash_x t$ By the definition of $\mathcal{A}_q$, $q_2''' \in f_2^a(q_2', q_2'')$ and so $q_2''' \in [\![t_1' \mid a[t_2'']]\!]_{\mathcal{A}_2}$. Furthermore, $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q_2''')$ and so $q_2 \in [\![t_1' \mid a[t_2'']]\!]_{\mathcal{A}_2}$. Observe that $t_1' \mid a[t_2''] \in t_1' \mid a[t_1'' \oslash_x t] \subseteq (t_1 \mid a[t_1'']) \oslash_x t$ and so the choice $t_2 = t_1' \mid a[t_2'']$ fulfills the requirements.

$\Longleftarrow$: Suppose that there is a tree t_2 such that $q_2 \in [\![t_2]\!]_{\mathcal{A}_2}$ and $t_2 \in t_1 \oslash_x t$.

Base case: $t_1 = \varepsilon$.
In this case, $t_1 \oslash_x t = \emptyset$ and so there is no such t_2 and the implication holds trivially.

Inductive case: $t_1 = t_1' \mid y$ for some $y \in \Omega$.
In this case, either:

- $t_2 = t_2' \mid y$ for some $t_2' \in t_1' \oslash_x t$; or
- $y = y$ and $t_2 = t_1' \mid t$.

In the first case, it must be that $q_2 \in (\varepsilon\text{-}\mathsf{closure}_2 \circ f_2^y)(q_2')$ for some $q_2' \in [\![t_2']\!]_{\mathcal{A}_2}$. By the inductive hypothesis, $(q_2', 1) \in [\![t_1']\!]_{\mathcal{A}_q}$. By the construction of $\mathcal{A}_q$, we have that $(q_2, 1) \in (\varepsilon\text{-}\mathsf{closure}_q \circ f_q^y)((q_2', 1))$ and so $(q_2, 1) \in [\![t_1' \mid y]\!]_{\mathcal{A}_q} = [\![t_1]\!]_{\mathcal{A}_q}$, as required.

In the second case, it must be that $q_2 \in (\!|t|\!)_{\mathcal{A}_2}(q_2') = q(q_2')$ for some $q_2' \in [\![t_1']\!]_{\mathcal{A}_2}$. By Lem. 30, $(q_2', 0) \in [\![t_1']\!]_{\mathcal{A}_q}$. By the construction of $\mathcal{A}_q$, we have that $(q_2, 1) \in f_x^q((q_2', 0))$ and so $(q_2', 1) \in [\![t_1' \mid x]\!]_{\mathcal{A}_q} = [\![t_1]\!]_{\mathcal{A}_q}$, as required.

Inductive case: $t_1 = t_1' \mid a[t_1'']$ for some $a \in \Upsilon$.
In this case, either:

- $t_2 = t_2' \mid a[t_1'']$ for some $t_2' \in t_1' \oslash_x t$; or

– $t_2 = t'_1 \mid a[t''_2]$ for some $t''_2 \in t''_1 \oslash_x t$.

In the first case, it must be that $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q'''_2)$, for some $q'''_2 \in f^a_2(q'_2, q''_2)$, for some $q'_2 \in [\![t'_2]\!]_{\mathcal{A}_2}$, $q''_2 \in [\![t''_1]\!]_{\mathcal{A}_2}$. By the inductive hypothesis, $(q'_2, 1) \in [\![t'_1]\!]_{\mathcal{A}_q}$. By Lem. 30, $t''_2, 0) \in [\![t''_1]\!]_{\mathcal{A}_q}$. Furthermore, $(q'''_2, 1) \in f^a_q((q'_2, 1), (q''_2, 0))$ and so $(q'''_2, 1) \in [\![t'_1 \mid a[t''_1]]\!]_{\mathcal{A}_q}$. Also, $(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}_2((q'''_2, 1))$ and so $(q_2, 1) \in [\![t'_1 \mid a[t''_1]]\!]_{\mathcal{A}_q} = [\![t_1]\!]_{\mathcal{A}_q}$, as required.

In the second case, it must be that $q_2 \in \varepsilon\text{-}\mathsf{closure}_2(q'''_2)$, for some $q'''_2 \in f^a_2(q'_2, q''_2)$, for some $q'_2 \in [\![t'_1]\!]_{\mathcal{A}_2}$, $q''_2 \in [\![t''_2]\!]_{\mathcal{A}_2}$. By Lem. 30, $(q'_2, 0) \in [\![t'_1]\!]_{\mathcal{A}_q}$. By the inductive hypothesis, $(t''_2, 1) \in [\![t''_1]\!]_{\mathcal{A}_q}$. Furthermore, $(q'''_2, 1) \in f^a_q((q'_2, 0), (q''_2, 1))$ and so $(q'''_2, 1) \in [\![t'_1 \mid a[t''_1]]\!]_{\mathcal{A}_q}$. Also, $(q_2, 1) \in \varepsilon\text{-}\mathsf{closure}_2((q'''_2, 1))$ and so $(q_2, 1) \in [\![t'_1 \mid a[t''_1]]\!]_{\mathcal{A}_q} = [\![t_1]\!]_{\mathcal{A}_q}$, as required.

□

Proposition 6. *The automaton defined in Def. 19 accepts the language* $L_1 \oslash\!\!-^{\exists}_{x} L_2$.

Proof.

$$
\begin{array}{rc}
 & t \in L_1 \oslash\!\!-^{\exists}_{x} L_2 \\
\iff & \exists t_1, t_2.\, t_1 \in L_1 \wedge t_2 \in L_2 \wedge t_2 \in t_1 \oslash_x t \\
\iff & \exists t_1, t_2.\, \exists q_2.\, t_1 \in L_1 \wedge q_2 \in [\![t_2]\!]_{\mathcal{A}_2} \wedge q_2 \in A_2 \wedge t_2 \in t_1 \oslash_x t \\
\iff & \exists q.\, q = (\!|t|\!)_{\mathcal{A}_2} \wedge \exists t_1, t_2.\, \exists q_2.\, t_1 \in L_1 \wedge q_2 \in [\![t_2]\!]_{\mathcal{A}_2} \wedge q_2 \in A_2 \wedge t_2 \in t_1 \oslash_x t \\
\iff & \exists q.\, q = (\!|t|\!)_{\mathcal{A}_2} \wedge \exists t_1.\, \exists q_2.\, t_1 \in L_1 \wedge (q_2, 1) \in [\![t_1]\!]_{\mathcal{A}_q} \wedge (q_2, 1) \in A_q \\
\iff & \exists q.\, q = (\!|t|\!)_{\mathcal{A}_2} \wedge \exists t_1.\, [\![t_1]\!]_{\mathcal{A}_1} \cap A_1 \neq \emptyset \wedge [\![t_1]\!]_{\mathcal{A}_q} \cap A_q \neq \emptyset \\
\iff & \exists q.\, q = (\!|t|\!)_{\mathcal{A}_2} \wedge q \in A \\
\iff & \exists q.\, q = [\![t]\!]_{\mathcal{A}} \wedge q \in A \\
\iff & [\![t]\!]_{\mathcal{A}} \cap A \neq \emptyset.
\end{array}
$$

□

A.5 Decidability with Quantifiers

The following lemma is assumed.

Lemma 32 (Duality of Freshness).

$$c, \sigma \models \text{И}\alpha.\, P \iff \forall x \in \Omega.\, x \sharp c, \sigma \implies c, \sigma[\alpha \mapsto x] \models P$$

Lemma 33 (Environment Extensionality). *For all* c, σ, P, α, x *where* $\alpha \notin dom(\sigma)$ *and* $\alpha \notin fv(P)$,

$$c, \sigma \models P \quad \iff \quad c, \sigma[\alpha \mapsto x] \models P.$$

Proof. The proof is by induction on the structure of the formula P. The majority of cases are trivial: since $\alpha \notin fv(P)$, the criteria for the satifaction relation are independent of whether α is bound in the environment. The only non-trivial case is when $P = \reflectbox{N}\beta.\, P'$, since this deals with σ. In this case:

$$\begin{array}{rcl} c, \sigma \models \reflectbox{N}\beta.\, P' & \iff & \exists y.\, y \sharp c, \sigma \ \wedge \ c, \sigma[\beta \mapsto y] \models P' \\ \text{(by swapping } y \text{ and } z) & \iff & \exists z.\, z \sharp c, \sigma[\alpha \mapsto x] \ \wedge \ c, \sigma[\beta \mapsto z] \models P' \\ \text{(by inductive hypothesis)} & \iff & \exists z \sharp c, \sigma[\alpha \mapsto x] \ \wedge \ c, \sigma[\alpha \mapsto x][\beta \mapsto z] \models P' \\ & \iff & c, \sigma[\alpha \mapsto x] \models \reflectbox{N}\beta.\, P'. \end{array}$$

□

We make use of the environment extensionality lemma extensively and often implicitly.

Lemma 1 (Encoding Existential with Freshness). *For all P,*

$$\exists\alpha.\, P \ \equiv \ \reflectbox{N}\alpha.\, P \circ_\alpha \left(\mapsto \wedge \neg \bigvee_{\beta \in fv(P) \setminus \{\alpha\}} \beta \right) \vee P \vee \bigvee_{\beta \in fv(P) \setminus \{\alpha\}} P[\beta/\alpha].$$

Consequently, every formula can be rewritten to an equivalent formula that contains no existential quantifiers.

Proof. By environment extensionality, we assume without loss of generality that $dom(\sigma) = fv(P) \setminus \{\alpha\}$. Let

$$P' = P \circ_\alpha \left(\mapsto \wedge \neg \bigvee_{\beta \in fv(P) \setminus \{\alpha\}} \beta \right) \vee P \vee \bigvee_{\beta \in fv(P) \setminus \{\alpha\}} P[\beta/\alpha].$$

$\Longrightarrow$: Suppose

$$c, \sigma \models \exists\alpha.\, P$$

and hence

$$\exists x.\, c, \sigma[\alpha \mapsto x] \models P.$$

We consider the possible cases for x.

If $x \sharp c, \sigma$ then $c, \sigma \models \reflectbox{N}\alpha.\, P$ and so $c, \sigma \models \reflectbox{N}\alpha.\, P'$.

If $x \in range(\sigma)$ (and so $x = \sigma(\beta)$ for some β) then $c, \sigma \models P[\beta/\alpha]$ (by induction). Hence $c, \sigma \models \reflectbox{N}\alpha.\, P'$.

If $x \sharp \sigma$ but $x \in fn(c)$ then for $y \sharp \sigma, c$

$$c = c[y/x] \; y \; x$$

$$c[y/x], \sigma[\alpha \mapsto y] \models P$$

$$x, \sigma[\alpha \mapsto y] \models \mapsto \wedge \neg \bigvee_{\beta \in fv(P) \setminus \{\alpha\}} \beta \qquad (\text{since } x \sharp \sigma\ )$$

so

$$\begin{array}{lrl} & c, \sigma[\alpha \mapsto y] \models P \circ_\alpha \left(\mapsto\!\!\!| \wedge \neg \bigvee_{\beta \in fv(P) \setminus \{\alpha\}} \beta \right) \\ \therefore & c, \sigma[\alpha \mapsto y] \models P' \\ \therefore & c, \sigma \models \text{И}\alpha.\, P'. \end{array}$$

The three cases we have considered for x cover all possibilities, and hence we can conclude, as required, $c, \sigma \models \text{И}\alpha.\, P'$.

$\Longleftarrow$: Suppose

$$c, \sigma \models \text{И}\alpha.\, P'$$

and hence

$$\exists x.\, x \sharp c, \sigma \;\wedge\; c, \sigma[\alpha \mapsto x] \models P'.$$

One of the disjuncts of P' must be satisfied; we consider the possible cases.

If

$$c, \sigma[\alpha \mapsto x] \models P' \circ_\alpha \left(\mapsto\!\!\!| \wedge \neg \bigvee_{\beta \in fv(P) \setminus \{\alpha\}} \beta \right)$$

then there exist c' and y with $y \sharp \sigma$ such that

$$\begin{array}{c} c = c' \textcircled{x}\, y \qquad \text{and} \\ c', \sigma[\alpha \mapsto x] \models P. \end{array}$$

By swapping x and y, we see that

$$c, \sigma[\alpha \mapsto y] \models P.$$

Hence

$$c, \sigma \models \exists \alpha.\, P.$$

If $c, \sigma[\alpha \mapsto x] \models P$ then $c, \sigma \models \exists \alpha.\, P$.

If $c, \sigma[\alpha \mapsto x] \models P[\beta/\alpha]$ (where $\sigma(\beta) = y$) then $c, \sigma[\alpha \mapsto y] \models P$ (by induction on P). Hence, $c, \sigma \models \exists \alpha.\, P$.

In each case, we have that $c, \sigma \models \exists \alpha.\, P$, as required. $\square$

Lemma 2 (Prenex Normalisation). *The following logical equivalences hold.*

$$a[\text{И}\alpha.\, P] \equiv \text{И}\alpha.\, a[P] \tag{1}$$
$$P' \mid (\text{И}\alpha.\, P) \equiv \text{И}\alpha.\, P' \mid P \tag{2}$$
$$(\text{И}\alpha.\, P) \mid P' \equiv \text{И}\alpha.\, P \mid P' \tag{3}$$
$$P' \circ_\beta (\text{И}\alpha.\, P) \equiv \text{И}\alpha.\, P' \circ_\beta P \tag{4}$$
$$(\text{И}\alpha.\, P) \circ_\beta P' \equiv \text{И}\alpha.\, P \circ_\beta P' \tag{5}$$
$$P' \multimap^{\exists}_{\beta} (\text{И}\alpha.\, P) \equiv \text{И}\alpha.\, P' \multimap^{\exists}_{\beta} (P \wedge \neg\Diamond\alpha) \tag{6}$$
$$(\text{И}\alpha.\, P) \multimap^{\exists}_{\beta} P' \equiv \text{И}\alpha.\, (P \wedge \neg\Diamond\alpha) \multimap^{\exists}_{\beta} P' \tag{7}$$
$$P' \circ\!\!-^{\exists}_{\beta} (\text{И}\alpha.\, P) \equiv \text{И}\alpha.\, P' \circ\!\!-^{\exists}_{\beta} (P \wedge \neg\Diamond\alpha) \tag{8}$$
$$(\text{И}\alpha.\, P) \circ\!\!-^{\exists}_{\beta} P' \equiv \text{И}\alpha.\, (P \wedge \neg\Diamond\alpha) \circ\!\!-^{\exists}_{\beta} P' \tag{9}$$
$$\Diamond\text{И}\alpha.\, P \equiv \text{И}\alpha.\, \Diamond P \tag{10}$$
$$\neg\text{И}\alpha.\, P \equiv \text{И}\alpha.\, \neg P \tag{11}$$
$$P' \wedge (\text{И}\alpha.\, P) \equiv \text{И}\alpha.\, P' \wedge P. \tag{12}$$

Consequently, every $\exists$-free formula is equivalent to a formula in which all quantifiers appear at the head of the formula — the prenex normal form*.*

Proof. Fix c, σ.

Equivalence (1):

$$\begin{array}{ll} & c, \sigma \models a[\text{И}\alpha.\, P] \\ \iff & \exists c' \in C_\Omega.\, c = a[c'] \,\wedge\, \exists y \in \Omega.\, y \sharp c', \sigma \,\wedge\, c', \sigma[\alpha \mapsto y] \models P \end{array}$$

$(fn(c) = fn(c'))$

$$\begin{array}{ll} \iff & \exists y \in \Omega.\, y \sharp c, \sigma \,\wedge\, \exists c'.\, c = a[c'] \,\wedge\, c', \sigma[\alpha \mapsto y] \models P \\ \iff & c, \sigma \models \text{И}\alpha.\, a[P] \end{array}$$

Equivalence (2):

$$\begin{array}{ll} & c, \sigma \models P' \mid (\text{И}\alpha.\, P) \\ \iff & \exists c_1, c_2 \in C_\Omega.\, c = c_1 \mid c_2 \,\wedge\, c_1, \sigma \models P' \,\wedge \\ & \qquad \exists y \in \Omega.\, y \sharp c_2, \sigma \,\wedge\, c_2, \sigma[\alpha \mapsto y] \models P \end{array}$$

(swapping, since $fn(c_2) \subseteq fn(c)$)

$$\begin{array}{ll} \iff & \exists z \in \Omega.\, z \sharp c, \sigma \,\wedge\, \exists c_1, c_2.\, c = c_1 \mid c_2 \,\wedge \\ & \qquad c_1, \sigma[\alpha \mapsto z] \models P' \,\wedge\, c_2, \sigma[\alpha \mapsto z] \models P \\ \iff & c, \sigma \models \text{И}\alpha.\, P \mid P'. \end{array}$$

The proof of (3) is symmetric to that of (2).

Equivalence (4): Let $x = \sigma(\beta)$.

$$\begin{array}{rl} & c, \sigma \models P' \circ_\beta (\text{И}\alpha.\, P) \\ \iff & \exists c_1, c_2 \in C_\Omega.\, c = c_1 \circledast c_2 \,\wedge\, c_1, \sigma \models P' \wedge \\ & \quad \exists y \in \Omega.\, y \sharp c_2, \sigma \,\wedge\, c_2, \sigma[\alpha \mapsto y] \models P \end{array}$$

(swapping, since $fn(c_2) \subseteq fn(c)$)

$$\begin{array}{rl} \iff & \exists z \in \Omega.\, z \sharp c, \sigma \,\wedge\, \exists c_1, c_2.\, c = c_1 \circledast c_2 \wedge \\ & \quad c_1, \sigma[\alpha \mapsto z] \models P' \,\wedge\, c_2, \sigma[\alpha \mapsto z] \models P \\ \iff & c, \sigma \models \text{И}\alpha.\, P' \circ_\beta P. \end{array}$$

Equivalence (5): Let $x = \sigma(\beta)$.

$$\begin{array}{rl} & c, \sigma \models (\text{И}\alpha.\, P) \circ_\beta P' \\ \iff & \exists c_1, c_2 \in C_\Omega.\, c = c_1 \circledast c_2 \,\wedge\, c_2, \sigma \models P' \wedge \\ & \quad \exists y \in \Omega.\, y \sharp c_1, \sigma \,\wedge\, c_1, \sigma[\alpha \mapsto y] \models P \end{array}$$

(swapping, since $fn(c_1) \subseteq fn(c) \cup \{x\}$ and $y \sharp \sigma \implies y \neq x$)

$$\begin{array}{rl} \iff & \exists z \in \Omega.\, z \sharp c, \sigma \,\wedge\, \exists c_1, c_2 \in C_\Omega.\, c = c_1 \circledast c_2 \wedge \\ & \quad c_1, \sigma[\alpha \mapsto z] \models P \,\wedge\, c_2, \sigma[\alpha \mapsto z] \models P' \\ \iff & c, \sigma \models \text{И}\alpha.\, P \circ_\beta P'. \end{array}$$

Equivalence (6): Let $x = \sigma(\beta)$.

$$\begin{array}{rl} & c, \sigma \models P' \multimap^{\exists}_\beta (\text{И}\alpha.\, P) \\ \iff & \exists c_1, c_2 \in C_\Omega.\, c_2 = c \circledast c_1 \,\wedge\, c_1, \sigma \models P' \wedge \\ & \quad \exists y \in \Omega.\, y \sharp c_2, \sigma \,\wedge\, c_2, \sigma[\alpha \mapsto y] \models P \\ \iff & \exists c_1, c_2 \in C_\Omega.\, c_2 = c \circledast c_1 \,\wedge\, c_1, \sigma \models P' \wedge \\ & \quad \exists y \in \Omega.\, y \sharp \sigma \,\wedge\, c_2, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha \end{array}$$

(swapping)

$$\begin{array}{rl} \iff & \exists z \in \Omega.\, z \sharp c, \sigma \,\wedge\, \exists c_1, c_2 \in C_\Omega.\, c_2 = c \circledast c_1 \wedge \\ & \quad c_1, \sigma[\alpha \mapsto z] \models P' \,\wedge\, c_2, \sigma[\alpha \mapsto z] \models P \wedge \neg\Diamond\alpha \\ \iff & c, \sigma \models \text{И}\alpha.\, P' \multimap^{\exists}_\beta (P \wedge \neg\Diamond\alpha). \end{array}$$

Equivalence (7): Let $x = \sigma(\beta)$.

$$\begin{array}{rl} & c, \sigma \models (\text{И}\alpha.\, P) \multimap^{\exists}_\beta P' \\ \iff & \exists c_1, c_2 \in C_\Omega.\, c_2 = c \circledast c_1 \,\wedge\, c_2, \sigma \models P' \wedge \\ & \quad \exists y \in \Omega.\, y \sharp c_1, \sigma \,\wedge\, c_1, \sigma[\alpha \mapsto y] \models P \\ \iff & \exists c_1, c_2 \in C_\Omega.\, c_2 = c \circledast c_1 \,\wedge\, c_2, \sigma \models P' \wedge \\ & \quad \exists y \in \Omega.\, y \sharp \sigma \,\wedge\, c_1, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha \end{array}$$

(swapping)

$$
\begin{aligned}
\iff \quad & \exists z \in \Omega.\, z \sharp c, \sigma \;\wedge\; \exists c_1, c_2 \in C_\Omega.\, c_2 = c ⓧ c_1 \wedge \\
& \quad c_2, \sigma[\alpha \mapsto z] \models P' \;\wedge\; c_1, \sigma[\alpha \mapsto z] \models P \wedge \neg\Diamond\alpha \\
\iff \quad & c, \sigma \models \text{И}\alpha.\, (P \wedge \neg\Diamond\alpha) \multimap^{\exists}_{\beta} P'.
\end{aligned}
$$

Equivalence (9): Let $x = \sigma(\beta)$.

$$
\begin{aligned}
& c, \sigma \models P' \circ\!\!-^{\exists}_{\beta} (\text{И}\alpha.\, P) \\
\iff \quad & \exists c_1, c_2 \in C_\Omega.\, c_2 = c_1 ⓧ c_1 \;\wedge\; c_1, \sigma \models P' \wedge \\
& \quad \exists y \in \Omega.\, y \sharp c_2, \sigma \;\wedge\; c_2, \sigma[\alpha \mapsto y] \models P \\
\iff \quad & \exists c_1, c_2 \in C_\Omega.\, c_2 = c_1 ⓧ c_1 \;\wedge\; c_1, \sigma \models P' \wedge \\
& \quad \exists y \in \Omega.\, y \sharp \sigma \;\wedge\; c_2, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha
\end{aligned}
$$

(swapping)

$$
\begin{aligned}
\iff \quad & \exists z \in \Omega.\, z \sharp c, \sigma \;\wedge\; \exists c_1, c_2 \in C_\Omega.\, c_2 = c_1 ⓧ c_2 \wedge \\
& \quad c_1, \sigma[\alpha \mapsto z] \models P' \;\wedge\; c_2, \sigma[\alpha \mapsto z] \models P \wedge \neg\Diamond\alpha \\
\iff \quad & c, \sigma \models \text{И}\alpha.\, P' \circ\!\!-^{\exists}_{\beta} (P \wedge \neg\Diamond\alpha).
\end{aligned}
$$

Equivalence (8): Let $x = \sigma(\beta)$.

$$
\begin{aligned}
& c, \sigma \models (\text{И}\alpha.\, P) \circ\!\!-^{\exists}_{\beta} P' \\
\iff \quad & \exists c_1, c_2 \in C_\Omega.\, c_2 = c_1 ⓧ c_1 \;\wedge\; c_2, \sigma \models P' \wedge \\
& \quad \exists y \in \Omega.\, y \sharp c_1, \sigma \;\wedge\; c_1, \sigma[\alpha \mapsto y] \models P \\
\iff \quad & \exists c_1, c_2 \in C_\Omega.\, c_2 = c_1 ⓧ c_1 \;\wedge\; c_2, \sigma \models P' \wedge \\
& \quad \exists y \in \Omega.\, y \sharp \sigma \;\wedge\; c_1, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha
\end{aligned}
$$

(swapping)

$$
\begin{aligned}
\iff \quad & \exists z \in \Omega.\, z \sharp c, \sigma \;\wedge\; \exists c_1, c_2 \in C_\Omega.\, c_2 = c_1 ⓧ c_2 \wedge \\
& \quad c_2, \sigma[\alpha \mapsto z] \models P' \;\wedge\; c_1, \sigma[\alpha \mapsto z] \models P \wedge \neg\Diamond\alpha \\
\iff \quad & c, \sigma \models \text{И}\alpha.\, P' \circ\!\!-^{\exists}_{\beta} (P \wedge \neg\Diamond\alpha).
\end{aligned}
$$

Equivalence (11):

$$
\begin{aligned}
& c, \sigma \models \neg\text{И}\alpha.\, P \\
\iff \quad & \neg(\exists y \in \Omega.\, y \sharp c, \sigma \;\wedge\; c, \sigma[\alpha \mapsto y] \models P) \\
\iff \quad & \forall z \in \Omega.\, z \sharp c, \sigma \implies c, \sigma[\alpha \mapsto z] \models P \\
\iff \quad & c, \sigma \models \text{И}\alpha.\, \neg P.
\end{aligned}
$$

Equivalence (12):

$$
\begin{array}{ll}
& c, \sigma \models P' \wedge (\text{И}\alpha.\, P) \\
\iff & \quad c, \sigma \models P \;\wedge\; \exists y \in \Omega.\, y \sharp c, \sigma \;\wedge\; c, \sigma[\alpha \mapsto y] \models P \\
\iff & \quad \exists y \in \Omega.\, y \sharp c, \sigma \;\wedge\; c, \sigma[\alpha \mapsto y] \models P' \;\wedge\; c, \sigma[\alpha \mapsto y] \models P \\
\iff & c, \sigma \models \text{И}\alpha.\, P' \wedge P.
\end{array}
$$

□

Lemma 2 (Deciding Satisfiability). *For all environments σ, formulae P, and hole variables α with $\alpha \notin dom(\sigma)$,*

$$
\begin{array}{ll}
& \exists c \in C_{\Omega}.\, c, \sigma \models \text{И}\alpha.\, P \\
\iff & \exists y \in \Omega.\, y \sharp \sigma \;\wedge\; \exists c \in C_{\Omega}.\, c, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha \\
\iff & \forall y \in \Omega.\, y \sharp \sigma \implies \exists c \in C_{\Omega}.\, c, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha.
\end{array}
$$

Proof.

$$
\begin{array}{ll}
& \exists c \in C_{\Omega}.\, c, \sigma \models \text{И}\alpha.\, P \\
\iff & \exists c \in C_{\Omega}.\, \exists y \in \Omega.\, y \sharp c, \sigma \;\wedge\; c, \sigma[\alpha \mapsto y] \models P \\
\iff & \exists y \in \Omega.\, y \sharp \sigma \;\wedge\; \exists c \in C_{\Omega}.\, c, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha \\
\iff & \forall y \in \Omega.\, y \sharp \sigma \implies \exists c \in C_{\Omega}.\, c, \sigma[\alpha \mapsto y] \models P \wedge \neg\Diamond\alpha.
\end{array}
$$

□