A Congruence-Based Perspective on Finite Tree Automata

Pierre Ganty[*]**, Elena Gutiérrez, Pedro Valero**

IMDEA Software Institute

Madrid, Spain

pierre.ganty@imdea.org, elenagutiv@gmail.com, pevalme@fb.com

Abstract. We provide new insights on the determinization and minimization of tree automata using congruences on trees. From this perspective, we study a Brzozowski's style minimization algorithm for tree automata. First, we prove correct this method relying on the following fact: when the automata-based and the language-based congruences coincide, determinizing the automaton yields the minimal one. Such automata-based congruences, in the case of word automata, are defined using pre and post operators. Now we extend these operators to tree automata, a task that is particularly challenging due to the reduced expressive power of deterministic top-down (or equivalently co-deterministic bottom-up) automata. We leverage further our framework to offer an extension of the original result by Brzozowski for word automata.

1.   Introduction

Finite tree automata are a well-studied [1, 2, 3] automata model processing tree structures, as opposed to the classical finite-state automata, which process words, i.e., trees where every node has at most one child. Examples of applications of tree automata include model checking [4, 5], natural language processing [6] and representing the nested structured of tree-based metalanguages, such as XML [7].

There exist two classes of tree automata, which differ in the way they process the input tree: *bottom-up tree automata* (BTAs) and *top-down tree automata* (TTAs). In their non-deterministic flavor, both types of tree automata have the same expressive power: they are both finite representations of the *regular tree languages*.

---

[*]Address of correspondence: IMDEA Software Institute, Madrid, Spain.

Like word automata, tree automata (both BTAs and TTAs) can be deterministic (DBTAs and DT-TAs) or non-deterministic, offering the classical trade-off, where deterministic automata are easier to reason about while non-deterministic ones are more concise. This situation has motivated the study of techniques to reduce the number of states of deterministic automata [8, 9, 10] as well as methods for building deterministic automata that are *minimal* in the number of states [11, 12, 13]. For both word and tree automata the minimal deterministic automaton is unique (up to isomorphisms).

Unlike word automata, where every regular language is definable by a deterministic automaton, there exist regular tree languages that cannot be defined by a deterministic TTA (equivalently, they cannot be defined by a co-deterministic BTA). It turns out that DTTAs and co-deterministic BTAs (co-DBTAs) define a subclass of regular tree languages called *path-closed languages* [14].

In this paper, we address a Brzozowski's style algorithm for minimizing TAs [12]. That is, given a BTA or a TTA defining the language L, the algorithm combines a co-determinization and a determinization operation to build the minimal DBTA or DTTA for L. In this sense, the algorithm works in the same fashion as the classical Brzozowski's algorithm [15] (also known as the double-reversal method) for finding the minimal deterministic word automaton for a given regular language. Note that in the tree case, the method only applies to tree automata generating path-closed tree languages since it requires the construction of a co-DBTA or a DTTA for the given language.

Brzozowski's algorithm relies on the fact that determinizing a co-deterministic word automaton returns the minimal deterministic automaton. The reason why this method is also coined as the *double-reversal* method is that it builds a co-deterministic automaton for the input language by combining a reverse, determinization and reverse operation.

1.1. Contributions

We study Brzozowski's minimization algorithm for tree automata from the perspective of *congruences* on trees. In this sense, we build on work by Ganty et al. [16] who applied congruences on words to the study of word automata minimization techniques. In this work, we use congruences of finite index on *trees* and *contexts*. The latter are trees for which exactly one node is labeled by a distinguished symbol of arity 0.

Concretely, we use so-called *upward congruences* [17, 1] over trees, which are equivalences on trees that behave well w.r.t. the concatenation of symbols on top of the tree; and *downward* congruences [13], i.e., equivalences on contexts that behave well when concatenating contexts on the leaves. Given a tree language L and a congruence satisfying some required properties, we show how to build deterministic and co-deterministic automata defining L.

We leverage two kinds of congruences: *language-based* congruences, defined in terms of a regular tree language, and *automaton-based* congruences, relative to an automaton. Hence we show how to use upward automata-based congruences to construct DBTAs and DTTAs, which are isomorphic to those obtained with the standard subset constructions, while using the upward language-based congruences results in minimal DBTAs and DTTAs. We also show that a similar reasoning holds for downward congruences and (minimal) co-DBTAs and co-DTTAs.

While upward congruences allows us to build deterministic BTAs, we observe that downward congruences must satisfy an extra condition to guarantee that they yield BTAs that are *co-deterministic*.

Our first contribution is to identify the class of congruences satisfying this condition, which we coin *strongly downward congruences* (Definition 3.8). Roughly speaking, since not every tree language has a finite representation in the form of a co-DBTA (recall that only path-closed languages do), the so-called strongly downward congruences attempt to capture the notion of path-closedness in their definition.

Secondly, unlike the language-based upward [17, 1] and downward congruences [13], which have been studied previously in the context of TA minimization; to the best of our knowledge, our automata-based congruences are novel. For this purpose, *we define an operator* $\mathrm{post}(\cdot)$ *for TAs* (Definition 4.5), in the same fashion to the existing one for word automata, and we use it to define the automata-based upward congruence (Definition 4.12). Analogously, *we introduce an operator* $\mathrm{pre}(\cdot)$ (Definition 4.5) which allows us to define its downward counterpart. While the definition of $\mathrm{post}(\cdot)$ is a straightforward generalization of the word case, that is not the case of $\mathrm{pre}(\cdot)$. One more time, our $\mathrm{pre}(\cdot)$ operator aims to capture the notion of path-closedness that is imposed by our goal of providing *co-deterministic* BTA constructions.

Then, by chaining together the right constructions we obtain a simple *proof of correctness of a Brzozowski's style algorithm* (Corollary 5.2) producing the minimal DBTA for a given tree language. This algorithm was first proposed by Björklund and Cleophas [12]. However, they did not include a proof, which, far from being conceptually new, could have resulted in several lines of technical details. We believe that our proof does bring new insights in form of new non-trivial notions adapted from the classical word automata.

Finally, *we generalize Brzozowski's algorithm* similarly to what Brzozowski and Tamm [18] have done for the case of automata over words. More precisely, we give a sufficient and necessary condition on BTAs such that their determinization produces the minimal DBTA (Theorem 5.3). This condition lifts the limitation to path-closed languages all the way up to regular tree languages.

In the main part of the document we only consider BTAs. In Appendix A.1 we adapt our results to TTAs enabled by a "reversal" operation on tree automata.

2. Preliminaries

We write $\mathbb{N}$ for the natural numbers and $\mathbb{N}_+$ for $\mathbb{N} \setminus \{0\}$. A tree domain is a finite set of sequences over $\mathbb{N}_+$ describing a tree structure. Formally, a *tree domain* D is a finite non-empty set $D \subseteq (\mathbb{N}_+)^*$ such that for each $v \in (\mathbb{N}_+)^*, n \in \mathbb{N}_+$:

(i) if $v \cdot n \in D$ then $v \in D$, i.e., D is prefix-closed, and

(ii) if $v \cdot n \in D$ and $n > 1$ then $v \cdot (n-1) \in D$.

The elements of D are called *nodes* and every tree domain contains a node ε called the *root*. For example, $D = \{\varepsilon, 1, 2, 3, 3 \cdot 1, 3 \cdot 2\}$ is a tree domain, while $D' = \{\varepsilon, 1, 2, 3, 1 \cdot 2, 1 \cdot 3\}$ is not. Note that D' does not satisfy condition (ii) since $1 \cdot 1 \notin D'$.

Given an *alphabet*, i.e., a finite non-empty set of *symbols*, a *tree* is a total function that maps nodes onto symbols. Formally, given an alphabet A and a tree domain $D \subseteq (\mathbb{N}_+)^*$, an A-labeled *tree* is a

function $t\colon D \to A$. We use $\mathrm{dom}(t)$ to denote the tree domain of a tree t, and $t(v)$ to denote the label of a node $v \in \mathrm{dom}(t)$.

The rank of a node is the number of its children. Formally, given a tree t, the *rank* of $v \in \mathrm{dom}(t)$, denoted $\langle v \rangle_t$, is $|\{k \in \mathbb{N}_+ \mid v \cdot k \in \mathrm{dom}(t)\}|$. Nodes of rank 0 are *leaves*, and $\ell(t)$ is the set of leaves of t. We say that a tree is *monadic* iff $\forall v \in \mathrm{dom}(t)\colon \langle v \rangle_t \leq 1$. Note that, in the context of word automata, monadic trees over an alphabet A can be interpreted as words over A.

The *height* of a tree t is defined as $\mathrm{h}(t) \stackrel{\mathrm{def}}{=} 1 + \max\{|v| \mid v \in \mathrm{dom}(t)\}$, where $|v|$ denotes the *length* of node v when interpreted as a sequence in $(\mathbb{N}_+)^*$.

Given a tree t with root $v \in \mathrm{dom}(t)$ the *subtree* t' rooted at v is such that $\mathrm{dom}(t') = \{u \in (\mathbb{N}_+)^* \mid v \cdot u \in \mathrm{dom}(t)\}$ and $t'(u) = t(vu)$, for every $u \in \mathrm{dom}(t')$. Let $t(v) = f$ for some $v \in \mathrm{dom}(t)$ and let t_i, with $i = 1..\langle v \rangle_t$, be the subtree of t at node $v \cdot i$. Then we denote the subtree t' rooted at node v as $f[t_1, \ldots, t_{\langle v \rangle_t}]$. Given a symbol a, we often write a, instead of $a[\,]$, to describe the tree $t = a[\,]$. For instance, we write $f[a, b]$ instead of $f[a[\,], b[\,]]$.

Example 2.1. We describe our running example for the next definitions. Let $\tilde{t}\colon D \to A$ be the tree shown below, defined as $\tilde{t} \stackrel{\mathrm{def}}{=} f[a, b, g[a, c]]$. In this case, $A = \{a, b, c, g, f\}$ and $\mathrm{dom}(\tilde{t}) = D = \{\varepsilon, 1, 2, 3, 3 \cdot 1, 3 \cdot 2\}$ with $\tilde{t}(\varepsilon) = f$, $\tilde{t}(1) = \tilde{t}(3 \cdot 1) = a$, $\tilde{t}(2) = b$, $\tilde{t}(3) = g$ and $\tilde{t}(3 \cdot 2) = c$. Thus, $\langle 1 \rangle_{\tilde{t}} = \langle 2 \rangle_{\tilde{t}} = \langle 3 \cdot 1 \rangle_{\tilde{t}} = \langle 3 \cdot 2 \rangle_{\tilde{t}} = 0$, and $\ell(\tilde{t}) = \{1, 2, 3 \cdot 1, 3 \cdot 2\}$. On the other hand, $\langle \varepsilon \rangle_{\tilde{t}} = 3$ and $\langle 3 \rangle_{\tilde{t}} = 2$. Finally, the height of $\tilde{t}$ is $\mathrm{h}(\tilde{t}) = 3$. Figure 1 shows a depiction of the tree $\tilde{t}$.
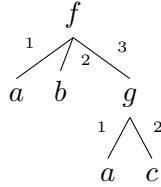

Figure 1.   Depiction of the tree $\tilde{t}$ from Example 2.1.

Next we introduce *ranked trees* building upon *ranked alphabets*. A *ranked alphabet* A is an alphabet partitioned into pairwise disjoint subsets $A = \bigcup_{k \in \mathbb{N}} A_k$ where A_k are the symbols of rank k. Given $f \in A$, the unique index k such that $f \in A_k$ is the *rank* of f and we denote it by $\langle f \rangle$. The rank of a ranked alphabet A is the greatest index k such $A_k \neq \varnothing$.

Given a ranked alphabet A, let $\mathcal{T}_A$ denote the set of all A-labeled *ranked trees* such that, in every tree $t \in \mathcal{T}_A$, the rank of every node $v \in \mathrm{dom}(t)$ coincides with the rank of $t(v)$, i.e., $\forall v \in \mathrm{dom}(t)\colon \langle v \rangle_t = \langle t(v) \rangle$. For an unranked alphabet A, let $\mathcal{T}_A$ denote the set of all A-labeled (*unranked*) trees. In Example 2.1, $\tilde{t} \in \mathcal{T}_A$ is a ranked tree with $A = A_0 \cup A_1 \cup A_2 \cup A_3$ where $A_0 = \{a, b, c\}$, $A_1 = \varnothing$, $A_2 = \{g\}$ and $A_3 = \{f\}$.

A *tree language* over an alphabet A is a subset $L \subseteq \mathcal{T}_A$. Define the *path language* of a tree $t \in \mathcal{T}_A$, denoted by $\pi(t)$, as the subset of $A(\mathbb{N}_+ A)^*$ given by:

$$\pi(t) \stackrel{\mathrm{def}}{=} \begin{cases} \{f\} & \text{if } t = f[\,] \\ \bigcup_{i=1}^{\langle f \rangle}\{f \cdot i \cdot w \mid w \in \pi(t_i)\} & \text{if } t = f[t_1, \ldots, t_{\langle f \rangle}] \end{cases}.$$

The *path language of a tree language* L is defined as $\pi(L) \stackrel{\text{def}}{=} \bigcup_{t \in L} \pi(t)$. A tree language $L \subseteq \mathcal{T}_A$ is *path-closed* iff $\{t \in \mathcal{T}_A \mid \pi(t) \subseteq \pi(L)\} = L$. For example, the path language of tree $\tilde{t}$ is $\pi(\tilde{t}) = \{f1a, f2b, f3g1a, f3g2c\}$. On the other hand, the path language of the tree language $L_1 \stackrel{\text{def}}{=} \{f[a, b], f[b, a]\}$ is $\pi(L_1) = \{f1a, f2b, f1b, f2a\}$ and it is not path-closed since $f[a, a] \notin L_1$ and $\pi(f[a, a]) = \{f1a, f2a\} \subseteq \pi(L_1)$; while $L_2 \stackrel{\text{def}}{=} \{f[a, b], f[b, a], f[a, a], f[b, b]\}$ satisfies that $\pi(L_2) = \pi(L_1)$ and it is path-closed.

2.1. Contexts and substitution

Let A be a ranked alphabet and let $\square \notin A$ be a special symbol with $\langle \square \rangle = 0$. A *context* over A is a tree $t \in \mathcal{T}_{A \cup \{\square\}}$ such that $t(v) = \square$ for exactly one node $v \in \text{dom}(t)$ that we call the *pivot* of t and that we denote $\text{piv}(t)$. Note that if A is a ranked alphabet, so is $A \cup \{\square\}$. We define the $\square$-height of a context $x \in \mathcal{C}_A$, denoted $\text{h}_\square(x)$ as follows $\text{h}_\square(x) \stackrel{\text{def}}{=} 1 + |\text{piv}(x)|$.

The set of all contexts over an alphabet A is denoted by $\mathcal{C}_A$. Since contexts are trees defined over a special alphabet, all notions defined for trees, including the ones we introduce next, apply to contexts. For clarity, we typically use t, r to denote trees and x, y for contexts when the distinction is important.

We define a substitution operator for trees as follows. Let $t, t' \in \mathcal{T}_A$ and let $v \in \text{dom}(t)$. The *tree substitution* $t[\![t']\!]_v$ is the result of replacing the subtree rooted at node v in t with t'. Formally, $t[\![t']\!]_v(u) = t(u)$ for all $u \in \text{dom}(t) \setminus v \cdot (\mathbb{N}_+)^*$ and $t[\![t']\!]_v(v \cdot u) = t'(u)$, for all $u \in \text{dom}(t')$. We omit the subindex v from $t[\![t']\!]_v$ when t is a context and $v = \text{piv}(t)$. For instance, recall the tree $\tilde{t} = f[a, b, g[a, c]]$ from Example 2.1 and let $\tilde{r} = h[a, b, c]$. We have $\tilde{t}[\![\square]\!]_2 = f[a, \square, g[a, c]]$ and $\tilde{t}[\![\tilde{r}]\!]_{31} = f[a, b, g[h[a, b, c], c]]$. Note that $\square[\,]$ satisfies $\square[\![t]\!] = t$, for every $t \in \mathcal{T}_A$, and thus it is called the *identity* context. For monadic trees, tree substitution coincides with word concatenation, where contexts correspond to prefixes of words. In particular, $\square[\,]$ corresponds to ε, the empty word.

Definition 2.2. (Upward and Downward Quotients)
Given a tree language $L \subseteq \mathcal{T}_A$, a tree $t \in \mathcal{T}_A$, we define the *upward quotient* of L w.r.t. t as $L t^{-1} \stackrel{\text{def}}{=} \{x \in \mathcal{C}_A \mid x[\![t]\!] \in L\}$. Similarly, we define the *downward quotient* of L w.r.t. a context $x \in \mathcal{C}_A$ as $x^{-1} L \stackrel{\text{def}}{=} \{t \in \mathcal{T}_A \mid x[\![t]\!] \in L\}$.

Observe that $t \in x^{-1} L \Leftrightarrow x \in L t^{-1}$. It is worth remarking that, in the monadic case, $L t^{-1}$ and $x^{-1} L$ correspond to the so-called *right quotient* of L by t and *left quotient* of L by x, respectively, where t and x are words over A.

2.2. Bottom-up tree automata

Definition 2.3. (Bottom-up tree automaton)
A *bottom-up tree automaton* (BTA) is a tuple $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ where Q is a finite set of *states*; Σ is a ranked alphabet of rank n; $\delta \colon \bigcup_{i=0}^n \Sigma_i \times Q^i \to \wp(Q)$ is the partial *transition function*, where Q^i denotes the set of i-tuples of elements in Q and $\wp(Q)$ denotes the powerset of Q; and $F \subseteq Q$ is the set of *final states*.

We denote a tuple $(f, q_1, \ldots, q_{\langle f \rangle}) \in \Sigma_{\langle f \rangle} \times Q^{\langle f \rangle}$ as $f[q_1, \ldots, q_{\langle f \rangle}]$. Note that if $\langle f \rangle = 0$, we denote the singleton tuple $(f) \in \Sigma_0 \times Q^0$ as $f[]$. We extend δ to sets S of tuples as $\delta(S) \stackrel{\text{def}}{=} \bigcup_{t \in S} \delta(t)$ and define the set of *initial states* of a BTA $\mathcal{A}$ as $\mathrm{i}(\mathcal{A}) \stackrel{\text{def}}{=} \{q \in Q \mid \exists a \in \Sigma_0 \colon q \in \delta(a[])\}$.

A BTA is *deterministic* (DBTA for short) iff every set of states in the image of δ is a singleton or is empty. Similarly, a BTA is *co-deterministic* (co-DBTA for short) iff F is a singleton and for each $q \in Q$ and $f \in \Sigma_n$, with $n \geq 1$, we have: if $q \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])$ and $q \in \delta(f[q_1', \ldots, q_{\langle f \rangle}'])$ then $q_i = q_i'$ for each $i = 1..\langle f \rangle$.

We define the *move* relation on a BTA, denoted by $\to_{\mathcal{A}} \in \mathcal{T}_{\Sigma \cup Q} \times \mathcal{T}_{\Sigma \cup Q}$ as follows. Let $t = x[\![f[t_1, \ldots, t_{\langle f \rangle}]]\!]$ and $t' = x[\![q[t_1, \ldots, t_{\langle f \rangle}]]\!]$ for some $x \in \mathcal{C}_{\Sigma \cup Q}, f \in \Sigma$ and $t_1, \ldots, t_{\langle f \rangle} \in \mathcal{T}_Q$. Then, $t \to_{\mathcal{A}} t' \stackrel{\text{def}}{\Leftrightarrow} q \in \delta(f[t_1(\varepsilon), \ldots, t_{\langle f \rangle}(\varepsilon)])$. We use $\to_{\mathcal{A}}^*$ to denote the reflexive and transitive closure of $\to_{\mathcal{A}}$. The *language* defined by $\mathcal{A}$ is $\mathcal{L}(\mathcal{A}) \stackrel{\text{def}}{=} \{t \in \mathcal{T}_\Sigma \mid \exists t' \in \mathcal{T}_Q \colon t'(\varepsilon) \in F \wedge t \to_{\mathcal{A}}^* t'\}$. Intuitively, a run of a BTA on a tree $t \in \mathcal{T}_\Sigma$ relabels the nodes of the t starting with its leaves and makes its way up towards the root of t as prescribed by the move relation. The run accepts when the root of t is labelled with an accepting state of the BTA.

Remark 2.4. BTAs and DBTAs define the class of *regular tree languages*, while co-DBTAs (which are equivalent to deterministic top-down automata, in the sense that they accept the same class of tree languages, as we shall see in Appendix A.1.2) define the subclass of path-closed tree languages [14]. It is decidable whether the language of a BTA is path-closed [1].

Example 2.5. Let $\mathcal{A} = \langle Q, \Sigma_0 \cup \Sigma_2, \delta, F \rangle$ be a BTA with $Q = \{q_0, q_1\}$, $\Sigma_0 = \{\mathbf{T}, \mathbf{F}\}$, $\Sigma_2 = \{\wedge, \vee\}$, $\{q_0\} = \delta(\wedge[q_0, q_1]) = \delta(\wedge[q_0, q_0]) = \delta(\wedge[q_1, q_0]) = \delta(\vee[q_0, q_0]) = \delta(\mathbf{F}[])$, $\{q_1\} = \delta(\wedge[q_1, q_1]) = \delta(\vee[q_1, q_0]) = \delta(\vee[q_0, q_1]) = \delta(\vee[q_1, q_1]) = \delta(\mathbf{T}[])$ and $F = \{q_1\}$.

Note that $\mathcal{L}(\mathcal{A})$ is defined as the set of all trees of the form $t \in \mathcal{T}_{\Sigma_0 \cup \Sigma_2}$ which yield to propositional formulas, over the binary connectives $\wedge$ and $\vee$ and the constants $\mathbf{T}$ (true) and $\mathbf{F}$ (false), that evaluate to $\mathbf{T}$. For instance, the following is a sequence of moves from a tree $t \in \mathcal{T}_{\Sigma_0 \cup \Sigma_2}$ such that $t \in \mathcal{L}(\mathcal{A})$.

$$
\overbrace{\underset{\mathbf{T}\ \mathbf{F}}{\overset{\vee}{}}\ \mathbf{T}}^{\wedge} \ \to_{\mathcal{A}} \ \cdots \ \to_{\mathcal{A}} \ \overbrace{\underset{q_1\ q_0}{\overset{\vee}{}}\ q_1}^{\wedge} \ \to_{\mathcal{A}} \ \overbrace{\underset{q_1\ q_0}{q_1}\ q_1}^{\wedge} \ \to_{\mathcal{A}} \ \overbrace{\underset{q_1\ q_0}{q_1}\ q_1}^{q_1} \ .
$$

Note that $\wedge[\square, q_1][\![\vee[q_1, q_0]]\!] \to_{\mathcal{A}} \wedge[\square, q_1][\![q_1[q_1, q_0]]\!]$ as $q_1 \in \delta(\vee[q_1, q_0])$. Observe that $\mathcal{A}$ is deterministic but not co-deterministic, since $\{q_0\} = \delta(\wedge[q_0, q_1]) = \delta(\wedge[q_1, q_0])$, for instance. Finally, the language $\mathcal{L}(\mathcal{A})$ is not path-closed since $\vee[\mathbf{F}, \mathbf{F}] \notin \mathcal{L}(\mathcal{A})$ but
$\pi(\vee[\mathbf{F}, \mathbf{F}]) = \{\vee 1\mathbf{F}, \vee 2\mathbf{F}\} \subseteq \pi(\mathcal{L}(\mathcal{A}))$. $\diamondsuit$

For each $q \in Q$ and $S \subseteq Q$, define the *upward* and *downward* language of q w.r.t. S, denoted by $\mathcal{L}_{\uparrow}^{\mathcal{A}}(q, S)$ and $\mathcal{L}_{\downarrow}^{\mathcal{A}}(q, S)$, respectively, as follows:

$$
\mathcal{L}_{\uparrow}^{\mathcal{A}}(q, S) \stackrel{\text{def}}{=} \{c \in \mathcal{C}_\Sigma \mid \exists t' \in \mathcal{T}_Q \colon c[\![q]\!] \to_{\mathcal{A}}^* t', t'(\varepsilon) \in S\}
$$
$$
\mathcal{L}_{\downarrow}^{\mathcal{A}}(q, S) \stackrel{\text{def}}{=} \{t \in \mathcal{T}_\Sigma \mid \exists t' \in \mathcal{T}_Q \colon t \to_{\mathcal{A}}^* t', t'(\varepsilon) = q, \ell(t') \subseteq S\} \ .
$$

We will simplify the notation and write $\mathcal{L}_\uparrow^{\mathcal{A}}(q)$ when $S = F$, and $\mathcal{L}_\downarrow^{\mathcal{A}}(q)$ when $S = \mathrm{i}(\mathcal{A})$. Also, we will drop the superscript $\mathcal{A}$ when the BTA $\mathcal{A}$ is clear from the context. Note that, in the monadic case, $\mathcal{L}_\uparrow^{\mathcal{A}}(q)$ corresponds to the so-called *right language* of state q, i.e., the set of words that can be read from q to a final state of the corresponding word automaton; while $\mathcal{L}_\downarrow^{\mathcal{A}}(q)$ corresponds to the *left language* of q, i.e., the set of words that can be read from an initial state of the corresponding word automaton to state q. When generalizing to trees we have that $\mathcal{L}_\uparrow(q)$ is the set of contexts such that the result of being processed by $\mathcal{A}$ (starting from state q instead of $\Box$) have their root labelled with a final/accepting state. Similarly, $\mathcal{L}_\downarrow(q, S)$ is the set of trees such that their processing by $\mathcal{A}$ return trees with q as root and leaves labelled by S. Finally, it is easy to see that $\mathcal{L}(\mathcal{A}) = \bigcup_{q \in F} \mathcal{L}_\downarrow(q)$ for every BTA $\mathcal{A}$.

Given a ranked alphabet Σ, $f \in \Sigma$ and $T_1, \ldots, T_{\langle f \rangle} \subseteq \mathcal{T}_\Sigma$, let $f[T_1, \ldots, T_{\langle f \rangle}] \stackrel{\text{def}}{=} \{f[t_1, \ldots, t_{\langle f \rangle}] \mid t_i \in T_i, i = 1..\langle f \rangle\}$. This notation allows us to give an inductive characterization of downward languages of a BTA (which will be useful in the proofs of Lemmas 3.3 and 3.7).

Lemma 2.6. Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA. For every $q \in Q$:

$$\mathcal{L}_\downarrow(q) = \{f[\mathcal{L}_\downarrow(q_1), \ldots, \mathcal{L}_\downarrow(q_{\langle f \rangle})] \mid \exists f \in \Sigma \colon q \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])\} \ .$$

Proof:
Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA and let $f \in \Sigma$ and $q, q_1, \ldots, q_{\langle f \rangle} \in Q$ be such that $q \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])$. Then,

$$\forall i \in \{1..\langle f \rangle\} \colon t_i \in \mathcal{L}_\downarrow(q_i) \Leftrightarrow$$
$$\forall i \in \{1..\langle f \rangle\} \colon \exists t_i' \in \mathcal{T}_Q \colon t_i'(\varepsilon) = q_i, \ t_i \to_{\mathcal{A}}^* t_i' \Leftrightarrow$$
$$\exists t' \in \mathcal{T}_Q \colon t'(\varepsilon) = q, \ f[t_1, \ldots, t_{\langle f \rangle}] \to_{\mathcal{A}}^* t' \Leftrightarrow$$
$$f[t_1, \ldots, t_{\langle f \rangle}] \in \mathcal{L}_\downarrow(q) \ .$$

Note that the first and last double-implication hold by definition of downward language of a state w.r.t. $\mathrm{i}(\mathcal{A})$. The second implication holds since, by H.I., $q \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])$. $\qquad\square$

Example 2.7. In Example 2.5, $\mathcal{L}_\uparrow(q_0) \neq \mathcal{L}_\uparrow(q_1)$. In fact, $\Box[\ ] \in \mathcal{L}_\uparrow(q_1)$, since $q_1 \in F$ while $\Box[\ ] \notin \mathcal{L}_\uparrow(q_0)$, since $q_0 \notin F$.

On the other hand, $\mathbf{F}[] \in \mathcal{L}_\downarrow(q_0)$ and $\mathbf{T}[] \in \mathcal{L}_\downarrow(q_1)$. Finally, consider the set of trees $\wedge[\mathcal{L}_\downarrow(q_1), \mathcal{L}_\downarrow(q_1)]$. The reader may check that $\wedge[\mathcal{L}_\downarrow(q_1), \mathcal{L}_\downarrow(q_1)] \subseteq \mathcal{L}_\downarrow(q_1)$ and $\wedge[\mathcal{L}_\downarrow(q_1), \mathcal{L}_\downarrow(q_1)] \not\subseteq \mathcal{L}_\downarrow(q_0)$. In fact, $\mathcal{L}_\downarrow(q_0) \cap \mathcal{L}_\downarrow(q_1) = \varnothing$ as $\mathcal{A}$ is deterministic. $\qquad\Diamond$

A state $q \in Q$ of a BTA is *unreachable* (resp. *empty*) iff its downward (resp. upward) language is empty. The *minimal* DBTA for a regular tree language is the DBTA with the least number of states which is unique modulo isomorphism. Let $\mathcal{A} \equiv \mathcal{A}'$ denote that two BTAs $\mathcal{A}$ and $\mathcal{A}'$ are isomorphic.

Given a BTA $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$, the *bottom-up determinization* builds a DBTA $\mathcal{D} \stackrel{\text{def}}{=} \langle \wp(Q), \Sigma, \delta', F' \rangle$ where $\delta'(f[R_1, \ldots, R_{\langle f \rangle}]) \stackrel{\text{def}}{=} \{q \mid \exists q_1 \in R_1, \ldots, q_{\langle f \rangle} \in R_{\langle f \rangle} \colon q \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])\}$ for each $f \in \Sigma$, $R_i \in \wp(Q)$ and $i = 1..\langle f \rangle$, and $F' \stackrel{\text{def}}{=} \{R \in \wp(Q) \mid R \cap F \neq \varnothing\}$, such that $\mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{A})$ [19].

We denote $\mathcal{A}^\mathsf{D}$ the result of applying the bottom-up determinization to $\mathcal{A}$ and removing unreachable states.

Given a BTA $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$, the *bottom-up co-determinization* builds a co-DBTA $\mathcal{E} \overset{\text{def}}{=} \langle \wp(Q),$ $\Sigma, \delta', F' \rangle$, where $F' \overset{\text{def}}{=} \{F\}$ and δ' is defined as follows. Given $f \in \Sigma \setminus \Sigma_0$ and $R \in \wp(Q)$, $R \in \delta'(f[R_1, \ldots, R_{\langle f \rangle}])$, where $R_i \overset{\text{def}}{=} \{q_i \in Q \mid \exists q \in R, q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_{\langle f \rangle} \in Q \colon q \in \delta(f[q_1, \ldots, q_i, \ldots, q_{\langle f \rangle}])\}$. Moreover, for every $a \in \Sigma_0$ and $R \in \wp(Q)$ such that $\exists q \in R \colon q \in \delta(a[\,])$, we have that $R \in \delta'(a[\,])$. Note that if $\mathcal{L}(\mathcal{A})$ is a path-closed language and $\mathcal{A}$ has no unreachable states then $\mathcal{L}(\mathcal{E}) = \mathcal{L}(\mathcal{A})$ [19]. We denote $\mathcal{A}^\mathsf{cD}$ the result of applying the bottom-up co-determinization to $\mathcal{A}$ and removing empty states.

Example 2.8. Let us give an example of the bottom-up co-determinization by removing the symbol $\wedge$ from Example 2.5. Let $\mathcal{A} = \langle Q, \Sigma_0 \cup \Sigma_2, \delta, F \rangle$ be a BTA with $Q = \{q_0, q_1\}$, $\Sigma_0 = \{\mathbf{T}, \mathbf{F}\}$, $\Sigma_2 = \{\wedge\}$, $\{q_0\} = \delta(\wedge[q_0, q_1]) = \delta(\wedge[q_0, q_0]) = \delta(\wedge[q_1, q_0]) = \delta(\mathbf{F}[\,])$, $\{q_1\} = \delta(\wedge[q_1, q_1]) = \delta(\mathbf{T}[\,])$ and $F = \{q_1\}$.

Note that $\mathcal{L}(\mathcal{A})$ is defined as the set of all trees of the form $t \in \mathcal{T}_{\Sigma_0 \cup \Sigma_2}$ which yield to propositional formulas over the binary connective $\wedge$ and the constants $\mathbf{T}$ and $\mathbf{F}$ that evaluate to $\mathbf{T}$. It is routine to check that $\mathcal{L}(\mathcal{A})$ is path-closed, hence there exists a co-DBTA defining $\mathcal{L}(\mathcal{A})$.

Since $\mathcal{A}$ has no unreachable states, we use the bottom-up co-determinization to build a co-DBTA $\mathcal{E}$ such that $\mathcal{L}(\mathcal{E}) = \mathcal{L}(\mathcal{A})$. We obtain $\mathcal{E} = \langle \wp(Q), \Sigma_0 \cup \Sigma_2, \delta', F' \rangle$ where δ' is defined as $\{q_1\} \in \delta'(\wedge[\{q_1\}, \{q_1\}])$ and $\{q_0\}, \{q_0, q_1\} \in \delta'(\wedge[\{q_0, q_1\}, \{q_0, q_1\}])$. On the other hand, $\{q_1\}, \{q_0, q_1\} \in \delta'(\mathbf{T}[\,])$ and $\{q_0\}, \{q_0, q_1\} \in \delta'(\mathbf{F}[\,])$. Finally, let $F' = \{\{q_1\}\}$.

Note that states $\{q_0\}$ and $\{q_0, q_1\}$ are empty. Thus, we would remove them (and the corresponding entries of the transition function δ') to build $\mathcal{A}^\mathsf{cD}$. $\Diamond$

Remark 2.9. Note that if $\mathcal{L}(\mathcal{A})$ is path-closed but $\mathcal{A}$ has unreachable states then we cannot guarantee that $\mathcal{E}$, the BTA that results from applying the bottom-up co-determinization operation, is such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{E})$.

For instance, consider the BTA $\mathcal{A} = \langle Q, \Sigma_0 \cup \Sigma_2, \delta, F \rangle$ from Example 2.8 and define $\mathcal{A}' \langle Q',$ $\Sigma_0 \cup \Sigma_2', \delta', F' \rangle$ by setting $Q' \overset{\text{def}}{=} Q \cup \{q_2\}$, $\Sigma_2' \overset{\text{def}}{=} \Sigma_2 \cup \{\star\}$ and $F' \overset{\text{def}}{=} F \cup \{q_2\}$. Finally, define δ' as the union of δ and the following entries: $\{q_2\} = \delta'(\star[q_1, q_2]) = \delta'(\star[q_2, q_1]) = \delta'(\star[q_2, q_2])$. Note that q_2 is an unreachable state and thus, $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$. However, $\mathcal{L}(\mathcal{A}') \subset \mathcal{L}(\mathcal{E}')$. In fact, let $\mathcal{E}' = \langle \wp(Q'), \Sigma_0 \cup \Sigma_2', \delta_{\mathcal{E}'}, F_{\mathcal{E}'} \rangle$. Then, the reader may check that, since $\{q_1, q_2\} \in \delta_{\mathcal{E}'}(\mathbf{T}[\,])$, $\{q_1, q_2\} \in \delta_{\mathcal{E}'}(\star[\{q_1, q_2\}, \{q_1, q_2\}])$ and $F_{\mathcal{E}'} = \{q_1, q_2\}$, the tree $\star[\mathbf{T}, \mathbf{T}] \in \mathcal{L}(\mathcal{E}')$, while $\star[\mathbf{T}, \mathbf{T}] \notin \mathcal{L}(\mathcal{A})$.

2.3. Equivalences and congruences

Given a set X, an *equivalence relation* $\sim \,\subseteq X \times X$ induces a *partition* $P_\sim$ of X given by a family $P_\sim = \{B_i\}_{i \in \mathcal{I}}$, with $\mathcal{I} \subseteq \mathbb{N}$, of pairwise disjoint subsets of X (called *blocks* or *equivalence classes*) the union of which is X. The partition $P_\sim$ is said to be *finite* when $\sim$ has *finite index*, i.e., $\sim$ consists of finitely many equivalence classes.

Given $t \in X$, let $P_\sim(t) \overset{\text{def}}{=} \{r \in X \mid t \sim r\}$ be the unique block containing t. This definition can be extended in a natural way to a set $S \subseteq X$ as $P_\sim(S) = \bigcup_{t \in S} P_\sim(t)$. Unless stated otherwise, we consider equivalence relations of finite index.

Given two equivalence relations $\sim_1, \sim_2$, we say that $\sim_1$ is *finer than or equal to* $\sim_2$ when $\sim_1 \subseteq \sim_2$. Sometimes we also say that $\sim_2$ is *coarser than or equal to* $\sim_1$. Observe that $\sim_1 \subseteq \sim_2$ is equivalent to write $\forall t, r \in X: t \sim_1 r \Rightarrow t \sim_2 r$.

Analogously to the notion of left and right congruences on words [16], we introduce *upward* and *downward* congruences on trees and contexts, respectively. Intuitively, upward congruences are equivalences on trees that behave well when substituting trees into contexts, while downward congruences are equivalences on contexts that behave well when substituting pivots for trees.

Definition 2.10. (Upward and Downward congruences)
Given a ranked alphabet Σ, an equivalence $\sim^u$ on $\mathcal{T}_\Sigma$ is an *upward congruence* iff for every $t, r \in \mathcal{T}_\Sigma$ and context $c \in \mathcal{C}_\Sigma$: $t \sim^u r \Rightarrow c[\![t]\!] \sim^u c[\![r]\!]$.
Similarly, an equivalence $\sim^d$ on $\mathcal{C}_\Sigma$ is a *downward congruence* iff for every $x, y, c \in \mathcal{C}_\Sigma$: $x \sim^d y \Rightarrow x[\![c]\!] \sim^d y[\![c]\!]$.

In the sequel, we follow the convention that upward congruences are denoted with a u superscript while downward congruences are denoted with a d superscript. Note that similar definitions of upward [17, 1] and downward [13] congruences have been previously proposed in the literature. In the monadic case, an upward congruence means that two congruent words remain so after prefixing the same word to their left, formally, $u \sim v$ implies $wu \sim wv$ for all w. Similarly, a downward congruence in the monadic case means that two congruent words remain so after appending the same word to their right, formally, $u \sim v$ implies $uw \sim vw$ for all w.

Example 2.11. Given a tree language $L \subseteq \mathcal{T}_\Sigma$, the congruence defined as $t \sim r \overset{\text{def}}{\Leftrightarrow} Lt^{-1} = Lr^{-1}$, for every $t, r \in \mathcal{T}_\Sigma$, is an upward congruence. Let us prove this fact by contradiction.

Suppose that $t \sim r$, for every $t, r \in \mathcal{T}_\Sigma$, but $c[\![t]\!] \not\sim c[\![r]\!]$, for some $c \in \mathcal{C}_\Sigma$. In other words, $Lt^{-1} = \{x \in \mathcal{C}_\Sigma \mid x[\![t]\!] \in L\} = \{x \in \mathcal{C}_\Sigma \mid x[\![r]\!] \in L\} = Lr^{-1}$, but there exists $y \in \mathcal{C}_\Sigma$ such that $y[\![c[\![t]\!]]\!] \in L$ and $y[\![c[\![r]\!]]\!] \notin L$. It follows that, by defining $x = y[\![c]\!]$, we have: $x[\![t]\!] \in L$ and $x[\![r]\!] \notin L$, which contradicts the fact $Lt^{-1} = Lr^{-1}$.

This upward congruence is also known as the *Myhill-Nerode relation for tree languages* [17].

Similarly, the congruence defined as $x \sim y \overset{\text{def}}{\Leftrightarrow} x^{-1}L = y^{-1}L$, for every $x, y \in \mathcal{C}_\Sigma$, is a downward congruence. Lemma 4.2 proves this result. $\diamondsuit$

3. Tree automata constructions from congruences

In this section we present two tree automata constructions that are built upon a given tree language and a congruence. Generally speaking, upward congruences yield deterministic constructions while downward congruences yield co-deterministic ones.

First we introduce the deterministic construction. Given a regular tree language L and an upward congruence $\sim^u$ that preserves the equivalence between upward quotients, i.e., $t \sim^u r \Rightarrow Lt^{-1} = Lr^{-1}$ for every $t, r \in \mathcal{T}_\Sigma$, the following construction yields a DBTA defining exactly L.

Definition 3.1. (Automata construction $\mathsf{H}^{\mathsf{u}}(\sim^{\mathsf{u}}, L)$)
Let $\sim^{\mathsf{u}}$ be an upward congruence and let $L \subseteq \mathcal{T}_\Sigma$. Define the BTA $\mathsf{H}^{\mathsf{u}}(\sim^{\mathsf{u}}, L) \stackrel{\text{def}}{=} \langle Q, \Sigma, \delta, F \rangle$ where
$Q = \{P_{\sim^{\mathsf{u}}}(t) \mid t \in \mathcal{T}_\Sigma\}$, $F = \{P_{\sim^{\mathsf{u}}}(t) \mid t \in L\}$ and the transition function δ is defined as follows:

  (i) for every $a \in \Sigma_0$, $\delta(a[\,]) = \{P_{\sim^{\mathsf{u}}}(a[\,])\}$, and

  (ii) for every $f \in \Sigma_n$, with $n \geq 1$, and trees $t_1, ..., t_{\langle f \rangle}, t \in \mathcal{T}_\Sigma$: $P_{\sim^{\mathsf{u}}}(t) \in \delta(f[P_{\sim^{\mathsf{u}}}(t_1), ..., P_{\sim^{\mathsf{u}}}(t_{\langle f \rangle})])$
     iff $f[P_{\sim^{\mathsf{u}}}(t_1), ..., P_{\sim^{\mathsf{u}}}(t_{\langle f \rangle})] \subseteq P_{\sim^{\mathsf{u}}}(t)$.

Remark 3.2. Note that $\mathsf{H}^{\mathsf{u}}(\sim^{\mathsf{u}}, L)$ is *well-defined* since we assume that $\sim^{\mathsf{u}}$ has finite index and is an upward congruence. Moreover, $\mathsf{H}^{\mathsf{u}}(\sim^{\mathsf{u}}, L)$ is a *deterministic* BTA since for every $f \in \Sigma$ and $t_1, \ldots, t_{\langle f \rangle}, t \in \mathcal{T}_\Sigma$ there exists *exactly one* block $P_{\sim^{\mathsf{u}}}(t) \in Q$ such that $f[P_{\sim^{\mathsf{u}}}(t_1), \ldots, P_{\sim^{\mathsf{u}}}(t_{\langle f \rangle})]$ $\subseteq P_{\sim^{\mathsf{u}}}(t)$. Finally, observe that $\mathsf{H}^{\mathsf{u}}(\sim^{\mathsf{u}}, L)$ might contain empty states but every state is reachable.

Lemma 3.3. Let $L \subseteq \mathcal{T}_\Sigma$ be a tree language and let $\sim^{\mathsf{u}}$ be an upward congruence such that $t \sim^{\mathsf{u}} r \Rightarrow L\, t^{-1} = L\, r^{-1}$ for every $t, r \in \mathcal{T}_\Sigma$. Then $\mathcal{L}(\mathsf{H}^{\mathsf{u}}(\sim^{\mathsf{u}}, L)) = L$.

Proof:
To simplify the notation, we denote $P_{\sim^{\mathsf{u}}}$, the partition induced by $\sim^{\mathsf{u}}$, simply by P. Let $\mathsf{H} = \mathsf{H}^{\mathsf{u}}(\sim^{\mathsf{u}}, L)$. First, we prove that:

$$\mathcal{L}_\downarrow^{\mathsf{H}}(P(r)) = P(r), \text{ for each } r \in \mathcal{T}_\Sigma \ . \tag{1}$$

($\subseteq$). We show that, for all $t \in \mathcal{T}_\Sigma$, $t \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(r)) \Rightarrow t \in P(r)$. We proceed by induction in the height of the tree t.

- *Base case:* Let t be of height 0, i.e. $\exists a \in \Sigma_0$ such that $t = a[\,]$. Then,

$$a[\,] \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(r)) \Leftrightarrow \quad [\text{Def. of } \mathcal{L}_\downarrow^{\mathsf{H}}(P(r))]$$
$$\exists t \in \mathcal{T}_Q\colon a[\,] \to_{\mathsf{H}}^* t \wedge t(\varepsilon) = P(r) \Leftrightarrow$$
$$P(r) \in \delta(a[\,]) \Leftrightarrow \quad [\text{Def. 3.1}]$$
$$a[\,] \in P(r) \ .$$

- *Inductive step:* Now we assume by hypothesis of induction that $t \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(r)) \Rightarrow t \in P(r)$ for all trees of height up to n. Let t be a tree of height $n+1$, i.e. $t = f[t_1, t_2]$ for some $f \in \Sigma$ and $t_1, t_2 \in \mathcal{T}_\Sigma$ height up n. Note that, w.l.o.g., we assume $\langle f \rangle = 2$ for the sake of clarity. Then,

$$t \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(r)) \Leftrightarrow$$
$$\exists t' \in \mathcal{T}_Q\colon t \to_{\mathsf{H}}^* t' \wedge t'(\varepsilon) = P(r) \Leftrightarrow$$
$$\exists r_1, r_2 \in \mathcal{T}_\Sigma\colon t_1 \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(r_1)), t_2 \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(r_2)) \text{ and}$$
$$P(r) \in \delta(f[P(r_1), P(r_2)]) \ ,$$

where the second double-implication holds by Lemma 2.6, since $t = f[t_1, t_2]$.

By Definition 3.1, the fact that $P(r) \in \delta(f[P(r_1), P(r_2)])$ is equivalent to $f[P(r_1), P(r_2)] \subseteq P(r)$. Since $t = f[t_1, t_2]$ and, by I.H., $t_i \in P(r_i)$ with $i \in \{1, 2\}$, we conclude that $t \in f[P(r_1), P(t_2)]$.

($\supseteq$). We show that, for all $t \in \mathcal{T}_\Sigma$, $t \in P(r) \Rightarrow t \in \mathcal{L}_\downarrow^\mathsf{H}(P(r))$. We proceed by induction in the height of the tree t.

- *Base case:* Let t be of height 0, i.e. $\exists a \in \Sigma_0$ such that $t = a[\,]$. Then,

$$
\begin{aligned}
a[\,] \in P(r) \Rightarrow \quad & \text{[Def. 3.1]} \\
P(r) \in \delta(a[\,]) \Rightarrow \quad & \text{[Def. of } \mathcal{L}_\downarrow^\mathsf{H}(P(r))\text{]} \\
a[\,] \in \mathcal{L}_\downarrow^\mathsf{H}(P(r)) \; . &
\end{aligned}
$$

- *Inductive step:* Now we assume by hypothesis of induction that $t \in P(r) \Rightarrow t \in \mathcal{L}_\downarrow^\mathsf{H}(P(r))$ holds for all trees t of height up to n. Let t be a tree of height $n+1$, i.e. $t = f[t_1, t_2]$ for some $f \in \Sigma$ and $t_1, t_2 \in \mathcal{T}_\Sigma$ have height up to n. Note that, w.l.o.g., we assume $\langle f \rangle = 2$. Then,

$$
\begin{aligned}
t \in P(r) \Rightarrow \quad & [t = f[t_1, t_2]] \\
f[t_1, t_2] \in P(r) \Rightarrow \quad & [\sim^\mathsf{u} \text{ is an upward cong.}] \\
f[P(t_1), P(t_2)] \subseteq P(r) \Rightarrow \quad & \text{[Def. 3.1]} \\
P(r) \in \delta(f[P(t_1), P(t_2)]) \Rightarrow \quad & \\
t \in \mathcal{L}_\downarrow^\mathsf{H}(P(r)) \; , &
\end{aligned}
$$

where the last implication holds by Lemma 2.6 and the fact that, by I.H, $t_i \in \mathcal{L}_\downarrow^\mathsf{H}(P(t_i))$, with $i \in \{1, 2\}$.

We conclude this proof by showing that $\mathcal{L}(\mathsf{H}) = L$. To do that, we first prove that $P(L) = L$.

On the one hand, $L \subseteq P(L)$ by reflexivity of $\sim^\mathsf{u}$. Now we show that $P(L) \subseteq L$. Let $t \in \mathcal{T}_\Sigma$ such that $t \in P(L)$. Then, there exists $r \in L$ with $t \sim^\mathsf{u} r$. On the other hand, $t \sim^\mathsf{u} r$ implies $Lt^{-1} = Lr^{-1}$. Then, we conclude that $r \in L \Rightarrow \square \in Lr^{-1} \Rightarrow \square \in Lt^{-1} \Rightarrow t \in L$.

Finally,

$$
\begin{aligned}
\mathcal{L}(\mathsf{H}) = \quad & \text{[Def. of } \mathcal{L}(\mathsf{H})\text{]} \\
\bigcup_{q \in F} \mathcal{L}_\downarrow^\mathsf{H}(q) = \quad & \text{[Def. 3.1]} \\
\bigcup_{t \in L} \mathcal{L}_\downarrow^\mathsf{H}(P(t)) = \quad & \text{[Equation (1)]} \\
\bigcup_{t \in L} P(t) = \quad & [P(L) = L] \\
L \; . &
\end{aligned}
$$

$\square$

Example 3.4. Let us give an example of the DBTA construction of Definition 3.1. Consider $L = \mathcal{L}(\mathcal{A})$ where $\mathcal{A}$ is the BTA defined in Example 2.5. Namely, $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ with $Q = \{q_0, q_1\}$; $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\mathbf{T}, \mathbf{F}\}$ and $\Sigma_2 = \{\wedge, \vee\}$; $\{q_0\} = \delta(\wedge[q_0, q_1]) = \delta(\wedge[q_0, q_0]) = \delta(\wedge[q_1, q_0]) = \delta(\vee[q_0, q_0]) = \delta(\mathbf{F}[\,])$, $\{q_1\} = \delta(\wedge[q_1, q_1]) = \delta(\vee[q_1, q_0]) = \delta(\vee[q_0, q_1]) = \delta(\vee[q_1, q_1]) = \delta(\mathbf{T}[\,])$

and $F = \{q_1\}$. Thus, L is the set of all trees $t \in \mathcal{T}_\Sigma$ that yield to propositional formulas, over the binary connectives $\wedge$ and $\vee$ and the constants $\mathbf{T}$ and $\mathbf{F}$, that evaluate to $\mathbf{T}$.

On the other hand, consider the upward congruence from Example 2.11, i.e., $t \sim^{\mathsf{u}} r \overset{\text{def}}{\Leftrightarrow} Lt^{-1} = Lr^{-1}$, for every $t, r \in \mathcal{T}_\Sigma$. Note that, $\sim^{\mathsf{u}}$ defines two equivalence classes:

- $P_{\sim^{\mathsf{u}}}(\mathbf{T}[\,]) = L$ and

- $P_{\sim^{\mathsf{u}}}(\mathbf{F}[\,]) = L^{\complement}$, where $L^{\complement}$ is the complement of L, i.e., the set $\mathcal{T}_\Sigma \setminus L$.

Specifically, $\{\vee[\mathbf{F},\mathbf{T}], \vee[\mathbf{T},\mathbf{F}], \vee[\mathbf{T},\mathbf{T}], \wedge[\mathbf{T},\mathbf{T}]\} \subseteq P_{\sim^{\mathsf{u}}}(\mathbf{T}[\,])$ and
$\{\wedge[\mathbf{F},\mathbf{T}], \wedge[\mathbf{T},\mathbf{F}], \wedge[\mathbf{F},\mathbf{F}], \vee[\mathbf{F},\mathbf{F}]\} \subseteq P_{\sim^{\mathsf{u}}}(\mathbf{F}[\,])$.

Thus, $\mathsf{H}^{\mathsf{u}}(\sim^{\mathsf{u}}, L) = \langle Q', \Sigma_0 \cup \Sigma_2, \delta', F' \rangle$ where $Q' = \{L, L^{\complement}\}, F' = \{L\}$ and δ' is defined as $\{L\} = \delta'(\mathbf{T}[\,]) = \delta'(\wedge([L, L])) = \delta'(\vee([L^{\complement}, L])) = \delta'(\vee([L, L^{\complement}])) = \delta'(\vee([L, L]))$ and $\{L^{\complement}\} = \delta'(\mathbf{F}[\,]) = \delta'(\wedge([L^{\complement}, L])) = \delta'(\wedge([L, L^{\complement}])) = \delta'(\wedge([L^{\complement}, L^{\complement}])) = \delta'(\vee([L^{\complement}, L^{\complement}]))$.   ◇

Similarly, given a regular tree language L that is path-closed and a downward congruence $\sim^{\mathsf{d}}$, that preserves the equivalence between downward quotients, i.e., $x \sim^{\mathsf{d}} y \Rightarrow x^{-1}L = y^{-1}L$ for every $x, y \in \mathcal{C}_\Sigma$, we give a tree automata construction that yields a BTA that defines exactly L. To this aim, we first recall an alternative characterization of path-closed languages due to Nivat and Podelski [13]. For simplicity, we state the characterization for symbols $f \in \Sigma$ with $\langle f \rangle \leq 2$, although it holds for alphabets of rank greater than 2. Let $L \subseteq \mathcal{T}_\Sigma$ be a regular tree language, then L is *path-closed* iff:

$$\forall x \in \mathcal{C}_\Sigma, \forall f \in \Sigma\colon$$
$$x[\![f[t_1, t_2]]\!], x[\![f[r_1, r_2]]\!] \in L \Rightarrow x[\![f[t_1', t_2']]\!] \in L \text{ where } t_1' \in \{t_1, r_1\}, t_2' \in \{t_2, r_2\} \ . \qquad (2)$$

We recall that the definition of being path-closed is given at the end of Section 2.

Before turning to the construction using downward congruences we introduce a notation. Given $C \subseteq \mathcal{C}_\Sigma$ and $c \in \mathcal{C}_\Sigma$, let $C[\![c]\!] \overset{\text{def}}{=} \{c'[\![c]\!] \mid c' \in C\}$.

Definition 3.5. (Automata construction $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L)$)
Let $\sim^{\mathsf{d}}$ be a downward congruence and let $L \subseteq \mathcal{T}_\Sigma$. Define the BTA $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L) \overset{\text{def}}{=} \langle Q, \Sigma, \delta, F \rangle$ where $Q = \{P_{\sim^{\mathsf{d}}}(x) \mid x \in \mathcal{C}_\Sigma, x^{-1}L \neq \varnothing\}$, $F = \{P_{\sim^{\mathsf{d}}}(\square)\}$ and the transition function δ is defined as follows:

(i) for every $a \in \Sigma_0$, $P_{\sim^{\mathsf{d}}}(x) \in Q\colon P_{\sim^{\mathsf{d}}}(x) \in \delta(a[\,])$ iff $P_{\sim^{\mathsf{d}}}(x)[\![a]\!] \subseteq L$, and

(ii) for every $f \in \Sigma_n$ with $n \geq 1$, $P_{\sim^{\mathsf{d}}}(x), P_{\sim^{\mathsf{d}}}(x_i) \in Q$ with $i = 1..\langle f \rangle\colon P_{\sim^{\mathsf{d}}}(x) \in \delta(f[P_{\sim^{\mathsf{d}}}(x_1), ..., P_{\sim^{\mathsf{d}}}(x_{\langle f \rangle})])$ iff $\exists t_1, ..., t_{\langle f \rangle} \in \mathcal{T}_\Sigma\colon P_{\sim^{\mathsf{d}}}(x)[\![f[t_1, ..., t_{\langle f \rangle}][\square]_i]\!] \subseteq P_{\sim^{\mathsf{d}}}(x_i)$, for every $i = 1..\langle f \rangle$.

Remark 3.6. Note that $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L)$ is *well-defined* since we assume that $\sim^{\mathsf{d}}$ has finite index and is a downward congruence. On the other hand, by definition, $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L)$ does not contain empty states and the final set of states is a singleton. Furthermore, $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L)$ does not contain unreachable states either. This is also a consequence of its definition, since Q only includes those blocks $P_{\sim^{\mathsf{d}}}(x)$ s.t. $x^{-1}L \neq \varnothing$. However, $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L)$ is not necessarily co-deterministic since it might be the case that, for some $f \in$

Σ_n, $P_{\sim^d}(x)$, $P_{\sim^d}(x_i)$, $P_{\sim^d}(y_i) \in Q$ with $i = 1..\langle f \rangle$, there exists $t_1, \ldots, t_{\langle f \rangle}, r_1, \ldots, r_{\langle f \rangle} \in \mathcal{T}_\Sigma$ such that $P_{\sim^d}(x)[\![f[t_1, \ldots, t_{\langle f \rangle}][\Box]_i]\!] \subseteq P_{\sim^d}(x_i)$, $P_{\sim^d}(x)[\![f[r_1, \ldots, r_{\langle f \rangle}][\Box]_i]\!] \subseteq P_{\sim^d}(y_i)$ with $x_i \not\sim^d y_i$.

Now we prove that if L, the language used to construct $\mathsf{H}^d(\sim^d, L)$, is path-closed then $\mathcal{L}(\mathsf{H}^d(\sim^d, L)) = L$. Notice that if L is not path-closed then only $L \subseteq \mathcal{L}(\mathsf{H}^d(\sim^d, L))$ is guaranteed (the reader may check this fact by looking at the proof of the lemma).

Lemma 3.7. Let $L \subseteq \mathcal{T}_\Sigma$ and $\sim^d$ be a downward congruence such that $x \sim^d y \Rightarrow x^{-1}L = y^{-1}L$ for every $x, y \in \mathcal{C}_\Sigma$. Then, $\mathcal{L}(\mathsf{H}^d(\sim^d, L)) \supseteq L$ holds and, if, incidentally, L is path-closed then $\mathcal{L}(\mathsf{H}^d(\sim^d, L)) \subseteq L$.

Proof:
To simplify the notation, we denote $P_{\sim^d}$, the partition induced by $\sim^d$, simply by P. Let $\mathsf{H} = \mathsf{H}^d(\sim^d, L) = (Q, \Sigma, \delta, F)$. First, we prove that for every $P(x) \in Q$:

$$\mathcal{L}_\downarrow^{\mathsf{H}}(P(x)) \subseteq x^{-1}L, \text{if } L \text{ is path-closed, and} \tag{3}$$

$$\mathcal{L}_\downarrow^{\mathsf{H}}(P(x)) \supseteq x^{-1}L \ . \tag{4}$$

($\subseteq$). We show that, if L is path-closed, then for all $t \in \mathcal{T}_\Sigma$, $t \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x)) \Rightarrow t \in x^{-1}L$. We proceed by induction in the height of the tree t.

- *Base case:* Let t be of height 0, i.e. $\exists a \in \Sigma_0$ such that $t = a[]$. Then.

$$
\begin{aligned}
a[] \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x)) \Rightarrow & \quad [\text{Def. of } \mathcal{L}_\downarrow^{\mathsf{H}}(P(x))] \\
P(x) \in \delta(a[]) \Rightarrow & \quad [\text{Def. 3.5}] \\
P(x)[\![a]\!] \subseteq L \Rightarrow & \quad [P(x) \supseteq \{x\}] \\
x[\![a]\!] \in L \Rightarrow & \quad [\text{Def. of downward quotient}] \\
a[] \in x^{-1}L \ . &
\end{aligned}
$$

- *Inductive step:* Now we assume by hypothesis of induction that $t \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x)) \Rightarrow t \in x^{-1}L$ for all trees of height up to n. Let t be a tree of height $n+1$, i.e. $t = f[t_1, t_2]$ for some $f \in \Sigma$ and $t_1, t_2 \in \mathcal{T}_\Sigma$ have height up to $n \geq 0$. Note that, w.l.o.g., we assume $\langle f \rangle = 2$ for the sake of clarity. Then,

$$
\begin{aligned}
t \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x)) \Leftrightarrow & \\
\exists t' \in \mathcal{T}_Q : t'(\varepsilon) = P(x), t \rightarrow_{\mathsf{H}}^* t' \Leftrightarrow & \\
\exists P(x_1), P(x_2) \in Q : t_1 \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x_1)), t_2 \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x_2)) \text{ and} & \\
P(x) \in \delta(f[P(x_1), P(x_2)]), & \tag{5}
\end{aligned}
$$

where the first double-implication holds by definition of $\mathcal{L}_\downarrow^{\mathsf{H}}(P(x))$ and the second one holds by Lemma 2.6, since $t = f[t_1, t_2]$. By Definition 3.5, Equation (5) is equivalent to:

$$
\begin{aligned}
\exists r_1, r_2 \in \mathcal{T}_\Sigma : P(x)[\![f[r_1, r_2][\Box]_1]\!] \subseteq P(x_1), & \\
P(x)[\![f[r_1, r_2][\Box]_2]\!] \subseteq P(x_2) \ . &
\end{aligned}
$$

By definition of P, we have that:

$$\exists r_1, r_2 \in \mathcal{T}_\Sigma \colon x_1 \sim^{\mathsf{d}} x[\![f[r_1, r_2][\![\Box]\!]_1]\!],$$
$$x_2 \sim^{\mathsf{d}} x[\![f[r_1, r_2][\![\Box]\!]_2]\!] \ . \tag{6}$$

Note that, by H.I., $t_1 \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x_1)), t_2 \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x_2)) \Rightarrow t_1 \in x_1^{-1}L, t_2 \in x_2^{-1}L$. Moreover, $x \sim^{\mathsf{d}} y$ implies that $x^{-1}L = y^{-1}L$. Therefore, it follows from Equations (5) and (6) that $\exists r_1, r_2 \in \mathcal{T}_\Sigma \colon t_1 \in x[\![f[r_1, r_2][\![\Box]\!]_1]\!]^{-1}L$ and $t_2 \in x[\![f[r_1, r_2][\![\Box]\!]_2]\!]^{-1}L$, which can be rewritten as $\exists r_1, r_2 \in \mathcal{T}_\Sigma \colon x[\![f[t_1, r_2]]\!] \in L$ and $x[\![f[r_1, t_2]]\!] \in L$. Finally, since L is path-closed and $t = f[t_1, t_2]$, we conclude that $x[\![t]\!] \in L$, i.e., $t \in x^{-1}L$.

($\supseteq$). We next show that, for all $t \in \mathcal{T}_\Sigma, t \in x^{-1}L \Rightarrow t \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x))$. We proceed by induction on the height of t.

- *Base case:* Let t be of height 0, i.e. $\exists a \in \Sigma_0$ such that $t = a[\,]$. Then,

$$\begin{aligned}
a[\,] \in x^{-1}L \Rightarrow \quad & \text{[Def. of downward quotient]} \\
x[\![a]\!] \in L \Rightarrow \quad & [\sim^{\mathsf{d}} \text{ is a downward cong.}] \\
P(x)[\![a]\!] \subseteq L \Rightarrow \quad & \text{[Def. 3.5]} \\
P(x) \in \delta(a[\,]) \Rightarrow \quad & \text{[Def. of } \mathcal{L}_\downarrow^{\mathsf{H}}(P(x))] \\
a[\,] \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x)) \ . \quad &
\end{aligned}$$

- *Inductive step:* Now we assume by hypothesis of induction that $t \in x^{-1}L \Rightarrow t \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x))$ holds for all trees t of height up to n. Let t be a tree of height $n+1$, i.e. $t = f[t_1, t_2]$ for some $f \in \Sigma$ and $t_1, t_2 \in \mathcal{T}_\Sigma$ with height up to n. Note that, w.l.o.g., we assume $\langle f \rangle = 2$ for the sake of clarity. Let $x_1 = x[\![f[\Box, t_2]]\!]$ and $x_2 = x[\![f[t_1, \Box]]\!]$. Since $\sim^{\mathsf{d}}$ is a downward congruence, we have that $P(x)[\![f[\Box, t_2]]\!] \subseteq P(x_1)$ and $P(x)[\![f[t_1, \Box]]\!] \subseteq P(x_2)$. Therefore, by Definition 3.5, $P(x) \in \delta(f[P(x_1), P(x_2)])$.

  On the other hand, note that $t_1 \in x_1^{-1}L$ and $t_2 \in x_2^{-1}L$. By I.H., $t_1 \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x_1))$ and $t_2 \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x_2))$. Relying on Lemma 2.6, we conclude that $t \in \mathcal{L}_\downarrow^{\mathsf{H}}(P(x))$.

We finish by proving the two statements of the Lemma.

$$\begin{aligned}
\mathcal{L}(\mathsf{H}) = \quad & \text{[Def. of } \mathcal{L}(\mathsf{H})] \\
\bigcup_{q \in F} \mathcal{L}_\downarrow^{\mathsf{H}}(q) = \quad & \text{[Def. 3.5]} \\
\mathcal{L}_\downarrow^{\mathsf{H}}(P(\Box)) \supseteq \quad & \text{[Equation (4)]} \\
(\Box)^{-1}L = L \quad &
\end{aligned} \qquad\qquad
\begin{aligned}
\mathcal{L}(\mathsf{H}) = \quad & \text{[Def. of } \mathcal{L}(\mathsf{H})] \\
\bigcup_{q \in F} \mathcal{L}_\downarrow^{\mathsf{H}}(q) = \quad & \text{[Def. 3.5]} \\
\mathcal{L}_\downarrow^{\mathsf{H}}(P(\Box)) \subseteq \quad & \text{[Equation (3), } L \text{ path-closed]} \\
(\Box)^{-1}L = L \ . \quad &
\end{aligned}$$

$\square$

Next we introduce the notion of *strongly downward congruence* which defines an extra condition on a downward congruence $\sim^{\mathsf{d}}$ that guarantees that $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L)$ is a co-DBTA. This condition is an ad-hoc design so as to avoid the issue raised at Remark 3.6. Intuitively, strongly downward means that every congruent pair x and y of contexts remains congruent after plugging the context $t[\![\Box]\!]_i$ in x and the context $r[\![\Box]\!]_i$ in y where t and r are chosen such that $x[\![t]\!], y[\![r]\!] \in L$. Recall that given a tree $t \in \mathcal{T}$, $t[\![\Box]\!]_i \in \mathcal{C}$ denotes the context obtained by replacing the subtree rooted at i with $\Box$.

Definition 3.8. (Strongly downward congruence)
Let $\sim^{\mathsf{d}}$ be a downward congruence and let $L \subseteq \mathcal{T}_\Sigma$. We say $\sim^{\mathsf{d}}$ is *strongly downward* w.r.t. L iff for every $x, y \in \mathcal{C}_\Sigma$ with $x \sim^{\mathsf{d}} y$, $t \in x^{-1}L, r \in y^{-1}L$ and $t(\varepsilon) = r(\varepsilon)$ where $t, r \in \mathcal{T}_\Sigma$, we have that $x[\![t[\![\Box]\!]_i]\!] \sim^{\mathsf{d}} y[\![r[\![\Box]\!]_i]\!], \forall i \in \{1..\langle t(\varepsilon)\rangle\}$.

Finally, we obtain the following lemma.

Lemma 3.9. Let $\sim^{\mathsf{d}}$ be a strongly downward congruence w.r.t. L such that $x \sim^{\mathsf{d}} y \Rightarrow x^{-1}L = y^{-1}L$ for every $x, y \in \mathcal{C}_\Sigma$. Then $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L)$ is a co-DBTA.

Proof:
We prove that $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L) = (Q, \Sigma, \delta, F)$ is a co-DBTA by contradiction.

If $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L)$ is not co-deterministic, then either $|F| > 1$, which is not possible by Definition 3.5; or $\exists f \in \Sigma, \exists x_1, \ldots, x_{\langle f\rangle}, y_1, \ldots, y_{\langle f\rangle}, x \in \mathcal{C}_\Sigma$ such that $P_{\sim^{\mathsf{d}}}(x) \in \delta(f[P_{\sim^{\mathsf{d}}}(x_1), \ldots, P_{\sim^{\mathsf{d}}}(x_{\langle f\rangle})])$ and $P_{\sim^{\mathsf{d}}}(x) \in \delta(f[P_{\sim^{\mathsf{d}}}(y_1), \ldots, P_{\sim^{\mathsf{d}}}(y_{\langle f\rangle})])$ and $\exists i_0 \in 1..\langle f\rangle$ for which $x_{i_0} \not\sim^{\mathsf{d}} y_{i_0}$.

Then, by Definition 3.5, $\exists t_1, \ldots, t_{\langle f\rangle}, r_1, \ldots, r_{\langle f\rangle} \in \mathcal{T}_\Sigma : [\![P_{\sim^{\mathsf{d}}}(x)]\!] f[t_1, \ldots, t_{\langle f\rangle}][\![\Box]\!]_{i_0} \subseteq P_{\sim^{\mathsf{d}}}(x_{i_0})$, and $P_{\sim^{\mathsf{d}}}(x)[\![f[r_1, \ldots, r_{\langle f\rangle}][\![\Box]\!]_{i_0}]\!] \subseteq P_{\sim^{\mathsf{d}}}(y_{i_0})$ with $x_{i_0} \not\sim^{\mathsf{d}} y_{i_0}$. Thus, $x[\![f[t_1, \ldots, t_{\langle f\rangle}][\![\Box]\!]_{i_0}]\!] \sim^{\mathsf{d}} x_{i_0}$ and $x[\![f[r_1, \ldots, r_{\langle f\rangle}][\![\Box]\!]_{i_0}]\!] \sim^{\mathsf{d}} y_{i_0}$.

On the other hand, since $P_{\sim^{\mathsf{d}}}(x_{i_0})$ and $P_{\sim^{\mathsf{d}}}(y_{i_0})$ are states of H^{d}, we have that $x_{i_0}^{-1}L \neq \varnothing$ and $y_{i_0}^{-1}L \neq \varnothing$. Since $x[\![f[t_1, \ldots, t_{\langle f\rangle}][\![\Box]\!]_{i_0}]\!] \sim^{\mathsf{d}} x_{i_0}$ and $x[\![f[r_1, \ldots, r_{\langle f\rangle}][\![\Box]\!]_{i_0}]\!] \sim^{\mathsf{d}} y_{i_0}$, we also have that $x[\![f[t_1, \ldots, t_{\langle f\rangle}][\![\Box]\!]_{i_0}]\!]^{-1}L \neq \varnothing$ and $x[\![f[r_1, \ldots, r_{\langle f\rangle}][\![\Box]\!]_{i_0}]\!]^{-1}L \neq \varnothing$. Thus, for some $t_{i_0}, r_{i_0} \in \mathcal{T}_\Sigma$, $t = f[t_1, \ldots, t_{i_0}, \ldots, t_{\langle f\rangle}] \in x^{-1}L$ and $r = f[r_1, \ldots, r_{i_0}, \ldots, r_{\langle f\rangle}] \in x^{-1}L$.

Finally, since $\sim^{\mathsf{d}}$ is a strongly downward congruence w.r.t. L, $x \sim^{\mathsf{d}} x$ (trivially), $t, r \in x^{-1}L$ and $t(\varepsilon) = r(\varepsilon)$, we have that $x[\![f[t_1, \ldots, t_{\langle f\rangle}][\![\Box]\!]_{i_0}]\!] \sim^{\mathsf{d}} x[\![f[r_1, \ldots, r_{\langle f\rangle}][\![\Box]\!]_{i_0}]\!]$. By transitivity of $\sim^{\mathsf{d}}$, we have that $x_{i_0} \sim^{\mathsf{d}} y_{i_0}$, which is a contradiction.

Thus, we conclude that $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L)$ is co-deterministic. □

Example 3.10. Now we give an example of the construction of Definition 3.5. Consider the BTA $\mathcal{A}$ defined in Example 2.8, i.e., $\mathcal{A} = \langle Q, \Sigma, \delta, F\rangle$ with $Q = \{q_0, q_1\}$; $\Sigma = \Sigma_0 \cup \Sigma_2$ with $\Sigma_0 = \{\mathbf{T}, \mathbf{F}\}$ and $\Sigma_2 = \{\wedge\}$; $\{q_0\} = \delta(\wedge[q_0, q_1]) = \delta(\wedge[q_0, q_0]) = \delta(\wedge[q_1, q_0]) = \delta(\mathbf{F}[])$, $\{q_1\} = \delta(\wedge[q_1, q_1]) = \delta(\mathbf{T}[])$ and $F = \{q_1\}$. Recall that $L = \mathcal{L}(\mathcal{A})$ is the set of all trees $t \in \mathcal{T}_\Sigma$ that yield to propositional formulas, over the binary connective $\wedge$ and the constants $\mathbf{T}$ and $\mathbf{F}$, that evaluate to $\mathbf{T}$. Hence, $\mathcal{L}(\mathcal{A})$ is path-closed.

On the other hand, consider the downward congruence from Example 2.11, i.e., $x \sim^{\mathsf{d}} y \overset{\text{def}}{\Leftrightarrow}$ $x^{-1}L = y^{-1}L$, for every $x, y \in \mathcal{C}_\Sigma$. As we shall see in Lemma 4.2, since L is path-closed, $\sim^{\mathsf{d}}$ is a strongly downward congruence w.r.t. L. It turns out that $\sim^{\mathsf{d}}$ defines two equivalence classes:

- $P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{T}, \square])$, with $(\wedge[\mathbf{T}, \square])^{-1}L = L$, and
- $P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{F}, \square])$, with $(\wedge[\mathbf{F}, \square])^{-1}L = \varnothing$.

Intuitively, $P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{T}, \square])$ is composed of those contexts x such that $x[\![\mathbf{T}[\,]]\!]$ corresponds to a propositional formula that evaluates to $\mathbf{T}$. For instance, $\{\wedge[\square, \mathbf{T}], \square[\,]\} \subseteq P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{T}, \square])$. On the other hand, $P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{F}, \square])$ is composed of those such that $x[\![\mathbf{T}[\,]]\!]$ is a formula that evaluates to $\mathbf{F}$. For example, $\wedge[\square, \mathbf{F}] \in P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{F}, \square])$.

Thus, $\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{u}}, L) = \langle Q', \Sigma, \delta', F' \rangle$ with $Q' = \{P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{T}, \square])\}$ and $F' = \{P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{T}, \square])\}$ since $P_{\sim^{\mathsf{d}}}(\square) = P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{T}, \square])$. Note that $P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{F}, \square]) \notin Q'$ since $(\wedge[\mathbf{F}, \square])^{-1}L = \varnothing$. In fact, including $P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{F}, \square])$ in Q' would result in an unreachable state since:

(i) for every $a \in \{\mathbf{F}, \mathbf{T}\}$, $P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{F}, \square])[\![a]\!] \cap L = \varnothing$, and

(ii) (trivially) $\nexists t \in (\wedge[\mathbf{F}, \square])^{-1}L : t(\varepsilon) = \wedge$ and $P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{F}, \square])[\![t[\![\square]\!]_i]\!] \subseteq P_{\sim^{\mathsf{d}}}(x_i)$, with $x_i \in \mathcal{C}_\Sigma$ and $i \in \{1, 2\}$.

Finally, $P_{\sim^{\mathsf{d}}}(\wedge[1, \square]) \in \delta'(\wedge[P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{T}, \square]), P_{\sim^{\mathsf{d}}}(\wedge[\mathbf{T}, \square])])$ (set $t = \wedge[\mathbf{T}, \mathbf{T}]$ in Definition 3.5). $\Diamond$

It is worth noting that the DBTA H^{u} built in Example 3.4 and the co-DBTA H^{d} from Example 3.10 are indeed the *minimal* DBTA and co-DBTA for their corresponding languages, respectively. This is due to the fact that we used the Myhill-Nerode relation for tree languages (and its counterpart downward congruence). We will recall the main properties of these congruences and introduce their approximation using BTAs in the following section.

4. Language-based equivalences and their approximation using BTAs

In this section, we instantiate the automata constructions from the previous section using two classes of congruences: *language-based* congruences, whose definition relies on a regular tree language; and *BTA-based* congruences, whose definition relies on a BTA.

Definition 4.1. (Language-based equivalences)
Let $L \subseteq \mathcal{T}_\Sigma$ be a tree language and let $t, r \in \mathcal{T}_\Sigma$ and $x, y \in \mathcal{C}_\Sigma$. The *upward and downward language-based equivalences* are, respectively:

$$t \sim^{\mathsf{u}}_L r \overset{\text{def}}{\Leftrightarrow} L\,t^{-1} = L\,r^{-1}$$
$$x \sim^{\mathsf{d}}_L y \overset{\text{def}}{\Leftrightarrow} x^{-1}L = y^{-1}L \ .$$

These equivalence relations are always congruences. The upward language-based equivalence is also known as the Myhill-Nerode relation for tree languages. As shown by Kozen [17], given a tree language $L \subseteq \mathcal{T}_\Sigma$, the Myhill-Nerode relation is an upward congruence of finite index iff L is regular.

Note that this congruence is the coarsest upward congruence enabling Lemma 3.3 (set $\sim^{\mathsf{u}}$ to be $\sim^{\mathsf{u}}_L$). Similarly, Nivat and Podelski [13] showed that the downward language-based equivalence has finite index iff L is regular and it is the coarsest downward congruence that enables Lemma 3.7. Next we prove that $\sim^{\mathsf{d}}_L$ is a strongly downward congruence w.r.t. L when L is path-closed, hence it is the coarsest downward congruence enabling Lemma 3.9 (set $\sim^{\mathsf{d}}$ to $\sim^{\mathsf{d}}_L$).

Lemma 4.2. Let $L \subseteq \mathcal{T}_\Sigma$ be a path-closed language. Then, $\sim^{\mathsf{d}}_L$ is a strongly downward congruence w.r.t. L.

Proof:
First, we show that $\sim^{\mathsf{d}}_L$ is a downward congruence by contradiction. Let $x, y, c \in \mathcal{C}_\Sigma$ be such that $x \sim^{\mathsf{d}}_L y$ and $x[\![c]\!] \not\sim^{\mathsf{d}}_L y[\![c]\!]$. Then, w.l.o.g, $\exists t \in \mathcal{T}_\Sigma$ such that $x[\![c[\![t]\!]]\!] \in L$ and $y[\![c[\![t]\!]]\!] \notin L$, hence $c[\![t]\!] \in x^{-1}L$ while $c[\![t]\!] \notin y^{-1}L$, which contradicts the fact that $x \sim^{\mathsf{d}}_L y$.

Now we show that $\sim^{\mathsf{d}}_L$ is strongly downward w.r.t. L. Consider $x, y \in \mathcal{C}_\Sigma$ s.t. $x \sim^{\mathsf{d}}_L y$ and $t, r \in \mathcal{T}_\Sigma$ with $t \in x^{-1}L$, $r \in y^{-1}L$ and $t(\varepsilon) = r(\varepsilon)$. Assume that $\sim^{\mathsf{d}}_L$ is not a strongly downward congruence, i.e. there exists $i_0 \in 1..\langle f \rangle$ such that $x[\![t[\![\square]\!]_{i_0}]\!] \not\sim^{\mathsf{d}}_L y[\![r[\![\square]\!]_{i_0}]\!]$. Then, there exists $s \in \mathcal{T}_\Sigma$ such that $x[\![t[\![s]\!]_{i_0}]\!] \in L$, while $y[\![r[\![s]\!]_{i_0}]\!] \notin L$.

On the one hand, since $r \in y^{-1}L$, there exists $s' \in \mathcal{T}_\Sigma$ such that $y[\![r[\![s']\!]_{i_0}]\!] \in L$. Since $x \sim^{\mathsf{d}}_L y$, i.e., $x^{-1}L = y^{-1}L$, we have that $x[\![r[\![s']\!]_{i_0}]\!] \in L$. Therefore, since $x[\![t[\![s]\!]_{i_0}]\!] \in L$, $x[\![r[\![s']\!]_{i_0}]\!] \in L$ and L is path-closed, by definition (see (2) in Section 3), we have that $x[\![r[\![s]\!]_{i_0}]\!] \in L$. Since $x \sim^{\mathsf{d}}_L y$, it follows that $y[\![r[\![s]\!]_{i_0}]\!] \in L$ as well, which is a contradiction.

We conclude that $\sim^{\mathsf{d}}_L$ is a strongly downward congruence w.r.t. L. $\qquad\square$

Now, we propose congruences based on the states of a given BTA. These BTA-based congruences are finer than (or equal to) the corresponding language-based ones and are thus said to *approximate* the language-based congruences.

In order to define the downward BTA-based congruences, we first introduce the notion of *root-to-pivot equivalence* between two contexts w.r.t. a BTA $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ and a subset of states $S \subseteq Q$. Intuitively, two contexts $x, y \in \mathcal{C}_\Sigma$ are root-to-pivot equivalent w.r.t. S iff

(i) their pivots (the nodes with the label $\square$) coincide and so do the paths from their roots to their pivots, and

(ii) either x and y belong to the upward language of some state (w.r.t. S), or none of them does.

Definition 4.3. (Root-to-pivot equivalence)
Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA and $S \subseteq Q$. We say that $x, y \in \mathcal{C}_\Sigma$ are *root-to-pivot equivalent* w.r.t. S, denoted by $x \sim^S_{\wr} y$, iff

(i) $\mathrm{piv}(x) = \mathrm{piv}(y)$ and $\forall p, p' \in (\mathbb{N}_+)^*$ if $p{\cdot}p' = \mathrm{piv}(x)$ then $x(p) = y(p)$, and

(ii) $x \in \mathcal{L}_\uparrow(q, S)$ for some $q \in Q$ if and only if $y \in \mathcal{L}_\uparrow(q', S)$ for some $q' \in Q$.

The root-to-pivot equivalence need not have finite index.

We will simply write $x \sim_{\wr} y$ when $S = F$. Let us fix intuitions with the following example.

Example 4.4. Consider the BTA $\mathcal{A}$ given in Example 2.5, i.e., $\mathcal{A} = \langle Q, \Sigma_0 \cup \Sigma_2, \delta, F \rangle$ with $Q = \{q_0, q_1\}$, $\Sigma_0 = \{\mathbf{T}, \mathbf{F}\}$, $\Sigma_2 = \{\wedge, \vee\}$, $\{q_0\} = \delta(\wedge[q_0, q_1]) = \delta(\wedge[q_0, q_0]) = \delta(\wedge[q_1, q_0]) = \delta(\vee[q_0, q_0]) = \delta(\mathbf{F}[\,])$, $\{q_1\} = \delta(\wedge[q_1, q_1]) = \delta(\vee[q_1, q_0]) = \delta(\vee[q_0, q_1]) = \delta(\vee[q_1, q_1]) = \delta(\mathbf{T}[\,])$ and $F = \{q_1\}$. Thus, $\mathcal{L}(\mathcal{A})$ is defined as the set of all trees of the form $t \in \mathcal{T}_{\Sigma_0 \cup \Sigma_2}$ which yield to propositional formulas, over the binary connectives $\wedge$ and $\vee$ and the constants $\mathbf{T}$ and $\mathbf{F}$, that evaluate to $\mathbf{T}$.

Let $S = F$, then we have that $x = \vee[\vee[\mathbf{T}, \mathbf{F}], \square]$ and $y = \vee[\vee[\mathbf{T}, \mathbf{T}], \square]]$ are root-to-pivot equivalent w.r.t. S, i.e., $x \sim_\wr y$. In fact, $\mathrm{piv}(x) = \mathrm{piv}(y) = 2$ and for all $p, p' \in (\mathbb{N}_+)^*$ if $p \cdot p' = \mathrm{piv}(x)$ then $x(p) = y(p)$. Moreover, $q_1 \in Q$ is s.t. $x \in \mathcal{L}_\uparrow(q_1, S)$ and $y \in \mathcal{L}_\uparrow(q_1, S)$.

On the other hand, we have that $x' = \wedge[\wedge[\mathbf{T}, \mathbf{T}], \square]$ and $y' = \wedge[\wedge[\mathbf{T}, \mathbf{F}], \square]$ are *not* root-to-pivot equivalent w.r.t. S. Note that, despite of having that $\mathrm{piv}(x') = \mathrm{piv}(y') = 2$ and for all $p, p' \in (\mathbb{N}_+)^*$ if $p \cdot p' = \mathrm{piv}(x)$ then $x(p) = y(p)$, $\nexists q \in Q \colon y' \in \mathcal{L}_\uparrow(q, S)$, while $x' \in \mathcal{L}_\uparrow(q_1, S)$. In fact, $(y')^{-1}L = \varnothing$ while $(x')^{-1}L \neq \varnothing$. Intuitively, in the absence of unreachable states, the second condition in the definition of root-to-pivot equivalence prevents from defining two equivalent contexts when one of them defines an empty downward quotient and the other does not. $\diamond$

Now we define the $\mathrm{post}(\cdot)$ and $\mathrm{pre}(\cdot)$ operators for trees. Given a BTA $\mathcal{A}$, a tree $t \in \mathcal{T}_\Sigma$ and a set $S \subseteq Q$, $\mathrm{post}_t^{\mathcal{A}}(S)$ is the set of states appearing at the root after $\mathcal{A}$ is done processing the tree t provided that this processing labelled the leaves of t with states in S. On the other hand, given a context $x \in \mathcal{C}_\Sigma$ and $S \subseteq Q$, $\mathrm{pre}_x^{\mathcal{A}}(S)$ contains the states q such that the result of $\mathcal{A}$ processing $x[\![q]\!]$ has its root labelled with a state in S. Furthermore, $\mathrm{pre}_x^{\mathcal{A}}(S)$ contains the states q obtained as above this time starting from $y[\![q]\!]$ where $y \in \mathcal{C}_\Sigma$ is root-to-pivot equivalent to x.

Definition 4.5. (Post and pre operators)
Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA and let $t \in \mathcal{T}_\Sigma$, $x \in \mathcal{C}_\Sigma$ and $S \subseteq Q$. Define:

$$\mathrm{post}_t^{\mathcal{A}}(S) \stackrel{\mathrm{def}}{=} \{q \in Q \mid t \in \mathcal{L}_\downarrow^{\mathcal{A}}(q, S)\}$$

$$\mathrm{pre}_x^{\mathcal{A}}(S) \stackrel{\mathrm{def}}{=} \{q \in Q \mid x \in P_{\sim_\wr}^S(\mathcal{L}_\uparrow^{\mathcal{A}}(q, S))\} \ .$$

As usual, we will omit the superscript $\mathcal{A}$ when it is clear from the context. Note that, for every $x \in \mathcal{C}_\Sigma, S \subseteq Q \colon \mathrm{pre}_x(S) = \varnothing \Leftrightarrow \nexists q \in Q \colon x \in \mathcal{L}_\uparrow(q, S)$.

Let us illustrate these definitions with an example.

Example 4.6. As in the previous example, consider the BTA $\mathcal{A}$ from Example 2.5. Given the tree $t = \wedge[\vee[\mathbf{T}, \mathbf{F}], \wedge[\mathbf{T}, \mathbf{T}]]$, we have that $\mathrm{post}_t(\{q_0, q_1\}) = \{q_1\}$, while $\mathrm{post}_t(\{q_0\}) = \varnothing$.

On the other hand, given the context $x = \vee[\wedge[\mathbf{T}, \mathbf{F}], \square]$, we have that $x \in P_{\sim_\wr}(\mathcal{L}_\uparrow(q_1))$ since $x \in \mathcal{L}_\uparrow(q_1)$. Thus, $q_1 \in \mathrm{pre}_x(F)$. Moreover, $x \in P_{\sim_\wr}(\mathcal{L}_\uparrow(q_0))$ since $y = \vee[\wedge[\mathbf{T}, \mathbf{T}], \square]$ is s.t. $y \sim_\wr x$ and $y \in \mathcal{L}_\uparrow(q_0)$. Hence, $\mathrm{pre}_x(F) = \{q_0, q_1\}$.

Finally, given the context $x' = \wedge[\vee[\mathbf{F}, \mathbf{F}], \square]$, $\mathrm{pre}_{x'}(F) = \varnothing$ since $\nexists q \in Q \colon x' \in \mathcal{L}_\uparrow(q)$. $\diamond$

Note that the notion of root-to-pivot equivalence in the definition of $\mathrm{pre}_x^{\mathcal{A}}(S)$ allows us to prove that the downward BTA-based equivalence (see Definition 4.12) is

  (i) strongly downward w.r.t. $\mathcal{L}(\mathcal{A})$ (see Lemma 4.13(b)) and

(ii) mimics the construction of each set R_i, from the definition of bottom-up co-determinization given in Section 2, i.e., we produce the same set of states as that of $\mathcal{A}^{\mathrm{cD}}$ by constructing the sets $\mathrm{pre}_x^{\mathcal{A}}(F)$, with $x \in \mathcal{C}_\Sigma$.

The reader is referred to Example 4.14 for fixing these intuitions.

Our next step is to establish basic properties of the $\mathrm{post}(\cdot)$ and $\mathrm{pre}(\cdot)$ operator, the upward and downward languages they induce and their relationships with upward and downward quotients. Before to establish those properties, we need three technical lemmas.

Lemma 4.7. Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA without unreachable states and such that $L = \mathcal{L}(\mathcal{A})$ is a path-closed language. Then, $x \sim_\wr y \Rightarrow x^{-1}L = y^{-1}L$, for every $x, y \in \mathcal{C}_\Sigma$.

Proof:
Recall that $\sim_\wr$ shortly denotes $\sim_\wr^F$. By definition, $x \sim_\wr y$ iff:

(i) $\mathrm{piv}(x) = \mathrm{piv}(y)$ and $\forall p, p' \in (\mathbb{N}_+)^*$ if $p \cdot p' = \mathrm{piv}(x)$ then $x(p) = y(p)$, and

(ii) $\exists q \in Q \colon x \in \mathcal{L}_\uparrow(q) \Leftrightarrow \exists q' \in Q \colon y \in \mathcal{L}_\uparrow(q')$.

Note that if $x \sim_\wr y$ then the $\square$-height of x and y coincide. The proof goes by induction on the $\square$-height of x and y.

- *Base case:* Let $\mathrm{h}_\square(x) = \mathrm{h}_\square(y) = 1$. Then $x = y = \square[\,]$ and $x^{-1}L = y^{-1}L = L$.

- *Inductive step:* Assume that the hypothesis holds for all contexts up to $\square$-height n. Let $x, y \in \mathcal{C}_\Sigma$ be such that $x \sim_\wr y$ and $\mathrm{h}_\square(x) = \mathrm{h}_\square(y) = n+1$.

  First, assume that $\nexists q \in Q \colon x \in \mathcal{L}_\uparrow(q)$ and $\nexists q' \in Q \colon y \in \mathcal{L}_\uparrow(q')$. Then, since $\mathcal{A}$ has no unreachable states we have that $x^{-1}L = y^{-1}L = \varnothing$.

  On the other hand, assume that $\exists q \in Q \colon x \in \mathcal{L}_\uparrow(q)$ and $\exists q' \in Q \colon y \in \mathcal{L}_\uparrow(q')$. Since $\mathcal{A}$ has no unreachable states, we have that $\exists t, r \in \mathcal{T}_\Sigma \colon x[\![t]\!], y[\![r]\!] \in L$.

  It is easy to check that $\exists x', x'', y', y'' \in \mathcal{C}_\Sigma \colon \mathrm{h}_\square(x') = \mathrm{h}_\square(y') = n-1$, $\mathrm{h}_\square(x'') = \mathrm{h}_\square(y'') = 2$, $x'[\![x'']\!] = x$ and $y'[\![y'']\!] = y$. On the other hand, since $\exists q \in Q \colon x \in \mathcal{L}_\uparrow(q)$ and $\exists q' \in Q \colon y \in \mathcal{L}_\uparrow(q')$ then $\exists \tilde{q} \in Q \colon x' \in \mathcal{L}_\uparrow(\tilde{q})$ and $\exists \tilde{q}' \in Q \colon y' \in \mathcal{L}_\uparrow(\tilde{q}')$. Thus, we have that $x' \sim_\wr y'$.

  W.l.o.g. and for the sake of clarity, let us consider $f \in \Sigma$ with $\langle f \rangle = 2$ such that $x'' = f[\square, t_2]$ and $y'' = f[\square, r_2]$ with $t_2, r_2 \in \mathcal{T}_\Sigma$. Then:

$$x[\![t]\!] \in L, y[\![r]\!] \in L \Leftrightarrow$$
$$x'[\![x''[\![t]\!]]\!] \in L, y'[\![y''[\![r]\!]]\!] \in L \Rightarrow^{\dagger}$$
$$y'[\![x''[\![t]\!]]\!] \in L, y'[\![y''[\![r]\!]]\!] \in L \Leftrightarrow$$
$$y'[\![f[t, t_2]]\!] \in L, y'[\![f[r, r_2]]\!] \in L \Rightarrow^{\dagger\dagger}$$
$$y'[\![f[t, r_2]]\!] \in L \Leftrightarrow$$
$$y'[\![y''[\![t]\!]]\!] \in L \Leftrightarrow$$
$$y[\![t]\!] \in L \ .$$

Note that implication † holds since, by H.I., $(y')^{-1}L = (x')^{-1}L$. On the other hand, implication †† holds since L is path-closed.

We conclude that $x^{-1}L \subseteq y^{-1}L$. The proof for the reverse inclusion is symmetric. Therefore, $x^{-1}L = y^{-1}L$.   □

Lemma 4.8. Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA without unreachable states and such that $L = \mathcal{L}(\mathcal{A})$ is path-closed then, for every $x, y \in \mathcal{C}_\Sigma$:

$$\exists q \in Q \colon x[\![y]\!] \in \mathcal{L}_\uparrow(q) \text{ iff } \exists q_1 \in Q \colon y \in \mathcal{L}_\uparrow(q_1, \mathrm{pre}_x(F)) \ .$$

Proof:
Recall that $\mathcal{L}_\uparrow(q)$ denotes $\mathcal{L}_\uparrow(q, F)$.

($\Rightarrow$). We assume that $\exists q \in Q \colon x[\![y]\!] \in \mathcal{L}_\uparrow(q)$, i.e., $\exists q \in Q, \exists t, s \in \mathcal{T}_Q \colon y[\![q]\!] \to_{\mathcal{A}}^* t$ and $x[\![t(\varepsilon)]\!] \to_{\mathcal{A}}^* s, s(\varepsilon) \in F$. By setting $q_1 = q$, we have that $y[\![q_1]\!] \to_{\mathcal{A}}^* t$, where $t(\varepsilon) \in \mathrm{pre}_x(F)$, since $\exists z \in \mathcal{C}_\Sigma \colon z \sim_\wr^F x$ s.t. $z \in \mathcal{L}_\uparrow(t(\varepsilon))$ by setting $z = x$. Thus, $\exists q_1 \in Q \colon y \in \mathcal{L}_\uparrow(q_1, \mathrm{pre}_x(F))$.

($\Leftarrow$). Assume now that $\exists q_1 \in Q \colon y \in \mathcal{L}_\uparrow(q_1, \mathrm{pre}_x(F))$, i.e., $\exists t \in \mathcal{T}_Q \colon y[\![q_1]\!] \to_{\mathcal{A}}^* t, t(\varepsilon) \in \mathrm{pre}_x(F)$. We will show that $\exists q \in Q \colon y[\![q]\!] \to_{\mathcal{A}}^* t'$ and $x[\![t'(\varepsilon)]\!] \to_{\mathcal{A}}^* s', s'(\varepsilon) \in F$.

By hypothesis, $t(\varepsilon) \in \mathrm{pre}_x(F)$. In the case that $x \in \mathcal{L}_\uparrow(t(\varepsilon))$, then indeed, $\exists q \in Q \colon y[\![q]\!] \to_{\mathcal{A}}^* t'$ and $x[\![t'(\varepsilon)]\!] \to_{\mathcal{A}}^* s', s'(\varepsilon) \in F$, by setting $q = q_1$ and $t'(\varepsilon) = t(\varepsilon)$. Now, let us assume that $t(\varepsilon) \in \mathrm{pre}_x(F)$ but $x \notin \mathcal{L}_\uparrow(t(\varepsilon))$. By definition of $t(\varepsilon) \in \mathrm{pre}_x(F)$, we have that $\exists \tilde{x} \in \mathcal{C}_\Sigma \colon \tilde{x} \sim_\wr^F x$ s.t. $\tilde{x} \in \mathcal{L}_\uparrow(t(\varepsilon))$, where $\tilde{x} \neq x$. Since $\mathcal{A}$ has no unreachable states, $y[\![q_1]\!] \to_{\mathcal{A}}^* t$ and $\tilde{x} \in \mathcal{L}_\uparrow(t(\varepsilon))$, we have that $\exists \tilde{t} \in \mathcal{T}_\Sigma \colon y[\![\tilde{t}]\!] \in \tilde{x}^{-1}L$. Since $\tilde{x} \sim_\wr^F x$ and L is path-closed, by Lemma 4.7, we have that $\tilde{x}^{-1}L = x^{-1}L$. Thus, $y[\![\tilde{t}]\!] \in x^{-1}L$. We conclude that, since $\mathcal{A}$ has no unreachable states, $\exists q \in Q \colon x[\![y]\!] \in \mathcal{L}_\uparrow(q)$.   □

Lemma 4.9. Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA without unreachable states and such that $\mathcal{L}(\mathcal{A})$ is path-closed. Then, for every $x_1, x_2, y_1, y_2 \in \mathcal{C}_\Sigma$ with $\mathrm{h}_\square(x_1) = \mathrm{h}_\square(x_2)$, $x_1[\![y_1]\!] \in \mathcal{L}_\uparrow(q)$ and $x_2[\![y_2]\!] \in \mathcal{L}_\uparrow(q')$ for some $q, q' \in Q$:

$$x_1[\![y_1]\!] \sim_\wr x_2[\![y_2]\!] \Leftrightarrow x_1 \sim_\wr x_2 \text{ and } y_1 \sim_\wr^{\mathrm{pre}_{x_1}(F)} y_2 \ .$$

Proof:
Recall that x, y are *root-to-pivot equivalent* w.r.t. $S \subseteq Q$, denoted by $x \sim_\wr^S y$, iff:

(i) $\mathrm{piv}(x) = \mathrm{piv}(y)$ and $\forall p, p' \in (\mathbb{N}_+)^*$ if $p \cdot p' = \mathrm{piv}(x)$ then $x(p) = y(p)$, and

(ii) $\exists q \in Q \colon x \in \mathcal{L}_\uparrow(q, S) \Leftrightarrow \exists q' \in Q \colon y \in \mathcal{L}_\uparrow(q', S)$.

The reader can easily check that the double implication in the statement: $x_1[\![y_1]\!] \sim_\wr x_2[\![y_2]\!] \Leftrightarrow x_1 \sim_\wr$ x_2 and $y_1 \sim_\wr^{\mathrm{pre}_{x_1}(F)} y_2$ holds w.r.t. condition (i). In particular, the left-to-right implication holds as $\mathrm{h}_\square(x_1) = \mathrm{h}_\square(x_2)$.

We now focus on proving that the double implication holds w.r.t. condition (ii) in the above definition. Recall that $\mathcal{L}_\uparrow(q)$ denotes $\mathcal{L}_\uparrow(q, F)$.

($\Rightarrow$). By hypothesis, $\exists q \in Q\colon x_1[\![y_1]\!] \in \mathcal{L}_\uparrow(q)$ and $\exists q' \in Q\colon x_2[\![y_2]\!] \in \mathcal{L}_\uparrow(q')$. Then, clearly, $\exists q_1 \in Q\colon x_1 \in \mathcal{L}_\uparrow(q_1)$ and $\exists q_2 \in Q\colon x_2 \in \mathcal{L}_\uparrow(q_2)$. Therefore, $x_1 \sim_\lessdot x_2$. Also, by hypothesis and Lemma 4.8, $\exists q_1' \in Q\colon y_1 \in \mathcal{L}_\uparrow(q_1', \mathrm{pre}_{x_1}(F))$ and $\exists q_2' \in Q\colon y_2 \in \mathcal{L}_\uparrow(q_2', \mathrm{pre}_{x_2}(F))$. Since $x_1 \sim_\lessdot x_2$ implies that $\mathrm{pre}_{x_1}(F) = \mathrm{pre}_{x_2}(F)$, we have that $y_1 \sim_\lessdot^{\mathrm{pre}_{x_1}(F)} y_2$.

($\Leftarrow$). Finally we show the right-to-left implication:

$$\exists q \in Q\colon x_1[\![y_1]\!] \in \mathcal{L}_\uparrow(q) \Leftrightarrow$$
$$\exists q_1 \in Q\colon y_1 \in \mathcal{L}_\uparrow(q_1, \mathrm{pre}_{x_1}(F)) \Leftrightarrow$$
$$\exists q_2 \in Q\colon y_2 \in \mathcal{L}_\uparrow(q_2, \mathrm{pre}_{x_2}(F)) \Leftrightarrow$$
$$\exists q' \in Q\colon x_2[\![y_2]\!] \in \mathcal{L}_\uparrow(q') \ .$$

Note that the first and last double-implications hold by Lemma 4.8, while the second one holds by hypothesis. Finally, we conclude that $x_1[\![y_1]\!] \sim_\lessdot x_2[\![y_2]\!]$. $\qquad\square$

We now can turn back to the lemmas on the $\mathrm{post}(\cdot)$ and $\mathrm{pre}(\cdot)$ operators, the upward and downward languages they induce and their relationship to upward and downward quotient.

Lemma 4.10. Let $\mathcal{A} = (Q, \Sigma, \delta, F)$ be a BTA with $L = \mathcal{L}(\mathcal{A})$ and let $t \in \mathcal{T}_\Sigma$ and $x \in \mathcal{C}_\Sigma$. Then the following hold:

(a) $\bigcup_{q \in \mathrm{post}_t(\mathrm{i}(\mathcal{A}))} \mathcal{L}_\uparrow(q) = L\, t^{-1}$, and

(b) If $\mathcal{A}$ has no unreachable states and L is path-closed, then $\bigcup_{q \in \mathrm{pre}_x(F)} \mathcal{L}_\downarrow(q) = x^{-1}L$.

Proof:
We start by recalling that $\mathrm{i}(\mathcal{A}) = \{q \in Q \mid \exists a \in \Sigma_0\colon q \in \delta(a[\,])\}$ as defined in Section 2.2.

(a) $\bigcup_{q \in \mathrm{post}_t(\mathrm{i}(\mathcal{A}))} \mathcal{L}_\uparrow(q) = L t^{-1}$.

Recall that $\mathcal{L}_\uparrow(q)$ simply denotes $\mathcal{L}_\uparrow(q, F)$. To simplify further the notation, denote $\mathrm{i}(\mathcal{A})$ as I.

$$L\, t^{-1} =$$
$$\{x \in \mathcal{C}_\Sigma \mid x[\![t]\!] \in L\} =$$
$$\{x \in \mathcal{C}_\Sigma \mid \exists t' \in \mathcal{T}_Q\colon x[\![t]\!] \to_\mathcal{A}^* t',\ t'(\varepsilon) \in F,\ \ell(t') \subseteq I\} =^\dagger$$
$$\{x \in \mathcal{C}_\Sigma \mid \exists t' \in \mathcal{T}_Q\colon x[\![q]\!] \to_\mathcal{A}^* t',\ t'(\varepsilon) \in F,\ q \in \mathrm{post}_t(I)\} =$$
$$\bigcup_{q \in \mathrm{post}_t(I)} \mathcal{L}_\uparrow(q) \ .$$

Note that, from the statement to the left of the equality $\dagger$ we have that $t \in \mathcal{L}_\downarrow(q, I)$ for some $q \in Q$, which is equivalent to the fact $q \in \mathrm{post}_t(I)$.

(b) If $\mathcal{A}$ has no unreachable states and L is path-closed, then $\bigcup_{q \in \mathrm{pre}_x(F)} \mathcal{L}_\downarrow(q) = x^{-1}L$.

Recall that $\mathcal{L}_\downarrow(q)$ denotes $\mathcal{L}_\downarrow(q, \mathrm{i}(\mathcal{A}))$, $\mathcal{L}_\uparrow(q)$ denotes $\mathcal{L}_\uparrow(q, F)$ and $\sim_\wr$ denotes $\sim_\wr^F$. We have that:

$$x^{-1}L =$$
$$\{t \in \mathcal{T}_\Sigma \mid x[\![t]\!] \in L\} =^\dagger$$
$$\{t \in \mathcal{T}_\Sigma \mid \exists y \in \mathcal{C}_\Sigma \colon x \sim_\wr y, y[\![t]\!] \in L\} =$$
$$\{t \in \mathcal{T}_\Sigma \mid \exists t' \in \mathcal{T}_Q \colon t \to_{\mathcal{A}}^* t', t'(\varepsilon) = q, x \in P_{\sim_\wr}(\mathcal{L}_\uparrow(q))\} =$$
$$\{t \in \mathcal{T}_\Sigma \mid \exists t' \in \mathcal{T}_Q \colon t \to_{\mathcal{A}}^* t', t'(\varepsilon) = q, q \in \mathrm{pre}_x(F)\} =$$
$$\bigcup_{q \in \mathrm{pre}_x(F)} \mathcal{L}_\downarrow(q) \ ,$$

where equality † holds by Lemma 4.7. Namely, since L is path-closed and $\mathcal{A}$ has no unreachable states, then for every $x, y \in \mathcal{C}_\Sigma$ s.t. $x \sim_\wr y$ we have that $x^{-1}L = y^{-1}L$. $\qquad\square$

Lemma 4.11. Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA. Let $S \subseteq Q$, $f \in \Sigma$, $t_1, \ldots, t_{\langle f \rangle} \in \mathcal{T}_\Sigma$ and let $x, y \in \mathcal{C}_\Sigma$. Then the following hold:

(a) $\mathrm{post}_{f[t_1, \ldots, t_{\langle f \rangle}]}(S) = \delta(\{f[\mathrm{post}_{t_1}(S), \ldots, \mathrm{post}_{t_{\langle f \rangle}}(S)]\})$.

(b) If $\mathcal{A}$ has no unreachable states and $\mathcal{L}(\mathcal{A})$ is path-closed then $\mathrm{pre}_{x[\![y]\!]}(F) = \mathrm{pre}_y(\mathrm{pre}_x(F))$.

Proof:

(a) $\mathrm{post}_{f[t_1, \ldots, t_{\langle f \rangle}]}(S) = \delta(\{f[\mathrm{post}_{t_1}(S), \ldots, \mathrm{post}_{t_{\langle f \rangle}}(S)]\})$.

$$\mathrm{post}_{f[t_1, \ldots, t_{\langle f \rangle}]}(S) =$$
$$\{q \mid f[t_1, \ldots, t_{\langle f \rangle}] \in \mathcal{L}_\downarrow(q, S)\} =$$
$$\{q \mid \forall i \in 1..\langle f \rangle, \exists t'_i \in \mathcal{T}_Q \colon t_i \to_{\mathcal{A}}^* t'_i, t'_i(\varepsilon) = q_i,$$
$$\ell(t'_i) \subseteq S, q \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])\} =$$
$$\{q \mid \forall i \in 1..\langle f \rangle, \exists q_i \in \mathrm{post}_{t_i}(S) \colon q \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])\} =$$
$$\delta(\{f[\mathrm{post}_{t_1}(S), \ldots, \mathrm{post}_{t_{\langle f \rangle}}(S)]\}) \ .$$

Note that the first three equalities hold by definition of $\mathrm{post}(\cdot)$, by definition of downward language of q w.r.t. S and by definition of the transition function δ respectively. Finally, the last equality holds by definition of δ extended to sets.

(b) If $\mathcal{A}$ has no unreachable states and $\mathcal{L}(\mathcal{A})$ is path-closed then $\mathrm{pre}_{x[\![y]\!]}(F) = \mathrm{pre}_y(\mathrm{pre}_x(F))$.

First, note that $\mathcal{L}_\uparrow(q)$ denotes $\mathcal{L}_\uparrow(q, F)$, and $\sim_\wr$ denotes $\sim_\wr^F$. Also recall that, by definition of root-to-pivot equivalence, $x \sim_\wr y$ iff:

(a) $\mathrm{piv}(x) = \mathrm{piv}(y)$ and $\forall p, p' \in (\mathbb{N}_+)^*$ if $p \cdot p' = \mathrm{piv}(x)$ then $x(p) = y(p)$, and

(b) $\exists q \in Q \colon x \in \mathcal{L}_\uparrow(q) \Leftrightarrow \exists q' \in Q \colon y \in \mathcal{L}_\uparrow(q')$.

($\Rightarrow$). We first prove that $\forall q \in Q \colon q \in \mathrm{pre}_{x[\![y]\!]}(F) \Rightarrow q \in \mathrm{pre}_x(\mathrm{pre}_y(F))$. By definition we have that:

$$q \in \mathrm{pre}_{x[\![y]\!]}(F) \Leftrightarrow$$
$$x[\![y]\!] \in P_{\sim_\natural^F}(\mathcal{L}_\uparrow(q)) \Leftrightarrow$$

$$\exists z \in \mathcal{C}_\Sigma \text{ with } z \sim_\natural x[\![y]\!] \colon z \in \mathcal{L}_\uparrow(q) \ . \tag{7}$$

Note that for every $z \in \mathcal{C}_\Sigma$ and for every $i \in 1.. \, \mathrm{h}_\square(z)$, there exists $\tilde{x}, \tilde{y} \in \mathcal{C}_\Sigma$ such that $z = \tilde{x}[\![\tilde{y}]\!]$ and $\mathrm{h}_\square(\tilde{x}) = i$. Using this, we rewrite statement (7) as follows:

$$q \in \mathrm{pre}_{x[\![y]\!]}(F) \Leftrightarrow$$
$$\exists \tilde{x}, \tilde{y} \in \mathcal{C}_\Sigma, \tilde{x}[\![\tilde{y}]\!] \sim_\natural x[\![y]\!], \mathrm{h}_\square(\tilde{x}) = \mathrm{h}_\square(x) \colon \tilde{x}[\![\tilde{y}]\!] \in \mathcal{L}_\uparrow(q) \ .$$

Using the definition of $\mathcal{L}_\uparrow(q)$ and $\to_{\mathcal{A}}^*$:

$$q \in \mathrm{pre}_{x[\![y]\!]}(F) \Leftrightarrow$$
$$\exists \tilde{x}, \tilde{y} \in \mathcal{C}_\Sigma, \tilde{x}[\![\tilde{y}]\!] \sim_\natural x[\![y]\!], \mathrm{h}_\square(\tilde{x}) = \mathrm{h}_\square(x), \exists t, r \in \mathcal{T}_Q \colon$$
$$\tilde{y}[\![q]\!] \to_{\mathcal{A}}^* r, \tilde{x}[\![r(\varepsilon)]\!] \to_{\mathcal{A}}^* t, t(\varepsilon) \in F \ . \tag{8}$$

Note that we are under the conditions of Lemma 4.9. Specifically, $\mathcal{A}$ has no unreachable states and $\mathcal{L}(\mathcal{A})$ is path-closed, by hypothesis. Furthermore, $\mathrm{h}_\square(x) = \mathrm{h}_\square(\tilde{x})$, $\tilde{x}[\![\tilde{y}]\!] \in \mathcal{L}_\uparrow(q)$ and, since $x[\![y]\!] \sim_\natural \tilde{x}[\![\tilde{y}]\!]$, we have that $\exists q' \in Q \colon x[\![y]\!] \in \mathcal{L}_\uparrow(q')$. Therefore, by Lemma 4.9, we have that $\tilde{x}[\![\tilde{y}]\!] \sim_\natural x[\![y]\!] \Leftrightarrow \tilde{x} \sim_\natural x$, and $\tilde{y} \sim_\natural^{\mathrm{pre}_x(F)} y$. Thus:

$$q \in \mathrm{pre}_{x[\![y]\!]}(F) \Rightarrow$$
$$\exists \tilde{x}, \tilde{y} \in \mathcal{C}_\Sigma, (\tilde{x} \sim_\natural x), (\tilde{y} \sim_\natural^{\mathrm{pre}_x(F)} y), \exists t, r \in \mathcal{T}_Q \colon$$
$$\tilde{y}[\![q]\!] \to_{\mathcal{A}}^* r, \tilde{x}[\![r(\varepsilon)]\!] \to_{\mathcal{A}}^* t, t(\varepsilon) \in F \ . \tag{9}$$

Notice that $r(\varepsilon) \in \mathrm{pre}_x(F)$ since $\tilde{x} \sim_\natural x$ and $\tilde{x} \in \mathcal{L}_\uparrow(r(\varepsilon))$. Then:

$$\exists \tilde{x}, \tilde{y} \in \mathcal{C}_\Sigma, (\tilde{x} \sim_\natural x), (\tilde{y} \sim_\natural^{\mathrm{pre}_x(F)} y), \exists t, r \in \mathcal{T}_Q \colon$$
$$\tilde{y}[\![q]\!] \to_{\mathcal{A}}^* r, \tilde{x}[\![r(\varepsilon)]\!] \to_{\mathcal{A}}^* t, t(\varepsilon) \in F \Leftrightarrow$$
$$\exists \tilde{y} \in \mathcal{C}_\Sigma, \tilde{y} \sim_\natural^{\mathrm{pre}_x(F)} y, \exists r \in \mathcal{T}_Q \colon$$
$$\tilde{y}[\![q]\!] \to_{\mathcal{A}}^* r, r(\varepsilon) \in \mathrm{pre}_x(F) \Leftrightarrow$$
$$\exists \tilde{y} \in \mathcal{C}_\Sigma, \tilde{y} \sim_\natural^{\mathrm{pre}_x(F)} y \colon \tilde{y} \in \mathcal{L}_\uparrow(q, \mathrm{pre}_x(F)) \Leftrightarrow$$
$$y \in P_{\sim_\natural^{\mathrm{pre}_x(F)}}(\mathcal{L}_\uparrow(q, \mathrm{pre}_x(F))) \Leftrightarrow$$
$$q \in \mathrm{pre}_y(\mathrm{pre}_x(F)) \ .$$

($\Leftarrow$). Now we show that $\forall q \in Q \colon q \in \mathrm{pre}_y(\mathrm{pre}_x(F)) \Rightarrow q \in \mathrm{pre}_{x\llbracket y \rrbracket}(F)$. Observe that, from the hypothesis $q \in \mathrm{pre}_y(\mathrm{pre}_x(F))$, we can follow the same reasoning as the one above bottom-up up to Equation (9), as every statement is chained with a $\Leftrightarrow$ symbol. Thus, we have that:

$$q \in \mathrm{pre}_x(\mathrm{pre}_y(F)) \Leftrightarrow$$
$$\exists \tilde{x}, \tilde{y} \in \mathcal{C}_\Sigma, (\tilde{x} \sim_{\mathrm{l}} x), (\tilde{y} \sim_{\mathrm{l}}^{\mathrm{pre}_x(F)} y), \exists t, r \in \mathcal{T}_Q \colon$$
$$\tilde{y}\llbracket q \rrbracket \rightarrow_{\mathcal{A}}^* r, \tilde{x}\llbracket r(\varepsilon) \rrbracket \rightarrow_{\mathcal{A}}^* t, t(\varepsilon) \in F \ . \tag{10}$$

Now we check that we are under the hypotheses of Lemma 4.9. First, we have that $\mathrm{h}_\square(x) = \mathrm{h}_\square(\tilde{x})$ since $\tilde{x} \sim_{\mathrm{l}} x$. Also, $\exists q \in Q \colon \tilde{x}\llbracket \tilde{y} \rrbracket \in \mathcal{L}_\uparrow(q)$ (see Equation (10)). Finally, using that $\tilde{x} \sim_{\mathrm{l}} x$ (which implies that $\exists q_1 \in Q \colon x \in \mathcal{L}_\uparrow(q_1)$) and $\tilde{y} \sim_{\mathrm{l}}^{\mathrm{pre}_x(F)} y$ (which implies that $\exists q_2 \in Q \colon y \in \mathcal{L}_\uparrow(q_2, \mathrm{pre}_x(F))$) together with Lemma 4.8, we conclude that $\exists q' \in Q \colon x\llbracket y \rrbracket \in \mathcal{L}_\uparrow(q')$. Thus, we are indeed under the conditions of Lemma 4.9 and we have that:

$$q \in \mathrm{pre}_x(\mathrm{pre}_y(F)) \Rightarrow$$
$$\exists \tilde{x}, \tilde{y} \in \mathcal{C}_\Sigma, \tilde{x}\llbracket \tilde{y} \rrbracket \sim_{\mathrm{l}} x\llbracket y \rrbracket, \mathrm{h}_\square(\tilde{x}) = \mathrm{h}_\square(x), \exists t, r \in \mathcal{T}_Q \colon$$
$$\tilde{y}\llbracket q \rrbracket \rightarrow_{\mathcal{A}}^* r, \tilde{x}\llbracket r(\varepsilon) \rrbracket \rightarrow_{\mathcal{A}}^* t, t(\varepsilon) \in F \ . \tag{11}$$

Finally, from Equation (11), we can follow the same reasoning as the one given above from Equation (8) bottom-up up to the statement $q \in \mathrm{pre}_{x\llbracket y \rrbracket}(F)$, as every statement is chained with a $\Leftrightarrow$ symbol.

We conclude that $\forall q \in Q, q \in \mathrm{pre}_{x\llbracket y \rrbracket}(F) \Leftrightarrow q \in \mathrm{pre}_y(\mathrm{pre}_x(F))$.

$\square$

We are now in position to introduce BTA-based equivalences.

Definition 4.12. (BTA-based equivalences)
Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA and let $t, r \in \mathcal{T}_\Sigma$ and $x, y \in \mathcal{C}_\Sigma$. The *upward and downward BTA-based equivalences* are respectively given by:

$$t \sim_{\mathcal{A}}^{\mathrm{u}} r \overset{\mathrm{def}}{\Leftrightarrow} \mathrm{post}_t(\mathrm{i}(\mathcal{A})) = \mathrm{post}_r(\mathrm{i}(\mathcal{A}))$$
$$x \sim_{\mathcal{A}}^{\mathrm{d}} y \overset{\mathrm{def}}{\Leftrightarrow} \mathrm{pre}_x(F) = \mathrm{pre}_y(F) \ .$$

Note that these equivalences have finite index since tree automata have finitely many states. Next, we show that $\sim_{\mathcal{A}}^{\mathrm{u}}$ and $\sim_{\mathcal{A}}^{\mathrm{d}}$ enable Lemmas 3.3 and 3.9 respectively.

Lemma 4.13. Let $\mathcal{A}$ be a BTA with $L = \mathcal{L}(\mathcal{A})$. Then,

(a) $\sim_{\mathcal{A}}^{\mathrm{u}}$ is an upward congruence and $\sim_{\mathcal{A}}^{\mathrm{u}} \subseteq \sim_L^{\mathrm{u}}$.

(b) If $\mathcal{A}$ has no unreachable states and L is path-closed then $\sim_{\mathcal{A}}^{\mathrm{d}}$ is a strongly downward congruence w.r.t. L and $\sim_{\mathcal{A}}^{\mathrm{d}} \subseteq \sim_L^{\mathrm{d}}$.

Proof:

Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ and, to simplify the notation, let I be the set $\mathrm{i}(\mathcal{A})$.

(a) $\sim_{\mathcal{A}}^{\mathsf{u}}$ is an upward congruence and $\sim_{\mathcal{A}}^{\mathsf{u}} \subseteq \sim_{L}^{\mathsf{u}}$.

We first prove that $\sim_{\mathcal{A}}^{\mathsf{u}}$ is an upward congruence. Let $f \in \Sigma$ and $t_i, r_i \in \mathcal{T}_{\Sigma}$ for $i \in 1..\langle f \rangle$ be such that $t_i \sim_{\mathcal{A}}^{\mathsf{u}} r_i$. By Definition 4.12, we have that $\mathrm{post}_{t_i}(I) = \mathrm{post}_{r_i}(I)$, for each $i \in 1..\langle f \rangle$. Therefore, by Lemma 4.11(a), we conclude that $\mathrm{post}_{f[t_1,\ldots,t_{\langle f \rangle}]}(I) = \mathrm{post}_{f[r_1,\ldots,r_{\langle f \rangle}]}(I)$, i.e., $f[t_1, \ldots, t_{\langle f \rangle}] \sim_{\mathcal{A}}^{\mathsf{u}} f[r_1, \ldots, r_{\langle f \rangle}]$.

Next, we show that $\forall t, r \in \mathcal{T}_{\Sigma} : t \sim_{\mathcal{A}}^{\mathsf{u}} r \Rightarrow L\,t^{-1} = L\,r^{-1}$. Since $t \sim_{\mathcal{A}}^{\mathsf{u}} r$, i.e., $\mathrm{post}_t(I) = \mathrm{post}_r(I)$, we have that $\bigcup_{q \in \mathrm{post}_t(I)} \mathcal{L}_{\uparrow}(q) = \bigcup_{q \in \mathrm{post}_r(I)} \mathcal{L}_{\uparrow}(q)$. By Lemma 4.10(a), we conclude that $L\,t^{-1} = L\,r^{-1}$.

(b) If $\mathcal{A}$ has no unreachable states and L is path-closed then $\sim_{\mathcal{A}}^{\mathsf{d}}$ is a strongly downward congruence w.r.t. L and $\sim_{\mathcal{A}}^{\mathsf{d}} \subseteq \sim_{L}^{\mathsf{d}}$.

First, we show that $\forall x, y \in \mathcal{C}_{\Sigma} : x \sim_{\mathcal{A}}^{\mathsf{d}} y \Rightarrow x^{-1}L = y^{-1}L$. Since $x \sim_{\mathcal{A}}^{\mathsf{d}} y$, i.e., $\mathrm{pre}_x(F) = \mathrm{pre}_y(F)$, we have that:

$$\bigcup_{q \in \mathrm{pre}_x(F)} \mathcal{L}_{\downarrow}(q) = \bigcup_{q \in \mathrm{pre}_y(F)} \mathcal{L}_{\downarrow}(q) \ .$$

Since $\mathcal{A}$ has no unreachable states and L is path-closed, by Lemma 4.10(b), we conclude that $x^{-1}L = y^{-1}L$.

Now we prove that $\sim_{\mathcal{A}}^{\mathsf{d}}$ is a downward congruence. Let $x, y \in \mathcal{C}_{\Sigma} : x \sim_{\mathcal{A}}^{\mathsf{d}} y$, i.e., $\mathrm{pre}_x(F) = \mathrm{pre}_y(F)$. We will show that for every $c \in \mathcal{C}_{\Sigma}$, $x[\![c]\!] \sim_{\mathcal{A}}^{\mathsf{d}} y[\![c]\!]$, i.e., $\mathrm{pre}_{x[\![c]\!]}(F) = \mathrm{pre}_{y[\![c]\!]}(F)$. Relying on Lemma 4.11(b) we have that, $\mathrm{pre}_{x[\![c]\!]}(F) = \mathrm{pre}_c(\mathrm{pre}_x(F)) = \mathrm{pre}_c(\mathrm{pre}_y(F)) = \mathrm{pre}_{y[\![c]\!]}(F)$. Thus, we conclude that $x[\![c]\!] \sim_{\mathcal{A}}^{\mathsf{d}} y[\![c]\!]$.

Finally, we show that $\sim_{\mathcal{A}}^{\mathsf{d}}$ is strongly downward w.r.t. L. Let $x, y \in \mathcal{C}_{\Sigma} : x \sim_{\mathcal{A}}^{\mathsf{d}} y, t \in x^{-1}L, r \in y^{-1}L$ and $t(\varepsilon) = r(\varepsilon)$, with $t, r \in \mathcal{T}_{\Sigma}$. We will prove that $x[\![t[\Box]_i]\!] \sim_{\mathcal{A}}^{\mathsf{d}} x[\![r[\Box]_i]\!]$ and $y[\![t[\Box]_i]\!] \sim_{\mathcal{A}}^{\mathsf{d}} y[\![r[\Box]_i]\!]$, for every $i \in 1..\langle f \rangle$. Note that if the previous holds then, since $\sim_{\mathcal{A}}^{\mathsf{d}}$ is a downward congruence, i.e., $x[\![t[\Box]_i]\!] \sim_{\mathcal{A}}^{\mathsf{d}} y[\![t[\Box]_i]\!]$ and using transitivity of $\sim_{\mathcal{A}}^{\mathsf{d}}$, we can conclude that $x[\![t[\Box]_i]\!] \sim_{\mathcal{A}}^{\mathsf{d}} y[\![r[\Box]_i]\!]$, for every $i \in 1..\langle f \rangle$.

For the clarity of the argument, let us assume w.l.o.g. that $t = f[t_1, t_2]$ and $r = f[r_1, r_2]$. Thus, we have to prove that $x[\![f[\Box, t_2]]\!] \sim_{\mathcal{A}}^{\mathsf{d}} x[\![f[\Box, r_2]]\!]$ and $x[\![f[t_1, \Box]]\!] \sim_{\mathcal{A}}^{\mathsf{d}} x[\![f[r_1, \Box]]\!]$, as well as $y[\![f[\Box, t_2]]\!] \sim_{\mathcal{A}}^{\mathsf{d}} y[\![f[\Box, r_2]]\!]$ and $y[\![f[t_1, \Box]]\!] \sim_{\mathcal{A}}^{\mathsf{d}} y[\![f[r_1, \Box]]\!]$. Note that, by definition, $x[\![f[\Box, t_2]]\!] \sim_{\mathcal{A}}^{\mathsf{d}} x[\![f[\Box, r_2]]\!]$ is equivalent to $\mathrm{pre}_{x[\![f[\Box, t_2]]\!]}(F) = \mathrm{pre}_{x[\![f[\Box, r_2]]\!]}(F)$.

Assume $q \in \mathrm{pre}_{x[\![f[\Box, t_2]]\!]}(F)$, i.e., $x[\![f[\Box, t_2]]\!] \in P_{\sim_{\frac{\cdot}{\cdot}}}(\mathcal{L}_{\uparrow}(q))$. We will show that

$q \in \mathrm{pre}_{x[\![f[\Box, r_2]]\!]}(F)$.

As we have shown previously, $x \sim_{\mathcal{A}}^{\mathsf{d}} y \Rightarrow x^{-1}L = y^{-1}L$. Therefore, since $t \in x^{-1}L$ and $r \in y^{-1}L$, then $r \in x^{-1}L$ and $t \in y^{-1}L$. Hence, as $\mathcal{A}$ has no unreachable states and $r \in x^{-1}L$, we have that $\exists q' \in Q, \exists t' \in \mathcal{T}_Q : x[\![f[q', r_2]]\!] \to_{\mathcal{A}}^{*} t', t'(\varepsilon) \in F$, i.e., $\exists q' \in Q : x[\![f[\Box, r_2]]\!] \in \mathcal{L}_{\uparrow}(q')$. Note that, by Definition 4.3 (root-to-pivot equivalence w.r.t. F),

$x[\![f[\square, t_2]\!]\!] \sim_\wr x[\![f[\square, r_2]\!]\!]$. Since, by hypothesis, $x[\![f[\square, t_2]\!]\!] \in P_{\sim_\wr}(\mathcal{L}_\uparrow(q))$, we have that $x[\![f[\square, r_2]\!]\!] \in P_{\sim_\wr}(\mathcal{L}_\uparrow(q))$. By definition of $\mathrm{pre}_{x[\![f[\square, r_2]\!]\!]}(F)$, we conclude that $q \in \mathrm{pre}_{x[\![f[\square, r_2]\!]\!]}(F)$ and thus, $\mathrm{pre}_{x[\![f[\square, t_2]\!]\!]}(F) \subseteq \mathrm{pre}_{x[\![f[\square, r_2]\!]\!]}(F)$.

The proof of $\mathrm{pre}_{x[\![f[\square, r_2]\!]\!]}(F) \subseteq \mathrm{pre}_{x[\![f[\square, t_2]\!]\!]}(F)$ goes similarly.

Thus, we have that $x[\![f[\square, t_2]\!]\!] \sim_\mathcal{A}^\mathsf{d} x[\![f[\square, r_2]\!]\!]$. On the other hand, the proof of $x[\![f[t_1, \square]\!]\!] \sim_\mathcal{A}^\mathsf{d} x[\![f[r_1, \square]\!]\!]$ is symmetric.

Finally, note that the remainder of the proof, i.e., $y[\![f[\square, t_2]\!]\!] \sim_\mathcal{A}^\mathsf{d} y[\![f[\square, r_2]\!]\!]$ and $y[\![f[t_1, \square]\!]\!] \sim_\mathcal{A}^\mathsf{d} y[\![f[r_1, \square]\!]\!]$ is symmetric, replacing x by y. $\qquad\square$

The next example illustrates the need of root-to-pivot equivalence to define $\mathrm{pre}(\cdot)$ in order to ensure that the automata-based downward congruence is indeed strongly downward.

Example 4.14. Consider the BTA $\mathcal{A} = \langle Q, \Sigma_0 \cup \Sigma_2, \delta, F \rangle$ with $Q = \{q_a, q_b, q_c, q_f, q_\bullet\}$, $\Sigma_0 = \{a, b, c\}$, $\Sigma_2 = \{f\}$, $F = \{q_f\}$, and s.t. δ is defined as follows: $\{q_f\} = \delta(f[q_a, q_a]) = \delta(f[q_b, q_b]) = \delta(f[q_c, q_c]) = \delta(f[q_\bullet, q_\bullet])$; $\{q_a, q_\bullet\} = \delta(a[])$; $\{q_b, q_\bullet\} = \delta(b[])$, and $\{q_c, q_\bullet\} = \delta(c[])$. We will construct $\mathsf{H}^\mathsf{d}(\sim^\mathsf{d}, \mathcal{L}(\mathcal{A})) = \langle Q', \Sigma_0 \cup \Sigma_2, \delta', F' \rangle$ where $\sim^\mathsf{d}$ is a downward congruence defined as follows. For each $x, y \in \mathcal{C}_\Sigma$:

$$x \sim^\mathsf{d} y \overset{\text{def}}{\Leftrightarrow} \mathrm{wpre}_x^\mathcal{A}(F) = \mathrm{wpre}_y^\mathcal{A}(F) \ ,$$

where $\mathrm{wpre}_x^\mathcal{A}(S) \overset{\text{def}}{=} \{q \in Q \mid x \in \mathcal{L}_\uparrow(q, S)\}$, with $S \subseteq Q$. Note that the definition of $\mathrm{wpre}_x^\mathcal{A}(S)$ is similar to that of $\mathrm{pre}_x^\mathcal{A}(S)$ (Definition 4.5) except that we drop root-to-pivot equivalence.

It is easy to check that $\sim^\mathsf{d}$ defines the following blocks:

- $P_{\sim^\mathsf{d}}(X_1)$, where $X_1 = \{f[\square, a], f[a, \square]\}$, since $\mathrm{wpre}_x(F) = \{q_a, q_\bullet\}, \forall x \in X_1$;

- $P_{\sim^\mathsf{d}}(X_2)$, where $X_2 = \{f[\square, b], f[b, \square]\}$, since $\mathrm{wpre}_x(F) = \{q_b, q_\bullet\}, \forall x \in X_2$;

- $P_{\sim^\mathsf{d}}(X_3)$, where $X_3 = \{f[\square, c], f[c, \square]\}$, since $\mathrm{wpre}_x(F) = \{q_c, q_\bullet\}, \forall x \in X_3$;

- $P_{\sim^\mathsf{d}}(\square)$ since $\mathrm{wpre}_x(F) = \{q_f\}$; and

- $P_{\sim^\mathsf{d}}(X_4)$, where $X_4 = \mathcal{C}_{\Sigma_0 \cup \Sigma_2} \setminus (X_1 \cup X_2 \cup X_3 \cup \{\square\})$ since $\mathrm{wpre}_x(F) = \varnothing, \forall x \in X_4$.

By Definition 3.5, $Q' = \{P_{\sim^\mathsf{d}}(X_1), P_{\sim^\mathsf{d}}(X_2), P_{\sim^\mathsf{d}}(X_3), P_{\sim^\mathsf{d}}(\square)\}$, $F' = P_{\sim^\mathsf{d}}(\square)$ and δ' is defined as follows: $\{P_{\sim^\mathsf{d}}(X_1)\} = \delta'(a[])$; $\{P_{\sim^\mathsf{d}}(X_2)\} = \delta'(b[])$; $\{P_{\sim^\mathsf{d}}(X_3)\} = \delta'(c[])$; and $\{P_{\sim^\mathsf{d}}(\square)\} = \delta'(f[P_{\sim^\mathsf{d}}(X_i), P_{\sim^\mathsf{d}}(X_j)]), \forall i, j \in \{1, 2, 3\}$. For instance, note that $P_{\sim^\mathsf{d}}(\square) \in \delta'(f[P_{\sim^\mathsf{d}}(X_3), P_{\sim^\mathsf{d}}(X_1)])$ since there exist $t_1 = a[]$, $t_2 = c[]$ such that $\square[\![f[a, c][\![\square]\!]_1]\!] \in P_{\sim^\mathsf{d}}(X_3)$ and $\square[\![f[a, c][\![\square]\!]_2]\!] \in P_{\sim^\mathsf{d}}(X_1)$.

However, observe that $\mathsf{H}^\mathsf{d}(\sim^\mathsf{d}, \mathcal{L}(\mathcal{A}))$ is not co-deterministic since, for instance, $P_{\sim^\mathsf{d}}(\square) \in \delta'(f[P_{\sim^\mathsf{d}}(X_3), P_{\sim^\mathsf{d}}(X_1)])$ and $P_{\sim^\mathsf{d}}(\square) \in \delta'(f[P_{\sim^\mathsf{d}}(X_1), P_{\sim^\mathsf{d}}(X_3)])$ where $P_{\sim^\mathsf{d}}(X_1) \neq P_{\sim^\mathsf{d}}(X_3)$. This is because $\sim^\mathsf{d}$ is not a strongly downward congruence w.r.t. $\mathcal{L}(\mathcal{A})$. In fact, in Definition 3.8, let $x = y = \square$ and $t = f[a, c]$, $r = f[a, a]$ where $x[\![t]\!], y[\![r]\!] \in \mathcal{L}(\mathcal{A})$. Then, note that (trivially) $x \sim^\mathsf{d} y$, but $x[\![t[\![\square]\!]_1]\!] = f[\square, a] \not\sim^\mathsf{d} f[\square, c] = x[\![r[\![\square]\!]_1]\!]$.

Now we will build $\mathsf{H}^\mathsf{d}(\sim_\mathcal{A}^\mathsf{d}, \mathcal{L}(\mathcal{A})) = \langle Q', \Sigma_0 \cup \Sigma_2, \delta', F' \rangle$ where $\sim_\mathcal{A}^\mathsf{d}$ is the automata-based downward congruence (Definition 4.12). Note that $\sim_\mathcal{A}^\mathsf{d}$ defines the following classes of equivalence:

- $P_{\sim^d_{\mathcal{A}}}(X)$, where $X = \{f[\square, a], f[a, \square], f[\square, b], f[b, \square], f[\square, c], f[c, \square]\}$, as $\text{pre}_x(F) = \{q_a, q_b, q_c, q_\bullet\}, \forall x \in X$;

- $P_{\sim^d_{\mathcal{A}}}(\square)$ as $\text{pre}_x(F) = \{q_f\}$; and

- $P_{\sim^d_{\mathcal{A}}}(X_4)$, where $X_4 = \mathcal{C}_{\Sigma_0 \cup \Sigma_2} \setminus (X \cup \{\square\})$.

Notice that $X = X_1 \cup X_2 \cup X_3$. According to Definition 3.5, $Q' = \{P_{\sim^d_{\mathcal{A}}}(X), P_{\sim^d_{\mathcal{A}}}(\square)\}, F' = \{P_{\sim^d_{\mathcal{A}}}(\square)\}$ and δ' is defined as follows: $\{P_{\sim^d_{\mathcal{A}}}(X)\} = \delta'(a[]) = \delta'(b[]) = \delta'(c[])$ and $\{P_{\sim^d_{\mathcal{A}}}(\square)\} = \delta'(f[P_{\sim^d_{\mathcal{A}}}(X), P_{\sim^d_{\mathcal{A}}}(X)])$.

Note that $\mathsf{H}^d(\sim^d_{\mathcal{A}}, \mathcal{L}(\mathcal{A}))$ is co-deterministic as $\sim^d_{\mathcal{A}}$ is a strongly downward congruence w.r.t. $\mathcal{L}(\mathcal{A})$. The reader may check that $\mathsf{H}^d(\sim^d_{\mathcal{A}}, \mathcal{L}(\mathcal{A}))$ is isomorphic to the co-DBTA $\mathcal{A}^{cD}$ that results from applying the co-determinization operation to $\mathcal{A}$ and removing empty states. $\diamond$

In the light of the Lemma 4.13, the upward BTA-based congruences are indeed finer than (or equal to) the language-based ones, i.e., $\sim^u_{\mathcal{A}} \subseteq \sim^u_L$. So are their downward counterparts if $\mathcal{A}$ has no unreachable states and $L(\mathcal{A})$ is path-closed. The following theorem gives a sufficient condition for the language-based and the BTA-based congruences to coincide.

Theorem 4.15. Let $\mathcal{A}$ be a BTA such that $L = \mathcal{L}(\mathcal{A})$ is a path-closed language. Then,

(a) If $\mathcal{A}$ is a co-DBTA with no empty states, then $\sim^u_{\mathcal{A}} = \sim^u_L$.

(b) If $\mathcal{A}$ is a DBTA with no unreachable states then $\sim^d_{\mathcal{A}} = \sim^d_L$.

Proof:

(a) If $\mathcal{A}$ is a co-DBTA with no empty states, then $\sim^u_{\mathcal{A}} = \sim^u_L$.

By Lemma 4.13(a), we have that $\sim^u_{\mathcal{A}} \subseteq \sim^u_L$.

Next, we show that if $\mathcal{A}$ is a co-DBTA without empty states, then $\sim^u_L \subseteq \sim^u_{\mathcal{A}}$. To simplify the notation, let I denote the set $\text{i}(\mathcal{A})$.

First, note that since $\mathcal{A}$ has no empty states, $Lt^{-1} = Lr^{-1} = \varnothing$ iff $\text{post}_t(I) = \text{post}_r(I) = \varnothing$. Now we consider the case $Lt^{-1} = Lr^{-1} \neq \varnothing$ and proceed by contradiction. Assume that $L\,t^{-1} = L\,r^{-1}$ and $\text{post}_t(I) \neq \text{post}_r(I)$. Recall that:

$$L\,t^{-1} =$$
$$\{x \in \mathcal{C}_\Sigma \mid x[\![t]\!] \in L\} =$$
$$\{x \in \mathcal{C}_\Sigma \mid \exists t' \in \mathcal{T}_Q : x[\![t]\!] \to^*_{\mathcal{A}} t', t'(\varepsilon) \in F, \ell(t') \subseteq I\} =^\dagger$$
$$\{x \in \mathcal{C}_\Sigma \mid \exists t' \in \mathcal{T}_Q : x[\![q]\!] \to^*_{\mathcal{A}} t', t'(\varepsilon) \in F, q \in \text{post}_t(I)\}\ .$$

Note that, from the statement to the left of the equality $\dagger$ we have that $t \in \mathcal{L}_\downarrow(q, I)$ for some $q \in Q$, which is equivalent to the fact $q \in \text{post}_t(I)$.

By hypothesis, $Lt^{-1} = Lr^{-1} \neq \varnothing$, and thus we can assume that there exists $q' \in Q$ with $q' \neq q$ such that $q \in \text{post}_t(I) \cap (\text{post}_r(I))^\complement$ and $q' \in \text{post}_r(I)$. Then, there exists $t' \in \mathcal{T}_\Sigma : x[\![q]\!] \to^*_{\mathcal{A}}$

t', $t'(\varepsilon) \in F$, and there exists $r' \in \mathcal{T}_\Sigma : x[\![q']\!] \to_\mathcal{A}^* r'$, $r'(\varepsilon) \in F$. Since $|F| = 1$, $Lt^{-1} = Lr^{-1}$ and $q \neq q'$, there exists necessarily $q_0 \in Q$ and $f \in \Sigma$ such that $|\delta^{-1}(q_0) \cap \{f\} \times Q^{\langle f \rangle}| > 1$. This is a contradiction since $\mathcal{A}$ is co-deterministic. Therefore, we conclude that if $Lt^{-1} = Lr^{-1}$ then $\mathrm{post}_t(I) = \mathrm{post}_r(I)$.

(b) If $\mathcal{A}$ is a DBTA with no unreachable states then $\sim_\mathcal{A}^{\mathsf{d}} = \sim_L^{\mathsf{d}}$.

Since $\mathcal{A}$ has no unreachable states and $L = \mathcal{L}(\mathcal{A})$ is path-closed, by Lemma 4.13(b), we have that $\sim_\mathcal{A}^{\mathsf{d}}$ is a strongly downward congruence w.r.t. L and $\sim_\mathcal{A}^{\mathsf{d}} \subseteq \sim_L^{\mathsf{d}}$.

Now we show that, in particular, if $\mathcal{A}$ is a DBTA then $\sim_\mathcal{A}^{\mathsf{d}} = \sim_L^{\mathsf{d}}$. Given $x, y \in \mathcal{C}_\Sigma$ with $x^{-1}L = y^{-1}L$, we will show that $\mathrm{pre}_x(F) = \mathrm{pre}_y(F)$.

By Lemma 4.10(b) and the fact that $x^{-1}L = y^{-1}L$, we have that:

$$x^{-1}L = \bigcup_{q \in \mathrm{pre}_x(F)} \mathcal{L}_\downarrow(q) = \bigcup_{q \in \mathrm{pre}_y(F)} \mathcal{L}_\downarrow(q) = y^{-1}L \ . \tag{12}$$

First, note that $\mathcal{A}$ has no unreachable states, i.e., $\forall q \in Q : \mathcal{L}_\downarrow(q) \neq \varnothing$. In particular, $\forall q \in \mathrm{pre}_x(F) : \mathcal{L}_\downarrow(q) \neq \varnothing$.

Second, we prove that if $\mathcal{A}$ is a DBTA then, for every pair of states $q, q' \in Q$, $\mathcal{L}_\downarrow(q) = \mathcal{L}_\downarrow(q')$ implies $q = q'$. The proof goes by contradiction. Assume that $q \neq q'$. Then, since $\mathcal{L}_\downarrow(q) = \mathcal{L}_\downarrow(q')$, for every $t \in \mathcal{L}_\downarrow(q)$, there exists $t', t'' \in \mathcal{T}_Q$ with $t' \neq t''$ s.t. $t \to_\mathcal{A}^* t', t'(\varepsilon) = q$ and $t \to_\mathcal{A}^* t'', t''(\varepsilon) = q''$. Therefore, for some $t_0 \in \mathcal{T}_{\Sigma \cup Q}$ satisfying that $t \to_\mathcal{A}^* t_0 \to_\mathcal{A}^* t'$ and $t \to_\mathcal{A}^* t_0 \to_\mathcal{A}^* t''$, we have that $|\delta(t_0)| > 1$. This is a contradiction since $\mathcal{A}$ is deterministic. Thus, we conclude that if $\mathcal{A}$ is a DBTA then $\mathcal{L}_\downarrow(q) = \mathcal{L}_\downarrow(q')$ implies $q = q'$.

Finally, we prove that $\mathrm{pre}_x(F) = \mathrm{pre}_y(F)$ by contradiction. Assume w.l.o.g. that $q \in \mathrm{pre}_x(F) \cap (\mathrm{pre}_y(F))^{\complement}$. Thus, as we have shown before, for each $q' \in \mathrm{pre}_y(F)$ with $q' \neq q$, we have that $\mathcal{L}_\downarrow(q) \neq \mathcal{L}_\downarrow(q')$. Therefore, equality (12) does not hold, which yields to contradiction. We conclude that, necessarily, $\mathrm{pre}_x(F) = \mathrm{pre}_y(F)$. □

Finally, the following lemma shows that the blocks of $\sim_L^{\mathsf{u}}$ (resp. $\sim_L^{\mathsf{d}}$) can be described as intersections of complemented (using the symbol $\complement$ in superscript) and uncomplemented downward (resp. upward) quotients of L. Similarly, the blocks of $\sim_\mathcal{A}^{\mathsf{u}}$ correspond to intersections of complemented and uncomplemented downward languages of states of $\mathcal{A}$.

For the counterpart characterization of the blocks of $\sim_\mathcal{A}^{\mathsf{d}}$, the result is not symmetric to that of the blocks of $\sim_\mathcal{A}^{\mathsf{u}}$. This is expected as $\sim_\mathcal{A}^{\mathsf{d}}$ is based on our definition of $\mathrm{pre}(\cdot)$, which in turn is defined not only in terms of upward languages but also uses the notion of root-to-pivot equivalence. We will use this lemma to give the generalization of Brzozowski's method for the construction of the minimal DBTA (in Section 5.1).

Lemma 4.16. Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA without unreachable states and such that $L = \mathcal{L}(\mathcal{A})$ and $I = \mathrm{i}(\mathcal{A})$.

Then, for every $t \in \mathcal{T}_\Sigma$, $x \in \mathcal{C}_\Sigma$, we have that:

$$P_{\sim_L^u}(t) = \bigcap_{y \in L t^{-1}} y^{-1}L \ \cap \bigcap_{y \notin L t^{-1}} (y^{-1}L)^\complement$$

$$P_{\sim_L^d}(x) = \bigcap_{r \in x^{-1}L} L r^{-1} \ \cap \bigcap_{r \notin x^{-1}L} (L r^{-1})^\complement$$

$$P_{\sim_\mathcal{A}^u}(t) = \bigcap_{q \in \mathrm{post}_t(I)} \mathcal{L}_\downarrow(q) \ \cap \bigcap_{q \notin \mathrm{post}_t(I)} (\mathcal{L}_\downarrow(q))^\complement$$

$$P_{\sim_\mathcal{A}^d}(x) = \bigcap_{q \in \mathrm{pre}_x(F)} P_{\sim_\wr}(\mathcal{L}_\uparrow(q)) \ \cap \bigcap_{q \notin \mathrm{pre}_x(F)} (P_{\sim_\wr}(\mathcal{L}_\uparrow(q)))^\complement \ .$$

Proof:
Let us start with the equalities on the upward congruences $\sim_L^u$ and $\sim_\mathcal{A}^u$.

For each $r \in \mathcal{T}_\Sigma$ we have that:

$$r \in \bigcap_{y \in Lt^{-1}} y^{-1} L \ \cap \bigcap_{y \notin Lt^{-1}} (y^{-1} L)^\complement \Leftrightarrow^\dagger$$

$$\forall y \in \mathcal{C}_\Sigma \colon y \in Lt^{-1} \Leftrightarrow r \in y^{-1} L \Leftrightarrow^{\dagger\dagger}$$

$$\forall y \in \mathcal{C}_\Sigma \colon y \in Lt^{-1} \Leftrightarrow y \in Lr^{-1} \Leftrightarrow$$

$$Lt^{-1} = Lr^{-1} \Leftrightarrow$$

$$r \in P_{\sim_L^u}(t) \ ,$$

where double-implication $\dagger$ holds by definition of set intersection and $\dagger\dagger$ holds since, for every $r \in \mathcal{T}_\Sigma, y \in \mathcal{C}_\Sigma \colon r \in y^{-1}L \Leftrightarrow y \in Lr^{-1}$.

On the other hand, for each $r \in \mathcal{T}_\Sigma$:

$$r \in \bigcap_{q \in \mathrm{post}_t(I)} \mathcal{L}_\downarrow(q, I) \ \cap \bigcap_{q \notin \mathrm{post}_t(I)} (\mathcal{L}_\downarrow(q, I))^\complement \Leftrightarrow$$

$$\forall q \in Q \colon q \in \mathrm{post}_t(I) \Leftrightarrow r \in \mathcal{L}_\downarrow(q, I) \Leftrightarrow^\dagger$$

$$\forall q \in Q \colon q \in \mathrm{post}_t(I) \Leftrightarrow q \in \mathrm{post}_r(I) \Leftrightarrow$$

$$\mathrm{post}_t(I) = \mathrm{post}_r(I) \Leftrightarrow$$

$$r \in P_{\sim_\mathcal{A}^u}(t) \ ,$$

where double-implication $\dagger$ holds as, for each $r \in \mathcal{T}_\Sigma, q \in Q \colon \ r \in \mathcal{L}_\downarrow(q, I) \Leftrightarrow q \in \mathrm{post}_r(I)$.

Now we move to the equalities on downward congruences. First, for each $z \in \mathcal{C}_\Sigma$ we have that:

$$z \in \bigcap_{r \in x^{-1}L} L r^{-1} \ \cap \bigcap_{r \notin x^{-1}L} (L r^{-1})^\complement \Leftrightarrow$$

$$\forall r \in \mathcal{T}_\Sigma \colon r \in x^{-1}L \Leftrightarrow z \in L r^{-1} \Leftrightarrow^\dagger$$

$$\forall r \in \mathcal{T}_\Sigma \colon r \in x^{-1}L \Leftrightarrow r \in z^{-1}L \Leftrightarrow$$

$$x^{-1}L = z^{-1}L \Leftrightarrow$$

$$z \in P_{\sim_L^d}(x) \ ,$$

where double-implication † holds since, for each $z \in \mathcal{C}_\Sigma, r \in \mathcal{T}_\Sigma \colon r \in z^{-1}L \Leftrightarrow z \in L\, r^{-1}$.

Finally, we prove the last equality. For each $z \in \mathcal{C}_\Sigma$ we have that:

$$z \in \bigcap_{q \in \mathrm{pre}_x(F)} P_{\sim_\flat}(\mathcal{L}_\uparrow(q)) \; \cap \; \bigcap_{q \notin \mathrm{pre}_x(F)} (P_{\sim_\flat}(\mathcal{L}_\uparrow(q)))^\complement \Leftrightarrow$$

$$\forall q \in Q \colon q \in \mathrm{pre}_x(F) \Leftrightarrow z \in P_{\sim_\flat}(\mathcal{L}_\uparrow(q)) \Leftrightarrow^\dagger$$

$$\forall q \in Q \colon q \in \mathrm{pre}_x(F) \Leftrightarrow q \in \mathrm{pre}_z(F) \Leftrightarrow$$

$$\mathrm{pre}_x(F) = \mathrm{pre}_z(F) \Leftrightarrow$$

$$z \in P_{\sim_{\mathcal{A}}^{\mathsf{d}}}(x) \;,$$

where double-implication † holds as, for each $z \in \mathcal{C}_\Sigma, q \in Q \colon \; z \in P_{\sim_\flat}(\mathcal{L}_\uparrow(q)) \Leftrightarrow q \in \mathrm{pre}_z(F)$.   □

4.1. Determinization and minimization of BTAs using congruences

In what follows, we will use $\mathsf{Min}^{\mathsf{u}}, \mathsf{Min}^{\mathsf{d}}$ and $\mathsf{Det}^{\mathsf{u}}, \mathsf{Det}^{\mathsf{d}}$ to denote the constructions $\mathsf{H}^{\mathsf{u}}, \mathsf{H}^{\mathsf{d}}$ when applied, respectively, to the language-based congruences induced by a regular tree language and the automata-based congruences induced by a BTA.

Definition 4.17. ($\mathsf{Min}^{\mathsf{u}}, \mathsf{Min}^{\mathsf{d}}$ and $\mathsf{Det}^{\mathsf{u}}, \mathsf{Det}^{\mathsf{d}}$)
Let $\mathcal{A}$ be a BTA with $L = \mathcal{L}(\mathcal{A})$. Define:

$$\mathsf{Min}^{\mathsf{u}}(L) \overset{\text{def}}{=} \mathsf{H}^{\mathsf{u}}(\sim_L^{\mathsf{u}}, L) \qquad\qquad \mathsf{Det}^{\mathsf{u}}(\mathcal{A}) \overset{\text{def}}{=} \mathsf{H}^{\mathsf{u}}(\sim_{\mathcal{A}}^{\mathsf{u}}, L)$$

$$\mathsf{Min}^{\mathsf{d}}(L) \overset{\text{def}}{=} \mathsf{H}^{\mathsf{d}}(\sim_L^{\mathsf{d}}, L) \qquad\qquad \mathsf{Det}^{\mathsf{d}}(\mathcal{A}) \overset{\text{def}}{=} \mathsf{H}^{\mathsf{d}}(\sim_{\mathcal{A}}^{\mathsf{d}}, L) \;.$$

All the above constructions yield to BTAs defining the language L (recall that, additionally, we need L to be path-closed when using the downward congruences). Concretely, $\mathsf{Det}^{\mathsf{u}}(\mathcal{A})$ and $\mathsf{Det}^{\mathsf{d}}(\mathcal{A})$ correspond to the bottom-up determinization and co-determinization operations defined in Section 2.

On the other hand, since $\mathsf{Min}^{\mathsf{u}}(L)$ and $\mathsf{Min}^{\mathsf{d}}(L)$ are constructed upon the language-based congruences, the resulting BTAs are minimal. More precisely, $\mathsf{Min}^{\mathsf{u}}(L)$ yields the minimal DBTA for L, and $\mathsf{Min}^{\mathsf{d}}(L)$ yields the minimal co-DBTA, as long as L is path-closed. Finally, since $\mathsf{Det}^{\mathsf{d}}(\mathcal{A})$ is a co-DBTA satisfying the conditions of Theorem 4.15(a), when we apply to it the determinization operation ($\mathsf{Det}^{\mathsf{u}}$) we obtain the minimal DBTA for L. Note that the fact that we construct it upon the co-DBTA $\mathsf{Det}^{\mathsf{d}}(\mathcal{A})$ requires that L is path-closed. A similar result holds when co-determinizing ($\mathsf{Det}^{\mathsf{d}}$) the DBTA $\mathsf{Det}^{\mathsf{u}}(\mathcal{A})$. All these notions are summarized by the following theorem.

Theorem 4.18. Let $\mathcal{A}$ be a BTA with $L = \mathcal{L}(\mathcal{A})$. Then the following properties hold:

(a) $\mathcal{L}(\mathsf{Min}^{\mathsf{u}}(L)) = L = \mathcal{L}(\mathsf{Det}^{\mathsf{u}}(\mathcal{A}))$.

(b) If L is path-closed then $\mathcal{L}(\mathsf{Min}^{\mathsf{d}}(L)) = L = \mathcal{L}(\mathsf{Det}^{\mathsf{d}}(\mathcal{A}))$.

(c) $\mathsf{Det}^{\mathsf{u}}(\mathcal{A}) \equiv \mathcal{A}^{\mathsf{D}}$.

(d) If L is path-closed and $\mathcal{A}$ has no unreachable states then $\mathsf{Det}^{\mathsf{d}}(\mathcal{A}) \equiv \mathcal{A}^{\mathsf{cD}}$.

(e) $\mathsf{Min}^{\mathsf{u}}(L)$ is isomorphic to the minimal DBTA for L.

(f) If L is path-closed then $\mathsf{Min}^{\mathsf{d}}(L)$ is isomorphic to the minimal co-DBTA of L.

(g) If L is path-closed then $\mathsf{Det}^{\mathsf{u}}(\mathsf{Det}^{\mathsf{d}}(\mathcal{A})) \equiv \mathsf{Min}^{\mathsf{u}}(L)$.

(h) If L is path-closed then $\mathsf{Det}^{\mathsf{d}}(\mathsf{Det}^{\mathsf{u}}(\mathcal{A})) \equiv \mathsf{Min}^{\mathsf{d}}(L)$.

Proof:

Let $\mathcal{A} \stackrel{\text{def}}{=} \langle Q, \Sigma, \delta, F \rangle$.

(a) $\mathcal{L}(\mathsf{Min}^{\mathsf{u}}(L)) = L = \mathcal{L}(\mathsf{Det}^{\mathsf{u}}(\mathcal{A}))$.

It is well-known that $\sim_L^{\mathsf{u}}$ is an upward congruence [17, 1] and, by Lemma 4.13(a), so is $\sim_{\mathcal{A}}^{\mathsf{u}}$. By definition, we have that $t \sim_L^{\mathsf{u}} r \Rightarrow Lt^{-1} = Lr^{-1}$. On the other hand, also by Lemma 4.13(a), we have that $\sim_{\mathcal{A}}^{\mathsf{u}} \subseteq \sim_L^{\mathsf{u}}$. Therefore, by Lemma 3.3, $\mathcal{L}(\mathsf{Min}^{\mathsf{u}}(L)) = L = \mathcal{L}(\mathsf{Det}^{\mathsf{u}}(\mathcal{A}))$.

(b) If L is path-closed then $\mathcal{L}(\mathsf{Min}^{\mathsf{d}}(L)) = L = \mathcal{L}(\mathsf{Det}^{\mathsf{d}}(\mathcal{A}))$.

Lemmas 4.2 and 4.13(b) show that both $\sim_L^{\mathsf{d}}$ and $\sim_{\mathcal{A}}^{\mathsf{d}}$ are downward congruences satisfying the condition of Lemma 3.7. Therefore, $\mathcal{L}(\mathsf{Min}^{\mathsf{d}}(L)) = L = \mathcal{L}(\mathsf{Det}^{\mathsf{d}}(\mathcal{A}))$.

(c) $\mathsf{Det}^{\mathsf{u}}(\mathcal{A}) \equiv \mathcal{A}^{\mathsf{D}}$.

Recall that, given $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$, $\mathcal{A}^{\mathsf{D}}$ denotes the DBTA that results from applying the bottom-up determinization to $\mathcal{A}$ and removing all unreachable states. Let $\mathcal{A}^{\mathsf{D}} = \langle Q_d, \Sigma, \delta_d, F_d \rangle$ and $\mathsf{Det}^{\mathsf{u}}(\mathcal{A}) = \langle \widetilde{Q}, \Sigma, \widetilde{\delta}, \widetilde{F} \rangle$.

Let P be the partition induced by $\sim_{\mathcal{A}}^{\mathsf{u}}$ and let $\varphi : \widetilde{Q} \to Q_d$ be the mapping assigning to each state $P(t) \in \widetilde{Q}$, the set $\mathrm{post}_t^{\mathcal{A}}(\mathsf{i}(\mathcal{A})) \in Q_d$ with $t \in \mathcal{T}_{\Sigma}$. Observe that φ is well-defined since, by def. 4.12, $t \sim_{\mathcal{A}}^{\mathsf{u}} r$ iff $\mathrm{post}_t^{\mathcal{A}}(\mathsf{i}(\mathcal{A})) = \mathrm{post}_r^{\mathcal{A}}(\mathsf{i}(\mathcal{A}))$. We show that φ is BTA isomorphism between $\mathsf{Det}^{\mathsf{u}}(\mathcal{A})$ and $\mathcal{A}^{\mathsf{D}}$. First, we show that $\varphi(\mathsf{i}(\mathsf{Det}^{\mathsf{u}}(\mathcal{A}))) = \mathsf{i}(\mathcal{A}^{\mathsf{D}})$. To simplify the notation, let us denote $\mathsf{i}(\mathcal{A})$ as I.

$$\varphi(\mathsf{i}(\mathsf{Det}^{\mathsf{u}}(\mathcal{A}))) =$$

$$\varphi(\{P(t) \in \widetilde{Q} \mid \exists a \in \Sigma_0 \colon P(t) \in \widetilde{\delta}(a[\,])\}) = \quad \text{[Def. of } \varphi\text{]}$$

$$\{\mathrm{post}_t^{\mathcal{A}}(I) \mid \exists a \in \Sigma_0 \colon P(t) \in \widetilde{\delta}(a[\,])\} = \quad \text{[Def. 3.1]}$$

$$\{\mathrm{post}_t^{\mathcal{A}}(I) \mid \exists a \in \Sigma_0 \colon a \in P(t)\} = \quad \text{[Def. of } P\text{]}$$

$$\{\mathrm{post}_t^{\mathcal{A}}(I) \mid \exists a \in \Sigma_0 \colon \mathrm{post}_a(I) = \mathrm{post}_t(I)\} \ .$$

Rewriting the above equation we have that:

$$\varphi(\mathsf{i}(\mathsf{Det}^{\mathsf{u}}(\mathcal{A}))) =$$

$$\{\mathrm{post}_a(I) \mid a \in \Sigma_0\} = \quad \text{[Def. of } \mathrm{post}_a(I)\text{]}$$

$$\{q \in Q \mid q \in \delta(a[\,]), a \in \Sigma_0\} = \quad \text{[Def. of } \mathsf{i}(\mathcal{A}^{\mathsf{D}})\text{]}$$

$$\mathsf{i}(\mathcal{A}^{\mathsf{D}}) \ .$$

Next, we show that φ is surjective, i.e. $\forall S \in Q_d, \exists t \in \mathcal{T}_{\Sigma} \colon S = \mathrm{post}_t^{\mathcal{A}}(I)$. We proceed by induction on the structure of $\mathcal{A}^{\mathsf{D}}$, i.e., we set the base case to $S = \mathsf{i}(\mathcal{A}^{\mathsf{D}})$ and we use the transition function δ_d onwards to reach all $S \in Q_d$. Recall that $\mathcal{A}^{\mathsf{D}}$ has no unreachable states.

- *Base case:* Let $S = \mathrm{i}(\mathcal{A}^\mathsf{D})$. Then, $S = \mathrm{post}_a(I)$, with $a \in \Sigma_0$.

- *Inductive step:* W.l.o.g., let us assume that $f \in \Sigma$ with $\langle f \rangle = 2$. Let $S, S_1, S_2 \in Q_d$ be such that $S \in \delta_d(f[S_1, S_2])$. Assume $\exists t_1, t_2 \in \mathcal{T}_\Sigma$ such that $S_i = \mathrm{post}_{t_i}(I)$ with $i \in \{1, 2\}$. Then,

$$S =$$
$$\{q \in Q \mid \exists q_1 \in S_1, q_2 \in S_2, q \in \delta(f[q_1, q_2])\} =$$
$$\{q \in Q \mid \exists q_1 \in \mathrm{post}_{t_1}(I), q_2 \in \mathrm{post}_{t_2}(I),$$
$$q \in \delta(f[q_1, q_2])\} =$$
$$\mathrm{post}_{f[t_1, t_2]}(I) \ ,$$

  where the first equality holds by definition of $\mathcal{A}^\mathsf{D}$, the second equality holds by definition of S_i with $i \in \{1, 2\}$, and the third equality holds using Lemma 4.11(a).
  Since $\mathcal{A}^\mathsf{D}$ keeps only reachable states, it follows that $\forall S \in Q_d, \exists t \in \mathcal{T}_\Sigma \colon \varphi(P(t)) = S$, i.e. φ is surjective.

It is routine to check that φ is injective for otherwise there exists $t \not\sim_\mathcal{A}^\mathsf{u} r$ with $\mathrm{post}_t^\mathcal{A}(\mathrm{i}(\mathcal{A})) = \mathrm{post}_r^\mathcal{A}(\mathrm{i}(\mathcal{A}))$ which contradicts definition 4.12. We thus conclude from above (φ is injective and surjective) that φ is bijective.

Next, we show that $\varphi(\widetilde{F}) = F_d$:

$$\varphi(\widetilde{F}) = \quad [\text{Def. 3.1}]$$
$$\varphi(\{P(t) \mid t \in L\}) = \quad [\text{Def. of } \varphi]$$
$$\{\mathrm{post}_t^\mathcal{A}(I) \mid t \in L\} = \quad [\text{Def. of } \mathcal{L}(\mathcal{A})]$$
$$\{\mathrm{post}_t^\mathcal{A}(I) \mid \mathrm{post}_t^\mathcal{A}(I) \cap F \neq \varnothing\} =$$
$$F_d \ .$$

Note that the last equality holds by definition of $\mathcal{A}^\mathsf{D}$ and using the fact that φ is bijective.

Finally, w.l.o.g., assume that $f \in \Sigma$ with $\langle f \rangle = 2$, and let $P(t), P(t_1), P(t_2) \in \widetilde{Q}$. By definition of $\mathcal{A}^\mathsf{D}$ we have that:

$$\varphi(P(t)) \in \delta_d(f[\varphi(P(t_1)), \varphi(P(t_2))]) \Leftrightarrow$$
$$\mathrm{post}_t^\mathcal{A}(I) = \{q' \in Q \mid \exists q_i' \in \mathrm{post}_{t_i}(I) \colon q' \in \delta(f[q_1', q_2'])\} \ .$$

Using Lemma 4.11(a), we have that the above equation is equivalent to:

$$\mathrm{post}_t^\mathcal{A}(I) = \mathrm{post}_{f[t_1, t_2]}(I) \Leftrightarrow \quad [\text{Def. } \sim_\mathcal{A}^\mathsf{u}]$$
$$t \sim_\mathcal{A}^\mathsf{u} f[t_1, t_2] \Leftrightarrow \quad [\text{Def. of } P]$$
$$P(t) = P(f[t_1, t_2]) \Leftrightarrow^\dagger \ [f[P(t_1), P(t_2)] \subseteq P(f[t_1, t_2])]$$
$$f[P(t_1), P(t_2)] \subseteq P(t) \Leftrightarrow \quad [\text{Def. 3.1}]$$
$$P(t) \in \widetilde{\delta}(f[P(t_1), P(t_2)]) \ .$$

Note that double-implication † holds since $\sim^{\mathsf{u}}_{\mathcal{A}}$ is an upward congruence and thus, $f[P(t_1), P(t_2)] \subseteq P(f[t_1, t_2])$.

(d) If L is path-closed and $\mathcal{A}$ has no unreachable states then $\mathsf{Det}^{\mathsf{d}}(\mathcal{A}) \equiv (\mathcal{A})^{\mathsf{cD}}$.

Recall that, given $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$, $\mathcal{A}^{\mathsf{cD}} = \langle Q_d, \Sigma, \delta_d, F_d \rangle$ denotes the co-DBTA that results from applying the co-determinization operation to $\mathcal{A}$ and removing all empty states. Let $\mathsf{Det}^{\mathsf{d}}(\mathcal{A}) = \langle \widetilde{Q}, \Sigma, \widetilde{\delta}, \widetilde{F} \rangle$. Let P be the partition induced by $\sim^{\mathsf{d}}_{\mathcal{A}}$ and let $\varphi : \widetilde{Q} \to Q_d$ be the mapping assigning to each state $P(x) \in \widetilde{Q}$, the set $\mathrm{pre}^{\mathcal{A}}_x(F) \in Q_d$. Observe that φ is well-defined since, by def. 4.12, $x \sim^{\mathsf{d}}_{\mathcal{A}} y$ iff $\mathrm{pre}^{\mathcal{A}}_x(F) = \mathrm{pre}^{\mathcal{A}}_y(F)$. We show that φ is an isomorphism between $\mathsf{Det}^{\mathsf{d}}(\mathcal{A})$ and $\mathcal{A}^{\mathsf{cD}}$. First, we show that $\varphi(\widetilde{F}) = F_d$.

$$
\begin{aligned}
\varphi(\widetilde{F}) = & \quad \text{[Def. 3.5]} \\
\varphi(\{P(\square)\}) = & \quad \text{[Def. of } \varphi\text{]} \\
\{\mathrm{pre}^{\mathcal{A}}_{\square}(F)\} = & \quad \text{[Def. of } \mathrm{pre}^{\mathcal{A}}_{\square}(F)\text{]} \\
\{F\} = & \quad \text{[Def. of } F_d\text{]} \\
F_d \ . &
\end{aligned}
$$

Next, we show that φ is surjective, i.e. $\forall S \in Q_d, \exists x \in \mathcal{C}_\Sigma$ with $x^{-1}L \neq \varnothing : S = \mathrm{pre}^{\mathcal{A}}_x(F)$. We proceed by induction on the structure of $\mathcal{A}^{\mathsf{cD}}$, i.e., we set the base case to $S = F_d$ and we use the transition function δ_d backwards to reach all $S \in Q_d$. Recall that $\mathcal{A}^{\mathsf{cD}}$ has no empty states and the set of states of $\mathsf{Det}^{\mathsf{d}}(\mathcal{A})$, i.e., $\widetilde{Q}$, is defined as $\widetilde{Q} = \{P(x) \mid x \in \mathcal{C}_\Sigma, x^{-1}L \neq \varnothing\}$.

- *Base case:* Let $S = F_d$. Then $S = \mathrm{pre}^{\mathcal{A}}_x(F)$ with $x = \square$.
- *Inductive step:* Now, let $f \in \Sigma$ and let $S, S_1, \ldots, S_{\langle f \rangle} \in Q_d$ be such that $S \in \delta_d(f[S_1, \ldots, S_{\langle f \rangle}])$. By I.H., $\exists x \in \mathcal{C}_\Sigma$ with $x^{-1}L \neq \varnothing : S = \mathrm{pre}^{\mathcal{A}}_x(F)$. Now let us show that $\forall i \in 1..\langle f \rangle, \exists z \in \mathcal{C}_\Sigma$ with $z^{-1}L \neq \varnothing : S_i = \mathrm{pre}^{\mathcal{A}}_z(F)$. By definition of $\mathcal{A}^{\mathsf{cD}}$ we have that, for all $i \in 1..\langle f \rangle$:

$$
\begin{aligned}
S_i = \{q_i \in Q \mid \exists q' \in S, \exists q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_{\langle f \rangle} \in Q : \\
q' \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])\} \ .
\end{aligned}
$$

Since $\mathcal{A}$ has no unreachable states, the above set is equivalent to:

$$
\begin{aligned}
\{q_i \in Q \mid \exists t \in \mathcal{T}_\Sigma, \exists r \in \mathcal{T}_Q : t(\varepsilon) = f, t[\![q_i]\!]_i \to^*_{\mathcal{A}} r, \\
r(\varepsilon) \in S\} \ .
\end{aligned}
\tag{13}
$$

Note that $q_i \in Q$ belongs to the set in (13) iff $q_i \in \mathrm{pre}_{\widetilde{t}[\![\square]\!]_i}(S)$, for every $\widetilde{t} \in \mathcal{T}_\Sigma$ with $\widetilde{t}[\![\square]\!]_i \sim^S_{\text{?}} t[\![\square]\!]_i$. In fact, $\exists t \in \mathcal{T}_\Sigma, \exists r \in \mathcal{T}_Q : t(\varepsilon) = f, r(\varepsilon) \in S$ and $t[\![q_i]\!]_i \to^*_{\mathcal{A}} r$ implies that $q_i \in \mathrm{pre}_{\widetilde{t}[\![\square]\!]_i}(S)$ by setting $\widetilde{t}[\![\square]\!]_i = t[\![\square]\!]_i$. On the other hand, if $q_i \in \mathrm{pre}_{\widetilde{t}[\![\square]\!]_i}(S)$ with $\widetilde{t}[\![\square]\!]_i \sim^S_{\text{?}} t[\![\square]\!]_i$ then, by definition of $\mathrm{pre}_{\widetilde{t}[\![\square]\!]_i}(S)$, $\exists y \in \mathcal{C}_\Sigma$ with $y \sim^S_{\text{?}} \widetilde{t}[\![\square]\!]_i : y \in \mathcal{L}^{\mathcal{A}}_{\uparrow}(q_i, S)$. Therefore, $\exists t \in \mathcal{T}_\Sigma, \exists r \in \mathcal{T}_Q : t(\varepsilon) = f, t[\![q_i]\!]_i \to^*_{\mathcal{A}} r$ and $r(\varepsilon) \in S$ by setting $t[\![\square]\!]_i = y$.

Thus, the set in (13) is equivalent to:

$$\{q_i \in Q \mid \exists \widetilde{t} \in \mathcal{T}_\Sigma \colon \widetilde{t}[\![\Box]\!]_i \sim^S_{\tilde{?}} t[\![\Box]\!]_i, q_i \in \mathrm{pre}^{\mathcal{A}}_{\widetilde{t}[\![\Box]\!]_i}(S)\} \ .$$

By induction hypothesis we have that the above set is equivalent to:

$$\{q_i \in Q \mid \exists \widetilde{t} \in \mathcal{T}_\Sigma, \exists x \in \mathcal{C}_\Sigma \colon \widetilde{t}[\![\Box]\!]_i \sim^S_{\tilde{?}} t[\![\Box]\!]_i, \ x^{-1}L \neq \varnothing,$$
$$q_i \in \mathrm{pre}^{\mathcal{A}}_{\widetilde{t}[\![\Box]\!]_i}(\mathrm{pre}^{\mathcal{A}}_x(F))\} \ .$$

Finally, using Lemma 4.11(b), the latter is equivalent to:

$$\{q_i \in Q \mid \exists \widetilde{t} \in \mathcal{T}_\Sigma, \exists x \in \mathcal{C}_\Sigma \colon \widetilde{t}[\![\Box]\!]_i \sim^S_{\tilde{?}} t[\![\Box]\!]_i, \ x^{-1}L \neq \varnothing,$$
$$q_i \in \mathrm{pre}^{\mathcal{A}}_{x[\![\widetilde{t}[\![\Box]\!]_i]\!]}(F)\} \ .$$

Therefore, for all non-empty $S_i \in Q_d$, there exists $z(= x[\![\widetilde{t}[\![\Box]\!]_i]\!]) \in \mathcal{C}_\Sigma$ with $z^{-1}L \neq \varnothing$ such that $S_i = \mathrm{pre}^{\mathcal{A}}_z(F)$ and, since $\mathcal{A}^{\mathsf{cD}}$ has no empty states, it follows that φ is surjective.

It is routine to check that φ is injective for otherwise there exists $x \not\sim^{\mathsf{d}}_{\mathcal{A}} y$ with $\mathrm{pre}^{\mathcal{A}}_x(F) = \mathrm{pre}^{\mathcal{A}}_y(F)$ which contradicts definition 4.12. We thus conclude from above (φ is injective and surjective) that φ is bijective.

On the other hand, $\varphi(\mathrm{i}(\mathsf{Det}^{\mathsf{d}}(\mathcal{A}))) = \mathrm{i}(\mathcal{A}^{\mathsf{cD}})$ since:

$$\varphi(\mathrm{i}(\mathsf{Det}^{\mathsf{d}}(\mathcal{A}))) =$$
$$\varphi(\{P(x) \in \widetilde{Q} \mid \exists a \in \Sigma_0 \colon P(x) \in \widetilde{\delta}(a[])\}) =$$
$$\{\mathrm{pre}^{\mathcal{A}}_x(F) \mid \exists a \in \Sigma_0 \colon P(x) \in \widetilde{\delta}(a[])\} =$$
$$\{\mathrm{pre}^{\mathcal{A}}_x(F) \mid \exists a \in \Sigma_0 \colon P(x)[\![a]\!] \subseteq L\} \ .$$

By definition of $\mathcal{L}(\mathcal{A})$ and Lemma 4.7 we have that the above set is equivalent to:

$$\{\mathrm{pre}^{\mathcal{A}}_x(F) \mid \exists a \in \Sigma_0, \exists q \in \mathrm{pre}_x(F) \colon q \in \delta(a[])\} \ .$$

Finally, since φ is bijective we have that the latter is equivalent to:

$$\{q_d \in Q_d \mid \exists a \in \Sigma_0, \exists q \in q_d \colon q \in \delta(a[])\} = \ [\text{Def. of } \delta_d]$$
$$\{q_d \in Q_d \mid \exists a \in \Sigma_0 \colon q_d \in \delta_d(a[])\} = \ [\text{Def. of } \mathrm{i}(\mathcal{A}^{\mathsf{cD}})]$$
$$\mathrm{i}(\mathcal{A}^{\mathsf{cD}}) \ .$$

Finally, we prove that:

$$\varphi(P(x)) \in \delta_d(f[\varphi(P(x_1)), \ldots, \varphi(P(x_{\langle f \rangle}))]) \text{ iff}$$
$$P(x) \in \widetilde{\delta}(f[P(x_1), \ldots, P(x_{\langle f \rangle})]) \ ,$$

where $P(x), P(x_1), \ldots, P(x_k) \in \widetilde{Q}$ and $f \in \Sigma$.

Note that, by definition of $\mathcal{A}^{\mathsf{cD}}$ and φ, $\varphi(P(x)) \in \delta_d(f[\varphi(P(x_1)), \ldots, \varphi(P(x_{\langle f \rangle}))])$ iff, for each $i \in 1..\langle f \rangle$:

$$\mathrm{pre}_{x_i}^{\mathcal{A}}(F) = \{q_i \in Q \mid \exists q \in \mathrm{pre}_x^{\mathcal{A}}(F), \exists q_1, \ldots, q_{\langle f \rangle} \in Q \colon$$
$$q \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])\} \ .$$

Since $\mathcal{A}$ has no unreachable states, for all $i \in 1..\langle f \rangle$, we have that:

$$\mathrm{pre}_{x_i}^{\mathcal{A}}(F) = \{q_i \in Q \mid \exists q \in \mathrm{pre}_x^{\mathcal{A}}(F), \exists t \in \mathcal{L}_{\downarrow}(q) \colon$$
$$t(\varepsilon) = f, q_i \in \mathrm{pre}_{t[\![\square]\!]_i}^{\mathcal{A}}(\{q\})\} \ .$$

By Lemma 4.11(b), we have that the above equality is equivalent to:

$$\exists t \in \mathcal{T}_\Sigma \colon t(\varepsilon) = f, \mathrm{pre}_{x_i}^{\mathcal{A}}(F) = \mathrm{pre}_{x[\![t[\![\square]\!]_i]\!]}^{\mathcal{A}}(F) \Leftrightarrow$$
$$\exists t \in \mathcal{T}_\Sigma \colon t(\varepsilon) = f, x_i \sim_{\mathcal{A}}^{\mathsf{d}} x[\![t[\![\square]\!]_i]\!] \Leftrightarrow$$
$$\exists t \in \mathcal{T}_\Sigma \colon t(\varepsilon) = f, P(x_i) = P(x[\![t[\![\square]\!]_i]\!]) \Leftrightarrow^\dagger$$
$$\exists t \in \mathcal{T}_\Sigma \colon t(\varepsilon) = f, P(x)[\![t[\![\square]\!]_i]\!] \subseteq P(x_i) \Leftrightarrow$$
$$P(x) \in \tilde{\delta}(f[P(x_1), \ldots, P(x_k)]) \ .$$

Note that double-implication $\dagger$ holds since $\sim_{\mathcal{A}}^{\mathsf{d}}$ is a downward congruence and thus, $P(x)[\![t[\![\square]\!]_i]\!] \subseteq P(x[\![t[\![\square]\!]_i]\!])$.

(e) $\mathsf{Min}^{\mathsf{u}}(L)$ is isomorphic to the minimal DBTA for L.

Let P be the partition induced by the Nerode's upward congruence $\sim_L^{\mathsf{u}}$. As shown by Comon et al. [1], the minimal DBTA for L is the BTA $\mathcal{M} = \langle Q', \Sigma, \delta', F' \rangle$ where $Q' = \{P(t) \mid t \in \mathcal{T}_\Sigma\}$, $F' = \{P(t) \mid t \in L\}$ and $\delta'(f[P(t_1), \ldots, P(t_{\langle f \rangle})]) = P(f[t_1, \ldots, t_{\langle f \rangle}])$ for every $f \in \Sigma$ and $t, t_1, \ldots, t_{\langle f \rangle} \in \mathcal{T}_\Sigma$. Hence, it is easy to check that $\mathcal{M} \equiv \mathsf{Min}^{\mathsf{u}}(L)$.

(f) If L is path-closed then $\mathsf{Min}^{\mathsf{d}}(L)$ is isomorphic to the minimal co-DBTA for L.

Trivially, $\sim^{\mathsf{d}} \overset{\mathrm{def}}{=} \sim_L^{\mathsf{d}}$ is the coarsest strongly downward congruence w.r.t. L that satisfies:

$$\forall x, y \in \mathcal{C}_\Sigma \colon x \sim^{\mathsf{d}} y \Rightarrow x^{-1}L = y^{-1}L \ . \tag{14}$$

Next, we show that $\mathsf{Min}^{\mathsf{d}}(L)$ is the minimal co-DBTA for L by contradiction.

Let $\mathcal{A}' = \langle Q, \Sigma, \delta, F \rangle$ be a co-DBTA for L with strictly fewer states than $\mathsf{Min}^{\mathsf{d}}(L)$. Note that we can assume that $\mathcal{A}'$ has no unreachable states, otherwise we could simply remove them and obtain a smaller equivalent co-DBTA.

First, we show that, for every co-DBTA $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ with no unreachable states, the set $\mathrm{pre}_x^{\mathcal{A}}(F)$ is a singleton, for every $x \in \mathcal{C}_\Sigma$. We proceed by contradiction. Let $F = \{q_f\}$, $x \in \mathcal{C}_\Sigma$ and assume that $\exists q, q' \in Q$ with $q \neq q'$ and such that $q, q' \in \mathrm{pre}_x^{\mathcal{A}}(F)$, i.e., $x \in P_{\sim_i}(\mathcal{L}_{\uparrow}(q))$ and $x \in P_{\sim_i}(\mathcal{L}_{\uparrow}(q'))$. By definition of $P_{\sim_i}$, $\exists y \sim_i x \colon y \in \mathcal{L}_{\uparrow}(q)$ and $\exists y' \sim_i x \colon y \in \mathcal{L}_{\uparrow}(q')$,

i.e., $\exists t, t' \in \mathcal{T}_Q : y[\![q]\!] \rightarrow_{\mathcal{A}}^* t, y'[\![q']\!] \rightarrow_{\mathcal{A}}^* t'$ and $t(\varepsilon) = t'(\varepsilon) = q_f$. Observe that $y \sim_{\dot\varsigma} y'$, by transitivity of $\sim_{\dot\varsigma}$.

Since $q \neq q'$, $t(\varepsilon) = t'(\varepsilon) = q_f$ and $y \sim_{\dot\varsigma} y'$, there exists necessarily $y_1, y_1' \in \mathcal{C}_{\Sigma \cup Q}$ with $y_1 \sim_{\dot\varsigma} y_1'$, $r_i, r_i' \in \mathcal{T}_Q$, $f \in \Sigma$ and $q_0 \in Q$ s.t.:

$$y[\![q]\!] \rightarrow_{\mathcal{A}}^* y_1[\![f[r_1, \ldots, r_{\langle f \rangle}]]\!] \rightarrow_{\mathcal{A}} \tilde{x}[\![q_0[r_1, \ldots, r_{\langle f \rangle}]]\!] \rightarrow_{\mathcal{A}}^* t$$

$$\text{and}$$

$$y'[\![q']\!] \rightarrow_{\mathcal{A}}^* y_1'[\![f[r_1', \ldots, r_{\langle f \rangle}']]\!] \rightarrow_{\mathcal{A}} \tilde{y}[\![q_0[r_1', \ldots, r_{\langle f \rangle}']]\!] \rightarrow_{\mathcal{A}}^* t' ,$$

where $r_i \neq r_i'$, for some $i \in 1..\langle f \rangle$. It follows that $\exists q_0 \in \delta(f[r_1(\varepsilon), \ldots, r_{\langle f \rangle}(\varepsilon)])$ and $q_0 \in \delta(f[r_1'(\varepsilon), \ldots, r_{\langle f \rangle}'(\varepsilon)])$ with $r_i(\varepsilon) \neq r_i'(\varepsilon)$ for some $i \in 1..\langle f \rangle$, which contradicts the fact that $\mathcal{A}$ is co-deterministic. Thus, we conclude that if $\mathcal{A}$ is a co-DBTA with no unreachable states, $\mathrm{pre}_x^{\mathcal{A}}(F)$ is a singleton, for every $x \in \mathcal{C}_\Sigma$.

Therefore, $\sim_{\mathcal{A}'}^d$, which by Lemma 4.13(b) is a strongly downward congruence w.r.t. L that satisfies Equation (14), has as many equivalence classes as states has $\mathcal{A}'$. Since $\mathcal{A}'$ has fewer states than $\mathsf{Min}^d(L)$, it follows that $\sim_{\mathcal{A}'}^d \subset \sim_L^d$, which contradicts the fact that $\sim_L^d$ is the coarsest strongly downward congruence that satisfies Equation (14).

Therefore $\mathcal{A}'$ has, at least, as many states as $\mathsf{Min}^d(L)$ and, as a consequence, $\mathsf{Min}^d(L)$ is the minimal co-DBTA for L.

(g) If L is path-closed then $\mathsf{Det}^u(\mathsf{Det}^d(\mathcal{A})) \equiv \mathsf{Min}^u(L)$.

It follows from Corollary A.4 and 3.9 that $\mathsf{Det}^d(\mathcal{A})$ is a co-DBTA with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathsf{Det}^d(\mathcal{A}))$. Furthermore, by Remark 3.6, $\mathsf{Det}^d(\mathcal{A})$ has no empty states. Therefore, by Theorem 4.15(a), we have that $\sim_{\mathsf{Det}^d(\mathcal{A})}^u = \sim_{\mathcal{L}(\mathcal{A})}^u$ so $\mathsf{Det}^u(\mathsf{Det}^d(\mathcal{A})) \equiv \mathsf{Min}^u(L)$.

(h) If L is path-closed then $\mathsf{Det}^d(\mathsf{Det}^u(\mathcal{A})) \equiv \mathsf{Min}^d(L)$.

It follows by Remark 3.2 that $\mathsf{Det}^u(\mathcal{A})$ is a DBTA with no unreachable states. Furthermore, by Lemma 3.3, $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathsf{Det}^u(\mathcal{A}))$. Thus, by Theorem 4.15(b), we have that $\sim_{\mathsf{Det}^u(\mathcal{A})}^d = \sim_{\mathcal{L}(\mathcal{A})}^d$, so $\mathsf{Det}^d(\mathsf{Det}^u(\mathcal{A})) \equiv \mathsf{Min}^d(L)$. □

5. A congruence-based perspective on Brzozowski's method

Brzozowski's double-reversal method [15] is a classical algorithm for minimizing word automata. It relies on the fact that determinizing a co-deterministic automaton $\mathcal{N}$ yields the minimal deterministic automaton for the language of $\mathcal{N}$. In this section, we show, as an easy consequence of Theorem 4.18(c), (d), and (g), that this algorithm can be adapted for finding the minimal DBTA for the language of a given BTA. To the best of our knowledge this is the first proof of correctness of this method for minimizing bottom-up tree automata. A precise statement is given next followed by a justification.

Corollary 5.1. Let $\mathcal{A}$ be a BTA without unreachable states such that $L = \mathcal{L}(\mathcal{A})$ is a path-closed language. Then, $(\mathcal{A}^{cD})^D \equiv \mathsf{Min}^u(L)$.

To be precise, Brzozowski's double-reversal method for word automata constructs the intermediate co-deterministic automaton by combining a reverse operation, followed by a determinization operation and then a reverse operation again. However, the reverse construction applied to BTAs, i.e., the operation that flips the direction of the transition function and switches final states by initial states, does not yield a BTA. On the contrary, it produces, a *top-down tree automaton* (TTA).

Top-down tree automata (see [1] for a detailed description of this model), as opposed to the class of BTAs, start their computations from the root, which is an *initial* state of the automaton, down to the leaves. Since TTAs can be interpreted as the *reverse* of BTAs, indeed both classes of automata define the regular tree languages, all the results of Section 4 have TTA-equivalents that we defer to Appendix A.1. It is worth noting that, as co-DBTAs, deterministic TTAs are strictly less expressive than the general TTAs and define the subclass of path-closed languages. Since Brzozowski's method goes through the construction of an intermediate co-DBTA $\mathcal{A}^{\mathsf{cD}}$, this algorithm is restricted to BTAs defining path-closed tree languages.

Relying on the definitions and notation we introduce in Appendix A.1, we conclude this section by stating Brzozowski's method using the reverse operation. Namely, given a BTA $\mathcal{A}$, $\mathcal{A}^R$ denotes the reverse TTA of $\mathcal{A}$ and $(\mathcal{A}^R)^{\mathsf{D}}$ denotes the result of applying the counterpart determinization operation on TTAs. Thus, $((\mathcal{A}^R)^{\mathsf{D}})^R \equiv \mathcal{A}^{\mathsf{cD}}$ and the following holds.

Corollary 5.2. Let $\mathcal{A}$ be a BTA without unreachable states such that $L = \mathcal{L}(\mathcal{A})$ is a path-closed language. Then, $(((\mathcal{A}^R)^{\mathsf{D}})^R)^{\mathsf{D}} \equiv \mathsf{Min}^{\mathsf{u}}(L)$.

5.1. Generalization of Brzozowski's method

Brzozowski's double-reversal methods builds a co-DBTA in order to guarantee, by Theorem 4.15(a), that determinizing the automaton produces the minimal DBTA.

In the word automata case, Brzozowski and Tamm [18] showed that going through a co-deterministic automata is not necessary and defined a class of automata, which strictly contains the co-deterministic ones, for which determinizing the automata yields the minimal deterministic automaton. Next we generalize that result from word to trees which, incidentally, allows us to drop the restriction to the path-closed languages.

Theorem 5.3. Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA with $L = \mathcal{L}(\mathcal{A})$. Then $\mathsf{Det}^{\mathsf{u}}(\mathcal{A}) \equiv \mathsf{Min}^{\mathsf{u}}(L)$ iff $\forall q \in Q \colon P_{\sim_L^{\mathsf{u}}}(\mathcal{L}_{\downarrow}(q)) = \mathcal{L}_{\downarrow}(q)$.

Proof:
First, we show that if $\mathsf{Det}^{\mathsf{u}}(\mathcal{A})$ is the minimal DBTA for L then we have $\forall q \in Q \colon P_{\sim_L^{\mathsf{u}}}(\mathcal{L}_{\downarrow}(q)) = \mathcal{L}_{\downarrow}(q)$.

To simplify the notation, let I denote $\mathsf{i}(\mathcal{A})$. Then, we have that:

$$P_{\sim_L^{\mathsf{u}}}(\mathcal{L}_{\downarrow}(q)) =$$
$$\{r \in \mathcal{T}_{\Sigma} \mid \exists t \in \mathcal{L}_{\downarrow}(q) \colon L\, r^{-1} = L\, t^{-1}\} = \quad [\sim_L^{\mathsf{u}} = \sim_{\mathcal{A}}^{\mathsf{u}}]$$
$$\{r \in \mathcal{T}_{\Sigma} \mid \exists t \in \mathcal{L}_{\downarrow}(q) \colon \mathrm{post}_r^{\mathcal{A}}(I) = \mathrm{post}_t^{\mathcal{A}}(I)\} \ .$$

Note that $\sim_L^{\mathsf{u}} = \sim_{\mathcal{A}}^{\mathsf{u}}$ by hypothesis, since $\mathsf{Det}^{\mathsf{u}}(\mathcal{A}) \equiv \mathsf{Min}^{\mathsf{u}}(L)$. On the other hand, by definition, $q \in \mathrm{post}_t^{\mathcal{A}}(I) \Rightarrow t \in \mathcal{L}_{\downarrow}(q)$. Thus, we have the following set inclusion:

$$\{r \in \mathcal{T}_{\Sigma} \mid \exists t \in \mathcal{L}_{\downarrow}(q)\colon \ \mathrm{post}_r^{\mathcal{A}}(I) = \mathrm{post}_t^{\mathcal{A}}(I)\} \subseteq$$
$$\{r \in \mathcal{T}_{\Sigma} \mid q \in \mathrm{post}_r^{\mathcal{A}}(I)\} =$$
$$\mathcal{L}_{\downarrow}(q) \ .$$

By reflexivity of $\sim_L^{\mathsf{u}}$, we have that $\mathcal{L}_{\downarrow}(q) \subseteq P_{\sim_L^{\mathsf{u}}}(\mathcal{L}_{\downarrow}(q))$, and thus we conclude $P_{\sim_L^{\mathsf{u}}}(\mathcal{L}_{\downarrow}(q)) = \mathcal{L}_{\downarrow}(q)$.

Now, assume that $P_{\sim_L^{\mathsf{u}}}(\mathcal{L}_{\downarrow}(q)) = \mathcal{L}_{\downarrow}(q)$, for each $q \in Q$. Then, for every $t \in \mathcal{T}_{\Sigma}$,

$$P_{\sim_{\mathcal{A}}^{\mathsf{u}}}(t) =^{\dagger}$$
$$\bigcap_{q \in \mathrm{post}_t^{\mathcal{A}}(I)} \mathcal{L}_{\downarrow}(q) \cap \bigcap_{q \notin \mathrm{post}_t^{\mathcal{A}}(I)} (\mathcal{L}_{\downarrow}(q))^{\complement} =^{\dagger\dagger}$$
$$\bigcap_{q \in \mathrm{post}_t^{\mathcal{A}}(I)} P_{\sim_L^{\mathsf{u}}}(\mathcal{L}_{\downarrow}(q)) \cap \bigcap_{q \notin \mathrm{post}_t^{\mathcal{A}}(I)} (P_{\sim_L^{\mathsf{u}}}(\mathcal{L}_{\downarrow}(q)))^{\complement} \ . \tag{15}$$

Note that equality $\dagger$ holds by Lemma 4.16 and $\dagger\dagger$ holds since $P_{\sim_L^{\mathsf{u}}}(\mathcal{L}_{\downarrow}(q)) = \mathcal{L}_{\downarrow}(q)$ by hypothesis.

It follows from (15) that $P_{\sim_{\mathcal{A}}^{\mathsf{u}}}(t)$ is a *union* of blocks of $P_{\sim_L^{\mathsf{u}}}$, for each $t \in \mathcal{T}_{\Sigma}$. In other words, $\sim_L^{\mathsf{u}} \subseteq \sim_{\mathcal{A}}^{\mathsf{u}}$. On the other hand, by Lemma 4.13(a), $\sim_{\mathcal{A}}^{\mathsf{u}} \subseteq \sim_L^{\mathsf{u}}$. Therefore, $P_{\sim_{\mathcal{A}}^{\mathsf{u}}}(t)$ necessarily corresponds to one single block of $P_{\sim_L^{\mathsf{u}}}$, namely, $P_{\sim_L^{\mathsf{u}}}(t)$. Since $P_{\sim_{\mathcal{A}}^{\mathsf{u}}}(t) = P_{\sim_L^{\mathsf{u}}}(t)$ for each $t \in \mathcal{T}_{\Sigma}$, we conclude that $\mathsf{Det}^{\mathsf{u}}(\mathcal{A}) \equiv \mathsf{Min}^{\mathsf{u}}(L)$. $\square$

Given a regular tree language L, the minimal DBTA for L is $\mathsf{Min}^{\mathsf{u}}(L)$ (Theorem 4.18(e)). On the other hand, we show (see proof of Lemma 3.3) that the states of the minimal DBTA are in one-to-one correspondence with the blocks of $P_{\sim_L^{\mathsf{u}}}$, i.e., for every state q of $\mathsf{Min}^{\mathsf{u}}(L)$, there exists a tree $t \in \mathcal{T}_{\Sigma}$ such that $\mathcal{L}_{\downarrow}(q) = P_{\sim_L^{\mathsf{u}}}(t)$ and vice-versa. Therefore, for every $S \subseteq \mathcal{T}_{\Sigma}$, $P_{\sim_L^{\mathsf{u}}}(S) = S$ iff S is a union of downward languages of states of the minimal DBTA for L. This property and Theorem 5.3 allows us to give an alternative characterization of the class of automata for which the determinization operation yields the minimal DBTA.

Corollary 5.4. Let $\mathcal{A}$ be a BTA with $L = \mathcal{L}(\mathcal{A})$. Then $(\mathcal{A})^{\mathsf{D}} \equiv \mathsf{Min}^{\mathsf{u}}(L)$ iff the downward language of every state in $\mathcal{A}$ is a union of downward languages of the minimal DBTA for L.

It is worth pointing that similarly to path-closedness the conditions of Corollary 5.4 and Theorem 5.3 are decidable since isomorphism is decidable, $\mathsf{Det}^{\mathsf{u}}(\mathcal{A})$ and $(\mathcal{A})^{\mathsf{D}}$ are effectively computable and so is $\mathsf{Min}^{\mathsf{u}}(L)$ [1].

Finally, we give the counterpart result of Theorem 5.3 for co-DBTAs. Namely, we show a sufficient and necessary condition on a BTA $\mathcal{A}$ that guarantees that when we co-determinize it ($\mathcal{A}^{\mathsf{cD}}$) we obtain the minimal co-DBTA for $\mathcal{L}(\mathcal{A})$.

Theorem 5.5. Let $\mathcal{A} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA without unreachable states and such that $L = \mathcal{L}(\mathcal{A})$ is a path-closed language. Then $\mathsf{Det}^{\mathsf{d}}(\mathcal{A}) \equiv \mathsf{Min}^{\mathsf{d}}(L)$ iff $\forall q \in Q\colon P_{\sim_L^{\mathsf{d}}}(\mathcal{L}_{\uparrow}(q)) = P_{\sim_{\frac{\mathsf{d}}{\mathsf{i}}}}(\mathcal{L}_{\uparrow}(q))$.

Proof:

First, we will show that if $\mathsf{Det}^{\mathsf{d}}(\mathcal{A}) \equiv \mathsf{Min}^{\mathsf{d}}(\mathcal{A})$ then $P_{\sim_L^{\mathsf{d}}}(\mathcal{L}_\uparrow(q)) = P_{\sim_\wr}(\mathcal{L}_\uparrow(q)), \forall q \in Q$.

$$P_{\sim_L^{\mathsf{d}}}(\mathcal{L}_\uparrow(q)) =$$

$$\{x \in \mathcal{C}_\Sigma \mid \exists y \in \mathcal{L}_\uparrow(q) \colon y^{-1}L = x^{-1}L\} = \quad [\sim_L^{\mathsf{d}} = \sim_{\mathcal{A}}^{\mathsf{d}}]$$

$$\{x \in \mathcal{C}_\Sigma \mid \exists y \in \mathcal{L}_\uparrow(q) \colon \mathrm{pre}_y^{\mathcal{A}}(F) = \mathrm{pre}_x^{\mathcal{A}}(F)\} \ .$$

Note that $\sim_L^{\mathsf{d}} = \sim_{\mathcal{A}}^{\mathsf{d}}$ by hypothesis, since $\mathsf{Det}^{\mathsf{d}}(\mathcal{A}) \equiv \mathsf{Min}^{\mathsf{d}}(L)$. On the other hand, since $y \in \mathcal{L}_\uparrow(q) \Rightarrow q \in \mathrm{pre}_y^{\mathcal{A}}(F)$, we have the following set inclusion:

$$\{x \in \mathcal{C}_\Sigma \mid \exists y \in \mathcal{L}_\uparrow(q) \colon \mathrm{pre}_y^{\mathcal{A}}(F) = \mathrm{pre}_x^{\mathcal{A}}(F)\} \subseteq$$

$$\{x \in \mathcal{C}_\Sigma \mid q \in \mathrm{pre}_x^{\mathcal{A}}(F)\} =$$

$$P_{\sim_\wr}(\mathcal{L}_\uparrow(q)) \ .$$

Since L is path-closed and $\mathcal{A}$ has no unreachable states, by Lemma 4.7, $x \sim_\wr y \Rightarrow x^{-1}L = y^{-1}L$, for every $x, y \in \mathcal{C}_\Sigma$. Therefore, $P_{\sim_\wr}(\mathcal{L}_\uparrow(q)) \subseteq P_{\sim_L^{\mathsf{d}}}(\mathcal{L}_\uparrow(q))$, hence we conclude that $P_{\sim_L^{\mathsf{d}}}(\mathcal{L}_\uparrow(q)) = P_{\sim_\wr}(\mathcal{L}_\uparrow(q))$.

Now, we will prove that if $P_{\sim_L^{\mathsf{d}}}(\mathcal{L}_\uparrow(q)) = P_{\sim_\wr}(\mathcal{L}_\uparrow(q))$, for each $q \in Q$, then $\mathsf{Det}^{\mathsf{d}}(\mathcal{A}) \equiv \mathsf{Min}^{\mathsf{d}}(L)$. For every $x \in \mathcal{C}_\Sigma$:

$$P_{\sim_{\mathcal{A}}^{\mathsf{d}}}(x) =^{\dagger}$$

$$\bigcap_{q \in \mathrm{pre}_x^{\mathcal{A}}(F)} P_{\sim_\wr}(\mathcal{L}_\uparrow(q)) \cap \bigcap_{q \notin \mathrm{pre}_x^{\mathcal{A}}(F)} (P_{\sim_\wr}(\mathcal{L}_\uparrow(q)))^{\complement} =^{\dagger\dagger}$$

$$\bigcap_{q \in \mathrm{pre}_x^{\mathcal{A}}(F)} P_{\sim_L^{\mathsf{d}}}(\mathcal{L}_\uparrow(q)) \cap \bigcap_{q \notin \mathrm{pre}_x^{\mathcal{A}}(F)} (P_{\sim_L^{\mathsf{d}}}(\mathcal{L}_\uparrow(q)))^{\complement} \ . \tag{16}$$

Note that equality $\dagger$ holds by Lemma 4.16 and $\dagger\dagger$ holds since $P_{\sim_\wr}(\mathcal{L}_\uparrow(q)) = P_{\sim_L^{\mathsf{d}}}(\mathcal{L}_\uparrow(q))$ by hypothesis.

It follows from (16) that $P_{\sim_{\mathcal{A}}^{\mathsf{d}}}(x)$ is a *union* of blocks of $P_{\sim_L^{\mathsf{d}}}$. In other words, $x \sim_L^{\mathsf{d}} y \Rightarrow x \sim_{\mathcal{A}}^{\mathsf{d}} y$, for every $x, y \in \mathcal{C}_\Sigma$. By Lemma 4.13(b), we have that $x \sim_{\mathcal{A}}^{\mathsf{d}} y \Rightarrow x \sim_L^{\mathsf{d}} y$. Thus, $P_{\sim_{\mathcal{A}}^{\mathsf{d}}}(x)$ necessarily corresponds to one single block of $\sim_L^{\mathsf{d}}$, namely, $P_{\sim_L^{\mathsf{d}}}(x)$. Since $P_{\sim_{\mathcal{A}}^{\mathsf{d}}}(x) = P_{\sim_L^{\mathsf{d}}}(x)$, for each $x \in \mathcal{C}_\Sigma$, we conclude that $\mathsf{Det}^{\mathsf{d}}(\mathcal{A}) \equiv \mathsf{Min}^{\mathsf{d}}(L)$. $\qquad\square$

Corollary 5.6. Let $\mathcal{A}$ be a BTA with $L = \mathcal{L}(\mathcal{A})$. Then $(\mathcal{A})^{\mathsf{cD}} \equiv \mathsf{Min}^{\mathsf{d}}(L)$ iff the set of contexts root-to-pivot equivalent to the upward language of every state in $\mathcal{A}$ is a union of upward languages of the minimal co-DBTA for L.

It is worth pointing that the conditions of Corollary 5.6 and Theorem 5.5 are decidable since isomorphism is decidable, $\mathsf{Det}^{\mathsf{d}}(\mathcal{A})$ and $(\mathcal{A})^{\mathsf{cD}}$ are effectively computable and so is $\mathsf{Min}^{\mathsf{d}}(L)$ [20, § 2.11].

6. Related work and conclusions

In this paper, we build on previous work on word automata [16] and present a congruence-based perspective on the determinization and minimization operations for bottom-up (and top-down, see Appendix A.1) tree automata. As a consequence, we obtain the first, to the best of our knowledge, proof of correctness of the double-reversal method for BTAs. Björklund and Cleophas, in their taxonomy of minimization algorithms for tree automata [12], proposed a double-reversal method for DBTAs. They observed that the reverse operation is embedded within the notion of top-down and bottom-up determinization although they did not include a proof of correctness of the algorithm.

Courcelle et al. [21] also studied the problem of determinizing and minimizing word and tree automata by offering a geometrical and general view on these operations. Roughly speaking, our framework is an instantiation of theirs using a concrete decomposition of their binary relations (namely, deterministic and co-deterministic decompositions). However, they focus on the so-called *lr-determinism* of BTAs, which is a relaxed version of our notion of *co-determinism* for BTAs that is defined in terms of the downward languages of the states of the BTA instead of being a purely syntactic notion, as our definition. As a consequence, the class of languages that are *lr-determinizable*, i.e., the so-called *homogeneous languages*, includes the path-closed languages. While theirs is a more general setting, our framework is constructive, in the sense that, it is defined upon congruences that allows us to extract automata constructions.

We also give a generalization of the double-reversal method in the same lines as the generalization of Brzozowski and Tamm [18] for the case of word automata, which further evidences the connection between congruences and determinization of automata. Note that the double-reversal method only applies to path-closed languages since it requires a co-determinization step, which is only possible for that class of languages. However, the generalized double-reversal method drops this restriction since, for every regular language, its minimal DBTA is already an automaton satisfying the condition of the generalized double-reversal method.

Figure 2 summarizes the relations between these tree automata constructions. Note that it includes the counterpart TTA congruence-based constructions (TDet^d, TDet^u, TMin^d and TMin^u) whose definition we deferred to the Appendix.

As a final note, we show how previous results on word automata follow from our results when we only consider monadic trees. First, note that, in the monadic case:

 (i) every tree language is path-closed;

 (ii) every downward congruence is strongly downward w.r.t. a given tree language, and

(iii) the notion of root-to-pivot equivalence between contexts collapses to the standard notion of equality.

A consequence of the latter is that our definition of $\text{pre}(\cdot)$ for tree automata coincides with the standard one for word automata. As a result of these observations, in the monadic case, Corollary 5.1 (Brzozowski's double-reversal method for BTAs) collapses to the counterpart result for word automata [15]. Finally, Theorem 5.3 (generalization of the double-reversal method) also collapses to the counterpart generalization for word automata given by Brzozowski and Tamm [18], later addressed in [16].
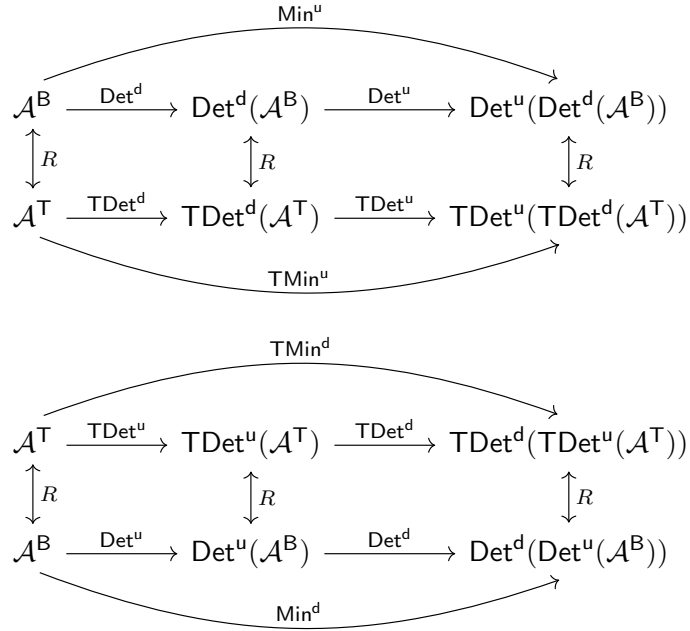
Figure 2. Relations between the automata constructions $\mathsf{Det}^d, \mathsf{Det}^u, \mathsf{TDet}^d, \mathsf{TDet}^u, \mathsf{Min}^d, \mathsf{Min}^u, \mathsf{TMin}^d$ and TMin^u. Note that constructions $\mathsf{Min}^d, \mathsf{Min}^u, \mathsf{TMin}^d$ and TMin^u are applied to the language defined by the automaton in the origin of the labeled arrow, while the others are applied directly to the automaton. The upper arcs of the diagrams follow from Theorem 4.18(g) and Theorem A.12(h). The squares and the bottom arc follow from Corollary A.11, since $\mathcal{A}^\mathsf{T} = (\mathcal{A}^\mathsf{B})^R$. Incidentally, the diagram shows a new relation which follows from the definition of $\mathsf{H}^{uR}, \mathsf{H}^{dR}$ and the fact that $\mathcal{A}^\mathsf{T} = (\mathcal{A}^\mathsf{B})^R$: $\mathsf{TDet}^d(\mathsf{TDet}^d(\mathcal{A}^\mathsf{T})) \equiv \mathsf{TMin}^u(\mathcal{A}^\mathsf{T})$, the minimal co-deterministic TTA for $\mathcal{L}(\mathcal{A}^\mathsf{T})$.

6.1. Future work

We have not considered yet how to build an automaton that satisfies the condition of Theorem 5.3. In particular, it is worth considering whether lr-deterministic automata satisfy this condition.

On the other hand, Ganty et al. [22] showed that quasiorders, i.e., reflexive and transitive binary relations, are related to residual word automata in the same way congruences are related to deterministic word automata. We believe that relaxing the equivalences presented in this work to obtain quasiorders would allow us to offer a new perspective on residual tree automata, defined by Carme et al. [23].

Acknowledgments

References

[1] Comon H, Dauchet M, Gilleron R, Jacquemard F, Lugiez D, Löding C, Tison S, Tommasi M. Tree Automata Techniques and Applications. Available on: `http://www.grappa.univ-lille3.fr/tata`, 2008. Release November, 18th 2008.

[2] Gécseg F, Steinby M. Minimal ascending tree automata. *Acta Cybern.*, 1978. **4**(1):37–44.

[3] Gécseg F, Steinby M. Tree Automata. Akadéniai Kiadó, Budapest, Hungary, 1984.

[4] Abdulla PA, Legay A, d'Orso J, Rezine A. Tree regular model checking: A simulation-based approach. *J. Log. Algebraic Methods Program.*, 2006. **69**(1-2):93–121. doi:10.1016/j.jlap.2006.02.001.

[5] Bouajjani A, Habermehl P, Rogalewicz A, Vojnar T. Abstract regular (tree) model checking. *Int. J. Softw. Tools Technol. Transf.*, 2012. **14**(2):167–191. doi:10.1007/s10009-011-0205-y.

[6] Knight K, May J. Applications of weighted automata in natural language processing. In: Handbook of Weighted Automata, pp. 571–596. Springer, 2009. doi:10.1007 / 978-3-642-01492-5_14.

[7] Hosoya H. Foundations of XML processing: the tree-automata approach. Cambridge University Press, 2010. doi:10.1017 / CBO9780511762093.

[8] Abdulla PA, Högberg J, Kaati L. Bisimulation Minimization of Tree Automata. *Int. J. Found. Comput. Sci.*, 2007. **18**(4):699–713. doi:10.1142/S0129054107004929.

[9] Almeida R, Holík L, Mayr R. Reduction of Nondeterministic Tree Automata. In: TACAS, volume 9636 of *Lecture Notes in Computer Science*. Springer, 2016 pp. 717–735. doi:10.1007/978-3-662-49674-9_46.

[10] Högberg J, Maletti A, May J. Backward and forward bisimulation minimization of tree automata. *Theor. Comput. Sci.*, 2009. **410**(37):3539–3552. doi:10.1016/j.tcs.2009.03.022.

[11] Brainerd WS. The Minimalization of Tree Automata. *Inf. Control.*, 1968. **13**(5):484–491.

[12] Björklund J, Cleophas L. A Taxonomy of Minimisation Algorithms for Deterministic Tree Automata. *J. UCS*, 2016. **22**(2):180–196. ISSN:0948-695X.

[13] Nivat M, Podelski A. Minimal Ascending and Descending Tree Automata. *SIAM J. Comput.*, 1997. **26**(1):39–58.

[14] Virágh J. Deterministic ascending tree automata I. *Acta Cybern.*, 1980. **5**(1):33–42.

[15] Brzozowski JA. Canonical regular expressions and minimal state graphs for definite events. *Mathematical Theory of Automata*, 1962. **12**(6):529–561. ID:118363215.

[16] Ganty P, Gutiérrez E, Valero P. A Congruence-based Perspective on Automata Minimization Algorithms. In: MFCS, volume 138 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019 pp. 77:1–77:14.

[17] Kozen D. On the Myhill-Nerode theorem for trees. *Bull. EATCS*, 1992. **47**:170–173.

[18] Brzozowski JA, Tamm H. Theory of átomata. *Theor. Comput. Sci.*, 2014. **539**:13–27. doi:10.1016/j.tcs.2014.04.016.

[19] Cleophas L. Tree algorithms: two taxonomies and a toolkit. Ph.D. thesis, Department of Mathematics and Computer Science, 2008. doi:10.6100/IR633481.

[20] Gécseg F, Steinby M. Tree Automata, 2015. `1509.06233`.

[21] Courcelle B, Niwinski D, Podelski A. A Geometrical View of the Determinization and Minimization of Finite-State Automata. *Math. Syst. Theory*, 1991. **24**(2):117–146. doi:10.1007/BF02090394.

[22] Ganty P, Gutiérrez E, Valero P. A Quasiorder-Based Perspective on Residual Automata. In: MFCS, volume 170 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020 pp. 40:1–40:14. arXiv:2007.00359 [cs.FL].

[23] Carme J, Gilleron R, Lemay A, Terlutte A, Tommasi M. Residual Finite Tree Automata. In: Developments in Language Theory, volume 2710 of *Lecture Notes in Computer Science*. Springer, 2003 pp. 171–182. doi:10.1007 / 3-540-45007-6_13.

A. Appendix

A.1. Top-down tree automata as reverse of bottom-up tree automata

In the main part of the document, we focus on bottom-up tree automata. This decision is motivated by the fact that deterministic BTAs are strictly more expressive than their top-down counterparts. On the other hand, TTAs can be interpreted as the reverse of BTAs, and thus we can easily extend the results obtained for bottom-up tree automata to their top-down counterparts. This will be the goal of this section.

Our ultimate purpose is to give a complete perspective on Brzozowski's method to minimize BTAs as a technique that combines a reverse operation followed by determinization operation twice (see Figure 2). As a product, we obtain a counterpart method for the minimization of TTAs. In the following, we use $\mathcal{A}^{\mathsf{B}}$ to denote a BTA and $\mathcal{A}^{\mathsf{T}}$ for a TTA.

A.1.1. Top-down tree automata

Definition A.1. (Top-down tree automaton)
A *top-down tree automaton* (TTA for short) is a tuple $\mathcal{A}^{\mathsf{T}} = \langle Q, \Sigma, \delta, I \rangle$ where Q is a finite set of *states*; Σ is a ranked alphabet of rank n; $\delta : Q \to \wp(\bigcup_{i=0}^{n} \Sigma_i \times Q^i)$ is the *transition function* and $I \subseteq Q$ is the set of *initial states*. Given a TTA $\mathcal{A}^{\mathsf{T}}$, we define its set of *final states* as $\mathrm{f}(\mathcal{A}^{\mathsf{T}}) \overset{\text{def}}{=} \{q \in Q \mid \exists a \in \Sigma_0 \colon a[] \in \delta(q)\}$.

A TTA is *deterministic* (DTTA for short) iff I is a singleton and for every state $q \in Q$ and symbol $f \in \Sigma_n$, with $n \geq 1$, we have: if $f[q_1, \ldots, q_{\langle f \rangle}] \in \delta(q)$ and $f[q'_1, \ldots, q'_{\langle f \rangle}] \in \delta(q)$ then $q_i = q'_i$ for each $i = 1..\langle f \rangle$. Similarly, a TTA is *co-deterministic* (co-DTTA for short) iff every set of states in the image of δ^{-1} is a singleton or is empty.

We define the *move* relation on a TTA, denoted by $\to_{\mathcal{A}^{\mathsf{T}}} \in \mathcal{T}_{\Sigma \cup Q} \times \mathcal{T}_{\Sigma \cup Q}$, as follows. Let $t = x[\![q[t_1, \ldots, t_{\langle f \rangle}]]\!]$ and $t' = x[\![f[t_1, \ldots, t_{\langle f \rangle}]]\!]$ for some $x \in \mathcal{C}_{\Sigma \cup Q}, q \in Q, f \in \Sigma$ and $t_1, \ldots, t_{\langle f \rangle} \in \mathcal{T}_Q$. Then $t \to_{\mathcal{A}^{\mathsf{T}}} t' \overset{\text{def}}{\Leftrightarrow} f[t_1(\varepsilon), \ldots, t_{\langle f \rangle}(\varepsilon)] \in \delta(q)$. We use $\to^*_{\mathcal{A}^{\mathsf{T}}}$ to denote the transitive closure of $\to_{\mathcal{A}^{\mathsf{T}}}$. The *language* defined by $\mathcal{A}^{\mathsf{T}}$ is $\mathcal{L}(\mathcal{A}^{\mathsf{T}}) \overset{\text{def}}{=} \{t \in \mathcal{T}_\Sigma \mid \exists t' \in \mathcal{T}_Q \colon t'(\varepsilon) \in I, t' \to^*_{\mathcal{A}^{\mathsf{T}}} t\}$. The language definition might seem counterintuitive because it starts with a tree t' in $\mathcal{T}_Q$ and ends up with a tree t in $\mathcal{T}_\Sigma$ which is the one "recognized" by the TTA. Intuitively, a run of a TTA accepting a tree $t \in \mathcal{T}_\Sigma$ first "guesses" for each node $v \in \mathrm{dom}(t)$ a state labelling v and then tries to reinstate the original labels in

a top down fashion using the move relation. We made this unusual choice to maintain coherence with the move relation for BTA. In the example below we depicted such a sequence of moves from right to left because we think it is easier for the reader to parse the sequence of moves backwards.

Example A.2. Let $\mathcal{A}^{\mathsf{T}} = \langle Q, \Sigma_0 \cup \Sigma_2, \delta, I \rangle$ be a TTA with $Q = \{q_0, q_1\}$, $\Sigma_0 = \{\mathbf{T}, \mathbf{F}\}$, $\Sigma_2 = \{\wedge, \vee\}$, $I = \{q_1\}$, and

$$\delta(q_0) = \{\wedge[q_0, q_1], \wedge[q_0, q_0], \wedge[q_1, q_0], \vee[q_0, q_0], \mathbf{F}[\,]\},$$
$$\delta(q_1) = \{\wedge[q_1, q_1], \vee[q_1, q_0], \vee[q_0, q_1], \vee[q_1, q_1], \mathbf{T}[\,]\} \ .$$

As in Example 2.5, $\mathcal{L}(\mathcal{A}^{\mathsf{T}})$ is defined as the set of all trees of the form $t \in \mathcal{T}_{\Sigma_0 \cup \Sigma_2}$ which yield to propositional formulas, over the binary connectives $\wedge$ and $\vee$ and the constants $\mathbf{T}$ and $\mathbf{F}$, that evaluate to $\mathbf{T}$. For instance, the following is a sequence of moves accepting a tree $t \in \mathcal{T}_{\Sigma_0 \cup \Sigma_2}$ (listed last below) such that $t \in \mathcal{L}(\mathcal{A})$.

Note that, $\vee[\square, q_1][\![q_1[q_1, q_0]]\!] \rightarrow_{\mathcal{A}^{\mathsf{T}}} \vee[\square, q_1][\![\vee[q_1, q_0]]\!]$ as $\vee[q_1, q_0] \in \delta(q_1)$. Observe that $\mathcal{A}^{\mathsf{T}}$ is co-deterministic, but not deterministic since $\wedge[q_0, q_1], \wedge[q_1, q_0] \in \delta(q_0)$. $\diamondsuit$

For each $q \in Q$ and $S \subseteq Q$, define the *upward* and *downward language* of q, respectively, as follows:

$$\mathcal{L}_{\uparrow}^{\mathcal{A}^{\mathsf{T}}}(q, S) \stackrel{\text{def}}{=} \{c \in \mathcal{C}_{\Sigma} \mid \exists t \in \mathcal{T}_Q \colon t \rightarrow_{\mathcal{A}^{\mathsf{T}}}^{*} c[\![q]\!], t(\varepsilon) \in S\}$$
$$\mathcal{L}_{\downarrow}^{\mathcal{A}^{\mathsf{T}}}(q, S) \stackrel{\text{def}}{=} \{t \in \mathcal{T}_{\Sigma} \mid \exists t' \in \mathcal{T}_Q \colon t' \rightarrow_{\mathcal{A}^{\mathsf{T}}}^{*} t, t'(\varepsilon) = q, \ell(t') \subseteq S\} \ .$$

We will simplify the notation and write $\mathcal{L}_{\uparrow}^{\mathcal{A}^{\mathsf{T}}}(q)$ when $S = I$ and $\mathcal{L}_{\downarrow}^{\mathcal{A}^{\mathsf{T}}}(q)$ when $S = \mathrm{f}(\mathcal{A}^{\mathsf{T}})$. Also, we will drop the superscript $\mathcal{A}^{\mathsf{T}}$ when the TTA $\mathcal{A}^{\mathsf{T}}$ is clear from the context.

A state $q \in Q$ of a TTA is *unreachable* (resp. *empty*) iff its upward (resp. downward) language is empty. It is straightforward to check that for every TTA $\mathcal{A}^{\mathsf{T}}$ we have that $\mathcal{L}(\mathcal{A}^{\mathsf{T}}) = \bigcup_{q \in I} \mathcal{L}_{\downarrow}(q)$.

Given a TTA $\mathcal{A}^{\mathsf{T}} = \langle Q, \Sigma, \delta, I \rangle$ without empty states, the *top-down determinization* [19] builds the DTTA $\langle \wp(Q), \Sigma, \delta', \{I\} \rangle$ where the transition function δ' is defined as follows. For each $f \in \Sigma \setminus \Sigma_0$ and $R \in \wp(Q)$, we have that $f[R_1, \ldots, R_{\langle f \rangle}] \in \delta'(R)$, where $R_i \stackrel{\text{def}}{=} \{q_i \in Q \mid \exists q \in R, q_1, \ldots, q_{i-1}, q_{i+1}, \ldots, q_{\langle f \rangle} \in Q \colon f[q_1, \ldots, q_i, \ldots, q_{\langle f \rangle}] \in \delta(q)\}$. On the other hand, for every $a \in \Sigma_0$ and $R \in \wp(Q)$ such that $\exists q \in R \colon a \in \delta(q)$, we have that $a[\,] \in \delta'(R)$. We abuse notation (w.r.t. the notation introduced for BTAs) and denote by $(\mathcal{A}^{\mathsf{T}})^{\mathsf{D}}$ the result of applying the top-down determinization to $\mathcal{A}^{\mathsf{T}}$.

As shown by Cleophas [19], the automaton $(\mathcal{A}^{\mathsf{T}})^{\mathsf{D}}$ is top-down deterministic and, whenever $\mathcal{L}(\mathcal{A}^{\mathsf{T}})$ is path-closed, $\mathcal{L}(\mathcal{A}^{\mathsf{T}}) = \mathcal{L}((\mathcal{A}^{\mathsf{T}})^{\mathsf{D}})$.

A.1.2. Reverse tree automata constructions

We define the *reverse TTA* of a BTA $\mathcal{A}^{\mathsf{B}} = \langle Q, \Sigma, \delta, F \rangle$ as $(\mathcal{A}^{\mathsf{B}})^R \stackrel{\text{def}}{=} \langle Q, \Sigma, \delta_r, I \rangle$, where $f[q_1, \ldots, q_{\langle f \rangle}] \in \delta_r(q) \stackrel{\text{def}}{\Leftrightarrow} q \in \delta(f[q_1, \ldots, q_{\langle f \rangle}])$ and $I \stackrel{\text{def}}{=} F$. Analogously, we define the *reverse BTA* of a TTA

$\mathcal{A}^{\mathsf{T}} = \langle Q, \Sigma, \delta, I \rangle$ as $(\mathcal{A}^{\mathsf{T}})^R \stackrel{\text{def}}{=} \langle Q, \Sigma, \delta_r, F \rangle$, where $q \in \delta_r(f[q_1, \ldots, q_{\langle f \rangle}]) \stackrel{\text{def}}{\Leftrightarrow} f[q_1, \ldots, q_{\langle f \rangle}] \in \delta(q)$ and $F \stackrel{\text{def}}{=} I$. Observe that the BTA and TTA shown in Examples 2.5 and A.2 are the reverse of each other.

Let $\mathcal{A}^{\mathsf{B}}$ be a BTA and let $\mathcal{A}^{\mathsf{T}}$ be its reverse, i.e., $\mathcal{A}^{\mathsf{T}} \stackrel{\text{def}}{=} (\mathcal{A}^{\mathsf{B}})^R$. It is easy to check that $\mathcal{L}(\mathcal{A}^{\mathsf{B}}) = \mathcal{L}(\mathcal{A}^{\mathsf{T}})$ and $\mathcal{A}^{\mathsf{B}}$ is a co-DBTA (resp. DBTA) iff $\mathcal{A}^{\mathsf{T}}$ is a DTTA (resp. co-DTTA). Moreover, we have that $(\mathcal{A}^{\mathsf{B}})^{\mathsf{cD}} \equiv (\mathcal{A}^{\mathsf{T}})^{\mathsf{D}}$ and a state of $\mathcal{A}^{\mathsf{B}}$ is unreachable (resp. empty) iff it is empty (resp. unreachable) in $\mathcal{A}^{\mathsf{T}}$.

Note that we can define the TTA-equivalent of the constructions H^{u} and H^{d} simply by reversing their transition functions and setting their final states as the initial ones.

Definition A.3. Let $L \subseteq \mathcal{T}_\Sigma$, and $\sim^{\mathsf{u}}$ and $\sim^{\mathsf{d}}$ be an upward and downward congruence, respectively. Define:

$$\mathsf{H}^{\mathsf{uR}} \stackrel{\text{def}}{=} \mathsf{H}^{\mathsf{u}}(\sim^{\mathsf{u}}, L))^R \qquad \mathsf{H}^{\mathsf{dR}} \stackrel{\text{def}}{=} (\mathsf{H}^{\mathsf{d}}(\sim^{\mathsf{d}}, L))^R$$

Clearly, H^{uR} yields a co-DTTA iff H^{u} yields a DBTA, and H^{dR} yields a DTTA iff H^{d} yields a co-DBTA. Thus, the following result is a consequence of Lemma 3.3, and the one after is a consequence of Lemmas A.4 and 3.9.

Corollary A.4. Let $L \subseteq \mathcal{T}_\Sigma$ be a tree language and let $\sim^{\mathsf{u}}$ be an upward congruence such that $t \sim^{\mathsf{u}} r \Rightarrow L\,t^{-1} = L\,r^{-1}$ for every $t, r \in \mathcal{T}_\Sigma$. Then, $\mathsf{H}^{\mathsf{uR}}(\sim^{\mathsf{u}}, L)$ is a co-DTTA with $\mathcal{L}(\mathsf{H}^{\mathsf{uR}}(\sim^{\mathsf{u}}, L)) = L$.

Corollary A.5. Let $L \subseteq \mathcal{T}_\Sigma$ be a path-closed language and let $\sim^{\mathsf{d}}$ be a strongly downward congruence w.r.t. L such that $x \sim^{\mathsf{d}} y \Rightarrow x^{-1}L = y^{-1}L$ for every $x, y \in \mathcal{C}_\Sigma$. Then, $\mathsf{H}^{\mathsf{dR}}(\sim^{\mathsf{d}}, L)$ is a DTTA with $\mathcal{L}(\mathsf{H}^{\mathsf{dR}}(\sim^{\mathsf{d}}, L)) = L$.

Next, we define congruences based on the states of a given TTA. These TTA-based congruences are finer than (or equal to) the corresponding language-based ones and are thus said to *approximate* the language-based congruences. To that end, we first define the $\mathrm{post}(\cdot)$ and $\mathrm{pre}(\cdot)$ operators for TTAs.

Definition A.6. Let $\mathcal{A}^{\mathsf{T}} = \langle Q, \Sigma, \delta, I \rangle$ be a TTA and let $t \in \mathcal{T}_\Sigma$, $x \in \mathcal{C}_\Sigma$ and $S \subseteq Q$. Define:

$$\mathrm{post}_x^{\mathcal{A}^{\mathsf{T}}}(S) \stackrel{\text{def}}{=} \{q \in Q \mid x \in P_{\sim_{\updownarrow}^S}(\mathcal{L}_{\uparrow}^{\mathcal{A}^{\mathsf{T}}}(q, S))\}$$

$$\mathrm{pre}_t^{\mathcal{A}^{\mathsf{T}}}(S) \stackrel{\text{def}}{=} \{q \in Q \mid t \in \mathcal{L}_{\downarrow}^{\mathcal{A}^{\mathsf{T}}}(q, S)\} \ .$$

Note that we will omit the superscript $\mathcal{A}^{\mathsf{T}}$ when it is clear from the context.

Definition A.7. Let $\mathcal{A}^{\mathsf{T}} = \langle Q, \Sigma, \delta, I \rangle$ be a TTA and let $t, r \in \mathcal{T}_\Sigma$ and $x, y \in \mathcal{C}_\Sigma$. The *upward and downward TTA-based equivalences* are respectively given by:

$$t \sim_{\mathcal{A}^{\mathsf{T}}}^{\mathsf{u}} r \stackrel{\text{def}}{\Leftrightarrow} \mathrm{pre}_t(\mathsf{f}(\mathcal{A}^{\mathsf{T}})) = \mathrm{pre}_r(\mathsf{f}(\mathcal{A}^{\mathsf{T}}))$$

$$x \sim_{\mathcal{A}^{\mathsf{T}}}^{\mathsf{d}} y \stackrel{\text{def}}{\Leftrightarrow} \mathrm{post}_x(I) = \mathrm{post}_y(I) \ .$$

Lemma A.8. Let $\mathcal{A}^\mathsf{B}$ be a BTA and $\mathcal{A}^\mathsf{T} \overset{\text{def}}{=} (\mathcal{A}^\mathsf{B})^R$. Then, the following hold:

(a) $\sim^\mathsf{u}_{\mathcal{L}(\mathcal{A}^\mathsf{B})} = \sim^\mathsf{u}_{\mathcal{L}(\mathcal{A}^\mathsf{T})}$ and $\sim^\mathsf{d}_{\mathcal{L}(\mathcal{A}^\mathsf{B})} = \sim^\mathsf{d}_{\mathcal{L}(\mathcal{A}^\mathsf{T})}$.

(b) $\sim^\mathsf{u}_{\mathcal{A}^\mathsf{B}} = \sim^\mathsf{u}_{\mathcal{A}^\mathsf{T}}$ and $\sim^\mathsf{d}_{\mathcal{A}^\mathsf{B}} = \sim^\mathsf{d}_{\mathcal{A}^\mathsf{T}}$.

Proof:
Let $\mathcal{A}^\mathsf{B} = \langle Q, \Sigma, \delta, F \rangle$ be a BTA with $\mathcal{A}^\mathsf{T} = (\mathcal{A}^\mathsf{B})^R$.

(a) $\sim^\mathsf{u}_{\mathcal{L}(\mathcal{A}^\mathsf{B})} = \sim^\mathsf{u}_{\mathcal{L}(\mathcal{A}^\mathsf{T})}$ and $\sim^\mathsf{d}_{\mathcal{L}(\mathcal{A}^\mathsf{B})} = \sim^\mathsf{d}_{\mathcal{L}(\mathcal{A}^\mathsf{T})}$.

Trivial, since $\mathcal{L}(\mathcal{A}^\mathsf{B}) = \mathcal{L}(\mathcal{A}^\mathsf{T})$.

(b) $\sim^\mathsf{u}_{\mathcal{A}^\mathsf{B}} = \sim^\mathsf{u}_{\mathcal{A}^\mathsf{T}}$ and $\sim^\mathsf{d}_{\mathcal{A}^\mathsf{B}} = \sim^\mathsf{d}_{\mathcal{A}^\mathsf{T}}$.

Relying on Definitions 4.5 and A.6, it is easy to see that, for every $t \in \mathcal{T}_\Sigma, x \in \mathcal{C}_\Sigma$ and $S \subseteq Q$: $\text{post}_t^{\mathcal{A}^\mathsf{B}}(S) = \text{pre}_t^{\mathcal{A}^\mathsf{T}}(S)$ and $\text{post}_x^{\mathcal{A}^\mathsf{T}}(S) = \text{pre}_x^{\mathcal{A}^\mathsf{B}}(S)$. Thus, by Definitions 4.12 and A.7, $\sim^\mathsf{u}_{\mathcal{A}^\mathsf{B}} = \sim^\mathsf{u}_{\mathcal{A}^\mathsf{T}}$ and $\sim^\mathsf{d}_{\mathcal{A}^\mathsf{B}} = \sim^\mathsf{d}_{\mathcal{A}^\mathsf{T}}$.   □

As a consequence of Lemma 4.13 and A.8(b), we have the following corollary.

Corollary A.9. Let $\mathcal{A}^\mathsf{T}$ be a TTA with $L = \mathcal{L}(\mathcal{A}^\mathsf{T})$. Then,

(a) $\sim^\mathsf{u}_{\mathcal{A}^\mathsf{T}}$ is an upward congruence and $\sim^\mathsf{u}_{\mathcal{A}^\mathsf{T}} \subseteq \sim^\mathsf{u}_L$.

(b) If $\mathcal{A}$ has no empty states and L is path-closed then $\sim^\mathsf{d}_{\mathcal{A}^\mathsf{T}}$ is a strongly downward congruence w.r.t. L and $\sim^\mathsf{d}_{\mathcal{A}^\mathsf{T}} \subseteq \sim^\mathsf{d}_L$.

A.1.3.   Determinization and minimization of TTAs using congruences

We can now define the TTA-equivalents of the constructions from Definition 4.17.

Definition A.10. Let $\mathcal{A}^\mathsf{T}$ be a TTA with $L = \mathcal{L}(\mathcal{A}^\mathsf{T})$. Define:

$$\mathsf{TDet}^\mathsf{u}(\mathcal{A}^\mathsf{T}) \overset{\text{def}}{=} \mathsf{H}^\mathsf{uR}(\sim^\mathsf{u}_{\mathcal{A}^\mathsf{T}}, \mathcal{A}^\mathsf{T}) \qquad\qquad \mathsf{TMin}^\mathsf{u}(L) \overset{\text{def}}{=} \mathsf{H}^\mathsf{uR}(\sim^\mathsf{u}_L, L)$$
$$\mathsf{TDet}^\mathsf{d}(\mathcal{A}^\mathsf{T}) \overset{\text{def}}{=} \mathsf{H}^\mathsf{dR}(\sim^\mathsf{d}_{\mathcal{A}^\mathsf{T}}, \mathcal{A}^\mathsf{T}) \qquad\qquad \mathsf{TMin}^\mathsf{d}(L) \overset{\text{def}}{=} \mathsf{H}^\mathsf{dR}(\sim^\mathsf{d}_L, L) \ .$$

It follows from Lemma A.8 that the constructions $\mathsf{TDet}^\mathsf{u}(\mathcal{A}^\mathsf{T})$ and $\mathsf{TDet}^\mathsf{d}(\mathcal{A}^\mathsf{T})$ are related to $\mathsf{Det}^\mathsf{u}(\mathcal{A}^\mathsf{B})$ and $\mathsf{Det}^\mathsf{d}(\mathcal{A}^\mathsf{B})$, respectively, through the reverse construction. The same holds for the constructions $\mathsf{TMin}^\mathsf{u}(L)$, $\mathsf{TMin}^\mathsf{d}(L)$ and $\mathsf{Min}^\mathsf{u}(L)$, $\mathsf{Min}^\mathsf{d}(L)$, respectively.

Corollary A.11. Let $\mathcal{A}^\mathsf{B}$ be a BTA and $\mathcal{A}^\mathsf{T} \overset{\text{def}}{=} (\mathcal{A}^\mathsf{B})^R$ with $L = \mathcal{L}(\mathcal{A}^\mathsf{T}) = \mathcal{L}(\mathcal{A}^\mathsf{B})$. Then,

$$\mathsf{TDet}^\mathsf{u}(\mathcal{A}^\mathsf{T}) \equiv (\mathsf{Det}^\mathsf{u}(\mathcal{A}^\mathsf{B}))^R \qquad\qquad \mathsf{TMin}^\mathsf{u}(L) \equiv (\mathsf{Min}^\mathsf{u}(L))^R$$
$$\mathsf{TDet}^\mathsf{d}(\mathcal{A}^\mathsf{T}) \equiv (\mathsf{Det}^\mathsf{d}(\mathcal{A}^\mathsf{B}))^R \qquad\qquad \mathsf{TMin}^\mathsf{d}(L) \equiv (\mathsf{Min}^\mathsf{d}(L))^R \ .$$

Finally, we obtain the following result as the TTA-equivalent of Theorem 4.18.

Corollary A.12. Let $\mathcal{A}^\mathsf{T}$ be a TTA with $L = \mathcal{L}(\mathcal{A}^\mathsf{T})$. Then the following properties hold:

(a) $\mathcal{L}(\mathsf{TMin}^\mathsf{u}(L)) = L = \mathcal{L}(\mathsf{TDet}^\mathsf{u}(\mathcal{A}^\mathsf{T}))$.

(b) If L is path-closed then $\mathcal{L}(\mathsf{TMin}^\mathsf{d}(L)) = L = \mathcal{L}(\mathsf{TDet}^\mathsf{d}(\mathcal{A}^\mathsf{T}))$.

(c) $\mathsf{TDet}^\mathsf{u}(\mathcal{A}^\mathsf{T}) \equiv (\mathcal{A}^\mathsf{T})^{\mathsf{cD}}$.

(d) If L is path-closed and $\mathcal{A}^\mathsf{T}$ has no empty states then $\mathsf{TDet}^\mathsf{d}(\mathcal{A}^\mathsf{T}) \equiv (\mathcal{A}^\mathsf{T})^\mathsf{D}$.

(e) TMin^u is isomorphic to the minimal co-DTTA for L.

(f) If L is path-closed then $\mathsf{TMin}^\mathsf{d}(L)$ is isomorphic to the minimal DTTA for L.

(g) If L is path-closed then $\mathsf{TDet}^\mathsf{u}(\mathsf{TDet}^\mathsf{d}(\mathcal{A}^\mathsf{T})) \equiv \mathsf{TMin}^\mathsf{u}(L)$.

(h) If L is path-closed then $\mathsf{TDet}^\mathsf{d}(\mathsf{TDet}^\mathsf{u}(\mathcal{A}^\mathsf{T})) \equiv \mathsf{TMin}^\mathsf{d}(L)$.

It follows from Corollary A.12(c), (d) and (h) that, given a TTA $\mathcal{A}$, we have that $((((\mathcal{A}^\mathsf{T})^R)^\mathsf{D})^R)^\mathsf{D}$ is the minimal DTTA for $\mathcal{L}(\mathcal{A}^\mathsf{T})$.