Domains of Partial Attributed Tree Transducers

Zoltán Fülöp, Sebastian Maneth
Domains of Partial Attributed Tree Transducers

Zoltán Fülöp *  
Department of Computer Science  
József Attila University  
Árpád tér 2., H-6720 Szeged, Hungary  
E-mail: fulop@inf.u-szeged.hu

Sebastian Maneth  
LIACS  
Leiden University, PO Box 9512  
2300 RA Leiden, The Netherlands  
E-mail: maneth@liacs.nl

Abstract

The domains of partial attributed tree transducers (patt’s) are the tree languages recognized by tree walking automata in universal acceptance mode. Moreover, the domains of patt’s having monadic output are the tree languages recognized by deterministic tree walking automata.

Key words: Formal Languages; Attribute Grammars; Tree Automata

1 Introduction

An attribute grammar (ag) [Knu68] can be seen as a device that translates derivation trees of a context-free grammar $G_0$ into expressions over some signature. Abstracting from $G_0$ we obtain an attributed tree transducer (att) [Fü181]. An att $A$ is a total deterministic device that translates all trees in $T_\Sigma$ (the set of trees over a ranked alphabet $\Sigma$) into output trees in $T_\Delta$.

If we allow $A$ to be a partial att (patt), then the domain of $A$ is a tree language $L$ over $\Sigma$, i.e., a subset of $T_\Sigma$. Now the question arises, which tree languages $L$ can be the domain of patt’s (equivalently: partial ag’s)? From [Bar81] it is known that $L$ is inside $\text{REGT}$, the class of regular tree languages. We show that $L$ is the domain of a patt if and only if it can be recognized by a tree walking automaton (twA) in universal acceptance mode. A twA $M$ is a sequential finite-state device that ‘walks’ on the nodes of a tree $s$ (possibly changing its state). The transition of $M$ is defined nondeterministically, however, the tree $s$ is recognized by $M$ in universal acceptance mode if and only if every possible walk of $M$ on $s$ leads to a final state (cf., e.g., [Kam83]). We also show that restricting the output of a patt to monadic trees corresponds to restricting the twA to determinism, which is simply a deterministic tree walking automaton (dtwa) [Au71].

The class of deterministic top-down recognizable tree languages (cf. [GS84]) is properly included in the class of domains of patt’s, because (i) a deterministic top-down tree automaton (dtA) is a patt with synthesized attributes only and (ii) the domain $\{\sigma(a, b), \sigma(b, a)\}$ of $A_1$ from Example 1 cannot be recognized by a dtA. It is open if every recognizable

*Research of this author was supported by grant No. T 030084 of the Hungarian Scientific Foundation
Acknowledgement We are grateful to Joost Engelfriet for helpful discussions.

2 Preliminaries

For two nonnegative integers m, n, we denote by [m, n] the set \{i | m \leq i \leq n\}. Moreover, [1, n] is abbreviated by [n]. Let \( \mathbb{N} = \{1, 2, \ldots\} \) and denote by \( \mathbb{N}^* \) the monoid of words (i.e. sequences of positive integers) generated by \( \mathbb{N} \). The empty word is denoted by 0, hence for every \( u \in \mathbb{N}^* \), \( u0 = 0u = u \).

Let \( A \) be a ranked alphabet. For \( k \in \mathbb{N} \) we denote by \( \Sigma^{(k)} \) the set of symbols in \( \Sigma \) having rank \( k \). If \( \sigma \in \Sigma^{(k)} \), then we define this fact by \( \sigma^{(k)} \). We define \( \text{maxrank}(\Sigma) \) to be the integer \( \max \{k \mid \Sigma^{(k)} \neq \emptyset\} \). \( \Sigma \) is called monadic, if \( \text{maxrank}(\Sigma) \leq 1 \)

Let \( A \) be a set. Then \( |A| \) denotes the cardinality of \( A \). As usual, we denote by \( T_\Sigma(A) \) the set of trees over \( \Sigma \) indexed by \( A \); for \( A = \emptyset \), we write \( T_\Sigma \) for \( T_\Sigma(A) \). Let \( s \in T_\Sigma(A) \). The set of occurrences (or, nodes) of \( s \) is the subset \( O(s) \) of \( \mathbb{N}^* \) defined inductively as follows. For \( s \in A \), \( O(s) = \{0\} \), and for \( s = \sigma(s_1, \ldots, s_k), \sigma \in \Sigma^{(k)}, k \geq 0, \) and \( s_1, \ldots, s_k \in T_\Sigma(A), \)

O(s) = \{0\} \cup \{iu \mid i \in [k], u \in O(t_i)\}. The label of \( s \) at \( u \in O(s) \) and the subtree of \( s \) (rooted) at \( u \) are denoted by \( \text{lab}(s, u) \) and \( s/u \), respectively.

We fix the set \( X = \{x_1, x_2, \ldots\} \) of variable symbols. For \( m \geq 0 \), let \( X_m = \{x_1, \ldots, x_m\} \) and abbreviate \( T_{\Sigma,m} \) by \( T_\Sigma(X_m) \). Moreover, \( T_{\Sigma,m} \) is the subset of \( T_{\Sigma,m} \) consisting of trees \( s \in T_{\Sigma,m} \) in which every variable of \( X_m \) occurs exactly once. For \( s \in T_{\Sigma,m} \) and trees \( s_1, \ldots, s_m \), we denote by \( s[s_1, \ldots, s_m] \) the tree obtained from \( s \) by replacing, for every \( i \in [m], \) every occurrence of \( x_i \) in \( s \) by \( s_i \).

A tree language is a subset of \( T_\Sigma \). A tree transformation is a partial mapping \( \tau : T_\Sigma \rightarrow T_\Delta \), for some ranked alphabet \( \Delta \). The domain of \( \tau \) is \( \text{dom}(\tau) = \{s \in T_\Sigma \mid \exists t \in T_\Delta : \tau(s) = t\} \).

For a class \( C \) of tree transformations, \( \text{dom}(C) = \{\text{dom}(\tau) \mid \tau \in C\} \).

3 Partial Attributed Tree Transducers

A partial attributed tree transducer (for short, patt) is a tuple \( A = (S, I, \Sigma, \Delta, a_0, R) \), where \( S \) and \( I \) are disjoint, unary ranked alphabets of synthesized and inherited attributes, respectively, \( \Sigma \) and \( \Delta \) are ranked alphabets (disjoint with \( S \cup I \)) of input and output symbols, respectively, and \( a_0 \in S \) is the initial attribute. The set \( R = \bigcup \{R_\sigma \mid \sigma \in \Sigma\} \cup R_{\text{root}} \) is a finite set of rules, such that for every \( \sigma \in \Sigma^{(k)} \) with \( k \geq 0 \),

\(1\) for every \( a \in S \), the set \( R_\sigma \) contains at most one rule of the form \( a(0) \rightarrow t[c_1(i_1), \ldots, c_m(i_m)] \), where \( m \geq 0, t \in T_{\Sigma,m}, c_1, \ldots, c_m \in (S \cup I), i_1, \ldots, i_m \in [0, k] \),
(2) for every \( b \in I \) and \( i \in [k] \), the set \( R_\sigma \) contains at most one rule of the form 
\[
b(i) \to t[c_1(i_1), \ldots, c_m(i_m)],
\]
where \( t[c_1(i_1), \ldots, c_m(i_m)] \) is as in (1).

Finally,

(3) for every \( b \in I \), the set \( R_\text{root} \) contains at most one rule of the form 
\[
b(0) \to t[c_1(0), \ldots, c_m(0)],
\]
where \( m \geq 0, t \in T_{\Sigma_m} \), and \( c_1, \ldots, c_m \in (S \cup I) \).

The patt \( A \) is in Bochmann normal form (Bnf), if in (1) and (2), \( \forall j \in [m] \): if \( c_j \in S \) then 
\( i_j \in [k] \) and if \( c_j \in I \) then \( i_j = 0 \), and in (3), \( c_1, \ldots, c_m \in S \). If \( \Delta \) is monadic, then \( A \) is called monadic.

Denote \( S \cup I \) by \( \text{Att} \). Let \( s \in T_\Sigma \). We denote by \( \text{Att}(O(s)) \) the set \( \{ a(u) \mid a \in \text{Att} \text{ and } u \in O(s) \} \). Notice that especially \( a_0(0) \in O(s) \). The \textit{derivation relation induced by} \( A \) on \( s \) is the binary relation \( \Rightarrow_{A,s} \) over \( T_\Delta(\text{Att}(O(s))) \) such that, for \( \xi_1, \xi_2 \in T_\Delta(\text{Att}(O(s))) \), 
\( \xi_1 \Rightarrow_{A,s} \xi_2 \) iff there is an occurrence \( v \) of \( \xi_1 \) with \( \xi_1/v = c(u) \in \text{Att}(O(s)) \), and one of the following three conditions hold.

(1) \( c \in S, \) \( \text{lab}(s, u) = \sigma \in \Sigma^{(k)}, k \geq 0, \) and there is a rule \( c(0) \to t[c_1(i_1), \ldots, c_m(i_m)] \) in 
\( R_\sigma \) such that \( \xi_2 \) is obtained from \( \xi_1 \) by substituting the tree 
\( t[c_1(u_1), \ldots, c_m(u_m)] \) for \( \xi_1/v \).

(2) \( c \in I, \) \( u = u', u' \in \mathbb{N}^*, \) \( \text{lab}(s, u') = \sigma \in \Sigma^{(k)}, k \geq 1, i \in [k], \) and there is a rule 
\( c(i) \to t[c_1(i_1), c_2(i_2), \ldots, c_m(i_m)] \) in \( R_\sigma \) such that \( \xi_2 \) is obtained from \( \xi_1 \) by substituting the tree 
\( t[c_1(u'_{i_1}), \ldots, c_m(u'_{i_m})] \) for \( \xi_1/v \).

(3) \( c \in I, \) \( u = 0, \) and there is a rule 
\( c(0) \to t[c_1(0), \ldots, c_m(0)] \) in \( R_\text{root} \) such that \( \xi_2 \) is 
obtained from \( \xi_1 \) by substituting the tree 
\( t[c_1(0), \ldots, c_m(0)] \) for \( \xi_1/v \).

The normal form \( nf(\Rightarrow_{A,s}, \xi) \) of a \( \xi \in T_\Delta(\text{Att}(O(s))) \) with respect to \( \Rightarrow_{A,s} \) is unique, if it exists (see, e.g., [FHV93]). The \textit{tree transformation} \( \tau_A \) realized by \( A \) is defined as 
\( \tau_A(s) = nf(\Rightarrow_{A,s}, a_0(0)) \) if \( nf(\Rightarrow_{A,s}, a_0(0)) \in T_\Delta \) (and undefined otherwise). The classes of tree transformations induced by patt’s and monadic patt’s are denoted by \( \text{PATT} \) and \( \text{PATT}_{\text{mon}} \), respectively.

**Example 1** Let \( A_1 = (S, I, \Sigma, \Delta, a_0, R) \) be the patt with \( S = \{ \text{start}, \text{what}, \text{syn}_a, \text{syn}_b \} \), 
\( I = \{ \text{ok}, \text{inh}_a, \text{inh}_b \} \), \( \Sigma = \{ \sigma^{(2)}, a^{(0)}, b^{(0)} \} \), \( \Delta = \{ a^{(1)}, b^{(1)}, \#^{(0)} \} \), and \( R \) as follows.

\[
\begin{align*}
R_\sigma & = \{ \text{start}(0) \to \text{what}(1), \text{inh}_a(1) \to \text{syn}_a(2), \text{inh}_b(1) \to \text{syn}_b(2), \text{ok}(2) \to \text{ok}(0) \} \\
R_a & = \{ \text{what}(0) \to a(\text{inh}_a(0)), \text{syn}_a(0) \to a(\text{ok}(0)) \} \\
R_b & = \{ \text{what}(0) \to b(\text{inh}_b(0)), \text{syn}_b(0) \to b(\text{ok}(0)) \} \\
R_\text{root} & = \{ \text{ok}(0) \to \# \}.
\end{align*}
\]

Note that \( A_1 \) is monadic and in Bnf. Consider \( s = \sigma(a, b) \). Then \( \text{start}(0) \Rightarrow_{A_1,s} \text{what}(1) \Rightarrow_{A_1,s} a(\text{inh}_a(1)) \Rightarrow_{A_1,s} a(\text{syn}_a(2)) \Rightarrow_{A_1,s} a(b(\text{ok}(2))) \Rightarrow_{A_1,s} a(b(\text{ok}(0))) \Rightarrow_{A_1,s} a(b(\#)) \).

Hence, \( nf(\Rightarrow_{A_1,s}, \text{start}(0)) = a(b(\#)) \in T_\Delta \) and thus \( \tau_{A_1}(\sigma(a, b)) = a(b(\#)) \). For \( s = \sigma(b, a) \) we obtain \( \tau_{A_1}(s) = b(a(\#)) \). Moreover, for every \( s' \in T_\Sigma \) with \( s' \not\in \{ \sigma(a, b), \sigma(b, a) \} \), \( \tau_{A_1}(s') \) is undefined because \( nf(\Rightarrow_{A_1,s}, \text{start}(0)) \not\in T_\Delta \). Thus, the domain \( \text{dom}(\tau_{A_1}) \) of \( A_1 \) is the tree language \( \{ \sigma(a, b), \sigma(b, a) \} \). \( \square \)
4 Tree-Walking Automata

The following definition of tree-walking automaton is the one of [EHvB99]. Note that we allow “stay” moves only at the root of a tree; this is not a restriction because any stay move at a non-root node can be simulated by an up move followed by a down move.

A tree-walking automaton (for short, twa) is a tuple $M = (Q, \Sigma, \delta, q_0, F)$, where $Q$ is a finite set of states, $\Sigma$ is a ranked alphabet, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and $\delta : (Q \times \Sigma \times [0, m]) \rightarrow \mathcal{P}(Q \times \{\uparrow, \downarrow, \ldots, \downarrow_m\})$ is a function, where $m = \maxrank(\Sigma)$. The twa $M$ is deterministic (for short, $M$ is a dtwa) if $|\delta(q, \sigma, i)| \leq 1$ for all $q \in Q$, $\sigma \in \Sigma$, and $i \in [0, m]$.

Let $s \in T_\Sigma$ be a tree. The set of configurations (of $M$ on $s$) is $Q \times O(s)$. A configuration $(q, u)$ is accepting, if $q \in F$. The walk-relation of $M$ on $s$ is the binary relation $\vdash_{M,s}$ over $Q \times O(s)$ such that, for $(q, u), (q', u') \in Q \times O(s)$, $(q, u) \vdash_{M,s} (q', u')$ iff $lab(s, u) = \sigma$, $u = vi, v \in \mathbb{N}^*$, $i \in [0, m]$ (with $i \neq 0$ if $u \neq 0$), and either (i) $(q', \downarrow_j) \in \delta(q, \sigma, i)$ and $u' = u_j$, or (ii) $(q', \uparrow) \in \delta(q, \sigma, i)$ and $u' = v$.

A walk of (of $M$ on $s$) is a (possibly infinite) sequence $C_0, C_1, \ldots$ of configurations such that $C_0 = (q_0, 0)$ and, for every $i \geq 0$, $C_i \vdash_{M,s} C_{i+1}$. A walk is accepting if it contains an accepting configuration, and maximal if it is either infinite or it is finite and for its last configuration $C$ there is no configuration $C'$ with $C \vdash_{M,s} C'$. The tree $s$ is recognized by $M$ (in universal mode) if every maximal walk on $s$ is accepting. The tree language recognized by $M$ (in universal mode) is denoted by $L_U(M)$. The classes of tree languages recognized (in universal mode) by twa’s and deterministic twa’s are denoted UTWA and DTWA, respectively. Note that if $M$ is deterministic, then there is exactly one maximal walk on $s$.

Example 2 Let $M_1 = (Q, \Sigma, \delta, q_0, F)$ be the twa with $Q = \{q_0, q_1, q_a, q_b, q_f\}$, $\Sigma$ as in the previous example, $F = \{q_f\}$, and the transition function $\delta$ is defined as follows.

- $\delta(q_0, \sigma, 0) = \{(q, \downarrow_1)\}$
- $\delta(q, a, 1) = \{(q, \uparrow)\}$, $\delta(q, b, 1) = \{(q_b, \uparrow)\}$
- $\delta(q_a, \sigma, 0) = \{(q_a, \downarrow_2)\}$, $\delta(q_b, \sigma, 0) = \{(q_b, \downarrow_2)\}$
- $\delta(q_a, b, 2) = \{(q_f, \uparrow)\}$, $\delta(q, a, 2) = \{(q_f, \uparrow)\}$, and
- $\delta(q, \gamma, i) = \emptyset$ for all $q, \gamma, i$ not defined above.

Note that $M_1$ is a dtwa. Consider $s = \sigma(a, b)$. Then $(q_0, 0) \vdash_{M_1,s} (q, 1) \vdash_{M_1,s} (q_0, 0) \vdash_{M_1,s} (q_a, 2) \vdash_{M_1,s} (q_f, 0)$. Hence $\sigma(a, b) \in L_U(M_1)$. It is easy to see that the tree language $L_U(M_1)$ recognized by $M_1$ is $\{\sigma(a, b), \sigma(b, a)\}$. \(\square\)

5 Main Result

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a twa. Since we consider universal acceptance, we may assume w.l.o.g. that $M$ stops walking once it is in a final state (the ‘stopping assumption’), i.e.,
for every \( q \in F, \sigma \in \Sigma, \) and \( i \in [0, m] \): \( \delta(q, \sigma, i) = \emptyset \). This is because all walks that contain a final state are accepting. Then, \( L_U(M) \) equals \( \{ s \in T_\Delta \mid \{ (q_0, 0) \} \models_{M, s} \emptyset \} \), where \( \models_{M, s} \) is the following extension of \( \models_{M} \) to sets. For \( K, K' \subseteq Q \times O(s), K \models_{M, s} K' \), if \( \exists (q, u) \in K \) with \( q \in F \) or \( N \neq \emptyset \) and \( K' = N \cup (K - \{(q, u)\}) \), where \( N = \{(q', u') \mid (q, u) \models_{M, s} (q', u')\} \). Note that in case \( q \in F \), by the stopping assumption \( N = \emptyset \).

**Theorem** (i) \( dom(PAT(T)) = UTWA \) and (ii) \( dom(PAT_{mon}) = DTWA \).

The proof is split up into the following two lemmas. We assume the stopping assumption for the twa in the proof Lemma 3.

**Lemma 3** (i) \( UTWA \subseteq dom(PAT(T)) \) and (ii) \( DTWA \subseteq dom(PAT_{mon}) \).

**Proof.** First we prove (i). Let \( M = (Q, \Sigma, \delta, q_0, F) \) be a twa and let \( m = \text{maxrank}(\Sigma) \).

We now construct the patt \( A = (S, I, \Sigma, \Delta, a_0, R) \) with \( dom(\tau_A) = L_U(M) \). Let \( S = Q \times [0, m] \cup \hat{Q} \times [0, m], I = Q \times \Sigma \cup \hat{Q} \times \Sigma, a_0 = (q_0, 0) \) and \( \Delta = \{ \#_0 \} \cup \{ \#_\nu \mid \nu = \delta(q, \sigma, i) \} \) for some \( q \in Q, \sigma \in \Sigma, i \in [m] \). For every \( \sigma \in \Sigma(k) \) with \( k \geq 0 \), \( R_\sigma \) is the smallest set satisfying conditions (1) - (4).

1. For all \( q \in Q \) and \( i \in [0, m] \), if \( \delta(q, \sigma, i) = \{(q_1, \text{move}_1), \ldots, (q_\nu, \text{move}_\nu)\} \) with \( \nu > 0 \), then let \( \langle q, i \rangle(0) \rightarrow \#_\nu(c_1(i_1), \ldots, c_\nu(i_\nu)) \) be in \( R_\sigma \), where, for every \( j \in [\nu], c_j(i_j) = \langle q_j, l \rangle(0) \), if \( \text{move}_j = \downarrow \) and \( c_j(i_j) = \langle q_j, \sigma \rangle(0) \), if \( \text{move}_j = \uparrow \).

2. For every \( \gamma \in \Sigma, q, i \in Q \), if \( \delta(q, \gamma, i) = \{(q_1, \text{move}_1), \ldots, (q_\nu, \text{move}_\nu)\} \) with \( \nu > 0 \), then let \( \langle q, \gamma \rangle(i) \rightarrow \#_\nu(c_1(i_1), \ldots, c_\nu(i_\nu)) \) be in \( R_\sigma \), where, for every \( j \in [\nu], c_j(i_j) = \langle q_j, l \rangle(0) \), if \( \text{move}_j = \downarrow \) and \( c_j(i_j) = \langle q_j, \gamma \rangle(0) \), if \( \text{move}_j = \uparrow \).

3. For all \( \bar{q} \in \hat{Q}, i \in [k], \gamma \in \Sigma \), let \( \langle \bar{q}, i \rangle(0) \rightarrow \langle q, i \rangle(i) \) and \( \langle \bar{q}, \gamma \rangle(i) \rightarrow \langle q, \sigma \rangle(0) \) be in \( R_\sigma \). (Note that these rules do not depend on the transition function of \( M \), hence we call them “built in” rules.)

4. For every \( q \in F \) and \( i \in [0, m] \), let \( \langle q, i \rangle(0) \rightarrow \#_0 \) and for every \( q \in F, \gamma \in \Sigma \) and \( i \in [k] \), let \( \langle q, \gamma \rangle(i) \rightarrow \#_0 \) be in \( R_\sigma \).

The set \( R_{\text{root}} \) is the smallest set satisfying the following conditions. For all \( \sigma \in \Sigma \) with \( k \geq 0 \) and \( q \in Q \), if \( \delta(q, \sigma, 0) = \{(q_1, \text{move}_1), \ldots, (q_\nu, \text{move}_\nu)\} \) with \( \nu > 0 \), then let \( \langle q, \sigma \rangle(0) \rightarrow \#_\nu(c_1(i_1), \ldots, c_\nu(i_\nu)) \) be in \( R_{\text{root}} \), where, for every \( j \in [\nu], c_j(i_j) = \langle q_j, l \rangle(0) \), if \( \text{move}_j = \downarrow \) and \( c_j(i_j) = \langle q_j, \sigma \rangle(0) \), if \( \text{move}_j = \uparrow \).

Moreover, for every \( q \in Q \) and \( \sigma \in \Sigma, \) let \( \langle \bar{q}, \sigma \rangle \rightarrow \langle q, \sigma \rangle \) be in \( R_{\text{root}} \) (built in rules) and for every \( q \in F \) and \( \sigma \in \Sigma \), let \( \langle q, \sigma \rangle(0) \rightarrow \#_0 \) be in \( R_{\text{root}} \).

Note that \( A \) is not in Bnf because there may be rules in \( R_{\text{root}} \) which contain inherited attributes in their right-hand side.

Let \( \text{Att'} = Q \times [0, m] \cup Q \times \Sigma \). A tree \( t \in T_\Delta(\text{Att'}(O(s))) \) is well-formed if, for every synthesized attribute occurrence \( \langle q, i \rangle(u) \) in \( t \), \( i \) is the last letter of \( u \) (in case \( u = 0 \) also \( i = 0 \)) and, for every inherited attribute occurrence \( \langle q, \sigma \rangle(u) \) in \( t \), \( \text{lab}(s, u) = \sigma \) holds. Moreover, for a well-formed tree \( t \) in \( T_\Delta(\text{Att'}(O(s))), \) let \( \text{Conf}(t) = \{(q, u) \mid \langle q, d \rangle(u) \text{ occurs in } t\} \).

**Claim.** Let \( s \in T_\Sigma \).

(i) For all \( t, t' \in T_\Delta(\text{Att'}(O(s))) \), if \( t \) is well-formed and \( t \Rightarrow_{A, s} t_1 \Rightarrow^*_{A, s} t' \) such that the
rule applied in the first step is not a built in one and all rules applied in the further steps are built in ones, then \( t' \) is also well-formed and \( \text{Conf}(t) \models_{M,s} \text{Conf}(t') \).

(ii) For all \( K, K' \subseteq Q \times O(s) \), if \( K \models_{M,s} K' \), then, for every well-formed \( t \in T_\Delta(\text{Att}'(O(s))) \) with \( \text{Conf}(t) = K \), there is a well-formed \( t' \in T_\Delta(\text{Att}'(O(s))) \) with \( \text{Conf}(t') = K' \) and \( t \Rightarrow_{A,s}^* t' \).

It is straightforward to prove this claim by going through the different cases of \( c \in \text{Att}' \) for \( c(u) \) in \( t \); we omit this. Then, the equality \( \text{dom}(\tau_A) = L_U(M) \) can be proved as follows.

\[ \text{dom}(\tau_A) \subseteq L_U(M) \text{: Let } s \in \text{dom}(\tau_A), \text{ then there is a } t \in T_\Delta \text{ with } \langle (q_0,0), 0 \rangle \Rightarrow_{A,s}^* t. \text{ Since } t_1, \ldots, t_n \in T_\Delta(\text{Att}(O(s))) \text{ and } t'_1, \ldots, t'_n \in T_\Delta(\text{Att}'(O(s))) \text{ such that } \langle (q_0,0), 0 \rangle \Rightarrow_{A,s} t_1 \Rightarrow_{A,s}^* t'_1 \Rightarrow_{A,s} t_2 \cdots \Rightarrow_{A,s} t_n \Rightarrow_{A,s}^* t'_n = t, \text{ the rules applied in the single steps } \langle (q_0,0), 0 \rangle \Rightarrow_{A,s} t_1, \cdots, t'_n \Rightarrow_{A,s}^* t_n \text{ are not built in ones, and all rules applied in the other steps are built in ones. Hence, by applying (i) successively we obtain } \langle (q_0,0) \rangle = \text{Conf}((\langle (q_0,0), 0 \rangle) \models_{M,s} \text{Conf}(t) = \emptyset, \text{ i.e., } s \in L_U(M). \]

\[ L_U(M) \subseteq \text{dom}(\tau_A) \text{: Let } s \in L_U(M), \text{ i.e., let } \{(q_0,0)\} \models_{M,s} \emptyset. \text{ Then, by successive application of (ii), there is a tree } t \in T_\Delta(\text{Att}'(O(s))) \text{ with } \text{Conf}(t) = \emptyset \text{ and } \langle (q_0,0), 0 \rangle \Rightarrow_{A,s}^* t. \text{ Then } t \in T_\Delta \text{ (because } \text{Conf}(t) = \emptyset \text{) and } s \in \text{dom}(\tau_A). \]

Clearly, if \( M \) is deterministic, then \( A \) is a monadic. This proves (ii). \( \square \)

Consider some twa \( (q, k') \in \delta(r, \gamma, i) \) and \( (r', i') \in \delta(q, \sigma, k) \). Then the set \( R \) of the pat\( A \) constructed in the proof of Lemma 3 contains the following rules.

\[ R_s \supseteq \{ \langle q, k \rangle(0) \rightarrow \ldots \langle r, \sigma \rangle(0) \ldots, \} \text{, } R_\gamma \supseteq \{ \langle r, \gamma \rangle(i) \rightarrow \ldots \langle q, k \rangle(i) \ldots, \} \text{, } \]

\[ \langle r, \gamma \rangle(i) \rightarrow \ldots \langle q, k \rangle(i) \ldots, \} \text{, } \]

The top-most two rules are due to condition (1) of the construction in the proof of Lemma 3, and all other rules are due to condition (2).

**Lemma 4** (i) \( \text{dom}(PATT) \subseteq UTTA \) and (ii) \( \text{dom}(PATT_{\text{mon}}) \subseteq DTWA \).

**Proof.** First we prove (i). Let \( A = (S, I, \Sigma, \Delta, a_0, R) \) be a pat\( A \) in Bnf and let \( m = \text{maxrank}(\Sigma) \). We now construct the twa \( M = (Q, \Sigma, \delta, q_0, \{ f \}) \) with \( L_U(M) = \text{dom}(\tau_A) \).

Let \( Q = S \cup (I \times [0,m]) \cup \{ f \}, q_0 = a_0, \) and define transition function \( \delta \) as follows (for technical convenience \( a \in S \) is also denoted by \( \langle a, 0 \rangle \) in the sequel).

For every \( j \in [0,m] \), \( \sigma \in \Sigma^{[k]}, k \geq 0, \) and rule \( a(\nu) \rightarrow r[c_1(i_1), \ldots, c_n(i_n)] \) in \( R_s, \) let \( \delta((a, \nu), \sigma, j) = \{ (f, \uparrow) \} \) if \( n = 0 \) (i.e., \( t \in T_\Delta \)), and otherwise let \( \delta((a, \nu), \sigma, j) \) be the smallest set such that for every \( c(i) \in \{ c_1(i_1), \ldots, c_n(i_n) \} \),

- if \( c \in S, \) then \( \langle c, \downarrow \rangle \in \delta((a, \nu), \sigma, j) \) and
- if \( c \in I, \) then \( \langle (c, j), \uparrow \rangle \in \delta((a, \nu), \sigma, j) \).

Note that \( i = 0 \) in case \( c \in I, \) because \( A \) is in Bnf. For every \( \sigma \in \Sigma^{[k]}, k \geq 0, \) and rule \( b(0) \rightarrow r[c_1(0), \ldots, c_n(0)] \) in \( R_{\text{roots}}, \) let \( \delta((b, 0), \sigma, 0) = \{ (f, \uparrow) \} \) if \( n = 0, \) and otherwise let \( \delta((b, 0), \sigma, 0) = \{ (c_1, \uparrow), \ldots, (c_n, \uparrow) \} \). Finally, let \( \delta(q, \sigma, i) = \emptyset \) for all \( q, \sigma, i \) not defined above.
Now, for a tree \( t \in T\Delta(\text{Att}(O(s))) \), define \( \text{Conf}(t) = \{(a, u) \mid a \in S, a(u) \text{ occurs in } t\} \cup \{(i, a, v) \mid a \in I, a(u) \text{ occurs in } t, u = vi \text{ such that if } u \neq 0 \text{ then } i \neq 0\} \).

**Claim:** Let \( s \in T_{\Sigma} \). Then (i) for all \( t, t' \in T\Delta(\text{Att}(O(s))) \), if \( t \Rightarrow_{A,s} t' \) then \( \text{Conf}(t') = \text{Conf}(t) \). The equality \( \text{Conf}(t) = \text{Conf}(t') \) by applying (i) successively we obtain \( \{(q_0, 0)\} \). Then, by applying (i) successively we obtain \( \{(q_0, 0)\} \). The equality \( \text{Conf}(t) = \text{Conf}(t') \) can be proved as follows.

\( \text{dom}(\tau_A) \subseteq L_U(M) \): Let \( s \in \text{dom}(\tau_A) \), i.e., let \( \{(q_0, 0)\} \Rightarrow_{A,s} t \) for some \( t \in T\Delta \). Then, by applying (i) successively we obtain \( \{(q_0, 0)\} \). The equality \( \text{Conf}(t) = \text{Conf}(t') \) can be proved as follows.

\( L_U(M) \subseteq \text{dom}(\tau_A) \): Let \( s \in L_U(M) \), i.e., let \( \{(q_0, 0)\} \). Then, by (ii), there is a tree \( t \in T\Delta(\text{Att}(O(s))) \) such that \( \{(q_0, 0)\} \Rightarrow_{A,s} t \) and \( \text{Conf}(t) = \text{Conf}(t') \). Then \( t \in T\Delta \) hence \( s \in \text{dom}(\tau_A) \).

Clearly, if \( A \) is monadic, then \( M \) is deterministic, which shows (ii).

Consider some part with \( R_\sigma = \{a'(0) \rightarrow \sigma(b'(0), a'(1)), b(1) \rightarrow \sigma(b(0), a(1)), b(2) \rightarrow a(1)\} \), where \( \sigma \) has rank 2. Applying the construction in the proof of Lemma 4 we obtain, for every \( j \in [m] \), \( \delta(a', \sigma, j) = \{(b', j)^\uparrow, (a', \downarrow_1)\} \), \( \delta(b, 1, \sigma, j) = \{(b, j)^\uparrow, (a, \downarrow_1)\} \), and \( \delta((b, 2), j) = \{(a, \downarrow_1)\} \).

**References**


