Forest-regular Languages and Tree-regular Languages

MURATA Makoto

May 26, 1995

1 Introduction

Forest-regular languages were studied by Pair et al[PQ68] and Takahashi [Tak75]. They are extensions of tree-regular languages [Tha87]. We borrow some concepts from these papers but adopt definitions more similar to those for string-regular languages.

2 Forests and trees

Definition 2.1 (forest). A *forest* over Σ is:

(1) ϵ (the null forest),

(2) $a\langle u\rangle$, where a is a symbol in Σ and u is a forest, or

(3) uv, where u and v are forests.

The set of forests over Σ is denoted by F_Σ. For any forest $u, v, w \in F_\Sigma, u(vw) = (uv)w$ and $u\epsilon = \epsilon u = u$. We abbreviate $a\langle\epsilon\rangle$ as a.

Remark. Since $abc \cdots = a\langle\epsilon\rangle b\langle\epsilon\rangle c\langle\epsilon\rangle \ldots$, a string is also a forest.

Definition 2.2 (tree). A *tree* is a forest of the form $a\langle u\rangle$. The set of trees over Σ is denoted by T_Σ.

Definition 2.3 (forest width). The *width* of a forest u, denoted $|u|$, is the number of trees at the top level of u. That is, $|\epsilon| = 0, |a\langle u\rangle| = 1$, and $|uv| = |u| + |v|$.

Definition 2.4 (forest domain). We assign to each $u \in F_\Sigma$ a subset of $\{1, 2, 3, \ldots\}^+$, denoted $Dom(u)$, such that:

(1) if $u = \epsilon$, then $Dom(u) = \emptyset$,

(2) if $u = a\langle v\rangle$, then $Dom(u) = \{1\} \cup \{1\, v_1 v_2 \ldots v_k \mid k \geq 0, v_1 v_2 \ldots v_k \in Dom(v)\}$,

(3) if $u = vw$, then $Dom(u) = Dom(v) \cup \{(w_1 + |v|)w_2w_3 \ldots w_k \mid k \geq 0, w_1w_2 \ldots w_k \in Dom(w)\}$

$Dom(u)$ is called the *forest domain* of u and the elements of $Dom(u)$ are called *addresses*.

Example 2.5. $Dom(a) = \{1\}$. $Dom(ab) = \{1, 2\}$. $Dom(a\langle bc\rangle d) = \{1, 11, 12, 2\}$.

Remark. If $d \in Dom(u)$ and $d1 \notin Dom(u)$, then d is the address of a leaf node.

Definition 2.6 (forest function). Corresponding to each $u \in F_\Sigma$, there is a function $\overline{u}$ from $Dom(u)$ to Σ as follows:

(1) If $u = a\langle v\rangle$, then $\overline{u}(1) = a$ and $\overline{u}(1\,v_1v_2 \ldots v_k) = \overline{v}(v_1v_2 \ldots v_k)$.

(2) If $u = vw$ and $u_1u_2 \ldots u_k \in Dom(v)$, then $\overline{u}(u_1u_2 \ldots u_k) = \overline{v}(u_1u_2 \ldots u_k)$.

(3) If $u = vw$ and $u_1u_2 \ldots u_k \notin Dom(v)$, then $\overline{u}(u_1u_2 \ldots u_k) = \overline{w}((u_1 \Leftrightarrow |v|)u_2 u_3 ... u_k)$.

Example 2.7. For $u = a\langle bc\rangle d$, $\overline{u}(1) = a$, $\overline{u}(11) = b$, $\overline{u}(12) = c$, and $\overline{u}(2) = d$.

Definition 2.8 (subtree). Given a forest u and a forest address d in $Dom(u)$, the *subtree rooted at d in u*, denoted u/d, is a tree such that $Dom(u/d) = \{1\,v_1v_2 \ldots v_k \mid d\,v_1v_2 \ldots v_k \in Dom(f)\}$ and $\overline{u/d}(1\,v_1v_2 \ldots v_k) = \overline{u}(d\,v_1v_2 \ldots v_k)$.

Example 2.9. $(a\langle bc\rangle d)/1 = a\langle bc\rangle$ and $(a\langle bc\rangle d)/2 = d$.

3 Forest automaton and tree automaton

Definition 3.1 (deterministic forest automaton). A *deterministic forest automaton* (DFA) is a quadruple $<Q, \Sigma, \alpha, F>$, where:

(1) Q is a finite set of states,

(2) Σ is an alphabet,

(3) α is a function (called *transition function*) from $\Sigma \times Q^*$ to Q such that for every $q \in Q$ and $x \in \Sigma$, $\{q_1q_2 \ldots q_k \mid k \geq 0, \alpha(x, q_1q_2 \ldots q_k) = q\}$ is string-regular, and

(4) F is a string-regular set over Q.

Remark. As a convention, instead of $\{q_1q_2 \ldots q_k \mid k \geq 0, \alpha(x, q_1q_2 \ldots q_k) = q\}$, we write $\hat{\alpha}(x, q)$. $\hat{\alpha}$ may be assumed as a function from $\Sigma \times Q$ to the power set of Q^*.

Definition 3.2 (deterministic tree automaton). A DFA $<Q, \Sigma, \alpha, F>$ is a *deterministic tree automaton* (DTA) if $F \subseteq Q$.

Definition 3.3 (transition function extension). The domain of a transition function α can be extended to $F_\Sigma \times Q^*$ as follows:

(1) if $u = \epsilon$, $\alpha(u, q_1 q_2 \dots q_k) = \epsilon$,

(2) if $u = a\langle v \rangle$ $(a \in \Sigma, v \in F_\Sigma)$, $\alpha(u, q_1 q_2 \dots q_k) = \alpha(a, \alpha(v, q_1 q_2 \dots q_k))$

(3) if $u = vw$ $(v, w \in F_\Sigma)$, $\alpha(u, q_1 q_2 \dots q_k) = \alpha(v, q_1 q_2 \dots q_k)\alpha(w, q_1 q_2 \dots q_k)$.

Definition 3.4 (accepted language). A DFA $M =< Q, \Sigma, \alpha, F >$ *accepts* a forest u $(\in F_\Sigma)$ if $\alpha(u, \epsilon) \in F$. The *language accepted* by M, $L(M)$, is the set of forests accepted by M.

Example 3.5. Consider a DFA $M =< \{q_0, q_1\}, \{a, b\}, \alpha, \{q_0 q_1\} >$, where:

$$\hat{\alpha}(a, q_0) = L((q_0|q_1)^*),$$
$$\hat{\alpha}(a, q_1) = \emptyset,$$
$$\hat{\alpha}(b, q_0) = \emptyset, \text{ and}$$
$$\hat{\alpha}(b, q_1) = L((q_0|q_1)^*).$$

Then, $L(M)$ is the set of forests u over $\{a, b\}$ such that $|u| = 2$, $\overline{u}(1) = a$ and $\overline{u}(2) = b$.

Remark. If DFA M is also a DTA, $L(M) \subseteq T_\Sigma$.

Definition 3.6 (forest-regular language). A language L $(\subseteq F_\Sigma)$ is *forest-regular* if L is accepted by a DFA.

Definition 3.7 (tree-regular language). A language L $(\subseteq T_\Sigma)$ is *tree-regular* if L is accepted by a DTA.

Definition 3.8 (state forest). For a DFA $M =< Q, \Sigma, \alpha, F >$, the *state forest* for u $(\in F_\Sigma)$ is a forest u_M $(\in F_Q)$ such that $Dom(u_M) = Dom(u)$ and $\alpha(u/d, \epsilon) = \overline{u_M}(d)$ for every $d \in Dom(u)$.

Example 3.9. Let u be $a\langle b \rangle b\langle a\langle ab \rangle\rangle$. Then, for the DFA M in Example 3.5, u_M is $q_0\langle q_1 \rangle q_1\langle q_0\langle q_0 q_1 \rangle\rangle$.

Remark. $L(M) = \{u \mid \overline{u_M}(1)\,\overline{u_M}(2) \dots \overline{u_M}(|u|) \in F, u \in F_\Sigma\}$.

Definition 3.10 (non-deterministic forest automaton). A *non-deterministic forest automaton* (NDFA) is a quadruple $< Q, \Sigma, \alpha, F >$, where:

(1) Q, Σ, and F are as specified in the definition of DFA, and

(2) α is a relation (called *transition relation*) from $\Sigma \times Q^*$ to Q such that for every $q \in Q$ and $x \in \Sigma$, $\{q_1 q_2 \dots q_k \mid k \geq 0, \alpha(x, q_1 q_2 \dots q_k, q)\}$ is string-regular.

Remark. As a convention, instead of $\{q_1 q_2 \dots q_k \mid k \geq 0, \alpha(x, q_1 q_2 \dots q_k, q)\}$, we write $\hat{\alpha}(x, q)$. $\hat{\alpha}$ may be assumed as a function from $\Sigma \times Q$ to the power set of Q^*.

Definition 3.11 (non-deterministic tree automaton). A NDFA $< Q, \Sigma, \alpha, F >$ is a *non-deterministic tree automaton* (NDTA) if $F \subseteq Q$.

Definition 3.12 (transition relation extension). A transition relation α can be extended as a relation from $F_\Sigma \times Q^*$ to Q as follows:

(1) if $u = \epsilon$, $\alpha(u, q_1 q_2 \dots q_k, r_1 r_2 \dots r_l)$ if and only if $r_1 r_2 \dots r_l = \epsilon$,

(2) if $u = a\langle v \rangle$ $(a \in \Sigma, v \in F_\Sigma)$, $\alpha(u, q_1 q_2 \dots q_k, r_1 r_2 \dots r_l)$ if and only if $l = 1, \alpha(a, s_1 s_2 \dots s_m, r_1)$ and $\alpha(v, q_1 q_2 \dots q_k, s_1 s_2 \dots s_m)$ for some $s_1 s_2 \dots s_m \in Q^*$

(3) if $u = vw$ $(v, w \in F_\Sigma)$, $\alpha(u, q_1 q_2 \dots q_k, r_1 r_2 \dots r_l)$ if and only if, for some $j (1 \le j \le n)$, $\alpha(v, q_1 q_2 \dots q_k, r_1 r_2 \dots r_j)$ and $\alpha(w, q_1 q_2 \dots q_k, r_{j+1} r_{j+2} \dots r_l)$.

Definition 3.13 (accepted language). An NDFA $M = < Q, \Sigma, \alpha, F >$ *accepts* a forest u $(\in F_\Sigma)$ if $\alpha(u, \epsilon, q_1 q_2 \dots q_k)$ for some $q_1 q_2 \dots q_k \in F$ $(k \ge 0)$. The *language accepted* by M, $L(M)$, is the set of forests accepted by M.

Example 3.14. Consider an NDFA $M = < \{q_0\}, \{a, b\}, \alpha, \{q_0\}^* >$, where:

$$\hat{\alpha}(a, q_0) = L(q_0^*), \text{ and}$$
$$\hat{\alpha}(b, q_0) = L(q_0^+)$$

Then, $L(M)$ is the set of forests over $\{a, b\}$ such that nodes labeled by b always have more than one subordinate node.

Remark. If NDFA M is also a NDTA, $L(M) \subseteq T_\Sigma$.

Theorem 3.15 (equivalence of DFA's and NDFA's). *A language L $(\subseteq F_\Sigma)$ is accepted by a NDFA if and only if L is forest-regular.*

Proof of "if". Straightforward. ☐

Proof of "only if". As in the string case, subset construction provides this proof. Assume that L is accepted by an NDFA $M = < Q, \Sigma, \alpha, F >$. Let $R = 2^Q$ and let f be a character-substitution[1] such that $f(q) = \{r \in R \mid q \in r\}$. We define a function β from $\Sigma \times R^*$ to R as $\beta(x, r_1 r_2 \dots r_l) = \{q \in Q \mid \alpha(x, q_1 q_2 \dots q_l, q)$ for some $q_i \in r_i (1 \le i \le l)\}$. Observe that $\hat{\beta}(x, r) = \bigcap_{q \in r} f(\hat{\alpha}(x, q)) \Leftrightarrow \bigcup_{q \in Q - r} f(\hat{\alpha}(x, q))$ and is thus string-regular. Let M' be a DFA $< R, \Sigma, \beta, f(F) >$. Then, $L(M') = L(M)$. ☐

Corollary 3.16 (equivalence of DTA's and NDTA's). *A language L $(\subseteq T_\Sigma)$ is accepted by a NDTA if and only if L is tree-regular.*

Definition 3.17 (state forest). For an NDFA $M = < Q, \Sigma, \alpha, F >$, a *state forest* for u $(\in F_\Sigma)$ is a forest v $(\in F_Q)$ such that $Dom(v) = Dom(u)$ and $\alpha(u/d, \epsilon, \overline{v}(d))$ for every $d \in Dom(u)$.

---

[1] A function h from Δ to the power set of Φ^* is a *character-substitution* if $h(x)$ is string-regular for every $x \in \Delta$, where Δ and Φ are alphabets. The domain of h can be extended to Δ^* by $h(x_1 x_2 \dots x_k) = h(x_1) h(x_2) \dots h(x_k)$ $(k \ge 0)$ and then to the power set of Δ^* by $h(L) = \bigcup_{x \in L} \{h(x)\}$. As is well known, the image of a string-regular set under a character-substitution is string-regular.

Definition 3.18 (unambiguous NDFA). An NDFA $M = < Q, \Sigma, \alpha, F >$ is *unambiguous* if for every $u \in F_\Sigma$, there exists at most one state forest u_M such that $\overline{u_M}(1)\,\overline{u_M}(2)\ldots\overline{u_M}(|u|) \in F$.

Example 3.19. The NDFA M in Example 3.14 is unambiguous. For example, if $u = a\langle b\rangle b$, then $u_M = q_0\langle q_0\rangle q_0$.

4 Forest-regular expression and tree-regular expression

Definition 4.1 (forest with substitution symbols). Let S be a finite set of *substitution symbols*. We define $F_\Sigma[S]$ as the set of forests $u \in F_{\Sigma\cup S}$ such that if $d\,1 \in Dom(u)$ then $\overline{u}(d) \notin S$ (in other words, substitution symbols appear only as leaf nodes). Elements in $F_\Sigma[S]$ are called *forests over* Σ *with substitution symbols in* S.

Definition 4.2 (vertical concatenation). For $s \in S$ and sets $U, V (\subseteq F_\Sigma[S])$, $U \circ_s V$ is the set of all forests $w \in F_\Sigma[S]$ for which there exists $u \in U$ such that w is obtained by replacing each occurrence of s in u by some element of V. Various occurrences of s may be replaced by different elements of V.

Remark. $U \circ_s (V \circ_s W) = (U \circ_s V) \circ_s W$, but $U \circ_s (V \circ_t W)$ may be different from $(U \circ_s V) \circ_t W$. For example, $(\{a\langle st\rangle\} \circ_s \{b\}) \circ_t \{c\} = \{a\langle bc\rangle\}$ but $\{a\langle st\rangle\} \circ_s (\{b\} \circ_t \{c\}) = \{a\langle bt\rangle\}$.

Definition 4.3 (vertical closure). For $s \in S$ and a set $U(\subseteq F_\Sigma[S])$, we define U^{*s} as $X_0 \cup X_1 \cup X_2 \ldots$, where $X_0 = \{s\}$ and $X_{n+1} = X_n \cup (U \circ_s X_n)$.

Example 4.4. $\{a\langle sbs\rangle\}^{*s} = \{s, a\langle sbs\rangle, a\langle a\langle sbs\rangle bs\rangle, a\langle sba\langle sbs\rangle\rangle, a\langle a\langle sbs\rangle ba\langle sbs\rangle\rangle, a\langle sba\langle a\langle sbs\rangle bs\rangle\rangle, a\langle a\langle sbs\rangle ba\langle sbs\rangle\rangle, \ldots\}$

Definition 4.5 (forest-regular expression). A *forest-regular expression (FRE) over* Σ *with substitution symbols in* S is:

(1) $\emptyset$,

(2) ϵ,

(3) s, where $s \in S$,

(4) $a\langle r\rangle$, where r is an FRE,

(5) $r_1 \,|\, r_2$, where r_1 and r_2 are FRE's,

(6) $r_1 r_2$, where r_1 and r_2 are FRE's,

(7) r^*, where r is an FRE,

(8) $r_1 \circ_s r_2$, where $s \in S$ and r_1, r_2 are FRE's, or

(9) r^{*s}, where $s \in S$ and r is an FRE.

Remark. A string-regular expression over Σ is also an FRE.

Definition 4.6 (tree-regular expression). A FRE r over Σ with substitution symbols in S is a *tree-regular expression* (TRE) if r is:

(1) $\emptyset$,

(2) s, where $s \in S$,

(3) $a\langle r\rangle$, where r is a FRE,

(4) $r_1 \,|\, r_2$, where r_1 and r_2 are TRE's,

(5) $r_1 \circ_s r_2$, where $s \in S$, r_1 is a TRE and r_2 is a FRE, or

(6) r^{*s}, where $s \in S$ and r is a TRE.

Definition 4.7 (represented language). The set of forests represented by an FRE r, denoted $L(r)\,(\subseteq F_\Sigma[S])$, is inductively defined as follows:

$$\begin{aligned}
L(\emptyset) &= \emptyset \\
L(\epsilon) &= \{\epsilon\} \\
L(s) &= \{s\} \\
L(a\langle r\rangle) &= \{a\langle u\rangle \mid u \in L(r)\} \\
L(r_1 \,|\, r_2) &= L(r_1) \cup L(r_2) \\
L(r_1 r_2) &= \{u_1 u_2 \mid u_1 \in L(r_1), u_2 \in L(r_2)\} \\
L(r^*) &= \{\epsilon\} \cup \{u_1 u_2 \dots u_k \mid k > 0, f_i \in L(r)(1 \le i \le k)\} \\
L(r_1 \circ_s r_2) &= L(r_1) \circ_s L(r_2) \\
L(r^{*s}) &= L(r)^{*s}
\end{aligned}$$

Example 4.8. $L(a\langle s\rangle^{*s} \circ_s b) = \{b, a\langle b\rangle, a\langle a\langle b\rangle\rangle, a\langle a\langle a\langle b\rangle\rangle\rangle, a\langle a\langle a\langle a\langle b\rangle\rangle\rangle\rangle, \dots\}$ and $L(a\langle s^*\rangle \circ_s b) = \{a, a\langle b\rangle, a\langle bb\rangle, a\langle bbb\rangle, a\langle bbbb\rangle, \dots\}$.

Remark. If a FRE r is also a TRE, then $L(r) \subseteq F_\Sigma[S] \cap T_{\Sigma\cup S}$.

Remark. When an FRE r is also a string-regular expression, $L(r)$ coincides with the set of strings represented by r.

Theorem 4.9 (equivalence of FRE's and (N)DFA's). *A language L ($\subseteq F_\Sigma$) is represented by a FRE if and only if L is forest-regular.*

Proof of "if". Assume that L is accepted by a DFA $M = <Q, \Sigma, \alpha, F>$. As in the string case, we inductively construct an FRE from M.

In preparation we extend the domain of α to $F_\Sigma[Q] \times Q^*$ as follows:

(1) If $u = q$ ($\in Q$), then $\alpha(u, q_1 q_2 \dots q_k) = q$.

(2) If $u = a\langle v\rangle$ ($v \in F_\Sigma[Q]$), then $\alpha(u, q_1 q_2 \dots q_k) = \alpha(a, \alpha(v, q_1 q_2 \dots q_k))$.

(3) If $u = vw$ $(v, w \in F_\Sigma[Q])$, then $\alpha(u, q_1 q_2 \ldots q_k) = \alpha(u, q_1 q_2 \ldots q_k)\alpha(v, q_1 q_2 \ldots q_k)$

Now, for each $q \in Q$ and sets $Q_1, Q_2 \subseteq Q$, let $R[q, Q_1, Q_2]$ be the set of trees u in $F_\Sigma[Q_2]$ such that $\alpha(u, \epsilon) = q$ and $\alpha(u/d, \epsilon) \in Q_1$ for non-leaf address d ($d \in Dom(u)$ and $d\,1 \notin Dom(u)$). In other words, $R[q, Q_1, Q_2]$ is the set of trees carrying M from $Q_2 \cup \Sigma$ to q through Q_1. By induction on the cardinality of Q_1 we prove that $R[q, Q_1, Q_2]$ is represented by some FRE over Σ with substitution symbols in Q.

Base case) Since $R[q, \emptyset, Q_2]$ consists of trees of depth ≤ 1,

$$R[q, \emptyset, Q_2] = \{x \in Q_2 \cup \Sigma \mid \alpha(x, \epsilon) = q\} \cup \bigcup_{x \in \Sigma} \{x\langle u\rangle \mid u \in (Q_2 \cup \Sigma)^* \text{ and } \alpha(x\langle u\rangle, \epsilon) = q\}.$$

Since $\{x \in Q_2 \cup \Sigma \mid \alpha(x, \epsilon) = q\}$ is finite, some FRE r_1 represents this set. Let $U[x]$ be $\{u \in (Q_2 \cup \Sigma)^* \mid \alpha(x\langle u\rangle, \epsilon) = q\}$. Consider a homomorphism[2] g from $(Q_2 \cup \Sigma)^*$ to Q_2^* such that $g(q) = q$ when $q \in Q_2$ and $g(y) = \alpha(y, \epsilon)$ when $y \in \Sigma$. Then, $g(U[x]) = \hat{\alpha}(x, q) \cap Q_2^*$. By the definition of DFA, $g(U[x])$ is string-regular. Since g is a homomorphism, $U[x]$ is also string-regular. Let $u[x]$ be a string-regular expression over $Q_2 \cup \Sigma$ that represents $U[x]$. Then, an FRE $r_1 \mid a_1\langle u[a_1]\rangle \mid a_2\langle u[a_2]\rangle \mid \ldots \mid a_{card(\Sigma)}\langle u[a_{card(\Sigma)}]\rangle$ represents $R[q, \emptyset, Q_2]$, where $\{a_1, a_2, \ldots, a_{card(\Sigma)}\} = \Sigma$.

Inductive case) Observe that the following equation holds.

$$R[q, Q_1 \cup \{p\}, Q_2] = R[q, Q_1, Q_2 \cup \{p\}] \circ_p R[p, Q_1, Q_2 \cup \{p\}]^{*p} \circ_p R[p, Q_1, Q_2]$$

Intuitively, this equation implies "to go from $Q_2 \cup \Sigma$ to q through $Q_1 \cup \{p\}$, go from $Q_2 \cup \Sigma$ to p through Q_1, go from $Q_2 \cup \{p\} \cup \Sigma$ to p through Q_1 for zero or more times, and finally go from $Q_2 \cup \{p\} \cup \Sigma$ to q through Q_1." By the induction hypothesis, $R[q, Q_1, Q_2 \cup \{p\}]$, $R[p, Q_1, Q_2 \cup \{p\}]$, $R[p, Q_1, Q_2]$ can be represented by FRE's over Σ with substitution symbols in Q, say r_1, r_2, r_3. Thus, $R[q, Q_1 \cup p, Q_2]$ can be represented by $r_1 \circ_p r_2^{*p} \circ_p r_3$. This completes the inductive proof.

Having proved that $R[p, Q_1, Q_2]$ is represented by some FRE, we are ready to prove that $L(M)$ is as well. For every $q \in Q$, consider an FRE r_q over Σ with substitution symbols in Q such that $L(r_q) = R[q, Q, \emptyset]$. Let r_F be a string-regular expression which represents F. By replacing each q in r_F with r_q, we obtain an FRE that represents $L(M)$.

□

Proof of "only if". Let r be an FRE over Σ with substitution symbols in S (a finite set) such that r represents a forest language L ($\subseteq F_\Sigma$). We are going to construct an NDFA that accepts L.

[2] A *homomorphism* h is a character-substitution such that $h(x)$ contains a single string for each x. An *inverse homomorphic image* of a language L is $\{x \mid h(x) \in L\}$. It is known that an inverse homomorphic image of a string-regular set is string-regular.

For each sub-expression r' of r, we inductively construct an NDFA $M[r']$ that accepts $L(r')$. Since $L(r')$ might not be a subset of F_Σ, we use $\Sigma \cup S$ rather than Σ as an alphabet. If this inductive construction yields $M[r] = <\Sigma \cup S, Q, \alpha, F>$, the NDFA we want is $<\Sigma, Q, \alpha \cap (\Sigma \times Q^* \times Q), F>$.

Inductive construction

Case 1 $r' = \emptyset$.

$$M[\emptyset] = <\emptyset, \Sigma \cup S, \emptyset, \emptyset> .$$

Case 2 $r' = \epsilon$.

$$M[\epsilon] = <\emptyset, \Sigma \cup S, \emptyset, \{\epsilon\}> .$$

Case 3 $r' = s \ (\in S)$.

$$M[s] = <\{s\}, \Sigma \cup S, \{(s, \epsilon, s)\}, \{s\}> .$$

Case 4 $r' = a\langle r_1 \rangle$ ($a \in \Sigma$, r_1 is an FRE). Let $M[r_1]$ be $<Q_1, \Sigma \cup S, \alpha_1, F_1>$. Then,

$$M[a\langle r_1 \rangle] = <Q_1 \cup \{q_F\}, \Sigma \cup S, \beta, \{q_F\}> \text{, where:}$$

$$\hat{\beta}(x, q) = \begin{cases} F_1 & \text{if } q = q_F \text{ and } x = a, \\ \emptyset & \text{if } q = q_F \text{ and } x \neq a, \\ \hat{\alpha_1}(x, q) & \text{if } q \in Q_1. \end{cases}$$

Case 5 $r' = r_1 \,|\, r_2$ (r_1, r_2 are FRE's). Let $M[r_1]$ be $<Q_1, \Sigma \cup S, \alpha_1, F_1>$, and let $M[r_2]$ be $<Q_2, \Sigma \cup S, \alpha_2, F_2>$. By renaming states not contained in S, we can assume $Q_1 \cap Q_2 \subseteq S$. Then,

$$M[r_1 \,|\, r_2] = <Q_1 \cup Q_2, \Sigma \cup S, \beta, F_1 \cup F_2> \text{, where:}$$

$$\hat{\beta}(x, q) = \begin{cases} \hat{\alpha_1}(x, q) & \text{if } q \in Q_1 \Leftrightarrow S, \\ \hat{\alpha_2}(x, q) & \text{if } q \in Q_2 \Leftrightarrow S, \\ \{\epsilon\} & \text{if } q \in S \cap (Q_1 \cup Q_2). \end{cases}$$

Case 6 $r' = r_1 r_2$ (r_1, r_2 are FRE's). Let $M[r_1]$ be $<Q_1, \Sigma \cup S, \alpha_1, F_1>$, and let $M[r_2]$ be $<Q_2, \Sigma \cup S, \alpha_2, F_2>$. Again, we assume $Q_1 \cap Q_2 \subseteq S$. Then, $M[r_1 r_2]$ is the same as $M[r_1 \,|\, r_2]$ except that the last constituent is $F_1 F_2$ rather than $F_1 \cup F_2$.

$$M[r_1 r_2] = <Q_1 \cup Q_2, \Sigma \cup S, \beta, F_1 F_2> \text{, where:}$$

$$\hat{\beta}(x, q) = \begin{cases} \hat{\alpha_1}(x, q) & \text{if } q \in Q_1 \Leftrightarrow S, \\ \hat{\alpha_2}(x, q) & \text{if } q \in Q_2 \Leftrightarrow S, \\ \{\epsilon\} & \text{if } q \in S \cap (Q_1 \cup Q_2). \end{cases}$$

Case 7 $r' = r_1^*$ (r_1 is an FRE). Let $M[r_1]$ be $<Q_1, \Sigma \cup S, \alpha_1, F_1>$. Then,

$$M[r^*] = <Q_1, \Sigma \cup S, \alpha_1, F_1^*> .$$

Case 8 $r' = r_1 \circ_s r_2$ ($s \in S$, and r_1, r_2 are FRE's). Let $M[r_1]$ be $<Q_1, \Sigma \cup S, \alpha_1, F_1>$, and let $M[r_2]$ be $<Q_2, \Sigma \cup S, \alpha_2, F_2>$. Again, we assume $Q_1 \cap Q_2 \subseteq S$. Let f be a character-substitution such that $f(x) = F_2$ when $x = s \in Q_1$, and $f(x) = \{x\}$ when $x \in Q_1 \Leftrightarrow \{s\}$. Then,

$$M[r_1 \circ_s r_2] = <(Q_1 \Leftrightarrow \{s\}) \cup Q_2, \Sigma \cup S, \beta, f(F_1)> \text{, where:}$$

$$\hat{\beta}(x,q) = \begin{cases} f(\hat{\alpha_1}(x,q)) & \text{if } q \in Q_1 \Leftrightarrow S, \\ \hat{\alpha_2}(x,q) & \text{if } q \in Q_2 \Leftrightarrow S, \\ \{\epsilon\} & \text{if } q \in S \cap (Q_1 \cup Q_2). \end{cases}$$

Case 9 $r' = r_1^{*s}$ ($s \in S, r_1$ is an FRE). Let $M[r_1]$ be $<Q_1, \Sigma \cup S, \alpha_1, F_1>$. Let f be a character-substitution such that $f(x) = F_1 \cup \{x\}$ when $x = s \in Q_1$, and $f(x) = \{x\}$ when $x \in Q_1 \Leftrightarrow \{s\}$. Then,

$$M[r^{*s}] = <Q_1, \Sigma \cup S, \beta, F \cup \{s\}> \text{, where:}$$

$$\hat{\beta}(x,q) = \begin{cases} f(\hat{\alpha_1}(x,q)) & \text{if } q \in Q_1 \Leftrightarrow S, \\ \{q\} & \text{if } q \in S \cap Q_1. \end{cases}$$

□

Corollary 4.10 (equivalence of TRE's and (N)DTA's). *A language L ($\subseteq T_\Sigma$) is represented by a TRE if and only if L is tree-regular.*

5 Forest-regular grammar and tree-regular grammar

Definition 5.1 (forest-regular grammar). A *forest-regular grammar* (FRG) is a quadruple $<N, \Sigma, P, X>$, where:

(1) N is a finite set of non-terminals,

(2) Σ is an alphabet,

(3) P is a finite set of *production rules*, each of which is of the form $A \to x\langle r \rangle$ ($A \in N, x \in \Sigma$, and r is a string-regular expression over N),

(4) X is a string-regular set over N.

Definition 5.2 (tree-regular grammar). An FRG $<N, \Sigma, P, X>$ is a *tree-regular grammar* if $X \subseteq N$.

Definition 5.3 (derivation). For an FRG $G = < N, \Sigma, P, X >$ and $u, v \in F_\Sigma[N])$, $u \to v$ or v is *directly derived* from u if for some $A \to a\langle r \rangle \in P$, v is obtained by replacing an occurence of A in u by an element of $\{a\langle w \rangle \mid w \in L(r)\}$. The transitive closure of $\to$ is denoted by $\underset{*}{\to}$.

Definition 5.4 (generated language). The *language generated by* G, $L(G)$, is $\{t_1 t_2 \ldots t_k \in F_\Sigma \mid k \geq 0, A_1 A_2 \ldots A_k \in X, A_i \underset{*}{\to} t_i (1 \leq i \leq k)\}$.

Example 5.5. Consider an FRG $G = < \{A\}, \{a, b\}, P, \{A\}^* >$, where $P = \{A \to a\langle A^* \rangle, A \to b\langle A^+ \rangle\}$. Then, $L(G)$ is the language accepted by G in Example 3.14.

Remark. If FRG G is also a TRG, $L(G) \subseteq T_\Sigma$.

Theorem 5.6 (equivalence of FRG's and (N)DFA's). *A language L ($\subseteq F_\Sigma$) is generated by a FRG if and only if L is forest-regular.*

Proof of "if". Assume that L is accepted by an NDFA $M = < Q, \Sigma, \alpha, F >$. For every $q \in Q$ and $x \in \Sigma$, let $r_{q,x}$ be a string-regular expression over Q such that $L(r_{q,x}) = \hat{\alpha}(x, q)$. The set of production rules P is defined as $\bigcup_{q \in Q, x \in \Sigma} \{q \to x\langle r_{q,x} \rangle\}$. Then, $< Q, \Sigma, P, F >$ is an FRG and generates $L(M)$. □

Proof of "only if". Assume that L is generated by an FRG $G = < N, \Sigma, P, X >$. A relation α from $\Sigma \times N^*$ to N is defined as $\alpha(a, A_1 A_2 \ldots A_k, A) \Leftrightarrow$ for some $A \to a\langle r \rangle \in P$, $A_1 A_2 \ldots A_k \in L(r)$. Then, $< N, \Sigma, \alpha, X >$ is an NDFA and accepts $L(G)$. □

Corollary 5.7 (equivalence of TRG's and (N)DTA's). *A language L ($\subseteq T_\Sigma$) is generated by a NDTA if and only if L is tree-regular.*

6 Properties of forest-regular languages and tree-regular languages

Theorem 6.1 (Boolean algebra). *The class of forest-regular languages from a Boolean algebra.*

Proof. We only have to prove closure under negation and closure under union. Let L_1 and L_2 be forest-regular languages over Σ. By definition, some DFA $M = < Q, \Sigma, \alpha, F >$ accepts L_1. The negation of L_1, $F_\Sigma \Leftrightarrow L_1$, is accepted by DFA $< Q, \Sigma, \alpha, Q^* \Leftrightarrow F >$ and is thus forest-regular. By Theorem 4.9, some FRE r_1 and r_2 represent L_1 and L_2, respectively. The union of L_1 and L_2 is represented by $r_1 \mid r_2$ and is thus forest-regular. □

Remark. Given two NDFA's, it is possible to directly construct an NDFA for the intersection of the two accepted languages.

Let NDFA $M_1 = < Q_1, \Sigma, \alpha_1, F_1 >$ and let $M_2 = < Q_2, \Sigma, \alpha_2, F_2 >$. We define two character-substitutions f_1 and f_2 as $f_1(q_1) = \{(q_1, q_2) \mid q_2 \in Q_2\}$ ($q_1 \in Q_1$)

and $f_2(q_2) = \{(q_1, q_2) \mid q_1 \in Q_1\}$ $(q_2 \in Q_2)$, respectively. Then, the intersection of $L(M_1)$ and $L(M_2)$ is accepted by NDFA $< Q_1 \times Q_2, \Sigma, \beta, F_1 \times F_2 >$, where $\hat{\beta}(x, (q_1, q_2)) = f_1(\hat{\alpha_1}(x, q_1)) \cap f_2(\hat{\alpha_2}(x, q_2))$.

Corollary 6.2 (Boolean algebra). *The class of tree-regular languages from a Boolean algebra.*

Proof. The same as the previous proof except that the final state set for the negation DTA is $Q \Leftrightarrow F$ rather than $Q^* \Leftrightarrow F$. □

BIBLIOGRAHICS NOTES

Our definition of FRE's is derived from [PQ68] but differs in not using projections and not using "enracinement". Our definition can also be considered as a forest-version of Thatcher and Wright's tree regular expressions [TW68]. We define FRG's similarily to [PQ68, Tak75] but again we avoid projections. Alternatively, our definition can be considered as a forest-version of Brainerd's tree regular grammars (called "tree generating regular systems") [Bra69]. Our definitions of NDFA's and DFA's are derived from (non-)deterministic tree automata of [Tha67] except that we have extended them to forests. We proved the equivalence between FRE's and (N)DFA's according to Arbib and Give'on's proof [AG68], which is simpler than those in [TW68].

References

[AG68] M.A. Arbib and Y. Give'on. Algebra automata i: Parallel programming as a polegomena to the categorical approach. *Information and Control*, 12:331–345, 1968.

[Bra69] Walter S. Brainerd. Tree generating regular systems. *Information and Control*, 14:217–231, 1969.

[PQ68] C. Pair and A. Quere. Définition et etude des bilangages réguliers. *Information and Control*, 13:565–593, 1968.

[Tak75] Masako Takahashi. Genelalizations of regular sets and their application to a study of context-free languages. *Information and Control*, 27:1–36, 1975.

[Tha67] J. W. Thatcher. Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. *Journal of Computer and System Sciences*, 1:317–322, 1967.

[Tha87] James W. Thatcher. Tree automata: An informal survey. In A.V. Aho, editor, *Currents in the theory of computing*, pages 143–172. Prentice-Hall, 1987.

[TW68] J.W. Thatcher and J.B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2(1):57–81, 1968.