AUTOMATA WITH RANKED STATE SETS

Dieter Schütt[1]

Technical Report 71-B4

Department of Computer and Information Science

University of Massachusetts, Amherst 01002

Abstract. The notion of finite state automaton is extended to systems which process directed ordered acyclic graphs. The generalization is mainly achieved by associating two ranking relations with the state set of an automaton. 'Ranked automata' can be considered as pattern-generating systems or as control devices for finite state automata. As was done for tree automata, closure properties of ranked automaton definable sets and languages induced by such automata are investigated.

[1]On leave from Amherst. New address:

Institut für Angewandte Mathematik und Informatik, 53 Bonn, Wegelerstr. 6, W. Germany.

0. Preliminaries

Let V be an alphabet (i.e. a non-empty, finite set).

A directed ordered graph (DOG) is a map $\Gamma : V \to V^*$.

A node v of Γ, i.e. an element of V, is called

(1) a leaf of Γ if $\Gamma(v) = \Lambda$

(2) an input node for $v' \in V$ if v' is a symbol in $\Gamma(v)$

(3) a root of Γ if v is not an input node for any $v' \in V$.

Note that our concept of DOGs differs from that given by Arbib and Give'on [1] in that we reverse the edges.

A sequence $v_1, \ldots, v_n$ of nodes of Γ is

(1) an undirected path of Γ if for $i = 1, \ldots, n-1$ either v_i is an input node for v_{i+1} or v_{i+1} is an input node for v_i

(2) a directed path of Γ if for $i = 1, \ldots, n-1$ v_i is an input note for v_{i+1}.

A DOG is connected if every pair of nodes are joined by an undirected path. A DOG which has no directed path that begins and ends with the same node is called acyclic.

Henceforth, all DOGs considered will be connected, acyclic DOGs $\Gamma : V \to V^*$ with the property that $\Gamma(v) = \Gamma(v')$ for any two nodes v, v' which are input nodes for the same node v''.

A tree is a DOG for which each node has at most one input node.

A linear ordered DOG (LOG) is a pair $(\Gamma, \lessdot)$ where $\Gamma : V \to V^*$ is a DOG and $\lessdot$ is a linear order on V such that

(1) if $\Gamma(v) = v_1 \cdots v_\ell$ $(\ell \geq 1)$ then $v \lessdot v_1 \lessdot \cdots \lessdot v_\ell$

(2) if v' is an input node for v, v" is not an input node for v, and v' $\lessdot$ v" (v" $\lessdot$ v') then v $\lessdot$ v" (v" $\lessdot$ v).

Clearly, there are DOGs for which no linear order of that kind exists. However, every tree is a LOG.

If $(\Gamma,\lessdot)$ is a LOG then

(1) a root of Γ is the smallest element of V

(2) if v $\lessdot$ v' $\lessdot$ v" and Γ(v) = Γ(v") then Γ(v') = Γ(v)

(3) if v $\lessdot$ v' $\lessdot$ v" and v, v" have the same input node v"ϵV then so has v'.

Example 1. V = {1,...,8}

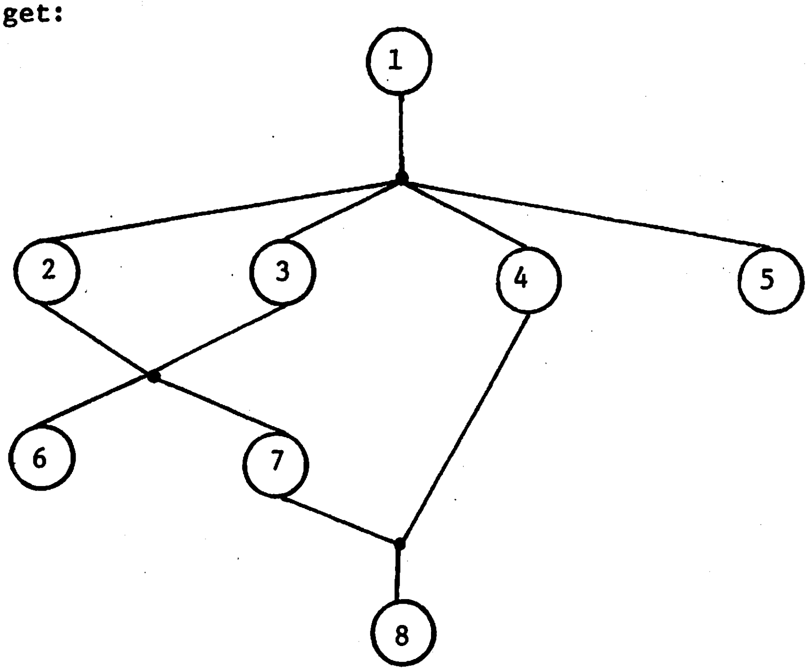
v	1	2	3	4	5	6	7	8
Γ(v)	2345	67	67	8	Λ	Λ	8	Λ

The (unique) linear order $\lessdot$ on V such that $(\Gamma,\lessdot)$ is a LOG is given by the sequence

1, 2, 3, 6, 7, 4, 8, 5.

For the pictorical representation of a LOG we choose the same method as used in [2], that is we gather together the 'outputs' of all nodes which 'feed' the same sequences of nodes.

Thus we get:



Let Σ be an alphabet.

A <u>Σ-LOG</u> is a triple $D = (\Gamma, \lessdot, \mu)$ where $(\Gamma, \lessdot)$ is a LOG and $\mu : V \to \Sigma$ is a labelling function.

By $\mathcal{L}_\Sigma$ we denote the set of Σ-LOGs.

Note that Σ-LOGs are derivation structures in the sense of Buttelmann [2].

For a Σ-LOG D and the sequence $v_1, \ldots, v_n$ of all its leaves in increasing order, the <u>frontier of D</u> is defined by

$fr(D) = \mu(v_1) \cdots \mu(v_n)$.

This completes our introduction to the graphs we shall be using. Now we shall continue with some definitions about automata.

An alphabet Q is <u>ranked</u> if a finite relation $r \subseteq Q \times \mathbb{N}$ ($\mathbb{N}$ is the set of positive integers) is specified. The set $\{q \mid (q,n) \epsilon r\}$ of n-ary elements is denoted by Q_n^r.

A <u>ranked</u> (finite state) <u>automaton</u> (RA) is a 7-tuple

$$A = (Q, \downarrow, \uparrow, \Sigma, \delta, Q_o, F)$$

where Q, Σ are (finite) alphabets,

Q_o, $F \subseteq Q$,

$\downarrow$, $\uparrow$ are finite subsets of $Q \times \mathbb{N}$, and

(0) $\delta \subseteq \{(q^k, x^k, \bar{q}^\ell) \mid q \epsilon Q_k^\downarrow \wedge \bar{q} \epsilon Q_\ell^\uparrow \wedge x \epsilon \Sigma\}$ or

(0') $\delta \subseteq \{(q^\ell, x^k, \bar{q}^k) \mid q \epsilon Q_\ell^\uparrow \wedge \bar{q} \epsilon Q_k^\uparrow \wedge x \epsilon \Sigma\}$.

In the first case (0) A is a <u>topdown</u> ranked automaton (T-RA), in the second case (0') a <u>bottomup</u> ranked automaton (B-RA).[1]

---

[1] Convention: (i) and (i') indicate the topdown and bottomup case, respectively.

Q is the set of states, Q_o the set of initial states, F the set of final states, Σ the set of input symbols, and δ the transition relation of A.

Note that δ is a finite set.

Remarks. (1) A B-RA is a generalized finite automaton defined by Buttelmann [2]. Since we restrict ourselves to transitions of the form $(q^k, x^k, \bar{q}^\ell)$ rather than $(q_1 \ldots q_k, x_1 \ldots x_k, \bar{q}_1 \ldots \bar{q}_\ell)$ the converse is not true. A T-RA is a generalized finite automaton if $(q^k, x^k, \bar{q}^\ell) \epsilon \delta$ implies $k = \ell$.

(2) If $\downarrow$ is a map and $Q_1^\downarrow = Q$ then A is a topdown or bottomup tree automaton.

(3) If $\downarrow, \uparrow$ are maps and $Q_1^\downarrow = Q_1^\uparrow = Q$ then A is a finite state automaton.

A RA A is called

(1) (total) left-deterministic (LT - RA or LB - RA) if for $\bar{q} \epsilon Q_\ell^\uparrow$ and $x^k (x \epsilon \Sigma,\ k \epsilon \mathbb{N}$ s.t. $Q_k^\downarrow \neq \emptyset)$ there is exactly one $q \epsilon Q$ s.t. $(q^k, x^k, \bar{q}^\ell) \epsilon \delta$
or for $\bar{q} \epsilon Q_k^\downarrow$ and $x^k (x \epsilon \Sigma)$ there is exactly one $q \epsilon Q$ and $\ell \epsilon \mathbb{N}$ s.t. $(q^\ell, x^k, \bar{q}^k) \epsilon \delta$

(2) (total) right-deterministic (RT-RA or RB-RA) if for $q \epsilon Q_k^\downarrow$ and $x^k (x \epsilon \Sigma)$ there is exactly one $\bar{q} \epsilon Q$ and $\ell \epsilon \mathbb{N}$ s.t. $(q^k, x^k, \bar{q}^\ell) \epsilon \delta$
or for $q \epsilon Q_\ell^\uparrow$ and $x^k (x \epsilon \Sigma,\ k \epsilon \mathbb{N}$ s.t. $Q_k^\downarrow \neq \emptyset)$ there is exactly one $\bar{q} \epsilon Q$ s.t. $(q^\ell, x^k, \bar{q}^k) \epsilon \delta$.

The <u>finite state automaton induced by the RA</u> A is a 5-tuple $\tilde{A} = (Q, \Sigma, \tilde{\delta}, Q_o, F)$ where $\tilde{\delta} \subseteq Q \times \Sigma \times Q$ is defined by $(q,x,\bar{q})\epsilon\tilde{\delta}$ iff $(q^k,x^k,\bar{q}^{\ell})\epsilon\delta$ or $(q^{\ell},x^k,\bar{q}^k)\epsilon\delta$ for some $k,\ell\epsilon\mathbb{N}$.

Clearly, determinism of A generally does not imply determinism of $\tilde{A}$.

In order to compute upon a Σ-LOG D with a RA A we introduce the notion of a 'run' (cf.[6]).

A <u>run of A on D</u> $= (\Gamma,\lessdot,\mu)$ is a map $\rho : V \rightarrow Q$ such that (1) if v is a root of Γ then $\rho(v)\epsilon Q_o$

(2) if $v_1,\ldots,v_k$ is a maximal sequence of nodes of Γ ordered by $\lessdot$ such that $\Gamma(v_1) = \ldots = \Gamma(v_k) = v_1' \ldots v_{\ell}'$ $(\ell \geq 1)$ then $(\rho(v_1)\ldots\rho(v_k),\ \mu(v_1)\ldots\mu(v_k),\ \rho(v_1')\ldots\rho(v_{\ell}'))\epsilon\delta$

or (1') if v is a leaf of Γ then there is a $q\epsilon Q_o$ such that $(q,\mu(v),\ \rho(v))\epsilon\delta$

(2') if $v_1,\ldots,v_k$ is a maximal sequence of nodes of Γ ordered by $\lessdot$ such that $\Gamma(v_1) = \ldots = \Gamma(v_k) = v_1' \ldots v_{\ell}'$ $(\ell \geq 1)$ then $(\rho(v_1')\ldots\rho(v_{\ell}'),\ \mu(v_1)\ldots\mu(v_k),\ \rho(v_1)\ldots\rho(v_k))\epsilon\delta$.

Thus a run is an assignment of states to nodes of Γ in conformity with the transition relation of A.

The automaton <u>A accepts D</u> if there exists a run of A on D such that

(3) if v is a leaf of Γ then there is a $q\epsilon F$ such that $(\rho(v),\ \mu(v),q)\epsilon\delta$

(3') if v is a root of Γ then $\rho(v)\epsilon F$.

T(A) denotes the set of Σ-LOGs accepted by A.

A subset T of $\mathcal{L}_{\Sigma}$ is <u>RA-definable</u> if for some RA A, T = T(A).

The <u>language of A</u> is defined by $L(A) = \{fr(D)\,|\,D\epsilon T(A)\}$.

Remark. A hierarchy of generalized automata analogous to the familiar hierarchy of generative grammars may be introduced as in [2]. Note that in [2], acceptance of Σ-LOGs means acceptance of DOGs with only one root.

Example 2. $Q = Q_o = F = \{q\}$

$\downarrow = \{(q,1),\ (q,2)\}$

$\uparrow = \{(q,1),\ (q,2),\ (q,4)\}$

$\Sigma = \{a,b\}$

$\delta = \{(q,a,q),$

$(q,a,q^4),$

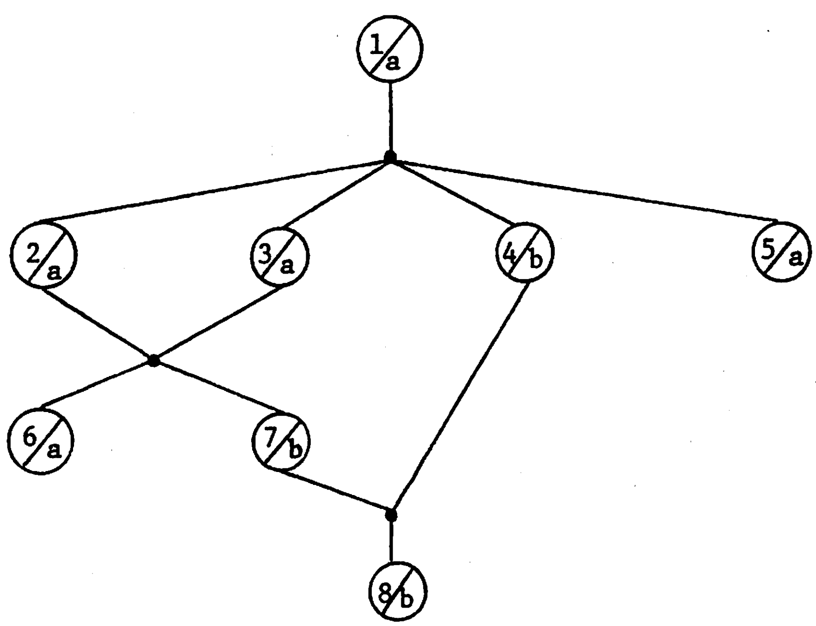
$(q,b,q),$

$(q^2,a^2,q^2),$

$(q^2,b^2,q)\}$

It can easily be checked that the DOG of example 1 is accepted by the T - RA $A = (Q, \downarrow, \uparrow, \Sigma, \delta, Q_o, F)$ if we define μ as follows

v	1	2	3	4	5	6	7	8
$\mu(v)$	a	a	a	b	a	a	b	b



Note that T(A) is not definable by any RA for which the relations $\downarrow,\uparrow$ are maps.

1. Comparison of T - RA and B - RA

The main result of this section will be that the classes of T - RA and B - RA are comparable whereas the classes of RT - RA and LT - RA (LB - RA and RB - LA) are incomparable with respect to definable sets.

A consequence of this result is that the classes of tree automata and ranked automata are incomparable with respect to definable sets.

Theorem 1. A subset of $\mathcal{L}_\Sigma$ is T - RA definable iff it is B - RA definable.

Proof: Let $A = (Q, \downarrow, \uparrow, \Sigma, \delta, Q_o, F)$ be a T - RA or B - RA.

Define $Q_o' = F$

$$F' = Q_o$$

$$\delta' = \{(\bar{q}^\ell, x^k, q^k) \mid (q^k, x^k, \bar{q}^\ell) \epsilon \delta\}$$

or $\delta' = \{(\bar{q}^k, x^k, q^\ell) \mid (q^\ell, x^k, \bar{q}^k) \epsilon \delta\}$.

Then $A' = (Q, \downarrow, \uparrow, \Sigma, \delta', Q_o', F')$ is a B - RA or T - RA such that $T(A') = T(A)$.

Corollary 1. A subset of $\mathcal{L}_\Sigma$ is RT - RA (LT - RA) definable iff it is LB - RA (RB - RA) definable.

Theorem 2. There are subsets of $\mathcal{L}_\Sigma$ which are T - RA (B - RA) definable but not RT - RA (LB - RA) definable.

The proof of this theorem is based on an example given by Magidor and Moran [6].

Proof: Define the T - RA A by

$$Q = \{q_1, q_2, q_3, q_4\}$$

$$Q_o = \{q_1, q_2\}$$

$$F = \{q_4\}$$

$$\Sigma = \{a,b\}$$

δ	a	b
q_1	q_1^2, q_2^2, q_4	q_3
q_2	q_3	q_1^2, q_2^2, q_4
q_3	q_3	q_3

T(A) is the set of Σ-LOGs which are trees with the property that each node which is not a leaf is an input node for exactly two other nodes and that any two nodes which have the same input node are labelled with the same element of Σ.

Note that any run upon a Σ-LOG not in T(A) is trapped into the absorbing state q_3.

From the fact that we treat RA with finite state sets it follows that a RT - RA A' such that $T(A') \supseteq T(A)$ must accept a tree which has at least one node which is an input node for two nodes labelled with different elements of Σ.

<u>Remark</u>. Magidor and Moran [6] have shown that there are finite sets of trees which are not definable by any deterministic sinking (i.e. topdown tree) automaton. However, that does not hold for RT - RA because a deterministic RA is allowed to have more than one initial state.

<u>Theorem 3</u>. There are subsets of $\mathcal{L}_\Sigma$ which are RT - RA (LB - RA) definable but not LT - RA (RB - RA) definable.

Proof: Define the RT - RA A by

$Q = \{q_1, q_2, q_3\}$

$Q_o = \{q_1\}$

$F = \{q_2\}$

$\Sigma = \{a, b\}$

δ	a	b
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_3	q_3

Clearly, $T(A) = \{a^m b^n \mid m \geq 0 \wedge n \geq 1\}$ is not LT - RA definable.

<u>Remark</u>. Buttelmann [2] proved that sets which are definable by generalized finite automata are also definable by right-deterministic ones. By theorems 3 and 4 however, B - RA definable sets need not be RB - RA definable.

<u>Theorem 4</u>. There are subsets of $\mathcal{L}_\Sigma$ which are LT - RA (RB - RA) definable but not RT - RA (LB - RA) definable.

Proof: Define the LT - RA A by

$Q = Q_o = F = \{q\}$

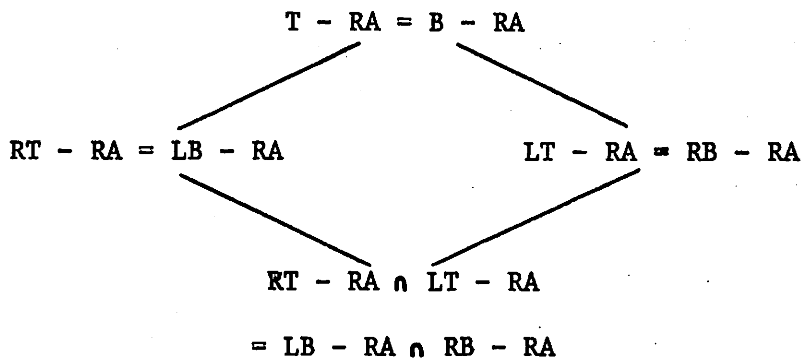
$\Sigma = \{a\}$

δ	a
q	q^2, q^3

One readily checks that T(A) is not RT - RA definable.

<u>Corollary 2</u>. The classes of RT - RA and LT - RA (LB - RA and RB - RA) are incomparable.

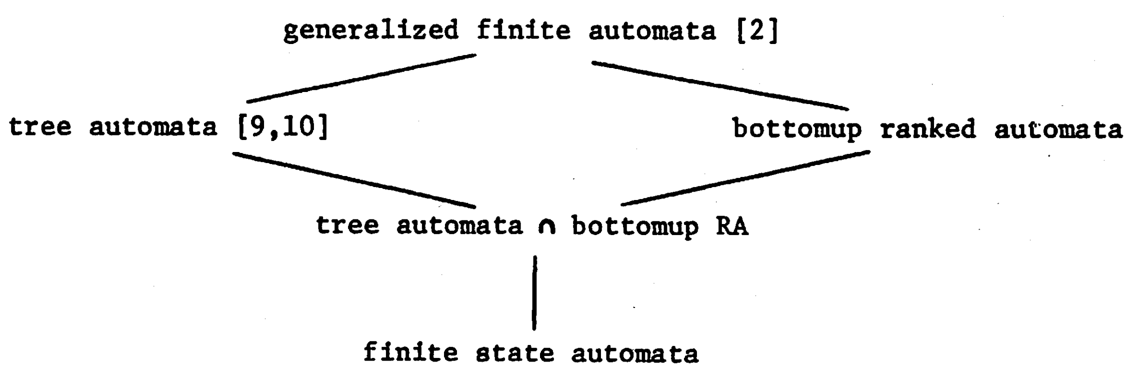
Thus, for the different classes of ranked automata we have the following hierarchy with respect to definable sets.



Moreover, since the above diagram is still true for ranked automata which are tree automata, but tree automaton definable sets are definable by deterministic bottomup tree automata [9], it follows

<u>Corollary 3</u>. There are tree automaton definable sets which are not RA - definable.

As a consequence, we get a hierarchy of automata with respect to definable sets of DOGs with only one root.



2. Closure Properties of RA definable sets

In this section it will be shown that RA definable sets are closed under union, intersection, projection, and inverse projection but not under complementation and product.

Theorem 1. If T and T' are RA definable subsets of $\mathcal{L}_\Sigma$, then so are $T \cup T'$ and $T \cap T'$.

Proof: Let A and A' be T - RA such that T(A) = T and T(A') = T'. Assume that $Q \cap Q' = \emptyset$.

Now define the T - RA $A'' = (Q'', \downarrow'', \uparrow'', \Sigma, \delta'', Q''_o, F'')$ by

$Q'' = Q \cup Q'$		$Q'' = Q \times Q'$
$Q''_o = Q_o \cup Q'_o$	or	$Q''_o = Q_o \times Q'_o$
$F'' = F \cup F'$		$F'' = F \times F'$
$\downarrow'' = \downarrow \cup \downarrow'$		$\downarrow'' = \{((q,q'),n) \mid (q,n) \in \downarrow \wedge (q',n) \in \downarrow'\}$
$\uparrow'' = \uparrow \cup \uparrow'$		$\uparrow'' = \{((q,q'),n) \mid (q,n) \in \uparrow \wedge (q',n) \in \uparrow'\}$
$\delta'' = \delta \cup \delta'$		$\delta'' = \{((q,q')^k, x^k, (\bar{q},\bar{q}')^\ell) \mid (q^k,x^k,\bar{q}^\ell) \in \delta \wedge (q'^k,x^k,\bar{q}'^\ell) \in \delta'\}$.

Then $T(A'') = T \cup T'$ or $T(A'') = T \cap T'$.

Corollary 1. LT - RA and RT - RA definable subsets of $\mathcal{L}_\Sigma$ are closed under union and intersection.

Obviously, RA definable sets are not closed under complementation. Even if we restrict ourselves to a certain subset of $\mathcal{L}_\Sigma$ and to the class of ranked automata which are tree automata we cannot prove a complementation closure theorem. To be more precise, if we specify a finite relation $\sigma \subseteq \Sigma \times N$ and define a T - RA $A_\sigma = (Q, \downarrow, \uparrow, \Sigma, \delta, Q_o, F)$ as follows

$Q = Q_o = F = \{q\}$

$\downarrow = \{(q,1)\}$

$\uparrow = \{(q,\ell) \mid \exists\ x \epsilon \Sigma : (x,\ell) \epsilon \sigma\}$

$\delta = \{(q,x,q^{\ell}) \mid (x,\ell) \epsilon \sigma\}$,

then we get (cf. [9])

<u>Theorem 2</u>. There is a relation σ and a tree RA definable subset T of $T(A_\sigma)$ such that $T(A_\sigma)$-T is not tree RA definable.

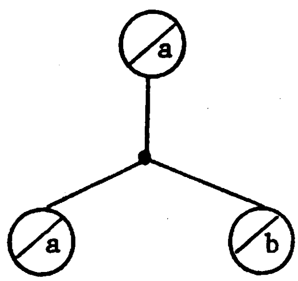
Proof: Let $\Sigma = \{a,b\}$, $\sigma = \{(a,1),(a,2),(b,1),(b,2)\}$,

and A be the tree RA in the proof of theorem 2 of section 1.

Clearly, $T(A) \subseteq T(A_\sigma)$.

Suppose $\bar{A}$ is a tree T - RA s.t. $T(\bar{A}) \supseteq T(A_\sigma) - T(A)$.

Then $\bar{A}$ accepts the following element of $T(A_\sigma) - T(A)$

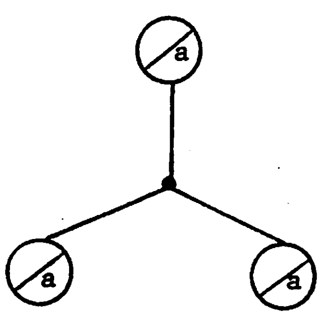


Thus $(\bar{q}_1, a, \bar{q}_2^{2})$ (i $\neq$ j does not necessarily imply $\bar{q}_i \neq \bar{q}_j$)

$(\bar{q}_2, a, \bar{q}_3)$

$(\bar{q}_2, b, \bar{q}_4)$ are transitions of $\bar{\delta}$ where $\bar{q}_1 \epsilon \bar{Q}_o$ and $\bar{q}_3, \bar{q}_4 \epsilon \bar{F}$.

But then $\bar{A}$ accepts also a tree which is in T(A):



Hence, $T(A_\sigma) - T(A)$ is not tree RA definable.

Now let $D = (\Gamma,<\!\cdot,\mu) \in \mathcal{L}_\Sigma$, $D' = (\Gamma,<\!\cdot,\mu') \in \mathcal{L}_{\Sigma'}$, and $\pi : \Sigma \to \Sigma'$.

D' is called the <u>projection of D w.r.t. π</u> if $\mu'(v) = \pi\mu(v)$ for all nodes v of Γ.

For $T \subseteq \mathcal{L}_\Sigma$, $\bar{\pi}(T)$ denotes the set of elements of $\mathcal{L}_{\Sigma'}$ which are projections of elements of T w.r.t. π.

For $T' \subseteq \mathcal{L}_{\Sigma'}$, $\bar{\pi}^{-1}(T')$ denotes the set of all $D \in \mathcal{L}_\Sigma$ for which there is at least one $D' \in T'$ such that D' is the projection of D w.r.t. π.

<u>Theorem 3</u>. If T is a RA definable subset of $\mathcal{L}_\Sigma$ and $\pi : \Sigma \to \Sigma'$, then $\bar{\pi}(T)$ is a RA definable subset of $\mathcal{L}_{\Sigma'}$.

Proof: Let $A = (Q, \downarrow, \uparrow, \Sigma, \delta, Q_o, F)$ be a T - RA s.t. $T(A) = T$.

Define $\delta' = \{(q^k, \pi(x)^k, \bar{q}^\ell) \mid (q^k, x^k, \bar{q}^\ell) \in \delta\}$

Clearly, the T - RA $A' = (Q, \downarrow, \uparrow, \Sigma', \delta', Q_o, F)$ defines $\bar{\pi}(T)$.

Conversely, we get:

<u>Theorem 4</u>. If T' is a RA definable subset of $\mathcal{L}_{\Sigma'}$ and $\pi : \Sigma \to \Sigma'$, then $\bar{\pi}^{-1}(T')$ is a RA definable subset of $\mathcal{L}_\Sigma$.

Proof: Let $A' = (Q, \downarrow, \uparrow, \Sigma', \delta', Q_o, F)$ be a T - RA s.t. $T(A') = T'$.

Define $\delta = \{(q^k, x^k, \bar{q}^\ell) \mid \exists\ x \in \Sigma : (q^k, \pi(x)^k, \bar{q}^\ell) \in \delta'\}$.

One easily checks that the T - RA $A = (Q, \downarrow, \uparrow, \Sigma, \delta, Q_o, F)$ defines $\bar{\pi}^{-1}(T')$.

<u>Remark</u>. As a projection means simply relabelling of a DOG it is not surprising to find that RA definable sets are closed under

'projection' and 'inverse projection'. The corresponding theorems for tree automata can be found in [9].

Now let D and D' be elements of $\mathcal{L}_\Sigma$, $\alpha\epsilon\Sigma^+$, $v_1,\ldots,v_n$ a segment of the sequence of leaves of D ordered by $\lessdot$ s.t. $\mu(v_1)\ldots\mu(v_n) = \alpha$, and $v_1',\ldots,v_n'$ a segment of the sequence of roots of D' ordered by $\lessdot'$ s.t. $\mu(v_1')\ldots\mu(v_n') = \alpha$. Assume that $V \cap V' = \emptyset$.

An <u>α-composition of D with D'</u> is a Σ-LOG D" with the property that

(1) $V'' = V \cup (V' - \{v_1',\ldots,v_n'\})$

$$\Gamma''(v) = \begin{cases} \Gamma(v) & \text{if } v\epsilon V - \{v_1,\ldots,v_n\} \\ \Gamma'(v_i') & \text{if } v = v_i,\ 1 \leq i \leq n \\ \Gamma'(v) & \text{otherwise} \end{cases}$$

(2)

$$\mu''(v) = \begin{cases} \mu(v) & \text{if } v\epsilon V \\ \mu'(v) & \text{otherwise} \end{cases}$$

(3) its linear order preserves $\lessdot$ and $\lessdot'$.

For subsets T, T' of $\mathcal{L}_\Sigma$ and $\mathcal{T}\subseteq\Sigma^+$ the (weak) <u>product of T and T' at $\mathcal{T}$</u> is the set $T \circ_{\mathcal{T}} T'$ of Σ-LOGs which can be obtained by composing an element of T with some elements of T' w.r.t. elements of $\mathcal{T}$.

<u>Remark</u>. Our concept of composition differs slightly from that given by Buttelmann [2], in that we don't specify the location of a composition and we don't require that $v_1',\ldots v_n'$ is the sequence of all roots of D'. The weak product is the analogue of that given by Magidor and Moran [6] for sets of trees.

<u>Theorem 5</u>. RA definable subsets of $\mathcal{L}_\Sigma$ are not closed under product.

Proof: Let the T - RA be given by

$Q = \{q_1, q_2, q_3\}$

$Q_o = \{q_1\}$

$F = \{q_2, q_3\}$

$\Sigma = \{a,b\}$

$\delta = \{(q_1^2, a^2, q_2^3),$

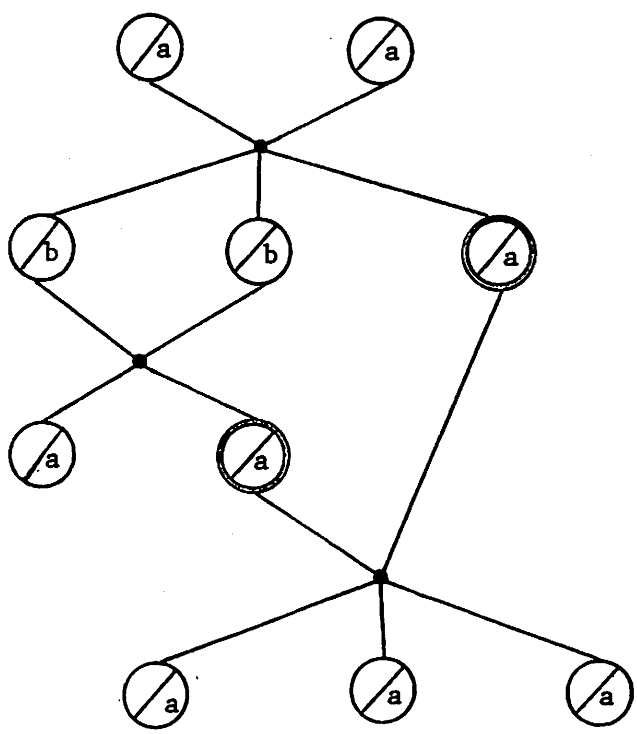
$(q_2, a, q_2),$

$(q_2^2, b^2, q_3^3),$

$(q_3, a, q_3)\}$

Suppose $\bar{A}$ is a T - RA s.t. $T(\bar{A}) \cong T(A) \circ_{\{a^2\}} T(A)$.

Then $\bar{A}$ accepts the following element of $T(A) \circ_{\{a^2\}} T(A)$



Thus $(\bar{q}_1^2, a^2, \bar{q}_2^3)$

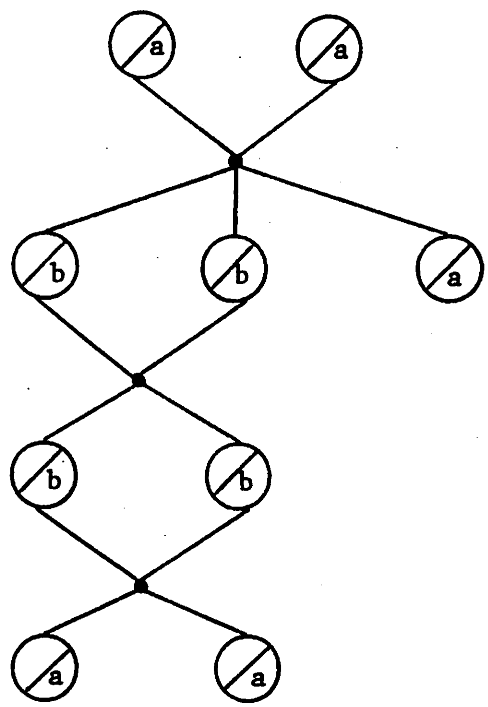
$(\bar{q}_2^2, b^2, \bar{q}_2^2)$ (i ≠ j does not necessarily imply $\bar{q}_i \neq \bar{q}_j$)

$(\bar{q}_2^2, a^2, \bar{q}_3^3)$

$(\bar{q}_2, a, \bar{q}_4)$

$(\bar{q}_3, a, \bar{q}_5)$ are transitions of $\bar{\delta}$ where $\bar{q}_1 \epsilon \bar{Q}_o$ and $\bar{q}_4, \bar{q}_5 \epsilon \bar{F}$.

But then $\bar{A}$ accepts a Σ-LOG which is not in $T(A) \circ_{\{a^2\}} T(A)$:



Hence, $T(A) \circ_{\{a^2\}} T(A)$ is not RA definable.

__Remark.__ As a consequence of theorem 5, we cannot prove a Kleene characterization theorem for RA definable sets as it was done by Thatcher and Wright [9] for tree automaton definable sets. A more promising approach could be that one recently given by Helton [3].

3. Languages induced by ranked automata

In this section we intend to study formal languages[1] induced by sequences of leaves or by certain paths in RA definable graphs. The notion of a control language for ranked automata yields some new problems.

It is well-known that tree automaton definable sets are exactly projections of sets of derivation trees of context-free grammars [10]. That does not hold for ranked automata which are tree automata since the class of languages defined by tree RA is a proper subclass of the class of context-free languages (hint: the regular language $\{a^m b^n | m, n \in \mathbb{N}\}$ is not the language of any tree RA).

However, if we associate with a tree T-RA A a finite set M of 'matrices' of transitions of A and require in addition of a run ρ of A upon a tree that, for any maximal sequence $v_1, \ldots, v_\nu$ of nodes in increasing order which have the same input node, the matrix

$$[(\rho(v_1), \mu(v_1), \rho(v'_{1,1}) \ldots \rho(v'_{1,\ell_1})), \ldots, (\rho(v_\nu), \mu(v_\nu), \rho(v'_{\nu,1}) \ldots \rho(v'_{\nu,\ell_\nu}))]$$

is in M then we get:

<u>Theorem 1</u>. For each context-free language L there is a tree T - RA A and a set M such that the language defined by A w.r.t. M is L.

Proof: Let L be a context-free language and $G = (V_N, V_T, P, S)$ be a context-free grammar generating L for which the start symbol S does not appear on the right of any production of G.

---

[1] For definitions of grammars and languages we refer to Chapter 2.2 and 2.3 of [4].

Define the T - RA A by:

$Q = \{q_o, q_1, q_2\}$

$Q_o = \{q_o\}$

$F = \{q_2\}$

$\Sigma = V_N \cup V_T$

$\downarrow = \{(q_o,1),(q_1,1),(q_2,1)\}$

$\uparrow = \{(q_1,\ell(w)) \mid \exists\ \xi \epsilon V_N : (\xi,w)\epsilon P\} \cup \{(q_2,1)\}$

$\delta = \{(q_o,S,q_1^{\ell(w)}) \mid (S,w)\epsilon P\} \cup \{(q_1,a,q_2) \mid a\epsilon V_T\}$
$\quad \{(q_1,\xi,q_1^{\ell(w)}) \mid \exists\ \xi\epsilon V_N - \{S\} : (\xi,w)\epsilon P\}.$

Let M consist of all matrices of the form

$[(q_1,v_1,\bar{q}_1^{\ell_1}),\ldots,(q_1,v_\nu,\bar{q}_\nu^{\ell_\nu})]$ where $v_1 \ldots v_\nu$ is a string appearing on the right of any production of G and

$(\bar{q}_1^{\ell_1},\ldots,\bar{q}_\nu^{\ell_\nu})$ is a ν-tuple with the property that for $i = 1,\ldots,\nu$

$$\bar{q}_i^{\ell_i} = \begin{cases} q_1^{\ell_i} & \text{if } v_i \epsilon V_N \text{ and } \ell_i = \ell(w) \text{ for some } w \text{ s.t. } (v_i,w)\epsilon P \\ q_2 & \text{if } v_i \epsilon V_T. \end{cases}$$

Then the language defined by A w.r.t. M is L.

Now we will investigate sets of strings of labels induced by certain paths of RA definable graphs.

Let A be an arbitrary T - RA and $D = (\Gamma,\lessdot,\mu)\epsilon T(A)$ with exactly one root and one leaf.

A directed path $v_1,\ldots,v_n$ of Γ that starts with the root and ends with the leaf is called <u>central</u> if

for each $v\epsilon V - \{v_1,\ldots,v_n\}$ there is an $i\epsilon\{1,\ldots,n\}$ such that v_i is an input node for v.

Denote by $L_{cp}(A)$ the set of strings of labels induced by central paths in Σ-LOGs with exactly one root and one leaf defined by A.

<u>Remarks</u>. (1) If $v_1, \dots v_n$ is a central path of Γ then a run ρ of A on D is completely determined by n − 1 transitions of the form

$$\left.\begin{array}{l} (\rho(v_1),\ \mu(v_1), \dots \rho(v_2) \dots) \\ (\dots \rho(v_2) \dots,\ \mu(v_2), \dots \rho(v_3) \dots) \\ \qquad\vdots \\ (\dots \rho(v_{n-1}) \dots,\ \mu(v_{n-1}), \dots \rho(v_n) \dots) \end{array}\right\} \qquad (*)$$

(2) $L_{cp}(A) \subseteq T(\tilde{A})$, i.e. $L_{cp}(A)$ is a (not necessarily regular) subset of the regular language defined by the finite state automaton $\tilde{A}$ induced by A.

If $\downarrow$, $\uparrow$ are maps and $Q_1^{\downarrow} = Q_1^{\uparrow} = Q$ then $L_{cp}(A) = T(\tilde{A})$. Thus the class of regular languages is a subclass of the class of languages of the form $L_{cp}(A)$. As the next theorem shows, this inclusion is proper.

<u>Theorem 2</u>. There is a T − RA A such that $L_{cp}(A)$ is context-free. There is also a context-free language L for which no RA A exists such that $L_{cp}(A) = L$.

Proof: To prove the first sentence define the T − RA A by

$Q = \{q, \bar{q}\}$

$Q_o = \{q\}$

$F = \{\bar{q}\}$

$\Sigma = \{a, b, \#\}$

$\downarrow = \{(q,1),\ (q,2)\}$

$\uparrow = \{(q,1),\ (q,2),\ (\bar{q},1)\}$

$\delta = \{(q,a,q^2),\ (q^2,b^2,q),(q,\#,\bar{q})\}$

Then $L_{cp}(A) = \{a^{m_1} b^{n_1} \ldots a^{m_k} b^{n_k} \# | k,m_i,n_i \geq \wedge m_1 + \ldots + m_k = n_1 + \ldots n_k\} \cup \{\#\}$ is a context-free language (the proof of this statement is tedious but elementary and left to the reader).

It is a much easier task to find an example proving the second sentence of the theorem. Almost any non-regular linear language considered in the literature is such a language, so for example the language $\{a^n b^n \# | n \geq 1\}$.

Remark. It is still an open question whether a language of the form $L_{cp}(A)$ can be context-sensitive.

In order to generalize this concept we introduce the notion of a control language (cf. [7]).

A control language for a T - RA A is a subset C of $(\mathbb{N} \times \mathbb{N})^+$.
To apply C, let $D = (\Gamma,\lessdot,\mu) \epsilon T(A)$ with exactly one root and one leaf.
Assume $v_1,\ldots,v_n$ is a central path of Γ and ρ is a run of A on D which is determined by n - 1 transitions of the form (*). Then, by definition, $\mu(v_1) \ldots \mu(v_n)$ is in $L_C(A)$ if the word

$$(1,\ell(\ldots\rho(v_2)\ldots))(\ell(\ldots\rho(v_2)\ldots),\ \ell(\ldots\rho(v_3)\ldots))\ldots(\ell(\ldots\rho(v_{n-1})\ldots),\ell(\ldots\rho(v_n)\ldots))$$

is in C.

Remarks. (1) Obviously, if $C = \{(k,\ell) | \exists\ q,\bar{q} \epsilon Q : (q,k) \epsilon \downarrow \wedge (\bar{q},\ell) \epsilon \uparrow\}^+$ then $L_C(A) = L_{cp}(A)$.

Consequently, each regular language can be represented in the form $L_C(A)$.

(2) In analogy to developmental systems [5] a ranked automaton A may be considered as a pattern-generating system, the control language C as a set of environmental inputs for A.

Two natural questions arise:

Which language L can be represented in the form $L = L_C(A)$?

If L is such a language what can we say about the complexity of C?

In what follows, we give some first results.

Theorem 3. Let A be a ranked automata such that $\downarrow$, $\uparrow$ are maps and $Q_1^{\downarrow} = Q_1^{\uparrow} = Q$.

If $C \subseteq \{(1,1)\}^+$ is regular then so is $L_C(A)$.

Proof: Let $C \subseteq \{(1,1)\}^+$ be a regular language. Then $\hat{C} = \{w \in \Sigma^+ \mid \exists\, w' \in C : \ell(w') = \ell(w)\}$ is also regular. Hence, the intersection of $\hat{C}$ with the regular language $L_{cp}(A) = T(\tilde{A})$ is regular.

Theorem 4. There is a T - RA A and a linear language C such that $L_C(A)$ is context-sensitive.

Proof: Define A by

$Q = \{q, \bar{q}\}$

$Q_o = \{q\}$

$F = \{\bar{q}\}$

$\Sigma = \{a, b, c, \#\}$

$\downarrow = \{(q,1),\ (q,2)\}$

$\uparrow = \{(q,1),\ (q,2),\ (\bar{q},1)\}$

$\delta = \{(q,a,q^2),\ (q^2,b,q),\ (q,c,q),\ (q,\#,\bar{q})\}$.

Let $C = \{(1,2)^m(1,1)^n(2,1)^m | m,n \geq 1\}$.

Then C is linear and $L_C(A) = \{a^n\ b^n\ c^n\ \# | n \geq 0\}$ context-sensitive.

Acknowledgement

The author would like to thank Michael A. Arbib for helpful suggestions on this work.

References

1. M. A. ARBIB and Y. GIVE'ON, Algebra Automata I: Parallel Programming as a Prolegomenon to the Categorical Approach, Information and Control 12 (1968), 331-345.

2. H. W. BUTTELMANN, On Generalized Finite Automata and Unrestricted Generative Grammars, Proceedings of the ACM Symposium on Theory of Computing (1971), 63-77.

3. F. J. HELTON, Algebra Automata, Stack Automata, and Operator Semigroups, Ph.D. thesis, Mathematics Department, Stanford University (1969).

4. J. E. HOPCROFT and J. D. ULLMAN, Formal Languages and their Relation to Automata, Addison-Wesley, Reading, Mass. (1969).

5. A. LINDENMAYER and G. ROZENBERG, Developmental Systems and Languages, Proceedings of the ACM Symposium on Theory of Computing (1972), 214-221.

6. M. MAGIDOR and G. MORAN, Finite Automata over Finite Trees, Technical Report No. 30, The Hebrew University of Jerusalem (1969).

7. A. SALOMAA, On Grammars with Restricted Use of Productions, Ann. Acad. Sci. Fennicœ, Ser. A.I. 454 (1969).

8. D. SCHÜTT, Baumautomaten, ITAS - Seminarbericht Nr. 36, Gesellschaft für Mathematik und Datenverarbeitung mbH Bonn (1971).

9. J. W. THATCHER and J. B. WRIGHT, Generalized Finite Automata Theory with an Application to a Decision Problem of Second-Order Logic, Math. Systems Theory 2 (1968), 57-81.

10. J. W. THATCHER, Generalized^2 Sequential Machine Maps, J. Comput. System Sci. 4 (1970), 339-367.