Synchronized Bottom-Up Tree Automata and L-Systems

E. Fachini and M. Napoli
Dipartimento di Informatica e Applicazioni
Universita' di Salerno
84100 Salerno, Italy

Introduction.

In this paper a new type of bottom-up tree automaton, called synchronized bottom-up tree automaton, is considered. This automaton processes a tree in a bottom-up way and one level at a time. Moreover, more than one transition function is allowed, but only one of them at a time can be applied to nodes at the same level of a tree.

The tree language recognized by these automata are the images, under projection, of the set of derivation trees of EPTOL languages.

The model introduced in this paper is a generalization of the bottom-up tree automaton. Its behaviour, relative to ETOL systems, is the same as the bottom-up tree automaton behaviour relating to context free grammars (7).

Furthermore, many properties of the bottom-up tree automata continue to hold for the class of automata here introduced. In fact, in the case that one transition function is allowed, the class of recognized tree languages is a boolean algebra and has a decidable equivalence problem. In the general case, the membership, the emptiness and the finiteness problems turn out to be decidable.

As it has been observed in the case of context-free languages, the introduction of tree automata recognizing sets of derivation trees of L-languages allows to state properties or to give simpler proofs of already known properties about the corresponding classes of L-languages.

We consider a subclass of trees in which a special symbol e labels a node representing an aborting computation, and we introduce a particular synchronized bottom-up tree automaton, called e-synchronized bottom-up tree automaton, which recognizes languages of this kind of trees.

The language recognized by these automata are the images, under a proper projection, of the sets of derivation trees of ETOL systems.

We are also able to construct, for every tree language L recognized by an e-synchronized bottom-up tree automaton, a synchronized bottom-up tree automaton which recognizes the set of trees obtained from L by pruning the dead branches. This result supplies a different method to construct an EPTOL system equivalent to a given ETOL system.

In section 1 preliminary concepts and definitions are introduced.

Section 2 contains the definitions of the considered classes of synchronized bottom-up tree automata and the proofs of their properties.

In section 3 the relationships between the synchronized bottom-up tree automata and L-systems are pointed out.

In section 4 some decision problems are dealt with.

1. Terminology.

We suppose the principal notions of tree languages and L-systems theories. We just give a definition of the set of derivation trees of an ETOL system, which is a slight modification of the one given by Engelfriet in (2).

Given a ranked alphabet Σ, we will consider the tree language over Σ containing only trees whose paths from the root to the leaves have the same lenght.

Definition 1.1. Let Σ be a ranked alphabet, consider the tree language $H_\Sigma=\bigcup_{k>=0} H^k_\Sigma$ where H^k_Σ is defined recursively as follows:
$H^0_\Sigma = \Sigma_0$
$H^{k+1}_\Sigma = \{a(t_1...t_r) \mid a \in \Sigma_r \text{ and } t_i \in H^k_\Sigma \text{ for } 1<=i<=r\}$.

Definition 1.2. Let $G=(\Sigma,\mathcal{P},S,\Delta)$ be an ETOL system. Given a symbol $e \notin \Sigma$, define Ω to be the ranked alphabet $\Sigma \cup \{e\}$ such that $\Omega_0=\Delta \cup \{e\}$, $\Omega_1=\{a\in\Sigma$ such that there exist $b \in \Sigma\cup\{\varepsilon\}$ and $P\in \mathcal{P}$ such that $a\to b$ is in $P\}\cup\{e\}$ and for $r>=2$, $\Omega_r=\{a\in\Sigma$ such that there exist $P\in \mathcal{P}$ and $w \in \Sigma^+$ such that $a\to w$ is in P and $|w|=r\}$ ($|x|$ is the lenght of the word x). For $a \in \Sigma$ and $\pi \in \mathcal{P}^*$, the set of derivation trees with top a and control word π, denoted $D^a_\pi(G)$, is defined recursively as follows:
1. for $a\in\Delta$, $a\in D^a_\lambda(G)$ (where λ is the empty word of $\mathcal{P}^*$);
2. for $a\in\Sigma$ and $P\in\mathcal{P}$, if $a\to\varepsilon$ is in P then $a(e^n)e$ is in $D^a_{\pi P}(G)$, for every $n>=0$ and for every $\pi \in \mathcal{P}^*$ such that $|\pi|=n$;
3. for $n>=1$, $a,a_1,...,a_n\in\Sigma$, $P\in\mathcal{P}$ and $t_1,...,t_n\in T_\Sigma$, if $a\to a_1...a_n$ is in P and $t_i\in D^{a_i}_\pi(G)$ for $1<=i<=n$, then $a(t_1...t_n)\in D^a_{\pi P}(G)$.

The set of derivation trees of G, denoted $D(G)\subseteq H_\Sigma$, is defined by $D(G)=\bigcup_{a\in S,\ \pi\in\mathcal{P}^*} D^a_\pi(G)$.

Note that if G is propagating then we do not need the symbol e and the clause 2. of the above definition.

For a given ranked alphabet Σ, besides the usual frontier function $fr:H_\Sigma\to\Sigma_0^*$, we will introduce the e-free frontier function for a symbol e, defined as follows.

Definition 1.3. Let Σ be a ranked alphabet. Chosen a symbol $e\in\Sigma_0$ we define a mapping $fr_e : H_\Sigma\to\Sigma_0^*$ recursively as follows:
i. for $a\in\Sigma_0$, $fr_e(a)=\varepsilon$ if $a=e$, otherwise $fr_e(a)=a$

ii. for $k>=1$, $a\in\Sigma_k$ and $t_i\in H_\Sigma$, for $1<=i<=k$,
$fr_e(a(t_1...t_k))=fr_e(t_1)fr_e(t_2)...fr_e(t_k)$.

It is easy to see that, for every ETOL system G, $fr_e(D(G))=L(G)$ and that for every EPTOL system G $fr(D(G))=L(G)$.

Definition 1.4. Given two ranked alphabets Σ and Δ such that $\Sigma_j\neq\emptyset \Rightarrow \Delta_j\neq\emptyset$ and given $R_j\subseteq\Sigma_j\times\Delta_j$, for every $j>=0$ such that $\Sigma_j\neq\emptyset$, a relabeling R with domain T_Σ and range T_Δ is the relation $R\subseteq T_\Sigma\times T_\Delta$ defined as follows:
$(a,b)\in R$ if $(a,b)\in R_0$;
$(a(t_1...t_k),b(t_1'...t_k'))\in R$ if $(t_i,t_i')\in R$ for $1<=i<=k$ and $(a,b)\in R_k$.

Given a relabeling R with domain T_Σ and range T_Δ and $t \in T_\Sigma$ let $R(t) = \{t' \in T_\Delta \mid (t,t') \in R\}$. If $L \subseteq T_\Sigma$, let $R(L)=\bigcup_{t\in L} R(t)$.

Definition 1.5. A projection P with domain T_Σ and range T_Δ is a relabeling such that for every $a \in \Sigma_j$ there exists a unique $b \in \Delta_j$ such that $(a,b) \in P$, for every j such that $\Sigma_j \neq \emptyset$.

In such a case we will write $t'=P(t)$ instead of $(t,t') \in P$.

A projection P with domain T_Σ and range T_Δ is said to be frontier preserving if for every $t \in T_\Sigma$ $fr(t)=fr(t')$ for every $t'=P(t)$.

2. Synchronized bottom-up tree automata.

In this section we introduce the k-synchronized bottom-up tree automata, abbreviated k-SBUTA, where k is an integer greater than zero, and we prove that the deterministic and non deterministic versions of such automata are equivalent.

We state that the class of tree languages recognized by any k-SBUTA is closed with respect to set theoretical union and intersection, whereas the closure with respect to complementation holds for the class of tree languages recognized by a 1-SBUTA.

We introduce a particular kind of k-synchronized bottom up tree automaton, called e-k-synchronized bottom-up tree automaton, in order to recognize trees in which some paths, labelled by words in $\{e\}^*$, represent aborted computations. Moreover, we prove that, given an e-k-synchronized bottom up tree automaton recognizing a tree language L, it is possible to construct a k-synchronized bottom-up tree automaton which recognizes the language obtained from L by cutting the dead subtrees from each tree.

Definition 2.1. Let $k \in N^+$. A k-NSBUTA is a 5-tuple $A=(\Sigma,Q,\rho_0,\rho,F)$ where

Σ is a ranked alphabet,
Q is a finite set of states,
$F \subseteq Q$ is a set of final states,
$\rho_0 \subseteq \Sigma_0 \times Q$ and
$\rho=\{\rho_1,\dots,\rho_k\}$ where $\rho_i = \{\rho_{i,j} \subseteq (Q^j \times \Sigma_j) \times Q \mid j>=1,\ \Sigma_j \neq \emptyset\}$.

The binary relation ρ_0 assignes initial states to the leaves; besides, if $(q_1,\dots,q_j,a,q) \in \rho_{i,j}$ then the state q can be assigned to a node labelled a if $q_1,\dots,q_j$ have been assigned to its sons. If k=1 then we will write ρ instead of ρ_1.

A k-NSBUTA $A=(\Sigma,Q,\rho_0,\rho,F)$ is deterministic if $\rho_0 : \Sigma_0 \to Q$ and $\rho_{i,j} : Q^j \times \Sigma_j \to Q$ are partial functions. A deterministic k-NSBUTA will be called k-SBUTA.

Let us define a binary relation $\vdash^{A,t,h}_{(N)SB} \subseteq (\bigcup_{n\in N} (\Sigma \times Q)^n)^2$ to describe the computation step of a k-(N)SBUTA A on $t \in T_\Sigma$, by using the h-th function (relation):

$(a_{1,1},q_{1,1},\dots,a_{1,r_1},q_{1,r_1},\dots,a_{s,1},q_{s,1},\dots,a_{s,r_s},q_{s,r_s}) \vdash^{A,t,h}_{(N)SB}$

$(a_1,q_1,\dots,a_s,q_s)$ if $1<=h<=k$ and for every $1<=i<=s$ it holds that $a_i \in \Sigma_{r_i}$, $r_i>0$, $a_{i,1},\dots,a_{i,r_i}$ are the labels of the sons of the nodes labelled a_i and $\beta_{h,r_i}(q_{i,1},\ldots,q_{i,r_i},a)=q_i$ $((q_{i,1},\ldots,q_{i,r_i},a,q_i)\ \beta_{h,r_i})$.

We define $x_1 \overset{A,t,\pi+}{\vdash} x_n$ if $\pi = j_1\ldots j_{n-1}$ and there exist $x_2,\ldots,x_{n-1} \in \bigcup_{n\in N} (\Sigma\times Q)^n$ such that $x_i \overset{A,t,j_i}{\vdash}$ (N)SB x_{i+1} for $1<=i<=n-1$, and $x1 \overset{A,t,\pi *}{\vdash}$ (N)SB x_n if $x_1=x_n$ and $\pi=\lambda$ or $x_1 \overset{A,t,\pi+}{\vdash}$ (N)SB x_n for some $\pi\in\{1,\ldots,k\}^+$.

We say that $t \in T_\Sigma$ is q-π-accepted by A if $q \in Q$ and $(a_1,q_1,\ldots,a_n,q_n) \overset{A,t,\pi *}{\vdash}$ (N)SB (a,q) where $a_1,\ldots,a_n$ are the labels of the leaves and a is the label of the root of t, $(a_i,q_i)\in\beta_0$ $(\beta_0(a_i)=q_i)$ for $1<=i<=n$ and $\pi\in\{1,\ldots k\}^*$.

We say that $t\in T_\Sigma$ is q accepted by A if it is q-π-accepted by A for some $\pi\in\{1,\ldots,k\}^*$. Moreover, $t\in T_\Sigma$ is accepted by A if it is q-accepted by A for some $q\in F$.

Let L(A) the set of trees accepted by a k-(N)SBUTA A. Note that for every k-NSBUTA $A=(\Sigma,Q,\beta_0,\beta,F)$, $L(A)\subseteq H_\Sigma$.

A language L is said to be k-(N)SB-recognizable if there exists a k-(N)SBUTA A such that L=L(A).

Let k-NSB-RECOG (k-SB-RECOG) be the set of the k-NSB-recognizable (k-SB-recognizable) languages and NSB-RECOG (SB-RECOG) the set of 1-NSB-recognizable (1-SB-recognizable) languages.

Theorem 2.1. For every $k>=1$, k-NSB-RECOG = k-SB-RECOG.
Proof. The proof exploits the usual subset construction.

In the following we will prove that $\bigcup_{k>=1}$ k-SB-RECOG and SB-RECOG are closed with respect to union and intersection, and that the closure with respect to the complementation holds for SB-RECOG.

Theorem 2.2. $\bigcup_{k>=1}$ k-SB-RECOG is closed with respect to intersection and union.

Proof. Let $A=(\Sigma,Q,\beta_0,\beta,F)$ be a k-SBUTA and $A'=(\Sigma',Q',\beta_0',\beta',F')$ be a k'-SBUTA. We can suppose, without loss of generality, that $Q\cap Q'=\emptyset$ and that $k>=k'$.

We now construct a k-NSBUTA A" such that $L(A")=L(A)\cup L(A')$, according to theorem 2.1 there exists a k-SBUTA equivalent to A".

Let $A"=(\Sigma",Q",\beta_0",\beta",F")$ be the k-NSBUTA defined as follows:
$\Sigma"=\Sigma\cup\Sigma'$, $Q"=Q\cup Q'$, $F"=F\cup F'$,
$\beta_0"=\{(a,q)\mid a\in\Sigma_0\cup\Sigma'_0,\ q\in Q\cup Q'$ and $\beta_0(a)=q$ or $\beta_0'(a)=q\}$,
$\beta"_{j,r}=\{(q_1,\ldots,q_r,a,q)\mid a\in\Sigma_r\cup\Sigma'_r,\ q_1,\ldots,q_r,q\in Q\cup Q'$ and $\beta_{j,r}(q_1,\ldots,q_r,a)=q$ or $\beta'_{j,r}(q_1,\ldots,q_r,a)=q\}$, for every $1<=j<=k$.

Consider now the k"-SBUTA A" such that $L(A")=L(A)\cap L(A')$, defined as follows:
$k"=kk'$, $\Sigma"=\Sigma\cap\Sigma'$, $Q"=Q\cap Q'$, $F"=F\cap F'$,
$\beta_0"(a)=(q,p)$ if $\beta_0(a)=q$ and $\beta_0'(a)=p$ for $a\in\Sigma_0\cap\Sigma'_0$ and
$\beta"_{(j-1)k'+l,r}((q_1,p_1),\ldots,(q_r,p_r),a)=(q,p)$ if $\beta_{j,r}(q_1,\ldots,q_r,a)=q$ and $\beta'_{l,r}(p_1,\ldots,p_r,a)=p$, for every $1<=j<=k$, $1<=l<=k'$ and $a\in\Sigma_r\cap\Sigma'_r$.

In both cases the proof is carried out by induction on the input tree.

Let $^\frown L=H-L$ be the complement of L in H .

Theorem 2.3. The class of SB-recognizable subset of H is a boolean algebra.

Proof. From the constructions given in the proof of theorem 2.2, it immediately follows that SB-RECOG is closed with respect to union and intersection. Let $A=(\Sigma,Q,\beta_0,\beta,F)$ be a SBUTA recognizing a language L, we construct a SBUTA $A'=(\Sigma,Q',\beta_0',\beta',F')$ recognizing ^L. Let $Q'=Q\cup\{p\}$, where $p\notin Q$, $F'=(Q-F)\cup\{p\}$, $\beta_0'(a)=\beta_0(a)$ if $\beta_0(a)$ is defined, otherwise $\beta_0'(a)=p$, $\beta'_r(q_1,\ldots,q_r,a)=\beta_r(q_1,\ldots,q_r,a)$ if $q_i\in Q$ and $\beta_r(q_1,\ldots,q_r,a)$ is defined, otherwise $\beta'_r(q_1,\ldots,q_r,a)=p$.

In the following we will prove that k-SB-RECOG is closed with respect to relabeling. As a consequence we obtain the closure with respect to projection and inverse projection.

Theorem 2.4. For every k>=1, k-SB-RECOG is closed with respect to relabeling.

Proof. Given a k-SBUTA $A=(\Sigma,Q,\beta_0,\beta,F)$ and a relabeling R with domain T_Σ and range T_Δ we will construct a k-NSBUTA $A'=(\Delta,Q,\beta_0',\beta',F)$, such that $L(A')=R(L(A))$, defined as follows: $\beta_0'=\{(b,q) \mid (a,b)\in R_0$ and $\beta_0(a)=q$ for $a\in\Sigma_0\}$, $\beta'_{j,r}=\{(q_1,\ldots,q_r,b,q)$ if $(a,b)\in R_r$ and $\beta_{j,r}(q_1,\ldots,q_r,a)=q$ for $a\in\Sigma_r\}$.

By induction on t it is easy to see that

i. for every $t\in H_\Sigma$, if t is q-π-accepted by A and there exists $t'\in H_\Delta$ such that $(t,t')\in R$ then t' is q-π-accepted by A';

ii. for every $t\in H_\Delta$, if t is q-π-accepted by A' then there exists $t'\in H_\Sigma$ such that $(t',t)\in R$ and t' is q-π-accepted by A.

From i. and ii. the thesis follows.

Corollary 2.1. For every k>=1, k-SB-RECOG is closed with respect to (frontier preserving) projection and inverse projection.

Proof. Obvious.

Since a tree can represent a computation process, it is quite natural to consider that some computation paths may abort. In order to represent trees with aborting paths in H_Σ, we will introduce a special symbol e to label the nodes of a chain which completes an aborting path. For example the tree $t=(a(b(d(d)b(bb))c)\in T_\Sigma$ can be represented by the tree $t'=(a(b(d(d)b(bb))c(e(e)))\in H_{\Sigma\cup\{e\}}$.

Definition 2.2. Given a ranked alphabet Σ, we will call e-ranked alphabet the alphabet $\Sigma'=\Sigma\cup\{e\}$ such that $e\notin\Sigma$, $\Sigma'_0=\Sigma_0\cup\{e\}$, $\Sigma'_1=\Sigma_1\cup\{e\}$ and $\Sigma'_j=\Sigma_j$ for every j>=1.

Definition 2.3. Let $k\in N^+$. A non deterministic synchronized bottom-up tree automaton over an e-ranked alphabet Σ, abbreviated e-k-NSBUTA, is a k-NSBUTA $A=(\Sigma,Q,\beta_0,\beta,F)$ such that there is a special state $q_e\in Q$ satisfying the following conditions:

i. if $(a,q)\in\beta_0$ then $a=e \Leftrightarrow q=q_e$,

ii. if $(q,a,q')\in\beta_{i,1}$ then $a=e\Leftrightarrow q=q'=q_e$ and $q'=q_e\Rightarrow a=e \wedge q=q_e$,

iii. for j>=2, if $(q_1,\ldots,q_j,a,q_{j+1})\in\beta_{i,j}$ then $q_h\neq q_e$ for every $1<=h<=j+1$,

iv. for every $1<=i<=k$ $(q_e,e,q_e)\in\beta_{i,1}$.

We will call e-k-SBUTA a deterministic bottom-up tree automaton over an e-ranked alphabet. The set of languages recognized by any e-k-(N)SBUTA will be called e-k-(N)SB-RECOG.

Lemma 2.1. For every e-k-NSBUTA A there exists an e-k-SBUTA A' such that L(A)=L(A').

Proof. A slight modification of the usual subset construction suffices to take in account the special symbol e.

Definition 2.4. Given two e-ranked alphabets Σ and Δ. a relabeling R with domain T_Σ and range T_Δ is said to be an e-relabeling if for every $(a,b) \in R_0 \cup R_1$ we have that a=e iff b=e. A projection which is an e-relabeling is called e-projection.

Lemma 2.2. For every $k>=1$, e-k-SB-RECOG is closed with respect to e-relabeling.

Proof. The construction given in the theorem 2.4 applies to the case of an e-k-SBUTA as well.

Given an e-ranked alphabet Σ, let Σ_{pe} be the ranked alphabet defined by: $(\Sigma_{pe})_0=\Sigma_0$ and $(\Sigma_{pe})_k=\bigcup_{r>=k}\Sigma_r$. Let pe be such a function that
$pe : H \to H_{pe}$ and $pe(a)=a$, for every $a \in \Sigma_0$,
$pe(a(t_1 \ldots t_k))=e$ if $pe(t_i)=e$ for every $1<=i<=k$,
$pe(a(t_1 \ldots t_k))=a(pe(t_{l_1}) \ldots pe(t_{l_m}))$ if $pe(t_{l_s})\neq e$ for $1<=s<=m$,
$0<l_1<\ldots<l_m<=k$ and $pe(t_{i_r})=e$ for every $i_r \neq l_s$, $1<=i_r<=k$.

In the following the concept of state diagram will be useful.

Definition 2.5. Given a k-SBUTA $A=(\Sigma,Q,\rho_0,\rho,F)$, let the state diagram $G=(V,E)$ be the directed labelled graph such that $V=2^Q$ and $(U,U') \in E$ with label j iff $1<=j<=k$ and $U'=\{\rho_{j,r}(q_1,\ldots,q_r,a) \mid a \in \Sigma_r$ and $q_i \in U$ for $1<=i<=r\}$.

Let $S_j(U)$ be the node U' such that $(U,U') \in E$ with label j.
Let $P(G) \subseteq 2^Q \times \{1,\ldots,k\}^* \times 2^Q$ such that $(U,\pi,U') \in P(G)$ iff $\pi \in \{1,\ldots,k\}^+$ and there exists a path from U to U' in G labelled π or $\pi=\lambda$ and U=U'. Let $C^U(G) = \{U' \in 2^Q \mid$ there exists $\pi \in \{1,\ldots,k\}^*$ and $(U,\pi,U') \in P(G)\}$.

Theorem 2.5. For every e-k-SBUTA A, there exists a k-SBUTA A' such that $L(A')=pe(L(A))-\{e\}$.

Proof. Given an e-k-SBUTA $A=(\Sigma,Q,\rho_0,\rho,F)$, let $G=(2^Q,E)$ be its state diagram. In the k-NSBUTA which we will construct, the states will be couples (q,U) where $q \in Q$ and U is a subset of Q containing the states p such that there exists a p-π-accepted tree, for a proper π, whose frontier belongs to $\{e\}^*$. Consider the k-NSBUTA $A'=(\Sigma',Q',\rho_0',\rho',F')$ where $\Sigma'=\Sigma_{pe}$, $Q'=Q\times 2^Q$, $F'=F\times 2^Q$, $\rho_0'=\{(a,(\rho_0(a),\{q_e\})) \mid a \in \Sigma_0-\{e\}\}$ and ρ' is defined as follows:
if $\rho_{j,r}(q_1,\ldots,q_r,a)=q$ then $((q_{i_1},U),\ldots,(q_{i_l},U),a,(q,S_j(U))) \in \rho'_{j,l}$ for $1<=l<=r$, $1<=i_1<\ldots<i_l<=r$ and $U \in C^{\{q_e\}}(G)$ iff there exist $h_1,\ldots,h_{r-l} \in$

$\{1,\ldots,r\} - \{i_1,\ldots,i_l\}$ such that $q_{h_n} \in U$, for every $1 <= n <= r-1$.

By induction on $|\pi|$ it is easy to prove that, for every $q \in Q$, $U \in C^{\{q_e\}}(G)$ and $\pi \in \{1,\ldots,k\}^*$ such that $(\{q_e\},\pi,U) \in P(G)$ it holds that $q \in U$ iff there exists $t \in H_\Sigma$ such that $pe(t)=e$ and t is q-π-accepted by A. By induction on t and by exploiting this result it is easy to prove that:

1. if $t \in H_\Sigma$ is q-π-accepted by A and $pe(t)=e$, then $pe(t)$ is (q,U)-π-accepted by A' for $(\{q_e\},\pi,U) \in P(G)$.
2. if $t \in H_{\Sigma pe}$ is (q,U)-π-accepted by A' then $(\{q_e\},\pi,U) \in P(G)$ and there exists $t' \in H_\Sigma$, such that $pe(t)=t'$, which is q-π-accepted by A.

So that $pe(L(A))-\{e\} = L(A')$. According to theorem 2.1 there exists a k-SBUTA A" such that L(A")=L(A').

3. L-systems and synchronized bottom-up tree automata.

In this section we relate the considered classes of synchronized bottom-up tree automata with classes of L-systems. We will prove that a language L is an ETOL language iff it coincides, up to the empty word, with the set of frontiers of trees belonging to a language recognized by a synchronized bottom-up tree automaton.

As an example of a result which can be obtained by exploiting the above correspondence between $\bigcup_{k>=1}$ k-SB-RECOG and ETOL languages, we show that for every ETOL system there exists a structurally equivalent ETOL system such that in every table the right hand side of a rule uniquelly determines its left hand side.

Theorem 3.1. For every ETOL system G, there exists an e-k-SBUTA A such that L(A)=D(G).

Proof. Given an ETOL system $G=(\Sigma,\mathcal{P},S,\Delta)$, consider the e-k-SBUTA $A=(\Omega,Q,\beta_0,\beta,F)$ such that $k=|\mathcal{P}|$, Ω is the e-ranked alphabet $\Sigma \cup \{e\}$, $Q=\{q_a \mid a \in \Sigma\} \cup \{q_e\}$, $F=\{q_a \mid a \in \Delta\}$,
$\beta_0(a)=q_a$ if $a \in \Delta \cup \{e\}$,
$\beta_{P,1}(q_e,e)= q_e$ for every $P \in \mathcal{P}$,
$\beta_{P,1}(q_e,a)=q_a$ for every $a \in \Omega_1$ such that $a \rightarrow \varepsilon \in P$ and finally
$\beta_{P,r}(q_{a_1},\ldots,q_{a_r},a)=q_a$ for every $a \in \Sigma_r$ such that $a \rightarrow a_1\ldots a_r \in P$, for $r>0$.

It is easy to see that L(A)=D(G).

Theorem 3.2. For every EPTOL system G, there exists a k-SBUTA A such that L(A)=D(G).

Proof. By eliminating the construction rules regarding the special symbol e, the construction given in the previous proof supplies the wanted k-SBUTA.

Theorem 3.3. Every e-k-SB-recognizable subset of H_Σ is the image, under a frontier preserving e-projection, of the set of derivation trees of an ETOL system $G=(\Omega,\mathcal{P},S,\Delta)$ for some alphabet Ω.

Proof. Let $A=(\Sigma,Q,\beta_0,\beta,F)$ be an e-k-SBUTA. Let us consider the alphabet $\Omega=\bigcup_{r>=1}(\Sigma_r-\{e\})\times(Q\cup\{R\})\cup(\Sigma_0-\{e\})$ where $R\notin\Sigma\cup Q$. Consider the ETOL $G=(\Omega,\mathcal{P},S,\Delta)$ defined as follows:
$\Delta=\{a\in\Sigma_0-\{e\}\mid\beta_0(a)$ is defined$\}$,
$S=\{(a,q)\mid a\in\Sigma_r,\ r>=0,\ q\in F\}\cup\{a\in\Sigma_0\mid\beta_0(a)\in F\}$,
$\mathcal{P}$ contains the tables $\mathbf{J}=\{(a,q)\rightarrow(a,q_1),\dots,(a,q_r)\mid\beta_{\mathbf{J},r}(q_1,\dots,q_r,a)=q$ for every $a\in\Sigma_r$, $r>0$ and $q\neq q_e$ and $q_j\neq q_e$ for $1<=j<=r\}\cup\{(a,q)\rightarrow\varepsilon\mid\beta_{\mathbf{J},1}(q_e,a)=q\}\cup\{(a,q)\rightarrow(a,R)\dots(a,R)\mid q\in Q,\ a\in\Sigma_r$ and $\beta_{\mathbf{J},r}(q_1,\dots,q_r,a)$ is not defined for every $q_i\in Q$, $1<=i<=r$ or $a\in\Sigma_r$ and $q=R\}\cup\{a\rightarrow R$ for every $a\in\Sigma_0\cup\{R\}\}$. Let Ω' be the ranked alphabet $\Omega\cup\{e\}$, hence $D(G)\subseteq H_{\Omega'}$. Consider the frontier preserving e-projection P with domain $T_{\Omega'}$ and range T_Σ such that $P(a)=a$ for every $a\in\Omega'_0$ and $P((a,q))=a$ for every $a\in\Sigma_r$ and $q\in Q\cup\{R\}$.

By induction on $t\in H_\Sigma$ it is easy to prove that $t\in L(A)$ iff there exists $t'\in D(G)$ such that $P(t')=t$, hence the thesis holds.

Theorem 3.4. Every k-SB-recognizable subset of H_Σ is the image, under a frontier preserving projection, of the set of derivation trees of an EPTOL system $G=(\Omega,\mathcal{P},S,\Delta)$ for some alphabet Ω.

Proof. The construction of theorem 3.3 applies to a k-SBUTA by giving an EPTOL system.

Note that the above results supply a new proof of the existence of an EPTOL system equivalent to a given ETOL system. In fact given an ETOL system $G=(\Sigma,\mathcal{P},S,\Delta)$ generating a language U, from theorem 3.1 there exists an e-k-SBUTA A such that $L(A)=D(G)$. From theorem 2.5 there exists a k-SBUTA A' such that $L(A')=pe(L(A))-\{e\}$ and $U-\{\varepsilon\}=fr_e(L(A))-\{\varepsilon\}=fr(L(A'))$. According to theorem 3.2 there exists an EPTOL system G' and a frontier preserving projection P such that $L(A')=P(D(G'))$, so that G' generates $U-\{\varepsilon\}$.

Corollary 3.1. A subset $U\subseteq\Delta^*$ is an E(T)OL language iff $U-\{\varepsilon\}=fr(V)$ for some (k)-SB-recognizable $V\subseteq H_\Delta$.

Proof. Given an ETOL language U, consider the EPTOL system $G=(\Sigma,\mathcal{P},S,\Delta)$, generating $U-\{\varepsilon\}$ and the k-SBUTA A such that $L(A)=D(G)$. Chosen a symbol a_0 in Δ, consider the following ranking of Δ: $a_0\in\Delta_n$ for every n such that $\Sigma_n\neq\emptyset$, $a\in\Delta_0$ for every $a\in\Delta$, and the frontier preserving projection P with domain T_Σ and range T_Δ such that $(a,a)\in P_0$ for every $a\in\Sigma_0$ and $(a,a_0)\in P_n$, for every $a\in\Sigma_n$, $n>0$.

In accordance to the corollary 2.1, there exists a k-SBUTA A' such that $L(A')=P(L(A))$. Furthermore $U-\{\varepsilon\}=fr(P(D(G)))=fr(L(A'))$.

Viceversa, let us consider a k-SB-recognizable subset $V\subseteq H_\Delta$. From theorem 3.4, there exists an EPTOL system $G=(\Omega,\mathcal{P},S,\theta)$ and a frontier preserving projection P with domain T_Ω and range T_Δ such that $P(D(G))=V$. Hence, $fr(P(D(G)))=fr(D(G))=fr(V)=U$ and so U is an ETOL language.

From the constructions given in theorems 3.1 and 3.4, it immediately follows that $U\subseteq\Delta^*$ is an EOL language iff $U-\{\varepsilon\}=fr(V)$ for some SB-recognizable $V\subseteq H_\Delta$.

Definition 3.1. Given an e-ranked alphabet Ω consider the e-ranked alphabet Ω^* defined by $(\Omega^*)_0=\Omega_0$ and $(\Omega^*)_k=\{*\}$ for every k such that $\Omega_k\neq\emptyset$. Moreover consider the frontier preserving e-projection R^* with domain T_Ω and range T_{Ω^*} such that $(a,a)\in(R^*)_0$ for every $a\in\Omega_0$ and

$(a,*) \in (R^*)_r$ for every $a \in \Omega_r$, $r>0$. Two ETOL systems are called structurally equivalent if $R^*(D(G_1))=R^*(D(G_2))$.

Definition 3.2. An ETOL system $G=(\Sigma,\mathcal{P},S,\Delta)$ is invertible if for every $P \in \mathcal{P}$ the right hand side of a rule in P uniquelly determines its left hand side.

Theorem 3.5. For every ETOL system G there exists a structurally equivalent invertible ETOL system G'.

Proof. Let $G=(\Sigma,\mathcal{P},S,\Delta)$ be an ETOL system with $|\mathcal{P}|=k$. Let us consider the e-k-SBUTA $A=(\Omega,Q,\beta_0,\beta,F)$ such that $L(A)=D(G)$ whose construction is given in theorem 3.1. Note that β_0 is an injective function. Consider the frontier preserving e-projection R^* above defined, according to lemma 2.2 there exists an e-k-SBUTA $A'=(\Omega^*,Q',\beta_0',\beta',F')$ such that $R^*(D(G))=L(A')$. Note that $\beta_0'=\beta_0$. Let us consider the ETOL system $G'=(\{*\}X(Q\cup\{R\})\cup\Delta,\mathcal{P}',\{(*,q) \mid q \in F'\},\Delta)$ constructed in the proof of theorem 3.3. Since A' is deterministic then G' is invertible, furthermore it is obvious that G' and G are structurally equivalent.

4. Decision problems.

In this section some decision problems about k-SB-RECOG are dealt with. In particular the membership, the emptiness and the finiteness problems for $\biguplus_{k>=1}$ k-SB-RECOG and the equivalence problem for SB-RECOG are shown to be decidable. As a consequence of these results we prove that the property of structural ambiguity is decidable for ETOL systems and the structural equivalence is decidable for EOL systems.

Theorem 4.1. The membership, the emptiness and the finiteness problems are decidable in the class $\biguplus_{k>=1}$ k-SB-RECOG.

Proof. Since every k-SB-recognizable tree language $L \subseteq H_\Sigma$, viewed as a language over the alphabet $\Sigma \cup \{(,)\}$, is an EPTOL language, the thesis follows from the decidability of the considered problems for EPTOL languages (see (6)).

Definition 4.1. A set of trees U is structurally ambiguous if U contains two different trees t_1 and t_2 such that $R^*(t_1)=R^*(t_2)$.

Corollary 4.1. It is decidable whether a k-SB-recognizable tree language is structurally ambiguous.

Proof. The proof is analogous to the one given by Paull and Unger in (5) for the class of tree languages recognized by finite state bottom-up tree automata. It suffices to consider in $H_{\Sigma\times\Sigma}$ the tree language W which contains all the trees such that the frontiers are labelled with pairs (a,a) and at least one internal node is labelled with a pair (a,a') with a=a'.
Clearly W$\in$SB-RECOG. Consider the relabelings R_1 and R_2 with domain T_Σ and range $T_{\Sigma\times\Sigma}$ defined as follows:
$(a,(a,a)) \in (R_i)_0$ for every $a \in \Sigma_0$, $i=1,2$,

$(a,(a,a')) \in (R_1)_r$ for every $a,a' \in \Sigma_r$ and $r>0$ and
$(a,(a',a)) \in (R_2)_r$ for every $a,a' \in \Sigma_r$ and $r>0$.

If $U \in$ k-SB-RECOG then, according to the theorems 2.2 and 2.4 it holds that $R_1(U) \cap R_2(U) \cap W \in k^2$-SB-RECOG. Finally, U is structurally ambiguous iff $R_1(U) \cap R_2(U) \cap W$ is non empty.

Theorem 4.2. The equivalence problem for SB-RECOG is decidable.

Proof. It immediately follows from theorems 2.3 and 4.1.

Corollary 4.2. It is decidable whether two EOL systems are structurally equivalent or not.

Proof. Given two EOL systems G_1 and G_2, $R^*(D(G_1))$ and $R^*(D(G_2))$ are SB-recognizable and then, according to theorem 4.2, it is decidable if $R^*(D(G_1))=R^*(D(G_2))$.

We would like to thank professor Jost Engelfriet for his valuable and helpful comments on this paper.

References.

(1) J. Engelfriet, Bottom-up and top-down tree transformations. A comparison, Math. Syst. Theory 9, 3(1975), 198-231.
(2) J. Engelfriet, Surface tree languages and parallel derivation trees, Theor. Comp. Sc. 2(1976), 9-27.
(3) J. Engelfriet, G. Rozenberg, G. Slutzki, Tree transducers, L-systems and two machines, J. C. S. S. 20(1980), 150,202.
(4) J. van Leeuwen, Variations of a new machines model, Proc. 17th Annual Symp. Found. Comput. Sci., Houston, Texas (1976), 228-235.
(5) M. C. Paull, S. H. Unger, Structural equivalence of context-free grammars, J. C. S. S. 2(1968), 427-463.
(6) G. Rozenberg, A. Salomaa, The mathematical theory of L-systems, Academic Press, New York (1980).
(7) J. W. Tatcher, Generalized sequential machines maps, J. C. S. S. 4 (1970), 339-367.