Matrix Grammars – Definition I

Definition:
i) A matrix grammar is a quintuple $G = (N, T, M, S, F)$, where

— N, T and S are specified as for a context-free grammar,

— $M = \{m_1, m_2, \ldots, m_n\}$ is a finite set of finite sequences of context-free rules (i. e., for $1 \leq i \leq n$,

$$m_i = (A_{i,1} \to w_{i,1}, A_{i,2} \to w_{i,2}, \ldots, A_{i,r_i} \to w_{i,r_i})$$

for some $r_i \geq 1$, $A_{i,j} \in N$, $w_{i,j} \in (N \cup T)^*$, $1 \leq j \leq r_i$) and

— F is a subset of the rules occurring in the matrices m_i, $1 \leq i \leq n$.

Matrix Grammars – Definition II

ii) Let $m = (A_1 \to w_1, A_2 \to w_2, \ldots, A_r \to w_r) \in M$. We say that $y \in V^*$ is derived from $x \in V^+$ by m (and write $x \Longrightarrow_m y$), if there exist words $x_1, x_2, \ldots, x_{r+1}$ such that

— $x = x_1$ and $y = x_{r+1}$,

— for $0 \leq i \leq r-1$, either $x_i = x_i' A_i x_i''$ and $x_{i+1} = x_i' w_i x_i''$ or A_i does not occur in x_i, $x_{i+1} = x_i$ and $A_i \to w_i \in F$.

iii) The language $L(G)$ generated by G consists of all words $z \in T^*$, which have a derivation

$$S \Longrightarrow_{m_{i_1}} w_1 \Longrightarrow_{m_{i_2}} w_2 \Longrightarrow_{m_{i_3}} \ldots \Longrightarrow_{m_{i_t}} = w_t = z$$

with $t \geq 1$ and $m_{i_j} \in M$ for $1 \leq j \leq t$.

Matrix Grammars – Normal Form

Definition: A matrix grammar $G = (N, T, M, S, F)$ is in normal form, if the following conditions are satisfied:

- $N = N_1 \cup N_2 \cup \{S, Z, \#\}$, $S, Z, \# \notin N_1 \cup N_2$, $N_1 \cap N_2 = \emptyset$,
- each matrix of M has one of the following forms
  — $(S \rightarrow XA)$ with $X \in N_1$, $A \in N_2$,
  — $(X \rightarrow Y, A \rightarrow w)$ with $X, Y \in N_1$, $A \in N_2$, $w \in (N_2 \cup T)^*$,
  — $(X \rightarrow Y, A \rightarrow \#)$ with $X \in N_1$, $Y \in N_1 \cup \{Z\}$, $A \in N_2$,
  — $(Z \rightarrow \lambda)$,
- there is only one matrix of the form $(S \rightarrow XA)$ in M,
- F consists of all rules of the form $A \rightarrow \#$ with $A \in N_2$.

Matrix Grammars – Results

$\mathcal{L}(MAT)$ denotes the set of all languages, which can be generated by matrix grammars.

Theorem: $\mathcal{L}(MAT) = \mathcal{L}(RE)$.

Theorem: For each recursively enumerable language L, there is a matrix grammar G in normal form such that $L(G) = L$.

Grammar Systems – Definition

Definition: i) A grammar system with n components is an $(n+3)$-tuple $G = (N, T, P_1, P_2, \ldots, P_n, S)$, where
— N, T and S are specified as in a context-free grammar, and
— $P_1, P_2, \ldots, P_n$ are finite sets of context-free rules.

ii) We say that $y \in (N \cup T)^*$ is derived from $x \in (N \cup T)^*$ by P_i, $1 \leq i \leq n$, (written as $x \Longrightarrow_{P_i}^{t} y$) if $x \Longrightarrow_{P_i}^{*} y$ (i. e., y can be obtained from x by a derivation which uses only rules of P_i) and no rule of P_i can be applied to y.

iii) The language $L(G)$ generated by G consists of all words $z \in T^*$ such that there is a derivation of the form

$$S \Longrightarrow_{P_{i_1}}^{t} w_1 \Longrightarrow_{P_{i_2}}^{t} w_2 \Longrightarrow_{P_{i_3}}^{t} \ldots \Longrightarrow_{P_{i_s}}^{t} w_s = z$$

with some $t \geq 1$, $1 \leq i_j \leq n$, $1 \leq j \leq s$.

Grammar Systems – Results

$\mathcal{L}_n(CF)$ denotes the set of all languages which can be generated by grammar systems with (at most) n components.

Theorem:

i) $\mathcal{L}(CF) = \mathcal{L}_1(CF) = \mathcal{L}_2(CF)$.

ii) For any $n \geq 3$, $\mathcal{L}_n(CF) = \mathcal{L}_3(CF)$.

Length Sets of Languages

For a language L and a family X of grammars, we define

$$N(L) = \{n \mid n = |w| \text{ for some } w \in L\},$$
$$N(X) = \{N(L) \mid L \in \mathcal{L}(X)\}.$$

Theorem : i) $N(REG) = N(CF) \subset N(CS) \subset N(RE)$.
ii) A set $M \subseteq \mathbf{N}_0$ belongs to $N(CF)$ if and only if there are numbers $r, s, q_1, q_2, \ldots q_r, p_1, p_2, \ldots, p_s \in \mathbf{N}_0$ such that

$$p \geq 1, \quad q_1 < q_2 < \ldots < q_r < p_1 < p_2 < \ldots < p_s,$$
$$M = \{q_1, q_2, \ldots, q_r\} \cup \bigcup_{i=1}^{s} \{p_i + np \mid n \in \mathbf{N}_0\}.$$

Membrane Systems – Definition I

Definition:
i) A membrane system with m membranes is a $(2m+3)$-tuple

$$\Gamma = (V, \mu, w_1, w_2, \ldots w_m, R_1, R_2, \ldots R_m, i),$$

where

— V is a finite alphabet (of objects occurring in the membranes),

— μ is a membrane structure (of m membranes),

— for $1 \leq j \leq m$, w_j is a word over V (giving the initial content of membrane j),

— for $1 \leq j \leq m$, R_j is a finite set of rules which can be applied to words in membrane j,

— i is a natural number such that $1 \leq i \leq m$ and the membrane i is a simple membrane (the output membrane).

Membrane Systems – Definition II

ii) A configuration of Γ is an m-tuple of multisets/words.

iii) For two configurations $C = (u_1, u_2, \ldots, u_m)$ and $C' = (u'_1, u'_2, \ldots, u'_m)$, we say that C is transformed to C' by Γ, written as $C \vdash C'$, if and only if C' is obtained from C by a maximal parallel application of rules of R_i to u_i for all i, $1 \leq i \leq m$.

iv) A configuration $C = (u_1, u_2, \ldots, u_m)$ is called halting iff no rule of R_i is applicable to u_i for $1 \leq i \leq m$.

v) The language $L(\Gamma)$ generated by a membrane system Γ is the set of all numbers n such that there is a halting configuration $C = (u_1, u_2, \ldots, u_m)$ of Γ with $|u_i| = n$.

Special Membrane Systems

A letter $c \in V$ is called a *catalyst* iff all rules where c occurs have the form $ca \rightarrow cw$ with $a \in V$ and $w \in (V \times Tar)^*$.

We say that a rule $u \rightarrow w$ with $w \in (V \times Tar)^*$ is
– *non-cooperating* iff $u \in V$,
– *cooperating* iff $|u| \geq 2$,
– *catalytic* iff $u = ca$ and $w = cw'$ for some catalyst c, some $a \in V$ and some $w' \in (V \times Tar)^*$.

We say that a membrane system is
– *non-cooperating* if all its rules are non-cooperating,
– *catalytic* if all its rules are non-cooperating or catalytic, and
– *cooperating* if it contains at least one rule which is cooperating and not catalytic.

Membrane Systems – Results

By $\mathcal{L}_n(P, nco)$, $\mathcal{L}_n(P, cat)$, and $\mathcal{L}_n(P, coo)$, we denote the families of languages which can be generated by non-cooperating, catalytic, and cooperating membrane systems with at most n membranes, respectively.

For $X \in \{nco, cat, coo\}$, $\mathcal{L}_*(P, X) = \bigcup_{n \geq 1} \mathcal{L}_n(P, X)$.

Fact: For $X \in \{nco, cat, coo\}$,
$\mathcal{L}_1(P, X) \subseteq \mathcal{L}_2(P, X) \subseteq \mathcal{L}_3(P, X) \subseteq \ldots \subseteq \mathcal{L}_n(P, X) \subseteq \ldots \subseteq \mathcal{L}_*(P, X)$.

Lemma: For $X \in \{nco, cat, coo\}$ and $n \geq 2$,
$\mathcal{L}_1(P, X) \subseteq \mathcal{L}_2(P, X) = \mathcal{L}_n(P, X) = \mathcal{L}_*(P, X)$.

Theorem: For $n \geq 1$, $\mathcal{L}_1(P, nco) = \mathcal{L}_n(P, nco) = \mathcal{L}_*(P, nco) = N(CF)$.

Theorem: For $n \geq 1$, $\mathcal{L}_1(P, coo) = \mathcal{L}_n(P, coo) = \mathcal{L}_*(P, coo) = N(RE)$.

Theorem: For $n \geq 2$,
$\mathcal{L}_1(P, cat) \subset \mathcal{L}_2(P, cat) = \mathcal{L}_n(P, cat) = \mathcal{L}_*(P, cat) = N(RE)$.

Membrane Systems with Symport/Antiport Rules I

Definition:
i) A membrane system with m membranes and symport/antiport rules is a construct

$$\Gamma = (V, \mu, E, w_1, w_2, \ldots, w_m, R_1, R_2, \ldots R_m, i)$$

where V, μ, $w_1, w_2, \ldots w_m$, $R_1, R_2, \ldots, R_m$ and i are specified as in membrane system, E is a subset of V and, for $1 \leq j \leq m$, R_j is a finite set of rules of the form (x, in) or (x, out) or $(x, out; y, in)$ with $x, y \in V^+$.

ii) A configuration of a membrane system with symport/antiport rules is a m-tuple $C = (u_1, u_2, \ldots, u_m)$ of words (or equivalently, multisets) over V.

Membrane Systems with Symport/Antiport Rules II

iii) Let j, $1 \leq j \leq m$, be a membrane and let j' be the unique membrane which contains membrane j. The application of a rule (x, in) of R_j to C results in taking the multiset x out $c_{j'}$ and adding to c_j; the application of (x, out) is performed by subtracting x from c_j and adding to $c_{j'}$; the application of $(x, out; y, in)$ consists in a parallel application of (x, out) and $y, in)$ as described. If j is the outer membrane, then E takes the rule of membrane j' where any element of E is present in E infinitely often.

The transformation of a configuration C into a configuration C' (written as $C \vdash C'$) is done by a maximal parallel application of the rules of all R_j, $1 \leq j \leq m$, to C.

iv) A configuration C is called halting if no rules from the sets R_j, $1 \leq j \leq m$, can be applied to C

Membrane Systems with Symport/Antiport Rules III

v) The language $L(\Gamma)$ generated by a membrane system Γ with symport/antiport rules is the set of all numbers n such that there is a halting configuration $C = (u_1, u_2, \ldots, u_m)$ of Γ with $|u_i| = n$.

Theorem:
For any set $L \in N(RE)$, there is a membrane system Γ with symport/antiport rules and two membranes such that $L(\Gamma) = L$.