CHOMSKY-SCHÜTZENBERGER REPRESENTATIONS FOR FAMILIES OF LANGUAGES AND GRAMMATICAL TYPES

Sorin Istrail
University "Al.I.Cuza"
Department of Mathematics and Computer Center
6600 - IASI, ROMANIA

ABSTRACT

The paper has two parts. In Part I, we shall present Chomsky-Schützenberger theorems for the families of context-sensitive (CS)and recursive-enumerable (RE) languages.

The results are obtained by generalizing the construction of the Dyck set from a "content-free" one to a "content-sensitive" one.

Also presented are fixed-point characterization theorems for CS and RE, which generalize the Algol-like theorem. While $\{X_i=F_i(X_1,\dots,X_t)$, $1\leq i\leq t$, is the system used in the Algol-like theorem, our theorems use $\{X_i=h_i(R_i\cap F_i(X_1,\dots,X_t))$, $1\leq i\leq t$, with F_i as above, h_i a finite substitution and R_i a regular set. The pair $\pi_i=(h_i,R_i)$ is called a π-function, defined as $\pi_i(L)=h_i(R_i\cap L)$.

Part II contains the study of systems of equations with right sides polynomials in π-functions, which turn out to be regular expressions over $\{\cdot,\cup,\cap,*$, finite substitution$\}$.

This is of interest not only because they realize the CS- and RE-steps, but also because they seem to provide with a "language" in which a variety of generative mechanisms from the literature can be expressed. This gives the base to an abstract, equational-based theory for presenting generative mechanisms: Grammatical types.

Within the theory we present general techniques for deriving Chomsky-Schützenberger representations for families of languages possessing a grammatical type definition. Among such families of languages we mention: CS, RE, programmed, Turing machines, Petri-nets, regular-control, scattered-context, L-systems, N(D)TIME(f), N(D)SPACE(f) (for $f(n)=n^k$ or $f(n)=k^n$), NP, P, EXPTIME.

PART I

I.1. FIXED-POINT THEOREMS

We shall give a fixed-point theorem for context-sensitive and recursive-enumerable languages, improving our results [I1], [I2].

THEOREM 1

A language $L\subseteq V^+$ $[\subseteq V^*]$ is context-sensitive [recursive-enumerable] if and only if it is a component of the minimal solution of a system of equations of the form:

$$\begin{cases} X_1 = h_1\ (R_1 \cap F_1(X_1,\dots,X_t)) \\ \text{- - - - - - - - - - - - - - - -} \\ X_t = h_t\ (R_t \cap F_t\ (X_1,\dots,X_t)) \end{cases}$$

where:

i) $h_1,\dots,h_t$ are λ-free [arbitrary] finite substitutions;

ii) $R_1,\dots,R_t$ are regular expressions with $\{\cdot, \cup, +\}$ not containing variables;

iii) $F_1,\dots,F_t$ are λ-free [arbitrary] polynomials in variables $X_1, \dots, X_t$ and having coefficients in V^*.

I.2. CHOMSKY-SCHÜTZENBERGER REPRESENTATIONS

Content-sensitive parentheses

We shall define a concept of "parentheses" generalizing the classical ones. The generalization is inspired by the reducibility of the known parentheses. While the classical reducibility is "content-free", i.e. $(\ \ ) \vdash \lambda$, our generalized reducibility is "content-sensitive", i.e. $[w] \vdash w'$ (reducing parentheses, but the result of the reducibility depends on the content).

DEFINITION 1

Let Par $= \{[_n,]_n \ / \ n \geq 1\}$ be an infinite set of pairs of parentheses symbols and V a finite alphabet disjoint with Par.

Let $V_1 \subset V \cup$ Par be a finite set.

A π-function over V_1 is a pair $\pi = (h,R)$ where:

i) h is a finite substitution on V_1, and

ii) R is a regular set over V_1.

We say that π is λ-free, if h is λ-free.

For $E \subset V_1^*$ we define $\pi(E) = h(E \cap R)$.

Let us remark that the system given by theorem 1 has now the form $\{X_i = \pi_i(F_i(X_1,\dots,X_t)),\quad 1 \leq i \leq t$, where π_i denotes the π-function $\pi_i = (h_i,R_i),\ \ 1 \leq i \leq t$.

DEFINITION 2

A content-sensitive parentheses over V_1 (cs-parentheses, for short) is given by $\rho = (\pi\ ;\ [_\pi,]_\pi)$ where:

a) π is a π-function

b) there exists n such that $[_\pi = [_n$ and $]_\pi =]_n$.

We shall denote by V_π and V'_π respectively V_1 and $V_1 \cup \{[_\pi,]_\pi\}$. The cs-parenthesis ρ is called λ-free if π is so.

The reducibility relation $\vdash_\rho$ associated to ρ, is given by $u \vdash_\rho v$ iff $u = u_1 [_\pi w]_\pi u_2$, $v = u_1 w' u_2$ and $w \in \pi(w')$.

If $V = \{y_1,\dots,y_q\}$ and the finite substitution h is defined by the rules $y_i \to z_{ij}$, $1 \leq i \leq q$, $1 \leq j \leq n_i$, let us consider $m=(m_{11},\dots,m_{1n_1},\dots,m_{qn_q})$ be the vector of distribution (i.e. number of usages) of rules of h in w' to obtain w. (rule $y_i \to z_{ij}$ is used m_{ij} times).
We say that w is obtained from w' by an m-factorization via π, and we write $u \vdash^{(m)}_{\rho} v$.
Given a finite set $\bar{\rho}$ of cs-parentheses, the reducibility generated by $\bar{\rho}$, denoted $\vdash^{*}_{\bar{\rho}}$ is the reflexive-transitive closure of $\bigcup_{\rho\in\bar{\rho}} \vdash_{\rho}$.

DEFINITION 3

Given a set of cs-parentheses $\bar{\rho} = \{\rho_1,\dots,\rho_n\}$, $\rho_i = (\pi_i; \sqsubset_{\pi_i}, \sqsupset_{\pi_i})$, the Dyck-set generated by $\bar{\rho}$, is the class of the empty word λ of $(\bigcup_{i=1}^{n} V'_{\pi_i})^*$. It is denoted $D_{\bar{\rho}}$.

EXAMPLES 1.

i) The restricted Dyck set $D_n'^*[B]$ equals $D_{\bar{\rho}}$, where $\bar{\rho} = \{\rho_1,\dots,\rho_n\}$, $V_i = \emptyset$, $\rho_i = (\pi; x_i, \bar{x}_i)$, $\pi = (1_{\{\lambda\}}, \{\lambda\})$, $1 \leq i \leq n$.
The Dyck set D_n^* equals $D_{\bar{\rho}'}$, where $\bar{\rho}' = \bar{\rho} \cup \{\rho'_1,\dots,\rho'_n\}$, $\rho'_i = (\pi; \bar{x}_i, x_i)$.

ii) The Dyck-set D_I (generalization due to Schützenberger [B]) where $I \subset \{1,\dots,n\}$, equals $D_{\bar{\rho}_I}$, where $\bar{\rho}_I = \bar{\rho} \cup \{\rho'_i \mid i \in I\}$.

iii) The set of non-necessarily nested parentheses over $\Sigma = \{x_1,\dots,x_n\}$, equals $D_{\bar{\rho}_0}$ where $\bar{\rho}_0 = \{\rho_{ij} \mid 1 \leq i, j \leq n\}$, $\rho_{ij} = (\pi_{ij}; x_i, \bar{x}_i)$, $\pi_{ij} = (1_{V^*}, \{x_j, \bar{x}_j\})$ and $V = \Sigma \cup \bar{\Sigma}$, $1 \leq i, j \leq n$.
The twin-shuffle T_{Σ} (Engelfriet, Rozenberg [ER]) equals $D_{\bar{\rho}'_0}$, where $\bar{\rho}'_0 = \bar{\rho}_0 \cup \{\rho'_{ij} \mid 1 \leq i,j \leq n\}$, $\rho'_{ij} = (\pi_{ij}, \bar{x}_i, x_i)$, $1 \leq i,j \leq n$. Note that $D_{\bar{\rho}_0}$ can also be called the restricted twin-shuffe.
Using the characterization theorem 1 and the content-sensitive parentheses we shall obtain Chomsky-Schützenberger representation theorems for the families of context-sensitive (filling a gap in the literature) and recursive-enumerable languages.

THEOREM 2

For every context-sensitive [recursive-enumerable] set $L \subseteq V^*$, there exists a regular set R such that

$$L = \varphi(\mathbb{D}_{\alpha} \cap R)$$

where α = CS [α = RE], is a homomorphism not depending on L and $\mathbb{D}_{CS}$ $\mathbb{D}_{RE}$ is the "universal" Dyck set over V for the family of context-sensitive [recursive-enumerable] sets.

PART II: GRAMMATICAL TYPES
(PRELIMINARY REPORT)

The study of π-functions in systems of equations is interesting not

only because they realize the context-sensitive step, but also because they seem to provide us with a "language in which a variety of generative mechanisms from the literature, can be expressed.
This gives the base to an abstract, equationsl-based theory for presenting generative mechanisms: Grammatical types.
The generalization of the notion of Dyck set from a "content-free" one to a "content-sensitive" one is performed by the way of π-functions. Their power of expressing generative actions is the base of obtaining, Chomsky-Schützenberger - like representation theorems for a variety of families of languages, possessing grammars or automata characterizations.

II.1. FIRST-ORDER GRAMMATICAL TYPES

The following families of languages have a first order type defining them: context-free, CS, RE, Petri nets, Programmed, Turing machines, regular-controlled (on Szilard words), ordered, scattered context, L-systems.
We shall define the first-order grammatical type α, by giving its syntax and its semantics.

Syntax:

i) Let Σ_α be a finite set, called the set of sorts, and s_T be a distinguished element of Σ_α called the terminal sort; also we denote $\Sigma'_\alpha = \Sigma_\alpha \setminus \{ s_T\}$.

Given $M'_\alpha \subset (\Sigma'_\alpha)^*$, the set of α-schematic actions (or α-monomials) is $M_\alpha = M'_\alpha \cup \{ s_T\}$. Let $POLY_\alpha$ be a regular subset of $\{s_T\} \cup (M'_\alpha \cdot \{+\})^* M'_\alpha$ called the α-polynomials, i.e. words of the form $p_1 + \ldots + p_n$, with $p_i \in M'_\alpha$, $1 \leqslant i \leqslant n$.

ii) $\Theta = \{X, Y\}$ is the set of variables; x_o is a special symbol called the initial; if $p \in POLY_\alpha$ then $p(X \cup x_o)$ is an α-term.

iii) An α-schematic system is given by: $S_\alpha : \{ X = t_X, Y = t_Y$, where t_X, t_Y are α-terms and $t_Y = s_T(X \cup x_o)$.

Semantics:

We shall consider three alphabets:

a) V a finite set;

(We suppose $V \subset V_{Terminal}$ which is the infinite collection of terminal symbols. However, we will always work with the arbitrary finite alphabet V).

b) $N = \{x_o, x_1, \ldots, x_m, \ldots \}$ an infinite auxiliary set;

c) Par $= \{ [_n,]_n \mid n \geqslant 1\} \cup \{ [_o,]_o, [_{-1},]_{-1}\}$ an infinite set of pairs of parentheses symbols.

A basic notion for defining the semantics of α-schematic systems is that of π-function. The class of π-functions over $V \cup N \cup$ Par, denoted Π, is the collection of all pairs $\pi = (h,R)$, where there exists a finite set $V_\pi \subset V \cup N \cup$ Par such that h is a finite substitution h: $V_\pi^* \longrightarrow 2^{V_\pi^*}$ and R a regular set over V_π. Such a pair $\pi = (h,R)$ defines the function $\pi: 2^{V_\pi^*} \longrightarrow 2^{V_\pi^*}$ given by $\pi(L) = h(L \cap R)$. In what follows, we intend to associate some meanings to the α-terms. Let us consider a sorting function $\zeta_\alpha : \Sigma \longrightarrow 2^{\Pi}$, and denote for every $s \in \Sigma_\alpha$, $\zeta_\alpha(s)$ by Π_s, the class of π-functions of sort s. We denote by Π' the closure of Π under composition "o" and union "$\cup$", and extend ζ_α to $POLY_\alpha$ as follows:

$$\zeta_\alpha(e_1 e_2) = \zeta_\alpha(e_1) \circ \zeta_\alpha(e_2), \text{ for all } e_1, e_2 \in \Sigma'^*;$$
$$\zeta_\alpha(p_1 + p_2) = \zeta_\alpha(p_1) \cup \zeta_\alpha(p_2)$$

for every $p_1, p_2 \in \{s_T\} \cup (M'_\alpha \{+\})^* M'_\alpha$.

Considering an α-term $p(X \cup x_o)$, we shall define one possible meaning for it as follows: if $f' \in \zeta_\alpha(p)$, we define the function f: $2^{V_f^*} \rightarrow 2^{V_f^*}$ by $f(L) = f'(L \cup \{x_o\})$ where V_f is an alphabet obtained by the adjunction of x_o to the alphabet of f'.

Now the set of meanings of the α-term $p(X \cup x_o)$, denoted $\zeta_\alpha(p(X \cup x_o))$ is given by: $\zeta_\alpha(p(X \cup x_o)) = \{f \mid f' \in \zeta_\alpha(p)\}$.

Given an α-schematic system $S_\alpha : \{X = t_X, Y = t_Y$ then an interpretation of S_α is any member of $\zeta_\alpha(t_X) \times \zeta_\alpha(t_Y)$.

If σ is an interpretation, denote it $\sigma = (t_X^\sigma, t_Y^\sigma)$.

An α-system is a pair $G = (S_\alpha, \sigma)$, i.e. $G: \{X = t_X^\sigma, Y = t_Y^\sigma$.

Each α-system possesses an unique minimal solution $G^{MIN} = (X^{MIN}, Y^{MIN})$.

A language L is said α-equational iff $L = Y^{MIN}$ for some α-system G, such that $G^{MIN} = (X^{MIN}, Y^{MIN})$.

The family of α-equational languages is denoted $EQUAT_\alpha$.

Our collection of π-functions must be augmented with two special ones.

Let be $\pi_{-1} = (1_{\{\lambda\}}, \{\lambda\})$ and $\pi_o = (h_o, \{[_{-1} \ ]_{-1})$, where h_o is given by $h_o([_{-1}) = \{[_{-1} x_o\}$ and $h_o(]_{-1}) = \{]_{-1}\}$.

Now be $\Pi(\alpha) = (\bigcup_{s \in \Sigma} \Pi_s) \cup \{\pi_o, \pi_{-1}\}$.

EXAMPLE 2. The type provided by CS_λ-grammars

The Corrolary 3 of [I1] gives a fixed-point characterization to the λ-free context-sensitive languages (CS_λ). An equivalent form of the system presented in Theorem 1, (equivalence being the coincidence of their first component of the minimal solution) is the system:

$$(1) \qquad \begin{cases} X_1 = V^* \cap (X_2 \cup \ldots \cup X_t \cup x_o \\ X_i = h_i(R_i \cap (X_2 \cup \ldots \cup X_t \cup x_o)), \quad 2 \leq i \leq t \end{cases}$$

where: i) V is a "terminal" alphabet; ii) all π-functions $\pi_i = (h_i, R_i)$ are over the alphabet $V_\pi = V \cup V_N$, with V_N a finite "nonterminal" alphabet; iii) $x_0 \in V_N$.

In the same sense as above, the system (1) is equivalent to:

$$(2)\qquad \begin{cases} Y = V^* \cap (X \cup x_0) \\ X = \bigcup_{i=2}^{t} h_i(R_i \cap (X \cup x_0)) \end{cases}$$

The equations of (1) have the following properties:

(a) The equation for X_1 simulates the "selection of terminal words", i.e. words containing only terminal symbols;

(b) There exists one equation, say that of X_2, which simulates the task "chooses nondeterministically one nonterminal";

(c) All the remaining equations, i.e. for X_i, $3 \leq i \leq t$, simulates different "applications of the context-sensitive rules" of the CS-grammar, i.e. rules of the form $pxq \longrightarrow puq$.

Again, more natural, an equivalent form of (2) is:

$$(3)\qquad \begin{cases} Y = \pi_1(X \cup x_0) \\ X = (\sum_{i=3}^{t} \pi_i \pi_2)(X \cup x_0) \end{cases}$$

According to the "sorts" of equations of (1), in order to define the type α, we consider three sorts s_1, s_2, s_3. I.e. $\Sigma_{CS_\lambda} = \{s_1, s_2, s_3\}$. The sorts will abstract the structure of the following sets of π-functions $\{\pi_1\}$, $\{\pi_2\}$ and respectively $\{\pi_3, \ldots, \pi_t\}$.

A) $\prod_{s_1}$, the set of π-functions of sort s_1, over $V \cup N$ contains exactly one π-function $\pi = (1_{V^*}, V^*)$;

B) $\prod_{s_2}$ is the collection of all pairs $\pi = (h, R)$ satisfying the conditions: B.1) There exist two finite subsets of N : V_N, $\overline{V}_N$ in bijection and disjoint such that $V_\pi = V \cup V_N \cup \overline{V}_N$ (we consider $\overline{V}_N = \{\overline{x} \mid x \in V_N\}$) ; B.2) $R = (V \cup V_N)^*$; B.3) h is defined by $h(x) = \{x, \overline{x}\}$, for $x \in V_N$ and $h(a) = a$, for $a \in V$.

C) $\prod_{s_3}$, is the collection of all pairs $\pi = (h, R)$ satisfying the conditions: C.1) the same as B.1; C.2) there exist $x \in V_N$, $p, q \in (V \cup V_N)^*$ such that $R = (V \cup V_N)^* p\overline{x}q (V \cup V_N)^*$; C.3) there exists $u \in (V \cup V_N)^+$ such that h is defined by $h(y) = y$, for $y \in V_\pi \setminus \{x\}$ and $h(\overline{x}) = u$.

The relevance of sorts, is in fact the decomposition in atomic actions of the generative device. The sorts s_1, s_2, s_3 are nothing else but names for "selection of terminal words", "chooses nondeterministically one nonterminal" and "application of the context-sensitive rule".

As we can see, in the format of the system (3) our CS_{λ}-polynomials are $POLY_{CS_{\lambda}} = \{s_1\} \cup \{s_3 s_2\ +\}^* s_3 s_2$. □

DEFINITION 4

A Dyck set of type α, $D_{\bar{\rho}}$ is the class of λ of V_1^* under the reducibility relation $\vdash^{*}_{\bar{\rho}}$, where $\bar{\rho} = \{\rho_1, \ldots, \rho_t\}$, $\rho_i = (\pi_i; [_{\pi_i},]_{\pi_i})$, $\pi_i \in \Pi(\alpha)$, $[_{\pi_i},]_{\pi_i}$ is a pair of parentheses symbols from Par, $1 \leq i \leq t$ and $V_1 = (\cup\{V_{\pi_i} \mid 1 \leq i \leq t\} \cup \{[_{\pi_i},]_{\pi_i} \mid 1 \leq i \leq t\})$.

Let $Dyck_{\alpha}$ be the family of Dyck sets of type α.

PROPOSITION 1

For any $L \in EQUAT_{\alpha}$, there exist a Dyck set $D \in Dyck_{\alpha}$, a regular set R and a homomorphism φ such that: $L = \varphi(D \cap R)$

PROOF. Let us consider an α-system $G: \{X = t_X, Y = t_Y$, where $t_X = p_1 + \ldots$ $\ldots + p_m(X \cup x_o)$ and $p_i = s_{i,1}, \ldots, s_{i,k_i}$, $1 \leq i \leq m$ and $Y^{MIN} = L$.

If $p_i^{\sigma} = s_{i,1}^{\sigma} \ldots s_{i,k_i}^{\sigma}$, let us denote $s_{i,e}^{\sigma}$ by $\pi_{i,e}$. Also s_T is denoted π_T. To each π-function $\pi_{i,e}$ used to define the interpretation of our α-system we shall associate a distinct pair of parentheses symbols from Par, obtaining a specific cs-parenthesis $\rho_{i,e} = (\pi_{i,e}; [_{\pi_{i,e}},]_{\pi_{i,e}})$, $1 \leq i \leq m$, $1 \leq e \leq k_i$. The terminal cs-parenthesis is $\rho_T = (\pi_T; [_{\pi_T},]_{\pi_T})$.

Let us denote $V'_{\pi_{i,e}} = V_{\pi_{i,e}} \cup \{[_{\pi_{i,e}},]_{\pi_{i,e}}\}$, $1 \leq i \leq m$, $1 \leq e \leq k_i$ and $V'_{\pi_T} = V_{\pi_T} \cup \{[_{\pi_T},]_{\pi_T}\}$. We shall define the reducibility relation $\vdash^{*}_{G}$ on the set V_G^*, where $V_G = (\cup\{V'_{\pi_{i,e}} \mid 1 \leq i \leq m, 1 \leq e \leq k_i\} \cup V'_{\pi_T} \cup \{[_o,]_o, [_{-1},]_{-1}\})$. The relation $\vdash^{*}_{G}$ equals $\vdash^{*}_{\bar{\rho}}$, where $\bar{\rho} = \{\rho_{i,e} \mid 1 \leq e \leq k_i, 1 \leq i \leq m\} \cup \{\rho_T, \rho_o, \rho_{-1}\}$, $\rho_j = (\pi_j; [_j,]_j)$, $j = o, -1$, $\pi_o = (h_o, \{[_{-1}\ ]_{-1}\})$, $\pi_{-1} = (1_{\{\lambda\}}, \{\lambda\})$ and h_o is defined by $h_o([_{-1}) = [_{-1} x_o$, $h_o(]_{-1}) =]_{-1}$

In order to obtain the result of the Proposition we take $D = D_{\bar{\rho}}$. Note that we add the cs-parentheses ρ_o, ρ_{-1} to realize the reducibility of the initial x_o to the empty word λ.

In order to construct the regular set R, we denote P_{ℓ} and P_r respectively the sets $P_{\ell} = \{[_{\pi_{i,e}} \mid 1 \leq i \leq m, 1 \leq e \leq k_i\}$ and $P_r = \{]_{\pi_{i,e}} \mid 1 \leq i \leq m, 1 \leq e \leq k_i\}$.

We define R as follows: if $R' = [_o [_{-1} [_{\pi_T}\ P_{\ell}^*\ V^* P_r^*\ ]_{\pi_T}]_{-1}]_o$

then $R = \underline{\text{if}}\ \lambda \notin L\ \underline{\text{then}}\ R'\ \underline{\text{else}}\ R \cup \{\lambda\}$.
Now it is manifest that if we define φ by: $\varphi(z) = \lambda$ for $z \in P_\ell \cup P_r \cup \{\sqsubset_0, \sqsubset_{-1}, \sqsupset_0, \sqsupset_{-1}\}$ and $\varphi(a) = a$, for $a \in V$ we have: $L = \varphi(D \cap R)$. $\square$

We shall prove in what follows a Chomsky-Schützenberger representation for $EQUAT_\alpha$, but at this time with a "universal" Dyck set, depending ony on the alphabet of the language, not on the language itself. That is, we exhibit a set $\mathbb{D}_\alpha$, that we call the universal Dyck set of the type α with respect to an alphabet V, such that for any $L \in EQUAT_\alpha$ there exist a regular set R_L and a homomorphism φ such that

$$L = \varphi(\mathbb{D}_\alpha \cap R_L)$$

This universal Dyck set of the type α encodes all possible reducibilities of all cs-parentheses provided by the π-functions of the set $\prod(\alpha)$.

Because the construction is lengthy we cannot include it here. However we shall give the main points.

A function Code is defined, which gives usual encodings for Par $\cup V \cup N$. By the way of Code we derive an encoding for a finite substitution and another for a regular set (a regular expression over $\cup, \cdot, *$). At this point, we can define Code (ρ) (which is a set of words).

Another code function $\hat{}$ occurs in our construction, but at this time giving codes over a disjoint alphabet, with that of Code.

In these terms, we can define the (universal) reducibility relation $\models_\alpha$. If $\rho = (\pi; \sqsubset_\pi, \sqsupset_\pi)$, $\pi \in \prod(\alpha)$ we put: $u \models_\alpha v$ iff $u = \hat{u}_1\ \hat{\sqsubset}_\pi$ Code(ρ) $\hat{w}\ \hat{\sqsupset}_\pi \hat{u}_2$, $v = \hat{u}_1\ \hat{w}'\ \hat{u}_2$ and $u_1 \sqsubset_\pi w \sqsupset_\pi u_2 \models_\rho u_1 w' u_2$.

If $\prod_1$ is a set of π-functions, then $cs(\prod_1)$ denotes the collection of all cs-parentheses obtained from π-functions of $\prod_1$ and well balanced pairs of parentheses of Par. In a similar way with the definition of τ_α we define ρ_α, a variant for cs-parentheses. We have $\rho_\alpha : \Sigma_\alpha \longrightarrow 2^{cs(\prod(\alpha))}$ given by $\rho_\alpha(s) = cs(\prod_s)$, for all $s \in \Sigma_\alpha$.

We extend ρ_α to subsets of Σ_α^* by:

$$\rho_\alpha(e_1 e_2) = \rho_\alpha(e_1) \circ \rho_\alpha(e_2)$$
$$\rho_\alpha(e_1 \cup e_2) = \rho_\alpha(e_1) \cup \rho_\alpha(e_2)$$

for every $e_1, e_2 \subset \Sigma_\alpha^*$.

The definitions above for composition and union of cs-parentheses are given in the usual way:

$$\models_{\rho_1 \rho_2} = \models_{\rho_1} \circ \models_{\rho_2} \quad \text{and} \quad \models_{\rho_1 \rho_2} = \models_{\rho_1} \cup \models_{\rho_2} .$$

Finally, we define $u \models_\alpha v$ iff $u \models_\eta v$ for some η.

$$\eta \in \mathcal{P}_\alpha(\mathcal{T}_\alpha(M_\alpha)) \cup \{\rho_0, \rho_1\}.$$

DEFINITION 5

The Universal Dyck set of type α over V, is the class of the empty word λ of $(V \cup K)^*$ with respect to the α-reducibility relation $\models^*_\alpha$. It is denoted $\mathbb{D}_\alpha$.

Extending the function Code to sets of compositions of cs-parantheses by Code $(E_1 \circ E_2)$ = Code (E_1) Code (E_2), Code $(E_1 \cup E_2)$ = Code (E_1) $\cup$ Code (E_2), we are ready to define a central notion in our theory.

DEFINITION 6

The Kernel of the grammatical type α over V, is the language Kerα = Code $(\mathcal{P}_\alpha(\mathcal{T}_\alpha(M_\alpha)))$.

PROPOSITION 2

If Kerα is a context-sensitive set, then the language $\mathbb{D}_\alpha$ is context-sensitive too.

THEOREM 3. (The Chomsky-Schützenberger representation)

Let α be a grammatical type.

For any language $L \in EQUAT_\alpha$, there exist a regular set R_L such that

$$L = \varphi(\mathbb{D}_\alpha \cap R_L),$$

where φ is a homomorphism depending only on α.

PROOF

Let us consider $L \in EQUAT_\alpha$ and the α-system $G=(S_\alpha, \sigma)$ defining it, i.e. $G : \{ X = t^\sigma_X,\ Y = t^\sigma_Y$ and $Y^{MIN} = L$.

We suppose that $t_X = (p^\sigma_1 + \ldots + p^\sigma_n)(X \cup x_o)$, $t_Y = \pi_T(X \quad x_o)$ and $p^\sigma_i = \pi_{i,1} \ldots \pi_{i,k_i}$. We shall use a collection of pairs of parentheses symbols from Par, say $[_{\pi_{i,j}}$, $]_{\pi_{i,j}}$, $1 \leq i \leq n$, $1 \leq j \leq k_i$ and $[_{\pi_T}$, $]_{\pi_T}$.

Let be the regular set Left $= \cup \{ \hat{[}_{\pi_{i,1}}$ Code $(\rho_{i,1})\ \hat{[}_{\pi_{i,2}}$ Code$(\rho_{i,2}) \ldots \hat{[}_{\pi_{i,k_i}}$ Code (ρ_{i,k_i}) / $1 \leq i \leq n\}$.

We shall define the regular R_L as follows:

$R_L = \hat{[}_o$ Code $(\rho_o)\ \hat{[}_{-1}$ Code$(\rho_{-1})\ \hat{[}_{\pi_T}$ Code$(\rho_T)(V \cup$ Left $\cup \{\hat{]}_{\pi_{i,j}} /$ $1 \leq i \leq n,\ 1 \leq j \leq k_i\})^*\ \hat{]}_{\pi_T}\ \hat{]}_{\pi_{-1}}\ \hat{]}_o$. The homomorphism φ is given by $\varphi(a)=a$, for $a \in V$, and $\varphi(z) = \lambda$, for $z \in k$.

Now we have indeed

$$L = \varphi(\mathbb{D}_\alpha \cap R_L).$$

$\square$

II.2. SECOND-ORDER GRAMMATICAL TYPES

We shall consider a more general concept as the one introduced above called second-order grammatical type.
Examples of families of languages which have second-order grammatical types defining them are:

$$N(D)TIME(f),\quad N(D)SPACE(f),\ N(D)RETURN(f)$$
$$(\text{for } f(n) = k^n \text{ or } f(n) = n^k;$$

AHO's indexed languages, EXPTIME, some generalizations of PETRI nets languages, NP, P.

The second-order grammatical type β

A second order grammatical type β is constructed from two (first-order) grammatical types α and α'.

Syntax:

(i) $\Sigma_\beta = \Sigma_\alpha \cup \Sigma'_\alpha \cup \{s''_T\}$ is the set of sorts; s''_T is called the second-order terminal sort.

The set of β-monomials (or β-schematic actions) is $M_\beta = M_\alpha \cup M_{\alpha'} \cup \{s''_T\}$. The β-polynomials are given by

$$POLY_\beta = POLY_\alpha \cup POLY_{\alpha'} \cup \{s''_T\},$$

(ii) $\Theta = \{X, Y, X', Y', Z\}$ is the set of variables.

The β-terms are the α-terms, the α'-terms and a specific second-order term $s''_T\ (Y,Y')$.

(iii) A β-schematic system is given by:

$$S_\beta : \{X=t_X,\ Y=t_Y,\ X'=t_{X'},\ Y'=t_{Y'},\ Z = s''_T\ (Y,Y')$$

where $S_\alpha : \{X = t_X,\ Y=t_Y$ and $S_{\alpha'} : \{X'=t_{X'},\ Y' = t_{Y'}\}$ are respecively α- and α'-schematic systems.

Semantics

To define semantics we need the notion of second order π-function. It is a two-variable function defined by the way of a finite substitution h applied to the intersection of arguments:

$$\pi : 2^{V_\pi^*} \times 2^{V_\pi^*} \longrightarrow 2^{V_\pi^*} \quad \text{given by}$$
$$\pi(L_1,L_2) = h(L_1 \cap L_2)$$

Given a β-schematic system S_β, an interpretation of it is a 5-tuple $\bar\sigma = (t_X^\sigma,\ t_Y^\sigma,\ t_{X'}^{\sigma'},\ t_{Y'}^{\sigma'},\ t^{\sigma''})$ where: $\sigma=(t_X^\sigma, t_Y^\sigma)$, $\sigma' = (t_{X'}^{\sigma'}, t_{Y'}^{\sigma'})$

are interpretations of S_α respectively S_α and $t^{\sigma''} = \pi(Y,Y')$ with π a second-order π-function.

Now, a β-system is a pair $G'' = (S_\beta, \bar{\sigma})$.

A language is said β-equational if it equals Z^{MIN} of a β-system. Notions as: $\vdash_{\overline{\beta}}$, $\mathbb{D}_\beta$, Ker β can be introduced following the corresponding generalizations.

THEOREM 4

For any $L \in EQUAT_\beta$ there exists a regular set R such that $L = \varphi(\mathbb{D}_\beta \cap R)$, where φ is a homomorphism depending only on β.

II.3. AN USEFUL RESULT

We derive as a corollary of the above results an useful theorem giving sufficient conditions for a family of languages $\mathcal{L}$ to possess a Chomsky-Schützenberger representation with $\mathbb{D}_\mathcal{L} \in \mathcal{L}$.

DEFINITION 7

A Chomsky-Schützenberger representation for a family of sets $\mathcal{L}$ is called standard if the universal Dyck set used to represent the sets in $\mathcal{L}$ belongs to $\mathcal{L}$.

A family of languages $\mathcal{L}$ is called grammatical type accessible if there exists a (first-order or second-order) grammatical type $\gamma \in \{\alpha, \beta\}$ such that $EQUAT_\gamma = \mathcal{L}$.

THEOREM 5

Any grammatical type accessible family of languages $\mathcal{L}$ (i.e. $\mathcal{L} = EQUAT_\gamma$) satisfying:

i) $Ker\gamma \in \underline{CS}$

ii) $\underline{CS} \subseteq \mathcal{L} \subseteq \underline{RE}$

possesses a standard Chomsky-Schützenberger representation.

APPENDIX 1

THE FIRST-ORDER GRAMMATICAL TYPE OF TURING MACHINES (TM)

We consider a Turing machine M as a rewriting system [S1] with Q the set of states, V_T the tape alphabet, $Q_1 \subseteq Q$ the final states set and F a set of rules. We shall use two new symbols x_0, y and the "barred" copies of Q and V_T, namely $\bar{Q}$, $\bar{V}_T$. $\# \in V_T$ is the boundary marker.

We define π-functions to simulate the rules of the TM. For notational simplicity a finite substitution h is specified by a set of context-free rules only for the nonidentical replacements. For example, $\{a \to a, a \to \bar{a}, s \to s, s \to \bar{s}\}$ is the substitution h_0, given by $h_0(a) = \{a, \bar{a}\}$, $h(s) = \{s, \bar{s}\}$, $h_0(\bar{s}) = \{\bar{s}\}$, $h_0(\bar{a}) = \{\bar{a}\}$, for $s \in Q$ and $a \in V_T$. Note that, $\{\ldots\}(X)$ means $h_0(X)$.

(1) Overprint: $s_i\ a \to s_j\ b$ is simulated by the equation

$$X = \{\bar{s}_i \to s_j,\ \bar{a} \to b\}(h_o(X) \cap V_T^* \bar{s}_i \bar{a} V_T^*)$$

(2) <u>Move-right</u>: $s_i ac \to as_j c$ is simulated by the equation

$$X = \{\bar{s}_i \to a, \bar{a} \to s_j,\ \bar{c} \to c\}(h_o(X) \cap V_T^* \bar{s}_i \bar{a}\bar{c} V_T^*)$$

(3) <u>Move-right and extends work-space</u>: $s_i\ a\# \to as_j r\#$ is simulated by

$$X = \{\bar{s}_i \to as_j, \bar{a} \to r, \bar{\#} \to \#\}(h_o(X) \cap V_T^* \bar{s}_i \bar{a}\ \bar{\#}).$$

Similar equations are constructed for (4) <u>Move-left</u> and (5) <u>Move-left and extends work-space</u> rules.

Let us remark that the above equations have the form: $X = \pi_i(\pi_o(X))$, $1 \leq i \leq 5$.

Note that we need to preserve the format of the input word. This is done by the usage of two placed symbols.

As a consequence, we have that the system

$$\begin{cases} X = (\pi_{init} + \sum_{\pi} \pi\pi_o)(X \cup x_o) \\ Y = \pi_{fin}(X \cup x_o) \end{cases}$$

has the property that $Y^{MIN} = L(M)$. (Here π_{init} generates an arbitrary word, to be tested for acceptance, and π_{fin} verifies the accurence of final state and transforms a two placed symbol in its "protected" initial content). It is clear that only 4 sorts we need, in order to define our TM-schematic systems.

We take $POLY_{TM} = \{s_{fin}\} \cup \{s_{init}+\}\{ss_o+\}^* ss_o$.

<u>APPENDIX 2</u>

<u>THE SECOND-ORDER TYPES PROVIDED BY N(D)TIME(f(n)), $f(n)=n^k$ or $f(n)=k^n$</u>

Let us consider the alphabet $W = \{(\frac{c}{d}) \mid c, d \in V \cup \{b\!\!\!/\}\}$ where $b\!\!\!/$ stands for the blank symbol. $V_T \subset V$ is the "terminal" alphabet. For $u, v \in (V \cup \{b\!\!\!/\})^*$, $u = c_1 \ldots c_p$, $v = d_1 \ldots d_p$, $d_i, c_i \in V \cup \{b\!\!\!/\}$, we write $\frac{u}{v}$ for $(\frac{c_1}{d_1}) \ldots (\frac{c_p}{d_p})$.

(1). The construction of the first-order type α'.

Let us assume $a \notin V$ and define the set

$$L_{q,f} = \{a^{q \cdot f(n)}\ \#_L B_1 \frac{w}{u} B_2 \#_R \mid w \in V_T^*\quad n = |w|,$$
$$B_1, B_2 \in \{(\frac{b\!\!\!/}{d}) \mid d \in V \cup \{b\!\!\!/\}\}^*,\ u \in (V \cup \{b\!\!\!/\})^*,\ |u| = |w|\}.$$

For $q=1$, we consider a grammar $G_{1,f}$ generating the set $L_{1,f}$ and we suppose that the letter a appears only in the terminal rule of $G_{1,f}$

of the form:

$(*) \qquad\qquad\qquad\qquad x_a \qquad a$

Let us observe that from $G_{1,f}$ we obtain for a given k the grammar $G_{k,f}$ just by taking instead of $(*)$ the rule $x_a \longrightarrow a^k$. We obtain in this way the family of grammars

$$\mathcal{G}_f = \{G_{k,f} \mid k \geqslant 1\}.$$

We shall construct a type α' provided by $\mathcal{G}_f$ as follows.
We consider the RE-system associated (see Example 2) to $G_{1,f}$:

$(S_{RE}, \sigma): \{X = t_X^\sigma,\ Y = t_Y^\sigma$, where $t_X^\sigma = (\sum_{i=3}^{t} \pi_i \pi_2)(X \cup x_o)$, $t_Y^\sigma = \pi_1(X \cup x_o)$.

Let us suppose that π_3 is the π-function "implementing" the rule $\bar{x}_a \longrightarrow a$, $\pi_3 = (h_3, R_3)$, where

$$R_3 = (W \cup \{\#_L, \#_R\} \cup W_N)^* \ \bar{x}_a (W \cup \{\#_L, \#_R\} \cup W_N)^*$$

and h_3 is given by: $h_3(\bar{x}_a) = a$, $h_3(y) = y$ otherwise. Note that W_N is for the nonterminal alphabet and $x_a \in W_N$.

To define our type α', we shall use t sorts, namely $\sum_{\alpha'} = \{s_1, s_2, \ldots$ $\ldots, s_t\}$, with $\zeta_{\alpha'}(s_i) = \{\pi_i\}$, $i \neq 3$ and $\zeta_{\alpha'}(s_3)$ is defined as the collection of all pairs $\pi = (h, R_3)$ where there exist $k \in \mathbb{N} \smallsetminus \{o\}$ such that h is given by: $h(\bar{x}_a) = a^k$, $h(y) = y$ otherwise.

Clearly taking $POLY_{\alpha'} = \{s_1\} \cup \{s_3 s_2 + s_4 s_2 + \ldots + s_t s_2\}$, we have

$$EQUAT_{\alpha'} = \{L(G) \mid G \in \mathcal{G}_f\}.$$

(2). The construction of the first-order type α .
It is a modified version of the type provided by Turing machines (see Appendix 1) which can be called "time-couting Turing machine type". The difference is that at each time one rewriting rule is applied a "countor" at the left end of the analysed word is increased by one.
Now for a β-system, $Z^{MIN} = h(Y^{MIN} \cap Y'^{MIN})$ and $EQUAT_\beta = NTIME(f)$.

<u>REMARKS</u>

1) Actually our construction shows that

$$EQUAT_\beta = \text{Cut-off } NTIME(f)$$

but in our cases, f being countable [HU] we have equality with NTIME(f).
2) If we restrict to deterministic TM we obtain exactly $EQUAT_\beta = DTIME(f)$.
3) It is easy to see that in all cases $Ker\beta \in \underline{CS}$.

APPENDIX 3 (Sketch)

Some other first order types: CF, matrix, regular-control, scattered-context, paralell CF, L-systems (Notation from [S1])

1) CF. In Example 2, in C2, we take $p=q=\lambda$.

2) Matrix CF. If $[x_1 \rightarrow w_1, \ldots, x_t \rightarrow w_t]$ is a matrix rule, it can be simulated by the equation $X = \pi_1 \pi_0 \pi_2 \pi_0 \ldots \pi_t \pi_0(X)$, π_0 is as an Example 2 i.e. "nondeterministically chooses one nonterminal", and π_i is the π-function implementing the CF rule $x_i \rightarrow w_i$.

3) Regular-control CF. A special symbol rule and Lab(F) the set of labels for the grammar rules. Let $C \subset (\mathrm{Lab}(F))^*$ be a regular set. If $r: x_r \rightarrow u_r$ then π_r, which simulates the rule r is given by $\pi_r = (h_r, R_r)$ where $h_r(\bar{x}_r) = \{u_r\}$, $h_r(\underline{rule}) = r\ \underline{rule}$ and $h_r(x) = x$ otherwise; $R_r = (\mathrm{Lab}(F))^*$. $\underline{rule}\ V^* \bar{x}_r V^*$, $V = V_N \cup V_T$.

Two more π-functions complete the construction of the type: π_T and π_{init}. We have $\pi_T = (h_T, C\ \underline{rule}\ V_T^*)$, $\pi_{init} = (h_{init}, \{x_0\})$, where $h_T(y) = \lambda$ for $y \in \mathrm{Lab}(F) \cup \{\underline{rule}\}$ and identity otherwise, $h_{init}(x_0) = \underline{rule}\ x_0$.

4) Paralell CF. The π-function which simulates the rule $x \rightarrow u$, is $\pi = (h, R)$, where $R = (((V_N \setminus \{x\}) \cup V_T)^* \bar{x})^* ((V_N \setminus \{x\}) \cup V_T)^*$ and h is given by $h(x) = u$, $h(y) = y$ otherwise.

5) L systems. For EOL systems we take, as usual, the following system of equations $\{X = h + h_{init}(X \cup x_0),\ Y = (X \cup x_0) \cap V_T^*$.
For ETOL systems we change the first equation as $X = (h_1 + \ldots + h_t + h_{init})(X \cup x_0)$. In the case of EIL systems, in the system of equations occurs a π-function of a new sort called "syncronization".

A table giving the sets POLY for each case ends our remarks:

TYPE	POLY	TYPE	POLY
CF	$\{s_T\} \cup (ss_0+)^* ss_0$	Reg-contr.	$\{s_T\} \cup (ss_0+)^* ss_0$
Matrix	$\{s_T\} \cup ((ss_0)^*+)^* (ss_0)$	Paralell	$\{s_T\} \cup (ss_0+)^* ss_0$
EOL	$\{s+s_{init}\} \cup \{s_T\}$		
EIL	$s_{init}+(s+)^* s \cup \{s_T\}$		
EIL	$s_{init}+s_{syn}+(ss_0+)^* ss_0 \cup \{s_T\}$		

REFERENCES

B BERSTEL, J. "Transductions and context-free languages" Teubner(1979)

CS CHOMSKY, N., SCHUTZENB RGER M.P. "The algebraic theory of context-free languages", in P.Braffort and D.Hirschberg (eds) Computer Programming and Formal Systems, pp.118-161, North Holland, (1963).

CN COCHET, Y., NIVAT,M. "Une generalisation des ensembles de Dyc Israel J.of Math. 9 (1971)

CM CULIK, K., MAURER, H.A. "On simple representations of language families" RAIRO (to appear)

ER ENGELFRIET, J., ROZENBERG, G. "Fixed-point languages, equality languages and reprezentations of recursive enumerable languages" 19th FOCS,(1978)

G GINSBURG, S. "Algebraic and automata-theoretic properties of formal languages" North Holland, Amsterdam (1975)

H HARRISON, M.A. "Introduction to formal language theory" Addison-Wesley, Reading (1978)

HU HOPCROFT,J., ULLMAN, J.D. "Introduction to Automata theory, Languages and Computation "Addison-Wesley, (1979).

I1 ISTRAIL, S. "A fixed-point theorem for recursive-enumerable languages and some considerations about fixed-point semantics of monadic programs" 6th Colloq.Automata, Languages and Programming, (1979) Lect.Notes Comput. Sci 71

I2 ISTRAIL, S. "Generalizations of Schützenberger-Ginsburg-Rice fixed-point theorem for context-sensitive and recursive-enumerable languages" Theor.Comput.Sci. (to appear)

I3 ISTRAIL, S. "Elementary bounded languages" Information and Contr. vol. 39, 2(1978), p. 177-191

I4 ISTRAIL, S. "Grammatical types of grammar- and L-forms" (in preparation).

IS ISTRAIL, S., SOIL, A. "Fixed-point theorems for Petri nets language" (submitted for publication)

M MONIEN, B. "A recursive and a grammatical characterization of the exponential-time languages" Theor.Comp.Sci. 3 (1977)

N NIVAT, M. "Transductions des languages de Chomschy" Anall.Inst. Fourier (1968)

P PETERSON, J.L. "Petri nets", Computing Surveys, vol.9,no.3,(1977)

R ROZENBERG, G. "Selective substitution grammars (towards a framework for rewriting systems) Part I: Definitions and examples" EIK (1977)

S SALOMAA, A. "Formal languages" Academic Press (1973)