Proceedings of Grammar Systems Week 2004, edited by E. Csuhaj-Varjú and Gy. Vaszil, MTA SZTAKI, Budapest, pages 158 - 169, 2004.

On Cooperating Distributed Grammar Systems with Competence Based Start and Stop Conditions

Jürgen Dassow

Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
PSF 4120, D–39016 Magdeburg
dassow@iws.cs.uni-magdeburg.de

Abstract

We define cooperating distributed grammar systems with start and stop conditions which are based on the competence of a component on the current sentential form. We distinguish six different types of competence conditions which result in 18 types of grammar systems. We summarize the results on the generative power known from the literature (where they are sometimes not related to competence) and determine the power of some further grammar systems.

1 Introduction

Cooperating distributed grammar systems were firstly investigated by R. Meersman and G. Rozenberg in [14]. A systematic study of these systems was started in [6]. Summaries of results are given in [9] and [11].

Intuitively, a cooperating distributed grammar system (CD grammar system for short) consists of some grammars or sets of productions which work on a common sentential form. A certain grammar, for which the start condition holds, starts the derivation, and it has to stop the derivation, if a certain stop condition is satisfied. Then another component satisfying the start condition continues the derivation until the stop condition holds etc.

Mostly one has considered the case where the start condition is true for any grammar and any sentential form and the stop condition is satisfied iff the derived sentential form y is obtained by k (or $\geq k$ or $\leq k$) direct derivation steps with respect to the chosen grammar or the grammar contains no rule which can be applied to y.

The idea behind these conditions was as follows. CD grammar systems have a motivation in the blackboard architecture of Artificial Intelligence. Here the grammars correspond to agents/experts, the nonterminals represent open problems and the application of a rule is a step to the solution. Thus the above conditions can be interpreted such that k (or $\geq k$ or $\leq k$) steps can be contributed to the solution. Therefore the stop condition is only satisfied if the component has a certain competence. However, this does not really reflect competence, because in the k-step derivation we can replace the same nonterminal in any step, i.e., we contribute only to a special subproblem and not to all subproblems. Therefore the condition considered in [14] is more appropriate, where the start condition requires that any nonterminal occurring in the sentential form can be replaced, i.e., the agent can contribute to any subproblem; we call this restriction full competence of the grammar on the sentential form.

In this paper we define the competence as the number of subproblems which can be (partially) solved by the grammar. Formally, the competence of a set P of productions on a sentential form x is the cardinality of the intersection of the set of nonterminals occurring in x and the set of nonterminals which can be rewritten by P. Now one can define competence k (or $\geq k$ or $\leq k$ or $\neq k$) and maximal and full competence. This leads to 18 types of CD grammar systems where the start condition as well as the stop condition is one of the competence types mentioned above or a negation of such a competence type.

We summarize the results on the generative power of such CD grammar systems obtained by M. ter Beek, E. Csuhaj-Varjú, M. Holzer, Gy. Vaszil and the author and add the results on two further systems.

The paper is organized as follows. In Section 2 we recall the definitions of some language families to which the families studied in the paper will be related. In Section 3 we introduce the concept of a cooperating distributed grammar system and the competence conditions. Section 4 contains the results. We finish with some remarks on topics of research to be done in this area.

2 Some Language Families

For an alphabet V, we denote the set of all (non-empty) words over V by V^* (and V^+, respectively). The length of a word $w \in V^*$ is denoted by $|w|$. For a letter $a \in V$ and a word $w \in V^*$, $\#_a(w)$ denotes the number of occurrences of a in w.

A *context-free grammar* is specified as a quadruple $G = (N, T, P, S)$ where
- N and T are disjoint alphabets of nonterminals and terminals, respectively,
- P is a finite subset of $N \times (N \cup T)^*$, and
- S is an element of N.

Instead of (A, w) for an element of P, we shall write $A \to w$. Elements of P are called context-free rules. We set $V_G = N \cup T$. The derivation process in a context-free grammar and the generated language are defined as usually (see e.g. [17]).

A *Russian parallel grammar* is a quintuple $G = (N, T, P_1, P_2, S)$, where
- N, T, and S are specified as in a context-free grammar, and
- P_1 and P_2 are finite sets of $N \times (N \cup T)^*$.

$x \in V_G^+$ directly derives $y \in V_G^*$ (written as $x \Longrightarrow y$), iff one of the following conditions hold:
- $x = x'Ax''$, $y = x'wx''$ for some $x', x'' \in V_G^*$ and $A \to w \in P_1$ or
- $x = x_0Ax_1Ax_2 \ldots x_{n-1}Ax_n$, $n \geq 0$, $x_i \in (V_G \setminus \{A\})^*$ for $1 \leq i \leq n$, $A \to w \in P_2$ and $y = x_0wx_1wx_2 \ldots x_{n-1}wx_n$.

The language $L(G)$ generated by the Russian parallel grammar G is defined as

$$L(G) = \{z \mid z \in T^*,\ S \Longrightarrow^* z\},$$

where $\Longrightarrow^*$ is the reflexive and transitive closure of $\Longrightarrow$.

A *random context grammar* is a quadruple $G = (N, T, P, S)$ where

– N, T and S are specified as in a context-free grammar, and

– P is a finite set of triples $r = (p, R, Q)$ where p is a context-free production and R and Q are subsets of N.

G is called a *forbidden* random context grammar if all rules of P are of the form $(p, \emptyset, Q)$. For $x, y \in V_G^*$, we say that x directly derives y, written as $x \Longrightarrow y$, iff there is a triple $r = (A \to w, R, Q) \in P$ such that

– $x = x'Ax''$ and $y = x'wx''$ for some $x', x'' \in V_G^*$,

– any letter of R is contained in x, and no letter of Q occurs in x.

The language L(G) generated by G is defined as

$$L(G) = \{w \mid w \in T^*,\ S \Longrightarrow^* w\},$$

where $\Longrightarrow^*$ is the reflexive and transitive closure of $\Longrightarrow$.

A *extended tabled interactionless L system* (ET0L system) is an $(r+3)$-tuple $G = (V, T, P_1, P_2, \ldots P_r, w)$ where

– V is an alphabet, T is a subset of V,

– w is a non-empty word over V and,

– for $1 \leq i \leq r$, P_i is a finite subset of $V \times V^*$ such that, for any $a \in V$, there is at least one element (a, v) in P_i.

Again, we shall write $a \to v$ instead of (a, v). $x \in V^+$ directly derives $y \in V^*$ (written as $x \Longrightarrow y$), if

– $x = x_1x_2 \ldots x_n$ for some $n \geq 0$, $x_i \in V$, $1 \leq i \leq n$,

– $y = y_1y_2 \ldots y_n$ and

– there is an j, $1 \leq j \leq r$ such that $x_i \to y_i \in P_j$ for $1 \leq i \leq n$.

The language $L(G)$ generated by the ET0L system G is defined as

$$L(G) = \{z \mid z \in T^*,\ w \Longrightarrow^* z\},$$

where $\Longrightarrow^*$ is the reflexive and transitive closure of $\Longrightarrow$.

A *random context ET0L system* (RCET0L system in short) is an $(r+3)$-tuple $G = (V, T, P_1, P_2, \ldots P_r, w)$ where

– for $1 \leq i \leq r$, $P_i = (P_i', R_i, Q_i)$ where R_i and Q_i are subsets of V,

– $G' = (V, T, P_1', P_2', \ldots P_r', w)$ is an ET0L system.

$x \in V^+$ directly derives $y \in V^*$ (written as $x \Longrightarrow y$), if

– any letter of R_i occurs in x and no letter of Q_i occurs in x, and

– $x \Longrightarrow y$ holds with respect P_i' in G'.

The language $L(G)$ generated by the RCET0L system G is defined as

$$L(G) = \{z \mid z \in T^*,\ w \Longrightarrow^* z\},$$

where $\Longrightarrow^*$ is the reflexive and transitive closure of $\Longrightarrow$.

By $\mathcal{L}(CF)$, $\mathcal{L}(RC)$, $\mathcal{L}(fRC)$, $\mathcal{L}(rp)$, L$(ET0L)$ and $\mathcal{L}(RCET0L)$ we denote the families of all context-free languages, random context languages, forbidden random context languages, Russian parallel languages, ET0L languages and random context

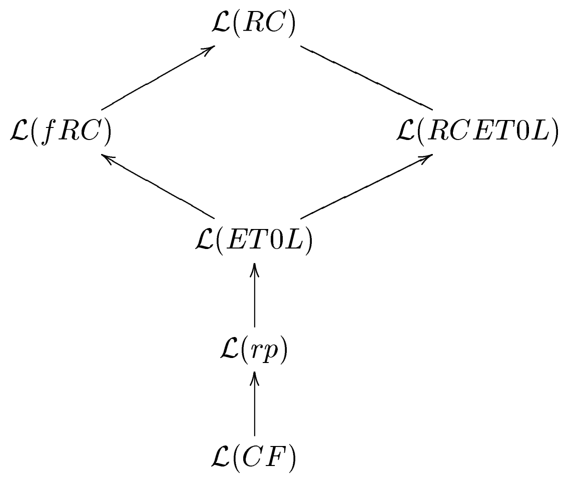

Figure 1:

ET0L languages. For a detailed information on these languages we refer to [10], [12], [17] and [16].

In Figure 1 we recall the hierarchy of the languages defined above.

3 Cooperating Distributed Grammar Systems

We now present the notion of cooperating distributed grammar systems.

A *cooperating distributed grammar system* (for short, CD grammar system is an $(n+5)$-tuple

$$G = (N, T, P_1, P_2, \dots, P_n, S, c, c')\,,$$

where
– N is a set of nonterminals and T is a set of terminals,
– for $1 \le i \le n$, the component P_i is a set of context-free productions,
– $S \in N$ is the start element,
– c is the start condition and c' is the stop condition, i.e., c and c' are predicates on

$$\{(P, w) \mid P \subset N \times (N \cup T)^* \text{ and } w \in (N \cup T)^*\}$$

(the conditions are defined on pairs consisting of a set of context-free productions and a word).

We say that

$$x = x_0 \Longrightarrow x_1 \Longrightarrow x_2 \Longrightarrow \dots \Longrightarrow x_m = y$$

is a derivation with respect to P_i and the conditions c and c' (written as $x \Longrightarrow^*_{P_i} y$) if
– $c(P_i, x)$ is true,
– for $1 \le j \le m+1$, $x_j \Longrightarrow x_{j+1}$ is a direct derivation step using a production of P_i
– for $1 \le j \le m-1$, $c'(P_i, x_j)$ is not true, and
– $c'(P_i, y)$ is true or y contains no letter of $dom(P_i)$.

The language generated by G consists of all words z over T such that there is derivation

$$S = z_0 \Longrightarrow^*_{P_{i_1}} z_1 \Longrightarrow^*_{i_2} z_2 \Longrightarrow^*_{P_{i_3}} \ldots \Longrightarrow^*_{P_{i_m}} z_m = z\,.$$

In this paper we discuss CD grammar systems where the start and stop conditions depend on the competence of the components to work on the sentential form. The competence is formalized in the following way.

Let N and T be two disjoint sets (of nonterminals and terminals, respectively). Further, let $V = N \cup T$ and let $P \subseteq N \times V^*$ be a (finite) set of context-free productions and w be a word over V. We set

$$nt(w) = \{A \mid A \in N \text{ and } \#_A(w) \geq 1\}$$

(i.e., $nt(w)$ is the set of nonterminals which occur in w),

$$dom(P) = \{A \mid A \to w \in P \text{ for some } w \in V^*\}$$

and

$$comp(P_i, w)) = \#(nt(w) \cap dom(P_i))\,.$$

We say that P has

competence k on w	iff	$comp(P_i, w) = k$,
competence $\leq k$ on w	iff	$comp(P_i, w) \leq k$,
competence $\geq k$ on w	iff	$comp(P_i, w) \geq k$,
maximal competence on w	iff	$comp(P_i, w) \geq comp(P_j, w)$ for $1 \leq j \leq n$,
full competence on w	iff	$nt(w) \subseteq dom(P_i)$.

The most obvious idea is to require that a component can start a derivation if it has a certain competence A and has to stop if it does not have the competence A. Therefore besides the competence conditions introduced above we have to consider the negations of the above conditions. Clearly, the negations can also be used as start conditions. Obviously, P has not competence $\geq k$ ($\leq k$) iff it has competence $\leq k-1$ ($\geq k+1$, respectively). Therefore we get the following additional notions.

P_i has competence $\neq k$ on w	iff	$comp(P_i, w) \neq k$,
P_i is not maximal competent on w	iff	there is a P_j, $1 \leq j \leq n$, $i \neq j$, with $comp(P_j, w) > comp(P_i, w)$,
P_i is not fully competent on w	iff	there is an $A \in nt(w)$ with $A \notin dom(P_i)$.

By these definitions, competence of P_i on w is given, if $dom(P_i)$ contains at least one element of $nt(w)$. Thus we also formulate:

P_i is competent on w	iff	$comp(P_i, w) \geq 1$,
P_i is not competent on w	iff	$comp(P_i, w) = 0$.

We use the following abbreviations for the competence conditions (in the order of their definition):

$$= k,\ \leq k,\ \geq k,\ max,\ full,\ \neq k,\ \neg max,\ \neg full,\ comp,\ \neg comp\,.$$

Because *comp* and ≥ 1, $\neq 1$ and ≥ 2 as well as ≤ 1 and $= 1$ coincide, we can restrict to $k \geq 2$ and have to add the condition $= 1$.

As an example we consider the CD grammar system

$$\begin{aligned}
G &= (\{S, A, B, A', B', F\}, \{a, b, c\}, P_1, P_2, P_3, P_4, P_5, P_6, S, max, \neg max),\\
P_1 &= \{S \to AB\},\\
P_2 &= \{A \to ab, B \to c\},\\
P_3 &= \{A \to aA'b, B \to F\},\\
P_4 &= \{A' \to F, B \to B'c\},\\
P_5 &= \{A' \to A, B' \to F\},\\
P_6 &= \{A \to F, B' \to B\}.
\end{aligned}$$

Obviously, after the application of P_1, P_2 and P_3 have maximal competence. If we apply P_2, then we have to terminate the derivation and get abc. If we apply P_3, we only can apply $A \to aA'b$ and loose the maximal competence. We have to apply in succession $B \to B'c$ of P_4, $A' \to A$ of P_5 and $B' \to B$ of P_6 and obtain $aAbBc$ (if we apply the other rule of the components we get a sentential form containing F and we cannot terminate the derivation). We can iterate this process. Therefore

$$L(G) = \{a^n b^n c^n \mid n \geq 1\}.$$

Note that we cannot combine an arbitrary competence condition as start condition with another arbitrary competence condition as a stop condition. Obviously, we cannot take the same condition as start condition as well as stop condition, because by definition the component can start the derivation but it also has to stop the derivation, which is impossible. However, there are also other situations of this type. For instance, $full$ and max cannot be taken as start and stop condition, respectively, since any full competent component is maximal, too.

Moreover, we have to exclude those combination which do not allow terminating derivations. For instance, this situation occurs in the case of start condition ≥ 4 and stop condition ≤ 2. If a derivation starts then the sentential word has at least four nonterminals. However, if the derived sentential form contains at most 2 nonterminals, we have to stop the derivation (and no component can continue the derivation by the start condition). Therefore we cannot derive terminal words. This situation also occurs if we have stop conditions of the types $\leq k$ with $k \geq 2$, $= 1$ and *comp*.

Furthermore, we shall exclude such situations where the start condition as well as the stop condition can be satisfied, but it is not necessarily true that both have to be valid. For instance, if $N = \{A, B, C\}$ and the start condition requires the competence ≥ 2 and the stop condition requires competence ≥ 3, then the derivation can start start on AB, but it cannot start on ABC since the start condition as well as the stop condition are satisfied. The pairs $(max, full)$ and $(full, = k)$ with $k \geq 2$ of start and stop conditions can also lead to conflicts.

Taking all these restrictions into consideration we have to investigate the following pairs of of start conditions and stop conditions, where $k \geq 2$ and $l \geq 2$:

(a) $(full, \neg full)$, $(full, \neg comp)$, $(max, \neg max)$, $(max, \neg comp)$, $(comp, \neg comp)$, $(= l, \neg comp)$, $(\geq l, \neg comp)$, $(\leq l, \neg comp)$, $(= 1, \neg comp)$, $(= 1, \geq k)$, $(= 1, = k)$,

(b) $(full, \neg max)$, $(\neg full, full)$, $(\neg full, \neg comp)$, $(\neg max, max)$, $(\leq l, = k)$ and $(\leq l, \geq k)$ with $l < k$, $(= l, = k)$ with $l \neq k$, $(= l, \geq k)$ with $l < k$, $(\neq k, = k)$

By $\mathcal{L}(c, c')$ we denote the family of all languages which can be generated by CD grammar systems with the start condition c and the stop condition c'. If we restrict the CD grammar systems which have at most n components, then we denote the corresponding family of languages by $\mathcal{L}_n(c, c')$.

By definition, we have the following statement.

Lemma 1 *For any $n \geq 1$ and any pair (c, c') of start and stop conditions,*

$$\mathcal{L}_n(c, c') \subseteq \mathcal{L}_{n+1}(c, c') \subseteq \mathcal{L}(c, c')\,.$$

4 Results

In this section we present the results which – at least partially – give the place of some families $\mathcal{L}(c, c')$ within the hierarchy given in Figure 1.

Theorem 2 *([6]) For any $n \geq 3$,*

$$\begin{aligned}\mathcal{L}(CF) &= \mathcal{L}_1(comp, \neg comp) = \mathcal{L}_2(comp, \neg comp)\\ &\subset \mathcal{L}_n(comp, \neg comp) = \mathcal{L}(comp, \neg comp) = \mathcal{L}(ET0L)\,.\end{aligned}$$

Theorem 3 *([14], [5], [4]) For any $n \geq 3$,*

$$\begin{aligned}\mathcal{L}(CF) &= \mathcal{L}_1(full, \neg full) \subset \mathcal{L}_2(full, \neg full)\\ &\subseteq \mathcal{L}_n(full, \neg full) = \mathcal{L}(full, \neg full) = \mathcal{L}(RC)\,.\end{aligned}$$

Theorem 4 *For any $n \geq 3$,*

$$\begin{aligned}\mathcal{L}(CF) &= \mathcal{L}_1(full, \neg comp) \subseteq \mathcal{L}_2(full, \neg comp)\\ &\subseteq \mathcal{L}_n(full, \neg comp) = \mathcal{L}(full, \neg comp) = \mathcal{L}(ET0L)\,.\end{aligned}$$

Proof. $\mathcal{L}(full, \neg comp) \subseteq \mathcal{L}(ET0L)$. Let $L \in \mathcal{L}(full, \neg comp)$. Then $L = L(G)$ for some CD grammar system $G = (N, T, P_1, P_2, \ldots, P_r, w, full, \neg comp)$. We set

$$\begin{aligned}N' &= N \cup \{A' \mid A \in N\} \cup \{F\},\\ h_i(a) &= \begin{cases} a & \text{if } a \in dom(P_i) \cup T \\ a' & \text{if } a \notin dom(P_i) \end{cases} \quad \text{for } 1 \leq i \leq n,\\ Q_i &= \{A \to h_i(w) \mid A \to w \in P_i\} \cup \{B \to F \mid B \notin dom(P_i)\}, \text{ for } 1 \leq i \leq n,\\ Q &= \{A' \to A \mid A \in N\},\\ G' &= (N', T, Q_1, Q_2, \ldots Q_r, Q, w, comp, \neg comp)\end{aligned}$$

Let $1 \leq i \leq n$. If a word x contains a symbol $B \notin dom(P_i)$, then Q_i derives in G' from x a word containing the letter F. Since there is no rule with left-hand side F in any component of G', we cannot terminate the derivation. Therefore in a terminating derivation we can only apply a component Q_i to x if $nt(x) \cap (N \setminus dom(P_i) = \emptyset$, or equivalently, $nt(x) \subseteq dom(P_i)$, i.e., P_i is fully competent on x. Therefore Q_i is applicable in a terminating derivation in G' to x if and only if P_i is fully competent on x if and only if P_i is applicable to x in G.

Moreover, in both cases we only stop if the component is not competent on the derived sentential form. Since we introduce primed versions of the letters of $N \setminus dom(P_i)$ applying Q_i we obtain

$$x \Longrightarrow_{P_i}^{*} y \quad \text{iff} \quad x \Longrightarrow_{Q_i}^{*} h_i(y)\,.$$

Furthermore, no component Q_i with $1 \leq i \leq n$ is competent on $h_i(y)$ since it only contains primed letters and terminals. Thus we have to continue with Q which replaces all primed letters A' by their original A. Thus

$$x \Longrightarrow_{P_i}^{*} y \quad \text{iff} \quad x \Longrightarrow_{Q_i}^{*} h_i(y) \Longrightarrow_{Q}^{*} y\,.$$

This implies $L(G) = L(G')$. By Lemma 2, $L = L(G) = L(G') \in \mathcal{L}(ET0L)$.

$\mathcal{L}(ET0L) \subseteq \mathcal{L}_3(full, \neg comp)$. In [6], Theorem 3, iii), it has been shown that, for any ET0L language L, there is a cooperating distributed grammar system $G = (N, T, P_1, P_2, P_3, S, comp, \neg comp)$ such that $L = L(G)$. Moreover, it is easy to see, that any component of G is competent on a sentential form x if and only if it is fully competent on x. Hence $G' = (N, T, P_1, P_2, P_3, S, full, \neg comp)$ generates L, too. Thus $\mathcal{L}(ET0L) \subseteq \mathcal{L}_3(full, \neg comp)$.

By Lemma 1, we obtain

$$\mathcal{L}(ET0L) \subseteq \mathcal{L}_3(full, \neg comp) \subseteq \mathcal{L}_n(full, \neg comp) \subseteq \mathcal{L}(full, \neg comp) \subseteq \mathcal{L}(ET0L)$$

for any $n \geq 4$, which implies

$$\mathcal{L}_m(full, \neg comp) = \mathcal{L}(full, \neg comp) = \mathcal{L}(ET0L)$$

for any $m \geq 3$. Obviously, $\mathcal{L}(CF) = \mathcal{L}_1(full, \neg comp)$. Now the assertion follows by Lemma 1. □

Theorem 5 *([7])* $\mathcal{L}(rp) \subset \mathcal{L}(max, \neg max) \subseteq \mathcal{L}(RC)$.

Theorem 6 $\mathcal{L}(max, \neg comp) = \mathcal{L}(RCET0L)$.

Proof. $\mathcal{L}(RCET0L) \subseteq \mathcal{L}(max, \neg comp)$. Let $L \in \mathcal{L}(RCET0L)$. Then there is a random context ET0L system

$$G = (V, T, (P_1, R_1, Q_1), (P_2, R_2, Q_2), \ldots, (P_r, R_r, Q_r), w)$$

such that $L = L(G)$. Let

$$\begin{aligned} V &= \{a_1, a_2, \ldots, a_n\}\,, \\ R_i &= \{a_{i,1}, a_{i,2}, \ldots, a_{i,s_i}\} \text{ for } 1 \leq i \leq r\,. \end{aligned}$$

We set

$$
\begin{aligned}
N &= \{a' \mid a \in V\} \cup \{a'' \mid a \in V\} \cup \{A, S, F\} \cup \{B_i \mid 1 \le i \le n+1\} \\
&\quad \cup \{A_{i,j,k} \mid 1 \le i \le r, 1 \le j \le s_i + 2, 1 \le k \le n+1\}, \\
Z_{init} &= \{S \to AA_{i,1,1}A_{i,1,2}\dots A_{i,1,n+1}w' \mid 1 \le i \le r\} \\
&\quad \cup \{S \to B_1B_2\dots B_{n+1}w'\}, \\
Z_{fin} &= \{B_i \to \lambda \mid 1 \le i \le n+1\} \cup \{a' \to F \mid a' \in N \setminus T'\} \cup \{a' \to a \mid a \in T\}, \\
Z_{i,j} &= \{A_{i,j,k} \to A_{i,j+1,k} \mid 1 \le k \le n+1\} \cup \{a'_{i,j} \to a''_{i,j}\} \\
&\qquad \text{for } 1 \le i \le r, 1 \le j \le s_i, \\
Z'_{i,j} &= \{A_{i,j,k} \to A_{i,j+1,k} \mid 1 \le k \le n+1\} \cup \{A \to F\} \\
&\qquad \text{for } 1 \le i \le r, 1 \le j \le s_i, \\
Z_{i,s_i+1} &= \{A_{i,s_i+1,k} \to A_{i,s_i+2,k} \mid 1 \le k \le n+1\} \cup \{a' \to a'' \mid a \in V \setminus Q_i\} \\
&\quad \cup \{b' \to F \mid b \in Q_i\} \text{ for } 1 \le i \le r, \\
Z_{i,s_i+2} &= \{A_{i,s_i+2,k} \to \lambda \mid 1 \le k \le n+1\} \cup \{a'' \to u' \mid a \to u \in P_i\} \\
&\quad \cup \{A \to AA_{i',1,1}A_{i',1,2}\dots A_{i',1,n+1} \mid 1 \le i' \le r\} \\
&\quad \cup \{A \to B_1B_2\dots B_{n+1}\} \text{ for } 1 \le i \le r, \\
G' &= (N, T, Z_{init}, Z_{fin}, Z_{1,1}, Z_{1,2}\dots, Z_{1,s_1+2}, Z_{2,1}, \dots, Z_{r,s_r+2}, \\
&\quad Z'_{1,1}, Z'_{1,2}\dots, Z'_{1,s_1}, Z'_{2,1}, \dots, Z'_{r,s_r}, S, max, \neg comp).
\end{aligned}
$$

If we start a derivation in G' (from S) we get $AA_{i,1,1}A_{i,1,2}\dots A_{i,1,n+1}w'$ for some i, $1 \le i \le r$, or $B_1B_2\dots B_{n+1}w'$.

We now discuss the continuation of the derivation of a sentential form of type $B_1B_2\dots B_{n+1}v'$ where v is a sentential form of G. Since v' contains at most n different letters, Z_{fin} is the only component with maximal competence and we derive the terminal word v or a word containing an occurrence of F, i.e., the derivation cannot be terminated. If v is a terminal word, then $v \in L(G)$ as well as $v \in L(G')$.

Now let us consider the derivation starting from $AA_{i,1,1}A_{i,1,2}\dots A_{i,1,n+1}v'$ for some i, $1 \le i \le r$ and some sentential form of G. Now $Z'_{i,1}$ has maximal competence, and its applications yields a word containing F such that we cannot terminate the derivation. If $a'_{i,1}$ is present, then $Z_{i,1}$ has also maximal competence, and its application yields $AA_{i,2,1}A_{i,2,2}\dots A_{i,2,n+1}v_1$ where v_1 is obtained from v' by replacing any occurrence of $a'_{i,1}$ by $a''_{i,1}$. Therefore we cannot terminate the derivation or all letters of R_i are present in v and we have a derivation

$$AA_{i,1,1}A_{i,1,2}\dots A_{i,1,n+1}v' \Longrightarrow^*_{Z_{i,1}} \dots \Longrightarrow^*_{Z_{i,r_i}} AA_{i,r_i+1,1}A_{i,r_i,2}\dots A_{i,r_i,n+1}v_2$$

where all occurrences of primed versions of letters of R_i in v' are replaced by their doubly primed versions to obtain v_2. Now Z_{i,r_i+1} is the only component with maximal competence, and its application yields a word containing F, if v' contains a primed version of a letter in Q_i, or we get $AA_{i,r_i+2,1}A_{i,r_i+2,2}\dots A_{i,r_i+2,n+1}v''$. Now we have to apply the only component Z_{i,r_i+2} with maximal competence which results in $AA_{i',1,1}A_{i',1,2}\dots A_{i',1,n+1}z'$ or $B_1B_2\dots B_{n+1}z'$ where $v \Longrightarrow_{P_i} z$ is a derivation step in the RCET0L system G.

Hence we can simulate in G' all derivations in G, and any terminating derivation In G' corresponds to a derivation in G. Therefore $L(G') = L(G) = L$ which proves $L \in \mathcal{L}(max, \neg comp)$.

$\mathcal{L}(max, \neg comp) \subseteq \mathcal{L}(RCET0L)$. Let $L \in \mathcal{L}(max, \neg comp)$. Then $L = L(H)$ for some CD grammar system $H = (N, T, P_1, P_2, \ldots, P_n, S, max, \neg comp)$. Obviously, for any word $w \in (N \cup T)^*$, the set $nt(w)$ uniquely determines the set of components which are of maximal competence on w. Let $f : 2^N \to 2^{\{P_1, P_2, \ldots, P_n\}}$ be the function where $f(M)$ is the set of components which have maximal competence on words w with $M = nt(w)$.

We now construct the random context ET0L system H' with the the underlying alphabet $V = N \cup T \cup \{A\} \cup \{[i] \mid 1 \leq i \leq n\} \cup \{A_M \mid M \subseteq N\}$, the terminal alphabet T, the start word AS and the following tables:

$$(\{A \to A_M\} \cup \{x \to x \mid x \in V \setminus \{A\}\}, \{A\} \cup M, N \setminus M) \text{ for } M \subseteq N$$

(this table is only applicable to a word Av with $M = nt(v)$, i.e., by this table we determine $nt(v)$, and we obtain $A_{nt(v)}v$),

$$(A_M \to [i] \mid P_i \in f(M)\} \cup \{x \to x \mid x \in V \setminus \{A_M\}\}, \{A_M\}, \emptyset) \text{ for } M \subseteq N$$

(by this table we derive $[i]v$ from $A_{nt(v)}v$, where P_i has maximal competence on v),

$$([i] \to [i]\} \cup P_i \cup \{x \to x \mid x \in V \setminus (dom(P_i) \cup \{[i]\}\}, \{[i]\}, \emptyset)\,,$$
$$([i] \to A\} \cup \{x \to x \mid x \in V \setminus \{[i]\}\}, \{[i]\}, dom(P_i))$$

(by these tables, in H' we obtain a derivation

$$[i]v \Longrightarrow [i]v_1 \Longrightarrow [i]v_2 \Longrightarrow \ldots \Longrightarrow [i]v_m \Longrightarrow Az$$

where $v \Longrightarrow^*_{P_i} z$ holds in G, P_i has maximal competence on v and P_i is not competent on z),

$$(\{A \to \lambda\} \cup \{x \to x \mid x \in N \setminus \{A\}, \{A\}, N)$$

(we cancel A and produce a word over T). By the explanations added to the tables, it is easy to see that $L(H') = L(H) = L$ which proves $L \in \mathcal{L}(RCET0L)$. □

Theorem 7 *([1])* $\mathcal{L}(fRC) \subseteq \mathcal{L}(=1, \geq 2) \subseteq \mathcal{L}(RC)$.

Theorem 8 *([8])* $\mathcal{L}(=1, \neg comp) = \mathcal{L}(\geq 1, \neg comp) = \mathcal{L}(ET0L)$.

Theorem 9 *([8]) For all* $k \geq 2$, $\mathcal{L}(=k, \neg comp) = \mathcal{L}(\geq k, \neg comp) = \mathcal{L}(RCET0L)$.

Theorem 10 *([8]) For all* $k \geq 1$, $\mathcal{L}(\leq k, \neg comp) = \mathcal{L}(ET0L)$.

5 Concluding Remarks

In the preceding section we have characterized – in some cases only partially – the families given in item (a) in the list at the end of Section 3. We do not know any result on the families mentioned in item (b) such that the generative power of these families has to be studied.

Above we have mentioned that some combinations of start and stop conditions do not allow the derivation of terminal words. Especially, this holds for the combinations $(= k, \neq k)$ and $(\geq k, \leq k-1)$ for any $k \geq 2$. In order to get terminating derivations in these cases one has some possibilities.

In [1] and [2] the authors allow that there is a special terminating phase. If this phase is started, then the component has to stop with a terminal word and for all intermediate sentential forms of this step the competence of component is at most k.

Another approach is presented in [3] where in each derivation of a component one has to perform a parallel derivation as in L systems.

For the classical types of stop conditions hybrid systems have been introduced, i.e., the start and stop condition is not associated with the system, it is only associated with a component, and the start conditions and/or the stop conditions can be different for different components (see e.g. [15] and [13]). Such hybrid systems can also be defined if one uses competence based start and stop conditions.

References

[1] M. ter Beek, E. Csuhaj-Varjú, M. Holzer and Gy. Vaszil, On competence in cooperating distributed grammar systems. Submitted, 2004.

[2] M. ter Beek, E. Csuhaj-Varjú, M. Holzer and Gy. Vaszil, On competence in cooperating distributed grammar systems, part II. Submitted, 2004.

[3] M. ter Beek, E. Csuhaj-Varjú, M. Holzer and Gy. Vaszil, On competence in cooperating distributed grammar systems with parallel rewriting. Submitted, 2004.

[4] H. Bordihn, On the number of components in CD grammar systems. In: E. Csuhaj-Varjú and Gy. Vaszil (Eds.), Proceedings of the Conference on Descriptional Complexity of Formal Systems, Budapest, 2003.

[5] H. Bordihn and E. Csuhaj-Varjú, On competence and completeness in CD grammar systems. *Acta Cybernetica* **12** (1996) 347–361.

[6] E. Csuhaj-Varjú and J. Dassow, On cooperating/distributed grammar systems. *Journal of Information Processing and Cybernetics* (EIK) **26** (1990), 49–63.

[7] E. Csuhaj-Varjú, J. Dassow and M. Holzer, On a competence-based cooperation strategy in CD grammar systems. Submitted, 2004.

[8] E. Csuhaj-Varjú, J. Dassow and M. Holzer, CD grammar systems with competence based entry conditions in their cooperation protocols. Submitted, 2004.

[9] E. Csuhaj-Varjú, J. Dassow, J. Kelemen and Gh. Păun, *Grammar Systems - A Grammatical Approach to Distribution and Cooperation.* Topics in Computer Mathematics 5, Gordon and Breach Science Publishers, Yverdon, 1994.

[10] J. Dassow and Gh. Păun, *Regulated Rewriting in Formal Language Theory.* EATCS Monograph on Theoretical Computer Science 18, Springer-Verlag, Berlin-Heidelberg-New York, 1989.

[11] J. Dassow, Gh. Păun and G. Rozenberg, Grammar systems. In: [17], Vol. II, Chapter 4, 155-213.

[12] J. Dassow, Gh. Păun and A. Salomaa, Grammar with controlled derivations. In: [17], Vol. II, Chapter 3, 101-154.

[13] H. Fernau, M. Holzer and R. Freund, Hybrid modes in cooperating distributed grammar systems: internal versus external hybridization. *Theor. Comp. Sci.* **259** (2001) 404–426.

[14] R. Meersman and G. Rozenberg, Cooperating grammar systems. In: J. Winkowski (ed.), Proceeding Mathematical Foundations of Computer Science 1978, Lecture Notes in Computer Science 64, Springer-Verlag, Berlin, 1978, 364–374.

[15] V. Mitrana, Hybrid cooperating/distributed grammar systems. *Computer and artrificial Intelligence* **12** (1993) 83–88.

[16] G. Rozenberg and A. Salomaa *The Mathematical Theory of L Systems.* Academic Press, New York, 1980.

[17] G. Rozenberg and A. Salomaa (Eds.), *Handbook of Formal Languages.* Vol. I – III, Springer Verlag, Berlin-Heidelberg-New York, 1997.