Cooperating Distributed Grammar systems: A Comparison of Generative Power

Ondřej Ryšavý

Modern Theoretical Computer Science

November 30, 2002

Abstract: *Cooperating Distributed Grammar systems have been introduced for describing multi-agent systems by means of formal grammars and languages. There are currently a lot of variations that differ by generative power. This article is focused on summarize definitions of selected CD grammar systems (homogenous, hybrid, and teams) and comparison of their generative power among themselves and against to languages of other systems, e.g. E0L/ET0L, CF, CS, and RE.*

Keywords: *formal language theory, Cooperating Distributed Grammar Systems, generative power, teams*

Acknowledgement*: This article was originally intended as final works in Modern Theoretical Computer Science course at Brno Technical University.*

1 Introduction

Cooperating distributed grammar systems have been introduced by E. Csuhaj-Varju and J. Dassow in 1988 for describing multi-agent systems by means of formal grammars and languages, based on blackboard architectures. The theory provides theoretical frameworks and tools mostly for describing multi-agent systems, such as distributed and cooperative systems of problem solving, collective robotics, parallel computers, computer networks, distributed databases, and other areas where set of agents work together in some well-defined manner. From other point of view, CD grammar systems also may take place in a gap between Context Free Grammars and Context Sensitive Grammars. We would like to have a grammar with a nature of Context Free grammar and with power closes to Context Sensitive grammar that could provides more suitable formal basis for many problems come from real world. In [2] reader can find some examples where Context Free Grammars are not sufficient. Several formal systems based on grammars have been studied to manage such problems, particularly systems uses regulated rewriting (matrix grammars, programmed grammars, E0L, ET0L) or cooperation of grammars (grammar systems). Since the usual formal language systems describe sequential systems on symbolic level, CD grammar systems do similar with cooperating systems, achieving this by using set of components. Informally saying, a grammar system consists of several grammars (automata, or other mechanisms representing languages) that cooperate according to some well-defined protocol in deriving sentential forms of a language. The components of the system correspond to the agents. The language or the string sequences identifying the current state of the system represent the behavior. At each moment, only one grammar is active and rewrites shared string. The conditions under which a component can become active or it is disabled and leaves the string to other components are specified by the *cooperation protocol*. The language consists of terminal strings generated under the conditions of system. Our interest will be concerned in investigation of what power we can reach by using CD grammar system with a given number of components and works with specified cooperation protocol.

2 Definitions

In this section definitions of cooperative distributed grammar systems will appear. It introduces a several type of CD grammar systems and assigns them notations that will be used later when generative capacity will be compared. Knowledge of notation and meaning of common terms of formal language theory is expected. We will use N for set of non-terminals, T as set of terminals, S will be start symbol and P set of productions as usual. Abbreviations *L(REG), L(CF), L(CS)*, and *L(RE)* are used to address family of regular, context free, context sensitive, and recursive languages. A CD grammar system consists of a finite set of grammars that cooperate in deriving words of a common language, where the grammars work on the sentential form in turns according to some cooperation protocol (the mode of derivation). We will consider that grammars, which may participate in grammar systems, are restricted on Context-Free Grammars. Their using in grammar systems allows to increase a power of grammar systems over the power of participating grammars, in contrast of cases when suppose other types [3]. We also avoid combination of different type of grammars due to complex definition of their power. The mode of derivation is important feature of grammar systems that together with size of grammar systems (number of participating grammars) produce generative power of grammar system. At the beginning a definition of common CD grammar systems is introduced. Further we will add their variations.

Definition 1: A cooperating distributed grammar system is a structure

> $\Gamma = (N, T, S, P_1, \ldots, P_n)$, $n \geq 1$ where
> N is set of nonterminals,
> T is set of terminals,
> $S \in N$ is an axiom,
> P_i is finite set of context-free productions over $N \cup T$, called component of Γ

Each component is represented with grammar that contains production rules over an alphabet $N \cup T$. All components share sets N and T, however only the nonterminals appear on the left side of productions of the grammar are considered as nonterminals inside the grammar. Using of a derivation mode controls a Cooperation between components. In each time only one grammar may be active. After it finishes, another one may continue. A derivation mode decides how many steps one active grammar may do in its turn. We will define set of derivation modes $D = \{T,*\} \cup \{=k, \leq k, \geq k : k > 0\}$ where $k > 0$ having items with following meaning:

> T mode – Component derives sentence until there is not any applicable production
> * mode – Component can do arbitrary amount of steps.
> = k mode – Each component has to do exactly k steps, before another one can continue.
> ≤ k mode – Component can do at most k derivation steps in each its turn.
> ≥ k mode – Component has to do at least k derivation steps.

We introduce a denotation CD_n for family of languages generated by CD grammar system with n components. If we do not take care of count of components then we use abbreviation $CD = \bigcup_{n \geq 1} CD_n$. The derivation mode defines manner of using productions of grammars. Since derivation mode has influence on generative power and due to conventions a CD grammar systems will be distinguished to that use the same derivation mode for all components and that do not. We call the CD grammar system, which uses exactly one mode for every component in every time, as homogenous, otherwise system is called as hybrid. Next two definitions express this precisely.

Definition 2: Homogenous CD grammar system is a grammar system in which one derivation mode holds for every component. We denote such family as $CD_n(f)$, where n is number of components and $f \in D$ is derivation mode valid for entire system.

Definition 3: Consider a CD grammar system $\Gamma = (N, T, S, P_1, \ldots, P_n)$. The language generated by Γ in derivation mode $f \in D$ is denoted by

$$L_f(\Gamma) = \left\{ z \in T^* \,\middle|\, S \Rightarrow_{i_1}^{f} w_1 \Rightarrow_{i_2}^{f} \ldots \Rightarrow_{i_m}^{f} w_m = z, m > 1, 1 \leq i_j \leq n \right\}.$$

Definition 4: Hybrid CD grammar system is a construct:

$\Gamma = (N, T, S, (P_1,f_1) \dots, (P_n,f_n))$, $n \geq 1$ where

N,T,S,P_i are above defined sets of CD grammar systems, and $f_i \in D$ is the derivation mode in the i-th component of Γ. We denote such defined family of hybrid grammar systems as HCD_n. We set denotation $HCD = \bigcup_{n \geq 1} HCD_n$ as in the case of homogenous CD grammar systems.

Definition 5: The language generated by a hybrid CD grammar system is set of strings:

$$L(G) = \left\{ w \in T^* \middle| S \Rightarrow_{i_1}^{f_{i_1}} w_1 \Rightarrow_{i_2}^{f_{i_2}} \dots \Rightarrow_{i_m}^{f_{i_m}} w_m = w \quad with\ m \geq 1, 1 \leq i_j \leq n, and\ 1 \leq j \leq m \right\}$$

So far we discussed CD grammar systems consisted of independent components. The next modification will assume that components do their works cooperatively. Team CD grammar systems, besides to usually CD grammar systems, defines set of teams, each of which consists of a set of components that work together. Each derivation step is made by all components of active team. Exactly one production rule of each component is selected for rewriting that happens concurrently.

Team CD grammar systems in some sense appear between CD grammar systems and PC grammar systems, thought the ability to communication between grammars. Prescribed team CD grammar systems are introduced first together with team CD grammar system with free teams. The teams are composed by the definition of grammar system and hold together during the whole system life. We may distinguish grammar systems that consist of teams of same size from ones composing groups that have not same number of components in each team. In chapter three we will compare generative power of these two kinds of systems.

Definition 5: A prescribed team CD grammar system is a construct:

$\Gamma = (N, T, w, P_1, \dots, P_n, Q_1, \dots, Q_m)$, $n,m \geq 1$ *where*

N, T, P_i are as usually, and

$w \in N^*$ is an axiom,

$Q_i = \{P_{i_1}, P_{i_2} \dots, P_{i_{s_i}}\}$, for $1 \geq i \geq m$, are sets of components called teams.

Definition 6: A Free team CD grammar system is a construct:

$\Gamma = (N, T, w, P_1, \dots, P_n)$, $n \geq 1$ *where*

N, T, P_i are as usually, and

$w \in N^*$ is an axiom,

In this case teams are constructed automatically from components $P_1, \dots, P_n$. We denote free team CD grammar system as $T_k CD_n(f)$.

Definition 6: A team CD grammar system is called fixed size whether all teams have the same number of members, otherwise it is defined as variable size. The family of languages generated by CD grammar systems with prescribed teams of fixed size is denoted as $PT_kCD_n(f')$, where $k \geq 1$ represents a size of team. Similarly languages generated by variable size teams are denoted $PT_*CD_n(f')$. The letter n expresses number of components, f’ $\in$ f $\cup$ {t_0, t_1, t_2} determines used rewriting mode. Similar is defined for free team CD grammar systems.

As in case of other CD grammar systems, teams use rewriting modes. Fourth already defined ones are incorporated for team grammars together with three teams’ specific:

- t_0 A team must rewrite until it can not longer rewrite as team
- t_1 A team must rewrite until no component can not rewrite any longer
- t_2 A team must rewrite until there is a component that cannot rewrite any longer

Derivation modes t_0 and t_1 seems to be clear. The derivation mode t_2 allows continuing even some component cannot be used temporarily. However the last rewriting step has to be done by all components.
Since CD grammar systems were distinguished to homogenous and hybrid, also teams have hybrid modifications. Two basically different definitions on hybrid teams can be found. One can consider a hybrid CD grammar system and form teams automatically according to some fixed strategy or one can take prescribe team CD grammar system and associate mode to each group. The first, based on hybrid CD grammar systems, will be introduced earlier. We assume a hybrid CD grammar system and automatically form teams by combining all components with certain mode of derivation into a team. This team will inherit derivation mode from its components.

Definition 7: Consider a hybrid CD grammar system:

$$\Gamma = (N, T, S, (P_1,f_1) \ldots, (P_n,f_n)),$$

Then teams $(Q_i, g_i) \subseteq \{(P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n)\}$ can be automatically formed in the following way:

$$(Q_i, g_i) = \{(P_j, f_j) : f_k = g_i, 1 \le k \le n\}.$$

Team (Q_i, g_i) is called an automatically formed team operating in mode g_i.

The family of languages generated by hybrid CD grammar systems with automatically formed teams of variable size is denoted by HT_*CD_n.

Definition 8: A (prescribed) hybrid team CD grammar system is a construct

$$\Gamma = (N, T, S, P_1, P_2, \ldots, P_n, (Q_1, f_1), (Q_2, f_2), \ldots, (Q_m, f_m)),$$

where $(N, T, S, P_1, P_2, \ldots, P_n)$ is a CD grammar system and (Q_i, f_i) is called a prescribed team operating in mode f_i.

HPTmCDn denotes the family of languages generated by prescribed hybrid CD grammar systems with variant team size.

3 Generative capacity

In this section generative power of several variations of CD grammar systems will be investigated. Often it will be compared to Context Free grammar or E0L/ET0L systems that are nearest neighbors of CD grammar systems with respect to generative power. All

CD grammar systems are parameterized by number of components. Furthermore homogenous ones and teams are parameterized by derivation mode, respectively size of team. Presented lemmas will be mostly given without proof, eventually with short proof sketches. Similarly to other areas of formal language theories appearing last time there remain several open problems waiting to settle. The most important issues on grammar systems are discussed along with summarization of main achievement and future expectations in [6]. As it was mentioned early we consider only Context Free grammars in the place of components of CD grammar systems. Also from previous definitions it is clear that we assume only style "all" in which a CD grammar system accepts. The accepting style has not been discussed in this article and reader may find more about styles in [1].

First lemma charges that homogenous CD grammar systems do not exceed generative power of context-free languages if it use =1, ≥1, ∗, and ≤k, for k≥1 modes, even they have arbitrary many components.

Lemma 1: $CF = CD(=1) = CD(\geq 1) = CD(*) = CD(\leq k),\ \ k \geq 1$
This equality seems to be clear for mode =1, since each turn consists of exactly one using of some production of arbitrary component. Intuitively this do not differ from the way how the derivation in CF grammars works: CF grammar may be break up to *n* components each comprises exactly one production that will be activated in order to derive of sentential form. Since turn endures one step, the string is "checked" each derivation step and for all strings consist only of terminals will become sentences of language. In the other direction an appropriate CF grammar to CD grammar systems with derivation mode =1 consists of union over all components, i.e. union of their productions and each string of terminals derived by using productions of components of grammar systems are sentences of language the equivalently CF grammar system will derive the same language. For the rest considered modes we can in the worst case use only one-step turns and the equality holds as well.

Another introduced lemma will not be evident as previous was. It appoints relation between CD grammar systems with different number of components working under the mode =k and ≥k. This mode is more restricted than modes supposed in lemma 1, because it allows selecting sentences from the set of all string of terminals generated by productions of components. This feature does not affect power of CD grammar systems only with one component and also it was not known whether it further increases power of grammar systems with three or more components.

Lemma 2: $CF = CD_1(f) \subset CD_2(f) \subseteq CD_3(f) \subseteq \ldots \subseteq CD(f),\ \ f \in \{=k, \geq k \mid k \geq 2\}$
The equality between CF and $CD_1(=k)$ may be proven using by pushdown automata. We will show a construction pushdown automata for $CD_1(=k)$ language. Let G = (N, T, S, P) is CD grammar system working under the derivation mode =k for k ≥ 2. We will create an equivalent (against accepted language) pushdown automata PA = (Q, T, Γ, δ, q, [S, 0] #, ∅) accepting language by an emptying stack, in a such way:

$$Q = \{q\} \cup \{r_j \mid 0 \leq j < k\},$$
$$\Gamma = N \cup T \cup \{\#\} \cup \{[\#, i] \mid 0 \leq i < k\} \cup \{[x, i] \mid x \in (N \cup T) \wedge 0 \leq i < k\},$$

$\delta = \{((q, e, [\#,0]),(q, e, e))\}$
$\cup\{((q, e, [A, i]), (q, [a, (i+1) \bmod k]\alpha)) \mid \exists(A \to a\alpha)\in P, A\in N, a\in N\cup T, \alpha\in(N\cup T)^* \wedge 0 \le i < k\}$
$\cup\{((q, e,[A, i]), (r_{(i+1) \bmod k}, e)) \mid \exists(A \to e)\in P, A\in N, \wedge 0 \le i < k\}$
$\cup\{((q, a, [a, i]),(r_i, e)) \mid 0 \le i < k\}$
$\cup\{((r_i, e, a), (q, [a, i])) \mid 0 \le i < k\}$

The pushdown automata holds number of derivation steps that were made by equivalent grammar on the top of stack. Always if the top symbol is reduced an auxiliary state is used to mark another symbol on the stack. Push down automata can accept string only when on symbol [#,0] will left on stack. It will appear on stack only if derivation has exactly m.k steps for m>0. For derivation mode ≥k equivalent pushdown automata can be constructed in similar manner.

To show that generative power increases in two components CD grammar systems using derivation mode =k we use an example of language that it is not context free and is generated by such grammar system. Consider CD grammar system with two components $G = (\{S, A, B, C, A', B'\}, \{a,b,c\}, S, P_1, P_2)$, where components contain productions:

P_1:	P_2:
$S\to aBc$	$A'\to A$
$B\to A'C'$	$A'\to e$
$A\to aA'$	$C'\to bC$
$C\to C'c$	$C'\to b$

The grammar system generates language $L(G) = \{ a^n b^n c^n \mid n\ge 1 \}$, which is not context free.

The third lemma deals with CD grammar systems using terminating derivation mode. The generative power of system contains at most two components is equal to Context Free grammars. By adding another one component it can be reached generative power of ET0L systems but they cannot exceeded it even the number of components will increase.

Lemma 3: $CF = CD_1(t) = CD_2(t) \subset CD_3(t) = CD(t) = ET0L$

We will show that Context Free grammars have same generative power as $CD_2(t)$ grammars at first. Suppose CD grammar system $G=(N, T, S, P_1, P_2)$. We will compose two set N_1 and N_2 that each will consist of nonterminals that appears on the left side of productions P_1 and P_2 respectively. Moreover each symbol A_i of set N_1 will be marked as A_i^1 to be distinguished from symbols of set N_2. Similarly the symbol A_j of the set N_2 will be marked as A_j^2. Thus the sets N_1 and N_2 will be disjunctive. Then we will modify sets P_1 and P_2 to P_1' and P_2'. For each production $A\to\alpha_1 A_1\alpha_2 \ldots \alpha_n A_n\alpha_{n+1}$ $(A, A_i\in N, \alpha\in T^*)$ from P_1 the set P_1' will have the identical production except to the A will be replace to A^1 and each A_i that appears on the left side of some production from P_1 will be replaced by A_i^1. The nonterminals A_j that do not appear on the left side of any production from P_1 but has an occurrence on the left side of some P_2 production will be replaced by A_j^2. Other symbols will be unchanged. The modification of set P_2 into the set P_2' will be done in the same way. Due to use terminating mode the language generated original CD grammar system will be same as the CD grammar systems:

$$G' = (N_1 \cup N_2 \cup \{S, S^1, S^2\}, T, S, P_1' \cup \{S \to S^1 | S^2\}, P_2').$$

Finally since the nonterminal symbols of each component that appear on the left side of productions can not be mixed during derivation up we may define an equivalent CF grammar:

$$G'' = (N_1 \cup N_2 \cup \{S, S^1, S^2\}, T, S, P_1' \cup P_2' \cup \{S \to S^1 | S^2\}).$$

We will use an example to show that generative power of CD grammar system in terminating derivation mode with three components is greater then for CF language. Again the language $L(G) = \{a^n b^n c^n \mid n \geq 1\}$ will be constructed:

$$G = (\{S, A, B, A', B'\}, \{a, b, c\}, S, P_1, P_2, P_3),$$

P_1:	P_2:	P_3:
$S \to AB$	$A \to aA'$	$A \to a$
$A' \to A$	$B \to bB'c$	$B \to bc$
$B' \to B$		

Each component has a special role. The first starts and restarts generating process. Second ensures that generating may continue and the last component finishes derivation by a replacing of all nonterminal symbols by terminals.

The complete prove of $CD(t) = ET0L$ can be found in [1]. We now sketch the proof of $CD(t) = CD_3(t)$. An arbitrary $CD(t)$ grammar system $\Gamma = (N, T, S, P_1, P_2, \ldots, P_n)$ can be simulated by $CD_3(t)$ grammar system $\Gamma' = (N', T, [S,1], P_1', P_2', P_3')$ constructed as:

$$N' = \{[A,i] \| A \in N, 0 \leq i \leq n\},$$

$$P_1' = \{[A,i] \to [w,i] \| A \to w \in P_i, 1 \leq i \leq n\},$$

$$P_2' = \{[A,i] \to [A,i+1] \| A \in N, 1 \leq i \leq n, i \text{ } odd\}$$

$$\cup \{[A,n] \to [A,0] \| A \in N, if \text{ } n \text{ } is \text{ } odd\},$$

$$P_3' = \{[A,i] \to [A,i+1] \| A \in N, 1 \leq i \leq n, i \text{ } even\}$$

$$\cup \{[A,n] \to [A,1] \| A \in N, if \text{ } n \text{ } is \text{ } even\},$$

where $[w,i]$ denotes the string obtained by replacing each nonterminal $A \in N$ appearing in w by $[A,i]$ and leaving the terminals unchanged.

It can been easily seen that $L(\Gamma) = L(\Gamma')$: the rules in P_i are simulated by P_1' on nonterminals of the form $[A,i]$. The components P_2', P_3' change only the second terms of such symbols, for all the nonterminals in the sentential form. Thus during their simulation by P_1', the rules in P_i are never mixed with rules in P_j, for $i \neq j$. Therefore Γ and Γ' generate the same language.

Without a proof the selected facts on relations that holds when hybrid CD grammar systems are considered will be covered of next lemma.

Lemma 4: When $f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}, n \geq 1$, then following holds:

$$CD_n(f) \subseteq HCD_n,$$
$$CD(f) \subseteq HCD_3,$$
$$CD_1(f) = HCD_1,$$
$$ET0L = CD_3(t) \subseteq HCD_3,$$
$$ET0L \subset HCD_4.$$

The second part of this chapter is focused on generative power of teams CD grammar systems. When the situation around CD grammar systems was little complex due to a lot of modifications, in the case of teams it is much more difficult. There are several way how to construct a team. One may adding teams, or components into teams, changing mode, selecting from several homogenous or hybrid variations. Under our assumptions on Team CD grammar systems we may construct set consists of all teams:

$$\begin{aligned} Teams = \{T_s CD_n(f) \big| s = \{k \mid k \geq 1\} \cup \{*\}, n \geq 1, f = \{\geq k, \leq k, = k \mid k \geq 1\} \\ \cup \{PT_s CD_n(f) \big| s = \{k \mid k \geq 1\} \cup \{*\}, n \geq 1, f = \{\geq k, \leq k, = k \mid k \geq 1\} \cup \{*, t_0, t_1, t_2\}\} \\ \cup \{HPT_m CD_n, m, n \geq 1\} \cup \{HT_* CD_n \mid n \geq 1\} \end{aligned}$$

We declare the article will compare CD grammar systems. To let it as evident as possible we pick only such teams that generates significant family of languages. The relations between languages of selected teams grammar systems are depicted in the figure 1. The diagram consists of three parts. One part uses Chomsky hierarchy to present clear insight into generative power of teams in grammar systems. Other two show relative power among various type of teams. In the diagram, a fill arrow indicates inclusion that is not known to be proper, whereas an open arrow indicates a proper inclusion. Families which are not connected are not necessary incomparable.

Lemma 6: *Let* $s \geq 2, f \in \{*, t_0, t_1, t_2\} \cup \{\leq k, =k, \geq k \;/\; k \geq 1\}, f' \in \{=1, \geq 1, *\} \cup \{\leq k \;/\; k \geq 1\}$, *then*

$T_sCD(f) \subseteq PT_sCD(f) \subseteq PT_*CD(f)$,
$T_*CD(f) \subseteq PT_*CD(f) \subseteq HPT_*CD$,
$PT_sCD(f) \subseteq HPT_sCD \subseteq HPT_*CD$,
$HCD = HT_1CD \subseteq HPT_sCD \subseteq HPT_*CD$,
$HT_1CD \subseteq HT_*CD \subseteq HPT_*CD$, *z*

$T_sCD(f) \subseteq T_{s+1}CD(f)$,
$PT_sCD(f) \subseteq PT_{s+1}CD(f)$,
$HPT_sCD(f) \subseteq HPT_{s+1}CD(f)$,

$CF = T_1CD(f') \subset T_2CD(f')$,
$ET0L = T_1CD(t_1) \subset T_2CD(t_1)$,
$T_{s+1}CD(t_1) \subseteq T_2CD(t_1)$,

It is not surprising that range of generative power of team grammar systems covers several families of languages of Chomsky hierarchy. When summarizing, one can see that there are derivation modes in which the forming of teams strictly increases the power of CD grammar systems. Whether the same holds in the case of hybrid CD grammar systems is still unknown [3]. Also it was proven that there is no difference in generative power between the derivation modes $*, \leq k, = k$ and $\geq k$ (for a $k \geq 1$). Moreover an interesting result is that each recursive enumerable language can be represented by hybrid prescribed team CD grammar system with team of size two. Reader can find more relations and proofs on relations from lemma 6 in [4] and [5].

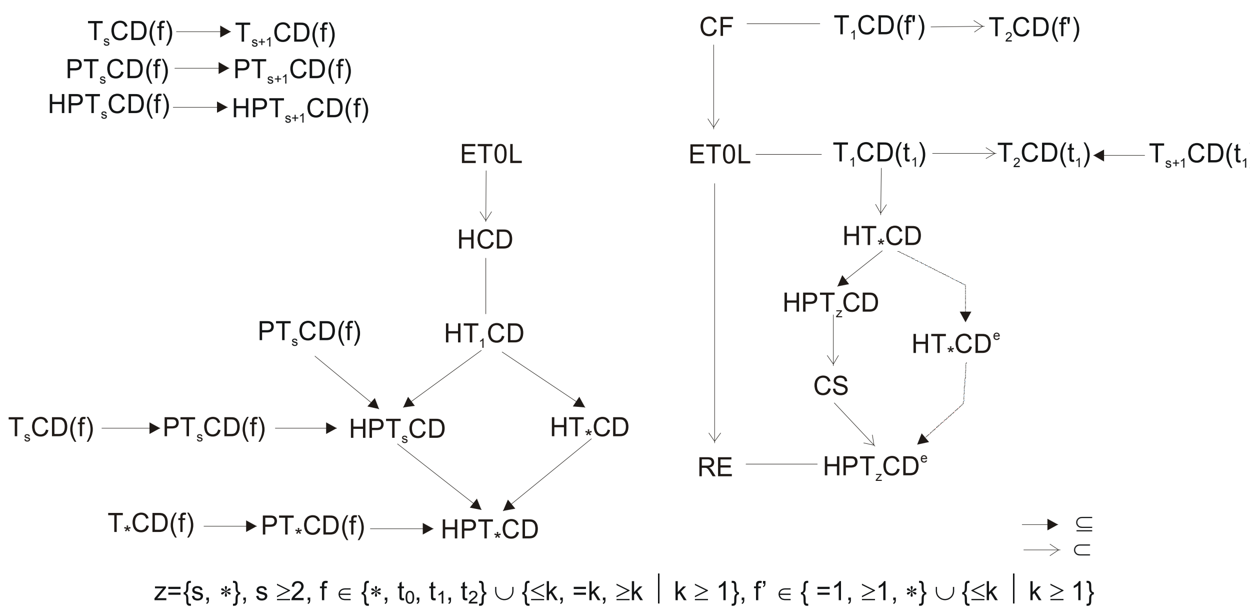

Figure 1: Selected relations between Team CD grammar systems

Many authors in their work stated relations to other families of grammars such as matrix grammars, programmed grammars, which were not discussed here. This comparison is interesting when one want to see how complex these grammars and grammar systems have to be to reach equivalent generative power. More can be found in [3], [4], and [5].

4 Conclusion

In this paper an overview on cooperating distributed grammar systems was presented. As was show in section 3 the class of cooperating grammar systems allows represent languages in the whole spectrum of Chomsky hierarchy. Other interesting variations that were not discussed here are Controlled Cooperating Distributed Grammar Systems and eco-grammar systems. Controlled Cooperating Distributed Grammar Systems deals with controlling of cooperation. Three different mechanisms how to select next working component are distinguished: external control, internal control and memories [1]. Since

an origin of grammar systems have foundation in AI area as a formal model for blackboard architecture, a grammar system introduced in [7] reflects an effort for describing live systems. An eco-grammar system introduces a special component called environment. The rest components are called agents. Each agent generates its own string as well as environment does. Productions applied on string depend on actual state of agent and environment. Thus agents affect an environment and vice-versa.
The grammar systems are still active area of formal language theory. They have been studied little more than a decade and due to existence of many complex variations there are still many open problems here [1]. The latest investigations are mainly focused on eco-grammar systems and on considering quantitative properties within cooperation protocol, e.g. priorities, or counting derivation steps of components according to study of game theoretic concepts.

5 References

[1] E. Csuhaj-Varjú, Grammar Systems - A Grammatical Approach to Distribution and Cooperation, 1992

[2] J. Dassow, G. Paun, Regulated Rewriting in Formal Language Theory, Springel-Verlang, 1989.

[3] M.H. ter Beek, Teams in grammar systems, IR-96-32 (master's thesis), Leiden University, 1996.

[4] L.Kari, A. Mateescu, Gh. Paun and A.Salomaa, Teams in cooperating grammar systems, J. Exper. Th. AI7 (1995), 347 - 359

[5] E.Scuhaj-Varjú and Gh. Paun, Limiting the size of teams in cooperating grammar systems. Bulletin EATCS 51(1993), 198 - 202.

[6] E. Csuhaj-Varjú, Grammar Systems: 12 years, 12 problems (Short version), In Proceedings of the MFCS 2000, Brno, 2000.

[7] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová and Gh.Paun, Eco(grammar) systems: a language theoretic model for AL. Manuscript, 1993