COOPERATING DISTRIBUTED GRAMMAR SYSTEMS AND GRAPH CONTROLLED GRAMMAR SYSTEMS WITH INFINITE NUMBER OF COMPONENTS

Filip Goldefus
Postgraduate Degree Programme (2), FIT BUT
E-mail: igoldefu@fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

ABSTRACT

This article deals with cooperative distributed grammar systems and cooperative distributed grammar systems controlled by graph with infinite number of components. Proofs of generative power for both types of grammar systems are provided. Both grammar systems are capable of generating arbitrary language, thus recursively enumerable languages. Graph controlled grammar systems with context-free productions, respectively with regular productions are more powerful than unrestricted grammars. Results are compared to grammar systems with finite number of components.

1 INTRODUCTION

Cooperative distributed grammar systems (see [1]) was constructed from the beginning on practical basis to simulate multiprocessor activities, for example group of agents working over one sentential form (consisting of symbols). This way of activity appears in many branches of computer science. Exclusive access to sources and their sharing is typical for hardware analysis and construction.

Cooperative distributed grammar systems were used as formalism estabilished on practical computing devices, which is constructable and works with limited generative power due the limits of finite sources. Cooperative distributed grammar systems are limited by number of components, typically finite. Altough cooperative distributed grammar system is mathematical model and limits of generative power were not inspected. Generative power of cooperative distributed grammars controlled by graph with infinite number of components and each component consist of finite set of regular productions is investigated in this paper. Cooperative ditributed grammar systems with regular productions and arbitrary finite number of components are as powerful as regular grammars.

Only one component of a cooperative distributed grammar system is active during derivation step or sequention of steps (depends on mode of derivation), performs a derivation step (or steps) and another (not exactly other) component is active. The way of passing active control is called protocol. There are many modes, but only mode $= 1$ and terminating mode are investigated. Mode $= 1$ means that a component of grammar system performs exactly one derivation step and then control is passed. Terminating mode means that a component of grammar system performs as many steps as possible and then control is passed.

Standard cooperative distributed grammar systems pass control between components nondeterministically. Cooperative distributed grammar systems controlled by graph eliminate nondeterminism by controlled passing of active component. This restriction increases generative power of cooperative grammar systems.

Results and proofs, which will be described in this article, are theoretical, but with regard to that grammar systems are theoretical constructs, then their limits has to be verified and explained.

Cooperative distributed grammar systems controlled by graph with finite number of components are practically usable in computer science, they have counterparts in processor controlling. Nondeterminism in hardware is nontypical and graph controlling eliminates nondeterminism. Another uninvestigated field is graph controlled cooperative distributed grammar systems with regular productions.

2 PRELIMINARIES

This paper assumes that reader is familiar with basic theory of automata and grammars (See [3]). For a word w over alphabet of terminal and nonterminal symbols $V = N \cup T$, $|w|$ denotes length of word and $w(i)$ denotes i-th symbol of w.

2.1 COOPERATIVE DISTRIBUTED GRAMMAR SYSTEMS

Definition 2.1. Cooperative distributed grammar system of degree n is (n+3)-tuple

$$\Gamma = (N, T, S, P_1, \dots, P_n),$$

where

- N is a finite set of nonterminal symbols (nonterminals),
- T is a finite set of terminal symbols (terminals),
- S is the starting nonterminal (axiom) and
- $(P_1, \dots, P_n)$ is a list (possibly inifinite) of finite sets of productions.
  - **context-free** - $X \to \alpha$, $X \in N$ and $\alpha \in V^*$,
  - **regular** - $X \to \alpha$, $X \in N$ and $\alpha \in T \cup TN$.

Cooperative distributed grammar systems work over one sentential form and thus configuration consists of string over V.

Every quaternion $G_i = (N, T, S, P_i)$ is i-th context-free grammar (component).

Definition 2.2. Let $\Gamma = (N, T, S, P_1, \dots, P_n)$ be a cooperative distributed grammar system, derivation step of i-th component using production $X \to w \in P_i$, where $u, v, w \in V^*$, $X \in N$, is relation of the form

$$uXv \Rightarrow_i uwv \; [X \to w]$$

Obviously, $\Rightarrow_i^*$ denotes reflexive and transitive closure of $\Rightarrow_i$.

Cooperative distributed grammar systems are based on cooperation, passing of control between components. Selection of a component is nondeterministic and cooperative distributed grammar systems controlled by graphs restrict this nondeterminism, by appointing which component is active in the next derivation step. Only one component rewrites actual sentential form, other wait for passing control.

Definition 2.3. Let $\Gamma = (N, T, S, P_1, \ldots, P_n)$ be a cooperative distributed grammar system,

- The one step derivation of i-th component (denoted by $\Rightarrow_i^{=1}$) is defined as

$$x \Rightarrow_i^{=1} y \text{ iff } x \Rightarrow_i y, \text{ for } x, y \in V^*$$

- The terminating derivation of i-th component (denoted by $\Rightarrow_i^t$) is defined as

$$x \Rightarrow_i^t y \text{ iff } x \Rightarrow_i^* y \text{ and there is no } z \in V^* \text{ such that } y \Rightarrow_i z$$

There exists other modes of derivation for cooperative distributed grammar systems, which can be found in e.g. [1].

Definition 2.4. The language of a cooperative distributed grammar system $\Gamma = (N, T, S, P_1, \ldots, P_n)$ is defined by $L(\Gamma) = \{w \in T^* \mid S \Rightarrow_{i_1}^f x_1 \Rightarrow_{i_2}^f \ldots \Rightarrow_{i_m}^f x_m = w,\ x_1, \ldots, x_m \in V^*,\ i_1, \ldots, i_m \in \{1, \ldots, n\},\ f \in \{=1, t\}\}$

Generally every cooperative distributed grammar system generates different languages depending on derivation mode.

2.2 GRAPH CONTROLLED COOPERATIVE DISTRIBUTED GRAMMAR SYSTEMS

Cooperative distributed grammar systems controlled by graph extend basic definition of cooperative distributed grammar systems limiting nondeterminism, which leads to increase of generative power ([2]).

Cooperative distributed grammar systems controlled by graph were studied only with components consisting one rule (see [2]). Definition which will follow uses components with arbitrary number (finite) of productions. This expansion leads to more flexible and natural connection with standard cooperative distributed grammar systems, every cooperative distributed grammar system is cooperative distributed grammar system controlled by graph with complete graph.

Definition 2.5. Cooperative distributed grammar system of degree n controlled by graph is $\Delta = (G, \Gamma)$, where Γ is cooperative distributed grammar system of degree n, $\Gamma = (N, T, S, P_1, P_2, \ldots, P_n)$ and G is a relation over sets of productions. Let $P = \{1, 2, \ldots, n\}$, so $G \subseteq P \times P$.

The relation G can be represented as graph which vertices are components of cooperative distributed grammar system Γ. Informally, standard cooperative distributed grammar systems are nondeterministic, thus selection of following component after execution of specified number of derivation steps is unpredictable. Limitation by control graph forbids such nondeterminism chosing set of consequent components, which can be used.

Definitions of one derivation step and two derivation modes are the same as in case of standard cooperative distributed grammar systems.

Definition 2.6. Let $\Delta = (G, \Gamma)$ be a cooperative distributed grammar system controlled by graph, the language is defined as

$$L(\Delta) = \{w \in T^* \quad | \quad S \Rightarrow^f_{i_1} x_1 \Rightarrow^f_{i_2} \ldots \Rightarrow^f_{i_m} x_m = w,\ x_1, \ldots, x_m \in V^*, i_1, \ldots, i_m \in \{1, \ldots, n\}, \\ f \in \{=1, t\}, (i_k, i_{k+1}) \in G,\ \text{for } k \in \{1, \ldots, m-1\}\}$$

Notice, that number of components of cooperative grammar systems is not limited, so we can investigate extreme theoretical properties of these systems especially their generative capacity.

3 MAIN RESULTS

Many books concerning cooperative distributed grammar systems investigate their generative capacity (See [1]). Considered grammar systems consisted unlimited but finite number of components. This paper investigates generative capacity of cooperative distributed grammar systems with infinite components.

Theorem 3.1. For an infinite language L over finite alphabet T, there exists a cooperative distributed grammar system Γ with infinite number of components and context sensitive productions working in mode $= 1$, such that $L(\Gamma) = L$.

Proof. Let $L = \{w_1, w_2, \ldots\}$ be an infinite language. The cooperative distributed grammar system of infinite degree $\Gamma = (N, T, S, P_1, P_2, \ldots)$ is constructed:

- $N = \{S\}$,
- $P_i = \{S \rightarrow w_i\}$, for $i \in \{1, 2, \ldots\}$.

Every word from L can be derivated by one step derivation. E.g. word w_i, $i \geq 1$ is obtained by using component i and

$$S \Rightarrow_i w_i,$$

using production $S \rightarrow w_i$, so $w_i \in L(\Gamma)$ and $L = L(\Gamma)$. □

Notice that Γ consists only singleton components, every component has only one production. Cooperative distributed grammar systems with infinite components are more powerful, than Turing machines. This grammar system has disadvantage, set of all productions $\bigcup_{i=1}^{\infty} P_i$ of Γ is infinite.

Theorem 3.2. For an infinite language L over finite alphabet T, there exists a cooperative distributed grammar system Γ with infinite number of components and regular productions working in terminating mode, such that $L(\Gamma) = L$.

Proof. Let $L = \{w_1, w_2, \ldots\}$ be an infinite language and $n = max\{|w| \,:\, w \in L\}$. The cooperative distributed grammar system of infinite degree $\Gamma = (N, T, S, P_1, P_2, \ldots)$ is constructed:

- $N = \{S, X_1, X_2, \ldots, X_{n-1}\}$,
- $P_i = \{S \rightarrow w_i(1)X_1, X_1 \rightarrow w_i(2)X_2, \ldots, X_{|w_i|-2} \rightarrow w(|w_i|-1)X_{|w_i|-1}, X_{|w_i|-1} \rightarrow w_i(|w_i|)\}$, for $i \in \{1, 2, \ldots\}$.

If $\varepsilon \in L$, then for some $i \geq 1$, $P_i = \{S \rightarrow \varepsilon\}$. Every component generates exactly one word from L. Terminating mode forces cooperative distributed grammar system to derivate one word, e. g. derivation of word w_i from language L, using component i

$$S \Rightarrow_i w_i(1)X_1 \Rightarrow_1 w_i(1)w_i(2)X_2 \Rightarrow_i \ldots \Rightarrow_i w_i(1)w_i(2)\ldots w_i(|w_i|-1)X_{|w_i|-1} \Rightarrow_i$$

$$\Rightarrow_i w_i(1)w_i(2)\ldots w_i(|w_i|-1)w_i(|w_i|) = w_i$$

and $w_i \in L(\Gamma)$, so $L = L(\Gamma)$. □

Note that components are able to contain more than one regular production and set of all productions $\bigcup_{i=1}^{\infty} P_i$ of Γ is finite.

Theorem 3.3. For an infinite language L over a finite alphabet T, there exists a graph controlled cooperative distributed grammar system Γ with infinite number of components and regular productions working in mode $= 1$, such that $L(\Gamma) = L$.

Proof. Let $L = \{w_1, w_2, \ldots\}$ be an infinite language. The cooperative distributed grammar system of infinite degree $\Delta = (G, \Gamma)$ with $\Gamma = (N, T, S, P_1, P_2, \ldots)$ is constructed:

- $N = \{S, X\}$,
- For every $w_i \in L$. Let $n = 1$ for $i = 1$, and $n = 1 + \sum_{k=1}^{i-1} |w_k|$ for $i > 1$, $P_n = \{S \rightarrow w_i(1)X\}, P_{n+1} = \{X \rightarrow w_i(2)X\}, \ldots, P_{n+|w_i|-1} = \{X \rightarrow w_i(|w_i|-1)X\}, P_{n+|w_i|} = \{X \rightarrow w_i(|w_i|)\}$.
- For every $w_i \in L$. Let $n = 1$ for $i = 1$, and $n = 1 + \sum_{k=1}^{i-1} |w_k|$ for $i > 1$, $(n, n+1), (n+1, n+2), (n+2, n+3), \ldots, (n+|w_i|-1, n+|w_i|)$ add to G.

If $\varepsilon \in L$, then for some $i \geq 1$, $P_i = \{S \rightarrow \varepsilon\}$.

Only components with S on left-hand side of a rule can be used at the beginning of derivation. Thus only components P_n with $n = 1$ for $i = 1$, and $n = 1 + \sum_{k=1}^{i-1} |w_k|$ for $i > 1$, can be used. Then for every component there exist only one consequent component due control graph G so every derivation is of the form

$$S \Rightarrow_n w_i(1)X \Rightarrow_{n+1} w_i(1)w_i(2)X \Rightarrow_{n+2} \ldots \Rightarrow_{n+|w_i|-2} w_i(1)w_i(2)\ldots w_i(|w_i|-1)X \Rightarrow_{n+|w_i|-1}$$
$$\Rightarrow_{n+|w_i|-1} w_i(1)w_i(2)\ldots w_i(|w_i|-1)w_i(|w_i|) = w_i$$

and derivates word $w_i \in L$. Hence $L(\Delta) = L$. □

REFERENCES

[1] Rozenberg, G., Salomaa, A.: Handbook of formal languages, vol. 2: linear modeling: background and application, Springer, 1997, ISBN 3-540-60648-3

[2] Fernau, H., Holzer, M.: Graph-controlled cooperating distributed grammar systems with singleton components, Magdeburg, DE, J. Autom. Lang. Comb. 7, 2002, s. 487–503.

[3] Hopcroft, J. E., Ullman, J. D.: ntroduction to Automata Theory, Languages and Computation, Addison-Wesley, 1979, ISBN 0-201-029880-X.