

Context Free Grammar Representation by the Colored Petri Net

Inga Lisnic¹, Irina Cojuhari²

^{1,2}Department of Software Engineering and Automatics

Technical University of Moldova

Chisinau, Republic of Moldova

e-mail: inga.lisnic@ati.utm.md, irina.cojuhari@ati.utm.md

Abstract— Specification technique like Colored Petri Net (CPN) allows for the formal description and analysis of the discrete event systems. In this paper, it is presented the methodology for Context Free Grammar design, specification and analysis by the using Colored Petri Net technique. It is proposed the approach for representation of the Context Free Grammar by the Colored Petri Net language.

Keywords—Colored Petri Net; Context Free Grammar, discrete event systems

I. INTRODUCTION

Agent-based systems are one of the most important areas of research and development that have been developed in information technology for the last 30 years. The concept of multi-agent systems can be found in a range of domains and offers the strong models for representing complex and discrete event systems. The multi-agent systems can be described and modelled by the formalism that is used for design of the discrete-event systems as Finite Automata, Petri Nets etc.

Petri Nets is one of the powerful modeling formalism in computer science, system engineering and many other domains. Petri Nets combine a well defined mathematical theory with a graphical representation of the dynamic behavior of systems. The graphical representation of Petri nets enable visualization of the modeled system state changes [6].

In 1991 by the Jensen and Rozenberg, the Petri Nets was extended to the Colored Petri Net (CPN), that is a powerful and recognized modelling tool that can be seen as one of the most advanced behavior modeling techniques that is used to model and specify the behavior of concurrent and distributed systems [1].

An important topic in the field of multi-agent systems is the concept of ‘environment’. The environment in which a multiagent system is acted, is fundamental in the analysis, design and operation of the system. In the

literature, there are developed the environment based model for formalizing multi-agent systems by means of formal languages, where is specified the way of interaction of the agent with its environment by the formal language theory [3].

The field of formal language theory it was initiated by Noam Chomsky in the 1950s—provides a measuring stick for linguistic theories that sets a minimal limit of descriptive adequacy. Chomsky suggested a series of massive simplifications and abstractions to the empirical domain of natural language. The immense success of this framework—influencing not only linguistics to this day, but also theoretical computer science and, more recently, artificial intelligence domain—suggests that these abstractions can essentially improve the design of the multi-agent systems and the way of interaction of the agents with environment [2, 9-10, 12].

By the classical classification, that is called Chomsky hierarchy, there are types of languages: regular, context-free, context-sensitive and computably enumerable languages. The each type of language specification can be defined by the grammar.

A context-free grammar is a set of recursive rules used to generate patterns of strings, that is studied in fields of theoretical computer science, compiler design, and linguistics. It is a formal grammar which is used to generate all possible patterns of strings in a given formal language and can describe all regular languages and more [8,11], but it is a known, that context-free grammars are not able to cover all phenomena of natural and programming languages. Context-sensitive grammars are powerful enough, but have bad features in case of decidability problems. Therefore, the context free grammars has a wide applicability and it is a natural idea to introduce grammars which use context-free rules and have a device which controls the application of the rules [4].

In this paper, it is proposed a new approach for context free grammar design and analysis by the Colored Petri Net modeling formalism.

II. COLORED PETRI NET

The Petri nets are graphical and mathematical modeling tools that are applicable to many concurrent, asynchronous, distributed systems and have been used in the study of the formal languages design and analysis [7].

A Petri Net is a directed bipartite graph defined by the 4-tuple $(P, T, Pre, Post)$, where [1]:

- P is a finite set of places;
- T is a finite set of transitions $(P \cap T = \emptyset)$;
- Pre is the backward incidence application;
- $Post$ is the forward incidence application.

Simulating the Petri Nets involves moving tokens from place to place by firing transitions according to predefined rules. Colored Petri Net is an extension of Petri Nets for modeling complex systems that adds another dimension for tokens as well as to selection criteria used in determining firing by the addition of different token types. In this case, tokens are represented by the different functions.

Colored Petri Net is a graphical language for constructing models of concurrent systems and analyzing their properties, where each place has an associated type determining the kind of data that the place may contain.

A Colored Petri Net is defined by the 9-tuple [1]

$$CPN = (\Sigma, P, T, A, Nod, Col, Grd, Expr, Init), \quad (1)$$

where:

- Σ is a finite set on non-empty types;
- P is a finite set of places;
- T is a finite set of transitions;
- A is a finite set of arcs such that:

$$P \cap T = P \cap A = T \cap A = \emptyset. \quad (2)$$

- Nod is a node function. It is defined from A into

$$Nod : A \rightarrow (P \times T) \cup (T \times P). \quad (3)$$

- Col is a color function. It is defined from $P \cup T$ into Σ such that:

$$Col : P \rightarrow \Sigma. \quad (4)$$

- Grd is a guard function. It is defined from T into expressions such that:

$$\forall t \in T : [Type(Grd(t)) = Bool \wedge Type(Var(Grd(t))) \subseteq \Sigma]. \quad (5)$$

The guard function is a logical expression that must take the true logical value for the transition to take place.

- $Expr$ is an arc expression function. It is defined from A into expressions such that:

$$\forall a \in A : [Type(Expr(a)) = Col(p(a))_{MS} \wedge Type(Var(Expr(a))) \subseteq \Sigma], \quad (6)$$

where $p(a)$ is the place of $N(a)$;

- $Init$ is an initialization function. It is defined from P into closed expressions such that:

$$\forall p \in P : [Type(Init(p)) = Col(p)_{MS}]. \quad (7)$$

$Type(v)$ – it is used for specification of the variable types, $Type(expr)$ – is used for specification of the expressions and $Var(expr)$ for the set of expressions' variables.

The syntactical elements of the Colored Petri Net are essentially places, transitions, arcs and inscriptions. The elements of the CPN are presented in the Fig. 1.

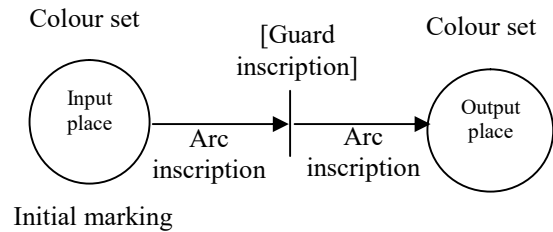


Figure 1. The elements of the Colored Petri Net [5].

The collection of tokens on the places determinate at any time the state of the discrete-event system. The arcs represent the relation between places and transactions and determinate a way how a state changes when an event occurs. The inscription above arcs are the functions, that determinate the actions and quantity of tokens moved between places. The transitions represent the possible actions and the guard function introduce the restriction to the possible actions that can occurs. The colored sets are the types of tokens attached to the respectively places and initial marking set the initial configuration of the Colored Petri Net [5].

III. CONTEXT FREE GRAMMAR

A grammar by the Chomsky classification is defined as a quadruple [12]

$$G = (V, \Sigma, P, S), \quad (8)$$

where:

- V - is a finite set of non-terminal symbols;
- Σ - is a finite set of terminal symbols;

$$V_N \cap V_T = \emptyset,$$

- $S \in N$ is a start symbol;
- P – is a finite set of productions of rules:

$$P \subseteq (V \cup \Sigma)^* V (V \cup \Sigma)^* \times (V \cup \Sigma)^* \quad (9)$$

The families of languages generated by context-sensitive, context-free, linear and regular grammars are denoted by CS, CF, LIN and REG, respectively. The next strict inclusions, named Chomsky hierarchy, hold [7, 9-10]:

$$\text{REG} \subset \text{LIN} \subset \text{CF} \subset \text{CS} \subset \text{RE}.$$

Usually, the rules $(u, v) \in P$ are written in the form $u \rightarrow v$, and the rule $u \rightarrow \varepsilon$ is called the erasing rule.

Let $G = (V_N, V_T, P, S)$ is a grammar. The language generated by the grammar G is denoted by $L(G)$ and represents the set of all strings of terminals that are derivable from the starting state S :

$$L(G) = \{x \mid S \rightarrow^* x, x \in \Sigma^*\}. \quad (10)$$

A string $x \in (V \cup \Sigma)^*$ directly derives a string $y \in (V \cup \Sigma)^*$ in grammar G , written as $x \Rightarrow y$ if there is a rule $u \rightarrow v \in P$, such that $x = x_1 u x_2$ and $y = x_1 v x_2$ for some $x_1, x_2 \in (V \cup \Sigma)^*$.

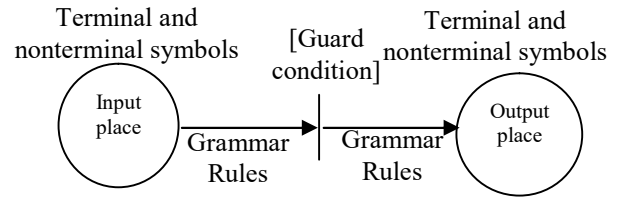
A grammar $G = (V_N, V_T, P, S)$ is called a context free grammar if each rule

$$U \rightarrow v \in P \text{ has } u \in V \text{ and } v \in (V \cup \Sigma)^*. \quad (13)$$

VI. CONTEXT FREE GRAMMAR REPRESENTATION BY THE COLORED PETRI NET

Forward reachability in Colored Petri Net, is a concept that a given marking M_f is reachable from M_0 , where M_0 is initial marking. The backward reachability, means that if a marking M_f is reachable from M_0 , M_0 is backward reachable from M_f . To perform the backward and forward reachability analysis, it is done based on the defined CPN structure and given transformation rules according to the given guard functions and transition functions [1].

The representation of the context free grammar by the Coloured Petri Net is given in the Fig. 2.



Initial marking

Figure 2. The representation of the context free grammar by the Colored Petri Net.

The construction of the Colored Petri Net presented in the Fig. 2 is based on the idea of using similarity between the firing of a transition based on function attached to each arc and guard condition and with the application of a production rule in a derivation in which tokens represent nonterminal and terminal symbols.

A context-free Colored Petri Net is defined as the 10-tuple

$$\text{CPN} = (\Sigma, P, T, A, \text{Nod}, \text{Col}, \text{Grd}, \text{Expr}, \text{Init}), \quad (14)$$

where :

- Σ is a finite set of terminal symbols;
- P is a finite set of places;
- T is a finite set of transitions;
- A is a finite set of arcs such that:

$$P \cap T = P \cap A = T \cap A = \emptyset. \quad (15)$$

- Nod is a node function. It is defined from A into

$$\text{Nod} : A \rightarrow (P \times T) \cup (T \times P). \quad (16)$$

- Col is a color function. It is defined from $P \cup T$ into Σ such that:

$$\text{Col} : P \rightarrow (V \cup \Sigma)^*, \quad (17)$$

where V is a set of nonterminal symbols.

Σ is a set of terminal symbols.

- Grd is a guard function. It is defined from T into expressions such that:

$$\forall t \in T : [\text{Type}(G(t)) = \text{Bool} \wedge \text{Type}(\text{Var}(G(t))) \subseteq (V \cup \Sigma)^*]; \quad (18)$$

The guard function is a logical expression that must take the true logical value for the transition to take place.

- Expr is an arc expression function. It is defined from A into expressions such that:

$$\forall a \in A: [\text{Type}(\text{Expr}(a)) = \text{Col}(p(a))_{\text{MS}} \wedge \text{Type}(\text{Var}(\text{Expr}(a))) \subseteq (V \cup \Sigma)^*], \quad (19)$$

where $p(a)$ is the place of $N(a)$;

- Init is an initialization function. It is defined from P into closed expressions such that:

$$\forall p \in P : [\text{Type}(\text{Init}(p)) = \text{Col}(p)_{\text{MS}}]. \quad (20)$$

V. CONCLUSIONS

In this paper, it is proposed a new approach for context free grammar design and analysis by the Colored Petri Net formalism. Colored Petri Net is a powerful modelling tool, that can be seen as one of the most advanced behavior modeling techniques, that is used to model and specify the behavior of concurrent and distributed discrete event systems. This approach can be easily used for modeling and design the agent-based systems.

Grammar systems can be considered as a formal model for a phenomenon of solving a given problem by dividing it into sub problems (grammars) to be solved by several parts. The control of rules in grammar systems also allows increasing computational power grammar systems, but the context free grammar have the limitation in description of the concurrent or synchronous event and in these way the Colored Petri Net introduce the parallel firing strategy, which can define the concurrently parallelism situations in formal language theory.

ACKNOWLEDGMENT

This work was supported by the project 20.80009.5007.09 "Elaboration and launches of the nanosatellites series with research missions from the

International Space Station, the monitoring and its post operation, promotion of space technologies".

REFERENCES

- [1] M. Bouali, P. Barger, W. Schon, "Colored Petri Net inversion for Backward Reachability Analysis," Second IFAC Workshop on Dependable Control of Discrete Systems, Bari : Italy, 2009, pp. 1-6.
- [2] G. Jager, J. Rogers, "Formal language theory: refining the Chomsky hierarchy," Philosophical Transactions of the Royal Society B: Biological Sciences, 367(1598), 2012, pp. 1956–1970. doi:10.1098/rstb.2012.0077.
- [3] Gemma Bel-Enguix, M. Dolores Jimenez-Lopez, "Agent-Systems and Formal Languages," In Proceedings of 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 2011, pp. 151 -154.
- [4] Jurgen Dassow, Sherzod Turaev, "Petri Net Controlled Grammars: the Power of Labeling and Final Markings," Romanian Journal of Information Science and Technology Volume 12, Number 2, 2009, pp. 191–207.
- [5] Kjeld Høyer Mortensen, *Coloured Petri Nets - a Pragmatic Formal Method for Designing and Analysing Distributed Systems*, DAIMI Report Series, 26(522). <https://doi.org/10.7146/dpb.v26i522.7051>
- [6] Wang Jiacun, *Petri Nets for Dynamic Event-Driven System Modeling*. Handbook of Dynamic System Modeling, chapter 24, 2007, pp. 1-17.
- [7] J. Dassow, G. Mavrankulov, M. Othman, S. Turaev, M.H. Selamat and R. Stiebe, "Grammars Controlled by Petri Nets," Petri In Manufacturing and Computer Science, 2012, doi:10.5772/50637
- [8] M. Al-A'ali, A. Khan, N. Al-Shamlan, "Simplification of context-free grammar through Petri net," Computers and Structures, 1996, pp. 1055–1058.
- [9] N. Chomsky, "Three models for the description of languages," *IRE Trans. on Information Theory* 2(3), 1956, pp. 113–124.
- [10] N. Chomsky, "On certain formal properties of grammars," *Information and Control*, 1959, pp. 137–167.
- [11] P. Darondeau, "On the Petri net realization of context-free graphs," *Theor. Computer Sci.*, 2001, pp. 573–598.
- [12] J. Hopcroft, J. Ullman, *Introduction to automata theory, languages, and computation*. Addison-Wesley Longman Publishing Co., Inc., 1990.