Development of timed RT-grammars for controlling processes in cyber-physical systems

N. Voit

Ulyanovsk state technical university, Ulyanovsk, Russia
e-mail: n.voit@ulstu.ru

Abstract. Author defines timed RT-grammars with infinite size, and represents a practical application of RT-grammars to the well-known generalized railroad crossing problem.

1 Introduction

Applying timed automaton for designing, specification, controlling and analyzing real-time systems is well-known practice [1]. Timed and hybrid automatons is used for analyzing and managing cyber-physical systems (CPS) [2].

When solving the tasks of CPS's design, specification, control and analysis, there are problems with access to resources, blocking, liveliness limitation (liveness, reversibility, boundedness, reachability, dead transitions, deadlocks, home states). The examples of CPS's tasks are the control of the nuclear reactor's temperature and the control of the railway level-crossing gate [3]. These tasks successfully apply timed context-free grammars. A large number of interacting CPS sets the task of CPS's monitoring and analyzing. It can be accomplished by various methods. At present, π - calculus is a promising but very young and evolving theory. It has many open questions and unresolved problems. Petri nets which are widely used do not have a universal framework for CPS's modeling and analyzing. In order to analyze various properties (liveliness, attainability, safety), CPSs are modeled in different types of Petri nets. In order to analyze CPS in the error-free systems' development in the conceptual design phase, the model checking method is widely used. However, it is mainly developed for experienced scientists and engineers, since it is complex to understand and use [4]. CPSs are also specified by managers who do not have training in formal models and informatics. Formal analysis requires a detailed representation of the process model in a formal language.

Although, relevant and having big practice famous is a problem researching mechanisms to analyze and control CPS. Authors offer that mechanism based on automaton RT-grammars.

2 Related works

Author researchs some works which consider CPSs, ω-grammars, timed grammars, timed automaton, the model checking technique. Several works have focused on specifying formal semantics and verification methods for workflows using Petri nets, process algebra, abstract state machine, see, e.g., [5-9].

Neda Saeedloei and Gopal Gupta [8] define timed grammars based on ω-grammars to analyse CPSs. In [9], they offer a logic programming framework for verifying complex continuous real-time systems, based on pushdown timed automata and timed automata, but this approche is very difficult to use in applications.

In [4], Yuan Wang, Yushun Fan offer a method for modeling and analyzing workflows based on Temporal Logic of Actions, and present the semantic of workflow process using TLA and illustrate how to modeling and analysis workflow using TLA. However, they consider only a few patterns, and this work is not theoretically grounded.

John L. Pfaltz and Azriel Rosenfeld have proposed two-dimensional pattern generating web-grammars [10]. The proposals generated by web-grammars are directed graphs with symbols at their vertices ("webs"). Zhang has offered the positional graphical grammar, later it has been developed by Costagliola [11, 12]. It is related to the context-free grammars. Wittenberg and Weitzman [13] have developed a relational graphical grammar. Zhang and Orgun [14] have described the preserving graphical grammars in their studies.

The mentioned methods have the following shortcomings.

1. Positional grammars developing on the base of plex-structure don't use touchpoints and can't be used for languages the objects of which have dynamically changeable number of inputs\outputs.

2. The relational grammars have poorly implemented the neutralization tools for generating a complete list of errors.

3. There is no semantic checking of text diagrams.

The common shortcomings of the above-mentioned grammars are: when designing the grammar for unstructured visual languages the number of rules is increased (if there is no change in the number of the language primitives for describing the all variants without structure, the significant quantity of rules will increase), the complexity of developing the grammar. Furthermore, it takes much time to analyze diagrams (analyzers based on the considered grammars have polynomial or exponential time of analysis).

3 Definition timed RT-grammars

The automaton timed RT-grammar is a R-grammar extension [15-19]. It is represented by six following components:

$$G = (V, \sum, C, E, R, r_0), \quad (1)$$

where $V = \{v_e,\ e = \overline{1,H}\}$ is an auxiliary alphabet (the alphabet of operations over internal memory); $\sum = \{a_l,\ l = \overline{1,L}\}$ is a terminal alphabet of a language; C –is a finite set of clock identifiers; $E -$ is a set of time expressions C (clock limitation and clock reset), is limited by the following expressions: from the beginning $\{c := 0\}$ and onwards $\{c \sim x\}$, and c is a variable, and x is a constant, $\sim \in \{=, <, \leq, >, \geq\}$; $R = \{r_i,\ i = \overline{0, I}\}$ is grammar scheme *G* (a set of names of production rules' complexes, each complex r_i consists of subset P_{ij} of production rules $r_i = \{P_{ij},\ j = \overline{1, J}\}$); $r_0 \in R$ is a RT-grammar axiom (the name of the initial complex of production rules), $r_k \in R$ is a conclusive complex of production rules. Production rule $P_{ij} \in r_i$ is given as $P_{ij}: a_l \xrightarrow[E]{W_v(\gamma_1, \dots, \gamma_n)} r_m$, where $W_v(\gamma_1, \dots, \gamma_n)$ is a *n*-th relation, which determines the type of operation over the internal memory depending on $v \in \{0,1,2\}$ (respectively, 0 – operation is not performed, 1 – record, 2 – read); $\gamma_1, \dots, \gamma_n \in V$; $r_i \in R$ is the name of the complex of a production rule's source; $r_m \in R$ is the name of the complex of a production rule's successor.

Example 1. The authors dealt with a language in which each sequence a is accompanied by the same number of sequence b, and there are at least two a and two b. For each pair of sequences a and b, the first symbol b should appear within 5 units of time from the first symbol a, and the final symbol b should appear within 20 units of time from the first character a. The context-free timed grammar of this language [8] is defined as:

$$\begin{aligned} &S \rightarrow R\,S, \\ &R \rightarrow a\,\{c := 0\}T\,b\,\{c < 20\}, \\ &T \rightarrow a\,T\,b, \\ &T \rightarrow a\,b\,\{c < 5\}. \end{aligned}$$

The authors gave a definition of this language:

$$L = \{a^n(b, c < 5)(b^m, c < 20) | m > 0, n = m + 1\} \tag{2}$$

Graphical RT-grammar representation (as a stack-like internal memory automata) analyzing the language (3) is shown in Figure 1.

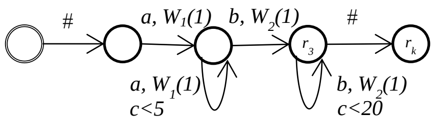

Figure 1. The timed stack memory RT-automata

Example 2. A classic example of a generalized railroad crossing problem [1] can be described by a timed automaton (Figure 2).

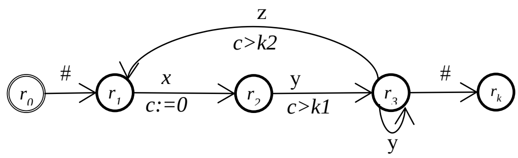

Figure 2. The timed automaton that formalizes a generalized railroad crossing problem

The train's states (locations) r_0 and r_k are axioms. In location r_1 the train is approaching to the railroad track crossing; the clock is set to zero. The location r_2 is before the railroad track crossing, so the railroad track is open only for the train located in this location; moreover, other railroad tracks are closed after time k_1. Then the train is in location r_3 up to time k_2, after that it leaves the location r3, and this track is open for the next train.

The context-free grammar [8] is given as:

$$S \rightarrow R\,S,$$
$$R \rightarrow x\,\{c := 0\}\,T\,z\,\{c > k_2\},$$
$$T \rightarrow y(c > k_1)T,$$
$$T \rightarrow yT,$$
$$T \rightarrow y.$$

A timed context-free language describing the cyber-physical system of railroad track crossing (see Fig.2) is defined as:

$$L = \{(x, c \coloneqq 0)(y, c > k_1)y^n(z, c > k_2) | n, k_1, k_2 > 0\} \quad (3)$$

7 Conclusion

The advantage of timed RT-grammars compared with LR(k)- and LL(k)-methods is a linear time-consumption for input language strings analysis with complete control preservation.

In the future works, the authors are going to deal with timed grammars for the analysis, control and translation of dynamic project flows' problems, in which the time factor occupies a significant place. Use of such grammars will eliminate a number of semantic errors at the stage of conceptual design of complex computer systems.

Acknowledgment

This research is supported by the grant of the Ministry of Education and Science of the Russian Federation, the project № 2.1615.2017/4.6. The reported study was funded by RFBR and Government of Ulyanovsk Region according to the research project № 16-47-732152. The reported study was funded by RFBR according to the research project № 17-07-01417.

References:

1. C. L. Heitmeyer, and N. A. Lynch, "The generalized railroad crossing: A case study in formal verification of real-time systems," IEEE RTSS, pp. 120-131 (1994).
2. Yu. G. Karpov, "MODEL CHECKING. Verifikaciya parallel'nyh i raspredelennyh programmnyh system," SPb.: BHV-Peterburg, 560 s (2010) (in Russian)
3. A. Lee Edward, "Cyber-physical systems: Design challenges," ISORC (2008)
4. Yuan Wang, Yushun Fan, "Using Temporal Logics for Modeling and Analysis of Workflows," Proceedings of E-Commerce Technology for Dynamic E-Business, IEEE International Conference on. DOI: 10.1109/CEC-EAST.2004.72 (2004)
5. Rajeev Alur and David L. Dill. Automata for modeling real-time systems. In ICALP, Lecture Notes in Computer Science, vol. 443, pp. 322-335. Springer (1990)
6. Rajeev Alur and David L. Dill. A theory of timed automata. Theoretical Computer Science, 126:183-235 (1994)
7. Rajesh Gupta. Programming models and methods for spatiotemporal actions and reasoning in cyber-physical systems. In NSF Workshop on CPS (2006)
8. Neda Saeedloei and Gopal Gupta Timed definite clause ω-grammars // Technical Communications of the International Conference on Logic Programming (Edinburgh), pp. 212–221 http://www.floc-conference.org/ICLP-home.html (2010)
9. Neda Saeedloei, Gopal Gupta, "A Logic Programming Realization of Timed Context-Free Grammars," Proceedings of the ICLP 2010, Pages 212-221 (2010)
10. K. Fu, Structural methods of pattern recognition. Moscow: Mir (1977)
11. G. Costagliola, A.D. Lucia, S. Orece, G. Tortora, "A parsing methodology for the implementation of visual systems,". (http://www.dmi.unisa.it/people/costagliola/www/home/papers/method.ps.gz.)
12. D. Q. Zhang, K. Zhang, "Reserved graph grammar: A specification tool for diagrammatic VPLs," IEEE Symposium on Visual Languages, pp. 284-291 (1997)
13. K. Wittenburg, L. Weitzman, "Relational grammars: Theory and practice in a visual language interface for process modeling,". http://citeseer.ist.psu.edu/wittenburg96relational.html (1996)
14. K. B. Zhang, K. Zhang, M. A. Orgun, "Using Graph Grammar to Implement Global Layout for A Visual Programming Language Generation System," (2002)
15. Alexander Afanasyev, Nikolay Voit, "Intelligent Agent System to Analysis Manufacturing Process Models," Proceedings of the First International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'16) vol.451 of the series Advances in Intelligent Systems and Computing. Russia, pp. 395-403 (2016)
16. Alexander Afanasyev, Nikolay Voit, Rinat Gaynullin, "The Analysis of Diagrammatic Models of Workflows in Design of the Complex Automated Systems," Proceedings of the First International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'16) vol. 450 of the series Advances in Intelligent Systems and Computing. Russia, pp. 227-236 (2016)
17. A.N. Afanasyev, N.N. Voit, R.F. Gainullin, "Diagrammatic models processing in designing the complex automated systems," 10th IEEE International Conference on Application of Information and Communication Technologies (AICT). Baku, Azerbaijan, pp. 441-445 (2016)
18. A.N. Afanasyev, N.N. Voit, E.Yu. Voevodin, R.F. Gainullin, "Control of UML diagrams in designing automated systems software," Proceedings of the 9th IEEE International conference on Application of Information and Communication Technologies: AICT – 2015, pp. 285-288 (2015)
19. A.N. Afanasev, N.N. Voit, E.Yu. Voevodin, R.F. Gainullin, "Analysis of Diagrammatic Models in the Design of Automated Software Systems," Object Systems – 2015: Proceedings of X International Theoretical and Practical Conference (Rostov-on-Don, 10-12 May, 2015) / Edited by Pavel P. Oleynik. – Russia, Rostov-on-Don: SI (b) SRSPU (NPI), pp. 124-129 (2015)