http://dx.doi.org/10.5755/j01.eee.19.1.3261 ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 19, NO. 1, 2013

Piece Linear-Aggregate Approach for Modelling and Analysis of Fuzzy Systems

H. Pranevicius[1,2], T. Kraujalis[2]
[1]*Department of Business Informatics, Kaunas University of Technology,*
Studentu St. 56–301, LT-51424 Kaunas, Lithuania, phone: +370 37 300376
[2]*Department of Applied Informatics, Vytautas Magnus University,*
Vileikos St. 8-409, LT–44404 Kaunas, Lithuania, phone: +370 37 327900
t.kraujalis@if.vdu.lt

***Abstract*—In this paper, we use a PLA model for performance and behaviour analysis of fuzzy system. This model makes connection between fuzzy logic and formalism. A case study contains an illustration how the proposed model can be fruitfully exploited to model traffic control systems based on fuzzy logic. Piece-Linear Aggregate model for traffic signal control system has been transformed into timed automaton for verification of safety, liveness, bounded- liveness and deadlock-freeness properties based on model checking. The system performance analysis was performed using Arena software package. A comparative analysis of traffic light controllers with fixed time and fuzzy logic algorithms is given.**

***Index Terms*—Fuzzy systems, formal languages, formal verification, computer simulation, traffic control, performance analysis.**

I. INTRODUCTION

Formal methods are widely used for modelling and verification of complex systems.

By the increasing interest in a fuzzy logic a lot of scientists were interested to use formal methods to model fuzzy systems. Various methods are being used for modelling fuzzy systems, such as: DEVS [1], [2]–[8] Petri-nets [3], Process algebra [4] and others. We think the researchers that use formal methods to model fuzzy systems pay not enough attention for the verification of these models. For formal modelling of fuzzy system we used piece linear- aggregate (PLA) formalism [9], which allows on the base of single formal description of system to create models for performance and behaviour analysis. The PLA formalism has been already applied to modelling computer network protocols, traffic systems [5], [10], medical applications [6], variable structure system, hybrid system [6], etc.

In this paper, we propose a general PLA model for modelling fuzzy systems. This model was used for performance and behaviour analysis of fuzzy traffic signal control system. For behaviour analysis the PLA model have been transformed to timed automata in order to verify the safety and liveness properties of a system using the UPPAAL model-checking tools.

II. PLA BASED MODEL OF FUZZY SYSTEMS.

PLA formalism can be defined like a universal and general methodology that provides tools to simulate and verify systems which behaviour is based on discrete event [6]. In the application of the aggregate approach for system modelling, the system is viewed as a whole of interacting aggregates. Each aggregate is represented as an object defined by a set of input signals X, output signals Y, events E and states Z.

A fuzzy logic systems (FLS) consists of four main parts: fuzzifier, rules, inference engine, and defuzzifier. These components and the general architecture of a FLS are shown in Fig. 1. In the first stage, the input parameters are fed to the fuzzification part to determine the degree of membership of crisp inputs in appropriate fuzzy sets. Following, the fuzzified input data are entered into the inference where the most appropriate rules are selected from the fuzzy rule base. Finally, the resulting fuzzy output is mapped to a crisp output in the defuzzification step.

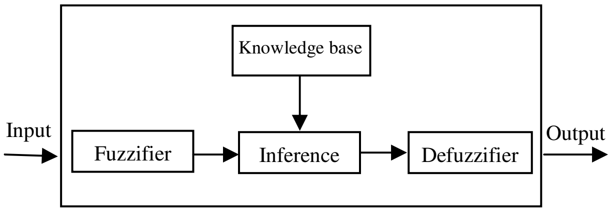

Fig. 1. Fuzzy logic system.

Using the scheme defined in Fig. 1 we created PLA model for modelling fuzzy systems. In PLA approach the fuzzy logic system aggregate is composed of several sub-aggregates (*Distributor*, *Fuzzifiers*, *Connector*, *SMF* and *Deffuzifier*) interacting with each other in order to implement a fuzzy system. Fig. 2 shows the aggregation model of fuzzy logic system. In aggregates Distributor, *Fuzzifier* and *SMF* there are no time management when input is received to the aggregate, it immediately generates an output. The model is completely generic, but depending on the application, the aggregates could be redefined.

A *Fuzzifier* transforms crisp values into membership grades of fuzzy sets. Formal specification of this aggregate in PLA formalism is presented below:

1. The set of input event $X = \{x\}, x \in \mathbb{R}$*;*
2. The set of output event $Y = \{y\},\ y \in [0,1]$*;*

Manuscript received March 19, 2012; accepted May 22, 2012.

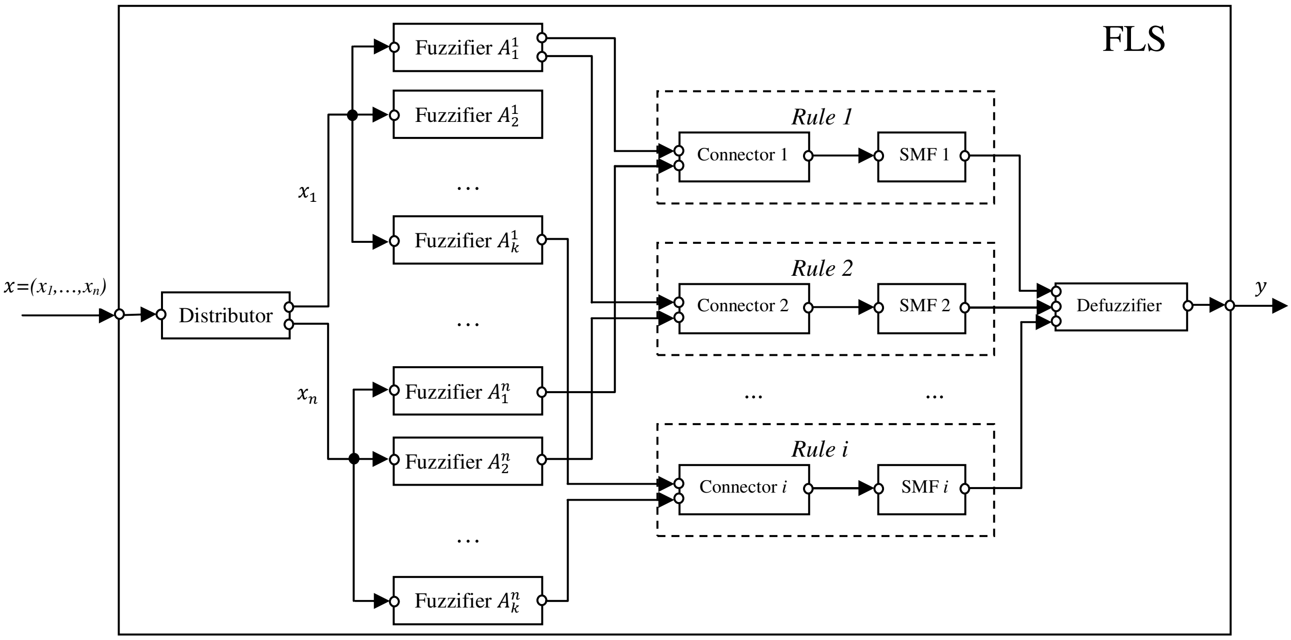

Fig. 2. The aggregation model of fuzzy logic system.

3. *The set of external event* $E' = \{e_1'\}$*;*
4. *The set of internal event* $E'' = \emptyset$*;*
5. *Discrete component of state* $\nu(t_m) = \emptyset$*;*
6. *Comtinues component* $z_\nu(t_m) = \emptyset$*;*
7. *Parametes. Membership function* $\mu: X \rightarrow [0,1]$*;*
8. *Output operators* $G(e_1')$: $y = \mu(x)$.

III. PLA MODEL OF THE TRAFFIC SIGNAL CONTROL SYSTEM

Case study demonstrates the application of PLA model, presented above, for modelling traffic signal control system.

We created a traffic signal control model for three- sided intersection, the exhaustive description could be found in [10]. The traffic flows at three-sided intersection is regulated by three-phase traffic signals, indicating separate signal phases for westbound, south- west, and east-south traffic. The lights are operated by a controller which is able to control traffic flows adaptively based on traffic conditions. The control device controlling algorithm is based on fuzzy logic, which decides, whether to stay in current green phase or move to the next.

The PLA model of fuzzy traffic signal control system based on fuzzy logic is presented in Fig. 3. The system consists of three aggregates: Intersection, TLC (traffic light controller) and FLC (fuzzy logic controller).

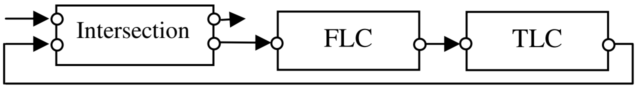

Fig. 3. The PLA model of traffic signal control system.

An input signal arrived from the aggregate *TLC* to aggregate *Intersection* causes an external transition and generates the output to the aggregate *FLC*. The output signal consists of the following attributes:

1) *Average number of vehicles in the lanes of current green;*
2) *Arrival rate of current green phase;*
3) *Average number of vehicles in the lanes of next green.*

Other output signal of this aggregate is the number of vehicles that passed the intersection during the green light.

The *FLC* receives signal from aggregate *Intersection* and produce the information for *TLC* to which state to transit after internal transition. When an internal transition occurs the *TLC* changes its state and produces an output signal to an aggregate *Intersection*. Fig. 4 shows a state graph for the aggregate *TLC*. Formal specification of aggregate *TLC* is presented below:

a) $X = \{x\}$, $x \in \{0,1\}$ – *received signal from FLC. If the value 0- the aggregate changes its state to the next, otherwise transit to the previous one;*
b) $Y = \{y\}$, *y- output signal to aggregate Intersection, indicating which phase was extended or terminated;*
c) $E' = \{e_1'\}$, e_1' *- input signal from aggregate FLC;*
d) $E'' = \{e_j''\}$, e_j'' *j- th phase has expired;* $j = \overline{1,3}$*;*
e) $e_j'' \mapsto \xi$, $\xi \in \{6,10\}$ – *activation time of the j- th phase;*
f) $\nu(t_m) = \{Ph(t_m)\}$; $Ph(t_m) \in \{1,2,3,4,5,6\}$,
g) *where 1 – first phase is active; 2 – waiting an input signal from aggregate FLC;3 – second phase is active;*
h) *Waiting an input signal from aggregate FLC;*
i) $z_\nu(t_m) = \{w(e_j'', t_m)\}, j = \overline{1,3}$ *time duration of j-th phase will be expired;*
j) $z(0) = \{1, \Delta t_1, \infty, \infty\}$, *initial state;*
k) $H(e_j'')$ *: /j- th phase has expired;* $j = \overline{1,3}$*/;*
   $Ph(t_{m+1}) = Ph(t_m) + 1$;
   $w(e_j'', t_{m+1}) = \infty$.

$G(e_j'')$: /Output signal is generated to aggregate *Intersection*/

$$y = k\ ,\ k = \begin{cases} 1, if\ Ph(t_{m+1}) = 1; \\ 2, if\ Ph(t_{m+1}) = 3; \\ 3, if\ Ph(t_{m+1}) = 5. \end{cases} \qquad (1)$$

$H(e_1')$: /Input signal is received from aggregate *FLC*/:

$$Ph(t_{m+1}) = \begin{cases} Ph(t_m) - 1 & , if\ x_1 = 1; \\ Ph(t_m) + 1, & if\ x_1 = 0\ \wedge Ph(t_m) < 6; \\ 1, & if\ x_1 = 0\ \wedge Ph(t_m) = 6\,. \end{cases} \quad (2)$$

$$w(e_k^{"}, t_{m+1}) = t_m + \Delta t\ ,\ k = \begin{cases} 1, if\ Ph(t_{m+1}) = 1; \\ 2, if\ Ph(t_{m+1}) = 3; \\ 3, if\ Ph(t_{m+1}) = 5. \end{cases} \quad (3)$$

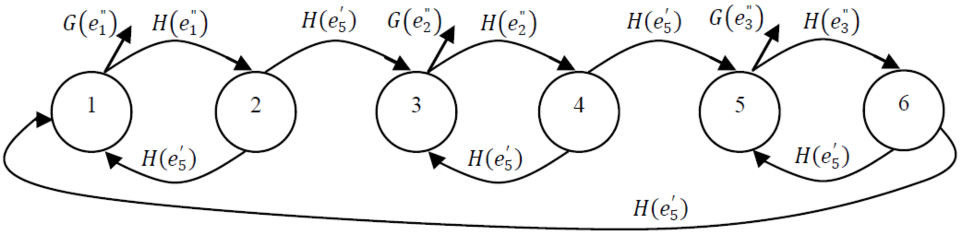

Fig. 4. State graph of the aggregate *TLC*.

IV. VERIFICATION OF FUZZY TRAFFIC SIGNAL CONTROL SYSTEM

The traffic signal light controller has been verified with UPPAAL model checker [7], which is widely used for modelling, validation and verification of real-time systems. It is an integrated tool environment consisting of a graphical user interface, which allows system descriptions to be defined graphically as networks of timed automata, and a model checking engine that verify properties. It can check for invariant, reachability, and liveness properties of the system expressed in the UPPAAL Requirement Specification Language. This language is a subset of Timed Computational Tree Logic (TCTL). Table I specifies the queries available in the verifier. Because in this verifier the systems are modelled using timed automata, to this purpose the formal traffic light control model specification written in PLA was transformed to timed automata.

Table I. UPPAAL Property Specification.

Syntax	Name	Property	Meaning
$A[]p$	Invariantly	Safety	For all paths p always holds
$A <> p$	Eventually	Liveness	For all paths p will eventually hold
$E <> p$	Possibly	Reachability	There exists a path where p eventually holds
$E[]p$	Potentially always	Safety	There exists a path where p always holds
$p --> $	Leads to	Liveness	Whenever p holds q will eventually hold

The analyzed system in UPPAL environment consists of the following timed automata: queues (Q1, Q2 and Q3), traffic lights (TL1, TL2 and TL3) and traffic light controller (TLC). The state graph illustrated in figure 4 for an aggregate *TLC* is the same and for *TLC* timed automata.

A safety, liveness and bounded- liveness properties of TLC were checked during verification. The verified safety property states that whenever the traffic is allowed in one direction (the light is green), the light is red on the other directions. To prove that a system satisfies a safety property, we used an invariant. In order to define system invariant three symbolic states were identified, using the predicates P_1, P_2, P_3. In order to define this predicates the following auxiliary predicates were used:

1) $T_1 : TL_1(t_m) = Green$*- light for the first approach is green;*
2) $T_2 : TL_2(t_m) = Green$*-light for the second approach is green;*
3) $T_3 : TL_3(t_m) = Green$*- light for the third approach is green;*
4) $E_1 : w(e_1^{"}, t_m) \neq \infty$ *- phase1 is active;*
5) $E_2 : w(e_2^{"}, t_m) \neq \infty$ *- phase2 is active;*
6) $E_3 : w(e_3^{"}, t_m) \neq \infty$ *- phase3 is active.*

Using the defined auxiliary predicates we can formulate predicates P_1, P_2, P_3 for symbolic states:

1) $P_1 = T_1 \wedge \overline{T_2} \wedge \overline{T_3} \wedge E_1 \wedge \overline{E_2} \wedge \overline{E_3}$ *;*
2) $P_2 = \overline{T_1} \wedge T_2 \wedge \overline{T_3} \wedge \overline{E_1} \wedge E_2 \wedge \overline{E_3}$*;*
3) $P_3 = \overline{T_1} \wedge \overline{T_2} \wedge T_3 \wedge \overline{E_1} \wedge \overline{E_2} \wedge E_3$.

The expression P_1 means that the traffic lights in the first phase must be as follows: the light for the first approach is green, for second and third approaches the lights are red. P_2 and P_3 are interpreted similarly. Thus, the system invariant is: $I = P_1 \vee P_2 \vee P_3$

One liveness and three bounded liveness properties were checked. These properties written in timed computational tree logic are presented in Table II:

1) Phases liveness. Whenever traffic light controller activates Phase1, it will eventually activate Phase2 and Phase3. We check similar properties for the rest phases.
2) Minimum duration of the phases. The duration of the phase must be more than or equal to 10 seconds.
3) Maximum duration of the phases. Depending on traffic flows, the duration of each phase can be extended till 40 seconds.
4) Traffic lights cycle time. The duration interval of the phase varies from 10 to 40 seconds, the length of the traffic lights cycle varies too, and its interval ranges between 30 to 120 seconds.

TABLE II. TCTL FORMULAS FOR ANALYSIS OF TLC SYSTEM.

Query	Property
TLC.Phase1--> TLC.Phase2 && TLC.Phase3	Liveness (1)
TLC.Phase1--> TLC.Phase2 && TLC.y >= 10:	Bounded liveness(2)
TLC.Phase1--> TLC.Phase2 && TLC.y <= 40:	Bounded liveness(3)
TLC.Phase1--> Phase3 && (TLC.y<=120 || *TLC.y>=30)*	Bounded liveness(4)

Finally, the model checker confirmed that the model has no deadlocks in any points during a runtime.

V. SIMULATION OF FUZZY TRAFFIC SIGNAL CONTROL SYSTEM

In this part a performance analysis of the fuzzy logic traffic lights control system was accomplished. A virtual environment for the junction based on the proposed modelling methodology was built using simulation software

“Arena”. Arena is a discrete event simulation software, which uses the SIMAN processor and simulation language. A comparative analysis between fuzzy logic control method and the fixed time controller has been made in traffic flows control under the same conditions without pedestrian crossing. The geometry of the intersection is illustrated in Fig. 5. Fig. 6 presents the phase order of the intersection model. The cycle sequence is 1–2–3–1– ....

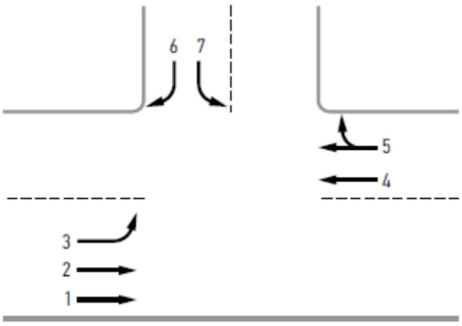

Fig. 5. The geometry of simulated intersection.

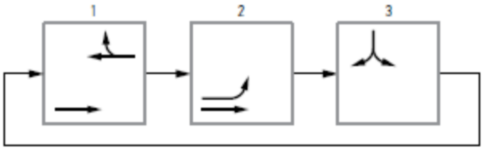

Fig. 6. Phase sequence of the tested intersection.

During the simulation, arrival rate of vehicles in the lane varied between 0.08 and 0.19 vehicles per second (288-680 veh/hr/lane). The criterion used for the evaluation is the average stopped delay, i.e. the delay which occurs when a vehicle is fully immobilized. The best control strategy is the one that provides the lowest delays. The results of the system performance are shown in Fig. 7.

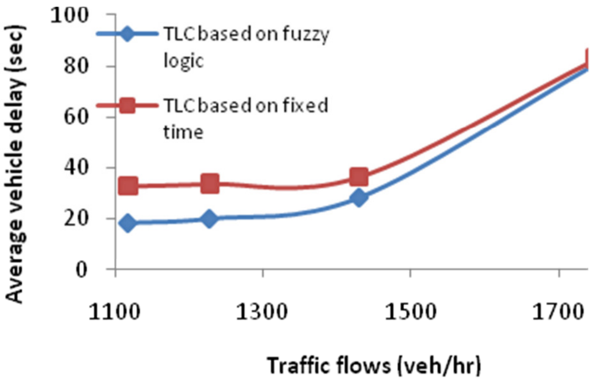

Fig. 7. Simulation results for different control algorithms.

During low and medium traffic flows the TLC based on fuzzy logic produced lower average stopped delay than fixed time technique. The fixed time controller produces similar time delay compared to FLC control strategy at heavy traffic flows, since the fixed time controller was optimized with Webster’s equation to control traffic flows at intersection under heavy traffic conditions. The overall average stopped delay value was 11.6% lower using fuzzy logic controller as compared to fixed time controller.

VI. CONCLUSIONS

In this paper, we presented a PLA model for modelling fuzzy systems. The PLA model of fuzzy systems is completely generic. Depending on the application, the model has to be redefined. The created PLA model of fuzzy system could be used for analyzing its behaviour, safety, liveness and bounded liveness properties. In this paper for system behaviour analysis we used UPPAAL model-checker. The PLA model of fuzzy system can be also used for creating simulators for system performance analysis.

REFERENCES

[1] B. P. Zeigler, H. Praehofer, T. G. Kim, *Theory of Modeling and Simulation, 2nd. ed.* Academic Press. 2000.

[2] P.-A. Bisgambiglia, E. de Gentili, P. Bisgambiglia, J.-F. Santucci, “iDEVS: new method to study inaccurate systems”, in *Proc. of the IEEE International Conference on Fuzzy Systems,* 2009, pp. 300–307. [Online]. Available: http://dx.doi.org/10.1109/FUZZY.2009.5277046

[3] J. Lee, K. F. R. Liu, W. Chiang, “A fuzzy Petri net-based expert system and its application to damage assessment of bridges”, in *Proc. of the IEEE Transactions on Systems, Man, and Cybernetics, Part B.*, 1999, pp. 350–370.

[4] L. D'Errico, M. Loreti, “A process Algebra Approach to Fuzzy Reasoning”, in *Proc. of IFSA/EUSFLAT Conf.,* 2009, pp. 1136–1141.

[5] H. Pranevicius, T. Kraujalis, “Fuzzy traffic control for three-sided intersection” in *Proc. of the 12th International Conference on Transport Means,* 2008, pp. 52–55.

[6] H. Pranevicius, L. Simaitis, M. Pranevicius, O. Pranevicius, “Piece-Linear Aggregates for Formal Specification and Simulation of Hybrid Systems: Pharmacokinetics Patient-Controlled Analgesia”, *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering),* no. 4, pp. 81–84, 2011.

[7] G. Behrmann, A. David, K. G. Larsen, “A tutorial on UPPAAL”, *Formal Methods for the Design of Real-Time Systems*, Springer, pp. 200–236, 2004. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-30080-9_7

[8] S. Sheikh-Bahei, P. Lino, J. Liu, M. Jamshidi, “An Intelligent Discrete Event System Approach to Modeling, Simulation and Control of Autonomous Agents”, *Intelligent Automation and Soft Computing Journal,* vol. 4, no. 10, pp. 337–348, 2004.

[9] H. Pranevicius, “Aggregate approach for specification, validation, simulation and implementation of network protocols”, *Lecture Notes in Computer Science,* pp. 433–477, 1991. [Online]. Available: http://dx.doi.org/10.1007/BFb0019364

[10] H. Pranevicius, T. Kraujalis, “Knowledge based traffic signal control model for signalized intersection”, *Transport*, vol. 27, no. 3, pp. 263–267, 2004. [Online]. Available: http://dx.doi.org/10.3846/16484142.2012.719545