Fuzzy-Timed Automata

F. Javier Crespo, Alberto Encina, Luis Llana

Fuzzy-timed automata*

F. Javier Crespo[1], Alberto de la Encina[1], Luis Llana[1]
javier.crespo@fdi.ucm.es, {albertoe,llana}@sip.ucm.es

DSIC, Universidad Complutense de Madrid, Spain

Abstract. Timed automata theory is well developed in literature. This theory provides a formal framework to model and test real-time systems. This formal framework supplies a way to describe transitions among states with timing constrains. These constraints are usually expressed with logic formulas involving the system clocks. The time domain of these clocks usually is considered dense, that is, the clocks take values in the real or rational numbers. Dealing with a domain like this can be hard, specially if we consider end points of intervals.
In this paper, we present a modification of the model that allows to use real time in an easier, more powerful and reliable approach for computing systems. Our proposed model exploits the concepts of fuzzy set theory and related mathematical frameworks to get a more flexible approach.

Keywords: Conformance Testing, Timed Automata, Fuzzy Set Theory

1 Introduction

Over the last decades Formal Methods have attracted the attention of researches all over the world. One of the first, and also one of the most important, enhancements to formal methods was the inclusion of temporal features. Just from the beginning, one the of the most important issues was the nature of time: whether the time is a discrete domain [8,16,15] or a dense one [19,3,5]. The authors in favor of a dense time domain argued that its expressive power is greater than the one of a discrete time domain. Those in favor of a discrete time domain argued that it is more realistic since, for instance, you can't measure $\sqrt{2}$ seconds.

As aforementioned, time can be considered discrete. In this case it is composed of sequential instants. Mathematically speaking a discrete time domain has a order relation that is isomorphic to the order relation of the natural numbers. This model implies the existence of abrupt jumps. Therefore, even the absence of vagueness, a discrete time model becomes imprecise in the real world.

On the other hand, time can be modeled to be dense. A dense time domain has an order relation similar to the one of the real numbers or the rational numbers: between two time instants there is always another time instant. In others words, there are not any abrupt jumps. When working with a dense time

* Research partially supported by the Spanish MCYT projects TIN2006-15578-C02-01 and TIN2009-14312-C02-01.

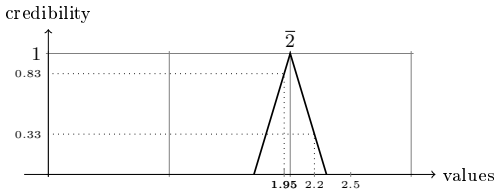


Fig. 1. Fuzzy number 2

domain it is usually necessary to discretize it. For instance, in Lazy Hybrid Automata [1], they consider that infinite precision is not possible so they discretize the continuous values by considering intervals.

Neither the discrete nor the dense models of time are capable to model properly the concept of time that humans beings have which it is inherently vague. In this paper, we propose the use of a structure that wraps the concept itself assuming the uncertainty and vagueness. In this context vagueness is not necessarily a criticism, but just a fact. The most popular approaches to handling vagueness and uncertainty as partial ignorance are Bayes theory, Shafer's evidence theory, the transferable belief model, and the possibility theory which are completely related to fuzzy sets.

Fuzzy set theory [20,21,14] provides a formal framework for the representation of vagueness. Our perception of reality is not perfect, although things can be *true* or *false*; our *environment* can make us doubt about a truth assessment. All measurement devices have an intrinsic error: if a thermometer indicates that the temperature is 35.4^{o}C, we know that the actual temperature is around that measurement. Something similar happens with time. We can claim that *it takes us an hour to go to work*; if the trip to work lasts 57 minutes in a particular day we know that the trip has lasted as usual, otherwise if it takes us 81 minutes we know that the trip has not lasted as usual. We can also be more accurate and give a degree of *confidence* of the measurement, which is a number in the interval $[0, 1]$ with being 1 the maximum degree of confidence and 0 the minimum. In this way, we can assess that 57 minutes is equal to 1 hour with a confidence degree close to 1, while 81 minutes has a confidence degree close to 0.

Therefore, a fuzzy number can be seen as a mapping from the set of real numbers to the interval $[0, 1]$. In Figure 1 we have depicted the fuzzy number 2, denoted by $\overline{2}$. In the figure we can observe that 1.95 is relatively close to 2 so it has a high confidence level 0.83. On the contrary, 2.2 is further from 2 so it has a lower confidence level 0.3, and 2.5, that is even further, has a confidence level of 0.

Timed automata theory is well developed in literature [2,3,7,17]. This theory provides a formal framework to model and test real-time systems. This formal framework supplies a way to describe transitions among states with timing constraints. The time model usually adopted in this theory is a dense one. Nevertheless, the numbers appearing in the time constraints they introduced always

range over the set of natural numbers $\mathbb{N}$. Hence, what they are really doing is a discretization of time. As aforementioned, we do not think this is the best way to model time. Hence, we adapt this framework to include fuzzy time constraints.

Once we have defined the fuzzy-timed specifications, it is necessary to check if the implementations meet them. One of the basic relationships between specifications and implementations is trace equivalence. But in this point there is a problem, let us suppose that a specification requires that an action a must be executed in the time interval $[1, 3]$. On the one hand, since we are in a fuzzy environment, we allow an implementation to execute action a at instant 3.0001; but, on the other hand, we cannot consider incorrect an implementation that *always* executes the action a within $[1, 3]$. This could be solved if the implementation relation were the trace inclusion. However, this relation is not completely satisfactory because an implementation that does not make any action at all is considered correct. Therefore, the implementation relation should be trace equivalence for the interval $[1, 3]$ and, at the same time, it should have some tolerance outside that interval.

Another important aspect of a theory is having the proper software tools. We have not developed any tool yet, but we have expressed our fuzzy implementation relations in terms of ordinary timed automata. In this way we can use the well known tools like Uppaal [4].

The relationship between fuzzy set theory and automata theory is not new. Fuzzy automata have been used to deal with different science fields: imprecise specifications [11], modeling Learning Systems [18] and many others. Fuzziness has been introduced in the different components of the automata: states, transitions, and actions [18,13,6]. A similar work of ours has been made in [7]. They use a many-valued logic in the transitions of the automata, and although they do not use fuzzy logic, their approach logic is similar to ours.

Probabilistic and stochastic models [12,9] might be considered related to the fuzzy model presented in this paper. In these models, the time when an action is performed follows a given random variable. Before setting a probability or a random variable in a model, a thorough statistical analysis should be performed. Unfortunately this requirement is difficult, if not impossible, to achieve. Thus, the specifier must choose the probability or a random variable based on her own experience. This experience fits better in a fuzzy environment.

The rest of the paper is organized as follows. Next, in Section 2 we introduce some concepts of fuzzy logic that are used along the paper. After that, in Section 3 we introduce the concept of fuzzy specifications, implementations and fuzzy time conformance. In Section 4 we present a case study to show the main characteristics of our model. Next, in Section 5 we show how to compute our fuzzy relations in terms of ordinary timed automata. Finally in Section 6 we give some conclusions and future work guidelines.

2 Preliminaries

In this paper we do not assume that the reader is familiar with fuzzy logic concepts. Therefore, we present some basic concepts of fuzzy logic.

2.1 Fuzzy relations

In ordinary logic, a set or a relation is determined by its characteristic function: a function that returns true if the element is in the set (or if some elements a related) and false otherwise. In the fuzzy framework we do not have that clear distinction between truth and falsehood; instead we have a complete range of values in the interval $[0, 1]$; the larger is the value, the more confidence we have in the assessment. In this paper we consider relations of real numbers $\mathbb{R}$. So a fuzzy relation is a mapping from the Cartesian product $\mathbb{R}^n$ into the interval $[0, 1]$

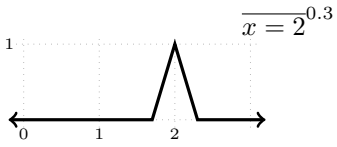
Definition 1. *A* fuzzy relation $\overline{A}$ *is a function* $\overline{A} : I\!R^n \mapsto [0, 1]$. *Let* $x \in I\!R^n$, *we say that* x *is* not included *in* $\overline{A}$ *if* $\overline{A} = 0$*; we say that* x *is* fully included *in* $\overline{A}$ *if* $\overline{A} = 1$. *The* kernel *of* $\overline{A}$ *is the set of elements that are fully included in* $\overline{A}$ □

The notion of α-cut is very important in fuzzy logic. Intuitively it establishes a credibility threshold. If we have a fuzzy relation $\overline{A}$, we accept that the relation is true for x if $\overline{A}(\overline{x})$ is above that threshold.

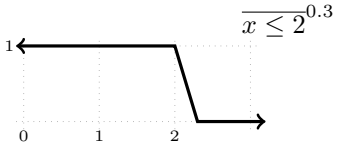
Definition 2. *Let* $\overline{A} : I\!R^n \mapsto [0, 1]$ *be a fuzzy relation and* $\alpha \in [0, 1]$. *We define the* α-cut of $\overline{A}$, *written* $\mathsf{cut}_\alpha(\overline{A})$, *as* $\mathsf{cut}_\alpha(\overline{A}) = \{x \in I\!R^n \mid \overline{A}(x) \geq \alpha\}$. □

Example 1. In this paper we consider the following fuzzy relations. Let us consider a non negative real number $\lambda \geq 0$,

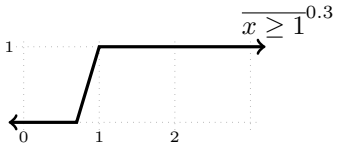
$$\overline{x = y}^\lambda = \begin{cases} 0 & \text{if } x \leq y - \lambda \text{ or } x > y + \lambda \\ 1 + \frac{x-y}{\lambda} & \text{if } y - \lambda < x \leq y \\ 1 - \frac{x-y}{\lambda} & \text{if } y < x \leq y + \lambda \end{cases}$$



$$\overline{x \leq y}^\lambda = \begin{cases} 1 & \text{if } x < y \\ 0 & \text{if } x > y + \lambda \\ 1 - \frac{x-y}{\lambda} & \text{if } y \leq x \leq y + \lambda \end{cases}$$



$$\overline{x \geq y}^\lambda = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{if } x \leq y - \lambda \\ 1 + \frac{x-y}{\lambda} & \text{if } y - \lambda < x \leq y \end{cases}$$



Note that these fuzzy relations are not order relations. We use these functions to describe fuzzy-timed automata that we present in Section 3. The only property we use, which simplifies the writing, is the commutativity of the equality $\overline{x = y}^\lambda = \overline{y = x}^\lambda$, viewed as binary operation over $\mathbb{R}$. □

2.2 Triangular Norms

A triangular norm (abbreviated t-norm) is a binary operation used in fuzzy logic to generalize the *conjunction* in propositional logic. In order to be able to generalize the conjunction, we have to analyze its basic properties: commutativity $p \wedge q = q \wedge p$, associativity $(p \wedge q) \wedge r = p \wedge (q \wedge r)$, identity true $\wedge\, p = p$, and nilpotency false $\wedge\, p =$ false.

Therefore, we require a t-norm to satisfy similar properties. We also require an extra property: monotonicity. Intuitively, the resulting truth value does not decrease if the truth values of the arguments increase.

Definition 3. *A* t-norm *is a function* $T : [0,1] \times [0,1] \mapsto [0,1]$ *which satisfies the following properties:*

- *Commutativity:* $T(x,y) = T(y,x)$.
- *Monotonicity:* $T(x,y) \leq T(z,u)$ *if* $x \leq z$ *and* $y \leq u$.
- *Associativity:* $T(x, T(y,z)) = T(T(x,y), z)$.
- *Number 1 is the identity element:* $T(x,1) = x$.
- *Number 0 is nilpotent:* $T(x,0) = 0$[1].

□

Since t-norms are associative, we can generalize them to lists of numbers:

$$T(x_1, x_2, \ldots, x_{n-1}, x_n) = T(x_1, T(x_2, \ldots, T(x_{n-1}, x_n) \ldots))$$

Example 2. The following t-norms are often used:

Łukasiewicz t-norm: $T(x,y) = \max(0, x + y - 1)$. We represent this t-norm with the symbol $\curlywedge$.

Gödel t-norm: $T(x,y) = \min(x,y)$. We represent this t-norm with the symbol $\overline{\wedge}$.

Product t-norm: $T(x,y) = x \cdot y$ (real number multiplication). We represent this t-norm with the symbol $\bigstar$.

We assume that any t-norm used has a symbol, we use the symbol $\triangle$ to represent a generic t-norm. We denote by $[\![\triangle]\!]$ the function associated with the symbol $\triangle$. For instance $[\![\overline{\wedge}]\!]$ is the min function. □

3 Fuzzy-Timed Automata

In this Section we introduce the basic notions of fuzzy-timed automata. But first, in order to make the paper self-contained, we first recall some common definitions of timed automata. Later we use and adapt these definitions to cope with fuzzy-timed automata. Apart from using these definitions later, we use ordinary timed automata to represent implementations of the fuzzy specifications.

Definition 4. Actions. *We assume that we have a finite alphabet of actions. The set of actions is denoted by* Acts. *Actions are ranged over by* $a, b, c, \ldots$

[1] As a matter of fact, this property can be easily deduced from the others.

Clocks. *A clock is a real valued variable. Clocks are ranged over by $x, y, z, \ldots$ We assume that we have a finite set of clocks denoted by* Clocks

Clock valuations. *A clock valuation $u : \mathsf{Clocks} \mapsto \mathbb{R}^+$ assigns non-negative real values to the clocks. The valuations of the clocks are denoted by $u, v, \ldots$ We denote by 0 the clock valuation where all clocks are set to 0.*
We use the following notation for valuations.

- *Let u be a valuation and let $d \in \mathbb{R}^+$. The valuation $u + d$ denotes the valuation $(u + d)(x) = u(x) + d$.*
- *Let u be a valuation and r a set of clocks. We denote the reset of all clocks of r in u, written as $u[r]$, as a valuation that maps all clocks in r to 0 and leaves invariant the rest of clocks.*

Clock constraints. *A clock constraint is a formula consisting of conjunctions of atomic relations of the form $x \bowtie n$ or $x - y \bowtie n$ where $n \in \mathbb{N}$, $x, y \in \mathsf{Clocks}$ and $\bowtie \in \{\leq, <, =, >, \geq\}$. We denote the set of clock constraints by $\mathcal{C}$.*
Let u be a valuation and $C \in \mathcal{C}$, we write $u \vDash C$ when the valuation u makes the clock constraint C true.

Timed Automata. *A timed automaton is a tuple (L, l_0, E, I) where:*

- *L is a finite set of locations.*
- *$l_0 \in S$ is the initial location.*
- *$E \subseteq L \times \mathsf{Acts} \times \mathcal{C} \times 2^{\mathsf{Clocks}} \times L$ is the set of edges; we write $l \xrightarrow{a,C,r} l'$ whenever $(l, a, C, r, l') \in E$.*
- *$I : L \mapsto \mathcal{C}$ is a function that assigns invariants to locations. As it is usual, the invariants of locations consist of conjunctions of atoms of the form $x \leq n$ where $n \in \mathbb{N}$.*

Operational semantics. *The operational semantics of a timed automaton is a timed labeled transition system whose states are pairs (l, u) where l is a location, u is a valuation of clocks, and its transitions are:*

- *$(l, u) \xrightarrow{d} (l, u + d)$ if $d \in \mathbb{R}^+$ and $u + d \vDash I(l)$.*
- *$(l, u) \xrightarrow{a} (l', u[r])$ if $l \xrightarrow{a,C,r} l'$ and $u \vDash C$ and $u[r] \vDash I(l')$.*

Automata traces. *Let $A = (L, l_0, E, I)$ be an automaton. A trace is a sequence $t = (d_1, a_1)(d_2, a_2) \ldots (d_n, a_n) \in (\mathbb{R}^+ \times \mathsf{Acts})^*$, written as $t \in \mathsf{tr}(A)$, if there is a sequence of transitions*

$$(l_0, 0) \xrightarrow{d_1} \xrightarrow{a_1} (l_1, u_1) \xrightarrow{d_2} \xrightarrow{a_2} (l_2, u_2) \cdots \xrightarrow{d_n} \xrightarrow{a_n} (l_n, u_n)$$

Conformance relation. *Among the different conformance notions in the timed automata literature we want to recall the trace inclusion and trace equivalence relations: $A_1 \leq_{tr} A_2 \Leftrightarrow \mathsf{tr}(A_1) \subseteq \mathsf{tr}(A_2)$ and $A_1 \equiv_{tr} A_2 \Leftrightarrow \mathsf{tr}(A_1) = \mathsf{tr}(A_2)$* □

As we have indicated, our objective is to define *fuzzy-timed* automata. In timed automata theory, time is expressed in the time constraints. Hence, we need to modify these constraints in order to be able to introduce fuzziness. In ordinary timed automata theory, the time constraints consist of conjunctions of inequalities. Instead of the ordinary crisp inequalities in Definition 4, we use their fuzzy counterparts appearing in Example 1. We could have more freedom in

allowing general convex fuzzy sets, but we have preferred to keep our constraints close to the original ones so we can use the theory developed for timed automata.

The role of a conjunction in Fuzzy Theory is played by t-norms. There is not a *canonical* t-norm, in Example 2 we have 3 of the more used t-norms. We have preferred to allow any of the available t-norms.

The time constraints be divided in two groups: The general fuzzy constraints that appear as the constraints attached to the actions; and the set of restricted constraints attached to location invariants where only inequalities of the form $\overline{x \leq n}^{\lambda}$ are allowed.

Definition 5.

1. *A* fuzzy constraint *is a formula built from the following B.N.F.:*

$$C ::= \mathsf{True} \mid C_1 \triangle C_2 \mid \overline{x \bowtie n}^{\lambda} \mid \overline{x - y \bowtie n}^{\lambda}$$

   where $\triangle$ *is a t-norm,* $\bowtie \in \{\leq, =, \geq\}$*,* $x, y \in \mathsf{Clocks}$*,* $\lambda \in \mathbb{R}^+$*, and* $n \in \mathbb{N}$*. We denote the set of fuzzy constraints by* $\mathcal{FC}$*.*
2. *A* fuzzy restricted constranint *is the subset of fuzzy constraints defined by the following B.N.F.:*

$$C ::= \mathsf{True} \mid C_1 \triangle C_2 \mid \overline{x \leq n}^{\lambda}$$

   where $\triangle$ *is a t-norm,* $x \in \mathsf{Clocks}$*,* $\lambda \in \mathbb{R}^+$*, and* $n \in \mathbb{N}$*. We denote the set of restricted fuzzy constraints by* $\mathcal{RFC}$*.*

□

In timed automata theory, the constraints are used to decide if the automata can stay in a location and to decide if a transition can be executed. All this is done by checking if a valuation satisfies the corresponding constraint. In fuzzy theory the notion of satisfaction is not crisp, we do not have a boolean answer but a range in the interval $[0, 1]$. Therefore, we do not have if a constraint is true or false but a *satisfaction grade* of a constraint.

Definition 6. *Let* u *be a clock valuation and* C *be a fuzzy constraint, we inductively define the* satisfaction grade of C in u*, written* $\mu_C(u)$*, as*

$$\mu_C(u) = \begin{cases} 1 & \text{if } C = \mathsf{True} \\ \overline{u(x) \bowtie n}^{\lambda} & \text{if } C = \overline{x \bowtie n}^{\lambda},\ \bowtie \in \{\leq, =, \geq\} \\ \overline{u(x) - u(y) \leq n}^{\lambda} & \text{if } C = \overline{x - y \leq n}^{\lambda},\ \bowtie \in \{\leq, =, \geq\} \\ [\![\triangle]\!](\mu_{C_1}(u), \mu_{C_2}(u)) & \text{if } C = C_1 \triangle C_2 \end{cases}$$

Let us remark that $\mu_C(u) \in [0, 1]$*.* □

Once the time constraints are defined, the definition of the fuzzy-timed automata is quite straightforward. It consists of replacing the ordinary time constraints by fuzzy time constraints.

Definition 7. *A* fuzzy-timed *automaton is a tuple* (L, l_0, E, I) *where:*

- L *is a finite set of locations.*
- $l_0 \in S$ *is the initial location.*
- $E \subseteq L \times \mathsf{Acts} \times \mathcal{FC} \times 2^{\mathsf{Clocks}} \times L$ *is the set of edges; we write* $l \xrightarrow{a,C,r} l'$ *whenever* $(l, a, C, r, l') \in E$.
- $I : L \mapsto \mathcal{RFC}$ *is a function that assigns invariants to locations.*

Let us note that the clock constraints in the edges have the general form as indicated in Definition 5-1, while the location invariants have the restricted form as indicated in Definition 5-2. □

Next we are going to define the operational semantics of fuzzy-timed automata. We need it to obtain the fuzzy traces that are used for the conformance relations. This operational semantics is given in terms of transitions, which are enabled when time constraints hold. Since we do not have crisp time constraints, the transitions must be decorated with a real number $\alpha \in [0, 1]$. This number indicates its certainty.

In order to define the operational semantics we need a t-norm. Let us explain the reason. In the definition of the operational semantics of a ordinary timed automaton, the action transitions requires the condition "$u \vDash C$ and $u[r] \vDash I(l')$". This conjunction must be transformed into its fuzzy version: a t-norm.

Definition 8. *Let* $fA = (L, l_0, E, I)$ *be a fuzzy-timed automaton and* $\triangle$ *be a t-norm. The* $\triangle$-operational semantics of fA *is the labeled transition system whose states are* $(l, u) \in L \times \mathsf{Clocks}$*, the initial state is* $(l_0, 0)$*, and the transitions are:*

1. $(l, u) \xrightarrow{d}_{\alpha} (l, u + d)$ *whenever* $\mu_{I(l)}(u + d) = \alpha$
2. $(l, u) \xrightarrow{a}_{\alpha} (l', u[r])$ *when ever* $l \xrightarrow{a,C,r} l'$, $\alpha_1 = \mu_{I(l')}(u[r])$, $\alpha_2 = \mu_C(u)$ *and* $\alpha = [\![\triangle]\!](\alpha_1, \alpha_2)$.

□

This operational semantics is not a pure timed transition system because the transitions are decorated with a real number $\alpha \in [0, 1]$ indicating the certainty to be executed. Anyway it is desirable that it has the main properties of timed transition systems: time determinism and time additivity.

Proposition 1. *Let* $fA = (L, l_0, E, I)$ *be a fuzzy-timed automaton and* $\triangle$ *be a t-norm, the* $\triangle$*-operational semantics of* fA *holds the following properties:*

Determinism. *If* $(l, u) \xrightarrow{d}_{\alpha_1} (l_1, u_1)$ *and* $(l, u) \xrightarrow{d}_{\alpha_2} (l_2, u_2)$*, then* $\alpha_1 = \alpha_2$, $l_1 = l_2$ *and* $u_1 = u_2$.

Additivity. *If* $(l, u) \xrightarrow{d_1}_{\alpha_1} (l_1, u_1) \xrightarrow{d_2}_{\alpha_2} (l_2, u_2)$ *then* $(l, u) \xrightarrow{d_1+d_2}_{\alpha_2} (l_2, u_2)$

Proof. To prove this it is enough to take into account that time transitions do not change location and the clock valuation is increased with the passing of time. □

As a further remark let us observe the α_2 of the transition $(l, u) \xrightarrow{d_1+d_2}_{\alpha_2} (l_2, u_2)$ in the time additivity property. In this case we do not have to combine it with

α_1 because the constraints considered in the locations have the form $\overline{x \leq n}^{\lambda}$, therefore $\alpha_1 \geq \alpha_2$. Intuitively if the automaton stays in a location, the likelihood of remaining in it can only decrease.

This operational semantics allows to define the traces of a fuzzy-timed automaton. The traces of a timed automaton consist of sequences of pairs, the first element of the pair expresses how long the automaton has stayed in a location and the second one the action that has made a location change. Now all these transitions are decorated with the certainty of being performed. So whenever the automaton stays in a location, we have an $\alpha \in [0,1]$ indicating the certainty of the automaton to stay in the location; and when a action transition is performed, we need a $\beta \in [0,1]$ indicating its certainty.

Definition 9. *Let $fA = (L, l_0, E, I)$ be a fuzzy-timed automaton and let $\triangle$ be a t-norm. We say that $t = (d_1, \alpha_1, a_1, \beta_1)(d_2, \alpha_2, a_2, \beta_2)\ldots(d_n, \alpha_n, a_n, \beta_n) \in (\mathbb{R}^+ \times [0,1] \times \mathsf{Acts} \times [0,1])^*$ is a $\triangle$-fuzzy trace of fA, written $t \in \mathsf{ftr}_{\triangle}(fA)$, if there is a sequence of transitions*

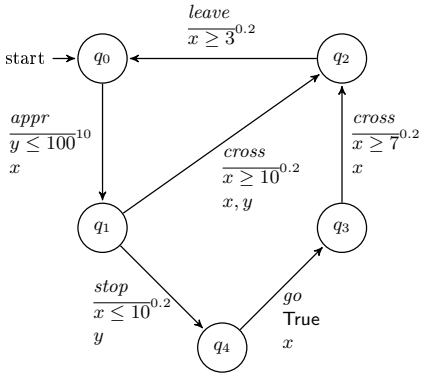
$$(l_0, 0) \xrightarrow{d_1}_{\alpha_1} \bullet \xrightarrow{a_1}_{\beta_1} (l_1, u_1) \xrightarrow{d_2}_{\alpha_2} \bullet \xrightarrow{a_2}_{\beta_2} (l_2, u_2) \cdots \xrightarrow{d_n}_{\alpha_n} \bullet \xrightarrow{a_n}_{\beta_n} (l_n, u_n)$$

in the $\triangle$-operational semantics of fA. In the previous sequence of transitions we have not specified the bullet states $\bullet$ to simplify the reading; they are always determined by the previous state: $(l_i, u_i) \xrightarrow{d_{i+1}}_{\alpha_{i+1}} (l_i, u_i + d_{i+1})$ □

As aforesaid, we consider implementations are real-time systems. When we check these systems we obtain real time measurements. We need to check if those measurements meet the fuzzy constraints given by the fuzzy specification. Therefore we assume that we have traces generated from a (non-fuzzy) timed automaton. These traces are of the form $(d_1, a_1)(a_2, d_2)\cdots(d_n, a_n)$. We have to compare them to the traces of the fuzzy automaton that have the form $(d_1, \alpha_1, a_1, \beta_1)(a_2, \alpha_n, d_2, \beta_2)\ldots(d_n, \alpha_n, a_n, \beta_n)$. In order to have a true/false assessment, first we have to combine the certainty values $(\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_n)$ with a t-norm and to compare the result to a given threshold $\alpha \in [0,1]$.

The conformance relations we want to mimic in our fuzzy framework are trace inclusion and trace equivalence. The first one is easier and corresponds to part 1 of Definition 10. Trace inclusion requires that all traces in the implementation are traces of the specification. In our case we do not have a crisp membership relation, instead we have the certainty values, a t-norm, and a threshold. Hence we compound the uncertainty values with the t-norm, we accept the trace if the obtained value is above the given threshold.

The relation corresponding to trace equivalence is in part 2 of Definition 10. It cannot be just trace equivalence. To understand the reason let us suppose that the specification allows the execution of action a in a given location with the fuzzy constraint $\overline{x \leq 3}^{0.2}$ and the threshold is 0.9. In this case we allow an implementation to execute the action a at time 3.01. But we do not want to force a correct implementation to execute that action at that time. That is, an implementation that always executes action a in the interval $[0,3]$ cannot be



q_0:	*Safe* True
q_1:	*Approaching* $\overline{x \leq 20}^{0.2} \bigstar \overline{y \leq 100}^{5}$
q_2:	*Crossing* $\overline{x \leq 5}^{0.2}$
q_3:	*Start* $\overline{x \leq 15}^{0.2}$
q_4	*Stop* True

Fig. 2. The *train* automaton

considered incorrect. To avoid this problem we require trace equivalence for the kernel (Definition 1) of the constraints and just trace inclusion for the rest of elements.

Definition 10. *Let A be a timed automaton, fA be a fuzzy-timed automaton, $\triangle_1$ and $\triangle_2$ be t-norms, and $\alpha \in [0,1]$.*

1. *A is $\triangle_1$-conformance with respect the $\triangle_2$-operational semantics of fA with an alpha cut of α, A $\mathsf{fconf}^{\alpha}_{\triangle_1,\triangle_2}$ fA, if for any trace $(d_1,a_1)\ldots(d_n,a_n) \in \mathsf{tr}(A)$ there exist a trace $(d_1,\alpha_1,a_1,\beta_1)\ldots(d_n,\alpha_n,a_n,\beta_n) \in \mathsf{ftr}_{\triangle_2}(fA)$ such that $[\![\triangle_1]\!](\alpha_1,\beta_1,\alpha_2,\beta_2,\ldots,\alpha_n,\beta_n) \geq \alpha$.*
2. *A is maximally $\triangle_1$-conformance with respect the $\triangle_2$-operational semantics of fA with an alpha cut of α, written A $\mathsf{fconfm}^{\alpha}_{\triangle_1,\triangle_2}$ fA, if A $\mathsf{fconf}^{\alpha}_{\triangle_1,\triangle_2}$ fA and for any fuzzy trace $(d_1,1,a_1,1)(d_2,1,a_2,1)\cdots(d_n,1,a_n,1) \in \mathsf{ftr}_{\triangle_2}(fA)$ we have $(d_1,a_1)(d_2,a_2)(d_n,a_n) \in \mathsf{tr}(A)$*

□

4 Case study

In this section, we present a case study of an automaton with five states. This automaton, which is adapted from the Uppaal demo directory, appears in Figure 2. It represents a train passing through a crossing.

First of all let us show the need of a t-norm to define the operational semantics. Let us suppose that we measure that a train has stayed in state q_0 for 101 time units. The credibility of the corresponding transition of the fuzzy-timed automaton depends on the used t-norm. We have a valuation u where the value of the clock y is 101, so we have $\overline{y \leq 100}^{10} = 0.9$ (the constraint in the transition) and $\overline{y \leq 100}^{5} = 0.8$ (the constraint in the invariant of state q_2). So we have $(101, 1, appr, 0.8) \in \mathsf{ftr}_{\bar{\wedge}}(train)$, $(101, 1, appr, 0.72) \in \mathsf{ftr}_{\bigstar}(train)$,

and $(101, 1, appr, 0.7) \in \mathsf{ftr}_{\curlywedge}(train)$ Thus, if we observe that an implementation A_1 verifies $(101, appr) \in \mathsf{tr}(A_1)$, we can say A_1 $\mathsf{fc\not{o}nf}^{0.8}_{\triangle,\bigstar}$ *train* and A_1 $\mathsf{fc\not{o}nf}^{0.8}_{\triangle,\curlywedge}$ *train*, but it might A_1 $\mathsf{fconf}^{0.8}_{\triangle,\overline{\wedge}}$ *train*.

Now let us focus on the loop among the states q_0, q_1, and q_2. Since the clock x is set to 0 in each transition, the following traces do not depend on any particular t-norm, that is, for any t-norm $\triangle$ we have

$$(20, 1, appr, 1)(10.02, 1, cross, 0.9)(4, 1, cross, 1)$$
$$(20, 1, appr, 1)(10.01, 1, cross, 0.95)(2.98, 1, cross, 0.9) \in \mathsf{ftr}_{\triangle}(train)$$

If we take the certainty values of the trace and we compound them under the different t-norms, we have

$$[\![\overline{\wedge}]\!](1, 1, 1, 0.9, 1, 1, 1, 1, 1, 0.95, 1, 0.9) = 0.9$$
$$[\![\curlywedge]\!](1, 1, 1, 0.9, 1, 1, 1, 1, 1, 0.95, 1, 0.9) = 0.77$$
$$[\![\bigstar]\!](1, 1, 1, 0.9, 1, 1, 1, 1, 1, 0.95, 1, 0.9) = 0.76$$

Therefore, if an implementation A_2 exhibit the trace

$$(20, appr)(10.02, cross)(4, cross)(20, appr)(10.01, cross)(2.98, cross) \in \mathsf{tr}(A_2)$$

We can say that A_2 $\mathsf{fc\not{o}nf}^{0.9}_{\curlywedge,\triangle}$ *train* and A_2 $\mathsf{fc\not{o}nf}^{0.9}_{\bigstar,\triangle}$ *train*; but A_2 is still a candidate to conforms the fuzzy automaton with respect the Gödel t-norm, that is, it might A_2 $\mathsf{fconf}^{0.9}_{\overline{\wedge},\triangle}$ *train*. Furthermore, if the loop continues with time measurements outside the kernel of the constraints, like in the trace above, the global credibility continuously decreases under the Łukasiewicz t-norm and the product t-norm even if each single credibility is above any threshold. Thus, there will be no $\alpha \in [0, 1]$ such that A_2 $\mathsf{fconf}^{\alpha}_{\curlywedge,\triangle}$ *train* or A_2 $\mathsf{fconf}^{\alpha}_{\bigstar,\triangle}$ *train*. On the contrary, if each single credibility is above the 0.9 threshold, we could have that A_2 $\mathsf{fconf}^{0.9}_{\overline{\wedge},\triangle}$ *train*.

5 Transforming fuzzy automata

In this Section we are going to present a transformation from fuzzy-timed automaton into ordinary timed automaton. This transformation is used to characterize the fuzzy relations in terms of ordinary timed automata. Hence, we can take advantage of all the theory and tools developed within the framework of timed automata. The basic idea is to apply a syntactical α-cut to the automaton constraints. Let us first remark that this α-cut does not work in the general case. To understand the reason let us consider the following example.

Example 3. Let us consider the fuzzy relation $C = \overline{x \leq 5}^{0.2} \bigstar \overline{y \leq 7}^{0.4}$. Then the set $\mathsf{cut}_{0.8}(C)$ cannot be expressed in terms of inequalities. We have

$$\begin{aligned}\mathsf{cut}_{0.8}(C) = {} & \{(a, b) \mid a \leq 5 \wedge b \leq 7\} \cup \\ & \{(a, b) \mid a \leq 5 \wedge 7 < b \leq 7.32\} \cup \\ & \{(a, b) \mid 5 < a \leq 5.16 \wedge b \leq 7\} \cup \\ & \{(a, b) \mid 5.2 \geq a > 5 \wedge 7.4 \geq b > 7 \wedge (1 - \tfrac{a-5}{0.2})(1 - \tfrac{b-7}{0.4}) \geq 0.8\}\end{aligned}$$

This set cannot be expressed in terms of conjunctions and inequalities so it cannot be part of a constraint of a timed automata.

But If we take the Gödel t-norm instead of the product t-norm we have

$$\mathsf{cut}_{0.8}(\overline{x \leq 5}^{0.2} \mathbin{\overline{\wedge}} \overline{y \leq 7}^{0.2}) = \{(a,b) \mid a \leq 5.12 \wedge b \leq 7.32\}$$

That can be expressed in terms of inequalities and conjunctions: $x \leq 5.16 \wedge y \leq 7.32$. Strictly speaking, this is not a timed constraint expression since 5.32 and 7.36 are not integers. This is not a real problem since the locations and actions are finite sets, therefore the amount of numbers like those is finite. Hence, in order to consider them integers it is enough to consider a time change unit. □

This example shows that if want to make a transformation that keeps the α-cuts, we have to restrict ourselves to fuzzy constraints where the only appearing t-norm is the Gödel one.

Definition 11. *A* fuzzy-Gödel constraint *is a formula generated by the following B.N.F.:*

$$C ::= \mathsf{True} \mid C_1 \mathbin{\overline{\wedge}} C_2 \mid \overline{x \bowtie n}^{\lambda} \mid \overline{x - y \bowtie n}^{\lambda}$$

where $\triangle$ is a t-norm, $\bowtie \in \{\leq, =, \geq\}$, $x, y \in \mathsf{Clocks}$, λ is a non-negative rational number, and $n \in \mathbb{N}$.

Let fA be a fuzzy-timed automaton. We say that is a Gödel fuzzy-timed automaton *if all the fuzzy constraints are fuzzy-Gödel constraints.* □

For fuzzy-Gödel constraints we can generalize what we have observed in Example 3. We first define the syntactical α-cut of such constraints

Definition 12. *Let C be a fuzzy-Gödel constraint and let α be a rational number in the interval $[0,1]$, then we inductively define the* α-cut of C *as:*

$$\mathsf{cut}_{\alpha}(C) = \begin{cases} \mathsf{True} & \text{if } C = \mathsf{True} \\ x \bowtie n + \alpha \cdot \lambda & \text{if } C = \overline{x <= n}^{\lambda}, \bowtie \in \{\leq, =, \geq\} \\ x - y \bowtie n + \alpha \cdot \lambda & \text{if } C = \overline{x - y <= n}^{\lambda}, \bowtie \in \{\leq, =, \geq\} \\ C_1 \wedge C_2 & \text{if } C = C_1 \mathbin{\overline{\wedge}} C_2 \end{cases}$$

□

Now we can prove that $\mathsf{cut}_{\alpha}(C)$ is equivalent to the α-cut of the fuzzy constraint C. That is, the set of clocks that give a satisfaction grade greater than α is the same that makes the formula $\mathsf{cut}_{\alpha}(C)$ true.

Proposition 2. *Let C be a fuzzy-Gödel constraint and let α be a rational number in the interval $[0,1]$. Then $u \vDash \mathsf{cut}_{\alpha}(C)$ if and only if $\mu_C(u) \geq \alpha$.*

Proof. It is straightforward by structural induction of C. □

Now that we have shown that the Gödel-fuzzy constraints behave properly when applying the α-cuts, we can extend the concept to Gödel fuzzy automata.

Definition 13. *Let $fA = (L, l_0, E, I)$ be a Gödel fuzzy-timed automaton and let α be a rational number in the interval $[0,1]$. We define the automaton $\mathsf{cut}_\alpha(fA)$ as the automaton $(L, l_0, E_\alpha, I_\alpha)$, where E_α and I_α are defined as:*

- $(l, a, \mathsf{cut}_\alpha(C), r, l') \in E_\alpha$ *iff* $(l, a, C, r, l') \in E$.
- *If* $I_\alpha(l) = \mathsf{cut}_\alpha(I(l))$.

□

First we want to prove that these transformations are correct, that is the obtained automaton conforms the fuzzy specification. The conformance relations in Definition 10 need two t-norms, one to define the operational semantics of fuzzy-timed automata, and another one to combine the values of the operational semantics. Again, the t-norms we need are the Gödel t-norm. To prove it we first need an auxiliary lemma that relates the transitions of a fuzzy-timed automata and the ones of its α-cut.

Proposition 3. *Let $fA = (L, l_0, E, I)$ be a Gödel fuzzy-timed automaton and let α be a rational number in the interval $[0,1]$. Then:*

1. $(l,u) \xrightarrow{d} (l, u+d)$ *is a transition of the operational semantics of* $\mathsf{cut}_\alpha(fA)$ *if and only if there exists $\alpha' \geq \alpha$ such that $(l,u) \xrightarrow{d}_{\alpha'} (l, u+d)$ is a transition of the $\overline{\wedge}$-operational semantics of fA.*
2. $(l,u) \xrightarrow{a} (l', u')$ *is a transition of the operational semantics of* $\mathsf{cut}_\alpha(fA)$ *if and only if there exists $\alpha' \geq \alpha$ such that $(l,u) \xrightarrow{a}_{\alpha'} (l', u')$ is a transition of the $\overline{\wedge}$-operational semantics of fA.*

Proof. The first thing we have to take into account is that the time constraints of $\mathsf{cut}_\alpha(fA)$ are obtained by applying the α-cuts. Hence the invariant of a location l in the automaton $\mathsf{cut}_\alpha(fA)$ is $I_\alpha(l) = \mathsf{cut}_\alpha(I(l))$. Also, by Proposition 2 we have $\mu_{I(l)}(u) \geq \alpha$ if and only if $u \vDash I_\alpha(l)$. So we have part 1.

For part 2, we also have to consider that the transitions of the automaton $\mathsf{cut}_\alpha(fA)$ have the form $(l, a, \mathsf{cut}_\alpha(C), r, l')$ where (l, a, C, r, l') is a transition of fA. Let us recall that we have consider the Gödel t-norm to build the operational semantics of fA therefore $(l,u) \xrightarrow{a}_{\alpha'} (l', u')$ if and only if $\alpha' = \min(\mu_{I(l')}(u[r]), \mu_{I(l)}(u))$. Therefore $\mu_{I(l')}(u[r]) \geq \alpha' \geq \alpha$ and $\mu_{I(l)}(u) \geq \alpha' \geq \alpha$. Again by Proposition 2, that is true if and only if $u[r] \vDash \mathsf{cut}_\alpha(I(l'))$ and $u \vDash \mathsf{cut}_\alpha(I(l))$. □

Proposition 4. *Let fA be a Gödel fuzzy-timed automaton and let α be a rational number in the interval $[0,1]$. Then* $\mathsf{cut}_\alpha(fA)\ \mathsf{fconfm}^{\alpha}_{\overline{\wedge},\overline{\wedge}}\ fA$

Proof. From the previous proposition we have $(d_1, a_1) \cdots (d_n, a_n) \in \mathsf{tr}(\mathsf{cut}_\alpha(fA))$ if and only if there exist $\alpha_i, \beta_i \geq \alpha$ such that $(d_1, \alpha_1, a_1, \beta_1) \cdots (d_n, \alpha_n, a_n, \beta_n) \in \mathsf{ftr}_{\overline{\wedge}}(fA)$. From which the result is immediate. □

From this proposition we get the main result of this Section: in order to prove if some automaton conforms a fuzzy specification it is enough to consider the α-cut of the fuzzy automaton.

Theorem 1. *Let A be an timed automaton, fA be a Gödel fuzzy-timed automaton, and $\alpha \in [0,1]$. Then*

- $A\ \mathsf{fconf}^{\alpha}_{\overline{\wedge},\overline{\wedge}}\ fA$ *iff* $A \leq_{tr} \mathsf{cut}_\alpha(fA)$
- $A\ \mathsf{fconfm}^{\alpha}_{\overline{\wedge},\overline{\wedge}}\ fA$ *iff* $A \leq_{tr} \mathsf{cut}_\alpha(fA)$ *and* $\mathsf{cut}_1(fA) \leq_{tr} A$.

□

6 Conclusions

In this paper, we have introduced a formalism that deals with time by using fuzzy logic. In our opinion, this is an important feature since it is difficult (if not impossible) to have a perfect and exact measurement of time. There have been many issues about the nature of time in formal methods, specially if it should be consider a discrete or dense entity. These issues do not have sense in a fuzzy framework because we are assuming, since the beginning, that we do not have an exact measurement of time. Although these measurements are imprecise, the fuzzy framework provides a precise measurement of this imprecision.

This work has been focused in the automata framework. As we have mentioned in the introduction, this fuzzy logic has been introduced successfully in the literature. However, as far as we know this is the first time where the fuzziness has been introduce in the time domain of the automata. We have defined the concept fuzzy-timed automata and defined a conformance relation. The concept of t-norm plays a central role in both the definition of the automata and the conformance relation. Among the different t-norms we have discover that the Gödel t-norm is specially important. By using it, we can apply a syntactical α-cut to the fuzzy-timed automata and reduce the fuzzy conformance relations to conformance relations between ordinary timed-automata.

This paper has a number of open loose ends that we plan to address in the future. In first place, the conformance relations are not completely satisfactory. We have introduced two conformance relations: $\mathsf{fconf}^{\alpha}_{\triangle_1,\triangle_2}$ and $\mathsf{fconfm}^{\alpha}_{\triangle_1,\triangle_2}$. The first one corresponds directly to trace inclusion while the second one is closer to trace equivalence. The first one has the same problems as the trace inclusion; while the second one has a conceptual problem because we require that certain fuzzy formula has a satisfaction grade of 1. We do not think this is very adequate in a fuzzy environment and we are working in an alternate possibility.

Another important issue to address is to implement a software tool. In Section 5, we have reduced the fuzzy relations to relation between ordinary automata. So, we can take advantage of the already existing software tools to verify the fuzzy relations, we are already working in a tool that performs this task. The problem is that to perform the transformation we can only work with Gödel t-norms. Apart from the fact that we would like to work with another t-norms, we think we can address this problem in an alternative way. The existing tools for timed automata like Uppaal [4] perform a discretization of time to transform a timed automaton into a non-timed automaton. What we plan is to take advantage of the fuzziness to try to avoid the discretization of time.

Acknowledgments. We like to express our gratefulness to the anonymous reviewers for their valuable comments. We also want to apologize to them for the bad English of the first submission. We hope we have improved it.

References

1. M. Agrawal and P. S. Thiagarajan. The discrete time behavior of lazy linear hybrid automata. In *Hybrid Systems: Computation and Control*, volume 3414 of *LNCS*,

pages 55–69. Springer, 2005.
2. R. Alur, C. Courcoubetis, and D. L. Dill. Model-checking for probabilistic real-time systems (extended abstract). In Leach-Albert et al. [10], pages 115–126.
3. R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
4. G. Behrmann, A. David, and K. G. Larsen. A tutorial on uppaal. In Marco Bernardo and Flavio Corradini, editors, *SFM*, volume 3185 of *LNCS*, pages 200–236. Springer, 2004.
5. J. Davies and S. Schneider. A brief history of timed csp. *Theor. Comput. Sci.*, 138(2):243–271, 1995.
6. M. Doostfatemeh and S. C. Kremer. General fuzzy automata, new efficient acceptors for fuzzy languages. In *IEEE International Conference on Fuzzy Systems*, pages 2097–2103. IEEE, 2006.
7. A. Fernández-Vilas, J. J. Pazos-Arias, and R. P. Díaz-Redondo. Extending timed automaton and real-time logic to many-valued reasoning. In *FTRTFT*, volume 2469 of *LNCS*, pages 185–204. Springer, 2002.
8. M. Hennessy and T. Regan. A process algebra for timed systems. *Information and Computation*, 117(2):221–239, 1995.
9. M. Z. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In C. Palamidessi, editor, *CONCUR*, volume 1877 of *LNCS*, pages 123–137. Springer, 2000.
10. J. Leach-Albert, B. Monien, and M. Rodríguez-Artalejo, editors. *Automata, Languages and Programming, 18th International Colloquium, ICALP 91*, volume 510 of *LNCS*. Springer, 1991.
11. S. I. Mensch and H. M. Lipp. Fuzzy specification of finite state machines. In G. Adshead and J. A. G. Jess, editors, *EURO-DAC*, pages 622–626. IEEE Computer Society, 1990.
12. M. G. Merayo, M. Núñez, and I. Rodríguez. Implementation relations for stochastic finite state machines. In András Horváth and Miklós Telek, editors, *EPEW*, volume 4054 of *LNCS*, pages 123–137. Springer, 2006.
13. M. Mraz, I. Lapanja, N. Zimic, and J. Virant. Fuzzy numnbers as input to fuzzy automata. In *IEEE*, pages 453–456. IEEE, 199.
14. H. T. Nguyen and E. A. Walker. *A first course in fuzzy logic*. CRC Press, Inc., Boca Raton, FL, USA, 1996.
15. X. Nicollin and J. Sifakis. The algebra of timed processes, atp: Theory and application. *Information and Computation*, 114(1):131–178, 1994.
16. Y. Ortega-Mallén. Operational semantics for timed observations. In Jan Vytopil, editor, *FTRTFT*, volume 571 of *LNCS*, pages 507–527. Springer, 1992.
17. J. Springintveld, F. W. Vaandrager, and P. R. D'Argenio. Testing timed automata. *Theoretical Computer Science*, 254(1-2):225–257, 2001.
18. W. G. Wee and K. S. Fu. A formulation of fuzzy automata and its application as a model of learning systems. *IEEE Trans. on Systems, Man and Cybernetics*, 5(3):215–223, 1969.
19. W. Yi. Ccs + time = an interleaving model for real time systems. In Leach-Albert et al. [10], pages 217–228.
20. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
21. L. A. Zadeh. Fuzzy logic and its application to approximate reasoning. In *IFIP Congress*, pages 591–594, 1974.