An improved and tool-supported fuzzy automata framework to analyze heart data*

Iván Calvo, Mercedes G. Merayo, Manuel Núñez

Departamento Sistemas Informáticos y Computación
Universidad Complutense de Madrid, Spain
email: {ivcalvo,mlmgarci,manuelnu}@ucm.es

Abstract. In this paper we present a new formalism that can be used to formally specify complex systems where uncertainty plays an important role. We introduce an improved version of a previous formalism, a *fuzzy* version of finite automata, by defining its syntax and semantics. We successfully applied this formalism to define and analyze information extracted from electrocardiograms (ECGs).

1 Introduction

The use of formal methods in the development of complex systems improves their reliability. The main obstacle to have a widespread use of formal methods is associated with their complexity and the lack of tools to support them. In addition, general purpose formalisms (such as timed automata [1]) are not suitable to be used in specific fields. This is the case of the application considered in this paper: modeling and analyzing the behavior of the heart. There are several approaches to formally model the heart [9,10,13,5] but they fail to take into account common characteristics in biological systems such as uncertainty and imprecision. If we use inaccurate models to analyze a system (whether biological or not), then we will not be able to obtain useful results.

There are many proposals to include fuzzy logic into automata [18,16,6,2]. The last of these proposals, produced in our research group, combined the best features of previous work. However, recent work [4] using this version of fuzzy automata has shown some of its weaknesses, in particular, while modeling and analyzing information about the heart. Actually, we are interested in modeling the behavior of the heart by taking into account data extracted from ECGs (electrocardiograms): heartbeats per minute and RR wave durations. Our model takes normal levels of ECGs from the study of numerous patients [17,7]. In order to assess the usefulness of the model, we will analyze real patients data to check whether our model detects existing illnesses.

We briefly comment on the main improvements of our new fuzzy automata. We have included a variable as a parameter of the actions. This fact strongly simplifies, while keeping the same expressive power, the previous framework

* Research partially supported by the projects DArDOS (TIN2015-65845-C3-1-R (MINECO/FEDER)) and SICOMORo-CM (S2013/ICE-3006).

based on *fields*. We have simplified the operational semantics by removing an additional clause that offered no substantial benefit. This modification allowed us to fix a potential source of problems in our previous formalism. We have clarified the way in which we obtain and process the data that we feed to the automaton. Moreover, the information returned for each patient after processing their data is structured in a more useful way. Specifically, we have disaggregated the obtained data and we currently provide different alternatives (together with its associated grade of confidence).

Finally, we review the main implementation details of our framework. In order to obtain the data that we feed to our automata, we used the `WFDB Software` [15] to extract *Inter-beat (RR) intervals* from the dataset that we consider in our case study [14]. We included calls to the functionalities `sqrs` and `ann2rr` in our patient data loading script. We obtain several `.cvs` files for each patient's header and record files. These two files are later used by our trace generating script, which produces the data sent to the automaton. Essentially, we format the data in a way that can be easily processed by our automaton. First, the automaton receives the gender and age of the patient. Next, for each minute, the environment sends a sequence of values and the automaton produces a diagnosis (*ok* or *alarm*). These sequences are formed by the actual number of *beats* recorded in the minute (BPM, one value per minute) followed by the length of each *RR interval*. Therefore, for each patient we obtain a sequence of n ok/alarm messages, being n the number of minutes in the record, labeled with the associated minute and the grade of confidence on the validity of the result.

The rest of the paper is structured as follows. Section 2 introduces the syntax of our formalism and its operational semantics. Section 3 defines our model of the heart and evaluates its usefulness with real data. Finally, in Section 4 we present our conclusions and some lines for future work.

2 An extended and improved version of fuzzy automata

Fuzzy automata [2] have been recently used in our research group and we have detected several deficiencies that we would like to fix in this improved version. In this section we introduce our new formalism. First, we briefly present some concepts related to fuzzy logic. The interested reader is referred to our previous work [3,4] for more details because, due to space limitations, we cannot review all the needed concepts and notations.

Fuzzy relations are similar to *boolean relations* but instead of returning true or false, they return a real value in the interval $[0, 1]$. The idea is that if we are sure that something holds then we have confidence equal to 1; otherwise, we will have a confidence less than 1, in particular, if we are sure that the relation does not hold then we have confidence equal to 0. Usual fuzzy relations can be found in our previous work [4]. In particular, we defined a relation $\overline{\alpha \leq \cdot \leq \beta}^{\delta}$. Intuitively, if a value x is such that $\alpha \leq x \leq \beta$ then we claim that the relation holds with confidence 1. If this is not the case and the distance from x to α or β is less than δ then we have a positive confidence (the confidence diminishes when the distance increases). Finally, if $x \notin [\alpha, \beta]$ and it is *far* from the interval then we have confidence 0 on x belonging to the interval.

We combine confidence values by using *t-norms*. In this paper we use two of them: the *Gödel t*-norm (computing the minimum of all the values and denoted by $\overline{\wedge}$) and the *Hamacher product t*-norm. This last *t*-norm (denoted by $\circledast$) is, as usual, associative. Therefore, it is enough to define it for two arguments δ_1 and δ_2 as $\frac{\delta_1 \cdot \delta_2}{\delta_1 + \delta_2 - \delta_1 \cdot \delta_2}$.

After this brief review, we can define our improved version of fuzzy automata. First, we introduce some additional notation.

Definition 1. *Let* Acts *be a finite set of actions (they will be used to model the actions that a system can perform). We will distinguish between* inputs*, preceded by* ?*, and* outputs*, preceded by* !.

A fuzzy constraint *is a formula where fuzzy relations are used instead of boolean relations and t-norms are used to combine relations instead of boolean operators. We denote by $\mathcal{FC}$ the set of fuzzy constraints.*

Let C be a fuzzy constraint with n parameters and $\overline{x} = (x_1, \ldots, x_n) \in \mathbb{R}_+^n$. We have that $\mu_C(\overline{x})$ denotes the satisfaction degree or grade of confidence (GoC) of C for $\overline{x}$ *(a formal definition can be found in our previous work [4]).*

In our case study, *inputs* will be used to receive information about the patient (e.g. BPM and RR). We will use *outputs* to send messages to the environment. For example, we can issue an alarm indicating that a potential problem has been found at a certain minute and with a certain grade of confidence. Again, we refer the interested reader to our previous work [4] for longer discussions and examples on fuzzy constraints. A simple example of a fuzzy constraint is $\overline{60 \leq x \leq 69}^{13}$. The idea is that if a patient is in the expected age range, that is $[60, 69]$, then the confidence is equal to 1. Otherwise, if the distance to the interval is more than 13 then the confidence is equal to zero. Finally, if the age is *close* to the interval, then the confidence linearly increases when the distance is reduced. Let us note that if $\delta = 0$ then fuzzy constraints become *usual* constraints.

In order to track some relevant data during the execution of the automaton, we introduce the following notion to deal with *variables* and *variable transformations*.

Definition 2. *Let X be a set of variables taking values in $\mathbb{R}_+$. We define the set of variable transformations $\mathcal{VT}$ as the set of expressions assigning a value to each variable of the set. We will use the following notation*

$$[y_1/x_1, \ldots, y_m/x_m]$$

where each y_i is a real valued expression over the set of variables X and each x_i is a variable in X. The semantics of this transformation is that each x_i takes the value obtained after evaluating y_i (possible taking into account the current values of the variables in X); if a variable x_i does not appear in the expression then we have that the variable does not change its value after the transformation.

Let $\mathfrak{R}$ be equal to $\bigcup_{i \geq 1} \mathbb{R}_+^i$, that is, $\mathfrak{R}$ is a set containing all the tuples, of any arity, with real number values.

Definition 3. *A* fuzzy automaton *is a tuple* $(S, \text{Acts}, X, s_0, T)$ *where:*

- S *is a finite set of states.*
- Acts *is a finite set of actions, partitioned into a set of inputs* I *and a set of outputs* O.
- X *is a set of variables ranging over* $\mathbb{R}_+$. *The set includes a variable* GoC, *which will be used to store the Hamacher grade of confidence associated with sequences of transitions. We assume that the initial value of* GoC *is* 1.
- s_0 *is the initial state.*
- $T \subseteq S \times (I \times X \cup O \times \mathfrak{R}) \times \mathcal{FC} \times \mathcal{VT} \times S$ *is the set of transitions. We assume that each transition implicitly applies the following variable transformation* $[\mu_C \circledast GoC/GoC]$.

Fuzzy automata are directed graphs where transitions have an associated condition, indicating the grade of confidence with which we can execute the transition, and a transformation of the variables. In addition, transitions can be labeled either by an input or by an output. Intuitively, a transition $(s, (a, \alpha), C, V, s') \in T$ denotes that if the automaton is in state s and receives/sends from/to the environment $a(\alpha)$, where a is an input/output action and α is a variable/tuple of positive real values, then the previous transition can be triggered if $\mu_C(\alpha) > 0$, the new values of the variables will be given by V, and the automaton will move to state s' . Usually, transitions labeled with an output will have a trivial fuzzy constraint (that is, it will be `True`). In order to simplify the graphical representation, if we have two transitions from one state to another one labeled by different fuzzy constraints C_1 and C_2 then we will only draw one transition labeled by $C_1 \| C_2$ (for example, see Figure 1, transition from q_{38} to q_{39}).

Next, we are going to define the operational behavior of fuzzy automata. This *operational semantics* will be used to obtain their (fuzzy) traces. We start in the initial state of the automaton, produce actions and trigger a transition labelled by the action if the attached value is included in the fuzzy relation induced by the constraint. We decorate transitions with a real number $\epsilon \in [0, 1]$ indicating its certainty. First, we define a single transition and then we concatenate transitions to conform traces.

Definition 4. *Let* $A = (S, \text{Acts}, X, s_0, T)$ *be a fuzzy automaton and* $\triangle$ *be a t-norm. Given states* $s_1, s_2 \in S$, *we have a* transition *from* s_1 *to* s_2, *after performing the action* $a \in \mathcal{A}ct$ *for* α *with confidence* ϵ, *denoted by* $s_1 \xrightarrow{(a(\alpha),V)}_{\epsilon} s_2$, *if the following conditions hold:*

- *There exists* $C \in \mathcal{FC}$ *such that* $(s_1, (a, \alpha), C, V, s_2) \in T$.
- $\mu_C(\alpha) = \epsilon$ *and* $\epsilon > 0$.
- *The new values of the variables belonging to* X *are given by* $V \in \mathcal{VT}$.

We say that a sequence $s_0 \xrightarrow{(a_1(\alpha_1),V_1)}_{\epsilon_1} s_1 \xrightarrow{(a_2(\alpha_2),V_2)}_{\epsilon_2} \cdots \xrightarrow{(a_n(\alpha_n),V_n)}_{\epsilon_n} s_n$ *of consecutive transitions starting in the initial state of the automaton* A *is a* $\triangle$-trace *of* A *if* $\epsilon = \triangle\{\epsilon_1, \ldots, \epsilon_n\}$ *is greater than zero and the values of the variables of* X *are the result of sequentially applying the variable transformations*

$V_1, \ldots, V_n$ to X. We call this composed variable transformation V. In this case we write $s_0 \xrightarrow{(a_1,\ldots,a_n,\alpha_1,\ldots,\alpha_n,V)}_{\epsilon} s_n$.

Example 1. Consider the component of the automata *Heart* given in Figure 1 where we assume that the value 0 denotes males and 1 denotes females. For example, we could observe a trace such as

$$(?checkGender(0)), (?checkAge(65)), (?minute(1)), (?readBPM(62)), (?readRR(977)), (?readRR(968)), (\cdots), (?noMorePendingRR()), (!ok(1, 1.0))$$

as the result of having the automaton working during a minute by analyzing a sample of a 65 years old male patient. As usual, inputs are preceded by ?, outputs are preceded by ! and $(\cdots)$ indicates that some $?readRR$ actions have been omitted from the trace due to presentation purposes.

3 Case Study

In this section we present the application of our fuzzy automata in a real scenario: prediction of heart problems. We define the automaton *Heart*, which is able to alert about the level of risk of a patient. In order to produce a diagnosis, we use the available information and physical evidence collected from electrocardiograms (ECGs). The information managed by the automaton is:

- Gender. We have 2 groups: Men and Women.
- Age. We have 8 groups of age.
- Heartbeats. The range of correct heartbeats per minute (BPM) for healthy patients, according to their gender and age.
- RR waves. The range of correct RR waves duration (measured in milliseconds) for healthy patients, according to their gender, age and BPM.

Additionally, we consider that our set of actions consists of the following operations:

- $?checkGender(gen)$. It reads the gender of the patient.
- $?checkAge(age)$. It reads the age of the patient.
- $?minute(m)$. It reads the current minute of the recording.
- $?readBPM(bpm)$. It reads the amount of beats in the current minute.
- $?readRR(rr)$. It reads the next RR interval in the current minute.
- $?noMorePendingRR(\cdot)$. It receives a notification that there are no more RR intervals in the current minute.
- $?endOfRecord(\cdot)$. It receives a notification to denote that there are no more minutes in the analyzed record.
- $!recordAlarm(min, GoC)$. It indicates, with a grade of confidence equal to GoC, that an alarm will be raised in the current minute.
- $!ok(min, GoC)$. It indicates, with a grade of confidence equal to GoC, no alarm will be recorded for the current minute.

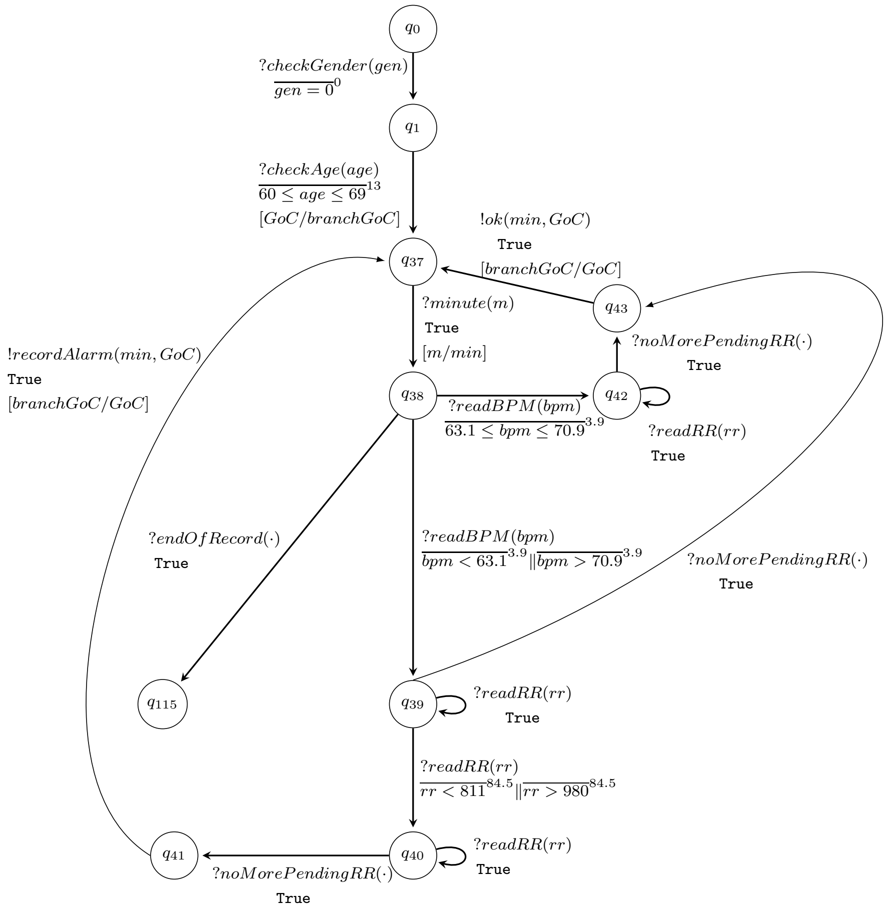

Fig. 1. Sub-automaton corresponding to male patients between 60 and 69 years old.

Our automaton has a total of 116 states and 226 transitions. Initially, our automaton has two transitions: one per gender. After that, each branch has 8 transitions, one per age group. Each of these age groups has an associated sub-automaton with 7 states and a transition to a common *final* state (the state q_{115}) to denote that all the data for this patient have been processed. Each minute, the automaton checks if the number of beats falls within the normal amount of beats per minute in the age range. If it does, then the automaton does not take into account the RR intervals in that minute and signals that minute to be *ok*. Otherwise, the automaton processes each RR interval. If there is at least one interval out of range, an alarm is raised (with a certain grade of confidence). As previously commented, we use variables to track some data. More specifically, in addition to the *built-in* variable *GoC*, we use a variable called *branchGoC*. In every transition entering an *age branch*, the recently computed *GoC* value is saved in the *branchGoC* variable. At the end of each minute, that value is stored in the *GoC* variable. Therefore, the *GoC* values associated with each minute are not affected by the values corresponding to the previous minutes. The value received when performing a *?minute* input action is also stored, in the variable *min*, and returned after processing the corresponding minute data. If at any moment of the study the state q_{41} is reached, then the automaton will record the current state of the patient as a case in which he suffers the risk of having a heart problem. For each minute, we process data until we find a potential problem during that minute slot (that is, the state q_{41} is traversed), having the samples a duration of around 30 minutes. Therefore, for each patient, the number of alarms that the automaton can raise is bounded by the number of minutes in the recording.

The values used to define our fuzzy constraints are taken from previous work. Normal ranges for heartbeats per minute, classified by gender, have been gathered from the work of Rijnbeek et al. [17]. In the case of the age, the δ value is obtained from the 20% of the highest value of each age range. The idea is that it is possible to wrongly classify a patient according to their age. For example, if we have a 53 years old male patient then we should classify him in the age group between 50 and 59. In addition, it may happen that the patient has a very healthy life style and, therefore, their heart is *younger*. Therefore, we should also classify them in the previous age group and decide whether their recorded data fits better in their *real* age group or in the *closer one* to the real one. In the case of heartbeats, we had the median, 2nd percentile and 98th percentile from the database, but they were not applicable as limits for our automaton because they are not characteristic data of the sample of patients. For that reason, we applied the estimations made by Hozo et al. [8]: if the size of a sample exceeds 25 then the median itself is the best estimator for the mean and the best estimator for the standard deviation is

$$\sigma \approx \frac{b-a}{6}$$

where a is the smallest value of the sample (the 2nd percentile in our case) and b is the largest value of the sample (the 98th percentile in our case). Fortunately, the database that we use [17] has information from 13354 patients and, therefore, we can base our limits on the median of each range while δ is based on the standard deviations of each range.

Concerning RR waves, we have used the data from the work of Haarmark et al. [7]. The problem in this case was that we only had the information of the RR waves duration for the patients of the age range $[30-39]$. So, if we had used these limits for all the patients, then the prediction would have been erroneous. Therefore, we also considered another related work [11] where the duration of the RR waves is derived from data corresponding to heartbeats. So, our limits are based on the application of the following formula

$$RR_{ms} \approx \frac{60000}{bpm}$$

to the BPM data obtained from the work of Rijnbeek et al. [17].

Table 1. Results corresponding to patients #100 and #104. Each cell includes a pair *GoC ok*/*GoC alarm* computed from the data observed during that minute.

	#100				#104			
	(50-59)	**(60-69)**	**(70-79)**	**(>80)**	**(50-59)**	**(60-69)**	**(70-79)**	**(>80)**
min. 1	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 2	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.00/0.41	0.54/1.00	0.68/0.75	0.12/0.12
min. 3	0.00/0.15	0.00/1.00	0.00/0.94	0.31/0.22	0.00/0.41	0.00/1.00	0.00/0.75	0.00/0.12
min. 4	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 5	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.00/0.41	0.00/1.00	0.29/0.75	0.12/0.12
min. 6	0.00/0.15	0.00/1.00	0.04/0.94	0.31/0.14	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 7	0.00/0.15	0.00/1.00	0.00/0.94	0.28/0.31	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 8	0.00/0.15	0.00/1.00	0.00/0.94	0.28/0.31	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 9	0.00/0.15	0.00/1.00	0.00/0.94	0.31/0.22	0.20/0.41	0.83/1.00	0.75/0.65	0.12/0.12
min. 10	0.00/0.15	0.00/1.00	0.00/0.94	0.31/0.26	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 11	0.00/0.15	0.00/1.00	0.00/0.94	0.31/0.26	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 12	0.00/0.15	0.00/1.00	0.00/0.94	0.31/0.31	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 13	0.00/0.15	0.00/1.00	0.00/0.94	0.31/0.22	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 14	0.00/0.15	0.00/1.00	0.04/0.94	0.31/0.14	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 15	0.00/0.15	0.00/1.00	0.04/0.94	0.31/0.14	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 16	0.13/0.15	0.47/1.00	0.51/0.94	0.31/0.14	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 17	0.00/0.15	0.00/1.00	0.04/0.94	0.31/0.14	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 18	0.00/0.15	0.00/1.00	0.04/0.94	0.31/0.14	0.38/0.41	1.00/0.59	0.75/0.25	0.12/0.09
min. 19	0.00/0.15	0.00/1.00	0.04/0.94	0.31/0.14	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 20	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 21	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 22	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 23	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 24	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 25	0.13/0.15	0.47/1.00	0.51/0.94	0.31/0.14	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 26	0.09/0.15	0.21/1.00	0.28/0.94	0.31/0.00	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 27	0.00/0.15	0.00/1.00	0.04/0.94	0.31/0.14	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 28	0.00/0.15	0.00/1.00	0.00/0.94	0.31/0.31	0.31/0.41	1.00/0.88	0.75/0.47	0.12/0.11
min. 29	0.00/0.15	0.00/1.00	0.00/0.94	0.31/0.22	0.38/0.41	1.00/0.59	0.75/0.25	0.12/0.09
min. 30	0.00/0.15	0.00/1.00	0.00/0.94	0.31/0.30				

The python scripts computing the Grades of Confidence given by the automaton and some relevant information used in this study are available at `https://github.com/FINDOSKDI/heartdiagnosis`.

In order to assess the usefulness of our automaton, we used the MIT/BIH Arrhythmia Database Directory [14] `https://physionet.org/physiobank/database/mitdb/`. This study includes 48 ECG recordings with a duration of 30 minutes from the Massachusetts Institute of Technology - Beth Israel Hospital arrhythmia database. All of them present some heart pathology. Specifically, 48% of the samples have been annotated in the database as representative cases of routine clinical recordings while the remaining 52% reflect uncommon cases of arrhythmias. As an example of the obtained results, in Table 1 we show minute data from each applicable age branch of the patients 100 and 104, commented in our previous work [4]. Each cell contains two numbers: the first one is the Hamacher GoC of sending an *ok* signal while the second one is the Hamacher GoC of raising an *alarm*, both referring to that minute and age/gender branch. This table clearly shows why our new approach represents a big step forward with respect to our previous work: we are able to produce and appropriately process several dozens of ok/alarm signals.

We recall that patient 100 is a 69 years old male and patient 104 is a 66 years old female. In both cases, the value of the corresponding *?checkAge* constraint is positive for four branches: ($50 \leq age \leq 59$), ($60 \leq age \leq 69$), ($70 \leq age \leq 79$) and ($age > 80$). The table is formed by four columns for patient 100 and four columns for patient 104. Each row contains the GoCs obtained in a minute. The record from patient 100 is 30 minutes long while the record from patient 104 is 29 minutes long. Let us briefly comment on the results obtained for these two patients. First, we notice that in both cases the maximum confidence is obtained in the $60 - 69$ branch. This means that there are not many RR or BPM values close to the limit of the normal range. The only case in which there is more confidence outside the $60 - 69$ age branch is in the $!ok$ results of the patient 100. We can see that the columns corresponding to the age ranges $70 - 79$ and > 80 present higher confidence in these cases. These values can indicate that the values observed from the heart of patient 100 could be normal for a much older person. This idea is reinforced by the observation that in the > 80 branch the confidence in the $!ok$ case is higher that the confidence in the $!alarm$ case. The most relevant difference we observe between these two patients is that patient 100 is outside his normal parameters in every minute, while patient 104 is showing a normal behavior most of the time, having some eventual alerts.

4 Conclusions and future work

Despite the numerous advances in healthcare, many patients do not receive a correct diagnosis. Some of these problems are provoked by either an incorrect processing of data or by wrong conclusions from relevant observations. Therefore, if data was more accurately processed and analyzed, then it would be possible to obtain improvements in this field. Our proposal goes in this line: automatically process data, extracted from electrocardiograms, to detect potential *malfunctions*. First, we have introduced a variant of finite automata where constraints indicating whether a certain transition can be performed are evaluated under a *fuzzy* point of view. We have modeled the behavior of the heart by taking into account data about the beats per minute and the duration of RR waves. In order to decide whether potential dangers have been observed, we use information about the gender and age of the patients. In the latter case, we also use a *fuzzy* approach because a patient can be classified in several age groups.

We are considering several lines of future work. First, we would like to obtain more data from patients with the aim of applying techniques, such as evolutive algorithms, swarm intelligence and neural networks, to improve the ability of our automata to detect illnesses. In cooperation with researchers in Medicine, we are working on alternative models where the *classification* of patients considers characteristics such as size/weight and medical record. Once we have more complex and complete models, we will test their suitability with alternative data sets [12].

References

1. R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

2. C. Andrés, L. Llana, and M. Núñez. Self-adaptive fuzzy-timed systems. In *13th IEEE Congress on Evolutionary Computation, CEC'11*, pages 115–122. IEEE Computer Society, 2011.
3. J. Boubeta-Puig, A. Camacho, L. Llana, and M. Núñez. A formal framework to specify and test systems with fuzzy-time information. In *14th Int. Work-Conf. on Artificial Neural Networks, IWANN'17, LNCS 10306*, pages 403–414. Springer, 2017.
4. A. Camacho, M. G. Merayo, and M. Núñez. Using fuzzy automata to diagnose and predict heart problems. In *19th IEEE Congress on Evolutionary Computation, CEC'17*, pages 846–853. IEEE Computer Society, 2017.
5. T. Chen, M. Diciolla, M. Kwiatkowska, and A. Mereacre. Quantitative verification of implantable cardiac pacemakers over hybrid heart models. *Information and Computation*, 236:87–101, 2014.
6. M. Doostfatemeh and S. C. Kremer. New directions in fuzzy automata. *International Journal of Approximate Reasoning*, 38(2):175–214, 2005.
7. C. Haarmark, C. Graff, M. Andersen, T. Hardahl, J. Struijk, E. Toft, J. Xue, G. Rowlandson, P. Hansen, and J. Kanters. Reference values of electrocardiogram repolarization variables in a healthy population. *Journal of electrocardiology*, 43(1):31–39, 2010.
8. S. Hozo, B. Djulbegovic, and I. Hozo. Estimating the mean and variance from the median, range, and the size of a sample. *BMC medical research methodology*, 5(1):1, 2005.
9. P. Hunter, A. Pullan, and B. Smaill. Modeling total heart function. *Annual review of biomedical engineering*, 5(1):147–177, 2003.
10. Z. Jiang, A.Connolly, and R. Mangharam. Using the virtual heart model to validate the mode-switch pacemaker operation. In *32nd Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society, EMBC'10*, pages 6690–6693. IEEE Computer Society, 2010.
11. Z. Khachaturian, J. Kerr, R. Kruger, and J. Schachter. A methodological note: Comparison between period and rate data in studies of cardiac function. *Psychophysiology*, 9(5):539–545, 1972.
12. M. Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013.
13. D. Méry and N. Singh. Formalization of heart models based on the conduction of electrical impulses and cellular automata. In *1st Int. Symp. on Foundations of Health Informatics Engineering and Systems, FHIES'11, LNCS 7151*. Springer, 2012.
14. G. Moody and R. Mark. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology*, 20(3):45–50, 2001.
15. G. B. Moody. RR intervals, heart rate and HRV Howto. `https://www.physionet.org/tutorials/hrv/`, 2016.
16. J. N. Mordeson and D. S. Malik. *Fuzzy automata and languages: theory and applications*. Chapman & Hall/CRC, 2002.
17. P. Rijnbeek, G. van Herpen, M. Bots, S. Man, N. Verweij, A. Hofman, H. Hillege, M. Numans, C. Swenne, J. Witteman, et al. Normal values of the electrocardiogram for ages 16–90 years. *Journal of electrocardiology*, 47(6):914–921, 2014.
18. W. G. Wee and K. S. Fu. A formulation of fuzzy automata and its application as a model of learning systems. *IEEE Transactions on Systems Science and Cybernetics*, 5(3):215–223, 1969.