A trading framework based on fuzzy Moore machines*

Iván Calvo, Mercedes G. Merayo[0000−0002−4634−4082], and Manuel Núñez[0000−0001−9808−6401]

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

Abstract. The everlasting competition between investing strategies has seen a remarkable impulse after automated trading algorithms took their place. Any failure in this kind of algorithms may end up implying huge monetary losses. Because of that, these systems may represent an important application of formal methods. Furthermore, considering the inherent uncertainty of stock markets and the usual imprecision in the definition of many investment strategies, any attempt to model these software systems is very challenging. In this paper we propose a complete framework, built upon the formalism of fuzzy automata, that can be used to define and evaluate a variety of automatic trading strategies based on the observation of candlestick patterns.

Keywords: Fuzzy automata · Automated trading · Candlestick patterns.

1 Introduction

The use of models helps to develop more reliable systems. This claim is well-known in most engineering disciplines [25]. It is interesting to realize that in Computer Science, despite the complexity of the current systems, it is not assumed that a *blueprint* should be used to guide the development process [22]. There is plenty of work in the academia introducing different approaches and techniques to support the formal development of computer systems and validate the correctness of the developed systems [11]. We think that the lack of good tools to support the theoretical approaches is an important deterrent to achieve a widespread use of formal methods in industry, although there are some remarkable experiences [3, 23, 30]. Therefore, our first aim when introducing a new formalism should be to provide tools to support its use.

There are many situations where the development of a system inherently needs to consider imprecise information. For example, this is the case if we have

* This work has been supported by the Spanish MINECO-FEDER (grant numbers DArDOS, TIN2015-65845-C3-1-R and FAME, RTI2018-093608-B-C31) and the Region of Madrid (grant number FORTE-CM, S2018/TCS-4314) co-funded by EIE Funds of the European Union.

to represent a simple system just tossing a coin. In order to formally design and analyze this kind of systems, *fuzzy logic* [34, 35] is very useful because, in addition to having an underlying mathematical theory, it has been shown that it can appropriately model systems where we need to use a certain degree of *imprecision*. We are interested in equipping a classical formalism with fuzzy logic capabilities so that we can model imprecision. Fortunately, there are many variants *fuzzifying* variants of finite automata [1, 13, 28, 29, 33]. Our previous work [6, 8, 9], which we use as initial step of the formalism presented in this paper, was built on top of these approaches and introduced a fuzzy version of classical Mealy machines. We successfully used our formalism to analyze heart data, extracted from electrocardiograms, in order to detect abnormal behaviors.

The next step in our research is to evaluate the versatility of our framework. In previous work we focused on the behaviour of medical systems. In this paper we consider a completely different field of application: we study how our framework can be used to specify a simple trading system and evaluate its suitability. To be more precise, since we do not need a formalism as expressive as the one that we used before, we will consider a fuzzy variant of Moore machines, a formalism slightly simpler than Mealy machines. We will present a simple trading model where we analyse the values of a certain stock, given as *candlesticks*, in order to detect a certain pattern. Candlesticks include four different pieces of information concerning how a certain stock was trading during a day. The *body* of the candlestick denotes the gap between the open and closing prices of the analysed stock. A green/red color denotes that the first price was lower/higher than the second one. In addition, the *shadow* of the candlestick shows the high and low prices for the session. Figure 1 shows the candlesticks corresponding to twenty sessions of the Apple stock.

Currently, there exists a rich literature explaining in detail trading strategies based on the observation of candlesticks patterns [5, 31]. In this paper, our main goal is not to specify a complex strategy but to show how our framework can encompass in a simple way a whole family of strategies, so that the user can fine-tune certain parameters according to the specific stock of interest. Specifically, we will define a framework where investment decisions will be based on the detection of *hammers*. A hammer is observed when a stock trades during a session significantly lower than its opening, but within the session is able to close near opening price. In order to consider that a candlestick has a hammer shape, it is usually assumed that the lower shadow must be at least twice the size of its body. The strategy looking for hammers considers that they mark a possible bottom in the value of a stock. More precisely, this strategy considers that the price of the stock might rise after we observe a hammer preceded by a sequence, at least three, of sessions where the value of the stock has declined.

In this paper we present a new formalism, a natural evolution and simplification of previous work, to represent fuzzy systems and apply it to specify a parametric strategy to trade a stock taking into account the occurrence of hammers in candlesticks series. Our model is parametric because it allows to trade different percentages of the total bankroll in a single operation. In the simplified version, $n = 1$, the investor buys/sell the complete bankroll in each single operation. For a general value of $n \in \mathbb{N}$, at a certain point of time the investor has, for a certain $0 \leq m \leq n$, $\frac{m}{n}$ of the bankroll invested in a stock and $\frac{n-m}{n}$ in liquidity. A single operation will buy/sell stock for $\frac{1}{n}$ of the bankroll. The framework is fully supported by a tool that allows users to process candlesticks information so that different strategies can be backtested.

The rest of the paper is structured as follows. In Section 2 we review basic concepts from fuzzy theory and introduce a variant of fuzzy automata, based on Moore machines, that is particularly well suited for the definition of trading strategies. In Section 3 we show how our formalism can represent candlesticks and patterns associated with different types of hammers. In Section 4 we present our strategy. Finally, in Section 5 we present our conclusions and sketch some lines for future work.

2 Fuzzy Moore machines

In this section we review some concepts associated with the definition of fuzzy automata and present our new formalism. The interested reader is referred to our previous work [8, 9] for more details. *Fuzzy relations* play the role of applying *soft* constraints. This is made by setting some δ, which expresses the level of tolerance that can be afforded when applying the given constraint. This means that, given some *crisp* relation, such as $x \leq y$, the fuzzy relation will *partially* hold even when $x > y$, provided that the difference between x and y is still lower than δ. More precisely, in this specific example, we define a function $\mathbb{R} \times \mathbb{R} \longrightarrow [0, 1]$ to be the fuzzy version of the $x \leq y$ relation:

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

Similarly, we can define the fuzzy analogous of the $x \geq y$ and $x = y$ relations:

$$\overline{x \geq y}^{\delta} \equiv \begin{cases} 1 & \text{if } x > y \\ \frac{\delta + x - y}{\delta} & \text{if } y \geq x \geq y - \delta \\ 0 & \text{if } y - \delta > x \end{cases}$$

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

In addition to fuzzy versions of the usual *arithmetic relations*, we need a fuzzy version of *Boolean relations* such that *partially true* statements can be joined together, forming logical conjunctions. The fuzzy analogous of the *crisp* Boolean conjunction is the concept of *triangular norms*, also known as t-norms.

Definition 1. *A t-norm is a function*

$$\triangle : [0,1] \times [0,1] \longrightarrow [0,1]$$

satisfying the following properties:

- *Commutativity:* $x \triangle y = y \triangle x$ *for all* $x, y \in [0,1]$.
- *Associativity:* $(x \triangle y) \triangle z = x \triangle (y \triangle z)$ *for all* $x, y, z \in [0,1]$.
- *Identity:* $1 \triangle a = a$ *for all* $a \in [0,1]$.
- *Monotonicity: if* $x_1 \leq y_1$ *and* $x_2 \leq y_2$ *then we have* $x_1 \triangle x_2 \leq y_1 \triangle y_2$ *for all* $x_1, x_2, y_1, y_2 \in [0,1]$.

In this paper we will consider only the Hamacher t-norm, defined as $(x,y) \mapsto \frac{xy}{x+y-xy}$. *This t-norm happens to be strictly monotonic, which means that the* $\leq$ *signs are actually strict* $<$ *signs in the monotonicity condition.*

The goal of introducing these notions is to finally have a *fuzzy* way to impose constraints on the behaviour of the system under certain inputs. These *fuzzy constraints* are defined as follows:

Definition 2. *A fuzzy constraint is a formula consisting of fuzzy relations, possibly combined with t-norms, which may contain free variables and constant values within the* $[0,1]$ *interval.*

Next, we will discuss the way in which we decided to *fuzzify* the usual notions brought from automata theory. In order to fix the alphabet, a distinction between *input* and *output* actions was used in our previous work. This distinction is useful to specify the way in which certain system outputs correspond to the given inputs. The way in which outputs are *raised* upon traversing some particular transition between states bears a slight resemblance with classical Mealy machines. It is at this point where we diverge from previous work: while we still want the model to serve as a specification of the correspondence between *inputs* and *outputs*, we will make no distinction between the symbols of the alphabet. Instead of that, each *state* of the automaton will be associated with a particular output value. This value will be raised every time that this state is traversed. With this distinction, the proposed formalism can be seen as a *fuzzified* version of Moore machines. Therefore, we will define our alphabet to be the set of *events* in our model.

Definition 3. *An* event *is a collection of variables, which are instantiated by the environment according to some predefined set of observations and expressions, together with a fuzzy constraint whose free variables are those variables. We identify each event with a name preceded by the # symbol.*

Let us illustrate this concept with an example that will be later used in the proposed model.

Example 1. We will denote by #opportunityCost the event consisting in an empty set of variables together with the constant constraint whose truth value is always 0.7.

Now, we have the elements to define our version of *fuzzy Moore machine.*

Definition 4. *A fuzzy Moore machine is a tuple*

$$(S, \mathcal{E}, \triangle, f, s_0, T)$$

whose components denote:

- S *is a finite set of states.*
- $\mathcal{E}$ *is a finite set of events.*
- $\triangle$ *is a strictly monotonic t-norm.*
- f *is a function taking a state from* S *and returning its associated output value.*
- s_0 *is the initial state.*
- $T \subseteq S \times \mathcal{E} \times S$ *is the set of transitions.*

The semantics of these machines is quite intuitive. The action of the environment must be formalized as a sequence of functions $\mathcal{O}_1, \ldots, \mathcal{O}_n$ mapping each expression to its instantiation value at the n-th time step. Having defined this, the satisfaction degree of an event #ev at a time step n, denoted by $\mu_n(\#\text{ev})$, will be defined in the natural way (the explicit formal definition can be found in our previous work [8]).

Finally, let us define some notation to establish the correspondence between observed events and produced outputs.

Definition 5. *Let* $F = (S, \mathcal{E}, \triangle, f, s_0, T)$ *be a fuzzy Moore machine,* $s_1, \cdots, s_n$ *be a sequence of states in* S *and* $\#ev_1, \cdots, \#ev_n$ *be a sequence of events. If for all* $1 \leq k \leq n$ *we have that* $\epsilon_k := \mu_k(\#ev_k) > 0$ *and* $(s_{k-1}, \#ev_k, s_k) \in T$ *then we write*

$$\#ev_1, \cdots, \#ev_n \Rightarrow_\epsilon f(s_1), \cdots, f(s_n)$$

where the value of ϵ *is given by* $\epsilon_1 \triangle \cdots \triangle \epsilon_n$.

3 Modeling candlestick patterns

In this section we will show how fuzzy constraints can be used to give a formal definition of various candlesticks patterns. Since the formalism is open to different ways in which the environment may instantiate the parameters of each individual observation, in this section we will review the basic elements present in our trading environment and discuss the precise definition of the observations that will be later used to define our model in Section 4. The definition of the observations is expressed in a formal language based on arithmetic expressions. We think that this formal language is expressive enough to define almost any possible trading strategy.

First, we will assume that we can obtain the *open*, *close*, *high* and *low* value of the stock under consideration on any day prior to the current, n-th, day. Note that this information is indeed freely available from multiple sources (including the Nasdaq website and The Wall Street Journal).

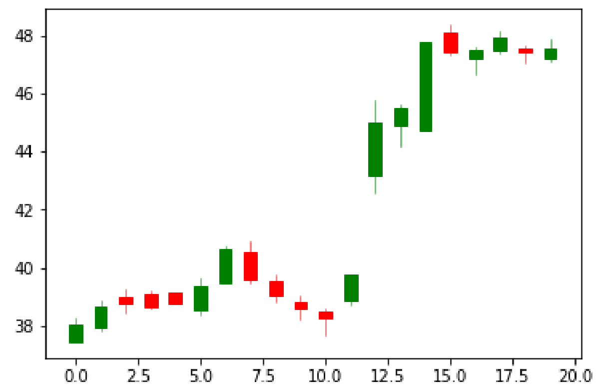

Fig. 1. Candlesticks chart from Apple.

Definition 6. *Let k be a natural number and let e be an expression of the form $(open|close|high|low) \sim k$. We denote by $\mathcal{O}_n(e)$ the corresponding value at the day $n - k$.*

We can extend the $\mathcal{O}$ notation to any arithmetical expression in a compositional way. Let e, e_1, e_2 be expressions of the previous form, $\neg$ be an unary arithmetic operator and $\bigcirc$ be a binary operator. Then we have

$$\mathcal{O}_n(\neg e) = \neg \mathcal{O}_n(e) \qquad \text{and} \qquad \mathcal{O}_n(e_1 \bigcirc e_2) = \mathcal{O}_n(e_1) \bigcirc \mathcal{O}_n(e_2)$$

Similarly, we will also consider real valued constant expressions. Finally, We define an observation to be any composition of these expressions. □

The expressive power of these formulas can be shown with a couple of illustrative examples.

Example 2. The difference between the value of the asset at the market close of the last two days can be expressed with the following formula:

$$(\text{close} \sim 1) - (\text{close} \sim 2)$$

A moving average of the midpoint of the range of prices of each time interval can be expressed as follows:

$$\frac{\frac{\text{low}\sim 1+\text{high}\sim 1}{2} + \frac{\text{low}\sim 2+\text{high}\sim 2}{2} + \frac{\text{low}\sim 3+\text{high}\sim 3}{2}}{3}$$

In order to define our strategy, we need to provide a formal definition of the *shape* of a hammer pattern and the one corresponding to a doji pattern. Since both patterns are based on the relative size of the components of the candlestick, we will define them by using fuzzy constraints defined over relative size observations.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
relopen	0.01	0.11	0.67	0.89	1.00	0.13	0.02	0.73	0.77	0.70	0.91	0.13	0.20	0.49	0.00	0.76	0.56	0.16	0.80	0.17
relclose	0.70	0.83	0.37	0.15	0.09	0.76	0.91	0.11	0.24	0.45	0.69	0.99	0.76	0.92	1.00	0.11	0.86	0.75	0.60	0.62

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
hammer	0.00	0.01	0.09	0.00	0.00	0.04	0.00	0.00	0.00	0.21	0.55	0.04	0.13	0.52	0.00	0.00	0.56	0.08	0.43	0.08
doji	0.01	0.00	0.56	0.00	0.00	0.09	0.00	0.11	0.23	0.65	0.68	0.00	0.20	0.38	0.00	0.07	0.56	0.15	0.71	0.35

Table 1. Relopen and relclose corresponding to Figure 1 (top) and satisfaction degree of hammer and doji patterns in candlesticks corresponding to Figure 1 (bottom)

Definition 7. *We define the* ***relopen*** *observation by the formula:*

$$\frac{open \sim 1 - low \sim 1}{high \sim 1 - low \sim 1}$$

We define the ***relclose*** *observation by the formula:*

$$\frac{close \sim 1 - low \sim 1}{high \sim 1 - low \sim 1}$$

In Table 1 (top) we see the **relopen** and **relclose** values corresponding to the candlesticks shown in Figure 1.

Once these relative size observations have been defined, we can give a *fuzzy* definition of *hammer* and *doji* patterns.

Definition 8. *The grade of confidence associated with the event of observing a* ***hammer*** *is given by the fuzzy constraint*

$$\overline{relopen \geq 0.8}^{0.7} \bigtriangleup \overline{relclose = 1}^{0.7}$$

Similarly, the grade of confidence associated with the event of observing a doji is given by the fuzzy constraint

$$\overline{relopen = relclose}^{0.7}$$

In Table 1 (bottom) we show the grade of confidence of each candlestick shown in Figure 1 being a *hammer* and a *doji*.

4 Strategy definition and evaluation

In this section we present a basic trading strategy, using our formalism, and show how a candlestick pattern recognition approach can be defined. In figure 2 we show the outline of our model, where some parameters of the corresponding automaton have been removed for the sake of clarity. First, we would like to mention that this model has only three states. Note that the numbers inside the states do not represent the name of the state but its associated output value. This means that a model might have different states associated to the same output value, which can be useful when modelling systems in which there are situations that are fundamentally different but may imply the same outcome. The three possible outcomes of our model, that is, the values of the set $\{0, \frac{1}{2}, 1\}$, encode the proportion of money invested in the stock under consideration. For example, an outcome of 1 at a given time indicates that we can invest all our available funds in the corresponding stock. On the contrary, an outcome of 0 indicates that we have to sell all our investment in the considered stock. The implications of a $\frac{1}{2}$ outcome are a little bit more subtle. Since the value of the stocks is asked to be equal to that of the available cash, a raise in the price of the asset can imply a sell order, while a fall in its price should imply a buy order. The initial state of our model is the one associated with a 0 outcome, meaning that the strategy starts with the account owning no stocks and, therefore, full liquidity. In this simple model, we assume that we only sell stocks that have been previously bought, ruling out the possibility of *short selling*. In order to illustrate our model, in this paper we have considered as stock of interest Apple Inc. (denoted by $AAPL$).

As mentioned in Section 3, the key element of our strategy is the observation of *hammer* patterns. Since an observation of a *hammer* after a downtrend may predict the reversal of the trend, the strategy will be based on waiting until one of these patterns appears. If a *hammer* appears, then there are two situations: the trend before it may be falling or not. In the first case, we want to buy as many stocks as possible, hence a transition moving to the 1 state is triggered. In the second case, we have to balance whether the risk of buying after no downtrend is too high or not. If it is not, then only a half of the available cash will be spent, so that a transition moving to the $\frac{1}{2}$ state will be triggered.

The way in which our formalism manages these decisions is, as explained in Section 2, to assign a confidence value to each possible sequence of outputs. After each time step, the current state associated with the highest grade of confidence is the one producing the output. In this case, that output is the target amount of capital that should be invested in the stock.

The described strategy has been successfully implemented and its python source code is available on github (https://github.com/FINDOSKDI/trading). The source code is an adaptation of the one from our previous tool, AUNTY [7], which was used to represent and execute models of our other version of this formalism, based on fuzzy Mealy machines. The code of the tool, without its graphical user interface, was inserted into a *jupyter notebook*, as shown in Figure 4, and adapted in order for it to be able to connect with *OHLC* datasets contained in *pandas DataFrames*.

It is worth to mention that our framework is integrated with quantopian, as shown in Figure 3. After each trading interval, the quantopian algorithm API provides a DataFrame with historical data corresponding to the previous intervals. This DataFrame is then fed to the automaton, which establishes the most suitable execution corresponding to that DataFrame and all previous ones. Then, the final state of that execution raises its corresponding output value, which signals the desired trading order to the quantopian algorithm API.

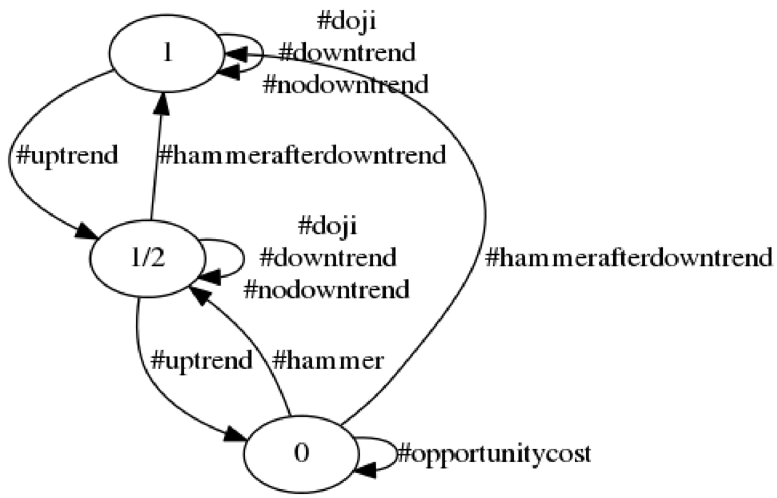

Fig. 2. Topology of our model.

5 Conclusions and future work

In this paper we have used a variant of fuzzy automata to represent a simple trading system where decisions about trading are *fuzzy*. The trading strategy is based on the observation of *hammers*, an indicator of a change of trend. Our strategy is parameterised so that the user can choose the portion of the available bankroll and portfolio that can be traded in a single operation. In order to ensure the usability of our model, the system is fully implemented, including the connection with quantopian to incorporate real data about candlesticks, and freely available.

This is only our initial step in this line of work and, therefore, there are many potential ways to continue our research in this field. A first group of improvements can target the usefulness of the model by incorporating more complicated features. Among them, we would like to introduce ways to calibrate beta, consider variations than can balance the investment between different stocks, strategies to optimize the conformation of the portfolio and introduce hard constraints regarding stop losses. In order to implement these improvements, we will consider as initial steps previous work at the implementation level, where different strategies are implemented in python [20], and at the specification level, using other appropriate high level formalisms [2, 4, 24].

A second group of improvements consider a more focused, and formal, analysis of the information. Specifically, in this paper we have presented a formalism to represent trading systems but we have not considered approaches to formally analyse these systems and to decide whether a certain *real* system (in this case, a certain stock) follows the behaviour specified by the system. First, we would like to take into account time and probabilities in a more formal approach.

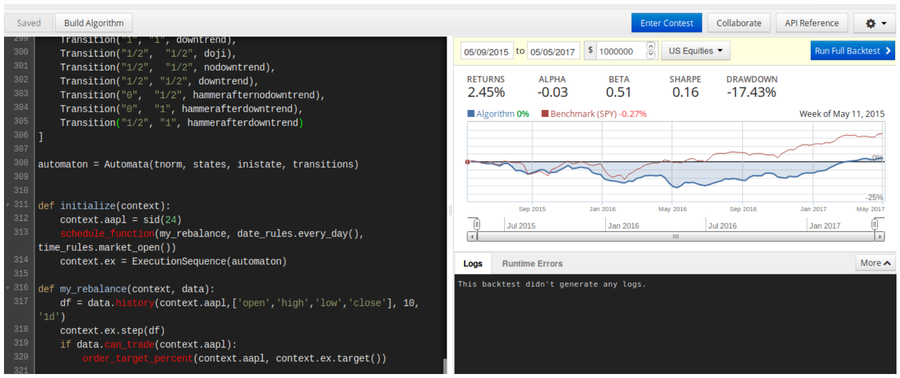

Fig. 3. Formalism integration with quantopian.

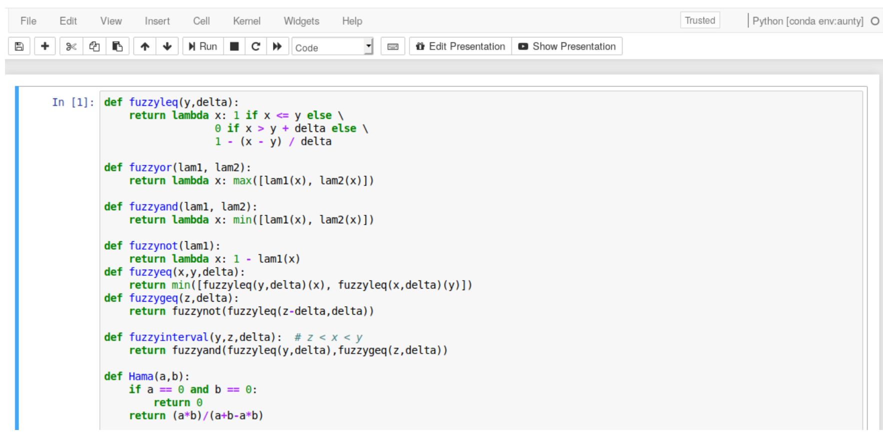

Fig. 4. Fragment of code from our framework, being integrated in a jupyter notebook.

In particular, we would like to establish whether the quotes of a certain stock *timely/probabilistically conforms* to a trading strategy [19, 32]. In this line of work, we would like to use formal testing approaches to analyse the relation between a stock and a trading strategy. Interestingly enough, we cannot use classical testing approaches where the tester interacts with the system because they would be unrealistic: except in very restricted markets, a typical trader cannot strongly influence a stock. Therefore, we need to use a *passive* approach where the tester observes the system without interacting with it [17, 26, 27]. Similarly, we need to use approaches to test in the distributed architecture [16, 18] because real-time data can be collected from sources distributed along different physical locations. We would also like to use recent work on testing from FSMs using Information Theory [21] to evaluate the likelihood of fault masking in the studied strategies. Finally, we would like to consider recent work on mutation testing [10, 12, 14, 15] to assess the validity of a given strategy when compared to likely worse ones given by *mutants*.

References

1. 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.
2. A. Bernal, M. E. Cambronero, A. Núñez, P. C. Cañizares, and V. Valero. Improving cloud architectures using UML profiles and M2T transformation techniques. *The Journal of Supercomputing*, 75(12):8012–8058, 2019.
3. S. Bonfanti, A. Gargantini, and A. Mashkoor. A systematic literature review of the use of formal methods in medical software systems. *Journal of Software: Evolution and Process*, 30(5):e1943, 2018.
4. J. Boubeta-Puig, Gregorio Díaz, H. Macià, V. Valero, and G. Ortiz. MEdit4CEP-CPN: An approach for complex event processing modeling by prioritized colored Petri nets. *Information Systems*, 81:267–289, 2019.
5. T. N. Bulkowski. *Encyclopedia of Candlestick Charts*. Wiley, 2008.
6. I. Calvo, M. G. Merayo, and M. Núñez. An improved and tool-supported fuzzy automata framework to analyze heart data. In *10th Asian Conference on Intelligent Information and Database Systems, ACIIDS'18, LNAI 10751*, pages 694–704. Springer, 2018.
7. I. Calvo, M. G. Merayo, and M. Núñez. `AUNTY`: A tool to automatically analyze data using fuzzy automata. In *3rd Int. Conf. on Computational Intelligence and Applications, ICCIA'18*, pages 102–106. IEEE Computer Society, 2018.
8. I. Calvo, M. G. Merayo, and M. Núñez. A methodology to analyze heart data using fuzzy automata. *Journal of Intelligent & Fuzzy Systems (to appear)*, 2019.
9. I. Calvo, M. G. Merayo, M. Núñez, and F. Palomo-Lozano. Conformance relations for fuzzy automata. In *15th Int. Work-Conf. on Artificial Neural Networks, IWANN'19, LNCS 11506*, pages 753–765. Springer, 2019.
10. P. C. Cañizares, A. Núñez, and M. G. Merayo. Mutomvo: Mutation testing framework for simulated cloud and HPC environments. *Journal of Systems and Software*, 143:187–207, 2018.
11. A. R. Cavalli, T. Higashino, and M. Núñez. A survey on formal active and passive testing with applications to the cloud. *Annales of Telecommunications*, 70(3-4):85–93, 2015.
12. P. Delgado-Pérez, Louis M. Rose, and I. Medina-Bulo. Coverage-based quality metric of mutation operators for test suite improvement. *Software Quality Journal*, 27(2):823–859, 2019.
13. M. Doostfatemeh and S. C. Kremer. New directions in fuzzy automata. *International Journal of Approximate Reasoning*, 38(2):175–214, 2005.
14. P. Gómez-Abajo, E. Guerra, J. de Lara, and M. G. Merayo. A tool for domain-independent model mutation. *Science of Computer Programming*, 163:85–92, 2018.
15. L. Gutiérrez-Madroñal, A. García-Domínguez, and I. Medina-Bulo. Evolutionary mutation testing for IoT with recorded and generated events. *Software - Practice & Experience*, 49(4):640–672, 2019.
16. R. M. Hierons, M. G. Merayo, and M. Núñez. Controllable test cases for the distributed test architecture. In *6th Int. Symposium on Automated Technology for Verification and Analysis, ATVA'08, LNCS 5311*, pages 201–215. Springer, 2008.

17. R. M. Hierons, M. G. Merayo, and M. Núñez. An extended framework for passive asynchronous testing. *Journal of Logical and Algebraic Methods in Programming*, 86(1):408–424, 2017.
18. R. M. Hierons, M. G. Merayo, and M. Núñez. Bounded reordering in the distributed test architecture. *IEEE Transactions on Reliability*, 67(2):522–537, 2018.
19. R. M. Hierons and M. Núñez. Implementation relations and probabilistic schedulers in the distributed test architecture. *Journal of Systems and Software*, 132:319–335, 2017.
20. Y. Hilpisch. *Python for Finance: Analyze Big Financial Data.* O'Reilly Media, 2014.
21. A. Ibias, R. M. Hierons, and M. Núñez. Using Squeeziness to test component-based systems defined as Finite State Machines. *Information & Software Technology*, 112:132–147, 2019.
22. L. Lamport. Who builds a house without drawing blueprints? *Communications of the ACM*, 58(4):38–41, 2015.
23. T. Lecomte, D. Déharbe, É. Prun, and E. Mottin. Applying a formal method in industry: A 25-year trajectory. In *20th Brazilian Symposium on Formal Methods: Foundations and Applications, SBMF'17, LNCS 10623*, pages 70–87. Springer, 2017.
24. N. López, M. Núñez, I. Rodríguez, and F. Rubio. A formal framework for e-barter based on microeconomic theory and process algebras. In *Innovative Internet Computer Systems, LNCS 2346*, pages 217–228. Springer, 2002.
25. L. Magnani and T. Bertolotti, editors. *Handbook of Model-Based Science.* Springer, 2017.
26. M. G. Merayo, R. M. Hierons, and M. Núñez. Passive testing with asynchronous communications and timestamps. *Distributed Computing*, 31(5):327–342, 2018.
27. M. G. Merayo, R. M. Hierons, and M. Núñez. A tool supported methodology to passively test asynchronous systems with multiple users. *Information & Software Technology*, 104:162–178, 2018.
28. J. N. Mordeson and D. S. Malik. *Fuzzy automata and languages: theory and applications.* Chapman & Hall/CRC, 2002.
29. M. Mraz, I. Lapanja, N. Zimic, and J. Virant. Fuzzy numbers as inputs to fuzzy automata. In *18th Int. Conf. of the North American Fuzzy Information Processing Society, NAFIPS'99*, pages 453–456. IEEE Computer Society, 1999.
30. C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, and M. Deardeuff. How Amazon web services uses formal methods. *Communications of the ACM*, 58(4):66–73, 2015.
31. S. Nison. *Japanese Candlestick Charting Techniques: A Contemporary Guide to the Ancient Investment Techniques of the Far East.* Prentice Hall, 2nd edition, 2001.
32. M. Núñez and I. Rodríguez. Conformance testing relations for timed systems. In *5th Int. Workshop on Formal Approaches to Software Testing, FATES'05, LNCS 3997*, pages 103–117. Springer, 2006.
33. 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.
34. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
35. L. A. Zadeh. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems.* Advances in Fuzzy Systems - Applications and Theory: vol. 6. World Scientific Press, 1996.