International Journal of Scientific World, 3 (1) (2015) 113-123
www.sciencepubco.com/index.php/IJSW
©Science Publishing Corporation
doi: 10.14419/ijsw.v3i1.4356
Research Paper

Use of Büchi automata and randomness for the description of biological processes

Konstantinos Giannakis*,Theodore Andronikos

Department of Informatics, Ionian University, Corfu, Greece
**Corresponding author E-mail: kgiann@ionio.gr*

Abstract

The main topic of this study is the modelling and verification of biological systems using ω-automata. This work focuses on the protein folding problem and the infinite behaviour it features in many cases. Specifically, stochastic computational models with infinite input are used in this paper's approach and indicative aspects of a biological problem are presented using both ω-automata and probabilistic Büchi automata (PBAs), making a novel attempt to establish their use in reasoning about biological processes. Necessary preliminary definitions and background towards this direction are provided. Finally, the pros and cons of each model are shown through examples. Overall, this work contributes by combining for the very first time PBAs with real biological mechanisms, with indicative examples, both with PBAs and NBAs (non-deterministic Büchi automata).

Keywords: *ω-Automata; Concurrent Systems; Infinite Sequences; Probabilistic Computation; Protein Folding*

1. Introduction

It is a fact that many abstract mathematical models of computation are studied in the theory of computation. Some of them include mechanisms, such as state machines (e.g. Turing machines, automata, etc.). Computer scientists study these abstract machines because of their simplicity in formulation and analysis, aiming to prove theories and results. Besides, any problem that can be solved (or decided) by a Turing machine can also be solved by a computer.

Apart from its physical description, computational theory deals with issues of computability and complexity of different systems, as well as of various natural and biological processes. In this paper we assume that the reader is familiar with basic notions from the theory of computation. Nevertheless, for an overview in automata theory and theory of computation we refer the reader to [1, 2, 3]. In general, variants of probabilistic automata are frequently used in various domains such as pattern recognition, signal processing etc.

Automata with infinite input are applicable in many fields, mainly in the verification of concurrent systems and reasoning about infinite systems (infinite games, biological systems etc.). In addition, decision problems for certain logics use notions of this type of automata. Similarly to their classic counterparts, automata on infinite sequences can be categorized into different classes according to either their branching mode or the acceptance condition under which they operate.

Verification of reactive and concurrent systems or reasoning about biological processes [4] are some of the research regions where one can meet these mechanisms. In biomolecular level, streams of events are omnipresent and they are vital for the well-being of the whole organism. Our world is full of example systems with non-terminating behaviour, not only in the physical level but also in the digital one. Life sciences study them trying to shed more and more light on various aspects of these phenomena. For example, such systems may be operating systems, biological processes etc. These systems have to undergo a computation process, either for their behaviour modelling or the use of them as computation tools themselves [5, 6].

In this work we try to typically formulate the intrinsically infinitary properties of specific biological structures. At first, we make an introduction to our concept, presenting the basic notions and definitions. We recall notions from the standard automata theory and then we proceed to some more advanced concepts. In short, we give the definitions of the deterministic (and non-deterministic) automaton and the corresponding ω-automaton. We explicitly state the most known acceptance conditions regarding the ω-automata and we primarily focus on the Büchi condition. Then we present the procedure proteins follow to fold and create their structure in an abstract level, concentrating to the formation of their primary and secondary structure.

This article begins with a review of the most relative literature, then in Section 3 the definitions needed for smooth transition to our main contribution are presented, which will be described and discussed thoroughly in Section 5. Next, Section 4 is devoted to the protein folding problem and its analogy with a computation concept. Discussion on our proposed model will be given in Section 5 where we present our approach in the modelling and verification of a biological problem in a novel way, using notions from the stochastic computation and ω-automata. Finally, in the last section we try to stimulate future work towards this direction with thoughts and conclusions derived from our approach.

2. Related work

Finite automata with infinite input is not a new concept. On the contrary, the first results were reported during the 1960s by notable researchers (Büchi, Rabin, McNaughton and others). In their works, especially in [7, 8, 9, 10], the connection between automata theory and logic emerged with plenty of surprising and useful results. Specifically, Büchi showed the equivalence between ω-automata and the monadic second-order logic of one successor (M1S in short) [7, 8]. These studies gave birth to a wealth of results and led to the formation of scientific branches, such as model checking, program verification etc. (e.g. see [11]).

Büchi automata were proposed in [7, 8] and a fine overview of them can be found in [12, 13, 14, 15, 16] by Thomas and Thomas et al. In [7] Büchi acceptance condition is discussed, while later, Muller and Rabin acceptance conditions (and the so-named ω-automata) were proposed. In [17] McNaughton showed that the above models are equivalent among themselves, both under the deterministic mode and the non-deterministic one and they all recognize the same class of languages, namely the ω-regular languages. Exception to this rule constitute the deterministic Büchi automata that are less powerful since there are ω-regular languages that are not recognized by them.

Vardi through [18, 19, 20, 21] has also some notable results. In [12, 13] there are elegant presentations of the main results on infinite computation and ω-automata, with additional results and theorems. The majority of these results lays out of the scope of our work, although anyone interested in the subject is encouraged to go over them. Besides these traditional uses of automata with infinite input, in [22] there is a novel use of Büchi automata in the querying of Linked Data (the hint in this work is that the authors took advantage of the constantly expanding web of linked data that shares a lot of similarities with infinite inputs)

As far as the probabilistic variants of the simple finite automata are concerned, they were introduced in [23], whereas [24] contains an extensive overview of them. Probabilistic automated verification is also discussed in details in [25, 26, 27]. A PFA (abbreviation for probabilistic finite automaton) can be seen as a normal Markov chain, having accepting behaviour rather than just transition-attached probabilities. Obviously, the transition matrix in the case of a simple PFA is basically a stochastic matrix. More recently, in [28] Gimbert et al. presented a solid review of open problems on probabilistic automata regarding their power and decidability.

The past decade, Baier et al. proposed the probabilistic version of Büchi automata, investigating in depth problems concerning the efficiency, expressiveness and decidability of these models of computation, producing prominent results and leaving ample space for future work on relevant logic issues and applications of them in both theoretic and handy implementations [29, 30, 31, 32, 33, 34]. They showed that the accepted language of a probabilistic ω-automaton can be defined by assigning a condition on the acceptance probability for the input words, providing three distinct probability semantics. In additions, they proved that probabilistic automata with Büchi acceptance are more expressive than non-deterministic ω-automata. Recently, Weidner in [35, 36] proposed a corresponding logic to the probabilistic extension of Büchi automata.

The fact that biological functionalities and biological systems have inherited infinite behaviour [37, 38] indicates them as ideal candidates for use and test of the theories and methodologies regarding ω-automata and infinite computation. This observation may have a two-fold meaning. On one hand, it could mean that particular biological systems can act as computing devices or on the other hand, that classic infinite computational models, such as Büchi automata, can be applied to describe and model these bio-systems. For example, in [39], Jha et al. offered a novel algorithm for statistical Model Checking using Bayesian Sequential Hypothesis Testing, with many advantages over other methods from the literature.

In [40], Freund et al. extended the well-established P systems [41] to act as devices with infinite sequence inputs. In [42] Petre followed the same rationale to introduce infinity in the molecular automata [43], a step to the DNA and molecular computing [5, 6]. In particular, Petre modelled the input of the Watson-Crick automaton as a DNA sequence with infinite length, in contrast to the standard variants that accept (or reject) words or objects of finite length. A variant of automata (specifically of P automata) was used to describe an actual biological model of mitochondrial fusion was described in [44].

In [45] Li and Daggett discussed the difficulties of modelling and controlling the folding procedure providing indicative simulation results. In particular, they highlighted the use of transition states in modelling protein folding/unfolding. Protein folding and protein architecture are covered in [46, 47, 48, 49, 50], as well. Shen et al. introduced SABIC, a novel method for protein structure alignment based on the internal coordinates (i.e. bond lengths, bond angles and torsion angles) of structure representation with prosperous experimental results on benchmark datasets [51]. Structural genomics is a term describing the systematic attempt to determine the structures of proteins representing every possible fold [52, 53].

The protein folding procedure can be simulated using several methods, such as molecular dynamics and Monte Carlo simulations [54]. In [55] there is a fine review on the use of stochastic hidden Markov processes in bioinformatics. Finally, more recently in [56], an alternative feature selection based method was proposed to predict protein structural classes.

In our work we describe and characterize infinitary properties of biological systems. The main contribution of this paper lies on the novel description of the protein folding process and structure using notions from automata theory and particularly standard and probabilistic Büchi automata along with ω-regular expressions. We present our proposed framework using detailed examples that show the expressing power of probabilistic Büchi automata over the standard ones.

3. Preliminary definitions

In this Section we provide the necessary definitions thar are vital for the comprehension of our proposed approach. At first, we will briefly recall the definitions of the simple, standard automata (deterministic and non-deterministic automata, DFA and NFA respectively). Then, the probabilistic extension of them follows. Afterwards, it is turn of the ω-automata and the acceptance conditions that govern them, focusing on the Büchi condition. We mention the rest of the known acceptance modes to complete the description. Later, we provide the definitions of the probabilistic Büchi automata.

Definition 3.1 *In short, a deterministic finite-state automaton (DFA) is a tuple* $(Q, \Sigma, \delta, q_0, F)$ *where*

1. Q *is a finite set of states,*
2. Σ *is a finite set of input symbols (alphabet),*
3. $\delta : Q \times \Sigma \longrightarrow Q$ *is the transition function,*
4. $q_0 \in Q$ *represents the initial state and*
5. $F \subseteq Q$ *is the set of accepting states.*

In the same manner one can define the non-deterministic finite automata (NFA), simply by altering the branching mode. In this case we have a different transition function; the new one is the $\delta : Q \times \Sigma \longrightarrow P(Q)$, where $P(Q)$ is the powerset of Q. DFA and NFA are completely equivalent in expressiveness [57].

The definition of the probabilistic finite automaton (PFA) is quite similar to the NFAs except from the fact that transitions are weighted with a probability measure.

Definition 3.2 *A probabilistic finite automaton (PFA) is a tuple* $(Q, \Sigma, \delta, q_0, F)$ *where*

1. Q *is a finite set of states,*

2. *Σ is the alphabet,*

3. *$\delta : Q \times \Sigma \longrightarrow D(Q)$ is the transition function, where $D(Q)$ is the set of all the probability distributions of the state set Q,*

4. *$q_0 \in Q$ represents the initial state and*

5. *$F \subseteq Q$ is the set of accepting states.*

Just like every stochastic system, the sum of the probabilities for every outgoing transition of every read symbol α must be equal to 1, that is

$$\sum_{i=1}^{|Q|} p_i(q,\alpha) = 1$$

and it is obvious that every probability p_i takes values from the set [0,1],

$$\delta(q,\alpha,q\text{'}) \in [0,1], \text{ for every } q \in Q \text{ and } \alpha \in \Sigma.$$

The set of languages recognized by probabilistic automata is called stochastic languages, having as a subset the class of the regular languages. Moreover, in the case of the probabilistic computation one has to define the probability semantics under which the automaton accepts or rejects according to the sum of the probabilities of the computation. More about the probabilities semantics will be mentioned in the Section where our main contribution lies (i.e. Section 5). The class of languages accepted by PFAs can be defined as

$$L_A(\lambda) = \{\ w \in A^* \mid P_A(w) \geq \lambda\ \},$$

where $0 \leq \lambda \leq 1$ is the probability cut-point [23].

An ω-automaton is an extension of the NFA running on infinite, rather than finite, input strings [13, 12, 15, 21]. Such automata do not terminate their computation, which is obvious since they read infinite input sequences. Thus, they possess a variety of acceptance conditions besides a set of accepting states. For simplicity reasons we provide the definition of the non-deterministic ω-automaton, since its difference to the deterministic version is akin to the difference between NFAs and DFAs.

Definition 3.3 *An ω-automaton is a tuple (Q, Σ, δ, q_0, Acc) where*

1. *Q is a finite set of states,*

2. *Σ is the input alphabet,*

3. *$\delta : Q \times \Sigma \longrightarrow P(Q)$, where $P(Q)$ is the powerset of Q,*

4. *$q_0 \in Q$ stands for the initial state and*

5. *Acc is an acceptance condition.*

The acceptance condition *Acc* declares which of the infinite runs are accepted by the state-machine. The most common and well-established of them are (see in [13]):

Büchi condition Automaton accepts the runs ρ for which $\text{In}(\rho) \cap F \neq \emptyset$ for a set $F \subseteq Q$, meaning that the states in F appear infinitely often during the run ρ.

Streett condition An infinite run $\rho = q_0q_1q_2\ldots$ is called Streett accepting if for each $i \in \{1, \ldots, i\}$ we have: $\text{In}(\rho) \cap F_i = \emptyset \vee \text{In}(\rho) \cap E_i \neq \emptyset$, with F_i, $E_i \in Q$.

Rabin condition the dual of the Street condition [9, 10].

Muller condition Automaton accepts the runs ρ for which $\text{In}(\rho) \in F$, that is the set of states occurring infinitely often in the run ρ forms a set in F.

A Büchi automaton is an ω-automaton under the Büchi acceptance condition. Thus, it accepts an infinite input sequence if and only if there exists a run of the automaton that visits at least one of the final states infinitely often. Non-deterministic Büchi automata recognize the ω-regular languages, whereas the same class is, also, recognized by Muller, Rabin and Streett automata, as well.

Probabilistic Büchi automata (PBA) have a syntax similar to the PFAs. In particular, they are finite-state automata where for each state q and each input symbol of an alphabet, a probability distribution specifies the probabilities for the successor states. It is obvious that they are governed by an acceptance condition like standard ω-automata. Their definition follows [30, 31].

Definition 3.4 *A probabilistic Büchi automaton (PBA) is a tuple (Q, Σ, δ, q_0, F) where*

1. *Q is a finite set of states,*
2. *Σ is the automaton's alphabet,*
3. *$\delta : Q \times \Sigma \longrightarrow D(Q)$ is the transition function, where $D(Q)$ is the set of all the probability distributions of the set of states Q,*
4. *μ represents the initial distribution for the starting state,*
5. *$F \subseteq Q$ is the set of accepting states.*

The acceptance condition in this case is the Büchi condition. Like in the definition of the PFAs, the standard probability laws apply for PBAs ($\sum_{i=1}^{|Q|} p_i(q,\alpha) = 1$ and $\delta(q,\alpha,q') \in [0,1]$).

4. The protein folding problem

In this part of our work we briefly discuss some biological phenomena with infinite traces, such as the protein folding mechanism and the mitochondria life-cycle, which can be characterized and described with elements of infinite computation (e.g. their population stability mechanisms). Abnormalities in these functions cause or boost potential diseases [58]. Proteins are biological molecules assembled from long chains of amino acids and each one of them has its own unique amino acid sequence. Proteins are responsible for a lot of vital biological duties, mainly acting as enzymes that catalyse biochemical reactions and assisting metabolism. In addition, they assist the DNA replication, molecules transportation etc.

Most proteins fold into 3-dimensional structures, either through their amino acids or by requiring molecular chaperones. There are 4 consecutive levels of protein's structures [46, 48, 50]:

1. Primary structure (the sequence of amino acids).
2. Secondary structure (a list of bases which are paired in a nucleic acid molecule, a vital consideration in the design of nucleic acid structures for DNA nanotechnology and DNA computing).
3. Tertiary structure (the overall shape of a protein molecule).
4. Quaternary structure (the structure formed by several protein molecules combined).

Protein folding is the process through which proteins reach to their final three-dimensional structure [46, 47, 48, 49, 50]. Proteins begin their folding session sometimes spontaneously and some other with enzymatic assistance by chaperones. Theoretically there is an infinite number of proteins that can be formed, each with a different sequence of amino acids. Nevertheless, only some of them can be form correct and functional structures.

Protein turnover is a term describing the equilibrium between protein synthesis and protein degradation. When this balance is violated, abnormalities and disfunctionalities appear with undesired effect for the organism. Formed proteins only "live" for a specific timeslot and afterwards they get degraded and recycled by the cell's mechanism of protein turnover. Proteins' life-span varies from a only some minutes to several years [58]. Unsuccessful folding, along with the produced misfolded proteins, is a factor to blame for several neurodegenerative diseases, allergies and other health problems [59, 60, 61].

Another indicative example of biological process for which there is a need for reasoning in infinite horizon is the fusion/fission mechanism of mitochondria [62, 63, 58]. Mitochondria population inside a cell has to remain stable, so as neither fusion rates nor fission rates overcome one another, otherwise abnormalities and diseases are caused [64]. Mitochondria dynamics are of great importance and the current literature is rich with research and experimental works on this domain.

Table 1: Table explaining the states and the symbols used in our NBAs and PBAs.

Symbol or State	Full Name
Nat	Native state
Mis	Misfolded state
Path	Pathway state
f	fold
u	unfold
d	degradation
s	stability

5. Protein folding and Büchi automata

In this part of our study we make a first attempt to describe and characterize biological models with inherited infinitary properties. We use as a case-study the protein folding problem, which was discussed in the previous section. The fact that phenomena like this have similarities with non-terminating systems supports our concept. Before we proceed to our formalizations, we should describe the abbreviations and terminology we use. Automata traditionally accept strings/words and, consequently, they recognize languages (sets of strings/words). As a consequence we translate the input of our machines into a simple string of a given alphabet. Moreover, we name each state according to the state of the biological system at the time of the computation/function, preferring names indicative enough of the actual state of the biological state.

Our rationale is easily seen from the abbreviation table (Table 1). We model the conformations of the proteins in each stage of the folding process as states of a NBA. The transitions are labelled with the action undertaken in each step of the computation, e.g. f stands for "fold" meaning that the automaton reads (or executes) a folding attempt. The above translation is essential in order to achieve the symbol-writing behaviour.

Firstly, we present two NBAs with their equivalent ω-regular expressions and then, we extend the first of them into a PBA, altering stochastically the appropriate transitions. We report the differences between these two models of computations. We note that in the next Section we provide some thoughts for future work in the same direction. The corresponding ω-regular expression for the NBA of the Fig. 1 is the $((fu)^*fd)^\omega$. Final state of the $\mathcal{A}$ is the *nat2* and initial is the *nat1*.

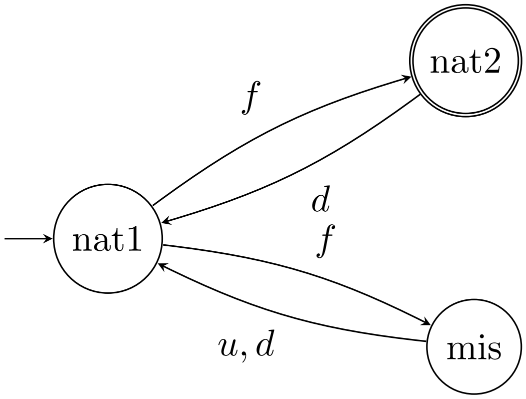

Figure 1: A simple Büchi automaton $\mathcal{A}$ describing protein folding. It accepts the language described by the ω-regular expression $((fu)^*fd)^\omega$.

Although the final state *nat2* is indeed being visited infinitely often, this model has a major drawback: the "bad" state *mis* is also being visited infinitely often, since the sequence $\{f,u\}$ (that is $\{fold,unfold\}$) appears infinitely often as one can clearly observe both in Fig. 1 and the corresponding ω-regular expression. This is actually an unwanted property for the protein formation procedure. Still, it is very useful in order to present our concept and then compare the two variants of Büchi automata.

The above observation led us to propose another NBA, more complex than $\mathcal{A}$, but with more realistic properties, i.e. the $\mathcal{A}'$ automaton of Fig. 2. The Büchi automaton $\mathcal{A}'$ in Fig. 2 is equivalent to the more complex ω-regular expression ((fu)+(fs$^+$d)+(s^+d))* (fs^+d)$^\omega$.

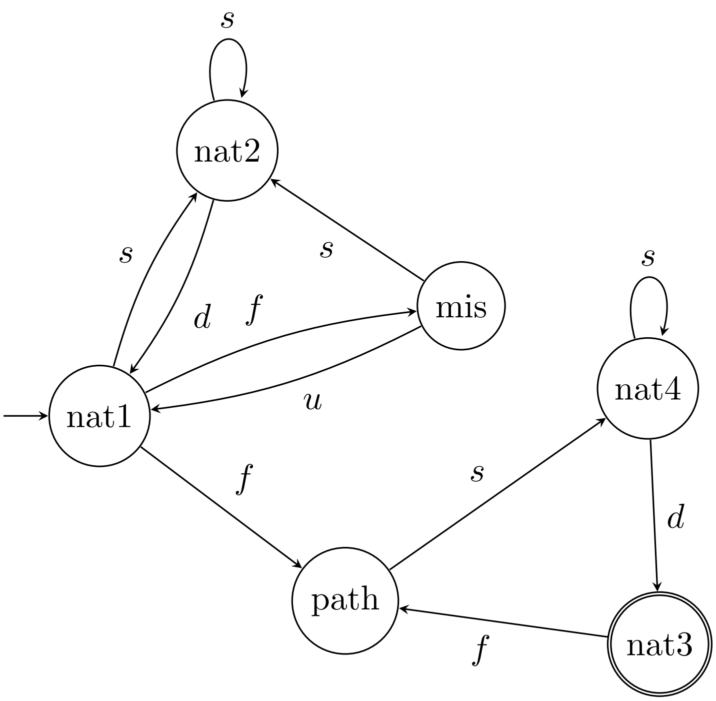

Figure 2: A complex Büchi automaton $\mathcal{A}'$ describing protein folding. It accepts the language described by the ω-regular expression ((fu)+(fs$^+$d)+(s^+d))* $(fs^+d)^\omega$.

Now, this machine is fairly more complex, with more states than the previous NBA; we write $|Q_{\mathcal{A}'}| > |Q_{\mathcal{A}}|$, where $|Q_{\mathcal{A}'}|$ is the number of states of automaton $\mathcal{A}'$ and $|Q_{\mathcal{A}}|$ the number of states of $\mathcal{A}$. In this case the undesirable state *mis* deriving from the pair {*fold,unfold*} is visited only finitely, since the part of the $\mathcal{A}'$ which yields the accepting behaviour occurs in an infinite horizon. This takes place after an arbitrary, distinct read of the symbol f that causes the transition from the *nat1* to the *path* state:

$$\ldots \xrightarrow{d,u} \mathit{nat1} \xrightarrow{f} \mathit{path} \xrightarrow{s} \ldots$$

This model is far more realistic, since it tries to capture the stability steps of the protein folding procedure, in which the protein finds itself in a situation where it does not need neither a folding action nor an unfold/degradation. This model can be extended and become even more realistic, verifying exact properties of the whole process with assisted by an actual biological model that describes in details the protein folding mechanism.

The fact that a protein turns up from its unfolded state to the folded one instantly and it “finds” its accurate structure among, almost, infinite configurations, emphasises the use of stochastic, rather than simply deterministic, methods to model the nature of the folding problem. As a consequence, we extend our proposed mechanism using probabilities, expressing the problem in terms of probabilistic Büchi automata (Fig. 3). The corresponding ω-regular expression for the PBA $\mathcal{P}$ depicted in Fig. 3 is the $(fd + fu)^*(fd)^\omega$.

Note that if we ignore the transition possibilities, the NBA we obtain is not equivalent to the PBA. This observation seems surprising and unnatural, but due to the infinite length of the input it is true. For the PBA in Fig. 3, the underlying NBA (the $\mathcal{A}$ from Fig. 1) recognizes the language $((fu)^*fd)^\omega$ by omitting the transition probabilities. PBAs are generally more expressive than the standard ω-automata. We point that the acceptance of a PBA is determined by the three semantics we pose on the computation run: the *probable semantics*, the *almost-sure semantics* and the *threshold* or *cut-point semantics*.

The probable semantics assigns to the PBA the set of infinite words w for which the runs have positive measure. Under the almost-sure semantics, a word w is accepted by the PBA if almost all runs for w are accepting and under the threshold or cut-point semantics the computation accepts with probability greater than a threshold λ. An equivalent NBA to the PBA of the Fig. 3 is depicted in Fig. 4. Both recognize the same language, but the latter has fewer states.

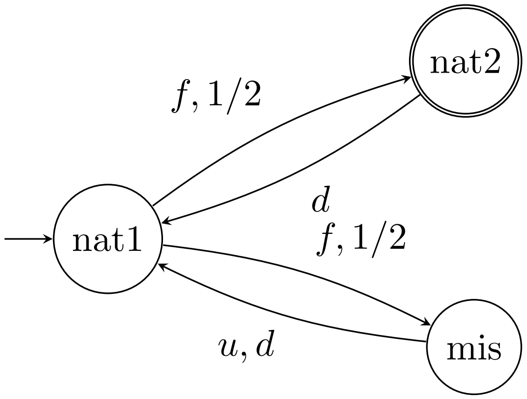

Figure 3: A probabilistic Büchi automaton $\mathcal{P}$ describing the protein folding. It accepts the language described by the ω-regular expression $(fd + fu)^*(fd)^\omega$.

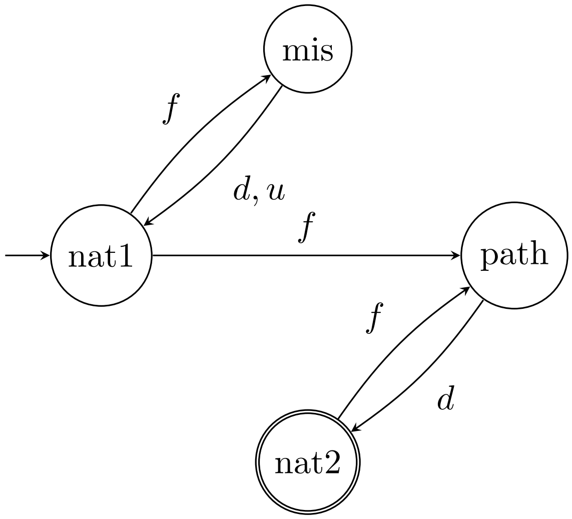

Figure 4: The equivalent NBA for the $\mathcal{P}$.

6. Future work

This work contributes to the field of Bioinformatics and especially in the branch of the modelling and verification of life systems with infinitary behaviour. We combined stochastic models of computation receiving inputs of infinite magnitude (probabilistic Büchi automata) and we established the use of them in natural phenomena of a living organism. We presented some of the advantages of this method (the minimization of the state-size) and we discussed satisfactory simulations of the protein folding mechanisms.

It is of great interest to test other biological models with similar properties to the protein folding, like fusion/fission mechanisms in a cellular mitochondrion, with realistic parameters and circumstances. Also, the choice of the probability distribution among the transition of the automata used in each case needs further investigations. For example, with which probabilities a protein folds into the wrong formation, leading to misfolding?

Another aspect of the biological function which is related to the DNA formation, that is under constant research, is the use of these subsystems as computational devices. DNA computing is a trending term in the current literature, as well as in industrial applications, and the potentials are prosperous in this domain. Close to the DNA computing notion, the quantum alternative mechanisms of computation [65], such as quantum automata, can be assisted from the research on biological processes, especially in the verification of the properties a quantum system should possess.

References

[1] M. Sipser. *Introduction to the Theory of Computation.* Thomson Course Technology, 2nd edition, 2006.

[2] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley, 2nd edition, 2000.

[3] H. Lewis and C. H. Papadimitriou. *Elements of the Theory of Computation.* Prentice-Hall, 2nd edition, 1997.

[4] A. Sadot, J. Fisher, D. Barak, Y. Admanit, M. J. Stern, E. J. A. Hubbard, and D. Harel. Toward verified biological models. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 5(2):223–234, 2008.

[5] L. M. Adleman et al. Molecular computation of solutions to combinatorial problems. *Science-AAAS-Weekly Paper Edition*, 266(5187):1021–1023, 1994.

[6] G. Păun, G. Rozenberg, and A. Salomaa. *DNA computing: new computing paradigms.* Springer, 1998.

[7] J. R. Büchi. Weak second-order arithmetic and finite automata. In *Zeitschrift fr mathematische Logik und Grundlagen der Mathematik.* 1960.

[8] J. R. Büchi. On a decision method in restricted second-order arithmetic. In *Proceedings of the 1960 International Congress for the Logic, Methology and Philosophy of Science*, pages 1–11. Stanford University Press, 1962.

[9] M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, (141):1–35, 1969.

[10] M. O. Rabin. *Automata on infinite objects and Church's problem.* Number 13. Amer Mathematical Society, 1972.

[11] S. Safra. On the complexity of omega-automata. In *Proceedings of the 29th Symposium on Foundations of Computer Science (FOCS88)*, pages 319–327. IEEE Computer Society, 1988.

[12] W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume III, pages 389–455. Springer-Verlag, 1996.

[13] W. Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 133–192. Elsevier Science, 1990.

[14] W. Thomas. Complementation of büchi automata revisited. In *Contributions on Theoretical Computer Science in Honor of Arto Saloma*, pages 109–120. Springer-Verlag, 1999.

[15] E. Grädel, W. Thomas, and T. Wilke. *Automata, logics, and infinite games: a guide to current research.* Springer, 2002.

[16] W. Thomas and H. Lescow. Logical specifications of infinite computations. *A Decade of Concurrency Reflections and Perspectives*, pages 583–621, 1994.

[17] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, (9):521–530, 1966.

[18] M. Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Banff Higher Order Workshop*, pages 238–266. Springer-Verlag, 1995.

[19] M.Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proceedings of the First Symposium on Logic in Computer Science*, pages 322–331, 1986.

[20] M.Y. Vardi. Automatic verification of probabilistic concurrent finite state programs. In *FOCS*, pages 327–338, 1985.

[21] M. Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.

[22] K. Giannakis and T. Andronikos. Querying linked data and büchi automata. In *Semantic and Social Media Adaptation and Personalization (SMAP), 2014 9th International Workshop on*, pages 110–114. IEEE, 2014.

[23] M. O. Rabin. Probabilistic automata. *Information and Control*, (6):230–245, 1963.

[24] A. Paz. *Introduction to probabilistic automata.* Academic Press, Inc., Orlando, FL, USA, 1971.

[25] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[26] L. De Alfaro. *Formal verification of probabilistic systems.* PhD thesis, Stanford University, Department of Computer Science, 1997.

[27] O. Goldreich. Randomized methods in computation. *Lecture Notes. http://www. wisdom. weizmann. ac. il/õded/rnd-sum. html*, 2001.

[28] H. Gimbert and Y. Oualhadj. Probabilistic automata on finite words: Decidable and undecidable problems. In *Automata, Languages and Programming*, pages 527–538. Springer, 2010.

[29] C. Baier, N. Bertrand, and M. Grösser. On decision problems for probabilistic büchi automata. In *FOSSACS*, pages 287–301. Springer, 2008.

[30] C. Baier and M. Grösser. Recognizing ω-regular languages with probabilistic automata. In *LICS*, pages 137–146, 2005.

[31] C. Baier, N. Bertrand, and M. Grösser. Probabilistic automata over infinite words: Expressiveness, efficiency, and decidability. In *DCFS, volume 3 of EPTCS*, pages 3–16, 2009.

[32] C. Baier, M. Grösser, and N. Bertrand. Probabilistic ω-automata. *Journal of the ACM*, 59(1):1–52, 2012.

[33] C. Baier, N. Bertrand, and M. Grösser. Probabilistic acceptors for languages over infinite words. In M. Nielsen et al., editor, *SOFSEM*, volume 5404 of Lecture Notes in Computer Science, pages 19–33. Springer-Verlag, 2009.

[34] C. Baier, N. Bertrand, and M. Grösser. The effect of tossing coins in omega-automata. In M. Bravetti and G. Zavattaro, editors, *CONCUR*, volume 5710, of Lecture Notes in Computer Science, pages 15–29. Springer-Verlag, 2009.

[35] T. Weidner. Probabilistic automata and probabilistic logic. In V. Sassone B. Rovan and P. Widmayer, editors, *MFCS 2012*, pages 813–824. Springer-Verlag, 2012.

[36] T. Weidner. Probabilistic ω-regular expressions. In *Language and Automata Theory and Applications*, pages 588–600. Springer, 2014.

[37] J. Fisher, D. Harel, and T. A. Henzinger. Biology as reactivity. *Communications of the ACM*, 54(10):72–82, 2011.

[38] D. Harel, Y. Setty, S. Efroni, N. Swerdlin, and I. R. Cohen. Concurrency in biological modeling: Behavior, execution and visualization. *Electronic Notes in Theoretical Computer Science*, 194(3):119–131, 2008.

[39] S. K. Jha, E. M. Clarke, C. J. Langmead, A. Legay, A. Platzer, and P. Zuliani. A bayesian approach to model checking biological systems. In *Computational Methods in Systems Biology*, pages 218–234. Springer, 2009.

[40] R. Freund, M. Oswald, and L. Staiger. ω-p automata with communication rules. In C. Martin-Vide, G. Mauri, G. Păun, G. Rozenberg, and A. Salomaa, editors, *Membrane Computing*, volume 2933 of *Lecture Notes in Computer Science*, pages 203–217. Springer Berlin Heidelberg, 2004.

[41] G. Păun. Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143, 2000.

[42] E. Petre. Watson-crick ω-automata. *Journal of Automata, Languages and Combinatorics*, 8(1):59–70, 2003.

[43] R. Freund, Gh Păun, G. Rozenberg, and A Salomaa. Watson-crick finite automata. *DNA Based Computers Three*, 48:297, 1999.

[44] K. Giannakis and T. Andronikos. Mitochondrial fusion through membrane automata. In *GeNeDis 2014*, pages 163–172. Springer, 2015.

[45] A. Li and V. Daggett. Identification and characterization of the unfolding transition state of chymotrypsin inhibitor 2 by molecular dynamics simulations. *Journal of molecular biology*, 257(2):412–429, 1996.

[46] C. M. Dobson. Protein folding and misfolding. *Nature*, 426(6968):884–890, 2003.

[47] F. U. Hartl. Chaperone-assisted protein folding: the path to discovery from a personal perspective. *Nature medicine*, 17(10):1206–1210, 2011.

[48] A. L. Horwich. Protein folding in the cell: an inside story. *Nature medicine*, 17(10):1211–1216, 2011.

[49] Y. Kang and C. M. Fortmann. An alternative approach to protein folding. *BioMed research international*, 2013, 2013.

[50] S. W. Englander, L. Mayne, and M. M. G. Krishna. Protein folding and misfolding: mechanism and principles. *Quarterly reviews of biophysics*, 40(04):287–326, 2007.

[51] Y.-F. Shen, B. Li, and Z.-P. Liu. Protein structure alignment based on internal coordinates. *Interdisciplinary Sciences: Computational Life Sciences*, 2(4):308–319, 2010.

[52] D. Baker and A. Sali. Protein structure prediction and structural genomics. *Science*, 294(5540):93–96, 2001.

[53] J.-M. Chandonia and S. E. Brenner. The impact of structural genomics: expectations and outcomes. *Science*, 311(5759):347–351, 2006.

[54] H. A. Scheraga, M. Khalili, and A. Liwo. Protein-folding dynamics: overview of molecular simulation techniques. *Annu. Rev. Phys. Chem.*, 58:57–83, 2007.

[55] Timo Koski. *Hidden Markov models for bioinformatics*, volume 2. Springer, 2001.

[56] H. Ding, H Lin, W. Chen, Z.-Q. Li, F.-B. Guo, J. Huang, and N. Rao. Prediction of protein structural classes based on feature selection technique. *Interdisciplinary Sciences: Computational Life Sciences*, 6(3):235–240, 2014.

[57] M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM journal of research and development*, 3(2):114–125, 1959.

[58] B. H. Toyama and M. W. Hetzer. Protein homeostasis: live long, won't prosper. *Nature Reviews Molecular Cell Biology*, 14(1):55–61, 2013.

[59] B. Alberts, D. Bray, K. Hopkin, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Essential cell biology*. Garland Science, 2013.

[60] D. J. Selkoe. Folding proteins in fatal ways. *Nature*, 426(6968):900–904, 2003.

[61] A. L. Goldberg. Protein degradation and protection against misfolded or damaged proteins. *Nature*, 426(6968):895–899, 2003.

[62] H. Chen and D. C. Chan. Mitochondrial dynamics-fusion, fission, movement, and mitophagy-in neurodegenerative diseases. *Human molecular genetics*, 18(R2):R169–R176, 2009.

[63] A. M. van der Bliek, Q. Shen, and S. Kawajiri. Mechanisms of mitochondrial fission and fusion. *Cold Spring Harbor perspectives in biology*, 5(6), 2013.

[64] D. L. Longo and S. L. Archer. Mitochondrial dynamics-mitochondrial fission and fusion in human diseases. *New England Journal of Medicine*, 369(23):2236–2251, 2013.

[65] Y. M. Li. Finite automata based on quantum logic and monadic second-order quantum logic. *Sci China Inf Sci*, (53):101–114, 2010.