Omega automata and its classes

Natasha Singh* and Ajay Kumar

Thapar Institute of Engineering and Technology, Patiala 147 003, India

ω-automata is a variant of finite automata which accepts infinite strings. It represents the behaviour of the infinite systems (hardware, operating system and control systems) which are not expected to end. A variety of conditions are used to represent the set of accepting strings in ω-automata. This paper summarizes various types of ω-automata, their transition functions and accepting conditions. In addition, this paper also summarizes the applicability of omega automata in various interdisciplinary fields.

Keywords: Büchi automata, co-Büchi automata and Muller automata, Rabin automata, Streett automata.

IN the theory of computation, infinite trees are used as a mathematical tool for representing the behaviour of systems such as protocols or circuits. Omega-automata are used to represent infinite strings (ω-word). ω-word is an infinite sequence of the symbol $a_1a_2a_3 \ldots$ where $a_1, a_2, a_3, \ldots \in \Sigma$. Language generated by omega-automata is used for representing non-terminating computations. Omega automata are used for specifying and verifying reactive systems, operating systems, concurrent and distributed systems. Omega automata are also applied to solve the significant problem on temporal logic.

Research on ω-language began in 1960 when Büchi studied monadic second-order theories. In this paper, we review various types of omega automata and their applicability. Omega automata can be described by a finite number of states, an input alphabet, an initial state, transition function and an accepting condition. Regular ω-language is the most important ω-language and can be described by ω-regular expression, Büchi automata, Muller automata and other equivalent formalism.

Kupferman *et al.*[1] studied three notions of typeness on ω-regular automata. These notions are useful for transitions between various types of omega automata. Further, they showed that the transitions from non-deterministic Büchi to non-deterministic co-Büchi is more complicated. Tao[2] discussed the infinity problem of ω-automata. He showed N logspace completeness if we can convert a generalized Büchi, Rabin, Muller or parity automata into an equivalent non-deterministic Büchi automata in logspace. Chen[3] introduced the concept of ω-grammar and ω-automata. He provided a systematic study on the generative power of ω-grammar and described various types of ω-grammar. He showed that the generative power of ω-context-free grammar is strictly weaker than ω-pushdown automata, whereas the generative power of ω-context-sensitive grammar and ω-Turing machine is the same. Redziejowski[4] proposed an improved construction of deterministic omega automata from ω-regular expression using the derivative. The proposed methodology was inspired by Safra's method and produces omega automata with fewer states. Kupferman and Vardi[5] described the optimal complementation construction for co-Büchi automata, Rabin and Streett automata.

Properties in model checking are explicitly specified using Linear-time Temporal Logic (LTL). Giannakopoulou and Lerda[6] proposed an efficient approach for the conversion of LTL to Büchi automata. Their implementation was released as a part of Java Path Finder Software. Fritz and Wilke[7] applied the quotienting by simulation equivalence technique to alternating Büchi automata. They introduced minimax and semi-elective quotient and proved that these newly introduced quotients are not difficult to compute than non-deterministic Büchi automata quotient.

Carton and Michel[8] introduced the concept of unambiguous Büchi automata for recognizing a rational set of ω-word. These automata start from an initial state and reach final states through exactly one path. Sheridan[9] established a relation between star normal form (SNF) and alternating automata. He proposed a bounded model checking using alternating automata and explored the difference between alternating, Büchi automata and SNF in terms of their applicability.

Boker and Kupferman[10] proposed an approach for conversion of non-deterministic Büchi word automata to deterministic and non-deterministic co-Büchi word automata. The obtained deterministic co-Büchi word automata have the upper bound $2^{O(n)}$, whereas the non-deterministic co-Büchi word automata have the upper bound $(n2^n)$ and lower bound $2^{\Omega(n)}$. Further, they converted Rabin, Muller, Streett and Parity word automata into non-deterministic Co-Büchi word automata. Piterman[11] described the construction of deterministic parity automata from non-deterministic Büchi and Streett automata with reduced states. The obtained automata have a variety of applications in satisfiability of CTL and reasoning of tree automata. The main advantage of parity acceptance condition allows more efficient algorithms than Rabin and Streett automata. Zeng and Tan[12] proposed a specification-based methodology for testing reactive system by

*For correspondence. (e-mail: natasha3699@gmail.com)

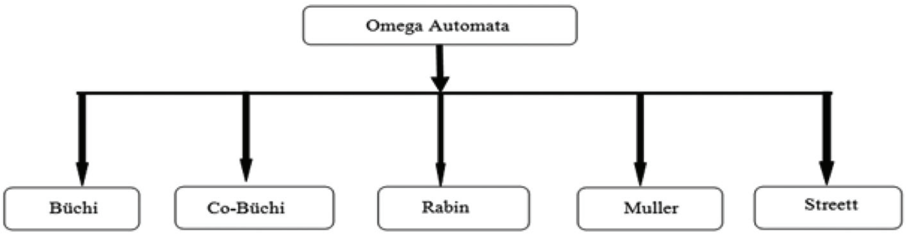

Figure 1. Classification of ω-automata.

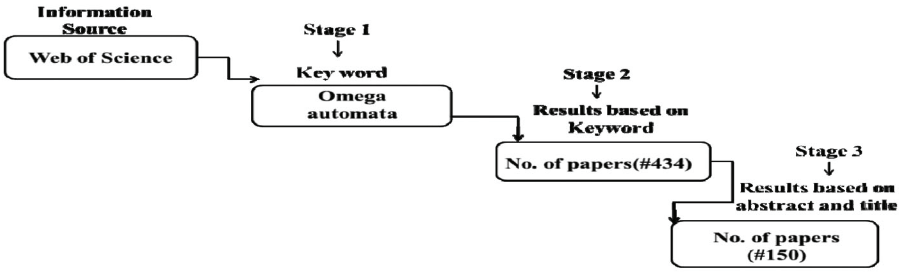

Figure 2. Criterion for selecting the paper relevant to omega automata.

Table 1. Results of keyword-based search and identified relevant paper

	No. of papers (keyword)	Relevant papers (based on abstract and title)
Omega automata	434	150
Büchi automata	352	81
Co-Büchi automata	21	5
Rabin automata	87	31
Muller automata	76	26
Streett automata	28	13

specifying requirement using Büchi automata. They built a testing process for specifying the properties of a reactive system. The proposed testing approach detected bugs in the system and its requirement. Further, they evaluated their approach using cross-coverage comparison and fault sensitivity analysis.

Baier *et al.*[13] applied the complementation operator on probabilistic Büchi automata (PBA). They proposed several algorithms for PBA. They showed that equivalence of PBA is an undecidable problem. Safra[14] proposed the construction of deterministic Büchi automata from non-deterministic Büchi automata by a generalization of subset construction. The proposed method is simple and provides single exponent upper bound in the general case. His proposed approach can be applied for obtaining the complementation of Büchi automata. Giannakis and Andronikos[15] described the query process of linked data using SPARQL and its verification is done using Büchi automata. Thiemann and Sulzmann[16] extended the partial derivatives and Brzozowski derivatives[17] for conversion of ω-regular expressions to non-deterministic Büchi automata by introducing the concept of a linear factor. Esik and Ivan[18] introduced the concepts of context-free grammar with Büchi and Müller acceptance condition. Further, they proved that a context-free grammar generated using Müller acceptance condition is not a Büchi context-free language. Tsay *et al.*[19] created an open repository on ω-automata. Readers can refer Thomas[20] for preliminaries related to ω-automata.

After accessing the applications of ω-automata in non-terminating systems, we realized the need for a systematic literature review. We summarized the existing research on ω-automata and their classes using abstract and title-based approaches.

Exploration of various types of omega automata

In this section, a brief overview of various classes of ω-automata has been presented. Figure 1 represents various types of ω-automata.

Search inclusion and exclusion criterion

This review starts by identifying research papers related to ω-automata and their applications from the Web of Science. Initially, we searched the paper using keyword omega automata, but it gave many irrelevant results related to automata. From the obtained search, we included the paper related to ω-automata by looking at the title and abstract. Figure 2 represents the result obtained using keyword-based search followed by an exclusion criterion adopted.

Table 1 clearly shows that irrelevant papers are larger. Hence the keyword-based approach was discarded, and the abstract and title-based search approach was included.

Table 2. Research questions and their motivation

Research question	Main motivation
RQ 1: How many papers were published between 1989 and 2017 related to ω-automata?	Identify the papers published after 1989 to till date
RQ 2: What are the different acceptance criteria for various types of ω-automata?	Identify the major differences between the acceptance criterion of different types of ω-automata
RQ 3: In which areas ω-automata can be applied.	Identify the major applications of ω-automata
RQ 4: What are the closure properties of various types of ω-automata?	Identify the closure properties of various types of ω-automata
RQ 5: To identify various researchers working in the field of ω-automata.	Recognize the active researchers working in the area of ω-automata

Table 3. Classification of papers

Property	Category
Year	1989–2017
Evaluation criteria	Based on title and abstract
Classification of papers	Papers related to omega, Büchi, co-Büchi, Muller, Rabin and Streett automata
Publication type	Journal article, book chapter and conference article

Table 4. Analysis of publication type referred in the survey

Publication type	No. of papers referred
Journal	63
Book	2
Conference article	64

Classification of papers

After identifying the relevant papers, it was observed that ω-automata research papers can be categorized based on the research questions in Table 2.

We included papers in our review based on the following classification criteria as shown in Table 3. Publication type analysis of the relevant papers referred in this survey is shown in Table 4.

Results

Year-wise status of publications (RQ 1)

After searching the research papers from Web of Science using keyword, followed by finding the relevant paper using title and abstract the figures are shown in Figure 3. Figure 4 represents the pie chart for the number of papers published since 1989 related to ω-automata. Figures 5–10 represent the graph plot of various research papers taken from various resources related to ω-automata.

Büchi Automata (RQ 2)

Julius Richard Büchi[21] proposed in 1962 the concept of Büchi automata which works on ω-strings. If some final state is visited infinitely often for running a ω-string, then the string is said to be accepted. It also represents ω-regular languages. Nitsche[22] proposed a power set construction for Büchi automata. Colcombet and Zdanowski[23] found tight lower bound for conversion of Büchi automata into deterministic Rabin automata. Shan *et al.*[24] proposed an approach for conversion of LTL to Büchi automata. Figure 11 represents two different types of Büchi automata.

Büchi automata are classified into unambiguous and ambiguous Büchi automata. Unambiguous Büchi automata start with an initial state and on reading ω-word, visit some final state through exactly one path. Ambiguous Büchi automata[8] on reading ω-string visit some final states infinitely often with more than one path.

Co-Büchi automata (RQ 2)

Co-Büchi automaton[25] is a variant of Büchi automata with different accepting conditions. A ω-word is accepted by co-Büchi automata if all the states occurring infinitely often in the run belong to the set of final states F. Wang *et al.*[26] worked on synthesis of co-Büchi automata specification.

Rabin automata (RQ 2)

Rabin automata[27] $M(Q, \Sigma, \delta, q_0, \Omega)$ with accepting component $\Omega = \{(E_1, F_1), (E_2, F_2), \ldots, (E_n, F_n) \mid E_i, F_i \subseteq Q\}$ and a run of ω-word is accepted if for $\forall i \in [1,\ldots, n]$ satisfy conditions $\mathrm{Inf}(r) \cap E = \phi$ and $\mathrm{Inf}(r) \cap F \neq \phi$.

Müller automata (RQ 2)

Muller automata[28] $M(Q, \Sigma, \delta, q_0, \Omega)$ with accepting component $F \subseteq Pow(Q)$ and a run of ω-word is accepted if $\mathrm{Inf}(r) \in F$.

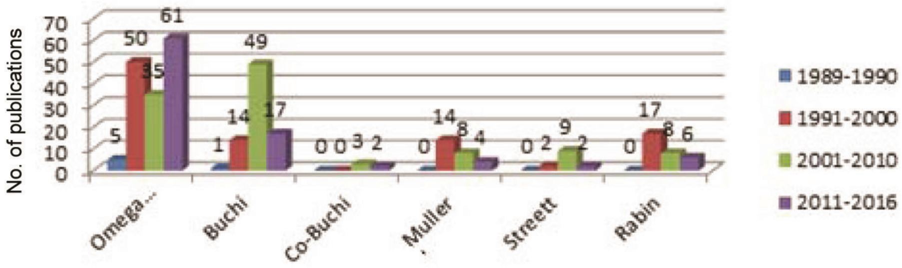

Figure 3. Year-wise distribution of a number of publications related to ω-automata.

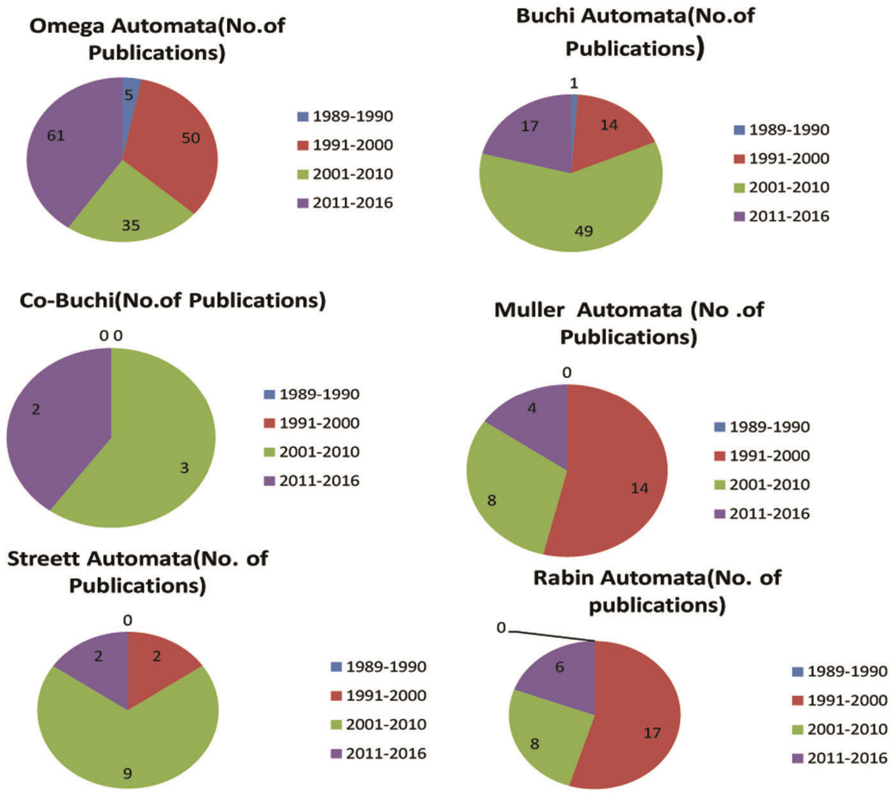

Figure 4. Year-wise number of publications of various types of ω-automata.

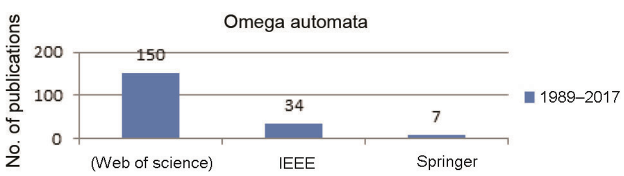

Figure 5. Graph plot of the number of research papers related to omega automata.

Streett automata (RQ 2)

Streett automata[29] $M(Q, \Sigma, \delta, q_0, \Omega)$ with accepting component $\Omega = \{(E_1, F_1), (E_2, F_2), \ldots, (E_n, F_n) \mid E_i, F_i \subseteq Q\}$ and a run of ω-word is accepted if for $\forall i \in [1,\ldots, n]$ satisfy conditions $\mathrm{Inf}(r) \cap E_i = \phi$ or $\mathrm{Inf}(r) \cap F_i \neq \phi$. Henzinger and Telle[30] applied lockup search for solving non-emptiness of Streett automata.

Table 5 represents the classification of various types of ω-automata based on their acceptance condition. Rabin and Streett automata accepting conditions are complement to each other.

The classification table presenting the various classes of omega automata, their acceptance conditions, comparison analysis and the transition modes is depicted in Table 5. The closure properties of various types of ω-automata are shown in Table 6.

Applications of omega automata (RQ 3)

The applications of various classes of omega automata are described with the help of Table 7.

Table 5. Classification of ω-automata

Omega automata	Year	Acceptance condition	Comparison
Büchi automaton[21]	1962	$\mathrm{Inf}(r) \cap F \neq \phi$	There exists at least one final state that is visited infinitely often during ω-run
Co-Büchi automaton[25]	1965	$\mathrm{Inf}(r) \subseteq F$	All states visiting infinitely often belongs to set F
Rabin automaton[27]	1969	$\mathrm{Inf}(r) \cap E = \phi$ and $\mathrm{Inf}(r) \cap F \neq \phi$	Accepting component $\Omega = \{(E_1, F_1), (E_2, F_2),\ldots,(E_n, F_n) \mid E_i, F_i \subseteq Q\}$
Muller automaton[28]	1963	$\mathrm{Inf}(r) \in F$	Infinitely visited states belong to the accepting component
Streett automaton[29]	1982	$\mathrm{Inf}(r) \cap E_i = \phi$ or $\mathrm{Inf}(r) \cap F_i \neq \phi$	Accepting component $\Omega = \{(E_1, F_1), (E_2, F_2),\ldots,(E_n, F_n) \mid E_i, F_i \subseteq Q\}$

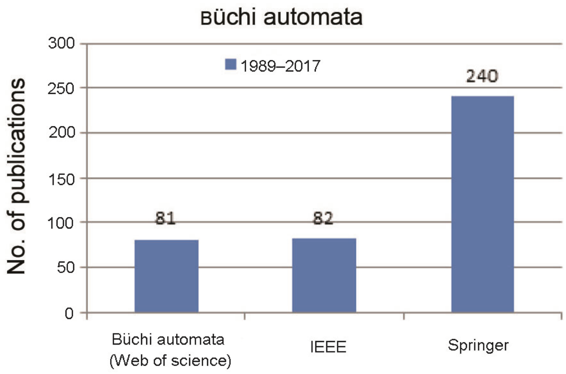

Figure 6. Graph plot of the number of research papers related to Büchi automata.

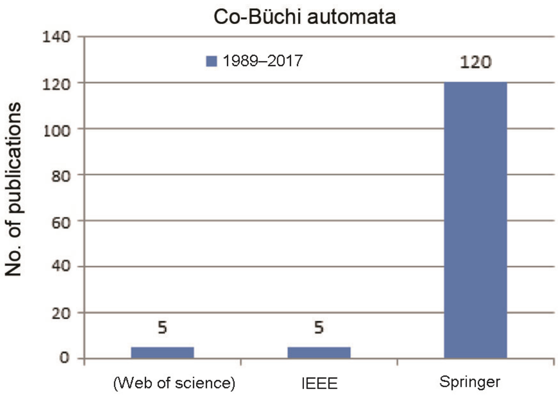

Figure 7. Graph plot of the number of research papers related to co-Büchi automata.

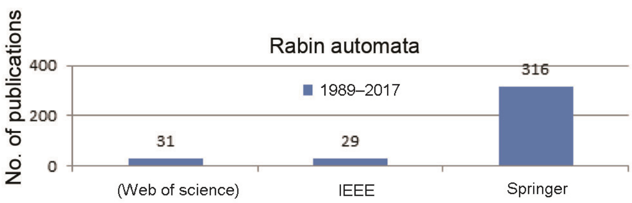

Figure 8. Graph plot for various research papers related to Rabin automata.

Minimization of non-deterministic Büchi, co-Büchi, Rabin, Muller and Streett automata is PSPACE-comp. Expressive power of Büchi, deterministic Rabin, Müller and Streett automata are same, whereas co-Büchi automata expressive power is less than Büchi automata.

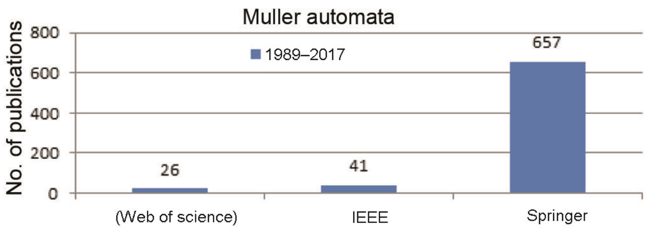

Figure 9. Graph plot for various research papers related to Muller automata.

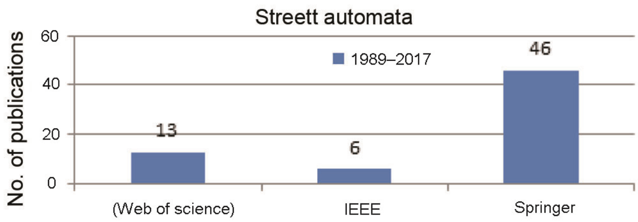

Figure 10. Graph plot for various research papers related to Streett automata.

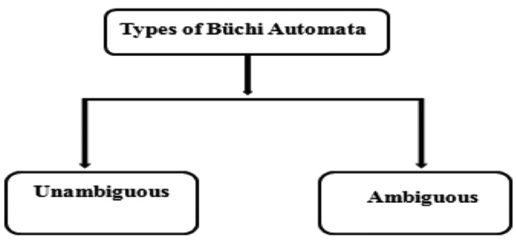

Figure 11. Types of Büchi automata.

Literature survey on ω-automata (RQ 5)

McNaughton and Yamada[31] proposed algorithms (supported by theorems) for converting state graph into a regular expression and vice versa. Ginzburg[32] devised a method for checking equivalence between two regular expressions using derivative of regular expressions and state graph. Kumar and Verma[33] determined the state complexity of non-deterministic finite automata from regular expressions. Garhwal and Jiwari[34] introduced the concept of fuzziness in parallel regular expression. Yan[35] determined the lower bound for the complementation of generalized Büchi automata. He showed that the

Table 6. Closure properties of various classes of ω-automata (RQ 4)

Types of ω-automata	Closed under operations
Büchi automata[21]	Union, concatenation, complement, intersection and omega closure
Co-Büchi automata[25]	Union, intersection, projection, determinization and omega closure
Rabin automata[27]	Union, intersection, omega closure and not closed under negation
Muller automata[28]	Boolean operation and complementation
Streett automata[29]	Determinization, union, intersection and omega closure

Table 7. Application of Büchi automata in various interdisciplinary fields (RQ 3)

Types of omega automata	Application areas
Büchi automata	Developing decision procedures for temporal logic[64]
	Bounded model checking of LTL formulae[65]
	Modelling of discrete event systems[66]
	Querying linked data using Büchi automata[15]
	Testing reactive systems[12]
	Formalization of digital forensic theory[67]
	Generation of profile trees for determinization[68]
	Generation of automata games[69]
	Generation of emptiness checking algorithms[70]
	Generation of safety and reachability games[71]
	Biological processes[72]
Co-Büchi automata	Game theory[71,73]
	Sequential synthesis of finite state machines using co-Büchi specifications[26]
	Puzzle games[73]
	Formal verification and synthesis[26]
Muller automata	Modelling asynchronous circuits and real time systems[74,75]
	Extension of existential monadic second-order logic[75]
Rabin automata	Game theory[75]
Streett automata	Game theory[63,75]
	Verification[76]

complementation of Büchi and generalized Büchi automata have the lower bounds $\Omega((0.76n)^n)$ and $(\Omega(nk))^n$ respectively.

Loding and Thomas[36] established the relation between monadic second-order logic and alternating weak automata on ω-strings. Miyano and Hayashi[37] introduced the concept of alternating ω-automata. They characterized the classes of alternating automata into four categories based on acceptance conditions. One of the acceptance conditions increases its power in comparison to non-deterministic. Baier and Grosser[38] introduced the concept of Probabilistic Büchi Automata (PBA). They showed that PBA is more expressive power than non-deterministic Büchi automata. They applied the concept of PBA in the verification of a Markov chain.

Park[39] modelled fair concurrency using the concept of ω-regular languages. He proved that ω-regular languages are closed under the operator fair concurrency. Staiger[40] proposed the concept of finite state ω-languages based on structural properties. The proposed finite state ω-languages are closely related to finite state automata. Study of ω-languages is a subset of topological space. Redziejowski[41] proposed a topology which results in a strongly-zero dimensional and completely regular topological space. Tsay *et al.*[42] designed a tool Graphical tool for Omega Automata and Logics (GOAL). This tool can be used for checking the equivalence of Büchi automata, conversion of linear temporal logic (LTL) to Büchi automata, design and testing of Büchi automata.

Angluin and Fisman[43] designed three variant of algorithms for finding unknown ω-regular expression based on learning from membership and equivalence queries. Chaturvedi *et al.*[44] established the relation between infinity games and ω-languages. They represented the winning strategies using *-languages which can be converted into ω-languages. Selivanov[45] introduced the concept of regular aperiodic ω-languages and characterized their Wadge degrees. d'Amorim and Roşu[46] proposed a technique for transforming Büchi automata to statistically optimal non-deterministic finite state machine with reduced size. Finkel[47] studied the closure properties of locally finite omega languages and proved that these languages are not closed under intersection and complement.

Thomas[48] described the relation of ω-words (represented by tree languages) using monadic second-order logic and showed that the k-ary tree is undecidable

Table 8. Key findings related to ω-automata

Author	Notable findings
Chen[3]	Discussed the generative power of ω-grammar
Redziejowski[55]	Conversion of ω-regular expressions to deterministic omega automata using the concept of derivatives
Giannakopoulou and Lerda[6]	Proposed tableau-based translation algorithm for constructing generalized Büchi automata
Carton and Michel[8]	Introduced the concept of unambiguous Büchi automata
Piterman[11]	Conversion of nondeterministic Büchi and Streett automata to deterministic parity automata
Safra[14]	Discussed the complexity of complementation problem for various types of ω-automata
Antimirov[57]	Introduced the concept of partial derivatives for regular expressions
Thiemann and Sulzmann[16]	Extended the conversion of omega-regular expressions to non-deterministic Büchi automata by introducing the concept of a linear factor in the partial derivative

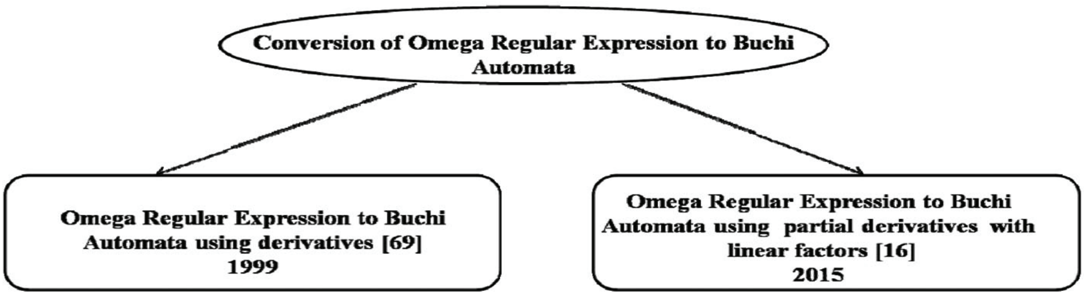

Figure 12. Various approaches for conversion of ω-regular expressions to Büchi automata.

in monadic second-order logic. Thistle and Wonham[49] found that automata can be controlled by imposing certain allowable restriction on the set of symbols. Thistle and Wonham[50] developed a fixpoint characterization of controllability subset for a deterministic Rabin automata. Using this fixpoint characterization, automata can be controlled to satisfy the acceptance criterion. Maler and Staiger[51] proposed several notions for ω-languages of syntactic congruence. They established a relation between syntactic congruence and its infinitary refinement. Thistle and Malhamé[52] worked on deadlock-free control of finite automata with respect to the specification specified in the Rabin acceptance conditions. State fairness condition implies that the controllability set can be computed in polynomial time.

Berry and Sethi[53] proposed the conversion of regular expressions to finite automata using the derivatives of regular expressions. Owens *et al.*[54] re-examined the regular expression derivative and reported their experiences with two functional languages. Redziejowski[55] applied the concept of partial derivatives for the conversion of ω-regular expressions to deterministic ω-automata. Antimirov[56] discussed the containment problem in algebra by applying algebraic specifications and term-rewriting methods. Antimirov[57] introduced the concept of partial derivatives for the conversion of regular expressions to non-deterministic finite automata based on the generalization of Brzozowski's derivatives. Caron *et al.*[58] extended the Antimirov's partial derivatives for the conversion of regular expressions (with complementation and intersection operators) to finite automata. Kumar and Verma[59] proposed a direct conversion from parallel regular expressions to deterministic finite automata.

Redziejowski[4] proposed an approach for the construction of deterministic automata from ω-regular expression using the derivative concept. Their proposed approach is inspired by Piterman's improvement. Brzozowski and Leiss[60] proposed the equations corresponding to Boolean automata. Their proposed equation was used to determine the language accepted by the sequential network.

Carton *et al.*[61] surveyed for the accepting conditions and equivalence of various ω-automata. Further, they studied the work of prophetic automata. Recently, Singh and Kumar[62] initiated the work for modelling various stages of cancer using Büchi automata.

Thiemann and Sulzmann[16] extended Brzozowski derivatives and partial derivatives for the construction of non-deterministic Büchi automata from ω-regular expressions by introducing the concept of a linear factor. This is shown with the help of Figure 12. The linear factor is defined as 3-tuples $\langle \alpha, \beta, \delta \rangle$ where α, β and δ represent current symbol read, remaining ω-regular expression to be processed and the current state which is final ($\delta = 1$) or non-final ($\delta = 0$) respectively.

Considering ω-regular expression $r = a^*b^*c^\omega$. On applying Thiemann and Sulzmann approach[16] for finding a linear factor, we obtain linear factor (LF) of r as

$$\{\langle a, a^*b^*c^\omega, 0\rangle, \langle b, b^*c^\omega, 0\rangle, \langle c, c^\omega, 1\rangle\} = Q = Q_0,$$

$$\delta(\langle a, a^*b^*c^\omega, 0\rangle, a) = LF(r) = Q,$$

$$\delta(\langle a, a^*b^*c^\omega, 0\rangle, b) = \langle\rangle,$$

$$\delta(\langle a, a^*b^*c^\omega, 0\rangle, c) = \langle\rangle,$$

$$\delta(\langle b, b^*c^\omega, 0\rangle, a) = \langle\rangle,$$

$$\delta(\langle b, b^*c^\omega, 0\rangle, b) = \{\langle b, b^*c^\omega, 0\rangle, \langle c, c^\omega, 1\rangle\},$$

$$\delta(\langle b, b^*c^\omega, 0\rangle, c) = \langle\rangle,$$

$$\delta(\langle c, c^\omega, 1\rangle, b) = \langle\rangle,$$

$$\delta(\langle c, c^\omega, 1\rangle, b) = \langle\rangle,$$

$$\delta(\langle c, c^\omega, 1\rangle, c) = \langle c, c^\omega, 1\rangle.$$

Accepting state is $F = (< c, c^\omega, 1 >)$.

Table 8 summarizes the key research finding in the area of ω-automata.

Open research challenges

The following are the research directions on which future work can be carried out: (i) Work can be carried out on ω-regular expression with an additional operator such as shuffle. (ii) Thiemann and Sulzmann[16] proposed the concept of a linear factor in the construction of Büchi automata. Work can be carried out for reducing the number of states by combining linear factors. (iii) A systematic comparative study can be carried out for generating Büchi automata from ω-regular expressions using existing approaches. (iv) Equivalence problem for ω-deterministic pushdown automata is still an open problem. (v) The concept of visibly pushdown automata and infinity games can be explored in the near future. (vi) Work can be carried out for extending Büchi automata with the involvement of quantum and establishing its relation with quantum logic. (vii) New approaches can be proposed for construction of Büchi automata from ω-regular expressions.

Conclusion

This paper reviewed and explored ω-automata and its types based on 76 relevant papers. Various classes of ω-automata have been investigated, characterized by their acceptance criteria and various application areas explored. In addition, this survey paper also reviewed various approaches used for conversion of ω-regular expressions to Büchi automata. Finally, this paper also facilitates readers with various open research challenges that future researchers can explore.

1. Kupferman, O., Morgenstern, G. and Murano, A., Typeness for ω-regular automata. *Int. J. Found. Comput. Sci.*, 2004, **17**(4), 324–338.
2. Tao, Y., Infinity problems and countability problems for ω-automata. *Inf. Proc. Lett.*, 2006, **100**, 151–153.
3. Chen, Z., On the generative power of ω-grammars and ω-automata. *Fundam. Infor.*, 2011, **111**(2), 119–145.
4. Redziejowski, R. R., An improved construction of deterministic omega-automaton using derivatives. *Fundam. Inform.*, 2012, **119**(3–4), 393–406.
5. Kupferman, O. and Vardi, M. Y., Complementation constructions for nondeterministic automata on infinite words. 11th International Conference on Tools and Algorithms for the Constructions and Analysis of Systems (TACAS 2005). *LNCS*, 2005, **3440**, 206–221.
6. Giannakopoulou, D. and Lerda, F., Efficient translation of LTL formulae into Büchi automata. RAICS Technical Report, Research Institute for Advanced Computer Science (RIACS), 1, 2001.
7. Fritz, C. and Wilke, T., Simulation relations for alternating Büchi automata. *Theoret. Comput. Sci.*, 2005, **338**(1–3), 275–314.
8. Carton, V and Michel, M., Unambiguous Büchi automata. *Theoret. Comput. Sci.*, 2003, **297**(1), 37–81.
9. Sheridan, D., Bounded model checking with SNF, alternating automata, and Büchi automata. *Electron. Notes Theoret. Comput. Sci.*, 2005, **119**(2), 83–101.
10. Boker, U. and Kupferman, O., Translating to co-Büchi made tight, unified and useful. *ACM Trans. Comput. Logic (TOCL)*, 2012, **13**(4), 29.
11. Piterman, N., From nondeterministic Büchi and Streett automata to deterministic parity automata. 21st Annual IEEE Symposium on Logic in Computer Science (LICS'06), Seattle WA, USA, 12–15 August 2006, 3(3), 255–264.
12. Zeng, B. and Tan, L., Test reactive systems with Büchi automata: acceptance condition coverage criteria and performance evaluation. IEEE International Conference on Information Reuse and Integration (IRI), San Francisco, CA, USA, 13–15 August 2015, pp. 380–387.
13. Baier, C., Bertrand, N. and Größer, M., On decision problems for probabilistic Büchi automata. Int. Conference on Foundations of Software Science and Computational Structures (FoSSaCS2008), LNCS, 2008, 4962, 287–301.
14. Safra, S., On the complexity of omega automata. Proceedings of the 29th Annual Symposium on Foundations of ComputerScience (SFCS'88), Washington, DC, USA, 24–26 October 1988, pp. 319–327.
15. Giannakis, K. and Andronikos, T., Querying linked data and Büchi automata. 9th International Workshop on Semantic and Social Media Adaptation and Personalization, 9th International Workshop of IEEE, Corfu, Greece, 6–7 November 2014, pp. 110–114.

16. Thiemann, P. and Sulzmann, M., From omega-regular expressions to Büchi automata via partial derivatives. *Int. Conf. Language Automata Theory Appl.*, LATA, LNCS, 2015, 8977, 287–298.
17. Brzozowski, J. A., Derivatives of regular expressions. *J. ACM*, 1964, **11**(4), 481–494.
18. Esik, Z. and Ivan, S., Büchi context-free languages. *Theoret. Comput. Sci.*, 2011, **412**(8–10), pp. 805–821.
19. Tsay, Y. K., Tsai, M. H., Chang, J. S., Chang, Y. W. and Liu, C. S., Büchi store: an open repository of omega-automata. *Int. J. Software Tools for Technol. Transf.*, 2013, **15**(2), 109–123.
20. Thomas, W., Automata on infinite objects. *Handbook of Theoretical Computer Science*, MIT Press Cambridge. MA, USA, 1990, 133–187.
21. Büchi, J. R., On a decision method in restricted second order arithmetic. Proc. International Congress on Logic, Method, and Philosophy of Science, Stanford, Stanford University Press, 1962, pp. 1–12.
22. Nitsche, U. U., A power-set construction for reducing Büchi automata to non-determinism degree two. *Inf. Proc. Lett.*, 2007, **101**(3), 107–111.
23. Colcombet, T. and Zdanowski, K., A tight lower bound for determinization of transition labeled Büchi automata. *International Colloquium on Automata, Languages, and Programming*, ICALP 2009, LNCS 5556, 2009, pp. 151–162.
24. Shan, L., Du, X. and Qin, Z., Efficient approach of translating LTL formulae into Büchi automata. *Front. Comput. Sci*., 2015, **9**(4), 511–523.
25. Büchi, J. R., Decision methods in the theory of ordinals. *Bull. Am. Math. Soc*., 1965, **71**, 767–770.
26. Wang, G., Mishchenko, A., Brayton, R. D. and Sangiovanni Vincentelli, A., Sequential Synthesis with Co-Büchi Specifications, ERL Technical Report, EECS Dept., UC Berkeley, 2006.
27. Rabin, M. O., Decidability of second-order theories and finite automata on infinite trees. *Trans. Am. Math. Soc.*, 1969, **141**, 1–35.
28. Muller, D. E., Infinite sequences and finite machines. 4th Annual Symposium on Switching Circuit Theory and Logical Design (SWCT), 1963, pp. 3–16.
29. Streett, R. S., Propositional dynamic logic of looping and converse. *Inform. Control*., 1982, **54**, 121–141.
30. Henzinger, M. R. and Telle, J. A., Faster algorithms for the nonemptiness of Streett automata and for communication protocol pruning. Scandinavian Workshop on Algorithm Theory, SWAT1996, LNCS, 1996, **1097**, 16–27.
31. McNaughton, R. and Yamada, H., Regular expressions and state graphs for automata. *IRE Tran. Electron. Comput.*, 1960, **EC-9(1)**, 39–47.
32. Ginzburg, A., A procedure for checking equality of regular expressions. *J. ACM*, 1967, **14**(2), 355–362.
33. Kumar, A. and Verma, A. K., Conversion of parallel regular expressions to non-deterministic finite automata using partial derivatives. *Chiang Mai J. Sci.*, 2014, **41**(5.2), 1409–1418.
34. Garhwal, S. and Jiwari, R., Parallel fuzzy regular expression and its conversion to epsilon-free fuzzy automaton. *Comput. J.*, 2016, **59**(9), 1383–1391.
35. Yan, Q., Lower bounds for complementation of ω-automata via the full automata technique. International Colloquium on Automata Languages and Programming (ICALP2006), LNCS, 2006, pp. 589–600.
36. Loding, C. and Thomas, W., Alternating automata and logics over infinite words. IFIP Int. Conf. Theoret. Comput. Sci. *LNCS*, 2000, **1872**, 521–535.
37. Miyano, S. and Hayashi, T., Alternating finite automata on ω-words. *Theoret. Comput. Sci.*, 1984, **32**(3), 321–330.
38. Baier, C. and Grosser, M., Recognizing ω-regular languages with probabilistic automata. 20th Annual IEEE Symposium on Logic in Computer Science (LICS'05), 2005, pp. 137–146.
39. Park, D., Concurrency and automata on infinite sequences. *Theor. Comput. Sci., LNCS*, 1981, **104**, 167–183.
40. Staiger, L., Finite-state ω-languages. *J. Comput. Syst. Sci.*, 1983, 1, **27**(3), 434–448.
41. Redziejowski, R. R., Infinite-word languages and continuous mappings. *Theor. Comput. Sci.*, 1986, **1**(43), 59–79.
42. Tsay, Y. K., Chen, Y. F., Tsai, M. H., Wu, K. N. and Chan, W. C., Goal: a graphical tool for manipulating Büchi automata and temporal formulae. Int. Conf. Tools Algorithms Construct. Anal. Syst. *LNCS*, 2007, **4424**, 466–471.
43. Angluin, D. and Fisman, D., Learning regular omega languages. *Theor. Comput. Sci.*, 2016, **650**, 57–72.
44. Chaturvedi, N., Olschewski, J. and Thomas, W., Languages versus ω-languages in regular infinite games. *Int. J. Found. Comput. Sci.*, 2012, **23**(5), 985–1000.
45. Selivanov, V. L., Fine hierarchy of regular aperiodic ω-languages. International Conference on Developments in Language Theory, DLT 2007, LNCS, 2007, **4588**, 399–410.
46. d'Amorim, M. and Roşu, G., Efficient monitoring of ω-languages. International Conference on Computer Aided Verification. *LNCS*, 2005, **3576**, 364–378.
47. Finkel, O., Closure properties of locally finite ω-languages. *Theoret. Comput. Sci.*, 2004, **322**(1), 69–84.
48. Thomas, W., Infinite trees and automaton-definable relations over ω-words. Ann. Symposium on aspects of Theoretical Computer Science (STACS 1990). *LNCS*, 1990, **415**, 263–277.
49. Thistle, J. G. and Wonham, W., Control of omega-automata, Church's problem, and the emptiness problem for tree omega-automata. 5th Workshop on Computer Science Logic (CSL'91), 1991, 367–381.
50. Thistle, J. G. and Wonham, W. M., Control of infinite behavior of finite automata. *SIAM J. Control Opt.*, 1994, **32**(4), 1075–1097.
51. Maler, O. and Staiger, L., On syntactic congruencies for ω-languages. 10th Annual Symposium on Theoretical Computer Science. *LNCS*, 1993, **665**, 586–594.
52. Thistle, J. G. and Malhame, R. P., Control of ω-automata under state fairness assumptions. *Syst. Control Lett*., 1998, **33**(4), 265–274.
53. Berry, G. and Sethi, R., From regular expressions to deterministic automata. *Theor. Comput. Sci.*, 1986, **48**(1), 117–126.
54. Owens, S., Reppy, J. and Turon, A., Regular-expression derivatives re-examined. *J. Funct. Program.*, 2009, **19**(2), 173–190.
55. Redziejowski, R. R., Construction of a deterministic ω-automaton using derivatives. *RAIRO-Theoret. Inform. Appl.*, 1999, **33**(2), 133–158.
56. Antimirov, V. M., Rewriting regular inequalities. 10th International Symposium on Fundamentals of Computation Theory, 1995, 116–125.
57. Antimirov, V., Partial derivatives of regular expressions and finite automaton constructions. *Theor. Comput. Sci.*, 1996, **155**(2), 291–319.
58. Caron, P., Champarnaud, J. M. and Mignot, L., Partial derivatives of an extended regular expression. International Conference on Language and Automata Theory and Applications, LATA, 2011. *LNCS*, 1996, **6638**, 179–191.
59. Kumar, A. and Verma, A. K., A novel algorithm for the conversion of parallel regular expressions to non-deterministic finite automata. *Appl. Math. Inform. Sci*., 2014, **8**(1), 95–105.
60. Brzozowski, J. A. and Leiss, E., On equations for regular languages, finite automata, and sequential networks. *Theor. Comput. Sci.*, 1980, **10**(1), 19–35.
61. Carton, O., Perrin, D. and Pin, J. E., Automata and semigroups recognizing infinite words. *Logic and Automata, History and Perspectives*, Amsterdam University Press, 2007.
62. Singh, N. and Kumar, A., Modelling finite and infinite behavior of cancer stages using Büchi and Finite Automata. *Curr. Sci.*, 2018, **115**(4), 1–5.

63. Vardi, M. Y. and Wolper, P., An automata-theoretic approach to automatic program verification. *Proc. First Symp. Logic Comput. Sci.*, 1986, 322–331.
64. Sistla, A. P., Vardi, M. Y. and Wolper, P., The complementation problem for Büchi automata with applications to temporal logic. *Theoret. Comput. Sci.*, 1987, **49**(2–3), 217–237.
65. Clarke, E., Kroening, D., Ouaknine, J. and Strichman, O., Completeness and complexity of bounded model checking. International Workshop on Verification, Model Checking, and Abstract Interpretation, VMCAI 2004, LNCS, 2004, **2937**, 85–96.
66. Young, S., Spanjol, D. and Garg, V. K., Control of discrete event systems modeled with deterministic Büchi automata. American Control Conference, IEEE, 2814–2818, 1992.
67. Jalan, S., Kumar, P. and Das, S., Formalization of digital forensic theory by using Büchi automaton. Third International Conference on Image Information Processing (ICIIP). *IEEE*, 2015, 102–108.
68. Fogarty, S., Kupferman, O. and Wilke, T., Profile trees for Büchi word automata with application to determinization. *Inf. Comput.*, 2015, **245**, 136–151.
69. Alfaro, L. D. and Henzinger, T. A., Concurrent omega-regular games. 15th Annual IEEE Symposium on Logic in Computer Science, 26–29 June 2000, pp. 141–154.
70. Zhao, L., Zhang, J. and Yang, J., Advances in on-the-fly emptiness checking algorithms for Büchi automata. 5th International Conference on Advanced Computational Intelligence (ICACI), IEEE, Nanjing China, 18–20 October 2012, pp. 113–118.
71. Etessami, K., Wilke, T. and Schuller, R. A., Fair simulation relations, parity games, and state space reduction for Büchi automata. *SIAM J. Comput.*, 2005, **34**(5), 1159–1175.
72. Giannakis, K. and Andronikos, T., Use of Büchi automata and randomness for the description of biological processes. *Int. J. Sci. World*, 2015, **3**(1), 113–123.
73. Serre, O., Parity games played on transition graphs of one-counter processes. International conference on foundations of software science and computation structures. *LNCS*, 2006, **3921**, 337–351.
74. Alur, R. and Dill, D. L., A theory of timed automata. *Theor. Comput. Sci.*, 1994, **126**, 183–235.
75. Bollig, B. and Kuske, D., Muller message-passing automata and logics. *Inf. Comput.*, 2008, **206**(9–10), 1084–1094.
76. Henzinger, M. R. and Telle, J. A., Faster algorithms for the nonemptiness of Streett automata and for communication protocol pruning. Scan. Workshop Algorithm Theory. *LNCS*, 1996, **1097**, 16–27.

Received 6 October 2017; revised accepted 8 August 2018

doi: 10.18520/cs/v115/i11/2042-2051