

THESIS FOR THE DEGREE OF LICENTIATE OF PHILOSOPHY

Synthesis, Alternation and Algebra

Exploring Regular Systems and Games

PRABHAT KUMAR JHA

Department of Computer Science and Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden, 2026

Synthesis, Alternation and Algebra

Exploring Regular Systems and Games

PRABHAT KUMAR JHA

© Prabhat Kumar Jha, 2026
except where otherwise stated.
All rights reserved.

ISSN 1652-876X

Department of Computer Science and Engineering
Division of Computing Science
Formal Methods unit
Chalmers University of Technology | University of Gothenburg
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2026.

*“Even human compositions are acceptable when they are instructive
of reason; otherwise the Vedas also are to be renounced (as
unreliable); because men are required always to abide by reason.
Words conformable with reason are to be received even if spoken by
children; otherwise they are to be rejected as straws though
pronounced by the Lord Brahmaa.”*
- Vashishtha to Rama in Yoga Vashishtha 2.18.2-3

Synthesis, Alternation and Algebra

Exploring Regular Systems and Games

PRABHAT KUMAR JHA

Department of Computer Science and Engineering

Chalmers University of Technology | University of Gothenburg

Abstract

This thesis explores the role of *alternation*, the structured interplay between existential and universal choice, in synthesis and formal reasoning about reactive systems. The first contribution shows that the supervisory control synthesis problem reduces to solving Büchi games via a compact and uniform reduction, improving the known complexity bound and establishing a formal bridge to reactive synthesis. The second contribution introduces *Alternating Kleene Algebra* (AKA), a sound and complete equational axiomatization for regular expressions with intersection, resolving an open problem of Antimirov and achieving optimal proof complexity. Together, the two contributions argue that alternation enables computational efficiency in synthesis and games, and that its algebraic foundations can potentially offer a path to compositional reasoning in reactive systems.

Keywords

Regular languages, Alternation, Kleene Algebra, Supervisor, Synthesis, LTL, Proofs, Completeness

List of Publications

Appended publications

This thesis is based on the following publications:

- [**Paper I**] Daniel Hausmann, **Prabhat Kumar Jha**, and Nir Piterman,
Games for Efficient Supervisor Synthesis
IEEE Control Systems Letters, vol. 7, pp. 2881-2885, 2023. doi:
10.1109/LCSYS.2023.3290727
- [**Paper II**] **Prabhat Kumar Jha**, and Nir Piterman, *Alternating Kleene*
Algebra
Manuscript. To be submitted

Authorship Statement

[**Paper I**] Games for Efficient Supervisor Synthesis

This paper was co-authored with Daniel Hausmann and Nir Piterman.

The ideas were result of discussions of our regular meetings and I wrote the paper under supervision of Daniel and Nir. I wrote the formalization and proofs which were continuously reviewed by Daniel and Nir.

[**Paper II**] Alternating Kleene Algebra

This paper is co-authored with Nir Piterman. I proposed the problem and the solution. The ideas were actively discussed with (and reviewed by) Nir. I wrote the paper completely under supervision of Nir.

Acknowledgment

I express my gratitude towards everyone who has been a part of this journey and have supported, even if I miss their name here, which is impossible to avoid without making the acknowledgement bigger than the work itself.

To my advisor Nir Piterman, for everything; from our first interactions till now. None of the work mentioned here could be possible if not for his active support and mentorship. The journey so far has not been easiest but could not have been even possible, so thank you Nir!

To my co-advisor Daniel Hausmann, for his explanations, discussions, review and insights. To Sahar Mohajerani and Martin Fabian, for introducing me to discrete event systems.

To my buddies Abhiroop and Matthi, for helping me feel motivated enough to join this PhD program. To Matthi and Anandi, for visiting my family in Delhi and being their first contact here in case of any need. To Robert, Therese and Ninni, for our hangouts, discussions, cooking, eating, celebrations, etc. To Henrik and Julia for inviting me to their wedding. To Luca, for many meetups and conversations. To David, for sharing office coversations. To Lorenzo (and other constructivists), the LEM still holds. To Beata, for being in our gang. To the svenska ganget, for friendships of Katya, Wincent, Luis, Niklas, Tejas, Janek, Victor, Erik, Great Dane, Niek, Alex, Julius, Liam, (and increasing) ! Of course, for watching LOTR. To Tejas, for cooking some good oily food.

To Bin (and other christian missionary folks), for trying to convert me again and again. While it will never work, I learnt some lessons in persistence from you, so thanks.

To my Indian families in Gothenburg, for all your love and care!

To my TIFR and CMI friends, for their support in shaping me to a large extent. To my TIFR days mentors Piyush, Akshay and Paritosh, for checking my proofs and for writing recommendations. To RP and Sushmita, for board games, snacks and kindness. To my CMI days teachers B Srivathsan, M Srivas, (Late) S Nambiyar, Jam, for teaching me logic and writing reference letters for me on various occasions.

To my Shibpur days friends, thank you for standing by me when I was very vulnerable.

To my physics teacher and mentor Swargiya Shri D. P. Singh. I miss you a lot. Still recovering from the fact that you left in August 2026. Without your support, I would not have managed any step on this journey. Eternally

grateful to you, sir.

To my other mentors from Patna days, Shri Manoj Kumar and Shri A K Deo sir, you are my aspirations and heroes, thank you!

To my teachers from primary and secondary schools: Swargiya Shri Ishwar Narayan sir, Shri Sanjay sir, Shri Vinay sir, Shri Muzim sir, Shri Dhyani sir, Shri Mithilesh sir, Shri Radheshyam sir, etc. For teaching me things that were not parts of syllabi.

To my family for supporting me in the journey in all ways. To Atharv, for being a light in my life.

This work is supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Contents

Abstract	iii
List of Publications	v
Authorship Statement	vii
Acknowledgment	ix
 I Summary	 1
1 Introduction	3
1.1 Goals and Thesis Statement	3
1.2 Synthesis: Two paradigms	4
1.2.1 Reactive Synthesis	4
1.2.2 Supervisory Control	4
1.2.3 The Gap Between the Paradigms	4
1.3 Games as a Common Framework	4
1.4 Connecting the Two Paradigms: A Reduction	5
1.5 Composability and Its Challenges	6
1.6 Alternating Kleene Algebra	6
1.7 The Remaining Gap	7
 2 Summary of Included Papers	 9
2.1 Games for Efficient Supervisor Synthesis	9
2.2 Alternating Kleene Algebra	11
 Bibliography	 13
 II Appended Papers	 15
 Paper I - Games for Efficient Supervisor Synthesis	
 Paper II - Alternating Kleene Algebra	

Part I

Summary

Chapter 1

Introduction

1.1 Goals and Thesis Statement

The design of reliable systems increasingly relies on automatic synthesis: the construction of correct controllers directly from formal specifications, without manual proof effort. Two major research traditions have developed their own answers to this challenge, namely reactive synthesis and supervisory control, employing different formalisms, different specification languages, and different algorithmic techniques. Despite their structural similarity, the two fields have developed largely in isolation, and the connections between them remain incompletely understood.

This thesis proposes that the appropriate unifying perspective for both paradigms is *alternation*: the structured interplay between controllable and uncontrollable behaviour, and between existential and universal choices. Alternation elucidates the relationship between the two paradigms. The thesis statement is:

Alternation enables computational efficiency in synthesis and games, and its algebraic foundations can potentially offer a path to compositional reasoning in reactive systems.

The first contribution demonstrates that encoding supervisory control as a Büchi game yields an improved complexity bound and establishes a formal connection to reactive synthesis. The second contribution shows that equipping Kleene Algebra with an intersection operator, the algebraic counterpart of universal choices, can be equipped with a complete Kozen-style proof system, and begins to build the algebraic infrastructure that compositional synthesis requires.

1.2 Synthesis: Two paradigms

1.2.1 Reactive Synthesis

Reactive systems are systems that continuously interact with a live environment rather than terminating. *Reactive synthesis* is the automatic construction of a controller for a reactive system from a formal specification of desired behaviour. Specifications are typically expressed in logics such as Linear Temporal Logic (LTL) or Computation Tree Logic (CTL). The synthesis problem is commonly solved by modelling the interaction of system and environment as a two-player game; winning strategies in such games correspond to the control structure of the desired system. One widely studied game objective is the Büchi objective, which requires a designated set of arena states to be visited infinitely often. Büchi games can be solved in $O(n^2)$ time Chatterjee and Henzinger 2012.

1.2.2 Supervisory Control

Supervisory control theory Ramadge and Wonham 1989 is concerned with the control of *discrete-event systems*. A *plant* is a finite-state transition system in which the event set is partitioned into *controllable* events, which a supervisor may enable or disable, and *uncontrollable* events, which occur whenever they are physically possible regardless of supervisor action. A *supervisor* observes the history of the plant and restricts the set of permitted controllable events in order to steer the plant towards desired behaviour.

Desired behaviour is specified by a set of *marked states* representing successful or acceptable configurations. A supervisor is *non-blocking* if, from every state reachable under its control, some marked state can still be reached. The synthesis problem asks for a supervisor that is both non-blocking and *maximally permissive*: one that disables as few controllable events as possible while maintaining the non-blocking property. The standard algorithm of Ramadge and Wonham Ramadge and Wonham 1989 computes the supervisor by iterative backward reachability, running in time $O(n^2e)$ where n is the number of plant states and e is the number of events.

1.2.3 The Gap Between the Paradigms

Both paradigms involve a controller interacting with an environment, and both are concerned with the reachability or liveness of the resulting closed-loop system. Nevertheless, the two communities have maintained largely separate literatures with distinct specification languages and different algorithmic traditions, despite working on problems that are structurally the same. Formalizing the connection between them is one of the central goals of this thesis.

1.3 Games as a Common Framework

Like reactive synthesis, supervisory control can be expressed as finding a winning strategy in a two-player game on a finite graph. One player represents

the controller (or supervisor) and the other represents the environment (or the uncontrollable events of the plant). A *play* is an infinite path through the game arena; a *strategy* for the controller prescribes which edge to follow at each controller vertex; and a strategy is *winning* if every play consistent with it satisfies a designated winning condition.

The game-theoretic connection between supervisory control and reactive synthesis was studied systematically by Ehlers, Lafortune, Tripakis, and Vardi Ehlers et al. 2017. They observed that both fields rely on iterative backward and forward searches in graph structures, and proposed a game-theoretic framework as a common substrate for cross-fertilization. In their comparative introduction, they showed that the supervisory control synthesis problem can be encoded as a two-player game with a graph of size $O(ne)$, achieved by encoding the supervisor’s choices of permitted controllable events linearly as game transitions. Their encoding handles three distinct types of game vertices depending on the properties of the corresponding plant states, and introduces additional structure to track controllable choices faithfully. A more important consequence of their encoding concerns the form of the winning condition. The non-blocking property, in their framework, cannot be expressed as a standard linear winning condition such as Büchi. Instead, it requires a branching-time specification, since the correctness criterion asks that, from every reachable state, *some* path reaches a marked state; this existential path quantifier interacts with the interleaving of controllable and uncontrollable events in a way that is not captured by any single linear play. Using their approach to solve the supervisory control problem results in a time complexity of $O(n^2e^3)$, which is worse than the direct Ramadge-Wonham algorithm. The same paper, and subsequent personal communications, implied that the non-blocking condition is inherently non-linear and cannot be captured by a linear winning condition. This conviction became a widely held belief in the area.

1.4 Connecting the Two Paradigms: A Reduction

Given the common game-theoretic substrate, a natural question is whether supervisory control can be *formally reduced* to reactive synthesis, and specifically to solving a game with a standard linear winning condition. The analysis of Ehlers et al. Ehlers et al. 2017, together with earlier work of Schmuck, Moor, and Schmidt Schmuck, Moor and Schmidt 2020, which encoded all supervisor choices explicitly and obtained a game of exponential size, suggested that no compact reduction to a linearly-conditioned game exists.

First contribution of this thesis disproves this belief. The key insight is a change of perspective: rather than encoding the supervisor’s choice of which controllable events to permit, the game arena encodes the environment’s choice of whether to cede control to the supervisor at all. This shift gives a game arena of size exactly $2n$, with two vertices per plant state, that is entirely uniform in the sense that every plant state is handled identically regardless of its properties. Over this arena, the non-blocking winning condition is shown to

be equivalent to a Büchi winning condition on the same arena. The maximally permissive non-blocking supervisor can therefore be read off from the winning region of a Büchi game, and the $O(n^2)$ Büchi algorithm of Chatterjee and Henzinger Chatterjee and Henzinger 2012 gives an improved overall complexity of $O(\max(n^2, ne))$ for supervisor synthesis.

This reduction is a concrete instance of the thesis statement: the alternating two-vertex encoding of each plant state, one vertex for uncontrollable behaviour and one for controllable, is precisely what linearizes the non-blocking condition into a Büchi condition. Alternation absorbs the complexity rather than externalizing it into a non-linear winning condition.

1.5 Composability and Its Challenges

The reduction mentioned in previous section establishes a clean algorithmic connection between the two paradigms. In practice, plants are rarely described as monolithic transition systems. A complex system is naturally modelled as the parallel composition of simpler components. If two components have n_1 and n_2 states respectively, their product plant has $n_1 \cdot n_2$ states; for k components each of size n , the product plant has n^k states. Monolithic synthesis on this product plant is computationally infeasible at scale. *Compositional supervisor synthesis* attempts to compute supervisors for individual components and combine them, but doing so correctly requires reasoning about how the closed-loop language of the composed system relates to the closed-loop languages of the components. The key operation is *intersection*: the closed-loop language of the composed system is the intersection of the component closed-loop languages, and a composed supervisor is correct if and only if this intersection satisfies the combined specification.

The game-theoretic approach of Paper A does not help here. Winning strategies for a product game do not decompose into winning strategies for component games: global interactions between components are encoded in the joint state space and cannot be recovered from individual game structures. Games, as a model, do not compose naturally.

The algebraic theory of regular languages offers a contrasting picture. Equational reasoning about regular languages is inherently compositional: an equality between language expressions can be established component by component and assembled into proofs about the composed system. If intersection could be handled within this algebraic framework, compositional reasoning about synthesis would become tractable. Standard Kleene Algebra, however, does not include intersection as a native operator. Extending Kleene Algebra with intersection, and providing a complete equational axiomatization, is therefore an important step towards algebraic foundations for compositional synthesis.

1.6 Alternating Kleene Algebra

Computation is closely linked with language classes. One of the most ubiquitous class of languages is *Regular languages*. These are languages of describing

specific repeating patterns. One of the ways to describe such a language is as an algebraic expression called *Regular expressions*.

Kleene Algebra gives a sound and complete proof system for regular expressions: any valid equality between two expressions is provable from the Kleene Algebra axioms alone, without appeal to the semantics of the expressions Kozen 1994a. This makes Kleene Algebra a practical tool for equational reasoning about programs and systems described by regular languages. The question is whether the same completeness result holds when regular expressions are enriched with intersection.

The difficulty is that while union distributes over concatenation, intersection does not, so new axioms are required and it is not obvious what they should be. Antimirov and Mosses Antimirov and Mosses 1995a studied rewriting rules for regular expressions with intersection, but their axiomatization was alphabet-dependent: the axioms referred explicitly to the underlying alphabet and were therefore non-equational in the standard sense. In particular, the valid identity $\alpha \cap \Sigma^* = \alpha$ was unprovable in their system. They left as an explicit open problem the existence of a complete, alphabet-independent equational axiomatization.

Second contribution of this thesis resolves this problem by introducing *Alternating Kleene Algebra* (AKA). AKA extends Kleene Algebra with an intersection operator $\otimes$ and additional axioms, none of which refer to the underlying alphabet. The algebra of regular languages with intersection is proved to be an AKA, and AKA is proved complete: any valid equality between regular expressions with intersection has a proof in AKA. The completeness proof algebraifies the product construction for deterministic finite automata, the standard automata-theoretic realization of intersection, by showing that the tensor product of transition matrices correctly models language intersection inside AKA.

AKA is the first step towards the algebraic infrastructure identified in Section 1.5. It provides equational reasoning about intersection of regular languages, which is the central operation required for compositional synthesis.

1.7 The Remaining Gap

The two papers together map the territory and identify the remaining gaps. Paper A shows that alternation enables an efficient game-theoretic treatment of supervisory control. Paper B shows that alternation admits a complete algebraic treatment at the level of regular languages. The gap between them is the missing algebraic theory of *alternating structures at the game level*, a theory that would support compositional reasoning about game-like systems with the same power that AKA supports reasoning about regular languages with intersection.

Concretely, AKA provides equational axioms for the intersection of regular languages as recognized by deterministic or nondeterministic finite automata. What is needed is an algebraic theory for dealing with *game-arenas*. Such a theory would bridge the gap between the contributions and compositional

synthesis.

Chapter 2

Summary of Included Papers

2.1 Games for Efficient Supervisor Synthesis

In this paper we reduce the supervisor synthesis problem to Büchi games.

Problem

The supervisory control synthesis problem asks for the construction of a maximally permissive non-blocking supervisor for a given plant. While the standard Ramadge–Wonham algorithm achieves $O(n^2e)$ complexity, the connections between supervisory control and reactive synthesis had remained algorithmically underexplored. Previous reductions to games either produced game graphs of exponential size in the number of events, because all possible supervisor choices were encoded as game vertices, or required non-linear winning conditions expressible only in branching-time logics, incurring a complexity of $O(n^2e^3)$. It was widely believed, and stated explicitly in the literature, that the non-blocking winning condition cannot be expressed as a linear winning condition such as Büchi. The problem is to find a compact, uniform reduction from supervisory control to a game with a standard linear winning condition, if one exists.

Scientific Contribution

The paper makes three contributions. First, it provides a compact and uniform reduction of the supervisory control synthesis problem to a non-blocking game of size $2n$, twice the number of plant states, in which every plant state is encoded identically as a pair of game vertices regardless of its type. Second, it proves that the non-blocking winning condition on this game arena is equivalent to a Büchi winning condition on the same arena, directly disproving the widely

held belief that the non-blocking condition is inherently non-linear. Third, applying the $O(n^2)$ Büchi game algorithm of Chatterjee and Henzinger to this arena gives an improved time complexity of $O(\max(n^2, ne))$ for supervisor synthesis, improving upon the Ramadge–Wonham algorithm.

Methodology

The reduction departs from the natural paradigm of encoding the supervisor's choice of which controllable events to permit. Instead, it encodes the *environment's* choice of whether to allow the supervisor to act. Given a plant $P = (X, x_0, X_m, E = E_c \uplus E_{uc}, \delta)$, the game arena G_P associates to each plant state $x \in X$ two vertices: an uncontrollable vertex $u(x)$, from which the environment either fires an uncontrollable event directly (moving to $u(x')$ for $\delta(x, e_{uc}) = x'$) or delegates control to the supervisor (moving to $c(x)$); and a controllable vertex $c(x)$, from which the supervisor fires any event (moving to $u(x')$ for any $\delta(x, e) = x'$). The arena has $|V| = 2|X|$ vertices and its construction is entirely uniform.

The equivalence between non-blocking and Büchi winning conditions is established in two lemmas. First we prove that the maximal winning strategy for the non-blocking game G_P^m (with marked vertices $V_m = \{c(x_m), u(x_m) \mid x_m \in X_m\}$) corresponds exactly to the maximally permissive non-blocking supervisor for P .

Then we prove that a vertex is in the maximal non-blocking winning strategy if and only if it is in the winning region of the Büchi game G_P^B , whose marked edges are all direct uncontrollable transitions together with both edges incident to the copies of each marked state. The forward direction constructs a Büchi strategy from the non-blocking strategy using a distance function measuring progress towards marked states; the backward direction constructs a non-blocking strategy from the Büchi winning region. Finally, we get the $O(\max(n^2, ne))$ complexity bound.

2.2 Alternating Kleene Algebra

In this paper we develop a complete axiomatization for regular languages with intersections.

Problem

Kleene Algebra (KA) is sound and complete for the equational theory of regular expressions: any valid equality between expressions is provable from the KA axioms. However, KA does not include intersection, which arises naturally in hardware specification languages, database query languages, and compositional reasoning about regular systems. Antimirov and Mosses studied rewriting rules for regular expressions with intersection but their axiomatization was alphabet-dependent and therefore not equational in the standard sense; in particular, the valid identity $\alpha \cap \Sigma^* = \alpha$ was not provable. They left as an explicit open problem the existence of a complete, alphabet-independent equational axiomatization.

Scientific Contribution

The paper makes three contributions. First, it introduces *Alternating Kleene Algebra* (AKA) as a sound and complete alphabet-independent equational axiomatization for regular expressions with intersection, resolving Antimirov's open problem. The AKA proves the valid sentences which are valid irrespective of the underlying alphabet. Second, it extends the AKA to Σ -AKA while keeping the list of axioms constant. The Σ -AKA proves the valid sentences which are true for a specific alphabet Σ . Third, it applies AKA to rewriting verification: checking whether a candidate rewriting of a regular expression is valid.

Methodology

AKA is defined as a structure $(K, +, \otimes, \cdot, *, 0, 1)$ where $(K, +, \cdot, *, 0, 1)$ is a KA and the intersection operator $\otimes$ satisfies eight additional axioms: absorption of 0 ($\alpha \otimes 0 = 0$), idempotency ($\alpha \otimes \alpha = \alpha$), commutativity, associativity, right-distributivity over $+$ ($(\alpha + \beta) \otimes \gamma = \alpha \otimes \gamma + \beta \otimes \gamma$), compatibility with the KA order ($\alpha + \beta = \beta \iff \alpha \otimes \beta = \alpha$), a weak distributivity with concatenation ($(\alpha \cdot \gamma) \otimes (\beta \cdot \delta) \geq (\alpha \otimes \beta) \cdot (\gamma \otimes \delta)$), and an axiom governing $\otimes$ on the identity ($1 \otimes \alpha \beta = (1 \otimes \alpha) \cdot (1 \otimes \beta)$). These are supplemented by alphabet-specific lemmas describing the interaction of individual letters with $\otimes$.

The completeness proof algebraifies the product construction for DFAs. Automata are represented as triples (u, M, v) with recognized language $u^T M^* v$. The central lemma proves that $(u_\alpha^T M_\alpha^* v_\alpha) \otimes (u_\beta^T M_\beta^* v_\beta) = (u_\alpha \otimes u_\beta)^T (M_\alpha \otimes M_\beta)^* (v_\alpha \otimes v_\beta)$ is a theorem of AKA, showing that the tensor product of matrices correctly models language intersection inside the algebra. The proof uses the Mixed-Product Property for simple matrices, which shows $(A \cdot B) \otimes (C \cdot D) = (A \otimes C) \cdot (B \otimes D)$ for matrices with entries that are sums of distinct alphabet letters, together with a vector form of Arden's Lemma and induction on

matrix powers. Completeness follows by uniqueness of minimal DFAs: any valid equality can be witnessed by a DFA isomorphism and translated into an algebraic proof.

Bibliography

- Abiteboul, Serge, Richard Hull and Victor Vianu (1995). *Foundations of Databases*. Addison-Wesley. ISBN: 0-201-53771-0 (cit. on p. 2 (II)).
- Anderson, Carolyn Jane, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger and David Walker (Jan. 2014). ‘NetKAT: Semantic Foundations for Networks’. In: *SIGPLAN Notices* 49.1, pp. 113–126 (cit. on pp. 2 (II), 3 (II)).
- Antimirov, Valentin M. and Peter D. Mosses (1995a). ‘Rewriting extended regular expressions’. In: *Theoretical Computer Science* 143.1, pp. 51–72. DOI: 10.1016/0304-3975(94)00214-Z (cit. on p. 7).
- (1995b). ‘Rewriting Extended Regular Expressions’. In: *Theoretical Computer Science* 143.1, pp. 51–72 (cit. on pp. 2 (II), 3 (II)).
- Arden, Dean N. (1961). ‘Delayed-logic and finite-state machines’. In: *2nd Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1961)*, pp. 133–151. DOI: 10.1109/FOCS.1961.13 (cit. on pp. 12 (II), 12 (II)).
- Calvanese, Diego, Giuseppe De Giacomo, Maurizio Lenzerini and Moshe Y. Vardi (2002). ‘Rewriting of Regular Expressions and Regular Path Queries’. In: *Journal of Computer and System Sciences* 64.3, pp. 443–465 (cit. on pp. 4 (II), 19 (II)).
- Chandra, Ashok K., Dexter C. Kozen and Larry J. Stockmeyer (Jan. 1981). ‘Alternation’. In: *Journal of the ACM* 28.1, pp. 114–133 (cit. on p. 3 (II)).
- Chatterjee, Krishnendu and Monika Henzinger (2012). ‘An $O(n^2)$ time algorithm for alternating Büchi games’. In: *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, pp. 1386–1399. DOI: 10.1137/1.9781611973099.109 (cit. on pp. 4, 6).
- Conway, John H. (1971). *Regular Algebra and Finite Machines*. Chapman and Hall (cit. on p. 6 (II)).
- Ehlers, Rüdiger, Stéphane Lafortune, Stavros Tripakis and Moshe Y. Vardi (2017). ‘Supervisory control and reactive synthesis: a comparative introduction’. In: *Discrete Event Dynamic Systems* 27.2, pp. 209–260. DOI: 10.1007/s10626-016-0232-7 (cit. on p. 5).
- Eisner, Cindy and Dana Fisman (2006). *A Practical Introduction to PSL*. Series on Integrated Circuits and Systems. Springer. ISBN: 0-387-26018-5 (cit. on p. 2 (II)).
- Fürer, Martin (1980). ‘The Complexity of the Inequivalence Problem for Regular Expressions with Intersection’. In: *Proceedings of the 7th Colloquium on*

- Automata, Languages and Programming*. Berlin, Heidelberg: Springer-Verlag, 234–245. ISBN: 3540100032 (cit. on pp. 3 (II), 3 (II)).
- Gelade, Wouter and Frank Neven (Jan. 2012). ‘Succinctness of the Complement and Intersection of Regular Expressions’. In: *ACM Trans. Comput. Logic* 13.1. ISSN: 1529-3785. DOI: 10.1145/2071368.2071372. URL: <https://doi.org/10.1145/2071368.2071372> (cit. on pp. 3 (II), 3 (II), 3 (II)).
- Hunt III, Harry B (1973). *The equivalence problem for regular expressions with intersection is not polynomial in tape*. Tech. rep. Cornell University (cit. on pp. 3 (II), 3 (II)).
- Kozen, Dexter (1994a). ‘A completeness theorem for Kleene algebras and the algebra of regular events’. In: *Information and Computation* 110.2, pp. 366–390. DOI: 10.1006/inco.1994.1037 (cit. on p. 7).
- (1994b). ‘A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events’. In: *Information and Computation* 110.2, pp. 366–390 (cit. on pp. 2 (II), 2 (II), 3 (II), 5 (II), 5 (II), 7 (II), 7 (II), 9 (II), 18 (II), 18 (II), 18 (II)).
- Kupferman, Orna, Moshe Y. Vardi and Pierre Wolper (Mar. 2000). ‘An automata-theoretic approach to branching-time model checking’. In: *J. ACM* 47.2, 312–360. ISSN: 0004-5411. DOI: 10.1145/333979.333987. URL: <https://doi.org/10.1145/333979.333987> (cit. on p. 3 (II)).
- Ong, Shawn, Stephanie Ma and Dexter Kozen (June 2025). ‘Probabilistic Kleene Algebra with Angelic Nondeterminism’. In: *Proceedings of the ACM on Programming Languages* 9.PLDI (cit. on p. 3 (II)).
- Ramadge, P. J. and W. M. Wonham (1989). ‘The control of discrete event systems’. In: *Proceedings of the IEEE* 77.1, pp. 81–98. DOI: 10.1109/5.21072 (cit. on p. 4).
- Różowski, Wojciech and Alexandra Silva (2024). ‘A Completeness Theorem for Probabilistic Regular Expressions’. In: *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS ’24)*. New York, NY, USA: Association for Computing Machinery (cit. on p. 3 (II)).
- Salomaa, Arto (Jan. 1966). ‘Two Complete Axiom Systems for the Algebra of Regular Events’. In: *Journal of the ACM* 13.1, pp. 158–169 (cit. on p. 2 (II)).
- Schmuck, Anne-Kathrin, Thomas Moor and Klaus Werner Schmidt (2020). ‘A reactive synthesis approach to supervisory control of terminating processes’. In: *21st IFAC World Congress*. Vol. 53. IFAC-PapersOnLine 2, pp. 2149–2156. DOI: 10.1016/j.ifacol.2020.12.2573 (cit. on p. 5).
- Thompson, Ken (1968). ‘Programming Techniques: Regular expression search algorithm’. In: *Communications of the ACM* 11.6, pp. 419–422. DOI: 10.1145/363347.363387 (cit. on pp. 2 (II), 2 (II)).