Awesome Automata

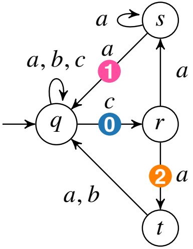
Algorithms and Applications

Ondřej Lengál

Brno University of Technology, Czech Republic

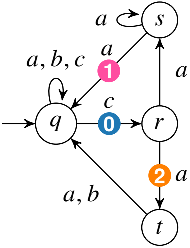
Habilitation (Scientific Council)

Automata in Computer Science



- (finite) automata one of the cornerstones of computer science
  - regex matching, parsing, state space representation, decision proc., . . .

Automata in Computer Science

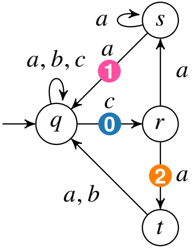


- (finite) automata one of the cornerstones of computer science
  - regex matching, parsing, state space representation, decision proc., . . .
- I am addressing two questions:

Q1: How to effectively use automata in applications?

- How can various automata be used for modelling different concepts?
  - memory states, system configurations, . . .
- Which automata model to use?
  - automata over finite/infinite words/trees/graphs/. . .

Automata in Computer Science



- (finite) automata one of the cornerstones of computer science
  - regex matching, parsing, state space representation, decision proc., . . .
- I am addressing two questions:

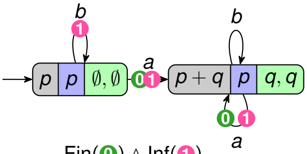
Q1: How to effectively use automata in applications?

- How can various automata be used for modelling different concepts?
  - memory states, system configurations, . . .
- Which automata model to use?
  - automata over finite/infinite words/trees/graphs/. . .

Q2: How to handle automata efficiently?

- many operations are hard (e.g., inclusion/equivalence)
  - PSPACE/EXPTIME/UNDEC for finite-state word/tree/weighted automata
  - $\rightsquigarrow$ practical algorithms
- how to achieve a compact representation?
  - e.g., adding bounded counters, registers, colours, synchronization, . . .

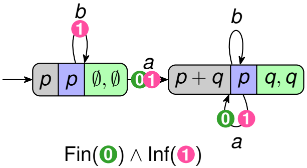
Büchi Automata Complementation



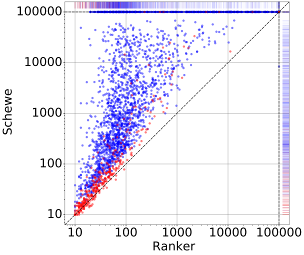
Fin(0) ∧ Inf(1)

- Büchi automata — finite-state automata over infinite words
- often used in formal methods:
  - modelling/model checking of reactive systems, deciding logics, program termination, . . .
- complementation: basic operation
  - implementation of negation in decision procedures and model checking
  - removal of traces in program verification
  - underlying operation for inclusion checking
  - complexity: $\mathcal{O}((0.76n)^n)$

Büchi Automata Complementation

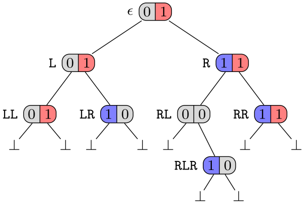


- Büchi automata — finite-state automata over infinite words
- often used in formal methods:
  - modelling/model checking of reactive systems, deciding logics, program termination, ...
- complementation: basic operation
  - implementation of negation in decision procedures and model checking
  - removal of traces in program verification
  - underlying operation for inclusion checking
  - complexity: $\mathcal{O}((0.76n)^n)$
- significantly improved the SOTA in Büchi complementation
  - rank-based [APLAS'19,CONCUR'21,TACAS'22,CAV'22]
  - elevator automata [TACAS'22,TACAS'23]
  - mix-and-match complementation algorithm [TACAS'23]
  - complementation of Emerson-Lei automata [FoSSaCS'25]



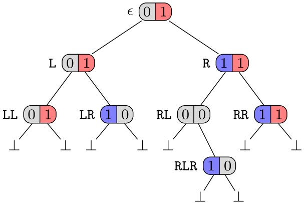
together with V. Havlena, B. Šmahlíková, Y. Li, A. Turrini, O. Alexaj, Y. Chen

Automata-Logic Connection



- finite word/tree automata used for deciding certain logics
  - monadic second-order logics over finite/infinite words/trees
  - first/second-order logics over automatic structures
  - first-order Presburger arithmetic (linear integer arithmetic)
- complexity: 2-NEXP-hard, TOWER-complete, UNDEC
- $\rightsquigarrow$ powerful heuristics necessary

Automata-Logic Connection



- finite word/tree automata used for deciding certain logics
  - monadic second-order logics over finite/infinite words/trees
  - first/second-order logics over automatic structures
  - first-order Presburger arithmetic (linear integer arithmetic)
- complexity: 2-NEXP-hard, TOWER-complete, UNDEC
- $\rightsquigarrow$ powerful heuristics necessary
- new techniques of symbolic reasoning over automata-based representation
  - nested antichains for weak monadic second-order logic WS1S [TACAS'15]
  - lazy automata techniques for WS1S/WS*k*S [TACAS'17,Acta'19,CADE'19,JAR'21]
  - algebraic reasoning with automata for Presburger arithmetic [CAV'24]
    - solver Amaya (several medals in NIA category of SMT-COMP'24)

together with L. Holík, V. Havlena, M. Hečko, T. Fiedor, T. Vojnar, P. Habermehl

Theory of Strings

- Satisfiability of formulae over string constraints such as:

$$\underbrace{x = yz}_{\text{equations}} \land \overbrace{yz \neq ua}^{\text{disequalities}} \land \underbrace{x \in (ab)^* a^+ (b|c)}_{\text{regular constraints}} \land \overbrace{|xy| = 2|uv| + 1}^{\text{length constraints}} \land \underbrace{\neg contains(uxz, zbcx)}_{\text{more complex operations}}$$

- Reasoning about string manipulation in programs
  - detecting security vulnerabilities, analysis of scripting languages, . . .

Theory of Strings

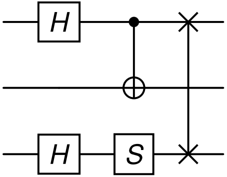
- Satisfiability of formulae over string constraints such as:

$$\underbrace{x = yz}_{\text{equations}} \land \overbrace{yz \neq ua}^{\text{disequalities}} \land \underbrace{x \in (ab)^* a^+ (b|c)}_{\text{regular constraints}} \land \overbrace{|xy| = 2|uv| + 1}^{\text{length constraints}} \land \underbrace{\neg contains(uxz, zbcx)}_{\text{more complex operations}}$$

- Reasoning about string manipulation in programs
  - detecting security vulnerabilities, analysis of scripting languages, ...
- automata-based string solving
  - compact encoding of constraints obtained in solving [APLAS'20,JSS'23]
  - stabilization-based string solving [FM'23,OOPSLA'23,TACAS'24,SAT'24,TACAS'25]
    - solver Z3-Noodler (winner of Strings category of SMT-COMP'24 under all scoring schemes)
  - automata-based handling of position constraints [PLDI'25]

together with L. Holík, V. Havlena, J. Síč, D. Chocholatý, M. Hečko, F. Blahoudek, Y. Chen

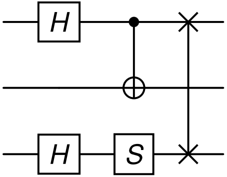
Analysis of Quantum Circuits

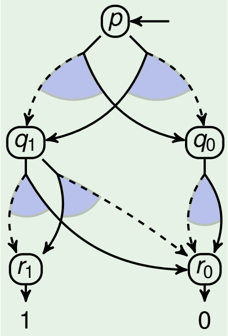


- quantum computation getting more traction
- quantum circuits notoriously hard to reason about
- current techniques: imprecise, highly manual, or do not scale

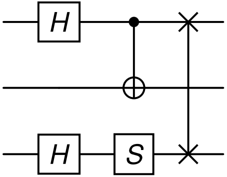
Analysis of Quantum Circuits

- quantum computation getting more traction
- quantum circuits notoriously hard to reason about
- current techniques: imprecise, highly manual, or do not scale
- new tree automata-based analysis framework [PLDI'23]
  - uses tree automata (TAs) for encoding sets of quantum states
  - efficient fully automated verification of a significant class of quantum circuits
  - stimulated development of novel formal models and algorithms
    - level-synchronized TAs: more compact encoding, parameterized verification of circuits with bounded superpositions [POPL'25]
    - symbolic (LS)TAs: verification for any amplitude values, while loops, simulation acceleration [CAV'23,ICCAD'24,TACAS'25]
    - weighted synchronized TAs: parameterized verification of circuits with unbounded superposition degree [WIP]

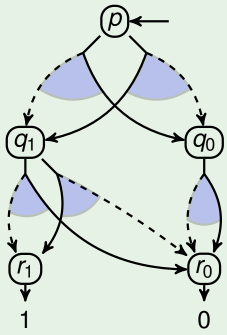




Analysis of Quantum Circuits



- quantum computation getting more traction
- quantum circuits notoriously hard to reason about
- current techniques: imprecise, highly manual, or do not scale
- new tree automata-based analysis framework [PLDI’23]
  - uses tree automata (TAs) for encoding sets of quantum states
  - efficient fully automated verification of a significant class of quantum circuits
  - stimulated development of novel formal models and algorithms
    - level-synchronized TAs: more compact encoding, parameterized verification of circuits with bounded superpositions [POPL’25]
    - symbolic (LS)TAs: verification for any amplitude values, while loops, simulation acceleration [CAV’23,ICCAD’24,TACAS’25]
    - weighted synchronized TAs: parameterized verification of circuits with unbounded superposition degree [WIP]



- Commun. ACM Research Highlight (June 2025)
  - “An Automata-Based Framework for Verification and Bug Hunting in Quantum Circuits”

together with Y. Chen, S. Jobranová, W. Tsai, K. Chung, R. Jiang, T. Chen, J. Lin, L. Holík, P. Abdulla . . .

Finite Automata in Network Intrusion Detection

- regexes used to describe network intrusions $\rightsquigarrow$ automata
- $>$ 100 Gbps networks: HW acceleration needed $\rightsquigarrow$ FPGAs
- limited resources!

Finite Automata in Network Intrusion Detection

- regexes used to describe network intrusions $\rightsquigarrow$ automata
- $>$ 100 Gbps networks: HW acceleration needed $\rightsquigarrow$ FPGAs
- limited resources!

- exact automata reductions insufficient
- $\rightsquigarrow$ approximate automata reduction
  - automata reduction w.r.t. network traffic model [TACAS'18,STTT'20]
    - reduction by 50 % states with 8.7×10^{-8} error
  - automata reduction w.r.t. network traffic sample and multi-stage architecture [FCCM'19]
    - 17 Gbps $\rightsquigarrow$ 200 Gbps (0 % Err) $\rightsquigarrow$ 400 Gbps (4 % Err)

V. Havlena, T. Vojnar, L. Holík, M. Češka, J. Semrič, J. Kořenek, D. Matoušek, J. Matoušek

Main Numerical Indicators (1/2)

- 54 publications at international conferences and journals
  - CORE A*: 12, CORE A: 27, CORE B: 5, unranked: 3
  - SJR Q1: 2, SJR Q2: 5
- 256 citations (SCOPUS w/o self-citations from all authors)
- h-index 11 (SCOPUS)
- 26× PC member of conferences/workshops
  - e.g. CAV'25, DATE'24, ATVA'19/20/23, SPIN'24/25, . . .
- 124 paper reviews
- 3× (co-)charing artifact evaluation committees
  - TACAS'19, ATVA'19, ESOP/FASE/FoSSaCS'25
- 1× PC co-chair (VMCAI'26)

Main Numerical Indicators (2/2)

- 4 best/distinguished papers at A/A* conferences
  - [CADE’19,FM’23,PLDI’23,OOPSLA’23]
- Commun. ACM Research Highlight (June 2025)
  - “An Automata-Based Framework for Verification and Bug Hunting in Quantum Circuits” [PLDI’23]
- participation in the development of tools in competitions
  - Z3-Noodler (winner of Strings @ SMT-COMP’24),
  - Amaya (multiple medals in NIA @ SMT-COMP’24),
  - SPEN (multiple medals at SL-COMP’14 and SL-COMP’18)
- PI of 2 GAČR grants, team member of 23 other grants
- Teaching:
  - Introduction to Logic for Computer Science (IZLO), Operating Systems (IOS), Advanced Mathematics (IAM), Static Analysis and Verification (SAV), Complexity Theory (SLOa), Theoretical Computer Science (TIN), Discrete Mathematics (IDM), Mathematical Seminar (SMT), Software Engineering (IUS)
  - 33 finished BSc/MSc theses, 1 finished PhD thesis