Complementing Büchi Automata with Ranker

Vojtěch Havlena Ondřej Lengál **Barbora Šmahlíková**

Brno University of Technology, Czech Republic

CAV'22

BA Complementation

Complementation:

- Given $\mathcal{A}$, get a BA $\mathcal{A}^{\complement}$ such that $\mathcal{L}(\mathcal{A}^{\complement}) = \overline{\mathcal{L}(\mathcal{A})}$
- We consider state/transition based Büchi automata

BA Complementation

Complementation:

- Given $\mathcal{A}$, get a BA $\mathcal{A}^{\complement}$ such that $\mathcal{L}(\mathcal{A}^{\complement}) = \overline{\mathcal{L}(\mathcal{A})}$
- We consider state/transition based Büchi automata

Motivation:

- Model checking of linear-time properties

$$\underbrace{S}_{\text{system}} \models \underbrace{\varphi}_{\text{property}} \rightsquigarrow \quad \mathcal{L}(\mathcal{A}_S) \subseteq \mathcal{L}(\mathcal{A}_\varphi) \quad \rightsquigarrow \quad \mathcal{L}(\mathcal{A}_S) \cap \mathcal{L}(\mathcal{A}_\varphi^{\complement}) = \emptyset$$

BA Complementation

Complementation:

- Given $\mathcal{A}$, get a BA $\mathcal{A}^{\complement}$ such that $\mathcal{L}(\mathcal{A}^{\complement}) = \overline{\mathcal{L}(\mathcal{A})}$
- We consider state/transition based Büchi automata

Motivation:

- Model checking of linear-time properties

$$\underbrace{S}_{\text{system}} \models \underbrace{\varphi}_{\text{property}} \rightsquigarrow \quad \mathcal{L}(\mathcal{A}_S) \subseteq \mathcal{L}(\mathcal{A}_\varphi) \quad \rightsquigarrow \quad \mathcal{L}(\mathcal{A}_S) \cap \mathcal{L}(\mathcal{A}_\varphi^{\complement}) = \emptyset$$

- Termination analysis of programs: Ultimate Automizer

BA Complementation

Complementation:

- Given $\mathcal{A}$, get a BA $\mathcal{A}^{\complement}$ such that $\mathcal{L}(\mathcal{A}^{\complement}) = \overline{\mathcal{L}(\mathcal{A})}$
- We consider state/transition based Büchi automata

Motivation:

- Model checking of linear-time properties

$$\underbrace{S}_{\text{system}} \models \underbrace{\varphi}_{\text{property}} \quad \rightsquigarrow \quad \mathcal{L}(\mathcal{A}_S) \subseteq \mathcal{L}(\mathcal{A}_\varphi) \quad \rightsquigarrow \quad \mathcal{L}(\mathcal{A}_S) \cap \mathcal{L}(\mathcal{A}_\varphi^{\complement}) = \emptyset$$

- Termination analysis of programs: Ultimate Automizer
- Decision procedures: implements negation
  - S1S: MSO over $(\omega, 0, +1)$
  - QPTL: quantified propositional temporal logic
  - FO over Sturmian words

BA Complementation

Complementation:

- Given $\mathcal{A}$, get a BA $\mathcal{A}^{\complement}$ such that $\mathcal{L}(\mathcal{A}^{\complement}) = \overline{\mathcal{L}(\mathcal{A})}$
- We consider state/transition based Büchi automata

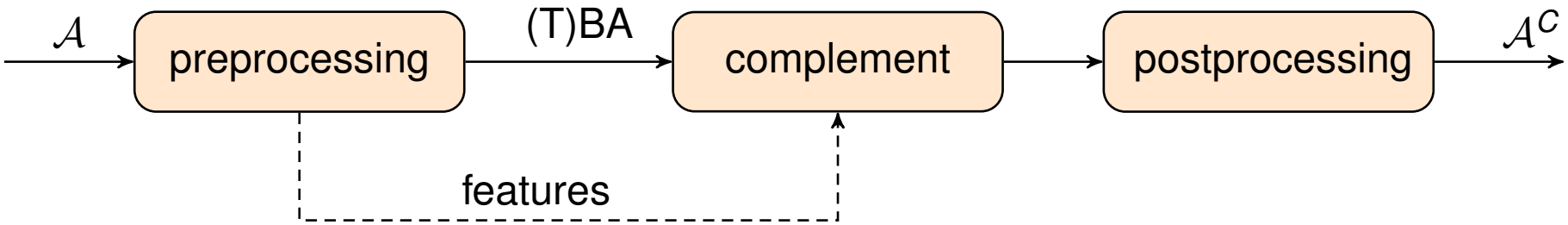
Motivation:

- Model checking of linear-time properties

$$\underbrace{S}_{\text{system}} \models \underbrace{\varphi}_{\text{property}} \quad \rightsquigarrow \quad \mathcal{L}(\mathcal{A}_S) \subseteq \mathcal{L}(\mathcal{A}_\varphi) \quad \rightsquigarrow \quad \mathcal{L}(\mathcal{A}_S) \cap \mathcal{L}(\mathcal{A}_\varphi^{\complement}) = \emptyset$$

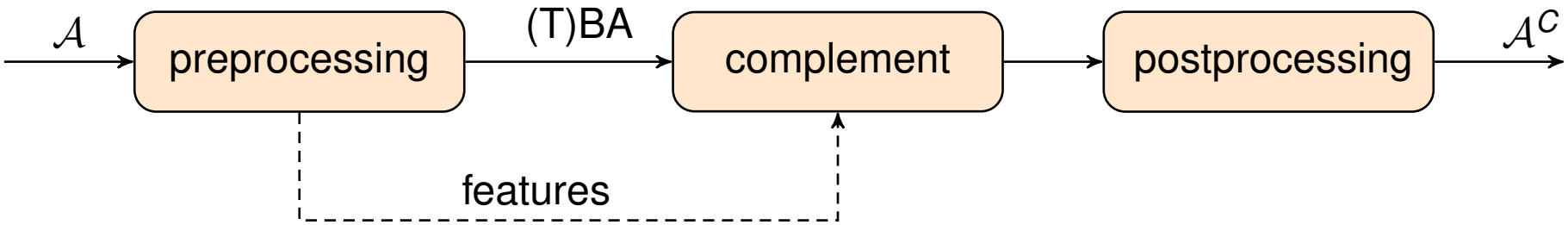
- Termination analysis of programs: Ultimate Automizer
- Decision procedures: implements negation
  - S1S: MSO over $(\omega, 0, +1)$
  - QPTL: quantified propositional temporal logic
  - FO over Sturmian words
- Basic operation for inclusion/equivalence checking

Ranker – architecture



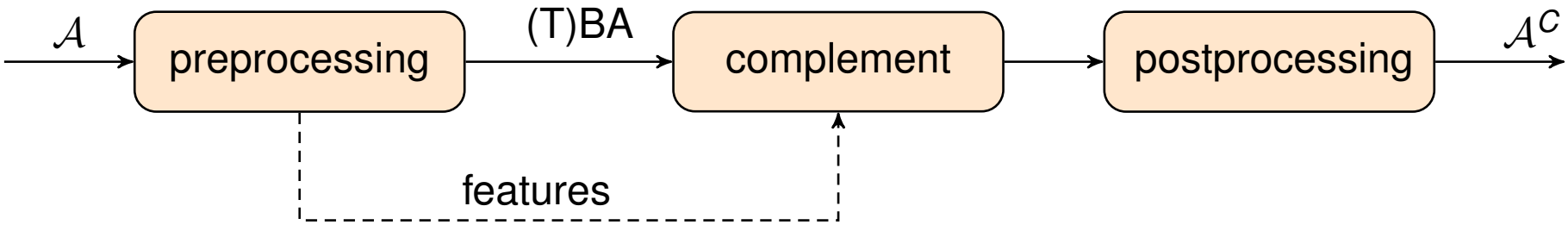
- Implemented in C++
- Input: HOA / BA, output: HOA

Ranker – architecture



- Implemented in C++
- Input: HOA / BA, output: HOA
- Preprocessing: direct simulation reduction, deelevation, saturation, feature extraction

Ranker – architecture



- Implemented in C++
- Input: HOA / BA, output: HOA
- Preprocessing: direct simulation reduction, deelevation, saturation, feature extraction
- Postprocessing: removing useless states, direct simulation reduction

Ranker – complementation

Inherently weak BAs:

- Every SCC is nonaccepting or it contains accepting state/transition in every cycle
- Optimized Miyano-Hayashi construction
- Simulation-based macrostates pruning/saturation

Ranker – complementation

Inherently weak BAs:

- Every SCC is nonaccepting or it contains accepting state/transition in every cycle
- Optimized Miyano-Hayashi construction
- Simulation-based macrostates pruning/saturation

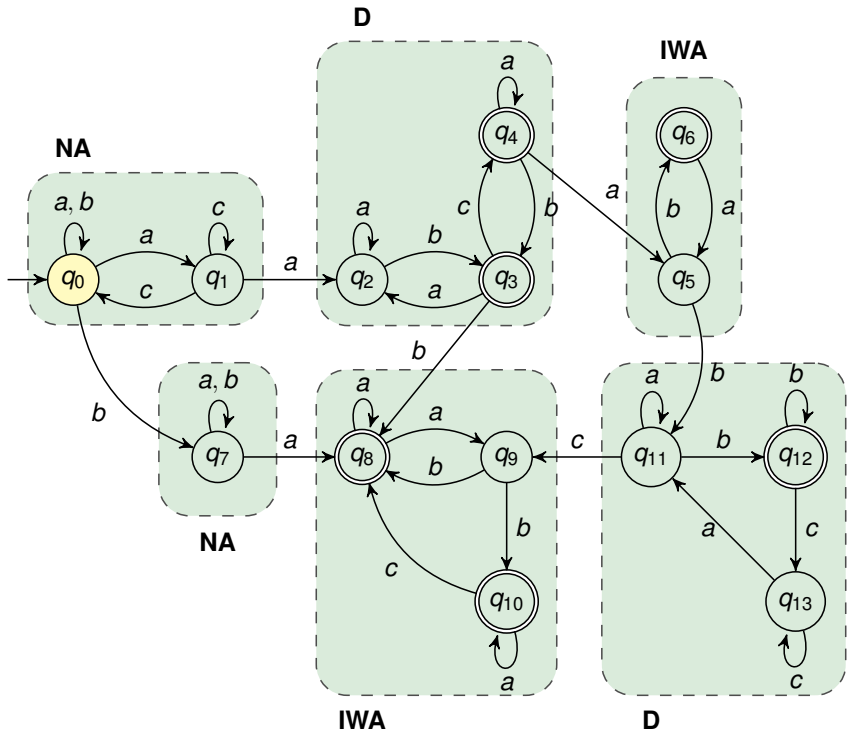
Semi-deterministic BAs:

- Nonaccepting and deterministic accepting part
- NCSB-MaxRank: optimized NCSB construction
- Nondeterminism reduction
- At most two successors for every macrostate and symbol

Ranker – complementation

Other BAs:

- Optimized rank-based construction
  - Friedgut, Kupferman, Vardi, Schewe
- Elevator automata[1]
  - Deterministic and inherently weak SCCs
  - Efficient procedure based on the structure
  - Extension to nonelevator automata
  - Deelevation: decreases the rank bound to 3



[1] Havlena, Lengál, and Šmahlíková. "Sky Is Not the Limit: Tighter Rank Bounds for Elevator Automata in Büchi Automata Complementation". In: *TACAS'22*.

Experimental Evaluation

- Random automata from [Tsai,Fogarty,Vardi,Tsay'11]
  - alphabet of 2 symbols
  - starting with 15 states
  - 4533 hard automata

Experimental Evaluation

- Random automata from [Tsai,Fogarty,Vardi,Tsay'11]
  - alphabet of 2 symbols
  - starting with 15 states
  - 4533 hard automata
- LTL automata
  - larger alphabets (up to 128 symbols)
  - from LTL formulae (from literature and randomly generated)
  - 1716 hard automata

Experimental Evaluation

- Random automata from [Tsai,Fogarty,Vardi,Tsay'11]
  - alphabet of 2 symbols
  - starting with 15 states
  - 4533 hard automata
- LTL automata
  - larger alphabets (up to 128 symbols)
  - from LTL formulae (from literature and randomly generated)
  - 1716 hard automata
- Automizer automata
  - BAs obtained from Ultimate Automizer
  - alphabets up to 2^{35} symbols
  - 906 hard automata

Experimental Evaluation

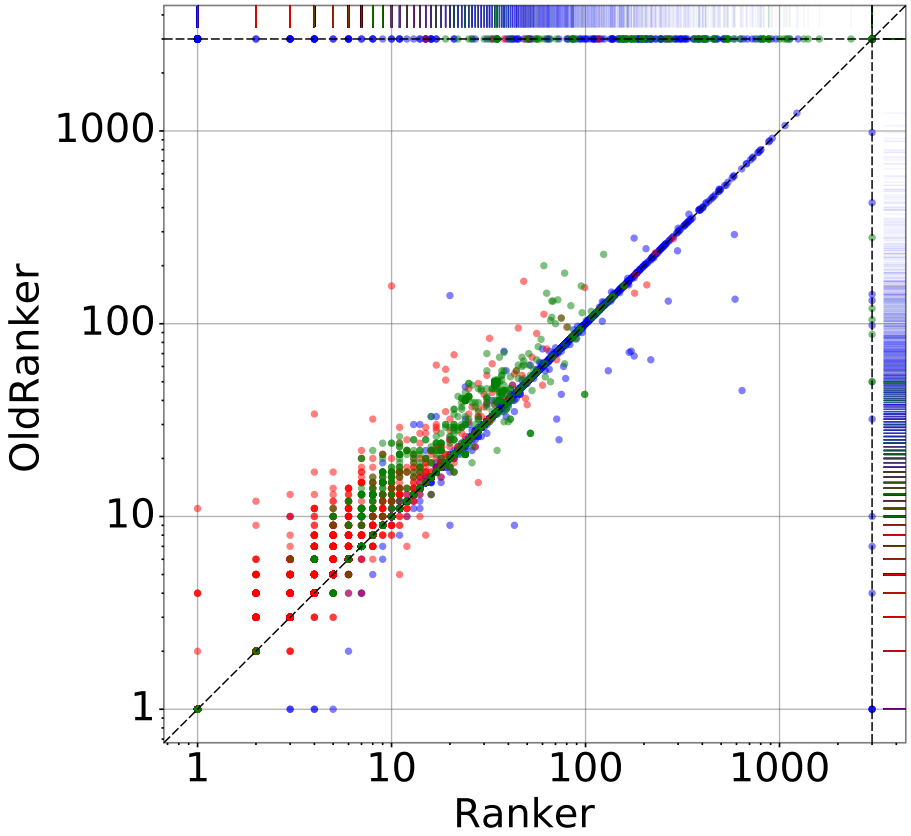
- Random automata from [Tsai,Fogarty,Vardi,Tsay'11]
  - alphabet of 2 symbols
  - starting with 15 states
  - 4533 hard automata
- LTL automata
  - larger alphabets (up to 128 symbols)
  - from LTL formulae (from literature and randomly generated)
  - 1716 hard automata
- Automizer automata
  - BAs obtained from Ultimate Automizer
  - alphabets up to 2^{35} symbols
  - 906 hard automata
- Total: 7155 state-based BAs, preprocessed with `autfilt`, timeout 5 min

Experimental Evaluation

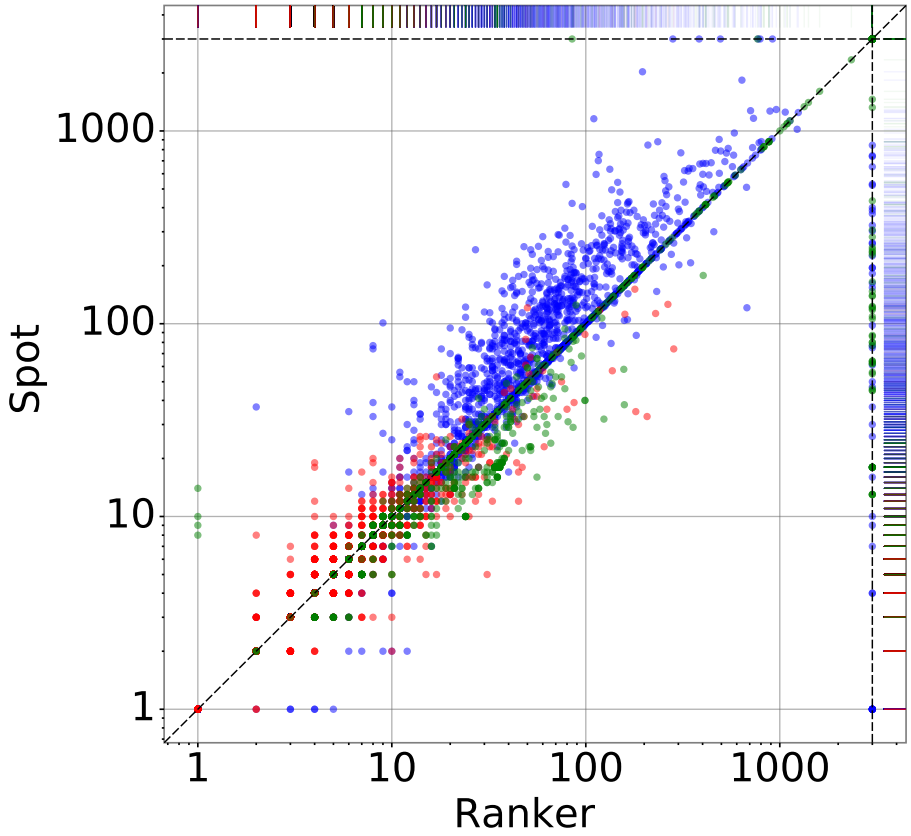
- RANKER compared with:
  - GOAL⚽ (SCHEWE, SAFRA, PITERMAN, FRIBOURG)
  - SPOT (Safra, Piterman, Redziejowski)
  - LTL2DSTAR
  - SEMINATOR 2
  - ROLL
- Focus on the number of states

`https://github.com/vhavlena/ranker`

Experimental Evaluation – States



(a) RANKER vs $\text{RANKER}_{\text{OLD}}$



(b) RANKER vs SPOT

- after postprocessing
- logarithmic axes

- **blue**: random
- **red**: LTL
- **green**: Automizer

Conclusion

- Timeouts decreased by 65 %
- RANKER has the smallest mean and median
- More wins than losses compared to any other tool

Conclusion

- Timeouts decreased by 65 %
- RANKER has the smallest mean and median
- More wins than losses compared to any other tool

Future work:

- Generalization to complementation of TELA
  - transition-based Emerson-Lei automata
- Language inclusion checking
- Decomposition-based complementation

Conclusion

- Timeouts decreased by 65 %
- RANKER has the smallest mean and median
- More wins than losses compared to any other tool

Future work:

- Generalization to complementation of TELA
  - transition-based Emerson-Lei automata
- Language inclusion checking
- Decomposition-based complementation

THANK YOU!

Experimental Evaluation – States

method	mean	median	wins	losses	timeouts
RANKER	38	11			158 (53 : 0 : 105)
RANKER$_{\text{OLD}}$	30	12	1554 (356 : 650 : 548)	264 (142 : 69 : 53)	458 (259 : 7 : 192)
PITERMAN ⚽	43	14	2881 (1279 : 966 : 636)	392 (263 : 68 : 61)	309 (12 : 4 : 293)
SAFRA ⚽	49	15	3109 (1348 : 1117 : 644)	274 (229 : 31 : 14)	599 (160 : 30 : 409)
SPOT	46	11	1347 (935 : 339 : 73)	1057 (327 : 343 : 387)	73 (13 : 0 : 60)
FRIBOURG ⚽	49	11	2223 (1177 : 503 : 543)	586 (245 : 207 : 134)	399 (93 : 2 : 304)
LTL2DSTAR	44	14	2794 (1297 : 924 : 573)	448 (283 : 88 : 77)	288 (130 : 13 : 145)
SEMINATOR 2	46	11	1626 (1297 : 291 : 38)	1113 (286 : 398 : 429)	419 (368 : 1 : 50)
ROLL	18	9	6050 (3824 : 1551 : 675)	620 (369 : 125 : 126)	1893 (1595 : 8 : 290)

Experimental Evaluation – Time

method	mean					median				
RANKER	3.72	(4.34	:	0.45	: 7.30)	0.05	(0.10	: 0.04	:	0.08)
RANKER$_{OLD}$	4.62	(5.33	:	0.72	: 9.69)	0.07	(0.19	: 0.03	:	0.15)
PITERMAN ⚽	8.06	(6.07	:	5.95	: 28.38)	5.12	(4.96	: 5.08	:	8.68)
SAFRA ⚽	11.58	(10.41	:	6.51	: 38.65)	5.41	(5.32	: 5.26	:	9.02)
SPOT	0.64	(0.57	:	0.02	: 2.28)	0.02	(0.02	: 0.01	:	0.02)
FRIBOURG ⚽	13.13	(14.14	:	6.06	: 23.88)	5.69	(6.82	: 4.92	:	6.57)
LTL2DSTAR	2.1	(2.25	:	0.34	: 5.15)	0.02	(0.02	: 0.01	:	0.05)
SEMINATOR 2	4.16	(6.33	:	0.03	: 1.88)	0.03	(0.08	: 0.01	:	0.03)
ROLL	23.65	(29.82	:	3.88	: 49.02)	3.34	(6.19	: 1.71	:	17.14)