Automata vs. linear-programming discounted-sum inclusion

Suguman Bansal, Rice U.

Swarat Chaudhuri, Rice U.

Moshe Y. Vardi, Rice U.

CAV 2018, FLoC 2018, Oxford, UK

July 17, 2018

Beyond qualitative reasoning …

Quantitative systems with quantitative properties

Quantitative reasoning

Algorithmic approaches

- Structural aspects - About the structure of systems
- Quantitative aspects - About quantities associated with systems

Algorithmic approaches

- Modular approach - Separates both reasonings
- Integrated approach - Combines both reasonings

Which approach is more viable in practice for Discounted-sum inclusion?

Contributions

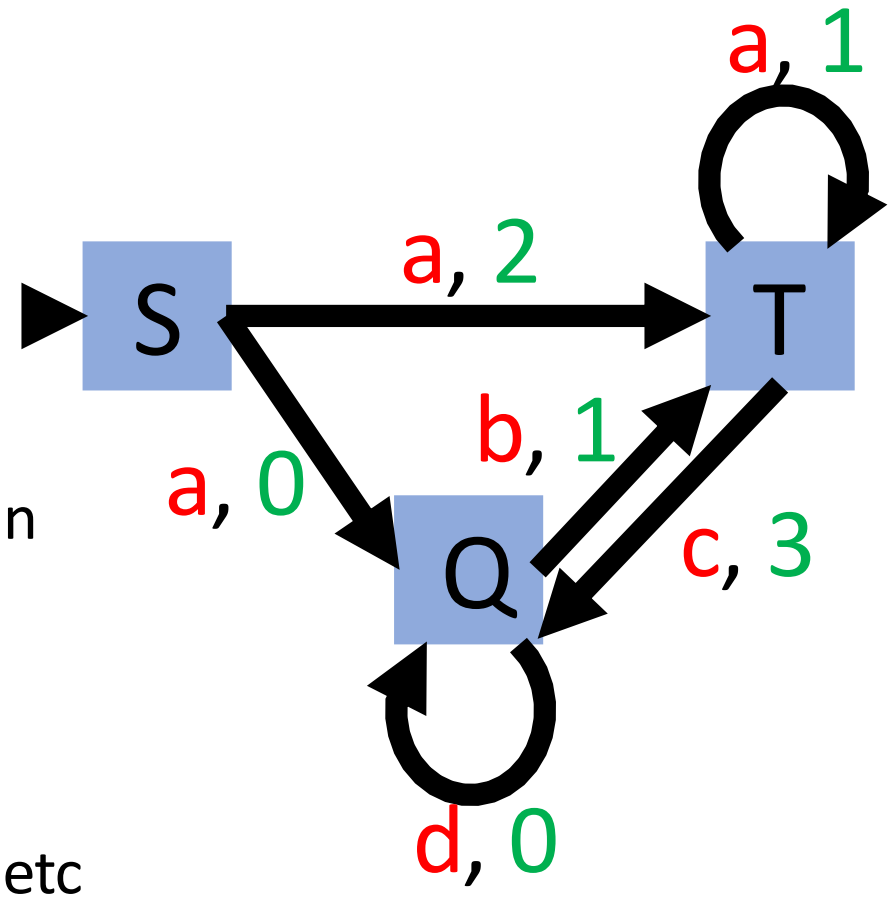
Understand performance of algorithmic approaches for discounted-sum inclusion

Conduct theoretical and empirical evaluation of tools of both approaches

Identify their strengths and weaknesses

Discounted-sum (DS) inclusion

- Discounted-sum (discount-factor $d > 0$)
  - Accumulates diminishing returns
- Discounted-sum automaton
  - Büchi automaton with weights on transitions
  - Weight of word – DS of weight sequence of its run
- DS inclusion between P and Q $(P \subseteq_d Q)$
  - Weight of every word is lower in P than Q
  - Applications in quantitative safety, game-theory, etc

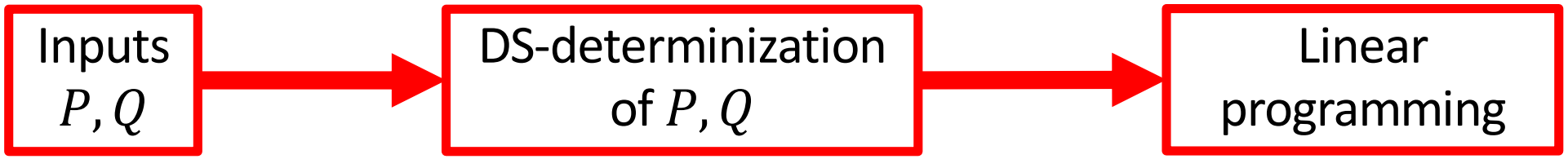


Modular approach

Algorithm

- Structural aspect - DS-determinization [Boker, Henzinger; LICS 2015]
- Quantitative aspect - Linear programming [Chatterjee, Doyen, Henzinger; ACM ToCL 2010]

Complexity - Exponential in time and space



Integrated approach [Bansal, Chaudhuri, Vardi; FoSSaCS 2018]

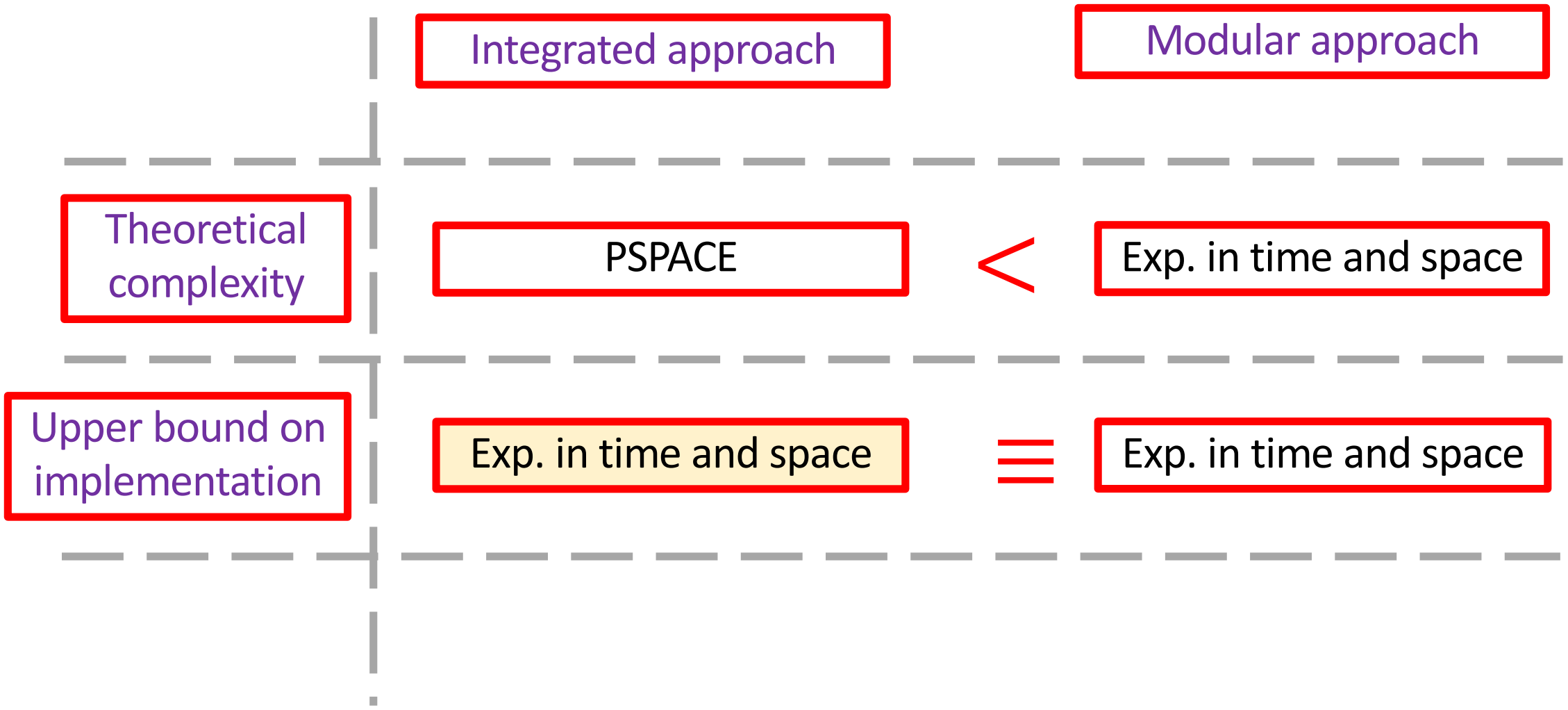
Algorithm: Polynomial-time reduction to Büchi language-inclusion

- *Comparator*-based reduction
  - Büchi automata accepts pair of bounded integer sequences (A, B) iff discounted-sum of A is lower

Complexity - PSPACE

Establishes PSPACE-completeness

Complexity comparison



Tool description

DetLP **(Modular approach)**	**QuIP** **(Integrated approach)**
DS-determinization [Boker, Henzinger; LICS 2015]	Improved comparator $O(n^2)$ to $O(n)$ states
+	+
Linear-programming solver CPLEX	Language-inclusion solver RABIT [Mayr, Clemente; POPL 2013]

Tool description

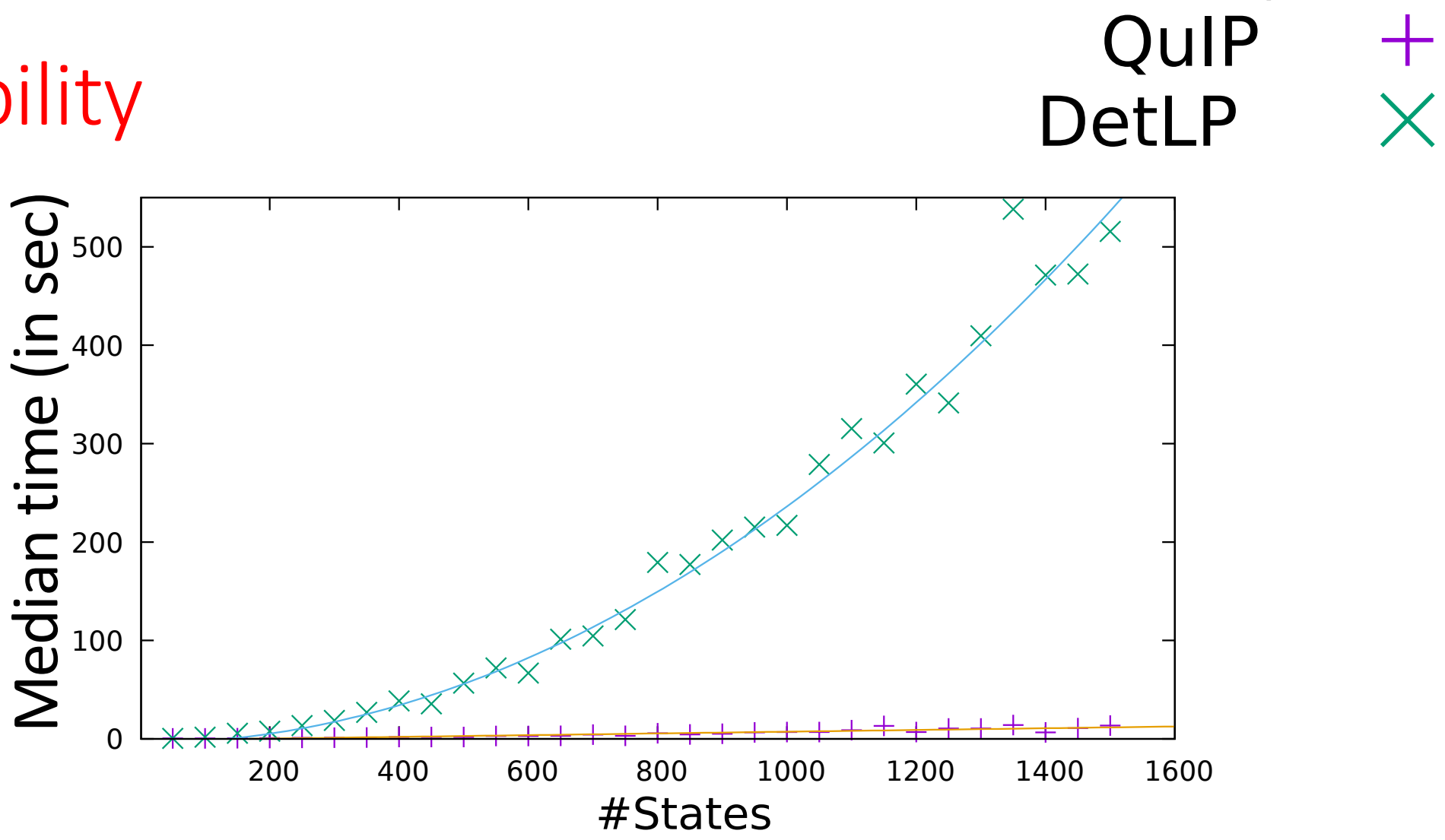
DetLP **(Modular approach)**	**QuIP** **(Integrated approach)**
DS-determinization [Boker, Henzinger; LICS 2015]	Improved comparator $O(n^2)$ to $O(n)$ states
+	+
Linear-programming solver CPLEX	Language-inclusion solver RABIT [Mayr, Clemente; POPL 2013]

Input instances represented with explicit states and transitions

Experimental evaluation

- Randomly generated benchmarks
  - Number of states range in 25-1500
  - Transition-density ranges in 2.5-4
  - Weight on edges ranges from 0-5
- For $P \subseteq_d Q$, fix P, increase Q
- 50 sets of inputs per parameter-tuple
  - Timeout 1000s
- Report median of all runs

Scalability



DetLP scales quadratically, QuIP scales linearly

Scalability inferences

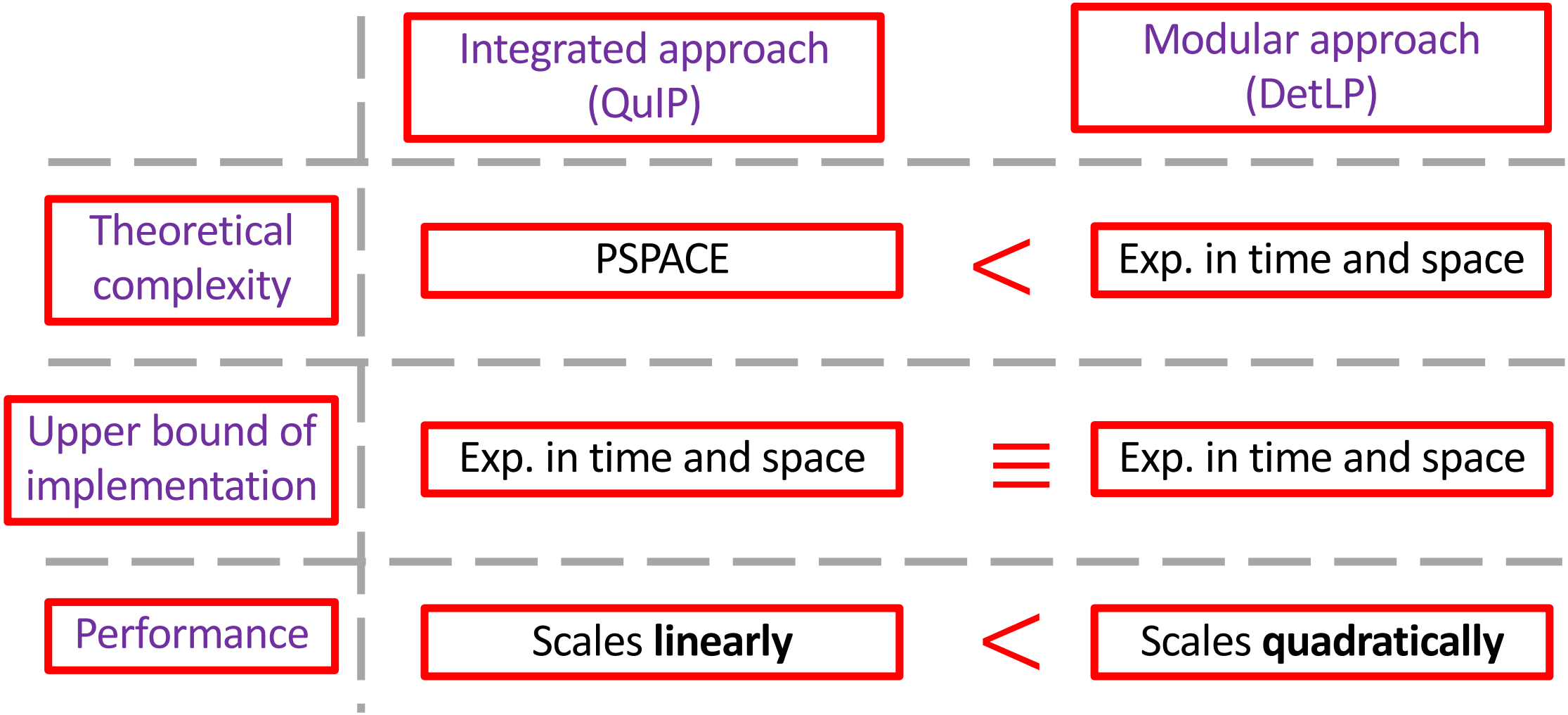
DetLP spends ~95% time in DS-determinization

No early termination of DS-determinization

Full DS-determinization before LP-constraints are set up

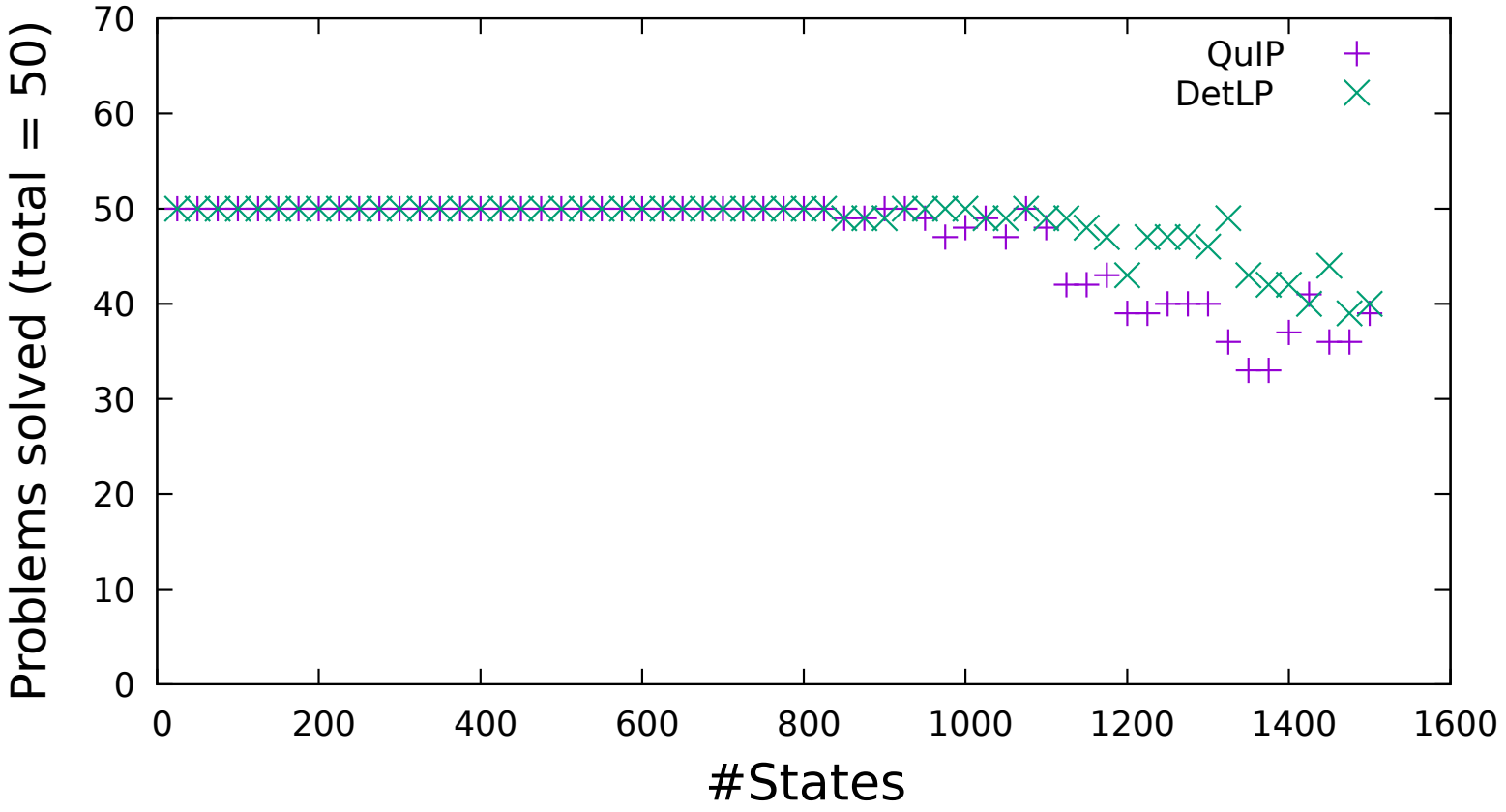
Modular approach suffers from modularity

Complexity comparison



Benchmarks solved

QuIP +
DetLP ×



Implementation of RABIT is not space-efficient

To summarize

- Integrated vs. modular approach – Nuanced
  - Integrated approach scales better
  - Modular approach solves more benchmarks
    - May change with space-efficient implementation of RABIT
- Disparity between theoretical upper bound and performance of tools
  - Better metrics for performance evaluation of algorithms
- Integrated approach with on-the-fly DS-determinization and LP
- Develop non-random benchmark suites by identifying application areas