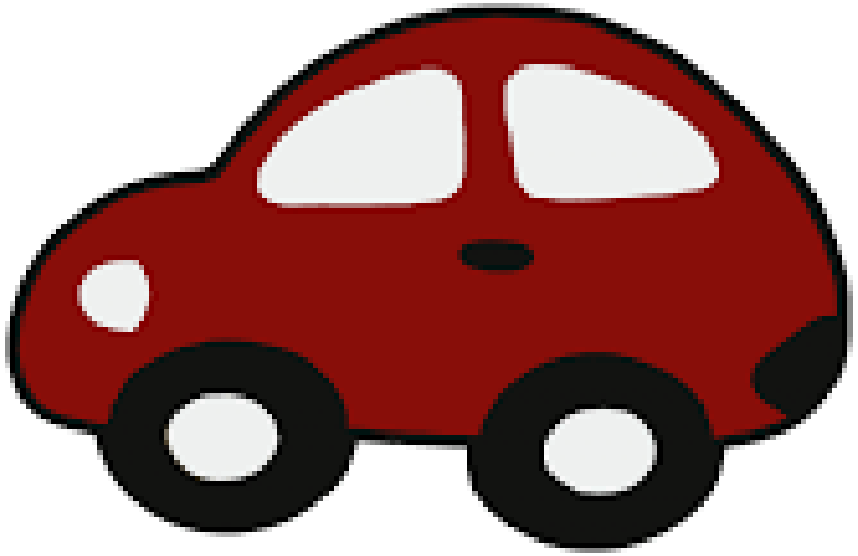
Safety and co-safety comparator automata for discounted sum inclusion

Suguman Bansal, Rice U.
Moshe Y. Vardi, Rice U.

@ CAV 2019, NYC

System efficiency

System efficiency



More efficient?

Compare **cost of** similar runs

Quantitative inclusion

[Chatterjee, Doyen, Henzinger; ACM ToCL 2010]

Formalizes comparison of quantitative systems

Cost computation determines complexity
High complexity – PSPACE-complete, undecidable, still unknown

Quantitative inclusion

=

Structural aspect + Quantitative aspect

States + transitions

Weights

Quantitative inclusion
=
Structural aspect + Quantitative aspect

States + transitions

Weights

Algorithmic approaches

Integrated approach	Hybrid approach
Combines both	Separates both

Quantitative inclusion

=

Structural aspect + Quantitative aspect

States + transitions

Weights

Algorithmic approaches

Integrated approach

Combines both

Hybrid approach

Separates both

Which approach is more viable in practice?

Discounted sum inclusion [**B.**, Chaudhuri, Vardi; CAV 2018]

Quantitative inclusion with discounted sum

Integrated approach	Hybrid approach
Comparator automata-based algorithm	Classical algorithm
[**B.**, Chaudhuri, Vardi; FoSSaCS 2018]	[Boker, Henzinger; LICS 2015]
	[Chatterjee, Doyen, Henzinger; ACM ToCL 2010]

Discounted sum inclusion [**B.**, Chaudhuri, Vardi; CAV 2018]

Quantitative inclusion with discounted sum

Integrated approach	Hybrid approach
Comparator automata-based algorithm	Classical algorithm
[**B.**, Chaudhuri, Vardi; FoSSaCS 2018]	[Boker, Henzinger; LICS 2015]
	[Chatterjee, Doyen, Henzinger; ACM ToCL 2010]
Lower run time	Higher run time
Higher memory consumption	Lower memory consumption

Contributions

Improvement of integrated approaches for DS inclusion

Contributions

Improvement of integrated approaches for DS inclusion

Novel automata-theoretic properties of comparator automata

Contributions

Improvement of integrated approaches for DS inclusion

Novel automata-theoretic properties of comparator automata

Lower complexity algorithm for DS inclusion

Contributions

Improvement of integrated approaches for DS inclusion

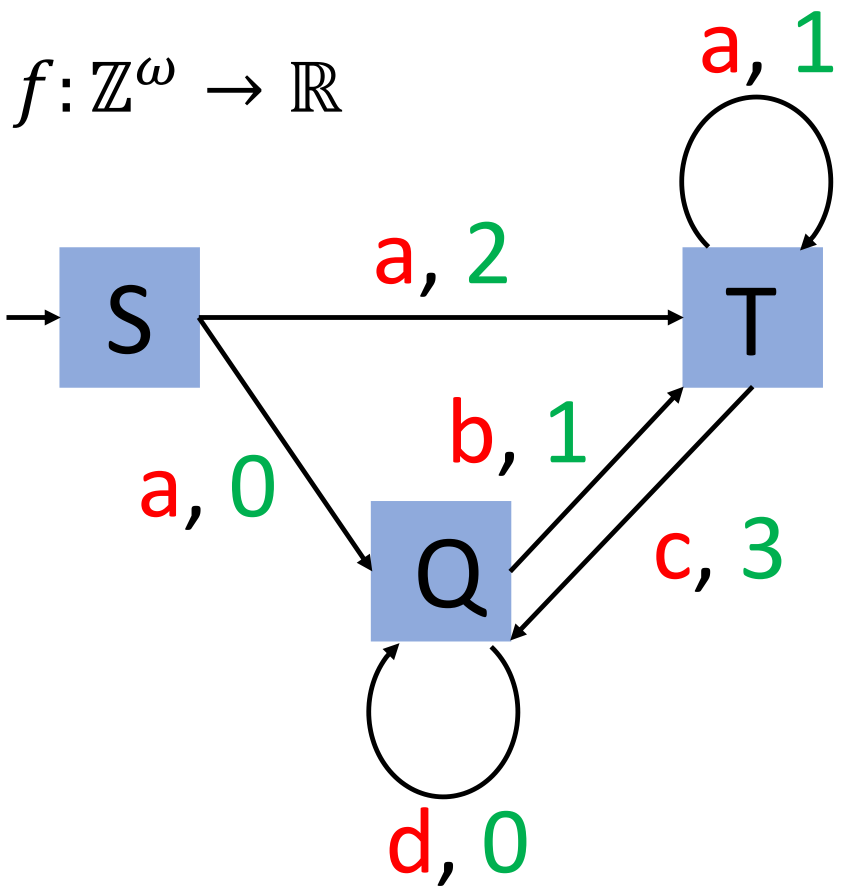
Novel automata-theoretic properties of comparator automata

Lower complexity algorithm for DS inclusion

Empirical analysis: Integrated approach is better in practice

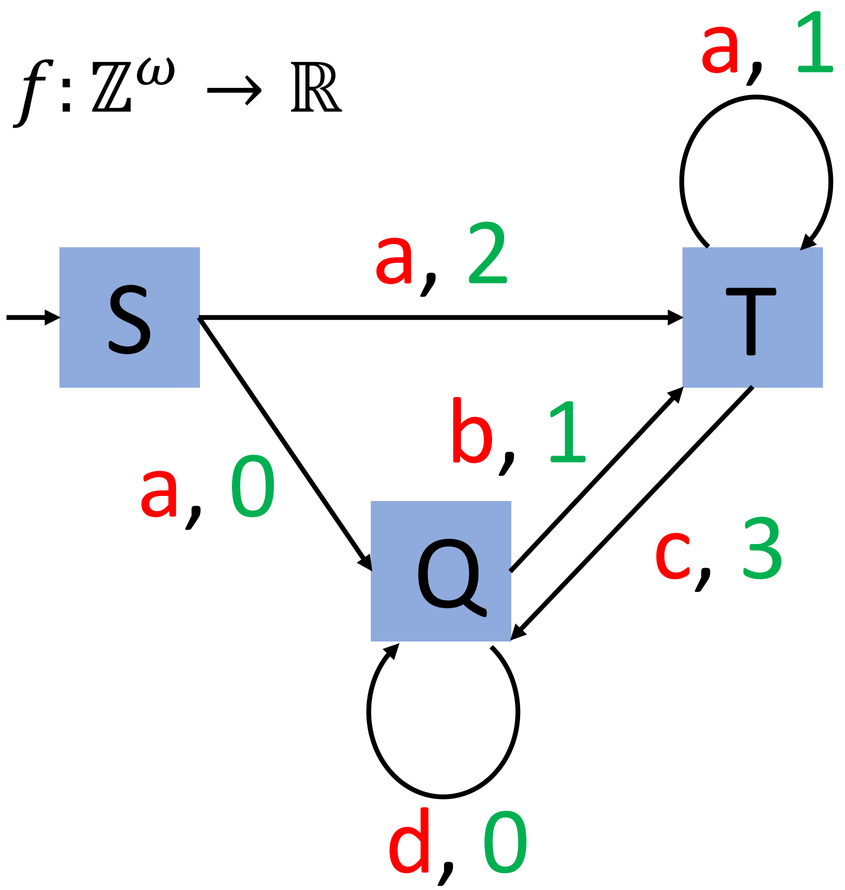
Quantitative inclusion

- Weighted automaton with aggregate function $f : \mathbb{Z}^\omega \to \mathbb{R}$
  - Büchi automaton with weights on transitions
  - Weight of a run: f applied to its weight sequence
  - Weight of word
    - Infimum/supremum of weight sequence of its run



Quantitative inclusion

- Weighted automaton with aggregate function $f: \mathbb{Z}^{\omega} \rightarrow \mathbb{R}$
  - Büchi automaton with weights on transitions
  - Weight of a run: f applied to its weight sequence
  - Weight of word
    - Infimum/supremum of weight sequence of its run

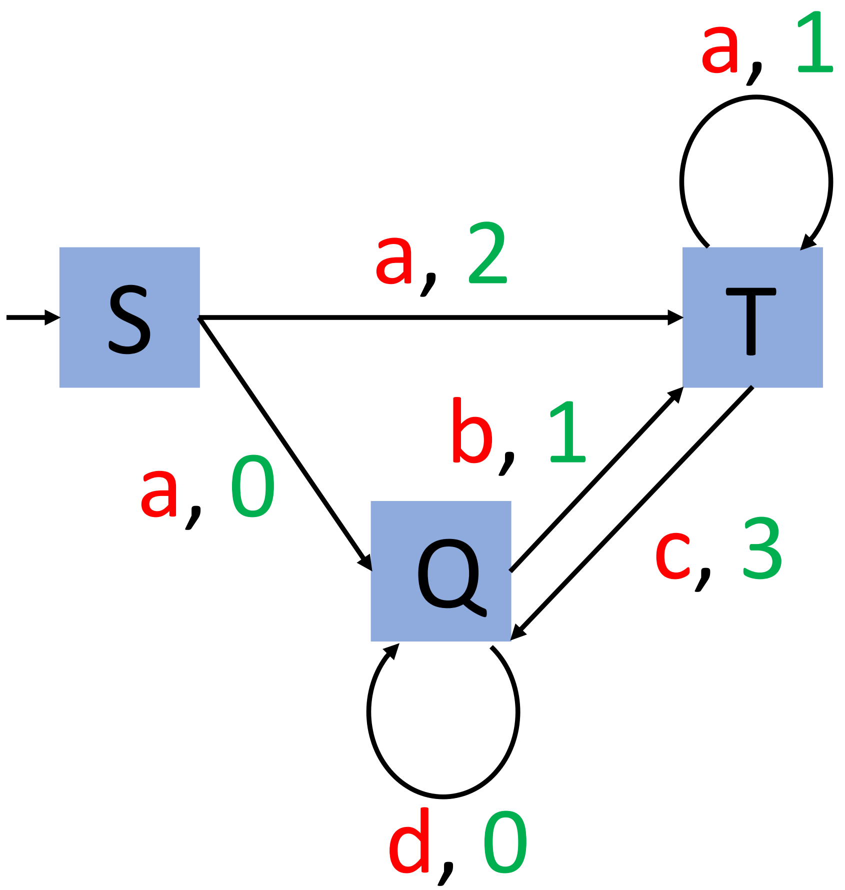


- Quantitative inclusion with f:
  - Given two weighted automata with f, P and Q
  - "Is the weight of every word lower in P than Q?"
  - Application in comparing efficiency of systems

Discounted sum (DS) inclusion

- Discounted sum (discount factor $d > 0$)
  - Accumulates diminishing returns
  - Applications in economics, finance etc
  - Approximates limit-average

- DS inclusion between P and Q ($P \subseteq_d Q$)
  - Quantitative inclusion with discounted-sum

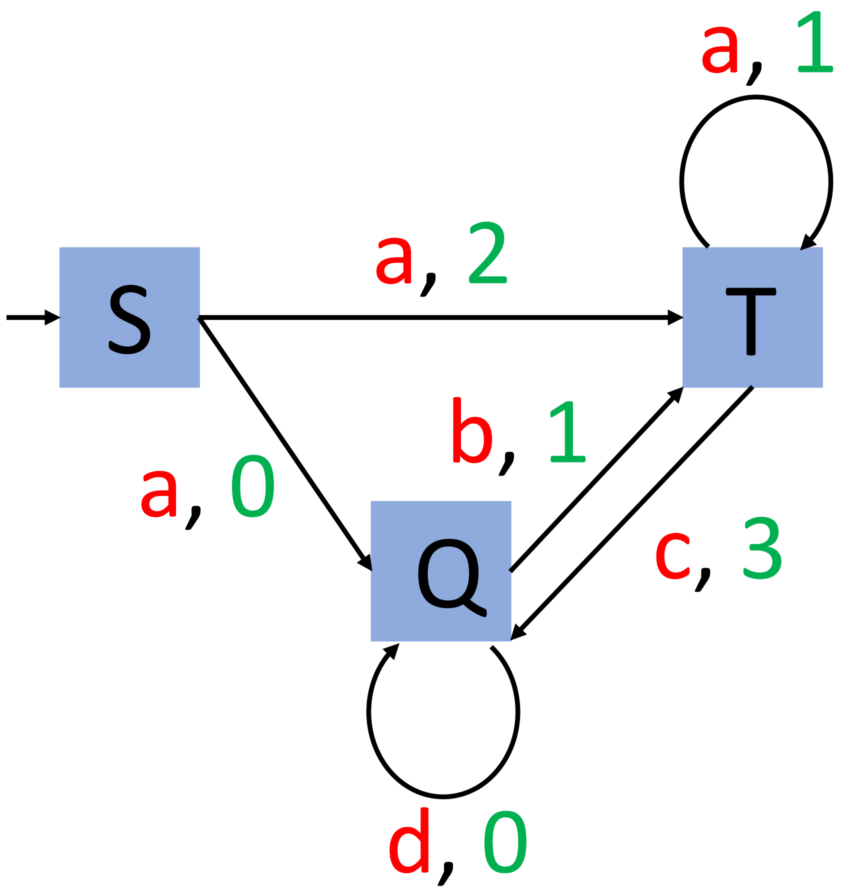


Discounted sum (DS) inclusion

- Discounted sum (discount factor $d > 0$)
  - Accumulates diminishing returns
  - Applications in economics, finance etc
  - Approximates limit-average

- DS inclusion between P and Q ($P \subseteq_d Q$)
  - Quantitative inclusion with discounted-sum

- Known theoretical results
  - Integer discount factor - PSPACE-complete
  - Non-integer discount factor - Decidability unknown



Comparator automata [B., Chaudhuri, Vardi; FoSSaCS 2018]

- For discount factor $d > 1$
- Discounted sum $DS(W, d) = w_0 + \frac{w_1}{d} + \frac{w_2}{d^2} + \cdots$

Comparator automata [B., Chaudhuri, Vardi; FoSSaCS 2018]

- For discount factor $d > 1$
- Discounted sum $DS(W,d) = w_0 + \frac{w_1}{d} + \frac{w_2}{d^2} + \cdots$

- Given:
  - Integer discount factor $d > 1$
  - Upper bound $\mu > 0$
  - Equality or inequality relation $R \in \{\leq, <, \geq, >, \neq, =\}$

- Comparator automata (comparator) for d, μ, R:
  - Non-deterministic Büchi automata
  - Alphabet $\Sigma = \{-\mu, \ldots, \mu\}$
  - Accepts an infinite length, weight sequence $W \in \Sigma^\omega$ iff $DS(W,d)\ R\ 0$ holds

Büchi comparator reduction

DS Inclusion
$P \subseteq_d Q$
Max weight μ

$\rightarrow$

Emptiness checking of an NBA

Büchi comparator

Büchi comparator reduction

DS Inclusion
$P \subseteq_d Q$
Max weight μ

$\rightarrow$

Emptiness checking of an NBA

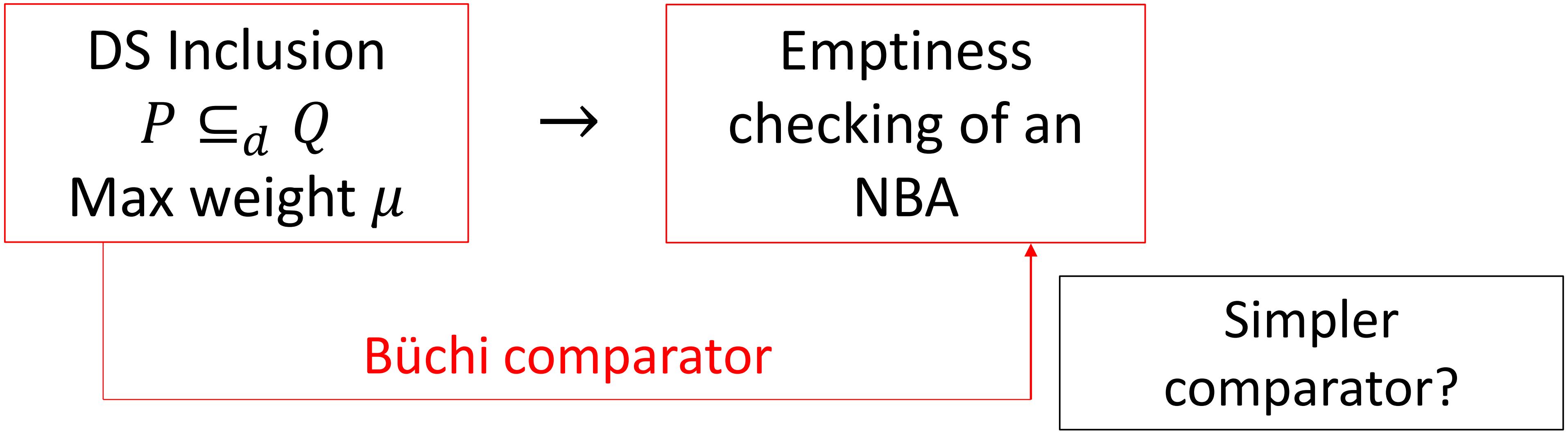
Büchi comparator

$2^{O(n \cdot \log n)}$ blow-up

$n = |P| \cdot |Q| \cdot \mu$

Cause: Büchi complementation

Büchi comparator reduction



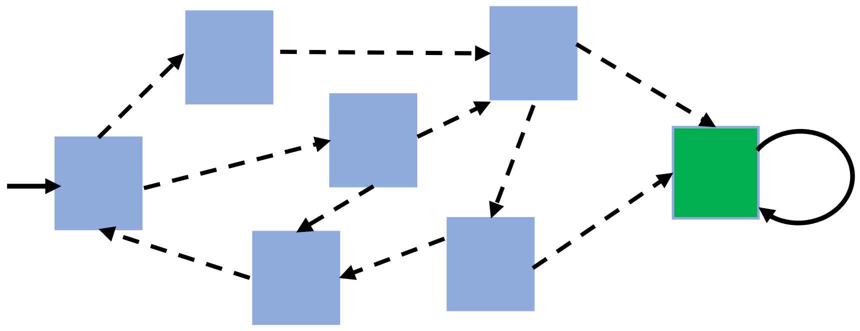
$2^{O(n \cdot \log n)}$ blow-up

$n = |P| \cdot |Q| \cdot \mu$

Cause: Büchi complementation

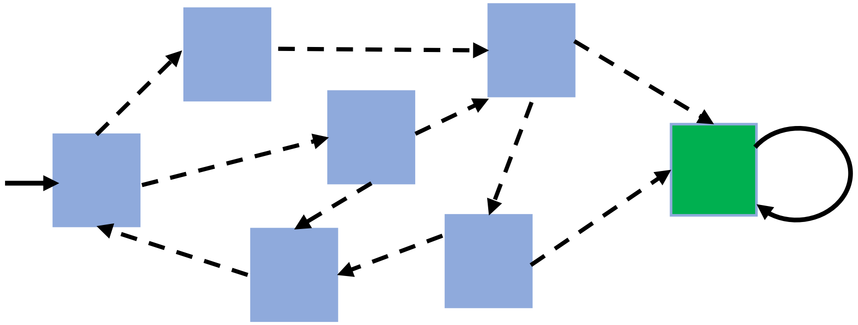
Co-safety automata [Kupferman, Vardi; CAV 1999]

- Deterministic Büchi automaton with one accepting sink



Co-safety automata [Kupferman, Vardi; CAV 1999]

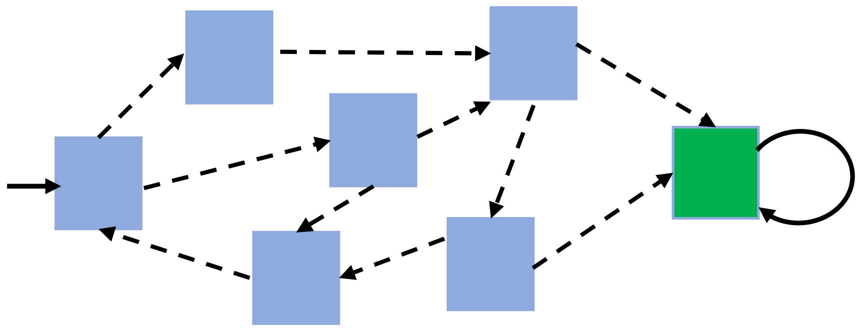
- Deterministic Büchi automaton with one accepting sink



- Language $\mathrm{L} \subseteq \Sigma^{\omega}$ is represented by a co-safety automaton if

Co-safety automata [Kupferman, Vardi; CAV 1999]

- Deterministic Büchi automaton with one accepting sink

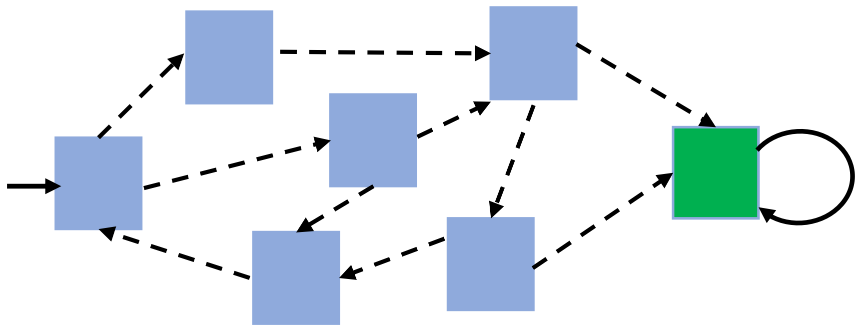


- Language $\mathrm{L} \subseteq \Sigma^{\omega}$ is represented by a co-safety automaton if

Büchi property: L is represented by a Büchi automaton

Co-safety automata [Kupferman, Vardi; CAV 1999]

- Deterministic Büchi automaton with one accepting sink



- Language $\mathrm{L} \subseteq \Sigma^{\omega}$ is represented by a co-safety automaton if

Büchi property: L is represented by a Büchi automaton

Good prefix property:

- Every word w ∈ L has a **good prefix** in L
- Finite word **x** is a **good prefix in L** if every infinite extension is present in L

Result: Comparator for $d, \mu, >$ is co-safety

Büchi prop.

Good prefix prop.

Result: Comparator for $d, \mu, >$ is co-safety

Büchi prop. ✓

Good prefix prop.

Good prefix: Diminishing tail

- $DS(W, d) = w_0 + \frac{w_1}{d} + \frac{w_2}{d^2} + \cdots$

Good prefix: Diminishing tail

- $DS(W, d) = w_0 + \frac{w_1}{d} + \frac{w_2}{d^2} + \cdots$

$$= DS(W[0 \ldots i], d) + \underbrace{\frac{1}{d^i} \cdot DS(W[i \ldots], d)}_{\text{Tail}}$$

Good prefix: Diminishing tail

- $DS(W,d) = w_0 + \frac{w_1}{d} + \frac{w_2}{d^2} + \cdots$

$$= DS(W[0 \ldots i], d) + \underbrace{\frac{1}{d^i} \cdot DS(W[i \ldots], d)}_{\text{Tail}}$$

- W is bounded. So, $DS(W[i \ldots], d)$ is bounded

Good prefix: Diminishing tail

- $DS(W,d) = w_0 + \frac{w_1}{d} + \frac{w_2}{d^2} + \cdots$

$$= DS(W[0 \dots i], d) + \underbrace{\frac{1}{d^i} \cdot DS(W[i \dots], d)}_{\text{Tail}}$$

- W is bounded. So, $DS(W[i \dots], d)$ is bounded
- As $i \to \infty$, Tail $\to 0$

Good prefix: Diminishing tail

- $DS(W,d) = w_0 + \frac{w_1}{d} + \frac{w_2}{d^2} + \cdots$
$$= DS(W[0 \ldots i], d) + \underbrace{\frac{1}{d^i} \cdot DS(W[i \ldots], d)}_{\text{Tail}}$$

- W is bounded. So, $DS(W[i \ldots], d)$ is bounded
- As $i \to \infty$, Tail $\to 0$

- If $DS(W,d) > 0$, then $DS(W[0 \ldots i], d)$ must be large enough

Good prefix: Diminishing tail

- $DS(W, d) = w_0 + \frac{w_1}{d} + \frac{w_2}{d^2} + \cdots$

$$= DS(W[0 \ldots i], d) + \underbrace{\frac{1}{d^i} \cdot DS(W[i \ldots], d)}_{\text{Tail}}$$

- W is bounded. So, $DS(W[i \ldots], d)$ is bounded
- As $i \to \infty$, Tail $\to 0$

- If $DS(W, d) > 0$, then $DS(W[0 \ldots i], d)$ must be large enough

Lemma: $DS(W, d) > 0$ iff W has a good prefix

Result: Comparator for $d, \mu, >$ is co-safety

Büchi prop.

Good prefix prop.

Kupferman, Vardi; CAV 1999

Co-safety automata

Result: Comparator for $d, \mu, >$ is co-safety

Büchi prop. ✓

Good prefix prop. ✓

Kupferman, Vardi; CAV 1999

Co-safety automata Exponential

Result: Comparator for $d, \mu, >$ is co-safety

Büchi prop.

Good prefix prop.

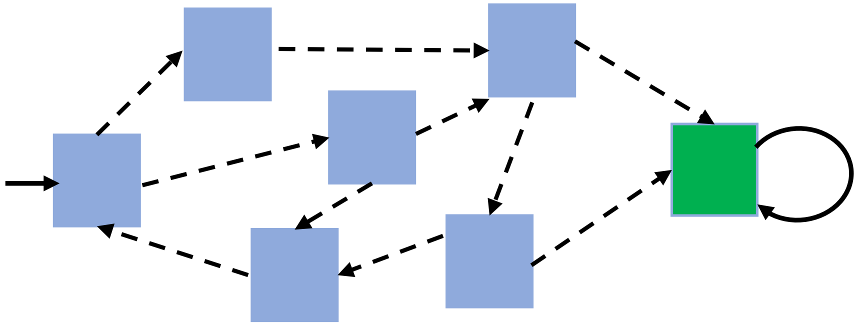
Co-safety automata

states in co-safety = O(# states in Büchi)

Comparators form safety/co-safety automata

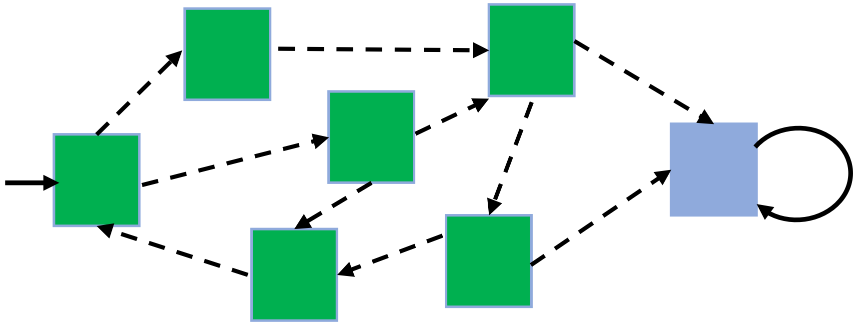
Co-safety automata

- Deterministic Büchi automata with an accepting sink



Safety automata

- Deterministic Büchi automata with a non-accepting sink



Theorem

- If $R \in \{<, >, \neq\}$, comparator for is d, μ, R is a co-safety automaton
- If $R \in \{\leq, \geq, =\}$, comparator for d, μ, R is a safety automaton
- Deterministic constructions have $O(\mu)$ states

Safety/Co-safety comparator reduction

DS Inclusion
$P \subseteq_d Q$
Max weight μ

$\rightarrow$

Emptiness checking of an NBA

Safety/Co-safety comparator

Safety/Co-safety comparator reduction

DS Inclusion
$P \subseteq_d Q$
Max weight μ

$\rightarrow$

Emptiness checking of an NBA

Safety/Co-safety comparator

$2^{O(n)}$ blow-up

$n = |P| \cdot |Q| \cdot \mu$

Cause: Subset construction

Practical performance

Memory consumption

Compare against

Hybrid

(DetLP)

Run-time performance

Compare against

Buchi comparator integrated

(QuIP)

Memory consumption

Prototype tool QuIPFly

- On-the-fly algorithm for emptiness checking
  - Subset construction
  - Deterministic safety/co-safety automata

Memory consumption

Prototype tool QuIPFly

- On-the-fly algorithm for emptiness checking
  - Subset construction
  - Deterministic safety/co-safety automata

- Low memory consumption
  - QuIPFly never constructs the full automata
  - DetLP must always construct the full automata

Memory consumption

Prototype tool QuIPFly

- On-the-fly algorithm for emptiness checking
  - Subset construction
  - Deterministic safety/co-safety automata

- Low memory consumption
  - QuIPFly never constructs the full automata
  - DetLP must always construct the full automata

- Confirmed experimentally as well

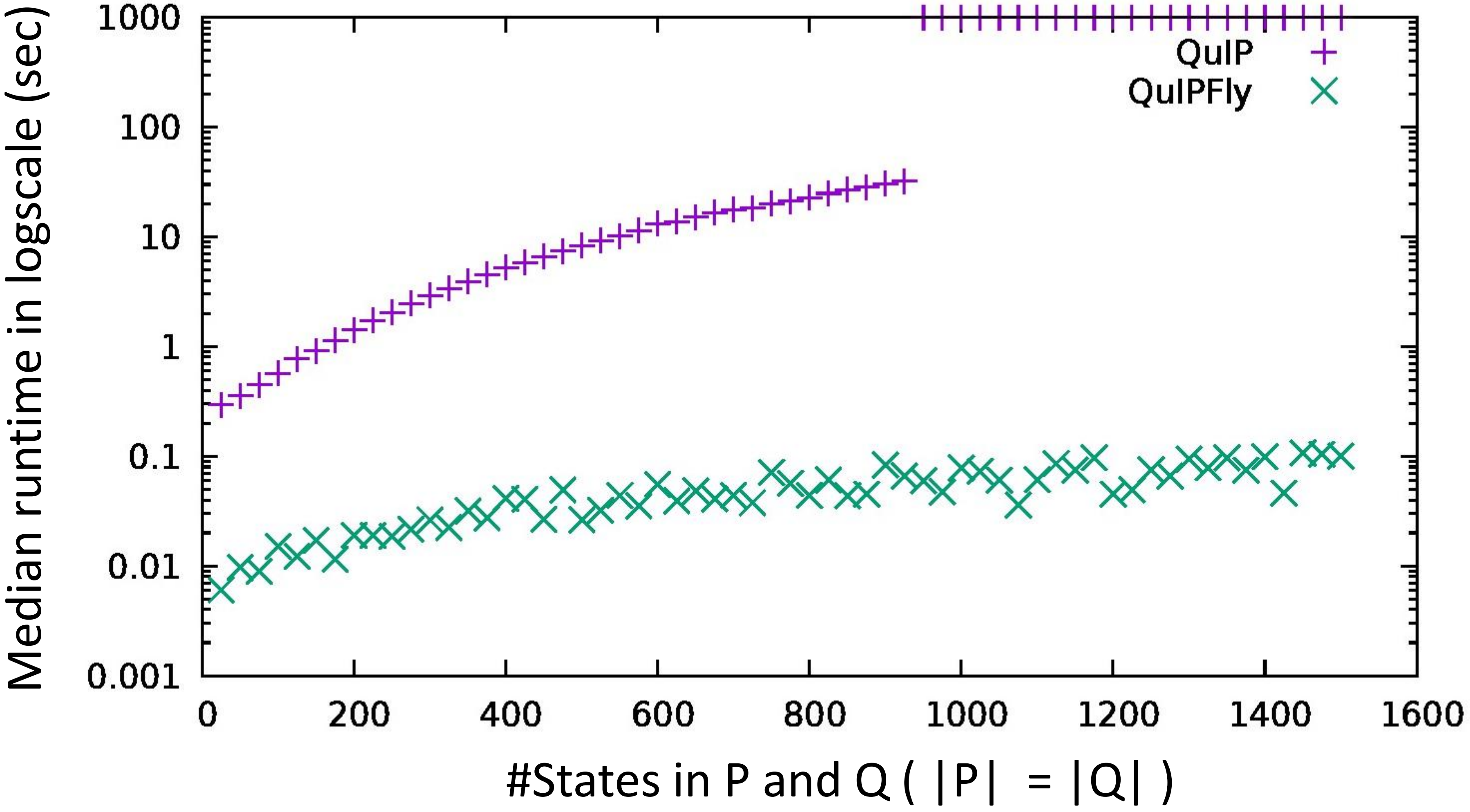
Run-time evaluation

- Compare against QuIP (Büchi comparator)

- Randomly generated benchmarks
  - Parameters: Number of states, transition density, weights on edges
  - 50 pairs of inputs per parameter-tuple

- Report median run time per parameter tuple
  - Median of 50 pairs of inputs

Run-time comparison



Practical performance

Prototype tool QuIPFly

Memory consumption

Compare against

Hybrid

(DetLP)

QuIPFly outperfoms

Run-time performance

Compare against

Buchi comparator integrated

(QuIP)

QuIPFly outperforms

Key takeaways

- Comparator automata are safety/co-safety automata
  - Compact deterministic constructions

Key takeaways

- Comparator automata are safety/co-safety automata
  - Compact deterministic constructions

- Impact on DS inclusion
  - Improved theoretical complexity for DS inclusion
  - Prototype implementation outperforms on run-time and memory usage

Key takeaways

- Comparator automata are safety/co-safety automata
  - Compact deterministic constructions

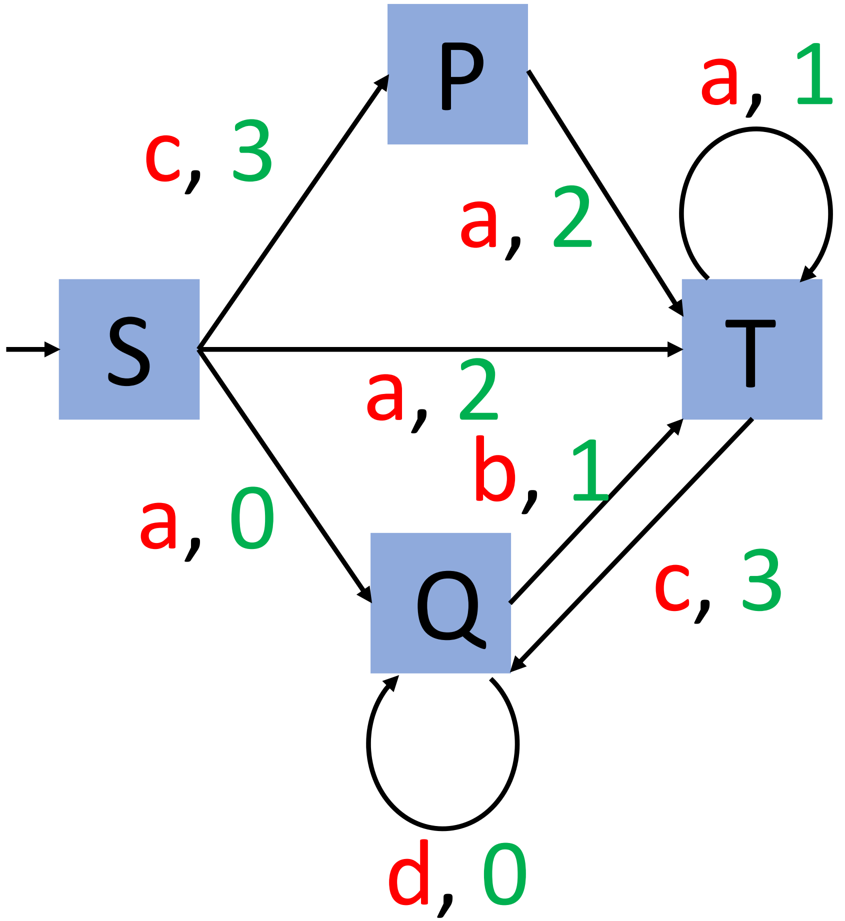
- Impact on DS inclusion
  - Improved theoretical complexity for DS inclusion
  - Prototype implementation outperforms on run-time and memory usage

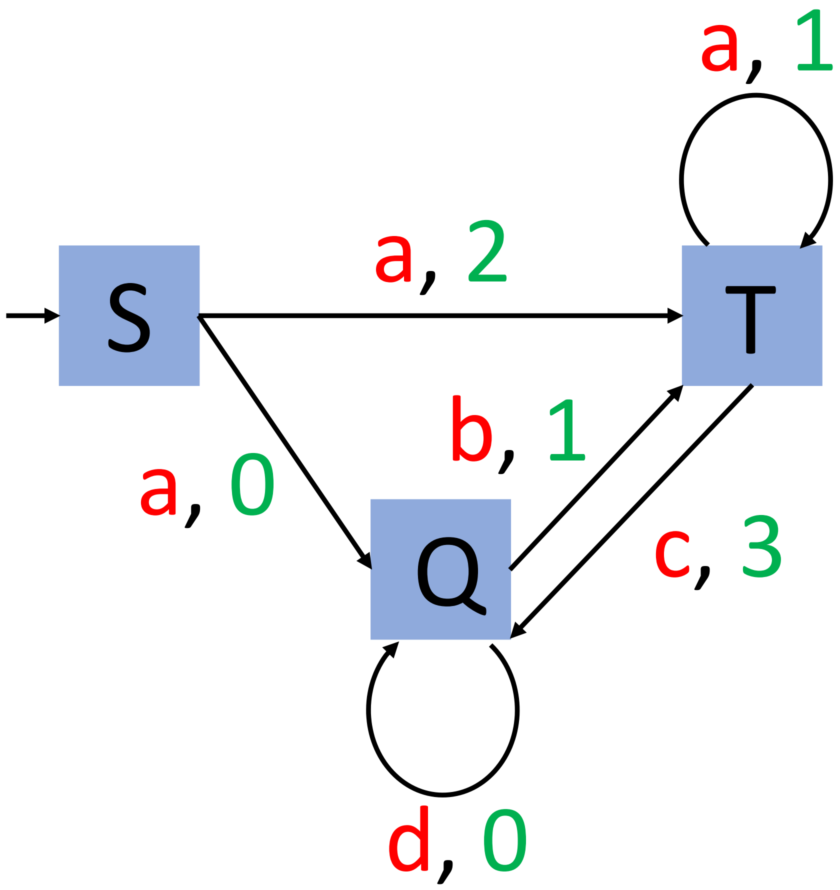
- Integrated approach for more problems in quantitative reasoning
  - Quantitative model checking, Quantitative synthesis, Approximations etc.
  - Non-integer discount factors – pragmatic solutions

Back-up slides

Quantitative inclusion

Formalizes comparison of quantitative systems





Empirical evaluation setup

- Prototype tool QuIPFly
  - Efficient on-the-fly algorithm
- Randomly generated benchmarks
  - Number of states range in 25-1500
  - Transition-density ranges in 3-5
  - Discount factor $d = 3$
  - Weight on edges range from 0 to $d - 1, d^2 + 1, d^3 + 1$
- 50 sets of inputs per parameter-tuple
- Other tools: DetLP (Classical algorithm), QuIP (Büchi comparator)

Gap value

- Finite weight sequence $W = w_0 \ldots w_n$

$$Gap(W, d) = d^n \cdot \left(w_0 + \frac{w_1}{d} + \cdots + \frac{w_n}{d^n}\right) = d^n \cdot DS(W, d)$$

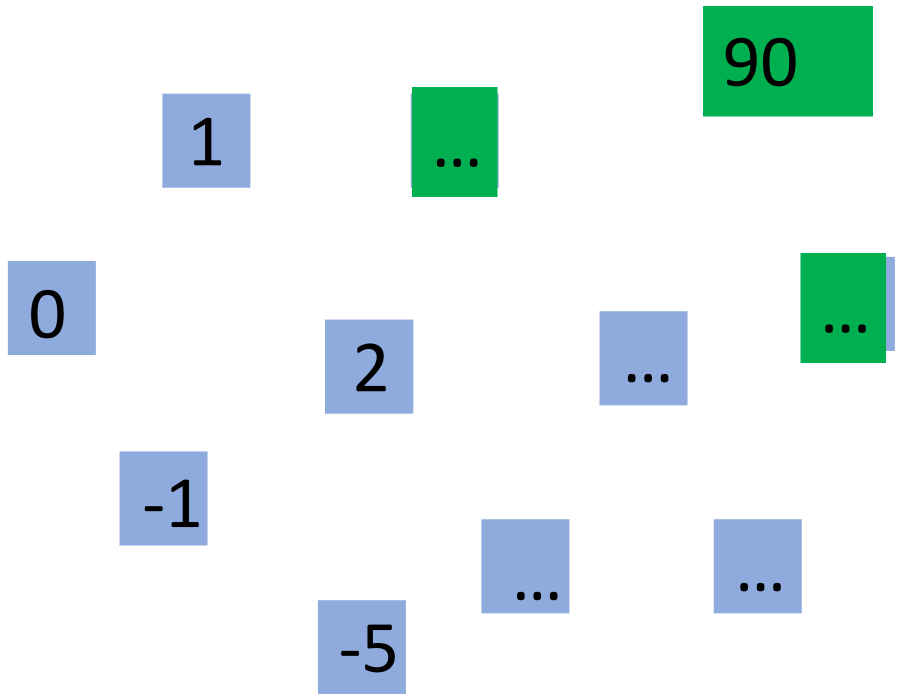
- $DS(W \cdot Y) = \frac{1}{d^n} \cdot (Gap(W, d) + \frac{1}{d} \cdot DS(Y, d))$

Theorem: W is good prefix of $C(d, \mu >)$ iff $Gap(W, d) > \frac{\mu}{d-1}$

Co-safety automaton: States

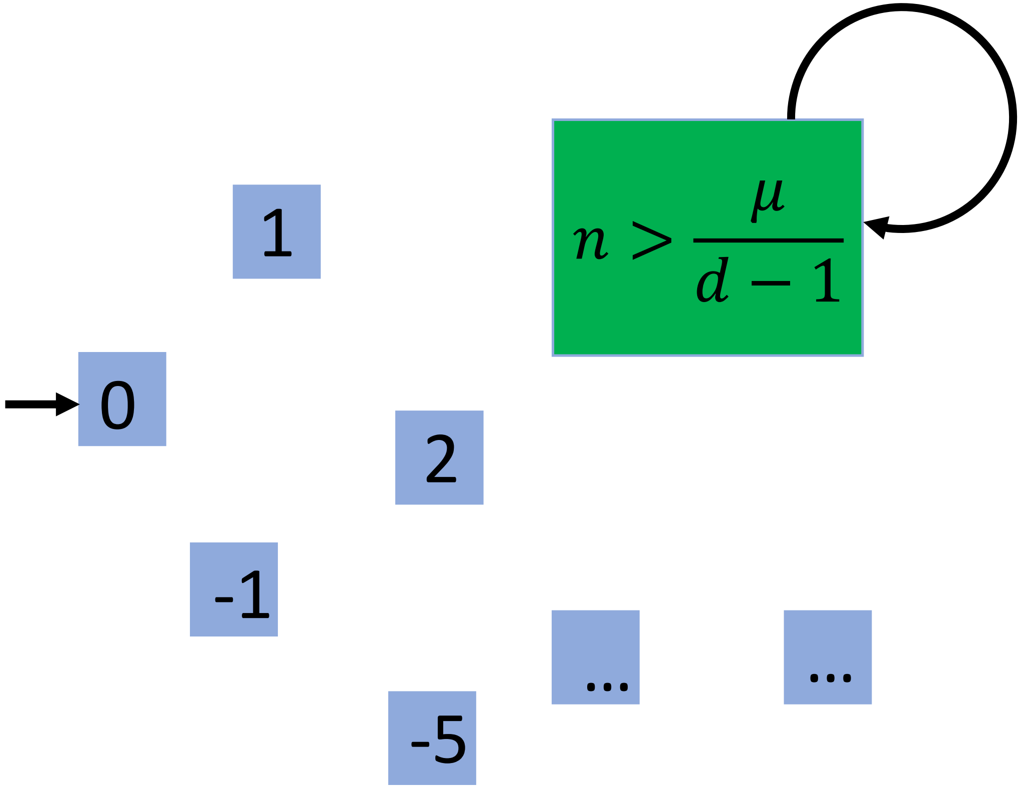
- Gap values represent states
- State is accepting if gap value is greater than threshold

- When d is an integer, gap is also an integer



Co-safety automaton: States

- Gap values represent states
- State is accepting if gap value is greater than threshold
- When d is an integer, gap is also an integer
- Collapse $n > \frac{\mu}{d-1}$ into one state
- Collapse $n < \frac{-\mu}{d-1}$ into one state

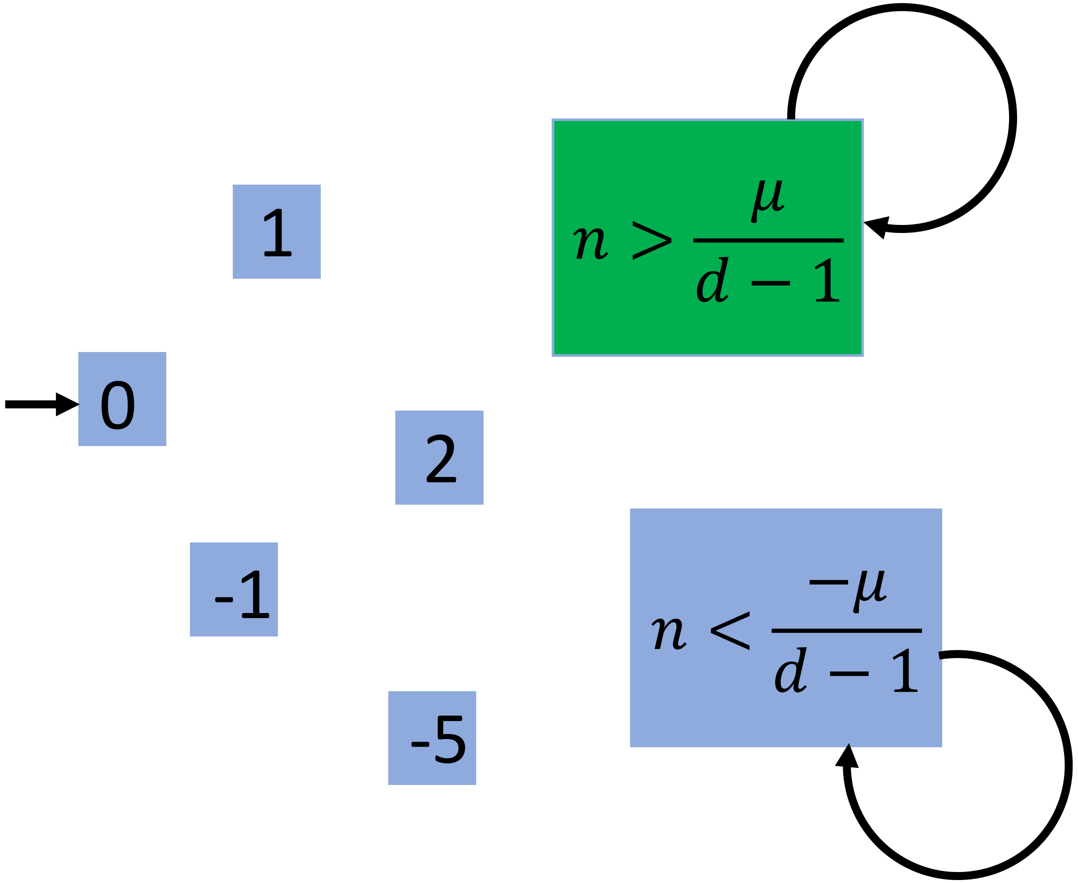


Co-safety automaton: States

- Gap values represent states
- State is accepting if gap value is greater than threshold

- When d is an integer, gap is also an integer

- Collapse $n > \frac{\mu}{d-1}$ into one state
- Collapse $n < \frac{-\mu}{d-1}$ into one state



Co-safety automaton: Transitions

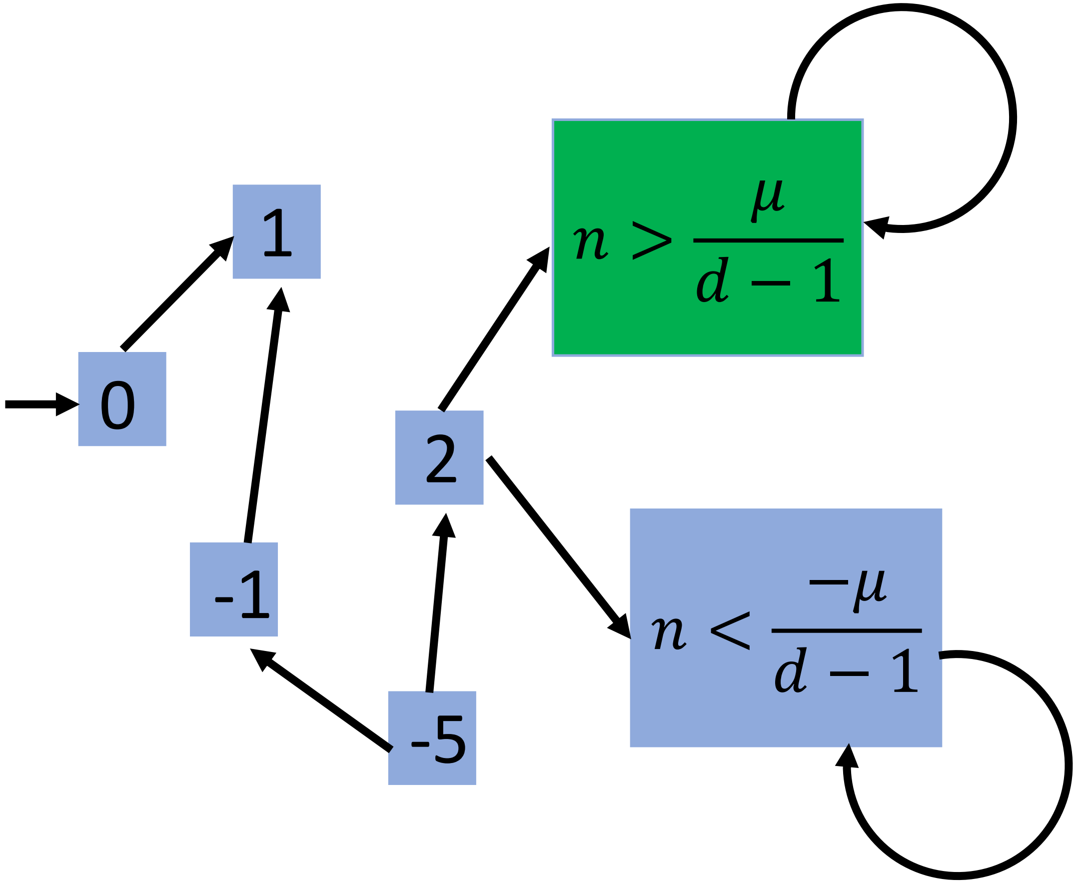
- Inductive definition of gap value

$$Gap(W \cdot w, d) \\ = d \cdot Gap(W, d) + w$$

- Transition from state s to t on alphabet a if

$$t = d \cdot s + a$$

- Deterministic transitions



Co-safety language and automata

Co-safety language

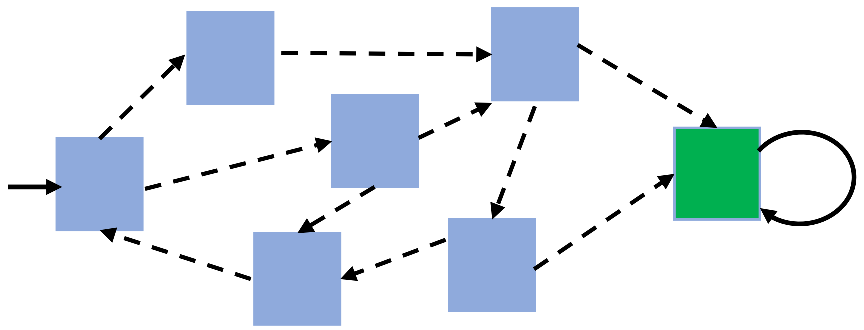
- Let Σ be a finite alphabet, $\mathrm{L} \subseteq \Sigma^\omega$ be a language

- $\mathrm{L} \subseteq \Sigma^\omega$ is a co-safety language if
  - Every word $w \in \mathrm{L}$ has a good prefix in L

Co-safety automata

- Co-safety language represented by Buchi automata

- Büchi automata with an accepting sink



Safety language and automata

Safety language

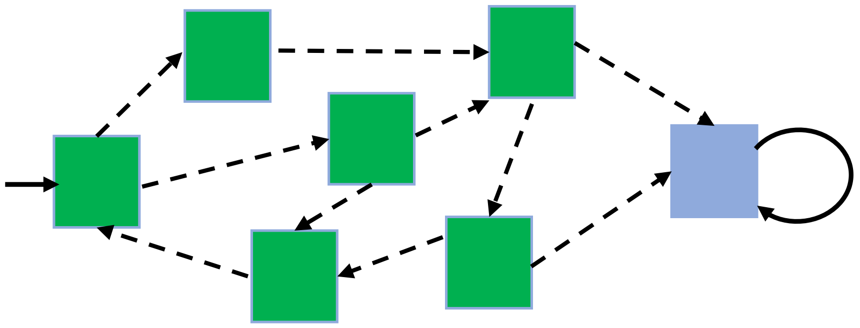
- Complement of a co-safety language

- $\mathrm{L} \subseteq \Sigma^{\omega}$ is a safety language if
  - Every word $w \in \mathrm{L}$ does not have good prefix in L

Safety automata

- Safety language represented by Buchi automata

- Büchi automata with a non-accepting sink



Insight I: Good prefix

- Let $C(d, \mu >)$ denote the language of comparator for $d, \mu, >$
- Claim: $C(d, \mu, >)$ forms a co-safety language
  - Suppose $w \in C(>)$ such that w does not have a good prefix
  - For all $i > 0$, i-length prefix w[i] can be extended by $y[i] \in \Sigma$^ω such that $DS(w[i] \cdot y[i]) \leq 0$
  - As $i \to \infty$,

$$w[i] \cdot y[i] \to w. \quad \text{Therefore,} \quad DS(w[i] \cdot y[i]) \to DS(w)$$

  - Contradiction , since $DS(w[i] \cdot y[i]) \leq 0$, while $DS(w) > 0$.

Theorem: Comparator for is $d, \mu, >$ is a co-safety language

Beyond qualitative reasoning ...

Quantitative reasoning: Reasoning about cost, payoff, efficiency…

Research direction:
Development of tools/techniques for quantitative reasoning by leveraging advancements in qualitative reasoning

System efficiency

Does system implementation conform to efficiency specification?

Algorithmic approaches

Comparator automata-based algorithm for DS inclusion

[**B.**, Chaudhuri, Vardi; FoSSaCS 2018]

Classical algorithm for DS inclusion

[Boker, Henzinger; LICS 2015]

[Chatterjee, Doyen, Henzinger; ACM ToCL 2010]

Lower runtime

Higher memory consumption

Larger runtime

Lower memory consumption

Comparator for $d, \mu, >$ is co-safety

Büchi

Good prefix

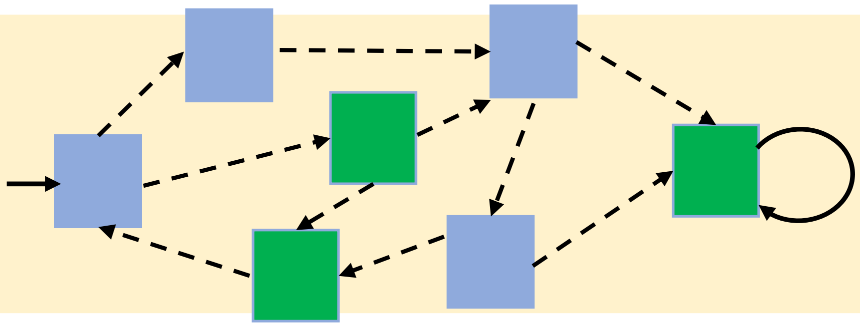
Deterministic

Existence

Construction

Comparator for $d, \mu, >$ is co-safety

Büchi



Good prefix

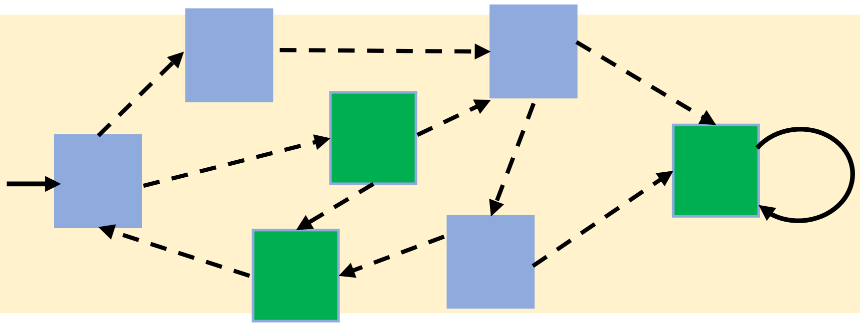
Deterministic

Existence

Construction

Comparator for $d, \mu, >$ is co-safety

Büchi



Good prefix ✓

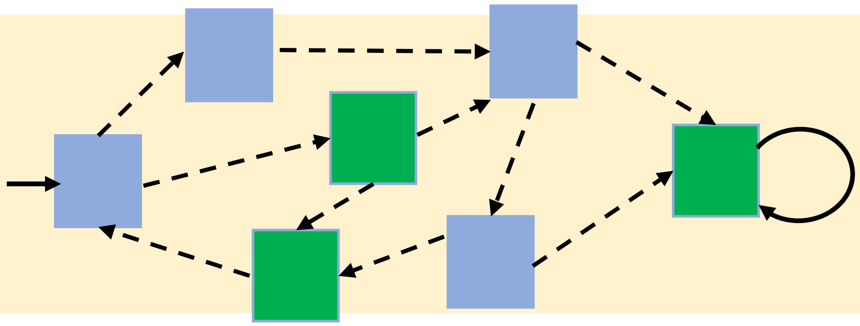
Deterministic

Existence

Construction

Comparator for $d, \mu, >$ is co-safety

Büchi



Good prefix

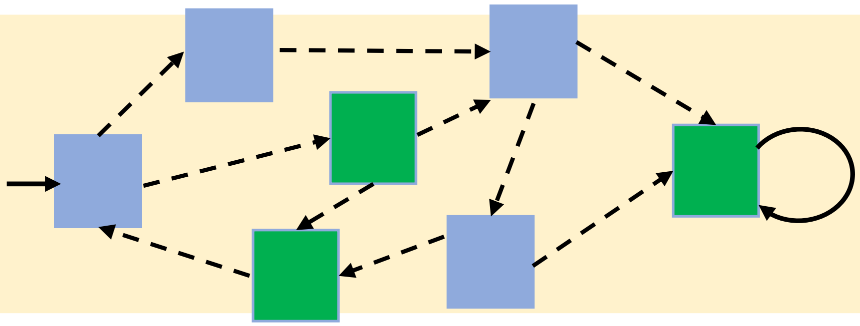
Deterministic Exponential

Existence

Construction

Comparator for $d, \mu, >$ is co-safety

Büchi



Good prefix

Deterministic $O(\frac{\mu}{d-1})$

Existence

Construction

Bottleneck in comparator-based DS inclusion

DS Inclusion
$P \subseteq_d Q$

Büchi comparator C

Language inclusion between NBA

Büchi complement. $2^{O(n \cdot log\ n)}$ blow-up

Is an NBA empty?
$2^{O(P \cdot Q \cdot C \cdot \log(P \cdot Q \cdot C))}$

Simpler comparator

Language inclusion between simpler automata

Is an NBA empty?
Smaller automata

Theoretical improvement

DS Inclusion
$P \subseteq_d Q$

Büchi comparator C

Language inclusion between Buchi automata

Büchi complement
$2^{O(n \cdot log\, n)}$ blow-up

Is an NBA empty?
$2^{O(P \cdot Q \cdot C \cdot \log(P \cdot Q \cdot C))}$

Deterministic safety/co-safety comparator C

Language inclusion between safety/co-safety automata

Subset construction
$2^{O(n)}$ blow-up

Is an NBA empty?
$2^{O(P \cdot Q \cdot C)}$

Good prefix: Diminishing tail

- $DS(W, d) = w_0 + \frac{w_1}{d} + \frac{w_2}{d^2} + \cdots$

$$= \quad DS(W[0 \ldots i], d) + \underbrace{\frac{1}{d^i} \cdot DS(W[i \ldots], d)}_{\text{Tail}}$$

- W is bounded. So, $DS(W[i \ldots], d)$ is bounded
- As $i \to \infty$, Tail $\to 0$

- If $DS(W, d) > 0$, then $DS(W[0 \ldots i], d)$ is large enough

Lemma: If $DS(W, d) > 0$, then W must have a good prefix