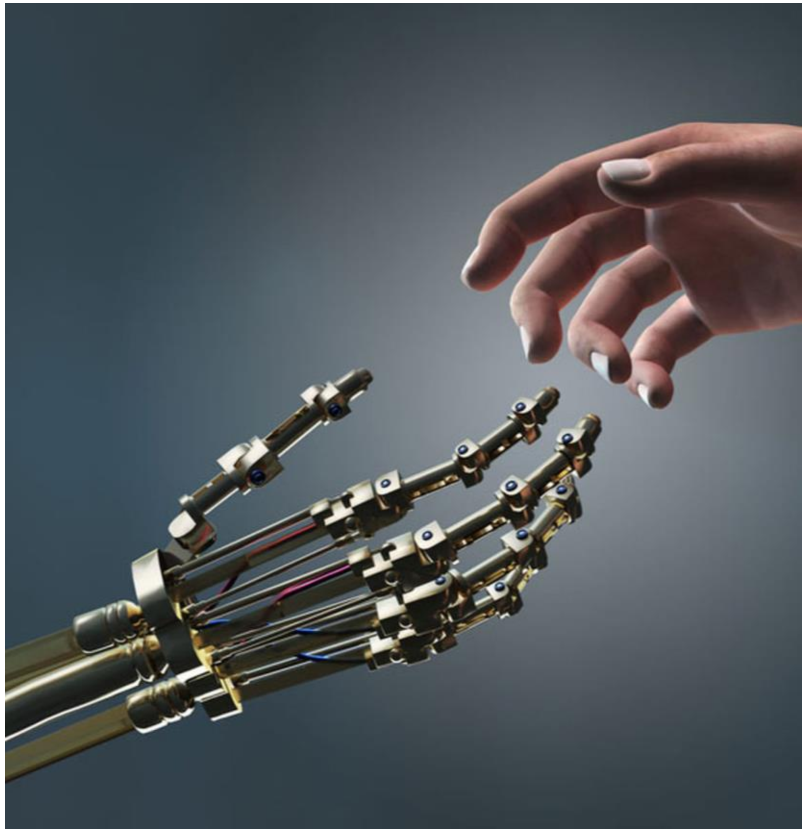
Towards designing intelligent machines via reactive synthesis

Suguman Bansal

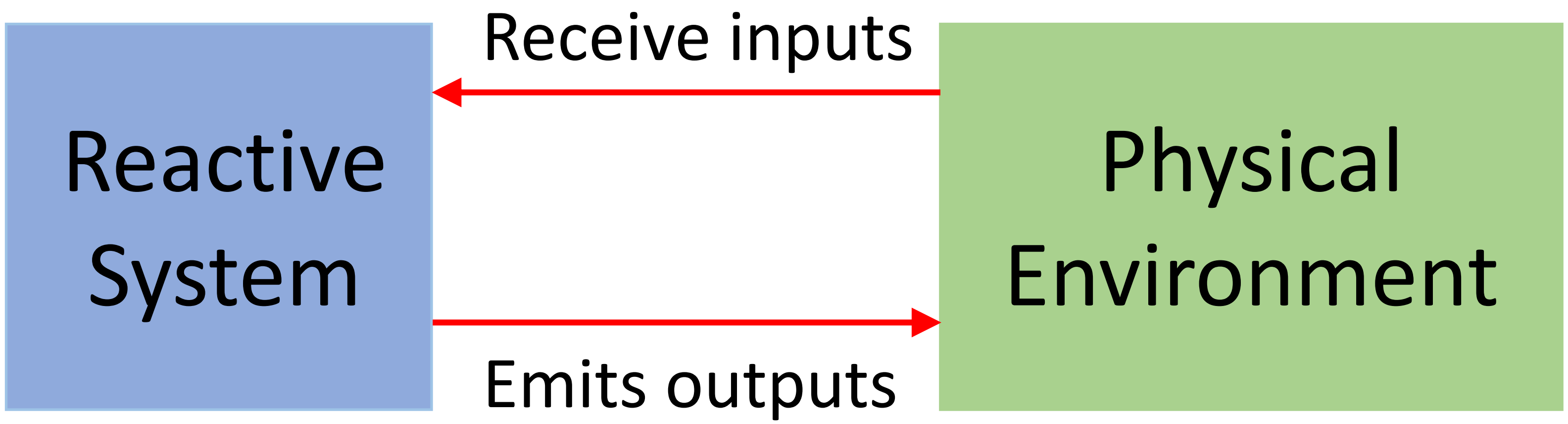
University of Pennsylvania

suguman@seas.upenn.edu

Intelligent machines
Interact with the environment

Reactive systems



Continuous cycle of interaction

“Reactive” systems today

Robot and human interactions

Autonomous vehicles

AARIAN MARSHALL AND ALEX DAVIES TRANSPORTATION 05.24.18 03:38 PM

UBER'S SELF-DRIVING CAR SAW THE WOMAN IT KILLED, REPORT SAYS

…The problem is that it’s hard to find images of every sort of situation that could happen in the wild. Can the system distinguish a tumbleweed from a toddler. …

Designing correct reactive systems is hard

Specifying intent of a reactive program is easier

Can we automatically generate a reactive program from its specification?

Reactive synthesis

Towards designing intelligent machines via reactive synthesis

Richness Specifications

Formal Guarantees

Scalability

On today’s agenda

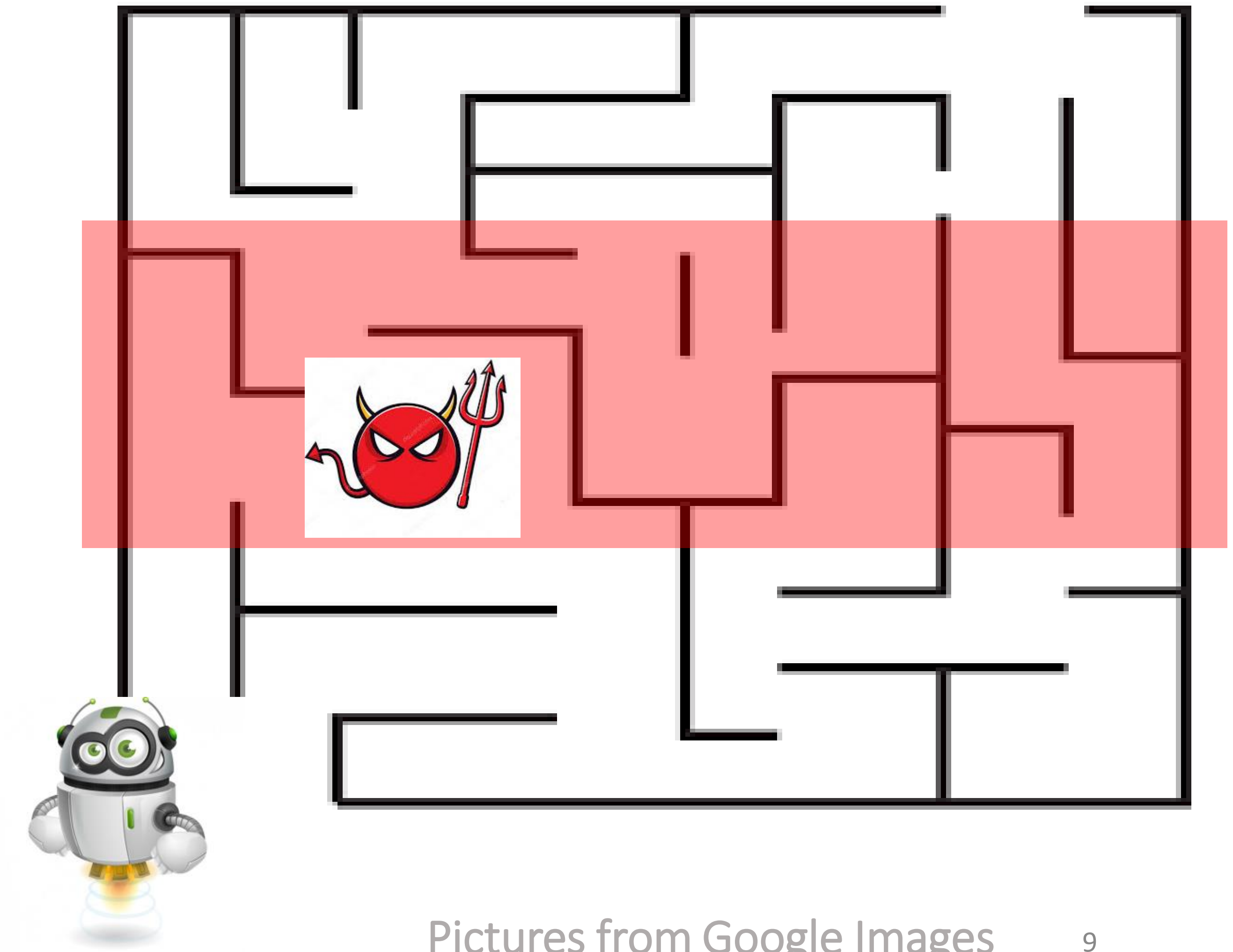
- Reactive synthesis for planning
    - Qualitative and Quantitative specifications
- Automata-based quantitative reasoning
- Generality of approach in Formal Quantitative Reasoning
    - Beyond reactive synthesis

Let’s help the robot plan

Robot in an uncontrollably changing environment

Robot must satisfy a given specification

Reactive synthesis to the rescue!



Pictures from Google Images

Synthesis from **rich** specifications

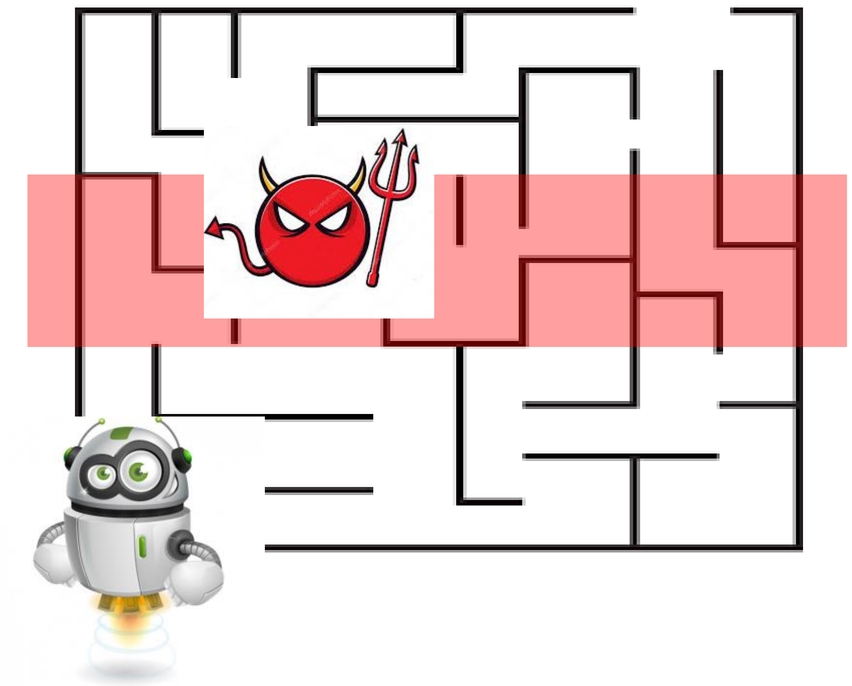
Specification = Quantitative + Qualitative

Qualitative: Temporal Goals

Given an **LTL formula**, every execution should satisfy the formula

Quantitative: Satisficing Goals

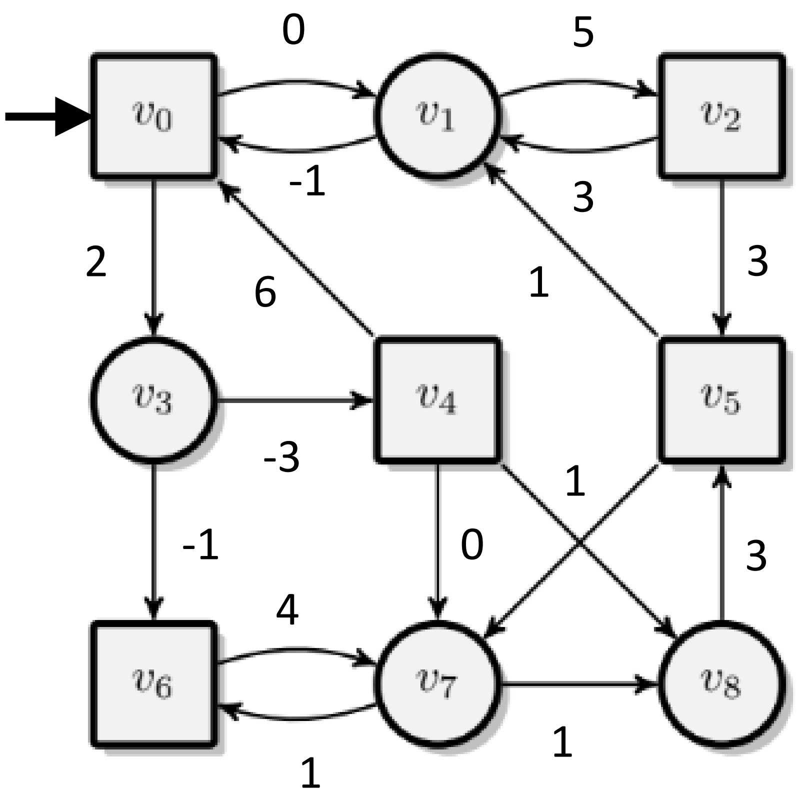
Given a **threshold value**, cost of every execution should exceed the threshold



Pictures from Google Images

Quantitative Game

- Two-player graph game with weights on edges
  - Plays begin in initial state; From each state, its player choses the next state



- Cost of a play (Discounted-sum):
  - For weight sequence A and discount factor $d > 1$,

$$DS(A, d) = A[0] + \frac{A[1]}{d} + \frac{A[2]}{d^2} + \cdots$$

- Adversarial players
  - System player: Maximizes cost of plays
  - Environment player: Minimizes cost of plays

Synthesis from Temporal and Satisficing Goals

Strategy: Decides the next state based on the history of a play

Problem: Generate a strategy for the system player that

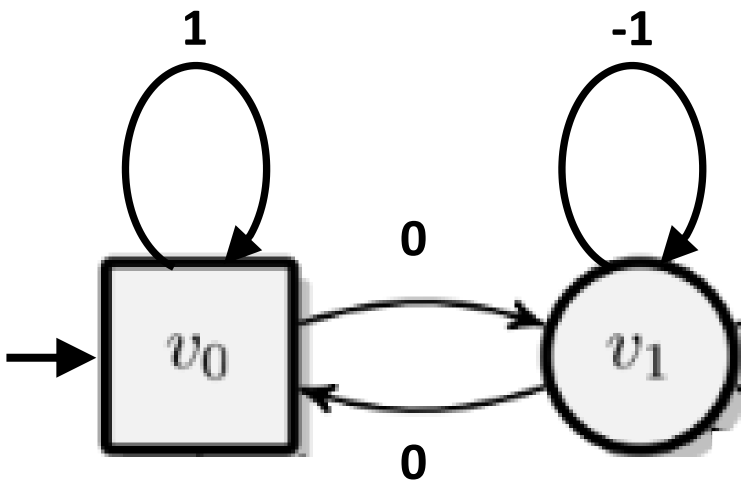
(a). satisfies a given LTL formula on all plays

(b). ensures the cost of all plays exceeds a given threshold value.

Example

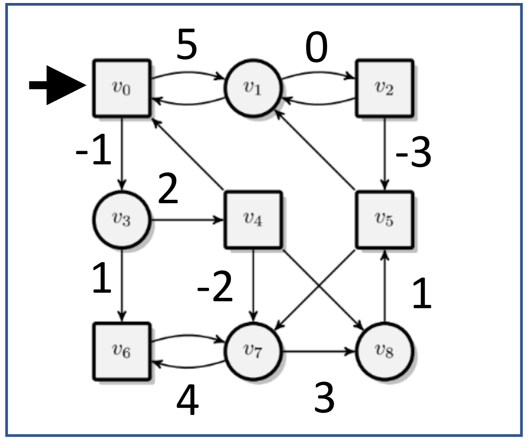
LTL Goal: Visit state v_1

Satisficing Goal: Ensure cost exceeds 0.5



Existing Solution Approaches

[Chatterjee et. al. 2017; Wen, Ehlers, Topcu, 2015; Kwiatkowska, Parker, Wiltsche; 2017]



Quantitative Game

$d > 1$

Temporal Goal → Automata-based methods

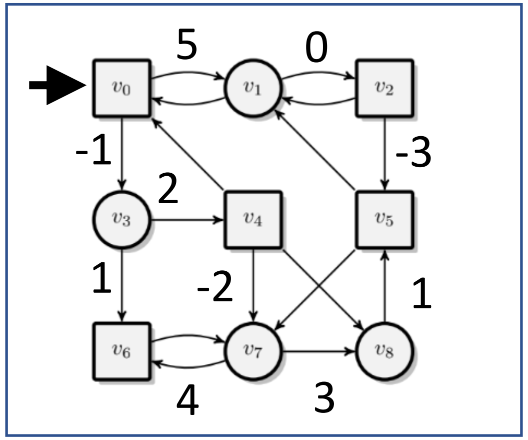
Satisficing Goal → Optimization-based methods

No sound algorithm so far!

Disparate methods do not combine well

Existing Solution Approaches

[Chatterjee et. al. 2017; Wen, Ehlers, Topcu, 2015; Kwiatkowska, Parker, Wiltsche; 2017]



Quantitative Game

$d > 1$

LTL Formula

Automata-based methods

Threshold Value

Optimization-based methods

No sound algorithm so far!

Disparate methods do not combine well

Existing Solution Approaches

[Chatterjee et. al. 2017; Wen, Ehlers, Topcu, 2015; Kwiatkowska, Parker, Wiltsche; 2017]

Automata-

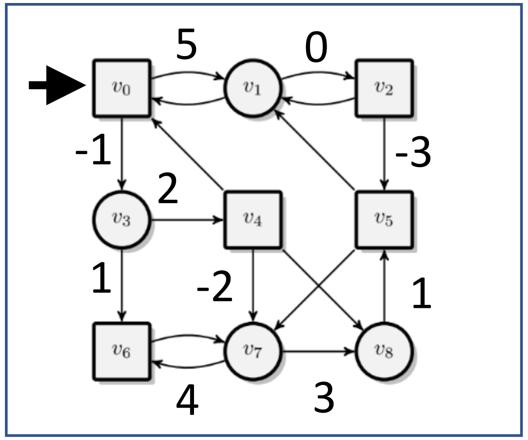
Devise integrated methods that combine both steps

$d > 1$

No sound algorithm so far!

Disparate methods do not combine well

Our Approach: Devising Integrated Solutions



Quantitative Game

$d > 1$

Temporal Goal → Automata-based methods

Satisficing Goal →

Automata-based Methods for Satisficing Goals?

On today's agenda

✓ Reactive synthesis for planning
  ✓ Qualitative and Quantitative specifications
  ✓ Existing algorithms fail to offer formal guarantees due to disparate-methods

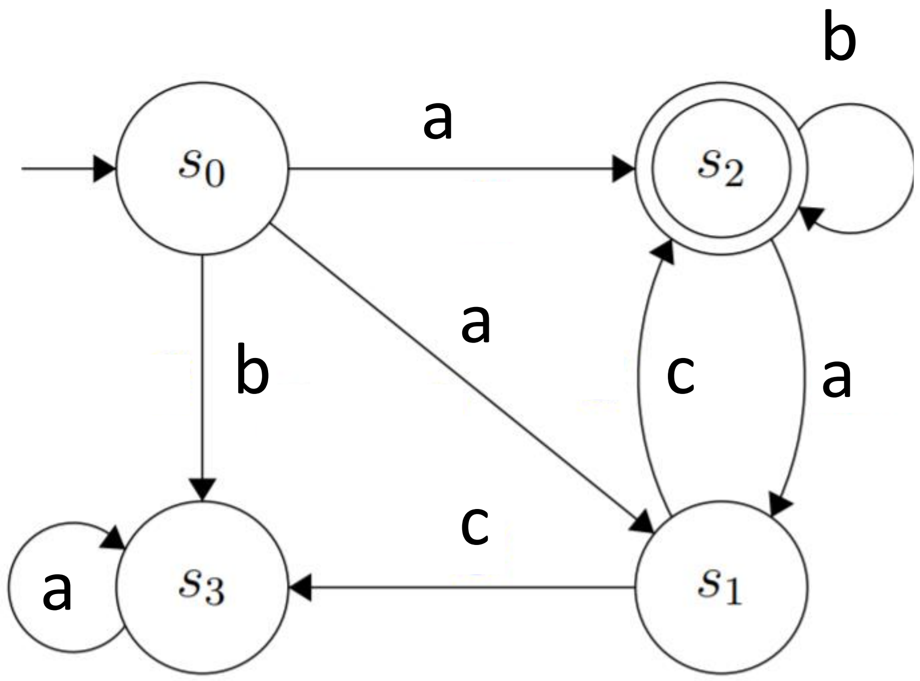
- Automata-based quantitative reasoning
  - Towards algorithms with formal guarantees and scalable performance

- Generality of approach in Formal Quantitative Reasoning
  - Beyond reactive synthesis

Non-deterministic Büchi Automata (NBA)

Automata over infinite-length words

- Finite states and finite alphabet
- Accepting states
- Transitions between states on alphabet
  - Deterministic BA: Exactly one transition from every state on every alphabet



- Run is accepting if it visits accepting states infinitely often
- Word is accepting if it has an accepting run

Comparison is the fundamental operation

Satisficing Goal: Given, threshold value v

System-player wins

If cost of plays exceeds v

$\equiv$ $DS(A, d) > v$, where A is weight-sequence of a play

Given, weight sequences A and B, is $DS(A, d) > DS(B, d)$?

How to perform comparison using automata?

Comparator

[Bansal, Chaudhuri, and Vardi. FoSSaCS 2018; Bansal et. al. CAV 2018; Bansal and Vardi, CAV 2019]

Weight sequences are infinite-length words

- Finite alphabet $\Sigma = \{-\mu, \ldots, \mu\}$ for integer $\mu > 0$

Given, discount factor $d > 1$

equality or inequality relation $R \in \{\leq, <, \geq, >, \neq, =\}$

integer $\mu > 0$

Let, $A, B \in \Sigma^{\omega}$ be two weight sequences

Comparator accepts (A, B) iff $DS(A, d)\ \boldsymbol{R}\ DS(B, d)$

Is Comparator an Automata?

With integer discount factors $d > 1$

- $DS(A,d) = a_0 + \frac{a_1}{d} + \frac{a_2}{d^2} + \cdots = \quad (a_0.a_1a_2\ldots)_d = A_d$
- $DS(A,d) \leq \; DS(B,d)$ iff $A_d \leq B_d$
- Is there C s.t. $DS(C,d) = B_d - A_d \geq 0$?

Caveat: $a_i \geq d$ and -ve

Caveat: Difference of infinite sequences?

Core Insight

- Consider ($d = 10$)

Most significant ← Least significant

X				
A		5	13	6
C	+	0	8	6
B				

Core Insight

- Consider ($d = 10$)

Most significant ← Least significant

X			1	0
A		5	13	6
C	+	0	8	6
B				2

$$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{i-1}$$

Core Insight

- Consider ($d = 10$)

Most significant $\leftarrow$ Least significant

X		2	1	0
A		5	13	6
C	+	0	8	6
B			2	2

$$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{i-1}$$

Core Insight

- Consider ($d = 10$)

Most significant ← Least significant

X		2	1	0
A		5	13	6
C	+	0	8	6
B		7	2	2

$i = 0, \quad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{i-1}$

Core Insight

- Consider ($d = 10$)

Most significant	→				Less significant	

X						
A	5	13	6	0	0	0 ....
C						
B	7	2	2	0	0	0 ....

$i = 0, \qquad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{i-1}$

Core Insight

- Consider ($d = 10$)

Most significant → Less significant

X		2						
A		5	13	6	0	0	0	....
C	+	0						
B		7	2	2	0	0	0	....

$$i = 0, \qquad a_0 + c_0 + x_0 = b_0$$

$$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{i-1}$$

Core Insight

- Consider ($d = 10$)

Most significant → Less significant

X		2	1					
A		5	13	6	0	0	0	....
C	+	0	8					
B		7	2	2	0	0	0	....

$i = 0, \qquad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{i-1}$

Core Insight

- Consider ($d = 10$)

Most significant → Less significant

X		2	1	0				
A		5	13	6	0	0	0	....
C	+	0	8	6				
B		7	2	2	0	0	0	....

$i = 0, \quad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{i-1}$

Core Insight

- Consider ($d = 10$)

Most significant → Less significant

X		2	1	0	0	0	0	....
A		5	13	6	0	0	0	....
C	+	0	8	6	0	0	0	....
B		7	2	2	0	0	0	....

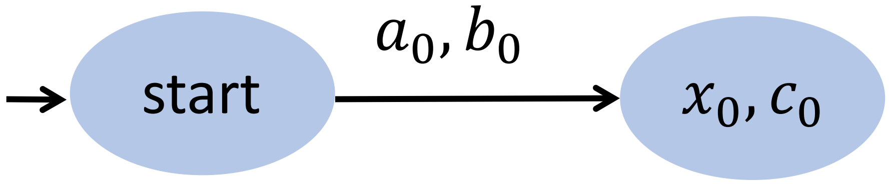
$i = 0, \quad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{i-1}$

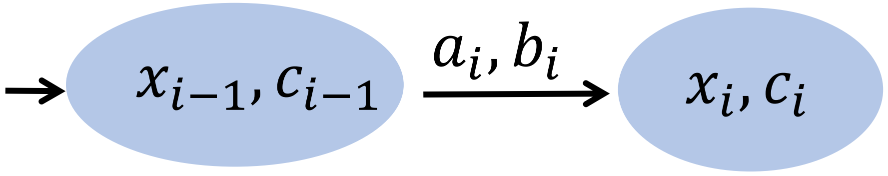
$\longrightarrow DS(A,d) + DS(C,d) = DS(B,d)$

Comparator: Construction

$a_0 + c_0 + x_0 = b_0$



$a_i + c_i + x_i = b_i + d \cdot x_{i-1}$



x_i, c_i are bounded integers, so automaton has finitely many states

Theorem

DS comparator for (integer) d, μ, R is an NBA with $O(\mu^2)$ states

Is Comparator an Automata

- With integer discount factors $d > 1$
  - $DS(A, d) = a_0 + \frac{a_1}{d} + \frac{a_2}{d^2} + \cdots = \quad (a_0 . a_1 a_2 \ldots)_d = A_d$
  - $DS(A, d) \leq \ DS(B, d)$ iff $A_d \leq B_d$
  - Is there C s.t. $DS(C, d) = B_d - A_d \geq 0$?
  - Non-determinism for arithmetic from most to lesser significant digits

Caveat: $a_i \geq d$ and -ve

Caveat: Difference of infinite sequences?

- With non-integer discount factors, comparator is not an automata

Theorem: Comparator is an NBA iff the discount-factor is an integer

Satisficing via Comparators

Satisficing Goal: Given, threshold value v

System-player wins

If cost of plays exceeds v

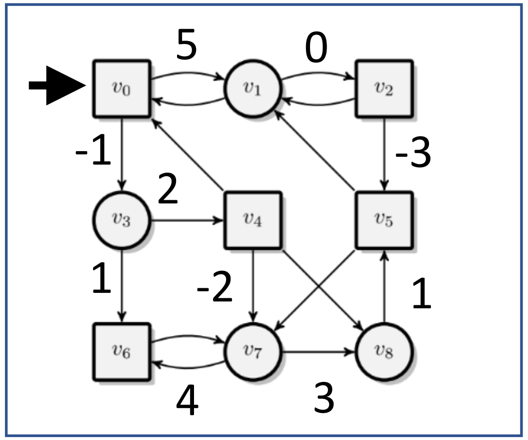
$\equiv$ $DS(A, d) > v$, where A is weight-sequence of a play

When the discount factor is an integer

$\equiv$ (A, V) is accepted by a comparator for $>$ where $DS(V, d) = v$

$\equiv$ Comparator for $>$ captures winning condition

Our Approach: Devising Integrated Solutions



Quantitative Game
Integer $d > 1$

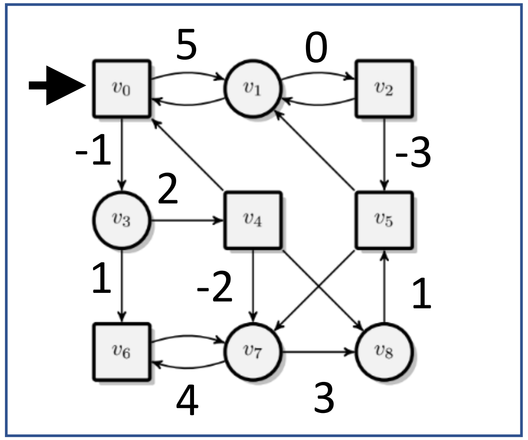
Temporal Goal → Automata-based methods

Satisficing Goal → Comparator-based methods

Algorithm is sound!

Automata-based methods combine well!

Our Approach: Devising Integrated Solutions



Quantitative Game

Integer $d > 1$

Temporal Goal	→	Automata-based methods
Satisficing Goal	→	Comparator-based methods

Algorithm is sound!

Automata-based methods combine well!

Approximate Comparator

[Bansal, et. al. 2021]

Given, non-integer discount factor $d > 1$,

approximation factor $0 < \varepsilon < 1$

Approximate comparator accepts (A, B), then

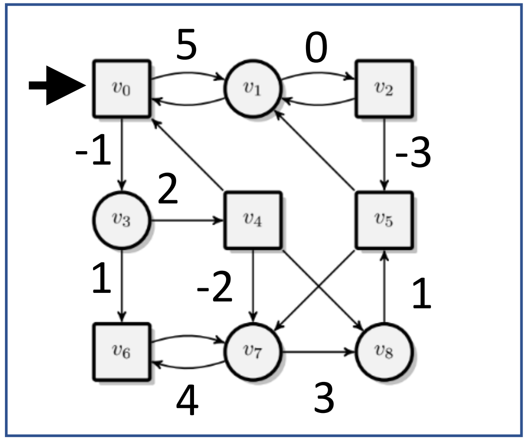
$$DS(A, d) > DS(B, d)$$

Approximate comparator rejects (A, B) , then

$$DS(A, d) \leq DS(B, d) \ + d \cdot \varepsilon$$

Theorem: Approximate comparator is an NBA (under some assumptions)

Our Approach: Devising Integrated Solutions



Quantitative Game

$d > 1$

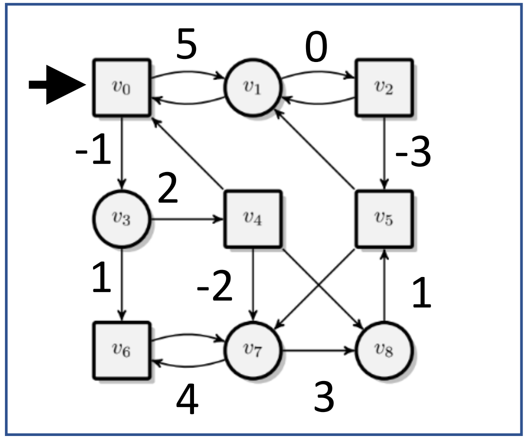
Temporal Goal → Automata-based methods

Satisficing Goal → Approx. comparator- or comparator-based methods

First theoretically sound algorithm for

Synthesis from qualitative and qualitative goals

Our Approach: Theoretical sound but ….



Quantitative Game

$d > 1$

Temporal Goal → Automata-based methods

Satisficing Goal → Approx. comparator- or comparator-based methods

All are NBAs → Deterministic NBAs

Exponential blow-up!

… not scalable

Deterministic Comparators

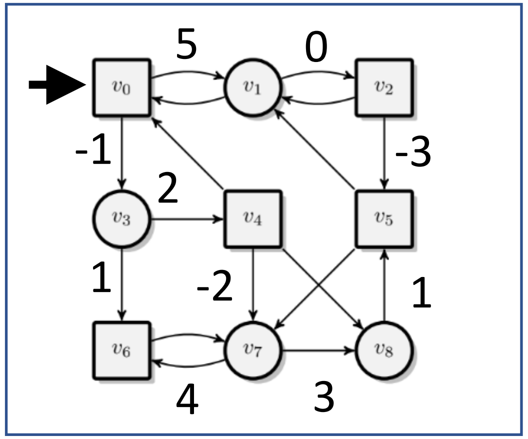
[Bansal and Vardi. CAV 2019]

Theorem: Comparators and Approximate Comparators are safety/co-safety automata

Safety and co-safety automata are deterministic NBAs that focus on finite prefixes

- Safety rejects words based on a finite prefix
- Co-safety accepts words based on a finite prefix
- $DS(A, d) = a_0 + \frac{a_1}{d} + \frac{a_2}{d^2} + \cdots = DS(A[0 \ldots i], d) + \frac{1}{d^i} \cdot DS(A[i \ldots], d)$
- As i increases, $\text{Tail} \to 0$. Eventually, only the finite prefix $A[0 \ldots i]$ matters!

Our Approach: Theoretically sound and ….



Quantitative Game

$d > 1$

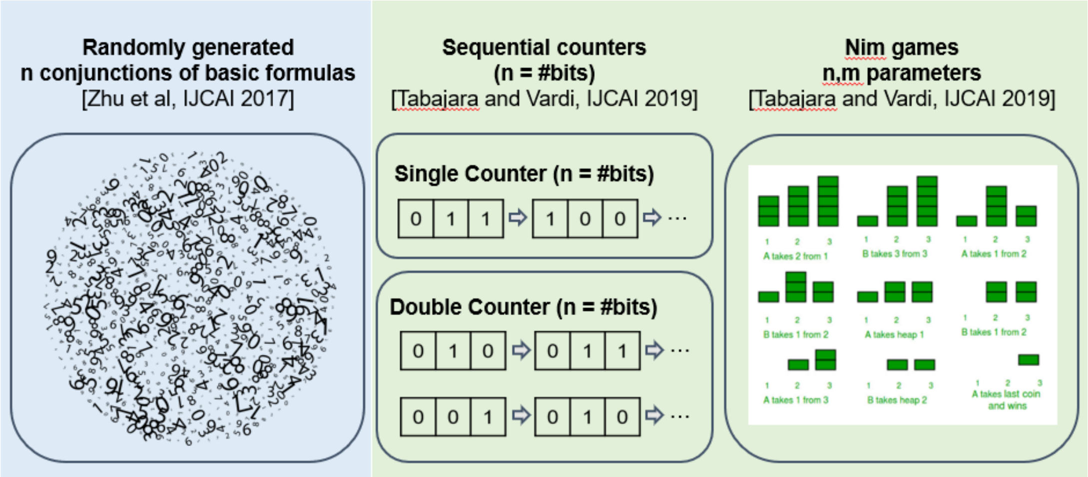
Temporal Goal → Automata-based methods

Satisficing Goal → **Deterministic** Approx. comparator- or comparator-based methods

… **scalable in practice**

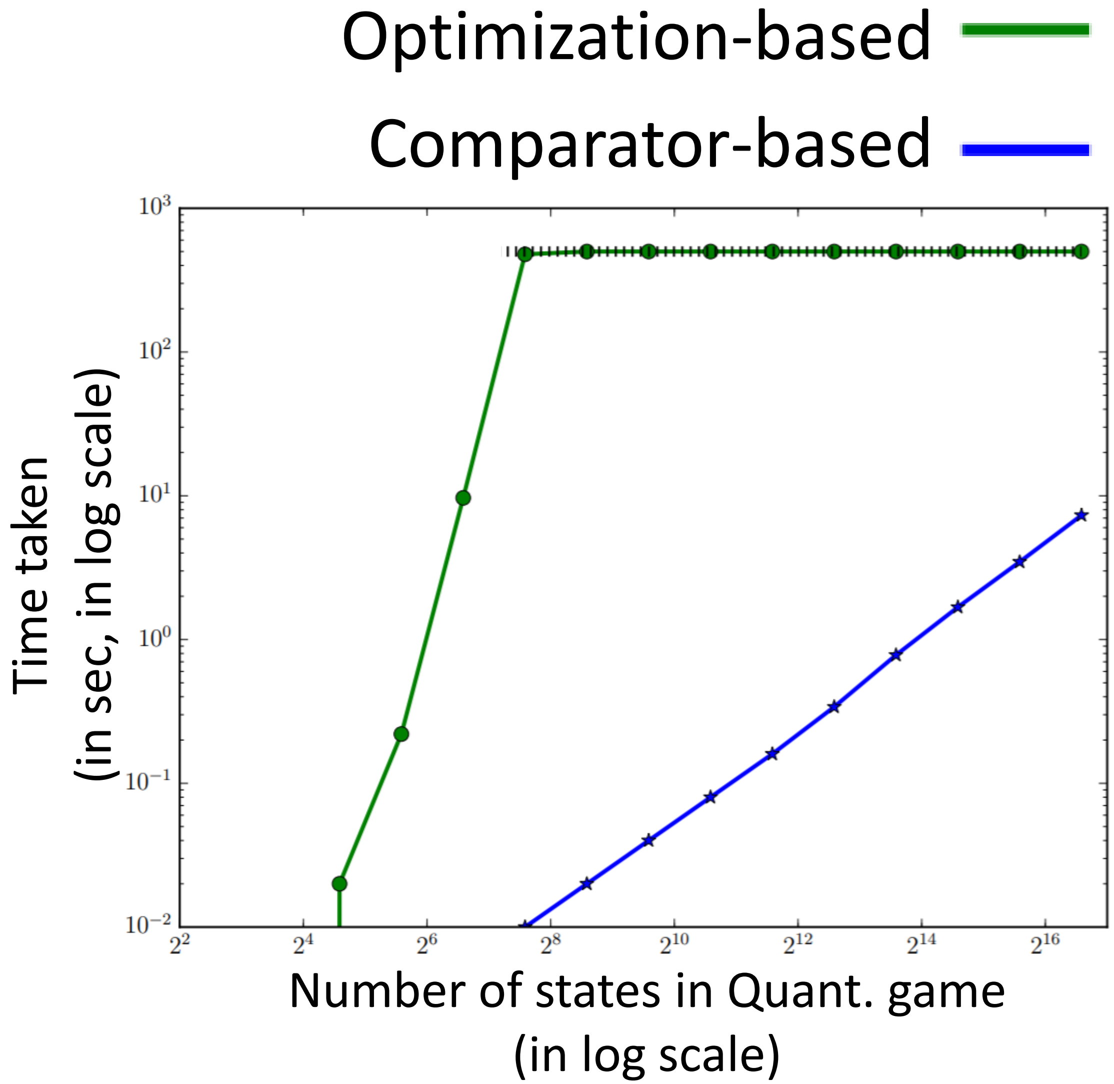
Synthesis from Satisficing Goals only: Optimization vs Comparators

300 benchmarks created from suite of LTLf benchmarks

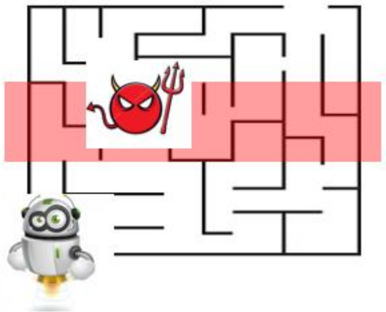


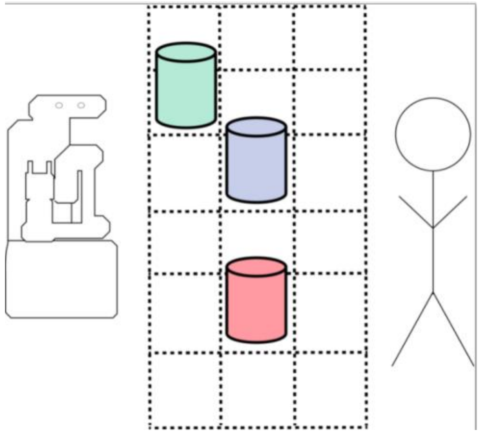
Optimization: ~140/300

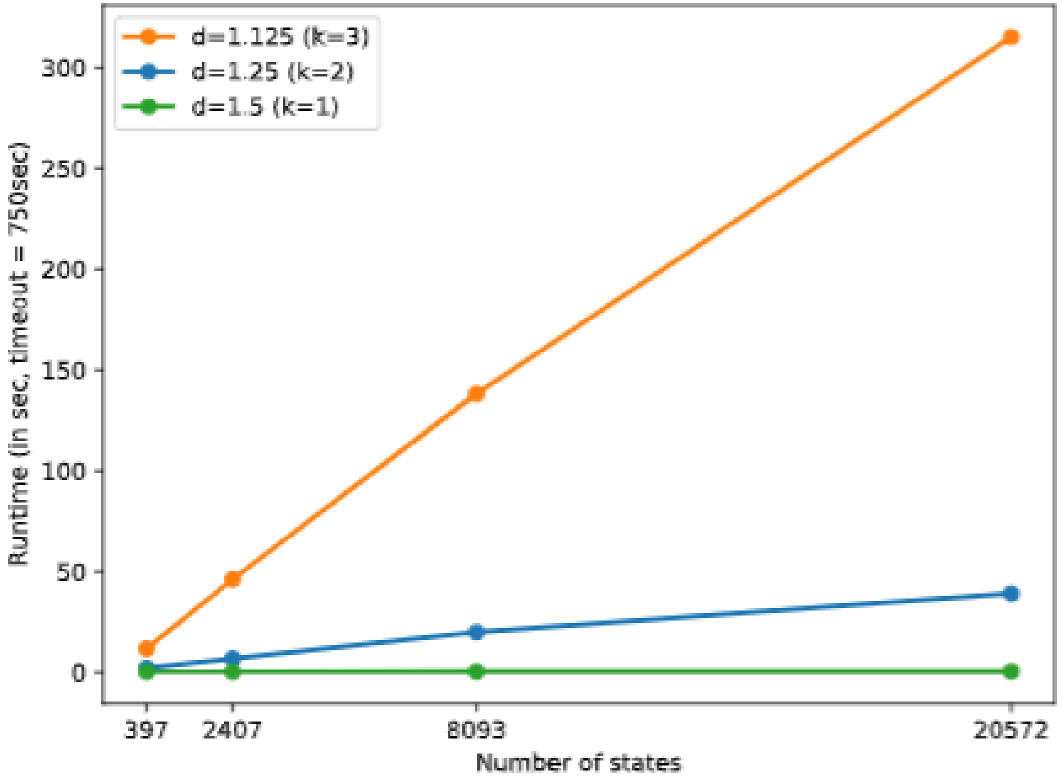
Comparator: ~ 285/300



Synthesis from Temporal and Satisficing Goals







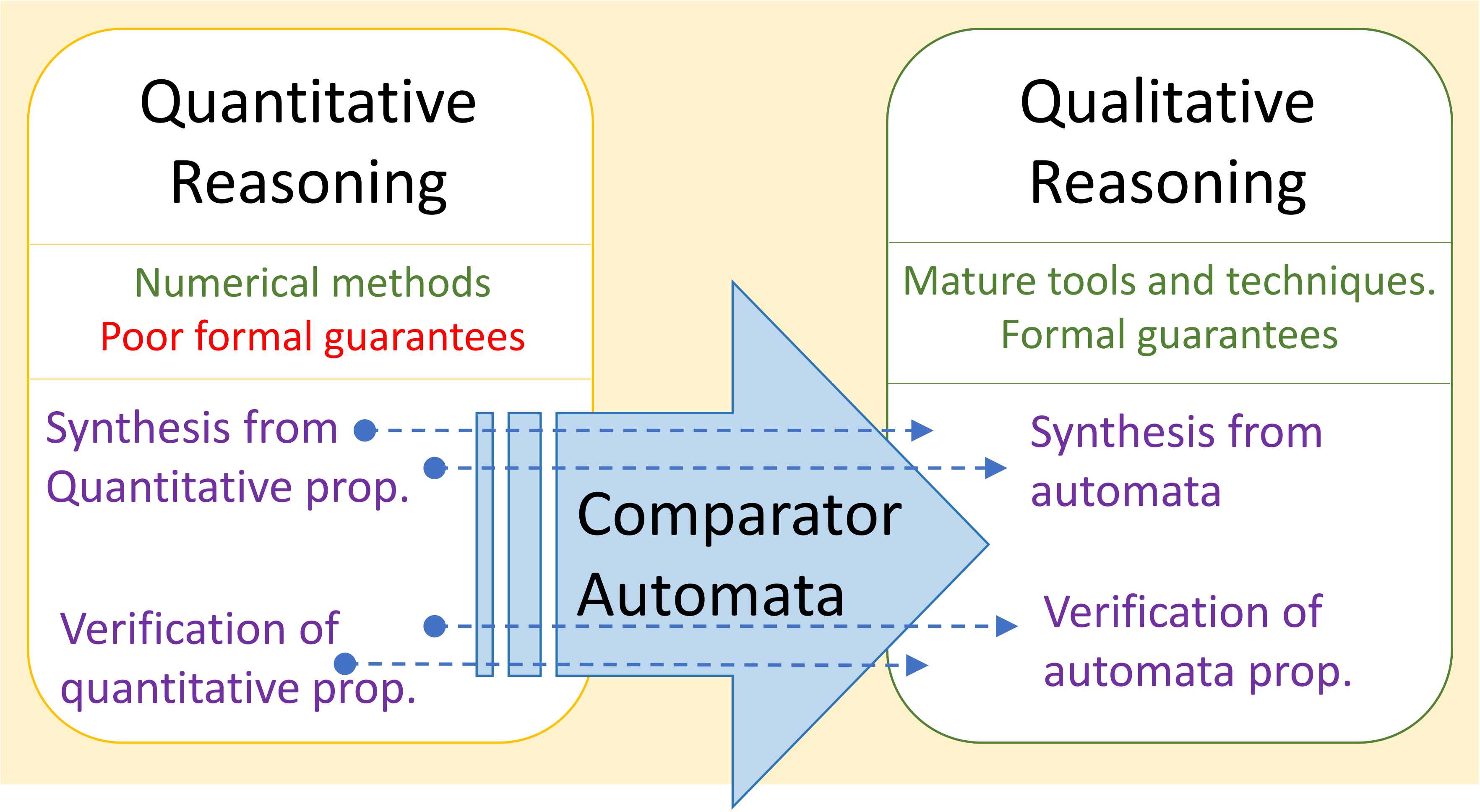
On today’s agenda

- ✓ Reactive synthesis for planning
  - ✓ Qualitative and Quantitative specifications
  - ✓ Existing algorithms fail to offer formal guarantees due to disparate-methods
- ✓ Automata-based quantitative reasoning
  - ✓ Towards algorithms with formal guarantees and scalable performance
  - ✓ Comparator-based algorithms
- Generality of approach in Formal Quantitative Reasoning
  - Beyond reactive synthesis

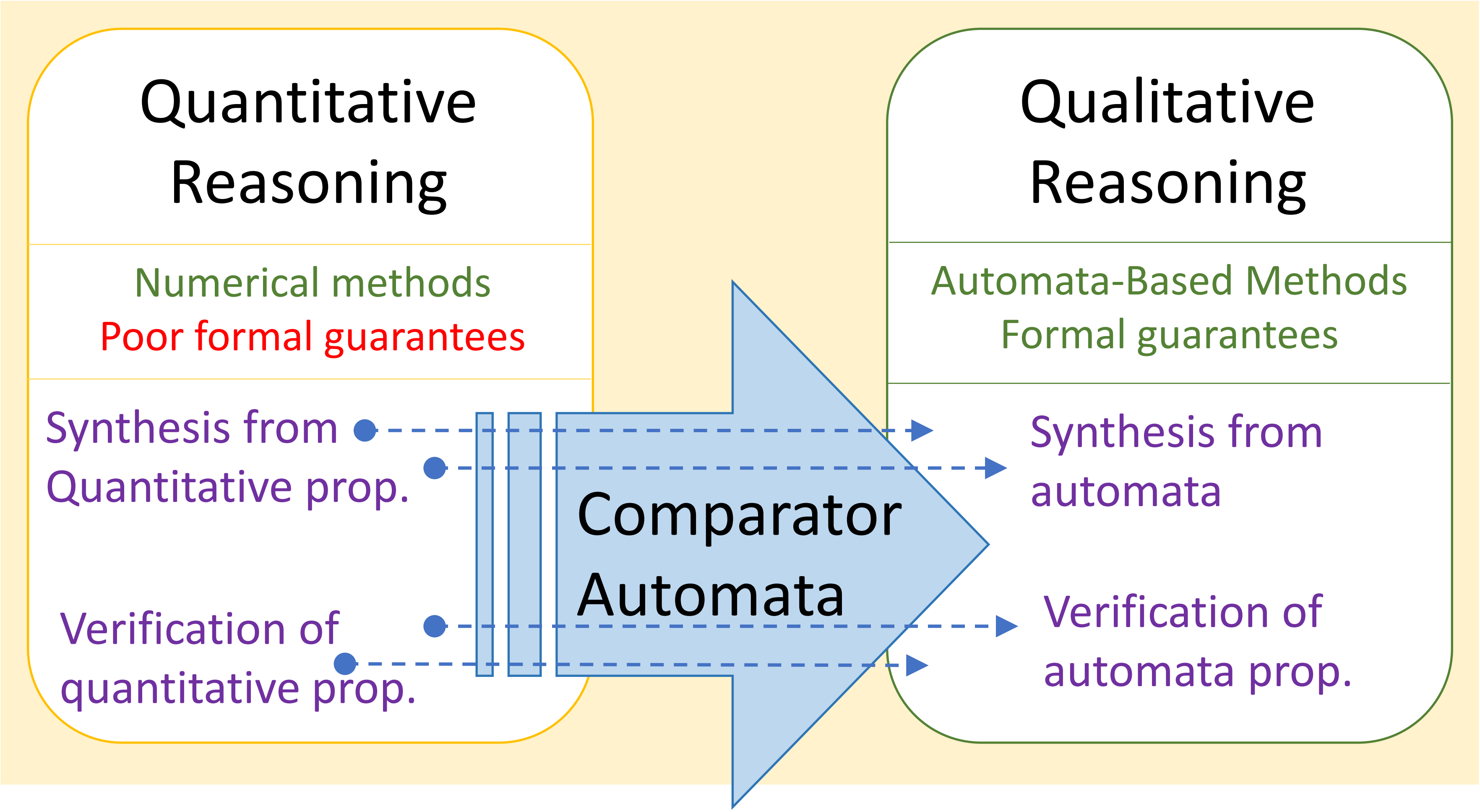
On today’s agenda

- ✓ Reactive synthesis for planning
  - ✓ Qualitative and Quantitative specifications
  - ✓ Existing algorithms fail to offer formal guarantees due to disparate-methods
- ✓ Automata-based quantitative reasoning
  - ✓ Towards algorithms with formal guarantees and scalable performance
  - ✓ Comparator-based algorithms
- Generality of approach in Formal Quantitative Reasoning
  - Beyond reactive synthesis

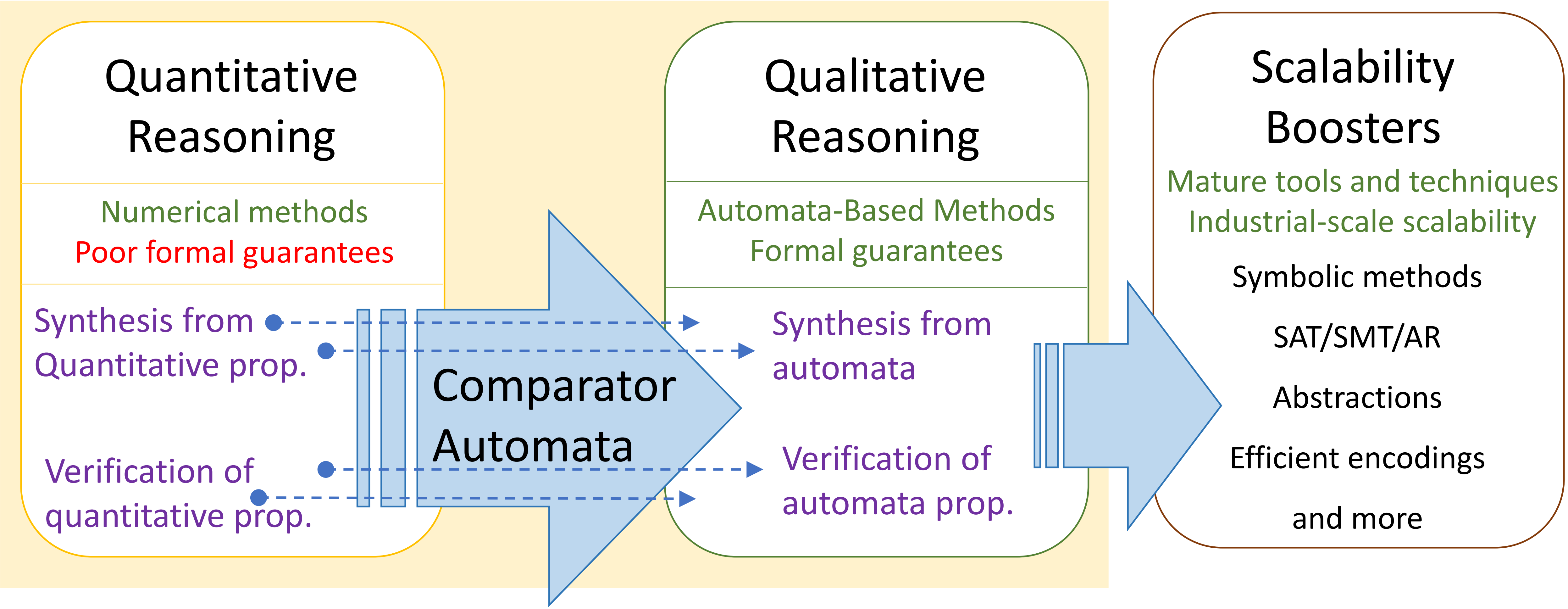
Automata-Based Quantitative Reasoning



Automata-Based Quantitative Reasoning



Automata-Based to Constraint Solving



Real-life applications in planning (robotics), formal guarantees on reinforcement learning

Towards designing intelligent systems via reactive synthesis

- Reactive synthesis from rich specifications
- This talk: Quantitaive + Qualitative specifications

- Automata-theoretic quantitative reasoning
  - Formal guarantees and scalability
  - Combine well with qualitative reasoning

- Beyond synthesis
  - Principled study of comparators and their capabilities
  - Quantitative reasoning -> Constraint solving

Many thanks to my collaborators

Swarat Chaudhuri
Rice U

Krishnendu Chatterjee
IST Austria

Moshe Y. Vardi
Rice U

Lydia Kavraki
Rice U

Andrew Wells
Tesla

Towards designing intelligent systems via reactive synthesis

- Reactive synthesis from rich specifications
- This talk: Quantitaive + Qualitative specifications

- Automata-theoretic quantitative reasoning
  - Formal guarantees and scalability
  - Combine well with qualitative reasoning

- Beyond synthesis
  - Principled study of comparators and their capabilities
  - Quantitative reasoning -> Constraint solving