Modeling, Analysis, and Synthesis of Quantitative System Requirements

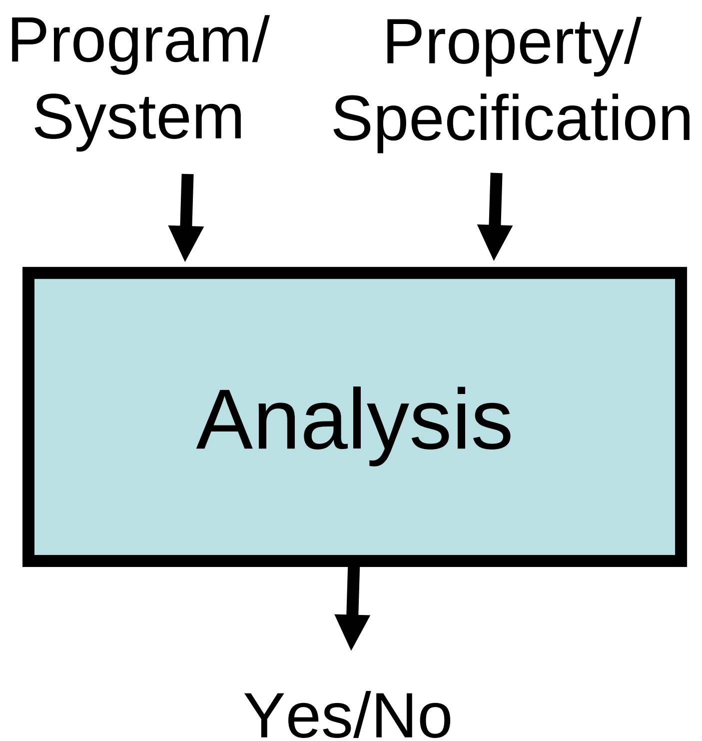
Tom Henzinger
IST Austria

Joint work with Krishnendu Chatterjee and Laurent Doyen.

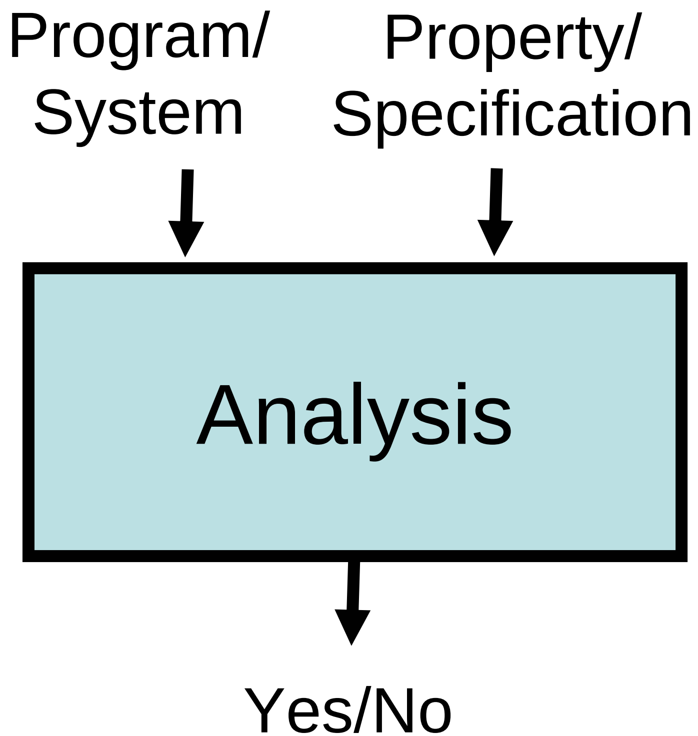
Outline

1. A Quantitative Systems Theory
2. Some Basic Open Problems
3. Some Promising Directions

Boolean Systems Theories

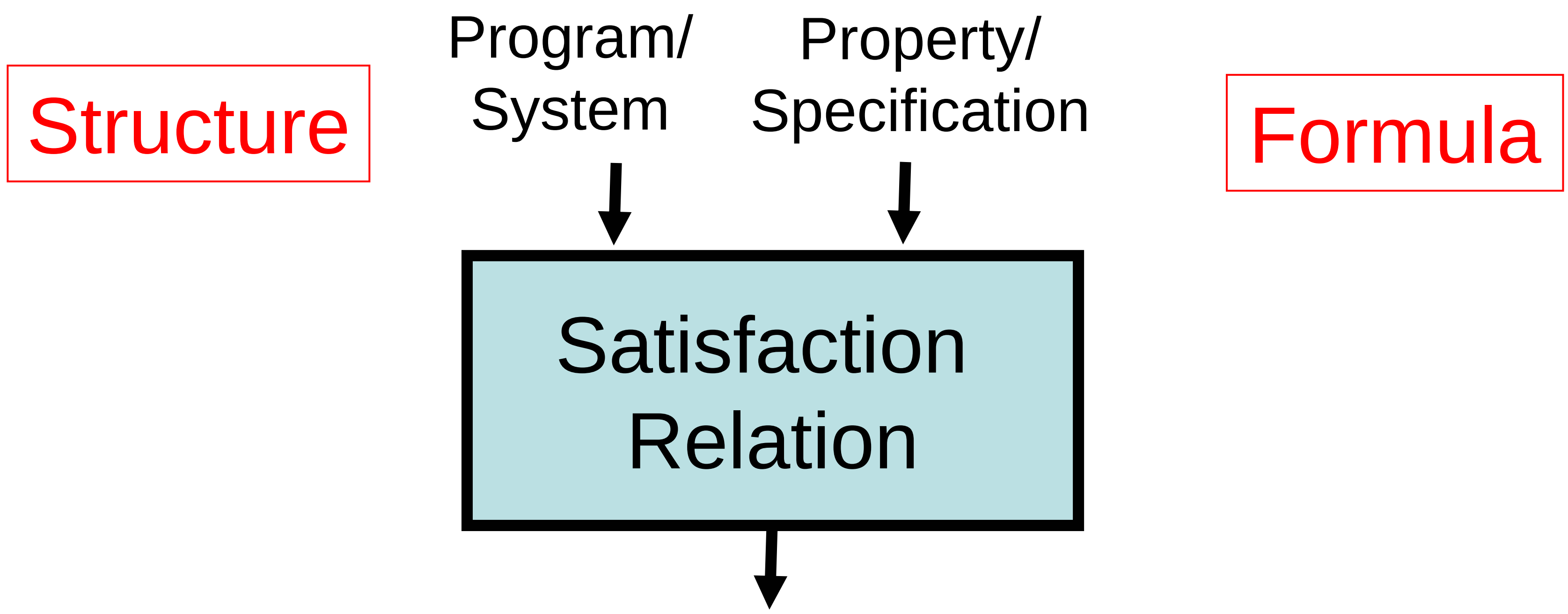


Boolean Systems Theories



-perhaps a proof
-perhaps some counterexamples
-perhaps even a proposed fix

Boolean Systems Theories



Yes/No

-perhaps a proof
-perhaps some counterexamples
-perhaps even a proposed fix

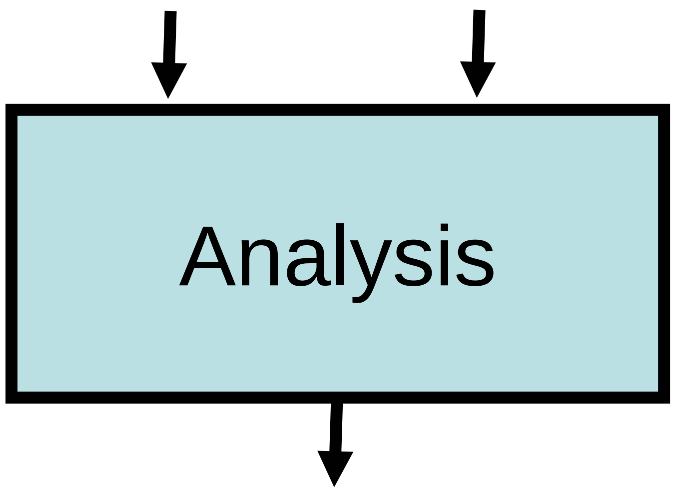
Boolean Systems Theories

Transition system.

Program/ System

Property/ Specification

Every request is followed by a grant.



Yes/No

-perhaps a proof
-perhaps some counterexamples
-perhaps even a proposed fix

Boolean Systems Theories

Timed automaton.

Quantitative Program/ System

Quantitative Property/ Specification

Every request is followed by a grant within 5 time units.

Analysis

Yes/No

-perhaps a proof
-perhaps some counterexamples
-perhaps even a proposed fix

Boolean Systems Theories

Quantitative Program/ System

Quantitative Property/ Specification

Markov process.

Every request is followed by a grant within probability 1/2.

Analysis

Yes/No

-perhaps a proof
-perhaps some counterexamples
-perhaps even a proposed fix

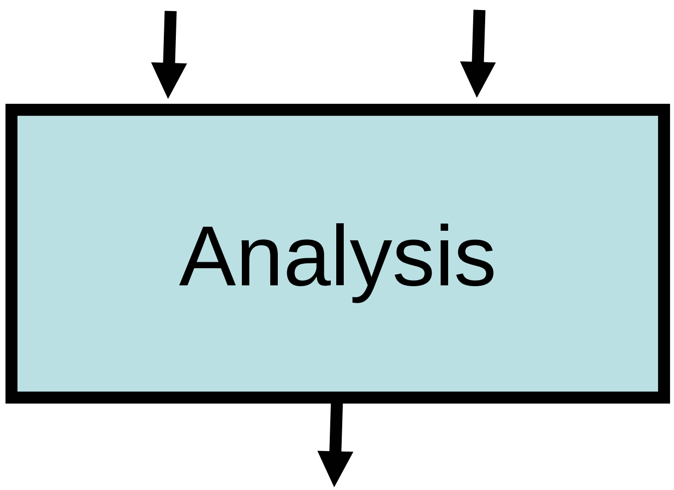
Boolean Systems Theories

Markov process.

Quantitative Program/ System

Quantitative Property/ Specification

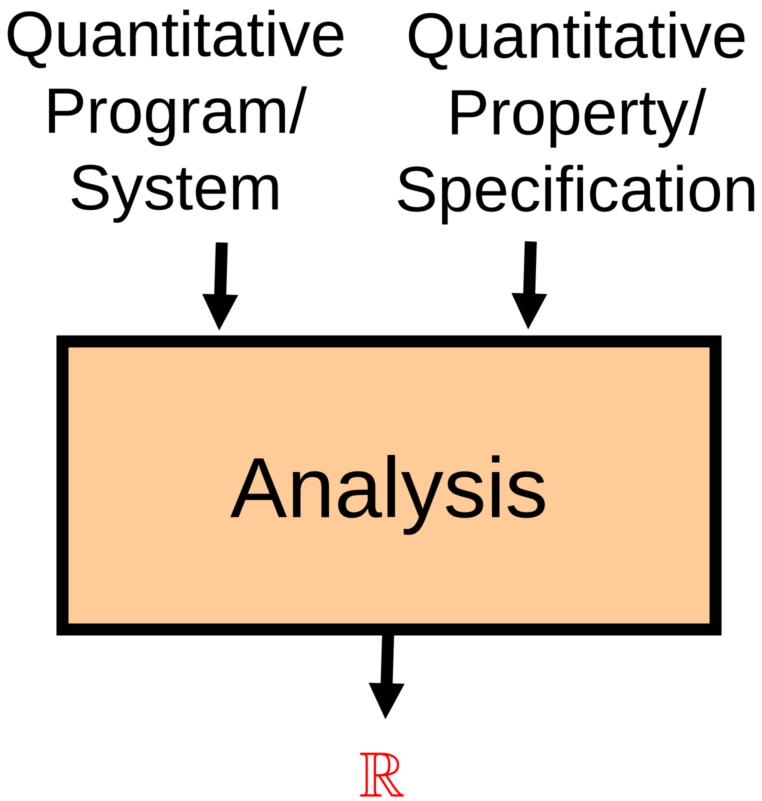
Every request is followed by a grant within probability 1/2.



$\mathbb{B}$

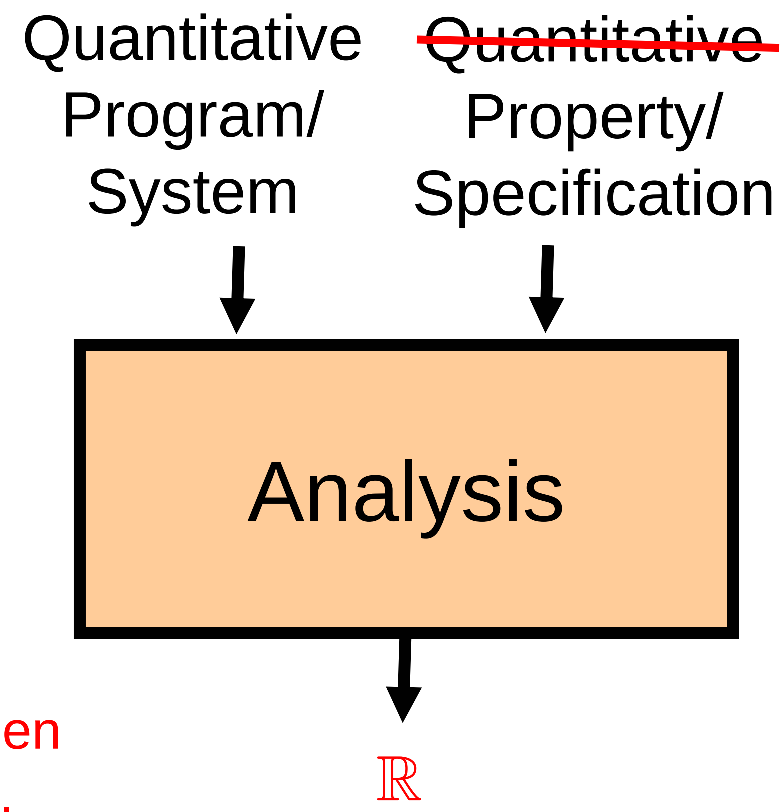
-perhaps a proof
-perhaps some counterexamples
-perhaps even a proposed fix

A Quantitative Systems Theory



-measure of "fit" between system and spec
-could be cost, quality, etc.

A Quantitative Systems Theory



Every request is followed by a grant.

The less time between requests and grants, the better.

-measure of "fit" between system and spec
-could be cost, quality, etc.

A Quantitative Systems Theory

~~Quantitative~~
Program/
System

~~Quantitative~~
Property/
Specification

Every request is followed by a grant.

Analysis

$\mathbb{R}$

The fewer unnecessary grants, the better.

-measure of “fit” between system and spec
-could be cost, quality, etc.

A Quantitative Systems Theory

Q1 Assigning values to behaviors

Q2 Assigning values to systems/properties

Q3 Assigning values to pairs of systems/properties

A Quantitative Systems Theory

Q1 Assigning values to behaviors

Boolean case: correct vs. incorrect behaviors

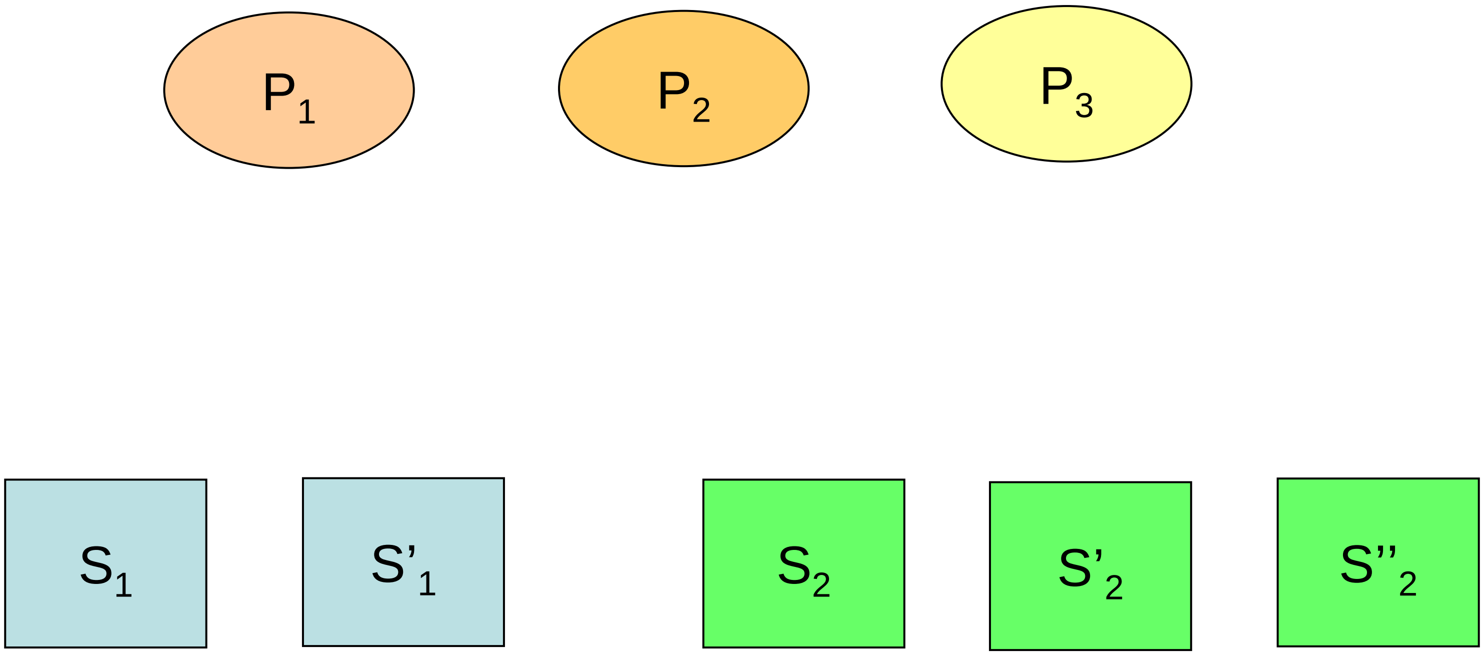
Q2 Assigning values to systems/properties

Boolean case: sets of behaviors (nondeterminism)

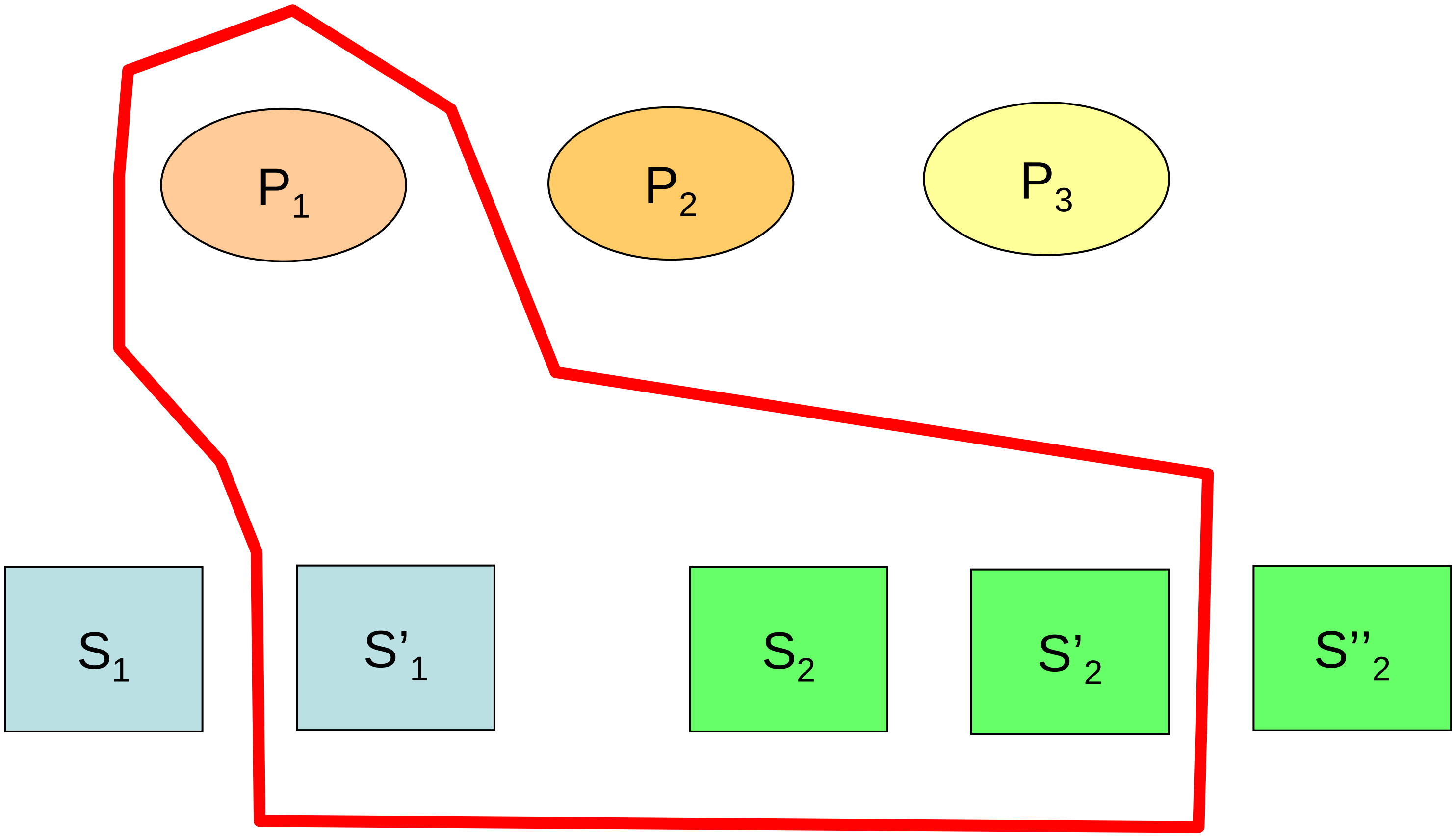
Q3 Assigning values to pairs of systems/properties

Boolean case: preorders (refinement)

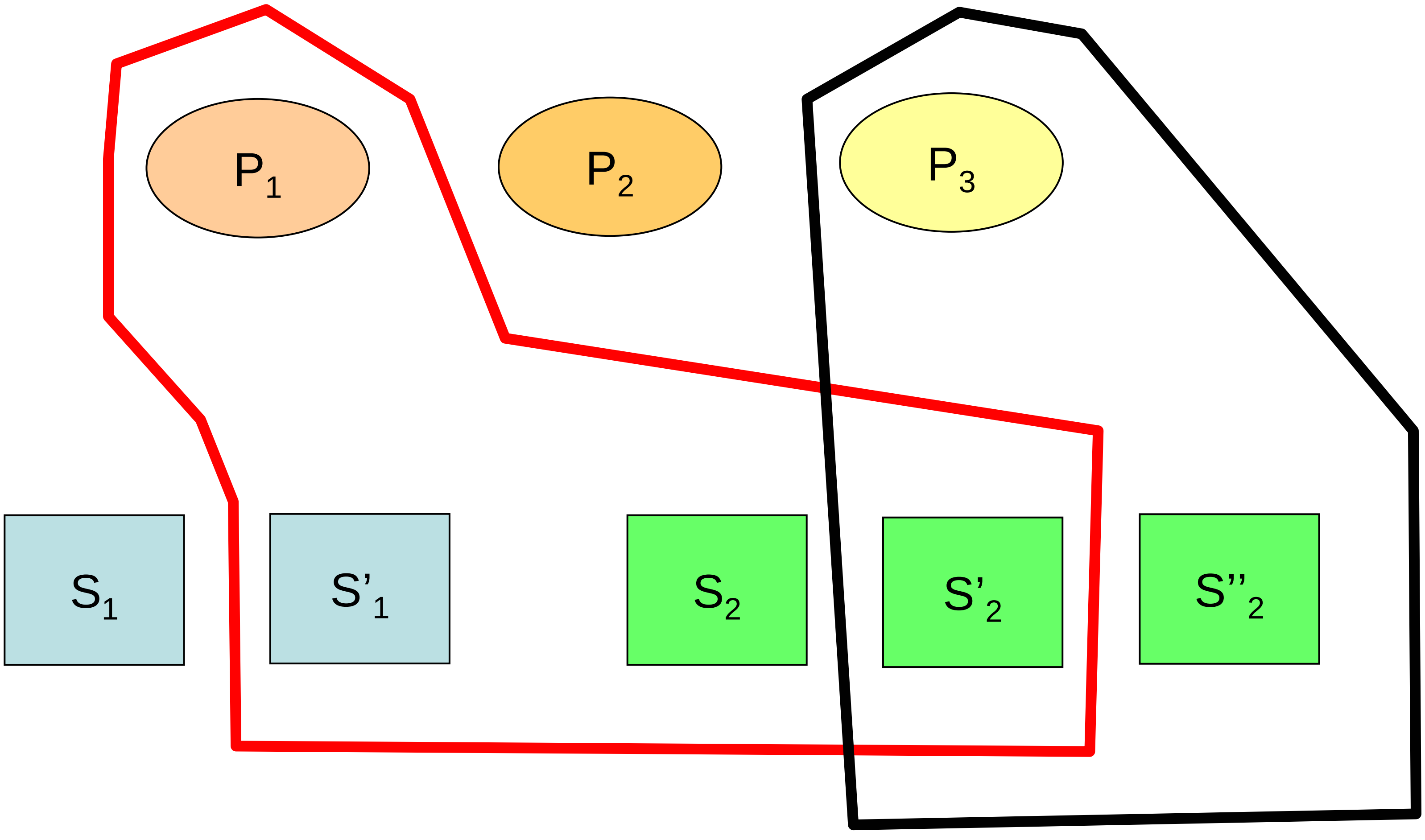
Boolean Systems Theories



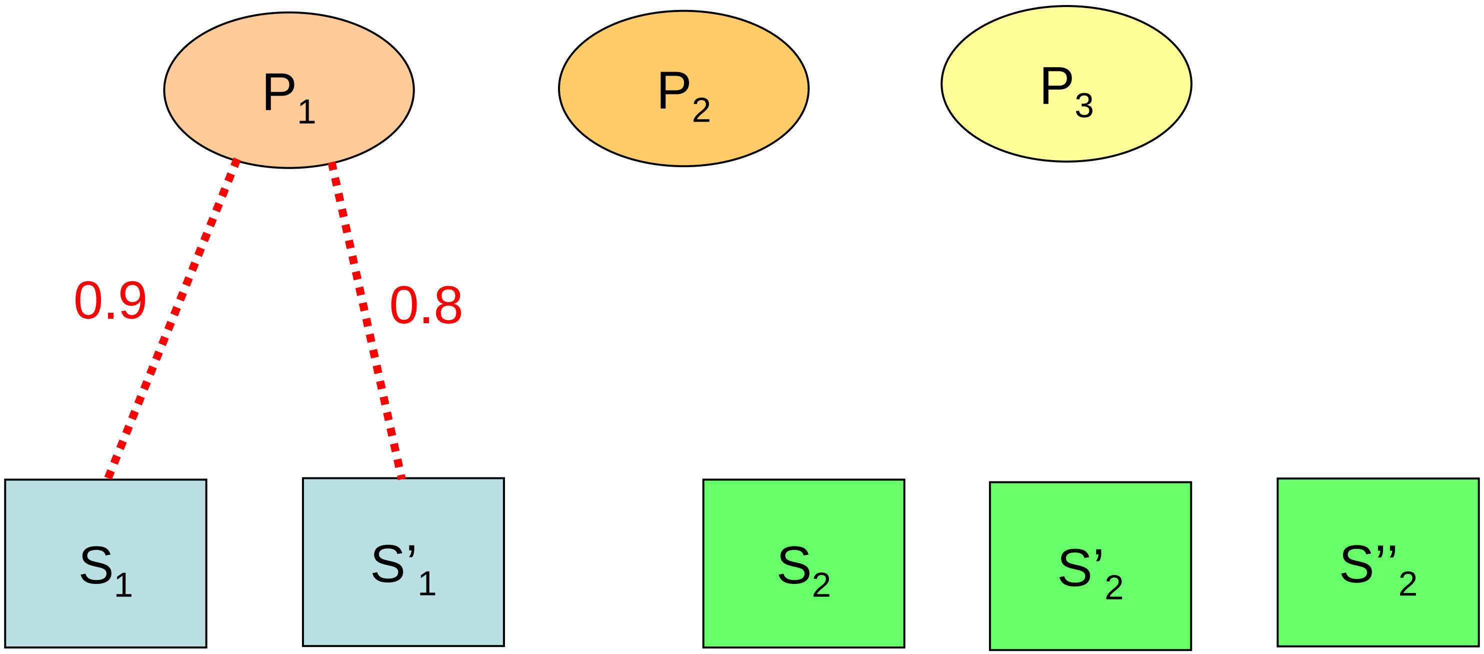
Boolean Systems Theories



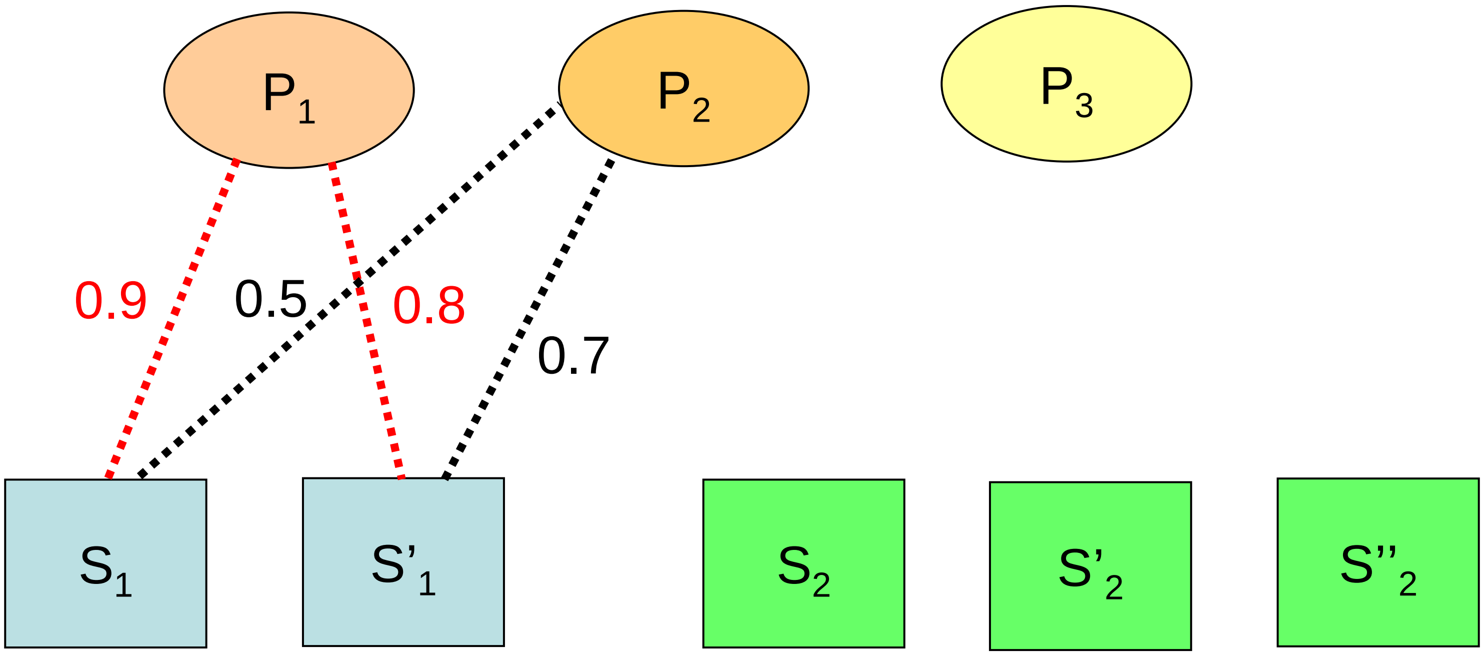
Boolean Systems Theories



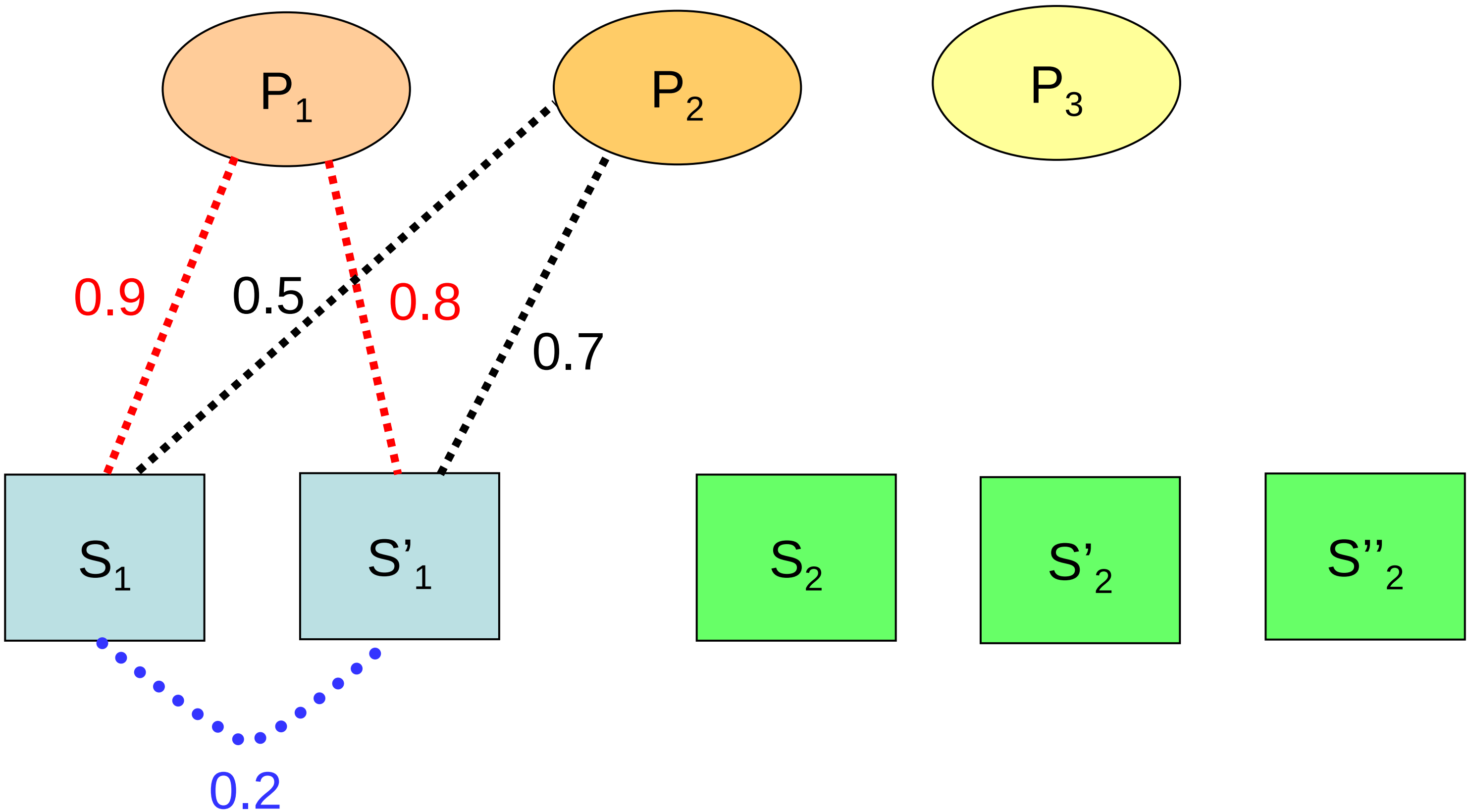
A Quantitative Systems Theory



A Quantitative Systems Theory



A Quantitative Systems Theory



Q1 Assigning Values To Behaviors

a. Probabilities

Q1 Assigning Values To Behaviors

a. Probabilities

b. Resource use

-worst case vs. average case (e.g. response time, QoS)
-peak vs. accumulative (e.g. power consumption)

Q1 Assigning Values To Behaviors

a. Probabilities

b. Resource use

-worst case vs. average case (e.g. response time, QoS)
-peak vs. accumulative (e.g. power consumption)

c. Quality measures

-discounting vs. long-run averaging (e.g. reliability)

Q1 Assigning Values To Behaviors: Safety

a: ok
b: fail

Discounted value ($0 < d < 1$):

aaaaaaaaaa...	1
aaaaaaab...	$1 - d^8$
aab...	$1 - d^3$
b...	0

Q1 Assigning Values To Behaviors: Safety

a: ok
b: fail

Discounted value ($0 < d < 1$):

aaaaaaaaaa...	1
aaaaaaab...	$1 - d^8$
aab...	$1 - d^3$
b...	0

Long-run average value:

aaaaaaaaaa...	1
abaabaaab...	1
aaabaaabaaab...	3/4
babbabbba...	0

Q2 Assigning Values To Systems

x: behaviors
w: observations (infinite words)
A,B: systems

$$A(w) \quad = \sup_x \{ \mathrm{val}(x) : \mathrm{obs}(x) = w \}$$

Q2 Assigning Values To Systems

x: behaviors
w: observations (infinite words)
A,B: systems

$$A(w) = \sup_x \{ val(x) : obs(x) = w \}$$
$$B(w) = \exp_x \{ val(x) : obs(x) = w \}$$

Q2 Assigning Values To Systems

x: behaviors
w: observations (infinite words)
A,B: systems

$$A(w) = \sup_x \{ val(x) : obs(x) = w \}$$
$$B(w) = \exp_x \{ val(x) : obs(x) = w \}$$

relative to input distribution

Q3 Assigning Distances To Systems

x: behaviors
w: observations (infinite words)
A,B: systems

$$A(w) = \sup_x \{ val(x) : obs(x) = w \}$$
$$B(w) = \exp_x \{ val(x) : obs(x) = w \}$$

$$diff(A,B) = \sup_w \{ |A(w) - B(w)| \}$$

Q3 Assigning Distances To Systems

x: behaviors
w: observations (infinite words)
A,B: systems

$A(w) = \sup_x \{ val(x) : obs(x) = w \}$
$B(w) = \exp_x \{ val(x) : obs(x) = w \}$

$diff(A,B) = \sup_w \{ |A(w) - B(w)| \}$

Boolean compositionality: if $A \leq A'$ then $A||B \leq A'||B$
Quantitative compositionality: $diff(A||B, A'||B) \leq f(diff(A,A'))$ [AFHMS]

Is there a Quantitative Systems Theory with

- an appealing mathematical formulation,
- useful expressive power, and
- good algorithmic properties?

(Like the boolean theory of ω-regularity.)

Outline

1 A Quantitative Systems Theory

2 Some Basic Open Problems:

- -Language inclusion for MDPs
- -Language inclusion for weighted automata

3 Some Promising Directions

Property = Language

Alphabet: Σ

$\Sigma = \{a,b,c\}$

Language: $L \subseteq \Sigma^\omega$

$L = (a^+b)^+(a^\omega \cup c^\omega) \cup (a^+b)^\omega$

$abaabaaabccccc... \in L$

$abcabc... \notin L$

Boolean Language

Alphabet: Σ

$\Sigma = \{a,b,c\}$

Language: $L \subseteq \Sigma^\omega$

$L = (a^+b)^+(a^\omega \cup c^\omega) \cup (a^+b)^\omega$

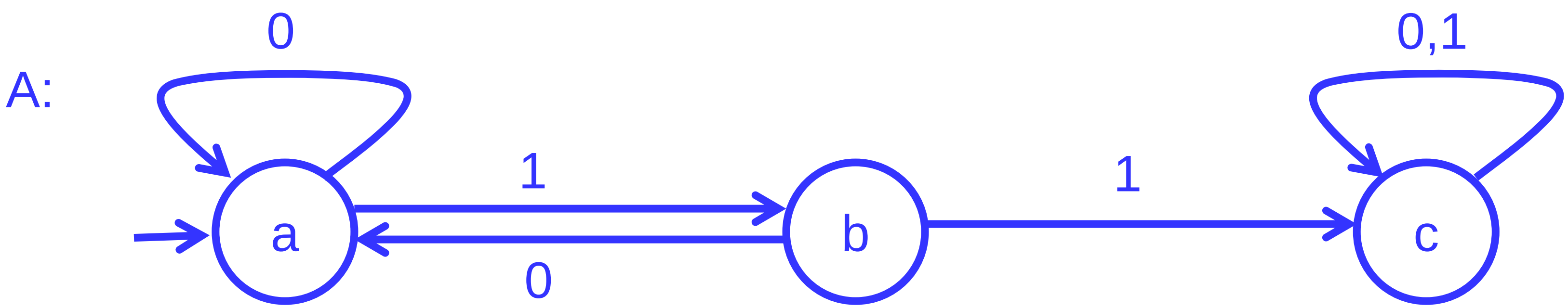
$abaabaaabcccccc... \in L$

$abcabc... \notin L$

$L: \Sigma^\omega \to \mathbb{B}$

Specification = Automaton

Q	states
$\lambda: Q \to \Sigma$	labeling
$q_0 \in Q$	initial state
Γ	choices
$\delta: Q \times \Gamma \to Q$	transition function

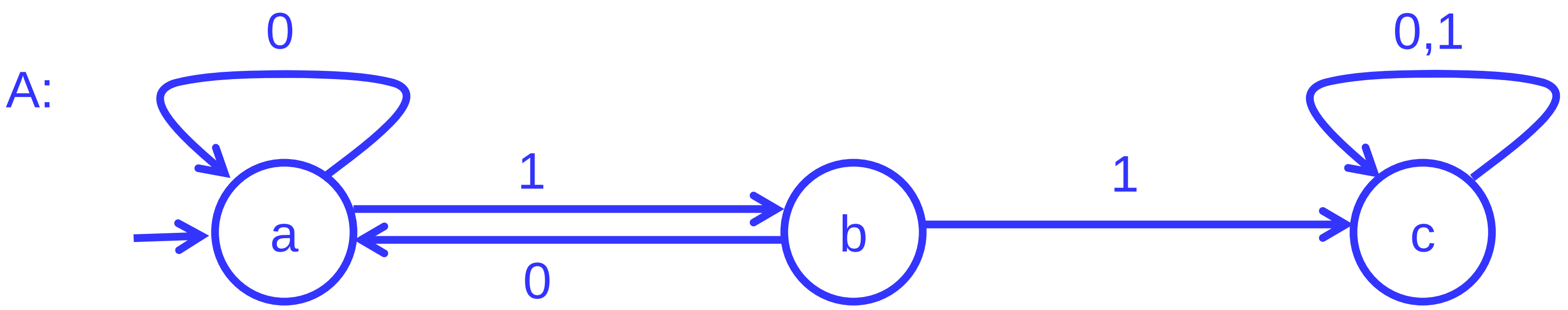


$\Gamma = \{0,1\}$

$L(A) = (a^+b)^+(a^\omega \cup c^\omega) \cup (a^+b)^\omega$

Specification = Automaton

Q	states
$\lambda: Q \to \Sigma$	labeling
$q_0 \in Q$	initial state
Γ	choices
$\delta: \; Q \times \Gamma \to Q$	transition function



“scheduler” 0101111... $\to$ aababccc... “outcome”

Specification = Automaton

Q	states
$\lambda: Q \to \Sigma$	labeling
$q_0 \in Q$	initial state
Γ	choices
$\delta: \; Q \times \Gamma \to Q$	transition function

Scheduler: $x: Q^+ \to \Gamma$

S ... set of schedulers

Outcome: $f(x) = q_0 q_1 q_2 \ldots$

where $\forall\, i : q_{i+1} = \delta(q_i, x(q_0 \ldots q_i))$

Language: $L = \{ \lambda(f(x)) : x \in S \}$

Satisfaction = Language Inclusion

Given two automata A and B, is $L(A) \subseteq L(B)$?

Satisfaction = Language Inclusion

Given two automata A and B, is $L(A) \subseteq L(B)$?

i.e. $\forall\, w \in \Sigma^\omega : L(A)(w) \leq L(B)(w)$

Satisfaction = Language Inclusion

Given two automata A and B, is $L(A) \subseteq L(B)$?

i.e. $\forall w \in \Sigma^\omega : L(A)(w) \leq L(B)(w)$

For finite/Buechi automata, PSPACE-complete.

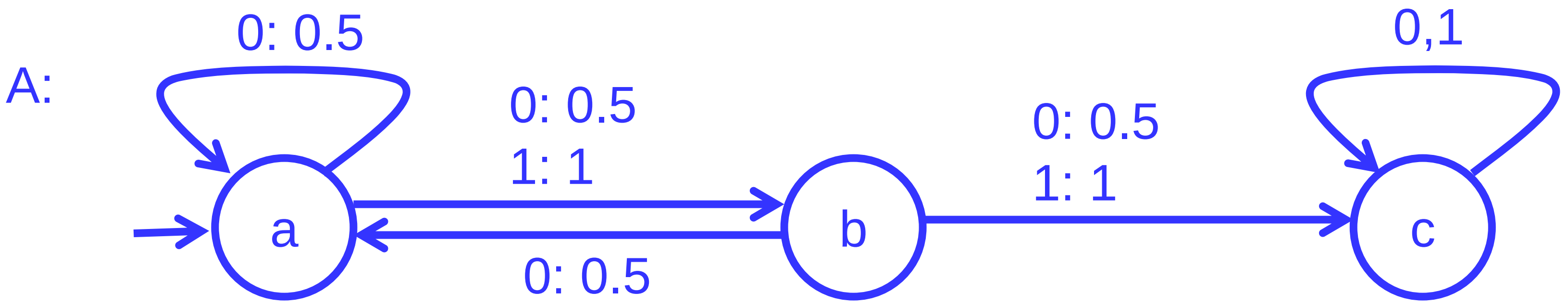
Probabilistic Language

Word:	element of Σ^ω
Probabilistic Word:	probability space on Σ^ω
Probabilistic Language:	set of probabilistic words

w: $ab\Sigma^\omega \rightarrow 1/2$
$aab\Sigma^\omega \rightarrow 1/4$
$aaab\Sigma^\omega \rightarrow 1/8$
...

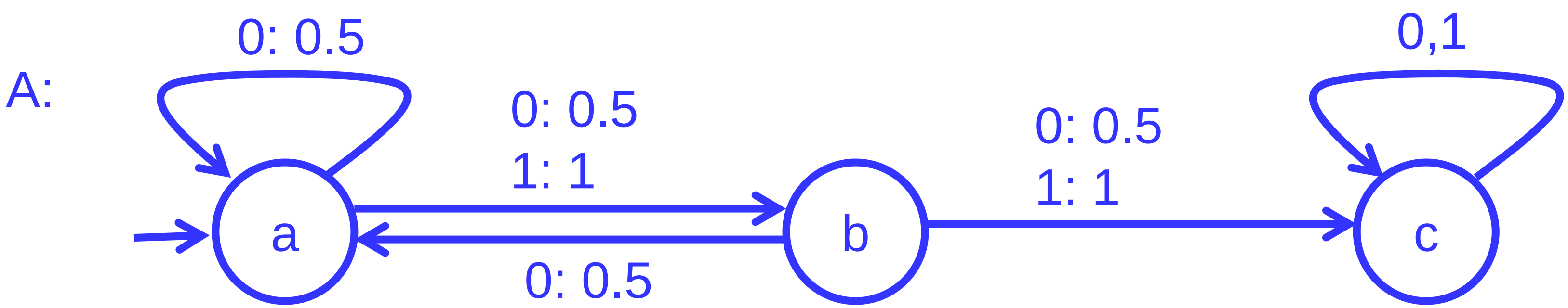
Markov Decision Process

Q	states
$\lambda: Q \to \Sigma$	labeling
$q_0 \in Q$	initial state
Γ	choices
$\delta: Q \times \Gamma \to D(Q)$	transition function



Markov Decision Process

Q	states
$\lambda: Q \to \Sigma$	labeling
$q_0 \in Q$	initial state
Γ	choices
$\delta: Q \times \Gamma \to D(Q)$	transition function



$0101111\ldots \to$ abccc... $\to 1/2$

aabccc... $\to 1/4$

...

Markov Decision Process

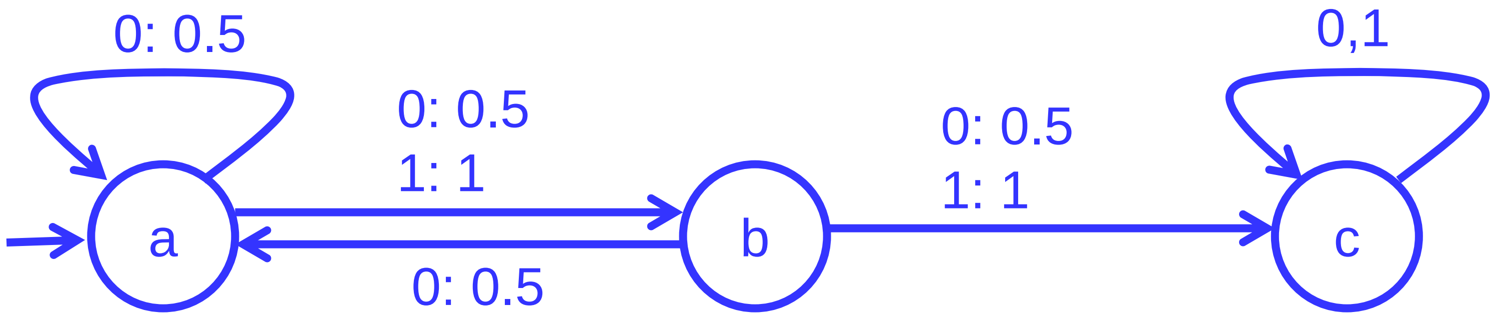
Q	states
$\lambda: Q \to \Sigma$	labeling
$q_0 \in Q$	initial state
Γ	choices
$\delta: \; Q \times \Gamma \to D(Q)$	transition function

Pure scheduler:	$x: Q^+ \to \Gamma$
Probabilistic scheduler:	$x: Q^+ \to D(\Gamma)$

Markov Decision Process

Q	states
$\lambda: Q \to \Sigma$	labeling
$q_0 \in Q$	initial state
Γ	choices
$\delta: \; Q \times \Gamma \to D(Q)$	transition function

A:



$\{0: 0.5, 1: 0.5\}^\omega \;\to\;$ abccc... $\to$ 9/16

aabccc... $\to$ 9/64

...

Probabilistic Language Inclusion

Given two MDPs A and B, is $L(A) \subseteq L(B)$?

Probabilistic Language Inclusion

Given two MDPs A and B, is $L(A) \subseteq L(B)$?

?

Probabilistic Language Inclusion

Given two MDPs A and B, is $L(A) \subseteq L(B)$?

?

Open even if specification B is deterministic (i.e. $|\Gamma| = 1$) and implementation scheduler required to be pure.

If both sides are deterministic, then it can be solved in polynomial time (equivalence of Rabin's probabilistic automata) [Tzeng, DHR].

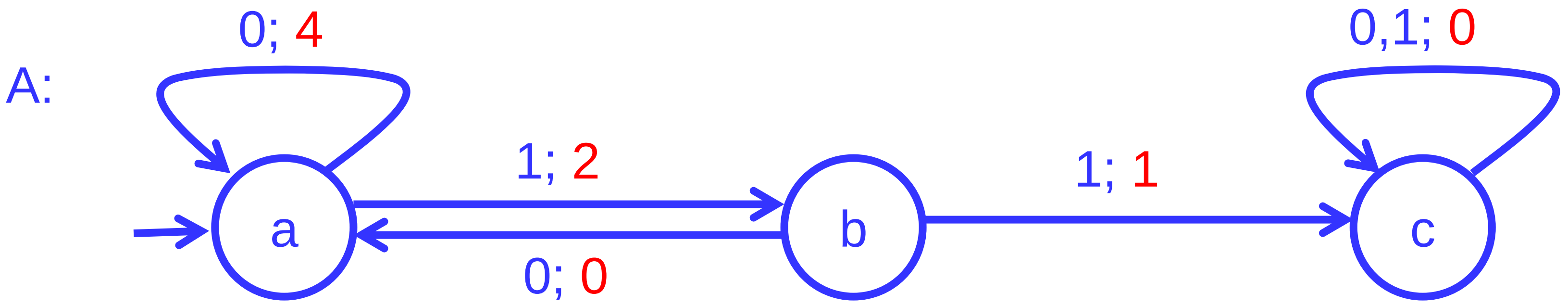
Quantitative Language

Language:	L: $\Sigma^\omega \to \mathbb{B}$
Quantitative Language:	L: $\Sigma^\omega \to \mathbb{R}$

$L(ab^\omega) = 1/2$
$L(aab^\omega) = 1/4$
$L(aaab^\omega) = 1/8$
...

Weighted Automaton

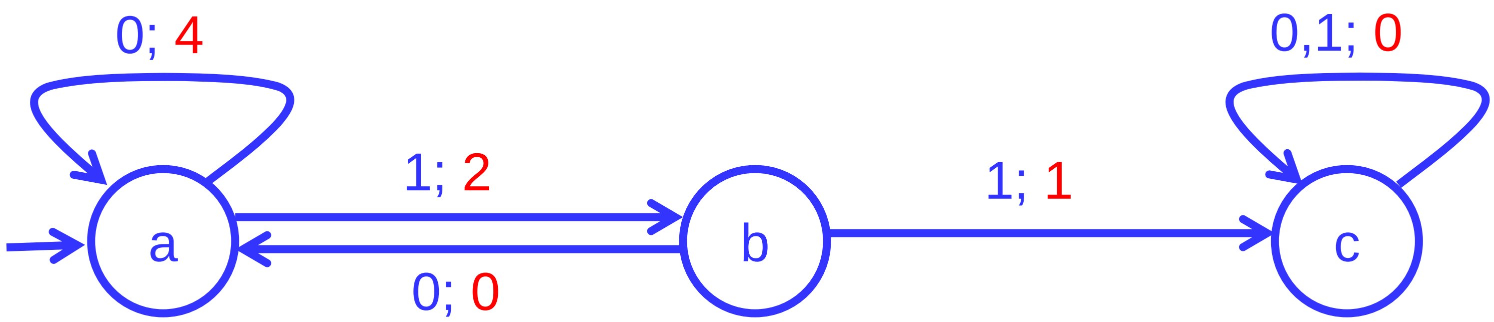
Q	states
$\lambda: Q \to \Sigma$	labeling
$q_0 \in Q$	initial state
Γ	choices
$\delta: Q \times \Gamma \to \mathbb{R} \times Q$	transition function



Weighted Automaton

Q	states
$\lambda: Q \to \Sigma$	labeling
$q_0 \in Q$	initial state
Γ	choices
$\delta: \; Q \times \Gamma \to \mathbb{R} \times Q$	transition function

A:



Value:

$0101111\ldots \to$ aababccc...; 4

$1111111\ldots \to$ abccc...; 2

Different Value Functions

Max value: $val(q_0 v_1 q_1 v_2 q_2 ...) = \sup\{ v_i : i \geq 1 \}$

Limsup value: $val = \lim_{n\to\infty} \sup\{ v_i : i \geq n \}$

Different Value Functions

Max value: $val(q_0 v_1 q_1 v_2 q_2 \ldots) = \sup\{ v_i : i \geq 1 \}$

(only 0 and 1 costs: finite automaton)

Limsup value: $val = \lim_{n \to \infty} \sup\{ v_i : i \geq n \}$

(only 0 and 1 costs: Buechi automaton)

Different Value Functions

Max value: $\mathrm{val}(q_0 v_1 q_1 v_2 q_2 \ldots) = \sup\{ v_i : i \geq 1 \}$

(only 0 and 1 costs: finite automaton)

Limsup value: $\mathrm{val} = \lim_{n \to \infty} \sup\{ v_i : i \geq n \}$

(only 0 and 1 costs: Buechi automaton)

Limavg value: $\mathrm{val} = \lim_{n \to \infty} 1/n \cdot \sum_{1 \leq i \leq n} v_i$

Different Value Functions

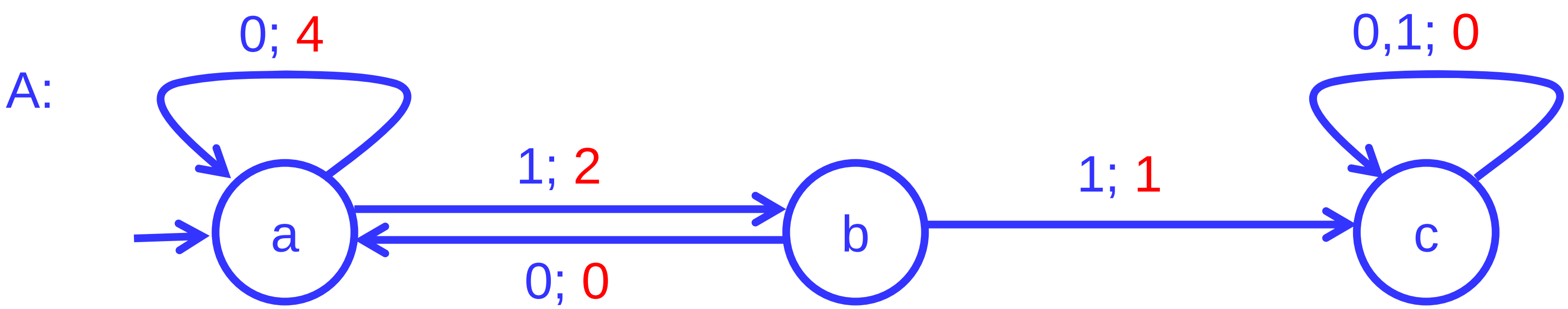
Max value: $val(q_0 v_1 q_1 v_2 q_2 \ldots) = \sup\{ v_i : i \geq 1 \}$
(only 0 and 1 costs: finite automaton)

Limsup value: $val = \lim_{n \to \infty} \sup\{ v_i : i \geq n \}$
(only 0 and 1 costs: Buechi automaton)

Limavg value: $val = \lim_{n \to \infty} 1/n \cdot \sum_{1 \leq i \leq n} v_i$

Discounted: $val = \sum_{i \geq 1} d^i \cdot v_i$ for some $0<d<1$

Weighted Automaton



Limsup value:

01010101... $\rightarrow$ aabababab...; 2
11111111... $\rightarrow$ abccc...; 0

Limavg value:

01010101... $\rightarrow$ aabababab...; 1
11111111... $\rightarrow$ abccc...; 0

Discounted:
(d = 0.5)

01010101... $\rightarrow$ aabababab...; 2.66...
11111111... $\rightarrow$ abccc...; 1.25

Quantitative Language Inclusion

Given two weighted automata A and B, is
$\forall\, w \in \Sigma^{\omega} : L(A)(w) \leq L(B)(w)$?

Quantitative Language Inclusion

Given two weighted automata A and B, is
$\forall$ w $\in$ Σ^ω : L(A)(w) $\leq$ L(B)(w) ?

For max and limsup values: PSPACE.
For limavg and discounted values: Open.

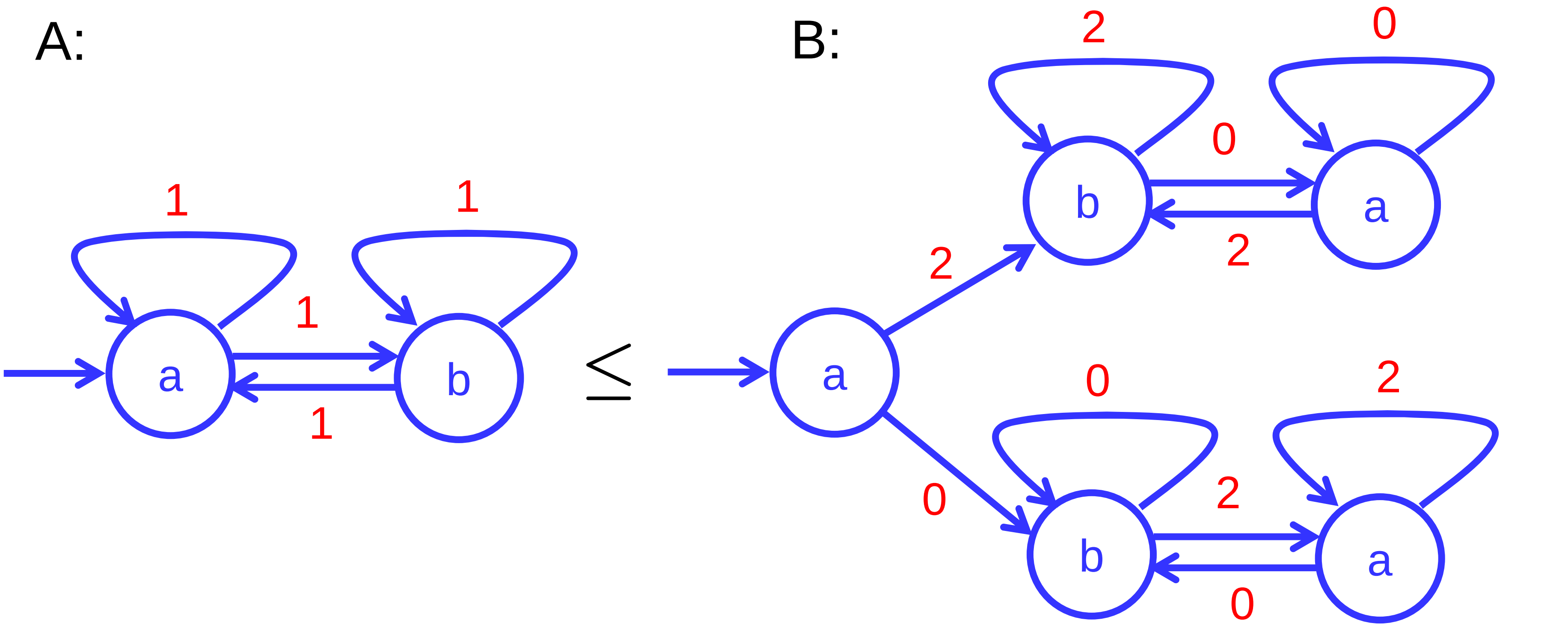
Quantitative Language Inclusion

Given two weighted automata A and B, is
$\forall\, w \in \Sigma^\omega : L(A)(w) \leq L(B)(w)$?

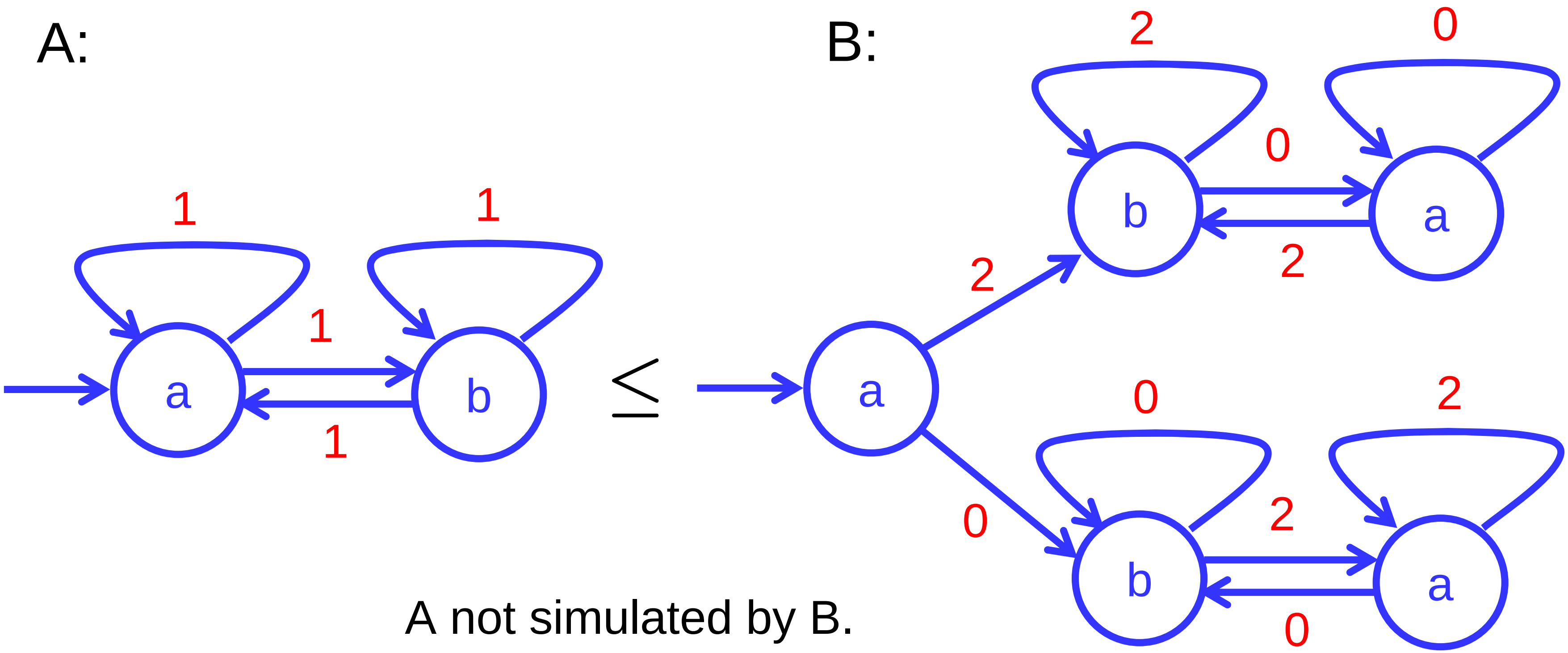
For max and limsup values: PSPACE.
For limavg and discounted values: Open.

If specification B is deterministic,
then it can be solved in polynomial time [CDH].

Quantitative Simulation

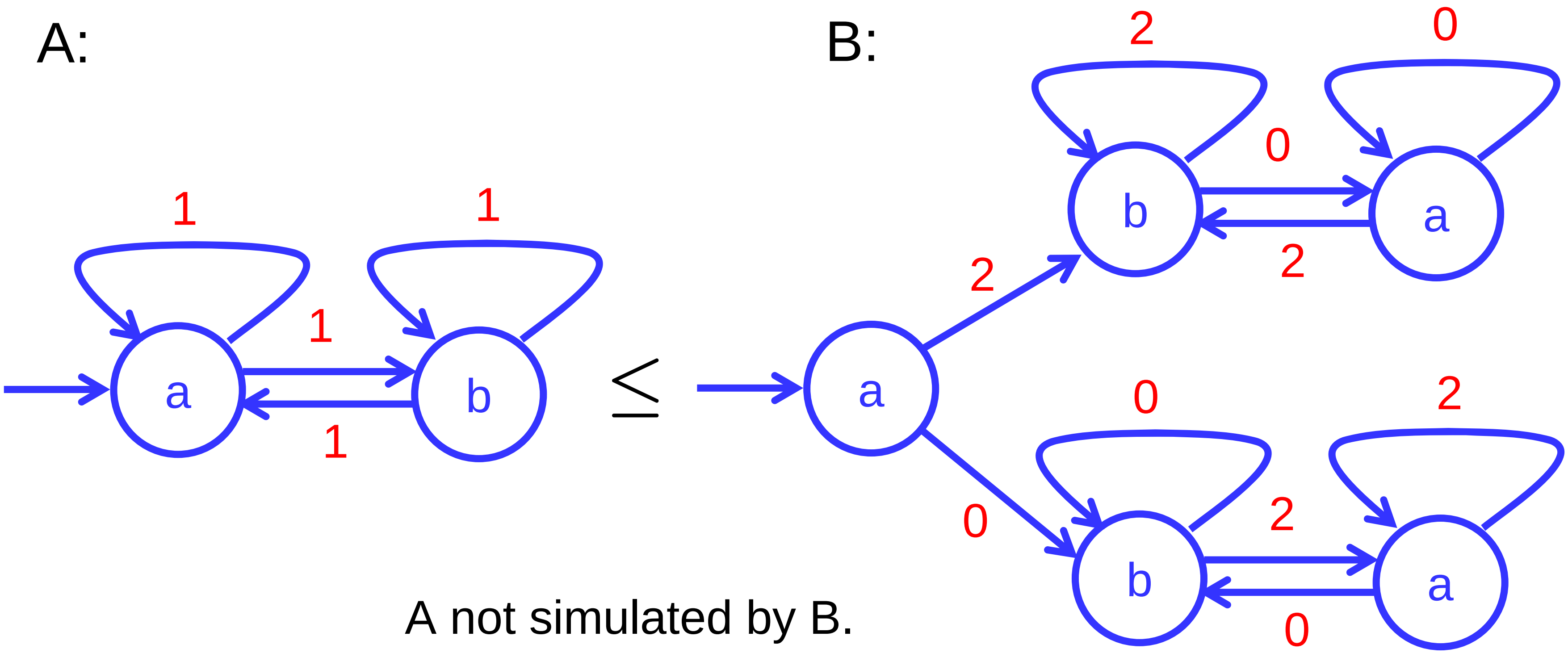


Quantitative Simulation



A not simulated by B.

Quantitative Simulation



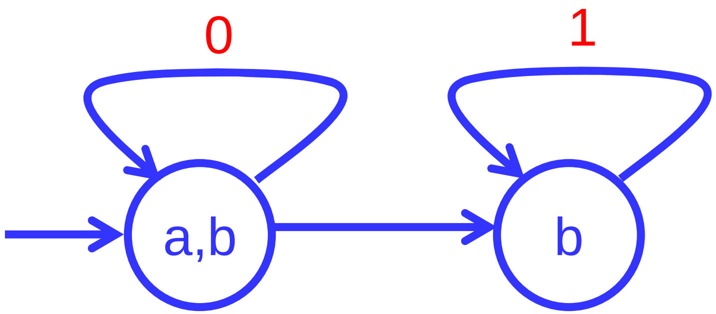
A not simulated by B.

Simulation game solvable in P for max values;
in NP ∩ coNP for limsup, limavg, discounted values [CDH].

Quantitative Expressiveness

E.g. limavg automata not determinizable [CDH]:

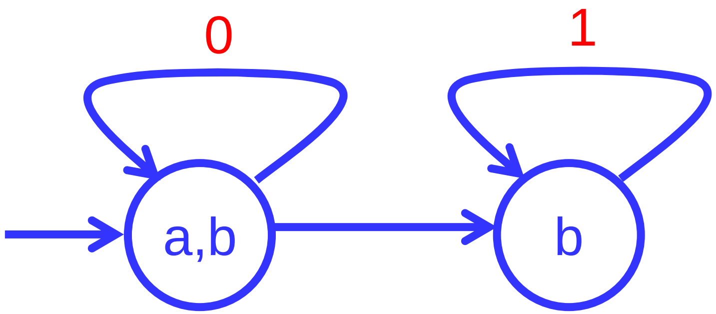
$\Sigma^* b^\omega$ expressible by a nondeterministic limavg automaton.



Quantitative Expressiveness

E.g. limavg automata not determinizable [CDH]:

$\Sigma^* b^\omega$ expressible by a nondeterministic limavg automaton.



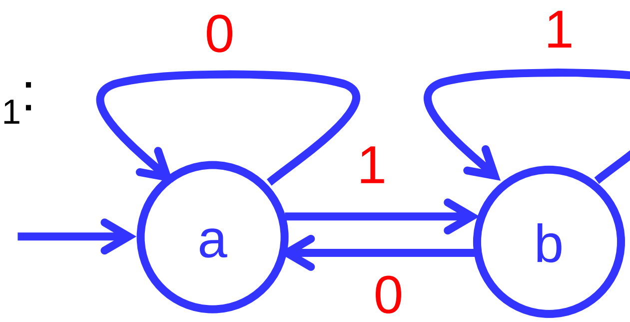
$\Sigma^* b^\omega$ not expressible by a deterministic limavg automaton.

Every b-cycle would need weight 1.
Consider $w_n = (ab^n)^\omega$.
Then $val(w_n) = 1$ for sufficiently large n, but $w_n \notin \Sigma^* b^\omega$.

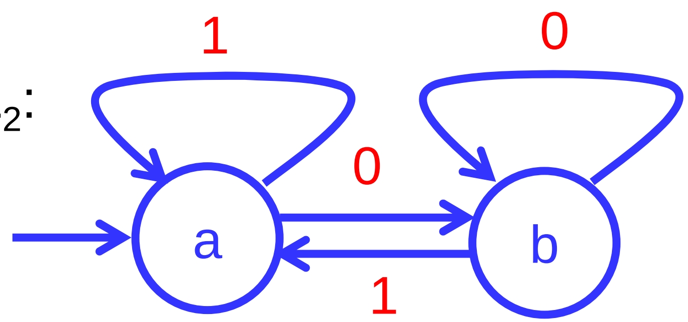
Quantitative Closure Properties

E.g. limavg automata not closed under min [CDH]:

L_1:

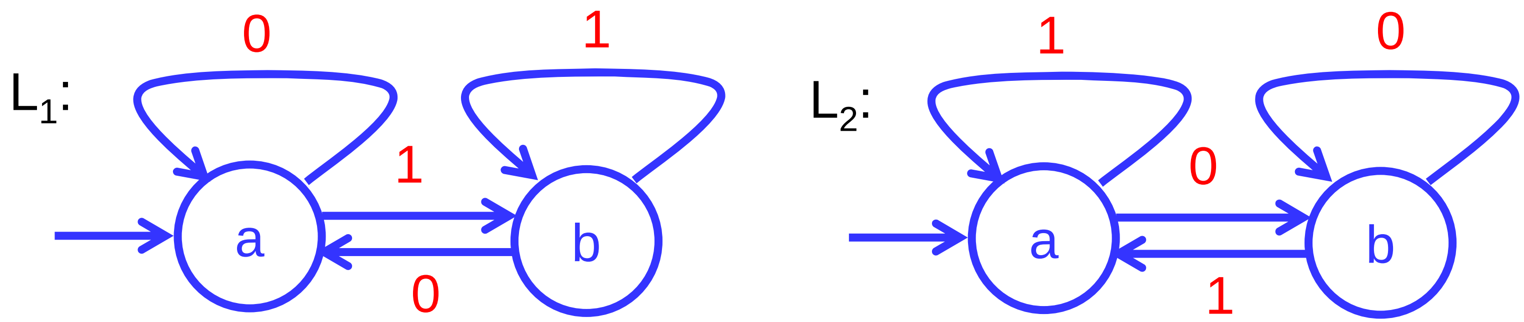


L_2:



Quantitative Closure Properties

E.g. limavg automata not closed under min [CDH]:



$\min(L_1,L_2)$ not expressible by a limavg automaton.

Consider $w_n = (a^n b^n)^\omega$ for large n.
Some a-cycle or b-cycle would need average positive weight.
Then some word ua^ω or ub^ω would have a positive value.

Outline

1 The Quantitative Verification Agenda

2 Some Basic Open Problems:

- -Language inclusion for MDPs
- -Language inclusion for weighted automata

3 Some Promising Directions

Outline

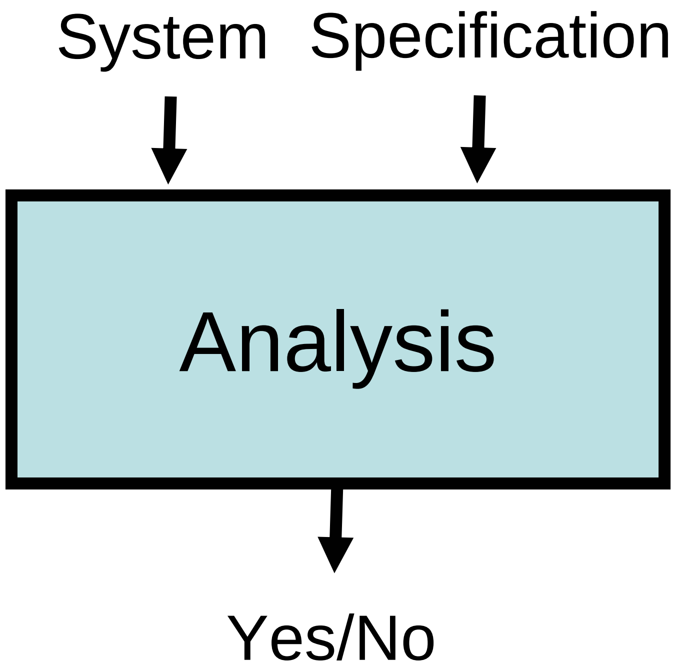
1 The Quantitative Verification Agenda

2 Some Basic Open Problems

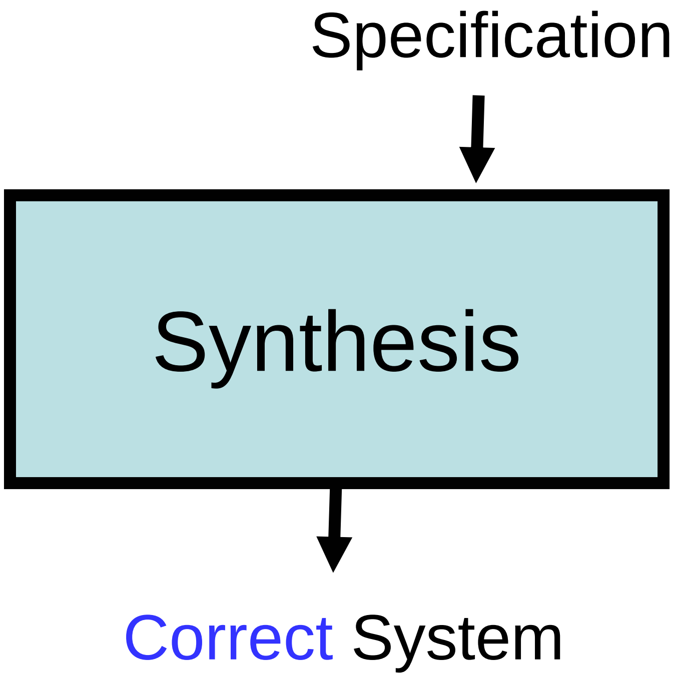
3 Some Promising Directions:

- -Quantitative Synthesis
- -Robust Systems

Boolean Systems Theories



Boolean Systems Theories



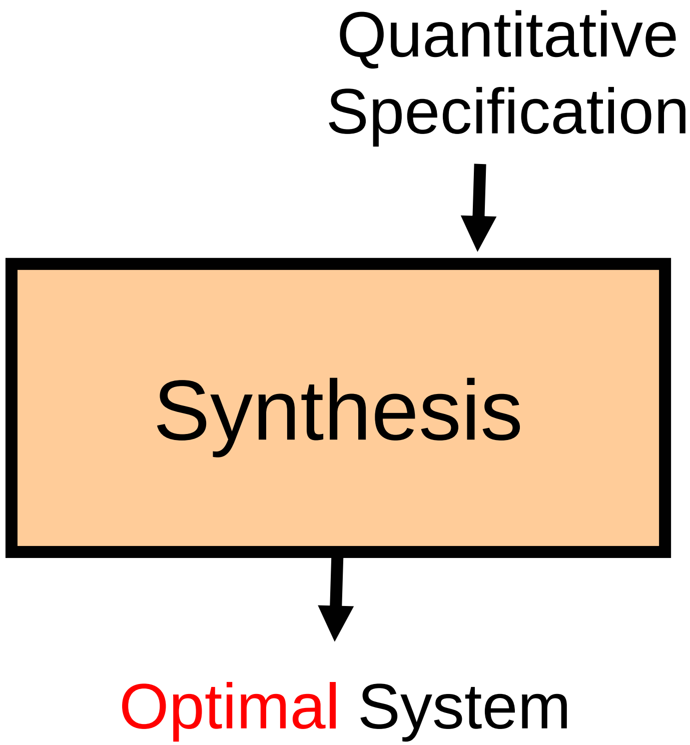
Boolean Systems Theories

ω–Regular
Automaton

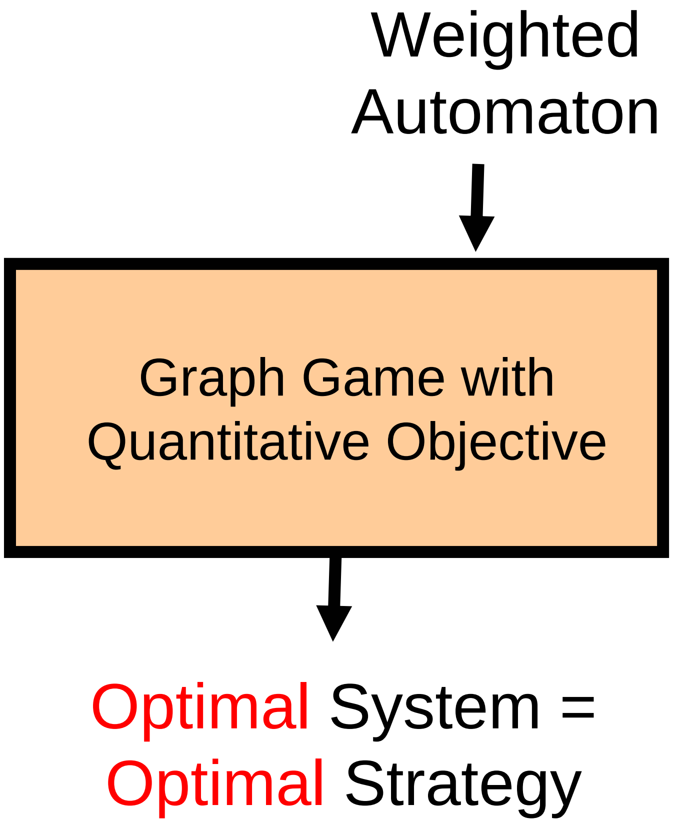
Graph Game with
ω–Regular Objective

Correct System =
Winning Strategy

Quantitative Synthesis



Quantitative Synthesis



Synthesis: From Automata to Games

Automaton states are partitioned into min and max states.

Game: minimizer against maximizer

- -in min states, minimizer chooses successor
- -in max states, maximizer chooses successor

- -minimizer tries to minimize value of a word
- -maximizer tries to maximize value of a word

Scheduler is replaced by two strategies, one for the minimizer and one for the maximizer:

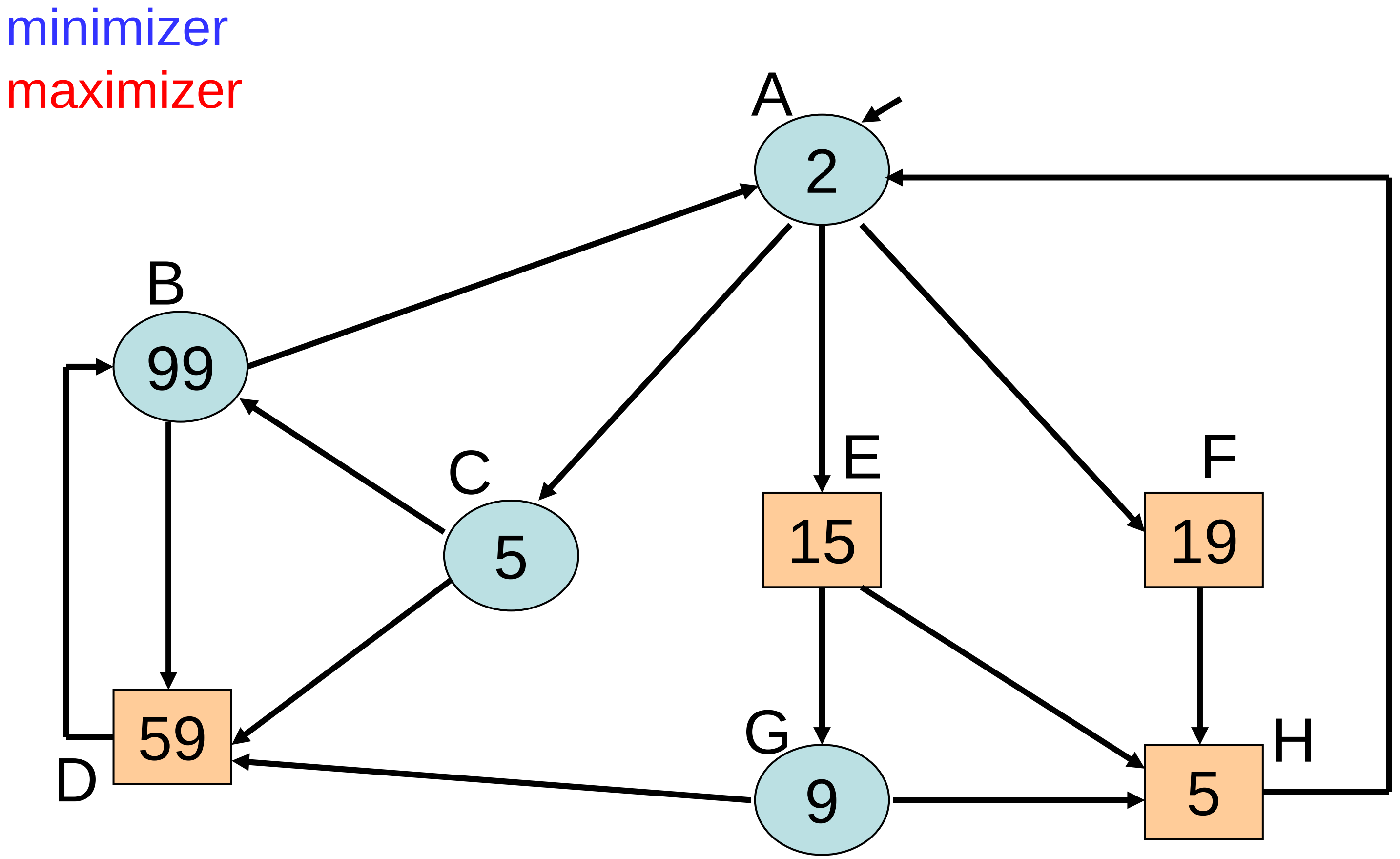
$L(w) = \sup \inf \ldots$

Games for Quantitative Synthesis

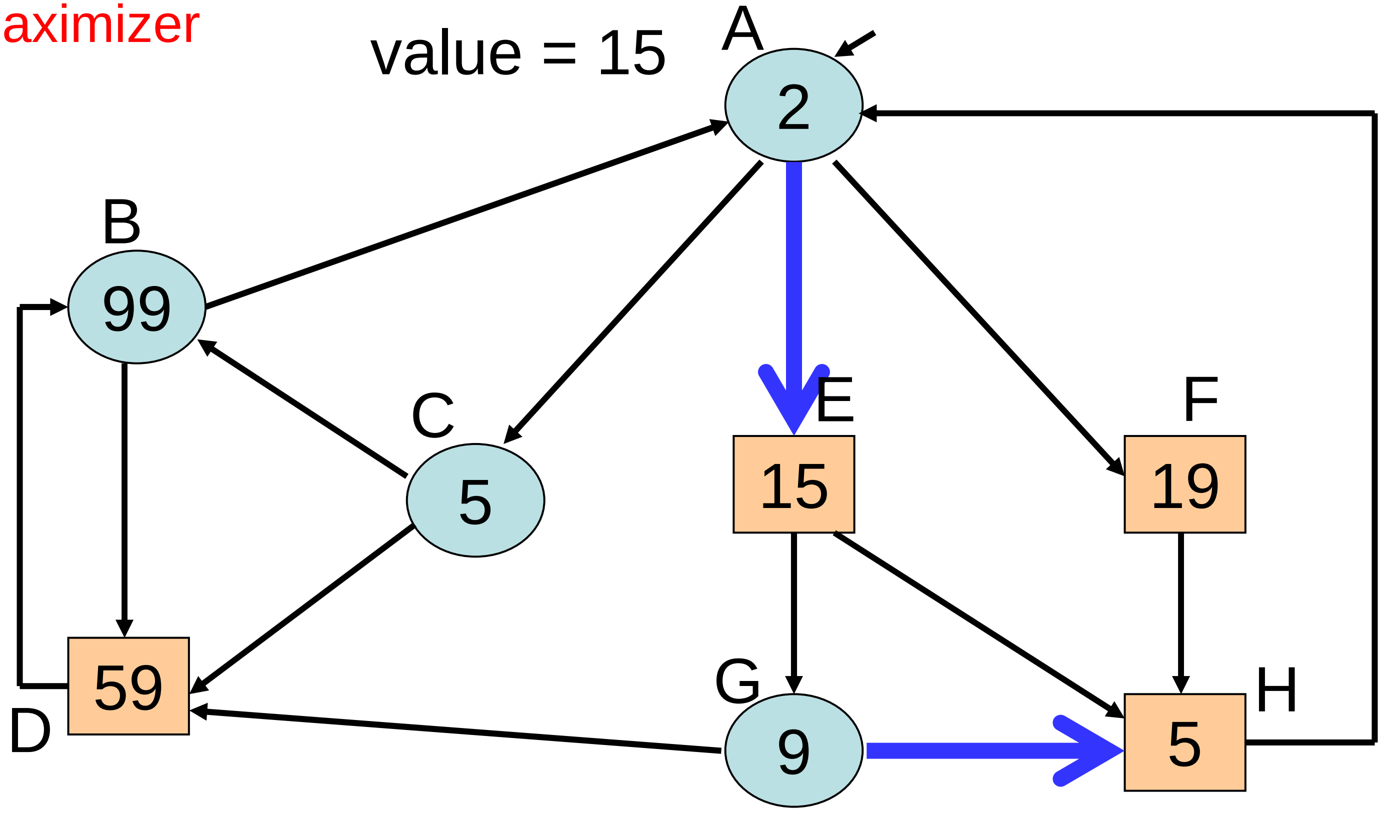
1 Constrained Resources

-every weight is a resource cost (e.g. power consumption)
-optimize peak resource use: max objective
-optimize accumulative resource use: sum objective
[Chakrabarti et al.]

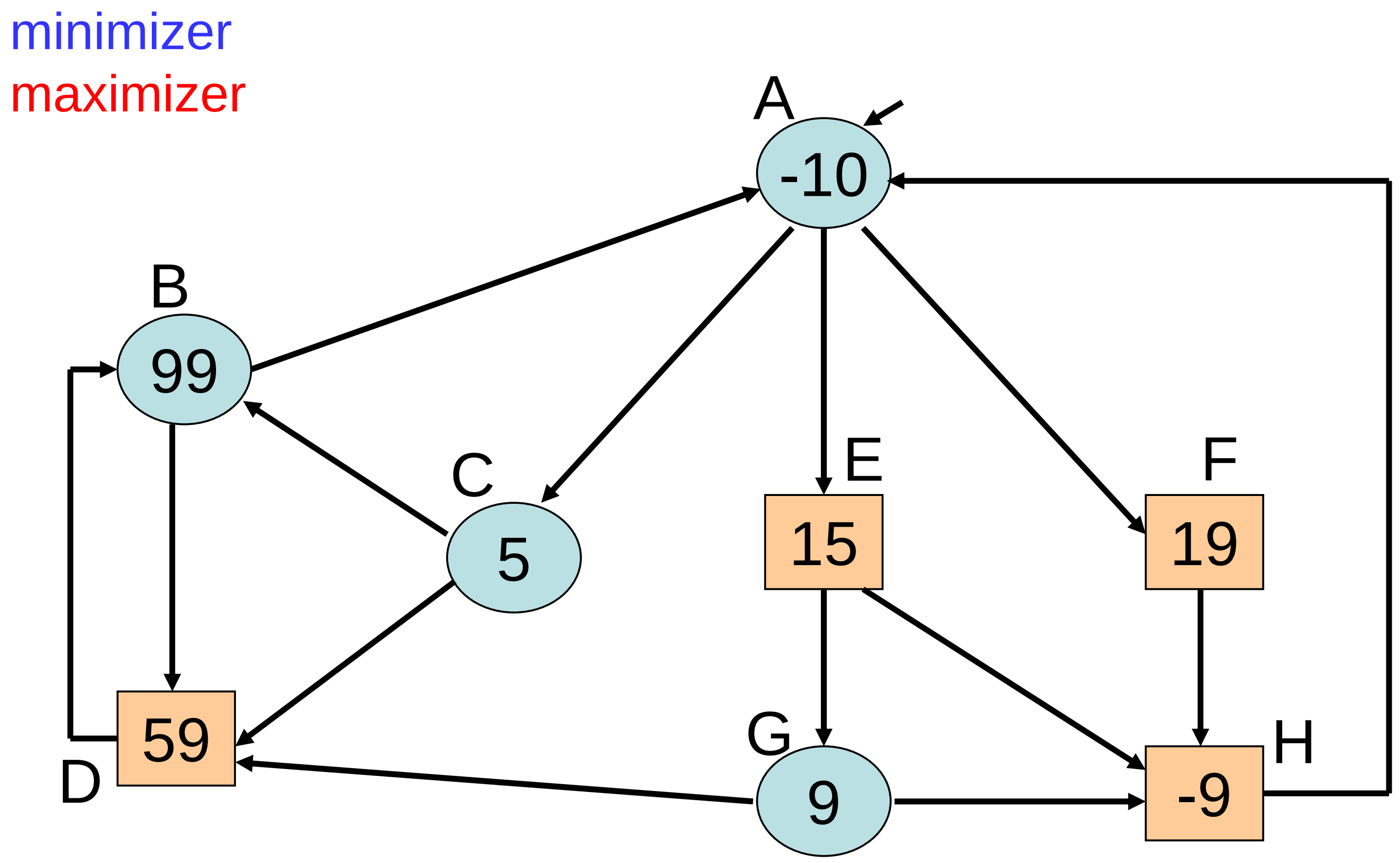
Max Game



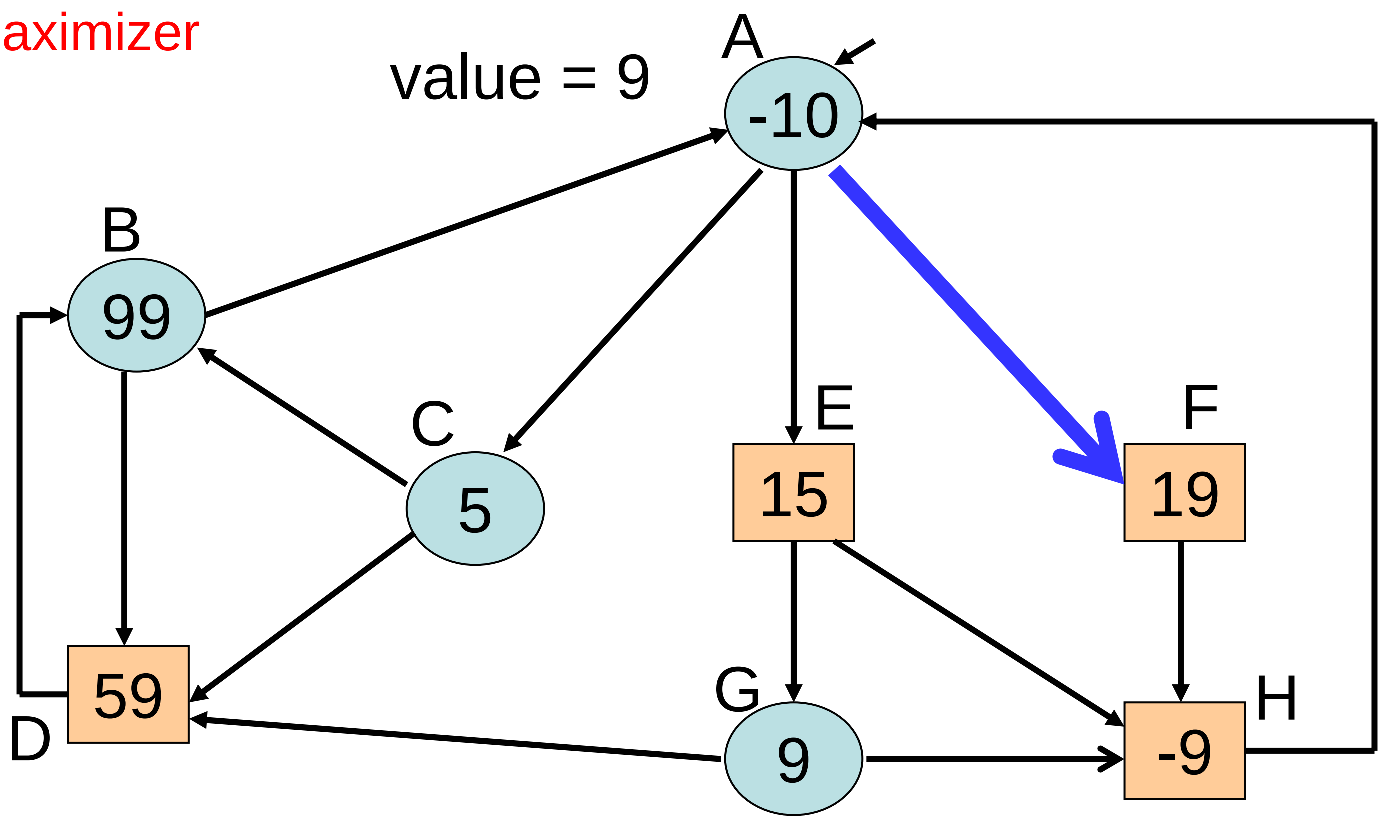
Max Game



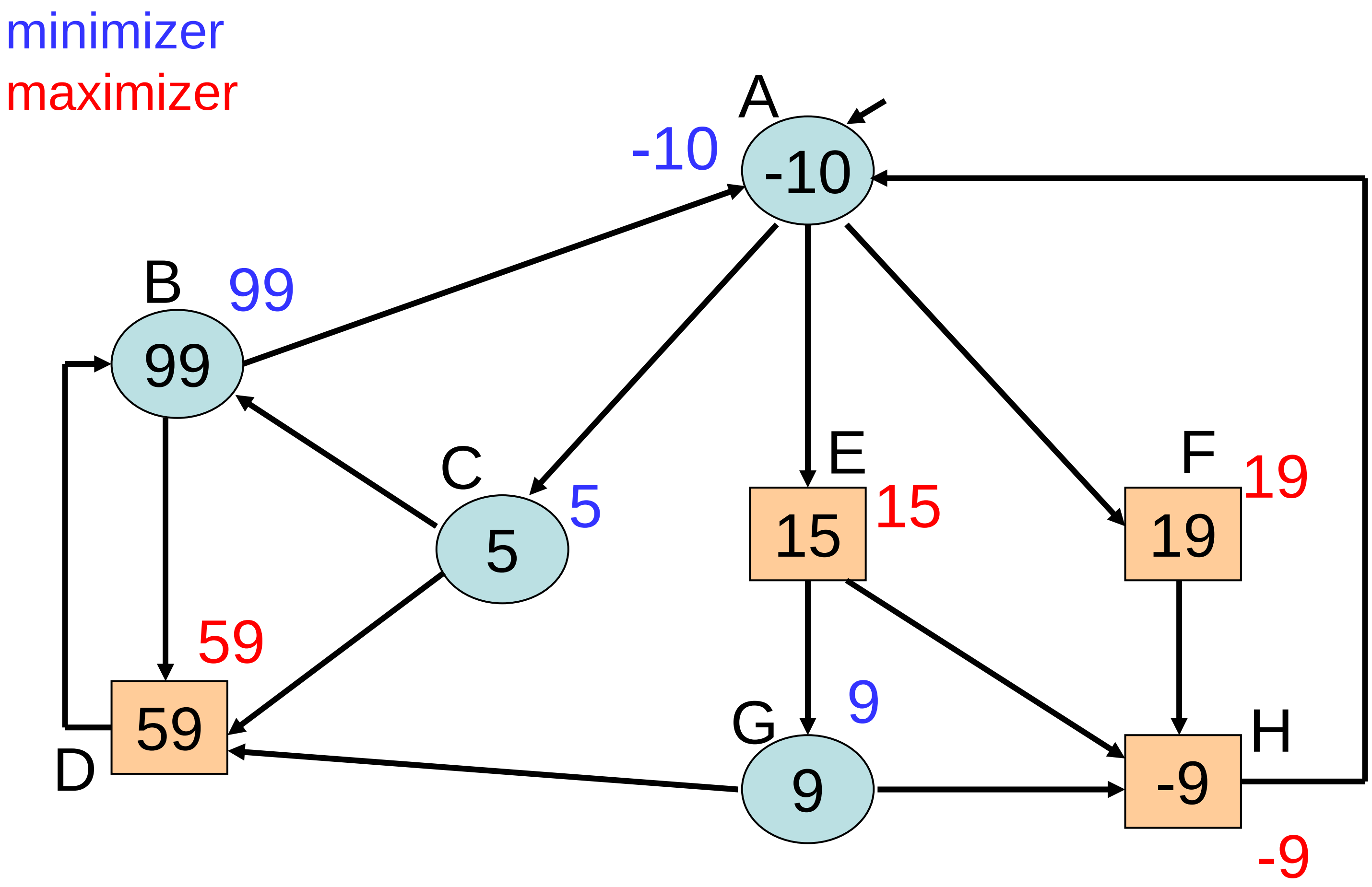
Sum Game



Sum Game

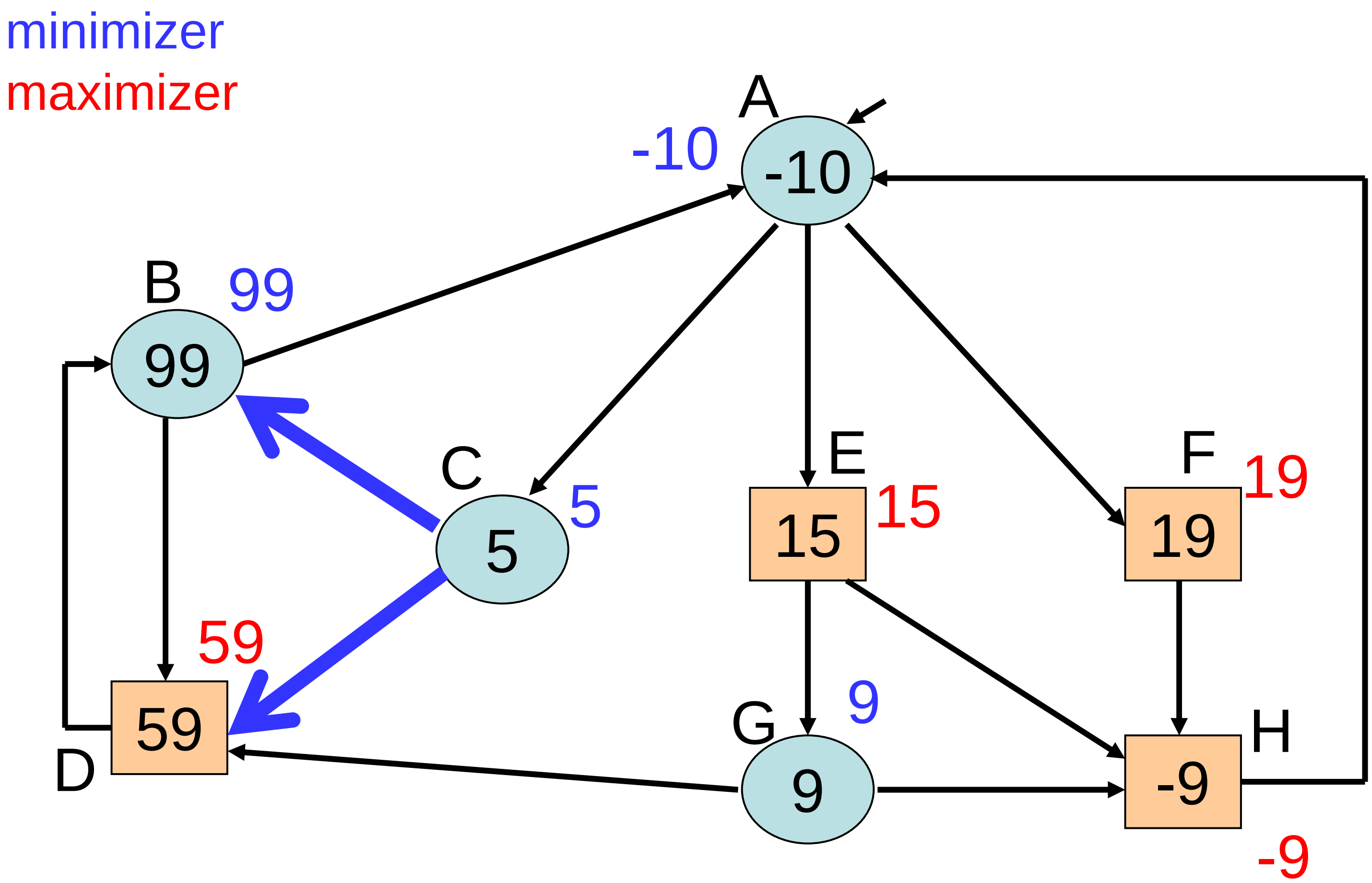


Sum Game



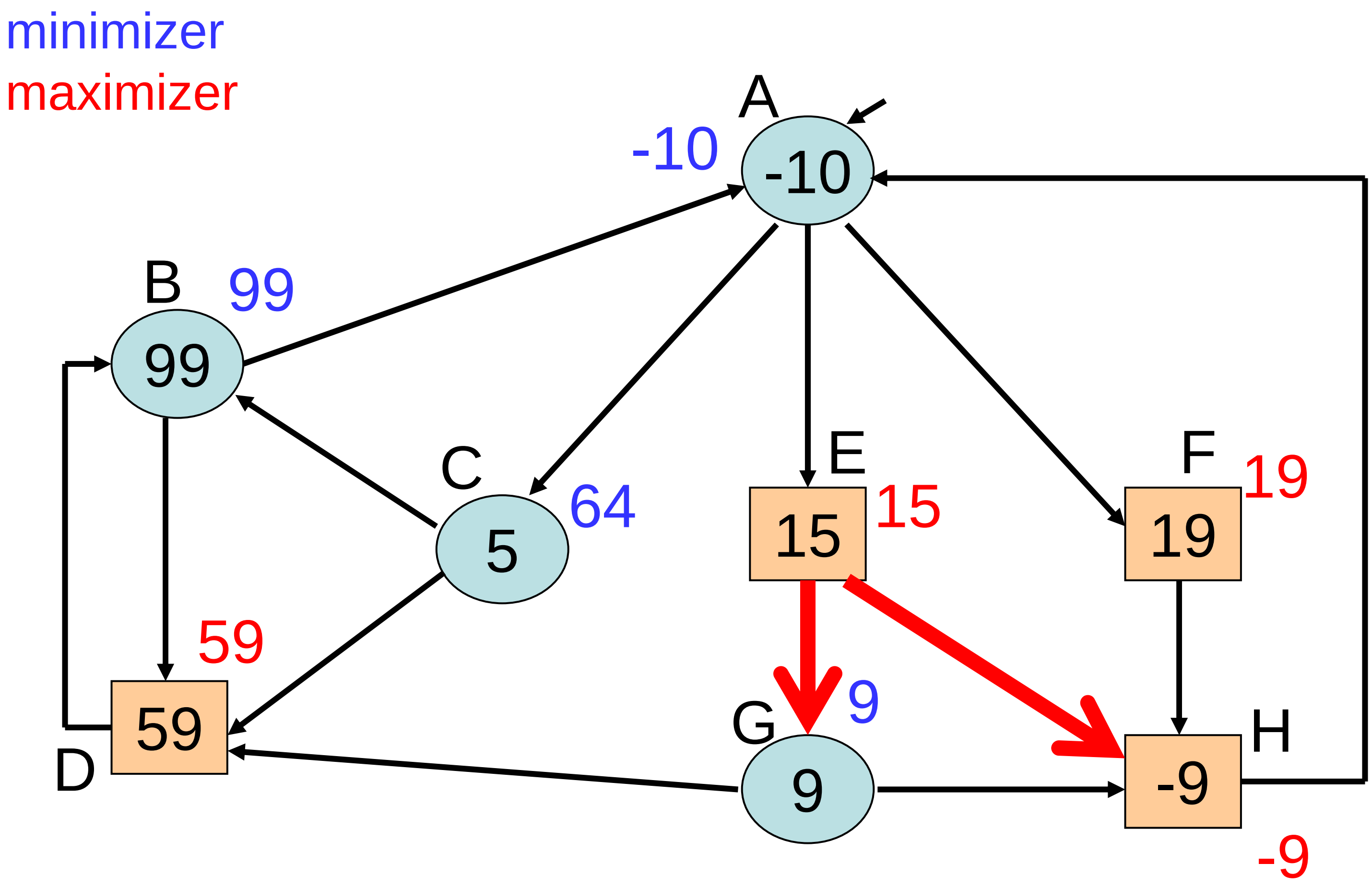
minimizer
maximizer
A
-10
-10
B
99
99
C
5
5
D
59
59
E
15
15
F
19
19
G
9
9
H
-9
-9

Sum Game



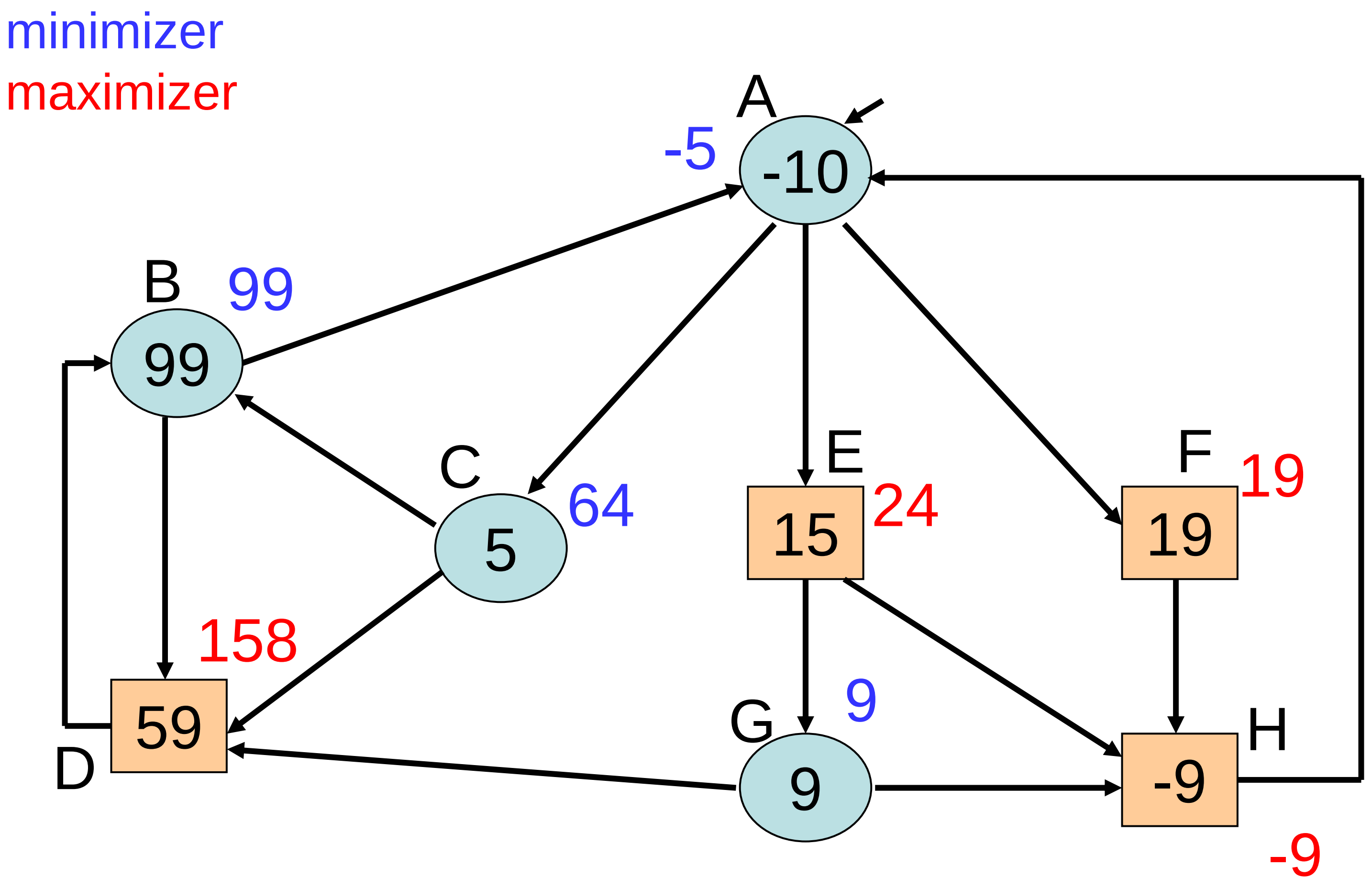
C: 5 + max(0,min(59,99)) = 64

Sum Game



E: 15 + max(0,max(9,-9)) = 24

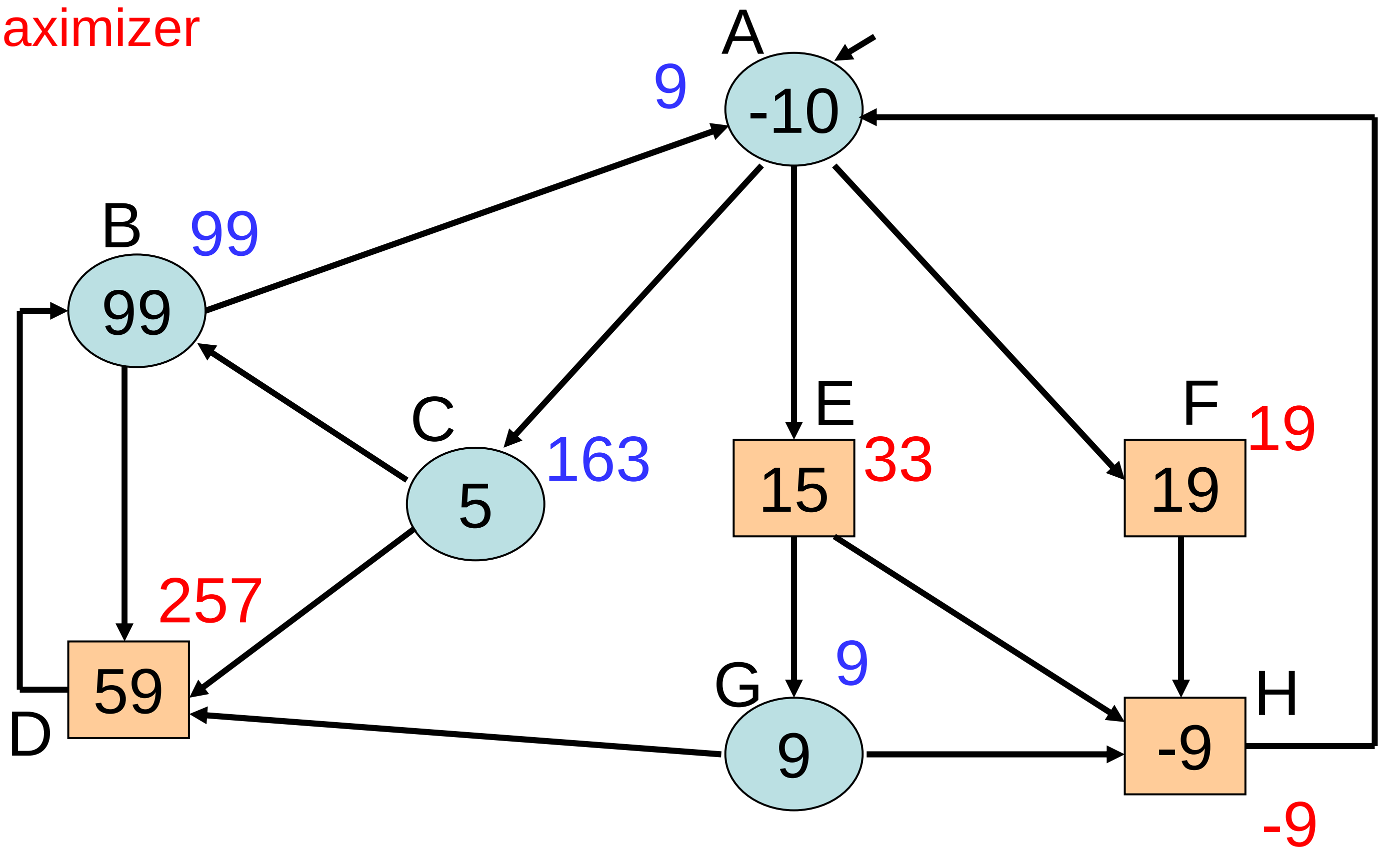
Sum Game



Iteration 1.

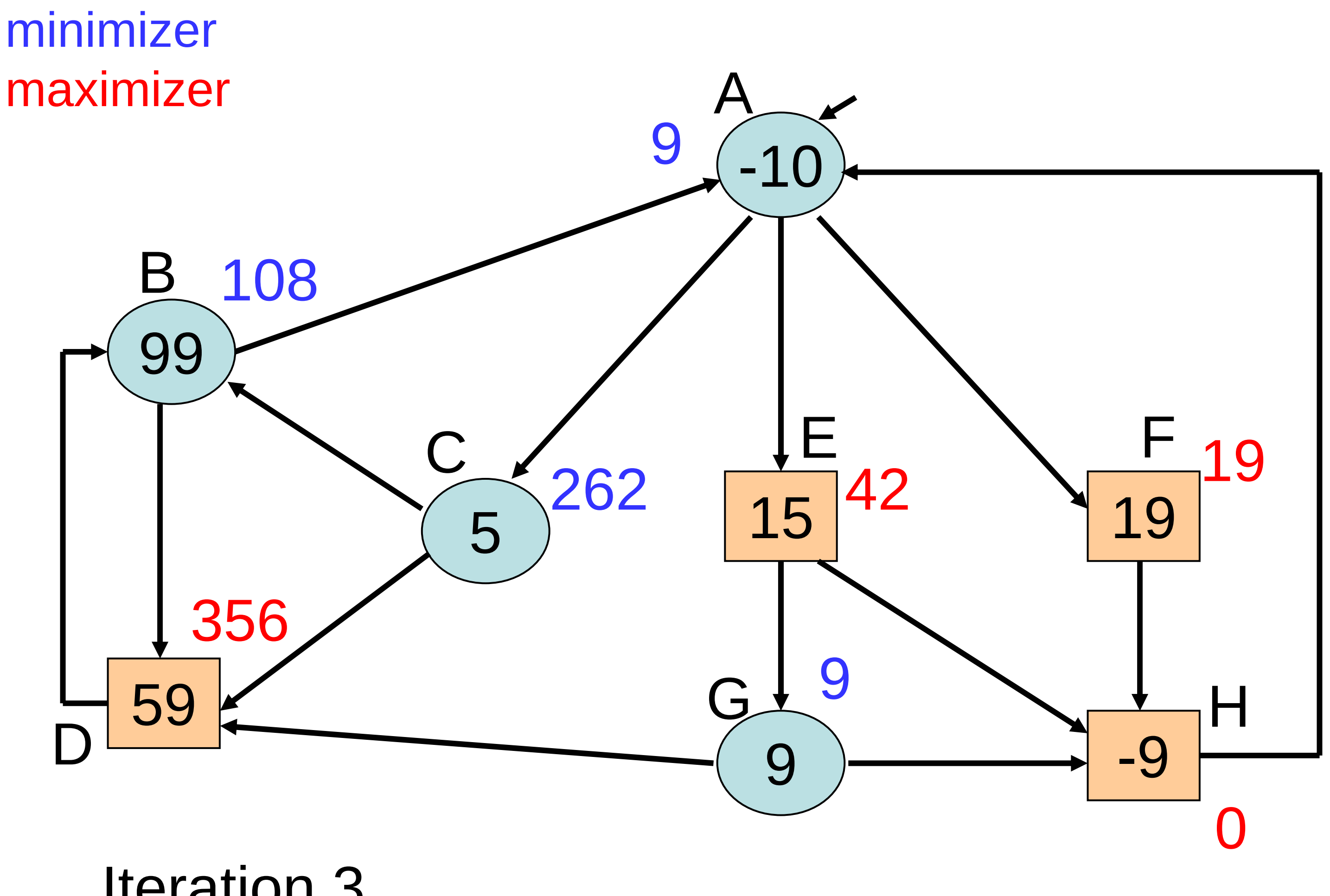
Sum Game

minimizer
maximizer



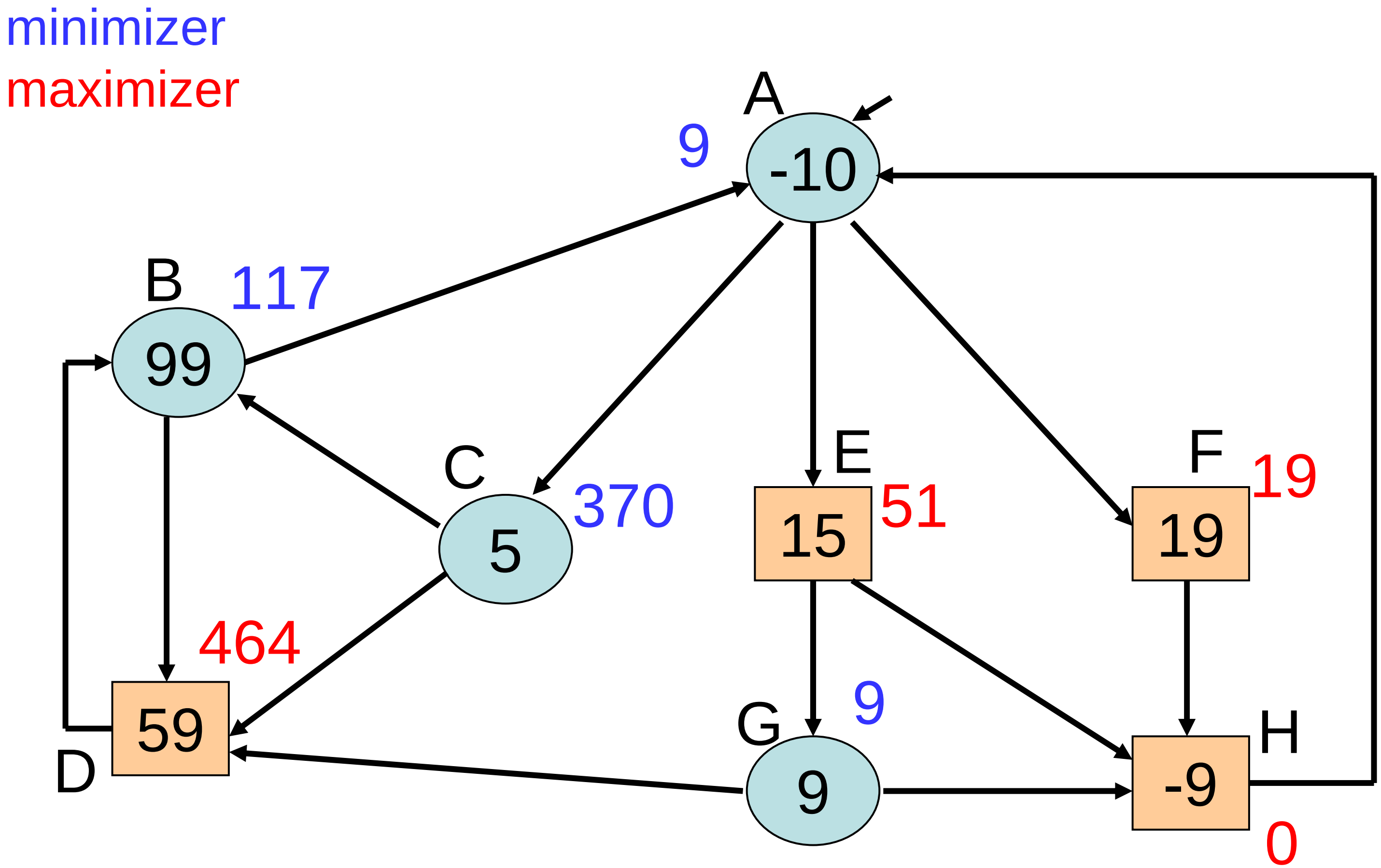
Iteration 2.

Sum Game



Iteration 3.

Sum Game



Iteration 4 = fixpoint.

Games for Quantitative Synthesis

1 Constrained Resources

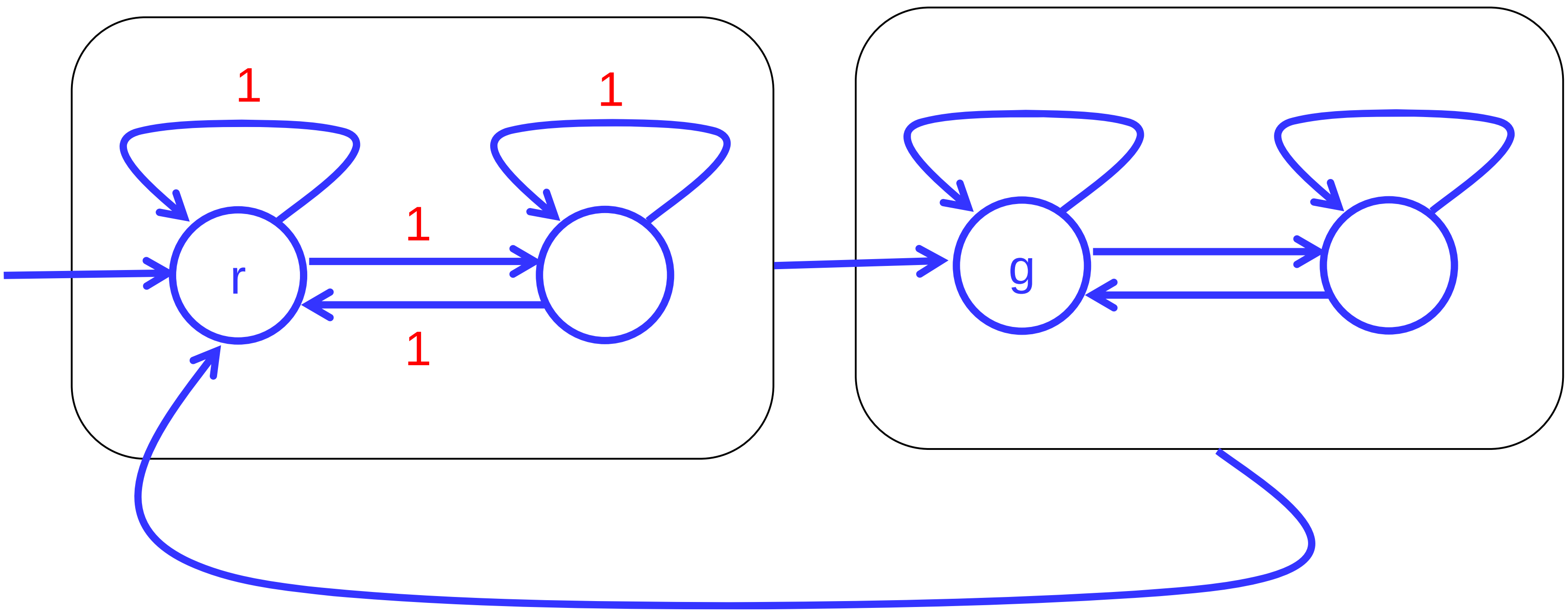
2 Preference between Different Implementations

-boolean spec, but certain implementations preferred
-formalized using lexicographic objectives
[Jobstmann et al.]

$$\langle\, f,\ g_1,\ \ldots\ g_n \,\rangle$$

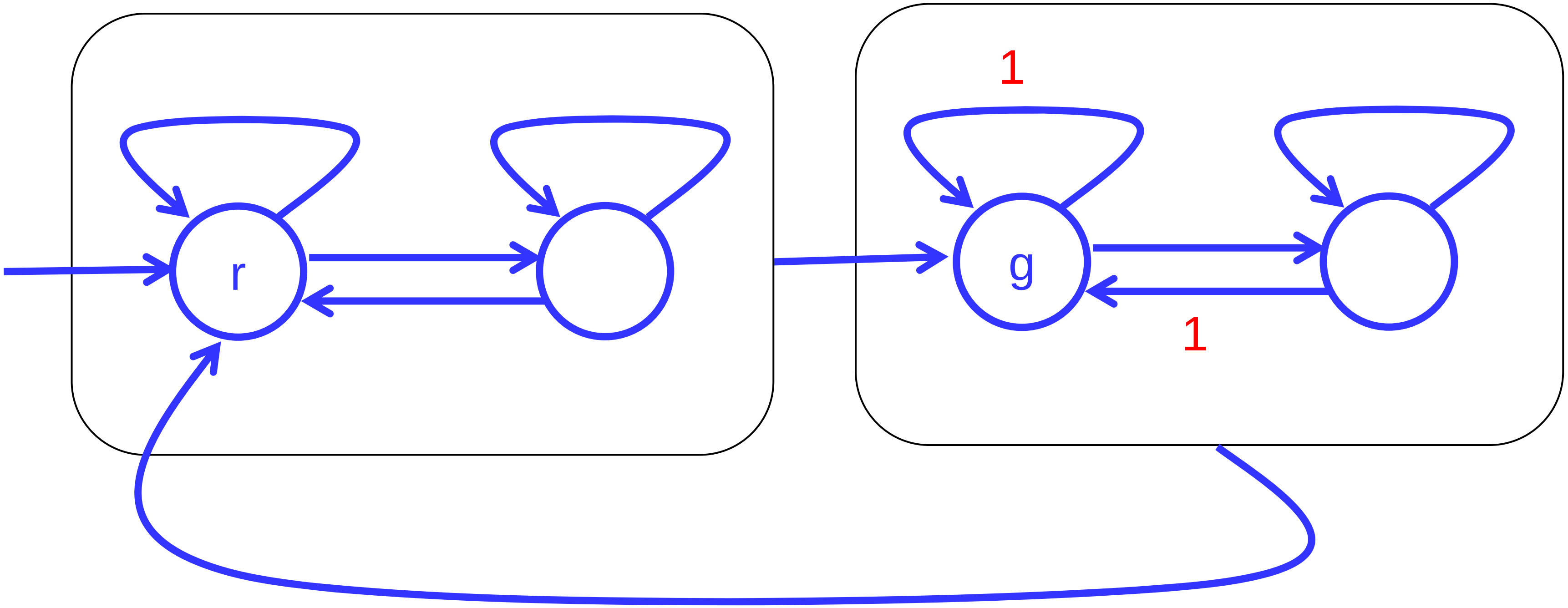
boolean objective   quantitative objectives

Request-Grant Limavg Automaton 1



Following a request, all steps until the next grant are penalized.

Request-Grant Limavg Automaton 2



Following a request, all repeated grants are penalized.

Outline

1 The Quantitative Verification Agenda

2 Some Basic Open Problems

3 Some Promising Directions:

-Quantitative Synthesis
-Robust Systems

Robust Systems

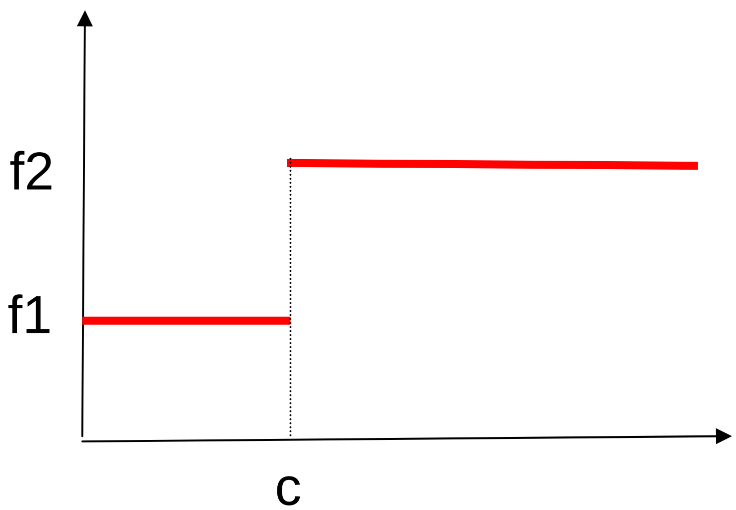
1 Robustness as Mathematical Continuity:

-small input changes should cause small output changes
-only possible in a quantitative framework

$\forall\, \varepsilon>0.\ \exists\, \delta>0.\ \text{input-change} \leq \delta \Rightarrow \text{output-change} \leq \varepsilon$

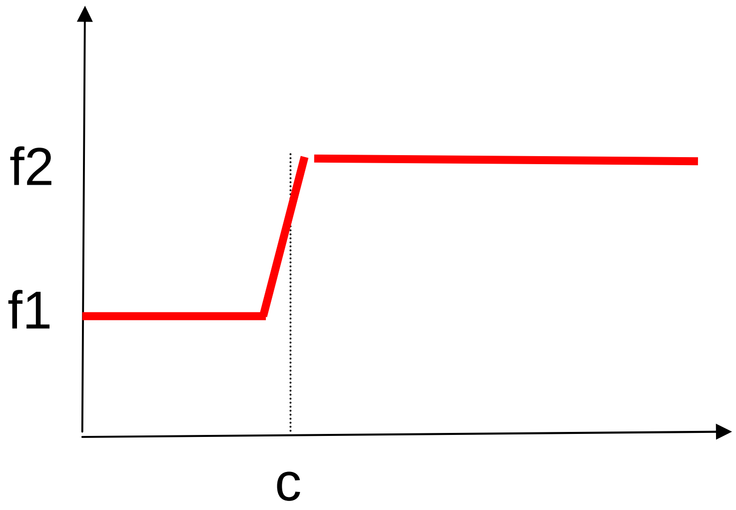
In general programs are not continuous.
But they can less continuous:

read sensor value x;
if x $\leq$ c then y = f1(x)
else y = f2(x);



In general programs are not continuous.
But they can less continuous:

read sensor value x;
if x $\leq$ c then y = f1(x)
else y = f2(x);



Or more continuous:

if x $\leq$ c - ε then y = f1(x);
if x $\geq$ c + ε then y = f2(x)
else y = (f2(c+ε)-f1(c-ε))(x-c+ε)/2ε + f1(c-ε);

[Majumdar et at., Gulwani et al.]

Robust Systems

1 Robustness as Mathematical Continuity:

-small input changes should cause small output changes
-only possible in a quantitative framework

$\forall\ \varepsilon>0.\ \exists\ \delta>0.$ input-change $\leq \delta \Rightarrow$ output-change $\leq \varepsilon$

Example of a Robustness Theorem [AHM]:

If discountedBisimilarity(A,B) > 1 - ε,
then $\forall$w : |A(w) – B(w)| < f(ε).

Robust Systems

1 Robustness as Mathematical Continuity:

-small input changes should cause small output changes
-only possible in a quantitative framework

2 Robustness w.r.t. Faulty Assumptions:

-environment may violate assumptions
-few environment mistakes should cause few system mistakes
-ratio of system to environment mistakes as quantitative
quality measure

[Greimel et al.]

Conclusions

-“Quantitative” is more than “timed” and “probabilistic.”

-Weighted automata offer a natural quantitative specification language.

-We need to move from boolean correctness criteria to quantitative system preference metrics.

-We have interesting point solutions, but no convincing overall framework.