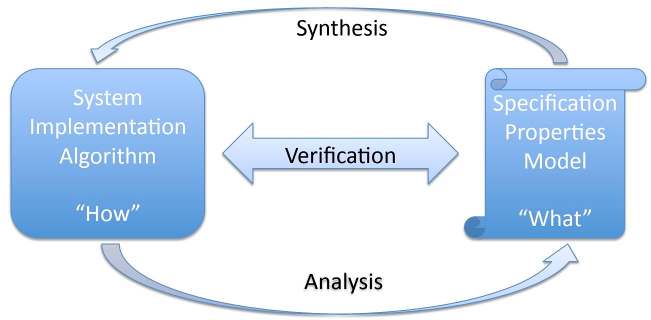
Quantitative Verification and Synthesis

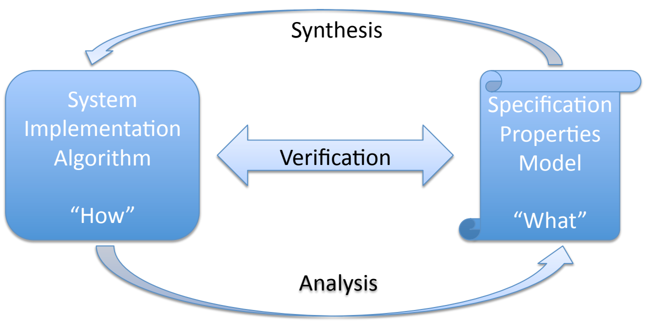
Barbara Jobstmann
CNRS-Verimag, Grenoble, France

Joint work with Roderick Bloem, Krishnendu Chatterjee, Tom Henzinger, and Rohit Singh.

Analysis – Verification – Synthesis



Analysis – Verification – Synthesis



Reactive Systems (e.g. servers, hardware,…)
- Interact with environment
- Non-terminating
- Finite data (or data abstractions)
- Control-oriented

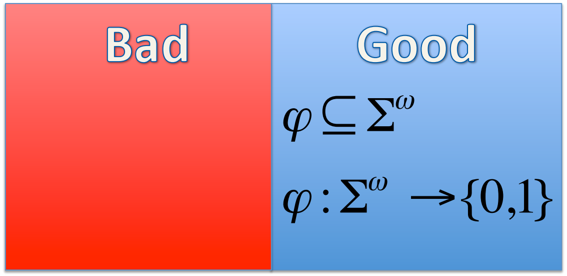
Specifications
- Set of good behaviors = language
- Temporal logic (LTL)

Resource Controller

1) always(not(g1 and g2))
2) always(r1 implies eventually(g1))
3) always(r2 implies eventually(g2))

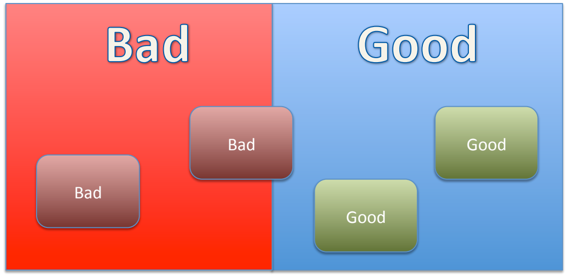
Classical Specifications

Σ ...alphabet
Σ^ω...set of all words/behaviors



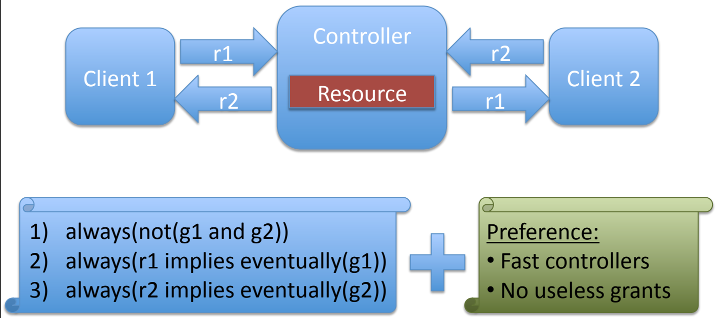
Classical Specifications

Σ ...alphabet
Σ^ω...set of all words/behaviors



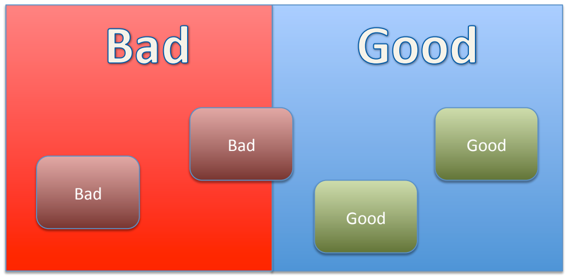
S satisfies φ iff $\forall w \in L(S) : \varphi(w) = 1$

Resource Controller



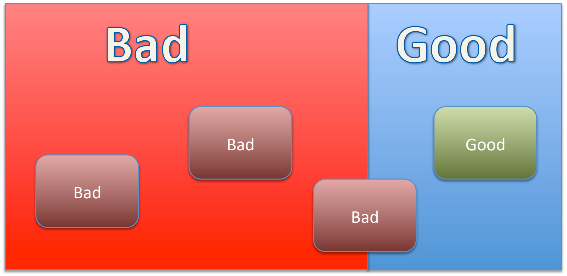
Preference?

- Write more properties



Preference?

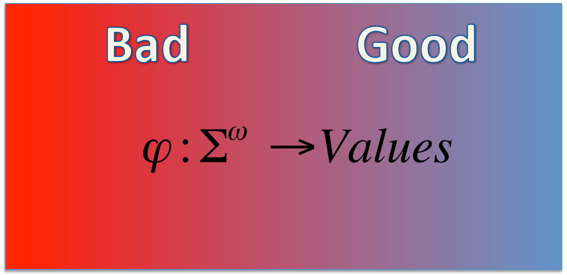
- Write more properties



1) Right properties might be hard to find
2) Properties usually difficult to express (soft constraints)
3) Properties can over-constrain system

Quantitative Specifications

Σ ...alphabet
Σ^{ω}...set of all words/behaviors



Quantitative Specifications

Σ ...alphabet
Σ^{ω}...set of all words/behaviors

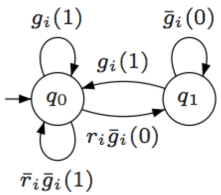
S better than S' iff value(S) ≥ value(S')

Two Main Questions

1. How to assign a value to a word?
2. How to assign a value to a system?

1. Value of a Word/Behavior

- Idea:
  - use weighted automata
  - give rewards for good behavior
- Example:
  - prefer fast reaction to request of client i



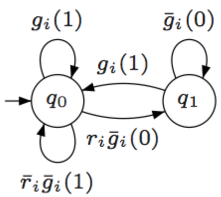
1. Value of a Word/Behavior

- Weighted automaton A defining rewards
- Value over a behavior
  - Min/max
  - **Average reward**: Mean-payoff value
    Given word w, let s_0,s_1,s_2… be the run of A on w

$$MP(w) = \lim_{n \to \infty} \sum_{i=0}^{n-1} \frac{r(s_i, s_{i+1})}{n}$$

$$value_A(w) := MP(w)$$

1. Value of a Word/Behavior



$w_1 = (rg \quad \bar{r}\bar{g} \quad rg)^\omega \qquad (111)^\omega \qquad value(w_1) = 1$

$w_2 = (r\bar{g} \quad \bar{r}\bar{g} \quad \bar{r}g)^\omega \qquad (001)^\omega \qquad value(w_2) = \frac{1}{3}$

$w_3 = (r\bar{g} \quad r\bar{g} \quad r\bar{g})^\omega \qquad (000)^\omega \qquad value(w_3) = 0$

Mean-payoff automaton/specification

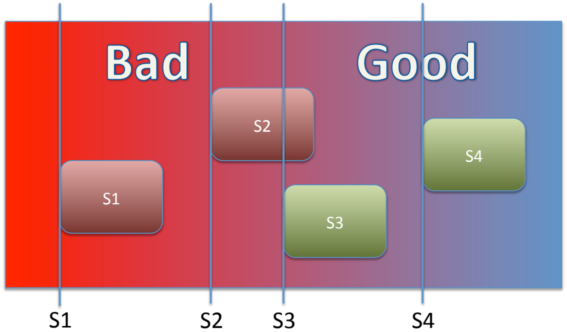
2. Value of a System

- Min/Max:
  Given a system S with set of behaviors L(S)

$$value(S) = \min_{w \in L(S)} \varphi(w)$$

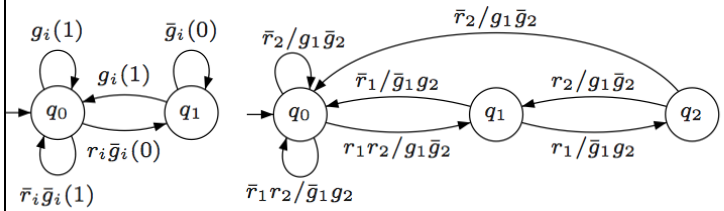
2. Value of a System

$$value(S) = \min_{w \in L(S)} \varphi(w)$$

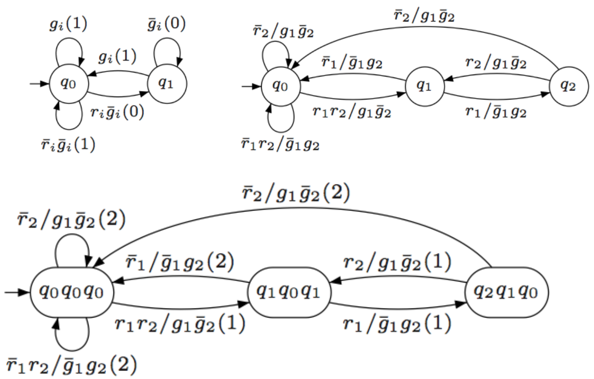


How to Compute Value of System?

- Given a mean-payoff automaton A and a reactive system S, compute value(S)

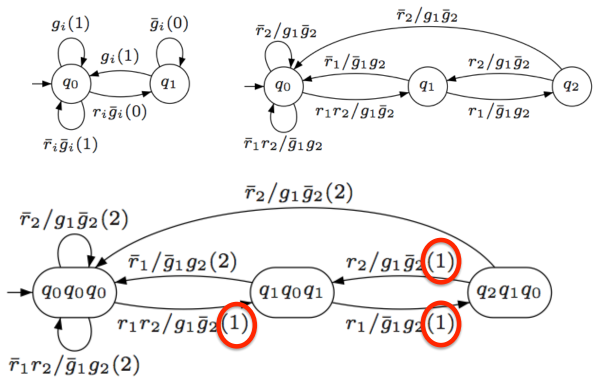


System × Specification



$$value(S) = \min_{w \in L(S)} \varphi(w)$$

System × Specification

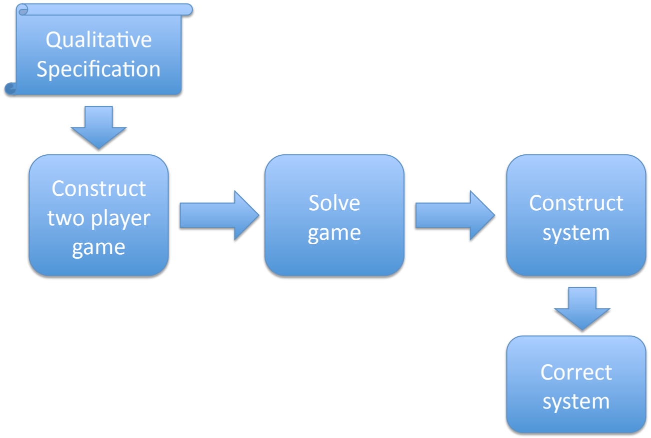


$$value(S) = \min_{w \in L(S)} \varphi(w)$$

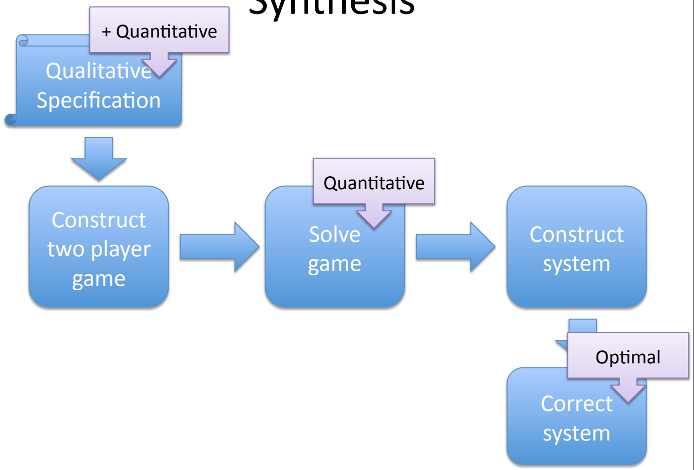
How to Construct Optimal System?

- Given a qualitative specification φ and a quantitative specification ψ, construct a reactive system S that
  (i) satisfies φ and
  (ii) optimizes ψ

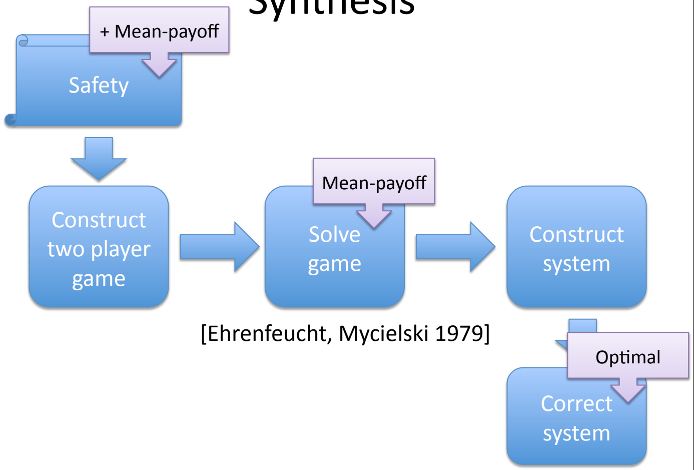
Synthesis of Reactive Systems



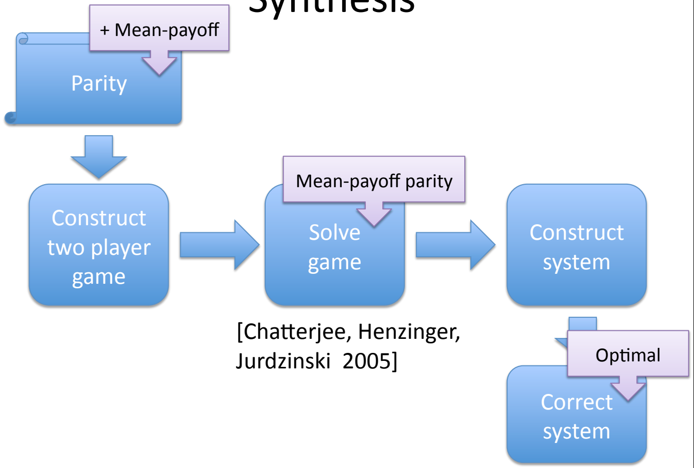
Synthesis



Synthesis



Synthesis

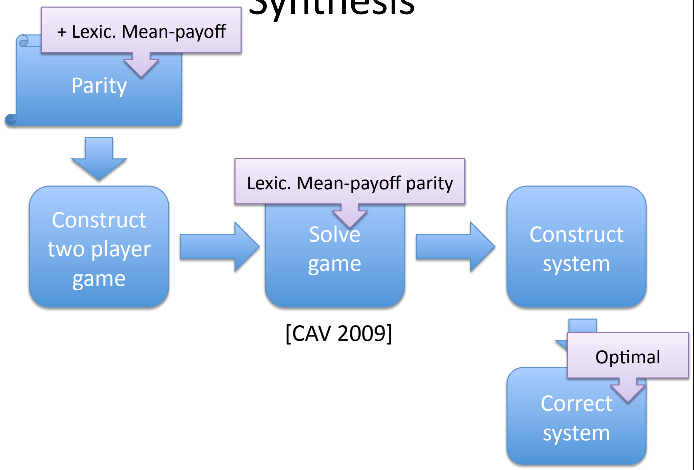


Lexicographic Extension

- Combining quantitative specifications
- Weighted automata with tuples:
  - Natural numbers $value_A : \Sigma^\omega \to R$
  - Vectors $lvalue_A : \Sigma^\omega \to R^d$

$$LMP(w) = \lim_{n \to \infty} \sum_{i=0}^{n-1} \frac{\vec{r}(s_i, s_{i+1})}{n}$$

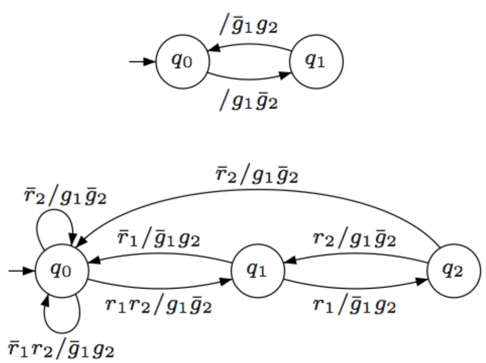
Synthesis



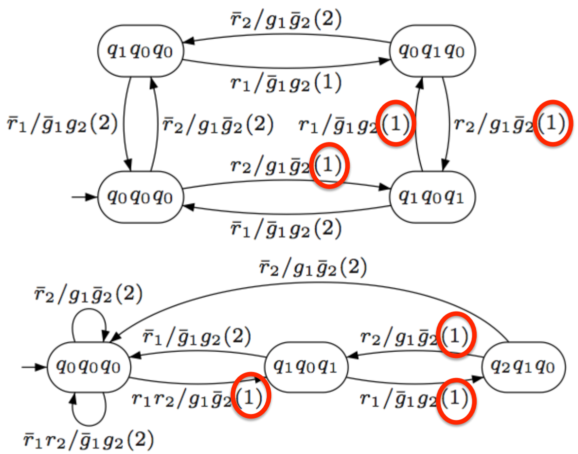
Value of System (revisited)

- Recall, resource controller
- Aim: fast response, $value_{A1}$ + $value_{A2}$
- Worse-case behavior?
- Worse-case input? $(r_1 r_2)^\omega$
- Best response? $(g_1 \bar{g}_2 \quad \bar{g}_1 g_2)^\omega$

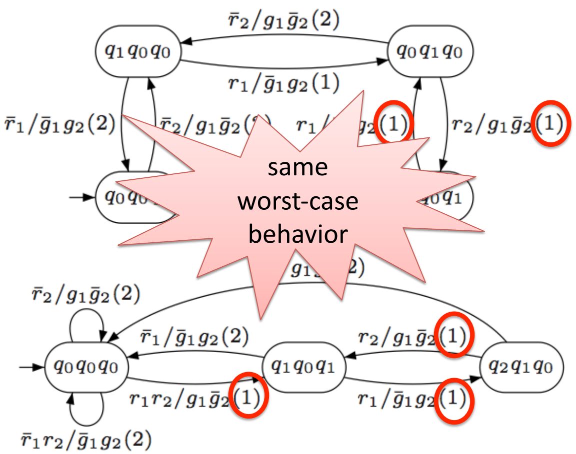
Resource Controllers



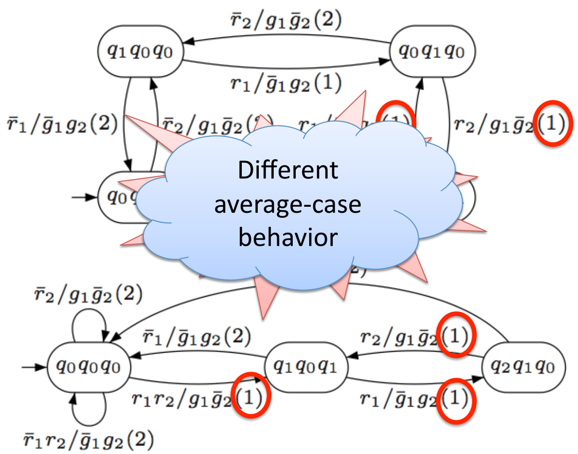
System × Specification



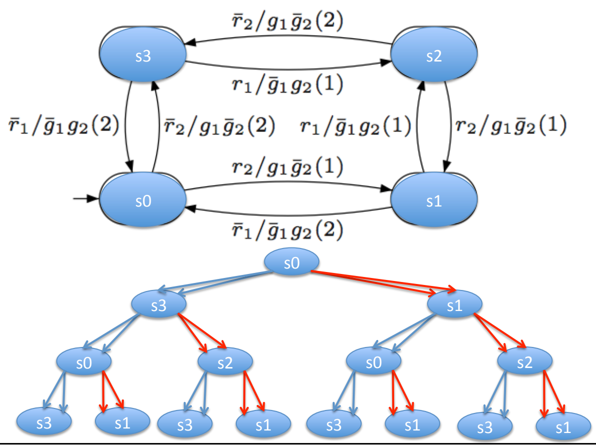
System × Specification



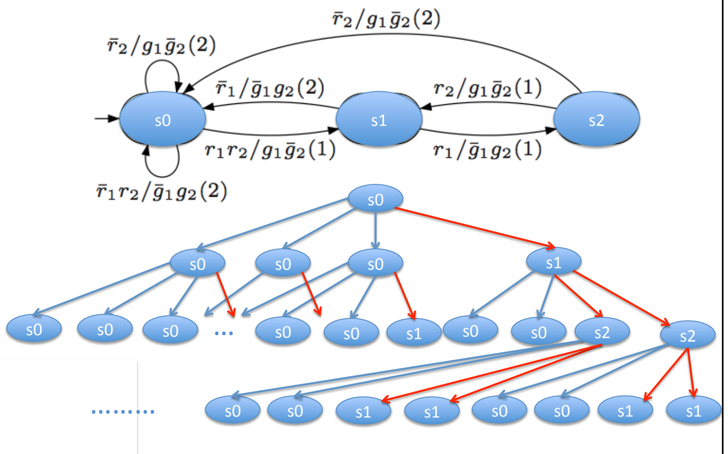
System × Specification



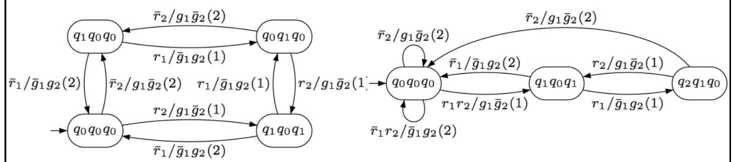
System × Specification



System × Specification



Average Over Input Sequences

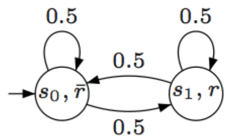


- In all states "equally frequently visited"
- Reward in every state: ½ · 1 + ½ · 2 = 1.5
- Average behavior: (4 · 1.5) / 4 = 1.5

- s0 is visited more often s0: 2/3, s1: 2/9, s2: 1/9
- Rewards:
  s0: ¾ · 2 + ¼ · 1 = 1.75
  s1,s2: ½ · 1 + ½ · 2 = 1.5
- Average behavior: 1.67

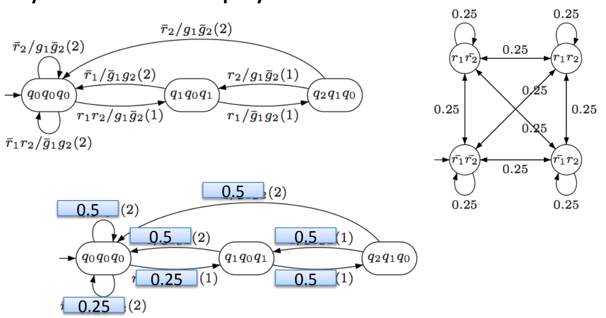
Markov Chain – Input Distribution

- How likely are different input values?



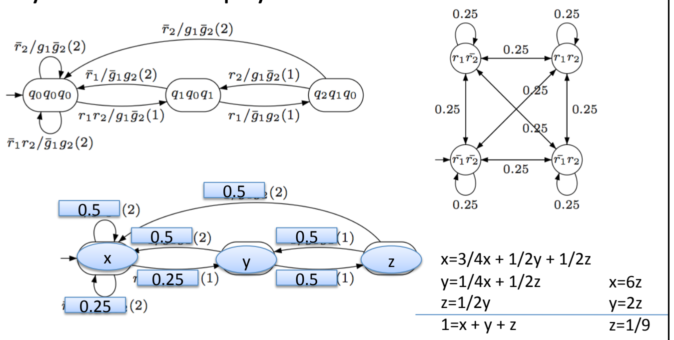
Compute Value wrt Probabilistic Environ. Assumption

- System × mean-payoff × Markov chain



Compute Value wrt Probabilistic Environ. Assumption

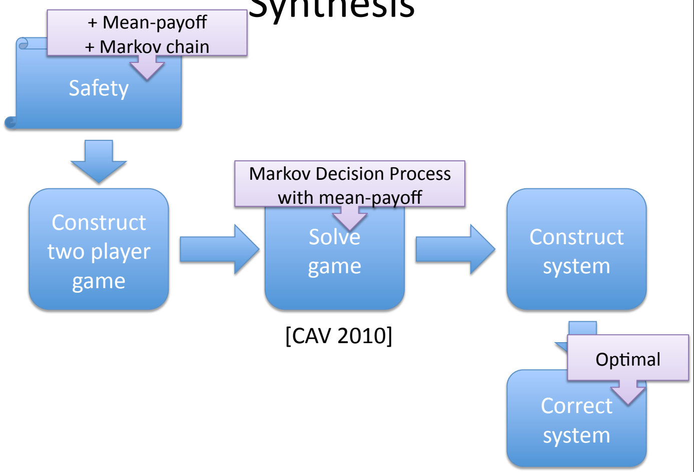
- System × mean-payoff × Markov chain



How to Construct Optimal System?

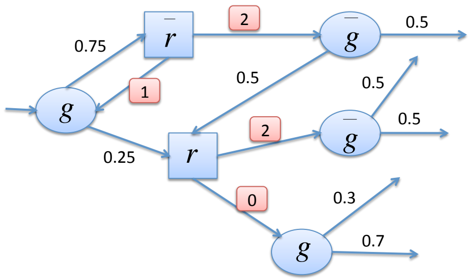
- Given a qualitative specification φ and a quantitative specification ψ and, a probabilistic environment assumption μ construct a reactive system S that (i) satisfies φ with probability 1 under μ (ii) optimizes ψ under μ

Synthesis

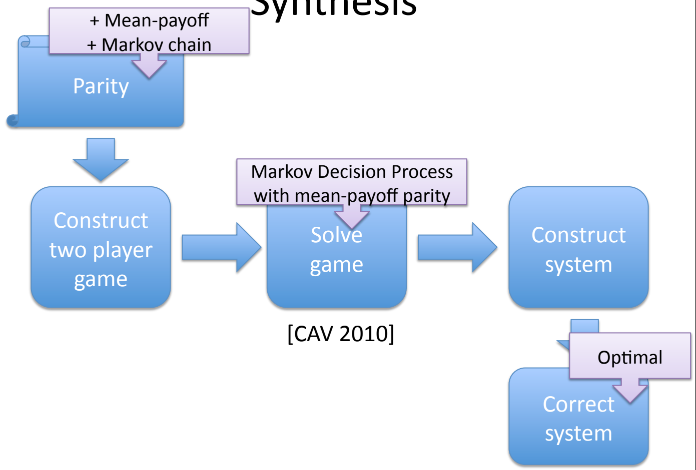


Markov Decision Process

- Safety × mean-payoff × Markov chain = MDP

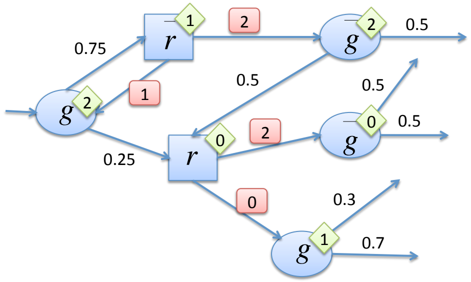


Synthesis



Markov Decision Process

- Safety × mean-payoff × Markov chain = MDP



MDPs with mean-payoff parity

- Polynomial-time algorithm [CAV 2010]
- End-components: set of states that is strongly-connected and closed for probabilistic player
- Key observation:
  MP-parity value in an end component with even minimal priority is equal to MP value
  - Computes maximal end components with even minimal priority
  - Fix value for these states to corresponding MP value
  - Compute way to best end component

Summary

- Quantities are good for verification & synthesis
  - to rank implementations wrt preference
  - to state "soft" properties
- Value of word
  - Mean-payoff automata (weighted automata that average over weights)
  - Lexicographic extension
- Value of system
  - Min/Max value over words: lexicographic MP-parity
  - Average value over words: MDP with MP-parity