

Quantitative μ -calculus and CTL based on constraint semirings

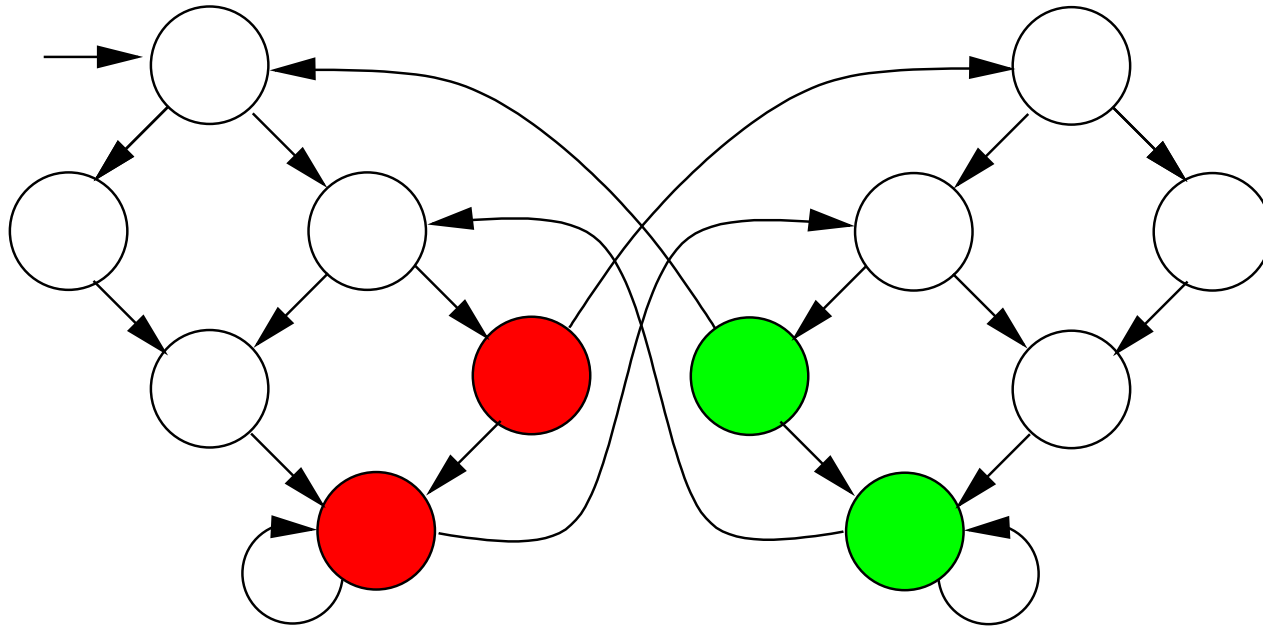
2nd Workshop on Quantitative Aspects of Programming Languages
27-28. March 2004

Alberto Lluch Lafuente, Ugo Montanari
Dipartimento di Informatica, Università di Pisa

Introduction

- *Classical* Model Checking: reason about **qualitative** (boolean) aspects of systems.
- *Quantitative* Model Checking: reason about **quantitative** aspects of systems.
 - ⇒ Examples: probabilistic, discounted, durational.
- *Classical* CSP: Problems with **boolean** constraints.
- *Soft* CSP: Problems with **soft** constraints.
 - ⇒ Examples: probabilistic, fuzzy, weighted, multi-valued, etc.
- C-semirings as framework for Soft CSP
 - ⇒ C-semirings as framework for Quantitative Model Checking?

Mutual Exclusion: Boolean Reasoning



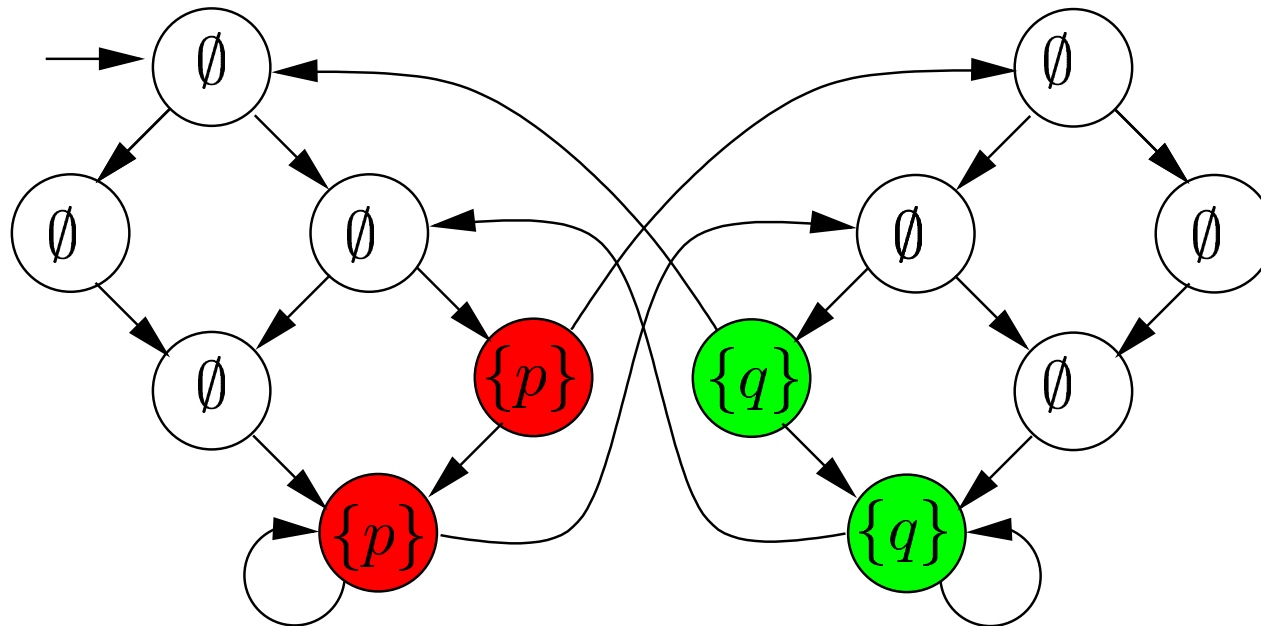
- Can process p reach the c.s.?

$$\llbracket \mathbf{EF} p_in_cs \rrbracket (s_0) = true$$

- Can process p reach the c.s. and stay forever?

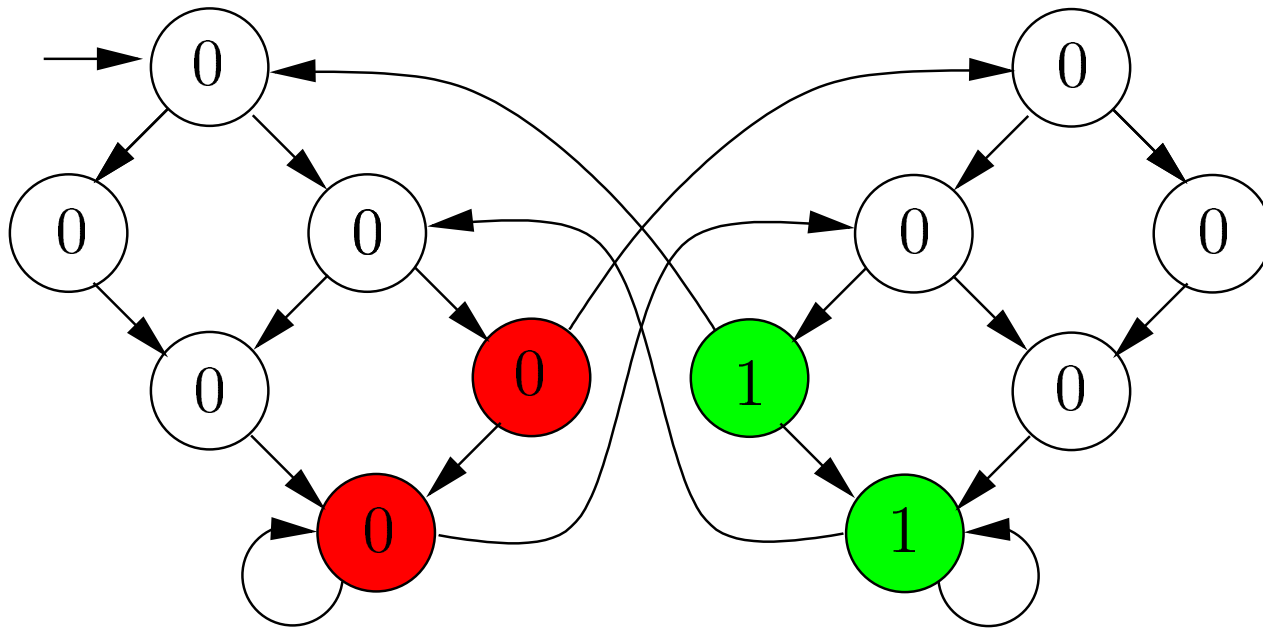
$$\llbracket \mathbf{EFEG} p_in_cs \rrbracket (s_0) = true$$

Mutual Exclusion: Reasoning about Processes



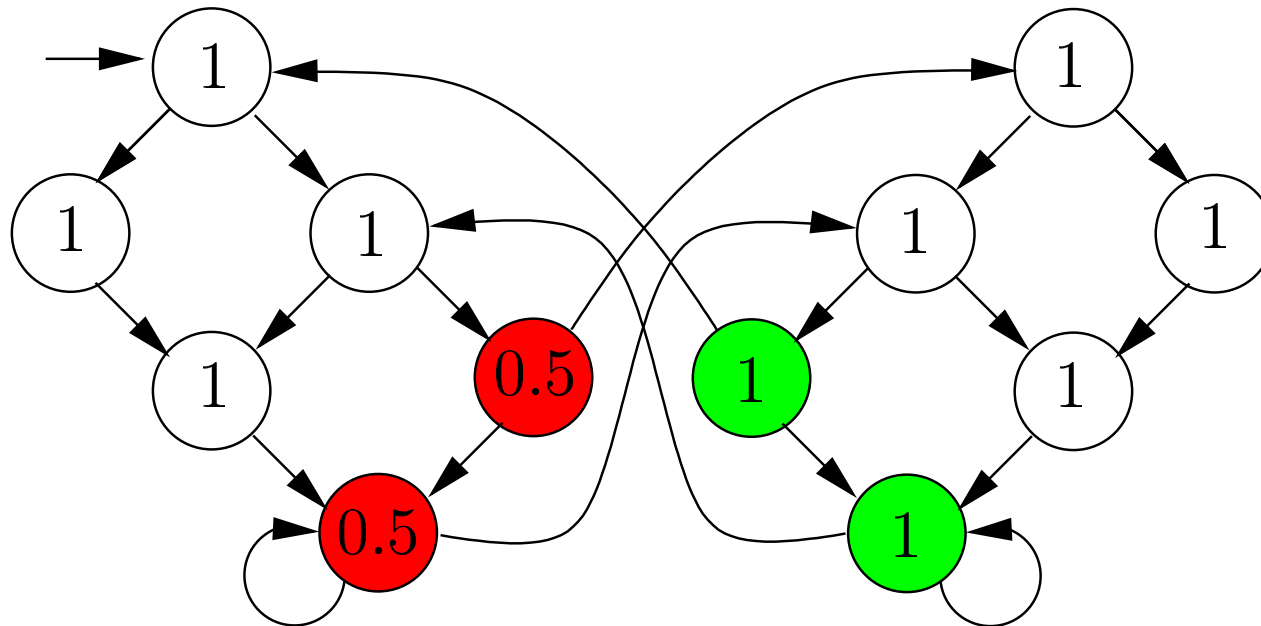
- Which processes can reach the c.s.?
- Which processes can reach the c.s. and stay forever?

Mutual Exclusion: Reasoning about Costs



- Min. cost for a process to use the c.s.?
- Min. cost for a process to use the c.s. forever?

Mutual Exclusion: Reasoning about Probabilities



- Max. probability for a process to use the c.s.?
- Max. probability for a process to use the c.s. forever?

C-Semirings (Definition)

A c-semiring is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ such that:

- A is a (possibly infinite) set;
- $\mathbf{0}$ and $\mathbf{1}$ are elements of A ;
- $+$: $A \times A \rightarrow A$
 - associative, commutative and idempotent;
 - $a + \mathbf{0} = a$ and $a + \mathbf{1} = \mathbf{1}$.
- \times : $A \times A \rightarrow A$
 - associative and commutative;
 - distributes over $+$;
 - $a \times \mathbf{0} = \mathbf{0}$ and $a \times \mathbf{1} = a$.

C-Semirings (Instances and Application to QoS)

- Boolean semiring: $\langle \{true, false\}, \vee, \wedge, false, true \rangle$
 \Rightarrow Service/link availability, etc.
- Optimization semiring: $\langle \mathbb{R}^+, min, +, +\infty, 0 \rangle$
 \Rightarrow Bandwidth, price, etc.
- Probabilistic c-semiring: $\langle [0, 1], max, \cdot, 0, 1 \rangle$
 \Rightarrow Availability rate, performance, etc.
- Fuzzy c-semiring: $\langle [0, 1], max, min, 0, 1 \rangle$
- Set-based c-semiring: $\langle 2^N, \cup, \cap, \emptyset, N \rangle$
 \Rightarrow Capabilities, access rights, etc.

C-Semirings (Properties I)

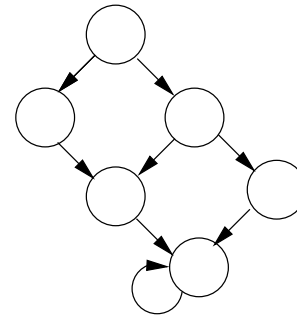
- Cartesian products, exponentials and power constructions of c-semirings are c-semirings.
 - ⇒ Combine multiple criteria, e.g. multiple QoS attributes.
- The additive operation induces a partial order:
 - $a \leq_S b$ iff $a + b = b$ (equiv. $\exists c \mid a + c = b$).
 - ⇒ E.g., partial order for $\langle \mathbb{R}^+, \min, +, +\infty, 0 \rangle$ is \geq :
$$a \geq b \text{ iff } \min(a, b) = b.$$

C-Semirings (Properties II)

- $\langle A, \leq_S \rangle$ is a complete lattice:
 - Every subset of A has a lub or join (\sqcup) which is $+$;
 - Every subset of A has a glb or meet (\sqcap).
 - \Rightarrow E.g., meet is *max* in $\langle \mathbb{R}^+, \min, +, +\infty, 0 \rangle$.
 - In practical cases the lattice is distributive.
- Sometimes the multiplicative operation is idempotent:
 - \times coincides with meet (\sqcap);
 - $+$ distributes over \times ;
 - $\langle A, \leq_S \rangle$ is a distributive lattice.
 - \Rightarrow E.g., boolean, fuzzy, set-based c-semirings.

Transition Systems

- Transition systems $M = \langle S, T \rangle$, where:
 - S is a set of states;
 - $T \subseteq S \times S$ is a set of transitions.
- We assume M to be *image-finite*, T to be total.
- Runs of a system are *maximal paths*.
- A path is a sequence $s_0, s_1, s_2 \dots$, such that $(s_i, s_{i+1}) \in T$.



Boolean Model checking

- Which states of M satisfy ϕ ?

$$[[\phi]] \subseteq 2^S$$

or

$$[[\phi]] : S \rightarrow \{true, false\}$$

C-Semiring Model Checking

- Which value of A associates ϕ to each state of M ?

$$[[\phi]] : S \rightarrow A.$$

, where $C = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ is a c-semiring.

Boolean CTL (Syntax)

$\phi ::= true \mid false \mid p \mid \neg\phi \mid \phi \vee \phi \mid \phi \wedge \phi \mid \kappa\psi \mid \kappa\mathbf{X}\phi$

$\kappa ::= \mathbf{E} \mid \mathbf{A}$

$\psi ::= \mathbf{F}\phi \mid \mathbf{G}\phi \mid [\phi\mathbf{U}\phi] \mid [\phi\mathbf{R}\phi]$

C-Semiring CTL (Syntax)

$\phi ::= a \mid v \mid f(\phi, \dots, \phi) \mid \phi + \phi \mid \phi \times \phi \mid \kappa\psi \mid \kappa\mathbf{X}\phi$

$\kappa ::= \mathbf{\Pi} \mid \mathbf{\Sigma} \mid \mathbf{\Pi}$

$\psi ::= \mathbf{F}\phi \mid \mathbf{G}\phi \mid [\phi\mathbf{U}\phi] \mid [\phi\mathbf{R}\phi]$

C-CTL (Path Semantics over TS I)

$$\begin{aligned} \llbracket a \rrbracket(s) &= a \\ \llbracket v \rrbracket(s) &= v(s) \\ \llbracket \phi_1 + \phi_2 \rrbracket(s) &= \llbracket \phi_1 \rrbracket(s) + \llbracket \phi_2 \rrbracket(s) \\ \llbracket \phi_1 \times \phi_2 \rrbracket(s) &= \llbracket \phi_1 \rrbracket(s) \times \llbracket \phi_2 \rrbracket(s) \\ \llbracket f(\phi_1, \dots, \phi_n) \rrbracket(s) &= f(\llbracket \phi_1 \rrbracket(s), \dots, \llbracket \phi_n \rrbracket(s)) \end{aligned}$$

C-CTL (Path Semantics over TS II)

$$\llbracket \kappa \mathbf{X} \phi \rrbracket (s) = \kappa_{(s,s') \in T} \llbracket \phi \rrbracket (s')$$

$$\llbracket \kappa \psi \rrbracket (s) = \kappa_{p \in \gamma(s)} \llbracket \psi \rrbracket (p)$$

$$\llbracket \mathbf{F} \phi \rrbracket (p) = \sum_{i \geq 0} \llbracket \phi \rrbracket (s_i^p)$$

$$\llbracket \mathbf{G} \phi \rrbracket (p) = \prod_{i \geq 0} \llbracket \phi \rrbracket (s_i^p)$$

$$\llbracket \phi_1 \mathbf{U} \phi_2 \rrbracket (p) = \sum_{i \geq 0} (\llbracket \phi_2 \rrbracket (s_i^p) \times \prod_{0 \leq j < i} \llbracket \phi_1 \rrbracket (s_j^p))$$

$$\llbracket \phi_1 \mathbf{R} \phi_2 \rrbracket (p) = \prod_{i \geq 0} (\llbracket \phi_2 \rrbracket (s_i^p) + \sum_{0 \leq j < i} \llbracket \phi_1 \rrbracket (s_j^p))$$

, where $\gamma(s)$ is the set of maximal paths starting at s .

C-CTL Path Semantics (Temporal Operators)

Let $p = s_0, s_1, s_2, \dots$

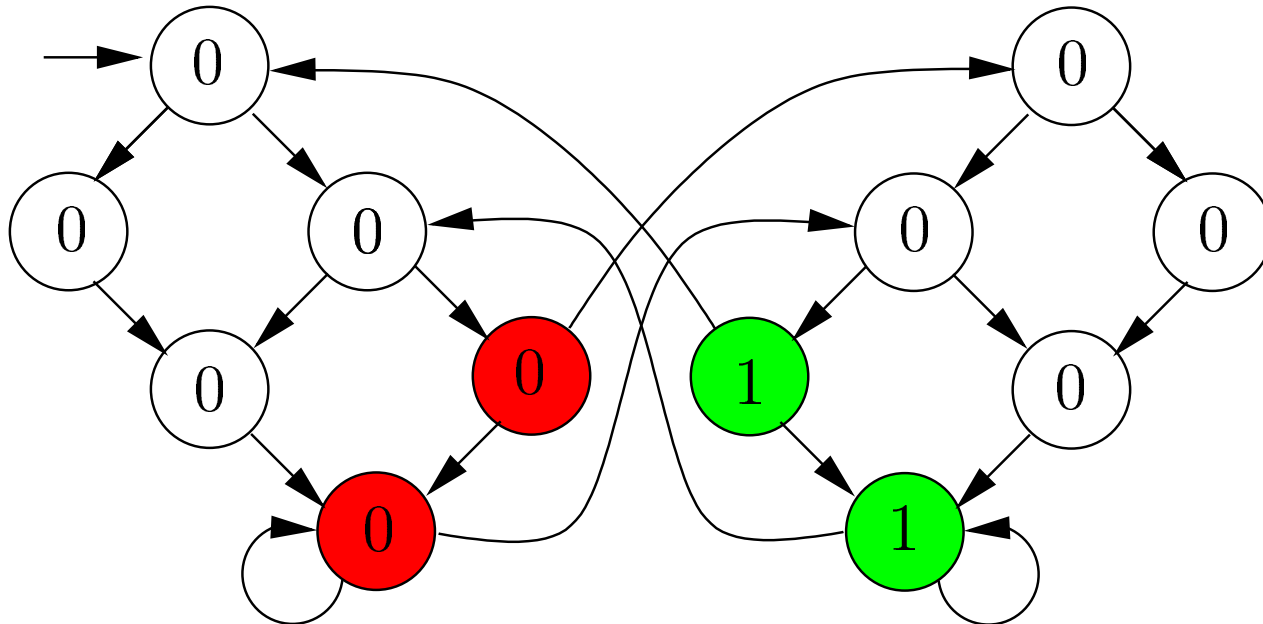
$$\llbracket \mathbf{F}\phi \rrbracket(p) = \llbracket \phi \rrbracket(s_0) + \llbracket \phi \rrbracket(s_1) + \llbracket \phi \rrbracket(s_2) + \dots$$

$$\llbracket \mathbf{G}\phi \rrbracket(p) = \llbracket \phi \rrbracket(s_0) \times \llbracket \phi \rrbracket(s_1) \times \llbracket \phi \rrbracket(s_2) \times \dots$$

$$\llbracket \phi_1 \mathbf{U} \phi_2 \rrbracket(p) = \sum \left\{ \begin{array}{l} \llbracket \phi_2 \rrbracket(s_1) \\ \llbracket \phi_1 \rrbracket(s_1) \times \llbracket \phi_2 \rrbracket(s_2) \\ \llbracket \phi_1 \rrbracket(s_1) \times \llbracket \phi_1 \rrbracket(s_2) \times \llbracket \phi_2 \rrbracket(s_3) \\ \dots \end{array} \right.$$

$$\llbracket \phi_1 \mathbf{R} \phi_2 \rrbracket(p) = \prod \left\{ \begin{array}{l} \llbracket \phi_2 \rrbracket(s_1) \\ \llbracket \phi_1 \rrbracket(s_1) + \llbracket \phi_2 \rrbracket(s_2) \\ \llbracket \phi_1 \rrbracket(s_1) + \llbracket \phi_1 \rrbracket(s_2) + \llbracket \phi_2 \rrbracket(s_3) \\ \dots \end{array} \right.$$

Mutual Exclusion: Reasoning about Costs $\langle \mathbb{R}^+, \min, +, +\infty, 0 \rangle$



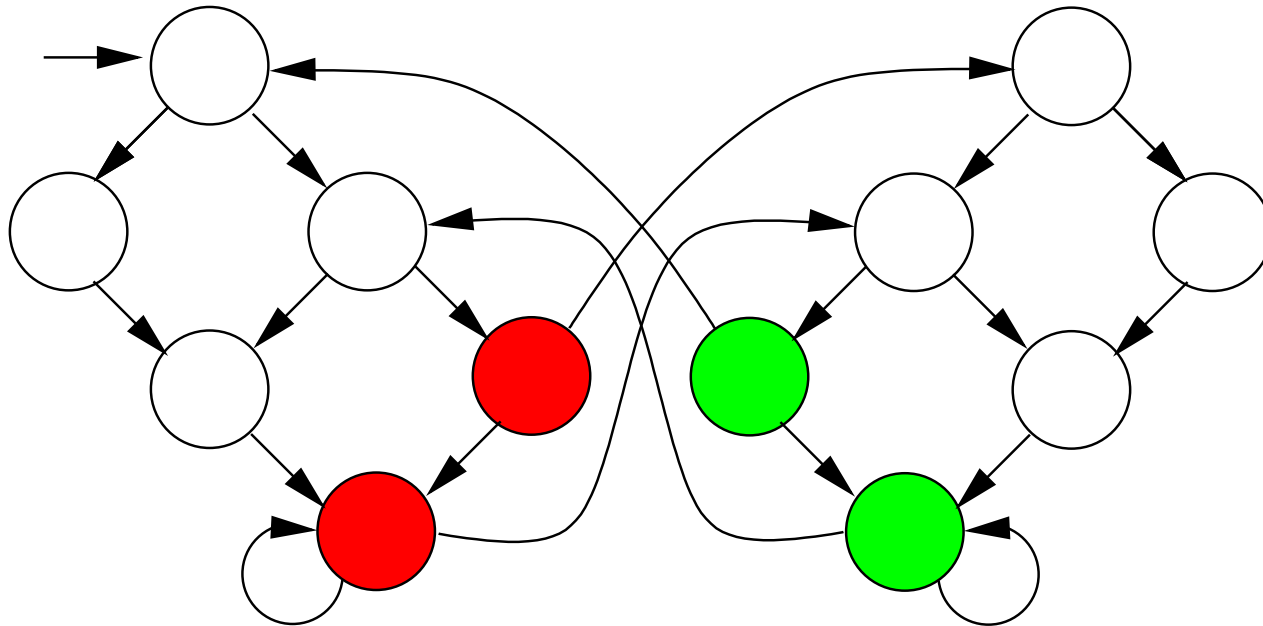
- Min. cost for a process to use the c.s.?

$$\llbracket \sum \mathbf{F} \text{cost} \rrbracket (s_0) = 0$$

- Min. cost for a process to use the c.s. forever?

$$\llbracket \sum \mathbf{F} \sum \mathbf{G} \text{cost} \rrbracket (s_0) = 0$$

Mutual exclusion (all together)



- Optimal QoS to reach the c.s.?

$$\llbracket \Sigma \mathbf{F} qos \rrbracket (s_0) = \{(\{p\}, 0, 0.5), (\{q\}, 1, 1)\}$$

- Optimal QoS to reach the c.s. and stay forever?

$$\llbracket \Sigma \mathbf{F} \Sigma \mathbf{G} qos \rrbracket (s_0) = \{(\{p\}, 0, 0), (\{q\}, \infty, 1)\}$$

C-CTL (Fixpoint Semantics over TS)

$$\llbracket \kappa \mathbf{F} \phi \rrbracket^f = \llbracket \mu z. \phi + \kappa \mathbf{X} z \rrbracket$$

$$\llbracket \kappa \mathbf{G} \phi \rrbracket^f = \llbracket \nu z. \phi \times \kappa \mathbf{X} z \rrbracket$$

$$\llbracket \kappa [\phi_1 \mathbf{U} \phi_2] \rrbracket^f = \llbracket \mu z. \phi_2 + (\phi_1 \times \kappa \mathbf{X} z) \rrbracket$$

$$\llbracket \kappa [\phi_1 \mathbf{R} \phi_2] \rrbracket^f = \llbracket \nu z. \phi_2 \times (\phi_1 + \kappa \mathbf{X} z) \rrbracket$$

Path vs. Fixpoint Semantics

If \times is **is** idempotent ...

- If \times is idempotent $\forall \phi \in \mathbf{c}\text{-CTL}$: $\llbracket \phi \rrbracket = \llbracket \phi \rrbracket^f$.

If \times is **not** idempotent ...

- $\forall \phi \in \mathbf{c}\text{-}\{\Sigma, \Pi\}\text{CTL}$: $\llbracket \phi \rrbracket \leq_V \llbracket \phi \rrbracket^f$.
- In practice: $\forall \phi \in \mathbf{c}\text{-CTL}$: $\llbracket \phi \rrbracket \leq_V \llbracket \phi \rrbracket^f$

Verification Algorithms for c-CTL

If \times is **is** idempotent ...

- $a \times a \times \dots = a$ and $a + a + \dots = a$
 \Rightarrow we can consider acyclic paths only.
- $O(|S|)$ fixpoint iterations are sufficient.

If \times is **not** idempotent ...

- Path and fixpoint semantics require different algorithms.
- Fixpoint iteration not feasible: might require ω iterations.
- Things are easier if $a \times a \times \dots = \mathbf{0}$ (unless $a=1$)

Bisimulation

- sRs' whenever:
 - $v(s) = v(s')$ for all $v \in AV$;
 - $s \rightarrow s_1$ then $s' \rightarrow s'_1 \mid s_1Rs'_1$;
 - $s' \rightarrow s'_1$ then $s \rightarrow s_1 \mid s_1Rs'_1$.

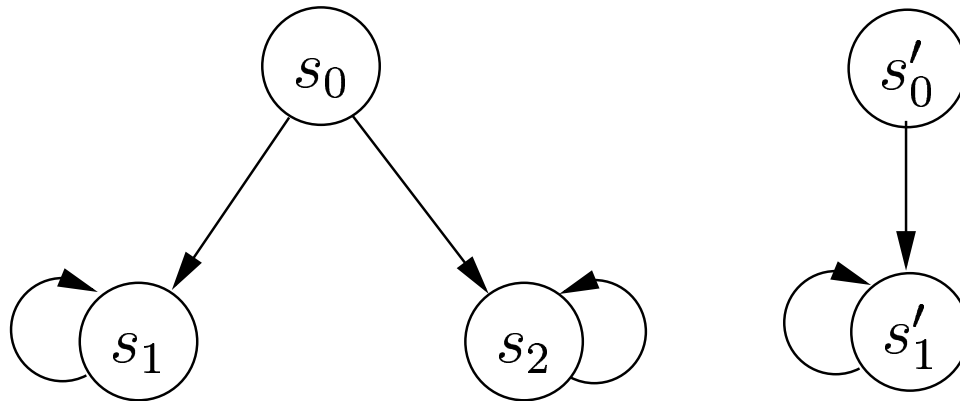
- If \times is **is** idempotent...

$$\forall \phi \in \text{c-CTL}: sRs' \text{ then } \llbracket \phi \rrbracket(s) = \llbracket \phi \rrbracket(s')$$

- If \times is **not** idempotent...

$$\forall \phi \in \text{c-}\{\Sigma, \sqcap\}\text{CTL}: sRs' \text{ then } \llbracket \phi \rrbracket(s) = \llbracket \phi \rrbracket(s')$$

Bisimulation (Counterexample)



$s_0 R s'_0$ but... $\llbracket \prod \mathbf{X}a \rrbracket(s_0) = a \times a \neq a = \llbracket \prod \mathbf{X}a \rrbracket(s'_0)$

Simulation

- sHs' whenever:
 - $v(s) \leq_S v(s')$ for all $v \in AV$;
 - $s \rightarrow s_1$ then $s' \rightarrow s'_1 \mid s_1Hs'_1$.

- If \times is **is** idempotent:

$$\forall \phi \in \text{c-CTL}: sHs' \text{ then } \llbracket \phi \rrbracket(s) \leq_S \llbracket \phi \rrbracket(s')$$

- If \times is **not** idempotent...

$$\forall \phi \in \text{c-}\{\Sigma, \sqcap\}\text{CTL}: sRs' \text{ then } \llbracket \phi \rrbracket(s) \leq_S \llbracket \phi \rrbracket(s')$$

Captured Problems

- Graph problems:
 - ⇒ Reachability, (multi-criteria) path optimization, etc.
- Boolean and quasi-boolean model checking:
 - ⇒ Multi-valued CTL [Chechik et al,03].
- Some probabilistic model checking approaches:
 - ⇒ Fuzzy CTL over transition systems [de Alfaro et al.,04].
- Some discounted model checking problems (via $c-L_\mu$):
 - ⇒ Discounted CTL [de Alfaro et al., 04].

Conclusions

- We have extended the μ -calculus and CTL to c-semirings.
- The usual connection between both logics is not general.
- Model checking CTL when \times is idempotent can be done via fixpoint iteration.
- The framework captures some quantitative model checking problems.
- Future work: algorithms, WAN applications, extension to spatial logics, etc.