Compositional Specification Theories for Stochastic Systems

Benoît Delahaye

Université de Nantes

2013-11-7

Joint work with B. Caillaud, J.-P. Katoen, K.G. Larsen, A. Legay, M.L. Pedersen, F. Sher, A. Wasowski

Outline

Specification Theories:

- A specification (interface) allows to represent the behavior of multiple components at the design level
- Allows independent reasoning
- Supports component-based design of large systems
- Reduces complexity of the design
- Existing Theories: Modal Specifications, Interface Automata...

Specification Theories:

- A specification (interface) allows to represent the behavior of multiple components at the design level
- Allows independent reasoning
- Supports component-based design of large systems
- Reduces complexity of the design
- Existing Theories: Modal Specifications, Interface Automata...

Our contribution(s): The first complete theories for Markov Chains and Probabilistic Automata

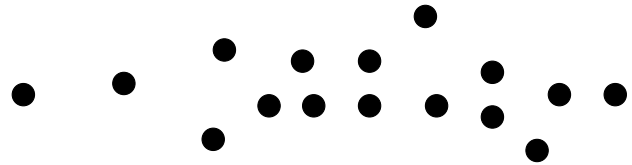
Specification Theories:

- A specification (interface) allows to represent the behavior of multiple components at the design level
- Allows independent reasoning
- Supports component-based design of large systems
- Reduces complexity of the design
- Existing Theories: Modal Specifications, Interface Automata...

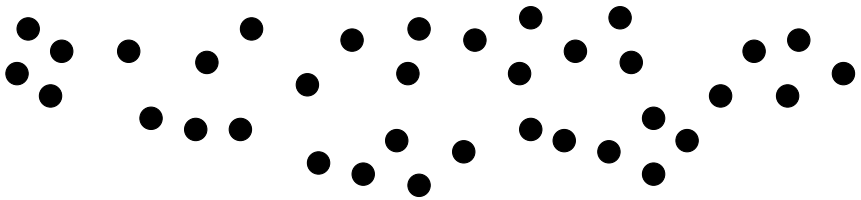
Our contribution(s): The first complete theories for Markov Chains and Probabilistic Automata

But ... What is a complete theory?

Specification Theories: Implementation/comparison(refinement)

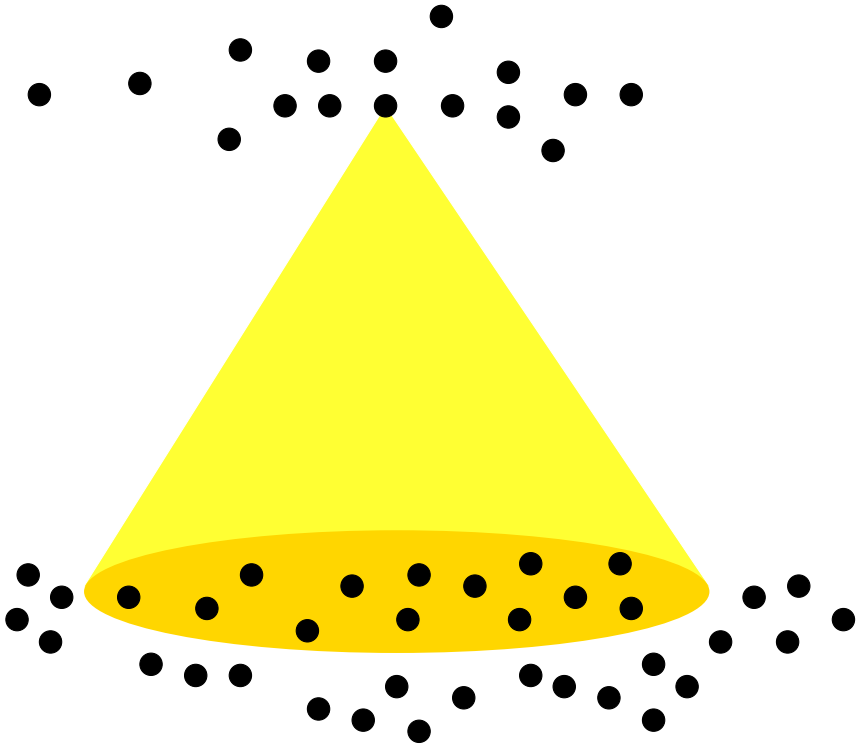


Specifications



Implementations

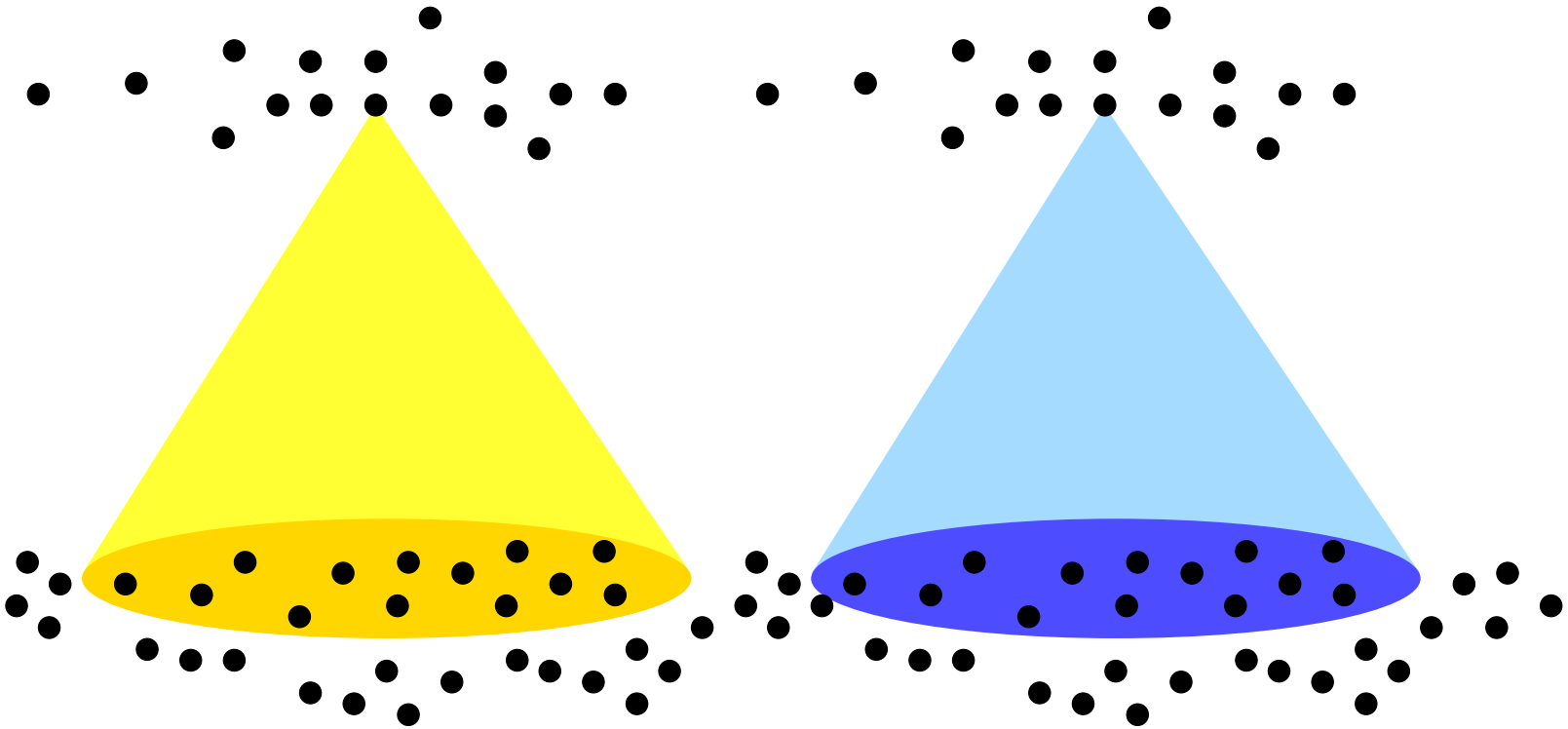
Specification Theories: Implementation/comparison(refinement)



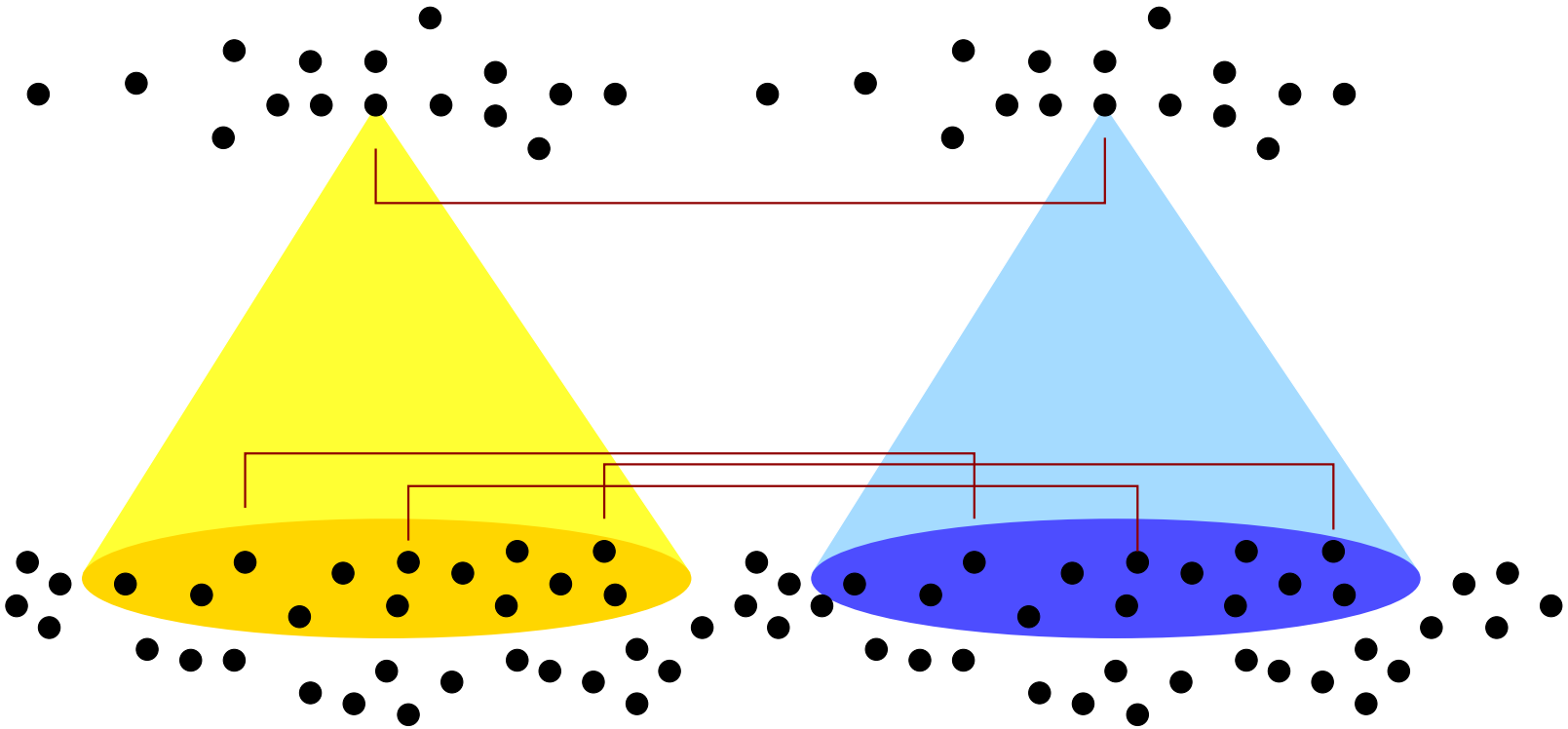
Specifications

Implementations

Specification Theories: Implementation/comparison(refinement)

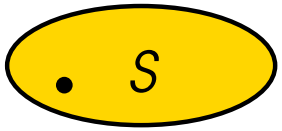


Specification Theories: Implementation/comparison(refinement)



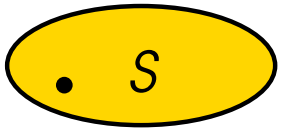
Single Component Operators

Consistency

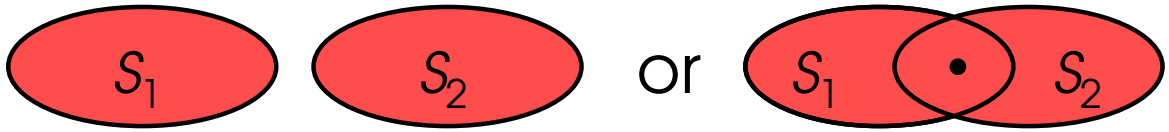
$S = \varnothing$ or  ?

Single Component Operators

Consistency

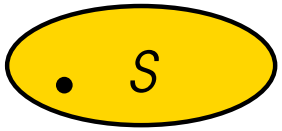
$S = \varnothing$ or  ?

Conjunction

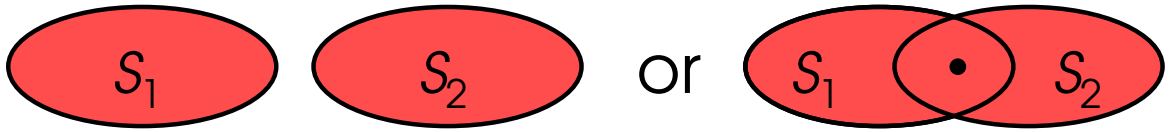
 ?

Single Component Operators

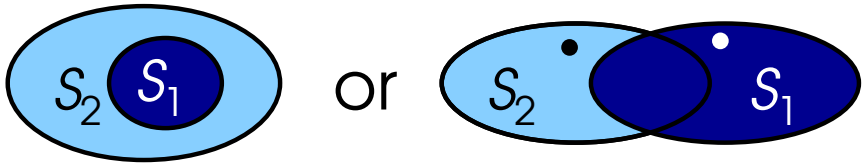
Consistency

$S = \varnothing$ or  ?

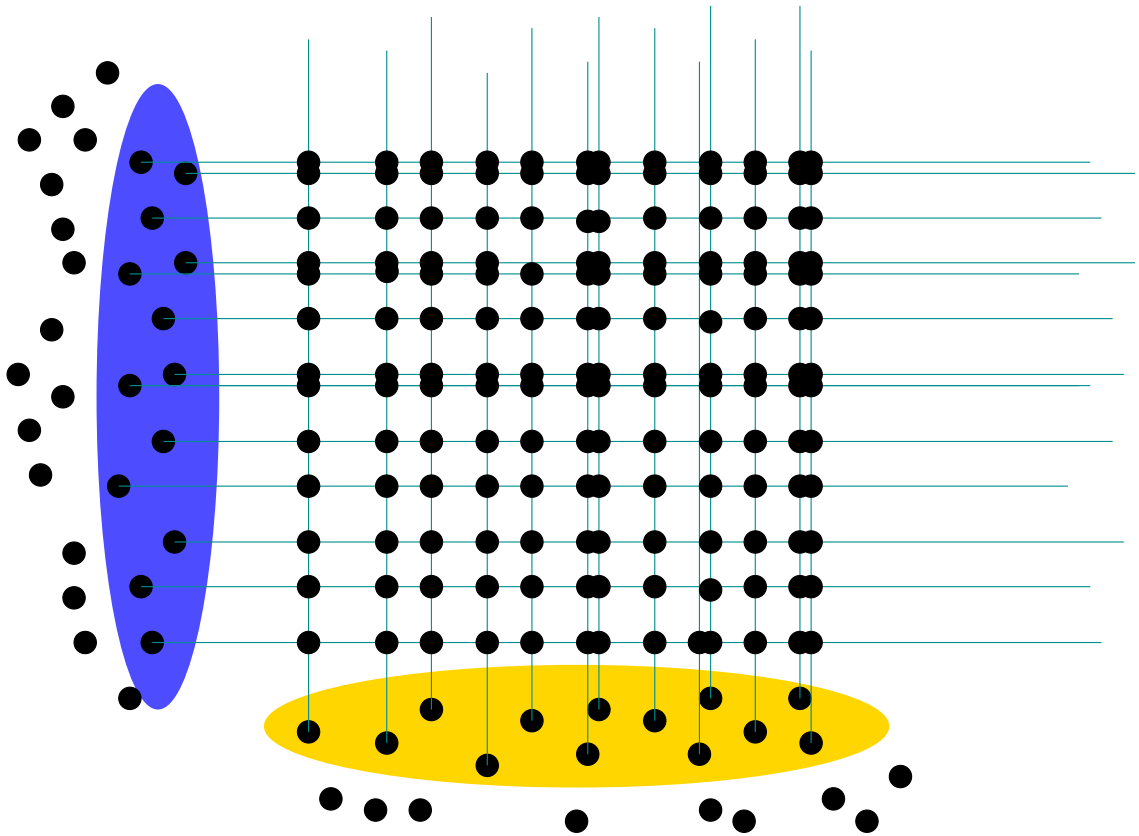
Conjunction

 or ?

Refinement

 or ?

Multiple Component Operators



Parallel Composition

Outline

Interval Markov Chains

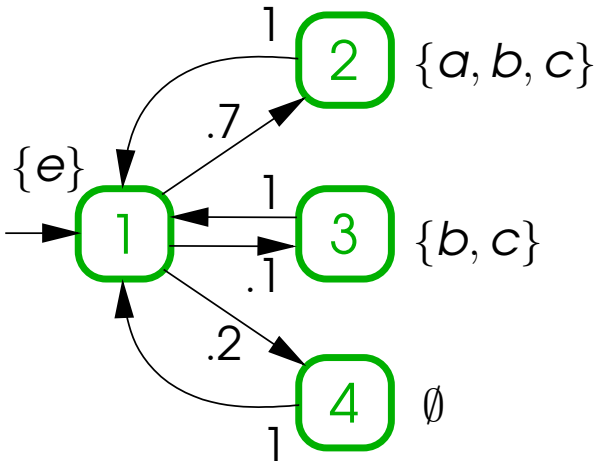
- B. Delahaye, K.G. Larsen, A. Legay, M.L. Pedersen, A. Wasowski. Consistency and Refinement for Interval Markov Chains. In *Journal of Logic and Algebraic Programming*, 2012.
- B. Delahaye, K.G. Larsen, A. Legay, M.L. Pedersen, A. Wasowski. Decision Problems for Interval Markov Chains. In *LATA, 5th International Conference on Language and Automata Theory and Applications*, Tarragona, Spain, 2011.

Markov Chains: An Example

$$M = (\{1, \ldots, n\}, o, M, A, V)$$

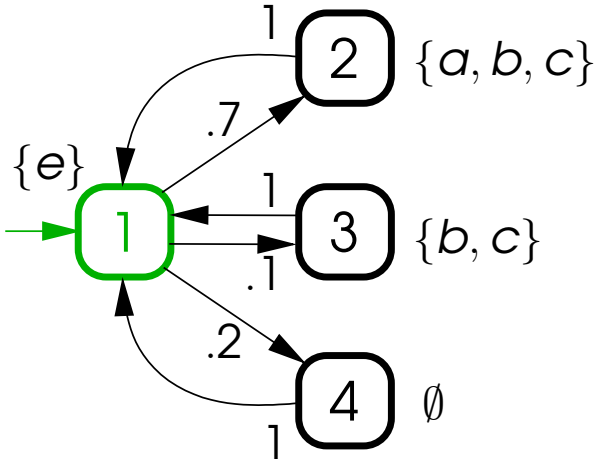
Markov Chains: An Example

$$M = (\{1, \ldots, n\}, o, M, A, V)$$



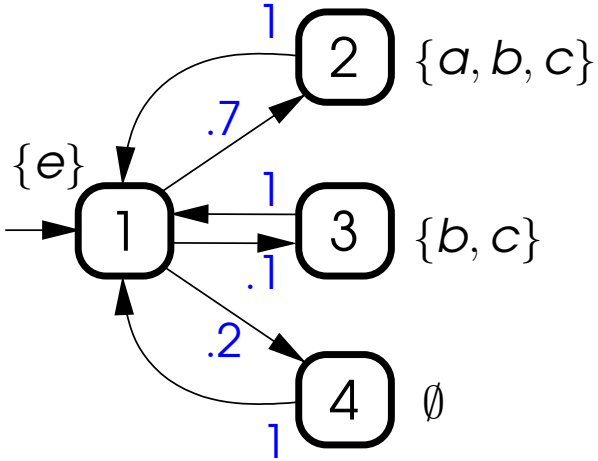
Markov Chains: An Example

$$M = (\{1, \ldots, n\}, o, M, A, V)$$



Markov Chains: An Example

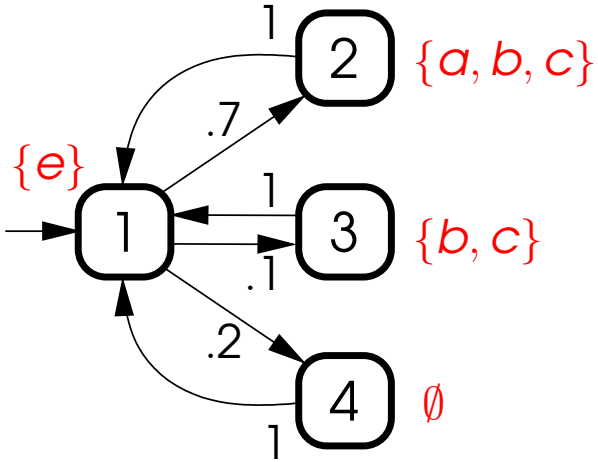
$$M = (\{1, \ldots, n\}, o, M, A, V)$$



$$\begin{pmatrix} 0 & 0.7 & 0.1 & 0.2 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Markov Chains: An Example

$$M = (\{1, \ldots, n\}, o, M, A, V)$$



$$\begin{pmatrix} 0 & 0.7 & 0.1 & 0.2 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

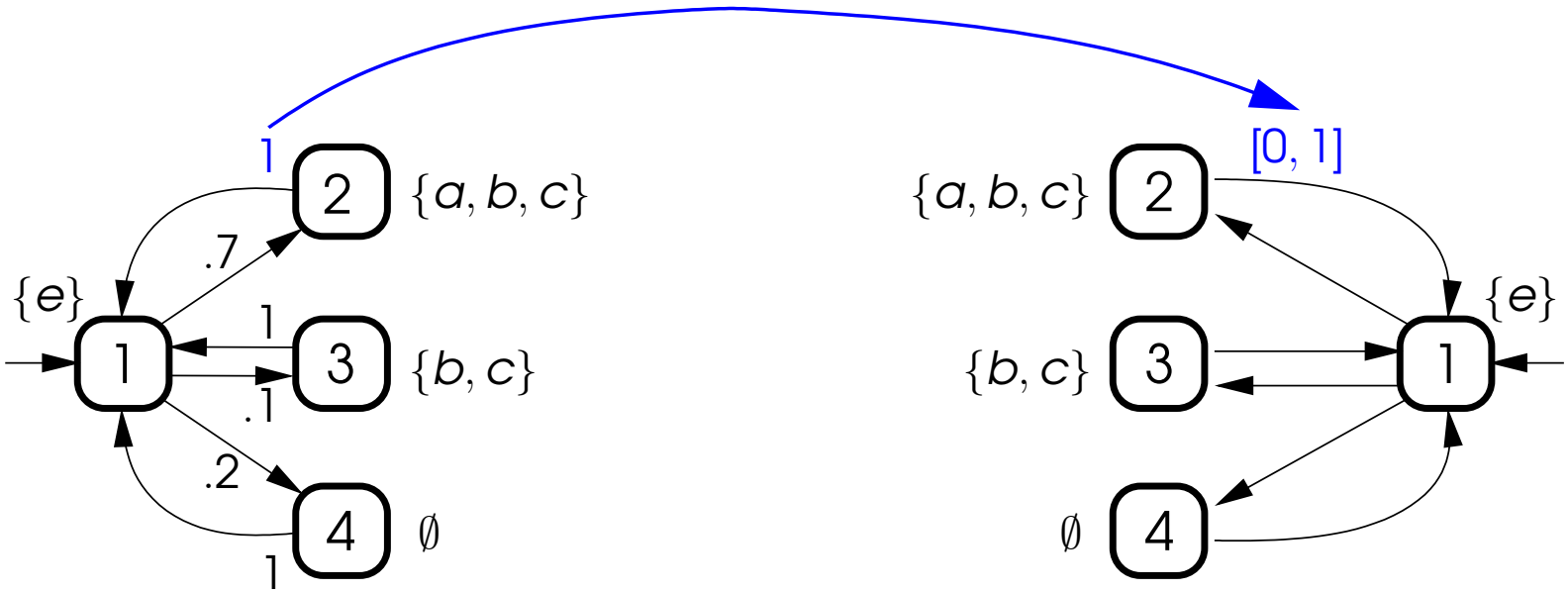
Interval Markov Chains: An Example

$$S = (\{1, \ldots, n\}, o, I, A, V)$$

Idea: Replace exact probabilities with intervals

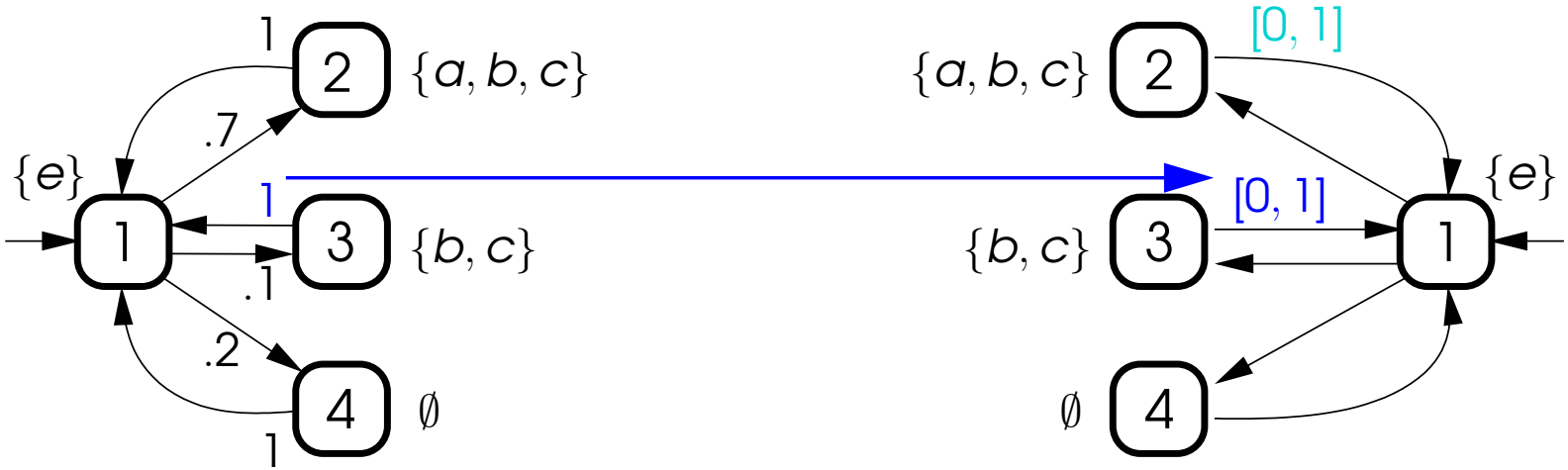
Interval Markov Chains: An Example

$$S = (\{1, \ldots, n\}, o, I, A, V)$$



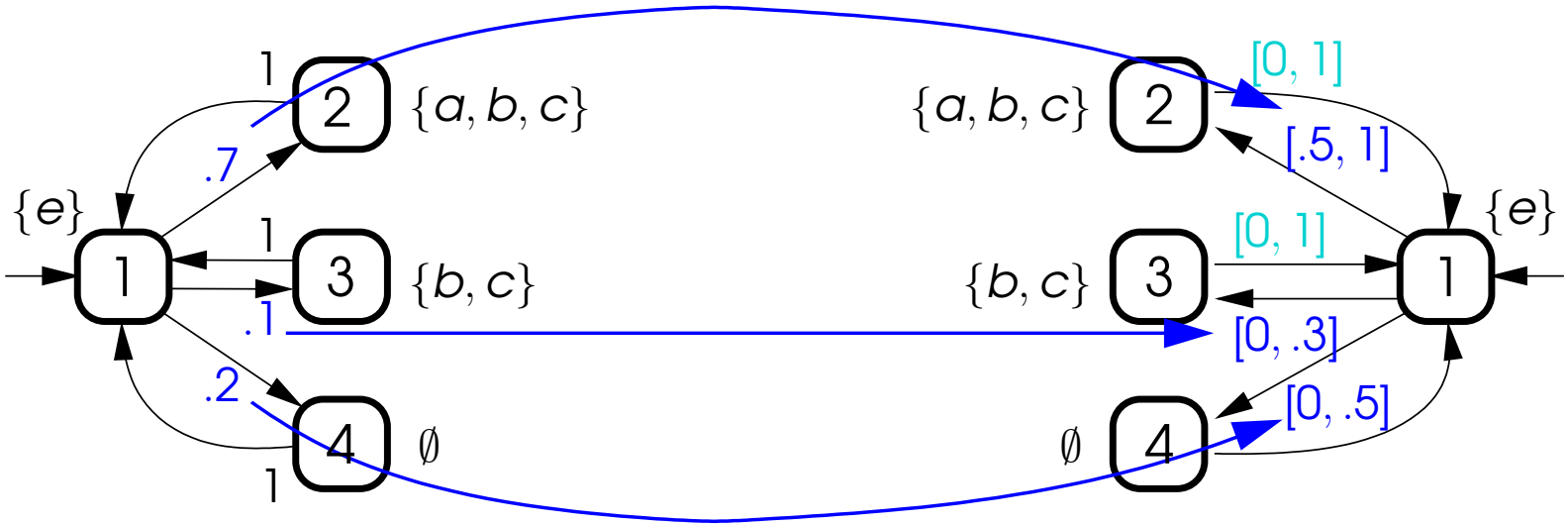
Interval Markov Chains: An Example

$$S = (\{1, \ldots, n\}, o, I, A, V)$$



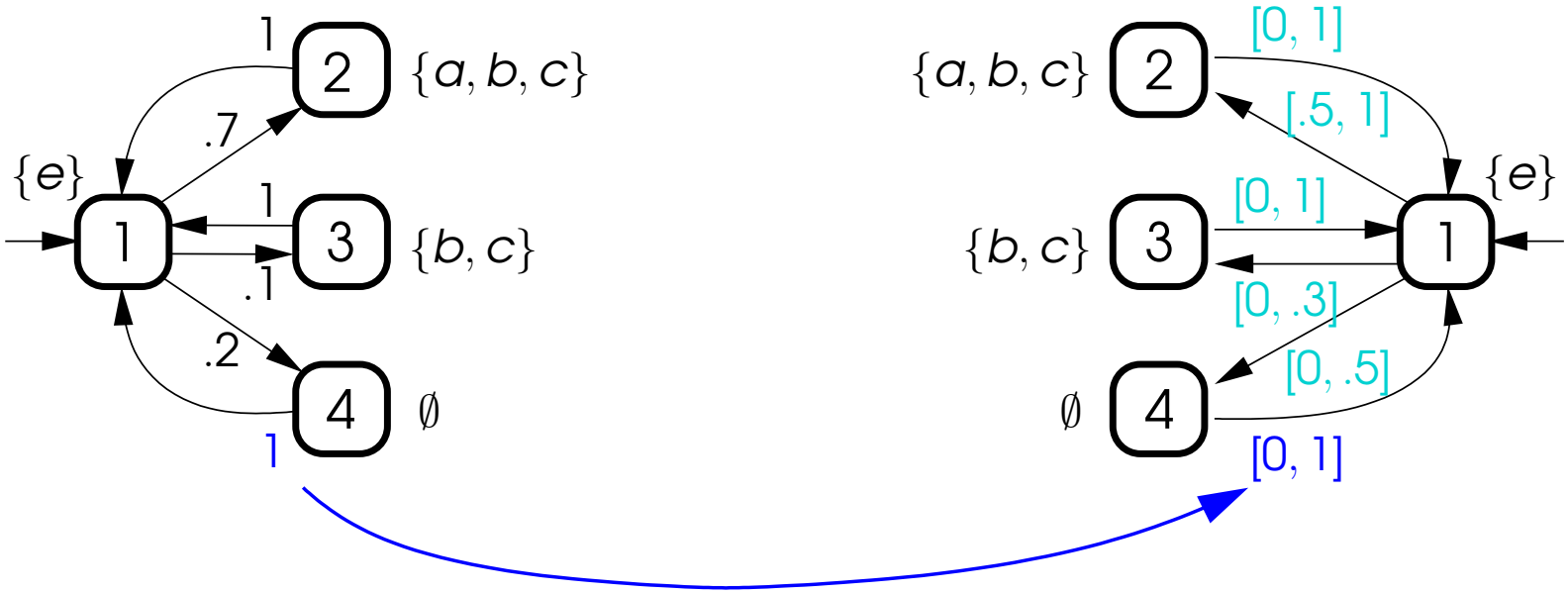
Interval Markov Chains: An Example

$$S = (\{1, \ldots, n\}, o, I, A, V)$$



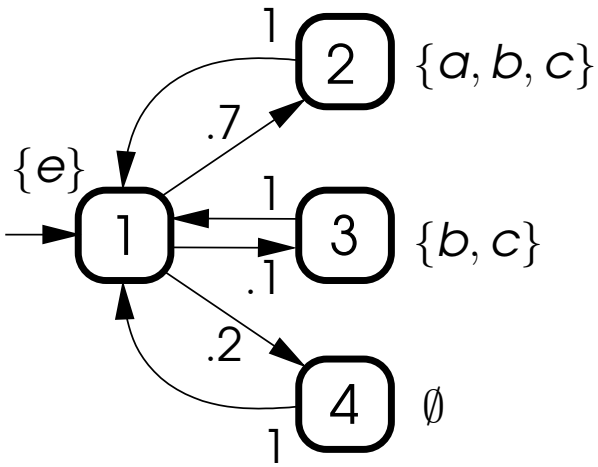
Interval Markov Chains: An Example

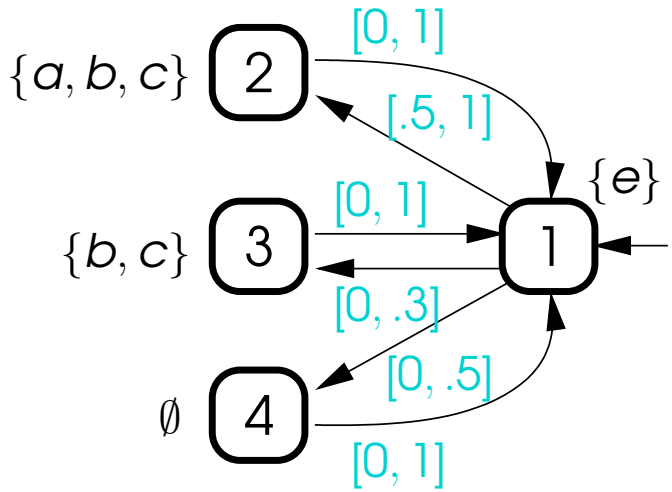
$$S = (\{1, \ldots, n\}, o, I, A, V)$$



Interval Markov Chains: An Example

$$S = (\{1, \ldots, n\}, o, I, A, V)$$





Our results concerning IMCs

- Consistency Checking: PTIME

Our results concerning IMCs

- Consistency Checking: PTIME
- Common Implementation: EXPTIME-complete

Our results concerning IMCs

- Consistency Checking: PTIME
- Common Implementation: EXPTIME-complete
- Semantic Refinement: EXPTIME-complete

Our results concerning IMCs

- Consistency Checking: PTIME
- Common Implementation: EXPTIME-complete
- Semantic Refinement: EXPTIME-complete
- Composition & Conjunction: Absence of closure

Absence of closure: An illustration (1)

Design of a coffee machine:

Absence of closure: An illustration (1)

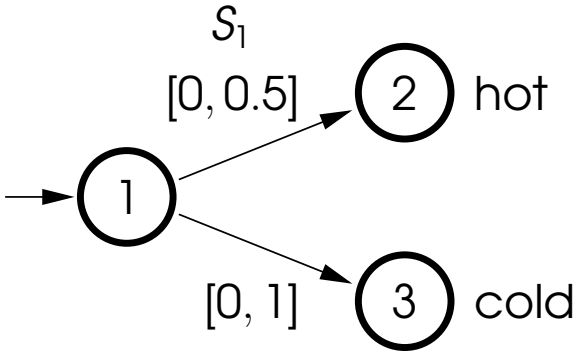
Design of a coffee machine:

- Requirement one (temperature): At most 50% of the drinks delivered are hot drinks

Absence of closure: An illustration (1)

Design of a coffee machine:

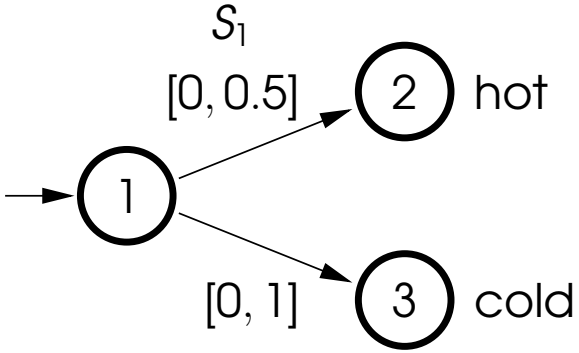
- Requirement one (temperature):



Absence of closure: An illustration (1)

Design of a coffee machine:

- Requirement one (temperature):

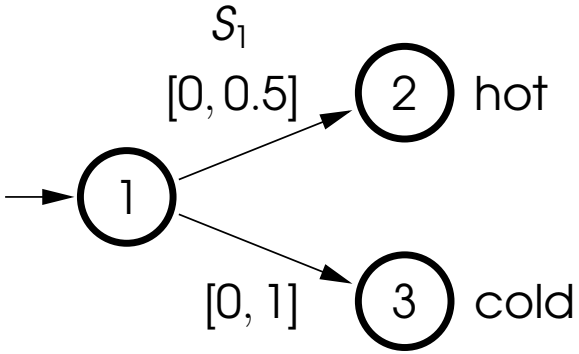


- Requirement two (choice of beverage): At least 20% of the drinks delivered are coffee

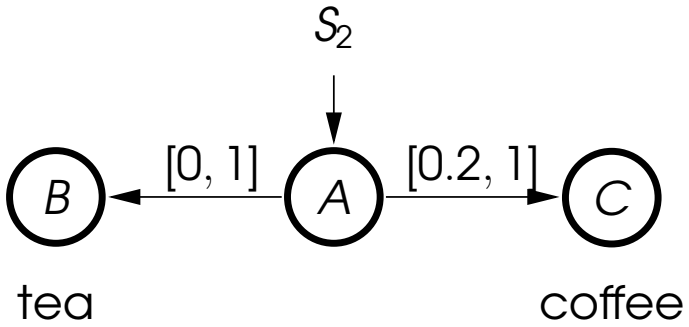
Absence of closure: An illustration (1)

Design of a coffee machine:

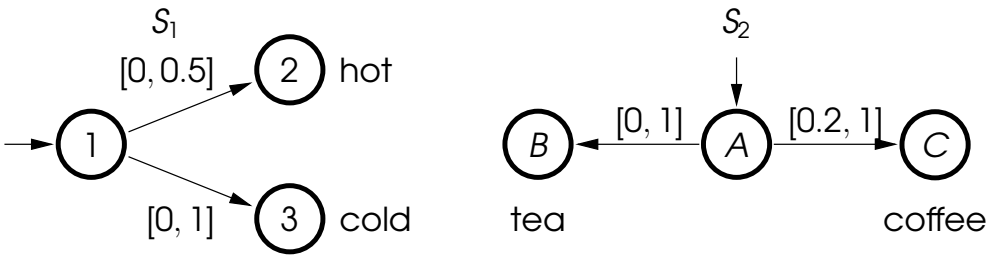
- Requirement one (temperature):



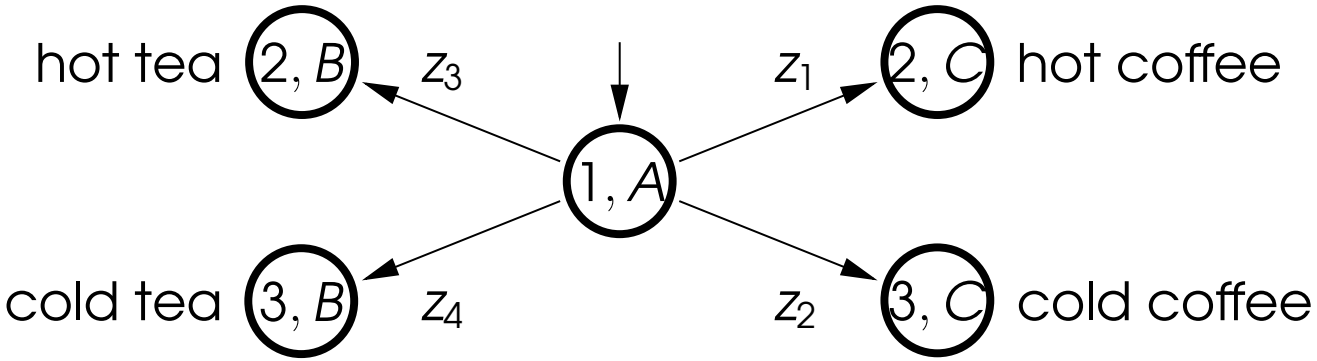
- Requirement two (choice of beverage):



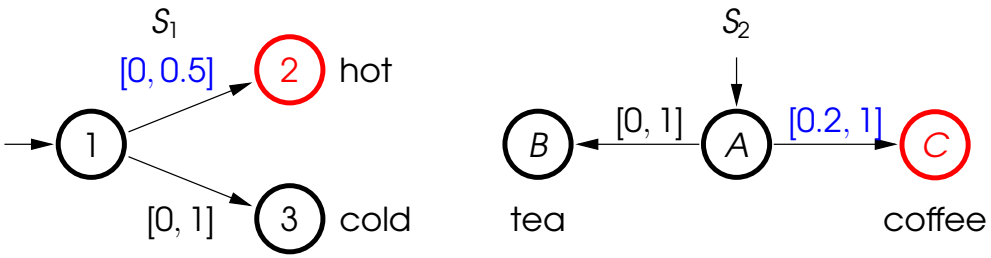
Absence of closure: An illustration (2)



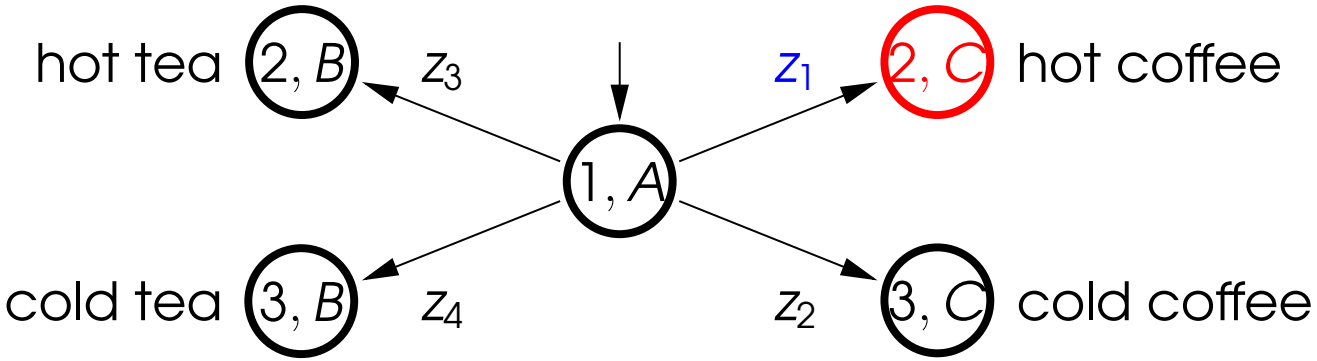
Conjunction of S_1 and S_2:



Absence of closure: An illustration (2)

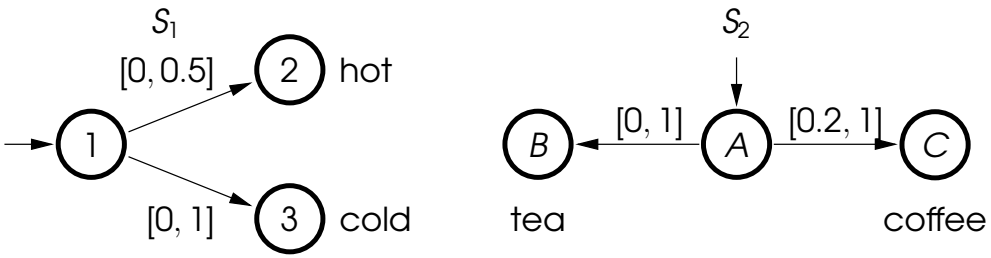


Conjunction of S_1 and S_2:

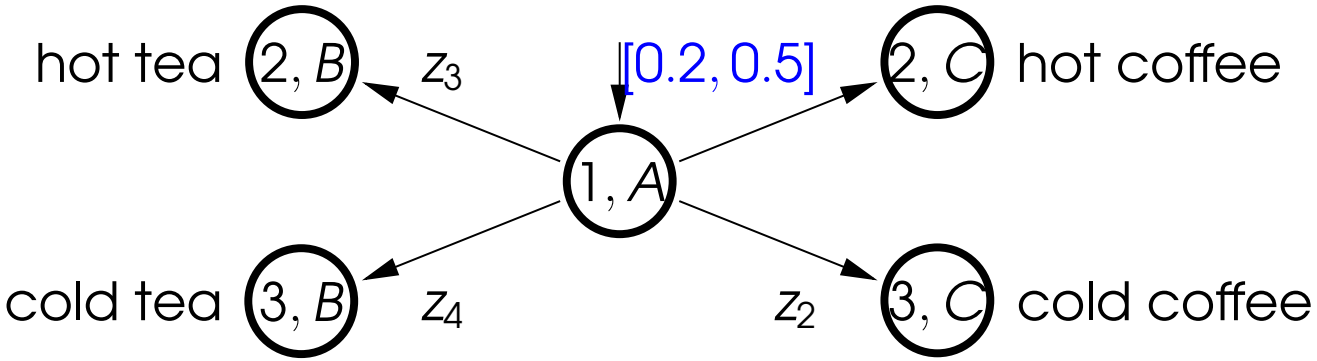


$$z_1 \in [0, 0.5] \cap [0.2, 1] \Rightarrow z_1 \in [0.2, 0.5]$$

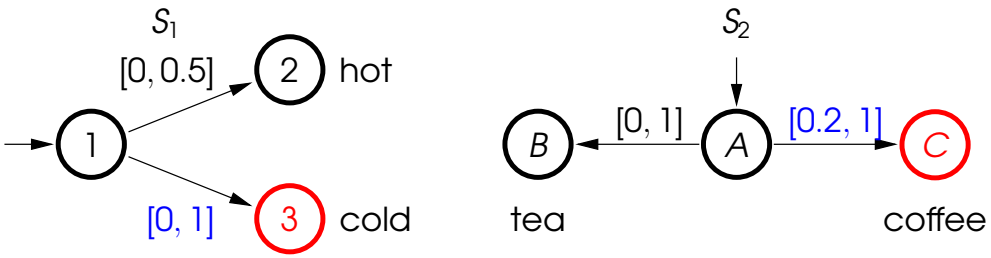
Absence of closure: An illustration (2)



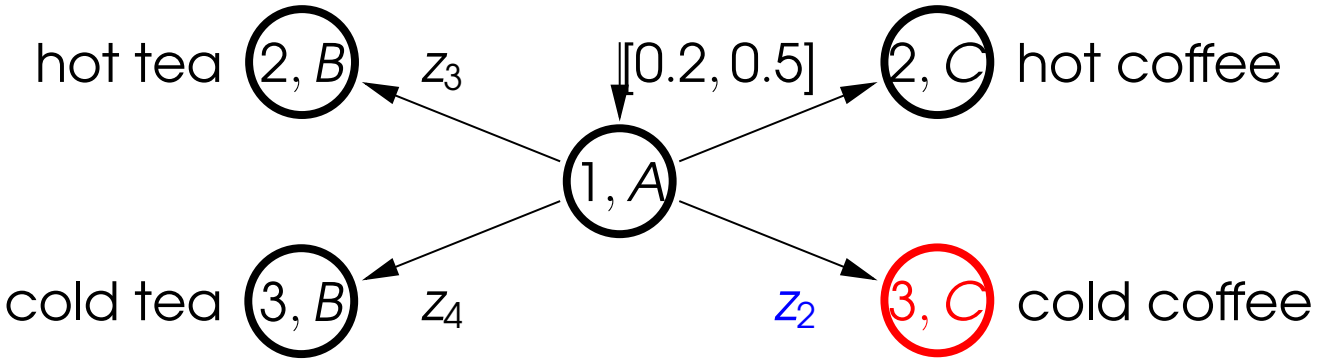
Conjunction of S_1 and S_2:



Absence of closure: An illustration (2)

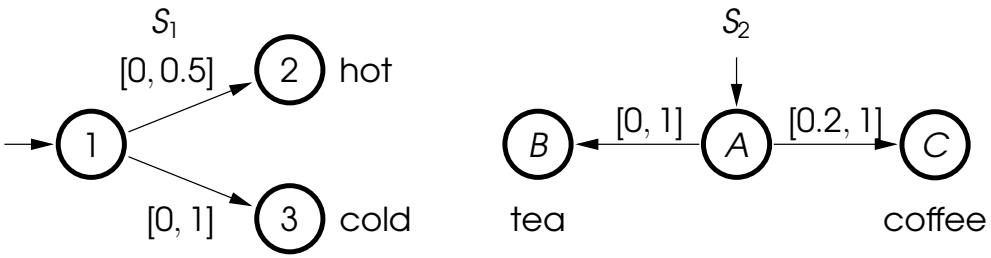


Conjunction of S_1 and S_2:

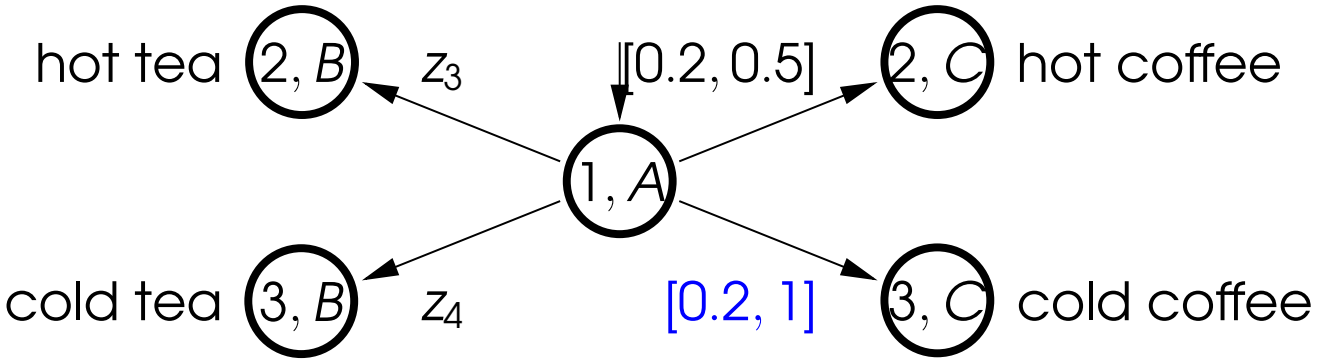


$$z_2 \in [0, 1] \cap [0.2, 1] \Rightarrow z_1 \in [0.2, 1]$$

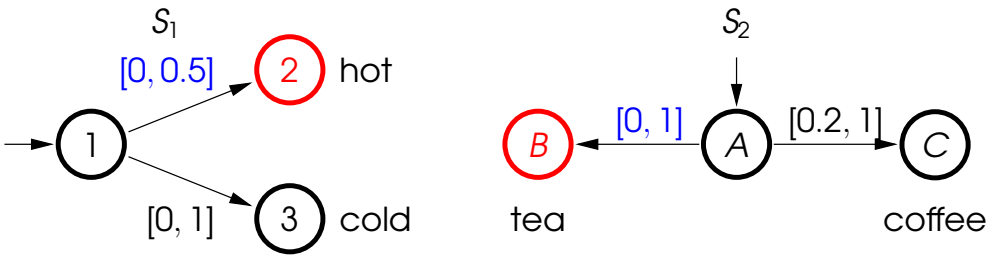
Absence of closure: An illustration (2)



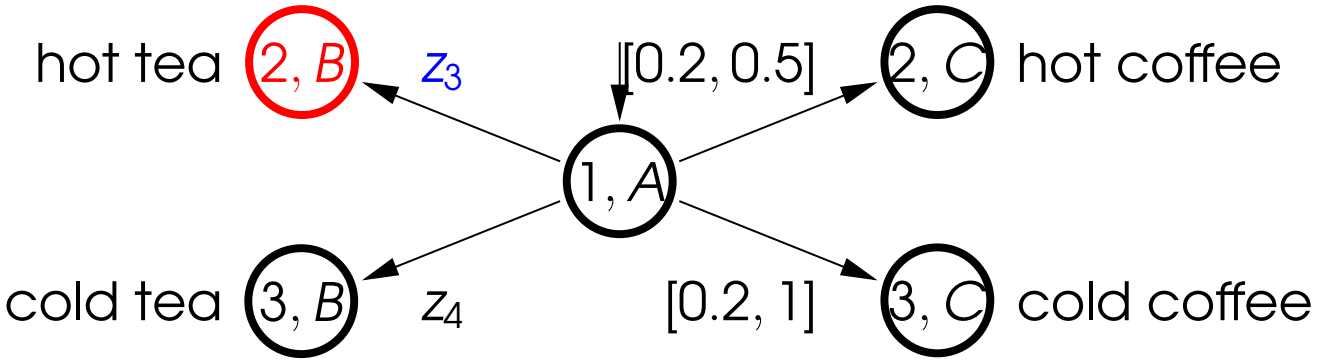
Conjunction of S_1 and S_2:



Absence of closure: An illustration (2)

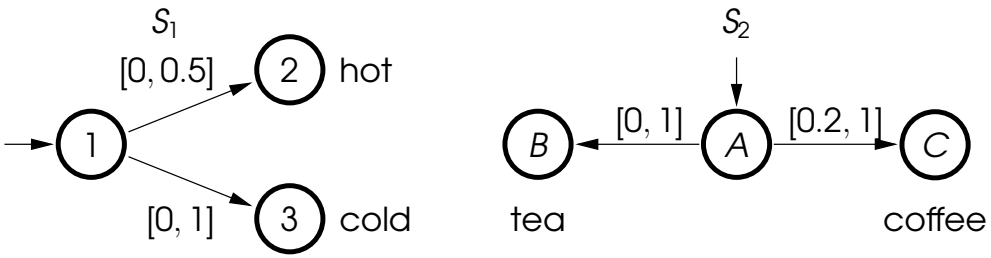


Conjunction of S_1 and S_2:

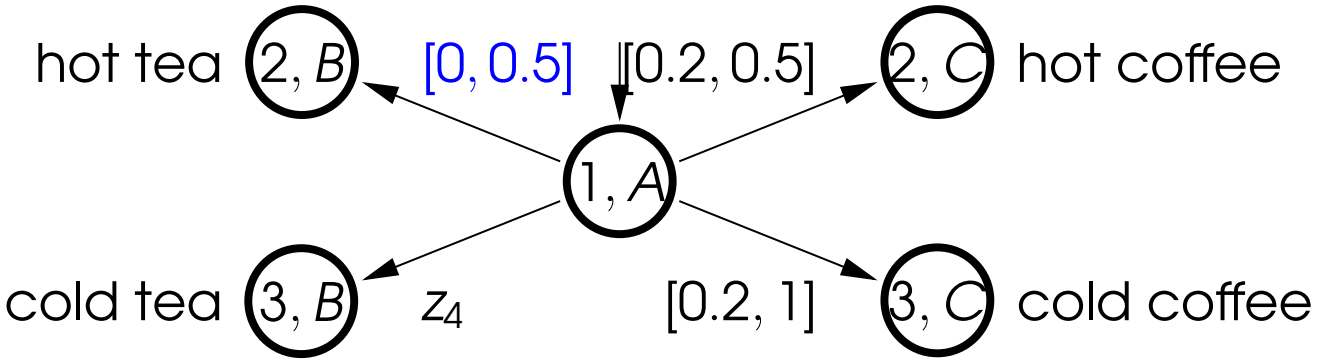


$$z_3 \in [0, 0.5] \cap [0, 1] \Rightarrow z_1 \in [0, 0.5]$$

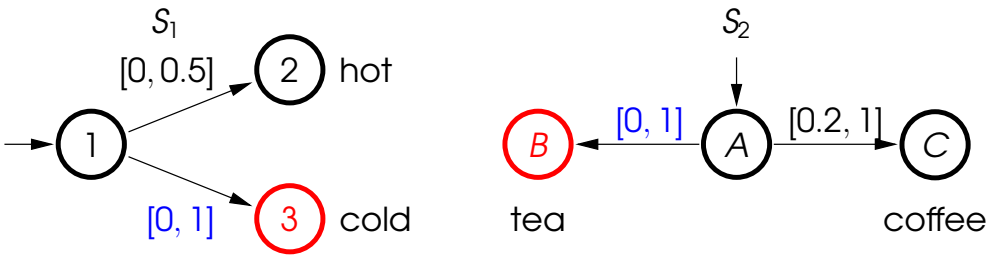
Absence of closure: An illustration (2)



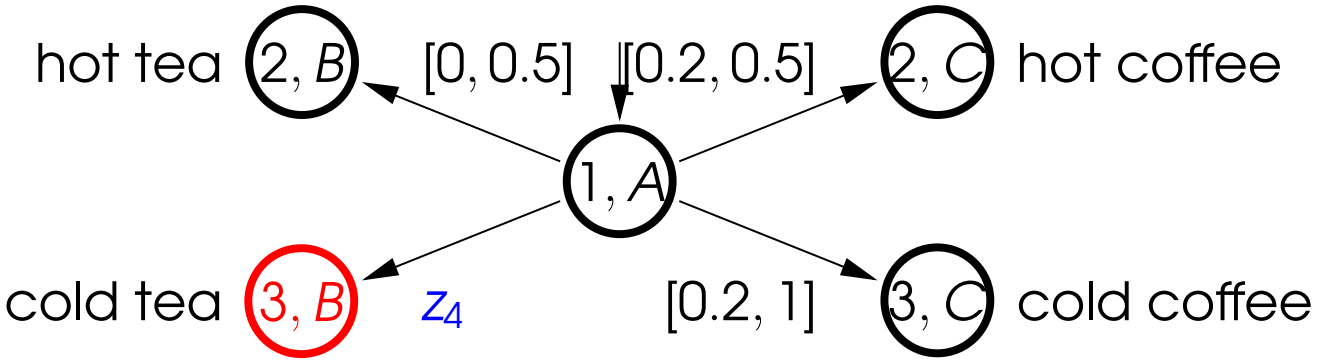
Conjunction of S_1 and S_2:



Absence of closure: An illustration (2)

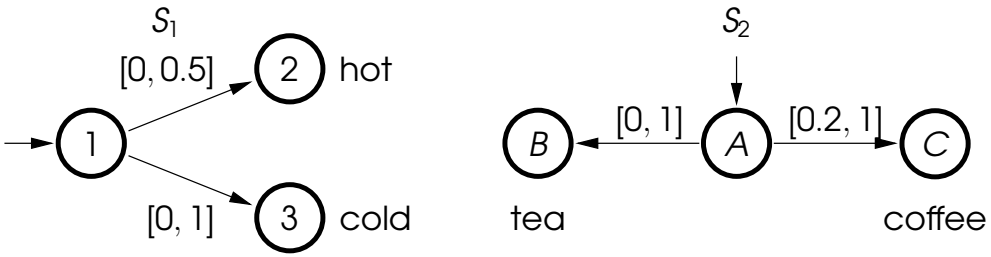


Conjunction of S_1 and S_2:

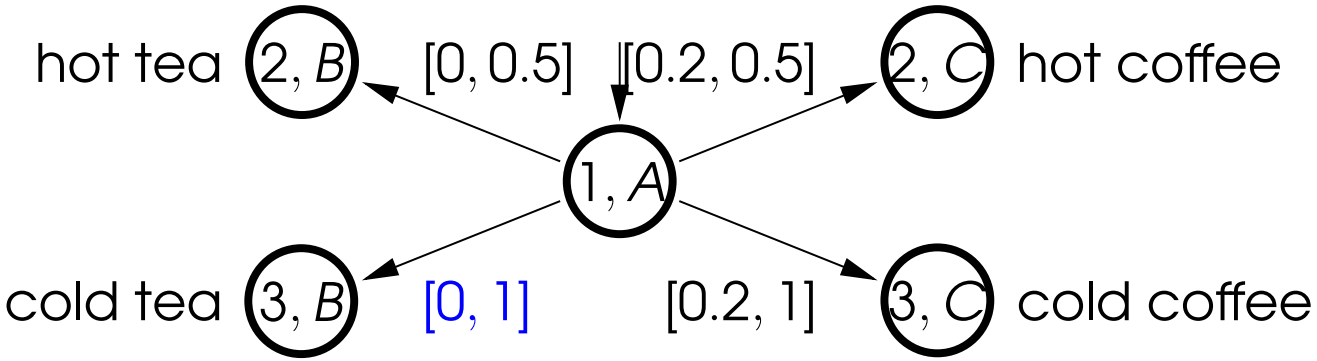


$$z_4 \in [0, 1] \cap [0, 1] \Rightarrow z_1 \in [0, 1]$$

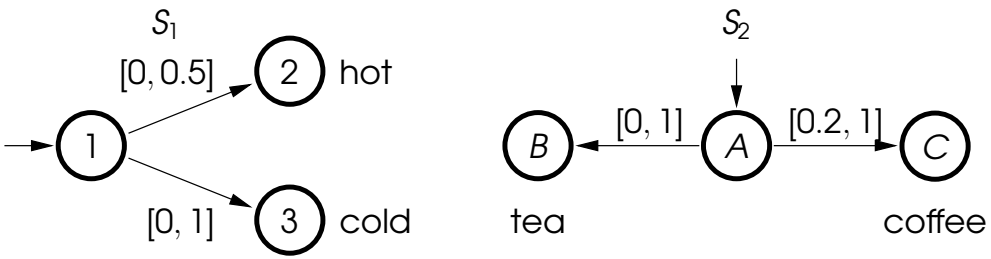
Absence of closure: An illustration (2)



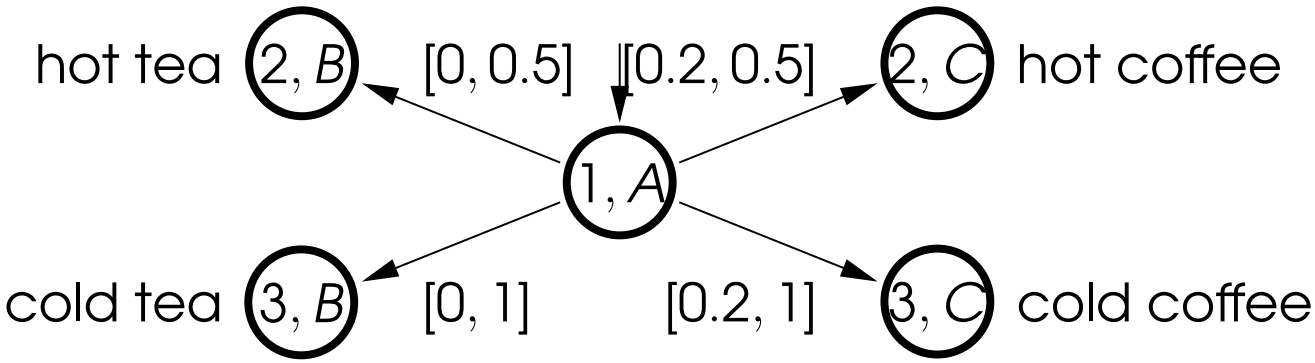
Conjunction of S_1 and S_2:



Absence of closure: An illustration (2)



Conjunction of S_1 and S_2:



$(z_1, z_2, z_3, z_4) = (0.3, 0.2, 0.3, 0.2)$ is an implementation, but prob. for reaching "hot", $z_1 + z_3 = 0.6$ violates S_1

Absence of closure: An illustration (3)

IMCs are not closed under compositional operations

- Conjunction
- Parallel Composition

Solution: Constraint Markov Chains

Outline

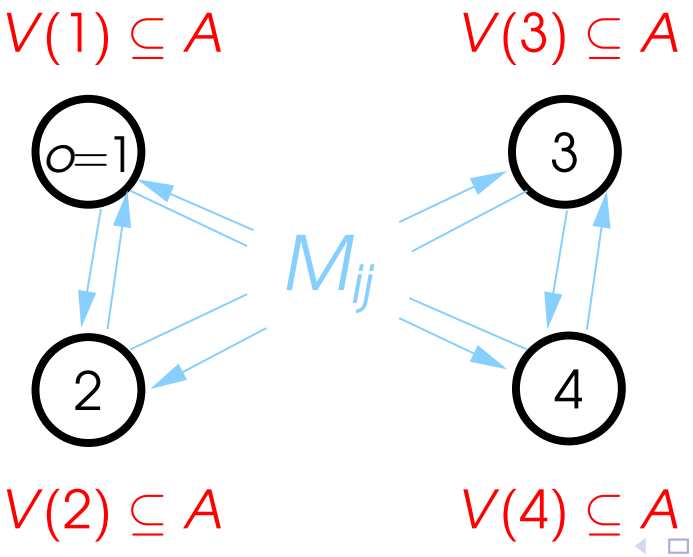
Constraint Markov Chains

- B. Delahaye, K.G. Larsen, A. Legay, M.L. Pedersen, A. Wasowski. New results for Constraint Markov Chains. In *Performance EVAluation*, 2012.
- B. Caillaud, B. Delahaye, K.G. Larsen, A. Legay, M.L. Pedersen, A. Wasowski. Constraint Markov Chains. In *Theoretical Computer Science*, 2011.
- B. Caillaud, B. Delahaye, K.G. Larsen, A. Legay, M.L. Pedersen, A. Wasowski. Compositional Design Methodology with Constraint Markov Chains. In *QEST, 7th International Conference on Quantitative Evaluation of SysTems*, Williamsburg, Virginia, United States of America, 2010.

Markov Chains (recap.)

$$(\{1,\ldots,n\}, o, M, A, V)$$

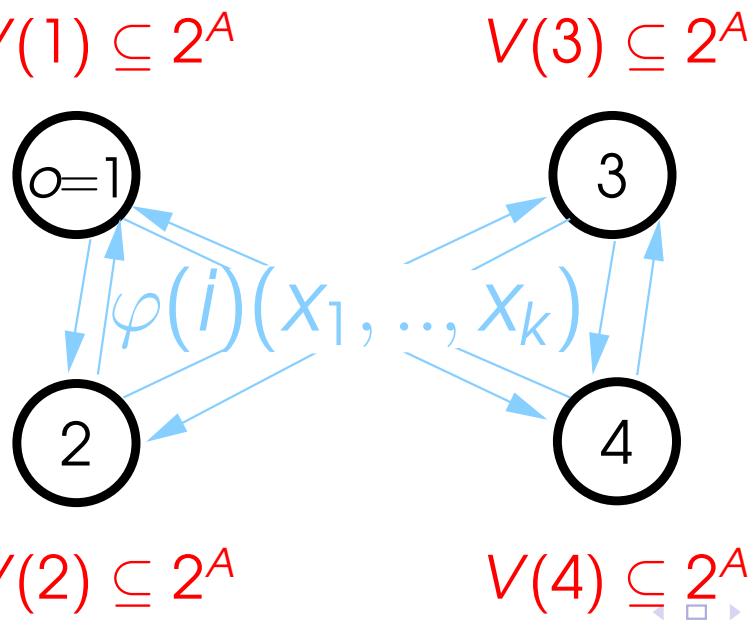
- states $\{1,\ldots,n\}$, o initial state,
- A is a set of atomic propositions, $V : \{1,\ldots,n\} \to 2^A$,
- $M \in [0,1]^{n\times n}$ is a probability transition matrix: $\sum_{j=1}^{n} M_{ij} = 1$ for $i = 1,\ldots,n$.



Constraint Markov Chains

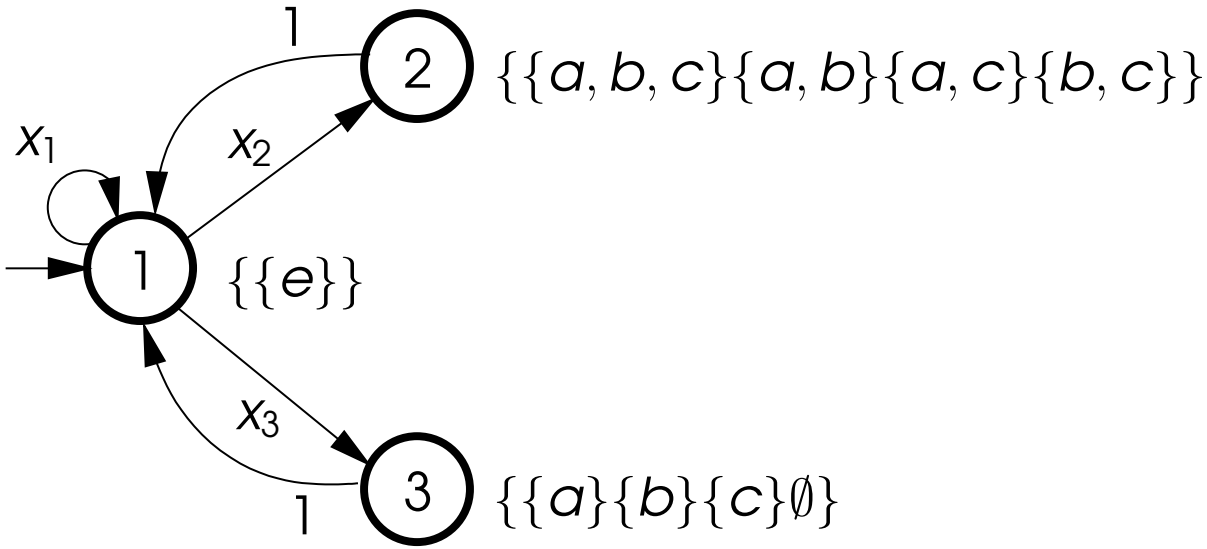
$$(\{1,\ldots,k\}, o, \varphi, A, V)$$

- states $\{1,\ldots,k\}$, o initial state
- A is a set of atomic propositions, $V : \{1,\ldots,k\} \to 2^{2^A}$
- $\varphi : \{1,\ldots,k\} \to [0,1]^k \to \{0,1\}$



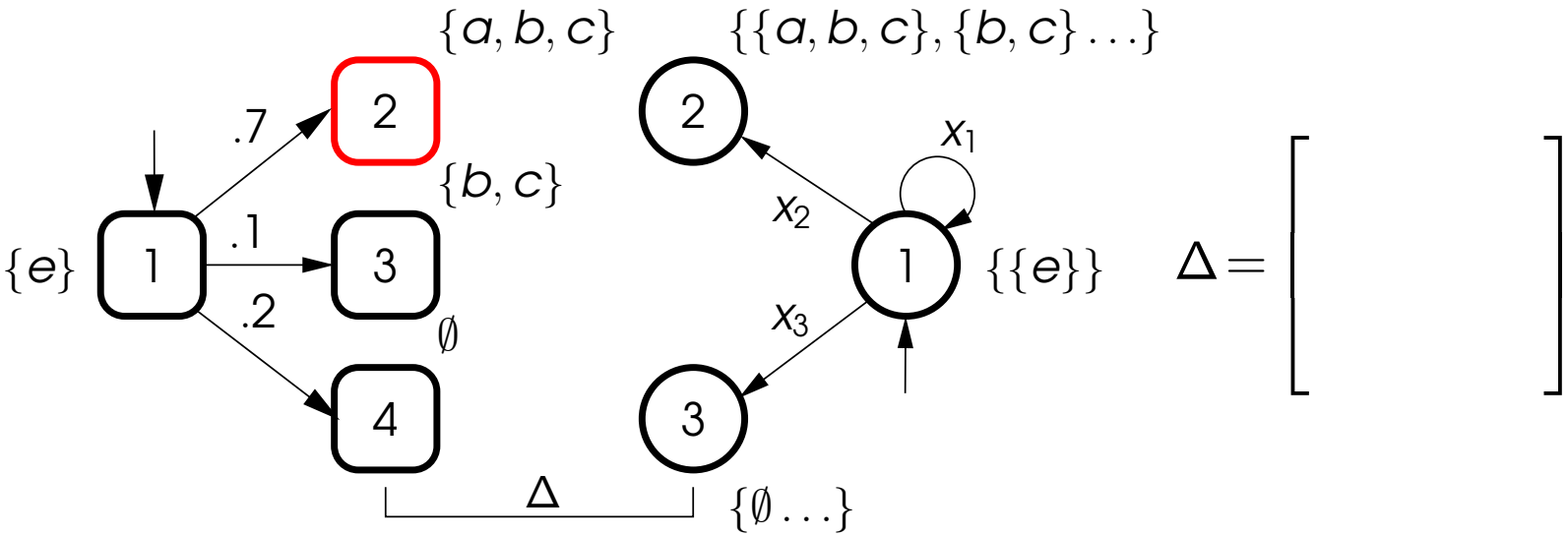
Constraint Markov Chain: An Example

$$(\{1,\ldots,k\}, o, \varphi, A, V)$$



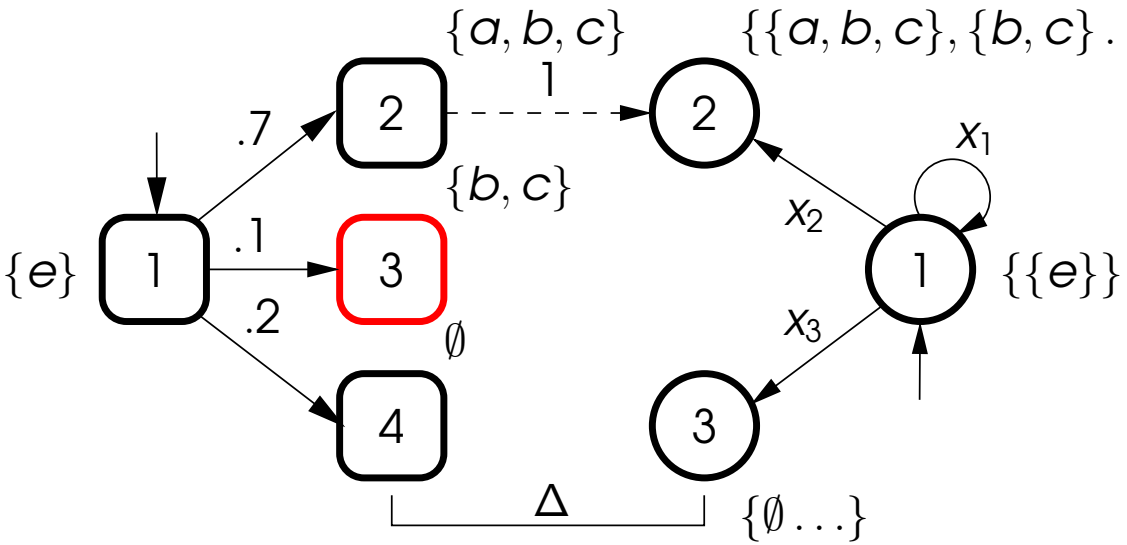
$$\varphi_1(1)(x_1, x_2, x_3) \equiv (x_1 = 0) \wedge (x_2 \geq 0.7) \wedge (x_2 + x_3 = 1)$$

Satisfaction Relation: An example



$$\varphi_1(1)(x_1, x_2, x_3) \equiv (x_1 = 0) \wedge (x_2 \geq 0.7) \wedge (x_2 + x_3 = 1)$$

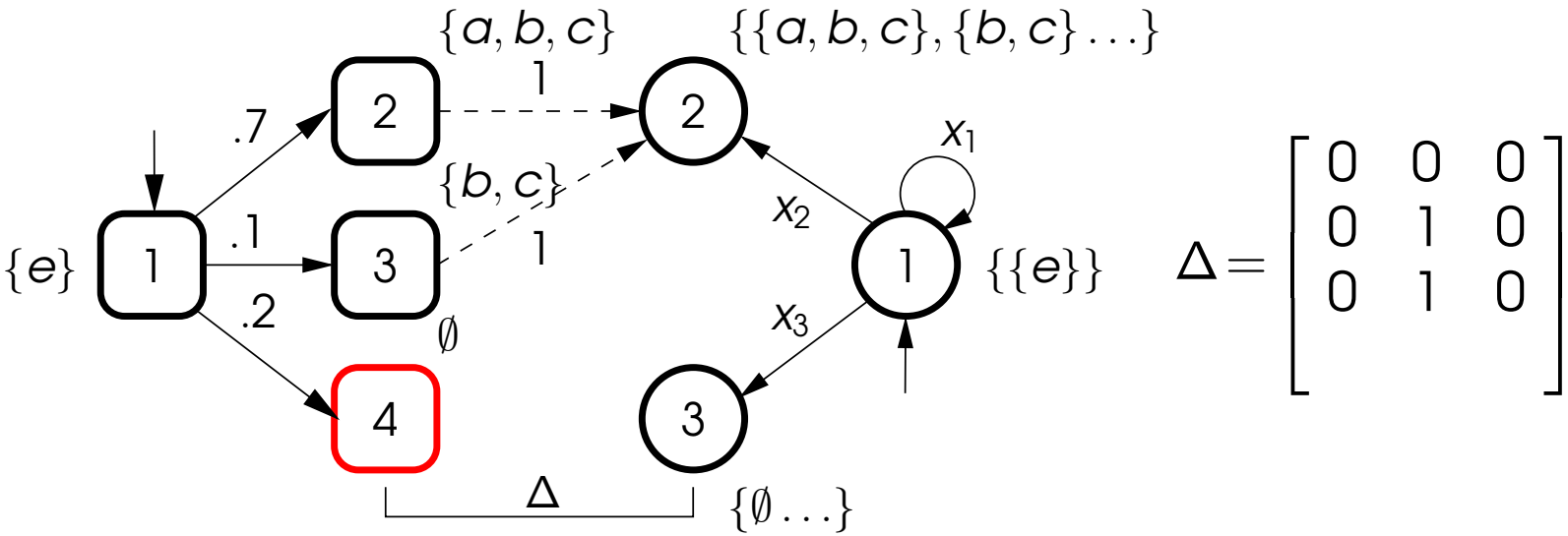
Satisfaction Relation: An example



$$\Delta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ & & \end{bmatrix}$$

$$\varphi_1(1)(x_1, x_2, x_3) \equiv (x_1 = 0) \wedge (x_2 \geq 0.7) \wedge (x_2 + x_3 = 1)$$

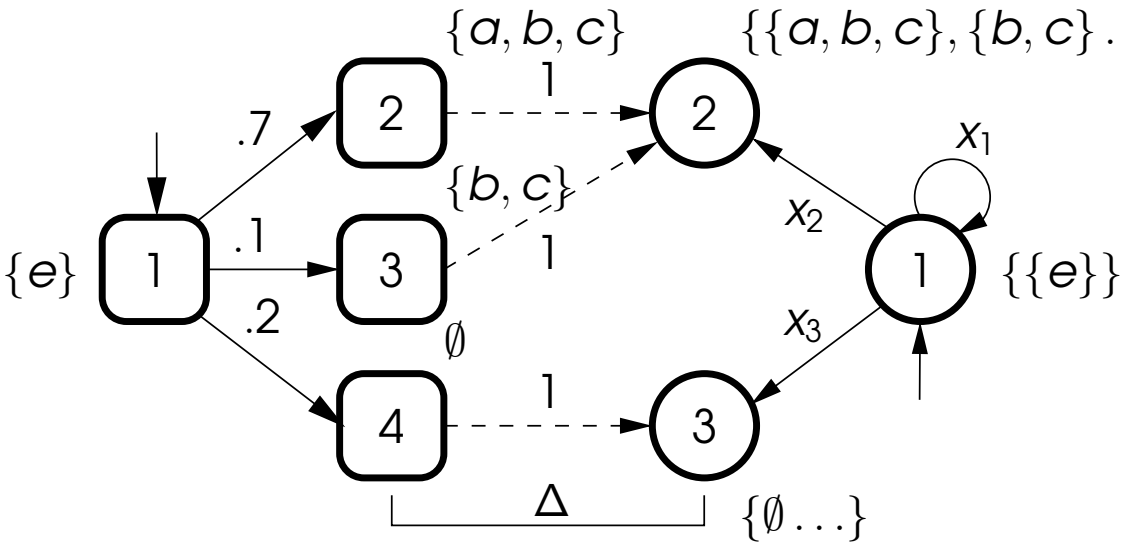
Satisfaction Relation: An example



$$\Delta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ & & \end{bmatrix}$$

$$\varphi_1(1)(x_1, x_2, x_3) \equiv (x_1 = 0) \wedge (x_2 \geq 0.7) \wedge (x_2 + x_3 = 1)$$

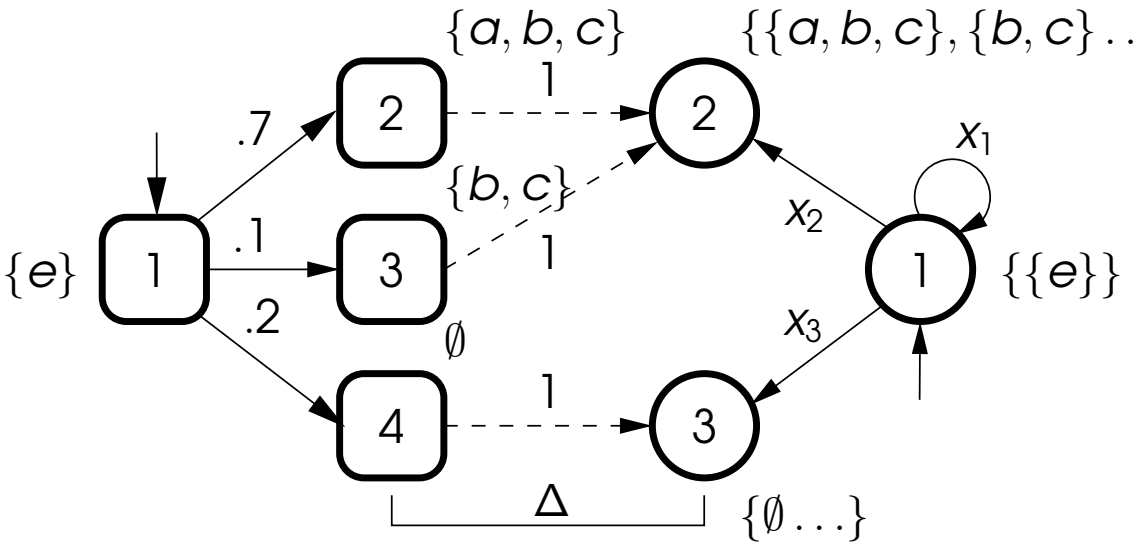
Satisfaction Relation: An example



$$\Delta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\varphi_1(1)(x_1, x_2, x_3) \equiv (x_1 = 0) \wedge (x_2 \geq 0.7) \wedge (x_2 + x_3 = 1)$$

Satisfaction Relation: An example

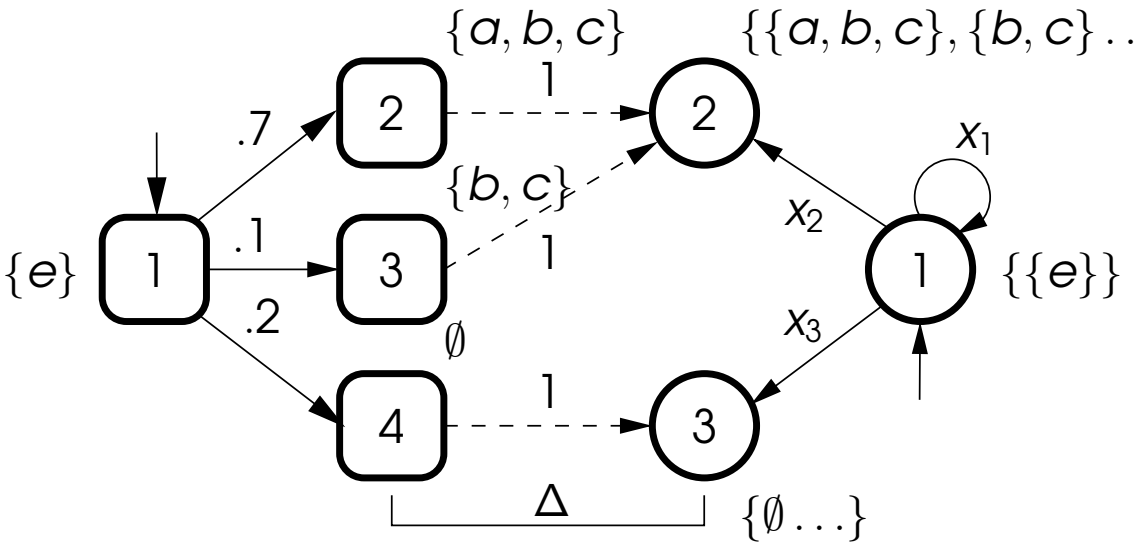


$$\Delta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\varphi_1(1)(x_1, x_2, x_3) \equiv (x_1 = 0) \wedge (x_2 \geq 0.7) \wedge (x_2 + x_3 = 1)$$

$$(0, 0.7, 0.1, 0.2) \times \Delta = (0, 0.8, 0.2)$$

Satisfaction Relation: An example



$$\Delta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\varphi_1(1)(x_1, x_2, x_3) \equiv (x_1 = 0) \wedge (x_2 \geq 0.7) \wedge (x_2 + x_3 = 1)$$

$$(0, 0.7, 0.1, 0.2) \times \Delta = (0, 0.8, 0.2)$$

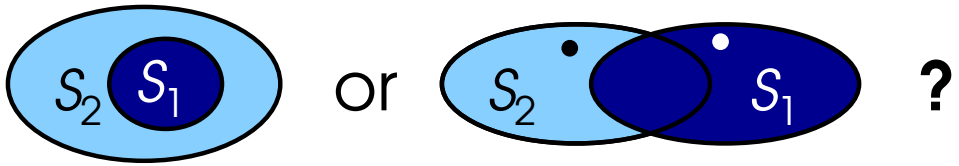
$$\Rightarrow \varphi_1(1)\left((0, 0.7, 0.1, 0.2) \times \Delta\right) \text{ holds}$$

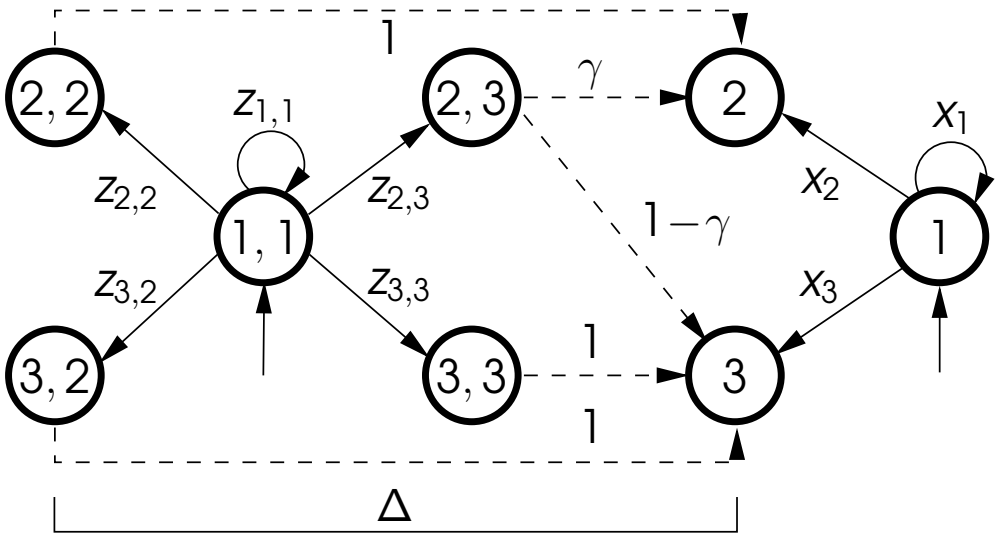
Satisfaction Relation: Formal Definition

Definition. Let $P = \langle \{1, \ldots, n\}, o_P, M, A, V_P \rangle$ be a MC and $S = \langle \{1, \ldots, k\}, o_S, \varphi, A, V_S \rangle$ be a CMC. Then $\mathcal{R} \subseteq \{1, \ldots, n\} \times \{1, \ldots, k\}$ is a *satisfaction relation* between states of P and S iff whenever $p \mathcal{R} u$, we have

1. $V_P(p) \in V_S(u)$, and
2. there exists a correspondence matrix $\Delta \in [0, 1]^{n \times k}$ such that
   - for all $1 \leq p' \leq n$ with $M_{pp'} \neq 0$, $\sum_{j=1}^{k} \Delta_{p'j} = 1$;
   - $\varphi(u)(M_p \times \Delta)$ holds, and if $\Delta_{p'u'} \neq 0$ then $p' \mathcal{R} u'$.

Weak Refinement: An Example





$$\Delta = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \gamma & 1-\gamma \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\gamma = \frac{0.7 - z_{2,2}}{z_{2,3}} \text{ if } z_{2,2} \leq 0.7 \text{ and } \gamma = \frac{0.8 - z_{2,2}}{z_{2,3}} \text{ otherwise}$$

Weak Refinement: Formal Definition

Definition. Let $S_1 = \langle \{1, \ldots, k_1\}, o_1, \varphi_1, A, V_1 \rangle$ and $S_2 = \langle \{1, \ldots, k_2\}, o_2, \varphi_2, A, V_2 \rangle$ be CMCs. The relation $\mathcal{R} \subseteq \{1, \ldots, k_1\} \times \{1, \ldots, k_2\}$ is a *weak refinement relation* iff whenever $v \mathcal{R} u$, we have

1. $V_1(v) \subseteq V_2(u)$ and
2. **for any distribution** $x \in [0, 1]^{1 \times k_1}$ **satisfying** $\varphi_1(v)(x)$, there exists a matrix $\Delta \in [0, 1]^{k_1 \times k_2}$ such that
   - for all S_1 states $1 \leq i \leq k_1$, $x_i \neq 0 \implies \sum_{j=1}^{k_2} \Delta_{ij} = 1$;
   - $\varphi_2(u)(x \times \Delta)$ holds and
   - $\Delta_{v'u'} \neq 0 \Rightarrow v' \mathcal{R} u'$.

Weak Refinement: main results

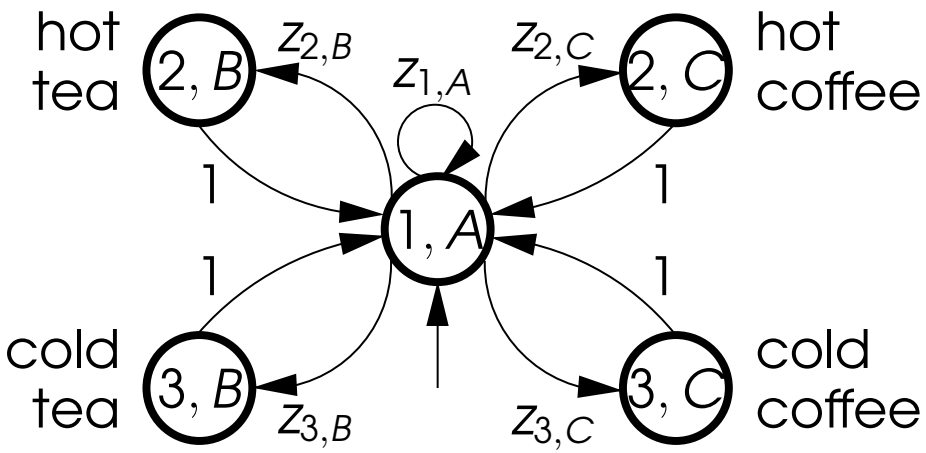
- Sound: $S_1 \preceq S_2$ implies $[\![S_1]\!] \subseteq [\![S_2]\!]$

Theorem

Weak Refinement is complete for deterministic systems: for S_1, S_2 $[\![S_1]\!] \subseteq [\![S_2]\!]$ implies $S_1 \preceq S_2$

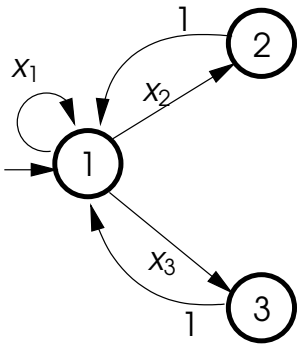
CMC S is *deterministic* iff $(\varphi(i)(x) \wedge (x_u \neq 0))$ and $(\varphi(i)(y) \wedge (y_v \neq 0))$, then $V(u) \cap V(v) = \emptyset$.

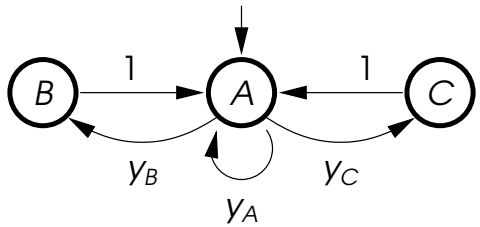
Conjunction: An Example



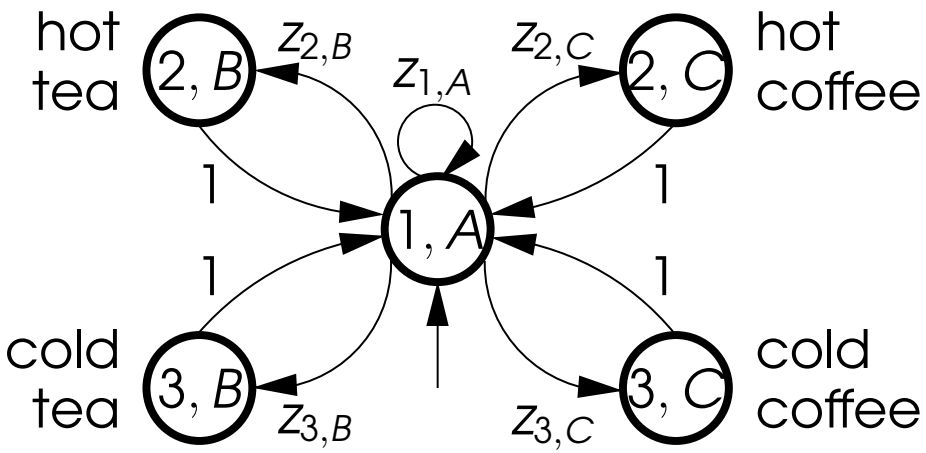
$$\varphi_1(1)(x) \equiv (x_1 = 0) \wedge (x_2 \leq 0.5) \wedge (x_2 + x_3 = 1)$$

$$\varphi_2(A)(y) \equiv (y_A = 0) \wedge (y_C \geq 0.2) \wedge (y_B + y_C = 1)$$





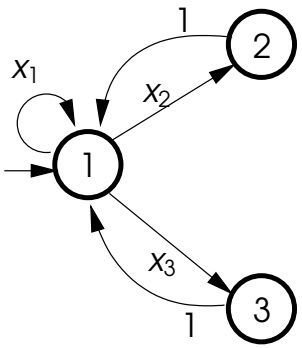
Conjunction: An Example

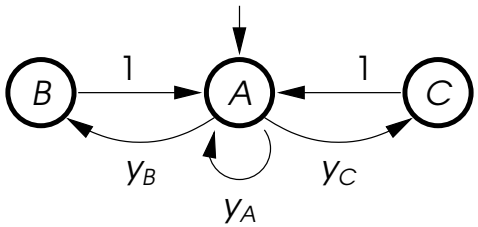


$$\varphi_1(1)(x) \equiv (x_1 = 0) \wedge (x_2 \leq 0.5) \wedge (x_2 + x_3 = 1)$$

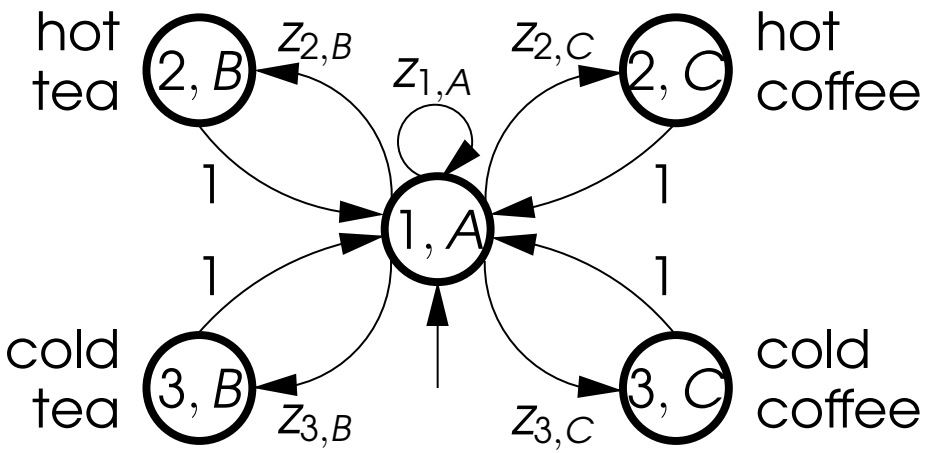
$$\varphi_2(A)(y) \equiv (y_A = 0) \wedge (y_C \geq 0.2) \wedge (y_B + y_C = 1)$$

$(z_{1,A}, z_{1,B}, \ldots, z_{3,C})$ is valid iff





Conjunction: An Example

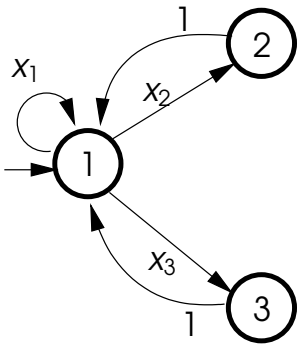


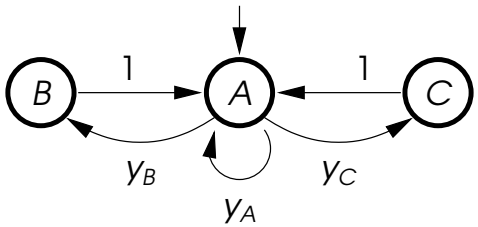
$$\varphi_1(1)(x) \equiv (x_1 = 0) \wedge (x_2 \leq 0.5) \wedge (x_2 + x_3 = 1)$$

$$\varphi_2(A)(y) \equiv (y_A = 0) \wedge (y_C \geq 0.2) \wedge (y_B + y_C = 1)$$

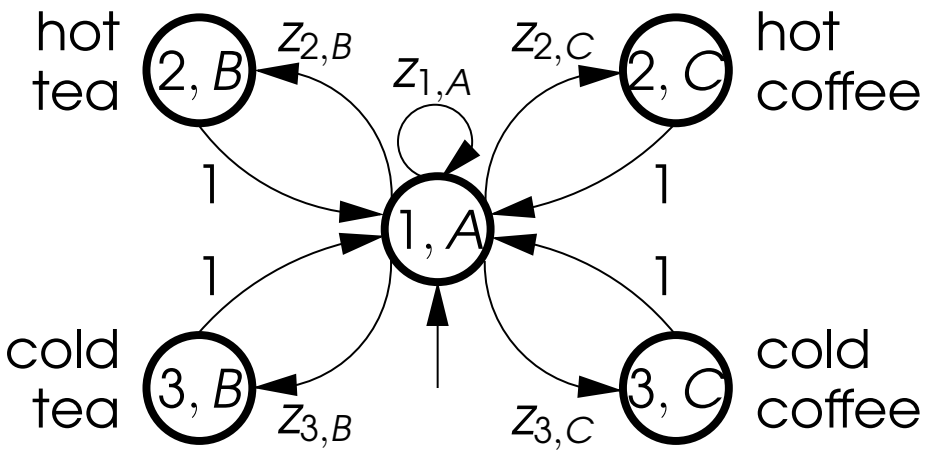
$(z_{1,A}, z_{1,B}, \ldots, z_{3,C})$ is valid iff

$$z_{1,A} + z_{1,B} + z_{1,C} = 0$$





Conjunction: An Example



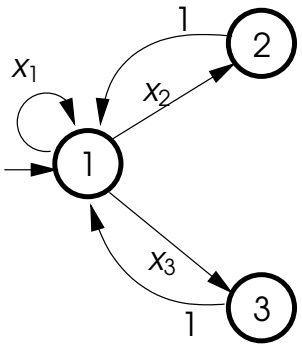
$$\varphi_1(1)(x) \equiv (x_1 = 0) \wedge (x_2 \leq 0.5) \wedge (x_2 + x_3 = 1)$$

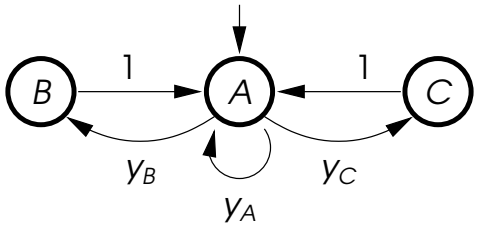
$$\varphi_2(A)(y) \equiv (y_A = 0) \wedge (y_C \geq 0.2) \wedge (y_B + y_C = 1)$$

$(z_{1,A}, z_{1,B}, \ldots, z_{3,C})$ is valid iff

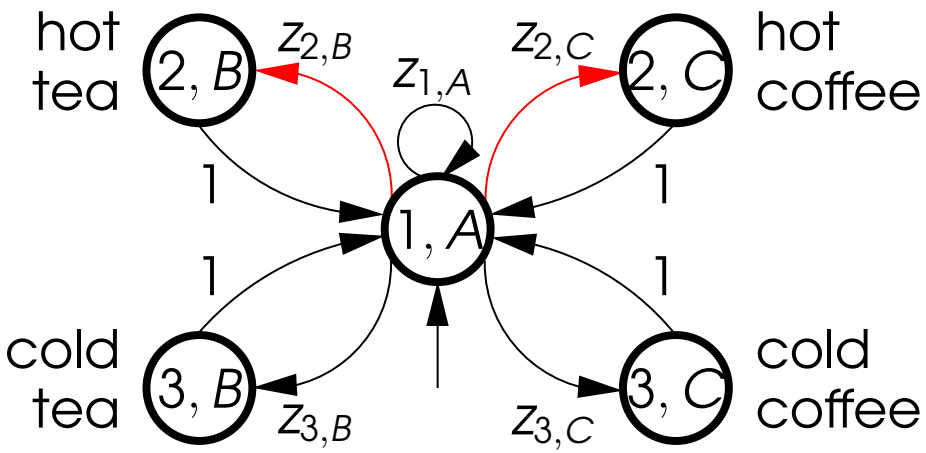
$$z_{1,A} + z_{1,B} + z_{1,C} = 0$$

$$z_{1,A} + z_{2,A} + z_{3,A} = 0$$





Conjunction: An Example



$$\varphi_1(1)(x) \equiv (x_1 = 0) \wedge (x_2 \leq 0.5) \wedge (x_2 + x_3 = 1)$$

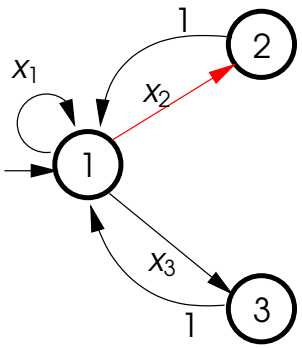
$$\varphi_2(A)(y) \equiv (y_A = 0) \wedge (y_C \geq 0.2) \wedge (y_B + y_C = 1)$$

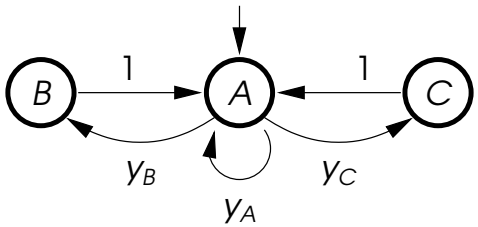
$(z_{1,A}, z_{1,B}, \ldots, z_{3,C})$ is valid iff

$$z_{1,A} + z_{1,B} + z_{1,C} = 0$$

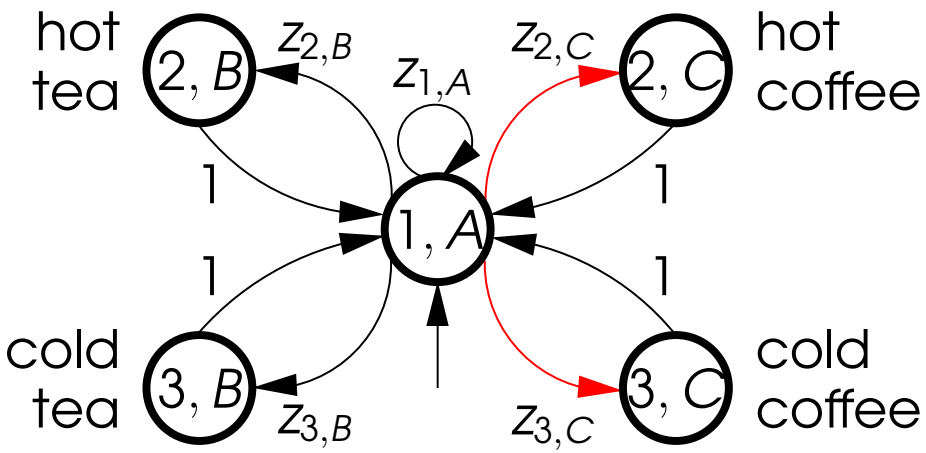
$$z_{1,A} + z_{2,A} + z_{3,A} = 0$$

$$z_{2,A} + z_{2,B} + z_{2,C} \leq 0.5$$





Conjunction: An Example

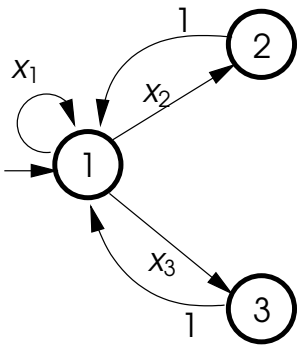


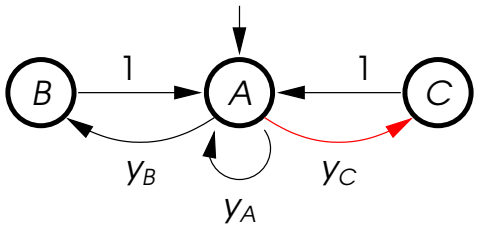
$$\varphi_1(1)(x) \equiv (x_1 = 0) \wedge (x_2 \leq 0.5) \wedge (x_2 + x_3 = 1)$$

$$\varphi_2(A)(y) \equiv (y_A = 0) \wedge (y_C \geq 0.2) \wedge (y_B + y_C = 1)$$

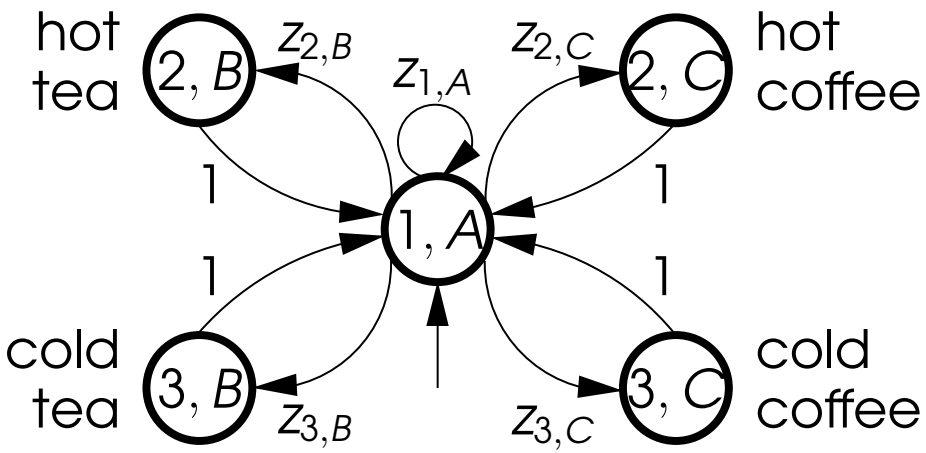
$(z_{1,A}, z_{1,B}, \ldots, z_{3,C})$ is valid iff

$$z_{1,A} + z_{1,B} + z_{1,C} = 0$$
$$z_{1,A} + z_{2,A} + z_{3,A} = 0$$
$$z_{2,A} + z_{2,B} + z_{2,C} \leq 0.5$$
$$z_{1,C} + z_{2,C} + z_{3,C} \geq 0.2$$





Conjunction: An Example

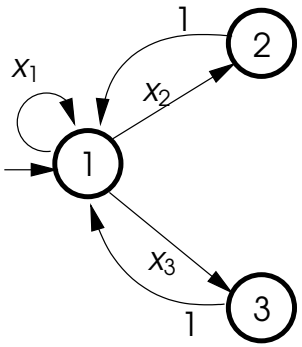


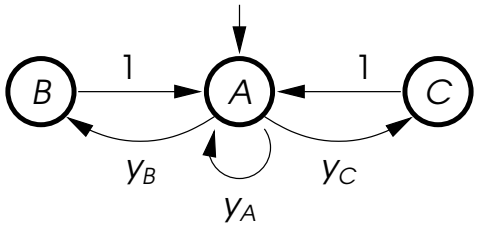
$$\varphi_1(1)(x) \equiv (x_1 = 0) \wedge (x_2 \leq 0.5) \wedge (x_2 + x_3 = 1)$$

$$\varphi_2(A)(y) \equiv (y_A = 0) \wedge (y_C \geq 0.2) \wedge (y_B + y_C = 1)$$

$(z_{1,A}, z_{1,B}, \ldots, z_{3,C})$ is valid iff

$$z_{1,A} + z_{1,B} + z_{1,C} = 0$$
$$z_{1,A} + z_{2,A} + z_{3,A} = 0$$
$$z_{2,A} + z_{2,B} + z_{2,C} \leq 0.5$$
$$z_{1,C} + z_{2,C} + z_{3,C} \geq 0.2$$
$$\sum_{i,j} z_{i,j} = 1$$

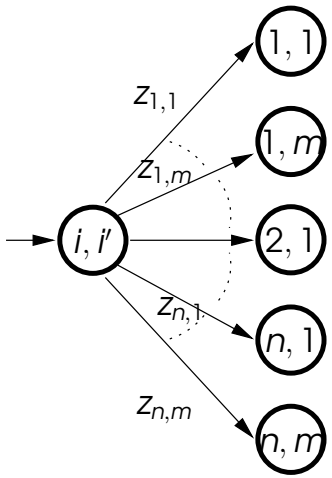




Conjunction: Formal Definition

Conjunction: Formal Definition

$$S_1 \wedge S_2 = \langle Q_1 \times Q_2, (i, i'), \varphi', A, V' \rangle$$



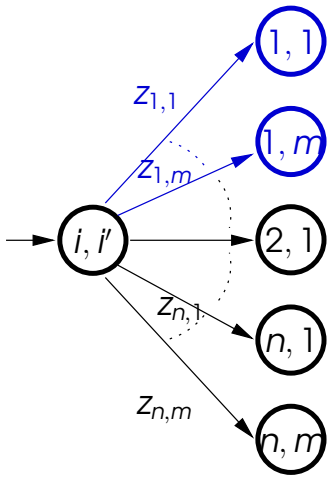
$$\varphi'((u, v))(x_{1,1}, x_{1,2}, \ldots, x_{2,1}, \ldots, x_{n,m}) \equiv$$

$$\varphi_1(u)(\qquad)$$

$$\wedge\, \varphi_2(v)(\qquad)$$

Conjunction: Formal Definition

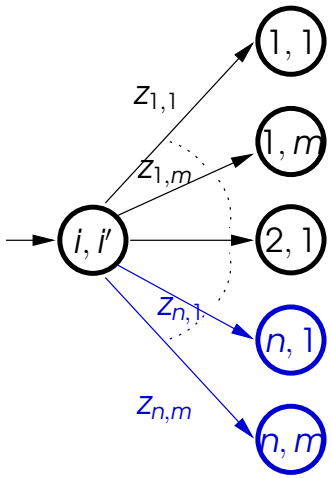
$$S_1 \wedge S_2 = \langle Q_1 \times Q_2, (i, i'), \varphi', A, V' \rangle$$



$$\varphi'((u,v))(x_{1,1}, x_{1,2}, \ldots, x_{2,1}, \ldots, x_{n,m}) \equiv$$
$$\varphi_1(u)(\sum_{j=1}^{m} x_{1,j}, \quad)$$
$$\wedge\, \varphi_2(v)(\quad)$$

Conjunction: Formal Definition

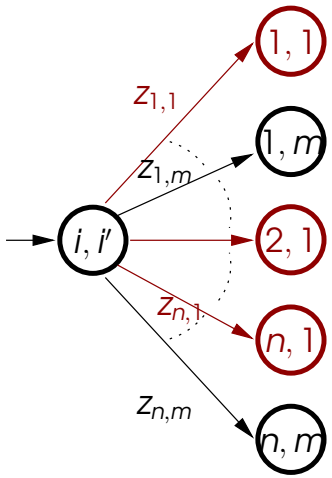
$$S_1 \wedge S_2 = \langle Q_1 \times Q_2, (i, i'), \varphi', A, V' \rangle$$



$$\varphi'((u,v))(x_{1,1}, x_{1,2}, \ldots, x_{2,1}, \ldots, x_{n,m}) \equiv$$
$$\varphi_1(u)(\sum_{j=1}^{m} x_{1,j}, \ldots, \sum_{j=1}^{m} x_{n,j})$$
$$\wedge\, \varphi_2(v)(\qquad\qquad\qquad\qquad)$$

Conjunction: Formal Definition

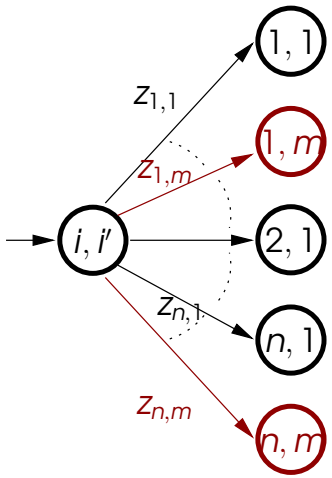
$$S_1 \wedge S_2 = \langle Q_1 \times Q_2, (i, i'), \varphi', A, V' \rangle$$



$$\varphi'((u,v))(x_{1,1}, x_{1,2}, \ldots, x_{2,1}, \ldots, x_{n,m}) \equiv$$
$$\varphi_1(u)\Big(\sum_{j=1}^{m} x_{1,j}, \ldots, \sum_{j=1}^{m} x_{n,j}\Big)$$
$$\wedge\, \varphi_2(v)\Big(\sum_{i=1}^{n} x_{i,1}, \qquad\qquad\Big)$$

Conjunction: Formal Definition

$$S_1 \wedge S_2 = \langle Q_1 \times Q_2, (i, i'), \varphi', A, V' \rangle$$



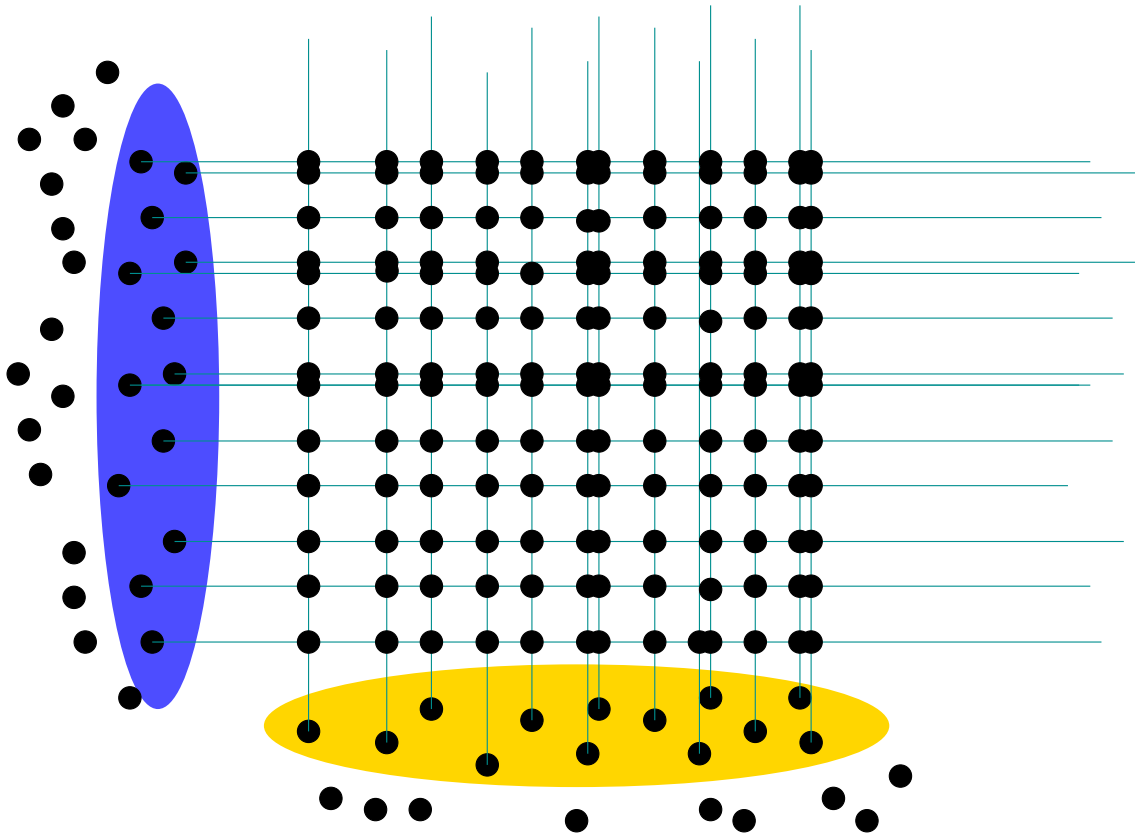
$$\varphi'((u,v))(x_{1,1}, x_{1,2}, \ldots, x_{2,1}, \ldots, x_{n,m}) \equiv$$
$$\varphi_1(u)(\sum_{j=1}^{m} x_{1,j}, \ldots, \sum_{j=1}^{m} x_{n,j})$$
$$\wedge\, \varphi_2(v)(\sum_{i=1}^{n} x_{i,1}, \ldots, \sum_{i=1}^{n} x_{i,m})$$

Conjunction

- Conjunction is equal to intersection of sets of implementations

- Obtain linear constraints, even when composing linear (including interval) constraints

Independent Parallel Composition



- Essentially a product
- State and transition constraints conjoined

Properties of Parallel Composition

- Weak refinement is a precongruence with respect to parallel composition:

Theorem:
$S'_1 \preceq S_1 \ \wedge \ S'_2 \preceq S_2$ implies $S'_1 \| S'_2 \ \preceq \ S_1 \| S_2$

Properties of Parallel Composition

- Weak refinement is a precongruence with respect to parallel composition:

Theorem:
$S_1' \preceq S_1 \wedge S_2' \preceq S_2$ implies $S_1' \parallel S_2' \preceq S_1 \parallel S_2$

- Gives rise to polynomial constraint $\varphi(z)$, due to multiplications of transition probabilities
  - $\exists x_1, \ldots, x_{k_1}, y_1, \ldots, y_{k_2} \in [0,1]$ such that $z_{i,j} = x_i \cdot y_j$, $z_{i,j} \in [0,1]$ and $\varphi_1(u)(x_1, \ldots, x_{k_1}) = \varphi_2(v)(y_1, \ldots, y_{k_2}) = 1$

Properties of Parallel Composition

- Weak refinement is a precongruence with respect to parallel composition:

Theorem:
$S_1' \preceq S_1 \ \wedge \ S_2' \preceq S_2$ implies $S_1' \parallel S_2' \ \preceq \ S_1 \parallel S_2$

- Gives rise to polynomial constraint $\varphi(z)$, due to multiplications of transition probabilities
  - $\exists x_1, \ldots, x_{k_1}, y_1, \ldots, y_{k_2} \in [0,1]$ such that $z_{i,j} = x_i \cdot y_j$, $z_{i,j} \in [0,1]$ and $\varphi_1(u)(x_1, \ldots, x_{k_1}) = \varphi_2(v)(y_1, \ldots, y_{k_2}) = 1$
- Non synchronizing

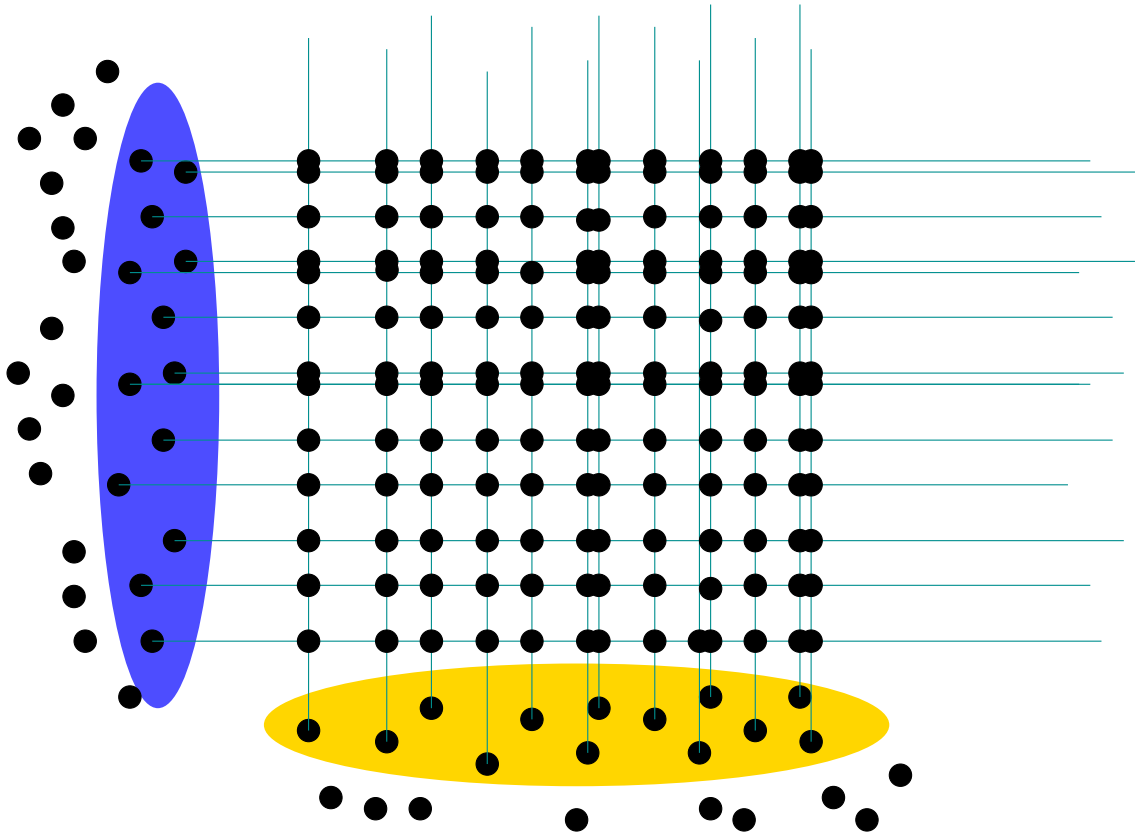
Properties of Parallel Composition

- Weak refinement is a precongruence with respect to parallel composition:

Theorem:
$S_1' \preceq S_1 \ \wedge \ S_2' \preceq S_2$ implies $S_1' \parallel S_2' \ \preceq \ S_1 \parallel S_2$

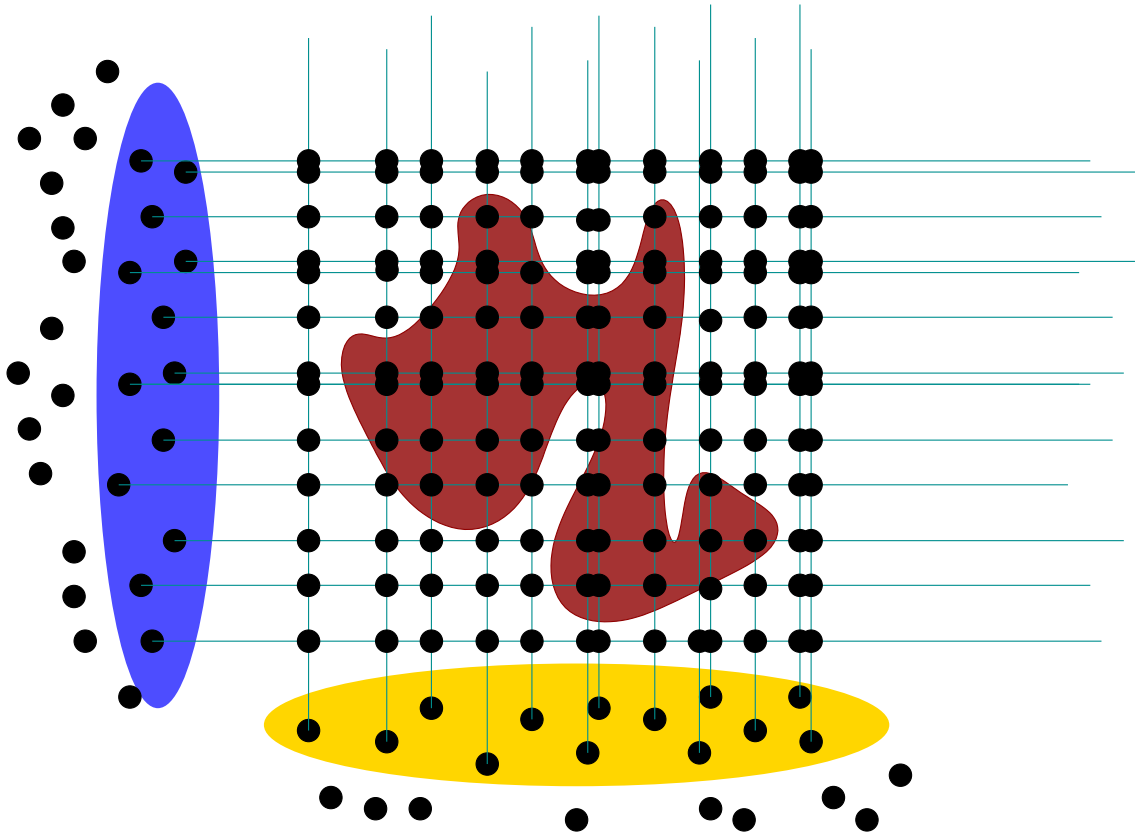
- Gives rise to polynomial constraint $\varphi(z)$, due to multiplications of transition probabilities
  - $\exists x_1, \ldots, x_{k_1}, y_1, \ldots, y_{k_2} \in [0,1]$ such that $z_{i,j} = x_i \cdot y_j$, $z_{i,j} \in [0,1]$ and $\varphi_1(u)(x_1, \ldots, x_{k_1}) = \varphi_2(v)(y_1, \ldots, y_{k_2}) = 1$
- Non synchronizing
- Preserves determinism

Synchronized Parallel Composition



$S_1 || S_2$

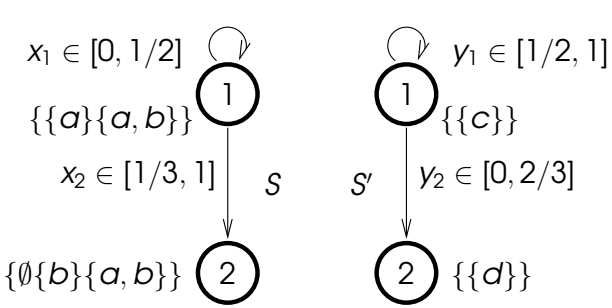
Synchronized Parallel Composition

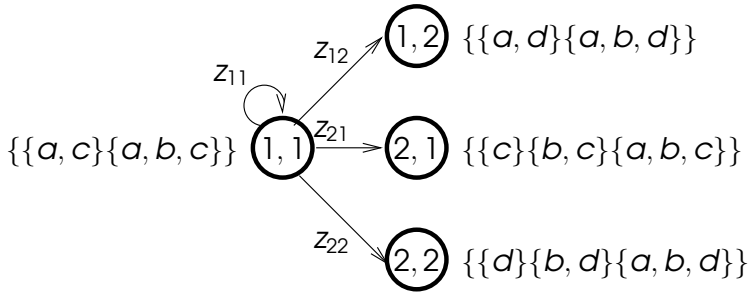


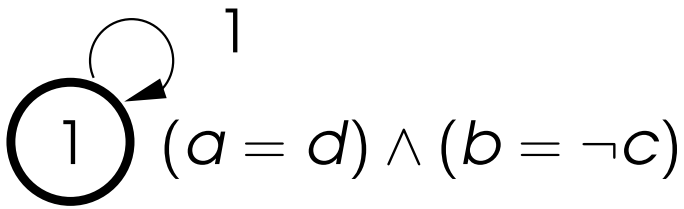
$$(S_1 || S_2) \wedge \text{Sync}$$

Synchronized Parallel Composition

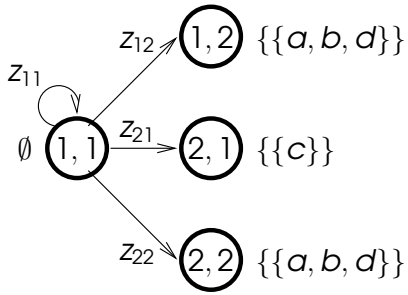
A synchronizer





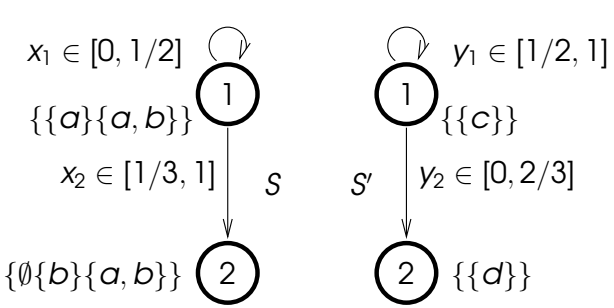


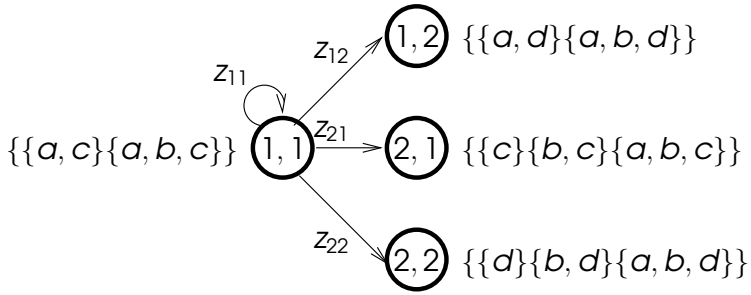
Sync

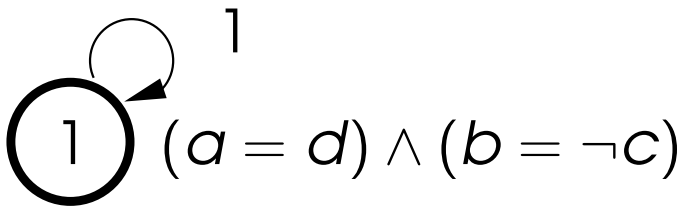


Synchronized Parallel Composition

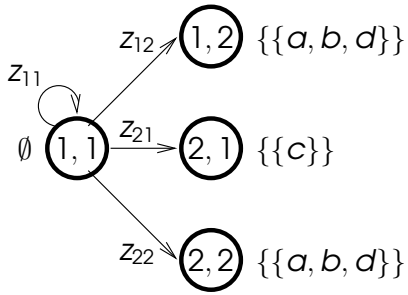
A synchronizer







Sync

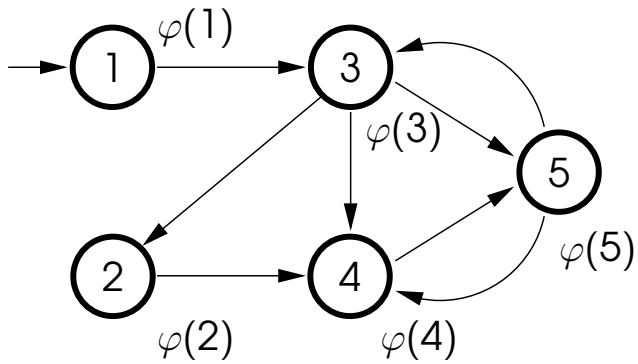


$$\left[((S_1 \parallel S_2) \wedge \mathrm{Sync}_{12}) \parallel S_3\right] \wedge \mathrm{Sync}_{123} = (S_1 \parallel S_2 \parallel S_3) \wedge S_{123}$$

Abstractions

Two notions of abstraction:

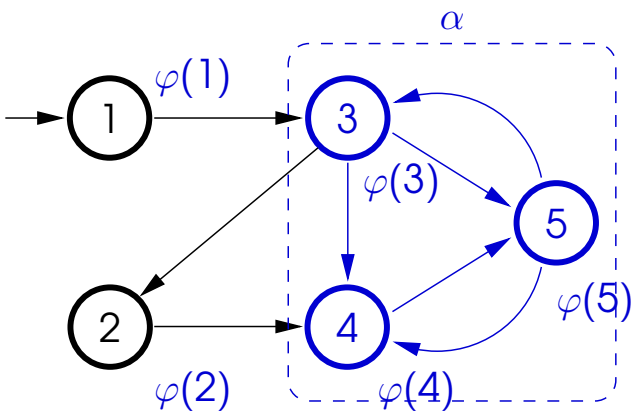
1. Grouping sets of state using a function α:
   - Linear transformation of the constraints
   - New structure



Abstractions

Two notions of abstraction:

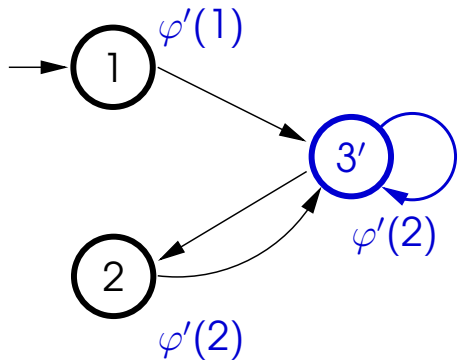
1. Grouping sets of state using a function α:
   - Linear transformation of the constraints
   - New structure



Abstractions

Two notions of abstraction:

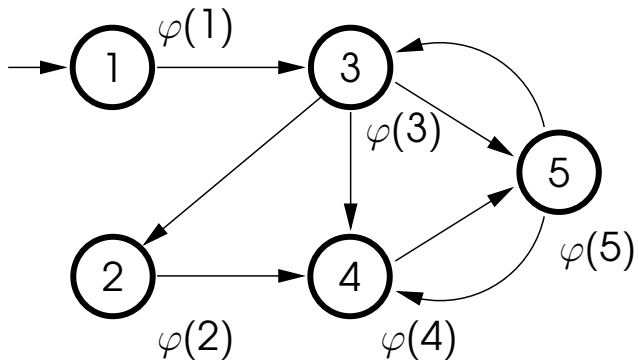
1. Grouping sets of state using a function α:
   - Linear transformation of the constraints
   - New structure



Abstractions

Two notions of abstraction:

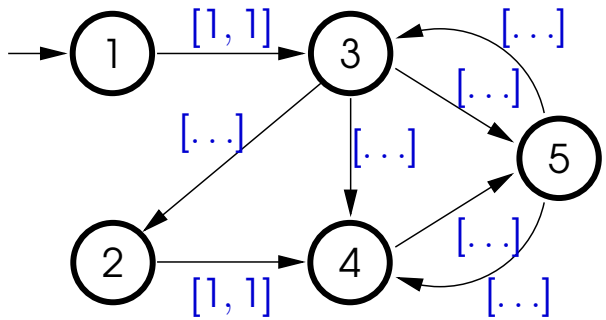
1. Grouping sets of state using a function α:
2. Abstraction to IMCs:
   - New Interval Constraints
   - Same structure



Abstractions

Two notions of abstraction:

1. Grouping sets of state using a function α:
2. Abstraction to IMCs:
   - New Interval Constraints
   - Same structure



Complexity

Complexity of solving the constraints: C
Total number of states: N
Upper Bounds:

- Consistency checking: $O(N^2 \times C)$
- Refinement: $O(N^2 \times C)$
- Determinization: Polynomial in N times C

Lower bound for thorough refinement: EXPTIME Hard (reduction from MTS)

Outline

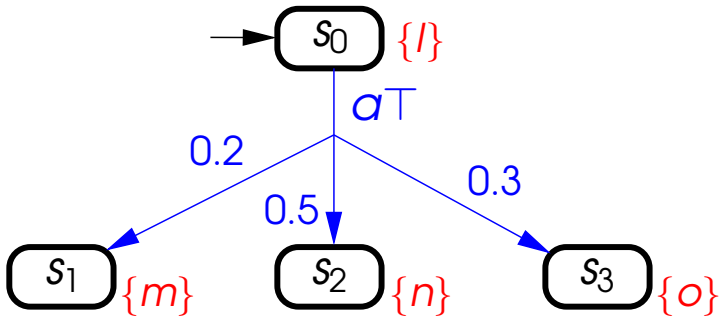
Abstract Probabilistic Automata

- B. Delahaye, K.G. Larsen, A. Legay, M.L. Pedersen, A. Wasowski. Stuttering in Abstract Probabilistic Automata. In *NWPT, 23rd Nordic Workshop on Programming Theory*, Västerås, Sweden, 2011.
- B. Delahaye, K.G. Larsen, A. Legay, M.L. Pedersen, A. Wasowski. APAC: a tool for reasoning about Abstract Probabilistic Automata (to appear). In *QEST, 8th International Conference on Quantitative Evaluation of SysTems*, Aachen, Germany, 2011.
- B. Delahaye, J.-P. Katoen, K.G. Larsen, A. Legay, M.L. Pedersen, F. Sher, A. Wasowski. New Results on Abstract Probabilistic Automata. In *ACSD, 11th International Conference on Application of Concurrency to System Design*, Newcastle, United Kingdom, 2011.
- B. Delahaye, J.-P. Katoen, K.G. Larsen, A. Legay, M.L. Pedersen, F. Sher, A. Wasowski. Abstract Probabilistic Automata. In *VMCAI, 12th International Conference on Verification, Model Checking, and Abstract Interpretation*, Austin, Texas, United States of America, 2011.

Probabilistic Automata

$$P = (S, A, L, AP, V, s_0)$$

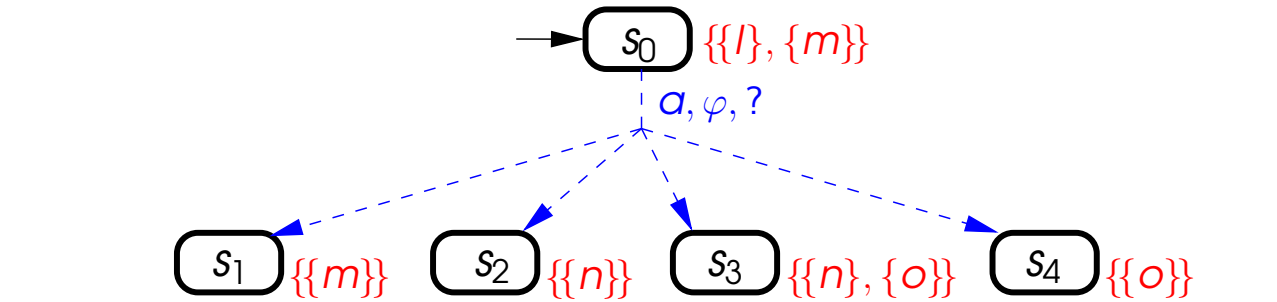
- states S, s_0 initial state,
- $L : S \times A \times Dist(S) \to \{\bot, \top\}$ is a two-valued transition function,
- A is a set of actions,
- AP is a set of atomic propositions, $V : S \to 2^{AP}$,



Abstract Probabilistic Automata

$$N = (S, A, L, AP, V, s_0)$$

- states S, s_0 initial state,
- $L : S \times A \times \mathbf{C(S)} \to \{\bot, ?, \top\}$ is a **three-valued** transition function,
- A is a set of actions,
- AP is a set of atomic propositions, $V : S \to \mathbf{2^{2^{AP}}}$,



$$\mu \in Sat(\varphi) \iff (\mu(s_1) + \mu(s_2) \geq 0.7) \wedge (\mu(s_3) + \mu(s_4) \geq 0.2)$$

Satisfaction

- Mix between satisfaction for MTS and satisfaction for CMCs
- New notation: $\mu \Subset_{\mathcal{R}} \varphi$ iff. $\exists \delta$ corresp. st. $\mu \times \delta \in \mathit{Sat}(\varphi)$...

Definition. $P = (S_P, A, L_P, AP, V_P, s_0^P)$ PA, $N = (S, A, L, AP, V, s_0)$ APA. $\mathcal{R} \subseteq S_P \times S$ is a sat rel iff. whenever $p \mathcal{R} s$,

- $V_P(p) \in V(s)$,
- $\forall a \in A, \mu \in \mathit{Dist}(S_P), L_P(p, a, \mu) = \top \implies \exists \varphi \in C(S), L(s, a, \varphi) \neq \bot$ and $\mu \Subset_{\mathcal{R}} \varphi$, and
- $\forall a \in A, \varphi \in C(S), L(s, a, \varphi) = \top \implies \exists \mu \in \mathit{Dist}(S_P), L_P(p, a, \mu) = \top$ and $\mu \Subset_{\mathcal{R}} \varphi$.

Refinement

- Mix of refinement for MTS and refinement for CMCs
- 3 syntactic definitions

Refinement

- Mix of refinement for MTS and refinement for CMCs
- 3 syntactic definitions

Definition. $N_1 = (S_1, A, L_1, AP, V_1, s_0^1)$, $N_2 = (S_2, A, L_2, AP, V_2, s_0^2)$ APAs. $\mathcal{R} \subseteq S_1 \times S_2$ is a strong ref rel iff. whenever $s_1 \mathcal{R} s_2$,

- $V_1(s_1) \subseteq V_2(s_2)$,
- $\forall a \in A, \varphi_1 \in C(S_1), L_1(s_1, a, \varphi) \neq \bot \implies \exists \varphi_2 \in C(S_2), L(s_2, a, \varphi_2) \neq \bot$ and
  $\exists \delta$ corresp , $\forall \mu_1 \in Sat(\varphi_1), \exists \mu_2 \in Sat(\varphi_2), \mu_1 \Subset^{\delta}_{\mathcal{R}} \mu_2$, and
- $\forall a \in A, \varphi_2 \in C(S_2), L(s_2, a, \varphi_2) = \top \implies \exists \varphi_1 \in C(S_1), L_1(s_1, a, \varphi_1) = \top$ and
  $\exists \delta$ corresp , $\forall \mu_1 \in Sat(\varphi_1), \exists \mu_2 \in Sat(\varphi_2), \mu_1 \Subset^{\delta}_{\mathcal{R}} \mu_2$.

Refinement

- Mix of refinement for MTS and refinement for CMCs
- 3 syntactic definitions

Definition. $N_1 = (S_1, A, L_1, AP, V_1, s_0^1)$, $N_2 = (S_2, A, L_2, AP, V_2, s_0^2)$ APAs. $\mathcal{R} \subseteq S_1 \times S_2$ is a weak ref rel iff. whenever $s_1 \mathcal{R} s_2$,

- $V_1(s_1) \subseteq V_2(s_2)$,
- $\forall a \in A, \varphi_1 \in C(S_1), L_1(s_1, a, \varphi) \neq \bot \implies \exists \varphi_2 \in C(S_2), L(s_2, a, \varphi_2) \neq \bot$ and $\forall \mu_1 \in Sat(\varphi_1), \mu_1 \Subset_{\mathcal{R}} \varphi_2$, and
- $\forall a \in A, \varphi_2 \in C(S_2), L(s_2, a, \varphi_2) = \top \implies \exists \varphi_1 \in C(S_1), L_1(s_1, a, \varphi_1) = \top$ and $\forall \mu_1 \in Sat(\varphi_1), \mu_1 \Subset_{\mathcal{R}} \varphi_2$.

Refinement

- Mix of refinement for MTS and refinement for CMCs
- 3 syntactic definitions

Definition. $N_1 = (S_1, A, L_1, AP, V_1, s_0^1)$, $N_2 = (S_2, A, L_2, AP, V_2, s_0^2)$ APAs. $\mathcal{R} \subseteq S_1 \times S_2$ is a weak weak ref rel iff. whenever $s_1 \mathcal{R} s_2$,

- $V_1(s_1) \subseteq V_2(s_2)$,
- $\forall a \in A, \varphi_1 \in C(S_1), L_1(s_1, a, \varphi) \neq \bot \implies$ $\forall \mu_1 \in Sat(\varphi_1) \exists \varphi_2 \in C(S_2), L(s_2, a, \varphi_2) \neq \bot$ and $\mu_1 \Subset_{\mathcal{R}} \varphi_2$, and
- $\forall a \in A, \varphi_2 \in C(S_2), L(s_2, a, \varphi_2) = \top \implies \exists \varphi_1 \in C(S_1), L_1(s_1, a, \varphi_1) = \top$ and $\forall \mu_1 \in Sat(\varphi_1), \mu_1 \Subset_{\mathcal{R}} \varphi_2$.

Refinement - Results

- Sound: $S_1 \preceq_X S_2$ implies $[\![S_1]\!] \subseteq [\![S_2]\!]$ for all X

Refinement - Results

- Sound: $S_1 \preceq_X S_2$ implies $[\![S_1]\!] \subseteq [\![S_2]\!]$ for all X

Theorem

Weak weak Refinement is complete for deterministic systems: for S_1, S_2, $[\![S_1]\!] \subseteq [\![S_2]\!]$ implies $S_1 \preceq_W S_2$

Determinism:

- Action determinism (MTS)
- Constraint determinism (CMCs)

Conjunction - First Idea

- How are modalities handled?

Conjunction - First Idea

- How are modalities handled? Like MTS:

$\wedge$	$\top$	?	$\bot$
$\top$	$\top$	$\top$	i
?	$\top$	?	$\bot$
$\bot$	i	$\bot$	$\bot$

Conjunction - First Idea

- How are modalities handled? Like MTS:

$\wedge$	$\top$	?	$\bot$
$\top$	$\top$	$\top$	i
?	$\top$	?	$\bot$
$\bot$	i	$\bot$	$\bot$

- How are constraints handled?

Conjunction - First Idea

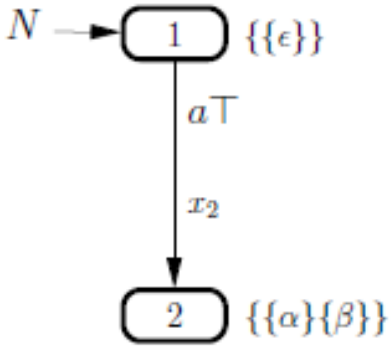
- How are modalities handled? Like MTS:

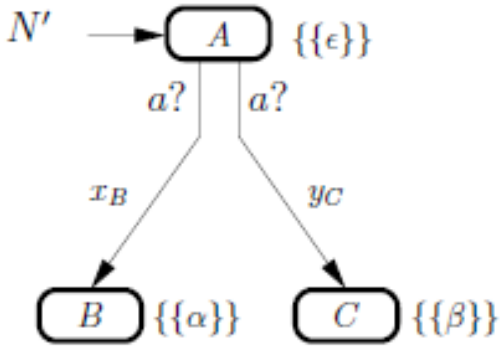
$\wedge$	$\top$	?	$\bot$
$\top$	$\top$	$\top$	i
?	$\top$	?	$\bot$
$\bot$	i	$\bot$	$\bot$

- How are constraints handled? Like CMCs:

$$\mu \in Sat(\varphi_\wedge) \iff \begin{cases} \mu\downarrow_1 \in Sat(\varphi_1) \\ \mu\downarrow_2 \in Sat(\varphi_2) \end{cases}$$

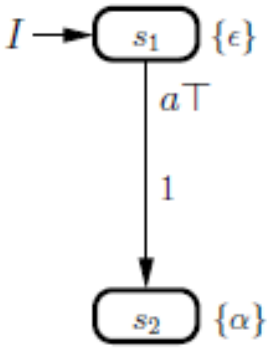
Conjunction - Not so simple

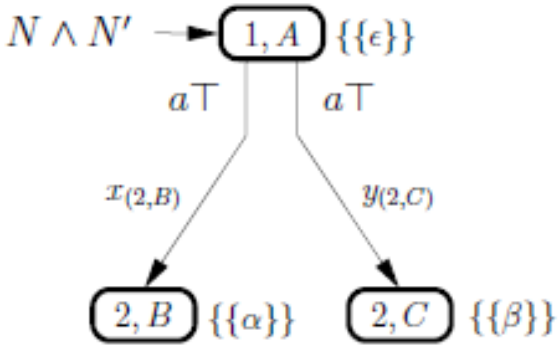




$$\varphi_x = (x_2 = 1)$$

$$\varphi'_x \equiv (x_B = 1)$$
$$\varphi'_y \equiv (y_C = 1)$$





$$\varphi''_x \equiv (x_{(2,B)} = 1)$$
$$\varphi''_y \equiv (y_{(2,C)} = 1)$$

Conjunction - General Definition

- Changes to the previous definition:

$\wedge$	$\top$	$?$	$\bot$
$\top$	$? + \top(*)$	$? + \top(*)$	i
$?$	$? + \top(*)$	$?$	$\bot$
$\bot$	i	$\bot$	$\bot$

Conjunction - General Definition

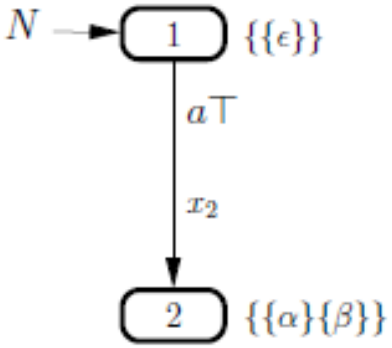
- Changes to the previous definition:

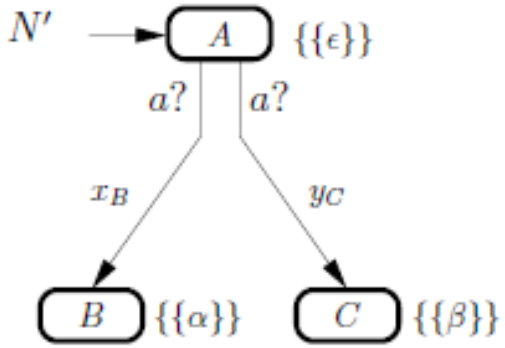
$\wedge$	$\top$	$?$	$\bot$
$\top$	$?+\top(*)$	$?+\top(*)$	i
$?$	$?+\top(*)$	$?$	$\bot$
$\bot$	i	$\bot$	$\bot$

- Additional transition:
  (*) When $s_1 \xrightarrow{a} \varphi_1$, we add $(s_1, s_2) \xrightarrow{a} \tilde{\varphi}_1$ such that

$$\tilde{\mu} \in Sat(\tilde{\varphi}_1) \iff \begin{cases} \tilde{\mu}\downarrow_1 \in Sat(\varphi_1) \\ \exists \varphi_2, L_2(s_2, a, \varphi_2) \neq \bot \wedge \tilde{\mu}\downarrow_2 \in Sat(\varphi_2) \end{cases}$$

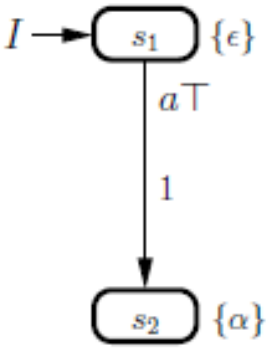
Conjunction - Example

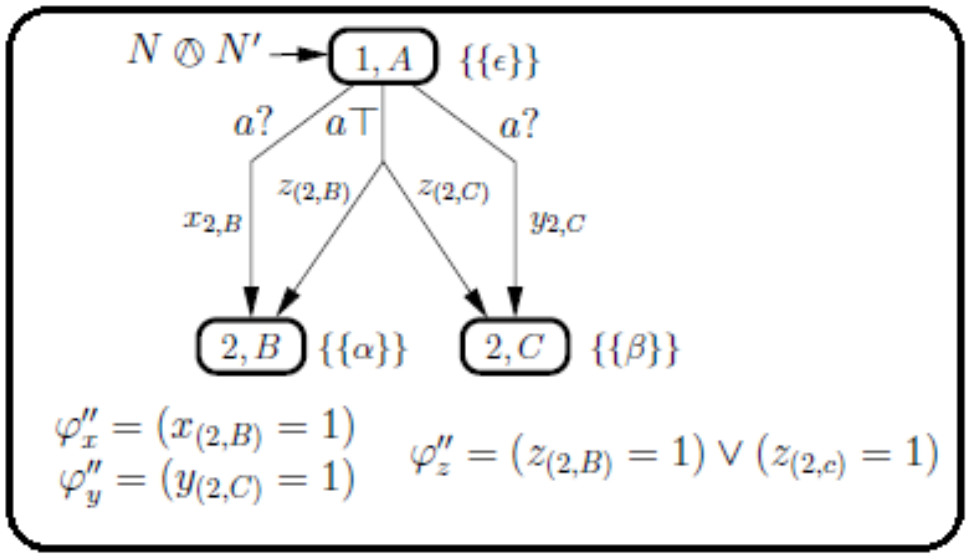




$\varphi_x = (x_2 = 1)$

$\varphi'_x \equiv (x_B = 1)$
$\varphi'_y \equiv (y_C = 1)$





Conjunction - Results

- General Conjunction ($\textcircled{\wedge}$) is equal to GLB wrt. weak weak ref

Conjunction - Results

- General Conjunction ($\textcircled{\wedge}$) is equal to GLB wrt. weak weak ref
- Deterministic APAs: First Definition ($\wedge$) works

Conjunction - Results

- General Conjunction ($\textcircled{\wedge}$) is equal to GLB wrt. weak weak ref
- Deterministic APAs: First Definition ($\wedge$) works
- In General: : $N_1 \wedge N_2 \preceq_W N_1 \textcircled{\wedge} N_2$

Conjunction - Results

- General Conjunction ($\textcircled{\wedge}$) is equal to GLB wrt. weak weak ref
- Deterministic APAs: First Definition ($\wedge$) works
- In General: : $N_1 \wedge N_2 \preceq_W N_1 \textcircled{\wedge} N_2$
- Under determinism: $N_1 \textcircled{\wedge} N_2 \preceq_W N_1 \wedge N_2$

Parallel Composition

$N_1 = (S_1, A_1, L_1, AP_1, V_1, s_0^1)$, $N_2 = (S_2, A_2, L_2, AP_2, V_2, s_0^2)$ APAs.

Parallel Composition

$N_1 = (S_1, A_1, L_1, AP_1, V_1, s_0^1)$, $N_2 = (S_2, A_2, L_2, AP_2, V_2, s_0^2)$ APAs.

- Disjoint atomic propositions (can be generalized):

$$AP_1 \cap AP_2 = \emptyset$$

Parallel Composition

$N_1 = (S_1, A_1, L_1, AP_1, V_1, s_0^1)$, $N_2 = (S_2, A_2, L_2, AP_2, V_2, s_0^2)$ APAs.

- Disjoint atomic propositions (can be generalized):

$$AP_1 \cap AP_2 = \emptyset$$

- Asymmetric actions

Parallel Composition

$N_1 = (S_1, A_1, L_1, AP_1, V_1, s_0^1)$, $N_2 = (S_2, A_2, L_2, AP_2, V_2, s_0^2)$ APAs.

- Disjoint atomic propositions (can be generalized):

$$AP_1 \cap AP_2 = \emptyset$$

- Asymmetric actions
- Synchronization set:

$$\bar{A} \subseteq A_1 \cap A_2$$

Parallel Composition

$N_1 = (S_1, A_1, L_1, AP_1, V_1, s_0^1)$, $N_2 = (S_2, A_2, L_2, AP_2, V_2, s_0^2)$ APAs.

- Disjoint atomic propositions (can be generalized):

$$AP_1 \cap AP_2 = \emptyset$$

- Asymmetric actions
- Synchronization set:

$$\bar{A} \subseteq A_1 \cap A_2$$

- Synchronous on $\bar{A}$, interleaving on $(A_1 \cup A_2) \setminus \bar{A}$

Parallel Composition

$N_1 = (S_1, A_1, L_1, AP_1, V_1, s_0^1)$, $N_2 = (S_2, A_2, L_2, AP_2, V_2, s_0^2)$ APAs.

- Disjoint atomic propositions (can be generalized):

$$AP_1 \cap AP_2 = \emptyset$$

- Asymmetric actions
- Synchronization set:

$$\bar{A} \subseteq A_1 \cap A_2$$

- Synchronous on $\bar{A}$, interleaving on $(A_1 \cup A_2) \setminus \bar{A}$
- Modalities for synchronous transitions:

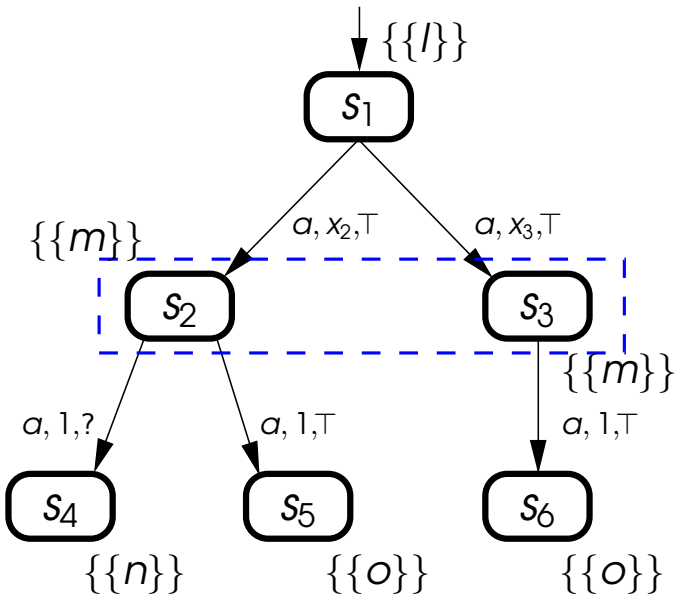
$\wedge$	$\top$	?	$\bot$
$\top$	$\top$	?	$\bot$
?	?	?	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$

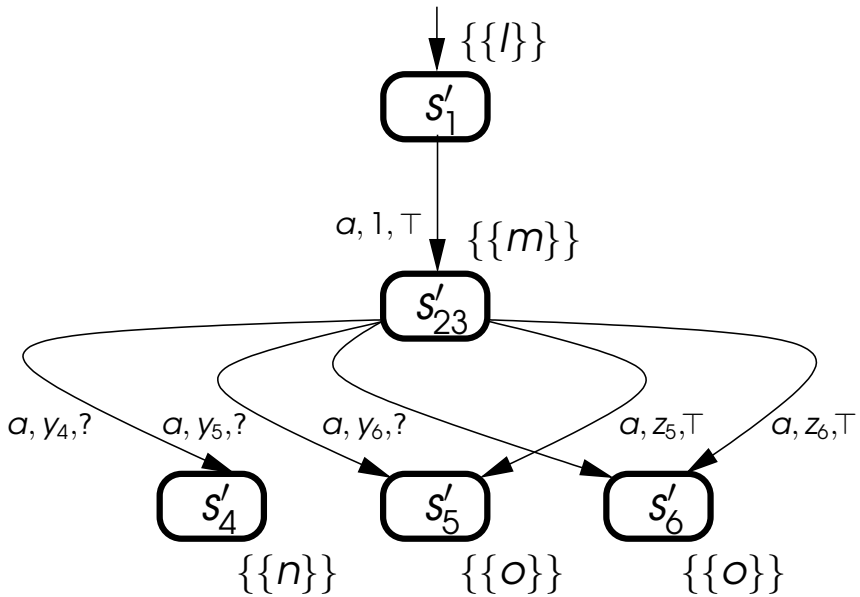
Abstractions 1/2

- State-based Abstraction: α

Abstractions 1/2

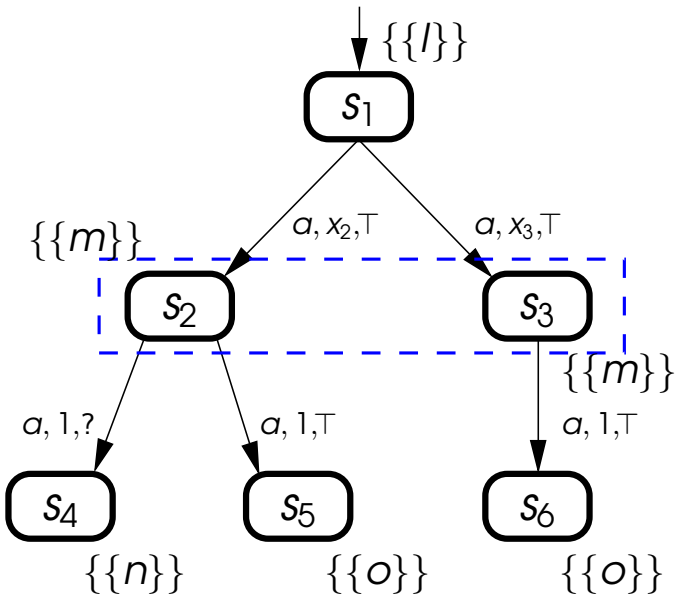
- State-based Abstraction: α

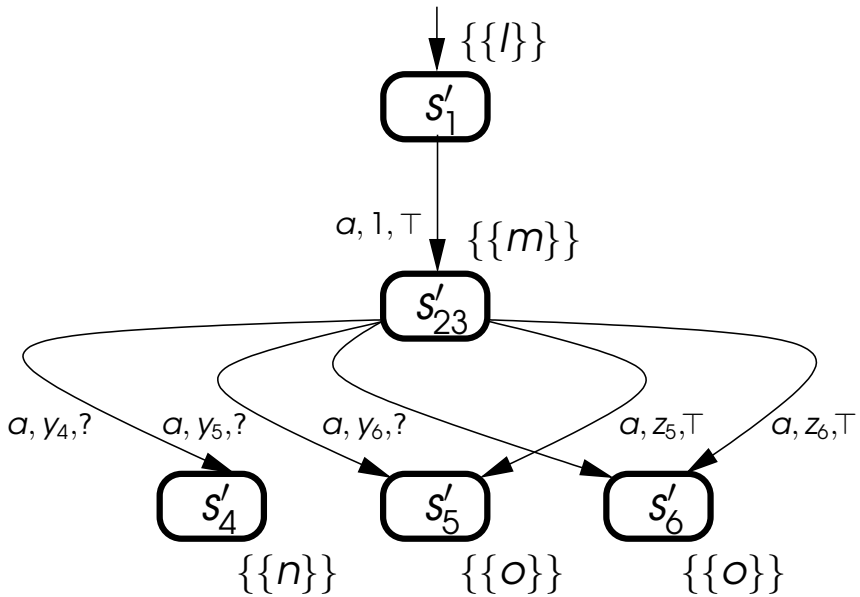




Abstractions 1/2

- State-based Abstraction: α

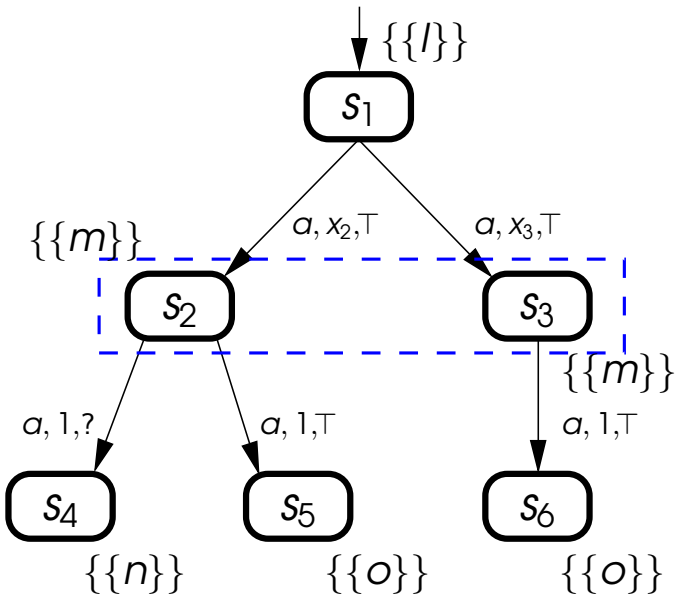


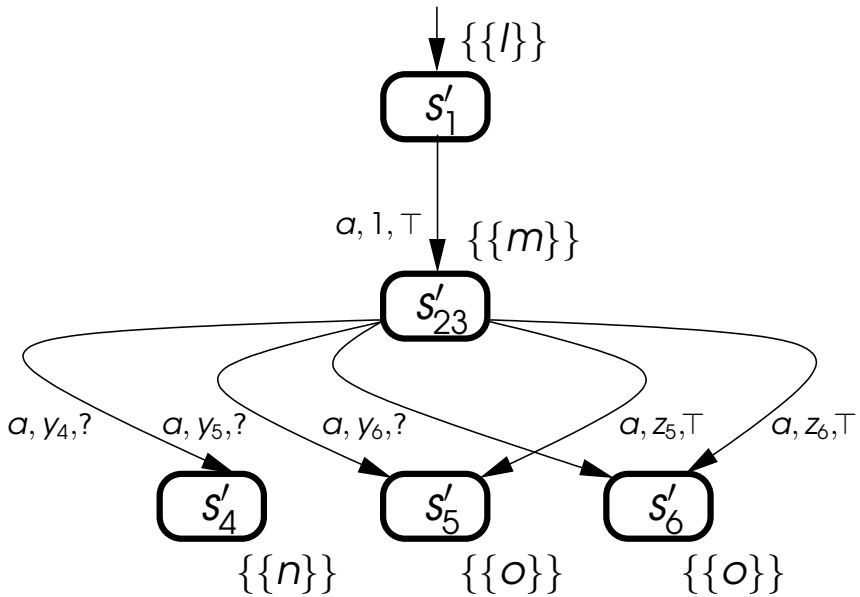


- $N \preceq_S \alpha(N)$

Abstractions 1/2

- State-based Abstraction: α





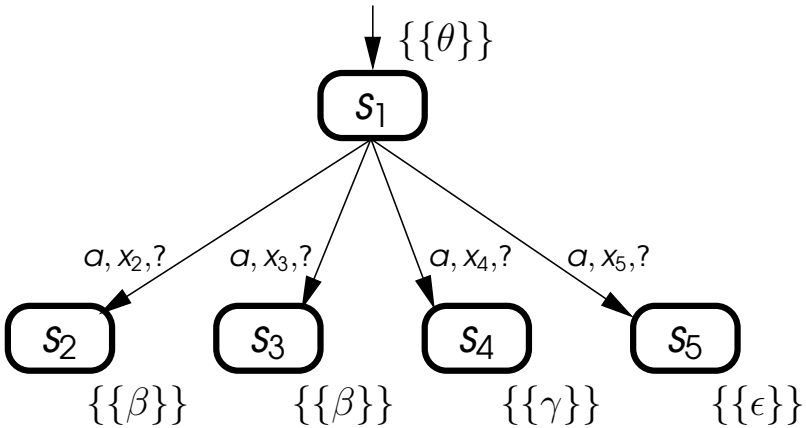
- $N \preceq_S \alpha(N)$
- For all $N_1, N_2, \alpha_1, \alpha_2$, $\alpha_1(N_1) \parallel_{\bar{A}} \alpha_2(N_2) = (\alpha_1 \times \alpha_2)(N_1 \parallel_{\bar{A}} N_2)$

Abstractions 2/2

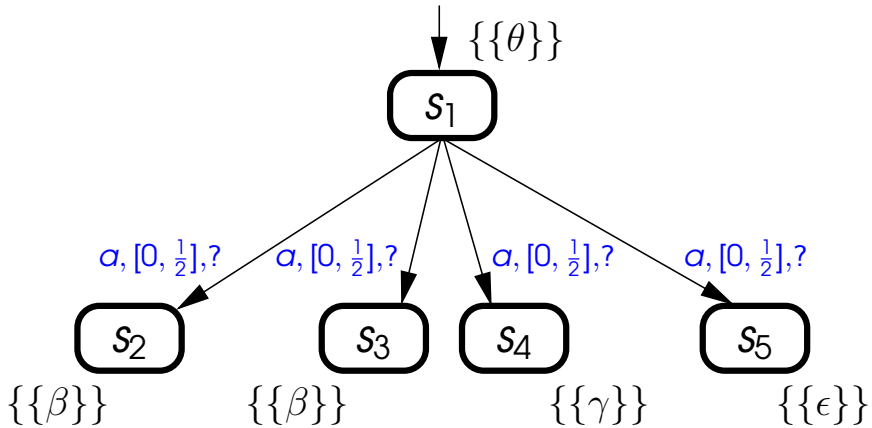
- Constraint-based Abstraction: χ

Abstractions 2/2

- Constraint-based Abstraction: χ

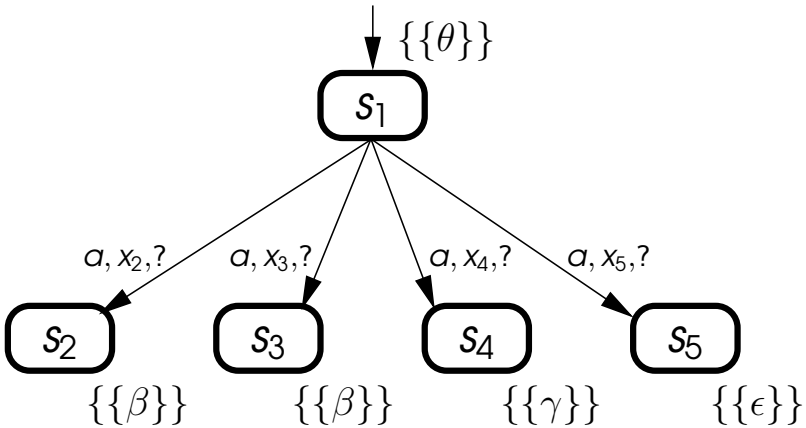


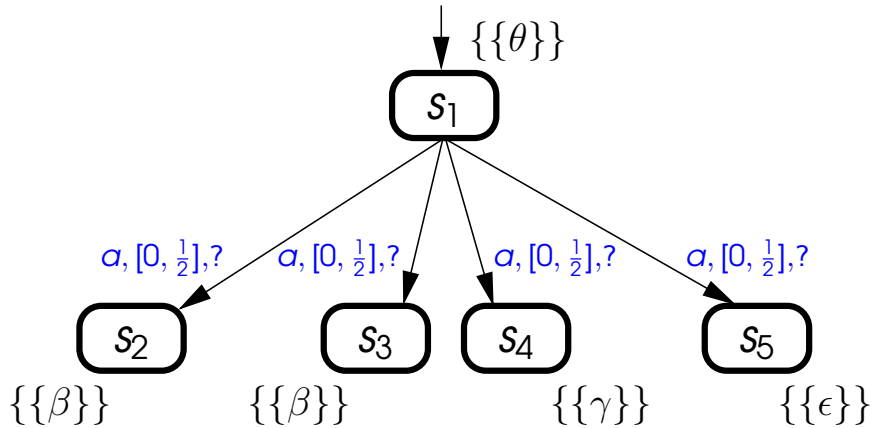
$\varphi_x \equiv x_2 + x_3 = \frac{1}{2} \wedge x_4 + x_5 = \frac{1}{2}$



Abstractions 2/2

- Constraint-based Abstraction: χ



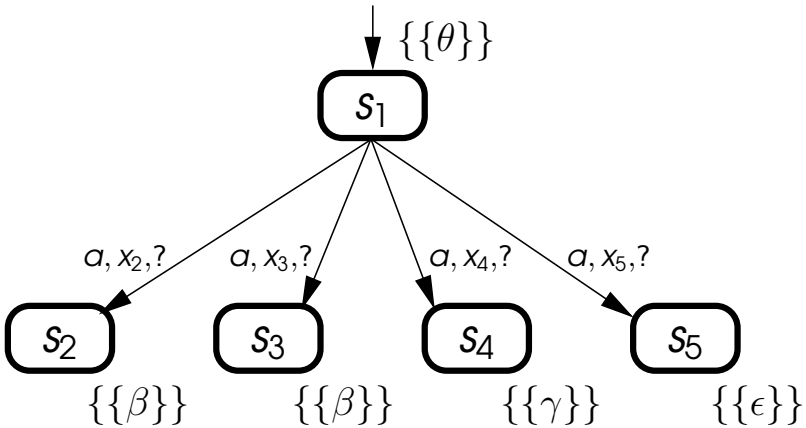


$\varphi_x \equiv x_2 + x_3 = \frac{1}{2} \wedge x_4 + x_5 = \frac{1}{2}$

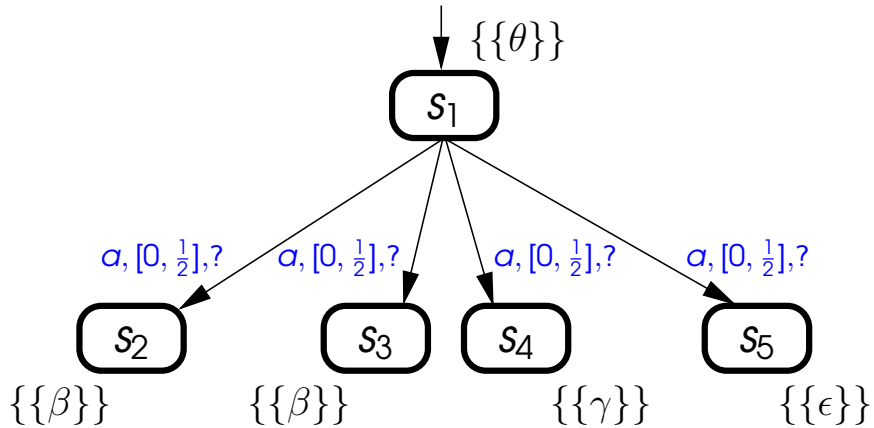
- $N \preceq_S \chi(N)$

Abstractions 2/2

- Constraint-based Abstraction: χ



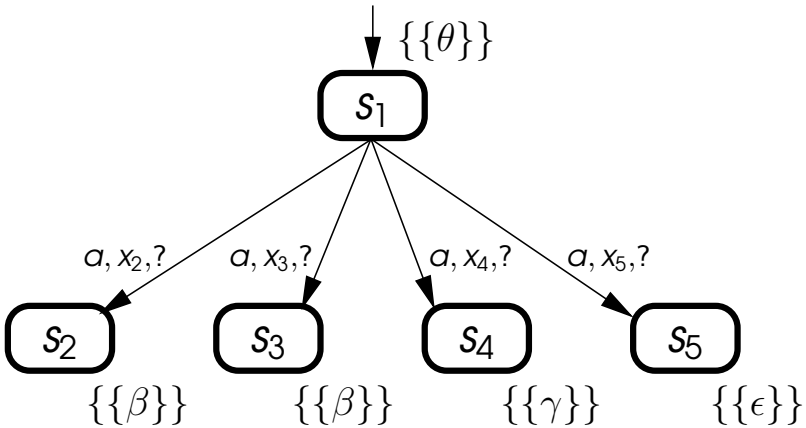
$\varphi_x \equiv x_2 + x_3 = \frac{1}{2} \wedge x_4 + x_5 = \frac{1}{2}$

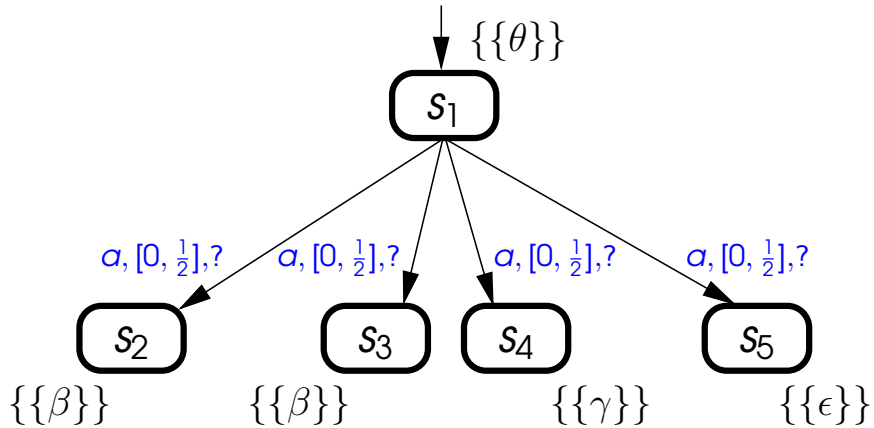


- $N \preceq_S \chi(N)$
- N in SVNF and IPA N' in SVNF, $N \preceq N'$ implies $\chi(N) \preceq N'$

Abstractions 2/2

- Constraint-based Abstraction: χ





$\varphi_x \equiv x_2 + x_3 = \frac{1}{2} \wedge x_4 + x_5 = \frac{1}{2}$

- $N \preceq_S \chi(N)$
- N in SVNF and IPA N' in SVNF, $N \preceq N'$ implies $\chi(N) \preceq N'$
- For all N_1, N_2, $\chi(N_1) \parallel_{\bar{A}} \chi(N_2) \preceq_S \chi(N_1 \parallel_{\bar{A}} N_2)$

Outline

Work in Progress

- Equip APAs with internal actions: Stuttering
- Consistency, Counter-Example generation, Difference
- Modal logic for APAs
- APAC: A tool for CMCs/APAs (only linear constraints)
- Extension to timed-stochastic Specs

Future work

- *Efficient* algorithms
- Soon: algorithms & tools for compositional modeling and design for probabilistic and real time systems
- Abstraction for model checking and games

The End

Thank you for your attention