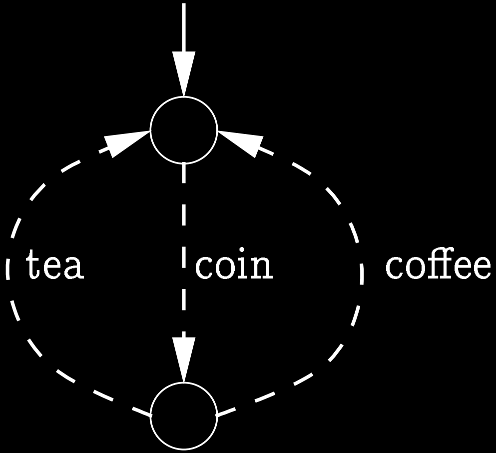
On Modal Refinement and Consistency

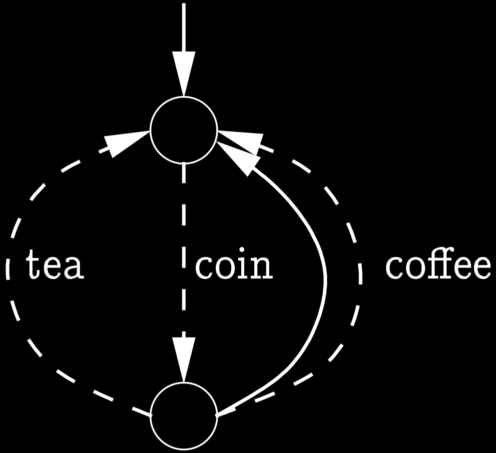
Kim G. Larsen, Ulrik Nyman and Andrzej Wąsowski

Aalborg University

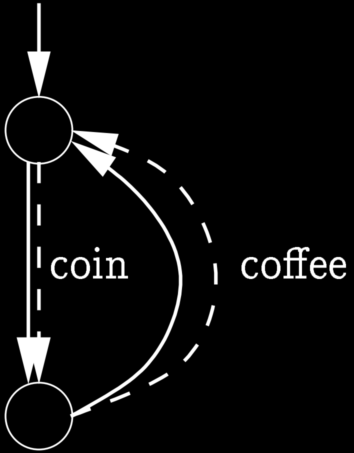
Modal Transition Systems



Modal Transition Systems



Modal Transition Systems



An implementation.

Outline

- Modal Transition Systems
- Part I: Refinement vs Implementation Inclusion
- Part II: Consistency
- Conjectures & Summary

Part I

Refinement
vs
Implementation Inclusion

Def. Modal transition system
$S = (states_S, \Sigma, \longrightarrow_S, \dashrightarrow_S)$

- $states_S$: a finite set of states
- Σ: an alphabet of actions
- $\longrightarrow_S \subseteq states_S \times \Sigma \times states_S$ (must)
- $\dashrightarrow_S \subseteq states_S \times \Sigma \times states_S$ (may)

Transition relations are finite.

Def. Modal transition system
$S = (states_S, \Sigma, \longrightarrow_S, \dashrightarrow_S)$

- $states_S$: a finite set of states
- Σ: an alphabet of actions

- $\longrightarrow_S \subseteq states_S \times \Sigma \times states_S$ (must)
- $\dashrightarrow_S \subseteq states_S \times \Sigma \times states_S$ (may)

Transition relations are finite.

Def. Modal Refinement

$S \leq_m T$ iff for any $a \in \Sigma$:

> whenever $S \overset{a}{\dashrightarrow} S'$ for some S' then for some T': $T \overset{a}{\dashrightarrow} T'$ and $S' \leq_m T'$
>
> whenever $T \xrightarrow{a} T'$ for some T' then for some S': $S \xrightarrow{a} S'$ and $S' \leq_m T'$

Generalizes simulation/bisimulation

Def. Modal Refinement

$S \leq_m T$ iff for any $a \in \Sigma$:

whenever $S \overset{a}{\dashrightarrow} S'$ for some S' then for some T': $T \overset{a}{\dashrightarrow} T'$ and $S' \leq_m T'$

whenever $T \xrightarrow{a} T'$ for some T' then for some S': $S \xrightarrow{a} S'$ and $S' \leq_m T'$

Generalizes simulation/bisimulation

Implementations

Def. A modal transition system I is an implementation iff $\longrightarrow_{\mathrm{I}} = \dashrightarrow_{\mathrm{I}}$.

Note: refinements of I are bisimilar.

Def. Implementation Inclusion $S \subseteq_m T$ iff $\forall$ implementations I.

$I \leq_m S$ implies $I \leq_m T$.

Implementations

Def. A modal transition system I is an implementation iff $\longrightarrow_I = \dashrightarrow_I$.

Note: refinements of I are bisimilar.

Def. Implementation Inclusion
$S \subseteq_m T$ iff $\forall$ implementations I.
$\quad I \leq_m S$ implies $I \leq_m T$.

Def. A refinement $\mathcal{R}$ is sound and complete wrt implementation inclusion if

$$S \mathcal{R} T \text{ iff } S \subseteq_m T \ .$$

Thm. Modal refinement is sound:

$$S \leq_m T \text{ implies } S \subseteq_m T \ .$$

Proof. Simple. $\square$

Def. A refinement $\mathcal{R}$ is sound and complete wrt implementation inclusion if

$$S \mathcal{R} T \quad \text{iff} \quad S \subseteq_m T \ .$$

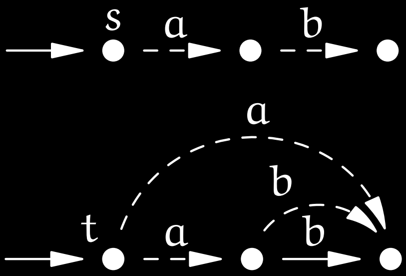
Thm. Modal refinement is sound:

$$S \leq_m T \quad \text{implies} \quad S \subseteq_m T \ .$$

Proof. Simple. $\square$

Thm. Modal refinement is incomplete

Proof.



$s \not\leq_m t, \quad$ while $\forall i. i \leq_m s$ iff $i \leq_m t \quad \square$

Theorem.

- Establishing implementation inclusion is co-NP hard
- even for syntactically consistent systems ($\dashrightarrow_S = \longrightarrow_S$).

Side note. Modal refinement is in P.

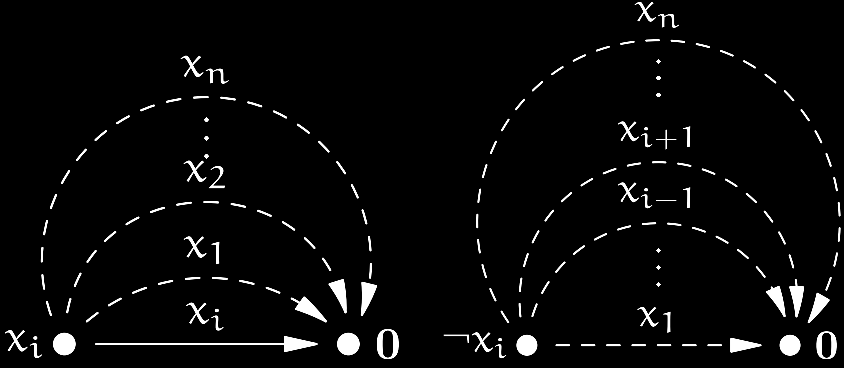
Proof. by reduction from validity checking (3-DNF-TAUTOLOGY).

Proof sketch.

- Construct T_φ (representing a tautology over $x_1 \ldots x_n$)
- Construct S_φ (representing φ)

Such that true $\implies \varphi$ is a tautology iff $[\![T_\varphi, \mathit{true}]\!] \subseteq [\![S_\varphi, \varphi]\!]$

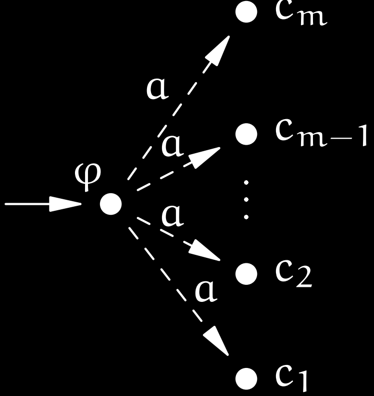
Representing x_i Representing $\overline{x_i}$



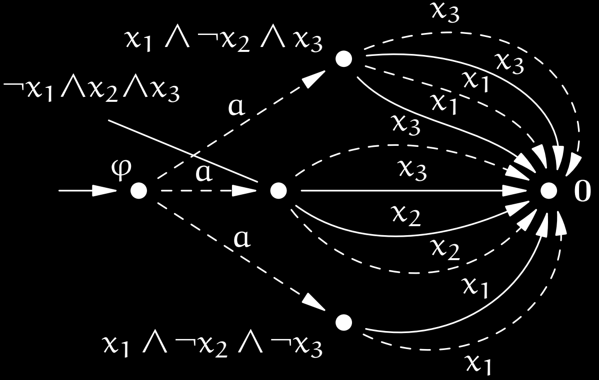
Combine to represent any satisfiable term.

A DNF formula:

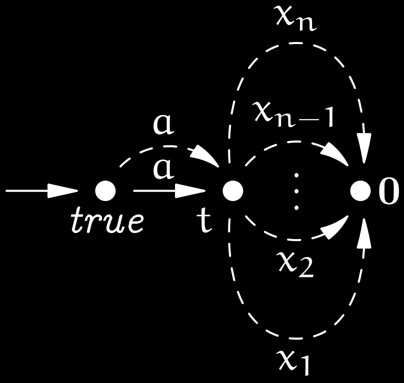
$c_1 \vee c_2 \vee \ldots \vee c_{m-1} \vee c_m$.



Reduction for $\varphi = (x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3)$.

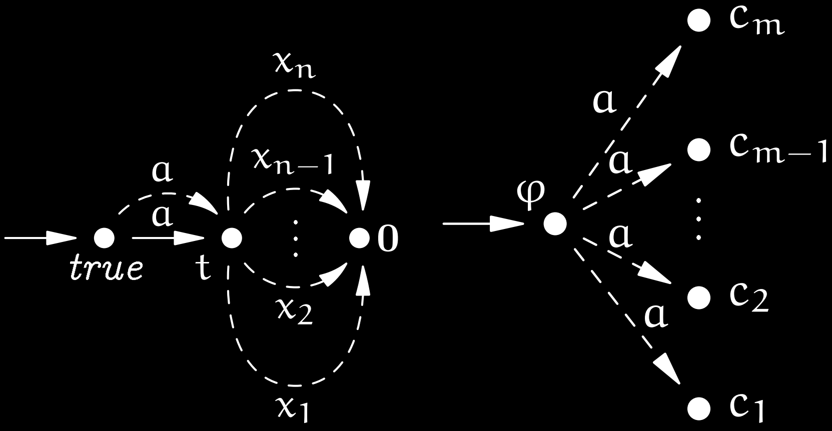


A true formula over the same variables.



Implementation inclusion

$\rightarrow \varphi$ is valid.



$\square$

Part II

Consistency

$(*)$ Syntactic consistency: $\longrightarrow \subseteq \dashrightarrow$

- No support for contradictions.
- Logic: consistency $\equiv$ existence of solutions under a satisfaction relation. Here:
  - refinement is satisfaction
  - implementations are solutions.
  - consistency: existence of implementation
- Characterize consistency using a computable criterion, like $(*)$

$(*)$ Syntactic consistency: $\longrightarrow \subseteq \dashrightarrow$

- No support for contradictions.
- Logic: consistency $\equiv$ existence of solutions under a satisfaction relation. Here:
  - refinement is satisfaction
  - implementations are solutions.
  - consistency: existence of implementation
- Characterize consistency using a computable criterion, like $(*)$

Def. Strong Consistency

A state S is strongly consistent iff there exists an implementation I such that

$$\mathrm{I} \leq_{\mathrm{m}} S \ .$$

Computing Consistency

For $\sigma, \sigma' \subseteq \mathit{states}_S$ we write:

$$\sigma \xrightarrow{a\lfloor S\rfloor} \sigma' \quad \text{iff} \quad \exists s \in \sigma.\ \exists s' \in \sigma'.\ s \xrightarrow{a} s'$$

$$\sigma \overset{a\lfloor S\rfloor}{\dashrightarrow} \sigma' \quad \text{iff} \quad \forall s \in \sigma.\ \exists s' \in \sigma'.\ s \overset{a}{\dashrightarrow} s'$$

(state sets are conjunctions of constraints)

Computing Consistency

Def. $\mathcal{B} \subseteq \mathcal{P}(states_S)$ is a strong consistency relation iff for all $a \in act$ and $\sigma \in \mathcal{B}$:

$$\forall s \in \sigma.\ s \xrightarrow{a} s'\ \exists \sigma' \in \mathcal{B}.$$
$$\sigma \xrightarrow{a}^{\lfloor S \rfloor} \sigma' \text{ and } \sigma \dashrightarrow{a}^{\lfloor S \rfloor} \sigma' \text{ and } s' \in \sigma'.$$

Thm. A state S is (strongly) consistent iff there exists a consistency relation with a class σ_S such that $S \in \sigma_S$.

Thm. Establishing strong consistency is NP-hard.

Proof. Reduction from 3-CNF-SAT.

Thm. A state S is (strongly) consistent iff there exists a consistency relation with a class σ_s such that $S \in \sigma_s$.

Thm. Establishing strong consistency is NP-hard.

Proof. Reduction from 3-CNF-SAT.

Consistency Results

Refinement	Lower bound	Upper bound
syntactic	linear	linear
strong	NP-hard	exp-time
weak	NP-hard	exp-time
may-weak	NP-hard	exp-time

Epilogue

Conjectures

- All consistencies are most likely PSPACE-complete (we have a proof sketch for the strong one).
- Establishing implementation inclusion is PSPACE-complete (currently working on this).

Summary

- Modal refinement is incomplete with respect to the implementation inclusion.
- Implementation inclusion is co-NP hard to establish.
- Characterized 4 consistencies
- All, but the syntactic one, are NP-hard.