On Modal Refinement and Consistency

Kim G. Larsen, Ulrik Nyman and Andrzej Wąsowski

Aalborg University
Modal Transition Systems

Diagram with states labeled "tea", "coin", and "coffee" connected by arrows.
Modal Transition Systems
Modal Transition Systems

An implementation.
Outline

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

Refinement vs Implementation Inclusion
Def. Modal transition system

\[ S = (\text{states}_S, \Sigma, \rightarrow_S, \longrightarrow_S) \]

- \( \Sigma \): an alphabet of actions
- \( \text{states}_S \): a finite set of states

- \( \rightarrow_S \subseteq \text{states}_S \times \Sigma \times \text{states}_S \) (must)
- \( \longrightarrow_S \subseteq \text{states}_S \times \Sigma \times \text{states}_S \) (may)

Transition relations are finite.
Def. Modal transition system

\[ S = \left( \text{states}_S, \Sigma, \longrightarrow_S, \longrightarrow_S \right) \]

- \( \Sigma \): an alphabet of actions
- \( \text{states}_S \): a finite set of states
- \( \longrightarrow_S \subseteq \text{states}_S \times \Sigma \times \text{states}_S \) (must)
- \( \longrightarrow_S \subseteq \text{states}_S \times \Sigma \times \text{states}_S \) (may)

Transition relations are finite.
Def. Modal Refinement

\[ S \leq_m T \text{ iff for any } a \in \Sigma:\]

whenever \( S \xrightarrow{a} S' \) for some \( S' \) then for some \( T': T \xrightarrow{a} T' \) 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 $S \xrightarrow{a} S'$ for some $S'$ then for some $T' : T \xrightarrow{a} T'$ 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 $\rightarrow_I = \rightarrow_I$.

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 $\iff \rightarrow_I = \neg
\neg \rightarrow_I$.

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 $R$ is sound and complete wrt implementation inclusion if

$$SRT \text{ iff } S \subseteq_m T.$$ 

Thm. Modal refinement is sound:

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

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

$$SRT \iff S \subseteq_m T.$$  

Thm. Modal refinement is sound:

$$S \leq_m T \implies S \subseteq_m T.$$  

Proof. Simple.
Thm. Modal refinement is incomplete

Proof.

$s \not\preceq_m t$, while $\forall i. i \preceq_m s$ iff $i \preceq_m t$
Theorem.

- Establishing implementation inclusion is co-NP hard
- even for syntactically consistent systems \((\rightarrow s = \longrightarrow s)\).

Side note. Modal refinement is in P.

Proof. by reduction from validity checking \((\text{3-DNF-TAUTOLOGY})\).
Representing $x_i$  Representing $\overline{x_i}$

Combine to represent any satisfiable term.
A DNF formula:
\[ c_1 \lor c_2 \lor \ldots \lor c_{m-1} \lor c_m. \]
A true formula over the same variables.
Implementation inclusion

$\rightarrow \phi$ is valid.
Part II

Consistency
(\ast) Syntactic consistency: \[ \rightarrow \subseteq \rightarrow \]

- No support for contradictions.
- Logic: consistency \equiv \text{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 (\ast)
(*) Syntactic consistency: $\rightarrow \subseteq \longrightarrow$

- 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

$$I \leq_m S.$$
Computing Consistency

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

$\sigma \xrightarrow{a \mid S} \sigma'$ iff $\exists s \in \sigma. \exists s' \in \sigma'. s \xrightarrow{a} s'$

$\sigma \xrightarrow{a \parallel S} \sigma'$ iff $\forall s \in \sigma. \exists s' \in \sigma'. s \xrightarrow{a} s'$

(state sets are conjunctions of constraints)
Def. $\mathcal{B} \subseteq \mathcal{P}(\text{states}_S)$ is a strong consistency relation iff for all $a \in \text{act}$ and $\sigma \in \mathcal{B}$:

$$\forall s \in \sigma. \ s \xrightarrow{a} s' \ \exists \sigma' \in \mathcal{B}. \ \sigma \xrightarrow{\text{a}\mid S} \sigma' \text{ and } \sigma \xrightarrow{\text{a}\mid S} \sigma' \text{ and } s' \in \sigma'.$$
Thm. A state $\mathcal{S}$ is (strongly) consistent iff there exists a consistency relation with a class $\sigma_s$ such that $\mathcal{S} \in \sigma_s$.

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 $\sigma_s$ such that $S \in \sigma_s$.

Thm. Establishing strong consistency is NP-hard.

Proof. Reduction from 3-CNF-Sat.
## Consistency Results

<table>
<thead>
<tr>
<th>Refinement</th>
<th>Lower bound</th>
<th>Upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>syntactic</td>
<td>linear</td>
<td>linear</td>
</tr>
<tr>
<td>strong</td>
<td>NP-hard</td>
<td>exp-time</td>
</tr>
<tr>
<td>weak</td>
<td>NP-hard</td>
<td>exp-time</td>
</tr>
<tr>
<td>may-weak</td>
<td>NP-hard</td>
<td>exp-time</td>
</tr>
</tbody>
</table>
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.