Robust models for generalized model checking

Michael Huth
Computing
Imperial College London

June 4, 2003

Acknowledgments: Patrice Godefroid, Rahda Jagadeesan, and David Schmidt.

Outline of talk

1. compositional model checking of modal transition systems (MTSs);
2. generalized model checking (GMC) for MTSs;
3. *robust* frameworks for generalized model checking;
4. GMC for MTSs is robust; and
5. type constructors preserving robustness of operational/denotational models.

1. Compositional checks of MTSs . . .

Modal transition systems (MTSs)

(Larsen & Thomsen'88) MTS M: a triple $(\Sigma, R^{\mathrm{a}}, R^{\mathrm{c}})$

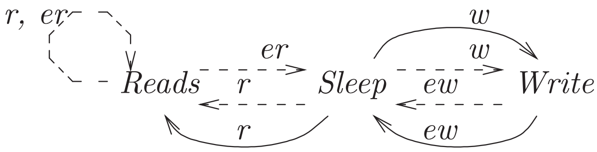
- Σ state set,
- $R^{\mathrm{a}} \subseteq \Sigma \times \mathtt{Act} \times \Sigma$ **must**-transitions,
- $R^{\mathrm{c}} \subseteq \Sigma \times \mathtt{Act} \times \Sigma$ may-transitions, and
- $R^{\mathrm{a}} \subseteq R^{\mathrm{c}}$: **must** implies **may**.

Two principal uses:

1. Under-specified models: **must**, **may** modalities for implementor.
2. Abstracted system: **must** $= \forall\exists$ and **may** $= \exists\exists$.

Analyzing a modal transition system

...R^{a} (solid) and R^{c} (dashed)



$\models^{\mathrm{a}}$ and $\models^{\mathrm{c}}$ abstract "for all ..." and "for some implementation" (respectively).

- $(M, \mathit{Sleep}) \models^{\mathrm{c}} (\exists r)(\exists r)(\exists er)(\exists w)\mathtt{true}$ holds: there is a sequence, at Sleep, of these four R^{c}-transitions in order.
- $(M, i) \models^{\mathrm{a}} \neg(\exists w)(\exists r)\mathtt{true}$ holds at all i: $(M, i) \models^{\mathrm{c}} (\exists w)(\exists r)\mathtt{true}$ doesn't — there is no sequence of two R^{c}-transitions labeled by w and then r.

Semantics for Hennessy-Milner logic

$$\phi ::= \mathtt{true} \;\mid\; \neg\phi \;\mid\; (\exists\alpha)\phi \;\mid\; \phi \wedge \phi$$

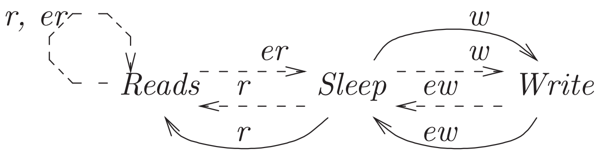
For $\mathrm{m} \in \{\mathrm{a}, \mathrm{c}\}$,

- $s \models^{\mathrm{m}} \mathtt{true}$
- $s \models^{\mathrm{m}} \neg\phi$ iff not $s \models^{\neg\mathrm{m}} \phi$, where $\neg\mathrm{a} = \mathrm{c}$ and $\neg\mathrm{c} = \mathrm{a}$
- $s \models^{\mathrm{m}} (\exists\alpha)\phi$ iff for some $(s, \alpha, s') \in R^{\mathrm{m}}$, $s' \models^{\mathrm{m}} \phi$
- $s \models^{\mathrm{m}} \phi_1 \wedge \phi_2$ iff $s \models^{\mathrm{m}} \phi_1$ and $s \models^{\mathrm{m}} \phi_2$.

Implies $s \models^{\mathrm{m}} \phi_1 \vee \phi_2$ iff $s \models^{\mathrm{m}} \phi_1$ or $s \models^{\mathrm{m}} \phi_2$, where $\phi_1 \vee \phi_2 \stackrel{\mathrm{def}}{=} \neg(\neg\phi_1 \wedge \neg\phi_2)$: loss of precision for $\mathrm{m} = \mathrm{a}$.

Extension to fixed points as usual.

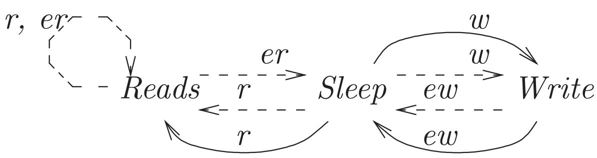
Revisit MTS check 1



- $s \models^{\mathrm{m}} \mathrm{true}$
- $s \models^{\mathrm{m}} (\exists\alpha)\phi$ iff for some $(s, \alpha, s') \in R^{\mathrm{m}}$, $s' \models^{\mathrm{m}} \phi$

$(M, \mathit{Sleep}) \models^{\mathrm{c}} (\exists r)(\exists r)(\exists er)(\exists w)\mathrm{true}$ holds: there is a sequence, at Sleep, of these four R^{c}-transitions in order.

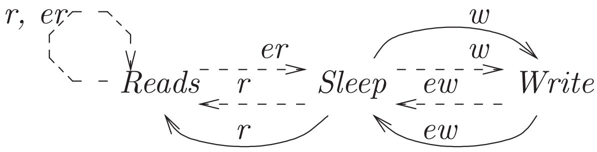
Revisit MTS check 2



- $s \models^{\mathrm{m}} \mathrm{true}$
- $s \models^{\mathrm{m}} \neg\phi$ iff not $s \models^{\neg\mathrm{m}} \phi$, where $\neg\mathrm{a} = \mathrm{c}$ and $\neg\mathrm{c} = \mathrm{a}$
- $s \models^{\mathrm{m}} (\exists\alpha)\phi$ iff for some $(s, \alpha, s') \in R^{\mathrm{m}}$, $s' \models^{\mathrm{m}} \phi$

$(M, i) \models^{\mathrm{a}} \neg(\exists w)(\exists r)\mathrm{true}$ holds at all i: $(M, i) \models^{\mathrm{c}} (\exists w)(\exists r)\mathrm{true}$ doesn't — there is no sequence of two R^{c}-transitions labeled by w and then r.

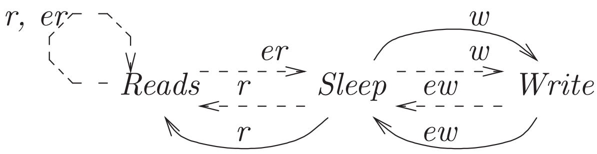
Revisit MTS check 3



- We have $Reads \models^{a} (\exists er)\mathrm{true} \vee \neg(\exists er)\mathrm{true}$ iff ($Reads \models^{a} (\exists er)\mathrm{true}$ or $Reads \models^{a} \neg(\exists er)\mathrm{true}$);
- $Reads \models^{a} (\exists er)\mathrm{true}$ **fails:** no s' with $(Reads, er, s') \in R^{a}$;
- $Reads \models^{a} \neg(\exists er)\mathrm{true}$ **fails** since $Reads \models^{c} (\exists er)\mathrm{true}$ holds: $(Reads, er, Reads) \in R^{c}$;
- $\Rightarrow$ loss of precision on propositional tautology!

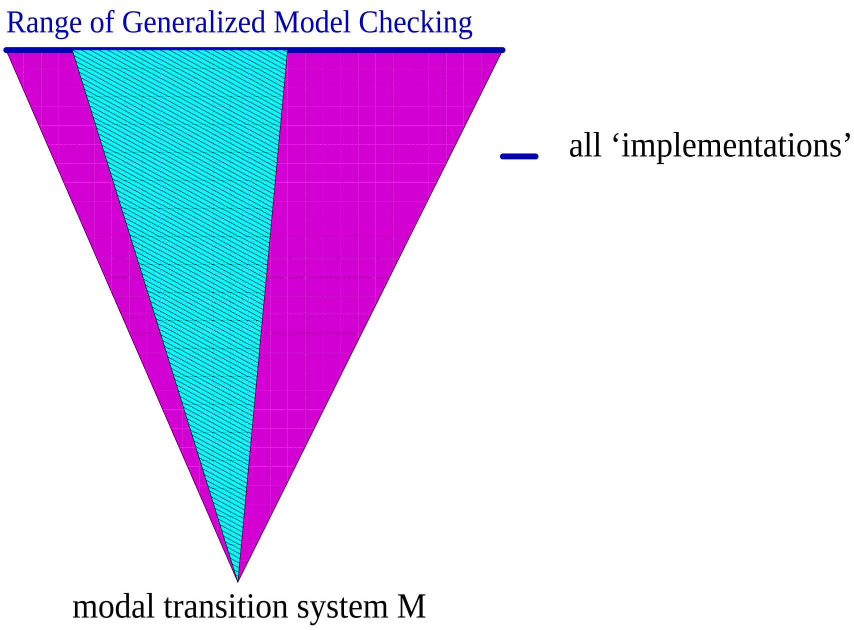
2. Generalized model checks of MTSs . . .

The MTS check re-visited



- $Reads \models^{\mathrm{a}} (\exists er)\mathrm{true} \vee \neg(\exists er)\mathrm{true}$ fails to hold;
- $Reads \models^{\mathbf{t}} (\exists er)\mathrm{true} \vee \neg(\exists er)\mathrm{true}$ holds; **thorough** semantics of (Bruns & Godefroid'00):
- $s \models^{\mathrm{t}} \phi$ iff for all "implementations" s' of s, $s' \models^{\mathrm{a}} \phi$;
- here: implementations = refining labeled transition systems.

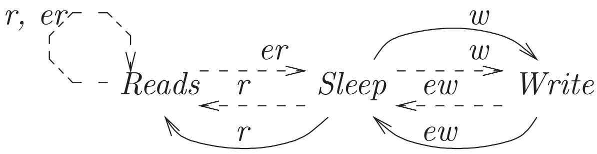
GMC 1



Refinement/abstraction of MTSs

Due to (Larsen & Thomsen'88): $(M, i)\,\mathsf{ref'd\_by}\,(N, j)$ iff
$\exists Q\colon (i, j) \in Q$ and whenever $(s, t) \in Q$

1. $(s, \alpha, s') \in R^{\mathrm{a}} \;\Rightarrow\; \exists (t, \alpha, t') \in R^{\mathrm{a}} \colon (s', t') \in Q$; and
2. $(t, \alpha, t') \in R^{\mathrm{c}} \;\Rightarrow\; \exists (s, \alpha, s') \in R^{\mathrm{c}} \colon (s', t') \in Q$.



abstracts / refined by

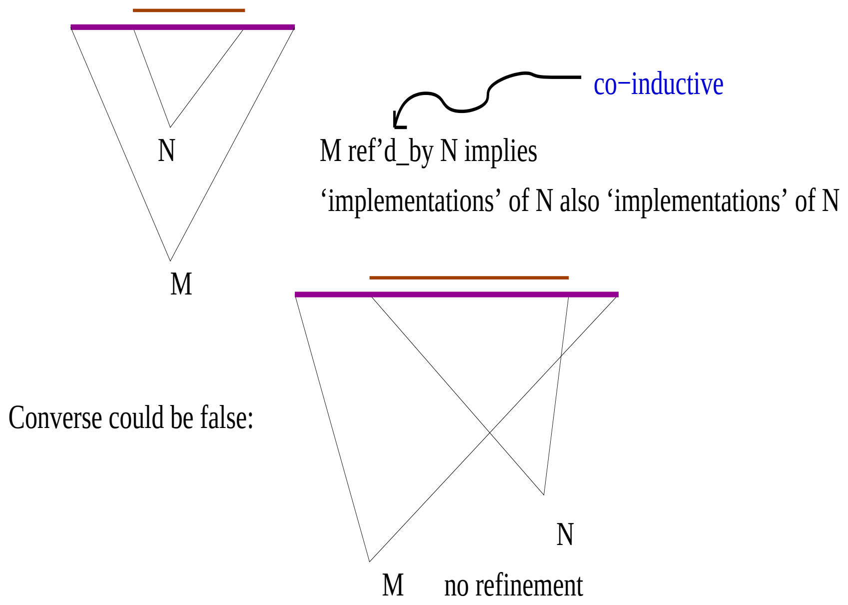
GMC 2

CTL GMC (Bruns & Godefroid'00):

- M pointed modal transition system, ϕ CTL formula;
- construct alternating Büchi word automata $\mathcal{A}_{(M,\phi)}$ over 1-letter alphabet, $\leq O(|\,M\,| \cdot 2^{O(|\phi|)})$ states;
- has non-empty language iff $\exists$ *concrete* pointed MTS $K\colon M\,\mathsf{ref'd\_by}\,K,\;\; K \models \phi$.

3. Robust generalized model checking . . .

Robustness



Conceptually . . .

- refinements define implementations: M impl′d_by K iff
  1. M ref′d_by K and
  2. $\forall K' \colon K$ ref′d_by $K' \Rightarrow K'$ ref′d_by K;
- conversely, any impl′d_by determines M ref′d_by N iff $\{K \mid N \text{ impl′d\_by } K\} \subseteq \{K \mid M \text{ impl′d\_by } K\}$;
- robustness =
  - this inclusion *characterizes* refinement; and
  - implementation and refinement relation determine each other.

4. Robustness for modal transition systems . . .

Domain-theoretic model of MTSs (HJS'03)

- Initial solution $\mathcal{D}$, over SFP domains, of

$$D = \prod_{\alpha \in \mathtt{Act}} \mathsf{M}[D]$$

- $\mathsf{M}[D]$: mixed powerdomain of D (Heckmann'90, Gunter'92);
- elements of $\mathsf{M}[D]$: (L, U) where L Scott-closed, U Scott-compact upper, and $L = \downarrow(L \cap U)$; and
- order on $\mathsf{M}[D]$: $(L, U) \leq (L', U')$ iff $L \subseteq L'$ and $U' \subseteq U$.

Example elements:

- $\perp_{\mathcal{D}} = (\emptyset, \mathcal{D})_{\alpha \in \mathtt{Act}}$ "universal stub" aka "state-wide divergence;" a compact "tree" and
- $(\emptyset, \emptyset)_{\alpha \in \mathtt{Act}}$ "state-wide deadlock;" a maximal compact "tree."

Characteristic semi-ring

(Heckmann'90) $\mathsf{M}[*]$ isomorphic to flat booleans $\mathbb{B}_\perp = \{\perp < \mathsf{false}, \mathsf{true}\}$, where

- $\perp = (\{\}, \{*\})$,
- $\mathsf{false} = (\{\}, \{\})$, and
- $\mathsf{true} = (\{*\}, \{*\})$.

For $(L, U) \in \mathsf{M}[D]$, L specifies **must** information (here R^{a}), U **may** information (here R^{c}).

The pair $(\{*\}, \{\})$ does not satisfy $L = \downarrow(L \cap U)$.

Adequacy of domain model $D = \prod_{\alpha \in \mathtt{Act}} \mathsf{M}[D]$

For all pointed MTS (M, i) and (N, j), 1.-4. in (HJS'03),

1. (M, i) has embedding $\langle\!| M, i |\!\rangle$ in $\mathcal{D}$;
2. (M, i) $\mathsf{ref'd\_by}$ $\langle\!| M, i |\!\rangle$ and $\langle\!| M, i |\!\rangle$ $\mathsf{ref'd\_by}$ (M, i);
3. (M, i) $\mathsf{ref'd\_by}$ (N, j) iff $\langle\!| M, i |\!\rangle$ $\mathsf{ref'd\_by}$ $\langle\!| N, j |\!\rangle$;
4. on $\mathcal{D}$, $\mathsf{ref'd\_by}$ equals order $\leq$ of $\mathcal{D}$;
5. $\langle\!| M, i |\!\rangle$ embeds reachable part of i only; not compact if i on infinite path;
6. $\mathbf{K}(\mathcal{D})$ in image of embedding; each $k \in \mathbf{K}(\mathcal{D})$ equals the embedding of its "tree;"
7. $(L, i) \in \mathrm{LTS} \Rightarrow \langle\!| L, i |\!\rangle \in \max(\mathcal{D})$; and
8. $d \in \max(\mathcal{D}) \Rightarrow d$ determines a 'topologically compact' LTS.

$\Rightarrow (\mathcal{D}, \leq)$ precise model for reasoning up to $\mathsf{ref'd\_by}$-equivalence.

Proof of robustness for MTSs 1

$$d \leq_t e \stackrel{\text{def}}{=} \{m \in \max(\mathcal{D}) \mid e \leq m\} \subseteq \{m \in \max(\mathcal{D}) \mid d \leq m\}.$$

Claim: $\leq_t$ equals $\leq$.

- Full abstraction and universality of $\mathcal{D}$ imply: claim captures equality of implementation and refinement relationship.
- $\leq$ transitive $\Rightarrow$ $\leq$ contained in $\leq_t$.
- Converse harder.

Proof of robustness for MTSs 2

$\max(\mathcal{D})$ Stone space and Lawson-closed in D — uses techniques of (Alessi et al.'03):

Exists skeleton $\mathrm{SK}(\mathcal{D})$ of $\mathcal{D}$, a finitely-branching tree in $\mathbf{K}(\mathcal{D})$ with finite principal ideals in the order induced by $\mathcal{D}$ such that every $k \in \mathbf{K}(\mathcal{D})$ has some $l \in \mathrm{SK}(\mathcal{D})$ with $k \leq l$ in $\mathcal{D}$

$$\mathrm{SK}_m(\mathcal{D}) \stackrel{\mathrm{def}}{=} \{\langle\!|\, M[m], i \,|\!\rangle \mid (M, i) \text{ is an LTS}\}$$

$(M[m], i)$: 'truncate' (M, i) after m transitions

$$\mathrm{SK}(\mathcal{D}) \stackrel{\mathrm{def}}{=} \bigcup_{m \geq 0} \mathrm{SK}_m(\mathcal{D}).$$

Proof of robustness for MTSs 3

$$\mathrm{V}_d \stackrel{\mathrm{def}}{=} \{e \in \mathcal{D} \mid d \leq_t e\}$$

Scott-open for $d \in \mathbf{K}(\mathcal{D})$:

- upper set: $\leq$ in $\leq_t$ and $\leq_t$ transitive,
- given $e \in \mathrm{V}_d$

$$\{\uparrow\! l \cap \max(\mathcal{D}) \mid l \in \downarrow\! e \cap \mathbf{K}(\mathcal{D})\}$$

  filtered collection of Scott-compact saturated sets in $\mathcal{D}$ ($\uparrow\! l$ and $\max(\mathcal{D})$ Lawson-closed); its intersection $\uparrow\! e \cap \max(\mathcal{D})$ is in Scott-open $\uparrow\! d$; Hofman-Mislove Theorem $\Rightarrow\ \exists l \in \downarrow\! e \cap \mathbf{K}(\mathcal{D})$ such that $\uparrow\! l \cap \max(\mathcal{D}) \subseteq \uparrow\! d$, i.e. $l \in \mathrm{V}_d$.

Proof of robustness for MTSs 4

If $k \leq_t l$ implies $k \leq l$ for all $k, l \in \mathbf{K}(\mathcal{D})$, then $\leq_t$ equals $\leq$:

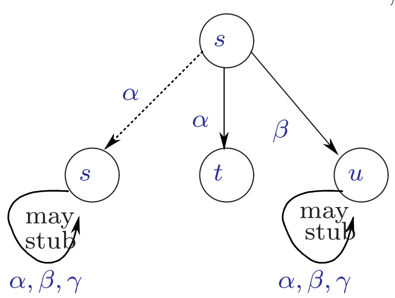
Proof by contradiction: let $d \leq_{\mathrm{t}} e$ and $d \not\leq e$.

- $\mathcal{D}$ algebraic and $d \not\leq e \;\Rightarrow\; \exists k \in \downarrow d \cap \mathbf{K}(\mathcal{D})$ with $k \not\leq e$
- so $k \leq d$ and $d \leq_{\mathrm{t}} e$ imply $k \leq_{\mathrm{t}} e$
- so e is in **Scott-open** V_k
- so $\exists l \in \downarrow e \cap \mathbf{K}(\mathcal{D})$ with $k \leq_{\mathrm{t}} l$
- so $k \not\leq e$ and $l \leq e$ imply $k \not\leq l$.

Proof of robustness for MTSs 4

"measure of partiality" $k \mapsto \deg(k) \colon \mathbf{K}(\mathcal{D}) \to \mathbb{N}$ with $\deg(k) = 0$ iff $k \in \max(\mathcal{D})$:

- each $k \in \mathbf{K}(\mathcal{D})$ determines/is determined by finite "tree"
- $\deg(k) =$ number of $R^{\mathrm{c}} \setminus R^{\mathrm{a}}$-transitions for each node and action in `Act` in "tree" of k; e.g. $\deg(s) = 7$ — s is unfolded



Proof of robustness for MTSs 5

Show $k \leq_t l \;\Rightarrow\; k \leq l$ by induction on $\deg(l)$.

Base case = previous item:

- $\deg(l) = 0 \;\Rightarrow\; l \in \max(\mathcal{D})$
- so $\uparrow l \cap \max(\mathcal{D}) = \{l\}$, but $\uparrow l \cap \max(\mathcal{D}) \subseteq \uparrow k \cap \max(\mathcal{D})$ as $k \leq_t l$
- so $k \leq l$.

Proof of robustness for MTSs 6

Show $k \leq_t l \;\Rightarrow\; k \leq l$ by induction on $\deg(l)$.

Inductive step:

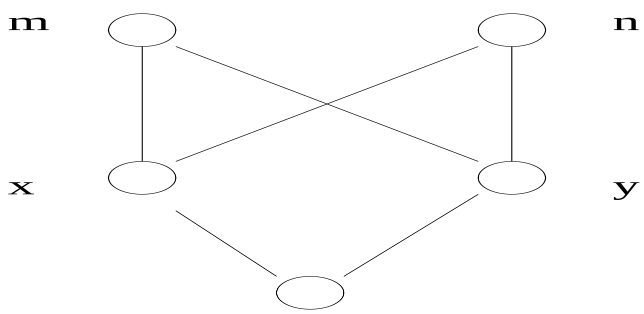
- $\deg(l) > 0 \;\Rightarrow\; \exists \alpha, l', l'' \colon (l', \alpha, l'') \in R^{\mathrm{c}} \setminus R^{\mathrm{a}}$ on "tree" of l
- l_+ "tree" obtained from "tree" of l be removing that transition
- l_- obtained from "tree" of l by promoting (l', α, l'') to R^{a}
- l_+ and l_- have smaller degree and have l as infimum (reasoned via algebraicity and refinement of "trees")
- use $k \leq_t l \leq l_+, l_-$ and induction: $k \leq l_+$ and $k \leq l_-$
- use $l = l_+ \wedge l_-$: get $k \leq l$.

Note: $\mathcal{D}$ has infima for "trees of distance 1; but not for **all** $(k, l) \in \mathbf{K}(\mathcal{D}) \times \mathbf{K}(\mathcal{D})$.

5. Robust type constructors . . .

Motivation

- Type constructors: building blocks of domain equations for adequate models of frameworks (e.g. $\prod$ and $\mathsf{M}[]$ for $\mathcal{D}$);
- domain $(D, \leq)$ is *robust* iff $\leq_t$ equals $\leq$;
- a non-example:



- which type constructors (e.g. functors and inverse limits in SFP) preserve robustness?

Preliminary results

- \- unary lift functor maps robust $\{*\}$ to non-robust $\{\perp < *\}$;
- \- lower powerdomain enforces a top element;
- \- upper powerdomain $\mathsf{U}[D]$: $X \leq_t Y$ iff $Y \cap \max(D) \subseteq X \cap \max(D)$;
- \+ Scott-continuous closure of robust domain;
- ? by previous item: convex powerdomain preserves robustness if mixed powerdomain does;
- ? mixed powerdomain: topological conjecture: "qualified yes;"
- \+ product functors and coalesced sum functors;
- \- any sum functors with a lift; and
- \+ Scott-continuous projections of robust domains, if maximal elements are projected to maximal elements in image.

Conclusions

(This page is intentionally left blank.)