

Checking Under-specified Models

Michael Huth¹

¹Department of Computing
Imperial College London

UK Model Checking Days, York, 27-28 September 2005

1 Under-specified models & applications

2 Research issues

3 References

- **state:** “a .NET component may have a `main` method”

Applications of under-specification

- **state:** “a .NET component may have a `main` method”
- **behavior:** “an audio plug-in may be present in a browser”

Applications of under-specification

- **state:** “a .NET component may have a `main` method”
- **behavior:** “an audio plug-in may be present in a browser”
- **interface:** “requires `balance >= 0`”

Applications of under-specification

- **state:** “a .NET component may have a `main` method”
- **behavior:** “an audio plug-in may be present in a browser”
- **interface:** “requires `balance >= 0`”
- **topology:** “a node may have no neighbor in its broadcast range”

Applications of under-specification

- **state:** “a .NET component may have a `main` method”
- **behavior:** “an audio plug-in may be present in a browser”
- **interface:** “requires `balance >= 0`”
- **topology:** “a node may have no neighbor in its broadcast range”
- **space-time:** “packets will get through in an ad-hoc network if no node is ever hostile.”

Under-specifying propositional models

- models $M: AtomicProp \rightarrow \{0, 1/2, 1\}$, e.g. $[p \mapsto 1/2, q \mapsto 0]$

Under-specifying propositional models

- models $M: AtomicProp \rightarrow \{0, 1/2, 1\}$, e.g. $[p \mapsto 1/2, q \mapsto 0]$
- M' refines M iff $\forall p: M(p) \neq 1/2 \Rightarrow M(p) = M'(p)$, e.g.
 $[p \mapsto 1, q \mapsto 0]$ refines $[p \mapsto 1/2, q \mapsto 0]$

Under-specifying propositional models

- models $M: AtomicProp \rightarrow \{0, 1/2, 1\}$, e.g. $[p \mapsto 1/2, q \mapsto 0]$
- M' refines M iff $\forall p: M(p) \neq 1/2 \Rightarrow M(p) = M'(p)$, e.g. $[p \mapsto 1, q \mapsto 0]$ refines $[p \mapsto 1/2, q \mapsto 0]$
- thorough: $M \models^{th} \phi$ iff all 2-valued refinement of M satisfy ϕ

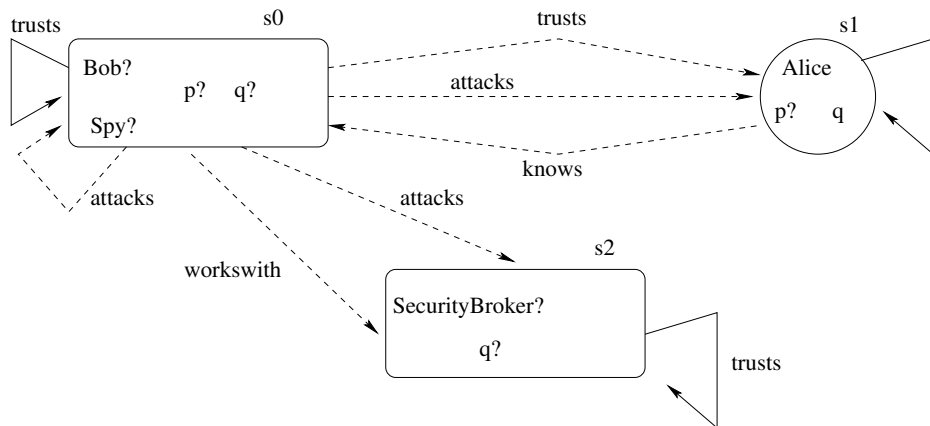
Under-specifying propositional models

- models $M: AtomicProp \rightarrow \{0, 1/2, 1\}$, e.g. $[p \mapsto 1/2, q \mapsto 0]$
- M' refines M iff $\forall p: M(p) \neq 1/2 \Rightarrow M(p) = M'(p)$, e.g. $[p \mapsto 1, q \mapsto 0]$ refines $[p \mapsto 1/2, q \mapsto 0]$
- thorough: $M \models^{th} \phi$ iff all 2-valued refinement of M satisfy ϕ
- **compositional: $M \models^{val} \phi$ interprets 1/2 as 0 (1) in positive (negative) contexts, implies $M \models^{th} \phi$**

Under-specifying propositional models

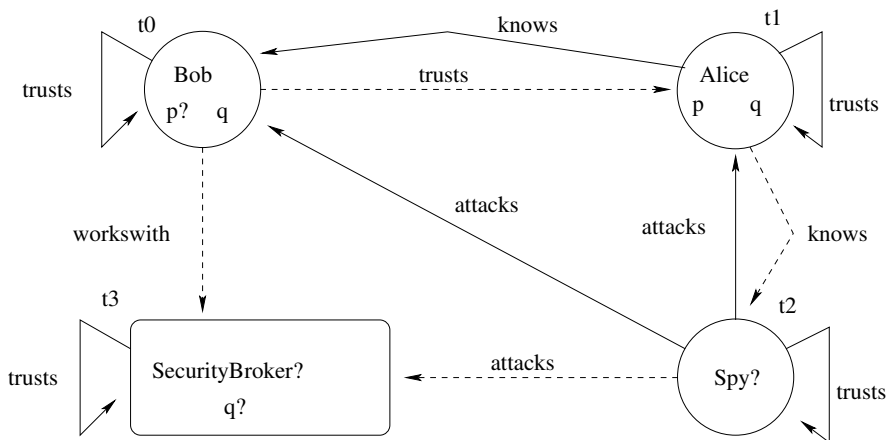
- models $M: AtomicProp \rightarrow \{0, 1/2, 1\}$, e.g. $[p \mapsto 1/2, q \mapsto 0]$
- M' refines M iff $\forall p: M(p) \neq 1/2 \Rightarrow M(p) = M'(p)$, e.g. $[p \mapsto 1, q \mapsto 0]$ refines $[p \mapsto 1/2, q \mapsto 0]$
- thorough: $M \models^{th} \phi$ iff all 2-valued refinement of M satisfy ϕ
- compositional: $M \models^{val} \phi$ interprets 1/2 as 0 (1) in positive (negative) contexts, implies $M \models^{th} \phi$
- loss of precision: $[p \mapsto 1/2] \models^{th} p \vee \neg p$ but $[p \mapsto 1/2] \not\models^{val} p \vee \neg p$

Under-specifying temporal models



? and dashed lines denote value 1/2, solid lines and atoms with no ? have value 1

Refinement



a refinement of the model on previous slide, refinement preserves guarantees and introduces no "new" possibilities

- compositional: $M \models^{val} \phi$ again interprets 1/2 as 0 (1) in negative (positive) contexts, noting $[\alpha]\phi = \neg\langle\alpha\rangle\neg\phi$

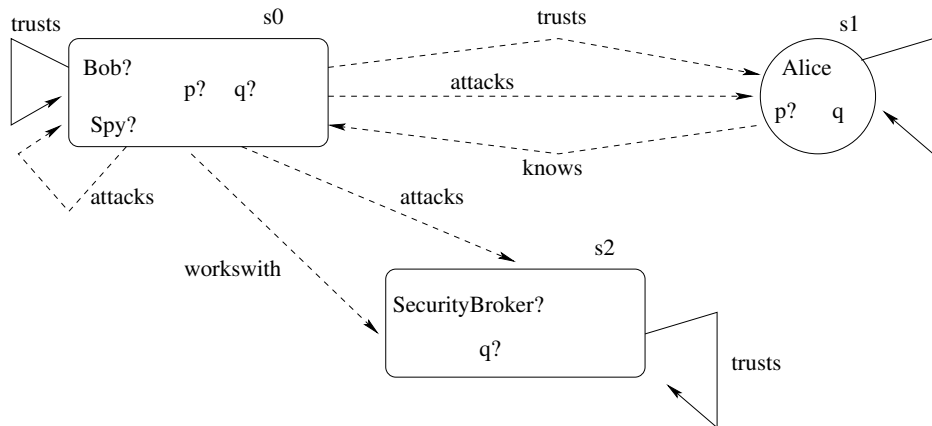
- compositional: $M \models^{val} \phi$ again interprets 1/2 as 0 (1) in negative (positive) contexts, noting $[\alpha]\phi = \neg\langle\alpha\rangle\neg\phi$
- thorough: $M \models^{th} \phi$ iff all 2-valued refinements of M satisfy ϕ

Compositional and thorough semantics

- compositional: $M \models^{val} \phi$ again interprets 1/2 as 0 (1) in negative (positive) contexts, noting $[\alpha]\phi = \neg\langle\alpha\rangle\neg\phi$
- thorough: $M \models^{th} \phi$ iff all 2-valued refinements of M satisfy ϕ
- **compositional semantics sound: $M \models^{val} \phi \Rightarrow M \models^{th} \phi$ for all M, ϕ**

- compositional: $M \models^{val} \phi$ again interprets 1/2 as 0 (1) in negative (positive) contexts, noting $[\alpha]\phi = \neg\langle\alpha\rangle\neg\phi$
- thorough: $M \models^{th} \phi$ iff all 2-valued refinements of M satisfy ϕ
- compositional semantics sound: $M \models^{val} \phi \Rightarrow M \models^{th} \phi$ for all M, ϕ
- **compositional semantics incomplete: inherited from propositional logic, e.g. $\langle\alpha\rangle\phi \vee \neg\langle\alpha\rangle\phi$**

Example re-visited



$s_0 \not\models^{val} \langle \text{attacks} \rangle Alice \vee \neg \langle \text{attacks} \rangle Alice$ as $(s_0, \text{attacks}, s_1)$ possible but $(s_0, \text{attacks}, \cdot)$ not guaranteed

(Ongoing work with Patrice Godefroid.)

- Identify specification patterns ϕ for which

$$\forall M: \quad M \models^{val} \phi \Leftrightarrow M \models^{th} \phi. \quad (1)$$

(Ongoing work with Patrice Godefroid.)

- Identify specification patterns ϕ for which

$$\forall M: \quad M \models^{val} \phi \Leftrightarrow M \models^{th} \phi. \quad (1)$$

- For such ϕ , efficient $M \models^{val} \phi$ is precise & sufficient.

(Ongoing work with Patrice Godefroid.)

- Identify specification patterns ϕ for which

$$\forall M: \quad M \models^{val} \phi \Leftrightarrow M \models^{th} \phi. \quad (1)$$

- For such ϕ , efficient $M \models^{val} \phi$ is precise & sufficient.
- Examples of ϕ satisfying (1):

(Ongoing work with Patrice Godefroid.)

- Identify specification patterns ϕ for which

$$\forall M: \quad M \models^{val} \phi \Leftrightarrow M \models^{th} \phi. \quad (1)$$

- For such ϕ , efficient $M \models^{val} \phi$ is precise & sufficient.
- Examples of ϕ satisfying (1):
 - “ s, t respond to p after q ” and

(Ongoing work with Patrice Godefroid.)

- Identify specification patterns ϕ for which

$$\forall M: \quad M \models^{val} \phi \Leftrightarrow M \models^{th} \phi. \quad (1)$$

- For such ϕ , efficient $M \models^{val} \phi$ is precise & sufficient.
- Examples of ϕ satisfying (1):
 - “ s, t respond to p after q ” and
 - “globally, p becomes true before q ”

(Ongoing work with Patrice Godefroid.)

- Identify specification patterns ϕ for which

$$\forall M: \quad M \models^{val} \phi \Leftrightarrow M \models^{th} \phi. \quad (1)$$

- For such ϕ , efficient $M \models^{val} \phi$ is precise & sufficient.
- Examples of ϕ satisfying (1):
 - “ s, t respond to p after q ” and
 - “globally, p becomes true before q ”
- **Non-example:** $\neg A[(EXp)U(p \rightarrow q)]$ does not enjoy (1).

Multiple-model checking

(Ongoing work with Altaf Hussain.)

- $M \models^{val} \phi$ and $M \models^{th} \phi$ reason about set $\mathcal{C}(M) = \{N \text{ 2-valued} \mid N \text{ refines } M\}$, link to “abstract interpretation.”

Multiple-model checking

(Ongoing work with Altaf Hussain.)

- $M \models^{val} \phi$ and $M \models^{th} \phi$ reason about set $\mathcal{C}(M) = \{N \text{ 2-valued} \mid N \text{ refines } M\}$, link to “abstract interpretation.”
- Requirements engineering, version control etc reason about

$$\bigcap_{i=1}^k \mathcal{C}(M_i). \quad (2)$$

Multiple-model checking

(Ongoing work with Altaf Hussain.)

- $M \models^{val} \phi$ and $M \models^{th} \phi$ reason about set $\mathcal{C}(M) = \{N \text{ 2-valued} \mid N \text{ refines } M\}$, link to “abstract interpretation.”
- Requirements engineering, version control etc reason about

$$\bigcap_{i=1}^k \mathcal{C}(M_i). \quad (2)$$

- For fixed k : have efficient check for consistency, i.e. (2) $\neq \{\}$?

(Ongoing work with Altaf Hussain.)

- $M \models^{val} \phi$ and $M \models^{th} \phi$ reason about set $\mathcal{C}(M) = \{N \text{ 2-valued} \mid N \text{ refines } M\}$, link to “abstract interpretation.”
- Requirements engineering, version control etc reason about

$$\bigcap_{i=1}^k \mathcal{C}(M_i). \quad (2)$$

- For fixed k : have efficient check for consistency, i.e. (2) $\neq \{\}$?
- If all M_i deterministic, (2) representable as $\mathcal{C}(\hat{M})$; not true for non-deterministic M_i , requires tree-automata-like models.

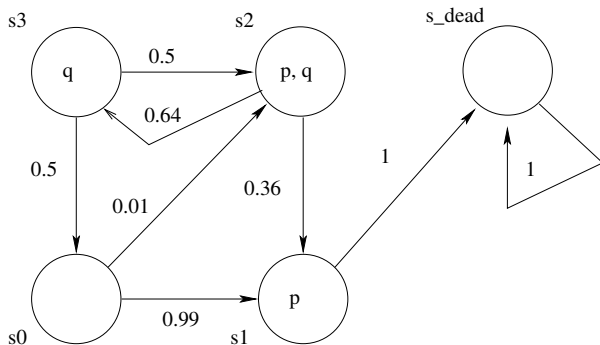
(Ongoing work with Altaf Hussain.)

- $M \models^{val} \phi$ and $M \models^{th} \phi$ reason about set $\mathcal{C}(M) = \{N \text{ 2-valued} \mid N \text{ refines } M\}$, link to “abstract interpretation.”
- Requirements engineering, version control etc reason about

$$\bigcap_{i=1}^k \mathcal{C}(M_i). \quad (2)$$

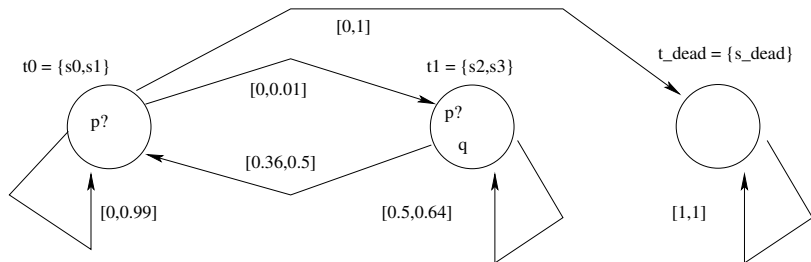
- For fixed k : have efficient check for consistency, i.e. (2) $\neq \{\}$?
- If all M_i deterministic, (2) representable as $\mathcal{C}(\hat{M})$; not true for non-deterministic M_i , requires tree-automata-like models.
- **Seek good analogue of efficient $M \models^{val} \phi$ in this setting.**

A probabilistic system



- discrete-time labeled Markov chain
- transition = probability measure over state space

An abstraction of that probabilistic system



- predicate abstraction of model on previous slide
- intervals approximate non-additive Choquet capacities

- Probabilistic model checking is expensive.

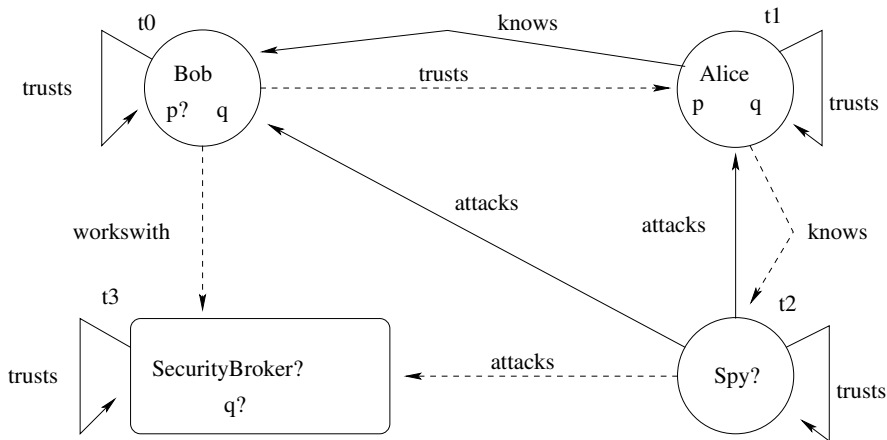
- Probabilistic model checking is expensive.
- Predicate abstraction and CEGAR for probabilistic systems possible?

- Probabilistic model checking is expensive.
- Predicate abstraction and CEGAR for probabilistic systems possible?
- Right abstract structures: measures, Choquet capacities, etc?

- Probabilistic model checking is expensive.
- Predicate abstraction and CEGAR for probabilistic systems possible?
- Right abstract structures: measures, Choquet capacities, etc?
- Complete (i.e. finite state) abstractions for probabilistic CTL or modal mu-calculus?

- Probabilistic model checking is expensive.
- Predicate abstraction and CEGAR for probabilistic systems possible?
- Right abstract structures: measures, Choquet capacities, etc?
- Complete (i.e. finite state) abstractions for probabilistic CTL or modal mu-calculus?
- Optimal finite state abstractions for finite set of properties of some probabilistic logic?

Nominals



How to abstract and model check nominals such as Alice and Bob?
(Ongoing work with Altaf Hussain.)

Some references

- Godefroid, P. & Huth, M. Model Checking Vs. Generalized Model Checking: Semantic Minimizations for Temporal Logics. Proc. of LICS 2005, pages 158–167.
- Huth, M. Refinement is Complete for Implementations. *Formal Aspects of Computing* 17(2):113–137, 2005.
- Huth, M. On Finite-State Approximants for Probabilistic Computation Tree Logic. To appear in *Theoretical Computer Science*, 2005.
- Hussain, A. & Huth, M. On model checking multiple hybrid views. Extended version of paper given at the 1th International Symposium on Leveraging Applications of Formal Methods, Paphos, Cyprus, October 2004. Invited journal submission.
- Detailed references to related and originating work (e.g. by Kim Larsen) can be found in these papers.