Hybrid logics, abstraction, and probabilities

Michael Huth¹

¹Department of Computing
Imperial College London

UCL, Intelligent Systems Seminar, 5 October 2005
1. Hybrid logics & abstraction

2. Abstraction & probabilities

3. Probabilistic nominals

4. References
Hybrid models and logics

- nominals, e.g. Bob, true at exactly one world
- formulas may refer to nominals \( n \), e.g. “at \( n, \phi \)” or “there is a cycle: \( \exists n, EF n \)” (not expressible in modal logic)
- an abstraction of the model above
Applications of abstraction through under-specification

- **state:** “a .NET component may have a main method”
Applications of abstraction through under-specification

- **state**: “a .NET component may have a main method”
- **behavior**: “an audio plug-in may be present in a browser”
Applications of abstraction through 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 abstraction through 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 abstraction through 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 $\geq 0$”
- **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 $M : \text{AtomicProp} \rightarrow \{0, 1/2, 1\}$, e.g. $[p \mapsto 1/2, q \mapsto 0]$
Under-specifying propositional models

- models $M : \text{AtomicProp} \rightarrow \{0, 1/2, 1\}$, e.g. $[p \leftrightarrow 1/2, q \leftrightarrow 0]$
- $M'$ refines $M$ iff $\forall p : M(p) \neq 1/2 \Rightarrow M(p) = M'(p)$, e.g. $[p \leftrightarrow 1, q \leftrightarrow 0]$ refines $[p \leftrightarrow 1/2, q \leftrightarrow 0]$
models $M : \text{AtomicProp} \rightarrow \{0, 1/2, 1\}$, 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 refinements of $M$ satisfy $\phi$
Under-specifying propositional models

- models $M: \text{AtomicProp} \rightarrow \{0, 1/2, 1\}$, 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: $M \models^\text{th} \phi$ iff all 2-valued refinements of $M$ satisfy $\phi$
- compositional: $M \models^\text{val} \phi$ interprets 1/2 as 0 (1) in positive (negative) contexts, implies $M \models^\text{th} \phi$
Under-specifying propositional models

- models $M: \text{AtomicProp} \rightarrow \{0, 1/2, 1\}$, 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 refinements of $M$
  satisfy $\phi$

- compositional: $M \models^{val} \phi$ interprets $1/2$ as $0$ ($1$) in
  positive (negative) contexts, implies $M \models^{th} \phi$

- loss of precision: $[p \mapsto 1/2] \models^{th} p \lor \neg p$ but
  $[p \mapsto 1/2] \not\models^{val} p \lor \neg p$
Hybrid logics, abstraction, and probabilities

Under-specifying temporal models

- abstracts \( \equiv \frac{1}{2} \), abstracts \( \equiv 1 \)
- SecurityBroker possibly true at two worlds, refinement can only realize one choice
Refinement

A refinement of the model on previous slide, refinement preserves guarantees and introduces no “new” possibilities.
Compositional and thorough semantics

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

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

- thorough: $M \models^{th} \phi$ iff all 2-valued refinements of $M$ satisfy $\phi$
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 $\phi$
- compositional semantics sound: $M \models_{val} \phi \Rightarrow M \models_{th} \phi$ for all $M, \phi$
Compositional and thorough semantics

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

\[ s_0 \not\models^{val} \langle \text{attacks} \rangle Alice \lor \neg \langle \text{attacks} \rangle Alice \text{ as } (s_0, \text{attacks}, s_1) \text{ possible but } (s_0, \text{attacks}, \cdot) \text{ not guaranteed } \]
Predicate (functional) abstraction

\[ M = (S, R \subseteq S \times S, L: (AP + Nom) \rightarrow \mathcal{P}(S)) \] hybrid Kripke structure
Predicate (functional) abstraction

- $M = (S, R \subseteq S \times S, L: (AP + Nom) \rightarrow \mathcal{P}(S))$ hybrid Kripke structure
- equivalence relation $s \equiv s'$ iff for all $i = 1, \ldots, n$ ($s \models \phi_i \iff s' \models \phi_i$)
Hybrid logics,
abstraction,
and
probabilities

Huth

Outline

Hybrid logics
& abstraction

Abstraction &
probabilities

Probabilistic
nominals

References

Predicate (functional) abstraction

- $M = (S, R \subseteq S \times S, L : (AP + Nom) \rightarrow \mathbb{P}(S))$ hybrid Kripke structure
- equivalence relation $s \equiv s'$ iff for all $i = 1, \ldots, n$ ($s \models \phi_i \iff s' \models \phi_i$)
- abstract model $(S/\equiv, R^{\exists}, R^{\forall}, L^{\exists}, L^{\forall})$
Predicate (functional) abstraction

- $M = (S, R \subseteq S \times S, L: (AP + Nom) \rightarrow \mathbb{P}(S))$ hybrid Kripke structure
- equivalence relation $s \equiv s'$ iff for all $i = 1, \ldots, n$
  \[ s \models \phi_i \iff s' \models \phi_i \]
- abstract model $(S/\equiv, R^\exists, R^\forall, L^\exists, L^\forall)$
- $tR^\exists t'$ iff $\exists s \in t \exists s' \in t': sRs'$, possible transitions \[ \in \{1/2, 1\} \]
- $tR^\forall t'$ iff $\forall s \in t \exists s' \in t': sRs'$, guaranteed transitions \[ \in \{1\} \]
Predicate (functional) abstraction

- $M = (S, R \subseteq S \times S, L: (AP + \text{Nom}) \rightarrow \mathcal{P}(S))$ hybrid Kripke structure
- equivalence relation $s \equiv s'$ iff for all $i = 1, \ldots, n$
  $\ (s \models \phi_i \iff s' \models \phi_i)$
- abstract model $(S/\equiv, R^\exists, R^\forall, L^\exists, L^\forall)$
- $tR^\exists t'$ iff $\exists s \in t \exists s' \in t': sRs'$, possible transitions
  $\in \{1/2, 1\}$
- $tR^\forall t'$ iff $\forall s \in t \exists s' \in t': sRs'$, guaranteed transitions
  $\in \{1\}$
- $t \in L^\exists(q)$ iff $\exists s \in t: s \in L(q)$, possible labels $\in \{1/2, 1\}$
- $t \in L^\forall(q)$ iff $\forall s \in t: s \in L(q)$, guaranteed labels $\in \{1\}$
Predicate (functional) abstraction

- $M = (S, R \subseteq S \times S, L: (AP + Nom) \rightarrow \mathcal{P}(S))$ hybrid Kripke structure
- equivalence relation $s \equiv s'$ iff for all $i = 1, \ldots, n$
  $\ (s \models \phi_i \iff s' \models \phi_i)$
- abstract model $(S/\equiv, R^\exists, R^\forall, L^\exists, L^\forall)$
- $tR^\exists t'$ iff $\exists s \in t \exists s' \in t': sRs'$, possible transitions
  $\in \{1/2, 1\}$;
- $tR^\forall t'$ iff $\forall s \in t \exists s' \in t': sRs'$, guaranteed transitions
  $\in \{1\}$
- $t \in L^\exists(q)$ iff $\exists s \in t: s \in L(q)$, possible labels $\in \{1/2, 1\}$;
- $t \in L^\forall(q)$ iff $\forall s \in t: s \in L(q)$, guaranteed labels $\in \{1\}$
- obtain: $\forall n \in Nom: L^\forall(n)$ empty or singleton; if singleton, then $L^\exists(n) = L^\forall(n)$;
- morale: turn these constraints into model axioms, even for relational abstractions
Example of predicate abstraction

- predicate abstraction of the model above with $\phi_1 = Alice$ and $\phi_2 = SecurityBroker$ results in this abstraction
Example of predicate abstraction con’t

- the predicate abstraction of the concrete model with 
  $\phi_1 = Alice$ and $\phi_2 = SecurityBroker$
Interlude: multiple-model checking

(ONGOING WORK WITH ALTAF HUSSAIN.)

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

(Ongoing work with Altaf Hussain.)

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

$$\bigcap_{i=1}^{k} C(M_i). \quad (1)$$
Interlude: multiple-model checking

(Ongoing work with Altaf Hussain.)

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

- Requirements engineering, version control etc reason about

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

- For fixed $k$: have efficient check for consistency, i.e. $(1) \neq \{\}$?
Interlude: multiple-model checking

(Ongoing work with Altaf Hussain.)

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

- Requirements engineering, version control etc reason about

$$C(M) = \bigcap_{i=1}^{k} C(M_i). \tag{1}$$

- For fixed $k$: have efficient check for consistency, i.e. (1) $\neq \{\}$?

- If all $M_i$ deterministic, (1) representable as $C(\hat{M})$; not true for non-deterministic $M_i$, requires tree-automata-like models.
Hybrid logics, abstraction, and probabilities

Huth

Outline

Hybrid logics & abstraction

Abstraction & probabilities

Probabilistic nominals

References

Interlude: multiple-model checking

(Ongoing work with Altaf Hussain.)

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

- Requirements engineering, version control etc reason about

\[
\bigcap_{i=1}^{k} C(M_i). \tag{1}
\]

- For fixed \( k \): have efficient check for consistency, i.e. \( (1) \neq \{\} \)?

- If all \( M_i \) deterministic, (1) representable as \( C(\hat{M}) \); not true for non-deterministic \( M_i \), requires tree-automata-like models.

- Seek good analogue of efficient \( M \models^\text{val} \phi \) 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
- nominals (if present) are treated as before
Probabilistic model checking is expensive.
Probabilities and abstraction

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

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

- 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?
Probabilities and abstraction

- 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?
Probabilistic nominals

- probabilistic system as before
- but now nominals governed by probability distribution
- atomic events for nominals of the form “n is at state s”
Hybrid probabilistic computation tree logic

\[ \phi ::= \ldots \quad PCTL \quad \ldots \quad | \; @_n^p \phi \; | \; \downarrow (n, \delta).\phi \; | \; \exists (n, \Delta').\phi \]

\( \Delta' \subseteq \Delta \) set of probability measures

- “at \( n \), \( \phi \) holds with probability \( \sqsupseteq p \):”
  \( s \models_L @_n^p \phi \) iff \( \sum \{ L(n, s') \mid s' \models_L[n \mapsto \delta_{s'}] \phi \} \sqsupseteq p \); reflects conditional probabilities of \( n \)'s being at \( s' \); where \( L[n \mapsto \delta](n) = \delta \) and \( L[n \mapsto \delta](m) = L(m) \) if \( m \neq n \).
Hybrid probabilistic computation tree logic

\[ \phi ::= \ldots PCTL \ldots \mid \Diamond_n^p \phi \mid \downarrow (n, \delta).\phi \mid \exists (n, \Delta').\phi \]

\(\Delta' \subseteq \Delta\) set of probability measures

- “at \(n\), \(\phi\) holds with probability \(\equiv p\):”
  \[ s \models_L \Diamond_n^p \phi \text{ iff } \sum \{ L(n, s') \mid s' \models_L [n \mapsto \delta_{s'}] \phi \} \equiv p; \text{ reflects conditional probabilities of } n\text{'s being at } s'; \text{ where} \]
  \[ L[n \mapsto \delta](n) = \delta \text{ and } L[n \mapsto \delta](m) = L(m) \text{ if } m \neq n \]

- “if \(n\) is rebound to \(\delta\), \(\phi\) holds:”
  \[ s \models_L \downarrow (n, \delta).\phi \text{ iff } s \models_L [n \mapsto \delta] \phi \]
Hybrid probabilistic computation tree logic

\[ \phi ::= \ldots \ PCTL \ldots \mid @_nP.\phi \mid \downarrow(n, \delta).\phi \mid \exists(n, \Delta').\phi \]

\[ \Delta' \subseteq \Delta \text{ set of probability measures} \]

- “at } n, \phi \text{ holds with probability } \equiv p:\”
  \[ s \models_L @_nP.\phi \text{ iff } \sum \{ L(n, s') \mid s' \models_L[n \mapsto \delta_{s'}] \phi \} \equiv p; \text{ reflects conditional probabilities of } n\text{’s being at } s'; \text{ where } L[n \mapsto \delta](n) = \delta \text{ and } L[n \mapsto \delta](m) = L(m) \text{ if } m \neq n \]

- “if } n \text{ is rebound to } \delta, \phi \text{ holds:”}
  \[ s \models_L \downarrow(n, \delta).\phi \text{ iff } s \models_L[n \mapsto \delta] \phi \]

- “it is possible to rebind } n \text{ in } \Delta' \text{ such that } \phi \text{ holds:”}
  \[ s \models_L \exists(n, \Delta').\phi \text{ iff for some } \delta \in \Delta': \ s \models_L[n \mapsto \delta] \phi \]
expressiveness of hybrid PCTL

- subsumes \((M, s) \models_L \downarrow n.\phi\) through \((M, s) \models_L (n, \delta_s).\phi\)
expressiveness of hybrid PCTL

- subsumes \((M, s) \models_L \downarrow n.\phi\) through \((M, s) \models_L (n, \delta_s).\phi\)
- subsumes \((M, s) \models_L \exists n.\phi\) through \((M, s) \models_L \exists(n, \{\delta_t \mid t \in \Sigma\}).\phi\)
expressiveness of hybrid PCTL

- subsumes \((M, s) \models_L \downarrow n.\phi\) through \((M, s) \models_L \downarrow (n, \delta_s).\phi\)
- subsumes \((M, s) \models_L \exists n.\phi\) through
  \((M, s) \models_L \exists(n, \{\delta_t \mid t \in \Sigma\}).\phi\)
- can express probabilistic recurrence, e.g. that state \(s\) is on a cycle with probability at least \(0.9999\), as

\[
s \models_L \downarrow (n, \delta_s).[true \cup n] \geq 0.9999\]
Example probabilistic check

\[ s_3 \models L \otimes_{n_3}^{>0.1} [true \cup n_3] \geq 0.01 \]

i.e. is sum of all \( L(n_3, s) \), with \( s \models L[s \mapsto \delta_s] [true \cup n_3] \geq 0.01 \), \( \geq 0.1 \)?

only \( s_0 \) and \( s_2 \) are relevant states \( s \) here.
at $s_0$ for $L[n_3 \rightarrow \delta_{s_0}]$, the probability that $s_0$ is on a cycle is $0.01 \cdot 0.64 \cdot 0.5 \cdot (\sum_{i=0}^{\infty} (0.64 \cdot 0.5)^i) = 0.00948529 \cdots \geq 0.01$ so $L(s_0, n_3) = 0.87$ does not contribute to that sum.
at $s_2$ for $L[n_3 \mapsto \delta_{s_2}]$, the probability that $s_2$ is on a cycle is $0.64 \cdot 0.5 + 0.64 \cdot 0.5 \cdot 0.01 = 0.3232 \geq 0.01$ so $L(s_2, n_3) = 0.13$ is only contributor to that sum $\Rightarrow s_3 \models_L \@^{>0.1}[true \cup n_3] \geq 0.01$ holds as $0.13 > 0.1$
Some references


- Detailed references to related and originating work (e.g. by Kim Larsen) can be found in these papers.