

Algorithms for Model Checking

2IW50

Lecture 7: Data Abstraction (Chapter 13)

Jaco van de Pol

`vdpol@cwi.nl`

`http://www.cwi.nl/~vdpol/amc.html`

Technische Universiteit Eindhoven
Centrum voor Wiskunde en Informatica

Situation

- We have seen:
 - If $M_1 \equiv M_2$ (M_1 and M_2 are bisimilar), then M_1 and M_2 satisfy the same CTL* formulas
 - If $M_1 \preceq M_2$ (M_2 simulates M_1), then all ACTL* formulas valid for M_2 hold for M_1 as well
- Consider a specification \mathcal{S} , with a very large state space M_1 .
- The question today is how to compute an M_2 such that:
 - $M_1 \preceq M_2$ (i.e.: useful for ACTL* formulas)
 - M_2 is smaller than M_1
 - without generating M_1 first.
- The approach will be **data abstraction**

O V E R V I E W

1. Refresh general specification language
2. Idea of data abstraction
3. Data abstraction for Kripke Structures
4. Data abstraction for Specifications
5. Example:
Bounded Retransmission Protocol

System Specification by formulas

The system specification is given by formulas from **first-order predicate logic**.

A specification is a tuple $(V, D, \mathcal{S}_0, \mathcal{R})$, where

- V is a set of **variables** v_1, \dots, v_n
- D is the **domain** of these variables.
- The formula $\mathcal{S}_0(V)$ represents the initial states
- Let V' be a copy of the variables: v'_1, \dots, v'_n
- The formula $\mathcal{R}(V, V')$ represents the transition relation.
 - V denotes the value of variables **before** the transition,
 - V' denotes the value of variables **after** the transition.

Here $\phi(V)$ denotes a first order formula, with all its free variables among V . Given a valuation $\alpha : V \rightarrow D$, the semantics of ϕ under α is written as $\llbracket \phi \rrbracket_\alpha \in \{true, false\}$.

Semantics by Kripke Structure

- Given a specification $(V, D, \mathcal{S}_0, \mathcal{R})$, we get the underlying Kripke structure (state space) (AP, S, S_0, R, L) as follows:
- The set of **states** S of the Kripke Structure will be the set of valuations $\alpha : V \rightarrow D$.
- The **atomic propositions** (AP) will be of the form $v = d$, where $v \in V$ and $d \in D$.
- The set of **initial states** S_0 will be the valuations α , such that $\llbracket \mathcal{S}_0 \rrbracket_\alpha$ is true.
- Let α and β be valuations $V \rightarrow D$. Define $\beta'(v'_i) = \beta(v_i)$ (for all i). There is a **transition** $R(\alpha, \beta)$, if $\llbracket \mathcal{R} \rrbracket_{\alpha \cup \beta'}$ is true.
- Finally, the set of **labels** of state α is $L(\alpha) := \{v_1 = \alpha(v_1), \dots, v_n = \alpha(v_n)\}$

Idea of Data Abstraction

- State variables range over some domain D .
- This is now called the **concrete domain**.
- A data abstraction consists of:
 - An **abstract** domain A
 - A surjective mapping $h : D \rightarrow A$.
- Example:
 - Concrete domain: \mathbb{N} (natural numbers)
 - Abstract domain: $\{even, odd\}$
 - $h(2n) = even, h(2n + 1) = odd$.
- **Abstract interpretation**: the art of evaluating an expression directly in the abstract domain.
- For this, all “concrete operations” must be mimicked by “abstract operations”
- In the example: $\llbracket m + n \rrbracket$ can be computed directly as $\llbracket m \rrbracket \hat{+} \llbracket n \rrbracket$ if we set

$even \hat{+} even = even$	$odd \hat{+} odd = even$
$even \hat{+} odd = odd$	$odd \hat{+} even = odd$

Abstraction for Kripke Structures

Let Kripke Structure (AP, S, S_0, R, L) be given, where $S = V \rightarrow D$.

Let abstract domain A with $h : D \rightarrow A$ be given.

- The idea will be:
 - to replace concrete by abstract data values
 - collapse states with the same abstract values
- The price to be paid is that we can now only express properties on abstract values.
 - Let \hat{v} denote the abstract version of variable v , ranging over A .
 - Change the set of atomic propositions:
 $\widehat{AP} := \{\hat{v} = a \mid v \in V, a \in A\}$.
 - Change the labeling:
 $L'(s) := \{\dots, \hat{v}_i = h(s), \dots\}$

- We will now compare the concrete $M' = (\widehat{AP}, S, S_0, R, L')$ with the abstract $M_r = (\widehat{AP}, S_r, S_0^r, R_r, L_r)$ to be defined next.

Abstraction for Kripke Structures (2)

Given the concrete $M' = (\widehat{AP}, S, S_0, R, L')$, we define the abstract $M_r = (\widehat{AP}, S_r, S_0^r, R_r, L_r)$:

- We want to collapse (identify) states with the same labels, so we define:

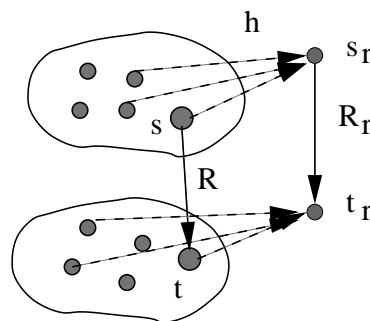
$$S_r := \{L'(s) \mid s \in S\} \quad (= \widehat{V} \rightarrow A)$$

- Of course, $L_r(s_r) := s_r$.
- M_r must simulate M' , so it should reflect all initial states:

$$S_0^r(s_r) \text{ :iff } \exists s \in S_0. L'(s) = s_r$$

- M_r must simulate M' , so it should reflect all transitions. So define $R_r(s_r, t_r)$ iff

$$\exists s, t \in S. s_r = L'(s) \wedge t_r = L'(t) \wedge R(s, t)$$



Abstraction at Specification Level

- Given a specification $(V, D, \mathcal{S}_0, \mathcal{R})$, and a data abstraction (A, h) , we want a spec for M_r .
- **Notation:** Write $[\phi](\widehat{V})$ to abbreviate the ideal abstraction of $\phi(V)$:

$$[\phi](\widehat{v}_1, \dots, \widehat{v}_n) :=$$

$$\exists v_1 \cdots \exists v_n.$$

$$h(v_1) = \widehat{v}_1 \wedge \cdots \wedge h(v_n) = \widehat{v}_n \wedge$$

$$\phi(v_1, \dots, v_n)$$

- Then the ideal abstract specification is defined as $(\widehat{V}, A, [\mathcal{S}_0(V)], [\mathcal{R}(V, V')])$,
- However, due to all the existential quantifications, this is costly to generate.
- Recall that in OBDDs, each $\exists x.B$ is implemented as $B|_{x=0} \vee B|_{x=1}$, possibly doubling the size.

Abstraction at Specification Level (2)

- Idea: generate an approximation more cheaply. by **pushing** the quantifications **inside**.
- Assume that \mathcal{S}_0 and \mathcal{R} are in positive normal form. We define $\mathcal{A}(\phi)$, the **abstract interpretation** of ϕ as follows:
 - $\mathcal{A}(P(x_1, \dots, x_m)) = [P](\widehat{x}_1, \dots, \widehat{x}_n)$
 - $\mathcal{A}(\neg P(x_1, \dots, x_m)) = [\neg P](\widehat{x}_1, \dots, \widehat{x}_n)$
 - $\mathcal{A}(\phi_1 \wedge \phi_2) = \mathcal{A}(\phi_1) \wedge \mathcal{A}(\phi_2)$
 - $\mathcal{A}(\phi_1 \vee \phi_2) = \mathcal{A}(\phi_1) \vee \mathcal{A}(\phi_2)$
 - $\mathcal{A}(\exists x. \phi) = \exists \widehat{x}. \mathcal{A}(\phi)$
 - $\mathcal{A}(\forall x. \phi) = \forall \widehat{x}. \mathcal{A}(\phi)$
- By induction on ϕ one can easily show that $[\phi] \Rightarrow \mathcal{A}(\phi)$ (the other way doesn't hold). (Here we use that h is surjective!)
- So given spec $(V, D, \mathcal{S}_0, \mathcal{R})$, we define its abstract version to be $(\widehat{V}, A, \mathcal{A}(\mathcal{S}_0), \mathcal{A}(\mathcal{R}))$.

Correctness of abstraction

- Let M' be the Kripke structure over \widehat{AP} , induced by specification (V, D, S_0, \mathcal{R}) .
- Let M_a be the Kripke structure induced by the abstract specification $(\widehat{V}, A, \mathcal{A}(S_0), \mathcal{A}(\mathcal{R}))$.
- **Claim:** $M' \preceq M_a$.
- **Proof:** the following H is a simulation relation: Let $s = \{v_1 = d_1, \dots, v_n = d_n\}$ and $s_a = \{\widehat{v}_1 = a_1, \dots, \widehat{v}_n = a_n\}$.

$$H(s, s_a) = \forall i (1 \leq i \leq n). h(d_i) = a_i$$

- By definition the (abstract) labels coincide.
- Let $s \rightarrow t$ in M' . Then $\mathcal{R}(s, t)$ holds, hence $[\mathcal{R}](h(s), h(t))$, hence $\mathcal{A}(\mathcal{R}(h(s), h(t)))$ holds. So $h(s) \rightarrow h(t)$ in M_a . Note that $H(s, h(s))$ and $H(t, h(t))$.
- Similarly if $s \in S_0$, then $s \in \mathcal{A}(S_0)$.

- Hence data abstraction is sound for ACTL*.