Static Program Analysis

Lecture 13: Abstract Interpretation II (Abstract Semantics of WHILE)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH Aachen University

noll@cs.rwth-aachen.de

http://www-i2.informatik.rwth-aachen.de/i2/spa11/

Summer Semester 2011

Informatik-Kolloquium

Fachgruppe Informatik

Lehrstuhl für Informatik 2

RWTH
RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

EINLADUNG

Zeit: Mittwoch, 01. Juni 2011, 15.00 Uhr

Ort: AH IV, Ahornstr. 55

Referentin: Frau Prof. Dr. Barbara M. Terhal,
Institute for Quantum Information, FB1,
RWTH Aachen

Titel: Quantum Complexity Theory

Abstract:

We will discuss the model of quantum computation and the important differences between quantum and classical computation. We then review the quantum version of the classical Cook-Levin theorem due to Kitaev which proves that a quantum (or matrix) version of the satisfiability problem is quantum NP or QMA-complete. We give an overview of more recent advances in quantum complexity theory and discuss various open questions.

Es laden ein: Die Dozenten der Informatik

Outline

Abstract Interpretation

- **Summary:** a theory of sound approximation of the semantics of programs
- **Basic idea:** execution of program on abstract values
  (similar to type-level bytecode interpreter)
- **Example:** parity (even/odd) rather than concrete numbers
- **Procedure:** run program on finite set of abstract values that cover all concrete inputs using abstract operations that cover all concrete outputs

  $\Longrightarrow$ soundness of approach
- **Preciseness** of information again characterized by partial order

Galois Connections I

Definition (Galois connection)

Let $(L, \sqsubseteq_L)$ and $(M, \sqsubseteq_M)$ be complete lattices. A pair (α, γ) of monotonic functions

$$\alpha : L \to M \qquad \text{and} \qquad \gamma : M \to L$$

is called a Galois connection if

$$\forall l \in L : l \sqsubseteq_L \gamma(\alpha(l)) \qquad \text{and} \qquad \forall m \in M : \alpha(\gamma(m)) \sqsubseteq_M m$$

Interpretation:

- $L = \{\text{sets of concrete values}\}$, $M = \{\text{sets of abstract values}\}$
- $\alpha =$ abstraction function, $\gamma =$ concretization function
- $l \sqsubseteq_L \gamma(\alpha(l))$: α yields over-approximation
- $\alpha(\gamma(m)) \sqsubseteq_M m$: no loss of precision by abstraction after concretization
- Usually: $l \neq \gamma(\alpha(l))$, $\alpha(\gamma(m)) = m$

Outline

Semantics of WHILE I

Definition (Execution relation for statements)

If $c \in \mathit{Cmd}$ and $\sigma \in \Sigma$, then $\langle c, \sigma \rangle$ is called a configuration. The execution relation (on configurations and states) is defined by the following rules:

$$(\text{skip})\ \frac{}{\langle \texttt{skip}, \sigma \rangle \to \sigma}$$

$$(\text{asgn})\ \frac{}{\langle x := a, \sigma \rangle \to \sigma[x \mapsto \mathit{val}_\sigma(a)]}$$

$$(\text{seq1})\ \frac{\langle c_1, \sigma \rangle \to \langle c_1', \sigma' \rangle}{\langle c_1 ; c_2, \sigma \rangle \to \langle c_1' ; c_2, \sigma' \rangle}$$

$$(\text{seq2})\ \frac{\langle c_1, \sigma \rangle \to \sigma'}{\langle c_1 ; c_2, \sigma \rangle \to \langle c_2, \sigma' \rangle}$$

Semantics of WHILE II

Definition (Execution relation for statements; continued)

$$(\text{if1})\ \frac{val_\sigma(b) = \text{true}}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \to \langle c_1, \sigma\rangle}$$

$$(\text{if2})\ \frac{val_\sigma(b) = \text{false}}{\langle \texttt{if } b \texttt{ then } c_1 \texttt{ else } c_2, \sigma\rangle \to \langle c_2, \sigma\rangle}$$

$$(\text{wh1})\ \frac{val_\sigma(b) = \text{true}}{\langle \texttt{while } b \texttt{ do } c, \sigma\rangle \to \langle c \texttt{;while } b \texttt{ do } c, \sigma\rangle}$$

$$(\text{wh2})\ \frac{val_\sigma(b) = \text{false}}{\langle \texttt{while } b \texttt{ do } c, \sigma\rangle \to \sigma}$$

Outline

An Execution Example

Example 13.1

- $c := \texttt{y := 1; while } \underbrace{\neg(\texttt{x=1})}_{b} \texttt{ do } \underbrace{\underbrace{\texttt{y := y*x}}_{c_1}\texttt{; }\underbrace{\texttt{x := x-1}}_{c_2}}_{c_0}$
- Claim: $\langle c, \sigma \rangle \to^+ \sigma_{1,6}$ for every $\sigma \in \Sigma$ with $\sigma(\texttt{x}) = 3$
- Notation: $\sigma_{i,j}$ means $\sigma(\texttt{x}) = i$, $\sigma(\texttt{y}) = j$
- Derivation: on the board

Determinism Property of Execution Relation

This operational semantics is well defined in the following sense:

Theorem 13.2

The execution relation for statements is deterministic, i.e., whenever $c \in \mathit{Cmd}$, $\sigma \in \Sigma$ *and* $\kappa_1, \kappa_2 \in \mathit{Cmd} \times \Sigma \cup \Sigma$ *such that* $\langle c, \sigma \rangle \to \kappa_1$ *and* $\langle c, \sigma \rangle \to \kappa_2$, *then* $\kappa_1 = \kappa_2$.

Proof.

omitted □

Outline

Safe Approximation of Functions I

Definition 13.3

Let (α, γ) be a Galois connection with $\alpha : L \to M$ and $\gamma : M \to L$, and let $f : L^n \to L$ and $f^\# : M^n \to M$ be functions of rank $n \in \mathbb{N}$. Then $f^\#$ is called a safe approximation of f if, whenever $m_1, \ldots, m_n \in M$,

$$\alpha(f(\gamma(m_1), \ldots, \gamma(m_n))) \sqsubseteq_M f^\#(m_1, \ldots, m_n).$$

Moreover it is called most precise safe approximation if the reverse inclusion is also true.

- **Interpretation:** the abstraction $f^\#$ of f covers all concrete results
- **Note:** monotonicity of f and/or $f^\#$ is *not* required (but usually given; see Lemma 13.5)

Safe Approximation of Functions II

Example 13.4

1. Parity abstraction (cf. Example 12.2): most precise approximations
   - $n = 0$: $1^\# = \{\text{odd}\}$
   - $n = 1$: $-^\#(P) = P$, $(-1)^\#(\{\text{even}\}) = \{\text{odd}\}$
   - $n = 2$: $\{\text{even}\} +^\# \{\text{odd}\} = \{\text{odd}\}$, $\{\text{even}\} \cdot^\# \{\text{odd}\} = \{\text{even}\}$
2. Sign abstraction (cf. Example 12.3): most precise approximations
   - $n = 0$: $1^\# = \{+\}$
   - $n = 1$: $-^\#(\{+\}) = \{-\}$, $(-1)^\#(\{+\}) = \{+, 0\}$
   - $n = 2$: $\{+\} +^\# \{+\} = \{+\}$, $\{+\} +^\# \{-\} = \{+, -, 0\}$
     $\{+\} \cdot^\# \{-\} = \{-\}$
3. Interval abstraction (cf. Example 12.4): most precise approximations
   - $n = 0$: $z^\# = [z, z]$
   - $n = 1$: $-^\#([z_1, z_2]) = [-z_2, -z_1]$, $(-1)^\#([z_1, z_2]) = [z_1 - 1, z_2 - 1]$
   - $n = 2$:
     - $[y_1, y_2] +^\# [z_1, z_2] = [y_1 + z_1, y_2 + z_2]$
     - $[y_1, y_2] -^\# [z_1, z_2] = [y_1 - z_2, y_2 - z_1]$

Safe Approximation of Functions III

Lemma 13.5

If $f : L^n \to L$ and $f^\# : M^n \to M$ are monotonic, then $f^\#$ is a safe approximation of f iff, for all $l_1, \ldots, l_n \in L$,

$$\alpha(f(l_1, \ldots, l_n)) \sqsubseteq_M f^\#(\alpha(l_1), \ldots, \alpha(l_n)).$$

Proof.

on the board □

Safe Approximation of Execution Relation I

- **Reminder:** concrete semantics of WHILE
  - states $\Sigma := \{\sigma \mid \sigma : \mathit{Var} \to \mathbb{Z}\}$ (Definition 12.6)
  - execution relation $\to \subseteq (\mathit{Cmd} \times \Sigma) \times (\mathit{Cmd} \times \Sigma \cup \Sigma)$ (Definition 12.9)
- Yields concrete domain $L := 2^{\Sigma}$ and concrete transition function:

Definition 13.6 (Concrete transition function)

The concrete transition function of WHILE is defined by the family of functions

$$\mathrm{next}_{c,c'} : 2^{\Sigma} \to 2^{\Sigma}$$

where $c \in \mathit{Cmd}$, $c' \in \mathit{Cmd} \cup \{\downarrow\}$ and, for every $S \subseteq \Sigma$,

$$\mathrm{next}_{c,c'}(S) := \{\sigma' \in \Sigma \mid c' \in \mathit{Cmd}, \exists \sigma \in S : \langle c, \sigma \rangle \to \langle c', \sigma' \rangle\} \text{ and}$$
$$\mathrm{next}_{c,\downarrow}(S) := \{\sigma' \in \Sigma \mid \exists \sigma \in S : \langle c, \sigma \rangle \to \sigma'\}$$

Safe Approximation of Execution Relation II

- **Reminder:** abstraction determined by Galois connection (α, γ) with $\alpha : L \to M$ and $\gamma : M \to L$
  - here: $L := 2^{\Sigma}$, M not fixed (usually $M = Var \to \ldots$ or $M = 2^{Var \to \cdots}$)
  - write Abs in place of M
  - thus $\alpha : 2^{\Sigma} \to Abs$ and $\gamma : Abs \to 2^{\Sigma}$
- Yields abstract semantics:

Definition 13.7 (Abstract semantics of WHILE)

Given $\alpha : 2^{\Sigma} \to Abs$, an abstract semantics is defined by a family of functions

$$\text{next}^{\#}_{c,c'} : Abs \to Abs$$

where $c \in Cmd$, $c' \in Cmd \cup \{\downarrow\}$, and each $\text{next}^{\#}_{c,c'}$ is a safe approximation of $\text{next}_{c,c'}$, i.e.,

$$\alpha(\text{next}_{c,c'}(\gamma(abs))) \sqsubseteq_{Abs} \text{next}^{\#}_{c,c'}(abs)$$

for every $abs \in Abs$. Notation:

- $\langle c, abs \rangle \Rightarrow \langle c', abs' \rangle$ for $\text{next}^{\#}_{c,c'}(abs) = abs'$ and
- $\langle c, abs \rangle \Rightarrow abs'$ for $\text{next}^{\#}_{c,\downarrow}(a) = abs'$