Semantics of Programming Languages
Solutions to Assignment 9

Patrick Maier, Jan Schwinghammer
http://www.ps.uni-sb.de/courses/sem-ws01/

INFORMATIK

Additionally to the declarations on the exercise sheet, let $\mathit{Store} = \mathit{Val}^{\mathit{Var}}$ and $PP = \{1, 2\}$.

Exercise 9.1 Collecting semantics as system of equations:

$$\begin{aligned} \mathit{acc}(1) &= S_0 \cup \mathit{assign}_{z:=x+y}(\mathit{acc}(2)) \\ \mathit{acc}(2) &= \mathit{dec}_y(\mathit{inc}_x(\mathit{acc}(1))) \end{aligned}$$

where the initial stores $S_0 = \{s_0\}$ with $s_0(x) = 17$, $s_0(y) = 4$, $s_0(z) = 21$, and

$$\begin{aligned} \mathit{assign}_{z:=x+y}(S) &= \{s' \in \mathit{Store} \mid \exists s \in S : s'(x) = s(x), s'(y) = s(y), s'(z) = s(x) + s(y)\}, \\ \mathit{inc}_x(S) &= \{s' \in \mathit{Store} \mid \exists s \in S : s'(x) = s(x) + 1, s'(y) = s(y), s'(z) = s(z)\}, \\ \mathit{dec}_x(S) &= \{s' \in \mathit{Store} \mid \exists s \in S : s'(x) = s(x), s'(y) = s(y) - 1, s'(z) = s(z)\} \end{aligned}$$

The least solution of the above system of equations (which is reached after running the program for an infinite time) is acc with

$$\begin{aligned} \mathit{acc}(1) &= \{s \in \mathit{Store} \mid s(x) \geq 17, s(y) \leq 4, s(z) = s(x) + s(y) = 21\} \\ \mathit{acc}(2) &= \{s \in \mathit{Store} \mid s(x) > 17, s(y) < 4, s(z) = s(x) + s(y) = 21\} \end{aligned}$$

Parity abstraction: Let $\langle P, \sqsubseteq\rangle$ with $P = \{\mathit{even}, \mathit{odd}, \bot, \top\}$ be the flat four-element lattice (i. e., *even* and *odd* are incomparable). Then $D^\# = P^{\mathit{Var}}$ ordered by the pointwise extension of $\sqsubseteq$ (which we will also denote by $\sqsubseteq$) is also a lattice. Abstraction α maps a set of stores $S \in 2^{\mathit{Store}}$ to an abstract store $a \in D^\#$ where for all $u \in \mathit{Var}$,

$$a(u) = \begin{cases} \bot & \text{if } S = \emptyset \\ \mathit{even} & \text{if } S \neq \emptyset \text{ and } \forall s \in S : s(u) \text{ even} \\ \mathit{odd} & \text{if } S \neq \emptyset \text{ and } \forall s \in S : s(u) \text{ odd} \\ \top & \text{otherwise} \end{cases}$$

Concretization γ maps an abstract store $a \in D^\#$ to a set of stores

$$\begin{aligned} \gamma(a) = \{s \in \mathit{Store} \mid \forall u \in \mathit{Var} : a(u) \neq \bot \text{ and } a(u) = \mathit{even} \Rightarrow s(u) \text{ even} \\ \text{and } a(u) = \mathit{odd} \Rightarrow s(u) \text{ odd}\} \end{aligned}$$

Abstract semantics (parity semantics) as system of equations:

$$\begin{aligned} \mathit{acc}^\#(1) &= a_0 \sqcup \mathit{assign}^\#_{z:=x+y}(\mathit{acc}^\#(2)) \\ \mathit{acc}^\#(2) &= \mathit{dec}^\#_y(\mathit{inc}^\#_x(\mathit{acc}^\#(1))) \end{aligned}$$

where a_0 is the initial abstract store with $a_0(x) = \mathit{odd}$, $a_0(y) = \mathit{even}$, $a_0(z) = \mathit{odd}$, and the abstract operations are $\mathit{assign}^{\#}_{z:=x+y}(a) = a'$ with $a'(x) = a(x)$, $a'(y) = a(y)$ and

$$a'(z) = \begin{cases} \bot & \text{if } a(x) = \bot \text{ or } a(y) = \bot \\ \mathit{even} & \text{if } a(x), a(y) \in \{\mathit{even}, \mathit{odd}\} \text{ and } a(x) = a(y) \\ \mathit{odd} & \text{if } a(x), a(y) \in \{\mathit{even}, \mathit{odd}\} \text{ and } a(x) \neq a(y) \\ \top & \text{otherwise} \end{cases}$$

Similarly, $\mathit{inc}^{\#}_x(a) = a'$ with $a'(y) = a(y)$, $a'(z) = a(z)$, and

$$a'(x) = \begin{cases} \bot & \text{if } a(x) = \bot \\ \mathit{even} & \text{if } a(x) = \mathit{odd} \\ \mathit{odd} & \text{if } a(x) = \mathit{even} \\ \top & \text{otherwise} \end{cases}$$

In the same manner, $\mathit{dec}^{\#}_y$ is defined.

The least solution of the system of abstract equations is $\mathit{acc}^{\#}$ where $\mathit{acc}^{\#}(1) = \mathit{acc}^{\#}(2) = a$ with $a(x) = a(y) = a(z) = \top$, i. e., the least solution is also the largest (hence least informative) solution. Note that this least solution is reached after two iterations of the loop body.

Exercise 9.2 Let $P = \{\mathit{even}, \mathit{odd}\}$. We define our new (disjunctive) abstract domain $\langle D^{\#}, \subseteq\rangle$ as $D^{\#} = 2^{(P^{\mathit{Var}})}$ ordered by inclusion $\subseteq$, i. e., abstract elements $A \in D^{\#}$ are sets of abstract stores in P^{Var}. The abstraction $\alpha : 2^{\mathit{Store}} \to D^{\#}$ is defined as $\alpha(S) = \bigcup\{\mathit{parity} \circ s \mid s \in S\}$ and the concretization $\gamma : D^{\#} \to 2^{\mathit{Store}}$ as $\gamma(A) = \{s \in \mathit{Store} \mid \exists a \in A : a = \mathit{parity} \circ s\}$. Here, the function $\mathit{parity} : \mathbb{Z} \to P$ is defined as usual, i. e., for all $i \in \mathbb{Z}$, $\mathit{parity}(i) = \mathit{even}$ if i is even and $\mathit{parity}(i) = \mathit{odd}$ otherwise.

Disjunctive parity semantics as system of equations:

$$\mathit{acc}^{\#}(1) = A_0 \cup \mathit{assign}^{\#}_{z:=x+y}(\mathit{acc}^{\#}(2))$$
$$\mathit{acc}^{\#}(2) = \mathit{dec}^{\#}_y(\mathit{inc}^{\#}_x(\mathit{acc}^{\#}(1)))$$

where $A_0 = \{a_0\}$ is the initial set of abstract stores with $a_0(x) = \mathit{odd}$, $a_0(y) = \mathit{even}$, $a_0(z) = \mathit{odd}$, and the abstract operations are

$$\mathit{inc}^{\#}_x(A) = \{a' \in P^{\mathit{Var}} \mid \exists a \in A : a'(x) \neq a(x), a'(y) = a(y), a'(z) = a(z)\},$$
$$\mathit{dec}^{\#}_x(A) = \{a' \in P^{\mathit{Var}} \mid \exists a \in A : a'(x) = a(x), a'(y) \neq a(y), a'(z) = a(z)\},$$
$$\mathit{assign}^{\#}_{z:=x+y}(A) = \{a' \in P^{\mathit{Var}} \mid \exists a \in A : a'(x) = a(x), a'(y) = a(y), a'(z) = a(x) +_P a(y)\},$$

where for all $s, t \in P$, $s +_P t = \mathit{even}$ if $s = t$ and $s +_P t = \mathit{odd}$ otherwise.

To compute the least solution $\mathit{acc}^{\#} = \langle A_1, A_2\rangle$ of the above system of equations, we approximate from below by $\langle A^i_1, A^i_2\rangle$ for $i = 0, 1, 2, 3, \ldots$ iterations of the loop:

$$\langle A^0_1, A^0_2\rangle = \langle\emptyset, \emptyset\rangle$$
$$\langle A^1_1, A^1_2\rangle = \langle\{a_0\}, \{a_1\}\rangle$$
$$\langle A^2_1, A^2_2\rangle = \langle\{a_0, a_1\}, \{a_0, a_1\}\rangle$$
$$\langle A^3_1, A^3_2\rangle = \langle\{a_0, a_1\}, \{a_0, a_1\}\rangle$$

where $a_0(x) = a_0(z) = odd$, $a_0(y) = even$ and $a_0(x) = even$, $a_0(y) = a_0(z) = odd$.

Exercise 9.3 Collecting semantics $acc : PP \to 2^{Store}$:

- $acc(p') = \{s' \in Store \mid \exists s \in acc(p) : s'(x) = c \text{ and } \forall u \in Var \setminus \{x\} : s'(u) = s(u)\}$
- $acc(p') = \{s' \in Store \mid \exists s \in acc(p) : s'(x) = s(y) + c \text{ and } \forall u \in Var \setminus \{x\} : s'(u) = s(u)\}$
- $acc(p') = \{s' \in Store \mid \exists s \in acc(p) : s'(x) = s(y) + s(z) \text{ and } \forall u \in Var \setminus \{x\} : s'(u) = s(u)\}$

Abstract semantics $acc^{\#} : PP \to D^{\#}$ where $D^{\#} = D_{\mathbb{Z}}^{Var}$:

- For all $u \in Var \setminus \{x\}$, $acc^{\#}(p')(u) = acc^{\#}(p)(u)$ and $acc^{\#}(p')(x) = c$.
- For all $u \in Var \setminus \{x\}$, $acc^{\#}(p')(u) = acc^{\#}(p)(u)$ and

$$acc^{\#}(p')(x) = \begin{cases} acc^{\#}(p)(y) + c & \text{if } acc^{\#}(p)(y) \in \mathbb{Z} \\ acc^{\#}(p)(y) & \text{otherwise} \end{cases}$$

- For all $u \in Var \setminus \{x\}$, $acc^{\#}(p')(u) = acc^{\#}(p)(u)$ and

$$acc^{\#}(p')(x) = \begin{cases} acc^{\#}(p)(y) + acc^{\#}(p)(z) & \text{if } acc^{\#}(p)(y) \in \mathbb{Z} \text{ and } acc^{\#}(p)(z) \in \mathbb{Z} \\ \top & \text{if } acc^{\#}(p)(y) = \top \text{ or } acc^{\#}(p)(z) = \top \\ \bot & \text{otherwise} \end{cases}$$

In program `twentyone`, after two iterations of the loop the abstract store maps x and y to $\top$, so the assignment $z := x + y$ will also map z to $\top$.

Exercise 9.4 The abstract domain for disjunctive constant propagation is $D^{\#} = 2^{(\mathbb{Z}^{Var})}$ ordered by inclusion $\subseteq$. Hence, the concrete domain and the abstract domain are exactly the same, and α and γ are the identity.

As already seen in exercise 9.1, computing the least solution of the system of equations, takes an infinite number of iterations. Therefore, the disjunctive analysis does not work here.

Exercise 9.5 We view a (GOTO-)program P as a labeled directed graph $P = \langle V_P, E_P \rangle$ with a finite set of vertices V_P (the program points) and labeled edges $E_P \subseteq V_P \times Store^{Store} \times V_P$, i. e., edges are labeled by functions which transform the store $Store = Val^{Var}$, where Var is the finite set of program variables.

Given a program $P = \langle V_P, E_P \rangle$, an initial program point $p_0 \in V_P$ and an initial store $s_0 \in Store$, we say that a program point $p \in V_P$ is *reachable* from $\langle p_0, s_0 \rangle$ in P if there exists a finite sequence $\langle p_0, s_0 \rangle, \langle p_1, s_1 \rangle, \ldots, \langle p_n, s_n \rangle$ such that $p = p_n$ and for all $i < n$, $\langle p_i, f_i, p_{i+1} \rangle \in E_P$ and $s_{i+1} = f_i(s_i)$. In other words, p is reachable in P if $acc(p) \neq \emptyset$.

It is well-known that every Turing machine can be simulated by such a program, if *Val* contains the natural numbers and the size of *Var* is not too small (two variables are needed to encode the tape; a couple of auxiliary variables may also be needed). Thus, the halting problem for Turing machines can be reduced to the question whether a particular program point of the simulating GOTO-program is reachable from the initial program point and the initial store. Hence, reachability of program points in GOTO-programs must be undecidable.

Looking at dead code elimination, we have the abstract domain $D^{\#} = \{\bot, \top\}$ and abstraction $\alpha : 2^{Store} \to D^{\#}$ with $\alpha(S) = \top$ if and only if $S \neq \emptyset$. So $\alpha(acc(p)) = \top$ if and only if p is reachable in P, which is undecidable.