Semantik von deterministischen Programmen

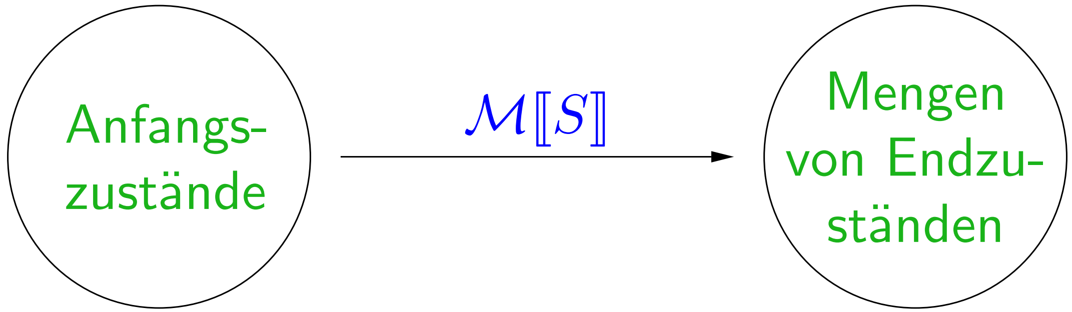
http://www.informatik.uni-bremen.de/theorie/teach/veri

Renate Klempien-Hinrichs

- Konfigurationen, Transitionen, Berechnungen
- Modifikation von Zuständen
- Semantiken von deterministischen Programmen

Semantik von Programmen

Die Semantik eines Programms S ist eine Abbildung $\mathcal{M}[\![S]\!]$, die jedem Anfangszustand eine Menge aus durch Abarbeitung von S erreichten Endzuständen zuordnet:



Definiere $\mathcal{M}[\![S]\!]$ als strukturierte operationelle Semantik. Dabei beschreibt eine Transition den Übergang von einer Konfiguration $\langle S\,,\ \sigma\rangle$ zur nächsten:

Zustand   Folgezustand

$$\langle \underbrace{S}_{\text{Programm}}\,,\ \overbrace{\sigma}^{\text{Zustand}}\rangle \to \langle \underbrace{S'}_{\text{Restprogramm nach einem Schritt}}\,,\ \overbrace{\sigma'}^{\text{Folgezustand}}\rangle$$

Programm   Restprogramm
nach einem Schritt

Transitionsregeln

Konvention: E steht für das leere Programm.
Für alle Programme S gilt: $E; S \equiv S; E \equiv S$.

(1) $\langle skip\,,\ \sigma\rangle \to \langle E\,,\ \sigma\rangle$

(2) $\langle x := t\,,\ \sigma\rangle \to \langle E\,,\ \sigma[x := \hat{\sigma}(t)]\rangle$

(3) $\dfrac{\langle S_1\,,\ \sigma\rangle \to \langle S_2\,,\ \tau\rangle}{\langle S_1; S\,,\ \sigma\rangle \to \langle S_2; S\,,\ \tau\rangle}$

(4) $\langle \mathbf{if}\ B\ \mathbf{then}\ S_1\ \mathbf{else}\ S_2\ \mathbf{fi}\,,\ \sigma\rangle \to \langle S_1\,,\ \sigma\rangle$ wobei $\sigma \models B$

(5) $\langle \mathbf{if}\ B\ \mathbf{then}\ S_1\ \mathbf{else}\ S_2\ \mathbf{fi}\,,\ \sigma\rangle \to \langle S_2\,,\ \sigma\rangle$ wobei $\sigma \models \neg B$

(6) $\langle \mathbf{while}\ B\ \mathbf{do}\ S\ \mathbf{od}\,,\ \sigma\rangle \to \langle S; \mathbf{while}\ B\ \mathbf{do}\ S\ \mathbf{od}\,,\ \sigma\rangle$ wobei $\sigma \models B$

(7) $\langle \mathbf{while}\ B\ \mathbf{do}\ S\ \mathbf{od}\,,\ \sigma\rangle \to \langle E\,,\ \sigma\rangle$ wobei $\sigma \models \neg B$

Dabei stehen σ, τ für beliebige Zustände, S, S_1, S_2 für beliebige deterministische Programme und B für einen beliebigen Booleschen Ausdruck.

Modifikation von Zuständen

Für einen Zustand σ, eine einfache oder indizierte Variable x vom Typ T und einen Wert $d \in \mathcal{D}_T$ ist der modifizierte Zustand $\sigma[x := d]$ wie folgt definiert, wobei y eine einfache oder Feldvariable ist:

- Falls x eine einfache Variable ist, gilt:

$$\sigma[x := d](y) = \begin{cases} d & \text{falls } x \equiv y, \\ \sigma(y) & \text{sonst.} \end{cases}$$

- Falls x eine indizierte Variable $a[t_1, \ldots, t_n]$ ist und $y \not\equiv a$, gilt:

$$\sigma[x := d](y) = \sigma(y)\,.$$

  Falls $y \equiv a$, gilt für alle Argumentwerte $d_1, \ldots, d_n \in \mathcal{D}$:

$$\sigma[x := d](y)(d_1, \ldots, d_n) = \begin{cases} d & \text{falls } d_i = \hat{\sigma}(t_i) \text{ für alle } i, \\ \sigma(y)(d_1, \ldots, d_n) & \text{sonst.} \end{cases}$$

- Für jeden Fehlerzustand $\phi \in \{\perp, \Delta, \mathbf{fail}\}$ ist $\phi[x := d] = \phi$.

Berechnungen

Sei S ein Programm und σ ein Zustand.

- Eine Transitionsfolge von S (startend in σ) ist eine endliche oder unendliche Folge $\langle S\,,\ \sigma\rangle = \langle S_0\,,\ \sigma_0\rangle \to \langle S_1\,,\ \sigma_1\rangle \to \cdots \to \langle S_i\,,\ \sigma_i\rangle \to \ldots$
- Eine Berechnung von S (startend in σ) ist eine maximale Transitionsfolge von S (startend in σ), d.h. sie ist entweder endlich und kann nicht verlängert werden oder sie ist unendlich.
- Eine Berechnung von S terminiert in τ, falls sie endlich ist und $\langle E\,,\ \tau\rangle$ die letzte Konfiguration ist.
- Eine Berechnung von S divergiert, falls sie unendlich ist. S kann von σ aus divergieren, falls es eine unendliche Berechnung von S gibt, die in σ startet.
- Falls es eine Folge $\langle S\,,\ \sigma\rangle = \langle S_0\,,\ \sigma_0\rangle \to \cdots \to \langle S_n\,,\ \sigma_n\rangle = \langle R\,,\ \tau\rangle$ mit $n \geq 0$ gibt, schreiben wir auch kurz: $\langle S\,,\ \sigma\rangle \xrightarrow{*} \langle R\,,\ \tau\rangle$.

Semantiken von Programmen

Sei S ein Programm.

- Die Semantik der partiellen Korrektheit von S ist die Abbildung

$$\mathcal{M}[\![S]\!] : \Sigma \rightarrow \mathcal{P}(\Sigma)$$

mit

$$\mathcal{M}[\![S]\!](\sigma) = \{\tau \mid \langle S\,,\ \sigma\rangle \xrightarrow{*} \langle E\,,\ \tau\rangle\}\,.$$

- Die Semantik der totalen Korrektheit von S ist die Abbildung

$$\mathcal{M}_{tot}[\![S]\!] : \Sigma \rightarrow \mathcal{P}(\Sigma \cup \{\perp\})$$

mit

$$\mathcal{M}_{tot}[\![S]\!](\sigma) = \begin{cases} \mathcal{M}[\![S]\!](\sigma) \cup \{\perp\} & \text{falls } S \text{ von } \sigma \text{ aus divergieren kann,} \\ \mathcal{M}[\![S]\!](\sigma) & \text{sonst.} \end{cases}$$

Folgen des Determinismus I

Lemma (Determinismus)
Für jedes deterministische Programm S und jeden Zustand σ gibt es genau eine Berechnung von S, die in σ startet.

Korollar (Zahl der Endzustände)
Für jedes deterministische Programm S und jeden Zustand σ enthält $\mathcal{M}[\![S]\!](\sigma)$ höchstens ein Element und $\mathcal{M}_{tot}[\![S]\!](\sigma)$ genau ein Element.

Folgen des Determinismus II

Lemma (keine Blockierung)
Für jedes deterministische Programm $S \not\equiv E$ und jeden Zustand σ gibt es eine Konfiguration $\langle S' \,,\, \sigma' \rangle$ mit $\langle S \,,\, \sigma \rangle \to \langle S' \,,\, \sigma' \rangle$.

Korollar (kein Endzustand)
Für jedes deterministische Programm S und jeden Zustand σ gilt:

(1) Die Berechnung von S startend in σ divergiert oder endet in einer Konfiguration der Form $\langle E \,,\, \tau \rangle$.

(2) $\mathcal{M}[\![S]\!](\sigma) = \emptyset$ gilt genau dann, wenn S von σ aus divergiert.