Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik
Prof. Dr. W.-P. de Roever
Harald Fecher, Immo Grabe

Verifikation nebenläfiger Programme

Sommer 2005 — **Serie 7** — 11. Mai 2005

Thema: MST (Einzelbearbeitung)

Ausgabetermin: 11. Mai 2005

Abgabe: 23. Mai (12:00 im Schrein)

Aufgabe 1 (4 Punkte) Zeigen Sie durch Gegenbeispiele, dass das Merging Lemma (Lemma 3.63) nicht gilt, wenn folgende computation histories benutzt werden:

- Die history speichert nur die initialen Zustände der Variablen für jeden Schritt. Eine history ist also eine Sequenz von Zuständen

  $$\langle \bar{\eta}_0, \bar{\eta}_1, \ldots, \bar{\eta}_m \rangle \ ,$$

  und jeder Transitions-Label $c \to [\bar{y} := f(\bar{y})]$ wird transformiert in $c \to [\bar{y}, h := f(\bar{y}), h \cdot \bar{y}]$.

- Eine history ist eine Sequenz von Prozess Identifikatoren

  $$\langle i_0, i_1, \ldots, i_m \rangle \ ,$$

  welche die Nummer des Prozesses speichert, welcher den Schritt ausführt. Jeder Label $c \to [\bar{y} := f(\bar{y})]$ in einem Prozess P_i wird entsprechend transformiert in $c \to [\bar{y}, h := f(\bar{y}), h \cdot i]$.

Aufgabe 2 (6 Punkte) Man kann auch eine direkte Methode angeben, um von einem Programm P zu zeigen, dass es φ-runtime-error-frei ist. Dazu muss Punkt (iii) in Definition 3.17 auf S. 149 abgeändert werden in

(iii) *Prove* local correctness *and* runtime-error freedom *of every* P_i'*: For every transition* $l \xrightarrow{a} l'$ *of* P_i'*, with* $a \equiv b \to f$*, prove*

$$\models Q_l \wedge b \to (Q_{l'} \circ f) \wedge \mathit{Def}(f) \ ,$$

und die zweite Klausel von Punkt (v) gelöscht werden; wobei zudem gefordert wird, dass g in Punkt (i) und h in Punkt (v) totale Funktionen sind.

Zeigen Sie, dass die resultierende Methode korrekt und semantisch vollständig ist.

Aufgabe 3 (6 Punkte) Beweisen Sie Lemma 3.45 auf Seite 184.

Hinweis: Folgende Definitionen und Lemmata können beim Beweis nützlich sein. Lemma 2 muss nicht extra bewiesen werden. Wird Lemma 3 oder Lemma 4 verwandt, so muss nur Lemma 3 bewiesen werden.

Definition 1 Eine reaktive Sequenz ρ sei (anders als im Vorlesungsskript) eine Sequenz

$$(\sigma_0, i_1, \sigma_1), (\sigma_1, i_2, \sigma_2), \ldots$$

Sei ι definiert wie folgt:

$$\begin{aligned} \iota(\sigma, \theta \cdot (i, \sigma')) &\stackrel{\text{def}}{=} \iota(\sigma', \theta) \cdot (\sigma', i, \sigma) \\ \iota(\sigma', \varepsilon) &\stackrel{\text{def}}{=} \varepsilon \end{aligned}$$

Definition 2 Der Projektionsoperator für reaktive Sequenzen sei für eine Menge I von Prozessindizes definiert wie folgt:

$$\begin{aligned} \varepsilon \downarrow I &\stackrel{\text{def}}{=} \varepsilon \\ (\rho \cdot (\sigma, i, \sigma')) \downarrow I &\stackrel{\text{def}}{=} \begin{cases} \rho \downarrow I \cdot (\sigma, i, \sigma') & \text{falls } i \in I \\ \rho \downarrow I & \text{falls } i \notin I \end{cases} \end{aligned}$$

Lemma 1 Für alle Computation histories θ, Mengen I sowie Zustände σ gilt:

$$\theta[I](\sigma) = \iota(\sigma, \theta) \downarrow I$$

Wir gehen hier von einer entsprechend „um den Prozessindex i“ erweiterten Definition von $\theta[I](\sigma)$ aus.

Lemma 2 Für alle reaktiven Sequenzen ρ, ρ' und Mengen I gilt:

$$(\rho \cdot \rho') \downarrow I = \rho \downarrow I \cdot \rho' \downarrow I$$

Lemma 3 Für alle reaktiven Sequenzen ρ und Mengen I, J mit $I \cap J = \emptyset$ gilt:

$$\rho \downarrow (I \cup J) \in (\rho \downarrow I) \mathbin{\tilde{||}} (\rho \downarrow J)$$

Lemma 4 Für alle reaktiven Sequenzen ρ und Mengen $I_1, \ldots I_n$ mit $I_i \cap I_j = \emptyset$ für alle $i \neq j$ gilt:

$$\rho \downarrow (I_1 \cup \ldots \cup I_n) \in (\rho \downarrow I_1) \mathbin{\tilde{||}} \ldots \mathbin{\tilde{||}} (\rho \downarrow I_n)$$