

## Max Wisniewski , Alexander Steen

Tutor: Ansgar Schneider

**Aufgabe 1**

Definieren Sie die WSKEA-Maschine derart um, dass bei arithmetischen Ausdrücken rechte Unterausdrücke vor linken ausgewertet werden. Konstruieren Sie ein Beispiel, für das ein abweichendes Ergebnis erzielt wird.

**Lösung:**

Dafür müssen lediglich zwei Zustandübergänge geändert werden, der Rest kann gleich bleiben:

(i) Zuerst müssen die rechten Ausdrücke einer Operation zuerst auf den Keller gelegt werden, damit diese zuerst ausgewertet werden:

$$\Delta < W|S|T_1\overline{OPT}_2.K|E|A > := < W|S|T_2.T_1.\overline{OP}.K|E|A >$$

(ii) Dann müssen wir beim Herunternehmen der Ergebnisse die korrekte der Operationsanwendung wieder herstellen:

$$\Delta < n_1.n_2.W|S| + .K|E|A > := < n_1 + n_2.W|S|K|E|A >, \text{ falls } n_1 + n_2 \text{ darstellbar.}$$

Diese Regel kann analog auf alle anderen arithmetische Operationen angewendet werden.

**Abweichendes Beispiel:****Links-vor-Rechts:**

$$\begin{aligned} \Delta < ()|()|\underline{read} - \underline{read}|(3, 2)|A > &:= \Delta < ()|()|\underline{read}, \underline{read}, -|(3, 2)|A > \\ &:= \Delta < (3)|()|\underline{read}, -|(2)|A > \\ &:= \Delta < (2, 3)|()| - |()|A > \\ &:= \Delta < (3 - 2)|()| - |()|A > \end{aligned}$$

**Rechts-vor-Links:**

$$\begin{aligned} \Delta < ()|()|\underline{read} - \underline{read}|(3, 2)|A > &:= \Delta < ()|()|\underline{read}, \underline{read}, -|(3, 2)|A > \\ &:= \Delta < (3)|()|\underline{read}, -|(2)|A > \\ &:= \Delta < (2, 3)|()| - |()|A > \\ &:= \Delta < (2 - 3)|()| - |()|A > \end{aligned}$$

Im ersten Fall wird im linken Ausdruck die 3 zuerst gelesen und im rechten die 2. Wir rechnen vollgültig  $3 - 2$  und erhalten eine 1. Im zweitenn Fall gehen wir erst in den rechten Ausdruck, lesen dort die 3 und gehen danach in den linken zweig und lesen dort als nächstes die 2. Wir rechnen also  $2 - 3$  und erhalten eine  $-1$ .

## Aufgabe 2

Erweitern Sie die WSKEA-Maschine um eine Komponente  $N$  für Nachrichten (Texte), in der kurze sinnvolle Meldungen eingetragen werden, wenn es keinen Folgezustand gibt oder wenn die Ausführung korrekt terminiert.

### Lösung:

Wir erweitern Zunächst unseren Zustand  $z$  um eine Nachricht. Da wir nur Nachrichten zurückgeben, wenn wir abstürzen oder fertig sind, brauchen wir keinen Stack und keine Queue. Es muss einfach ein Text gespeichert werden.  $\langle W|S|K|E|A|N \rangle$ , dabei sind  $W, S, L, E, A$  wie in der ursprünglichen WSKEA Maschine und  $N$  ist das eben erwähnt Feld für die einzelne Nachricht. Der Startzustand  $z_{P,E} = \langle \varepsilon|S_0|P.\varepsilon|E|\varepsilon \rangle$  ist äquivalent zum ursprünglichen Startzustand. Der letzte Eintrag für die Nachricht  $N$  ist einfach leer. Da wir nun auch im Fehlerfall eine Überführung haben wollen und nicht einfach abbrechen, definieren wir eine Funktion, die die unendliche Ausführung ermöglicht und bei Beendigung Idempotent sein sollte:

$$\Delta \langle W|S|K|E|A|n \rangle = \langle W|S|K|E|A|n \rangle, \quad \text{für } n \neq \varepsilon$$

Dies bedeutet, sollten wir eine Nachricht empfangen haben, lassen wir den Zustand unverändert, da es sich um einen Fehler handelt oder das Programm schon fertig ist. Damit wir nicht so viel schreiben müssen übernehmen wir alle anderen Überföhrungsfunktionen, wie gehabt, wenn die nachricht  $\varepsilon$  ist. Nun müssen wir nur noch Fälle betrachten in denen das Programm beendet wird oder ein Fehler auftritt.

$$\Delta \langle W|S|\varepsilon|E|A|\varepsilon \rangle = \langle W|S|\varepsilon|E|A| \text{ Programm erfolgreich beendet} \rangle$$

Einen Fehler können vielfältig auftreten, wie wir auf dem ersten Übungsblatt gesehen haben, wir werden es hier exemplarisch für  $IF$  und  $+$  zeigen

$$\Delta \langle n1.n2.W|S|+.K|E|A|\varepsilon \rangle = \langle W|S|K|E|A|\text{falsche Typ bei } + \rangle, \quad n1, n2 \text{ keine Zahl}$$

$$\Delta \langle n1.\varepsilon|S|+.K|E|A|\varepsilon \rangle = \langle W|S|K|E|A|\text{Zu wenig Zahlen für } + \rangle$$

$$\Delta \langle b.W|S|if.p1.p2.K|E|A|\varepsilon \rangle = \langle W|S|K|E|A|\text{Bedingung ist keine Zahl} \rangle, \quad b \text{ kein Bool}$$

## Aufgabe 3

Die Syntax von WHILE sei um das repeat-until-Konstrukt erweitert, wie in der Aufgabe. Ergänzen Sie die operationelle Semantik.

### Lösung:

Die übliche Semantik von REPEAT C UNTIL B ist, dass man C solange ausführt, wie B gilt. Der unterschied zu While ist, dass man C zunächst einmal ausführt, bevor man die Bedingung prüft. Um es un leicht zu machen, formen wir REPEAT deshalb auf WHILE um. Die Berechnung wird dadurch nicht verändert.

$$\Delta \langle W|S|\underline{REPEAT} C \underline{UNTIL} B.K|E|A \rangle = \langle W|S|C.\underline{WHILE} \underline{NOT} (B) \underline{DO} C \underline{OD}.K|E|A \rangle$$

Nach dieser Umformung wird zunächst einmal C ausgeführt. Nach dieser Ausführung wird B geprüft. Ist es wahr, wird die Schleife abgebrochen. Wenn nicht geht es in einen neuen Durchlauf über.

## Aufgabe 4

tba