

Abstrakte Maschinen

Sommersemester 2006

5. Übungsblatt

Abgabetermin: Fr., 16. Juni 2006

Aufgabe 1: Code-Erzeugung

4 Punkte

Gegeben sei ein Ausdruck $e \equiv \text{fn } x \Rightarrow \text{if } x=1 \text{ then } 1 \text{ else } x*\text{fac}(x-1)$, sowie die Adressumgebung $\rho = \{\text{fac} \mapsto (L, 1)\}$ und der Kellerpegel $kp = 5$. Berechnen Sie $\text{code}_V e \rho kp!$ Geben Sie zu jeder Instruktion den aktuellen Kellerpegel an!

Aufgabe 2: Adressumgebung

10 Punkte

Geben sei folgende Funktionsdefinition:

```
fn x,y,z => let x = 1;
            a = 3;
            b = 4
            in (a+(b+(x+(y+z))))
```

- Erzeugen Sie Call-by-value-Code für die Funktionsdefinition! Geben Sie zu jeder Instruktion den aktuellen Kellerpegel an. Der Ausgangskellerpegel sei $kp = 3$.
- Bestimmen Sie die Adressumgebung ρ , die zur Code-Generierung von $(a + (b + (x + (y + z))))$ verwendet wird!
- Bestimmen Sie die Adressen der Variablen a , b , x , y , und z relativ zu SP_0 (dem Stackpointer beim Betreten des Funktionsrumpfes).

Aufgabe 3: Reverse Engineering

6 Punkte

Betrachten Sie folgenden Call-by-value-Code eines PuF Ausdrucks:

```
alloc 1          getbasic          jumpz _5          _3:return 2
pushloc 0        pushloc 5          mark _7          _1:rewrite 1
mkvec 1         getbasic          pushloc 4        mark _8
mkfunval _0     sub                getbasic         loadc 16
jump _1        mkbasic           pushloc 4        mkbasic
_0:targ 2      pushloc 5          getbasic         loadc 12
pushloc 0      pushglob 0        sub             mkbasic
getbasic       apply           mkbasic         pushloc 5
pushloc 2     _4:jump _3        pushloc 4        apply
getbasic      _2:pushloc 0     pushglob 0      _8:slide 1
gr            getbasic         apply           halt
jumpz _2      pushloc 2        _7:jump _6
mark _4       getbasic         _5:pushloc 0
pushloc 3     le                _6:
```

- Bestimmen sie den Kellerpegel kp für jeden Programmpunkt beginnend bei $kp = 5!$
- Was berechnet dieses Programm? Versuchen sie hierzu den Code in Sinneinheiten zu zerlegen.