Technische Universität München
Fakultät für Informatik
Prof. Dr. H. Seidl

Peter Ziewer

Abstrakte Maschinen

Sommersemester 2006

5. Übungsblatt *Abgabetermin: Fr., 16. Juni 2006*

Aufgabe 1: *Code-Erzeugung* — *4 Punkte*

Gegeben sei ein Ausdruck $e \equiv$ `fn x => if x=1 then 1 else x*fac(x-1)`, sowie die Adressumgebung $\rho = \{fac \mapsto (L, 1)\}$ und der Kellerpegel $kp = 5$. Berechnen Sie $code_V\ e\ \rho\ kp$! 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))))
```

a) Erzeugen Sie Call-by-value-Code für die Funktionsdefinition! Geben Sie zu jeder Instruktion den aktuellen Kellerpegel an. Der Ausgangskellerpegel sei $kp = 3$.

b) Bestimmen Sie die Adressumgebung ρ, die zur Code-Generierung von $(a + (b + (x + (y + z))))$ verwendet wird!

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

a) Bestimmen sie den Kellerpegel kp für jeden Programmpunkt beginnend bei $kp = 5$!

b) Was berechnet dieses Programm? Versuchen sie hierzu den Code in Sinneinheiten zu zerlegen.