

11. Exercise sheet *Compiler Construction 2011*

Due to Wed., 26th Januar 2011, *before* the exercise course begins.

Hand in your solutions in groups of three or four!

Exercise 11.1: (2 points)

Consider the following intermediate code:

```

    :
7: LOAD(1, 2);      (dif, off)
8: ADD;
9: RET;
    :
26: CALL(38, 1, 3);  (ca, dif, loc)

```

Give the next four states of the abstract machine starting in:

$$(ca, d, p) := (7, -3, 9 : 4 : 26 : 3 : 7 : 4 : 3 : 36 : 5 : 10 : 4 : 40 : 1 : 2 : \dots)$$

Recall that the procedure stack has the form:

sl	dl	ra	v_1	\dots	v_n	\parallel	\dots
------	------	------	-------	---------	-------	-------------	---------

and the *base*-function is defined as:

$$\begin{aligned}
 \text{base}(p, 0) &:= 1 \\
 \text{base}(p, \text{dif} + 1) &:= \text{base}(p, \text{dif}) + p.\text{base}(p, \text{dif})
 \end{aligned}$$

Exercise 11.2: (4 points)

In addition to `while`-loops we want to have `for`-loops with implicit declaration of the counter variable in our example programming language:

`for (var I := A ; B ; C1) C2`

- a) Extend the translation function *ct* accordingly.
- b) Generate intermediate code for

`for (var x := 0; x < 10; x := x + 1) P();`

without parameters for the `CALL` instruction generated for `P()`.

Exercise 11.3:**(5 points)**

Let $P = DC$ be the following EPL program with data structures:

```
type Real := real;
  Point := record x: Real; y: Real end;
  PArray = array [0..5] of Point;
var p: PArray; z: int;
z := 0;
while z <= 5 do
  p[z].x := z;
  p[z].y := z * z;
```

- a) Determine the symbol table $st := update(D, st_\emptyset)$.
- b) Give the translation $ct(C, st)$.