

Virtual Machines

Exercise Sheet 2

Deadline: 24. May 2011, 14:00

Exercise 1: *break* & *continue*

5 Points

In the lecture we did not consider **break** and **continue** as general statements. In C however these statements can be freely used (for example inside loops). Give code generation definitions for **break**, **continue**. Redefine **for** and **while** loops to support this feature.

Exercise 2: *Switch Statement*

10 Points

In the lecture we saw a translation scheme for simple switch statements, but the full scheme is not that much more complicated.

```
switch(e){  
  case b11: ... case bn1:  
    [default:]  
    ss1  
  ...  
  
  case b1k: ... case bmk:  
    [default:]  
    ssk  
}
```

- No two case values b_j^i are the same.
- Only one **default:** per switch.
- Statements now may include **break**'s. (like in the 1st Ex.)
- We know the minimal ($\min(b)$) and maximal ($\max(b)$) cases.
- We have a mapping σ from case values to statement indexes:

$$\sigma(b_j^i) = i$$
$$\sigma(\max(b) + 1) = n \quad \text{where } n \text{ is default}$$

Define the full translation scheme for switches! Note, that if you translate a simple switch statement using the full scheme, you should get very similar code to the simple scheme. What happens to σ if you do?

Exercise 3: *The do ... while Loop*

2 Points

Define code generation for the “do *s* while (*e*)” loop.

Exercise 4: *Expressions*

3 Points

Produce CMA code for the following expression! Use the following address environment:
 $\rho = \{x \mapsto 4, y \mapsto 5, z \mapsto 6\}$!

- $e_1 \equiv x = (y = (y + 4)) * z$