Technische Universität München *Summer Semester 08*
Fakultät für Informatik *K N Verma*
Prof. Dr. H. Seidl *verma@in.tum.de*

Virtual Machines

Exercise Sheet 8

Deadline: 17 June 2008, during lecture, by email, or in room 02.07.041

Exercise 1:

10 Points

The following is the straightforward method of computing Fibonacci numbers using a recursive function.

```
letrec fib = fn x => if x <= 1 then 1 else (fib (x-1)) + (fib (x-2))
in fib 4
```

The following example shows how to accomplish the same using a tail-recursive function.

```
letrec fib_aux = fn n, next, result => if n <= 1 then result
                                else (fib_aux (n-1) next + result, next);
       fib_main = fn n => (fib_aux n 2 1)
in fib_main 4
```

Generate $code_V$ for this new expression with $sd = 0$ and $\rho = \{\}$ using CBV.

Exercise 2:

6 Points

Write a prolog program including following predicates:

a) *last*/2 where the first parameter is a list and the second one is the last element of this list (e.g. last([1,2,3],3)).
b) *reverse*/2 with two lists as parameters, where one is the reverse list of the other. (e.g. reverse([1,2,3],[3,2,1]))
c) *chain*/2 with two lists, where the first list includes the second one as connected chain.
(e.g. chain([1,2,3,4,5,6],[2,3,4]))

Note: You can write auxiliary predicates if needed.

Exercise 3:

4 Points

Produce $code_A/code_G$ for the following terms/goals !

a) $f(X, g(b, Y), g(\bar{X}, \bar{Z}))$
b) $p(f(g(X, h(\bar{Y}, \_), b), Z))$

Use the following address environment: $\rho = \{X \mapsto 1, Y \mapsto 2, Z \mapsto 3\}$!