

# Abstract Machines

*Summer Semester 2004*

6. Homework

Deadline: 9 June 2004 12:00

Exercise 1:

10 Points

Extend PuF with type `Tree`. `Trees` are constructed using the nullary constructor (constant) `LEAF` and the 3-ary constructor `NODE`. `NODE` constructs a `Tree` value from an arbitrary value and two `Tree` values. The syntax of expressions  $e$  is extended with:

$$e ::= \dots \mid LEAF \mid NODE(e_1, e_2, e_3) \\ \mid (\mathbf{case} \ e_0 \ \mathbf{of} \ LEAF \rightarrow e_1; \ NODE(info, left, right) \rightarrow e_2)$$

Define code generation functions for the new expressions. Extend the set of heap objects with new objects of type `Tree`. You may define new MaMa instructions.

Exercise 2:

4 Points

A restaurant offers different meals for the same price. A meal consists of one of the following: beef, pork, fish or vegetarian food. It also includes a salad or a soup or a dessert. Write a prolog program that computes all available meals.

Exercise 3:

6 Points

Write a prolog program including following predicates:

- $odd/2$  ( $even/2$ ) where the first parameter is a list and the second one a list containing only the odd (even) elements of the first parameter.  
(e.g. `odd([1,2,3,4,5],[1,3,5])`)
- $reverse/2$  with two lists as parameters, where one is the reverse list of the other.  
(e.g. `reverse([1,2,3],[3,2,1])`)
- $chain/2$  with two lists, where the first list includes the second one as connected chain.  
(e.g. `chain([1,2,3,4,5],[2,3,4])`)
- $remove/3$  which removes all occurrences of the first parameter in the second parameter, which is a list.  
(e.g. `remove(2,[1,2,3,2,5],[1,3,5])`)

Note: You can write auxiliary predicates if needed.