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

Virtual Machines

Exercise Sheet 10

Deadline: 1 July 2008, during lecture, by email, or in room 02.07.041

Exercise 1: **Most General Unifiers:** *9 Points*

When two terms are to be unified, they are compared. If they are both constants then the result of unification is success if they are equal else failure. If they are both variables then one is bound to the other and unification succeds, If one is a variable X and the other some structure t then if X occurs in t then unification fails, otherwise X is bound to t and unification succeeds. If both terms are structures then each pair of sub-terms is unified recursively and the unification succeeds if each pair of sub-terms unifies.

The result of unification is either failure or success with a set of variable bindings, known as a **"unifier"**. There may be many such unifiers for any pair of terms but there will be at most one **"most general unifier"**, other unifiers simply add extra bindings for sub-terms which are variables in the original terms.

Determine the most general unifiers for the following pairs of terms if possible or explain why the unification fails:

a) $f(X, g(Y, b))$ and $f(g(a, Z), X)$
b) $f(g(a, Z), X)$ and $f(X, g(b, Y))$
c) $g(X, f(X, X))$ and $g(f(a, a), f(Y, Y))$
d) $g(g(X, g(a, Z)), g(f(V), V))$ and $g(g(f(Y), Y), g(Z, g(Y, Z)))$
e) $a(b, X, d(e, Y, g(h, i, Z)))$ and $a(U, c, d(V, f, g(W, i, j)))$
f) $f(X, 5, Y, x(a, g(6, 7)))$ and $f(Y, 5, c, x(Z, g(6, X)))$

Example: The most general unifier of $f(X, g(Y, b), Z)$ and $f(g(a, U), g(a, V), W)$ is $[X/g(a, U), Y/a, Z/W, V/b]$. The terms $f(X, Y)$ and $f(g(Y), g(X))$ have no unifiers.

Exercise 2: *6 Points*

In the lecture we defined a run time function $deref$, which dereferences the reference chains.

(a) In which cases can nontrivial reference chains, i.e. such that $deref$ is recursively called at least once, appear? Give an example!

(b) Explain how long such reference chains can become?

Exercise 3: *7 Points*

The following PuP programm P is given:

```
rev (L1, R, L2) :- L1=[], L2=R.
rev (L1, R, A) :- L1 = [X|L], rev (L, [X|R], A).
reverse (L, R) :- rev (L, [], R).
?- reverse (X, [4,2,1]).
```

(a) Translate P to WiM code (without optimization).

(b) Execute the WiM code showing the sequence of (sub-)goals that are called and the stack and the heap after each of these goals has been processed. Where is backtracking done ?