Technische Universität München *Dr. K. N. Verma*
Fakultät für Informatik *verma@in.tum.de*
Prof. Dr. H. Seidl *Room: MI 02.07.041*

Abstract Machines

Summer Semester 2004

10. Homework *Deadline: 7 July 2004 12:00*

Exercise 1: *7 Points*

Our normalized representation of PuP (in clause heads only variables and no arbitrary terms are allowed) simplifies the translation pattern for clauses. The references to the argument terms are placed directly above the organizational cells (like local variables) if a clause is called. This implicitly unifies the head variables with the argument terms. Without normalization the space needed for the argument terms has to be reservered (above the org. cells) before the space for the local variables is allocated. Additionally the unification of the terms in the head of a clause with the argument terms must be ensured.

(a) Formalize above description to a code function for the translation of clauses which can contain arbitrary head terms.

(b) There are situations, in which the references to the argument terms, introduced above, are no longer needed after the unification with the head terms is done. Which?

Exercise 2: *6 Points*

Translate the following PuP program P considering the optimization of **tail recursion** as given in chapter 34 !

```
t(X) <- X=b
p <- q(X),t(X)
q(X) <- s(X)
s(X) <- t(X)
s(X) <- X=a
? p
```

(Hint: A translation of P without tail recursion is given in chapter 33 page 303.)

Exercise 3: *7 Points*

The following PuP clause is given:

```
p(X1) <- q1(X3,X2), q2(X1,X2), q3(a(X2)), q4(X3,X4), X4=a.
```

(a) Calculate the **life times** of the variables and order the variables according to their life **life times**!

(b) Translate the clause with regard to the calculated order and pop dead variables by means of the `trim` instruction!

(Hint: see chapter 35 for an example)