

Integrated Logic Systems (Part I)

Prof. Michael Schroeder, Sebastian Voigt

International Master Program in Computational Logic — summer term 2010

18.05.2010

Exercise 3.1

Consider the following program

```
member(H, [H|T]).
member(E, [H|T]) :- member(E, T).
```

together with the query $? - L = [a, b], member(X, L)$.

- Extend the WAM such that predicate $= / 2$ is predefined with the expected behaviour.
- Give the \mathcal{L}_3 Code for the program and the query.
- Trace the execution of the code from b) to determine two CAS.

Exercise 3.2

Anonymous variables (denoted by names with leading $_$) by definition only occur once and thus need not record any bindings. In arguments of clause heads they can simply be ignored. In non-argument positions the following instructions can be used (see also pages 43-44 in the WAM tutorial):

```
set_void n ≡ for i ← H to H + n - 1 do
    HEAP[i] ← ⟨REF, i⟩
H ← H + n
```

```
unify_void n ≡ case mode of
    read: S ← S + n
    write: for i ← H to H + n - 1 do
        HEAP[i] ← ⟨REF, i⟩
        H ← H + n
    endcase
```

Use the mentioned optimizations to give \mathcal{L}_3 code for query $? - p([b|_], a)$. and program

```
p(g(-, -), -).
p([_|_], X) :- p(X, -).
p(a, b).
```

and trace its execution.

Exercise 3.3

Consider an extension of \mathcal{L}_3 to \mathcal{L}_4 such that also negated atoms are allowed in rule bodies. The *negation as failure* principle in Prolog fails on a negated atom $\neg p(\vec{t})$ if $p(\vec{t})$ is satisfiable wrt. the given program, and succeeds otherwise.

a) Give a transformation that takes an arbitrary \mathcal{L}_4 program P and yields an \mathcal{L}_3 program P' such that WAM execution of P' appropriately implements negation as failure.

b) Apply the program transformation from a) to query $? - \neg q(X)$. and the following program:

$$\begin{aligned} q(X) & :- \neg p(X). \\ p(a). \end{aligned}$$

c) Is there any difference in the outcome of the execution from b) when changing the query to $? - p(X)$.? (You need not actually execute the WAM.)