Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik (20)
Johann Wolfgang Goethe-Universität Frankfurt am Main

Deduktionssysteme: Grundlagen und Anwendungen

Sommersemester 1999

Übungsblatt Nr. 6

Abgabe: Donnerstag 20. Mai 1999 vor der Vorlesung

Aufgabe 1 (35 Punkte)

Schreiben Sie ein Programm, das für einen Datentyp `PL1` mit dem prädikatenlogische Formeln dargestellt werden die frei vorkommenden Variablen ermittelt. Verwenden Sie als Programmiersprache Haskell, Mercury oder Prolog.

Der Haskell-Datentyp könnte z.B. so aussehen:

```
data PL1
  = Atom String [Term]
  | Not PL1
  | PL1 `V` PL1
  | PL1 :^: PL1
  | PL1 `Implies` PL1
  | PL1 `Iff` PL1
  | Forall String PL1
  | Exists String PL1

data Term
  = Var String
  | Term String [Term]
```

und wäre äquivalent zum Mercury-Typ `pl1`:

```
:- import_module list.

:-  type pl1
    ---> atom(string,list(term))
    ;    not(pl1)
    ;    or(pl1, pl1)
    ;    and(pl1, pl1)
    ;    implies(pl1, pl1)
    ;    iff(pl1, pl1)
    ;    forall(string, pl1)
    ;    exists(string, pl1).
```

```
:-  type term
    ---> var(string)
    ;    term(string, list(term)).
```

Aufgabe 2 (20 Punkte)

Bestimmen Sie eine Signatur zur Arithmetik auf pos. ganzen Zahlen mit Gleichheit und Ordnung. Verwenden Sie Symbole für die Signatur, die für die Arithmetik nicht verwendet werden. Formulieren Sie einige Sätze der Arithmetik auf pos. ganzen Zahlen und bestimmen Sie andere Interpretationen, für die diese Sätze ebenfalls gelten.

Aufgabe 3 (20 Punkte)

Formen Sie folgende Formeln zu Klauselnormalform um.

1. $\forall x : A(x) \implies \exists y : B(y) \wedge C(x,y)$
2. $\forall x : A(x) \implies \exists y,z : A(y) \wedge A(z) \wedge B(y) \wedge C(z) \wedge D(y,x) \wedge E(z,x)$