

## Vorlesung Perlen der Informatik

### Aufgabe 1 Herleitungsbaum

1. Überlegen Sie sich anhand der Prolog-Semantik, wie der Herleitungsbaum aussieht, der vom Prolog-Interpreter bei der Auswertung des Ziels:

`mult(X, X, X)`

erzeugt wird.

2. Probieren sie zum Testen aus, wie sich SWI-Prolog verhält. (*Tipp: Mit `trace.` kann man einen Trace einschalten.*)

### Aufgabe 2 Parsen mit Prolog

Gegeben sei folgende (sehr primitive) Grammatik für englische Sätze:

```
<sentence> → <noun-phrase> <verb-phrase>
<noun-phrase> → <determiner> <noun>
<verb-phrase> → <verb> <noun-phrase>
<verb-phrase> → <verb>
<determiner> → the
<noun> → apple
<noun> → man
<verb> → eats
<verb> → sings
```

Entwickeln Sie einen Parser für diese Grammatik in Prolog, also ein Prädikat auf Wortlisten, das wahr ist, wenn die Liste einen gültigen Satz repräsentiert.

1. In der ersten Version soll die Grammatik möglichst 1:1 umgesetzt werden.
2. Wie effizient ist das Verfahren? Wie kann man die Effizienz verbessern?

### Aufgabe 3 Differenzlisten

Häufiger ist es geschickter, anstelle normaler Listen sogenannte *Differenzlisten* zu verwenden. Diese haben als Schwanz eine *uninstantiierte* Variable, die als zusätzliches Argument mitübergeben wird. So wird z.B. die leere Liste durch das Argumentpaar `XS` und `XS`, und die Liste `[a, b, c]` durch das Argumentpaar `[a, b, c|XS]` und `XS` dargestellt.

- Schreiben Sie ein Prädikat `dlist_of_list(XS, YS, ZS)`, welches eine normale Liste `XS` in eine Differenzliste bestehend aus `YS` und `ZS` konvertiert. Schreiben Sie ebenfalls ein Prädikat `list_of_dlist(YS, ZS, XS)` für die Gegenrichtung.
- Schreiben Sie ein Prädikat `dlist_append(XS1, XS2, YS1, YS2, ZS1, ZS2)` zum Zusammenhängen zweier Differenzlisten.
- Schreiben Sie ein Prädikat `dlist_reverse(XS, YS, ZS)`, welches die „normale“ Liste `XS` in die Differenzliste bestehend aus `YS` und `ZS` umdreht.

#### Aufgabe 4 Disjunktive Normalform

Im folgenden seien aussagenlogische Formeln definiert durch

$$\phi ::= x \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2$$

wobei  $x$  eine Variable bezeichnet. Aussagenlogische Formeln der Gestalt

$$(\phi_{1,1} \wedge \dots \wedge \phi_{1,m_1}) \vee \dots \vee (\phi_{n,1} \wedge \dots \wedge \phi_{1,m_n})$$

lassen sich als Liste von Listen von Formeln, darstellen, d.h.

$$[[\phi_{1,1}, \dots, \phi_{1,m_1}], \dots, [\phi_{n,1}, \dots, \phi_{1,m_n}]]$$

Eine Formel der obigen Art ist in *disjunktiver Normalform* (DNF), wenn jedes  $\phi_{i,j}$  eine Variable oder die Negation einer Variable ist.

- Schreiben Sie unter Verwendung der obigen Listendarstellung für Formeln ein Prädikat  $\text{dnf}(F, Xs)$ , das erfüllt ist, wenn  $Xs$  die DNF von  $F$  ist. Beispiel:

$$\text{dnf}(\text{neg}(\text{or}(\text{neg}(\text{or}(\text{neg}(\text{var}(\text{p})), \text{var}(\text{q}))), \text{var}(\text{p}))), \text{var}(\text{p})), Xs).$$

$$Xs = [[\text{var}(\text{p}), \text{neg}(\text{var}(\text{q})), \text{neg}(\text{var}(\text{p}))], [\text{var}(\text{p}), \text{neg}(\text{var}(\text{p}))]]$$

- Schreiben Sie ein Prädikat  $\text{absurd}(Xs)$ , das genau dann erfüllt ist, wenn jede Liste in  $Xs$  eine Formel und ihre Negation enthält.
- Schreiben Sie mit Hilfe der Prädikate  $\text{dnf}$  und  $\text{absurd}$  ein Prädikat  $\text{taut}(F)$ , das genau dann erfüllt ist, wenn  $F$  eine Tautologie ist.