

**Übungen**  
**Deskriptive Programmierung**  
**SS 15**

**Blatt 10**

**Aufgabe 46** (einzureichen über eCampus, [9P]).

Betrachten Sie folgendes Programm:

```
member(X, [X|Xs]) .  
member(X, [Y|Xs]) :- member(X, Xs) .
```

und entwickeln Sie die Prolog-Ableitungsbäume (analog zur Vorlesung) ausgehend von folgenden Anfragen:

- (a) ?- member(X,[2,3]).
- (b) ?- member(4,[2,3]).
- (c) ?- member(X,[2,3]),X=4.
- (d) ?- member(d(X),[c(1),d(2),d(3),c(4),d(5)]).

**Hinweis:** Die Bäume sollten wirklich „gemalt“ sein (von Hand oder per Programm/Satzsystem). Fertigen Sie also nicht „nur“ ein irgendwie geartetes Textprotokoll der Ausführung an.

**Aufgabe 47** (einzureichen über eCampus, [2P]).

Füllen Sie die Lücken  $\square_1$  und  $\square_2$  in **not** ( $\square_1, \square_2$ ) und **not** ( $\square_1$ ); **not** ( $\square_2$ ) konsistent mit Literalen derart, dass sich verschiedene Prolog-Antworten ergeben.

**Aufgabe 48** (einzureichen über eCampus, [4P]).

Betrachten Sie folgendes Programm:

```
member(X, [X|Xs]) .  
member(X, [Y|Xs]) :- member(X, Xs) .  
  
disjoint([X|Xs], Ys) :- not(member(X, Ys)), !, disjoint(Xs, Ys).  
disjoint(Xs, Ys)      :- empty(Xs) .  
  
empty([]) .
```

---

<sup>1</sup>Bei Fragen wenden Sie sich bitte via E-Mail an Janis Voigtländer (jv@informatik.uni-bonn.de).

und „malen“ Sie den Prolog-Ableitungsbaum (analog zur Vorlesung) ausgehend von folgender Anfrage:

$?- \text{disjoint}([1|Xs],[2]).$

**Aufgabe 49** (zu lösen/einzureichen über Autotool, 5 Fehlversuche erlaubt, [2P]).

**Aufgabe 50** (zu lösen/einzureichen über Autotool, 5 Fehlversuche erlaubt, [2P]).

**Aufgabe 51** (zu lösen/einzureichen über Autotool, 5 Fehlversuche erlaubt, [3P]).

**Aufgabe 52** (einzureichen über eCampus, [5 Extrapunkte]).

1. Ersetzen Sie im Prolog-Parser für arithmetische Ausdrücke aus der Vorlesung den Teil zur Erkennung natürlicher Zahlen durch eine Version, die auch Zahlen erkennt, die aus mehreren Ziffern bestehen.
2. Definieren Sie einen Prolog-Parser für Listen natürlicher Zahlen in der üblichen Notation "[2,3,5,7]". Zum Beispiel soll also `list(L,"[2,3,5,7]", "")` erfolgreich sein und die Bindung `L=[2,3,5,7]` liefern.

**Hinweis:** In Regeln einer Definite Clause Grammar können auch normale Prolog-Literale aufgenommen werden, eingeschlossen in geschweifte Klammern: `{...}`.