

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

**Blatt 10**

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

Betrachten Sie folgendes Programm:

```
member(X, [X|Xs]) .  
member(X, [_|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**  $(\square_1); \text{not } (\square_2)$  konsistent mit Literalen derart, dass sich verschiedene Prolog-Antworten ergeben.

**Hinweis:** Das Blatt wird noch um weitere Aufgaben ergänzt.

---

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