



## Logische Programmierung und Deduktive Datenbanken — Klausur —

Name: \_\_\_\_\_

Studiengang: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

(Diese Daten werden zur Ausstellung des Leistungsnachweises benötigt.)

Aufgabe	Punkte	Max. Punkte	Zeit
1 (Verschiedenes)		10	20 min
2 (Datenbankanfragen)		16	30 min
3 (Bindungsmuster, SIP-Strategie)		9	15 min
4 (Polynomauswertung)		15	25 min
Summe		50	90 min

- Ich fühle mich gesundheitlich in der Lage, diese Prüfung abzulegen.  
(Bitte sprechen Sie mit dem Aufsichtspersonal, falls Sie sich krank fühlen.)
- Für den Fall, daß ich nicht korrekt zu dieser Prüfung angemeldet sein sollte, erkläre ich mich sowohl damit einverstanden, daß ich rückwirkend angemeldet werde, als auch damit, daß diese Prüfung nicht zählt (Entscheidung des Dozenten).

Unterschrift: \_\_\_\_\_

**Hinweise:**

- Bearbeitungsdauer: 90 Minuten
- Skript, Bücher, Notizen sind erlaubt.  
Notebooks, PDAs, Handys etc. dürfen nicht verwendet werden.
- Die Klausur hat 10 Seiten. Bitte prüfen Sie die Vollständigkeit.
- Verwenden Sie weder Rot- noch Bleistift zum Bearbeiten der Aufgaben.
- Bitte benutzen Sie den vorgegebenen Platz. Wenn Sie auf die Rückseite ausweichen müssen, markieren Sie klar, dass es eine Fortsetzung gibt.
- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Bei Aufgabe 1 zum Ankreuzen sollten Sie wenigstens raten, wenn Sie die richtige Lösung nicht wissen.
- Fragen Sie, wenn Ihnen eine Aufgabe nicht klar ist!

**Aufgabe 1 (Verschiedenes)****10 Punkte**

Bitte kreuzen Sie jeweils die korrekte Antwort an. Es ist genau eine Antwort pro Teilaufgabe richtig. (Mehr als ein Kreuz pro Teilaufgabe wird mit 0 Punkten bewertet.)

a) Welche Antwort gibt ein PROLOG-System auf die Anfrage  $x=y$  ?

- Yes
- No
- $x=y$

b) Welche Antwort gibt ein PROLOG-System auf die Anfrage  $6 \text{ is } 4+2$  ?

- Yes
- No
- $6=4+2$

c) Welche Antwort gibt ein PROLOG-System auf die Anfrage  $6=4+x$  ?

- Yes
- No
- $x=2$

d) Welche Antwort gibt ein PROLOG-System auf die Anfrage  $\sin(x)=\sin(\pi)$  ?

- Yes
- No
- $x=\pi$

e) Welche Antwort gibt ein PROLOG-System auf die Anfrage  $[X|Y]=[1,2,3]$  ?

- Yes
- No
- $X=1, Y=[2,3]$

f) Sind die Literale  $(X, (f(Z), Z))$  und  $(f(Y), (X, Z))$  „unifizierbar“? Falls ja, geben Sie einen allgemeinsten „Unifikator“ an (mgu), falls nein, geben Sie Variablenbindungen an, bis Sie auf einen Widerspruch stoßen. Die Antwort gilt nur mit schlüssiger Begründung.

- Ja mit den Bindungen: .....
- Nein, wegen: .....

g) Gegeben sei das Programm:

```
sibling(alfons,berta).
sibling(alfons,chris).
sibling(chris,berta).
sibling(X,Y) :- sibling(Y,X).
```

Was ist das Ergebnis der Anfrage `sibling(alfons,X)` ?

- Yes
- No
- Endlosschleife
- `X=berta <;>`  
`X=chris.`
- `X=berta <;>`  
`X=chris <;>`  
`X=berta <;>`  
`X=chris ...`

h) Gegeben sei das folgende Programm:

`p :- p.`

Was ist die Antwort eines PROLOG-Systems auf die Anfrage `p` ?

- Yes
- No
- Endlosschleife

Was würde eine deduktive Datenbank auf die gleiche Anfrage antworten?

- Yes
- No
- Endlosschleife

**Aufgabe 2 (Datenbankanfragen)****16 Punkte**

Gegeben sei die stark vereinfachte Datenbank eines Autohändlers in Form von Fakten:

- Das erste EDB-Prädikat enthält alle Daten der Hersteller. Sowohl die Nummer als auch der Name (Alternativschlüssel) sind eindeutig.

```
/* hersteller(_Nummer_, Name, Land) */  
hersteller(10, "Mercedes", "Deutschland").  
hersteller(20, "VW", "Deutschland").  
hersteller(30, "Toyota", "Japan").  
hersteller(40, "Fiat", "Italien").
```

- Jeder Hersteller stellt mehrere Basismodelle her, deren Bezeichnung global eindeutig ist.

```
/* modell(_Name_, Preis, Herstellernummer) */  
modell("C180", 32000, 10).  
modell("Polo", 15000, 20).  
modell("Passat", 24500, 20).  
modell("Corolla", 15000, 30).
```

- Die Basismodelle können mit Extras versehen werden, die zur Vereinfachung bei jedem Hersteller den gleichen Aufpreis haben sollen. Sowohl die Nummer als auch die Bezeichnung ist eindeutig.

```
/* extra(_Nummer_, Aufpreis, Bezeichnung) */  
extra(123, 300, "beheizbare Heckscheibe").  
extra(125, 150, "Nebelscheinwerfer").  
extra(145, 200, "Zentralverriegelung").  
extra(220, 1000, "Klimaautomatik").  
extra(345, 600, "ABS").  
extra(633, 3000, "Navigationsgeraet").
```

- Der Hersteller legt fest, mit welchen Extras sein Modell wahlweise ausgerüstet werden kann. Ein Auto kann mehrere Extras mit jeweils unterschiedlicher Bezeichnung und unterschiedlicher Nummer bekommen.

```
/* verfuegbar(Modell, Extranummer) */  
verfuegbar("C180", 123).  
verfuegbar("C180", 125).  
verfuegbar("C180", 145).  
verfuegbar("C180", 220).  
verfuegbar("C180", 345).  
verfuegbar("C180", 633).  
verfuegbar("Polo", 123).  
verfuegbar("Polo", 125).  
verfuegbar("Polo", 220).  
verfuegbar("Polo", 345).  
verfuegbar("Passat", 123).  
verfuegbar("Passat", 145).  
verfuegbar("Passat", 220).  
verfuegbar("Passat", 345).  
verfuegbar("Passat", 633).  
verfuegbar("Corolla", 123).  
verfuegbar("Corolla", 145).  
verfuegbar("Corolla", 220).  
verfuegbar("Corolla", 633).
```

- a) Für welche Modelle sind Navigationsgeräte verfügbar? Geben Sie ein Prädikat an, welches die Modellbezeichnung als Argument hat. Vorsicht: Es geht wirklich um Navigationsgeräte, nicht um Nummern, die wir relativ willkürlich für die Extras vergeben haben. Andererseits können wir auch für unnötige Joins Punkte abziehen.

Anfrage-Ergebnis im Beispiel:

```
?- modell_mit_navi(Modell).
```

```
Modell = "C180";  
Modell = "Passat";  
Modell = "Corola".
```

- b) Bestimmen Sie den Namen des Herstellers und die Modellbezeichnung aller deutschen Fahrzeuge, die mit Klimaautomatik geliefert werden können. Schreiben Sie ein Prädikat mit den Argumenten `Hersteller` für den Herstellernamen und `Modell` für die Modellbezeichnung.

Anfrage-Ergebnis im Beispiel:

```
?- deutsch_mit_klima(Hersteller, Modell).
```

```
Hersteller = "Mercedes", Modell = "C180";  
Hersteller = "VW",      Modell = "Polo";  
Hersteller = "VW",      Modell = "Passat".
```

- c) Die Bezeichnung eines Automodells soll ein Schlüssel sein. Definieren Sie ein Prädikat, welches wahr ist, wenn die Schlüsselbedingung verletzt ist.

- d) Ermitteln Sie alle in Japan und Italien hergestellte Modelle. Das zu entwerfende Prädikat soll die Argumente `Land`, `Hersteller` für den Herstellernamen und `Modell` für die Modellbezeichnung haben.

Anfrage-Ergebnis im Beispiel:

```
?- japan_oder_italien(Land, Hersteller, Modell).
```

```
Land = "Japan", Hersteller = "Toyota", Modell = "Corolla".
```

**Aufgabe 3 (Bindungsmuster, SIP-Strategie)****9 Punkte**

Gegeben seien die eingebauten Prädikate `cons` zum Erzeugen und Zerlegen von Listen mit `cons(Head,Tail,List)` und erlaubten Bindungsmustern `bbf` und `ffb`, sowie `succ(Predecessor,Successor)` mit Bindungsmustern `bf` und `fb`. Bestimmen Sie für die Regel

```
increment_head(XList,YList) :-  
    cons(XHead,Tail,XList),  
    cons(YHead,Tail,YList),  
    succ(XHead,YHead).
```

eine sinnvolle Reihenfolge der Rumpfliterale für das entsprechende „verzierte“ Prädikat `increment_head_fb` und geben Sie die Bindungsmuster für die Rumpfliterale an, mit dem diese aufgerufen werden.

.....  
.....  
.....

**Aufgabe 4 (Polynomauswertung)****15 Punkte**

Definieren Sie ein Prädikat `polyeval` zur Auswertung eines Polynomes beliebigen Grades. Das Polynom wird als Liste von Polynomkoeffizienten dargestellt. Beispielsweise stellt `[a,b,c,d]` die Polynomfunktion  $x \mapsto a + b \cdot x + c \cdot x^2 + d \cdot x^3$  dar. Sie können den Ausdruck in dieser Form auswerten, Sie können aber auch das Horner-Schema verwenden, welches den Ausdruck in der Form  $a + x \cdot (b + x \cdot (c + x \cdot d))$  berechnet.

Beispiele:

```
?- polyeval([1,2,3], 10, Y).
```

```
Y = 321.
```

```
?- polyeval([], 42, Y).
```

```
Y = 0.
```