

**Aufgabe 1**

(25 Punkte)

**Logik**

a) Wodurch unterscheiden sich klassische Logik und effektive Logik?

b) Zeigen Sie die Gültigkeit des *Claviusschen Gesetzes*  $(\neg A \rightarrow A) \rightarrow A$  mithilfe eines klassischen und eines effektiven Dialogs.

Hinweis: Verwenden Sie im effektiven Fall *tertium non datur* als Hypothese.

- c) Wann ist eine Aussage kontradiktorisch? Überprüfen Sie anhand einer Wahrheitstafel, ob die Aussage  $\neg((a \wedge b) \vee \neg c) \longrightarrow ((b \vee a) \vee c)$  kontradiktorisch ist!

- d) Welche Eigenschaft einer Aussage kann mit einem semantischen Tableaux gezeigt werden? Zeigen Sie dies mit Hilfe eines semantischen Tableaux für die *Peircesche* Aussage  $((a \longrightarrow b) \longrightarrow a) \longrightarrow a!$

**Aufgabe 2**

(8 Punkte)

**Quantorenlogische Resolution**

a) Was ist SLD-Resolution? Unter welcher Bedingung ist sie vollständig?

b) Was ist der occur-check?

**Aufgabe 3**

(7 Punkte)

**Kalküle**

a) Geben Sie einen Kalkül  $\mathbb{K}_{Strich}$  an, welcher alle Strichlisten ungerader Länge erzeugt.

b) Wann heißt eine Regel  $R$ , die nicht im Kalkül  $\mathbb{K}$  enthalten ist,  $\mathbb{K}$ -zulässig? Geben Sie eine  $\mathbb{K}_{Strich}$ -zulässige Regel an.

**Aufgabe 4**

(18 Punkte)

**Prolog**

a) Was ergeben die folgenden Unifikationen in Prolog?

i) `?- super(G, I, R, L) = super(C, 0, 0, L).`ii) `?- [ [A], [a,b,c] ] = [ B, [A | C] ].`

b) Gegeben sei das folgende Prolog-Programm, welches die Liste L2 erzeugt, welche aus der Liste L1 ohne die jeweils N-ten Elemente (beginnend beim ersten Element) besteht.

Beispiel einer Anfrage:

```
?- drop([1,2,3,4,5,6,7], 2, Q).
Q = [1, 3, 5, 7] ;
fail.
```

Korrigieren sie die drei Fehler im Programm:

```
drop(L1,N,L2) :- drop(L1,N,L2,N).
```

```
drop([],_,[],_).
```

```
drop(_|Xs,N,Ys,1) :-
    drop(Xs,N,Ys,N).
```

```
drop([X,Xs],N,[X|Ys],K) :-
    K > 1,
    K1 == K - 1,
    drop(Xs,K1,Ys,K1).
```

c) Kodieren Sie folgende Datenbasis als Prolog-Programm:

<b>Name</b>	<b>Geburtsjahr</b>
Anna	1990
Bert	1986
Clara	1981

Geben Sie eine Anfrage an das Prolog-System an, mit welcher Sie überprüfen, ob Anna (mindestens ein Jahr) älter ist als Bert. Geben Sie dabei nicht erneut das Geburtsjahr ein, sondern greifen Sie auf das Gespeicherte zurück.

**Aufgabe 5**

(25 Punkte)

**Prolog**

- a) Implementieren Sie das Prädikat `numlist(Start, Stop, Liste)` in Prolog. Das Prädikat erzeugt die Liste `Liste` der aufsteigenden Zahlenfolge, die bei `Start` beginnt und bei `Stop` endet. Verwenden Sie keine vordefinierten Prolog-Prädikate außer den arithmetischen!

Beispiel eines Programmaufrufs:

```
?- numlist(2, 7, R).  
R = [2, 3, 4, 5, 6, 7].
```

- b)** Implementieren Sie das Prädikat `pick(Element, Liste, RestListe)` in Prolog. Das Prädikat wählt aus der Liste `Liste` ein Element `Element` aus und gibt `Liste` ohne das `Element` als `RestListe` zurück. Verwenden Sie keine vordefinierten Prolog-Prädikate außer den arithmetischen!

Beispiel eines Programmaufrufs:

```
pick(2, [1,2,3,2,5], Q).  
Q = [1, 3, 2, 5] ;  
Q = [1, 2, 3, 5] ;  
fail.
```



- c) Ein magisches Quadrat der Kantenlänge 3 ist eine quadratische Anordnung der Zahlen  $1, 2, \dots, 9$ , so dass die Summe aller Zeilen, Spalten und der beiden Diagonalen jeweils gleich 15 ist. Ergänzen Sie die fehlenden Prädikate `generate/1` und `test/1` des Prolog-Programms

```
magic(Q) :- generate(Q), test(Q).
```

welches eine Liste `Q = [A, B, C, D, E, F, G, H, I]` erzeugt, die dem magischen Quadrat aus Abbildung 1 entspricht.

A	B	C
D	E	F
G	H	I

Abbildung 1: Ein magisches Quadrat

`generate/1` erzeugt alle Permutation der Zahlen 1 bis 9. `test/1` überprüft, ob die übergebene Permutation ein magisches Quadrat ist.

Verwenden Sie, mit Ausnahme von `numlist`, `pick` (siehe vorherige Teilaufgaben) und den arithmetischen, keine vordefinierten Prolog-Prädikate. Beispiel eines Programmaufrufs:

```
?- magic(Q).  
Q = [2, 7, 6, 9, 5, 1, 4, 3, 8] ;  
Q = [2, 9, 4, 7, 5, 3, 6, 1, 8] .
```

Platz zur Bearbeitung der Aufgabe finden Sie auf der folgenden Seite!

`generate/1`-Prädikat:

`test/1`-Prädikat: