

5. Februar 2015

Klausur

Prolog Grundkurs WS 2014/15

Nachname: _____ Vorname: _____

Matrikelnummer: _____

Studienfach: _____ Semester: _____

Unterschrift: _____

Zugelassene Hilfsmittel: Keine

Diese Klausur enthält 9 nummerierte Klausurseiten. Prüfen Sie bitte zuerst, ob die Klausur alle Seiten enthält.

Erklärung:

Hiermit erkläre ich, dass ich mit der Veröffentlichung der Klausurergebnisse unter Angabe von Matrikelnummer und Note bzw. Punktzahl auf der WWW-Seite der Lehrveranstaltung einverstanden bin.

Unterschrift: _____

Diesen Teil bitte nicht ausfüllen:

Aufgabe	1	2	3	4	5	6	7	Σ
Punktzahl	5	15	10	25	5	10	15	85
Erreicht								

Hinweise:

Schalten Sie Ihr Mobiltelefon aus!

Täuschungsversuche führen zum sofortigen Ausschluss von der Klausur. Die Klausur wird dann als **nicht bestanden** gewertet!

Sie können auch die Rückseiten benutzen. Falls Sie zusätzliches Papier benötigen, werden Ihnen Bögen zur Verfügung gestellt.

Viel Erfolg!

Multiple Choice

Aufgabe 1

[5 Punkte]

Kringeln Sie die richtige Antwort ein. Bitte machen Sie klar deutlich, was Ihre Antwort ist.

Bewertung: Pro richtiger Antwort 1 Punkt. Für falsche Antworten werden keine Punkte abgezogen. Nicht angekreuzte Fragen werden nicht bewertet. Es können insgesamt also nicht weniger als 0 Punkte erzielt werden.

- (a) Was ist Prolog?
 - A. Eine logische Programmiersprache
 - B. Das letzte Kapitel eines Buches
 - C. Monoide aus der Kategorie der Endofunktoren
- (b) Die Liste $L = [1,2,3| T]$ repräsentiert eine valide Prolog-Liste.
 - A. Ja
 - B. Nein
 - C. Kommt auf den Wert von T an
- (c) Welche Anfrage kann in Prolog bewiesen werden?
 - A. $X = 1 + 2, X = 3.$
 - B. $X \text{ is } 1 + 2, X = 3.$
 - C. $1 + 2 = X, X = 3.$
- (d) Die Liste $L = [1, 2 | [3, [4]]]$ kann auch dargestellt werden als:
 - A. $L = [1, 2, 3, 4]$
 - B. $L = [1, 2, [3, [4]]]$
 - C. $L = [1, 2, 3, [4]]$
- (e) Welches Resultat liefert der Prolog Interpreter für die Anfrage
?- member(X, [1,2]), member(X,[3,4]).
 - A. false
 - B. $X = 3$
 - C. $X = 2$

Erklären Sie die Funktionsweise der Prädikate

Aufgabe 2

[15 Punkte]

Gegeben sind vier Prolog Prädikate.

Beschreiben Sie ihre Funktionsweise oder geben Sie ein äquivalentes Prädikat, welches in der Vorlesung oder Übung besprochen wurde an.

Schreiben Sie zudem einen **sinnvollen** Beispielaufruf zu jedem Prädikat!

```
predicate_a(X,[X|_]).  
predicate_a(X,[_|T]):-  
    predicate_a(X,T).
```

```
predicate_b([],L,L).  
predicate_b([H|T1],L,[H|T2]):-  
    predicate_b(T1,L,T2).
```

```
predicate_c(X,[],[]).  
predicate_c(X,[X|T],T).  
predicate_c(X,[H|T1],[_|T2]):-  
    predicate_c(X,T1,T2).
```

```
predicate_d(X,Y,Z):-  
    X >= Y, !,  
    Z = X.  
predicate_d(X,Y,Y).
```

```
predicate_e(A,C):-  
    predicate_e(A,[],C).  
predicate_e([],B,B).  
predicate_e([H|T],B,C):-  
    predicate_e(T,[H|B],C).
```

Definite Clause Grammars - DCG's

Aufgabe 3

[10 Punkte]

Schreiben Sie mithilfe einer DCG eine **reguläre Grammatik (Typ 3 Chomsky Hierarchie)**, welche die folgende Sprache aufspannt:

$a^n b^m$ wobei $n, m \in \mathbb{N} \wedge \neg (n = 0 \wedge m = 0)$ (mindestens ein a oder ein b)

Die Namenswahl für ihre Regeln ist beliebig, als Beispielaufruf wählen wir hier s:

```
?- s([a], []).  
?- true.
```

```
?- s([b], []).  
?- true.
```

```
?- s([], []).  
?- false.
```

```
?- s([a,a,a,b,b], []).  
?- true.
```

```
?- s([a,b,b,a], []).  
?- false.
```

Listenverarbeitung in Prolog

Sie dürfen alle in der Vorlesung besprochenen Prädikate verwenden!

Aufgabe 4

[25 Punkte]

- (a) [6 Punkte] Schreiben Sie das Prädikat `eins_mit_drei/2`, welches als ersten Parameter eine Liste als Eingabe erwartet und als zweiten Parameter ebenfalls eine Liste zurückgibt, in der das erste und dritte Element der Eingabe vertauscht sind.

(Sie müssen den Fehlerfall für Listen kleiner der Länge 3 nicht betrachten)

Beispielaufruf: `?- eins_mit_drei([1,2,3,4],R), R = [3,2,1,4].`

- (b) [9 Punkte] Schreiben Sie das Prädikat `vorletzt/2`, welches das **vorletzte Element** einer Liste ausgibt.

Tipp: Verwenden Sie `reverse/2` oder benutzen Sie rekursive Listenverarbeitung. Außerdem müssen Sie nicht den Fall abfangen, falls die Liste kleiner der Länge 2 ist.

Beispielaufruf:

`?- vorletzt([a, b], X).`

`?- X = a.`

`?- vorletzt([a, b, c], b).`

`?- true.`

- (c) [10 Punkte] Max ist sehr auf seine Ernährung bedacht und notiert sich die Zutaten seines Müslis in Prolog Listen.

Schreiben Sie das Prädikat `muesli/1`, welches als Eingabe eine Liste erwartet und überprüft ob diese nicht mehr als 3 beliebige Elemente enthält, wobei **schokolade nicht** enthalten sein darf.

Tipp: Verwenden sie `member/2`, `length/2` sowie den `not` Operator

Beispielaufruf:

`?- muesli([banane, apfel]). true.`

`?- muesli([banane, apfel, schokolade]). false.`

`?- muesli([banane, apfel, kirschen, trauben]). false.`

Cut Operator

Aufgabe 5

[5 Punkte]

Optimieren Sie die folgenden Prolog Prädikate mithilfe des Cut Operators !
Versuchen Sie so viele Zeilen wie möglich mithilfe des Cut Operators einzusparen.

(a) _____
min(X,Y,X): –
 X < Y.
min(X,Y,Y): –
 X >=Y.

(b) _____
abs(X,R): –
 X >= 0,
 R = X.
abs(X,R): –
 X < 0,
 R is -X.

Differenzlisten

Hierfür dürfen Sie **keine** vordefinierten Prädikate verwenden!

Aufgabe 6

[10 Punkte]

Schreiben Sie `append_dl/4`, wobei die ersten drei Parameter **Differenzlisten** sind welche im letzten Argument als konkatenierte Differenzliste zusammengefügt werden.

Muster:

`append_dl((DL1,T1), (DL2,T2), (DL3,T3), (DLR, TR))`.

Beispielaufruf:

`append_dl(([1, 2 | T],T1), ([3 | T2],T2), ([4 | T3],T3), ([1,2,3,4 | T3], T3))`

BONUSAUFGABE

Univ Operator =..

Aufgabe 7

[15 Punkte]

Schreiben Sie das Prädikat `reverse_call/1`, welches als Eingabe ein bestehendes Prolog Prädikat erwartet und dessen Argumente umdreht und dann aufruft.

Beispielaufruf:

```
?- reverse_call(abs(X,-3)).
```

```
?- X = 3
```

Tips:

`abs(X,-3)` wird dann wie folgt aufgerufen: `abs(-3,X)`

`append(X,L2,L1)` wird dann wie folgt aufgerufen: `append(L1,L2,X)`.

Zur Erinnerung: Prädikat `=..` Liste

`reverse/2` wird ihnen helfen!