Übungsaufgaben zur Lehrveranstaltung

Programmierparadigmen der Künstlichen Intelligenz

des Studiengangs

Wirtschaftsinformatik

an der

Technischen Universität Ilmenau

apl. Prof. Dr.-Ing. habil. Rainer Knauf

Inhaltsverzeichnis

1 Einführung

Die Übungaufgaben sind entsprechend der Gliederung der Vorlesung kapitelweise einzeln durch nummeriert. Zum Kapitel „1. Einführung“ gibt es allerdings keine Übungsaufgaben.

2 Mathematisch–logische Grundlagen

1. Seien A und B einfache Aussagen. Man zeige

   (a) durch Bestimmung des Werteverlaufs der Wahrheitswerte,

   (b) durch äquivalente Umformungen,

   dass folgende zusammengesetzte Aussagen allgemeingültig (Tautologien) sind:

   - $(A \land B) \to A$
   - $(A \to B) \lor (A \land \neg B)$
   - $(A \to B) \leftrightarrow (\neg A \lor B)$

2. Seien A und B einfache Aussagen. Man zeige

   (a) durch Bestimmung des Werteverlaufs der Wahrheitswerte,

(b) durch äquivalente Umformungen,

dass folgende zusammengesetzte Aussagen Kontradiktionen sind:

- $\neg((A \wedge B) \rightarrow A)$
- $\neg((A \rightarrow B) \vee (A \wedge \neg B))$

3. Gegeben sei ein (endlicher) Individuenbereich I = {a,b}.
   Man zeige durch Umwandlung in variablenfreie Aussagen und äquivalente Umformungen, dass folgende prädikatenlogisch formulierten Aussagen allgemeingültig sind.

   (a) $\forall X(p(X) \vee \neg p(X))$
   (b) $\forall X p(X) \rightarrow \exists X p(X)$
   (c) $\neg\forall X p(X) \leftrightarrow \exists X \neg p(X)$
   (d) $\neg\exists X p(X) \rightarrow \forall X \neg p(X)$
   (e) $(\forall X p(X) \wedge \forall Y q(Y)) \leftrightarrow \forall X(p(X) \wedge q(X))$

4. Gegeben sei ein (endlicher) Individuenbereich I = $\{c_1, \ldots, c_n\}$.
   Man zeige durch Umwandlung in variablenfreie Aussagen und äquivalente Umformungen, dass folgende prädikatenlogisch formulierten Aussagen allgemeingültig sind.

   (a) $\forall X \neg p(X) \rightarrow \neg\exists Y p(Y)$
   (b) $(\forall X p(X) \vee \forall Y q(Y)) \rightarrow \forall X(p(X) \vee q(X))$

5. Man formuliere folgende Sachverhalte als prädikatenlogische Aussagen.

   Individuenbereich: Menschen (z.B. TUI-Angehörige)

   (a) Es gibt niemanden, der von jedem geliebt wird.
   (b) Es gibt Menschen, die nicht irren. (. . . nämlich diejenigen, die nichts von sich geben)
   (c) Ein denkender Mensch irrt besonders oft.
   (d) Wenn man von Leuten Pflichten fordert und ihnen keine Rechte gibt, muss man sie gut bezahlen.
   (e) Wir lieben nicht alle, die wir bewundern.
   (f) Wir lieben alle, die uns bewundern.

6. Man entscheide für folgende syntaktische Gebilde, ob es sich um Aussagen bzw. Ausdrücke handelt :

   (a) $\forall X gleich(X, X)$
   (b) $\forall X((gleich(X, Y) \wedge gleich(X, Z)) \rightarrow gleich(Y, Z))$
   (c) vater_von(dieter,X)
   (d) $\exists X bruder\_von(rainer, X)$
   (e) $\neg\exists X mutter\_von(X, eva)$
   (f) $\forall X \forall Y(liiert(X, Y) \leftarrow \exists Z(vater\_von(X, Z) \wedge mutter\_von(Y, Z)))$
   (g) vater_von bernd dieter
   (h) $\forall X liebt(X, Y)$

(i) $\forall X \wedge \forall Y liebt(X, Y)$

(j) $\forall X \forall Y (groesser(X, Y) \rightarrow groesser(f(X), f(Y)))$

(k) $\forall X \exists Y groesser(X, Y) \rightarrow groesser(f(X), f(Y))$

7. Man notiere folgenden prädikatenlogischen Ausdruck durch Anwendung der Vereinbarungen zur Einsparung von Klammern kürzer:

   $(\forall Z(((((\forall X \neg p(X) \wedge q(Y)) \vee p(Y)) \rightarrow q(Z)) \leftrightarrow p(Z)) \rightarrow p(X)) \wedge p(X))$

8. Man zeige folgende Äquivalenzen:

   (a) $\neg \forall X (weibl(X) \rightarrow \exists Y verheir(X, Y)) \equiv \exists X (weibl(X) \wedge \forall Y \neg verheir(X, Y))$

   (b) $\forall X (p(X) \leftarrow q(X) \vee r(X)) \equiv \forall X (p(X) \leftarrow q(X)) \wedge \forall X (p(X) \leftarrow r(X))$

9. Man forme folgenden prädikatenlogischen Ausdruck so um, dass die Quantoren nur noch vor dem Gesamtausdruck stehen :

   $\forall X \exists Y p(X, Y) \rightarrow \exists Y q(Y)$

10. Für einen Individuenbereich I = {a,b} zeige man, dass

    (a) $\{p(a), \forall X (p(X) \rightarrow q(X))\} \models q(a)$

    (b) $\{\forall X (p(X) \rightarrow q(X)), \forall X \neg q(X)\} \models \forall X \neg p(X)$

    (c) $\{p(a), \forall X (p(X) \rightarrow q(X))\} \models \exists X q(X)$

11. Für folgende syntaktische Gebilde entscheide man, ob es sich um HORN-Klauseln handelt:

    (a) $\forall X (gleich(X, X) \leftarrow true)$

    (b) $\forall X \forall Y (liiert(X, Y) \leftarrow \exists Z (vater(X, Z) \wedge mutter(Y, Z)))$

    (c) $\forall X \forall Y \forall Z (liiert(X, Y) \leftarrow vater(X, Z) \wedge mutter(Y, Z))$

    (d) $\forall X \forall Y \forall Z (grossvater(X, Z) \leftarrow (vater(X, Y) \wedge vater(Y, Z)) \vee$
    $(vater(X, Y) \wedge mutter(Y, Z)))$

    Sind die Aussagen 11b und 11c äquivalent ?

12. Man wandle in Klauselform um:

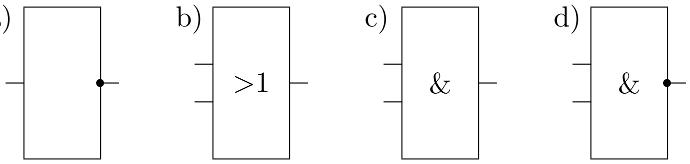
    (a) $\forall X p(X) \rightarrow \forall X p(X)$

    (b) $\exists X \neg p(X) \wedge (\exists X p(X) \vee \exists X (p(X) \wedge q(X))) \wedge \neg \exists X p(X)$

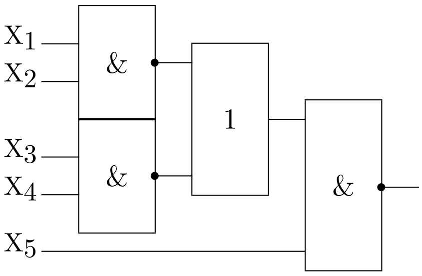
    Sind die entstehenden Klauseln HORN–Klauseln?

13. Für einen Individuenbereich I = {0,1} formuliere man HORN-Klauseln, die die Funktion folgender digitaler Schaltungen beschreiben:

a) b) c) d)



e)



14. Man zeige durch wiederholte Anwendung der Resolutionsmethode nach ROBINSON, dass aus den HORN-Klauseln

    K_1: befreundet(frank,steffen)
    K_2: befreundet(steffen,uwe)
    K_3: bekannt(X,Y) $\leftarrow$ befreundet(X,Z) $\wedge$ befreundet(Z,Y)

    die Hypothese **H: bekannt(frank,Wer)** folgt[1].

3 Einführung in die Logische Programmierung mit PROLOG

1. Über einem Individuenbereich I = {otto,emma,...} seien folgende Prädikate definiert:

Prädikat	**Bedeutung**
maennlich(X)	bildet [X] auf wahr ab, gdw. X eine männliche Person ist
weiblich(X)	bildet [Y] auf wahr ab, gdw. Y eine weibliche Person ist
vater_von(X,Y)	bildet [X,Y] auf wahr ab, gdw. X der Vater von Y ist
mutter_von(X,Y)	bildet [X,Y] auf wahr ab, gdw. X die Mutter von Y ist
verheiratet(X,Y)	bildet [X,Y] auf wahr ab, gdw. X und Y miteinander verheiratet sind

   Unter Zuhilfenahme dieser Prädikate formuliere man HORN-Klauseln, die Beziehungen zwischen den Elementen des Individuenbereiches beschreiben, indem sie die nachfolgenden Prädikate definieren:

[1] Die Variablenersetzung in der Hypothese liefert zugleich eine konstruktive Lösung der Beweisaufgabe.

Prädikat	Bedeutung
geschwister(X,Y)	bildet [X,Y] auf wahr ab, gdw. X und Y den gleichen Vater und die gleiche Mutter haben
cousin_von(X,Y)	bildet [X,Y] auf wahr ab, gdw. X Sohn eines Geschwisters eines Elternteils von Y ist
cousine_von(X,Y)	bildet [X,Y] auf wahr ab, gdw. X Tochter eines Geschwisters eines Elternteils von Y ist
schwager_von(X,Y)	bildet [X,Y] auf wahr ab, gdw. X der Mann einer Schwester von Y oder Bruder des Ehepartners von Y ist
schwaegerin_von(X,Y)	bildet [X,Y] auf wahr ab, gdw. X Frau eines Bruders von Y oder Schwester des Ehepartners von Y ist
nichte_von(X,Y)	bildet [X,Y] auf wahr ab, gdw. X Tochter eines Schwagers oder einer Schwägerin oder eines Geschwisters von Y ist
neffe_von(X,Y)	bildet [X,Y] auf wahr ab, gdw. X Sohn eines Schwagers oder einer Schwägerin oder eines Geschwisters von Y ist

2. Man notiere die HORN-Klauseln aus Aufgabe 13 des vorangegangenen Kapitels als Wissensbasis eines PROLOG-Programms.

3. Gegeben sei folgende Wissensbasis:

```
(1)  mutter_von(hilde,elsa).
(2)  vater_von(egon,elsa).
(3)  vater_von(paul,otto).
(4)  mutter_von(erna,otto).
(5)  mutter_von(elsa,gaby).
(6)  mutter_von(elsa,dieter).
(7)  vater_von(otto,gaby).
(8)  vater_von(otto,dieter).
(9)  grossvater_von(Grossvater,Enkel) :- vater_von(Grossvater,Vater),
                                          vater_von(Vater,Enkel).
(10) grossvater_von(Grossvater,Enkel) :- vater_von(Grossvater,Mutter),
                                          mutter_von(Mutter,Enkel).
```

(a) Man veranschauliche die Abarbeitung folgender Fragen anhand von Suchbäumen:
   i. ?-vater_von(paul,dieter).
   ii. ?-mutter_von(X,gaby).
   iii. ?-grossvater_von(egon,Wer).
   iv. ?-grossvater_von(Wer,dieter).

(b) Man erweitere die Wissensbasis um folgende Prädikate:

Prädikat	Bedeutung
geschwister(X,Y)	bildet (X,Y) auf wahr ab, gdw. X und Y gleichen Vater und gleiche Mutter haben und voneinander verschieden[2]sind.
geboren(X,Y)	bildet (X,Y) auf wahr ab, gdw. Y das Geburtsdatum der Person X ist. Man überlege sich eine geeignete Termstruktur für Y.
zwillinge(X,Y)	bildet (X,Y) auf wahr ab, gdw. X und Y am gleichen Tag geboren sind und die gleiche Mutter haben.

(c) Mit Hilfe dieser Prädikate formuliere man folgende Fragen und veranschauliche deren Bearbeitung durch ein PROLOG-System:

i. Wer wurde am 1. April geboren?
ii. Wer wurde im Jahre 1960 geboren?
iii. Wann hat Egon Geburtstag?
iv. Gibt es ein Zwillingspaar in der Familie?
v. Welche Geschwisterpaare gibt es?

4. Gegeben sei folgende Wissensbasis:
(1) teil_von(anker,anlasser).
(2) teil_von(zugmagnet,anlasser).
(3) teil_von(anlasser,elektrik).
(4) teil_von(elektrik,auto).

„teil_von(X, Y)“ soll [X,Y] auf wahr abbilden, gdw. X ein Bestandteil von Y ist.

Man erweitere diese Wissensbasis um ein Prädikat „teiles_teil(X, Y)“, welches [X,Y] auf wahr abbildet, gdw. X ein direkter oder indirekter Bestandteil von Y ist. Man veranschauliche die Abarbeitung von Fragen der Art

?-teiles_teil(X, Y)

mit Hilfe von Suchbäumen.

4 Noch 'ne KI–Sprache: LISP

1. Man notiere den entsprechenden LISP–Ausdruck für

(a) (36 : 9) : 3

(b) 36 : (9 : 3)

(c) $(7 * \frac{8-\frac{9}{11}}{14} - (\frac{7}{5})^2) * (11 - \frac{3*(4^2-\frac{12}{4})+5}{4})$

2. Man übersetze folgende LISP–Ausdrücke in arithmetische Infix–Ausdrücke:

(a) (+ (* 0.1 20)(/ 4 -32))

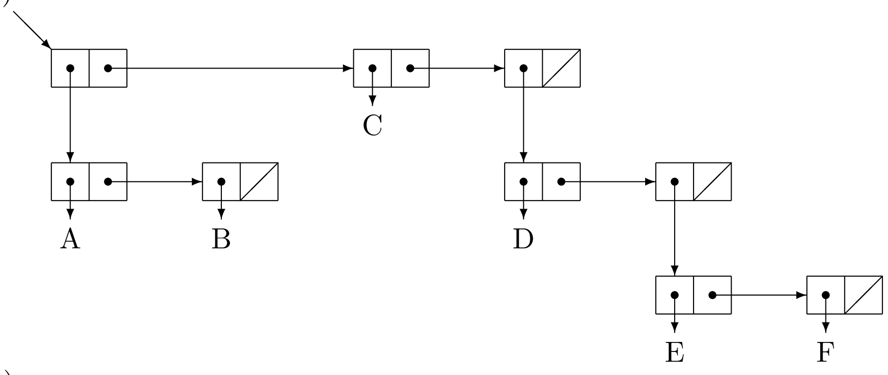
(b) (- (- (- 5 (- 2 100)) 10) (- 8.5 3.5))

3. Man notiere folgende Listen als LISP–Ausdruck

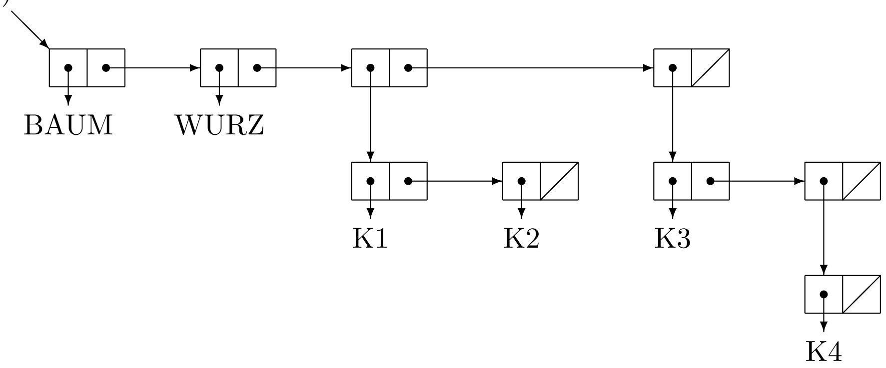
- in (vereinfachter) Listennotation und
- als Struktur verschachtelter gepunkteter Paare.

[2]Ein Prädikat „ungleich(X,Y)“ kann als eingebaut vorausgesetzt werden.

(a)



(b)



4. Man veranschauliche die interne Repräsentation folgender Listen:
   (a) (A.(B.(C.NIL)))
   (b) ((A.(B)).(C.NIL))
   (c) (A.((B.NIL).(C.((D.(E)).NIL))))
   (d) (A B C)
   (e) ((A B) C)
   (f) (A (B C))
   (g) ((A B C))
   (h) (A B (C D) (E F G) H (I J) K)

5. Man notiere einen LISP–Ausdruck für

$$(\frac{11}{2} - 3)^3 + (\frac{11}{2} - 3)^2 + \frac{11}{2}$$

   (a) mit vorheriger Zuweisung von Teilausdrücken an Variablen durch `SETQ`.
   (b) ohne die vorherige Zuweisung von Teilausdrücken an Variablen.

6. Man binde
   (a) an die Variable X den (numerischen) Wert durch Auswertung des LISP–Ausdrucks
   (b) an die Variable Y den LISP–Ausdruck

   von $3 \cdot (5 + \frac{7}{2})$.

7. Man greife mit Hilfe von `CAR` und `CDR` sowie abkürzenden Schreibweisen von Verschachtelungen dieser Funktionen auf

(a) das dritte Element
(b) das erste Element des zweiten Elements[3]
(c) das zweite Element des dritten Elements des ersten Elements
(d) die Liste der hinter dem dritten Element stehenden Elemente
(e) die Liste der hinter dem zweiten Element des ersten Elements stehenden Elemente
(f) die Liste der hinter dem dritten Element des zweiten Elements des ersten Elements

der Liste `LISTE` zu.

8. Man
   (a) konstruiere die Liste '((A B) () (C D) (E))
      i. mit Hilfe der Funktion `LIST`,
      ii. mit Hilfe der Funktion `CONS` (und ohne Verwendung von `LIST`),
   (b) binde das Ergebnis an die Variable `X` und
   (c) erzeuge eine Liste `LISTE_2`, deren erstes Element `27` ist und deren Restliste die an `X` gebundene Liste ist.

9. Welche Funktionswerte entstehen bei der Auswertung folgender LISP–Ausdrücke?
   (a) `(ATOM (A))`
   (b) `(ATOM 27)`
   (c) `(NUMBERP (27 12 15))`
   (d) `(NUMBERP (27))`
   (e) `(NUMBERP 27)`
   (f) `(NULL ())`
   (g) `(NULL NIL)`
   (h) `(NULL (NULL NIL))`
   (i) `(NULL (ATOM 'A))`
   (j) `(NULL '(ATOM A))`
   (k) `(NULL '(NULL NIL))`
   (l) `(LISTP (LIST (SETQ X 27) (+ X X)))`
   (m) `(LISTP NIL)`

10. (a) Man erzeuge vier Listen `LISTE_1`, `LISTE_2`, `LISTE_3` und `LISTE_4` mit folgenden Eigenschaften:
      - `LISTE_3` und `LISTE_4` sollen bei Ihrer Auswertung (B C) liefern und physisch identisch sein.
      - `LISTE_1` soll als erstes Element A, als zweites Element `LISTE_3` und als drittes Element D haben.
      - `LISTE_2` soll als erstes Element A, als zweites Element `LISTE_4` und als drittes Element D haben, aber nicht mit `LISTE_1` physisch identisch sein.
    (b) Man veranschauliche die o.g. Listenstrukturen in einer Skizze.

[3]. . . , welches dann natürlich seinerseits eine Liste ist, . . .

(c) Welche Funktionswerte entstehen bei der Auswertung folgender Ausdrücke?
   i. `(EQUAL LISTE_1 LISTE_2)`
   ii. `(EQUAL LISTE_3 LISTE_4)`
   iii. `(EQUAL LISTE_1 LISTE_4)`
   iv. `(EQL LISTE_1 LISTE_2)`
   v. `(EQL LISTE_3 LISTE_4)`

11. Man definiere eine Funktion, die die Liste der Elemente ab dem dritten Element einer gegebenen (Argument–) Liste liefert

    (a) mit Hilfe eines LAMBDA–Ausdruckes
    (b) mit Hilfe der Funktion `DEFUN` und einem anschließenden Aufruf

    und wende sie auf die Liste (A B C D) an.

12. Man weise einer Variablen `LISTE` die Liste (N E U) zu, tausche darin das 2. Element durch das Element A aus, kehre die Reihenfolge der Elemente der entstandenen um, hänge daran die Liste (E N R A) an, tausche die (Rest–) Liste der nach dem 4. Element stehenden Elemente durch (M L I) aus und kehre anschließend die gesamte dabei entstandene Liste um.

    Welche Liste entsteht dabei?

13. Man definiere folgende Funktionen rekursiv in LISP:

    (a) Die Funktion `B_LISTE` soll prüfen, ob das erste Element einer ihr als Argument übergebenen Liste ein B ist und in diesem Falle die Liste selbst als Funktionswert zurückgeben. Anderenfalls soll `NIL` der Funktionswert sein.
    (b) Die Funktion `FAK` soll die Fakultät einer ihr als Argument übergebenen positiven ganzen Zahl liefern.
    (c) Die Funktion `SUM_TO` soll die Summe der ihr in einer Liste übergebenen Elemente bilden. Die Elemente können Zahlen sein oder arithmetische Ausdrücke, bei deren Auswertung Zahlen entstehen.
    (d) Die Funktion `ATOMLISTE` soll prüfen, ob alle Elemente der ihr übergebenen Liste Atome sind und in diesem Falle `T` und anderenfalls `NIL` liefern.
    (e) Die Funktion `ATOMLISTE_1` soll prüfen, ob alle Elemente der ihr übergebenen Liste Atome sind und in diesem Falle die Liste selbst sowie anderenfalls `NIL` liefern.
    (f) Die Funktion `ERSTES_ATOM` soll prüfen, ob das erste Element der ihr übergebenen Liste atomar ist und in diesem Falle dieses Element liefern. Sollte das erste Element nicht atomar (und demnach eine Liste) sein, soll die Funktion rekursiv für das erste Element aufgerufen werden und dessen Funktionswert liefern.
    (g) Die Funktion `LIST` (Bildung einer Liste) ist unter Verwendung der Funktion `CONS` selbst zu definieren.

14. Für das LISP–Programm aus dem Abschnitt 4.6 der Vorlesung („Ein Beispiel: Tiefe–zuerst–Suche in Graphen“) zur Ermittlung eines Erreichbarkeitsbaumes eines gerichteten Graphen und den Beispielgraphen

```
((2 3)(1 5)(3 1)(1 4)(5 6)(4 2)(6 5)(6 3)(6 4))
```

(a) vollziehe man die Aufrufe der Funktion `SUCHSCHRITT` nach, d.h. gebe für jeden Aufruf die aktuellen Parameter, den sich ergebenden Restgraphen und den sich daraus ergebenden Neuaufruf an.

(b) gebe man die Anzahl der Aufrufe an, die nötig sind, um den Abbruch der Rekursion zu erreichen.

(c) veranschauliche man die interne Repräsentation der Aufrufvariablen `GRAPH`, `STACK` und `TEILGRAPH` beim 9. Aufruf der Funktion.

5 Weitere Teilgebiete der KI im Überblick

1. Gegeben sei eine 2-stellige assoziative und kommutative Funktion $*$, d.h. es gilt

$$\begin{aligned} (X * Y) * Z &= X * (Y * Z) \\ X * Y &= Y * X \end{aligned}$$

Man

(a) formuliere diese Aussagen im PK1 unter Verwendung des (infix notierten) Gleichheitsprädikats $\equiv /2$,

(b) füge dieser Aussagenmenge die Klauseln

$$\begin{aligned} p(*(b, *(a, b)), a) &\leftarrow true \\ \forall X \forall Y (q(*(X, Y)) &\leftarrow p(X, Y)) \end{aligned}$$

hinzu und

(c) zeige durch Hintereinanderanwendung von Resolutionsmethode und/oder Paramodulationsregel, dass aus der so entstehenden Aussagenmenge die Aussage

$$q(*(*(a, b), *(b, a)))$$

folgt.

2. Man formuliere folgende Logelei

(a) in Form von HORN–Klauseln des PK1
  i. ohne Sorten (und stattdessen mit Sortenprädikaten)
  ii. mit Sorten

(b) als PROLOG–Programm, wobei hier zusätzlich eine PROLOG–Frage zu formulieren ist, die den Beweis erbringt, dass es ein Tier gibt, welches ein getreidefressendes Tier frisst.

Wölfe, Füchse, Vögel, Tausendfüßler und Schnecken sind Tiere, und von jeder Art existiert mindestens ein Exemplar. Außerdem gibt es Getreide, welches zu den Pflanzen gehört.

Jedes Tier frisst entweder

- *alle Pflanzen oder*
- *alle Tiere, die kleiner sind als es selbst und Pflanzen fressen.*

Tausendfüßler und Schnecken sind kleiner als Vögel, diese sind kleiner als Füchse und diese sind kleiner als Wölfe.

Wölfe fressen keine Füchse und keine Pflanzen, wogegen Vögel zwar Tausendfüßler fressen, aber keine Schnecken. Sowohl Tausendfüßler als auch Schnecken fressen Pflanzen.

3. Man finde eine Aussage ϕ und 91 Aussagen $\phi_1, \ldots \phi_{91}$, so dass $\{\phi_1, \ldots, \phi_{91}\} \models \phi$ gilt, aber ϕ nicht mehr folgt, wenn man eine beliebige Prämisse wegläßt.

4. Man entscheide, ob folgende Termpaare s und t miteinander unifizierbar sind und bestimme im Erfolgsfall den allgemeinsten Unifikator $m.g.u.(s, t)$.

   (a) $s = f(X, Y)$ $\quad t = f(g(a, Y), Z)$

   (b) $s = f(X, Y)$ $\quad t = f(g(a, Y), Y)$

   (c) $s = f(X, Y)$ $\quad t = f(g(a, Y), X)$

   (d) $s = f(X, Y)$ $\quad t = f(g(a, Y), f(X))$

   (e) $s = f(X, Y, h(X, Y, g(X, Y)))$ $\quad t = f(g(a, Y), b, Z)$

5. Sei $\mathbf{N}$ die Menge der natürlichen Zahlen, $s : \mathbf{N} \to \mathbf{N}$ die Nachfolgerabbildung und $\mathbf{V}$ die Menge aller Vielfachen von 4. Das Prädikat $g(X)$ sei wahr genau dann, wenn X gerade ist.

   Gegeben seien folgende Voraussetzungen:

$$\begin{array}{l} \forall X \in \mathbf{V} g(X) \\ \forall X \in \mathbf{N}(g(X) \leftrightarrow \neg g(s(X))) \\ 4 \in \mathbf{V} \\ \forall W(W \in \mathbf{N} \to s(W) \in \mathbf{N}) \\ \forall U(U \in \mathbf{V} \to U \in \mathbf{N}) \end{array}$$

   Daraus folgt die Behauptung $g(s(s(4)))$.

   (a) Man wandle die Voraussetzungen und die Behauptung in eine Klauselmenge um. Man verwende dabei für $\mathbf{V}$ und $\mathbf{N}$ 1-stellige Sortenprädikate.

   (b) Man leite aus dieser Menge mittels Resolutionsmethode und Faktorenregel die leere Klausel ab; man führe dabei Schritte auf g vor Schritten auf $\mathbf{V}$ und $\mathbf{N}$ durch.

6. Man finde je ein Beispiel für eine Menge von Ausdrücken in Klauselform $\{K_1, \ldots, K_n\}$ mit

$$kt \bigwedge_{i=1}^{n} K_i$$

   , so dass die Hintereinanderanwendung von Resolutionsmethode und Faktorenregel

   (a) nach endlich vielen Schritten terminiert, d.h. keine neuen Klauseln mehr erzeugt und

   (b) nicht terminiert.

7. Kann es eine endliche Klauselmenge M geben, so dass man durch Resolutionsmethode und Faktorenregel die leere Klausel $\Box$ ableiten kann, aber $\Box$ nicht aus drei beliebig vorgegebenen Klauseln aus Mableitbar ist?

   Im positiven Falle gebe man ein Beispiel und im negativen Falle beweise man die Behauptung.

8. Man leite aus folgenden Klauselmengen die leere Klausel ($false \leftarrow true$) ab nach Resolutionsmethode und Faktorenregel ab und gebe für jeden Ableitungsschritt die Variablenersetzung an:

   (a) K_1: $p(X, a, g(X, b)) \leftarrow true$
       K_2: $false \leftarrow p(f(Y), Z, g(f(a), b))$

   (b) K_1: $q(g(X)) \leftarrow true$
       K_2: $q(f(Y)) \leftarrow q(Y)$
       K_3: $false \leftarrow q(f(f(g(Z))))$

   (c) K_1: $r(a) \leftarrow true$
       K_2: $r(f(f(X))) \leftarrow true$
       K_3: $false \leftarrow r(Y) \land s(Y)$
       K_4: $s(Z) \lor s(f(Z)) \leftarrow true$