/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Übung 6: Lösung logischer Aufgaben vom Typ "Insel der Ritter & Lügner",die in Textform gegeben sind

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Die Bewohner dieser Insel sind entweder Ritter oder Lügner. Ritter sagen immer die Wahrheit, Lügner immer die Unwahrheit. Die Aufgabe besteht darin, aus den Aussagen einiger Insulaner deren Identität zu bestimmen.

Z.B. sind folgende Aussagen zweier Insulaner a und b gegeben: a sagt dass jeder von ihnen ein Lügner ist. b sagt etwas unverständliches. Daraus lässt sich logisch ableiten, dass a ein Lügner und b ein Ritter ist.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Es ist ein PROLOG-programm zu entwickeln, das

1. die Aussagen der Insulaner in einen logischen Ausdruck in Präfix-Notation konvertiert: so wird aus der Aussage "a sagt dass jeder von ihnen ein Lügner ist" der logische Ausdruck

(<-> a (& (! a) (! b))). Benutzt werden für die logischen Operationen die gleichen Symbole wie in der Übung 3 von „Funktionale Programmierung“

2. den logischen Ausdruck auswertet, d.h. dessen Wurzeln bestimmt. Der Ausdruck

(<-> a (& (! a) (! b))) hat genau eine Wurzel: a=0, b=1.

3. das Ergebnis der Auswertung des logischen Ausdrucks wieder in eine verbale Darstellung

konvertiert. dabei gibt es drei Fälle:

* 1. der Ausdruck hat keine Wurzel => die Aussagen sind widersprüchlich - keine Schlussfolgerung möglich.
  2. der Ausdruck hat eine Wurzel => Bestimmung der Identität aller beteiligten Insulaner: die Wurzel a=0, b=1 bedeutet: a ist ein Lügner, b ist ein Ritter.
  3. der Ausdruck hat mehrere Wurzeln => das bedeutet, dass für die Insulaner deren Werte nicht in allen Wurzeln gleich sind, nicht bestimmt werden kann ob sie jeweils ein Ritter oder ein Lügner sind.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Zur Vereinfachung ist den eigentlichen Aussagen ein Satz vorangestellt, aus dem folgt, um welche Insulaner es sich handelt. Z.B. [es,gibt,folgende,insulaner,a,b]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Insgesamt sind alle logischen Aufgaben in dem File "aufgaben\_ritter\_luegner.pl" als Fakten gespeichert. Die Fakten haben die Struktur:

aufgabe(<nr>,[es,gibt,folgende,insulaner,...],<listen der Einzelaussagen>).\*/

Die Fakten werden durch "consult( <file>)" geladen.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Gegeben sind folgende mögliche Formen der Aussagen

|  |  |  |
| --- | --- | --- |
| N | AUSSAGE | LOGIK |
| 1 | a sagt etwas unverständliches | 1 |
| 2 | a sagt dass er e/k r/l ist | (<-> a a) |
| 3 | a sagt dass b e/k r/l ist | (<-> a b) |
| 4 | a sagt dass er e/k r/l ist und dass b e/k r/l ist | (<-> a a)&(<-> a a) |
| 5 | a sagt dass b e/k r/l ist und dass er e/k r/l ist | (<-> a b)&(<-> a a) |
| 6 | a sagt dass b e/k r/l ist und dass c e/k r/l ist | (<-> a a)&(<-> a b) |
| 7 | a sagt dass er e/k r/l ist oder dass b e/k r/l ist | (<-> a a)v(<-> a a) |
| 8 | a sagt dass b e/k r/l ist oder dass er e/k r/l ist | (<-> a b)v(<-> a a) |
| 9 | a sagt dass b e/k r/l ist oder dass c e/k r/l ist | (<-> a a)v(<-> a b) |
| 10 | a s d entweder er e/k r/l ist oder b ist e/k r/l | (<-> a(+ a b)) |
| 11 | a s d entweder b e/k r/l ist oder er ist e/k r/l | (<-> a(+ b e)) |
| 12 | a s d entweder b e/k r/l ist oder c ist e/k r/l | (<-> a(+ b c)) |
| 13 | a sagt dass wenn er e/k r/l ist dann ist b e/k r/l | (<-> a(-> a b)) |
| 14 | a sagt dass wenn b e/k r/l ist dann ist er e/k r/l | (<-> a(-> b a)) |
| 15 | a sagt dass wenn b e/k r/l ist dann ist c e/k r/l | (<-> a(-> b c)) |
| 16 | a sagt dass mindestens einer von ihnen ein r/l ist | (<-> a(v a b c)) |
| 17 | a sagt dass genau einer von ihnen ein r/l ist | (<-> a(v (& a (! b) (! c)) (& (! a) b (! c)) (& (! a) (! b) c))) |
| 18 | a sagt dass keiner von ihnen ein r/l ist | (<-> a(& (! a) (! b) (! c))) |
| 19 | a sagt dass jeder von ihnen ein r/l ist | (<-> a(& a b c)) |

Abkürzungen: s = sagt, d = dass, e/k = ein/kein , r/l = Ritter/Lügner. Die Semantik ist jeweils für die Kombination „e,r“ = „ein Ritter“ angegeben.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Einen Gesamtüberblick über die Aufrufstruktur der einzelnen Prädikate - welche Prädikate welche anderen Prädikate benutzen - finden sie in der Datei "Schema der Prädikatenaufrufstruktur.docx".

Empfohlen wird eine BOTTOM UP Vorgehensweise: Definieren und testen Sie zuerst die einfachen Prädikate danach schrittweise die auf diesen aufbauenden komplexeren Prädikate.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Laden der Aufgabendatei

/\*-----------------------------------------------------------------------------------------------------------------------\*/

:- consult('d:/LEHRE/KIBASICS/AUFGABEN/AUFGABEN.19/uebung6/

aufgaben\_ritter\_luegner.pl').

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1: Konvertierung des Aufgaben Textes in einen logischen Ausdruck

/\*-----------------------------------------------------------------------------------------------------------------------\*/

Die Aufgabentexte sind in einem gesonderten File, dessen Inhalt wie folgt geladen wird:

consult('q:/stiefel\WV\_GRUNDLAGEN.BCH/Uebung6/ aufgaben\_ritter\_luegner.pl‘).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.1: Definieren sie das Prädikat "insulaner\_l/3", das aus dem jeweiligen Fakt der Aufagbenstellung die Angabe über die involvierten Insulaner extrahiert. Das Prädikat arbeitet mit einer Differenzliste.

Die Argumente des Prädikats sind:

IL = Liste der Insulaner

AIL = Aussage zu den beteilgten Insulanern = 1. Teil der Differenzliste

REST = Restliste = 2. Teil der Differenzliste

/\*-----------------------------------------------------------------------------------------------------------------------\*/

?-aufgabe(1,AIL,AUSGL),insulaner\_l(IL,AIL,[]).

AIL=[es,gibt,folgende,insulaner,a,b],

AUSGL=[[a,sagt,dass,er,ein,ritter,ist],[b,sagt,dass,a,ein,luegner,ist]],

IL=[a,b]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.2: Definieren sie das Prädikat "konv/2", das {ein,kein} in {ritter,luegner} konvertiert. Durch die Konvertierung ist ein Vergleich RL=?=EK möglich wodurch anstelle der 4 Kombinationen von {ein,kein} und {ritter,luegner} nur 2 Kombinationen betrachtet werden müssen, da "ein ritter = kein Lügner" und "kein ritter = ein Lügner" ist.

Die Argumente des Prädikts sind:

EK = {ein,kein}

RL = {ritter,luegner}

/\*-----------------------------------------------------------------------------------------------------------------------\*/

konv(ein,ritter). konv(kein,luegner).

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% konv(ein,RL). RL=ritter

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.3: Definieren sie das Prädikat "el\_aus/4. Das Prädikat wandelt eine elementare Aussage wie "X ist ein/kein Ritter/Lügner" in eine negierte bzw. unnegierte Aussagenvariable um: x bzw. [!,x]. Dabei wird die mögliche unterschiedliche Wortreihenfolge berücksichtigt, die von dem Kontext bestimmt wird. Es gibt 3 Typen der Elementaraussagen in Abhängigkeit von der Position des Verbs "ist":

Typ1: "ist" am Satzanfang. Beispiel ... Dann ist x ein Ritter

Typ2: "ist" an 2. Stelle des Satzes. Beispiel ... X ist ein Ritter

Typ3: "ist" am Satzende. Beispiel ... Dass x ein Ritter ist

Diese Typinformation wird vom aufrufenden Prädikat geliefert.

Das Prädikat benutzt das Prädikat "konv/2".

Das Prädikat arbeitet mit einer Differenzliste.

Die Argumente des Prädikats sind:

Typ = Typ der Elementaraussage

SPR = Name des Sprechers - d.h. desjenigen, der die Aussage macht

IL = Liste der Insulaner, über die Aussagen gemacht werden

SEM = Aussagevariable

WORT\_L = Aussage über den Insulaner 1. Teil der Differenzliste

REST = Restliste 2. Teil der Differenzliste

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% el\_aus(1,a,[a,b],SEM,[ist,er,ein,luegner],[]). SEM = [!, a].

% el\_aus(2,a,[a,b],SEM,[er,ist,kein,ritter],[]). SEM = [!, a].

% el\_aus(3,a,[a,b],SEM,[er,ein,ritter,ist],[]). SEM = a.

% el\_aus(2,a,[a,b],SEM,[b,ist,kein,luegner],[]). SEM = b.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.4: Definieren sie das Prädikat "rest/4. Das Prädikat verarbeitet die drei Fälle, die nach der Verarbeitung einer Elementaraussage auftreten:

1. Es bleibt eine Wortliste übrig, die mit "und" beginnt.
2. Es bleibt eine Wortliste übrig, die mit "oder" beginnt.
3. Es bleibt eine leere Wortliste übrig.

Im Fall a und b wird erneut das Prädikat "el\_aus/4" aufgerufen. Das Ergebnis des Aufrufs wird entsprechend mit UND oder ODER mit dem Ergebnis des ersten Aufrufs von "el\_aus/4" zu einem logischen Ausdruck verknüpft.

Im Fall c wird nur das Ergebnis des ersten Aufrufs von "el\_aus/4" zurückgegeben.

Das Prädikat arbeitet mit einer Differenzliste.

Die Argumente des Prädikats sind:

SPR = Name des Sprechers - d.h. desjenigen, der die Aussage macht

IL = Liste der Insulaner, über die Aussagen gemacht werden

A1 = Aussagenvariable (Ergebnis des ersten Aufrufs von "el\_aus/4")

SEM = Aussagevariable bzw. Konjunktion oder Disjunktion von 2 Aussagevariablen

WORT\_L = Aussage über den Insulaner 1. Teil der Differenzliste

REST = Restliste 2. Teil der Differenzliste

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% rest(a,[a,b],a,SEM,[oder,dass,b,ein,luegner,ist],[]). SEM = [v, a, [!, b]].

% rest(a,[a,b],a,SEM,[und,dass,b,ein,luegner,ist],[]). SEM = [&, a, [!, b]].

% rest(a,[a,b],a,SEM,[],[]). SEM = a

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.5: Definieren sie das Prädikat "negation/2. Das Prädikat negiert eine unnegierte Aussagenvariable.

Die Argumente des Prädikats sind:

X = Aussagenvariable

NX = negierte Aussagenvariable

/\*-----------------------------------------------------------------------------------------------------------------------\*/

negation(X,['!',X]).

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% negation(a,NX). NX=['!',a]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.6: Definieren sie das Prädikat "sem\_min1/3. Das Prädikat konstruiert den äquivalenten logischen Ausdruck für einen Satz der Art "x sagt dass mindestens einer von ihnen ein/kein ritter/luegner ist".

Das Prädikat benutzt das Prädikat "negation/2".

Die Argumente des Prädikats sind:

IL = Liste der Insulaner, über die Aussagen gemacht werden

SEM = Disjunktion der negierten bzw. unnegierten Varaiblen aus IL

RL = {ritter,luegner}

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% sem\_min1([a,b,c],MIN,ritter). MIN = [v, a, b, c]

% sem\_min1([a,b,c],MIN,luegner). MIN = [v, [!, a], [!, b], [!, c]]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.7: Definieren sie das Prädikat "sem\_genau1/3. Das Prädikat konstruiert aus der Liste der Insulaner den logischen Ausdruck für den Sachverhalt, dass von den Insulanern genau einer ein Ritter/Lügner ist.

Das Prädikat benutzt die Systemprädikate "maplist/3", "findall/3", "select/3" und das Prädikat "negation/2".

Die Argumente des Prädikats sind:

IL = Liste der Insulaner, über die Aussagen gemacht werden

SEM = Logischer Ausdruck 1 aus n

RL = {ritter,luegner}

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% sem\_genau1([a,b,c],SEM,luegner). SEM=[v,[&,[!,a],b,c],[&,[!,b],a,c],[&,[!,c],a,b]]

% sem\_genau1([a,b,c],SEM,ritter). SEM=[v,[&,a,[!,b],[!,c]],[&,b,[!,a],[!,c]],[&,c,[!,a],[!,b]]]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.7: Definieren sie das Prädikat "sem\_keiner/3. Das Prädikat konstruiert aus der Liste der Insulaner den logischen Ausdruck für den Sachverhalt, dass keiner der Insulaner ein Ritter/Lügner ist.

Das Prädikat benutzt die Systemprädikate "maplist/3" und das Prädikat "negation/2".

Die Argumente des Prädikats sind:

IL = Liste der Insulaner, über die Aussagen gemacht werden

SEM = Logischer Ausdruck für "keiner aus n"

RL = {ritter,luegner}

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% sem\_keiner([a,b,c],KEIN,ritter). KEIN = [&, [!, a], [!, b], [!, c]]

% sem\_keiner([a,b,c],KEIN,luegner). KEIN = [&, a, b, c].

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.8: Definieren sie das Prädikat "sem\_alle/3. Das Prädikat konstruiert aus der Liste der Insulaner den logischen Ausdruck für den Sachverhalt, dass jeder der Insulaner ein Ritter/Lügner ist.

Das Prädikat benutzt die Systemprädikate "maplist/3" und das Prädikat "negation/2".

Die Argumente des Prädikats sind:

IL = Liste der Insulaner, über die Aussagen gemacht werden

SEM = Logischer Ausdruck für "alle aus n"

RL = {ritter,luegner}

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% sem\_alle([a,b,c],ALLE,ritter). ALLE = [&, a, b, c]

% sem\_alle([a,b,c],ALLE,luegner). ALLE = [&, [!, a], [!, b], [!, c]].

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.9: Definieren sie das Prädikat "aussage/5". Das Prädikat bestimmt die Art der logischen Aussage und davon abhängig wie der äquivalente logische Ausdruck konstruiert wird. Zur Konstruktion werden verschiedene weitere Prädikate benutzt: el\_aus/4, rest/4, sem\_min1/3, sem\_genau1/3, sem\_keiner/3, sem\_alle/3.

Das Prädikat arbeitet mit einer Differenzliste.

Die Argumente des Prädikats sind:

SPR = Name des Sprechers - d.h. desjenigen, der die Aussage macht

IL = Liste der Insulaner, über die Aussagen gemacht werden

SEM = Semantik (Logik) der Einzelaussage

WORT\_L = Satz mit Aussage über Insulaner 1. Teil der Differenzliste

REST = Restliste 2. Teil der Differenzliste

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% aussage(a,[a,b],SEM,[er,ein,ritter,ist],[]). SEM = a.

% aussage(a,[a,b],SEM,[er,ein,ritter,ist,und,dass,b,ein,luegner,ist],[]). SEM = [&, a, [!, b]]

% aussage(a,[a,b],SEM,[b,ein,ritter,ist,oder,dass,er,ein,luegner,ist],[]). SEM = [v, b, [!, a]]

% aussage(a,[a,b],SEM,[mindestens,einer,von,ihnen,ein,ritter,ist],[]). SEM = [v, a, b]

% aussage(a,[a,b],SEM,[genau,einer,von,ihnen,ein,luegner,ist],[]). SEM=[v,[&,[!,a],b],[&,[!,b],a]]

% aussage(a,[a,b],SEM,[keiner,von,ihnen,ein,ritter,ist],[]). SEM = [&, [!, a], [!, b]]

% aussage(a,[a,b],SEM,[jeder,von,ihnen,ein,ritter,ist],[]). SEM = [&, a, b]

% aussage(a,[a,b],SEM,[wenn,er,ein,ritter,ist,dann,ist,b,ein,luegner],[]). SEM = [->, a, [!, b]]

% aussage(a,[a,b,c],SEM,[entweder,b,ein,ritter,ist,oder,c,ist,ein,luegner],[]). SEM=[<-/->,b,[!,c]]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.10: Definieren sie das Prädikat "aussage\_or\_nil/5". Das Prädikat differenziert zwischen einer leeren und einer echten Aussage und verknüpft den Namen des Sprecher, der eine Aussage macht, mit dieser Aussage.

Das Prädikat benutzt das Prädikat "aussage/5".

Das Prädikat arbeitet mit einer Differenzliste.

Die Argumente des Prädikats sind:

SPR = Name des Sprechers - d.h. desjenigen, der die Aussage macht

IL = Liste der Insulaner, über die Aussagen gemacht werden

SEM = Semantik (Logik) der Einzelaussage

WORT\_L = Satz mit Aussage über Insulaner 1. Teil der Differenzliste

REST = Restliste 2. Teil der Differenzliste

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% aussage\_or\_nil(a,[a,b],SEM,[etwas,unverstaendliches],[]). SEM = 1

% aussage\_or\_nil(a,[a,b],SEM,[dass,er,ein,ritter,ist],[]). SEM = [<->, a, a]

% aussage\_or\_nil(a,[a,b],SEM,[dass,mindestens,einer,von,ihnen,ein,ritter,ist],[]). SEM=[<-> a,[v,a,b]]

% aussage\_or\_nil(a,[a,b],SEM,[dass,jeder,von,ihnen,ein,ritter,ist],[]). SEM = [<-> a,[&, a, b]]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.11: Definieren sie das Prädikat "satz/4". Das Prädikat bestimmt den einer Einzelaussage äquivalenten logischen Ausdruck und erzeugt im Fehlerfall eine Fehlermeldung.

Das Prädikat benutzt das Prädikat "aussage\_or\_nil/5".

Das Prädikat arbeitet mit einer Differenzliste.

Die Argumente des Prädikats sind:

IL = Liste der Insulaner

SEM = Semantik (Logik) der Einzelaussage

SATZ = Satz mit Aussage über Insulaner 1. Teil der Differenzliste

REST = Restliste 2. Teil der Differenzliste

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% satz([a],SA,[a,sagt,dass,er,ein,ritter,ist],[]). SA = [<->,a,a]

% satz([a,b],SA,[a,sagt,dass,er,ein,ritter,ist],[]). SA = [<->,a,a]

% satz([a,b],SA,[a,sagt,dass,b,ein,ritter,ist],[]). SA = [<->,a,b]

% satz([a,b],SA,[a,sagt,dass,er,ein,ritter,ist,und,b,ein,luegner,ist],[]). SA = [<->, a, [&, a, [!, b]]].

% satz([a],SA,[c,sagt,dass,er,ein,ritter,ist],[]). SA = fehler\_im\_satzteil1.

% satz([a],SA,[a,sagt,dass,er,einen,laptop,hat],[]).SA = fehler\_im\_satzteil2.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.12: Definieren sie das Prädikat "saetze/3". Das Prädikat wendet rekursiv das Prädikat "satz/4" auf die einzelnen Sätze der Aussagen der Aufgaben an und fasst deren Einzelsemantiken konjunktiv zusammen.

Die Argumente des Prädikats sind:

IL = Liste der Insulaner

AUSGl = Liste der Einzelaussagen

SEML = Liste der Semantik (Logik) der Einzelaussagen

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% saetze([a,b],[[a,sagt,dass,er,ein,ritter,ist],[b,sagt,dass,a,ein,luegner,ist]],SEM).

% SEM = [[<->, a, a], [<->, b, [!, a]]]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 1.12: Definieren sie das Prädikat "textmining/3". Das Prädikat holt den Aufgabentext entsprechend dem gegebenen Index aus dem Aufgabenfile, bestimmt die Menge der zu identifizierenden Insulaner und bildet mit der Konjunktion der Einzelaussagen die Gesamtsematik des Aufgabetextes in Form eines aussagenlogischen Ausdrucks.

Das Prädikat benutzt das Prädikat "saetze/3".

Die Argumente des Prädikats sind:

I = Index der Aufgabe

IL = Liste der Insulaner

SEM = extrahierte Semantik der Aufgabe

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% textmining(1,IL,SEM). SEM = [&,[<->,a,a],[<->,b,[!,a]]]

% textmining(2,IL,SEM). SEM = [&,1,[<->,b,[&,[!,a],[!,b]]]]

% textmining(3,IL,SEM). SEM = [&,[<->,a,[&,a,b]],[<->,b,[&,[!,a],[!,b]]]]

% textmining(4,IL,SEM). SEM = [&,[<->,a,[->,a,b]],[<->,b,[&,[!,a],[!,b]]]]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 2: Bestimmung der Wurzeln eines logischen Ausdrucks

/\*-----------------------------------------------------------------------------------------------------------------------\*/

Aufgabe 2.1: Definieren Sie eine Interpreter für geschachtelte aussagenlogische Ausdrücke in Präfixnotation.

Das Prädikat "int/2" bestimmt den logischen Wert von geschachtelten aussagenlogische Ausdrücke in Präfixnotation in Form einer geschachtelten Liste. Der Kopf der Listen enthält den logischen Operator, der auf den Rest der Liste, der die Argumente enthält angewendet wird. Die Anwendung erfolgt dann durch für jede logische Funktion spezielle Prädikate:

und/2, oder/2, neg/2, imp/3, equ/3, ant/3

Der aussagenlogische Ausdruck enthält nur konstante Werte (0,1).

Die Argumente des Prädikats sind:

AL = aussagenlogische Ausdruck in Form einer verschachtelte Liste

Y = Wert des aussagenlogischen Ausdrucks

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% int([&,1,1,1,1],Y). Y=1

% int([&,1,1,1,0],Y). Y=0

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% int([v,0,0,0,0],Y). Y=0

% int([v,0,0,0,1],Y). Y=1

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% int([v,[&,1,1,1,1],[&,1,1,1,0]],Y). Y=1

%int([v,[->,1,1],[&,1,1,1,0]],Y). Y=1

%int([v,[->,0,1],[->,[&,1,1,1,0],1]],Y). Y=1

%int([&,[->,1,0],[->,[v,1,1,1,0],1]],Y). Y=0

%int([v,[!,0],0,0,0],Y). Y=1

%int([v,[!,[&,1,1,1,0]],[&,1,1,1,0]],Y). Y=1

%int([!,[&,1,1,1,0]],Y). Y=1

/\*-----------------------------------------------------------------------------------------------------------------------\*/

%int([<->,0,0],Y). Y=1

%int([<->,1,1],Y). Y=1

%int([&,[0,<->,0],[1,<->,[!,0]]],Y).

%int([<->,[&,1,1,1,0],[v,[!,0],0,0,0]],Y). Y=0

%int([<->,[&,1,1,1,0],[<-/->,[!,0],0]],Y). Y=0

%int([<-/->,[!,0],0],Y). Y=1

%int([&,[<->,0,[&,0,1]],[<->,1,[&,[!,0],[!,1]]]],Y). Y=0.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 2.2: Definieren Sie die Prädikate für die einzelnen logischen Funktionen: und/2, oder/2, neg/2, imp/3, equ/3, ant/3.

Beachten Sie dabei, dass deren Argumente entweder Konstanten oder Unterausdrücke sind.

Die Argumente der Prädikate sind:

und(<liste der Eingabewerte>,<Ausgabewert>)

oder(<liste der Eingabewerte>,<Ausgabewert>)

neg(<Eingabewert>,<Ausgabewert>)

imp(<Eingabewert1>,<Eingabewert2>,<Ausgabewert>)

equ(<Eingabewert1>,<Eingabewert2>,<Ausgabewert>)

ant(<Eingabewert1>,<Eingabewert2>,<Ausgabewert>)

Die Prädikate benutzen das Prädikat "int/2" beim Auftreten von Unterausdrücken.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

% und([1,[v,0,1],[<-/->,0,1]],Y). Y=1

% und([1,1,1,1],Y). Y=1

% und([1,1,1,0],Y). Y=0

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% oder([0,0,0,0],Y). Y=0

% oder([0,0,0,1],Y). Y=1

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% und([[->,1,0],[->,[v,1,1,1,0],1]],Y). Y=0

% oder([[&,1,1,1,1],[&,1,1,1,0]],Y). Y=1

% oder([[->,1,1],[&,1,1,1,0]],Y). Y=1

% oder([[->,0,1],[->,[&,1,1,1,0],1]],Y). Y=1

% oder([[!,0],0,0,0],Y). Y=1

% oder([[!,[&,1,1,1,0]],[&,1,1,1,0]],Y). Y=1

% neg([[&,1,1,1,0]],Y). Y=1

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% equ(0,0,Y). Y=1

% equ(1,1,Y). Y=1

% equ([&,1,1,1,0],[v,[!,0],0,0,0],Y). Y=0

% equ([&,1,1,1,0],[<-/->,[!,0],0],Y). Y=0

% ant([!,0],0,Y). Y=1

% und([[<->,0,[&,0,1]],[<->,1,[&,[!,0],[!,1]]]],Y). Y=0.

% und([[<->,0,0],[<->,1,[!,0]]],Y). Y=1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 2.3: Definieren Sie das Prädikat "werte/2", das die Binärvektoren für die Wertebelegungen der Variablen des logischen Ausdrucks erzeugt.

Das Prädikat benutzt die Systemprädikate "length/2", "maplist/2" und das Prädikat "bin/2": bin(0). bin(1).

Die Argumente des Prädikats sind:

VL = Liste mit Variablennamen

WL = mit binären Werten gefüllte Werteliste

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% werte([a,b,c],WL). -> Alle 8 Wertevektoren

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

bin(0). bin(1).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 2.4: Definieren Sie das Prädikat "einsetzen/3" zum Einsetzen von Konstantenwerten in symbolischen logischen Ausrücken entsprechend den gegebenen Zuordnungen von Variablen und Werten.

Das Prädikat benutzt die Systemprädikate "member/2", "atom/1","number/1" und "compound/1"

Die Argumente des Prädikats sind:

VEXPR: Logischer Ausdruck mit Variablen (und Konstanten)

AL: Assoziationsliste mit (Variablen,Werte)-Paaren

WEXPR: Logischer Ausdruck mit Konstanten

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% einsetzen([v,x,y,z],[(x,1),(y,1),(z,0)],WEXPR). WEXPR = [v, 1, 1, 0]

% einsetzen([&,1,y,z],[(x,0),(y,1),(z,0)],WEXPR). WEXPR = [&, 1, 1, 0]

% einsetzen(['<->',0,x],[(x,1)],WEXPR). WEXPR = [<->, 0, 1]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 2.5: Definieren Sie das Prädikat "subst/4" zum Substituieren von Variablen durch Konstantenwerte in symbolischen logischen Ausrücken entsprechend den Variablen- und Konstantenlisten.

Die Argumente des Prädikats sind:

VEXPR: Logischer Ausdruck mit Variablen (und Konstanten)

VL: Liste der zu ersetzenden Variablen

VL: Liste der die Variablen ersetzenden Werte

WEXPR: Logischer Ausdruck mit Konstanten

Das Prädikat benutzt die Systemprädikate "maplist/3", das Prädikat "einsetzen/3" und das Prädikat bind/3. => bind(V,W,(V,W)).

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% subst([v,x,y,z],[x,y,z],[1,1,0],WEXPR). WEXPR = [v, 1, 1, 0]

% subst([&,1,y,z],[x,y,z],[0,1,0],WEXPR). WEXPR = [v, 1, 1, 0]

% subst(['<->',x,x],[x],[1],WEXPR). WEXPR = [<->, 1, 1]

% subst([&,[a,<->,a],[b,<->,[!,a]]],[a,b],[0,1],WEXPR). [&,[0,<->,0],[1,<->,[!,0]]]

% subst(['<->', ['&',a,b,c,d],['v',[!,e],f,g,h]],[a,b,c,d,e,f,g,h],[1,1,1,0,0,0,0,0],WEXPR).

% WEXPR = [<->, [&, 1, 1, 1, 0], [v, [!, 0], 0, 0, 0]]

% subst([&,[<->,a,[&,a,b]],[<->,b,[&,[!,a],[!,b]]]],[a,b],[0,1],WEXPR).

% WEXPR = [&,[<->,0,[&,0,1]],[<->,1,[&,[!,0],[!,1]]]]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 2.6: Definieren Sie das Prädikat "wurzel/2", das für einen gegebenen logischen Ausdruck eine seiner Wurzeln bestimmt.

Die Argumente des Prädikats sind:

LEXPR = logischer Ausdruck

VL = Variablenvektor

WL = Wurzel

Das Prädikat benutzt die Prädikate "werte/3", "subst/4" und "int/2".

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% wurzel([&,1,[<->,b,[&,[!,a],[!,b]]]],[a,b],W). W = [1, 0]

% wurzel([&,[<->,a,[&,a,b]],[<->,b,[&,[!,a],[!,b]]]],[a,b],W).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 2.7: Definieren Sie das Prädikat "log\_analyse/3", das für einen gegebenen logischen Ausdruck alle seine Wurzeln bestimmt.

Die Argumente des Prädikats sind:

LEXPR = logischer Ausdruck

VL = Variablenvektor

WLN = Wurzelliste

Das Prädikat benutzt das Systemprädikate "findall/3" und das Prädikat "wurzel/3".

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% log\_analyse([v,a,b,c],[a,b,c],WL).

% WL=[[0,0,1],[0,1,0],[0,1,1],[1,0,0],[1,0,1],[1,1,0],[1,1,1]]

% log\_analyse(['<->,a,['&',[!,a],[!,b]]],[a,b],WL). WL = [[0, 1]].

% log\_analyse([&,[<->,a,a],[<->,b,[!,a]]],[a,b],WL). WL = [[0, 1], [1, 0]]

% log\_analyse([&,1,[<->,b,[&,[!,a],[!,b]]]],[a,b],WL).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 3: Interpretation der Wurzeln eines logischen Ausdrucks

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 3.1: Definieren Sie das Prädikat "spalte/3", das eine Matrix - in unserem Fall die Wurzelmatrix in eine Spalte und die Restmatrix zerlegt. Das ist Voraussetzung für das Trans- ponieren der Wurzelmatrix.

Die Argumente des Prädikats sind:

MATRIX = Wurzelmatrix

SPALTE = Spalte der Wurzelmatrix

RESTMATRIX = Restmatrix (nach Spaltenabtrennung) der Wurzelmatrix

/\*-----------------------------------------------------------------------------------------------------------------------\*/

spalte([],[],[]). % Abbruch für quadratische Matrizen

spalte([[K]|RL],[K|R],[]):- spalte(RL,R,[]). % Nichtquadratische Matrizen

spalte([[ZK|ZR]|MR],[ZK|SR],[ZR|MR1]):- spalte(MR,SR,MR1).

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% spalte([[a11,a12,a13],[a21,a22,a23],[a31,a32,a33]],SP,RM). ->

% SP = [a11, a21, a31]

% RM = [[a12, a13], [a22, a23], [a32, a33]]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 3.2: Definieren Sie das Prädikat "transpon/2", das eine Matrix transponiert.

Die Argumente des Prädikats sind:

MATRIX = Wurzelmatrix

TRANSMATRIX = transponierte Wurzelmatrix

Das Prädikat benutzt das Prädikat "spalte/3"

/\*-----------------------------------------------------------------------------------------------------------------------\*/

transpon([],[]).

transpon([[ZK|ZR]|MR],[[ZK|SP]|IMR]):- spalte(MR,SP,MRR), % spalte(i,i,o)

transpon(MRR,IMR1),

spalte(IMR,ZR,IMR1). % spalte(o,i,i)

/\*-----------------------------------------------------------------------------------------------------------------------\*/

%transpon([[a,b,c]],TM). TM = [[a], [b], [c]]

%transpon([[a11,a12],[a21,a22]],TM). TM = [[a11, a21], [a12, a22]].

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 3.3: Definieren Sie das Prädikat "spalten\_gleichheit/2", das für eine Matrix feststellt, ob die Elemente ihrer Spalten gleich sind. Wenn sie gleich sind, erhält die Ergebnisliste an dieser Stelle den Wert des Elements, das in der Spalte vorkommt. Sind unterschiedliche Elemente in der betreffenden Spalte vorhanden erhält die Ergebnisliste an dieser Stelle den Wert 2.

Die Argumente des Prädikats sind:

TRANSMATRIX = transponierte Wurzelmatrix

EL = Ergebnisliste

Das Prädikat benutzt das Systemprädikat "forall/2"

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% spalten\_gleichheit([[0,0,1],[1,0,1],[1,1,1],[0,0,0]],EL). EL = [2, 2, 1, 0]

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 3.4: Definieren Sie das Prädikat "ausgabe/3", das das Ergebnis des Test auf Gleichheit der Elemente der Spalten der transponierten Wurzelmatrix und damit die Variablenwerte interpretiert. Das Ergebnis der Interpretation wird in der Ergebnisliste VWL zurückgegeben und als Seiteneffekt textuell ausgegeben.

Die Argumente des Prädikats sind:

GL = Liste der Werte des Gleichheitstests der Spaltenwerte

VL = Liste der Variablen (Insulanernamen)

VWL = Ergebnisliste

Das Prädikat benutzt das Systemprädikat "format/2"

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% ausgabe([0,1,2],[a,b,c],VWL).

%Der Insulaner a ist ein Lügner

%Der Insulaner b ist ein Ritter

%Es kann nicht festgestellt werden ob der Insulaner c ein Ritter oder ein Lügner ist

% VWL = [ (a, 0), (b, 1), (c, 2)] ;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 3.5: Definieren Sie das Prädikat "auswertung/3", das alle Prädikate für die Interpretation der Wurzeln eines logischen Ausdrucks zusammenfasst.

Die Argumente des Prädikats sind:

VL = Liste der Variablen (Insulanernamen)

WLN = Wuzelliste

VWL = Ergebnisliste

Das Prädikat benutzt die Prädikate "transpon/2", "spalten\_gleichheit/2" und "ausgabe/3".

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% auswertung([a,b,c],[[0,1,1],[1,0,1]],VWL).

% Es kann nicht festgestellt werden ob der Insulaner a ein Ritter oder ein Lügner ist

% Es kann nicht festgestellt werden ob der Insulaner b ein Ritter oder ein Lügner ist

% Der Insulaner c ist ein Ritter

% VWL = [ (a, 2), (b, 2), (c, 1)]

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% auswertung([a,b,c],[],VWL). VWL = []

% Die Aussagen sind widersprüchlich - keine Schlussfolgerung moeglich

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Aufgabe 4: Definieren Sie das Prädikat "solver/2", das alle Prädikate zur Lösung einer logischen Aufgabe zusammenfasst.

Die Argumente des Prädikats sind:

I = Index der Aufgabe

VWL = Ergebnisliste

Das Prädikat benutzt die Prädikate "textmining/3", "log\_analyse/3" und "auswertung/3".

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% solver(4,VWL). VWL = []

% Die Aussagen sind widersprüchlich - keine Schlussfolgerung moeglich

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% solver(5,VWL). VWL = [ (a, 0), (b, 0)] ;

% Der Insulaner a ist ein Lügner

% Der Insulaner b ist ein Lügner

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Test mit allen 22 Aufgaben

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

test\_int(I,N):- I>N,writeln('TestEnde').

test\_int(I,N):- I=<N,solver(I),format(': Fall ~d ist ok.~n',[I]),I1 is I+1,!,

test\_int(I1,N).

test\_int(I,N):- I=<N,not(solver(I)),format('Fehler im Fall ~d~n',[I]),

I1 is I+1, test\_int(I1,N).

/\*-----------------------------------------------------------------------------------------------------------------------\*/

% test\_int(1,22).

%Es kann nicht festgestellt werden ob der Insulaner a ein Ritter oder ein Lügner ist

%Es kann nicht festgestellt werden ob der Insulaner b ein Ritter oder ein Lügner ist: Fall 1 ist ok.

%Der Insulaner a ist ein Ritter

%Der Insulaner b ist ein Lügner: Fall 2 ist ok.

% Die Aussagen sind widersprüchlich - keine Schlussfolgerung moeglich

% : Fall 3 ist ok.

% Die Aussagen sind widersprüchlich - keine Schlussfolgerung moeglich

% : Fall 4 ist ok.

%

%Der Insulaner a ist ein Lügner

%Der Insulaner b ist ein Lügner: Fall 5 ist ok.

% Die Aussagen sind widersprüchlich - keine Schlussfolgerung moeglich

% : Fall 6 ist ok.

%

%Es kann nicht festgestellt werden ob der Insulaner a ein Ritter oder ein Lügner ist

%Es kann nicht festgestellt werden ob der Insulaner b ein Ritter oder ein Lügner ist: Fall 7 ist ok.

% Die Aussagen sind widersprüchlich - keine Schlussfolgerung moeglich

% : Fall 8 ist ok.

%

%Der Insulaner a ist ein Lügner

%Der Insulaner b ist ein Lügner

%Der Insulaner c ist ein Ritter: Fall 9 ist ok.

%Der Insulaner a ist ein Ritter

%Der Insulaner b ist ein Lügner

%Es kann nicht festgestellt werden ob der Insulaner c ein Ritter oder ein Lügner ist: Fall 10 ist ok.

%Es kann nicht festgestellt werden ob der Insulaner a ein Ritter oder ein Lügner ist

%Es kann nicht festgestellt werden ob der Insulaner b ein Ritter oder ein Lügner ist

%Der Insulaner c ist ein Ritter: Fall 11 ist ok.

%Es kann nicht festgestellt werden ob der Insulaner a ein Ritter oder ein Lügner ist

%Es kann nicht festgestellt werden ob der Insulaner b ein Ritter oder ein Lügner ist

%Es kann nicht festgestellt werden ob der Insulaner c ein Ritter oder ein Lügner ist: Fall 12 ist ok.

% Die Aussagen sind widersprüchlich - keine Schlussfolgerung moeglich

% : Fall 13 ist ok.

%

%Der Insulaner a ist ein Ritter

%Der Insulaner b ist ein Lügner: Fall 14 ist ok.

%Es kann nicht festgestellt werden ob der Insulaner a ein Ritter oder ein Lügner ist

%Der Insulaner b ist ein Ritter: Fall 15 ist ok.

%Der Insulaner a ist ein Lügner

%Der Insulaner b ist ein Ritter

%Es kann nicht festgestellt werden ob der Insulaner c ein Ritter oder ein Lügner ist: Fall 16 ist ok.

%Der Insulaner a ist ein Lügner

%Der Insulaner b ist ein Ritter: Fall 17 ist ok.

%Der Insulaner a ist ein Lügner

%Der Insulaner b ist ein Lügner: Fall 18 ist ok.

%Der Insulaner a ist ein Ritter

%Der Insulaner b ist ein Ritter: Fall 19 ist ok.

%Der Insulaner a ist ein Ritter

%Der Insulaner b ist ein Lügner: Fall 20 ist ok.

%Der Insulaner a ist ein Ritter

%Der Insulaner b ist ein Ritter

%Der Insulaner c ist ein Ritter: Fall 21 ist ok.

%Der Insulaner a ist ein Lügner

%Der Insulaner b ist ein Ritter

%Der Insulaner c ist ein Lügner: Fall 22 ist ok.

% TestEnde

% TestEnde