

Aufgabe 1

a)

Zustand: Menge von Stapeln mit mindestens 1 und höchstens 3 Elementen, jeder Stapel ist als Stack von Kisten organisiert, wobei jede Kiste durch ihre Nummer repräsentiert wird (vorne = oben)

Pop/1 sei eine Funktion, die einen um 1 Element reduzierten Stack zurückliefert

Push/2 liefert einen um das angegebene Element erweiterten Stack zurück

- und + seien Mengendifferenz und Aufnahmen eines Elements in eine Menge

Startzustand: $S_0 = \{ [5,3,4,1,2,6] \}$

Zielzustand: $S_Z = \{ [1,2,3,4,5,6] \}$

Aktionen:

Ablegen_1(S): $Z_n \rightarrow Z_{n+1}$ // 1 Element auf den Boden legen

$S \in Z_n, x = \text{top}(S)$

$S_1 = \text{pop}(S)$

$Z_{n+1} = Z_n - S + S_1 + [x]$

Ablegen_2(S): $Z_n \rightarrow Z_{n+1}$ // 2 Elemente auf den Boden legen

$S \in Z_n, x = \text{top}(S), y = \text{top}(\text{pop}(S))$

$S_1 = \text{pop}(\text{pop}(S))$

$Z_{n+1} = Z_n - S + S_1 + [x,y]$

Umlegen_1(Sa,Sb): $Z_n \rightarrow Z_{n+1}$ // 1 Element auf einen anderen Stapel legen

$S_a \in Z_n, S_b \in Z_n, S_a \neq S_b, x = \text{top}(S_a)$

$S_{a1} = \text{pop}(S_a), S_{b1} = \text{push}(x, S_b)$

$Z_{n+1} = Z_n - S_a + S_{a1} - S_b + S_{b1}$

Umlegen_2(Sa,Sb): $Z_n \rightarrow Z_{n+1}$ // 2 Elemente auf einen anderen Stapel legen

$S_a \in Z_n, S_b \in Z_n, S_a \neq S_b, x = \text{top}(S_a), y = \text{top}(\text{pop}(S_a))$

$S_{a1} = \text{pop}(\text{pop}(S_a)), S_{b1} = \text{push}(x, \text{push}(y, S_b))$

$Z_{n+1} = Z_n - S_a + S_{a1} - S_b + S_{b1}$

b)

$S_0 \rightarrow \{ [3, 4, 1, 2, 6], [5] \}$
 $S_1 \rightarrow \{ [4, 1, 2, 6], [3, 5] \}$

(bei anderer als Mengenrepräsentation existieren 4 Folgezustände, abhängig vom Zielort des Teils oder 2er-Stapels)

c)

Die sortierte Liste enthält Pfade, keine Knoten. $f=g+h$ berechnet sich aus der Heuristik plus 1 für jede durchgeführte Aktion (wegen uniformer Kosten).

Iteration sortierte Liste

0 $\langle \{ [2,3,1] \} \rangle -1$

1 $\langle \{ [2,3,1] \}, \{ [2,3], [1] \} \rangle -2, \langle \{ [2,3,1] \}, \{ [3,1], [2] \} \rangle -3$

2 $\langle \{ [2,3,1] \}, \{ [2,3], [1] \}, \{ [1,2,3] \} \rangle -2, \langle \{ [2,3,1] \}, \{ [3,1], [2] \} \rangle -3, \langle \{ [2,3,1] \}, \{ [2,3], [1] \}, \{ [1], [2], [3] \} \rangle -4$

in 2. wird der Pfad $\langle \{ [2,3,1] \}, \{ [2,3], [1] \}, \{ [2,3,1] \} \rangle -4$ wird entfernt, weil er einen Zyklus darstellt.

Aufgabe 2

a)

Durch das Entfernen inkonsistenter Werte sind die Variablen $M(1,1)$ und $M(1,2)$ bereits belegt.

1	3	2	2	2
2	1	3	1	1
	3	4	3	4
	1	2	1	2
4				3
3	1	2	1	2
	4	4	4	4

b)

Minimum Remaining Values (aller noch nicht belegten Variablen) in Kombination mit Most Constrained Variable: MRV sind

$\{ M(2,2), M(3,2), M(4,3), M(4,4) \}$ mit jeweils 2-stelligen Wertebereichen. MCV als Tie-Breaker bringt keine eindeutige Entscheidung: MCV sind $\{ M(2,2), M(3,2), M(4,3), M(4,4) \}$ mit jeweils 4 Constraints.

(Alternative: MCV sind die 8 Variablen $M(2,1), \dots, M(3,4)$ mit jeweils 5 Constraints zu noch nicht belegten Variablen. (Die übrigen Variablen haben eine Grad-Heuristik von 4).

c)

Belege $M(2,2)$ mit 1. Forward Checking führt in die rechts abgebildete Situation. $M(3,2)$ ist belegt, $\{ M(2,1), M(2,3), M(2,4) \}$ haben jeweils um den Wert „1“ reduzierte Wertebereiche. Die übrigen Domains werden nicht verändert.

1	3	2	2	2
2	3	4	1	1
	3	4	3	4
	1	2	1	2
4				3
3	2	1	2	1
	4	4	4	4

Aufgabe 3

a/1 ~ Aufgabenblatt

u/1 ~ unlösbar

s/1 ~ Student

b/2 ~ bekommen

sch/1 ~ Schein

l/2 ~ lösen

a/2 ~ abgeben

k/2 ~ Kommilitone

v/3 ~ verraten

L ~ diese Lösung

l/1 ~ Lösung

a)

$\exists x, y a(x) \wedge a(y) \wedge x \neq y \wedge (\forall z a(z) \rightarrow z=x \vee z=y)$

b)

$\neg \exists x a(x) \wedge u(x)$ oder $\forall x a(x) \rightarrow \neg u(x)$

c)

$(\forall x s(x) \rightarrow \forall y a(y) \rightarrow I(x,y)) \rightarrow \exists z sch(z) \wedge b(x,z)$
 $(\forall x,y s(x) \wedge a(y) \rightarrow I(x,y)) \rightarrow \exists z sch(z) \wedge b(x,z)$ (nicht ganz korrekt)
 Wenn „genau jeder Student“ bzw. „nur Studenten, die“ gemeint ist, dann Äquivalenz verwenden.

d)

$\forall x s(x) \rightarrow \exists y,z a(y) \wedge a(z) \wedge y \neq z \wedge a(s,y) \wedge a(s,z)$

e)

$\neg \exists x s(x) \wedge \forall y k(x,y) \rightarrow v(x,y,L) \wedge I(L)$

Aufgabe 4

$K1 = \{ \neg P(x,1,f(x)), \neg Q(x,x), \neg R(x,x) \}$
 $K2 = \{ P(y,z,f(z)), Q(y,f(y)), R(1,y) \}$

$\neg R(1,1)$ $\neg R(2,1)$ war Druckfehler

1. P gegen $\neg P$ resolvieren:
 $\{y/x, z/1, x/1\} \rightarrow \{ \neg Q(1,1), \neg R(2,1), Q(1,f(1)), R(1,1) \}$
2. Q gegen $\neg Q$ resolvieren: geht nicht, da nicht unifizierbar (x und g(x) nicht unifizierbar)
3. R gegen $\neg R$ resolvieren:
 $\{ x/2, y/2 \} \rightarrow \{ \neg P(2,1,f(2)), \neg Q(2,2), P(2,z,f(z)), Q(2,f(2)) \}$

Aufgabe 5

a)

1. Konjunktive Normalform der Wissensbasis bilden, 2. Anfrage negieren, 3. Resolventen bilden, bis leere Resolvente ableitbar oder kein neues Wissen generierbar

b)

1. KNF bilden

1. $\{ \neg G(x), \neg S(x), GU(x) \} \leftarrow \forall x G(x) \wedge S(x) \rightarrow GU(x)$
2. $\{ \neg M(x), S(x) \} \leftarrow \forall x M(x) \rightarrow S(x)$
3. $\{ \neg R(x), M(x) \} \leftarrow \forall x R(x) \rightarrow M(x)$
4. $\{ \neg R(x), \neg GF(x,y), R(y) \} \leftarrow \forall x \forall y R(x) \wedge GF(x,y) \rightarrow R(y)$
5. $\{ G(A1) \} \leftarrow G(A1)$
6. $\{ GF(A1) \} \leftarrow GF(S1, A1)$
7. $\{ R(S1) \} \leftarrow R(S1)$

2. Anfrage negieren und in KNF überführen

8. $\{ \neg GU(A1) \} \leftarrow \neg GU(A1)$

3. Resolventen bilden

- 8&1 $\{x/A1\} \rightarrow \{ \neg G(A1), \neg S(A1) \} \quad (9)$
- 9&5 $\{ \} \rightarrow \{ \neg S(A1) \} \quad (10)$
- 10&2 $\{x/A1\} \rightarrow \{ \neg M(A1) \} \quad (11)$
- 11&3 $\{x/A1\} \rightarrow \{ \neg R(A1) \} \quad (12)$
- 12&4 $\{x/z, y/A1\} \rightarrow \{ \neg GF(z, A1), \neg R(z) \} \quad (13)$
- 13&6 $\{z/S1\} \rightarrow \neg R(S1) \quad (14)$

14&7 $\{ \} \rightarrow \{ \}$

Aufgabe 6 - Planung

a)

Wir nummerieren die Bausteine folgendermaßen:



Prädikate:

- farbe/1 : Typprädikat
- größe/1 : Typprädikat
- etage/2 : Baustein im Haus mit Etage und Bausteinnummer
- höhe/1 : aktuelle Höhe des zu bauenden Hauses
- block/3 : Block mit Nummer, Größe und Farbe.
- dach/2 : Dach mit Nummer und Größe
- n/2 : Nachfolgerprädikat auf natürlichen Zahlen
- ge/2 : größer-gleich-Relation (falls mehr als nur 2 Größen vorhanden sind)
- v/1 : bezeichnet ein verfügbares Bauteil (eines, das noch nicht verbaut ist)

Bemerkungen: statt v/1 man könnte alternativ auch einen Ort "Kiste" als Konstante definieren und beispielsweise in/2 verwenden. Die Blocknummer ist wichtig, weil mehrere gleiche Blöcke zu unterscheiden sind – dasselbe gilt für Dächer. Dächer haben keine Farbe, weil diese beim Häuserbauen keine Rolle spielt.

Konstanten:

- S,G,W : Farben
- 1,2 : Größen (klein, groß)
- 1,...,9 : IDs für Bausteine

Bemerkung: 1 und 2 sind sowohl Größen, als auch IDs. Diese „Überladung“ von Konstanten bringt hier keine Probleme. Bei dieser Modellierung von Größen mittels Zahlen in Verbindung mit der größer-gleich-Relation kann man auch weitere Blockgrößen einführen.

b)

S0: { block(1), block(2), ..., block(6), dach(7), dach(8), dach(9),
 größe(1,1), größe(2,1), ..., größe(6,2), größe(7,1), größe(8,1), größe(9,2),
 farbe(1,S), farbe(2,S), ..., farbe(6,W),
 farbe(S), farbe(G), farbe(W), größe(1), größe(2), n(0,1), n(1,2), n(2,3),
 höhe(0),
 ge(1,1), ge(2,1), ge(2,2)
 v(1),v(2),...,v(9)
 }

c)

SZ: {etage(1,x),etage(2,y),etage(3,z),dach(z)}
 alternativ: {höhe(3),etage(3,x),dach(x)}

d)

ACT stack0(x) // ein beliebiges Bauteil als 1. Etage verbauen
 PRE höhe(0), v(x)

1. Teilklausur GKI, WS 2010/11, Musterlösung

```
ADD höhe(1), etage(1,x),  
DEL v(x), höhe(0)
```

```
ACT stackb( x )           // einen verfügbaren Block auf einen anderen Block legen  
  PRE höhe(h), etage(h,b), v(x), gröÙe(b,g), farbe(b,f), block(x), gröÙe(x,gx), farbe(x,f),  
    ge(g,gx), n(h,h1)  
  ADD höhe(h1), etage(h1,x)  
  DEL höhe(h), v(x)
```

Bemerkung: die Überprüfung der Farbe stellt sicher, dass b kein Dach sein kann.

```
ACT stackd( x )           // ein verfügbares, passendes Dach auf die aktuelle Etage legen  
  PRE höhe(h), etage(h,b), block(b), v(x), dach(x), gröÙe(x,g), gröÙe(b,g), n(h,h1)  
  ADD höhe(h1), etage(h1,x)  
  DEL höhe(h), v(x)
```

e)

Lösungsplan: stack0(1),stackb(2),stackd(7)

f)

Relevant und konsistent sind alle Aktionen, die eines der Teilziele erfüllen. dach/1 wird von keiner Aktion generiert (ist bereits erfüllt, geeignete Variablenbelegung vorausgesetzt). Gesucht sind also alle Aktionen, die eine etage(1,x) in ihrer ADD haben: Dies betrifft sowohl stack0 als auch stackb und stackd mit allen möglichen Belegungen für die Variable x (also alle Individuenkonstanten (insgesamt 12 Konstanten) --> $12 \cdot 3 = 36$ Aktionen)