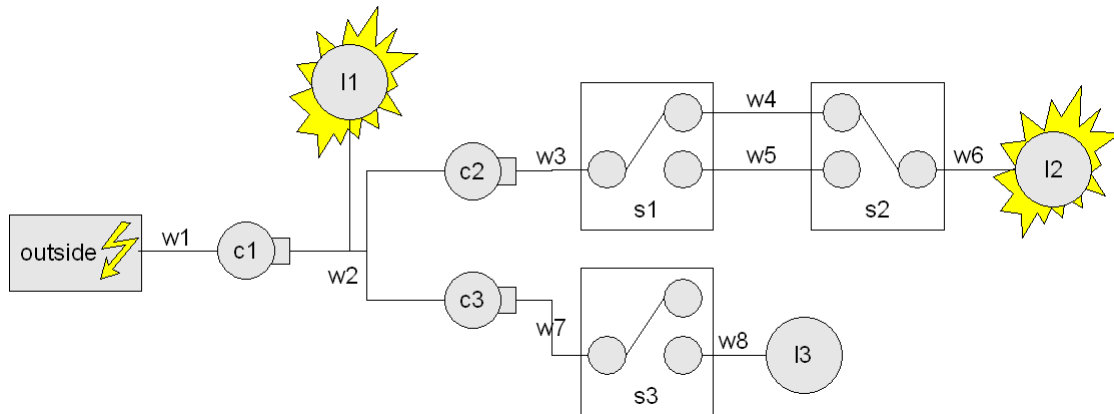


GWV – Grundlagen der Wissensverarbeitung  
 Aufgabenzettel 2 : Axiomatisierung / Regelbasierte Systeme  
 Abgabe 8.11.2009, 18.00h Besprechung am 9.11.2009.

Übungsaufgabe 2.1 : (Axiomatisierung (WBS))

von
6



Die folgende Aufgabe kann wahlweise auf Papier oder in Form eines Prologprogramms bearbeitet werden. (Letzteres ist eine gute Übung, um sich mit Prolog vertraut zu machen.)

In einem Gebäude findet sich die in Abbildung 1 dargestellte Elektroinstallation, dabei wurden folgende Abkürzungen verwendet: l = Lampe, w = Kabel (wire), c = Sicherung (circuit breaker), s = Schalter. Es ist davon auszugehen, dass alle Elemente der Elektroinstallation unbeschädigt sind.

1. Beschreiben Sie die oben gezeigte Schaltung mit Hilfe der Prädikat:  $light(X)$ ,  $up(S)$ ,  $down(S)$  und  $ok(E)$ . (1 Pkt.)

**light(L)** ist wahr, wenn L eine Lampe ist.

**up(S)** ist wahr, wenn ein Schalter nach oben steht.

**down(S)** ist wahr, wenn ein Schalter nach unten steht.

**ok(E)** ist wahr, wenn E nicht beschädigt ist. E kann eine Lampe oder eine Sicherung sein.

2. Durch das Prädikat  $connectedto(X,Y)$  kann modelliert werden, welche Elemente der Schaltung miteinander verbunden sind.

**connectedto(X,Y)** ist wahr, wenn X so mit Y verbunden sind, dass Strom von X nach Y fließen kann.

Beschreiben Sie die Verbindungen der oben gezeigten Schaltung mit Hilfe des  $connectedto$  Prädikats. Bedenken Sie, dass der Wahrheitswert des Prädikates in einigen Fällen von der Stellung eines Schalters oder dem Zustand einer Sicherung abhängen kann. (1 Pkt.)

3. Ob ein Element der Schaltung mit Strom versorgt wird, hängt davon ab, ob dieses Element mit einem Element verbunden ist, das mit Strom versorgt wird. Schreiben Sie ein Prädikat, das diesen transitiven Zusammenhang ausdrückt. Es ist davon auszugehen, dass die Stromversorgung des Gebäudes (outside) stets einwandfrei funktioniert:  $\text{live}(\text{outside})$  ist immer wahr. (2 Pkt.)

$\text{live}(W)$  ist wahr, wenn  $W$  mit Strom versorgt ist.

4. Entscheidend dafür, ob eine Lampe leuchtet ist, ob die Lampe mit Strom versorgt wird, und ob die entsprechende Lampe heil ist. Schreiben Sie ein Prädikat, das diesen Zusammenhang ausdrückt. (1 Pkt.)

$\text{lit}(L)$  ist wahr, wenn eine Lampe  $L$  leuchtet.

5. Begründen Sie mit Hilfe der von Ihnen aufgestellten Prädikate, dass Lampe  $l1$  und Lampe  $l2$  leuchten und dass Lampe  $l3$  nicht leuchtet. Falls Sie diese Aufgabe als Prologprogramm gelöst haben, können Sie den Beweis von Prolog führen lassen; benutzen Sie dazu das Prädikat "trace/0". Wie können sie diese Wissensbasis zur Diagnose nutzen, um beispielsweise herauszufinden, warum Lampe  $l3$  nicht leuchtet? (2 Pkt.)

**Übungsaufgabe 2.2: (Regelbasierte Systeme: Türme von Hanoi (WM))**

Für die Darstellung der in einem regelbasierten System vorgegebenen und erschlossenen Fakten (kurz für die Elemente des Arbeitsspeichers) wird folgendes Format festgelegt:

`(typ attribut1 : wert1 ... attributeN : wertN )`

Die Produktionsregeln haben die Form

`IF cond1 ... condN THEN aktion1 ...aktionM`

Das Antezedens besteht aus einer Liste von Bedingungen, das Sukzedens (oder Konsequenz) aus einer Liste von Aktionen. Eine Bedingung hat die Form

`(typ attribut1 : spez1 .... attributN : spezN)` bzw.

`- (typ attribut1 : spez1 .... attributN : spezN).`

Dabei ist - die Negation und jede Spezifikation ist entweder

- ein Atom,
- eine Variable,
- ein (arithmetisch) auswertbarer Ausdruck gekennzeichnet durch „[ ]“,
- ein boolescher Test gekennzeichnet mit „{ }“ oder
- die Konjunktion, Disjunktion oder Negation einer Spezifikation.

Beispielweise ist

`(person age: [n + 4] occupation: x)`

eine Bedingung, die genau dann durch ein Faktum des Typs `person` erfüllt wird, wenn es das Alter  $n + 4$  hat, wobei  $n$  an anderer bestimmt wurde.

VON
6

Die Bedingung

- (**person age**: {< 23  $\wedge$  > 6})

ist erfüllt, wenn es keine Person gibt, die älter als 6 und jünger als 26 ist.

Eine Aktion hat eine der folgenden Formen:

- **ADD cond**: Das bedeutet, dass ein Faktenelement der Form **cond** direkt in die Faktenbasis eingefügt wird.
- **REMOVE i**: Entferne das Faktum, welches mit der *i*-ten Bedingung im Antezedent der Regel gematched wurde. (Nur anwendbar, wenn die Bedingung positiv ist).
- **MODIFY i (attribut spez)**: Modifiziere das Faktum, welche mit der *i*-ten Bedingung in der Regel gematched wurde, derart, dass der aktuelle Wert des Attributs **attribut** durch **spez** ersetzt wird. (Nur anwendbar auf positive Bedingungen. )

Nun zur eigentlichen Aufgabe, dem Hanoi-Problem:

Gegeben sind drei Pflocke A,B,C. Zu Anfang sind  $n$  Scheiben auf *A* angeordnet, so dass die kleinste Scheibe oben ist. Ziel ist es, die Scheiben in der richtigen Anordnung (mit kleinster Scheibe zuoberst) auf Pflock *C* zu übertragen. In jedem Schritt darf nur die oberste Scheibe bewegt werden; eine Scheibe kann nur von einem Pflock auf den anderen bewegt werden; und zu keinem Zeitpunkt darf eine größere Scheibe auf einer kleineren Scheibe sein.

Zur Lösung soll folgendes iterative Verfahren verwendet werden: Zuerst werden die Pflocke in einem Kreis angeordnet, so dass sich im Uhrzeigersinn die Folge A, B, C und dann wieder A ergibt. Es wird angenommen, dass keine Scheibe mehr als einmal nacheinander bewegt wird. Dann kann immer nur genau eine Scheibe ohne Verletzung der obigen Bedingungen bewegt werden. Die Scheibe wird auf den nächstmöglichen Pflock bewegt. Falls  $n$  gerade ist, geschieht die Suche nach dem nächstmöglichen Pflock im Uhrzeigersinn, falls  $n$  ungerade ist, gegen den Uhrzeigersinn.

Schreiben Sie eine Folge von Produktionsregeln, die dieses Verfahren implementiert und testen sie ihre Regeln anhand des Falls von drei Scheiben.

Zu Anfang sollte die Faktenbasis für alle  $1 \leq i \leq n$  Fakten der Form (**auf pflock**: *A* **scheibe**: *i*) und ein Faktum (**start**) enthalten. Falls keine Regel mehr feuern kann, sollte die Faktenbasis (**ende**) und für alle  $1 \leq i \leq n$  (**auf pflock**: *C* **scheibe**: *i*) enthalten.

*Version: 2. November 2009*

*Summe der erreichbaren Punkte auf diesem Blatt: 12*

*Summe der bisher erreichbaren Punkte: 26*