Prof. Dr. Christoph Beierle, Prof. Dr. Gabriele Kern-Isberner

Modul 64211

Wissensbasierte Systeme

LESEPROBE

Fakultät für
Mathematik und Informatik

FernUniversität in Hagen

Selbsttestaufgabe 4.12 (Kreditvergabe 2) Wir gehen von der Situation in Selbsttestaufgabe 4.8 aus. Welcher der beiden folgenden Kunden bekommt einen Kredit? Instantiieren Sie Ihr Regelnetzwerk analog zu Abbildung 4.4 und beschreiben Sie die Arbeitsweise bei der Vorwärtsverkettung.

1. Peter Hansen ist schon seit 15 Jahren Kunde bei der Eckbank. Er hat sich nie unkorrekt verhalten und würde niemals sein Konto überziehen. Sein monatliches Gehalt von 2800 Euro geht regelmäßig auf sein Girokonto bei der Eckbank ein. Seit kurzem hat er eine Freundin, der er zum Geburtstag eine Weltreise schenken will. Dafür beantragt er einen Kredit. Seine Mutter bürgt für ihn.

2. Olivia Hölzer ist Professorin. Sie arbeitet neuerdings in Kantenhausen und hat seit kurzem ein Konto bei der Eckbank. Sie möchte für sich und ihre Familie ein Haus kaufen und fragt deshalb nach einem Kredit. ■

4.3.3 Zielorientierte Inferenz (Rückwärtsverkettung)

Die Vorwärtsverkettung ist sicherlich nützlich, wenn man einen Überblick über den allgemeinen Zustand eines Systems erhalten will. Oft ist man aber nur am Zustand ganz bestimmter Knoten (z.B. des Endknotens M) interessiert. Für diesen Fall bietet sich die *rückwärtsgerichtete Inferenz* oder *Rückwärtsverkettung* an.

Auch der Rückwärtsverkettung liegt das Prinzip der transitiven Verknüpfung von Regeln zugrunde. Hierbei geht man jedoch nicht von den gegebenen Daten aus, sondern von einem Zielobjekt, über dessen Zustand der Benutzer Informationen wünscht. Das System durchsucht dann die Regelbasis nach geeigneten Regeln, die also das Zielobjekt in der Konklusion enthalten. Die Objekte der Prämissen werden zu Zwischenzielen.

Bevor wir einen allgemeinen Algorithmus skizzieren und ihn auf das Regelwerk des Beispiels 4.7 anwenden, wollen wir die Idee an einem kleinen, trivialen Beispiel illustrieren.

Beispiel 4.13 Die Wissensbasis enthalte die (zweiwertigen) Objekte $O1, O2, O3$ und die Regeln

Regel 1: **if** $O1$ **then** $O2$
Regel 2: **if** $O2$ **then** $O3$

Wir wollen den Zustand des Zielobjektes $O3$ in Erfahrung bringen. Aufgrund von Regel 2 wissen wir, dass $O3$ wahr ist, wenn $O2$ wahr ist. Dieses wiederum ist erfüllt (Regel 1), wenn $O1$ wahr ist. Nun sucht das System gezielt nach Informationen über $O1$. □

Das allgemeine Vorgehen wird durch den Algorithmus in Abbildung 4.5 beschrieben. Dabei gehen wir davon aus, dass alle Regeln in Normalform vorliegen, und notieren eine Regel **if** $p_1 \wedge \ldots \wedge p_m$ **then** q als $p_1 \wedge \ldots \wedge p_m \rightarrow q$.

Zielorientierte Inferenz:

Eingabe: Eine Regelbasis RB (Objekte und Regeln),
eine (evidentielle) Faktenmenge $\mathcal{F}$,
eine Liste von Zielen (atomaren Anfragen) $[q_1, \ldots, q_n]$

Ausgabe: *yes*, falls alle q_i ableitbar sind, sonst *no*

BACKCHAIN($[q_1, \ldots, q_n]$)

wobei BACKCHAIN wie folgt rekursiv definiert ist:

```
BACKCHAIN([q_1, ..., q_n])                                          (1)
    if n = 0 then return(yes);                                      (2)
    if q_1 ∈ F                                                      (3)
      then return(BACKCHAIN([q_2, ..., q_n]))                       (4)
      else for each Regel p_1 ∧ ... ∧ p_m → q aus RB mit q_1 = q    (5)
           do if BACKCHAIN([p_1, ..., p_m, q_2, ..., q_n]) = yes    (6)
                then return(yes)                                    (7)
           endfor                                                   (8)
    endif                                                           (9)
    return(no)                                                     (10)
```

Abbildung 4.5: Algorithmus zur zielorientierten Inferenz

Die zielorientierte Inferenz ist dann wie in Abbildung 4.5 durch den Aufruf BACKCHAIN($[q_1, \ldots, q_n]$) definiert, wobei $[q_1, \ldots, q_n]$ die gegebene Liste von Zielen ist und BACKCHAIN rekursiv wie in den Zeilen (1) bis (10) in Abbildung 4.5 definiert ist.

Zu Beginn des Verfahrens besteht die Zielliste aus der Liste der angegebenen Zielobjekte; ist nur ein einzelnes Zielobjekt q_1 angegeben, also aus der Liste $[q_1]$. Ist diese Liste leer, liefert BACKCHAIN als Rückgabewert *yes*. Andernfalls sei q_1 das erste Element der Liste der Zielobjekte. Ist q_1 in der Faktenmenge $\mathcal{F}$ enthalten, wird BACKCHAIN mit der Restliste ohne q_1 aufgerufen. Andernfalls wird die Regelbasis RB gezielt nach Regeln durchsucht, die das Zielobjekt in der Konklusion enthalten. Findet es solche Regeln, so versucht das Verfahren, sie gezielt anzuwenden, indem für eine solche Regel BACKCHAIN rekursiv aufgerufen wird, wobei anstelle von q_1 die Prämissen der Regel als neue Ziele in die Liste der Zielobjekte aufgenommen werden. Ist für keine der Regeln mit q_1 als Konklusion dieser Aufruf erfolgreich, so kann dieses Zielobjekt nicht abgeleitet werden und BACKCHAIN liefert als Ergebnis *no* zurück. Beachten Sie aber, dass, sofern nicht weitere Vorkehrungen getroffen werden, BACKCHAIN bei rekursiven Regeln eventuell nicht terminiert, z.B. bei der Regelmenge $\{p \to q,\ q \to p\}$ und der Anfrage $[p]$.

Beispiel 4.14 Wir wenden den Algorithmus zur zielorientierten Inferenz aus