

Künstliche Intelligenz

Mitschrift von www.kuertz.name

Hinweis: Dies ist **kein offizielles Script**, sondern nur eine private Mitschrift. Die Mitschriften sind teilweise **unvollständig, falsch oder inaktuell**, da sie aus dem Zeitraum 2001–2005 stammen. Falls jemand einen Fehler entdeckt, so freue ich mich dennoch über einen kurzen Hinweis per E-Mail – vielen Dank!

Klaas Ole Kürtz (klaasole@kuertz.net)

Inhaltsverzeichnis

1	Einführung	1
2	Regelbasierte Systeme	2
2.1	Expertensysteme	2
2.2	Darstellung von Regeln	2
2.3	Arten von Prämissen	3
2.4	Regelauswertung	4
2.4.1	Rückwärtsverkettete Auswertung	4
2.4.2	Vorwärtsverkettete Auswertung	5

1 Einführung

folgt noch...

2 Regelbasierte Systeme

2.1 Expertensysteme

Die wichtigsten Komponenten eines *Expertensystems (XPS)* sind:

1. *Wissensbasis*: Faktenbasis, Regelbasis
2. *Wissenserwerbungsbasis*: dient dem Aufbau und der Wartung der Wissensbasis
3. *Erklärungskomponente*: gibt Erklärung zur Vorgehensweise oder zum Lösungsweg
4. *Interviewer-Komponente*: Dialog mit dem Benutzer
5. *Inferenzmaschine*: Schlußfolgerung, Problemlösungskomponente – der Kern des Systems

Eine *Shell* besteht nur aus den Punkten zwei bis fünf.

2.2 Darstellung von Regeln

Regeln stellen Wissen über allgemeine Beziehungen zwischen *Sachverhalten (Fakten)* dar, sie bestehen aus *Prämissen (Bedingungen)* und *Konklusion (Folgerung)*. Beispiel:

1. WENN der momentan betrachtete Buchstabe ein **q** ist
UND das Wort keine Abkürzung ist,
DANN ist der nächste Buchstabe ein **u**.
2. WENN der Motor zu heiß wird,
DANN prüfe Kühlflüssigkeit.

Zu den Regeln:

1. Die erste Regel ist eine sogenannte *Deduktionsregel* (Aussage oder Fakt im Konklusionsteil der Regel), man kann sie als logische Implikation interpretieren und mit *modus-ponens-Schema* auswerten: Falls die Regel „WENN *p* DANN *k*“ wahr ist und die Prämisse *p* wahr ist, so ist auch *k* wahr.
2. Die zweite Regel ist eine sogenannte *Aktionsregel* oder *Reaktionsregel*, da auf der rechten Seite eine Aktion steht.

Allgemeine Form einer Regel:

$$r : \text{WENN } p_1 \text{ UND } \dots \text{ UND } p_m \text{ DANN } k$$

Dabei ist die *rechte Handseite (RHS)* k eine *Aussage (Fakt oder Aktion)*. Die Bedingungen p_1, \dots, p_m sind atomare Aussagen, die bezüglich der Faktenbasis überprüft werden können.

Regeln werden häufig zu *Regelmengen (Regelsystemen)* zusammengefaßt:

$$\begin{aligned} r_1 & : \text{WENN } p_{1,1} \text{ UND } \dots \text{ UND } p_{1,m_1} \text{ DANN } k_1 \\ \dots & : \dots \\ r_n & : \text{WENN } p_{n,1} \text{ UND } \dots \text{ UND } p_{n,m_n} \text{ DANN } k_n \end{aligned}$$

Andere mögliche Darstellung:

$$r_i : p_{i,1}, \dots, p_{i,m_i} \rightarrow k_i \text{ für } i = 1, \dots, n$$

Dabei steht das Komma für eine Konjunktion. Eine Regel mit ODER in der Prämisse wird bezeichnet mit $p_1 \cup p_2 \rightarrow k$ und kann ersetzt werden durch zwei Regeln $r_1 : p_1 \rightarrow k$ und $r_2 : p_2 \rightarrow k$.

2.3 Arten von Prämissen

1. *Positive Prämissen* $+a$ sind erfüllt, wenn ein entsprechendes positives Faktum a in der Faktenbasis vorhanden ist (die Regel ist *anwendbar* bzw. *folgt*). D.h. die Prämisse ist erfüllt, wenn $a \in F$ ist für die Faktenbasis F .

Annahme: Alle in der Faktenbasis eingetragenen Fakten sind als „positive Fakten“ zu interpretieren. Fakten, die nicht eingetragen sind, werden automatisch als „negativ“ interpretiert. Diese Annahme nennt man auch *closed-world-assumption*: Was nicht in F ist, wird als falsch angenommen.

2. *Negative Prämisse* $-a$ ist erfüllt, wenn das Faktum a nicht in der Faktenbasis vorhanden ist. D.h. die Prämisse ist erfüllt, wenn $a \notin F$ ist für die Faktenbasis F .
3. *Prozedurale Prämisse*: Auswertung bzw. Prüfung der Prämisse wird i.A. von einer Prozedur durchgeführt.

2.4 Regelauswertung

2.4.1 Rückwärtsverkettete Auswertung

Bei der *rückwärtsverketteten Auswertung (zielorientierte Inferenz)* wird gestartet mit einer Anfrage bzw. Hypothese. Falls in F kein Faktum vorhanden ist, das die Hypothese bestätigt, sucht die Inferenzmaschine eine „passende“ Regel, also eine Regel, deren Konklusion die Hypothese darstellt. Die Prämisse (oder die Prämisseanteile) dieser Regel werden zu den neuen Hypothesen, die dann rekursiv wieder geprüft werden.

- Eingabe: Faktenbasis, Regelbasis, Hypothese (Ziel)
- Ausgabe: Bestätigung oder Nicht-Bestätigung der Hypothese

Beispiel: Diagnostik von Autoheizungen. Gegeben sei die Faktenbasis

$$F = \{(1) \text{ gebläsemotor_defekt}, (2) \text{ gebläse_klemmt}\}$$

Regeln wie oben¹. Hypothese sei (5) *heizung_defekt*. Verlauf der Rückwärtsverkettung:

Hypothese	Regel	nächster Schritt	Faktenbasis F
(5)		r_1	$\{(1), (2)\}$
(4)	r_1	r_2, r_3	$\{(1), (2)\}$
(1)	r_2	verifiziert	$\{(1), (2)\}$
(2)	r_2	verifiziert	$\{(1), (2), (4)\}$
(5)	r_1	verifiziert	$\{(1), (2), (4), (5)\}$

Schlußfolgerung: Die Heizung ist defekt! Falls keine oder zu wenig Fakten in F vorhanden sind, kann man die Faktenbasis durch Nachfrage an den Benutzer² ergänzen. Sei nun $F = \emptyset$, Hypothese wiederum (5).

Hypothese	Regel	nächster Schritt	Faktenbasis F	Eingabe
(5)		r_1	\emptyset	-
(4)	r_1	r_2, r_3	\emptyset	-
(1)	r_2	Nachfrage	\emptyset	<i>ja</i>
(2)	r_2	Nachfrage	$\{(1)\}$	<i>nein</i>
(1)	r_3	verifiziert	$\{(1)\}$	-
(3)	r_3	Nachfrage	$\{(1)\}$	<i>ja</i>

Schlußfolgerung: Die Hypothese wurde nicht verifiziert (wegen der negative Prämisse in r_3).

¹(3) *batterie_leer*, (4) *heizgebläse_defekt*

²„... aber ich habe noch nie Mörder und Mörderinnen gelesen!“

Negative Hypothesen

Liegt eine negative Hypothese $-a$ vor, so versucht man, die Hypothese $+a$ zu verifizieren. Schlägt dieser Versuch fehl, so wird die negative Hypothese bestätigt. Dies entspricht dem Prinzip der schwachen Negation:

- Das Prinzip der *schwachen Negation*: Eine negative Hypothese $-a$ wird dann akzeptiert, wenn alle Versuche, die positive Hypothese a zu beweisen, fehlschlagen.
- Das Prinzip der *starken Negation*: Eine negative Hypothese $-a$ wird dann akzeptiert, wenn $-a$ in der Faktenbasis vorhanden ist.

Im Weiteren wird nur das Prinzip der schwachen Negation angewandt. Für den Algorithmus sind Fakten- und Regelbasis vorgegeben. Die Rückwärtsverkettung wird durch die Boolesche Funktion `erfüllt?(hyp)` realisiert. Sie prüft, ob die Hypothese in F eingetragen ist oder ob es eine zu ihr passende Regel mit der Hypothese im Konklusionsteil gibt. Wenn ja, überprüfe, ob diese anwendbar ist (wobei `pre+` die positiven, `pre-` die negativen und `pre=` die prozeduralen Prämissen sind).

Steuerung der Regelverarbeitung

Die *Konfliktmenge* ist die Menge der Regeln, die momentan anwendbar sind. Die Regelansteuerung ist (bis jetzt) primitiv: Die Regeln werden in der Reihenfolge der Liste „Regelbasis“ abgearbeitet. Besser ist es, „wahrscheinlichere“ bzw. „günstigere“ Regeln zuerst zu verwenden. Dies wird durch Konfliktlösungsstrategien festgelegt.

Hier nur eine: Auswahl der Regel nach der *Priorität*: Jeder Regel ist eine Priorität zugeordnet, die aus strategischem Wissen oder Statistik folgt. Bei einer Regelsteuerung mit *dynamischen Prioritäten* wird die Priorität entweder statistisch berechnet oder nach einer *heuristischen Regel* (informiertes Verfahren): Addiere etwa, falls die Prämissen a_1, \dots, a_n der Regel h erfüllt sind, Δ zur Priorität der Regel r .

$$h: a_1, \dots, a_n \rightarrow \text{add}(r, \Delta)$$

2.4.2 Vorwärtsverkettete Auswertung

Bei der *vorwärtsverketteten Auswertung (datenorientierte Inferenz)* wird ...