6. Regelbasierte Systeme

Regel (über einer Sprache L): Konstrukt der Form

IF Bedingung$_1$, …, Bedingung$_n$ THEN Folgerung/Aktion

Falls Aktion im THEN-Teil: Produktionsregel, sonst Inferenzregel

Bedingungen und Folgerung in L formuliert

verschiedene Notationen gebräuchlich:

Bedingung$_1$, …, Bedingung$_n$ $\rightarrow$ Folgerung

Folgerung $\leftarrow$ Bedingung$_1$, …, Bedingung$_n$

Folgerung :- Bedingung$_1$, …, Bedingung$_n$

Die Folgerung nennt man auch Kopf der Regel, die Bedingungen Körper.

Viele der in den 80/90er Jahren entwickelten Expertensysteme regelbasiert.

Regeln vs. Implikationen

$A \supset B$ <u>ungerichtet</u>:
erlaubt Inferenz von B aus A, aber auch von $\neg A$ aus $\neg B$, bzw.
von B aus $A \supset B$ zusammen mit $\neg A \supset B$

$A \rightarrow B$ <u>gerichtet</u>:

nur wenn A gegeben, kann B hergeleitet werden: Kontrolle des Schließens

Falls L nur aus Atomen besteht (Horn Klausen): C folgerbar aus Implikationen gdw. C herleitbar aus entsprechenden Regeln

Beispiel:

Toddler
Toddler $\rightarrow$ ***Child***
Child, Male $\rightarrow$ ***Boy***
Infant $\rightarrow$ ***Child***
Child, Female $\rightarrow$ ***Girl***
Female

Schließen mit Regeln: backward vs. forward chaining

<u>Backward chaining</u> (Rückwärtsverkettung):

entspricht dem Finden von Regeln für noch offene Teilziele:

Ziel:	***Girl***
bewiesen falls beweisbar	***Child, Female***
bewiesen falls beweisbar	***Toddler, Female***
bewiesen falls beweisbar	***Female***
bewiesen falls beweisbar	**nichts mehr zu beweisen!**

Man spricht von backward chaining, da man vom Ziel zu den Prämissen (von Regel-Konklusion zu Regel-Prämisse) zurück geht.

Rekursive Beweisprozedur

input: endliche Liste von zu beweisenden Elementen $q_1, \ldots, q_n$
output: JA, falls aus gegebener Regelbasis R alle q_i herleitbar, NEIN sonst
procedure SOLVE[$q_1,\ldots,q_n$]
if n = 0 then return YES;
for each rule r ∈ R do
if r = [$p_1, \ldots, p_m \rightarrow q_1$] and SOLVE[$p_1, \ldots, p_m, q_2,\ldots,q_n$]
then return YES
end for;
return NO.

Implizite Suchstrategie: depth first left-to-right

Forward chaining (Vorwärtsverkettung)

Starte mit Fakten (Regeln ohne Vorbedingung) in R;
wende alle Regeln an, deren Vorbedingungen bereits abgeleitet sind;
iteriere, bis nichts Neues mehr abgeleitet werden kann.

Formaler:

Sei R Menge von Regeln.

Eine Menge S heißt abgeschlossen unter R, falls gilt:

$$[B_1, \ldots, B_n \rightarrow C] \in R \text{ und } B_1, \ldots, B_n \in S \text{ impliziert } C \in S.$$

Die Hülle von R ist die kleinste Menge, die abgeschlossen ist unter R.

Anmerkung: für jedes Element c der Hülle gibt es einen Beweis, d.h. eine Folge von Regeln $r_1, \ldots r_k$, so dass die Vorbedingungen von r_j jeweils in den Köpfen von $r_1 \ldots r_{j-1}$ enthalten sind und c Kopf von r_k ist. Die Hülle ist in diesem Sinne *gegründet*.

Iterative Berechnung der Hülle

Betrachte Operator T_R:

$$T_R(A) = \{q \mid p_1,\ldots, p_m \rightarrow q \in R, p_1, \ldots, p_m \in A\}$$

Definiere

$$T_R^0 = \varnothing$$

$$T_R^{i+1} = T_R(T_R^i)$$

Hülle von R kann berechnet werden durch wiederholtes Anwenden von T_R .

Start mit der leeren Menge (Hülle ist kleinster Fixpunkt von T_R).

Im Beispiel:

$$T_R^0 = \varnothing$$

$$T_R^1 = T_R(\varnothing) = \{\textit{Toddler, Female}\}$$

$$T_R^2 = T_R(T_R^1) = \{\textit{Toddler, Female, Child}\}$$

$$T_R^3 = T_R(T_R^2) = \{\textit{Toddler, Female, Child, Girl}\}$$

$$T_R^4 = T_R(T_R^3) = \{\textit{Toddler, Female, Child, Girl}\}$$

Forward vs. Backward

forward chaining Algorithmen leicht (im aussagenlogischen Fall sehr effizient) zu implementieren.
entsprechende Algorithmen existieren auch für first order Fall (Variablen in Regeln erlaubt, Instanzierung und Unifikation zu berücksichtigen).

<u>Vor- und Nachteile:</u>

***backward chaining* zielgerichtet: nur Regeln überprüft, die zu gesuchter Ableitung beitragen können; aber kann endlos laufen, z.B. wenn Regel der Form $p \rightarrow p$ vorhanden ist.**
***forward chaining* effizient in propositionalem Fall, terminiert möglicher Weise nicht im first order Fall, wenn Hülle unendlich werden kann.**

Vorwärtsverkettung kann besser sein!

2n Atome: $p_0, p_1, \ldots, p_{n-1}, q_0, q_1, \ldots, q_{n-1}$

4n-4 Regeln:

$p_0 \rightarrow p_1, p_1 \rightarrow p_2, \quad \ldots \quad p_{n-2} \rightarrow p_{n-1}$

$p_0 \rightarrow q_1, p_1 \rightarrow q_2, \quad \ldots \quad p_{n-2} \rightarrow q_{n-1}$

$q_0 \rightarrow p_1, q_1 \rightarrow p_2, \quad \ldots \quad q_{n-2} \rightarrow p_{n-1}$

$q_0 \rightarrow q_1, q_1 \rightarrow q_2, \quad \ldots \quad q_{n-2} \rightarrow q_{n-1}$

SOLVE[p_i] und SOLVE[q_i] liefern jeweils NO, aber erst nach 2^i Schritten (= Aufrufe von SOLVE). Einfacher Induktionsbeweis für p_i:

i = 1: ruft auf SOLVE[p_0] und SOLVE[q_0], also 2^1 Aufrufe

i > 1: ruft auf SOLVE[p_{i-1}] und SOLVE[q_{i-1}],

nach Induktionsvoraussetzung je 2^{i-1} Aufrufe von SOLVE, also insgesamt 2^i.

Forward chaining terminiert sofort: keine Vorbedingung einer Regel ableitbar