Kapitel L: V (Fortsetzung)

V. Produktionsregelsysteme

- Produktionsregelsysteme
- Forward-Chaining
- Backward-Chaining
- Verkettungsstrategien
- Produktionsregelsysteme mit Negation

Produktionsregelsysteme mit Negation

Definition 9 (PS mit Negation)

Ein Produktionsregelsystem mit Negation $P_N = (D, R_N)$ ist ein Produktionsregelsystem, bei dem der Bedingungsteil von Regeln auch die Negation `NOT` enthalten kann.

Beispiel:

$$\texttt{IF NOT}\, X \neq a \wedge \texttt{NOT}\, (Y = a \wedge Z \neq b)\ \texttt{THEN}\ W = a$$

Zwei Paradigmen zur Interpretation von `NOT`:

1. Negation-as-Failure
2. bezogen auf eine aktuelle, „statische“ Datenbasis

Produktionsregelsysteme mit Negation

1. Mit Hilfe von de Morgan lassen sich Regeln mit `NOT` so umformen, dass die Negation nur bei Atomen steht. (Negationsnormalform des Bedingungsteils)

$$\texttt{NOT}\,(\alpha_1 \wedge \ldots \wedge \alpha_n) \;\approx\; \texttt{NOT}\,(\alpha_1) \vee \ldots \vee \texttt{NOT}\,(\alpha_n)$$
$$\texttt{NOT}\,(\alpha_1 \vee \ldots \vee \alpha_n) \;\approx\; \texttt{NOT}\,(\alpha_1) \wedge \ldots \wedge \texttt{NOT}\,(\alpha_n)$$

Beispiel:

$$\texttt{IF NOT}\, X \neq a \wedge \texttt{NOT}\,(Y = a \wedge Z \neq b)\ \texttt{THEN}\ W = a$$

$$\approx\ \texttt{IF NOT}\, X \neq a \wedge (\texttt{NOT}\, Y = a \vee \texttt{NOT}\, Z \neq b)\ \texttt{THEN}\ W = a$$

Produktionsregelsysteme mit Negation

1. Mit Hilfe von de Morgan lassen sich Regeln mit `NOT` so umformen, dass die Negation nur bei Atomen steht. (Negationsnormalform des Bedingungsteils)

$$\texttt{NOT}\,(\alpha_1 \wedge \ldots \wedge \alpha_n) \;\approx\; \texttt{NOT}\,(\alpha_1) \vee \ldots \vee \texttt{NOT}\,(\alpha_n)$$
$$\texttt{NOT}\,(\alpha_1 \vee \ldots \vee \alpha_n) \;\approx\; \texttt{NOT}\,(\alpha_1) \wedge \ldots \wedge \texttt{NOT}\,(\alpha_n)$$

   Beispiel:

$$\texttt{IF NOT}\,X \neq a \wedge \texttt{NOT}\,(Y = a \wedge Z \neq b)\;\texttt{THEN}\;W = a$$

$$\approx\;\texttt{IF NOT}\,X \neq a \wedge (\texttt{NOT}\,Y = a \vee \texttt{NOT}\,Z \neq b)\;\texttt{THEN}\;W = a$$

2. Darauf aufbauend lässt sich die disjunktive Normalform herstellen und die Regeln aufspalten:

$$\texttt{IF NOT}\,X \neq a \wedge (\texttt{NOT}\,Y = a \vee \texttt{NOT}\,Z \neq b)\;\texttt{THEN}\;W = a$$

$$\approx\;\begin{array}{l}\texttt{IF NOT}\,X \neq a \wedge \texttt{NOT}\,Y = a\;\texttt{THEN}\;W = a\\ \texttt{IF NOT}\,X \neq a \wedge \texttt{NOT}\,Z \neq b\;\texttt{THEN}\;W = a\end{array}$$

Produktionsregelsysteme mit Negation

Interpretation von NOT als Negation-as-Failure

- wird in der Programmiersprache PROLOG verwandt
- hier rein aussagenlogischer Fall:
  Die „Bedingung NOT τ“ für ein Atom τ ist erfüllt, falls τ nicht ableitbar ist.
- Hintergrund dieser Interpretation ist die Closed World Assumption (CWA).

Annahme:

- Die Diskurswelt (Domäne, Situation) ist vollständig durch $P_N = (D, R_N)$ beschrieben.

$\Rightarrow$ Alle Fakten, die in der Diskurswelt gültig sind, sind auch ableitbar.

$\Rightarrow$

Failure bzgl. des Ableitens von τ

$\Leftrightarrow$

„NOT τ“ gilt in der Diskurswelt

Produktionsregelsysteme mit Negation

Definition 10 (Semantik von $\texttt{NOT}$ unter CWA)

In einem Produktionsregelsystem $P_N = (D, R_N)$ ist eine Bedingung $\texttt{NOT}\,\alpha$ genau dann erfüllt (wahr), wenn α nicht aus P_N ableitbar ist. Das heißt:

1. Ist α eine Konjunktion von Teilformeln α_i darf mindestens ein α_i nicht ableitbar sein, damit $\texttt{NOT}\,\alpha$ erfüllt ist.

2. Ist α eine Disjunktion von Teilformeln α_i so darf kein α_i ableitbar sein, damit $\texttt{NOT}\,\alpha$ erfüllt ist.

Bemerkungen:

- Dieser Erfüllbarkeitsbegriff kann unmittelbar in den Algorithmus BC-DFS integriert werden.
- Mit Negation-as-Failure wird eine neue Schlussregel eingeführt – in Zeichen:

$$(\alpha \not\vdash_{PS} \tau) \;\; \vdash_{\substack{PS_N \\ CWA}} \;\; \neg\tau$$

In Worten: Falls τ aus α nicht mittels $\vdash_{PS}$ ableitbar ist, so ist $\neg\tau$ unter der Closed-World-Assumption ableitbar.

Produktionsregelsysteme mit Negation

Algorithm: BC-DFS-N

Input: Startdatenbasis D, Regelmenge R, Formel α

Output: *true*, falls α ableitbar unter CWA, *false* sonst; evtl. Endlosschleife

```
BEGIN
  IF α = NOT α1 THEN RETURN (NOT BC-DFS-N(α1)) ENDIF
  IF α = α1 ∧ α2 THEN RETURN (BC-DFS-N(α1) AND BC-DFS-N(α2)) ENDIF
  IF α = α1 ∨ α2 THEN RETURN (BC-DFS-N(α1) OR BC-DFS-N(α2)) ENDIF
  IF α ∈ D THEN RETURN (true) ENDIF
```

Produktionsregelsysteme mit Negation

Algorithm: BC-DFS-N
Input: Startdatenbasis D, Regelmenge R, Formel α
Output: *true*, falls α ableitbar unter CWA, *false* sonst; evtl. Endlosschleife

```
BEGIN
  IF α = NOT α1 THEN RETURN (NOT BC-DFS-N(α1)) ENDIF
  IF α = α1 ∧ α2 THEN RETURN (BC-DFS-N(α1) AND BC-DFS-N(α2)) ENDIF
  IF α = α1 ∨ α2 THEN RETURN (BC-DFS-N(α1) OR BC-DFS-N(α2)) ENDIF
  IF α ∈ D THEN RETURN (true) ENDIF
  R* = {r | r = (IF γ THEN α) und r ∈ R}
  stop=false
  WHILE R* ≠ ∅ AND stop=false DO
      r = choose(R*)
      IF BC-DFS-N(premise(r))=true
      THEN stop=true
      ELSE R* = R* \ {r}
  END
  IF stop=true
  THEN RETURN (true)
  ELSE RETURN (false)
END
```


Produktionsregelsysteme mit Negation

Zyklische Regelmengen und $\texttt{NOT}$

Sei folgendes Produktionsregelsystem $P_N = (D, R_N)$ gegeben:

$$
\begin{aligned}
D = \{\} \qquad R_N = \{ & r_1 : \texttt{IF NOT}\, X = a \;\texttt{THEN}\; Y = b, \\
& r_2 : \texttt{IF NOT}\, Y = b \;\texttt{THEN}\; X = a \}
\end{aligned}
$$

- In R_N enthält eine Schleife für das Ziel $Y = b$ und für das Ziel $X = a$.
- Schleifen (unendliche Ableitungen) dürfen nicht mit der Nicht-Ableitbarkeit eines Faktes gleichgesetzt werden.

Produktionsregelsysteme mit Negation

Negation-as-Failure und Vorwärtsverkettung

Bei der Vorwärtsverkettung hängt die Erfüllung einer Bedingung von der aktuellen Datenbasis D ab.

$\Rightarrow$ Die Bildung der Konfliktmenge hängt vom aktuellen D ab.

Im Widerspruch dazu steht Negation-as-Failure:

- Die Erfüllung einer Bedingung hängt von der Ableitbarkeit ab.

$\Rightarrow$ Für (D, R_N) macht ein rein vorwärtsverkettendes Verfahren keinen Sinn, weil bei negierten Bedingungen die Ableitbarkeit von Atomen getestet werden muss.

$\Rightarrow$ Die Integration eines rückwärtsverkettenden Verfahrens und die Kombination beider Verkettungsstrategien ist notwendig.

Produktionsregelsysteme mit Negation

Negation-as-Failure und Vorwärtsverkettung

Sei folgendes Produktionsregelsystem $P_N = (D, R_N)$ gegeben:

$$D = \{\} \qquad R_N = \{r_1 : \texttt{IF } Z = a \texttt{ THEN } X = b,$$
$$r_2 : \texttt{IF NOT } Y = b \texttt{ THEN } Z = a,$$
$$r_3 : \texttt{IF } U = 1 \texttt{ THEN } Y = b\}$$

- Test, ob r_2 in die Konfliktmenge kommt.

$\Rightarrow$ Test, ob $Y = b$ abgeleitet werden kann.

$\Rightarrow$ Backward-Chaining

Alternative

Anwendung einer anderen Interpretation der Negation bei vorwärtsverkettenden Verfahren. Idee: Konfliktmengenbildung bei statischer Datenbasis D.

Produktionsregelsysteme mit Negation

Statische Interpretation von NOT und Vorwärtsverkettung

Definition 11 (Semantik von NOT unter D)

Eine Bedingung NOT α ist in Bezug auf eine Datenbasis D genau dann erfüllt, wenn α in Bezug auf D nicht erfüllt ist:

- Ist α ein Atom, so muss $\alpha \notin D$ gelten.
- Andernfalls wird das Erfülltsein von α entsprechend der Junktoren auf das Erfülltsein der Teilformeln zurückgeführt.

Produktionsregelsysteme mit Negation

Statische Interpretation von NOT und Vorwärtsverkettung

Lemma 2

Produktionsregelsysteme mit Negation und der Interpretation der Negation in Bezug auf die Datenbasis sind nicht kommutativ.

Beweis 2 (Lemma)

Sei folgendes Produktionsregelsystem $P_N = (D, R_N)$ gegeben:

$$D = \{\} \qquad R_N = \{r_1 : \texttt{IF NOT}\, X = a \,\texttt{THEN}\, Y = b,$$
$$r_2 : \texttt{IF NOT}\, Y = b \,\texttt{THEN}\, X = a\}$$

Wegen $D = \emptyset$ ist sowohl r_1 als auch r_2 anwendbar. Wähle r_1.

$\Rightarrow$ $(D, R_N) \mid\!\frac{1}{PS}\, (D_1, R_N)$ mit $D_1 = \{Y = b\}$.

$\Rightarrow$ Für D_1 ist die Bedingung von r_2 nicht länger erfüllt.

$\Rightarrow$ r_2 ist nicht anwendbar.

$\Rightarrow$ P_N nicht kommutativ. □

Produktionsregelsysteme mit Negation

Statische Interpretation von NOT und Vorwärtsverkettung

Algorithm: FC-N-test
Input: Startdatenbasis D, Regelmenge R_N, Atom τ^*
Output: *true*, falls $(D, R) \mid_{\overline{PS}} \tau^*$, *unknown* sonst

```
BEGIN
  D* = D
  R_tmp = R

  REPEAT
    R* = {(IF α THEN τ) ∈ R_tmp | α wahr bzgl. D*}
    IF R* ≠ ∅
    THEN BEGIN
      r = choose(R*)
      D* = D* ∪ {conclusion(r)}
      R_tmp = R_tmp \ {r}
    END
    ELSE R_tmp = ∅
  UNTIL R_tmp = ∅

  IF τ* ∈ D*
  THEN RETURN(true)
  ELSE RETURN(unknown)
END
```

Bemerkungen:

- FC-N-test terminiert bei jeder Eingabe.
- Aufgrund der Nicht-Kommutativität kommt dem Aufruf *choose*(R^*) eine besondere Bedeutung zu: Nicht jede Auswahl von Regeln liefert das Ergebnis *true*, auch wenn $(D, R_N) \mid\!\frac{}{_{PS}}\, \tau$ gilt.

Produktionsregelsysteme mit Negation

Statische Interpretation von `NOT` und Vorwärtsverkettung

Satz 3 (Korrektheit und Vollständigkeit von FC-N-test**)**
Es sei P_N ein Produktionsregelsystem mit Negation und τ ein Atom. Dann gilt FC-N-test(D, R_N, τ) = *true* ist möglich genau dann, wenn sich τ aus P_N ableiten lässt, d.h. $(D, R_N) \mid\!\frac{}{PS}\, \tau$ gilt.

Produktionsregelsysteme mit Negation

Beweis 3 (Korrektheit und Vollständigkeit von FC-N-test)

„⇒“ Korrektheit

Aus FC-N-test(D, R_N, τ) = *true* ist möglich folgt $(D, R_N) \mathrel{|}\!\!\underline{}_{PS}\, \tau$.

Klar, weil jeder Iterationsschritt des Algorithmus genau einem Schritt der Ableitung $\mathrel{|}\!\!\frac{1}{PS}$ entspricht.

„⇐“ Vollständigkeit

Aus $(D, R_N) \mathrel{|}\!\!\underline{}_{PS}\, \tau$ folgt, dass eine Ableitungsfolge für FC-N-test(D, R_N, τ) existiert mit FC-N-test(D, R_N, τ) = *true*.

- Nach Voraussetzung existiert eine Folge von Regelanwendungen

  $$(D, R_N) \mathrel{|}\!\!\frac{1}{PS} (D_1, R_N) \mathrel{|}\!\!\frac{1}{PS} \ \ldots\ \mathrel{|}\!\!\frac{1}{PS} (D_k, R_N) \text{ mit } \tau \in D_k,$$

  wobei D_i aus D_{i-1} durch Anwendung einer Regel entsteht.

- Wähle die entsprechenden Regeln in dieser Reihenfolge für die ersten k Schleifendurchläufe in FC-N-test.

$\Rightarrow$ $D_k \subseteq D^*$

$\Rightarrow$ τ wurde abgeleitet.

Produktionsregelsysteme mit Negation

Nicht-Determinismus von FC-N-test

- Aus $(D, R_N) \mid\!\frac{}{PS}\, \tau$ folgt nicht, dass FC-N-test(D, R_N, τ) den Rückgabewert *true* liefern muss.
- Im Falle der Nichtableitung von τ ist der Rückgabewert von FC-N-test *unknown*.
- Unter der Voraussetzung P $\neq$ NP lässt sich der Nichtdeterminismus von FC-N-test auch nicht so auflösen, dass ein polynomiell beschränktes deterministisches Verfahren zur Bestimmung der Ableitbarkeit entsteht:

Satz 4 (NP-Vollständigkeit des Ableitbarkeitsproblems)
Es sei P_N ein Produktionsregelsystem mit Negation und τ ein Atom. Das Entscheidungsproblem „Lässt sich τ aus P_N ableiten?“ – kurz: „Gilt $P_N \mid\!\frac{}{PS}\, \tau$?“ – ist NP-vollständig.

Produktionsregelsysteme mit Negation

Beweisidee (NP-Vollständigkeit des Ableitbarkeitsproblems)

1. Obere Schranke.

   $P_N \mid_{\overline{PS}} \tau$ ist in NP; Argumentation über FC-N-test.

2. Vollständigkeit. Reduktion von 3SAT auf $P_N \mid_{\overline{PS}} \tau$.

   Konstruktion einer Menge R_α von Regeln zu einer aussagenlogischen Formel α mit

$$\alpha \text{ erfüllbar} \quad \Leftrightarrow \quad P_\alpha = (\emptyset, R_\alpha) \mid_{\overline{PS}} \tau, \ \tau = (Y = 1)$$

Argumentation zu Punkt 2:

„⇒" Mit Erfüllbarkeit von α folgt $P_\alpha \mid_{\overline{PS}} (Y = 1)$: Die erfüllende Belegung $\Im$ der Atome in α lässt die Regeln so feuern, dass $Y = 1$ von P_α abgeleitet werden kann.

„⇐" Mit $P_\alpha \mid_{\overline{PS}} (Y = 1)$ folgt die Erfüllbarkeit von α: Aus den gefeuerten Regeln folgt eine erfüllende Belegung $\Im$ der Atome in α.