

Reguli în sistemele de producție

Sistemele de producție sunt o modalitate de formalizare a cunoașterii utilizate în multe aplicații practice (sisteme expert). Folosesc un raționament de tip întântuire încânt, cu reguli de un anumit format, numite reguli de producție.

Un sistem de producție menține o memorie de lucru (Working Memory WM). WM conține informații ce se schimbă în timpul operațiilor sistemului.

O regulă de producție constă dintr-o mulțime de condiții și o mulțime de acțiuni (consecințe):

IF condiții THEN acțiuni

Condițiile sunt teste aplicate stării curente a WM, iar consecințele sunt acțiuni ce modifică WM.

Operație de bază a unui sistem de producție este un ciclu de trei pași ce se repetă cât timp există reguli ce se pot aplica WM:

1. recunoaștere - determinarea regulilor aplicabile, adică a regulilor ale căror condiții sunt satisfăcute de WM curentă;
2. rezolvare conflictelor - dintre regulile găsite la pasul 1, numite mulțime de conflict, se alege regula care se execută;
3. acțiune - modifică WM aplicând consecințele regulilor selectate la pasul 2.

Memorie de lucru WM

Este alcătuită dintr-o mulțime de elemente WMEs (Working Memory Elements). Un WME are forma

(type attribute₁:value₁ ... attribute_n:value_n)

unde type, attribute_i și value_i sunt atomi.

Exemple: (person age:21 home:bucharest)
(student name:john dept:computerScience)

În sens declarativ, un WME este o propoziție existențială:

$$\exists x [type(x) \wedge attribute_1(x) = value_1 \wedge \dots \wedge attribute_n(x) = value_n]$$

WMEs reprezintă obiecte, iar relațiile dintre ele pot fi manipulate prin reșificări (adică transformarea unei propoziții într-un obiect)

$$\left[\begin{array}{c} \text{Purchases}(\text{john}, \text{bike}, \text{nov9}) \\ \downarrow \\ \text{Purchases}(p3) \wedge \text{agent}(p3) = \text{john} \wedge \text{object}(p3) = \text{bike} \dots \end{array} \right]$$

De exemplu, (binaryFact relation: olderThan firstArg: john secondArg: mary) poate fi folosit pentru a exprima faptul că John este mai bătrân ca Mary.

Reguli de producție

Condițiile unei reguli de producție sunt legate prin conjuncție. Pentru ca o regulă să fie aplicabilă la un moment dat, trebuie ca toate condițiile să fie adevărate.

O condiție poate fi pozitivă sau negativă (notată cu -cond) de formă:

$(\text{type attribute}_1 : \text{specification}_1 \dots \text{attribute}_k : \text{specification}_k),$

unde fiecare specificație este:

- un atom
- o variabilă
- o expresie evaluabilă $[\]$
- un test $\{ \}$
- \vee, \wedge sau \neg unei specificații

De exemplu, $(\text{person age} : [n+4] \text{ occupation} : x)$ este satisfăcută dacă există un WME de tipul person, cu atributul age $n+4$, unde n este specificat undeva. Dacă variabila x nu este legată, atunci x va fi legată cu valoarea atributului occupation; altfel, valoarea atributului occupation din WME trebuie să fie valoarea lui x .

- $(\text{person age} : \{ <23 \wedge >6 \})$ este satisfăcută dacă nu există niciun WME în WM al cărui tip este person iar valoarea atributului age este între 6 și 23 (negativă ca eșec).

Un WME se potrivește cu o condiție dacă tipurile sunt identice și pentru fiecare pereche atribut/specificație din condiție există o pereche corespundentă atribut/valoare în WME, iar valoarea se potrivește cu specificația. WME poate avea și alte atribute ce nu se referă în condiție.

Consecințele regulilor de producție au o interpretare strict procedurală. Toate acțiunile sunt executate în ordine și pot fi de trei tipuri:

1. ADD un nou WME
2. REMOVE i : șterge WME din WM ce se potrivește cu condiția i din regulă. Nu se aplică dacă condiția este negativă.
3. MODIFY i (attribute specification) : modifică WME ce se potrivește cu condiția i din regulă, înlocuind valorile curente ale atributelor prin specificații. Nu se aplică dacă condiția este negativă.

Obs. În ADD și MODIFY, variabilele care apar se referă la valorile obținute prin potrivirea condițiilor regulii.

Exemplul 1 IF (student name: x) THEN ADD (person name: x)
(echivalentul $\forall x. \text{Student}(x) \supset \text{Person}(x)$)

Exemplul 2 - presupunem că o regulă adăugă un WME de tipul birthday

```
IF (person age:  $x$  name:  $n$ ) (birthday who:  $n$ )
  THEN MODIFY 1 (age [ $x+1$ ])
      REMOVE 2
```

\pentru a împiedica regula să se execute din nou

Exemplul 3 - pentru indicarea unei etape de calcul

```
IF (starting)
  THEN REMOVE 1
      ADD (control phase: 1)
```

```
...
IF (control phase:  $x$ ) ... alte condiții
  THEN MODIFY 1 (phase [ $x+1$ ])
```

```
IF (control phase: 5)
  THEN REMOVE 1
```

Exemplul 4 Avem 3 cuburi de dimensiuni diferite într-o grămadă. Cu ajutorul unei brațe robotice ele pot fi mutate în pozițiile numite 1, 2, 3. Șeșul este ce cel mai mare cub se așază pe poziție 1, cel mijlociu pe poziție 2 și cel mai mic pe poziție 3.

WM

- 1 (counter value: 1)
- 2 (cube name: A size: 10 position: heap)
- 3 (cube name: B size: 30 position: heap)
- 4 (cube name: C size: 20 position: heap)

Reguli

- 1. IF (cube position: heap name: n size: s)
 - (cube position: heap size: $\{ > s \}$)
 - (cube position: hand)
 THEN MODIFY 1 (position hand)
- 2. IF (cube position: hand)
 - (counter value: i)
 THEN MODIFY 1 (position i)
 - MODIFY 2 (value $[i+1]$)

nu avem conflicte deoarece la un moment doar o regulă poate fi executată

Regula 1 $\rightarrow n = B, s = 30 \rightarrow$ WME 3 se modifică
(cube name: B size: 30 position: hand)

Regula 2 $\rightarrow i = 1 \rightarrow$ WME 3 se modifică
(cube name: B size: 30 position: 1)
WME 1 se modifică (counter value: 2)

Regula 1 $\rightarrow n = C, s = 20 \rightarrow$ WME 4 se modifică
(cube name: C size: 20 position: hand)

Regula 2 $\rightarrow i = 2 \rightarrow$ WME 4 se modifică
(cube name: C size: 20 position: 2)
WME 1 se modifică (counter value: 3)

Regula 1 $\rightarrow n = A, s = 10 \rightarrow$ WME 2 se modifică
(cube name: A size: 10 position: hand)

Regula 2 $\rightarrow i = 3 \rightarrow$ WME 2 se modifică
(cube name: A size: 10 position: 3)
WME 1 se modifică (counter value: 4)

WM $\left[\begin{array}{l} 1. (\text{counter value} : 4) \\ 2. (\text{cube name} : A \text{ size} : 10 \text{ position} : 3) \\ 3. (\text{cube name} : B \text{ size} : 30 \text{ position} : 1) \\ 4. (\text{cube name} : C \text{ size} : 20 \text{ position} : 2) \end{array} \right.$

În acest moment nicio regulă nu mai este aplicabilă, deci sistemul de producție se oprește.

Exemplul 5 Pentru un an dat, se se afișeze numărul de zile.

if (an \nmid 4) then (an obisnuit)
 else if (an \nmid 100) then (an bisect)
 else if (an \nmid 400) then (an obisnuit)
 else (an bisect)

WM [(wontDays year: 1956)]

Reguli $\left[\begin{array}{l} 1. \text{ IF (wontDays year: } n) \\ \text{ THEN REMOVE 1} \\ \text{ ADD (year mod 4 : } [n \% 4] \text{ mod 100 : } [n \% 100] \text{ mod 400 : } [n \% 400]) \\ 2. \text{ IF (year mod 4 : } \{ \neq 0 \}) \\ \text{ THEN REMOVE 1} \\ \text{ ADD (hasDays days : 365)} \\ 3. \text{ IF (year mod 4 : } 0 \text{ mod 100 : } \{ \neq 0 \}) \\ \text{ THEN REMOVE 1} \\ \text{ ADD (hasDays days : 366)} \\ 4. \text{ IF (year mod 100 : } 0 \text{ mod 400 : } \{ \neq 0 \}) \\ \text{ THEN REMOVE 1} \\ \text{ ADD (hasDays days : 365)} \\ 5. \text{ IF (year mod 400 : } 0) \\ \text{ THEN REMOVE 1} \\ \text{ ADD (hasDays days : 366)} \end{array} \right.$

Deoarece condițiile sunt disjuncte, doar o regulă se poate executa.

WM [(hasDays days: 366)]

Acum nicio regulă nu mai este aplicabilă, sistemul de producție se oprește.

Rezolvarea conflictelor

Există o serie de strategii de eliminare a unor reguli aplicabile, dacă dorim ca să lucrăm. Cele mai întâlnite abordări:

- ordine: se aplică prima regulă aplicabilă, în ordinea prezentării. Trebuie ținut cont de ca să lucrăm atunci când se construiesc regulile.
- specificitate: selectarea regulii ale cărei condiții sunt cele mai specifice

IF (bird) THEN ADD (canFly)

mai specific decât prima { IF (bird weight: { > 100g) THEN ADD (cannotFly)

IF (bird) (penguin) THEN ADD (cannotFly)

(~~deci~~ se poate aplica un alt criteriu pentru a pune o singură regulă aplicabilă

- momentul ultimei utilizări: se alege regula care se poate utiliza cu cel mai recent creat/modificat WME (sau cel mai puțin recent)
- respingerea unei reguli aplicabile care a mai fost aplicată cu aceleași valori ale variabilelor - previne comportamentul ciclic rezultat din aplicarea repetată a unei reguli prin potrivirea cu același WME.

Eficiența sistemelor de producție

Operația de potrivire a regulilor cu WMEs reprezintă aproximativ 30% din timpul de operare a unui sistem de producție.

Dacă observăm cheie cu condus la o implementare foarte eficientă:

- WM se modifică puțin la fiecare ciclu de execuție
- multe reguli cu condiții comune

Algoritmul RETE (1974) creează o rețea din condițiile regulilor. Deoarece regulile nu se schimbă în timpul operației sistemului de producție, rețeaua este creată o singură dată, urmând ca în timpul operației informațiile legate de potrivirea condițiilor cu WMEs să fie actualizate. Astfel, la fiecare ciclu de execuție doar o mică parte din WM trebuie re-potrivită cu condițiile regulilor, reducând drastic timpul de calcul al mulțimii de conflict.

Avantajele sistemelor de producție

- modularitate: deoarece regulile lucrează independent, se pot adăuga sau șterge relativ ușor.
- control: structură simplă de control, fără "puncte oarbe" în implementare
- transparență: regulile folosesc o terminologie ușor de înțeles iar raționamentul sistemului de producție poate fi urmărit și explicat în limbaj natural.

În sisteme cu mulți de reguli, aceste avantaje tind să se diminueze. Cu toate acestea, sistemele de producție sunt folosite cu succes în multe probleme practice (diagnostic medical, eligibilitate pentru obținerea unui credit, configurații de sisteme etc.)

MYCIN - sistem expert pentru diagnosticarea infecțiilor bacteriene, dezvoltat la Univ. Stanford în anii '70.

500 de reguli de producție pentru identificarea a 100 de cazuri de infecție.

Noutatea cea mai remarcabilă a fost introducerea unui nivel de certitudine a evidențelor și a încrederii în ipoteze.

XCON - sistem bazat pe reguli folosit în configurarea calculatoarelor, dezvoltat la Univ. Carnegie - Mellon în 1978.

10000 de reguli pentru sute de tipuri de componente

Sistemul a contribuit la creșterea interesului comercial pentru sisteme bazate pe reguli.

