エキスパートシステムにおける競合解消ルールの線型順序化

5J-9

5J - 9

黒瀬 博靖
(株) リコー
ソフトウェア研究所

1 はじめに

エキスパートシステムの一部として競合解消ルールがある。同時に発火可能なルールが複数個存在した場合のルールの選択手段である。競合解消ルールの与え方によってエキスパートシステムの動作は全く異なってくる。OPS5[1]においては競合解消ルールとして，LEX と MEA が採用されている。これは最新のワーキングメモリに対応するルール，より条件の厳しいルールから発火するという戦略である。これに対して，OPS5 の戦略はユーザにトリッキーなコーディングを要求するものであるという意見もある[3]。OPS83[2]や CL[8]では，ユーザに競合解消ルールのプログラミングを許している。またメタルールによる競合解消戦略の記述を行う手段も提唱されている[3]。確かにユーザプログラマブルであればいかなる動作も記述できる。しかしユーザに競合解消ルールをプログラムさせるのは混乱を招くものであり，あくまで OPS5 のパラダイムの中で戦略を考えるべきであるという意見もある[7,6]。

このように競合解消ルールに対する考え方はさまざまであるが，それによってシステムがどのように動くのかユーザに容易に認識できることが最も重要な条件であると思われる。そこで我々はルールの優先順位に着目した。ルールの優先順位とは，コンパイル時に任意のルールの組に対して定まる発火順序である。これによって競合解消ルールを作成すれば，ユーザは容易にシステムの動きをトレース出来る。発火可能なルールの組にたいしてどちらが優先されるかを記述すれば，任意の優先順位が記述できる。当リコーソフトウェア研究所で開発した推論エンジン GIE においては，全てのルールに自然数を割り当てることにより，ルールに線型順序を与え，競合解消をおこなっている[4,5]。

だが優先順位が必ず線型順序になるとは限らない。実際の局面で生じた競合集合のなかでは優先順位は線型順序になっているであろう。しかし同時発火可能な２つのルールを持ってきたときにどちらが優先か一意的に定められたとしても，その優先順位をまとめた関係は半順序にすらならない恐れがある。

競合解消ルールを順序関係でない優先順位として持っているとすると問題がある。静的に定まっているはずの優先順位を局面ごとに検索する必要が生ずるうえ，それが正当な優先順位かどうかチェックすることが出来ない。

そこで同等の力を持つルールセットと優先順位で，優先順位が線型順序になっているものを生成することによって問題解決を試みた。

2 定義

定義 1 ここで述べるルールとは次の形のものを指す。

$$C \longrightarrow A$$

Cは一階述語論理式である。A は任意の形を許す。C が成立したときに A が発火される。Cをルールの条件部，A をルールの実行部と呼ぶ。

定義 2 ルール集合 Sが両立可能であるとは，Sに含まれる全てのルールの条件部を同時に成立させるような解釈が存在する場合をいう。また２つのルール A , Bからなる集合が両立可能な場合，A と Bは両立可能であるという。

定義 3 任意の２つのルールに対して優先順位をつけることが出来る。ルール A がルール Bよりも優先されるとき，$A > B$と書く。$A > B$と $B > A$ が同時に成立することはない。また両立可能でないルール間の優先順位は意味を持たない。

原理 1 優先順位の基本原理

任意のルール集合 Sにおいて Sが両立可能ならば，Sの上の優先順位は線型順序をなす。

実際にルールを判定してみた結果としていくつかのルールが選ばれたとき，そのルールを優先する順番に並べることが出来ると言い替えることが出来る。これは優先順位に対する自然な要求である。これからはこの原理を満たすものに関する議論を進めていく。

定義 4 ルール集合 Rと R'が等価であるとはいかなるルールの解釈を選んでも同一の実行部が発火可能になることをいう。

定義 5 ルール集合 Oがルール集合 Rの原始集合であるとは，Rの集合としての分割 $\{R_i\}(i \in I)$と OからR_iへの上への１対１対応($\tau : o_i \to R_i$) が存在し，次の条件を満足することをいう

1. $o_i : C_i \longrightarrow A_i$ とすると，R_iに含まれるルール r_{ij}は全て $r_{ij} : C_{ij} \longrightarrow A_i$ という形をしている。
2. 全ての R_iにたいし $C_i \equiv \sum_{r_{ij} \in R_i} C_{ij}$ が成立する。

定義 6 ルール集合 Rの上の優先順位 Pとルール集合 R'の上の優先順位 P'が等価であるとは次の条件を満たすことを言う。

1. Rと R'は等価である。
2. いかなるルールの解釈においても発火可能な実行部の優先順位が一致する。

等価なルール集合と優先順位を使用すれば，エキスパートシステムの実行結果は常に同一となる。ルール集合 Rと R'がともにルール集合 R''を原始集合とする場合，Rと R'は等価なルール集合となる。

定義 7 ルール集合 Rとその原始集合 Oがあって，$o_i , o_j \in O , o_i > o_j$ という優先順位がある場合，その優先順位に対応する R上の優先順位を次のようにしてつける。原始集合の定義より，$\tau(o_i) = R_i , \tau(o_j) = R_j$がある。$r_{ik} \in R_i , r_{jl} \in R_j$となるルールの中で r_{ik}と r_{jl}が両立可能なものに対して $r_{ik} > r_{jl}$という優先順位をつける。このようにしてつけた R上の優先順位全体を Oから Rへの優先順位の自然な拡張と呼ぶ。

優先順位の自然な拡張は元の優先順位と等価である。

Linearization of conflict resolution strategy in an expert system
Hiroyasu Kurose
RICOH Co.,Ltd.

定義 8 $i \in J$のときC_{iJ}を以下のように定める。

$$C_{iJ} \equiv C_i \wedge \prod_{j \in J} C_j \wedge \prod_{k \in I-J} (\neg C_k)$$

明らかにルール集合 $\{C_i \longrightarrow A_i\}$ はルール集合 $\{C_{iJ} \longrightarrow A_i \mid J \subset I \wedge i \in J\}$ と等価である。この新しく作られた条件を元に等価なルールを作成していく。

3 補題

補題 1 ルール $R_{iJ}(C_{iJ} \longrightarrow A_i)$とルール $R_{kL}(C_{kL} \longrightarrow A_k)$が両立するならば$i \in J$, $k \in L$ かつ $J = L$

証明 両立の定義より, $C_{iJ} \wedge C_{kL}$ がある解釈の元で成立する。これより明らかに$i \in J$, $k \in L$が言える。

$$\begin{gathered} C_{iJ} \wedge C_{kL} \equiv \\ C_i \wedge \prod_{j \in J} C_j \wedge \prod_{m \in I-J} (\neg C_m) \wedge C_k \wedge \prod_{l \in L} C_l \wedge \prod_{n \in I-L} (\neg C_n) \equiv \\ C_i \wedge C_k \wedge \prod_{j \in J \cap L} C_j \wedge \prod_{m \in I-(J \cup L)} (\neg C_m) \wedge \prod_{j \in J-L} (C_j \wedge (\neg C_j)) \wedge \prod_{l \in L-J} (C_l \wedge (\neg C_l)) \end{gathered}$$

よって $J - L = L - J = \phi$の場合にのみ $C_{iJ} \wedge C_{kL}$ が真となる。

補題 2 ルール $R_{iJ}(C_{iJ} \longrightarrow A_i)$とルール $R_{kL}(C_{kL} \longrightarrow A_k)$が両立し，ルール R_{kL}とルール $R_{mN}(C_{mN} \longrightarrow A_m)$ が両立するならばルール集合 $S = \{R_{iJ}, R_{kL}, R_{mN}\}$ は両立する。

証明 補題1より, $J = L$, $L = N$が言える。よって$J = L = N$となる。同様に$i, k, m \in J$となる。よって

$$\begin{gathered} C_{iJ} \wedge C_{kL} \wedge C_{mN} \equiv \\ C_i \wedge C_k \wedge C_m \wedge \prod_{j \in J} C_j \wedge \prod_{m \in I-J} (\neg C_m) \equiv \\ C_i \wedge C_k \wedge \prod_{j \in J} C_j \wedge \prod_{m \in I-J} (\neg C_m) \equiv \\ C_{iJ} \wedge C_{kL} \end{gathered}$$

仮定より $C_{iJ} \wedge C_{kL} \wedge C_{mN}$は真となりうる。よって$S$は両立する。

補題 3 ルール集合 $S = \{R_{iJ} \mid \langle i, J \rangle \in I_s\}$ があったとする。Sの要素を点とし, Sの中で両立しているルール同士を弧で結ぶ。このグラフが連結ならば S全体も両立する。

証明 補題2から明らか。

4 定理

定理 1 任意のルール集合 Sとその上の優先順位 Pが与えられたとする。Sと等価なルール集合 S'と Pと等価な S'上の優先順位 P'が存在し, S'を点, P'を弧とした有向グラフが無閉路的となる。

証明 定義7, 8に従ってルール集合と優先順位を作る。これは元のルール，優先順位と等価になる。後はこの優先順位によるグラフが無閉路的であることを言えばよい。背理法を用いる。閉路が存在したとする。その閉路の上のルールの集合を Sとする。Sの上の優先順位のグラフは閉路となっている。よってこのグラフは連結であり，補題3より Sは両立する。基本原理より両立するルール集合においては優先順位は線型順序となる。これは優先順位が閉路をなすことと矛盾する。よってこのグラフは無閉路的となる。

系 1 任意のルール集合 Sとその上の優先順位 Pが与えられたとする。Sと等価なルール集合 S'と Pと等価な S'上の優先順位 P'が存在し, P'が順序関係となる。

証明 定理1と有向グラフの性質より明らか。

補題 4 集合Sの上に順序関係Rが定義されていたとする。$\{S_i \mid i \in I\}$をSの集合としての分割とする。RがS_iの上では線型順序であり，

$$(\forall x)(\forall y)((x \in S_i \wedge y \in S_j \wedge (x, y) \in R) \Rightarrow i = j)$$

が成立するとき, S上に Rを含む線型順序 R'が存在する。

証明 分割$\{S_i\}$の添字集合Iを整列させて順序を入れておく。R'を次のように定義する。

$$(x, y) \in R' \equiv \begin{cases} (x, y) \in R & (x \in S_i, y \in S_i) \\ i < j & (x \in S_i, y \in S_j, i \neq j) \end{cases}$$

Rが S_iの上では線型順序であることから, R'は線型順序となる。またもうひとつの条件から $R \subset R'$となる。

定理 2 任意のルール集合 Sとその上の優先順位 Pが与えられたとする。Sと等価なルール集合 S'と Pと等価な S'上の優先順位 P'が存在し, P'が線型順序となる。

証明 系1と補題4から系1で作成した優先順位 P'が線型順序となる。

5 おわりに

基本原理と自然な拡張を用いて，任意の優先順位を等価な線型順序に置き換えることが出来た。これによって自然数の値だけで任意の優先順位を記述することが出来るようになった。逆に Order Graph のループをチェックすることによって，その Order が正しく設定されているかどうか確かめることが出来る。またこれを用いると，局面によって発火すべきルールが一意に定まるので競合解消ルール自身が不要となる。

競合解消ルールとしてどのようなものが本当にコーディングしやすく，システムを旨く動作させることが出来るか，ということは今後の検討課題である。

参考文献

[1] Brownston, *Programming expert systems in OPS5*, Addison-Wesley (1985).

[2] Forgy, *The OPS83 User's Manual*, Production Technologies, Inc. (1986).

[3] Rosenthal, Adding Meta Rules to OPS5, *SIGPLAN Notices* *20*, 10 (1985), 173–180.

[4] 安次富他, ＤＢＭＳを兼ねるＬＩＳＰシステム上のエキスパートシステムの構成, 情報処理学会第 *35* 回全国大会講演論文集 (1987), 477–478 (7Cc-10).

[5] 黒瀬他, ＤＢＭＳ上のスキーマ診断システムＧＳＣ, 情報処理学会研究報告 88-DBS-65 (1988).

[6] 戸沢, プログラミング言語としてのＯＰＳ５, 情報処理学会研究報告 86-AI-45 (1986).

[7] 山岡, ＯＰＳ５のプログラミングパラダイムを使った生産計画シミュレーション, 情報処理学会研究報告 87-AI-54 (1987).

[8] 渡辺他, ＣＬにおけるルール指向プログラミング, 情報処理学会研究報告 86-AI-46 (1986).