

## 時間関係と因果関係を融合して扱う プロダクション・システム

(財)電力中央研究所)

篠原靖志 · 寺野隆雄

### 1. はじめに

電気事業に関連するエキスパート・システムの開発にあたっては、系統制御、プラント診断などいろいろな場面で時間概念を扱う必要が生ずる。その中には、わずかな観測事象をもとにその間に生じた諸事象の連なり全体を推定し、観測事象を矛盾なく説明するひとつのストーリーを再構成しなければならない性質のものがある。このようなシステムを開発するための要件は次のとおりである。

#### 1) 時間関係と因果関係の融合

部分的な時間関係と因果関係に基づく情報から、全体的な時間関係、因果関係を導かなければならない。また、過去、未来の情報を柔軟に記述でき、しかも、事象の因果関係を説明できなくてはならない。

#### 2) 事象の時間に関する半順序関係の利用

このようなシステムでは、たとえば、互いに因果関係のない2つの事象の前後関係を決定する必要はない。すなわち、個々の事象の生起に関する因果関係と時間的な半順序関係とから全体が説明できれば十分である。

#### 3) 専門知識の記述容易性

ドメイン・エキスパートが比較的容易に知識を記述できるものでなければならない。これには、[6]にもあるように、プロダクション・ルールに近い形で知識を記述することが適切である。

#### 4) 診断機能

対象システムの診断は、上で述べたようなストーリーに基づいて、それを説明することで行なう。なお、実際に生ずるような事象の診断を自動化するには、専門家のヒューリスティクスに加えて、対象とする分野の深い知識(deep knowledge)[4]を利用しなければならない。これをシステムに取り込めるようにしておく必要もある。

具体的なエキスパート・システムでも医療分野を中心に、いくつか時間関係を扱うものが知られている。しかし、これらは対象分野の性格上、隣合った時間を離散的に扱うだけのものが多い[3]、[7]。一方、最近、これとは別の方式で時間概念を扱うシステムがいくつか提案されている[1]、[5]、[8]、[9]。Templog[9]はPrologを時間論理に拡張して扱う。柳田等[8]は、Uranusの多重世界機構を利用してtree状の(潜在的に可能な)時間軸を取り扱える方法を示している。Allen[1]、Vilain[5]等は、時間区間を推論処理の基本とするシステムを提案しており、さらに[1]ではそのために比較的

効率のよい算法を示している。

本稿では、これらの研究を発展させ、上の要件を満たすべく試行中の、時間関係と因果関係を融合して扱うシステムについてその基本的な考え方を述べる。

## 2. 対象システムの状態と時間の表現

対象システムの属性は、システムを構成する各構成機器、または、構成機器間の属性とその取り得る値との組で次のように表わす。各属性の取り得る値はユニークとし、I/NIL、数値、ストリングいずれかの型をもつ。

(属性 機器 値) ; (属性 (機器1 機器2 . . . 機器n) 値)

以下では、属性、機器の組を述語と呼び、上の表現を(述語 値)または(P V)などと書く。

また、本システムでは時間情報は、[1]と同様にすべて時間区間として表わす。そして、対象システムの状態は、上で示した属性とその属性が成立する時間区間のリスト、成立しない時間区間のリストの組で次のように表現する。

(述語 値 (成立区間1 成立区間2 . . . )  
(不成立区間1 不成立区間2 . . . ))

以下では、これを(P V I not-I)などと書く。

この表現法をとることにより、”スイッチAが区間T1とT2でonであり T3 で offである”というシステムの状態は、次のように記述される。

(ON スイッチA T (T1 T2) (T3))

## 3. 知識ベース中のデータの表現法

知識ベースに格納するデータは、事実データ、時間関係データ、状態内ルール、時間依存ルール、の4種類である。このうち状態内ルール、時間依存ルールの2つは後述するシナリオから生成することも可能である。

事実データは対象システムの正常な状態もしくは機器の特性をあらわすデータの集合で、(P V I not-I c r)の形式である。ここでcとrはI/NILの値をもち、後述する推論の制御に用いる。

状態内ルールは時間については恒等的に成立するルールである。したがって状態内ルールには時間に関する情報を明記しない。すなわち、対象システムの状態を表わす項(P V I not-I)を(P V)などと略記する。状態の否定、連言は次のように解釈される。ただし、連言については状態内ルールの左辺(IF部)のみに記述できるものとする。

状態記述 : (P V) <=> (P V I not-I)

否定 : (P not-V) <=> (P V not-I I)

連言 : (P1 V1) & (P2 V2)

$$\langle = \rangle \quad (P1 \quad V1 \quad P2 \quad V2 \quad I1 \& I2 \quad \text{not-}I1 \mid \text{not-}I2)$$

状態内ルールは次のように記述され解釈される。

$IF(P \ V) \ THEN \ (Q \ W) \quad \langle = \rangle \quad IF(P \ V \ I \ \text{not-}I) \ THEN \ (Q \ W \ I \ ())$   
 $IF(\text{NOT} \ (P \ V)) \ THEN \ (Q \ W) \quad \langle = \rangle \quad IF(P \ V \ I \ \text{not-}I) \ THEN \ (Q \ W \ \text{not-}I \ ())$   
 $IF(P1 \ V1) \& (P2 \ V2) \ THEN \ (Q \ W)$   
 $\quad \langle = \rangle \quad IF(P1 \ V1 \ I1 \ \text{not-}I1) \& (P2 \ V2 \ I2 \ \text{not-}I2) \ THEN \ (Q \ W \ I1 \& I2 \ ())$   
 $IF(P1 \ V1) \mid (P2 \ V2) \ THEN \ (Q \ W)$   
 $\quad \langle = \rangle \quad IF(P1 \ V1 \ I1 \ \text{not-}I1) \& (P2 \ V2 \ I2 \ \text{not-}I2) \ THEN \ (Q \ W \ I1 \mid I2 \ ())$   
 $IF(P \ V) \ THEN \ (Q1 \ W1) \& (Q2 \ W2)$   
 $\quad \langle = \rangle \quad IF(P \ V \ I \ \text{not-}I) \ THEN \ (Q1 \ W1 \ I \ ()) \& (Q2 \ W2 \ I \ ())$   
 $IF(P \ V) \ THEN \ \text{NOT}(Q \ W) \quad \langle = \rangle \quad IF \ (P \ V \ I \ \text{not-}I) \ THEN \ (Q \ W \ () \ I)$

このような状態内ルールの適用例を図1に示す。ここでは、(Q W)の状態が、3つの状態内ルールの適用によって時間区間上で明確になっていく様子を示している。この表現・解釈からわかるとおり、本システムでは述語は、真、偽、不明の3つの値をとる三値論理の体系で考えることとなる。

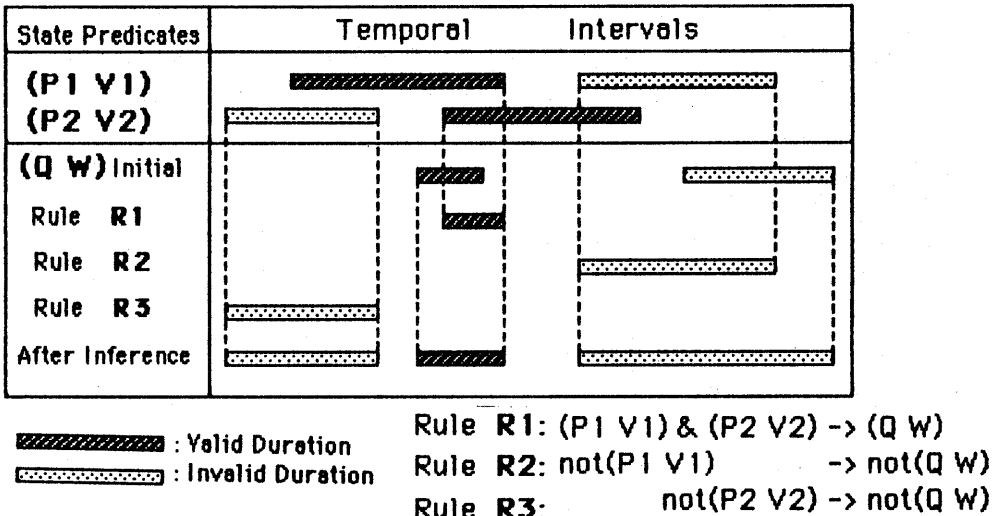


図1 状態内ルールの適用例

本システムでは、推論処理の過程で作られる対象事象全体を説明するストーリーは、時間区間をノードに、その間の半順序関係をアークに対応させたネットワークで表現される[1]。時間依存ルールは対象システムの時間的な状態を変更したり、新たな、時間区間を作成したりするのに使われる。すなわち、時間依存ルールの適用によって、対象となるストーリーはラフなものから徐々に精密なものへと変化していくことになる。利用者は、ルールまたは、シナリオの形で時間区間の関係と事象の因果関係の情報を記述

できるようになっている。時間依存ルールは次の記述項目をもつ。

認識部 (if部)

ルール名称、想定シナリオ名称：ルール名とそれをつくるもとになった事象の断片 (シナリオ) の名称を記述する。

状態認識部： マッチングすべき状態を示す。形式は、2節で述べた状態の記述と同一である。これにより、シナリオにマッチングする時間区間の候補を選び出す。

時間区間指定部： 上で認識された状態に現われる時間区間のうちシナリオ中の時間関係を満足する時間区間の組を定める。

・ VALID-DURATION, INVALID-DURATION : 指定したの時間区間変数の取り得る長さ、取り得ない長さを記述する。この情報によって、マッチングすべき区間の絞り込みを行なう。

・ VALID-TIME-RELATION, INVALID-TIME-RELATION : 指定した2つの時間区間変数の間の妥当な関係、妥当でない関係を記述する。2区間同士の関係は[1]にあるように before, equals, meets, overlaps, during, starts, finishes およびそれらの逆の関係13とおりで表わせる。ただし、これだけでは、記述が繁雑となるので、2つの区間の共通部分や包含関係を指定できるマクロ機能を備えている。

関数実行部： EXECUTE-PREDICATE, EXECUTE-FUNCTION の指定により、任意のLISP述語、関数を実行することができる。これらはおもに、利用者との質問応答に使用される。

実行部 (then部)

時間区間生成部：

・ MAKE-INTERVAL: 区間変数名を生成し、その区間の長さを指定する。

・ ASSERT-TIME-RELATION: 2つの区間変数名の間の関係を時間区間認識部と同様な方法で指定する。

状態定義部： 新しい状態を2節で述べた形式でASSERTする。

関数実行部： 認識部と同様に、任意のLISP述語、関数を実行するために使われる。

ブザーの例を考える。今 "振動板が動くのは電磁石に電流が流れるときである。" というシナリオと "スイッチAをがはいっているとき、ブザーの振動板が動き、電磁石への電流はとぎれる。" というシナリオとが与えられたとする。これに対応するルールはたとえば以下のように書くことができる。

(RULE-1 SCENARIO-1

(IF (ON VIBRATOR T \*I1 ()))

(ON CURRENT-OF-MAGNET T \*I2 ()))

(VALID-TIME-RELATION (\*I1 \*I2)(EQUAL)))

(THEN (ASSERT-TIME-RELATION (\*I1 \*I2)(EQUAL))))

```

(RULE-2 SCENARIO-2
  (IF (ON SWITCH-A T *I1 ()))
    (ON CURRENT-OF-MAGNET T ( ) *I4)
    (VALID-TIME-RELATION (*I1 *I4)(CONTAINS)))
  (THEN (MAKE-INTERVAL *I3)
    (MAKE-INTERVAL *I5)
    (ASSERT-TIME-RELATION (*I3 *I5)(BEFORE))
    (ASSERT-TIME-RELATION (*I5 *I4)(BEFORE))
    (ASSERT-TIME-RELATION (*I3 *I4)(BEFORE))
    (ASSERT-TIME-RELATION (*I3 *I1)(STARTS))
    (ASSERT-TIME-RELATION (*I1 *I4)(CONTAINS))
    (ON CURRENT-OF-MAGNET T *I3 ( ))
    (ON VIBRATOR T ( ) *I5)))

```

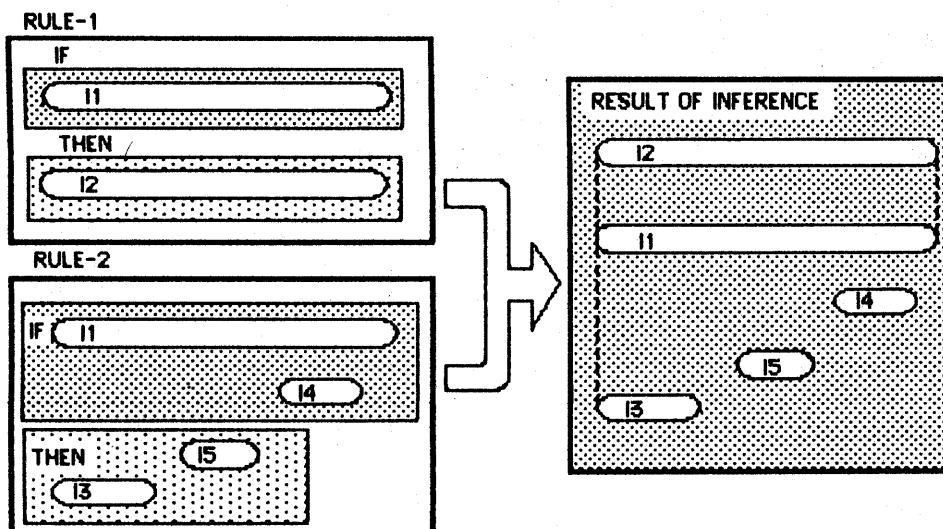


図2 時間依存ルールの適用

上の2つのルールとそれらを順次実行したときの時間ネットワークの変化は図2のようになる。本システムでは全体的な時間関係のうち、事象の半順序関係を規定する部分のみを柔軟に変更しながら処理を進めることができるので、Tree状の時間を扱うより自由度が大きく、しかも比較的効率がよいという特徴をもつ。ただし、ルールには、事象の時間関係と、因果関係とが混在しているため、推論処理の実行順序と事象の因果関係、時間関係はでたらめなものとなる。次節でこれを制御する機構について述べる。

#### 4. 推論処理の方式

本システムの推論機構は、推論を起動するトリガーとなる与えられた状態に対して、知識ベースの事実、ルールを適用し、事象の連なり全体を説明できる解がひとつ求まれば停止する。そして途中で処理に失敗した場合には、Prolog風の後戻り制御を行なう。そのため、いわゆるTruth Maintenance System[2]ほど複雑な処理は行なわない。

トリガー状態は対象システムについて外部から与えられる観測データであり、事実と同様に (P V I not-I \*c \*r) の形に記述する。ここで\*cはトリガー状態の原因を知りたい場合にTに、\*rは波及効果を知りたい場合にTにセットする。トリガー状態の\*c, \*rの値は少なくともいずれかがTでなければならない。

推論処理の制御方向は、推論処理開始時に定める処理モード、ならびに、トリガー状態の\*c, \*rの値、ならびに、ルール中に明記する\*c,\*rの情報にしたがう。(ルール中に明記する\*c, \*rの情報については、議論を簡単にするために、3節では記述と説明を省略したので注意されたい。) すなわち、処理モードが原因探求型のときは\*cのフラグがTのルールが起動され、処理モードが波及予測型のときは\*rの値がTのルールが起動される。。また、処理モードが両方向のときは、どのルールも使用される。推論処理の途中で生成される状態の\*c, \*rの値は知識ベースもしくはルールに上で示した制御方向によって決定する。制御方向による中間状態の\*c, \*rの値の決定法は下の表1にしたがう。また、推論過程におけるルールの適用条件は表2によって決定される。

表1 制御方向による\*c, \*r の値

| 制御方向      | *C    | *R    |
|-----------|-------|-------|
| C : 原因探求型 | T     | N I L |
| R : 波及予測型 | N I L | T     |
| 両方向型      | T     | T     |

表2 ルールの適用条件

| トリガー状態 | 適用可能ルールIF部    |
|--------|---------------|
| *c = T | c または r c = T |
| *r = T | r または r c = T |

時間ネットワークを処理する算法は、[1]で提案されているAllenの算法とほぼ同様である。ただし、処理を効率化するために導入した参照区間(Reference Interval)の定義法は異なる。[1]では、Duringの関係にあるより大きな時間区間を便宜的に参照区間に定めて推論処理を実行するのに対し、本システムでは、時間依存ルールを記述するもとなった事象のシナリオからDuring関係にある参照区間を原則的に選ぶこととしている。現在のバージョンではこの部分の自動選択は行なっていないが、次節で述べるように、事象のシナリオの概念を有効に利用することにより、これを自動化し、さらに

時間関係の推論処理効率を向上させることができる。

推論処理の算法の概略は次のとおりである。

```
WHILE ((トリガーとなりうる状態が存在する) &
        (適用可能な時間依存ルールが存在する))
  DO [(時間依存ルールを適用する);
       IF (知識ベースに矛盾が生じる)
         THEN (バックトラックする) ENDIF ;
       WHILE (適用可能な状態内ルールが存在する)
         DO {(状態内ルールを適用する);
              IF (知識ベースに矛盾が生じる)
                THEN (バックトラックする) ENDIF } ]
  IF (トリガーとなりうる状態が存在しない)      THEN (成功;EXIT)
  ELSE
    /*適用可能な時間依存ルール、状態内ルールがなくなった*/
    { 初期設定してあるモードにしたがって以下のいずれかの処置をとる:
      (1) アペンド・モード: (そのときのトリガー状態を表示して停止する)
      (2) バックトラック・モード: (他の可能性を探してバックトラックする) }
```

## 5. シナリオの概念の利用方法

本システムの推論方式では、ストーリーは時間順序にも因果関係にも無関係に作成されていく。したがって、推論に使用したルールを順次利用者に提示するという通常の説明機能では推論の内容を理解することが難しくなる。そこで、ルール作成のもとになるシナリオごとに説明文の準備しておいて、推論処理終了後それを提示する方法を採用している。

また、3節の2つのルールで例示したように、本システムの時間依存ルールの記述方法は単純な自然言語のシナリオに対してもかなり複雑である。さらに、どのような推論を行なうかによって、同一のシナリオからいくとおりものルールを作成しなければならない。そこで、シナリオを、時間関係を明記した事象・述語の列で同時に因果関係の情報をもつより厳密な形で定義しておけば、これを前処理することによって、プロダクション・ルールに展開することが可能になる。この機能については現在設計を進めている段階である。

さらに、シナリオ中の重要な時間区間を参照区間とすることで、シナリオをルールに展開した後の時間関係の推論処理を効率化することが可能である。これについては、Allen[1]の算法の改良も含めて検討中である。

## 6. おわりに

本稿ではルール中に明記した時間関係と因果関係を利用し、時間順序によらずに推論を行なうプロダクション・システムについて述べた。本システムは、実際のエキスパート・システムへの適用を目指して試行中である。

本システムは、ルール・システムとしてはかなり複雑になるため、現在のままではドメイン・エキスパートが自らシステムを使いこなすのは難しい。エキスパート・システムに共通の問題であるマンマシン・インタフェースの改善[4]が必要である。また、本システムは時間推論の処理が複雑なため実行速度が遅いという欠点をもつ。時間関係を扱うための算法を改善することも今後の課題である。ただし、本システムの推論方式はかなり柔軟であり、一般の物語生成システムの推論エンジンとしても利用可能であると考えている。

## 参考文献

- [1] Allen, J.F.: "Maintaining Knowledge about Temporal Interval." Comm. ACM, Vol. 26, No.11, pp. 832-843 (Nov. 1983).
- [2] Doyle, J.: "A Glimpse of Truth Maintenance." in Winston, P. H. et al. (eds.) "Artificial Intelligence: an MIT Perspective." Vol. 1, pp. 119-135 (1979).
- [3] Fagan, L.M., et al.: "Representation of Dynamic Clinical Knowledge: Measurement Interpretation in the Intensive Care Unit." Proc. 6th IJCAI, pp. 260-262 (1979).
- [4] Hayes-Roth, F., Waterman, D. A., and Lenat, D. B. (eds.): "Building Expert Systems." Addison-Wesley, 1983.
- [5] Vilain, M.: "A System for Reasoning about Time." Proc. AAAI-82, pp. 197-201 (1982).
- [6] Weiss, S. M., and Kulikowski, C.A.: "A Practical Guide to Designing Expert Systems." Chapman and Hall, 1983.
- [7] 小山照夫, 開原成充: "時間経過の概念を含む汎用医療コンサルテーション・システム." 情報処理学会論文誌, Vol. 23, No. 4, pp. 414-420 (1982).
- [8] 柳田昌宏, 中川裕志: "多重世界による時間概念の表現法." 情報処理学会ソフトウェア基礎論研究会資料12-2, 1985年2月.
- [9] 米崎直樹, 新 淳, 蓬萊尚幸: "時区間を基礎とする時間論理プログラミング." 情報処理学会知識工学と人工知能研究会資料39-11, 1985年3月.