知識処理のための推論高速化技法

増位　庄一

1. はじめに

「計算機に人間並みの高度で高速な問題解決能力を与える」ことは人工知能研究[1]の大きな目標のひとつである．「人間並みに高度な」問題解決能力を実現する方式として，「曖昧推論」「時制推論」「非単調推論」等種々の推論方式[2]が提案されており，これらはそれぞれ本特集においても詳しく解説されている．本稿では，これらの高度推論方式の研究とならんで重要と思われる，「人間並みに高速な」知識の処理を実現する技法について解説する．

2. 知識工学における探索問題

人工知能(Artificial Intelligence:AI)技術はその源を広大な状態空間中での解の探索技術に発している点でオペレーションズリサーチ(OR)の技術と同種である．AIの代表的かつ古典的問題であるゲームは，そのままORの問題でもある．両者の違いは，AIが専らアルゴリズムが見つけにくい大規模な問題に対する発見的（ヒューリスティック）解法[3]を追求したのに対し，ORが数理的解法の発見を第一義とした点にある．誤解を恐れずに言えば，目的関数や制約条件が数式として明確に定められる探索問題はORの受け待ちで，それ以外がAIの範疇であるといえる．このため人工知能の基礎は一般的で高速な問題解決方式の発見にあるとされ，探索木の爆発的成長を回避して効率的に解を探索する方法の開発が種々の角度から試みられた．1960年代のGPS (General Problem Solver)[4]やPLANNER[5]等はその試みの代表例である．

この人間の問題解決の一般的過程を法則化しようとする試みは，探索技術を大きく発達させたが，求めていた法則の発見には至らないまま，「知識こそが知能の力の根源である」とする知識工学[6]に人工知能研究の主流の座を明け渡すことになった．知識工学は人間のもつ知識を計算機上に具体化し，それを用いることにより従来の方法では非効率にしか行なえなかった解の探索を飛躍的に効率化しようとする．囲碁において「囲碁の規則」と「打ち方」だけを与えた場合と，さらにそれに加えて「定石集」を知識として与えた場合を想定すれば，後者に相当する知識工学の狙いと従来技術との相違点を理解していただけると思う．そしてこの「定石集」のようなその道の専門家（エキスパート）の知識を計算機上で取り扱えるようにしたシステムがエキスパートシステム[7]である．

この知識工学は，しかし，新たな「探索」の問題をもたらした．エキスパートシステムにおいて使われる知識は，ある状況においてのみ有効なきわめて局所性の高いものである．このため現時点ではどの知識が適用可能であるかを常に見きわめ

ますい　しょういち　日立製作所　システム開発研究所
〒215 川崎市麻生区王禅寺1099

ておく必要がある．上記の「囲碁」の場合でいうと，現在の局面で使える定石はどれかを各局面毎に調べておかねば，せっかくの「定石」の知識を活かすことができない．そして知識の量が増加すればするほどこの知識の適用可能性に関する「探索」には手間がかかり，このことにより知識の利用による解探索の効率化という知識工学の狙いが達成できなくなってしまう可能性もあった．

ルール1：：IF　夕焼け　THEN　明日は晴れ．

ルール2：：IF　Xは晴れ　THEN　Xには傘不要．

図 1　簡単なルールの例（Xは変数で任意の文字列にマッチする．この例ではX＝明日になる）

3. 知識処理の高速化の考え方

エキスパートシステムは，専門家の知識をうまく利用することで，問題解決の効率化を狙うものである．そのためには専門家の知識を，専門家と計算機が同時に理解できる形で記述できなければならない．このための知識表現としてプロダクションルール[3]，フレーム[9]，述語論理[10]等がある．プロダクションルールは，専門家のもつ経験的知識を，**図1**に示すようなIF～THEN～形式で表現するもので，(1)専門家の知識が表現しやすい，(2)ルールは相互に独立した形で記述できるため，その変更や追加が容易である，ことから広く利用されている．

プロダクションルールは，そのIF部に記述された条件がすべて成立すれば，THEN部の結論が導けることを意味する知識である．図1では，「夕焼け」が出れば「明日は晴れる」，「晴れた日」ならば「傘は要らない」という2つの知識がルール化されている．ルールによる推論とは，「夕焼け」を見た時に「明日は傘が要らない」という結論を導くことであり，**図2**に示すように「夕焼け」→→「晴れ」→「傘不要」という推論連鎖を形成することである．この推論連鎖の形成には，前述のように現在得られている「事実(「夕焼け」)」に関してどのルールの条件部が成立しているかを照合し，成立しているルールの結論を新しい「事実」として加えることが必要である．この追加により「晴れ」が新しい事実となり，連鎖が形成できる．

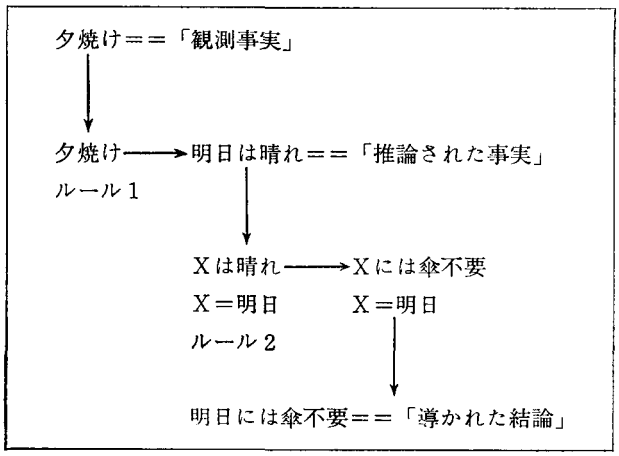

図 2　ルールによる推論連鎖（観測された事実から，中間的な事実，結論をルールを用いて導く）

ルール表現にもとづくエキスパートシステムの推論処理で最も時間のかかるのは，条件部の成立を判定する照合の過程である．処理負荷の90%以上がこの照合処理に費やされるといわれている．これはルールの条件毎にデータと照合し，その成否を調べなければならないためで，当然のことながらルールの数に比例してその処理時間が増大する．これを解決するためには，(1)ルールは相互に独立しており，その条件部は並列照合可能であるという性質を利用して，照合過程を並列処理する，(2)無駄な照合を避け，照合の回数をできる限り削減する，という2つの方策が考えられる．前者は，ICOTで研究が進められている並列推論マシーン[11]がその代表的な研究例である．そのほかにもプロダクションシステムのルールを直接実行する専用マシーン[12][13]の研究が米国を中心に盛んに行なわれている．後者に関しては，どのようにして無駄な照合処理を削減するか

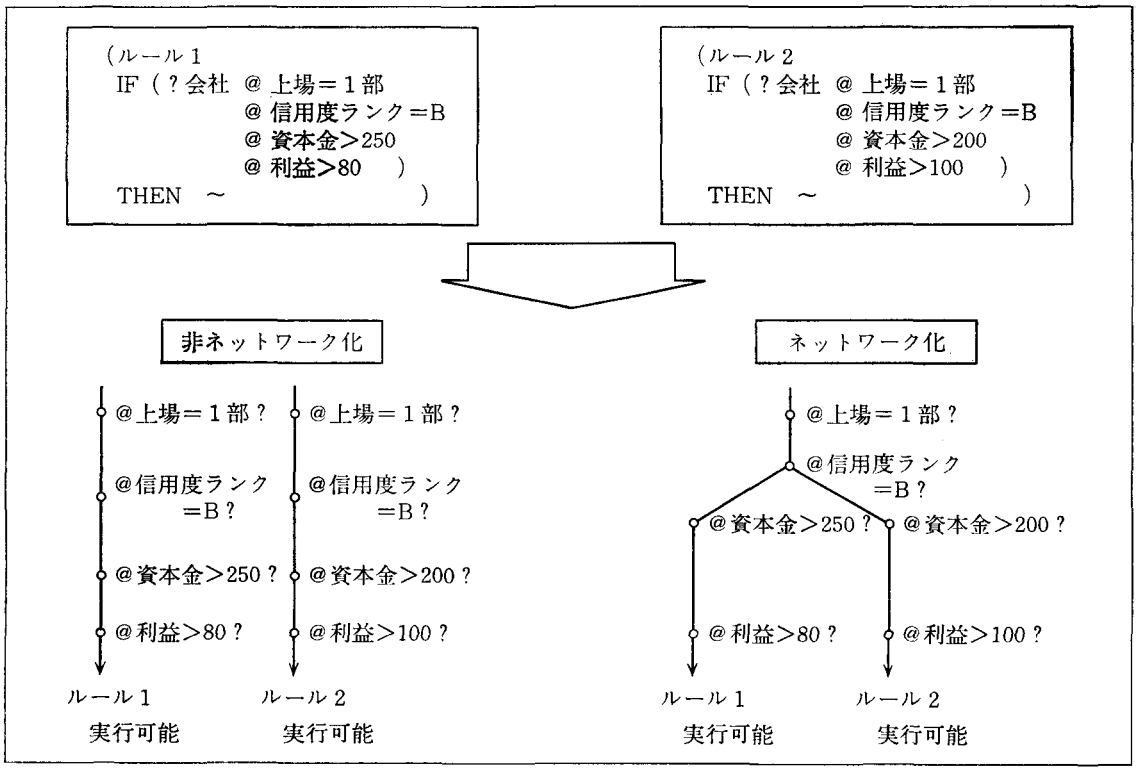

図 3 ルール条件部のサブネット化と重複部分の共通化

が重要な課題になり，(1)フィルターにより，照合すべきデータをふるいわける，(2)ルールを分割し，照合すべき条件数を減らす，(3)メタルールによって照合範囲を限定する，等が主要な方策となる．しかしこれらは知識を記述する場合にあらかじめ知識の構造が整理できるという考え方にもとづくものであり，知識が明示的に構造化できない場合の有効性は低い．知識処理に内在する無駄な照合を削減するという観点からの高速化の手法としては，米国カーネギーメロン大学の，C. L. FORGY が提唱した RETE アルゴリズム[14]が著名である．これは，ルールの条件部をネットワーク化し照合を効率的に行なうもので，(1)文字列操作をビット操作に変換し，比較演算を簡易化する，(2)ルールの条件部とデータとの比較において，変化したデータのみに関する照合だけをやり直すことで，処理の高速化を実現しようとするものである．

4. RETE アルゴリズムの概要

RETEアルゴリズムは，**図3**に示すように，ルールの条件部における1つ1つのデータ照合，たとえば「会社の上場が一部であるか？」「会社の信用度ランクはBであるか？」等を，それぞれ判定ノードとして独立させ，そのノードの連結によってルールの条件部を表現する．そしてある条件に対応する連結ノード（以下ではサブネットという）をすべて通過できるデータが存在する時，その条件が成立していると考える．このアルゴリズムでの照合は，「このデータは，どのルール条件に対応するサブネットを通過できるか？」というデータを中心にした方式で行なわれる．この RETE アルゴリズムでの処理高速化は，次の2つの考え方にもとづいている．

(1) ルール条件部の重複部分を共通化する．

ルールの条件部は，複数のルールで重複してい

る場合が多い．ルールの条件部を内部表現に変換する時，図3に示したように，この重複した部分をネットワーク化することにより共有させれば判定ノード数が減り処理効率があがる．

(2) 変化したデータに関してのみ再照合する．

1つのルールを実行しても変化するデータは少なく，このデータの変化によって成否が変更される条件部はそれほど多くない．したがって毎回すべてのルール条件部とデータの比較判定をすることは無駄である．変化したデータに関係するルール条件部だけについて判定処理すればこの無駄がなくなり，特に大規模ルールの場合には処理効率が飛躍的にあがる．これを可能にするには，どのデータがどのサブネットを満たしているかというデータ変化前の状態を覚えておく必要がある．**図4**の例では，資本金が250億以上で株式市場1部上場のA社の利益が，資本金が240億以上で2部上場のB社の利益より大きい場合にはルール1を，逆に少ない場合はルール2を実行するという2つのルールが示されている．

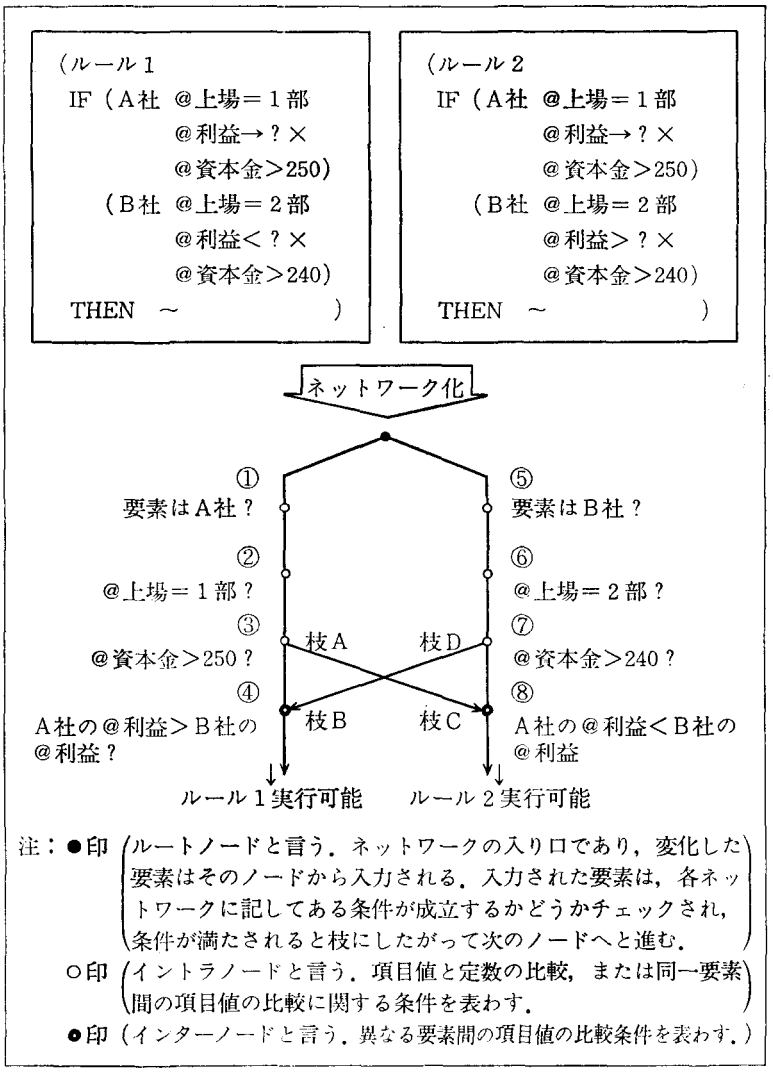

図4 変化したデータのみの再照合による高速化

この例では，A社に関するサブネットとB社に関するサブネットは，両者の利益を相互に比較するノード（◐の部分）において接続される．A社のデータは①②③の順に判定処理を受け，ノード④およびノード⑧において⑤⑥⑦の判定処理を受けたB社のデータと比較判定される．RETEアルゴリズムでは，単純判定を行なう○のノード（ノード①，ノード②等）をイントラノードと呼び，異なるサブネット間のデータの比較を行なう◐（ノード④，ノード⑧）をインタノードと呼ぶ．このインタノードに至るまでの判定処理をパスしたデータをインタノードに入る各枝（図4の例では，A，B，C，Dの枝）に記憶しておけば，変化したデータに関するサブネットの判定処理を行なうだけで相互の比較が可能となり，処理効率をあげられる．たとえば図4でA社の利益データのみが変わった場合，社Aに関してのサブネット，すなわち①②③の条件判定をするだけで，B社に

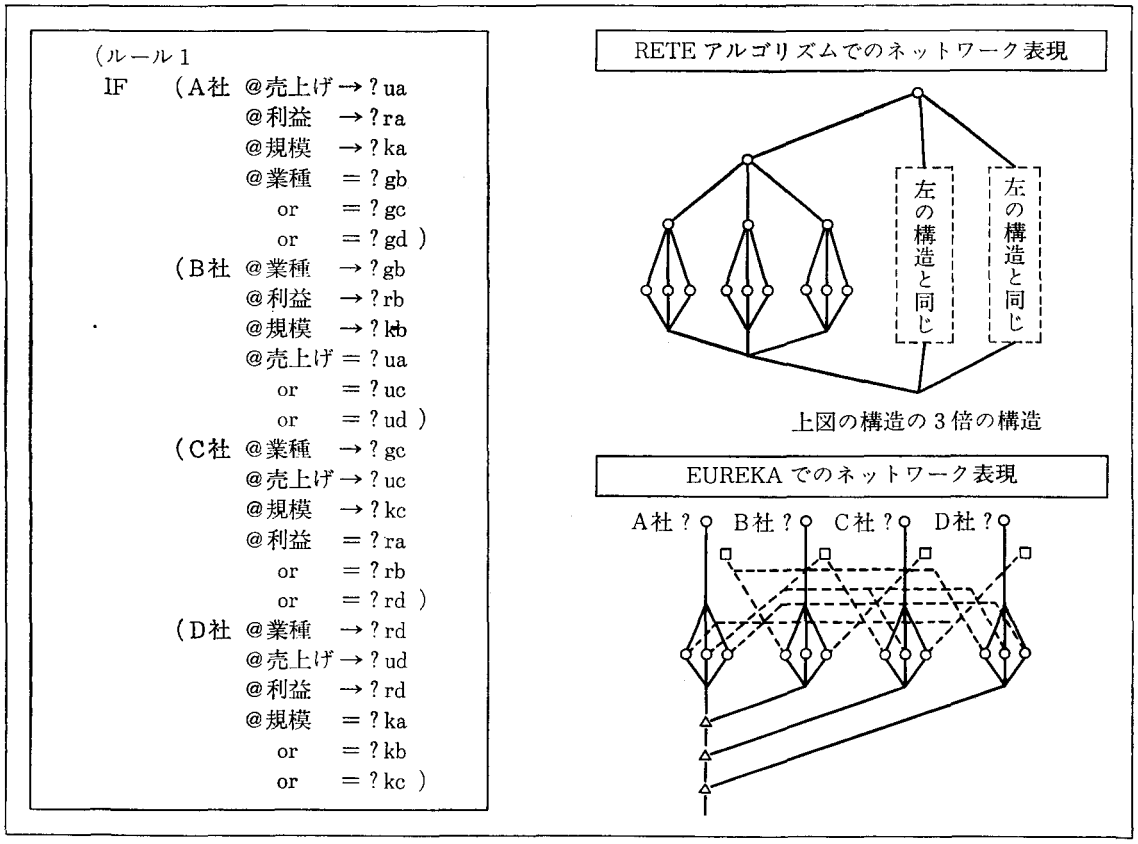

図 5 複雑な条件に対するネットワーク表現の比較

関する条件判定⑤⑥⑦を行なうことなく，ノード④およびノード⑧において相互のデータの比較ができる．

5. EUREKA の推論高速化方式

5.1 EUREKA での高速化の考え方

ここでは，ビジネス応用に適した複雑で抽象的記述が可能で，かつシステム制御等のリアルタイム制御に対応できるツールとしてわれわれが開発したエキスパートシステム構築ツールEUREKAの推論高速化の考え方について述べる．EUREKAは，前述のRETEアルゴリズムと同じネットワーク型のデータ中心の照合処理を高速化の基本方式としているが，それに加えて，ルールのネットワーク表現方法，ネットワーク処理の効率化方式等に工夫を加え，さらに数段の高速化を達成している．

(1) EUREKAのネットワーク表現[15]

RETEアルゴリズムでは，異なるサブネットの比較においては，直接インタノードでサブネット同士を結合している．このため比較が複雑になると，インタノードにおける処理がきわめて増加するとともにネットワーク自身が肥大化し，処理効率がいちじるしく落ちるという問題がある．そこで，EUREKAでは1つ1つのサブネットは独立に評価し，相互の比較が必要な場合は，各サブネットに付随させた仮想的な「候補ノード」を介して行ない，サブネット同士は，比較処理後にマージノードで結合するという新しいネットワーク構成を考案した．すなわち，比較処理と結合処理を分離し，**図5**に示すような複雑なルールの条件部の評価においてもネットワークの肥大化を避ける

ようにした．この方式は条件部が複雑化するほど効果が大きく，ビジネス分野のような抽象的で相互関連の複雑な知識が多い場合に特に適した処理方式である．

(2) ネットワーク処理の効率化 [16]

EUREKAではサブネットの処理量を低減するため，(a)ルールの重複部分の共通化，(b)ネットワーク中のノードの処理量の評価にもとづくノードの入れ替え，を行なっている．(b)は，サブネット上の各ノードの位置，すなわちノード判定の順序を変更することにより，推論実行時のサブネットの処理量の削減を図るものである．この考え方の基本は，処理中のデータがそのサブネットを通過できるか否かをできるだけ早く判定しようとするもので，処理の簡単なノード，制約の強いノード（判定が明確に行なえるノード）が優先処理されるように並べかえる．具体的には，処理が複雑なインタノードよりも簡単に判定ができるイントラノードが，分岐処理が複雑なOR結合よりも制約が強いAND結合が先に処理されるようにサブネット内のノードの位置を入れ替える．**図6**の例でA社の信用度がAからBに変化した場合，並べかえ前のサブネットでは，すべてのノードの判定処理が必要となるが，並べかえ後のサブネットではイントラノード④の処理のみを行なえば，サブネットが通過できないことが即座に判明する．

```
(ルール1
 IF  (A社 $利益   =?rb      …… ①
          or     >100       …… ②
      $売上げ=?ub            …… ③
      $信用度=A )            …… ④
     (B社 $利益   →?rb
      $売上げ→?ub )
 THEN ～                    )
```

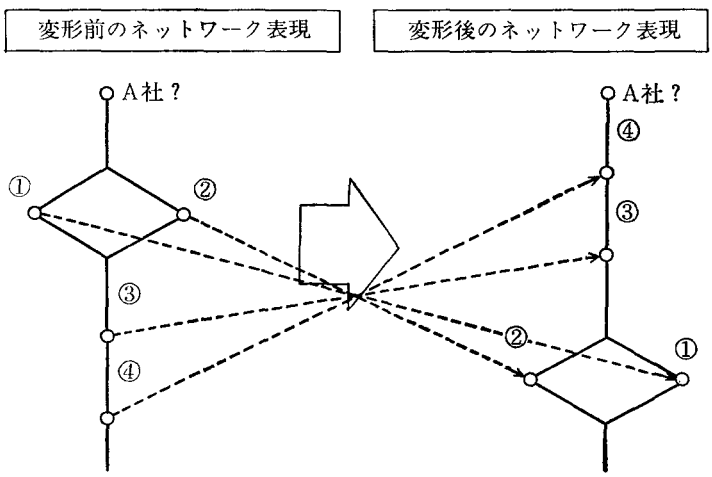

図 6 ノードの並べかえによるネットワーク処理の効率化

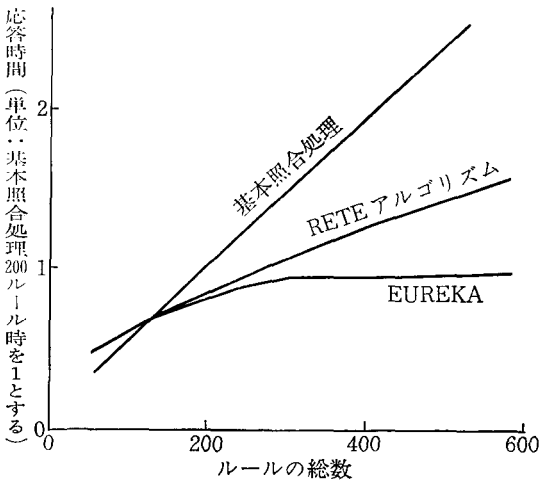

図 7 ルールの実行速度

5.2 EUREKAの処理速度の評価

EUREKAでは，前節で述べたような推論高速化方式により，基本的な推論処理およびRETEアルゴリズムによる推論処理に比べて，**図7**に示すような高速化を実現した．これは実用的なシステムによく見られる，条件部が複雑化しながらルール数が増加する場合にもEUREKAが十分対応可能であることを示している．すなわちEUREKAの特長は，ルール総数に依存しない応答速度を実現したこと，および複雑な条件に対しても優れた応答性が確保できることにあり，これらは今後その開発が期待されている大規模なエキスパートシ

ステム構築のためのツールとして好適である．また今までは実現がむずかしかった組合せ計画（乗員スケジューリング等のOR問題）のエキスパートシステム化も可能となった．

6. おわりに

ルールを用いた知識処理の高速化方式について述べた．ソフトウェア上の工夫による高速化の基本は，無駄な照合を避けるという点にある．その意味で基本となるRETEアルゴリズムを紹介し，その問題点をさらに改良したEUREKAの処理方式を示した．このEUREKAの技術は，制御用コンピュータHIDIC V90/25およびエンジニアリングワークステーションES-330上のEUREKA-II，またビジネスアプリケーション向けにクリエーティブワークステーション2050上のES/KERNELといった製品に活かされ，実用化されている．エキスパートシステムが実用となるには，ツールの高速化がキーになることは広く認識されつつある．アメリカにおいては，LISPで書かれていた処理系をC言語で書き直すことにより高速化を図ることが盛んに行なわれている．また処理の並列化というハードウェアによる高速化計画も着実に進展しており，今後とも「人間並みな，あるいはそれ以上の高速な知識処理」の実現をめざした開発が各所で進められることになろう．

参考文献

[1] Feigenbaum, E. A. et al. 編：人工知能ハンドブック，共立出版（1983）

[2] 石塚 満(編)：高次人工知能に向けてのパラダイム，人工知能学会誌，Vol. 2, No. 1（1987）

[3] Nilsson, N. J. : Principles of Artificial Intelligence, Tioga Pub.（1980）

[4] Newell, A. and Simon, H. A. : GPS, A Program that Simulates Human Thought, in Computer and Thought, McGraw-Hill（1963）

[5] Hewitt, C. : PLANNER : A Language for Proving Theorems in Robot, IJCAI 1（1969）

[6] Feigenbaum, E. A. : The Art of Artifical Intelligence, Themes and Case Studies of Knowledge Engineering, IJCAI 5（1977）

[7] 上野晴樹：エキスパートシステム概論，情報処理，Vol. 28, No. 2（1987）

[8] 小林重信：プロダクションシステム，情報処理，Vol. 26, No. 12（1985）

[9] 小川 均：フレーム論理に基づく知識表現言語，情報処理，Vol. 26, No. 12（1985）

[10] 中島秀之：論理に基づく知識の表現，情報処理，Vol. 26, No. 12（1985）

[11] Murakami, K. et al. : Research on Parallel Machine Architecture for Fifth-Generation Computer Systems, Computer, Vol. 18, No. 6（1985）

[12] Gupta, A. et al. : Parallel Algorithms and Architectures for Rule-Based Systems, Proc. 13th Annual Int'l Symp. Computer Architecture, IEEE/ACM（1986）

[13] Stolfo, S. J. and D. P. Miranker : The DADO Production System Machine, J. Parallel and Distributed Computing, Vol. 3, No. 2（1986）

[14] Forgy, C. L. : RETE : A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, ARTIFICIALL INTELLIGENCE, Vol. 19（1982）

[15] 田野，他：知識処理ソフトウェアEUREKAにおける推論機構の高速化，第31回情報処理学会全国大会論文集（1985）

[16] 田野，他：知識処理ソフトウェアEUREKAにおけるルールネットワークの効率化方式，第32回情報処理学会全国大会論文集（1986）