TM-1150

法的推論システム HELIC-IIの開発と現状

新田　克巳、大獄　能久、前田　茂、
小野　昌之、大崎　宏、坂根　清和

January, 1992

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

法的推論システム HELIC-II の開発と現状

新田克己, 大嶽能久, 前田 茂
小野昌之, 大崎 宏, 坂根清和

1 はじめに

新世代コンピュータ技術開発機構 (ICOT) は、第5世代コンピュータプロジェクトの研究センターである。論理型言語をベースにした知識処理技術を、ハードウェアから、ソフトウェア技術、データベース技術、知識処理技術に至るまで幅広く開発してきている。ICOT で開発されている並列推論マシン (PIM) は、論理型言語 KL1 で書かれたプログラムを並列に実行することによって高速に推論を行う。
PIM の応用プログラムの1つとして開発されたのが、法的推論実験システム HELIC-II である。HELIC-II は「ヘリクツ」生成システムである。後で説明するように訴訟における論理展開の材料を提供することを目的とするからである。以下では、法的推論の概要とシステムの機能について説明する。

2 法的推論とは

2.1 法的推論と解釈

法律に関する知識と聞くと、すぐに法令文が思い浮かぶ。法令文の多くは「もし〜ならば〜である。」の形式をしているので、これを IF-THEN ルールや論理式で表現することができる。与えられたデータに対して、演繹的にルール／論理式を適用すれば、法的な結論が機械的に得られると考えられる。
しかし、現実にはこのような単純な推論で解決できる問題は非常に限られている。その原因の1つは「ルールの抽象性」にある。法令文はありとあらゆる紛争を細かく場合分けして、作られているわけではない。少ない数（とはいっても山のように法律はあるが）の法令文で広い範囲の問題を解決するためには、法令文はどうしても抽象的にならざるを得ない。そのため、法令文（ルール）を現実の問題に適用するとき、その事件がルールの条件部に合致するかどうかの判定が困難となる。このようなとき、法律家は法令文の意味をより詳細化し、その後に判定を行う、という手順をとる。法令文の意味を詳細化、明確化する作業が法律の「解釈」である。
解釈を行うには、社会通念、上位法、他の条文との関係、法律の制定目的、など広い範囲の知識が使われる。それらの知識からくる制約は互いに矛盾することも多く、完全に満たすことはできないかもしれないが、それらのバランスをとりながら合理的な解釈を求めていかなくてはならない。
解釈や判断の材料として、判例を使うことができる。判例には、その事件の状況、両者の主張と対立点、裁判官の理由付け（解釈を含む）と結論が明確に述べられているからである。過去の判例の中から類似した事件を検索することによって、裁判官の判断の予測や、論争のヒントの獲得が可能になるのである。

2.2　刑法の場合

我々はHELIC-IIの応用として、刑法の論理構築問題に取り組んでいる。そこで、刑法における法的推論を簡単に説明する。
刑法の条文はおよそ270あり、その前半は総則、後半は個々の罪が規定されている。後者の例として、199条に、「人を殺す」ことが殺人罪の成立する条件であること、「死刑、無期懲役、3年以上の懲役」がその刑罰であることが定義されている。この一見明白であるルールでさえ、その適用においてはいろいろな問題を生じる。
まず、「人」とは何かが問題になる。胎児を含むのか、行為者自身を含むのかが条文上は明白でない。次に「殺す」とは何かが問われる。ある「行為」が「殺す」に該当するには、故意（殺意）があること、「死」という結果が生じたこと、その行為と死との間に「因果関係」があることが必要である。総則の中に「故意のない行為は原則として罰しない」(38条)などの規定がある。しかし、故意や因果関係の具体的な定義がないので、その内容は解釈にまかされている。
例えば、因果関係においては、「原因の1つであった行為にはその重要性にかかわらず因果関係を認めてしまう」説（条件説）、「経験則上その行為がその結果を生じるのが当然な場合に限って因果関係を認める」説（相当因果関係説）などがある。このように、因果関係について解釈が分かれているので、「甲を殺そうとして、ナイフで脅したら路上に飛びだして交通事故で死んでしまった」ケースにおける因果関係の判定などは、専門家によって異なる判断がなされるかもしれない。

2.3　例題

HELIC-IIで扱っている例題の1つを以下に示す。これは、司法試験の刑法の問題を簡単化したものである。

> 「冬のある日に、甲は貧しさに堪えかねて、実子である太郎（生後4月）を道端に捨てた。たまたま通りかかった乙は太郎を拾って、自動車で警察に届けようとしたが事故を起こしてしまった。太郎は負傷したが、乙は太郎が死んだものと誤解し、太郎を捨てて逃げた。太郎は凍死してしまった。甲と乙の罪について考察せよ。」

この例題では、甲に保護責任者遺棄罪、乙に過失致死罪が成立することは疑いがないが、甲に保護責任者遺棄致死罪が成立するか（遺棄と死の因果関係）、乙に業務上過失致死罪が成立するか（事故と死の因果関係）、乙に死体遺棄罪が成立するか（故意）などが問題となる。
そこで以下の判例を考える。これは実際に最高裁で扱われた事件を簡単化したものであり、司法試験の上記の問題の解説に引用された判例である。

> 「丙は丁を殺そうと思い、丁が寝ている間に丁の首を絞めた。丁は気絶したが、丙は丁が死んだものと誤解した。そこで犯行を隠すため、丙は丁を海岸に運び、そこに放置した。丁は砂を吸って窒息死した。」

この判例で、被告（丙）は、「首を絞めたときには殺意はあったが目的は達していない」、「海岸に運んだときには殺意はない」ということから、2つの行為（絞める、運ぶ）はそれぞれ、殺人未遂罪と過失致死罪にはなるが、殺人罪ではないと主張した。それに対して、検察側は、「首を絞めた行為と運んだ行為は一体の行為であって、殺意もあり、実際に死んでいる」から殺人罪であると主張し、さらに「仮に一体の行為でないとしても、首理由で因果関係を認めた。

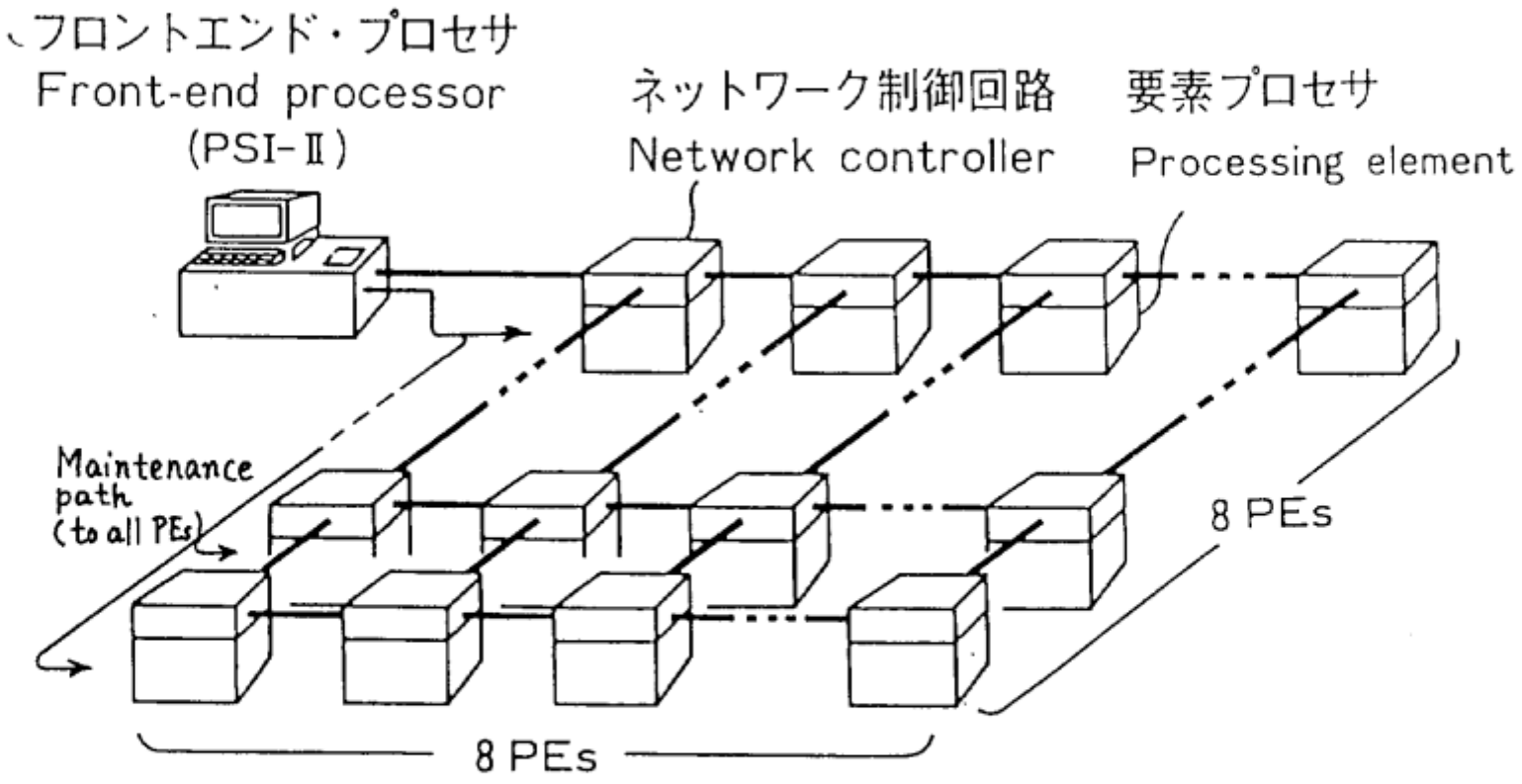

図1　Multi-PSIの構成

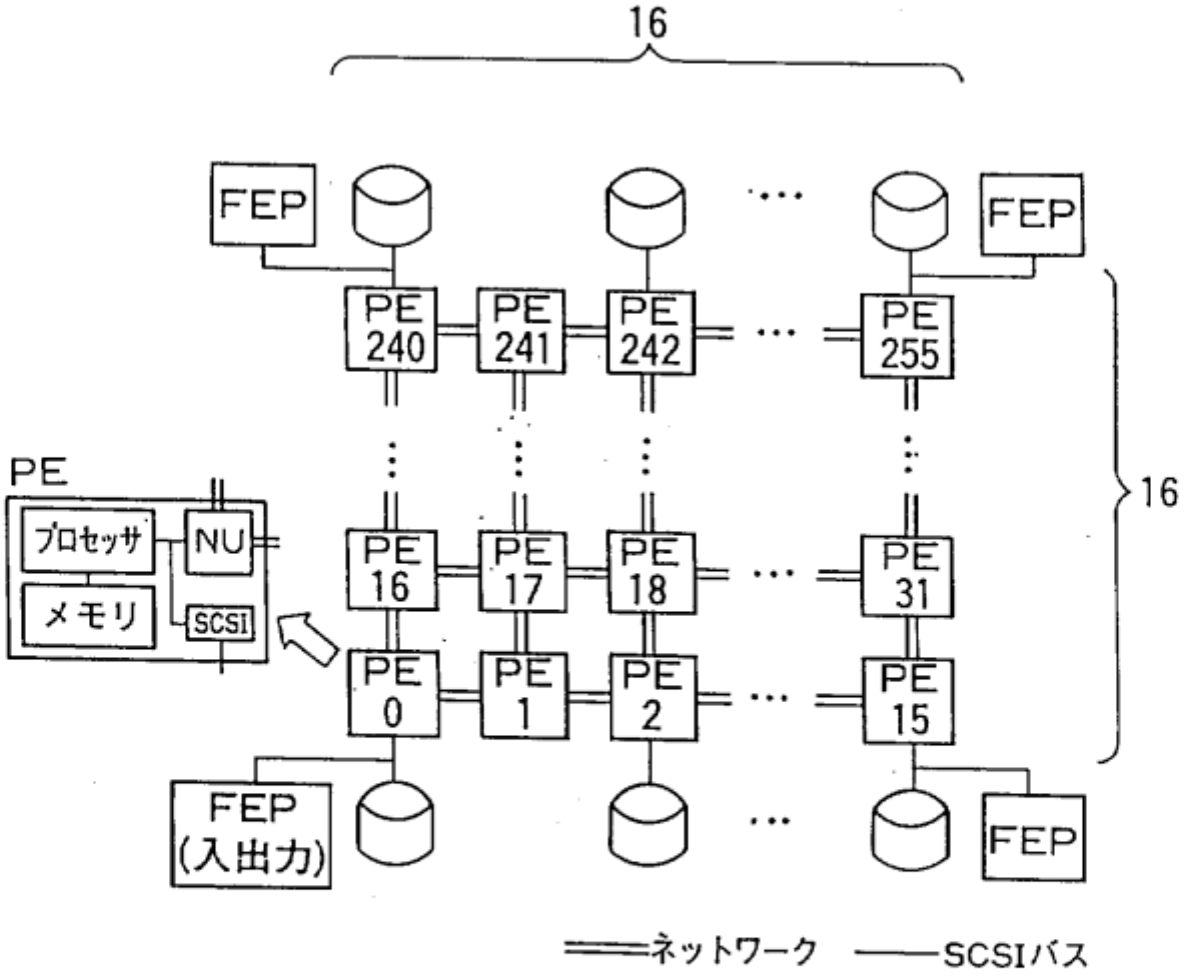
16
FEP
FEP
PE 240
PE 241
PE 242
PE 255
16
PE
プロセッサ
NU
メモリ
SCSI
PE 16
PE 17
PE 18
PE 31
PE 0
PE 1
PE 2
PE 15
FEP (入出力)
FEP
ネットワーク
SCSIバス

図 2　PIM/mの構成

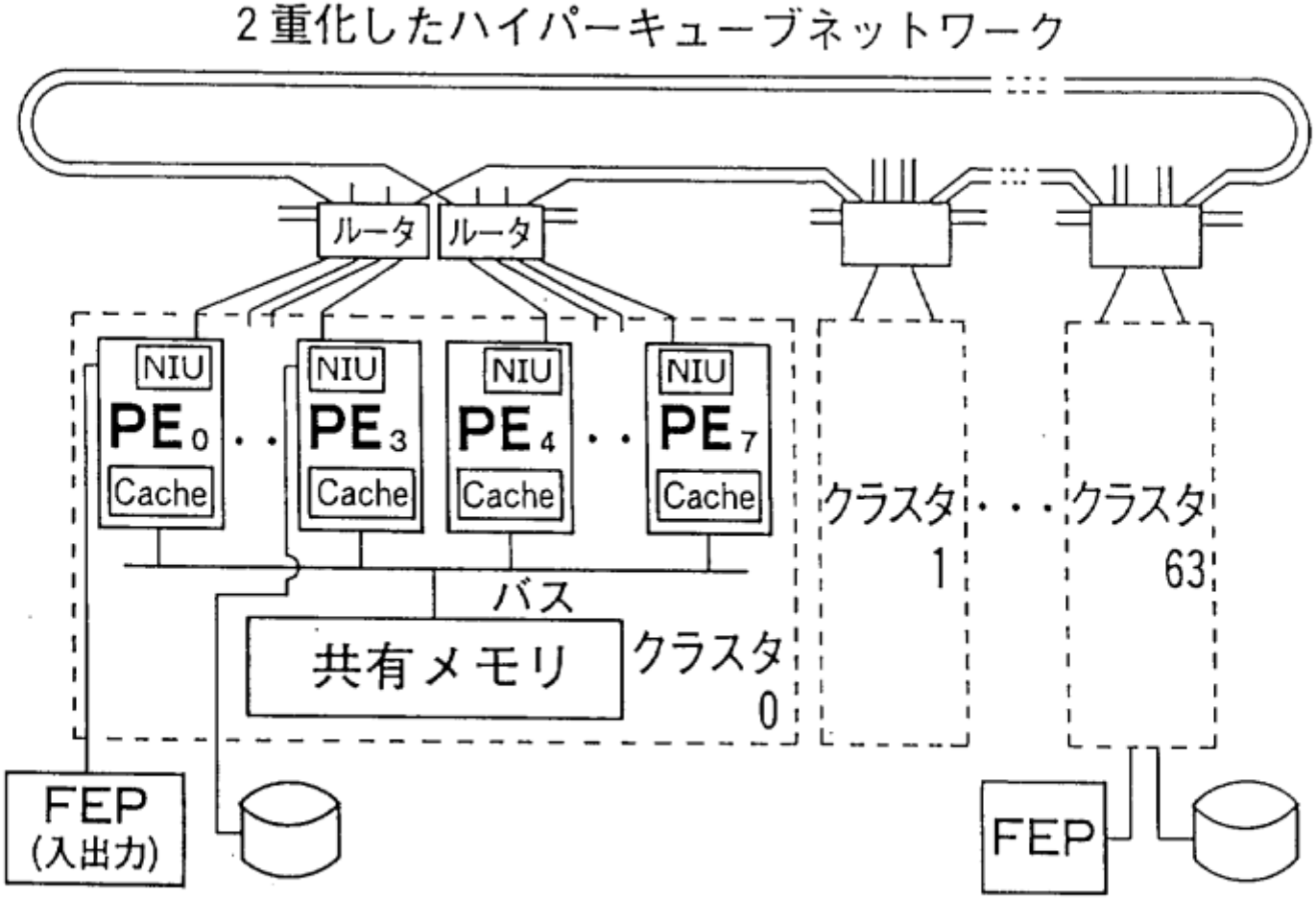
2重化したハイパーキューブネットワーク
ルータ
ルータ
NIU
PE0
Cache
NIU
PE3
Cache
NIU
PE4
Cache
NIU
PE7
Cache
バス
共有メモリ
クラスタ
0
クラスタ
1
クラスタ
63
FEP
(入出力)
FEP

図3　PIM/pの構成

さて、上記の甲事件と丙事件を比較し、

乙	丙
太郎	丁
交通事故	首を絞める
過失（殺意なし）	殺意あり
負傷	気絶
死んでいると誤解	死んでいると誤解
路上に捨てる	海岸に運ぶ
冬	砂
凍死	窒息死

n をそれぞれ対応させると、両者の類似点と相違点がはっきりする。類似性が高いのであれば、丙の事件での議論（双方の論理展開）が甲の事件に利用できる。逆に、相違点が多いのであれば、丙の判例は甲事件に利用しても信頼度が低くなる。

3 開発経緯

3.1 並列推論マシン

ICOT は今までに論理型言語で書かれたプログラムを実行するコンピュータとして逐次型推論マシン (PSI-I,PSI -II, PSI-III) と並列推論マシン実験機 Multi-PSI を開発してきた。 Multi-PSI は 64 個の処理要素 (PE) を 2 次元メッシュ状に結合した並列計算機である (図 1)。各 PE は CPU とメモリを有し、ほぼ 1 台の PSI-II に相当する機能を持つ。
論理型言語 KL1 で書かれたプログラムは、個々の PE に分配され、同期をとり、情報を交換しながら並列に実行される。 KL1 では、どの処理をどの PE で実行させるかを指定できるので、負荷分散と実行時間の関係を実験で得ることができる。
現在、第 5 世代プロジェクトでは、5 種類の並列推論マシン PIM の開発を終了しつつある。5 種類の PIM はそれぞれ異なるアーキテクチャを持つ。例えば、PIM/m は Multi-PSI と同様に、プロセッサが 2 次元メッシュに結合された分散メモリのマシンであるが (図 2)、PIM/p は 8 台のプロセッサと共有メモリが 1 つのクラスタを形成し、クラスタ間がネットワークで結合されている (図 3)。 これらの PIM の言語はいずれも KL1 なので、同じプログラムを修正することなく、異なる PIM 上で実行させることができる。

3.2 HELIC-II の開発

法的推論システム HELIC-II の開発は 1989 年後半から行っている。当初は、労働災害認定問題を題材にとり、「持病を持った人が過労で死亡した場合に過労死と認定されるかどうか」の判例を集めて、労災判断とその理由生成のシステムを試作した。
1990 年後半からは、対象を刑法に移し、犯意や因果関係などの判断に判例を利用している。対象を刑法に移したのは、労働災害では、関連する条文の数が少なく、法令文と判例の双方を利用するシステムとしては、両方の知識源のバランスがとりにくいこと、刑法のほうが一般向けの解説書が多いこと、一般の人に問題点が比較的理解し易いことによる（当初は、民法や商法も検討した）。刑法は、民法などと違い、類推の範囲がきびしく制限されるが（原則として類推解釈は禁止。ただし、法律での類推と AI での類推は同一ではない）、HELIC-II の基本メカニズムは、個々の法律の性質に依存しないので、どの法律にも応用可能である。

4 機能の特徴

HELIC-II は、法令文と判例を利用して法的な結論を導き出す推論システムである。推論システムとしての特徴を以下に示す。

1. 2つの推論エンジン (ルールベースエンジンと事例ベースエンジン) が協調して問題を解決するハイブリッドなシステムである。法令文は、ルールベースに格納されており、ルールベースエンジンは、法律的な結論を生成する。判例は、事例ベースに格納されており、事例ベースエンジンは、現事件と類似の判例を検索し、抽象的な法律概念の候補を生成する。2つのエンジンは共有の作業記憶で結合されており、データ駆動で推論が開始する。

2. 並列推論マシン (Multi-PSI, PIM) の上で並列に推論を行う。ルールベースエンジンにおいては、異なるルールの適用を複数のプロセッサにより並列に実行する。事例ベース推論においては、類似判例の検索と事例ルールの適用をそれぞれ異なるプロセッサにおいて並列に実行する。

5 機能概要

5.1 HELIC-II の構成

HELIC-II は図4のように2つのエンジン (ルールベースエンジンと事例ベースエンジン) からなる。法令文は節形式で記述され、ルールベースに格納される。判例は状況記述と事例ルールの組として、事例ベースに格納される。この他に法令文や判例に現れる概念 (用語) の辞書がある。概念の間には、上位／下位の関係に基づく階層構造が成立している (図5)。
新しい事件を作業記憶に入力すると、2つのエンジンは独立に推論を開始する。新しい事件の事実集合は通常は具体的な概念 (例えば、「運転する」や「殴る」など) のみで記述され、抽象的な概念 (例えば、「故意」や「因果関係」など) は含まれない。法令文には、抽象的な概念が多く含まれているので、初めはルールベースエンジンはほとんど実行を進めることはできず、データ待ちの状態となる。
一方、事例ベースエンジンは入力された問題の類似事例を次々と検索し、事例ルールを適用して、「故意」や「因果関係」の成立可能性についての仮説を生成する。この結果は作業記憶に送られ、それを利用してルールベースエンジンは推論を再開する。同一の判例を引用したとしても、検察側の事例ルールを適用すれば、「故意の成立」などを生成し、弁護側の事例ルールを適用すれば、「故意の不成立」を生成する。このように、互いに矛盾する仮説が事例ベースエンジンによって生成されるので、ルールベースエンジンは矛盾性をチェックし、矛盾しない仮説の組組ごとに法令文を適用した結論を生成することになる。

5.2 知識表現

HELIC-II においては、法令文を記述したルールベースと、判例を記述した事例ベースと、概念辞書という3つの知識が存在する。これらの表現例を 2.3 の例題に即して以下に示す。

(1) 法令文の表現

条文は、ルール形式で記述する。ルールには「～」(logical not) と「not」(negation as failure) という2種類の否定を書くことができる。前者は、明確に否定された情報 (例: 「殺意はなかった」を "～殺意" と記述する) を表現し、後者は、存在が証明できない情報 (例: 「殺意は立証できない」を "not 殺意" と記述する) を表現するのに用いられる。

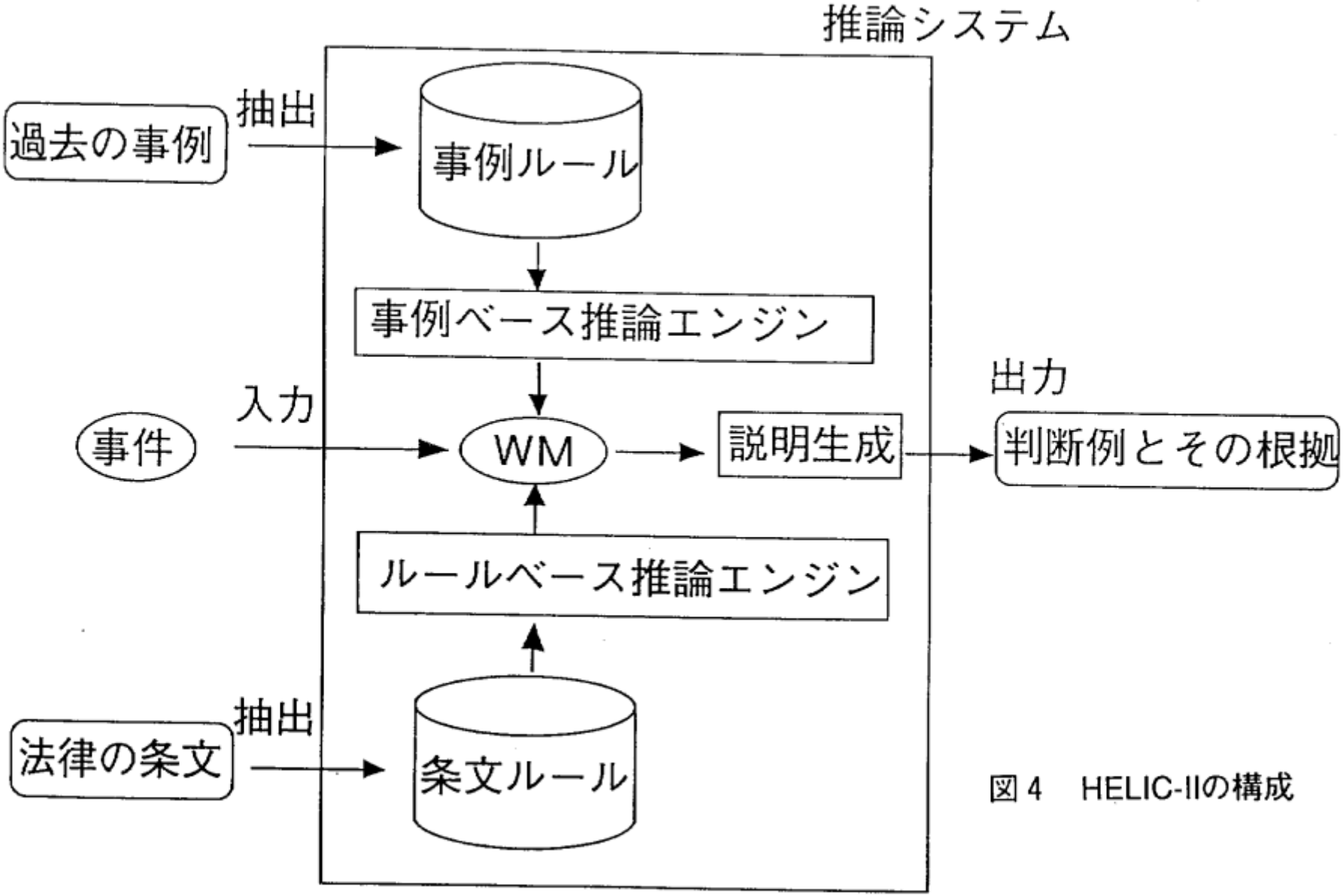

図4　HELIC-IIの構成

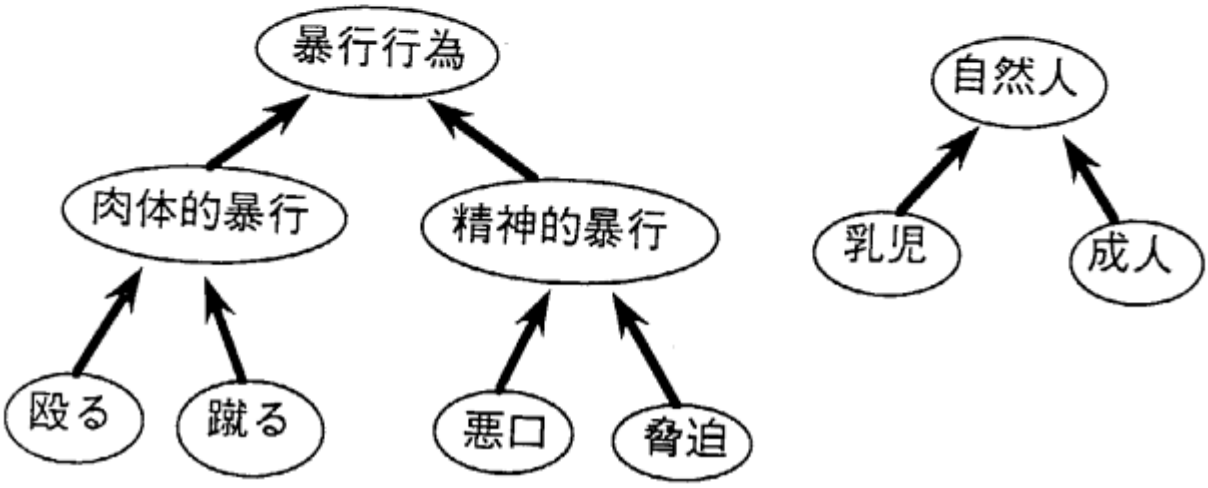

図５　概念階層の例

図 6 のルールは、 保護責任者遺棄致死罪 (218, 219 条) の記述の一部である。述語の第 1 引数はオブジェクト名であり、大文字で始まるのが変数である。「保護を要する人 (Minority) とその人に対して責任 (Obligation) を持つ人 (Person) がいるとする。 Person が Minority を遺棄 (Action) したために、 Minority が死んだならば、保護責任者遺棄致死罪の構成要件に該当する」ことを表している。

(2) 判例の表現

オリジナルの判例には、事実、双方の主張、理由、結論（判決）、 などの情報が含まれている。 この情報を「状況」と「事例ルール」という形式で表現する。
図 7 は、 丙事件における「状況」の表現例である。「状況」は事実 (fact, object) と事象 (event, action), および、それらの間の時間的関係で記述される。
「事例ルール」はこの事件の検察、または、被告の主張を「〜だから〜だ」の形で表現したものである。検察と被告の論理展開は事例ルールの連鎖である。事例ルールの条件部は、抽象的な概念ではなく、 具体的な概念（上記の状況記述の一部）で記述されている。従って、法令文の記述と異なり、変数は含まれていない。
図 8 の事例ルールは、「気絶している人を海岸に運ぶ行為は遺棄にあたる」という検察側の主張を表現している。事例ルールの条件部の中で、どの情報が法的に重要かを表す重みを付与することができる。

(3) 概念辞書の表現

概念辞書は、法令文や判例に現れる概念にどのような属性があるかを記述する。概念の名前、上位の概念、属性としてとりうるもののリストからなる。図 9 の例では、「絞める」と「なぐる」は暴行の下位概念であるという点で共通である。概念階層の中で、共通の上位概念に至るまでの距離によって 2 つの概念の類似度を判定する。 法令文や判例に現れるすべての概念はこの辞書に登録されていることを前提とするが、臨時に辞書を追加することが可能である。

(4) 新しい事件の表現

解決してもらいたい新しい事件の記述は判例の「状況」記述と同じ形式でなされる。例えば、2.3 章で紹介した甲事件は図 10 のように記述される。これを入力すると、推論が開始される。

5.3 推論機能

(1) ルールベースエンジンによる推論

ルールベース推論エンジンは、法令文のルールベースをもとにして、 演繹推論で法的な結論 (刑法では犯罪名など) を求める。このエンジンは、 ICOT で開発された並列定理証明プログラム MGTP(Model Generation Theorem Prover) を応用したものである。 MGTP は図 11 に示すように、空モデルから始めて、与えられたルールを順次適用することによって、すべてのルールを満足するモデルの集合を求める機能を有する。モデル生成は複数の PE で並列に行われるので、証明が高速に行われる。

(2) 事例ベースエンジンによる推論

事例ベース推論エンジンは、第 1 段階として、与えられた事件と類似の判例を検索し、第 2 段階として、その事例ルールを適用して、 仮定的な法的概念を生成する (図 12)。例えば、第 1 段階においては、甲事件 (図 10) について、 event の系列が似ている丙事件 (図 7) を検索する。

```
rule("最上位ルール", [条文=219],
  要扶助者(Minority), 自然人(Person),
  責任(Obligation,[主体=Person, 客体=Minority]),
  遺棄(Action,[主体=Person, 客体=Minority]),
  死亡(Death, [主体=Minority]),
  因果関係(Causality, [原因=Action, 結果=Death]),
  during(Action, Obligation)
   -->
  犯罪構成要件該当(Crime, [行為=Action,
        結果=Death, 犯罪=保護責任者遺棄致死罪])).
```

図6　法令文の記述例

```
case(丙事件,"相当因果関係説が採用された事例",
     [ meets(action1,state1),
       after(action1,action2), ...])

自然人(丙, [性=女, 年齢=34]).
自然人(丁, [性=男, 年齢=40]).
殴める(action1, [主体=丙, 客体=丁, 場所=首1]).
故意(intention, [主体=丙, 行為=action1,
                 目的=death1]).
死亡(death1, [主体=丁]).
caused(cause1, [行為=action1, 結果=state1]).
気絶(state1, [主体=丁]).
誤解(action2, [主体=丙, 認識=death2,
               事実=state1]).
運ぶ(action3, [主体=丙, 客体=丁, 目的地=海岸1]).
   .......
```

図７　判例の状況の記述例

```
rule01("遺棄行為",
   [条文=刑法199条，罪名=殺人罪，
     主張者=検察，対立ルール=rule03]),
   [気絶(state1,[主体=丁]),
    運ぶ(action3,[主体=丙，客体=丁，
                  目的地=海岸1]) ]
    ==>
   [遺棄(action4，[対象=action3，主体=丙，
                   客体=丁，場所=海岸1])])
```

図 8　事例ルールの記述例

concept(絞める，暴行，[主体，対象，場所])
concept(なぐる，暴行，[主体，対象，場所])
concept(交通事故，事故，[主体，乗り物，場所])

図９　概念の記述例

```
case(甲事件,"例題",
     [ meets(action101,state101),
       after(action101,action102), ...])

 自然人(甲, [性=女, 年齢=20]).
 乳児(太郎, [性=男, 年齢=0]).
 遺棄(action101, [主体=甲, 対象=太郎,
                  場所=道路1]).
    .......
 交通事故(accident, [主体=乙, 乗り物=車1])
 過失(negligence, [主体=乙, 行為=accident])
 caused(cause101, [行為=accident,
                   結果=state101]).
 負傷(state101, [主体=太郎]).
 死亡(death102, [主体=太郎]).
 誤解(action102, [主体=乙, 認識=death102,
                  事実=state101]),
 置き去り(action103,[主体=乙, 対象=太郎,
                     場所=道路2]) ]
    .......
```

図１０　新しい事件の記述例

C1; true → p(a) ; q(b).

C2; p(X) → q(X) ; r(X).

C3; r(X) → s(X).

C4; q(X) → false.

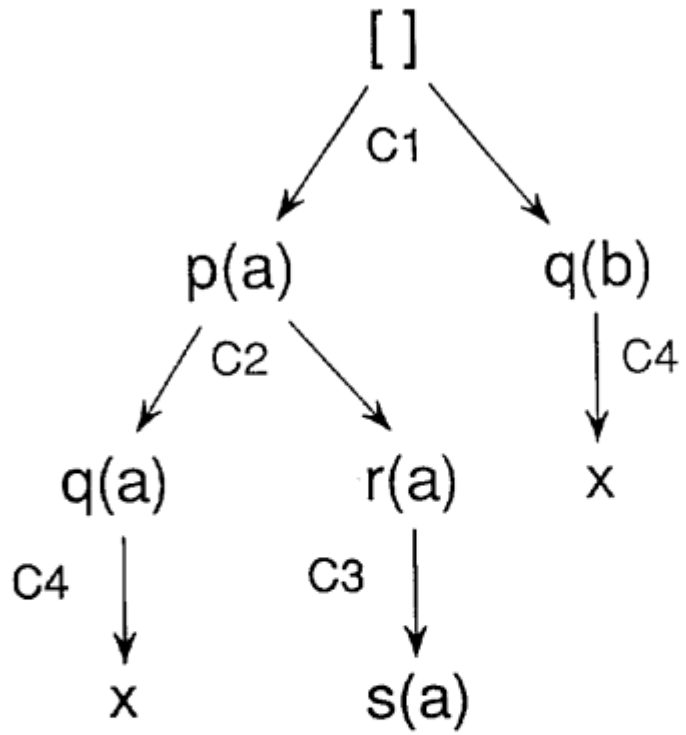

図１１　MGTPのモデル生成

{p(a),r(a),s(a)}

モデル

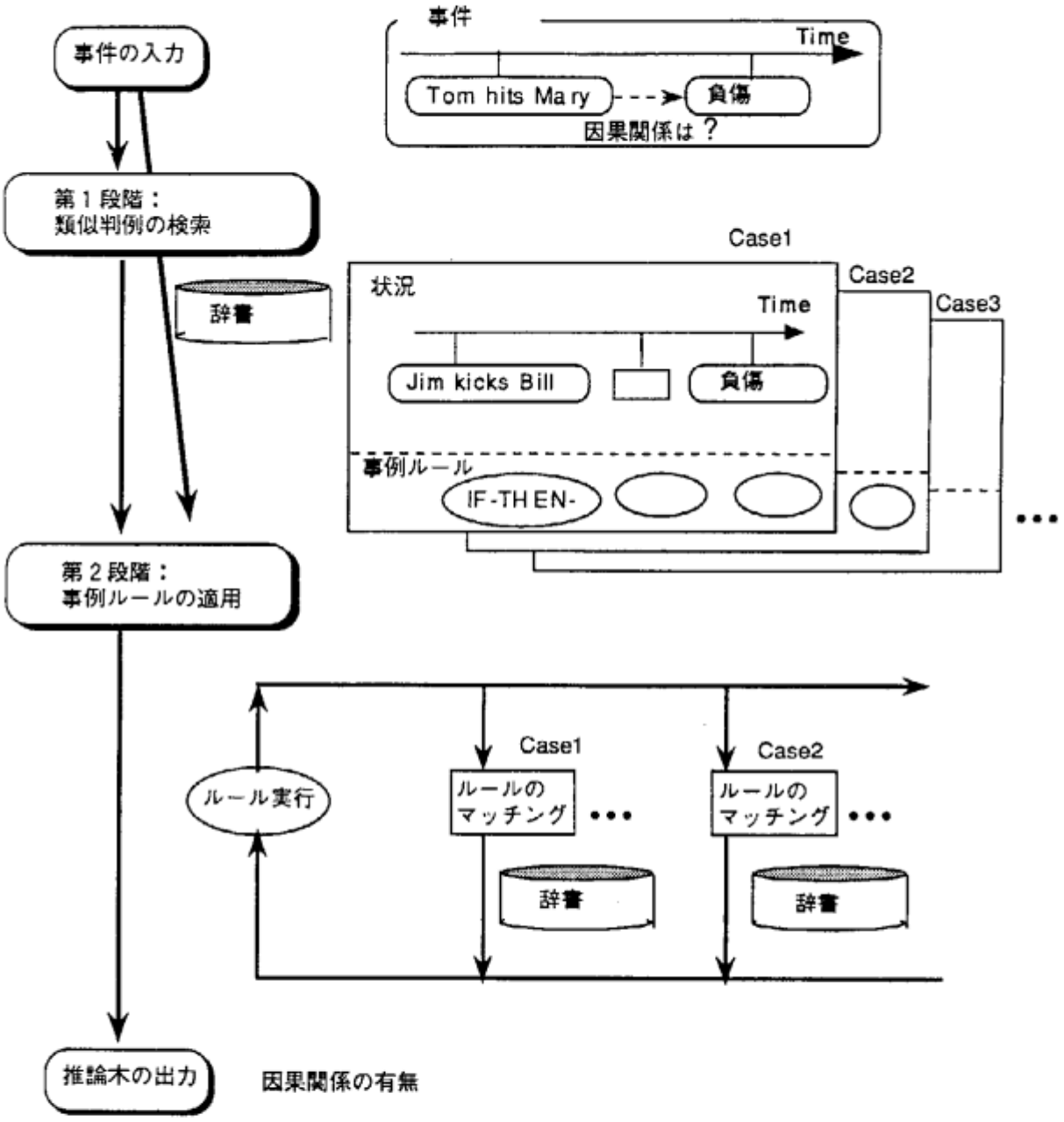

図１２　事例ベース推論エンジンのはたらき

第2段階の事例ルールの適用は、ルールの条件部と与えられた事件の個々の object(event) を対応付けることによって類似性を判定し、新しい object(event) を仮説として生成する。例えば、甲事件 (図 10) と図 8 の rule01 において、

自然人 (丙)	と	自然人 (乙)
自然人 (丁)	と	乳児 (太郎)
気絶 (state1)	と	負傷 (state101)
運ぶ (action3)	と	置き去り (action103)

を対応付けて「負傷 (state101) した人 (太郎) を置き去りにした行為 (action103) が遺棄にあたる」という結論を生成する。このように事例ルールのマッチングはプロダクションルールのような厳密なマッチングではなく、類似性に基づくマッチングである。従って、概念辞書を利用した object(event) 自体の類似性や、着目している object(event) の周辺情報の類似性が利用される。また、一部の条件部が満足されなくても、重要性の高い条件部分がマッチしていれば全体としてマッチングしていると判断される可能性がある。

5.4 出力結果

2.3 章であげた例題に対し、HELIC-II は図 13 のような推論木を 12 個生成した。それぞれの木の末端は初期データである。末端以外のノードは事例ベースエンジンやルールベースエンジンの出力であり、ルートには犯罪名などが現れる。
われわれは、最終結果 (犯罪名) ではなく、途中の推論過程を重視している。HELIC-II は、学説の違いによって異なる木 (推論過程) が生成するので、それらの木を比較することによって、どの学説がどの事実に着目しているか、わずかな事実の認識の違いがどのような結果の差になるか、などを知ることができる。これらを材料として、裁判の論理展開をするときのヒントを獲得したり、相手の攻撃に対する反論を用意したりすることができることを期待している。

5.5 性能評価

今までに、いろいろな法的推論システムが研究されてきているが、実行時間にまで考察したものは少ない。小さな問題を扱っているときには問題にならないが、多くの法令文や大量の判例を扱うときには、実行時間が問題となってくる。そこで、並列推論マシン (Multi-PSI, PIM) による速度の向上が期待される。
Multi-PSI を用いて同じ問題をいろいろな数の PE を用いて実行し、並列処理による速度向上を測定した (図 14)。特に事例ベース推論エンジンの効果が顕著に現れている。今後、学説によるルールの分岐数が増えると、ルールベースエンジンの並列効果は増大すると思われる。

6 まとめ

第5世代コンピュータの技術は、大規模知識情報処理に適しており、われわれは、遺伝子情報処理、CAD システム、囲碁システム、さまざまなエキスパートシステムを開発して、その性能を評価している。ここでは、PIM の1つの応用として、法的推論システム HELIC-II の概要を紹介した。法律は膨大な法令文と判例データベースを持っているので、大規模情報処理が必要な分野である。今後は、専門家の評価を受けながら、ルールベースや事例ベースや辞書の拡充と機能の充実を図る予定である。

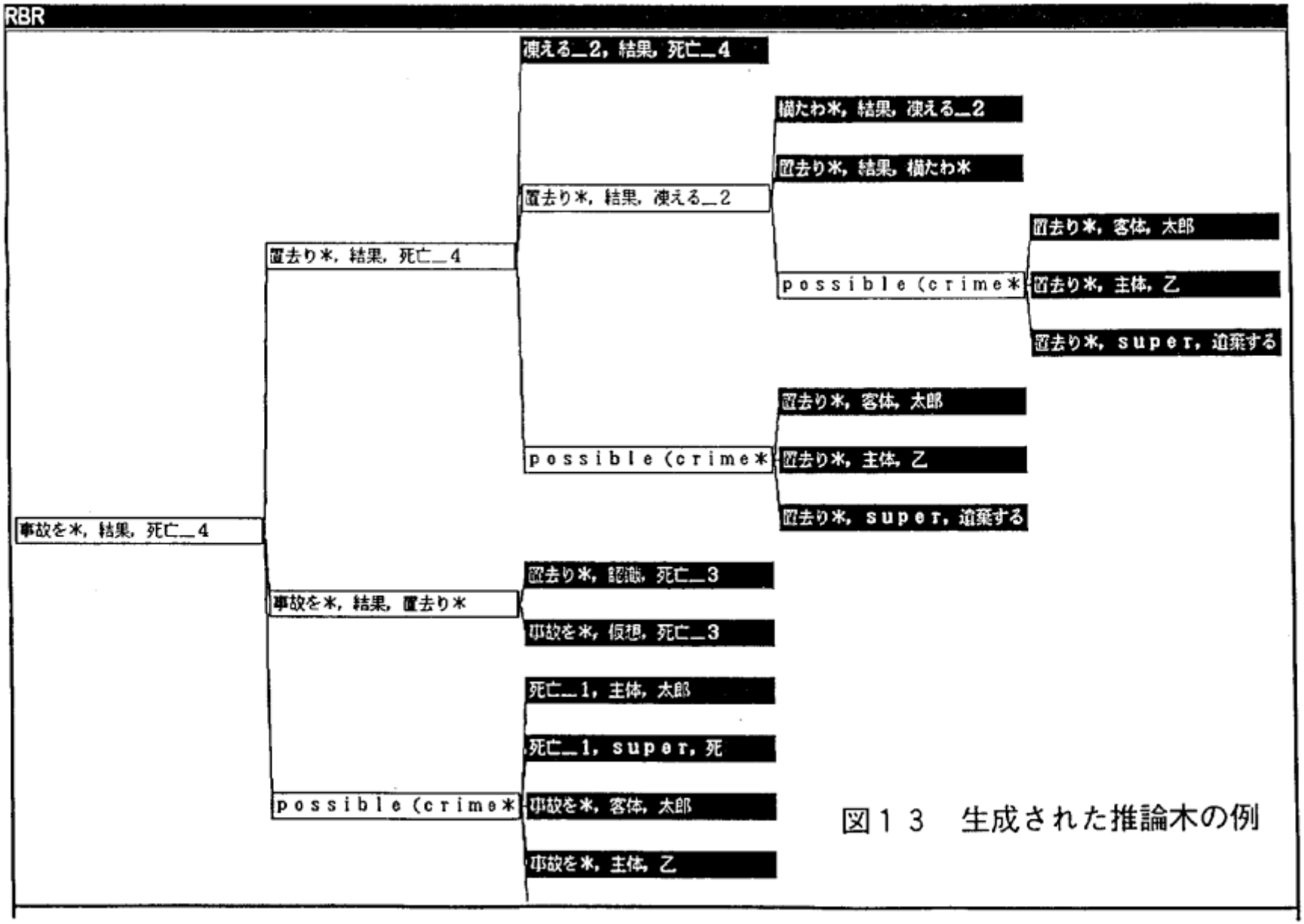

図１３　生成された推論木の例

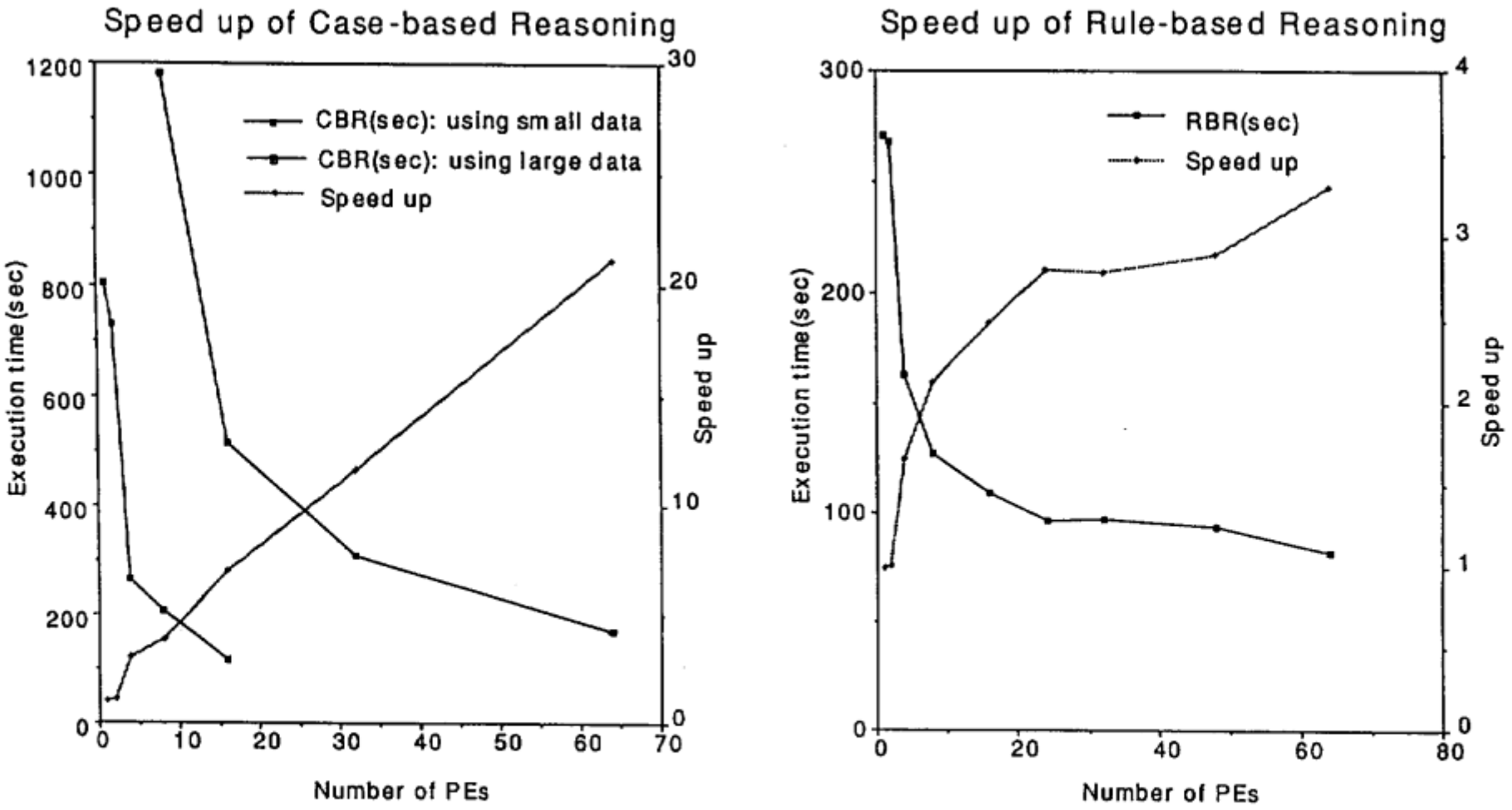
Speed up of Case-based Reasoning
CBR(sec): using small data
CBR(sec): using large data
Speed up
Execution time(sec)
Speed up
Number of PEs
Speed up of Rule-based Reasoning
RBR(sec)
Speed up
Execution time(sec)
Speed up
Number of PEs

図１４　並列推論の効果

参考文献

[1] 大嶽ほか,「法的推論システム HELIC-II (1) ～ (4)」, 情報処理学会第 43 回全国大会論文集 (6E-6 ～ 9)

[2] 前田ほか,「法的推論における並列推論」, SWoPP '91 (AI-77-6)

[3] 新田ほか,「事例を用いた法的推論とその並列化」, 情報処理学会研究会 (知識工学と人工知能 69-5)

[4] 藤田ほか,「A Model Generation Theorem Prover Using Ramified-Stack Algorithm」, ICOT TR-606