法的推論システム HELIC-II

大嶽能久[†] 新田克己[†] 前田茂[†]
小野昌之[†] 大崎宏[††] 坂根清和[†††]

法的推論システム HELIC-II について述べる．法的推論を計算機上で実現するためには，個々の事件の事実関係に解釈を与え，それに法的な概念を対応させる過程をいかに実現するかが大きな問題の一つとなる．HELIC-II は条文と判例を知識源とする hybrid システムである．条文に基づく推論はルールベース推論によって，判例に基づく推論は事例ベース推論によってそれぞれ実現されている．判例に基づく推論は過去の類似の判例を参照して法的な概念を生成する．条文に基づく推論はこれらの法的な概念を使って罪責を演繹的に求める．両者は相補的に機能し，あらゆる可能な法的判断を生成する．ルールベース推論エンジンは，並列定理証明器 MGTP (Model Generation Theorem Prover) をベースにして，それに幾つかの拡張を施した．事例ベース推論エンジンは，推論を類似事例の検索と適用の2段階に分けることにより，事例の記述を容易にすると同時に並列推論の効果を高めた．出力としては，これらが導かれた過程を表す推論木がユーザに提示される．さらに推論木の理解を容易にするために，自然言語風の詳細説明も提示することができる．HELIC-II は並列推論マシン上にインプリメントされ，並列推論によって高速に結論を導き出す．例題として刑法を対象とする実験システムを構築し，並列推論の効果を実証した．

Legal Reasoning System HELIC-II

Yoshihisa Ohtake,[†] Katsumi Nitta,[†] Shigeru Maeda,[†] Masayuki Ono,[†]
Hiroshi Osaki[††] and Kiyokazu Sakane[†††]

This paper presents HELIC-II, a legal reasoning system on the parallel inference machine. HELIC-II draws legal conclusions for a given case by referring to a statutory law (legal rules) and judicial precedents (old cases). This system consists of two inference engines. The rule-based engine draws legal consequences logically by using legal rules. The case-based engine generates legal concepts by referencing similar old cases. These engines complementally draw all possible conclusions, and output them in the form of inference trees. Using these inference trees HELIC-II can support argumentations in a legal suit. HELIC-II can draw conclusions quickly by parallel inference.

1. はじめに

法律の知識は条文（法令文）と過去の判例からなる．条文はルールであり，条文に基づく推論はルールベース推論によって実現することができる．しかし条文ルールはしばしば定義の曖昧な法的概念を含んでいる．そのような法的概念と実際の事件の事実関係との対応付けは自明ではなく，しかも一般的な解釈方法をあらかじめ与えておくこともできない．このような条文ルールを実際の事件に適用するには，それに含まれる法的概念を法律の専門家が，個々の事件に則して解釈する必要がある．その際しばしば過去の判例が参照され，その中の論理展開が再利用される．つまり法的推論はルールベース推論（RBR）と事例ベース推論（CBR）の複合パラダイムとしてモデル化することができる．

同様のモデル化に基づいてルールベース推論と事例ベース推論の hybrid システムが幾つか開発されている[7),8)]．しかし，実際的な法的推論システムを開発するに際しては，いくつかの困難がある．まず第一に条文ルール同様，過去の判例もまた非常に数多くあり，その中から類似の事例を検索し，それに基づいて結論を導き出すには多くの時間が必要である．第二に複数の推論エンジンを管理するためには，それらの推論エ

† (財)新世代コンピュータ技術開発機構
Institute for New Generation Computer Technology
†† (財)日本情報処理開発協会
Japan Information Processing Development Center
††† 新日本製鐵
Nippon Steel Corporation

ンジンが行う推論を制御するための複雑な機構が必要となる．われわれは法的推論システム HELIC-II を並列推論マシン上に開発し，これらの問題を並列推論によって解決した．

2. 法的推論とは

法的推論とは，法律の専門家が法律上の問題を解決する際に行う推論のことである．端的に言えば事実認定や解釈を伴う推論である[4]．

法令文の多くは「もし〜ならば〜である．」の形式をしているが，これを IF-THEN 形式でルール化したからといって，それらを使ったルールベース推論だけで法的な結論が機械的に得られるというようなものではない．なぜなら法令文はそれだけで完結した知識の体系として与えられているわけではないからである．

例えば刑法条文は「人を殺したるものは，殺人罪に処す」といったような記述で与えられるが，この場合の「殺した」という記述は，客観的な事実を指すものではなく，法律の専門家によって判定されるべき概念である．しかもこれらの概念には，どういう事実がそれに該当すると判定されるべきかという定義が，法令文はもとより，何者によっても明確には与えられていないために，その判断は一般に自明ではない．

このような法律の条文の意味は法律の専門家が，個々の事例ごとに，その事実関係に照らして解釈することによって与えるものである．言い換えれば，特定の条文が選択され適用されるためには，まず法的概念と具体的な事実とを結び付けるという操作が必要なのである．

ところで解釈および判断とは常にある程度主観的な要素を含むものである．それゆえに弁護側，検察側など，その置かれた立場によって，法律の専門家の出す結論が異なるということが起こってくるのである．そして裁判とはこのような対立する解釈および判断のうちどれが正当であるかが争われる場である．つまり法律の専門家による事実関係の解釈および判断こそが法的推論の核心部分であるともいえるのである．

法律の専門家は，上記のような判断を行う際には，過去の判例とりわけ最高裁の判例を重要な知識源として使っており，またそれは裁判での重要な論拠ともなる．

3. システム構成

以上のように法的推論は，演繹的な推論と事例に基づく推論との相補的な組み合わせとしてモデル化できる．前者が法律条文に基づいた推論を行い，後者が過去の判例に基づいて，具体的な事実と法的概念とを結び付ける推論を行うというものである．

HELIC-II はこのようなモデル化を反映して，**図1**に示すようにルールベース推論エンジンと事例ベース推論エンジンの二つの推論エンジンから構成されている．また HELIC-II は並列論理型言語 KL 1 で記述された並列プログラムであり，並列推論マシン PIM 上で並列処理される．

事例ベース推論エンジンとルールベース推論エンジンは作業記憶を介して通信しながら，独立に推論を進めることができる．新しい事件が入力されると，二つの推論エンジンはデータ駆動でそれぞれに推論を開始する．これらの通信やデータ駆動による推論制御は KL 1 処理系が提供しているものである．

条文は条文ルールと呼ばれる節形式のルールに変換して，ルールベースに格納している．判例は事件の事実関係の記述と事例ルールの組として事例化し，事例ベースに格納している．事例ルールとは検察弁護双方の主張および判決の各論理展開をルール化したものである．このほかの知識源として概念階層辞書がある．この辞書には条文や判例に現れる用語の概念的な階層関係などが定義されている．

入力は解きたい事件の事実関係の記述である．事実関係の記述とは事件の内容を，それを構成する個々の事象の定義と，それらの間の時間関係とし，それらを意味ネットで表現したものである．詳しくは次章で述べる．

推論過程では入力された新しい事件の事実関係に与えられた解釈と，因果関係や故意などについての判断

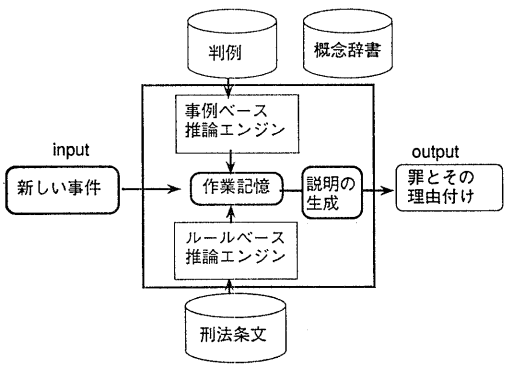


図 1 HELIC-II のシステム構成
Fig. 1 The architecture of HELIC-II.

が生成される．最終的な出力は，それらの解釈や判断に基づいて当事者に対して成立すると結論付けることのできるあらゆる可能な罪である．ただし出力形式としては，根拠となる事実関係から，ある解釈と判断を経てそれらの結論に至るまでの推論過程を推論木として出力する．

4. 知識表現

4.1 事実関係の表現

事実関係や法的な概念の定義は，新しい事件の記述や条文，判例の記述の基礎となるものである．

（1）事実関係の表現

事実関係は行為や状態の変化などの発生した事象の系列とそれらの間の時間関係，およびそれらに関与した個体の定義からなる．発生した事象の個々の個体の定義は，

概念（**Id**, [**属性名＝属性値**, …]).

という形で表される．これはあるオブジェクト(Id)が特定の概念あるいは範疇のインスタンスであることと，その性質を表す属性が（属性名，属性値）の対として定義されることを表す．

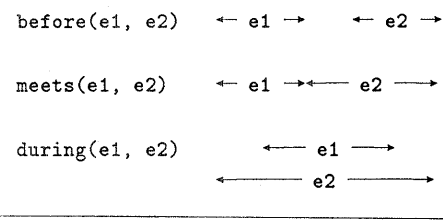
例えば，普通の人間は法人に対応して自然人という法的概念が与えられており，

自然人（**太郎**, [**性別＝男**, **年齢＝20**]).

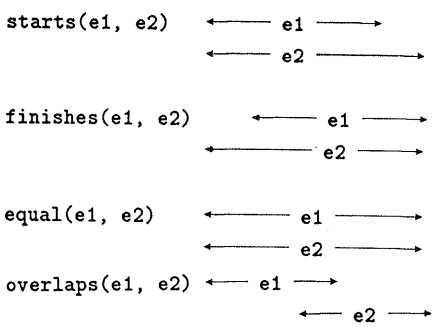
は isa（太郎，自然人），性別（太郎，男），年齢（太郎，20）という三つ組の述語表現と等価である．また発生した事象間の時間関係の定義は，

problem（**事件名**, "**コメント**", [**時間関係**, …]).

という形で表される．problem という述語によってある事件の時間関係の集合に事件名とコメントが関係付けられる．時間関係は Allen の体系[1]に基づいて before, meets, during, starts, finishes, equal, overlaps を用いて記述される☆．これらの時間関係を図式化すると，



☆ ただし，equal を除いたそれぞれに after, met-by, contains, started-by, finished-by, overlapped-by という逆関係がある．それらも事件や事例記述には許しているが内部では自動的に前者に統一して扱っている．



といった関係になる．また条文ルールでは before, during, equal だけが使われている．

入力データとなる事件の事実関係は，以上の時間関係の定義と発生した事象や個々の個体の定義の集合として入力される．

problem（**事件名**, "**コメント**", [**時間関係**, …]).
概念（**Id**, [**属性名＝属性値**, …]).
…

（2）概念の定義

推論に必要な法律条文や判例文に現れるすべての概念は，概念階層辞書に登録されていなければならない．これは特に事例ベース推論において類似事例を検索し，それに基づいた論理構築を行う際に二つの概念の類似度を判定するために用いられるものである．したがってデータの中に未定義語が含まれていると，適切なマッチングが行えない．

概念階層辞書はこのような概念の定義の集合であり，個々の概念の定義はその概念の名前，上位概念および属性として取りうるもののリストを以下の形式で記述したものである．

上位概念（概念，[**属性名＝属性値**, …]).

この辞書に定義してあれば概念の名前，属性名として使われる用語には特に制限はない．

4.2 条文の表現

法律の各条文は条文ルールの形式で表現される．この条文ルールとは条文を IF-THEN 形式で表現したものである．

ルール名（"**コメント**", [**条文＝[条文番号**, …]],
[**A 1, A 2**, …**An**]
--〉
[[**B1, B2**, …**Bm**], [**C1, C2**, …**Cl**], …]).

ルール名，コメントに続くのは関連条文の条文番号のリストである．*Ai*, *Bi*, *Ci* は事実関係の定義あるいは概念の定義を以下のようにパラメータ化したものである．

概念（変数1, [属性名＝変数2, …]）.

同一ルール中に現れる同一名の変数には同じ値が束縛される．条件部 [A1, A2, …An] の各 A*i* には論理否定（以後 ¬ で表す）と証明の失敗による否定（以後 not で表す）の2種類の否定を記述することができる．帰結部 [[B1, B2, …Bm], [C1, C2, …Cl], …] では，各 B*i*, C*i* には ¬ のみ記述することができる．また A1 から An まで，B1 から Bm まで，および C1 から Cl までは AND で結ばれ，[B1, B2, …, Bm] や [C1, C2, …Cl] はそれぞれ OR で結ばれる．

条文をルール化するには ¬ と not の2種類の否定が記述できることは必須条件である．例えば総則の規定にある犯罪の成立要件は「A がある犯罪行為を行い，なおかつ A がその行為を行ったことについての違法性はないとは言えず，かつまた A にその責任能力がないとも言えない場合，その犯罪は成立する」である．このようなルールは2種類の否定を使って以下のように記述することができる．

```
犯罪の成立要件（'' '', [条文＝[35, 36, 37, 41]],
  [構成要件該当性有り（K, [主体＝A,
                          行為＝Act,
                          罪名＝Crime]),
  not（¬違法性有り（I, [主体＝A,
                        行為＝Act])),
  not（¬責任能力有り（N, [主体＝A,
                          行為＝Act]))]
-->
[[犯罪成立（C, [主体＝A,
                行為＝Act,
                罪名＝Crime])]]).
```

4.3 事例（判例）の表現

判例には事件の事実関係および原告と被告の論理展開が記述されている．判例データは事件の事実関係の定義と，個々の論理展開を構成する，事実関係に対する解釈や法的判断をルール化した事例ルールとからなる．事例ルールは事件の事実関係からある罪の成立に至るまでの論理展開を，単一のルールとしてではなく，複数の段階に分けてルール化したものである．新たな事件に対する論理展開はこれらのルールの連鎖として構築される．

```
ルール名（''コメント'',
        [条文＝[関連条文, …],
         罪名＝罪の名前,
         学説＝背景学説名,
         主張者＝検察|弁護|法廷,
         勝敗＝勝訴|敗訴,
         対立ルール＝[ルール名, …]],
        [A1, A2, …An]
        -->
        [B1, B2, …Bm]).
```

A*i*, B*i* は事例の事実関係の定義の一部あるいは概念の定義である．ただし A*i* はそれぞれの属性に trivial, important, exact の3段階の重みが付けられている．

概念（Id, [属性名＝属性値/重み, …]）.

これらの重みは帰結部に現れる結論に対する，個々の要素の重要度を表すものである．帰結部には条件部に現れる事実関係に基づいてなされた判断が記述される．したがってここには条件部の事実よりも，より抽象度の高い概念が現れる．また A1 から An まで，および B1 から Bm までは AND で結ばれ，個々の要素には ¬ を付けることが許されている．

5. 推論方式

HELIC-II は，条文ルールベースと事例ベースという二つの異なる知識源を持ち，ルールベースエンジンと事例ベースエンジンという二つの推論エンジンが相補的に問題を解決するハイブリッド推論システムである．

定理証明器をベースとしたルールベース推論エンジンで，法的推論の論理的な側面をより厳密に扱い，概念階層と部分照合による状況の類似性判定により，類似事例での論理展開を再利用している．

5.1 判例を用いた事例ベース推論

事例ベース推論は，事例ベースと概念辞書を知識源とする．与えられた事件と事実関係の類似した過去の事件を事例ベースから検索し，そこでなされた判断を現在の事実に適用することにより，事実関係と法的概念とを結び付ける．

HELIC-II ではこのような推論を類似事例検索と類似論理構築の二つのフェーズに分けて処理している（図 2）．

5.1.1 類似事例検索

類似事例検索は，事例ベースの中から与えられた事件に類似する事例を選び出すものである．事件，事例ベース，概念階層辞書を入力とし，個々の事象の上位概念，時間関係，そして状況の三つに着目して類似する事例を検索する．

類似事例検索では事件の事実関係と事例ベース中の事例の事実関係を比較し，両者の個々の事象の時間関係に沿った類似照合を行う．例えば**図3**の上段の事象系列と下段の事象系列とが与えられた場合，時間の流れに沿って概念的に対応する事象の組を作ってゆく．このような事象の対応付けは部分照合によって行われるので，部分的に対応の取れない事象があっても照合可能となる．

個々の事象の時間関係に沿った類似照合以外にも，状況の類似性による照合も行っている．これはあらかじめ登録されているある特徴的な事実関係の型に着目し，事実関係にある見方を与えた照合も行っている☆．

4.3 節で述べたように1事例中には複数の事例ルールが定義されている．類似の事件を扱った事例でも，その中のすべての事例ルールが与えられた事件にとって参照の対象となるわけではない，そこで与えられた事件と参照する事例の間の類似点に基づいて，検索された事例中に定義されている事例ルールのうち，参照の対象となりうるものだけを選別して出力する．

このようにして，類似事例検索で類似事例と事例ルールを絞っておくことで，後段の類似論理構築における無用な照合処理を省くことができる．

5.1.2 類似論理構築

類似事例検索で検索された類似事例は，類似論理構築のフェーズに渡される．このフェーズでは事例ルールを使って，与えられた事件の事実関係に基づいて法的概念を次々と生成し，その推論連鎖により事実関係と条文中の法的概念とを結び付け，結論としてある罪を導き出す．この過程を論理構築という．

まず事例ルールの条件部と新しい事件の類似照合を行い，類似していればその実行部を実行する．類似照合は条件部と新しい事件をそれぞれ三つ組の述語表現(4.1 節(1))として考えたとき，条件部の各三つ組を(i)同じ述語をもち(ii)引数が概念辞書の中で共通の上位概念をもつような新しい事件の三つ組にできるだけ多くマッピングさせることで行う．

例えば**図4**のような事例ルール（の条件部に現れる事象）と新しい事件が与えられたとする．両者の各事象間で概念距離の近い物同士の対応を，できるだけ多く取ることにより，リンカーンの事例を参照した場合，さまざまな背景の中からキューバ侵攻の失敗がケネディ暗殺の背景として浮かび上がってくる．

類似性の判定は，条件部のマッピングされた各要素の重みを考慮した，条件部全体の類似度を計算することによって行われる．重みの指定のうち exact は特別な重み付けで，この指定のある要素がマッチしなかった場合には全体のマッチングが失敗する．important と trivial の指定されたものについては，概念階層辞書中での距離に相対的な重みづけをして類似度を計算し，それが与えられた閾値を越えた場合そのルールを発火させる．

5.2 条文に基づく演繹的推論

ルールベース推論エンジンは与えられた事件の事実関係と，事例ベース推論エンジンで生成された法的概念に条文ルールを適用して，最終的な特定の罪の成立を結論付ける．ただし事例ベース推論エンジンで生成される法的概念は検察や弁護等のある特定の立場からの主張であり，否定される可能性があるものという意味で仮説とよぶ．

ルールベース推論エンジンは並列定理証明器

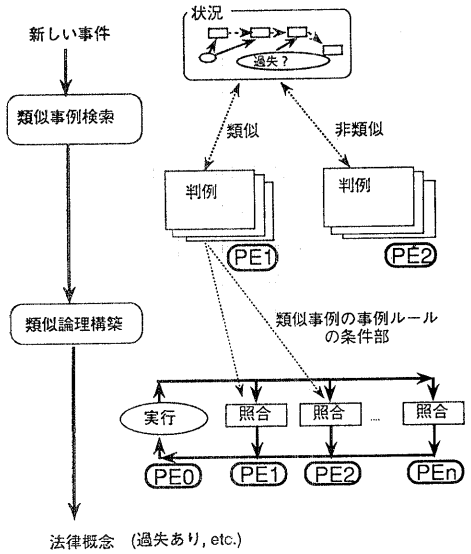


図2 判例を用いた事例ベース推論
Fig. 2 Reasoning by the case-based engine.

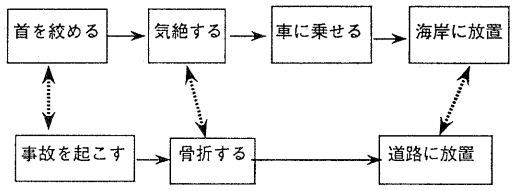


図3 類似事例検索での事実関係の照合
Fig. 3 The fact matching in 1st phase of CBR.

☆ この機能は現在はまだ部分的な限定されたものでしかない．

MGTP[2]を拡張した演繹的推論エンジンである．この推論エンジンの特徴は並列モデル生成機構による高速推論，矛盾検出などである．これらの機能は法律の論理的な側面を扱うのに適している．

図5に MGTP によるモデル生成の様子を示す．R1 から R6 がルールである．R1 のような無条件に発火できるルールによって初期事実が定義される．モデル生成は R1 から始まり，R2 と R3 が発火可能となる．R2 と R3 の右辺に現れる ';' は OR を表す．R2 が先に発火するとモデルは '泥酔' と '正常' で M2 と M3 に別れ，以後のモデル生成はそれぞれ別々に進められる．次に R3 が発火するとさらにモデルは '故意' と '過失' で M4 と M5 に別れ，M5 では R5 により過失傷害罪が導かれる．

法的推論システムのルールベース推論エンジンとするにあたって，主に以下の2点について MGTP を拡張した．

（i）否定を含むルールの扱い

MGTP は元々¬を扱うことができたが not は扱うことができなかった．しかし4.2節でも述べたように条文ルールのルールベース推論エンジンとしては not を扱えることは必須であり，この点拡張が必要であった[3]．

（ii）対立する結論を含むデータの扱い

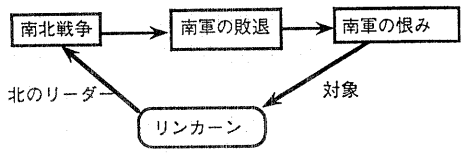
事例ベース推論エンジンからは検察側の主張と弁護側の主張の対立により，例えば一方がAと主張し他方が ¬A と主張するといった相反する仮説がそれぞれの事例ルールから得られることがある．

このような場合，事例ベース推論エンジンから生成される仮説集合から，上記のような相反する仮説を同時に含まない仮説の組み合わせをいくつか生成し，その組み合わせごとにルールベース推論を実行する．これは論理的には矛盾の生じないあらゆる可能世界ごとに，並行して演繹推論を行うことに相当する．また意味的には異なるそれぞれの主張を仮に認めた場合の論理展開を，各主張ごとに生成していることになる．

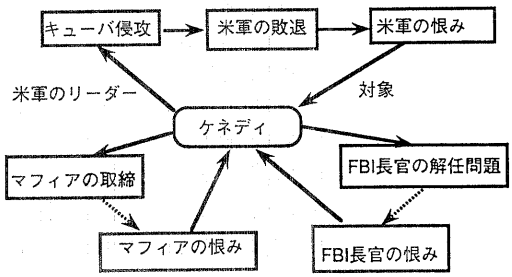
6. 実 行 例

実行例としてある，例題を解かせた場合について説明する．問題は昭和60年司法試験の刑法問題であり，以下のようなものである．

『甲女は，生後4個月の実子太郎の養育に疲れ，厳寒期のある夜，人通りの少ない市街地の歩道上に，誰かに拾われることを期待して太郎を捨てた．そこを通りかかった乙は，太郎に気付き，警察に送り届けようとして，自己の自動車に乗せて運転中，誤って自動車を電柱に衝突させ，太郎に瀕死の重症を負わせた．乙は，太郎が死んだものと思い，その場に太郎を置き去りにして自動車で逃走したところ，太郎は，その夜凍死した．』

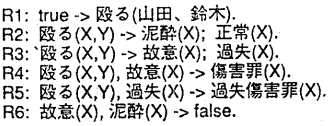


（a） リンカーン大統領暗殺の背景（事例ルール）
（a） An example of case rule



（b） ケネディ大統領暗殺の背景（新しい事件）
（b） An example of case facts

図4 類似論理構築での事実関係の照合
Fig. 4 The partial matching in 2nd phase of CBR.



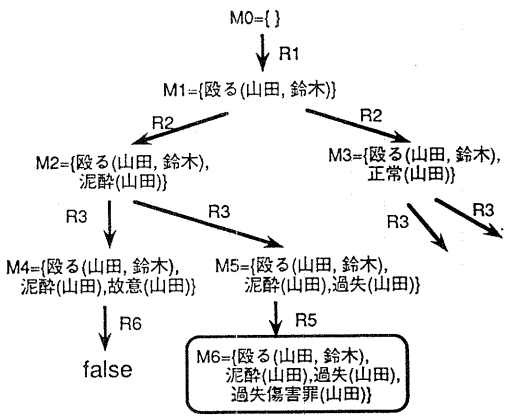


図5 MGTP によるモデル生成
Fig. 5 MGTP proof tree.

推論過程では各ルールの発火履歴が推論木データとして保持されている．これらの推論木の葉は与えられた事実関係の各要素であり，根は最終的な法的結論である．推論木は全体として，事実関係と法的結論の間をつなぐいくつかの中間的な法的概念の生成過程として構成される論理展開を表しており，これがシステムの出力として生成される．

この問題を，約200の条文ルールと約60の事例ルールと約100概念からなる概念辞書を用いて解かせたときに生成される推論木の一つを**図6**に示す．問題となる事実関係が与えられると，条文ルールにはまだ発火可能なルールがないので，まず事例ベース推論によって法的概念が生成される．事例ベース推論エンジンは事実関係の並列類似照合により，類似事例を検索し参照する．

例えば，交通事故を起こした者が病院に送り届けるために被害者をいったん保護したが，途中で翻意してその被害者を置き去りにして逃走したといった事例があり，その事例ルールとして，事故の加害者は被害者をいったん保護したという引き受け行為を行ったことにより，被害者の保護責任者であるとするルールが定義されていたとする．このルールを問題の事件に適用すれば，乙は太郎の保護責任者であったとする判断が導かれる．

他の事例から，置き去りにした行為を遺棄と認める判断が事例ベース推論により導かれると，刑法218条の条文ルールが発火可能となり，保護責任者遺棄罪が成立するとする結論が導かれる．さらに置き去りと太郎の死との間の因果関係を認める判断が事例ベース推論により導かれると，刑法219条の条文ルールが発火可能となり，保護責任者遺棄致死罪が成立するとする結論が導かれる．

この例題の場合，甲については保護責任者遺棄罪，保護責任者遺棄致死罪，等，乙については保護責任者遺棄致死罪，業務上過失致死罪，等あわせて12の罪が求まり，その約3分の1は市販の解説書に記されている模範解答に現れる，考えられる罪名と一致していた☆．

☆ 司法試験の問題には解釈が学説によって分かれるものが出題されることがある．したがってその答えも，どの学説を取るかで複数可能になる．

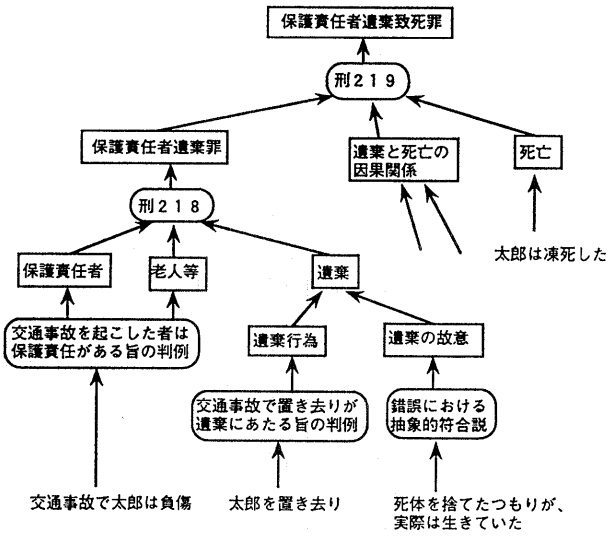


図6 保護責任者遺棄致死罪に対する推論木
Fig. 6 An inference tree.

7. HELIC-II の評価

HELIC-II は法的推論システムとしての側面と並列推論マシンの実証プログラムとしての側面を持つ．まず後者の側面について述べ，その後前者の側面について他の関係研究との比較を踏まえて述べる．

7.1 並列推論システムとしての評価

HELIC-II は並列論理型言語 KL1 によって書かれ，並列推論マシン PIM の上で稼働する．PIM の各プロセッサの利用効率を上げるためには，特定のプロセッサにアクセスが集中しないようにしたり，プロセッサの負荷が均等になるようにしたりするなどの工夫が必要である．本システムでは，判例の検索，条文の適用，否定の処理，仮説の処理，時間推論などに多くの並列性があり，それぞれの部分で並列効果を高めるようなプログラミングを行っている．

例えば事例ベース推論では図2に示したように，事例を異なるプロセッサに分配し，並列に検索・照合する．事例ルールは他の事例とは独立に処理できるので，事例を単位に処理を各プロセッサに割り振っている．これにより，各プロセッサごとの処理の独立性はきわめて高く，したがって高い並列処理効果が得られている．

6章で示した事件を例題として約130の事例データ☆を使って PIM/m[6] 上で，使用するプロセッサ数を

☆ 並列効果を計測するためのダミーデータを含む．

1から 64 まで変化させて実行したときの，二つの推論モジュールの台数効果を**図7**に示す．

台数効果は各プロセッサでの負荷バランスに大きく依存する．このグラフは比較的負荷バランスが良い場合の性能であり，事例ベース推論エンジンではほぼ台数に比例した性能向上が見られている．ただし負荷バランスが悪い場合にはこの図の半分程度まで台数効果が低下する．

ただしメモリー容量の関係から，この事例データでは8より少ないプロセッサ数では実行できなかったため，測定は8プロセッサから 64 プロセッサまでで行い，8より少ないプロセッサ数での台数効果はより小さい事例データを使って測定した．

データがある規模を越えると少ないプロセッサ数では実行できないという事実は，大規模な問題に対する並列推論の有効性を示しているといえる．また130事例が8プロセッサで実行できることから 64 プロセッサでは 1,040(=130×8) 事例まで実行可能と推定される．この規模であればかなり広い範囲の問題，例えば刑法なら重要判例をほとんど扱うことができると考えられ，これにより大規模や知識ベースを持つ知識処理システムの高速化の見通しを得ることができたことが並列処理システムとしての貢献と考える．

7.2 法的推論システムとしての評価

法律エキスパートシステムの開発は，AI の応用としては最も古い歴史を持つものの一つである．初期のシステムの代表的な手法は，法律の条文を論理式で表現し，新しい事件を命題の集合で表現して，演繹推論で結論を得るものであった．しかしこの手法では取り扱えない問題がしばしば生じた．その代表例は2章でも述べたような法律の解釈に関するものである．

解釈を実現する有力な方法として，判例を利用した事例ベース推論の技術が注目され，ルールベース推論と事例ベース推論の双方の推論モジュールを備えた法的推論システム[7)-10)]が研究されてきた．しかし判例の表現法および利用法には確立した手法がない．例えば Rissland[7)] は，判例をある種のフレームで表現し，Branting[9)] は意味ネットで表現しているが，判例の利用方法は HELIC-II とは異なる．

HELIC-II もまた二つの推論エンジンからなる hybrid システムである．判例を用いた事例ベース推論の手法は，意味ネットをベースにしている点で Branting のシステムに類似している．しかし HELIC-II は，

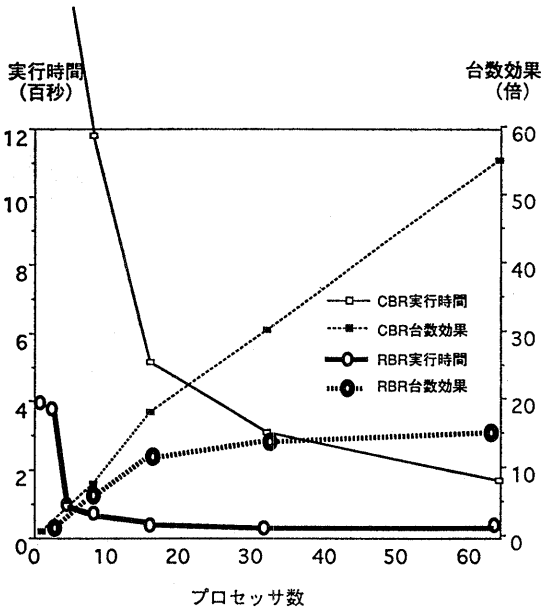


図7 CBR と RBR の台数効果
Fig. 7 Performance of CBR and RBR engine.

（1）判例記述の詳細度が Branting のシステムよりもはるかに高く，また時間情報や否定情報まで考慮している，（2）類似判断の方法が，リンクとノードの双方を考慮することでより詳細になっている．（3）そのため HELIC-II の方が類似判断に時間がかかるが，これを並列処理によって実用的な時間内で可能にしている．（4）事実関係から罪の成立に至るまでの論理展開を，単一の事例ルールではなく，複数の段階に分けてルール化したため，さまざまな抽象度での類似比較が可能となり，これにより判例データの抽象度のばらつきが吸収される．（5）判例の中で原告と被告の論理を両方とも事例ルール化して利用するため，両方の立場からの論理展開（推論木）を相互に比較することができる（**図8**）などの利点がある．

また条文を利用したルールベース推論においては，（1）2種類の否定を扱うことができ，（2）並列処理による高速な前向き推論を実現している．

その他二つの推論エンジンを組み合わせた hybrid システムとしては，（1）二つのエンジンがデータ駆動で並列に動作するため，Rissland のシステムのような複雑な制御機構が不要である，（2）判例を引用するごとに場合分けを行って，おのおのの場合ごとに条文を適用しているため，出力の整理が容易である．などの特徴がある．

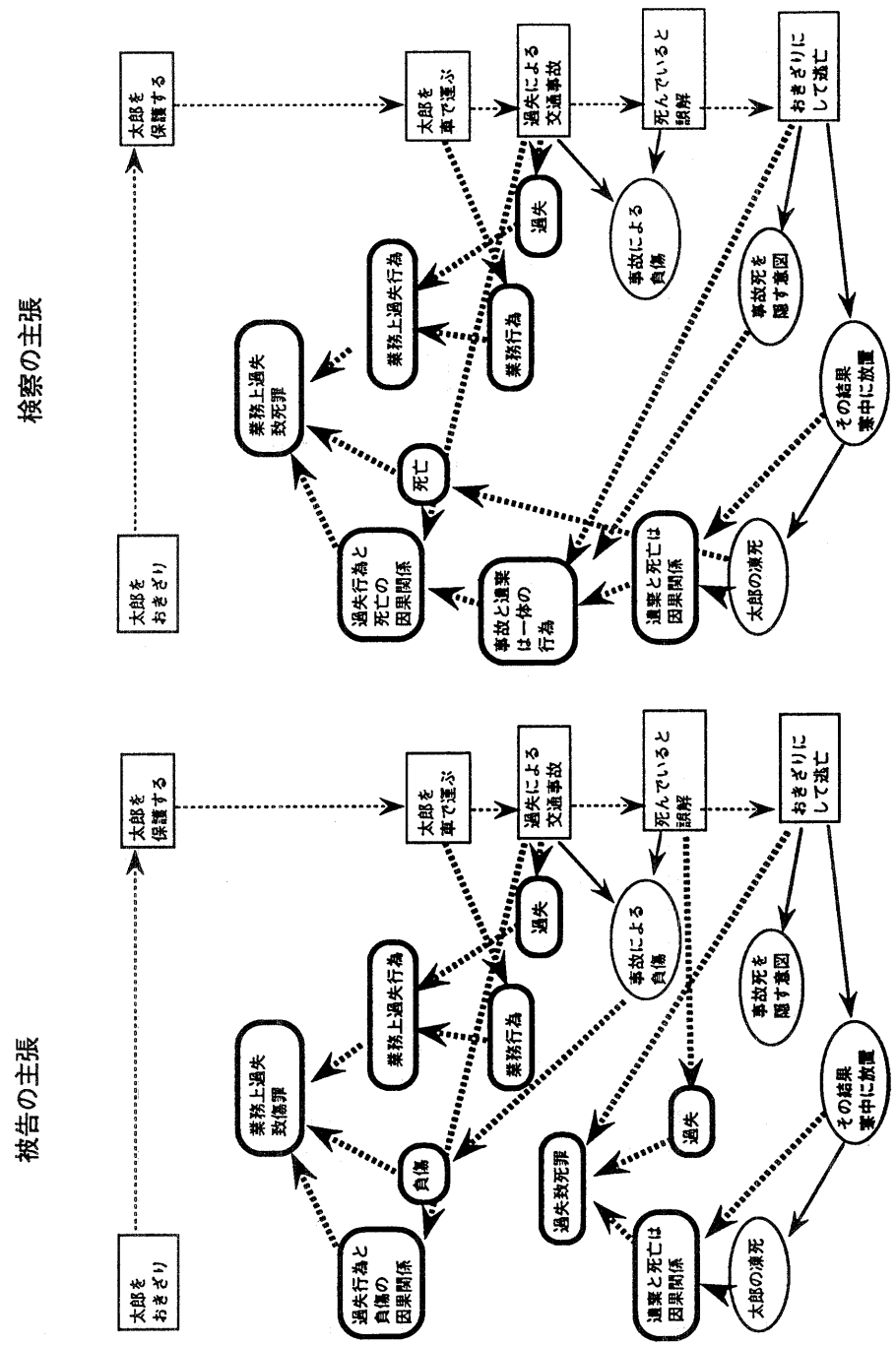


図 8 検察と被告それぞれの論理展開
Fig. 8 Inference trees from both side.

こうした推論システムとしての特徴だけでなく，法律分野での知識の表現方法の研究として，日本の刑法条文と刑法判例を分析し，ルールベース，判例の事例ベース，概念辞書などの知識ベースを試作し，司法試験の問題を解くのにどの程度の情報を判例から抽出すれば良いかの実験を行い，その結果を専門家にチェックしていただいたことにより，実用的な知識ベースを構築するための実質的な一歩となっている．

一般に意味ネットなどで情報を詳細に記述すれば，それだけ高度な精度の高い推論が可能になるが，半面多様性が増す分記述方法の統一が困難になるだけでなく，検索や推論に要する時間も増大する．判例からどのような情報を抽出し，どのようにそれを表現して利用するかは，一層の研究が必要である．

また現在は事例ベースがまだ小規模なものであるため，解ける問題の範囲はまだ刑法のかなり限定された分野のものに限られている．HELIC-II の問題解決には類似事例の存在は必須であり，適用可能な類似事例が事例ベース中に存在しない問題は解くことができない．したがって扱える事件の範囲を広げるには，事例ベースをさらに拡充する必要がある．

事例ベースが小規模なものにとどまっている理由は，主に事例データの作成が容易でない点にある．判例集[5]から典型的な判例を選択し，その文章から知識表現へ変換する作業は機械的には行えず，事例データの作成にはかなりの熟練が要求されるのが現状である．

8. おわりに

法的推論システム HELIC-II を紹介した．HELIC-II の研究の目的の一つは，ルールベース推論と事例ベース推論を備えた，より強力な法的推論のメカニズムを提供することである．HELIC-II では，

（1）基本的な枠組として不完全なルール集合（条文）と，ルールの運用実績である事例ベース（判例）の組合せによって問題を解く．これは柔軟な推論システムの構築手法として，法律のみならず多くの分野に適用可能である．

（2）事例ベース推論の中心的な役割は，二つの状況の類似判断である．これは非常に時間がかかる処理であるが，並列処理によって高速な照合を可能にした．

（3）事例ベース推論エンジンは仮説を生成し，ルールベース推論エンジンはそれぞれの仮説ごとに世界を分けてルールを適用する．したがって HELIC-II は並列の仮説推論システムと考えることができる．

などのように，知識処理に関する幅広い応用問題に適用できる特徴を有している．また今後の課題としては，

1. より詳細な知識表現

現在の枠組で刑法のかなりの範囲の問題が扱えるとの見通しが得られているものの，現在の HELIC-II 事例の事実関係の記述は，原判例の記述内容の核心部分を抽出して記述しており，その細部や様相の情報は切り捨てられている．そのため事例ルールとして論理展開の根拠が十分に記述しきれていない部分もあり，今後改良の余地がある．

2. 解釈生成機能

法的推論における解釈生成には幾つかの方法があり，過去の事例に基づくものはその一つにすぎない．例えば条文の類推解釈は，刑法では禁止されているが民法では典型的な解釈方法である．したがって民法の問題を解くためにはこのような手法も扱えることが必要である．

3. 論争機能

裁判での論理構築をする上で有益な情報を効果的に法律の専門家に提示するためのインタフェースとして，異なる観点に立った論争を模擬するというような機能を実現することは，システムの有用性を大きく高めるものと期待される．

謝辞 研究の機会を与えてくださったICOT渕一博所長（現，東京大学教授），内田俊一部長（現，ICOT所長）に感謝します．

参考文献

1) Allen, J. F.: Towards a General Theory of Action and Time, *Artif. Intell.*, Vol. 23, No. 2, pp. 123-154 (1984).
2) Fujita, H. et al.: *A Model Generation Theorem Prover in KL 1 Using a Ramified-Stack Algorithm*, ICOT TR-606 (1991).
3) Inoue, K. et al.: *Embedding Negation as Failure into a Model Generation Theorem Prover*, ICOT TR-722 (1991).
4) Nitta, K. et al.: HELIC-II: A Legal Reasoning System on Parallel Inference Machine, *Proc. Int. Conf. Fifth Generation Computer System*, Tokyo, Japan, pp. 1115-1124 (1992).
5) 大塚 仁編：判例コンメンタール（刑法），三省堂 (1976).
6) Uchida, S.: Summary of the Parallel Inference

Machine and its Basic Software, *Proc. Int. Conf. Fifth Generation Computer System*, Tokyo, Japan, pp. 33-49 (1992).
7) Rissland, E. L. et al.: Interpreting Statutory Predicates, *Proc. Int. Conf. on Artificial Intelligence and Law*, Vancouver, Canada, pp. 46-53 (1987).
8) Sanders, K.: Representing and Reasoning about Open-textured Predicates, *Proc. Int. Conf. on Artificial Intelligence and Law*, Oxford, UK, pp. 137-144 (1991).
9) Branting, K.: Representing and Reusing Explanations of Legal Precedents, *Proc. Int. Conf. on Aritificial Intelligence and Law*, Boston, USA, pp. 103-110 (1989).
10) Gardner, A.: *An Artificial Intelligence Approach to Legal Reasoning*, A Bradford Book, MIT Press (1987).

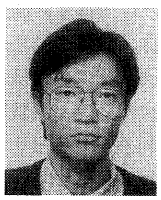
(平成5年4月12日受付)
(平成6年2月17日採録)

大嶽 能久（正会員）

1958年生．1982年京都大学工学部精密工学科卒業．1984年同大学院精密工学専攻修士課程修了．同年(株)東芝入社．1989年(財)新世代コンピュータ技術開発機構に出向．出向中は並列LSI-CADシステムおよび並列法的推論システムHELIC-IIの研究・開発に従事．1993年(株)東芝に復職．現在知識処理技術の研究に従事．

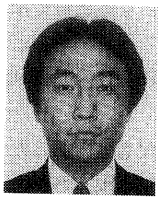
新田 克己（正会員）

1952年生．1975年東京工業大学工学部電子工学科卒業．1977年同大学院修士課程修了．1980年同大学院電子物理工学専攻博士課程修了．工学博士．1980年電子技術総合研究所に入所し，論理型言語の並列処理と応用の研究に従事．1988年弁理士試験合格．1989年～1992年に(財)新世代コンピュータ技術開発機構（ICOT）に出向し，並列推論マシンの応用プログラム（遺伝子解析，法的推論）の開発に従事．1993年4月から再びICOTに出向し，現在に至る．現在ICOT第2研究部部長．元岡賞，情報処理学会奨励賞，論文賞を各1回受賞．



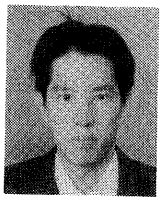
前田 茂（正会員）

1962年生．1985年北海道大学工学部電気工学科卒業．1987年同大学院情報工学専攻修士課程修了．同年(株)東芝入社．1989年(財)新世代コンピュータ技術開発機構に出向．1993年(株)東芝に復職．現在マルチメディア技術研究所において数値パターンを扱う知識処理技術の研究に従事．



小野 昌之（正会員）

1962年生．1985年上智大学理工学部化学科卒業．同年沖電気工業(株)入社．1989年(財)新世代コンピュータ技術開発機構に出向．1992年沖電気工業(株)に帰属．知識工学，並列処理に興味を持つ．現在技術情報システムの企画開発に従事．第43回情報処理学会全国大会奨励賞受賞．



大崎 宏（正会員）

1962年生．1985年青山学院大学理工学部化学科卒業．同年日本情報処理開発協会入社．ネットワーク関連の開発の後，知識獲得システムEPSILON，協調問題解決システムFREEDOMなどの知識処理システム構築に携わる．現在は法律エキスパートシステムHELIC-IIの概念辞書の構築に専念．

坂根 清和（正会員）

1957年生．1981年京都大学工学部電気工学第2学科卒業．1983年同大学院修士課程修了．同年新日本製鐵入社．現在エレクトロニクス・情報通信事業本部オートメーション事業部掛長．事例ベース推論と現代制御理論とを統合したプロセス制御システムの開発，制御プロセスのモデリング技術の開発に従事．1987～1991年(財)新世代コンピュータ技術開発機構に出向中は，定性推論，制約解析，定理証明器，法的推論システムの研究・開発に従事．システム制御情報学会会員．