

TM-1114

法的推論システム
HELIC-II

新田 克己、前田 茂、小野 昌之
大嶽 能久、坂根 清和、大崎 宏

September, 1991

© 1991, ICOT

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(1)

- 概要 -

大崎 宏† 大塚 能久‡ 新田 克己‡

†(財)日本情報処理開発協会

‡(財)新世代コンピュータ技術開発機構

1 はじめに

一般に、法律の専門家の行う推論は、法律条文だけを使って演繹的に行われるように思われがちである。しかし実際には、その前段階として、まず対象となる事件の具体的な事実を「解釈」を与えたり、抽象的な法的概念との関係づけが行われなければならない。その際法律の専門家は、過去の判例などを重要な知識源として使っている。このような推論を計算機上に実現するためには、過去の判例に基づいた推論機構が必要となる。

我々は法律の専門家が行うこのような推論を、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した。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た推論を行うというものである。

HELIC-II は、このようなモデルを計算機処理可能な形で具体化したものである。

また、事例に基づく推論では膨大な判例の検索を含む大規模な処理がなされる。そこで、HELIC-II を並列推論マシン Multi-PSI 上に構築することにより、それらを並列処理し高速化を実現している。なお、今回扱う対象は刑法における論理構築とした。

以下、本論文では HELIC-II の概要について述べる。

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を言う。端的に言えば事実認定や解釈を伴う推論であると言える。

では、なぜ事実認定や解釈が必要かを例を用いて説明する。

例えば、刑法条文の記述で「人を殺したるものは、殺人罪に処す」といったものがある。これをそのままルール形で書き表すならば、以下のようなものになるだろう。

IF 殺した(X,Y) THEN 殺人罪(X)

ここで「太郎が次郎を殺した」という「事実」がもし与えられたとすれば、太郎は殺人罪であることが結論づけられる。

殺した(太郎, 次郎) → 殺人罪(太郎)

しかし、実際の法的推論がこのような単純な推論によって行われているわけではない。なぜなら、「殺し

た」というのは法律上の概念であり、それが成り立つかどうかは事実を解釈することによって判断される。実際に与えられる具体的事実は、例えば「刺した」、「絞めた」、「毒を与えた」といったようなものである。そして、これらの事実が「殺した」という概念に相当するかどうかは必ずしも明らかではない。

このように、法律の条文は抽象的な概念に基づいて記述されている。そしてこのような概念には一般的な解釈ルールを与えることはできない。これらは個々の事例に基づいて、法律の専門家によって解釈されて始めて意味を与えられるものである。従って、特定の条文が選択され適用されるには、まずこのような抽象的な法的概念と具体的な事実を結び付けるという操作が必要となる。

法律の専門家は、このような判断の材料として、過去の判例などを重要な知識源として使っている。同様な推論を計算機上に実現する方法としては、過去の判例を事例とする、事例に基づく推論を用いるのが有望である。

3 HELIC-II の入出力

HELIC-II の入出力は以下である。

(1) 入力

事件の事実関係を入力する。

以下は事件の例である。

「甲女は、生後4ヶ月の息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、太郎に気づき、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、太郎に顔死の重傷を負わせた。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。」

この事件で争点となるのは、甲女が太郎を捨てた行為と、太郎が凍死した結果との因果関係、そして、乙が引き起こした自動車事故と太郎の凍死との因果関係である。これらの因果関係の認定方法は学説上の対立のある問題であり、判例の判断も一様ではない。

(2) 出力

入力された事件と類似した事例を調べて因果関係や故意などについての判断とその根拠を推論する。

次にそれらを使って条文を調べて罪状とその理由付けを出力する。利用者は、その結果を見て自分に有利な論理構築や相手の反論に役立てる。

4 推論方式

HELIC-II の推論は、条文に基づく推論と事例に基づく推論の2種類の推論方式からなる。

以下にそれぞれの推論方式について説明する。

4.1 条文に基づく推論

法律の条文は抽象化された法的概念で論理的に記述されている。そのため、条文に基づいた推論は論理的な推論である。これを実現するために並列定理証明器 MGTP (Model Generation Theorem Prover) [2] を拡張した推論エンジンを開発した。本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と、矛盾検出機能である。これらの機能は、法律の論理的な側面を利用するのに適している。

各条文からは、論理式の形で表現された条文ルールを抽出しており、推論エンジンはその条文ルールを適用して推論を進めていく。

4.2 事例に基づく推論

与えられた事件の事実関係と抽象的な法的概念との対応づけは、以下の2段階で行われる。なお、ここに出てくる事例ルールとは、判例から事実関係の解釈や論点に応じた論理展開を取り出したルールである。

- (1) 類似事例の検索
まず、与えられた事件と事象の並びが類似した事例を検索する。
- (2) 事例ルールの実行
次に、選択された類似事例中の事例ルールによって事件と抽象的な法的概念との結び付けを行う。

ここで事例ルールと条文ルールの違いを表1に示す。

事例ルール	条文ルール
個別判断	一般的なルール
具体的事実	抽象的概念
条件部は類似性による照合	条件部は完全なマッチング
適用結果は仮説的	適用結果は真

表1: 事例ルールと条文ルールの違い

5 システム構成・動作

HELIC-II のシステム構成を図1に示す。全体の動作の概略は以下である。

- (1) 与えられた事件の事実関係が入力として与えられる。システムの内部表現としては意味ネット表現が用いられているが、入力形式としては、フレームタイプの知識表現言語を用いている。事例データについても同様である。
- (2) 事例に基づく推論を行う事例ベース推論エンジンにより、過去の事例から類似事例が検索され、それらの事例ルールにより、事件の事実関係に解釈が与えられる。現在、約30事例が登録されている。
- (3) 事件の事実関係と法律上の概念との対応づけがなされると、条文に基づく推論を行うルールベース推論エンジンが条文ルールを適用する。刑法の条文から約200ルールを抽出し、条文ルールとして使用している。
- (4) 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領域(WM)を介して通信する。
- (5) 問題となる事件の当事者に対する可能な罪状を列挙する。同時に、それらの結論に至る推論過程をその根拠として出力する。

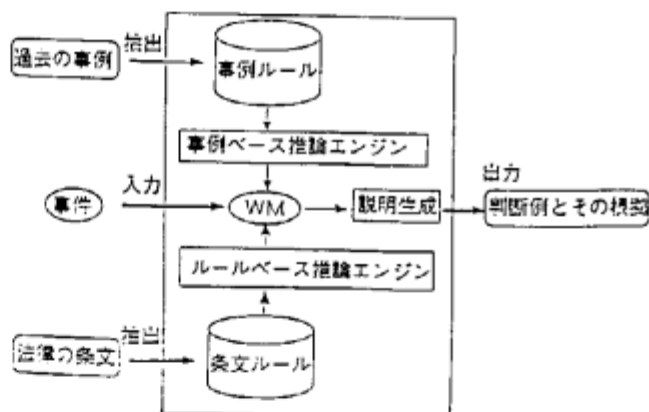


図1: システム構成

6 おわりに

条文に基づく推論と事例に基づく推論という2つの異なる推論方式を組み合わせた法的推論システム HELIC-II の概要について述べた。現在、事実表現、条文/事例ルールの記述形式、および推論方式についての見直し作業と、システム第2版の設計・試作を進めている。

参考文献

- [1] 新田 他: "事例を用いた法的推論とその並列化", 情報処理工学と人工知能研究会 69-5, 1990.
- [2] 長谷川 他: "KL1 による1階述語論理ブルーバ", KL1 Programming Workshop '91, 1991.

法的推論システム HELIC-II(2)

- 類似事例検索手法 -

小野 昌之 新田 克己

(財) 新世代コンピュータ技術開発機構

1 はじめに

過去の判例を使って新たな事件の問題を解決するには、単にその判例で使われた判断を適用するだけでは不十分であり、その判断に陽に記述されない条件や事件の時間的な流れを考慮する必要がある。また、事例の数に伴って増加する判断の適用にかかるコストを減らすことも必要である。その為、我々は法的推論システム HELIC-II における事例に基づく推論に、類似事例照合部と類似論理構築部の二段階の照合処理を導入し、過去の類似事例を利用して新たな事件での罪責とその理由付けを行わせている。この類似事例照合部では、事件の背景や事実の時間的な流れを比較して、類似事例の検索を行う。また類似論理構築部では、その検索された事例において使われた判断を用い、新たな事件に対する罪責やその理由付けを生成する。

類似事例照合部における類似事例検索は、新たな事件と事例の事実関係について意味ネットワークで表現した個々の事実の上位概念とその時間関係の類似を調べることによって実現している。またこの検索処理は、並列推論マシン Multi-PSI 上で高速な並列検索を行っている。本論文では、この類似事例検索方式と並列処理による効果について述べる。

2 類似事例検索

2.1 類似事例検索の問題点

HELIC-II において事例は、事実とその間の時間関係及び事例ルールで表現される。以下に簡単な例を使って、事実とその間の時間関係を示す。

事実：殴る (A,B,t1), 川に捨てる (A,B,t2),
水を吸う (B,t3), 死亡 (B,t4), 逃げる (A,t5)
時間関係：t1 < t2 < t3 < t4, t2 < t5

殴る (A,B,t1) とは A が B を区間 t1 の間に殴ったことを示す。また、t1 < t2 とは t2 は t1 より時間的に後に起こったことを示す。

A Searching Method of Similar Cases in a HELIC-II
Masayuki Ono, Katsumi Nitta
ICOT

類似事例検索とは、類似する事実関係を持つ事件を見つけることである。二つの事実間の類似点を探すのに、意味ネットワークと上位概念を用いた類似 [新田ほか 90] は有効である。しかし時間の取り扱いを考えていない為に、時系列方向の行為者・被行為者関係の整合をとれないといった問題が生じる。例えば、「A が B を殴り、A が B を蹴った」と「C が D を殴り、D が C を蹴った」は、上位概念は同じであるが行為者・被行為者関係が異なる。この為、時系列に移り変わる行為者・被行為者関係について時系列での一致をとる必要がある。

また事例間の事実の時間関係の比較の際では、時間的に前後逆転しているイベントを類似とすることがある。例えば、時間を区間で表現している為におこる区間が重なり合ったイベントをどのように類似と扱うか、つまり例のように時間軸が複数ある場合の問題がある。また事例では、認定された事実のみが事実として残る為、事例と新たな事件では記述内容に差がでてくる。照合の際にその扱いが問題になる。これらの問題を考慮した類似事例検索が必要である。

2.2 類似事例検索手法

前記で述べた問題を解決する為、類似事例照合部では事件と事例の事実関係の照合の際に、上位概念方向の類似、時系列方向の類似、時系列に移り変わる行為者・被行為者関係についての時系列での類似の三つの類似性を同時に見ることとしている。

上位概念方向の類似では、意味ネットワークと上位概念を用いた類似照合 [新田ほか 90] をベースに、人と人以外のものを区別する機能を加え拡張している。HELIC-II では共通の法的概念辞書を持つ為、それを使った上位概念による類似照合は大変難しくなっている。なぜなら、法的概念辞書を用いた場合、探索の深さによって類似であったりそうでなかったりするからである。例えば、「人が人を殴った」と「人が犬を殴った」を比較する場合、客体である人と犬の上位概念は同じ動物であるが、法的解釈において通常、これを類似と扱わない。そこで、通常、法的な解釈において類似と扱うべきではないもの、つまり人と人以外のものに対して、区別する機能を付けている。

時系列方向の類似では、事件、事例それぞれの事実の流れを類似時系列リストで表し、まずこの二つの間で時系列類似照合を行う。次にこの中で最も類似点が多い照

合パターンを求め、類似の判断を行っている。この時系列類似照合では、二つのリストを先頭から順に上位概念方向の類似照合を行い、もし類似していなければどちらかの対象イベントを一つずらし同様な照合を繰り返すといった効率の良いアルゴリズムを用いている。また擬似時系列リストとは、時間を区間で表現している為に起こる時間的な重なりや時間的に定義されないイベント関係を、擬似的に時系列で並ぶようにしたものである。前記の例の場合、[殴る, 川に捨てる, 逃げる, 水を吸う, 死亡]で表す。しかし、このリストを用いて時系列方向の類似照合を行わせる場合、「逃げる」と「水を吸う」のような時間軸が異なる場合には、正確な照合が行えない。そこで、擬似時系列リストを用いた照合を行った後に、区間の重なりや時間的に等価であるイベントについて調整を行う方式を採用している。また、照合の際、時間的に連続したイベントに対し、一対多、もしくは多対一の照合を行い、判例と事件の記述の差を吸収する。そして、これら二つの類似照合によって、時間的に逆転したイベント間でも、上位概念の一致さえあれば、時間的に類似と扱えるようにしている。

時系列に移り変わる行為者・被行為者関係についての時系列での一致では、時系列方向の類似の際に、行為者、被行為者に対応する人や物を事件・事例間で一致させたものを記憶し、後に起きたイベントの照合の際に利用するようにしている。

こうした照合の結果、類似事例照合部では、事件・事例間の照合パターンが幾つか求められる。そして、この照合パターンを評価関数で評価し、最も類似しているとされる照合パターンを事件・事例間の類似点とする。図1に、簡単な記号を用いた時系列リストに対する類似照合を示す。リスト中の $a(x,y)$ は x が y に a という行為をしたという意味を表す。また、照合パターン中の $-$ は類似しなかったことを意味する。

事件: $[a(x,y), b(x,y), a(x,y), c(x,y)]$
事例: $[a(x,y), a(y,x), b(x,y), c(x,y), c(x,y)]$

照合パターン: $\begin{matrix} \swarrow & \searrow \\ [a(x,y), b(x,y), -, c(x,y)] & [-, a(x,y), c(x,y)] \\ [a(x,y), -, b(x,y), c(x,y), c(x,y)] & [a(x,y), -, -, c(x,y), c(x,y)] \dots \end{matrix}$
類似点

図1: 類似照合

事例ルールは罪責の理由付けの為に、主にイベント間の関係などを定義したものである。類似事例照合部で推論された類似結果には、新たな事件の一部に対し事例全体が類似する一部類似と、新たな事件全体と事例全体が類似する全体類似がある。事例が全体類似であれば全ての事例ルールを必要とするが、一部類似なら類似していない部分に関する事例ルールは必要ない。そこで、類似と判断されたイベントを一つ以上事例ルール的前提部に含む事例ルールを、類似論理構築部へ送るようにしている。図2に類似事例照合部の概要を示す。

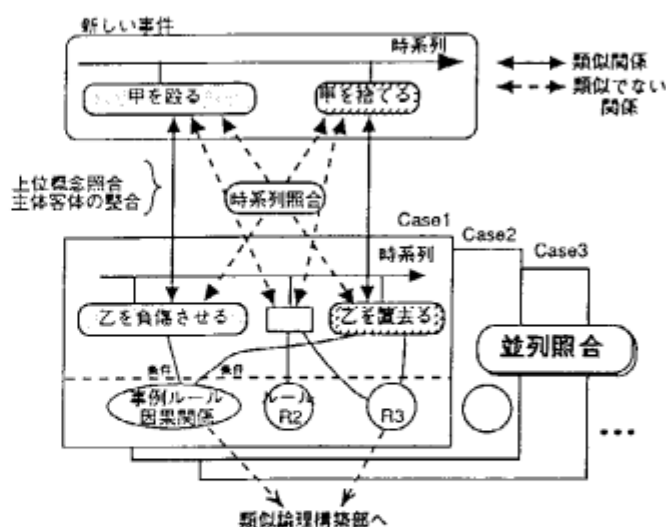


図2: 類似事例照合部

3 並列処理

HELIC-II は並列論理型言語 KL1 で記述されており、並列推論マシン Multi-PSI 上で推論を高速に実行する。類似事例照合部では、新たな事件と一事例の類似照合処理がそれぞれ独立に実行可能な為、この処理単位に並列化を行っている。現在、370 の評価用事例を用いた時の最も良い結果は、16 プロセッサの時に約 10 倍の速度向上である。これは 1 つ 1 つの事例照合の計算量が多くない為に、多くのプロセッサを使い切れないことを示している。しかし、事例の数に適したプロセッサを用意すれば、十分に高速化が得られると考えられる。このことから 64 台の PE を利用した場合、1000 以上の事例を扱うことができると予測され、大規模事例ベースにも十分耐えうる類似事例照合処理と思われる。

4 終わりに

多くの判例の中から類似している事例を効率良く検索することは法的推論にとって必須である。本論文では新たな事件と事例の間の類似判定に、上位概念方向の類似、時系列方向の類似、行為者・被行為者関係についての時系列での類似を利用した類似事例照合手法とその並列処理について述べた。しかし事例検索の際に、実際の裁判で論争になる部分(錯誤、因果等)の類似を考慮していない為、選択した事例の正当化が十分ではないと思われる。また負荷分散の際に、個々の照合処理における計算量は事例の中の事実の数とその複雑さの積に比例する為に、処理の負荷が均一になっていない。これらの改良を今後の課題としたい。

参考文献

[新田ほか 90] "事例を用いた法的推論とその並列化", 情報処理工学と人工知能研究会 69-5, 1990.

法的推論システム HELIC-II (3)*

- 事例ベース推論を用いた判例に基づく論理構築手法 -

前田 茂

新田 克己†

(財) 新世代コンピュータ技術開発機構‡

1 はじめに

法的推論システム [前田ほか 91] とは、法律の専門家が行う推論をモデル化し、計算機上で実現するものである。法律の条文に表れる抽象的概念と与えられる具体的な事件の事実の間の抽象度のギャップがあるために、従来のエキスパートシステムのように条文をそのままルールとして記述し、それを直接適用する手法を用いることは一般に困難である。

したがって、具体的事実と法的抽象概念とを結び付ける推論手法が必要となる。これは、条文の解釈の問題として捕らえることもできる。我々は、その手法として事例ベース推論が適切だと考える。つまり、与えられた事件の事実に対して類似の過去の事件を判例から検索し、そこでなされた判断を現在の事実に応用することにより、より抽象的な概念を作り出すことによってこれを行う。なぜなら、判例によってなされた判断は、抽象的な概念を生成するものであるからである。たとえば、ある行為とある結果の間に法的な因果関係が認められるという判断を行った判例があった時に、それを類似適用することによって、現在の事実の間に因果関係という抽象的な概念を作ることができる。

我々は、この判例の中の判断を事例ルールという形で表し、部分照会と概念階層辞書を用いた照会による類似検索により、判例の中の判断を利用して現在の事実と条文の中の抽象的な概念を結び付ける手法を考案した。また、事例ルールの推論連鎖により論理構築を行う。さらに我々は、この手法を ICOT の開発した並列推論マシン Multi-PSI 上で実現したので、その並列処理による速度向上の評価と共に報告する。

2 条文の解釈問題

法的推論とは、一般に条文を形式的に適用することによってなされるように考えられる。けれども、与えられた事件の事実関係の表現は具体的な概念で記述されており、これを直接適用することは難しい。たとえば、刑法 211 条で定義される業務上過失致死罪は、以下のように定義されるが、今回扱った事件 [大崎ほか 91] の事実である乙の運転行為が業務上の行為であるかどうか等の問題や、乙が太郎を放置した行為と、太郎の死亡との間に因果関係があるかどうかの問題は明確なルールがある訳ではないので、ルールによってその適用可能性を検証することはできない。

刑法第 211 条: 業務上過失致死罪

業務上必要な注意を怠り、よって人を死傷に致したものは、五年以下の懲役もしくは禁固または八千円以下の罰金に処す

したがって、このような具体的事実を元に、これを法的な抽象的な概念に結び付ける推論が必要である。これが、条文の解釈問題である。

*Making legal argument with cases using Case-Based Reasoning

†Shigeru MAEDA, Katsumi NITTA

‡Institute for New Generation Computer Technology

我々は、この問題を過去の判例によってなされた判断を用いることにより解決する手法を提案する。すなわち、新しい事実が与えられた時に、過去の類似の事件を検索し、そこでなされた判断、理由づけを用いることにより、新しい事実と法的な抽象概念を結び付ける (図 1)。

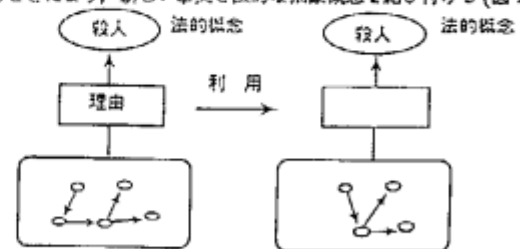


図 1: 判例を利用した法的解釈の生成

ここでは、類似事例の検索、適用の問題がある。つまり、過去の事例と現在の事件の間の類似性をどう判断するか、その類似性にしたがって過去の事例の判断をどう適用するかといった問題である。

我々は、この問題を解決するために、まず、過去の事例の判断の部分を判例から抽出し、4章で詳しく述べるような事例ルールという形で記述する。また、事例の事件および与えられる事件は、意味ネットワーク表現で記述することとした [新田ほか 90]。そして、基本的にはプロダクションシステムの推論サイクルをもちいる。類似検索については、意味ネットワークのオブジェクト、リンクの各々に対して概念辞書を用いて上位概念をたどり、ネットワークの部分照会を行うことでこれを行うこととした。そして、類似性を計算し、類似していると判断された場合は、事例ルールの後件部を概念辞書を用いて現在の事件の事実と表れるものに置き換えることで過去の事例の判断を適用する。

3 論理構築部の構成

図 2 に論理構築部の Multi-PSI 上での構成を示す。PE (Processor Element) 0 は、事例ルールの実行部の処理のプロセスを行う。その他の PE では、事例ルールの条件照会プロセスを並列に行う。ここでは、PE 毎に概念辞書を持ち類似照会を行う。これはプロセッサ間通信をなるべく減らすためである。

また、一般に一つの判例からは数個の事例ルールが抽出できるが、プロセッサへの割り当ての最小単位は判例単位である。これにより負荷の均等化がはかれる。なぜなら、一つの判例の中では事例ルールは具体的事実から徐々に抽象的な概念を生成するものへと細分されているので、推論の時系列にしたがって、最初は低い抽象度の概念を生成する事例ルールが実行し、次第に高い抽象度の概念を作り出す事例ルールが実行することが期待される。したがって、各 PE にほぼ同じ抽象度の概念を生成する事例ルールを分散させることにより、負荷が均等化される。

このようにプロセスを各 PE に割り当てた後、与えられた事件が

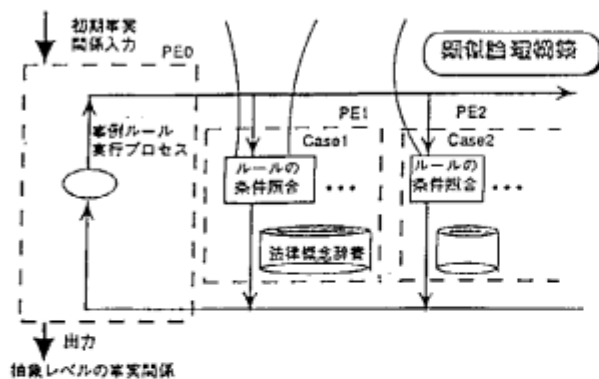


図 2: 推論構築部の構成

PE0 に入力されると、それらは各ルール照合 PE に送られ、事例ルールとの類似照合が行われる。発火した事例ルールの後件部で作られるべき概念は、すべてまた、PE0 に送られ、そこで与えられた事件に適用される。そしてまた以上のサイクルを繰り返す。このような推論サイクルによって次第に抽象的な概念が生成され、最終的には法的概念が生成される。

4 事例ルールによる論理構築

以下では、与えられた事件に対して次のような事件が類似の判例としてあったとしよう。

殺意を持って麻縄で首を絞めると、被害者が身動きをしなくなったので、死亡したものと思った被告人が、被害者を海辺の砂の上に放置したところ、被害者が砂を吸い込んで死亡した。

(判例: 大判大一二・四・三〇集二: 三七八)

ここで、判例の中の判断として被告が被害者を砂の上に放置したこと、被害者が砂を吸い込んで窒息死したこととの間に因果関係が認められたことは、以下のような事例ルールとして表される。

```

(rule001, 第 199 条, [因果関係条件説], "因果ありルール",
  [[放置する,
    (主体 = 被告, trivial),
    (客体 = 被害者, trivial),
    (結果 = 吸引する, important)],
  [吸引する,
    (主体 = 被害者, trivial),
    (結果 = 窒息死, important)],
  [窒息死,
    (主体 = 被害者, trivial)],
  after(放置する, 窒息死)]
=>
[(make, [放置する, (因果 = 窒息死)])])
  
```

かっこで始まる最初の 4 要素は、それぞれ、ルール ID、関連する条文名、関連学説、コメントである。次のリストの部分が判断に用いられた事実関係である。それは、意味ネットワークの一部分と、after などの部分意味ネットワーク間の時間関係からなる。そのほかに、概念辞書をたどる上位概念を制限する limit、照合した事実が異なることを表す different を記述できる。

事件が与えられた時にはまず、概念辞書を用いて事例ルールの個々のオブジェクト、リンクとの類似照合を取る。概念辞書としては、乙、太郎、被告、被害者は自然人、放置すると乙の置き去りは放置行為、窒息死と凍死は死などの上位概念を持つとする。したがって、事例ルールの前件部の各要素の大部分は与えられた事件と照合する。

次に、前件部の各要素の重みを考慮して事例ルール全体の類似度を計算する。重みは、exact, important, trivial で記述し、それぞれ、必ず照合しなければならないもの、照合に重要なもの、それほど重要でないものを表す。その類似度の計算は以下の通りである。

二つのオブジェクトまたはリンクの概念階層間の距離を d 、概念階層辞書の平均階層長を L とすると、概念辞書における両者の距離の割合 (同一概念の場合 1, 上位概念がルートしかない時 0) を M とすると、 $M = d/2(1-L) - L/(1-L)$ である。次に、各前件部要素の重みを W とすると、trivial, important のときの相対的加重はそれぞれ、 $W_i = 1/n(m+1)$, $W_i = 1/(M+1)$ である。ここで、 m, n はそれぞれ前件部内の important, trivial の個数であり、trivial 要素はすべて照合しても、たかだか important 要素一つの重みしか考えられないという仮定にたっている。また、前件部の要素に覆れるオブジェクトとリンクの合計は 3 なので、事例ルールの類似度 S は次の通りになる。

$$S = \sum_{j=1}^{m+n} (W_j * \frac{\sum_{k=1}^3 M_{jk}}{3})$$

各々の事例ルールに対して類似度を計算し、それが与えられた関連値を越えたルールを発火させる。そして、概念辞書を用いて事例ルールの後件部を与えられた事件に適用する。ここでは、放置するに対して置き去り、窒息死に対して凍死が対応するので、この事例ルールを用いると乙の置き去り行為と太郎の窒息死の間に因果関係があるという判断を行うことができる。このようにして、事例ルールを次々に適用することで、法的な論理構築を行う。

5 並列処理の評価

実験システムを、ICOT で開発された並列論理型言語 KL1 で記述し、並列推論マシン Multi-PSI 上で評価を行った。事例数は約 130 を入力した。メモリー容量の関係から、1PE では実行できない規模であったので、8PE から 64PE までで行った。このことから、大規模な問題での並列処理の有効性が示された。ここではほぼ台数に比例する効果が現れている。

PE 数	8	16	32	64
時間 (sec)	1181.7	516.2	309.1	170.4
台数効果	1.00	2.29	3.82	6.93

表 1: 大量事例での計測結果

6 おわりに

法的推論における条文に覆れる抽象的概念と現実の事実の間のギャップを埋めるために CBR の手法を用いて、事例ルール表現と概念辞書と部分照合による類似照合により問題を解決した。また、事例ルールの逐次適用により論理構築を行う。そして、並列推論によって効率化をはかった。

参考文献

- [前田ほか 91] 法的推論における並列推論, SWoPP '91, AI-77-6, 1991.
- [大崎ほか 91] 法的推論システム HELIC-II(1) - 概要 -, 情報第 43 回全国大会, 1991.
- [新田ほか 90] 事例を用いた法的推論とその並列化, 情報処理学会, 知識工学と人工知能 69-5, 1990.

法的推論システム HELIC-II (4)

並列定理証明器 MGTP を用いた法律条文に基づく推論

坂根 清和

新日本製鉄(株) ソフトウェア技術センター

新田 克己

(財) 新世代コンピュータ技術開発機構

1 はじめに

法的推論システム HELIC-II は、法律条文に基づく論理的な推論と、過去の判例に基づく推論(事例に基づく推論)とを相補的に用いることで、与えられた事件の事実関係に対する当事者の罪責に至る理由づけの説明を生成する。

条文に基づく推論は、並列定理証明器 MGTP(Model Generation Theorem Prover)[1]を法的推論用に拡張して実現した。各法律条文を論理式で表現することにより、MGTP のモデル生成過程を、論理的な説明生成過程として用いている。また、事実関係の相互矛盾、学説間の対立関係を論理式で表現することにより、複数学説環境の各々において矛盾のない説明の生成を行なう。さらに、並列推論マシン Multi-PS1 上で並列実行することにより、高速な推論を実現した。

2 並列定理証明器 MGTP

定理証明器 MGTP は、モデル生成法に基づく一階述語論理のための並列定理証明システムである。MGTP は並列論理型言語 KLI[2]で実装されており、与えられた領域限定(論理式の出でくる全ての変数が、前件部に現れる)非ホーン節問題を効率良く解く。MGTP に与える節は次のような含意式である。

$$A_1, \dots, A_n \rightarrow C_1; \dots; C_m. \dots (1)$$

ここに、 A_i, C_j は原子論理式で、前件部は連言を表し、後件部は選言を表す。モデル生成法では、与えられた節集合のモデルを、空集合から始めて次のモデル拡張則にしたがって構成的に求める。

- モデル拡張規則: 一般節 $A \rightarrow C$ において、置換 σ のもとに前件部 $A\sigma$ が M で充足されており、後件部 $C\sigma$ が M で充足されていないときは、モデル M に $C\sigma$ を加えて拡張する。
- モデル棄却規則: 負節 $A \rightarrow false$ において前件部 $A\sigma$ が M で充足されているとき、 M を棄却する。

MGTP の特徴としては、

(a) KLI 言語による効率的プログラミング; MGTP は変数を KLI の変数で表して、モデル生成時に先だてて各節をその前件部を KLI のガード部、後件部をボディ部で表現した KLI 述語に変換する。前件部の照合を KLI の高速なガードユニフィケーションと節インデクシングで実現している。

(b) 連言照合の冗長性の除去; モデル M に新たな要素 δ が追加されて行く過程で、連言照合に RMS(Ramified Stack Algorithm)[1]方式を用いて冗長な照合を回避する。

3 法律条文に基づく推論の知識表現

(i) 法律条文の論理表現

法律の各条文は、その適用条件 A_i を述べた部分と、その場合の解釈、刑罰などの結果 C_k を述べた部分から構成される。適用条件のうちには、「 $\sim B_j$ でなければ、 $\dots$ 」のような、否定の表現を含む。MGTP の各節は、全て正のリテラルで表現されるので、法律条文の論理的に等価な表現は、

$$A_1, \dots, A_n \rightarrow B_1; \dots; B_p; C_1; \dots; C_m. \dots (2)$$

となる。HELIC-II 内部では、新たな事件、過去の判例の事実関係を、{対象, 関係, 値} の3つ組で記述する意味ネットワークで表現しており、(2)式の A_i, B_j, C_k は3つ組形式の原子論理式である。

(ii) 論理的矛盾検出機能

事実関係には、相互に矛盾する関係がある。HELIC-II では、法律の要件を考慮する上で、重要となる矛盾関係を、負節で表現して公理系に与えた。たとえば、犯罪の構成要件として、結果に対する行為者の意図の存在が重要である。行為 A と、結果の状態 B との間には、意図があるか、意図がないかのいずれかである。両方の関係が推論で導かれた場合、矛盾であるので負節で表現する。

$$\{\text{行為 } A, \text{意図, 状態 } B\}, \{\text{行為 } A, \text{意図なし, 状態 } B\} \rightarrow false. \dots (3)$$

(iii) 法律上の対立学説の表現

法律条文の構成要件の解釈や、事実関係の解釈において、確定した通説がなく、いくつかの法学上の学説が存在する場合がある。たとえば、行為と結果の状態との間の「法律上の因果関係」は重要である。しかし、どのような場合に因果関係が認められるかについては、必ずしも明確でなく、以下の2つの学説が対立している[3]。

- 因果関係条件説: 結果の状態が生じる前提条件の1つとして、ある行為が関連している時、その行為と状態との間に法律上の因果関係を認める。
- 相当因果関係説: 社会通念上一般に因果関係が認知されている場合にこれを法律上の因果関係と考える。

2つの学説は対立するので、モデル生成時に推論を行なう環境を2つに分ける必要がある。次の非ホーン正節を MGTP に与える。

$$true \rightarrow \{\text{学説, apply, 因果関係条件説}\}; \\ \{\text{学説, apply, 相当因果関係説}\}. \dots (4)$$

4 条文に基づく推論用の MGTP の拡張

HELIC-II の条文に基づく推論の推論エンジンとして、MGTP を用いるために、以下の拡張を行なった。

(i) モデル導出過程の説明機能

法的推論においては、その最終結果のみではなく、それに至る導出過程の説明が重要である。モデル内の各原子論理式に対して、それがどの原子論理式から、どの節の帰結部として得られたかの履歴情報を付加する機能を加えた。具体的には、各節を KLI 述語に変換する時に、後件部の原子論理式を justification つきの表現で表す。たとえば、節番号 num1 の論理式

$$num1; p(X), q(X, Y), r(Y, Z) \rightarrow s(X, Z). \dots (5)$$

に対して、後件部の $s(X, Z)$ の代わりに $(s(X, Z), num1, [p(X), q(X, Y), r(Y, Z)])$ を用いる。モデル生成時には、この節の前件部を満足する変数 X, Y, Z の値がユニファイされるのみで、推論速度に影響を与えずに $s(X, Z)$ の justification が得られる。この justification により、初期事実関係から最終結論に至る説明木を構成する。また、説明木のグラフィック表示機能を付加した。

(ii) 論理式表現の拡張

法律の要件、適用結果は通常複数の関係で記述される。この論理表現は、前件部のみでなく、後件部も連言形式で表現する必要がある。(2)式の後件部の連言形式の各々を連言形で記述できるように表現上の拡張を行なった。次式で、後件部〔〕内は、連言の意味である。

$$A_1, \dots, A_n \rightarrow [C_{11}, \dots, C_{1p_1}]; \dots; [C_{m1}, \dots, C_{mp_m}] \dots (2')$$

5 MGTP による条文に基づく推論

HELIC-II では、MGTP を具象化項の集合のデータを公理系に与えて、それから導かれるモデルを推論結果として用いる前向き推論エンジンとして使用する。

判例に基づく推論の出力である新たな事件の解釈事実は、データストリームとして条文に基づく推論に伝えられる。条文に基づく推論では、KL1 言語の組み込み *merger* を用いて、正箇で宣言された事実関係と判例に基づく推論から到着した事実関係をマージする。マージされた全初期事実集合をモデル生成過程に与え、順次各節の条件部の連言照合に用いる、ストリーム並列を実現している。MGTP のモデル生成過程は、初期事実関係集合から導出可能な全ての事実関係を含む推論結果をモデルとして得る。図1に条文に基づく推論の流れを示す。

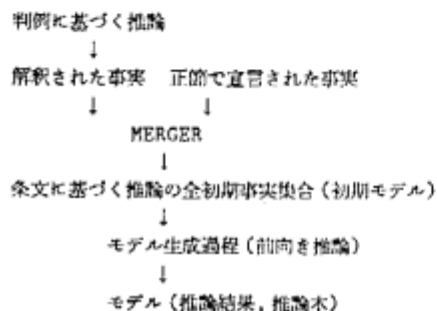


図1 条文に基づく推論の流れ

判例に基づく推論から出力される解釈事実は、その解釈がどの学説に基づくかという、“背景学説”を付加情報として持つ。モデル生成の木は(4)式で示すような対立学説環境に従い分岐して、判例に基づく推論による解釈事実のうち、各推論木の枝の学説環境に適合する解釈事実のみをモデルに加える。

6 推論結果

今回の実験システムは、刑法の条文のうち、遺棄、殺人、傷害、暴行、窃盗、強姦、住居侵入などの罪に関する各論理ルール、および、罪数関係、共犯、阻却事由などの刑法の総則ルール、ならびに、通説、学説に関する解釈ルールなど合計約200個の論理式を含む。

“甲女の事件”[4]の例題に対して、条文に基づく推論では、相当因果関係説の下では、乙が太郎を保護した時点で、乙の太郎に対する保護責任が生じるため、これを置き去りにすることにより、保護責任者遺棄罪が成立立つことが示された。因果関係条件説の下では、これに加えて、乙が起こした交通事故と太郎の凍死の間には、因果関係の存在が認められるので、“乙の起こした交通事故は、業務上(運転中の)過失致死罪に相当する”との推論結果が得られた。図2に乙の業務上過失致死罪を導く証明木の一部を示す。

7 並列推論による高速化

MGTP を用いたモデル生成アルゴリズムはいくつかの並列性を内在している。

(a) モデル分岐時の OR 並列性: 非カーン節によるモデル拡張時に分岐したモデルは、値域限定条件のために共有変数を持たないので、以後のモデル拡張を全く独立に行なうことができ自然な OR 並列性を提供する。

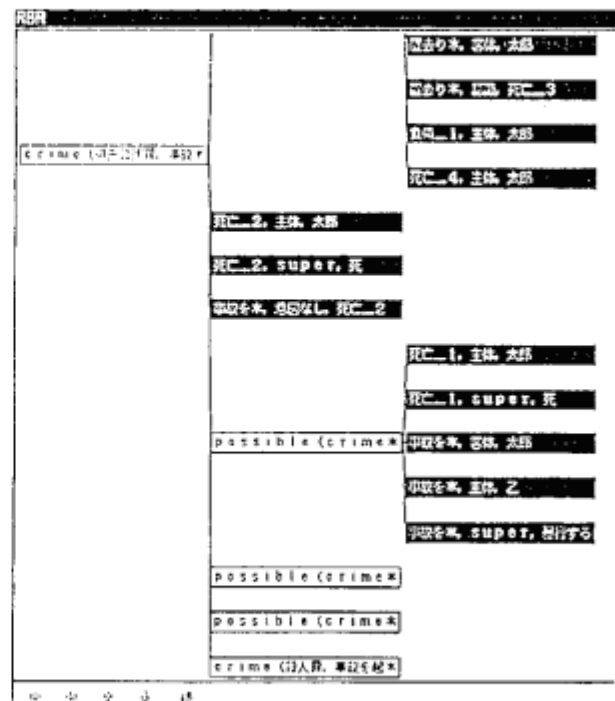


図2 証明木の例(乙の業務上過失致死罪の一部)

(b) 連言照合を複数論理式で同時に行なう OR 並列性: あるモデル候補の環境の下での各論理式の条件部の連言照合は独立で、OR 並列性がある。実際には5~10程度のルール毎に、1台のプロセッサを割り当てて、各プロセッサの負荷の均等化を行なう。

(c) 1つの節の条件部の連言照合における AND 並列性: 通常各条文の論理式表現の条件部は多数の条件節から構成される。よって、条件部の連言照合には AND 並列性が存在する。

今回の実験システムでは、(a),(b)の OR 並列性について、並列実行による高速化を行なった。最大64台のプロセッサを用いて、Multi-PSI上で並列実行させた際の条文に基づく推論単体で、1台のプロセッサでの推論速度に比べて、最高約3.5倍の速度向上が得られた。

8 まとめ

並列定理証明器 MGTP を用いた HELIC-II の法律条文に基づく推論の知識表現、推論方式、並列推論による高速化について述べた。

現在の実験システムでは、条文に基づく推論は、判例に基づく推論による解釈事実を一時的に受け入れるのみで、HELIC-II 全体としての推論も前向き一方である。現実の裁判においては、法律条文の構成要件を満足するために必要な解釈事実(注視点)を導き、それに従い、過去の類似判例を検索して自己に有利な事実解釈を導く。本システムも、両者の推論結果を教え合いながら、推論を進めて行く機構を作る必要がある。

参考文献

- [1] Fujita, H., Hasegawa, R., A Model Generation Theorem Prover Using Ramified-Stack Algorithm, ICOT TR-606, (1990).
- [2] Ueda, K., Chikayama, T., Design of the Kernel Language for the Parallel Inference Machine, The Computer Journal, (1990).
- [3] 大塚仁, 刑法入門, 有斐閣, (1981).
- [4] 大崎他, 法的推論システム HELIC-II(1) 概要, 情処 43 全大 (1991).