

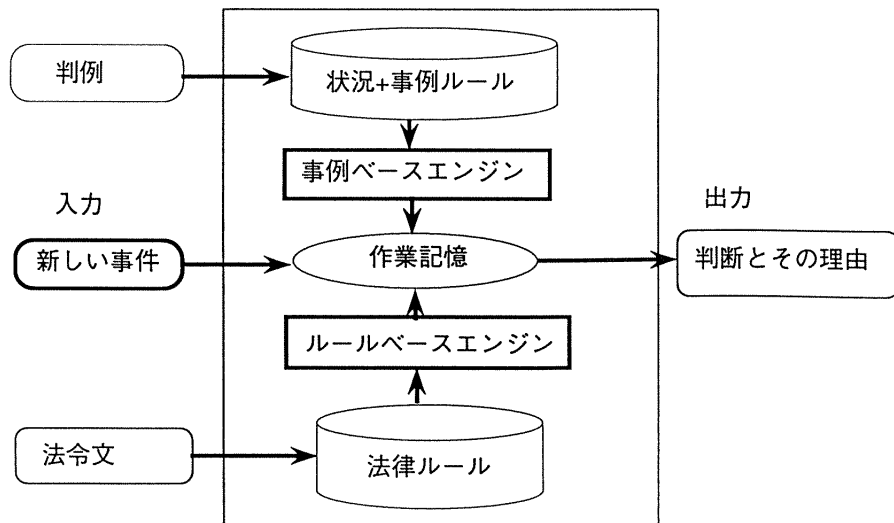
並列法的推論実験システム HELIC-II

概要

法律における推論はルールベース推論と事例ベース推論が結合したモデルで表すことができる。このモデルに基づいて、法的推論実験システム HELIC-II を並列推論マシン上に開発した。HELIC-II は法律条文と過去の事例（判例）を参照することによって、法的な結論を導き出すシステムである。

特徴

- 2つのエンジンからなる推論モデル
HELIC-II はルールベースエンジンと事例ベースエンジンという2つの推論機構を持つ。これらのエンジンは相補的に働いて法的結論を導く。
- 並列ルールベースエンジン
ルールベースエンジンは法律ルールを参照して、法的結論を演繹的に導く。このエンジンは並列定理証明器 MGTP (Model Generation Theorem Prover) をベースに、いくつかの機能を追加したものである。
- 並列事例ベースエンジン
事例ベースエンジンは過去の類似事例（判例）を参照することで法的概念を生成する。このエンジンは類似の事例を検索し、その事例のルールを新たな事例に適用することによって、新たな論理を構築するものである。



法的推論

法律の知識は法令文と過去の判例からなる。法令文は法律ルールの集合であるから、法令文に基づく推論はルールベース推論として実現している。しかし、法律ルールはしばしば定義の曖昧な法律用語（法的概念）を含んでいる。そうした法的概念の多くは、実際の事実に適用されて初めてその厳密な意味が決められるものである。このような法律ルールを実際の事実に適用するには、それらのルールを解釈することと、法的概念と具体的な事実との対応付けをすることが必要になる。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と事例ベース推論が結合したモデルで表すことができる。しかし、このモデルでは類似の事例を検索し、それに基づいて結論を導き出すのに多くの時間が必要であり、更にの複数の推論エンジンの推論を制御するための複雑な機構が必要となる。我々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。

HELIC-II の概要

HELIC-II は与えられた事件に関する法的結論を、法令文と過去の判例を参照することによって導き出し、推論木の形でそれらを出力するシステムである。HELIC-II はルールベース推論のエンジンと事例ベース推論のエンジンからなっている。ルールベースエンジンは法律ルールを参照し、法的結論を論理的に導くものである。以下は過失致死の法律ルールの例である。全ての法律ルールはこのような推論規則として表現される。

```
過失致死（“コメント”，[ 条文 = 210 ]，
  [ 自然人 ( A, [ ] ) , 自然人 ( B, [ ] ) , { { A \ = B } } ,
  行為 ( -行為, [ 主体 = A ] ) ,
  過失 ( -過失, [ 主体 = A, 行為 = -行為 ] ) ,
  因果 ( -因果, [ 原因 = -行為, 結果 = -死 2 ] ) ,
  死 ( -死 2, [ 主体 = B ] ) ]
  →
  [ [ 過失致死
    ( -, [ 主体 = A, 行為 = -行為 ] ) ] ] ) .
```

このルールは”過失”という法的概念を含んでいる。この”過失”が成り立つかどうかは、下記の甲女の事件のような個々の事例でケースバイケースで判断される問題である。

甲女の事件：

ある冬、甲女は生活に疲れて実子の太郎を道端に捨てた。太郎は生後4カ月であった。乙は道端で泣いている太郎を見つけ、車で警察に届ける

途中、事故を起こしてしまった。その事故で太郎は負傷したが、乙は太郎が死んだものと思い、再び道端に捨てて逃げ去った。その後、太郎は凍死した。

法律ルールからでは、この事例で甲女の行為に過失があったかどうかを判断することはできない。そこで、HELIC-II ではこのような過失の判断は事例ベース推論によって行う。事例ベースエンジンは、過去の類似事例を参照することによって、与えられた事実から（過失のような）法的概念を生成するものである（図1）。

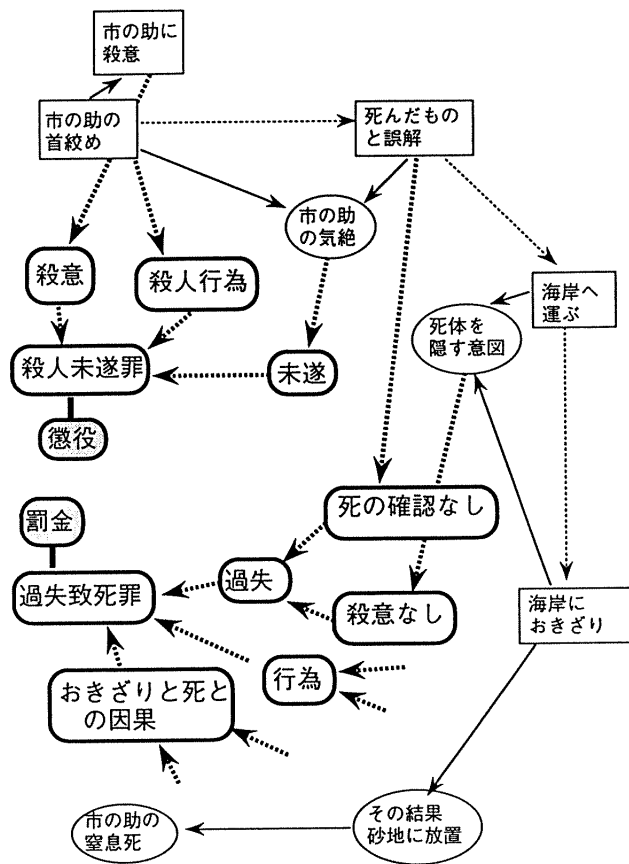


図1 浦田の事例における論理展開

図2は甲女の事件に対するHELIC-IIの結論の1つであり、浦田が市之助を殺害した事件の最高裁判例を引用して、結論を出している。

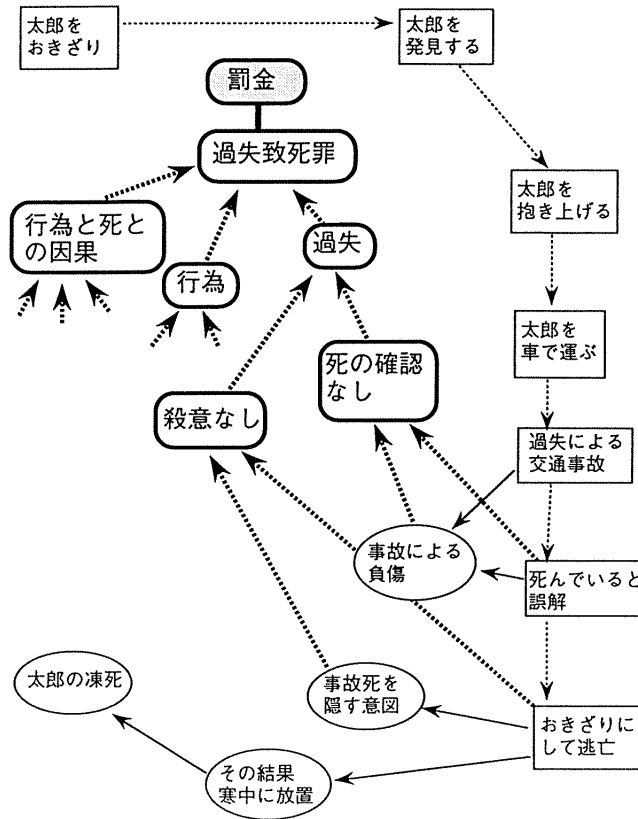


図2 甲女の事件における論理展開

HELIC-II の結論は自然言語の文章でユーザーに示される。以下の例は市之助事件で使われた被告の論理展開をまず示した後に出力される適用情報である。

市之助の状態【気絶する-1】と太郎の状態【負傷1】は共に【傷害】である。

浦田が市之助に行った行為【放置する-1】と乙が太郎に行った行為【置き去り2】は共に【放置する】である。

浦田は市之助の状態【気絶する-1】を確認しなかった。同様に乙は太郎の状態【負傷1】を確認しなかった。

浦田が市之助に行った行為【放置する-1】には市之助の死亡という目的がなかったと同様に乙が太郎に行った行為【置き去り2】には太郎の死亡という目的がなかった。

市之助も太郎も死亡した。

判例によれば、浦田が市之助に行った行為【放置する 1】は過失であると主張された。

したがって、乙が太郎に行った行為【置き去り 2】は過失であると認められる。

ルールベース推論

ルールベースエンジンは法的結論を法律ルールによる前向き推論によって導き出す。法的ルールは数が多いので、高速なルールベースエンジンが必要である。更に法的ルールには例外規則も含まれているので、ルールベースエンジンには、非単調推論を扱える機構が付加されていなければならない。HELIC-II のルールベースエンジンは並列定理証明器 MGTP(Model Generation Theorem Prover) をベースにしている。MGTP は非ホーン節の集合を与えると、全ての入力節を満たすモデルを並列推論によって生成する。MGTP を法律ルールのルールベースエンジンとして使い、さらにパイプライン効果によって高い処理性能を得るために、我々は MGTP にいくつかの機能拡張を施した。

1. 「証明の失敗による否定」の実現：

法律ルールには 2 種類の否定が含まれている。1 つは論理的否定（「～でない」）であり、もう 1 つは証明の失敗による否定（「～が証明できない」）である。元の MGTP は論理的否定だけを対象としていたので、我々は証明の失敗による否定も扱えるように拡張を行った。

2. 異なる前提条件に基づく推論の実現：

ルールベースエンジンは与えられた事実（新たな事件）と事例ベースエンジンの推論結果の両方を初期モデルとして使う。弁護側と検察側の主張はしばしば対立するので、事例ベースエンジンは論理的には矛盾するデータを生成することがある。そのため、論理的に矛盾を含まないようにモデルを分割して、ルールベース推論を行うという、データ管理の機能を開発した。

事例ベース推論

法的な事例（過去の判例）は弁護側と検察側の双方からの論証と、裁判官の判断、そして最終的な判決とからなる。我々は過去の判例を、その状況といくつかの事例ルールの組として記述している。状況は、事件の出来事に関する情報をイベントとオブジェクトの集合、そしてそれらの間の時間関係として記述する。双方からの論証は事例ルールとして記述される。事例ベースエンジンは過去の類似の判例を参照して、法的な概念を生成する。まず第 1 段階では類似の判例を事例ベースから検索する。第 2 段階では、選ばれた類似判例の事例ルール（双方の

論理) を新たな事件に適用することによって、新たな事件に対する論理が構築される。各事例ルールは類似性に基づく照合によって適用される。

並列推論の評価

図3はルールベースエンジンと、事例ベースエンジンの第2段階での速度向上を示している。我々はマルチP S Iの64 P Eを使って50倍以上の速度向上を得ている。

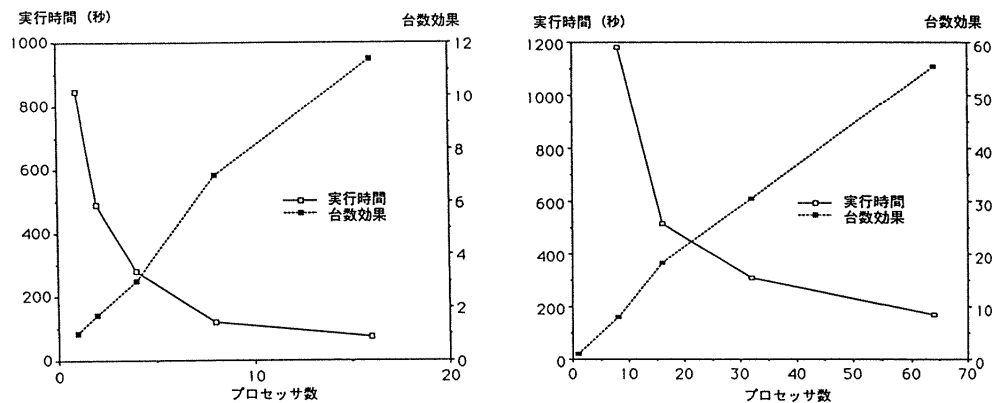


図3 ルールベースエンジンと事例ベースエンジンの速度向上

デモ概要

並列法的推論実験システム HELIC-II が、甲女の事件を解く様子をデモンストレーションする。まず甲女の事件の問題点を説明する。次に HELIC-II によって結論が導かれる様子を示す。更にその結論が導かれた推論過程を推論木により解説する。最後に自然言語形式の出力を使ってより詳細な説明を与える。