

事例を用いた法的推論とその並列化

新田 克己 星田 昌紀

新世代コンピュータ技術開発機構

法律の問題解決を行うときには、法律として与えられた知識をそのまま用いて三段論法を行うのではなく、解釈を伴う法的推論が行われる。解釈を行うときには他の法律や学説や過去の判決例などを参考にすることがあるので、法的推論はある種の事例ベース推論ととらえることが可能である。ここでは事例を用いた法的推論の計算モデルの構築を目的として、解釈の分析と、事例ルールによる事例の表現方法、論理の構築方法などを考察した。特に、事例の一般化や類推適用に伴う問題点について検討した。また、事例の検索と論理構築を並列処理することによって実行時間の短縮を図る方法を示し、論理型言語KL1による実験によってその効果を確認した。さらに法律システムの構築のための推論エンジンHELIC-Iの構成について説明した。

Legal Reasoning using Precedents and its Parallel Execution

Katsumi Nitta Masaki Hoshida

Institute for New Generation Computer Technology

Mita Kokusai Building 21F, 1-4-28 Mita, Minato-ku, Tokyo 108

Legal Reasoning is composed of finding of facts, interpretation of the law and application to the facts. To interpret the law, other law or precedents are often referred. Therefore, legal reasoning is a kind of case-based reasoning. In this paper, we discuss the model of legal reasoning such as representing precedents and constructing explanations. Especially, problems on generalizing precedents and making use of decisions of similar cases are discussed. Furthermore, the parallel way of executing legal reasoning in logic programming language KL1 is proposed and the inference engine for legal expert system is introduced.

1 はじめに

判決の予測、裁判における論理展開、法律にしたがった行動計画など、法律の専門家の活動を支援するシステムを開発するには、法律における推論を解析し、その計算モデルを構築する必要がある。

法律のシステムは、条文を論理式で表現し、法的結論をその論理式で証明する定理証明の問題として形式的に取り扱える場合が多い。しかし、現実の裁判などでは、法律で与えられる知識は具体的な問題を解決するには不完全であり、法の解釈がしばしば問題となる。解釈に依存した知識をどのように取り扱うかが、法律システムを構築する上での長い間の課題であった [1]。

解釈を行うためには、学説や他の法律や判例などが参照される。特に、判例は判決の予測を行う上での最も重要な情報源である。

ここでは判例を利用した推論のモデルとその問題点を検討する。また、法的推論の並列化の方法を提案し、法律システム構築のための推論エンジンの構成を述べる。

第2章では、法的推論を分析し、事例の役割について述べる。第3章では、事例の表現方法と事例の利用方法を説明し、事例の一般化に伴う問題点を検討する。第4章では、並列論理型言語 K L 1 による法的推論の並列化と推論エンジン H E L I C - I I の概要を説明する。

2 法的推論

2.1 法律と解釈

法律において意志決定をする場合、観測される事実に対し、法律を演繹的に適用すれば結論に至るように見える。しかし、現実にはそのような単純な図式では解決できない問題が存在する。

事実認定の段階においては、解決したい事件の記述が必ずしも完全ではないという問題がある。観測される事実には見落としがあるし、同じものを観察しても、主観的な判断が入る可能性がある。どの事実があれば、どの法的概念に該当するかという必要十分条件は決定できないことが多いので、他の事件での認定結果などが参考にされる場合がある。

認定された事実に対し、法の適用を行うときには、法律に現れる用語の多義性や、概念の抽象性や、論理体系としての不完全性などの問題がある [2] [3]。それにもかかわらず、法律が社会的規範として機能するのは、推論を行うときに [解釈] がなされるからである。

解釈とは、法の意味を具体化して認定された事実にあてはめる作業である。具体的には、用語に多義性があるときには、条文の目的に応じてその意味を個別に考えることに相当し、概念が抽象的なときには、概念を具体化する基準を生成することに相当し、適用できる法律間

矛盾があるときには、全体としての整合がとれるように調整規則を生成したり法律間の優先順位を生成したりすることに相当する。法律が時代遅れになったり、法律が不足したりするときには、既存の規則の修正や新たな規則の生成が行われる。

解釈は、具体的な事件の価値判断という側面があるから、事件が与えられる前に法律を解釈しきれるものではない。解釈は [法の創造] という要素がある法的推論における最も困難な技術であり、解釈が客観的に行えるかどうかは昔からの議論の対象であった [2]。

2.2 法的推論の種類

法律の専門家は、判決の予測、補償金の算出、財産の運用、論争における論理構築、契約文書作成、文献検索、などのいろいろな業務を行う。そのような業務遂行になされる法的推論を、吉野は正当化の推論と発見の推論に分類した [4]。正当化の推論とは、結論が前提条件から演繹的に導出されたものとして証明するもので、[意志決定] の機能と [論理構築] の機能がある。

[意志決定] とは、与えられた事件の記述に対して、法律や一般常識のルールを順次適用して、最終的な結論に達するものをいう。判決の予測や行動計画などの問題がこれに該当する。

[論理構築] とは、認定事実が特定の法律状態 (ゴール) にある事を主張するため、その理由を生成するものをいう。裁判で争いが生じると、原告も被告も、観測される事実がどうしてその概念に該当するか (または該当しないか) という論理 (説明) を展開する。これはゴールを達成するため、自分に有利な事実のみに焦点を当て、自分に有利な法律のみを適用しようとするものである。

判決文には原告 / 被告の論理とともに、裁判官の論理が記載される。そこに展開されている論理は、観測された事実と法律や判断ルールを適用し、最終結論に至っているように見える。しかし、実際には、裁判官は観測事実からある程度判決を仮定し、それを正当化するための説得の論理を構築していることがある [5]。

判決予測などの意志決定の場合には、観測される事実があらかじめ用意されたどの典型的パターンに該当するかという分類問題として捉えられるのに対し、論理構築の場合には、分類問題としての要素以外に、観測される事実と典型的なパターンとの相違を発見してそのパターンに分類されないことを主張するような発見的な要素がある [7]。

意思決定や論理構築のいずれの場合も問題となるのは解釈の問題である。新たな解釈を行うために発見の推論がなされる。解釈はできるだけ客観的な説明がつくよう

になされるが、その材料は法律内に限られないので、発想の着眼点を限定することは一般的には困難である。

2.3 事例を利用した推論

法律における推論の目標は、観測事実と結論をつなぐ論理（説明）を構築することである。

前述したように、論理を構築するには解釈という段階が必要なことがある。裁判では解釈は個々の裁判官の裁量にまかされているが、裁判官は解釈をまったく自由に行うわけではなく、学説や法目的や産業政策や社会常識や他の法律や過去の判例などを考慮し、なるべく客観的な説明がつくように行う。

解釈ではバックに法律体系に関する理解と社会常識などの膨大な知識を必要とし、異なる価値判断の間の調節をしなくてはならないが、その方法は確立したものではない。例えば、民法の解釈には、文理解釈、論理解釈、目的論的解釈、論理解釈、立法者意志解釈などの方法があるとしており [8]、どれを採用するかによって異なる結論が導かれる事が少なくない。これらの解釈の調整を行う原理は明確ではないし、それぞれの解釈を行うには、法律体系や個々の条文の趣旨や学説などを知識として、複雑な推論をしなければならない。

しかし、過去に類似の事件があったり、他の法律で類似の条文があったりするときには、そこで用いられた論理を拡張や類推によって利用して、解釈や判断の問題に対処することができる。現に、判決文には (1) 過去に生成された命題や判断基準をそのまま利用したもの、(2) 過去の命題や判断基準の趣旨を利用したもの、(3) 説得するための根拠の補強として、過去の命題や判断基準を引用したものなどがあり [5]、専門家も事例を利用した推論を行っている。ただし、判決の中の命題や判断基準の中で、どれが他の事件に応用できるかについては専門的な議論がある [5] [6]。

3 事例を用いた法的推論モデル

3.1 事例の表現

事例の中で最も重要なものは判決文である。判決文は過去の事件において、観測事実から結論に至るまでの推論過程を表すものである。その主な内容は、どのようなルールがどのような事実のもとで適用されたかという説明である。

これを図1のような図式で考えることができる [10] [11]。この図で R1、R2 は法律から抽出されたプロダクションルールを表し、r1、r2、r3 は判断結果から抽出された事例ルールを表し、FA、FB、FC は観測された事実を意味ネットで表現したものを表す。

この例では、FA は A に、FB は B に該当したなど

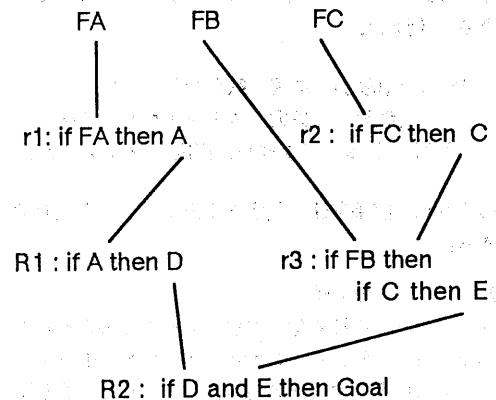


図1: 判決文の説明

とそれぞれ判断されて、最終的に Goal という結論に至ったことを表す。ここで、r3 は状況 FB のもとでルール (IF C THEN E) が生成されたことを表す。これは、互いに類似する2つの概念 C と E を、ある状況では同一視したいときなどに用いることができる。

このように、ルールには条文から抽出されるルールの他に、判例として与えられるルール、一般常識を表すルール、新たに裁判官によって創造されたルールなどを含む。一般的なルールと判断のルール（事例ルール）との違いは、前者は条件部がすべて満足されないと起動されないのに対し、後者は条件部がかなり満足されれば起動される可能性があること、前者の条件部は固定されているのに対して、後者の条件部の概念が拡張（一般化）されることである。また、前者の推論結果は常に正しいが、後者の推論結果は正しいとは限らない [10]。

意味ネットは、

{ 対象, 関係, 値 }

の3つ組の集合で表現する。事例ルールの中では

IF { {対象1, 関係1, 値1}, 重要度1 } ∧
{ {対象2, 関係2, 値2}, 重要度2 } ∧

{ {対象n, 関係n, 値n}, 重要度n }

THEN

述語 / リンク / ノード / ルールの生成

のように、意味ネットの表現がそのリンクの重要度とともに記述される。

事例ルールの実行部では、以下のように新しいリンク、ノード、述語、ルールを生成することができる。

● 新しいリンクの生成

推論の結果、ノード間に新しい関係が生じることがある。例えば、

```
IF { {事故 #1, 原因, 飲酒 #1}, 20 } ∧
    { {飲酒 #1, 原因, ストレス #1}, 20 }
THEN [make {事故 #1, 原因, ストレス #1} ]
```

のように、因果関係に関する新たなリンクを生成できる。

● 新しいノードの生成

ネットの一部を抽象化して別のノードと考えることができる。例えば、喧嘩でのいくつかの出来事のノードを統合して、[傷害行為]という抽象的な概念のノードを生成することができる。

● 新しい述語の生成

ノードの生成と実質的に同じであるが、その事件で成立する事項を述語として記録することもできる。

● 新しいルール生成

過去のルール生成の説明を利用して、類似の説明によってルールを生成することができる。すなわち、事例の中に

```
IF ~
THEN
[make (IF~THEN~)]
```

なるルールがあれば、類似の状況で類似のルールが生成できる。このようなルールは、過去のルールの適用条件の限定を表すのに使うことができ、ルール生成のためのいわゆる法理(いくつかの解釈を統一する原則)を表現するものとしても使うことができる。

事例ルールによって生成されたデータにはその根拠(使用した事実とルールの組)が付加される。事例ルールによる結論はあくまでも仮説に過ぎないから、後でその根拠を示す必要があるからである。

3.2 事例の適合

新しい事件が与えられたときの、過去の事例の利用方法について述べる。新しい事件で成り立つ事実は、意味ネットとその事件を特徴付ける述語の集合で表されるとする。

このような入力データに対し、法律のルール(そのほとんどがプロダクションルールとして記述されている)を直接適用しようとしても、通常はルールの条件部が満足されない。前に述べたように事実と法的概念の抽象

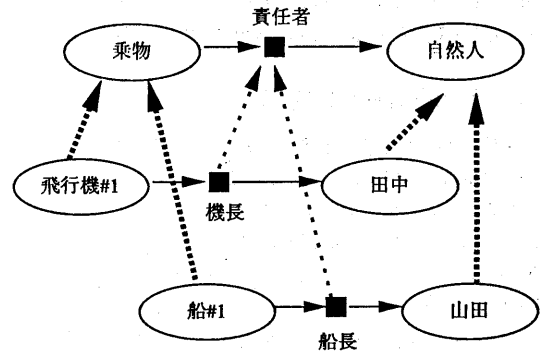


図 2: 階層構造

度が違うからである。そこでまず事例ルールを起動すると、観測事実の中からいくつかに着目されて抽象的な概念が生成され、その結果、法律のルールが起動されることになる。

事例ルールの場合、条件がすべて満足されなくともある程度の類似性があれば起動される [10]。例えば、

```
IF { {船 #1, 船長, 山田}, 20 } ∧
    { {骨折 #1, 主体, 山田}, 5 }
THEN ~
```

というルールがあったとき、

```
{ 飛行機 #1, 機長, 田中 }
```

というデータがあったとする。2つの3つ組の対象、関係、値のそれぞれが、与えられた階層構造のもとの同一の上位概念を持つ場合に、その3つ組は対応がとれるものとする。この例では、[船 #1]と[飛行機 #1]はともに[乗物]という概念の低位概念(またはインスタンス)であるという知識と[船長]と[機長]はともに[責任者]という概念の低位概念であるという知識と[山田]と[田中]はともに[自然人]の低位概念(またはインスタンス)であるという知識があれば、{船 #1, 船長, 山田}は{飛行機 #1, 機長, 田中}と対応がとれる(図2)。対応をとるにあたって、上位概念をたどる段数が少ないほど、両者の類似度は近いと判断される。また、上の例では

```
{ 骨折 #1, 主体, 山田 }
```

に対応する事実がないかもしれない。しかし、この条件の重要度が低いので、事例ルールの条件部は類似度が高いと判断され、このルールが起動される可能性は高い。

判決予測を行うときには、入力された事件の意味ネットが、事例ルールの適用によって次第に詳細化と抽象化

が行われ、あらかじめ用意された特定の事件のパターンに合致するとき最終結論に至る。

特定のゴールに向けての論理構築を行うときには、そのゴールへ達するためのサブゴールが生成され、入力された意味ネットを抽象化してそのサブゴールに結び付ける。

適切なルールがないときには、新たなルールの生成や着眼点の転換などで、事態を打開する必要がある。このような場合に、過去の事例で、どのような状況の下でどのようなルールの生成がなされたかを知ることは、新たなルール生成の発想の参考になる [9]。

3.3 判決の例

判決の例として、労働基準法 79 条

[労働者が業務上死亡した場合においては、使用者は、遺族に対して、平均賃金の 1000 日分の遺族補償を行わなければならない]

に関する判決を考えてみる。[業務上の死亡か否か]の判断基準は、業務遂行中の死亡であること、業務と死亡との間に因果関係があること、の 2 つであることが判例で確立している。

ここで因果関係の判定は容易ではない。他の原因が重なっていることもあるし、業務が間接的な原因になっているかもしれないからである。従って、多くの事例が類型化され、それぞれに判断基準がルールとして決められている [12]。

以下の事例は、農協職員山田氏が勤務時間後に、料理屋での会議に業務命令で参加し、その後、帰宅の途中で組合員鈴木氏の家に保険加入の勧誘の目的で行こうとして事故にあった事件である [長野地判 39.10.6 の改変]。

裁判では、業務上の死亡かどうか争われた。その論点の 1 つとして、被告(労働基準監督署)は、(1)山田氏は庶務係なので保険勧誘は本来の業務(職務)ではない、(2)鈴木氏の家へ勧誘に行けとの具体的命令は出していない、の理由で業務でないとしたのに対し、原告(山田氏の遺族)は、(1)保険勧誘の成績は評価の対象となっていたので実質的業務である、(2)保険勧誘を奨励する局長の訓示やキャンペーンがあったので包括的業務命令があったとみなせる、と主張した。

まず、この事件の観測された事実の意味ネット表現の一部を図 3 に示す。

被告が、保険勧誘は山田氏の直接の業務(職務)でないことを理由に、業務でないとしたことは、以下の事例ルールで表現できる(ルールの要部のみ示す)。

```
IF { {雇用関係 #1, 従業者, 山田氏}, 20 } ∧  
    { {雇用関係 #1, 雇用者, 郵便局}, 20 } ∧  
    {not {雇用関係 #1, 職務, 保険勧誘}, 100}  
THEN [make(職務外)]
```

それに対し、原告が、保険の勧誘が勤務評価の対象であったので、実質的に業務であったとしたことは以下のように表すことができる。

```
IF { {雇用関係 #1, 従業者, 山田氏}, 100 } ∧  
    { {雇用関係 #1, 雇用主, 郵便局 #1}, 100 } ∧  
    { {郵便局 #1, 業務, 保険勧誘}, 100 } ∧  
    { {郵便局 #1, 責任者, 局長 #1}, 100 } ∧  
    { {局長 #1, 職務, 勤務評価}, 100 } ∧  
    { {勤務評価 #1, 対象, 山田氏}, 100 } ∧  
    { {勤務評価 #1, 内容, 保険勧誘}, 100}  
THEN [make(職務範囲)]
```

ここで新しい事件について考える。これは、建築会社の材木運搬トラックの運転手である田中氏が材木の積み降ろし中に事故にあったものである。田中氏の材木の積み降ろしは雇用契約に含まれていないとすると、労災認定においてどのような判断がなされるかを予測する。

この事件を意味ネットで記述したものの一部を図 4 に示す。

このネットを図 3 と比較し、[雇用関係 #1] と [雇用関係 #2]、[山田氏] と [田中氏]、[郵便局] と [建築会社]、... のようにマッピングすれば対応がとれる。すなわち、山田事件によってなされた双方の主張(職務でない vs 成績評価の対象)の事例ルールを田中事件に適用すれば、田中事件でも同じ議論が成立することが推定される。

また、いずれの事件もさらに抽象化することによって、事業主の管理下のケース、事業主の管理下ではないが支配下のケース、事業主の管理下でも支配下でもないケースなどに分類され、それぞれのケースにおいて異なる判断基準が適用される [12]。

3.4 一般化の問題点

事例ルールの条件部と入力された意味ネットデータの照合において、事例と入力データは 2 種類の方法で抽象化されている。1 つはいくつかの概念を統合して新しい抽象的な概念を生成する方法であり、もう 1 つは階層構造の上位概念で一致するときには、両者は類似すると判断する方法である。前者の例としては、3.3 の山田事件で、局長の訓示やキャンペーンなどの事実を統合して、包括的業務命令と判断したことがあげられる。後者の例としては、山田氏と田中氏がともに自然人であるという知識を利用していることがあげられる。

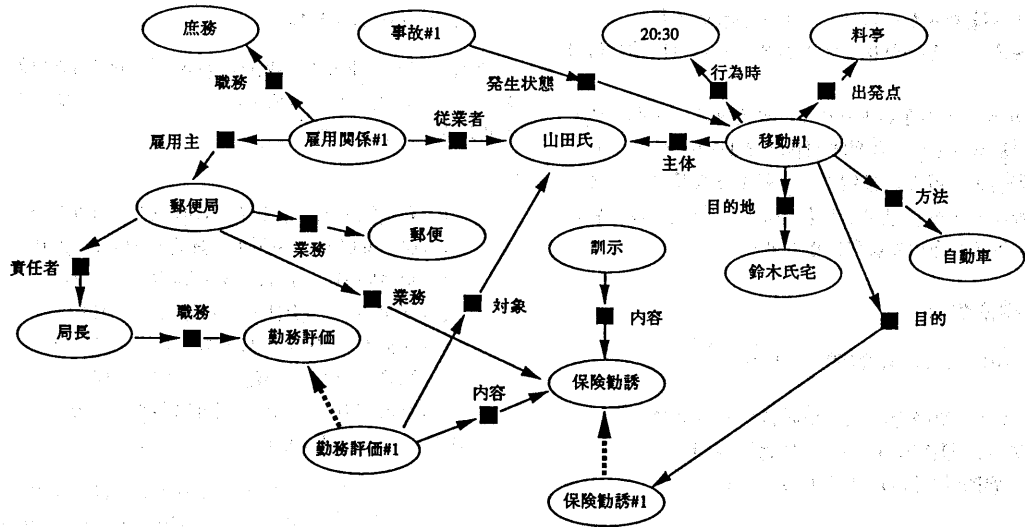


図 3: 山田事件の意味ネット表現

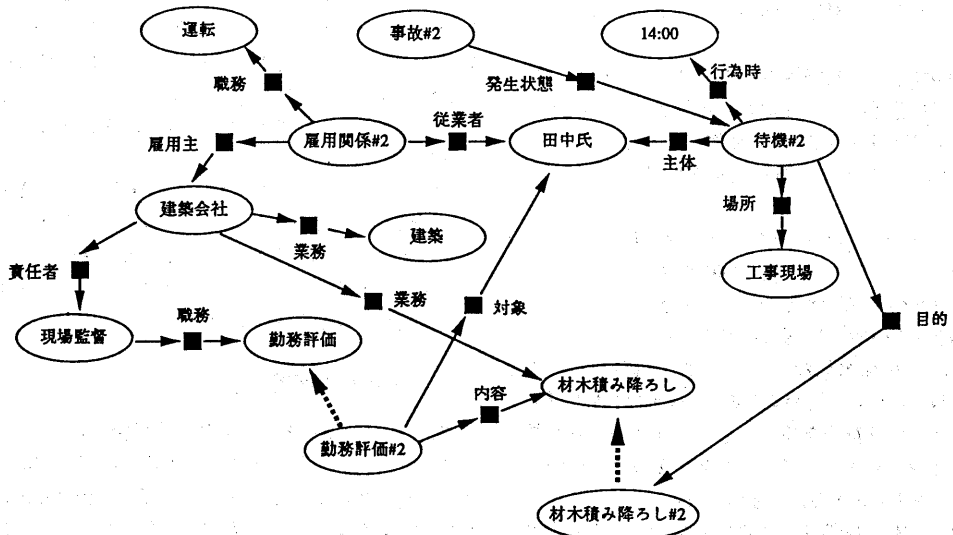


図 4: 田中事件の意味ネット表現

これらの抽象化は、事例で生成された判断基準や解釈を拡張して類似の事件に適用していることに相当する。しかし、過去の判例をどの程度拡張できるか（すなわち命題の適用範囲）については、法律的な検討が必要である [5]。

例えば、[相手と共謀して嘘の意思表示をしたときは無効であるが、そのことを知らなかった第三者には対抗できない] ことを内容とする民法 94 条に関して、共謀の事実がなくてもこの条文の適用を認めたいいくつかの判例がある。そのうちのいくつかは [誤った登記の黙認] という事実を [共謀] という概念に含めることによって結論を得ているので、[誤った登記の黙認] を [共謀] と判断する事例ルールをおくことによって、判決予測や論理構築をすることができる。ただし、これは無制限に [誤った登記の黙認] を [共謀] と判断して良いことを意味しているのではなく、この条文の適用に関してのみの判断である。しかし、もっと抽象的な観点からこれが [外観を信頼して取り引きをした善意の第三者の保護] という目的のための論理であることに着目すると、同一の論理が外観に関する他の事件（例えば商法 265 条関連の事件）にも使える可能性がある。

2 つの判決を比較すると、一見して矛盾しているように見えるものがある。例えば、[不動産関係の所有権等の変動は登記をしなければ第三者に対抗できない] ことを内容とする民法 177 条に関する 2 つの判例を考察する。昭和 28 年の最高裁判決では、[自作農創設特別措置法によって農地を買収する時には民法 177 条は適用されない] という命題を生成しているのに対し、昭和 38 年の最高裁判決では、[国税滞納処分による差し押えでは民法 177 条は適用される] という命題を生成している [5]。[農地買収処分] と [差し押え] はいずれも [公法上の処分] という概念の下位概念である。従って、前者の判決があるだけであれば、[公法上の処分では民法 177 条は適用される] という一般化の可能性があるが、両方の判決が存在する場合には、そこまでの一般化はできないことになる。

また、職業選択の自由を定めた憲法 22 条に関して、[公衆浴場の設置場所を制限した条例は、その偏在と濫立を防止することが環境衛生に必要なから合憲である] という判決と、[薬局の設置場所を制限した条例は、その偏在と濫立を防止することが国民の保健上の危険防止に直接つながらないから違憲である] という判決が出ている [5]。いずれも [保健衛生に関する公共性のある職業] であるが、そのような概念にまで判断を一般化することはできない。

類似の事件で判決の相違はどこから出たのかを観察すると、法目的やその事件に特有な事実などが反映している。異なる判決の事実を比較して、それをいくつかのカ

テゴリに分割し、そのカテゴリを越えない範囲でしか一般化は行うことはできない。現実にはそのようなカテゴリを設定するのは困難なので、事例ルールには命題そのものだけでなく、その背景となる事実をも条件部に記述して、適用できる範囲を制限することが必要である。この場合、より類似する事例のルールの結果がより信頼できることになる。

このような事例ルールの詳細化の他に、ここでは一般化の制限を、事例ルールの中で明示することでも対処している。すなわち、事例ルールの中で特に指定がなければ、無制限に階層構造の上位概念を照合するが、例えば以下のように船 #1 に制限 [船] が指定されているときには、それより上位概念は照合しない。

```
IF { {[船 #1, 船], 船長, 山田}, 20} ∧
    { {骨折 #1, 主体, 山田}, 5}
THEN ~
```

3.5 事例の索引

図 3 は事例の 1 つの記述を示したものであるが、法律家が事例を検索し、議論をするときにはもっと細かい事実に着目したり、逆にもっと抽象的な事件のパターンで十分なことがある。したがって、事件の記述は具体的なレベルから抽象的なレベルまで存在しうる。抽象的な記述を構成する方法は 1 通りではない。いろいろな見解によって抽象化の方法が異なる可能性があるし、前に見たように、反対の法律概念にまとめられる可能性もある。一見して、類似する事例でもわずかな事実の相違から結論がまったく異なることもある。

このように考えると、事例はその条件部の抽象度によって階層的に整理すること、原告に有利な判断と被告に有利な判断のように立場の相違によって整理すること、類似の事例をまとめておくことなどが必要がある。

索引付けのための特徴抽出には、3.1 で説明した述語生成の機能が利用できる。また、階層的な整理には、新しい概念を既存の階層構造内に挿入する研究の成果が利用できる可能性がある [13]。

3.6 推論結果

同じ観測事実を与えても、どの事例ルールを適用するかによって異なる結果が得られる。事例ルールの適用による新たなデータの生成は、1 つの仮説の生成とも考えられる。異なる事例ルールの適用によって異なる仮説が生成されるが、作業記憶の中でそれらは同一に取り扱われるのではなく、異なった世界で成り立つ事実として管理されなくてはならない。仮説の組み合わせによっては矛盾を生じるから、そのような仮説の組み合わせを防止するある種の真理維持機能が必要である。

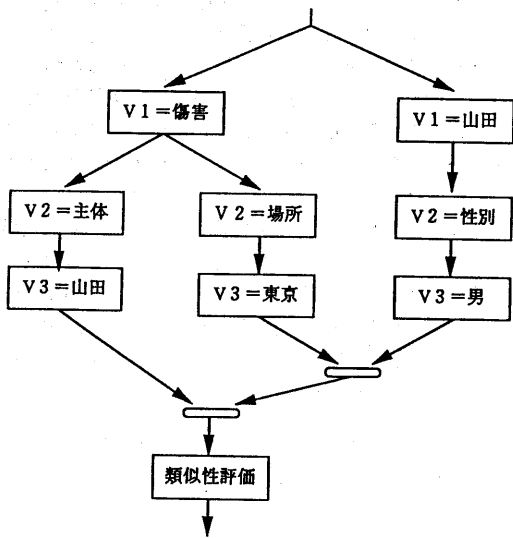


図 5: 判断ルールのコンパイル

4 事例検索と適合の並列化

4.1 データフローグラフ

一般に事例ベース推論は、膨大な事例を検索し、照合しなくてはならないから、実行時間が非常にかかることが予想される。そこで、ここでは並列処理による推論時間の短縮の可能性について検討する。

前に述べたように、法律における説明は一般のプロダクションルールと事例ルールで表される。プロダクションルールの条件部は RETE ネットなどのデータフローグラフに展開することで高速化が図れることが知られている。事例ルールの場合も RETE と類似のデータフローグラフに展開することができる。

例えば、

```
IF { {傷害, 主体, 山田}, 10 } ∧
    { {傷害, 場所, 東京}, 20 } ∧
    { {山田, 性別, 男}, 30 }
THEN ~
```

というルールは、図 5 のようなグラフに展開される。

図で、V1、V2、V3 とは、意味ネットの 3 つ組データの第 1 引数、第 2 引数、第 3 引数を表す。このデータフローグラフのルートから、新しい事件のデータをトークンとして流すと、条件を満足するトークンの組み合わせが出力される。各ノードにはメモリがあって、以前に通過したトークンを記憶している。従って、2 回目以降のサイクルでは、新たに生成されたデータのみを流せば良いことになる。類似性評価のノードでは、あら

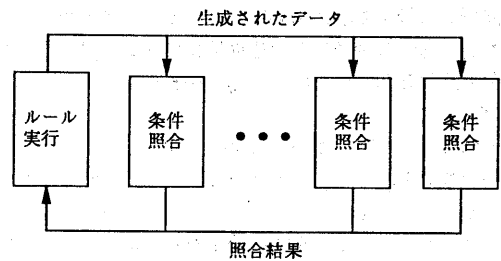


図 6: プロセッサの割り付け

かじめ設定された類似度に達しないトークンの組み合わせは通過できない。

このデータフローグラフが通常の RETE ネットと異なるのは以下の点である。

- 3 つ組の各引き数が同じ上位概念ノードを持つかどうかのチェックが必要である。
- RETE ネットの場合には全ての条件が満足されたトークンの組合せが出力されるのに対し、このグラフでは一部の条件が満たされないトークンの組み合わせでも、ある基準以上の類似度があれば出力される。

このデータフローグラフを並列論理型言語 KL1 で実現する。これは、事例ルールの条件部をコンパイルして、KL1 プロセスをノードに対応させ、ストリームをリンクに対応させたデータフローグラフを作成することによる。ルートノードのプロセスから、新しい事件の事実データをトークンとして流すと、条件照合の並列処理が実現できる。

このような方法で、推論エンジンを作成し、Multi-PSI による並列実験を行った。実験に用いたルールは、労働災害の 8 つの判例から抽出した事例ルールを複製した試験的なもの (約 300) である。現実の事実ルールに比べて、条件部がかなり簡単なものを用いている。実験では Multi-PSI の 1 台の要素プロセッサ (PE) が照合結果の収集とルールの実行を行い、他の PE が判断ルールのデータフローグラフを格納して条件照合を行うようにした (図 6)。

異なった事件に対する実行時間の測定結果を表 1 に示す (単位は秒)。この表で、台数とは条件照合で用いた PE の数を表す。条件照合以外にルール実行を行う PE が必要だから、実際にはこの数より 1 だけ多い PE が使われたことになる。台数が 0 のときは、条件照合とルール実行が同じ PE で行われているが、条件照合のフェーズとルール実行のフェーズを分けているので、台数が 1 のときと実行時間がほとんど変わらない。

台数	0	1	2	4	8	16	24
(a)	13.1	13.4	7.1	3.9	2.4	2.0	2.2
(b)	17.7	18.0	9.6	5.2	3.1	2.5	2.7

表1 実行時間

表2は、1台のPEを用いたときに比べて何倍の速度が得られたかを示したものである。表に示すように、台数が16以下のときには速度の向上が観測できるが、それ以上のときは速度が低下する。これはPE間のデータ転送のコストが影響したものである。

台数	0	1	2	4	8	16	24
(a)	*	1.0	1.9	3.4	5.6	6.7	6.2
(b)	*	1.0	1.9	3.4	5.8	7.2	6.6

表2 台数効果

4.2 法的推論エンジン

これまで述べてきたように、法的推論を行うためには事例を利用して、意思決定や論理構築を支援する機能が必要である。場合によっては、ルールを生成することもある。生成された論理が苦しまぎれのへ理屈であっても、実用に当たっては無いよりマシであるし、相手の反撃を未然に防止するのに役に立つこともある。また、考えの漏れを防ぐのにも使えることがある。

現在設計中の、へ理屈生成を行うための法的推論エンジンHELIC-11 (Hypothetical Explanation of Legal Inference Constructor - Intelligent Instructor) はこのような目的のためのエンジンである。HELIC-11は、解決しようとする事件の状況(と目標とする法的な状態)を入力すると、状況から目標へ至る論理(説明)を生成する(図7)。

このシステムは、プロダクションルールの推論を行うエンジンと事例ルールの推論を行うエンジンとからなる[14]。プロダクションルールのエンジンからはルールベースがアクセスでき、事例ルールのエンジンからは事例ベースがアクセスできる。解決しようとする事件を入力すると、これは意味ネットと述語の集合として作業記憶に格納される。双方のエンジンは独立に作業記憶にアクセスし、推論結果をその根拠と共に作業記憶に格納する。最終的に適用できるルールや事例がなくなった段階で、目標の状態に達したものをすべて選び、用いたルールと事例を出力する。

5 課題と問題点

法的推論における判決文の役割と、事例検索の並列化の可能性について述べた。技術的課題を以下に示す。

1. 事例の記述が何通りもできるのは問題である。異なった記述をしても抽象化で同じパターンにまとめられる保証はない。事件の記述のための制限と主要概念の辞書が必要である。
2. 事例ルールの数が多いときに組み合わせが膨大になる恐れがある。何らかの推論制御が必要である。
3. 立証責任の問題が重みに反映されなくてはならない。原告と被告は対等ではなく、条文によってどちらかがその法律概念に該当すること/該当しないことを立証する責任がある。当然、立証する方の負担が大きく、反証する方は相手の説明の欠陥だけをつければ良い。

最後に、学習との関係について述べる。判例とは1つの事件の解決の推論過程の説明を記述したものである。この推論過程の説明を一般化してルールを生成し、それを他の事件に適用することはいわゆる「説明に基づく学習(Explanation Based Learning)」として定式化できる可能性がある。特に解釈生成のルールが事例から生成できれば、実用的な価値はある。しかし、ルールを生成するための領域知識は整理されておらず、いきなり一般化された形で判決文に現れることもあるから、解釈生成ルールを学習によって作り出すことは容易ではないと思われる。また、3.4で述べたように、どこまで事例を一般化できるかという問題を解決するには個別に法律的な知識を必要とする。

謝辞 本研究の機会を与えていただいた内田俊一ICOT第4研究室室長に感謝致します。また、KL1プログラミングについて教えていただいたKL1-TGとPIMOS関係の方々、法的推論の基礎を教えていただいた法律エキスパートシステム研究会(明治学院大学 吉野一教授代表)の方々、勉強のための資料を教えていただいた夏井高人判事補、法律エキスパートシステムの動向などの大局的な指針を与えていただく池田純一弁護士に感謝致します。研究についての実質的な助言と批判を下された小松弘弁護士に感謝致します。

参考文献

- [1] A.L.Gardner: "Law Applications", Encyclopedia of Artificial Intelligence"
- [2] 五十嵐: 法学入門, 一粒社
- [3] 山本: リーガルマインドへの挑戦, 有斐閣選書
- [4] 吉野: 法律エキスパートシステムに関する調査研究報告書, 機械システム振興協会(1987)

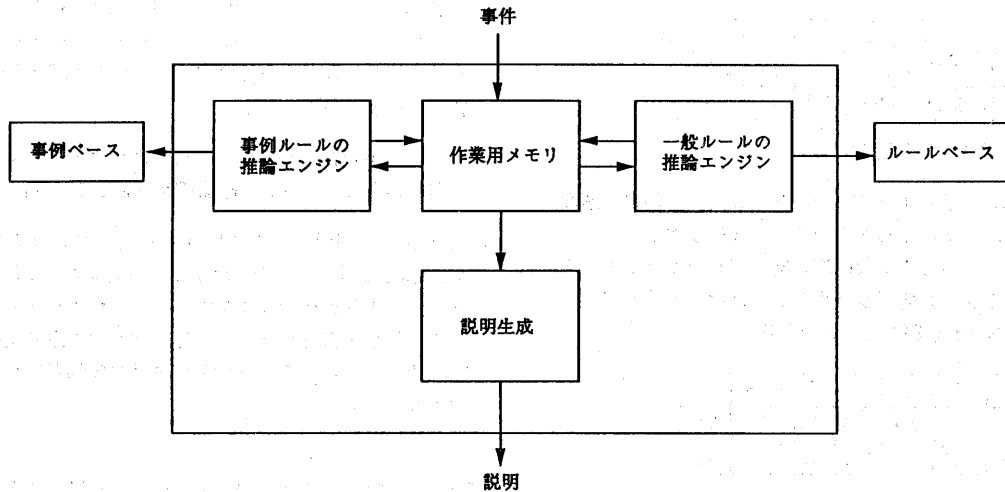


図 7: HELIC-II の構成

- [5] 中野：判例とその読み方，有斐閣
- [6] 法学セミナー Vol.34, No.3 (1989)
- [7] K.D.Ashley, et.al. ; "Compare and Contrast, A Test of Expertise", Workshop on Case-Based Reasoning (1988)
- [8] 遠藤：民法 (I)，有斐閣双書
- [9] 諏訪, 他："フラストレーションに基づく学習", ICOT KSA-WG 資料 (1988)
- [10] L.K.Branting : "Representing and Reusing Explanations of Legal Precedents", International Conference on Artificial Intelligence and Law (1989).
- [11] L.K.Branting : "The Role of Explanation in Reasoning from Legal precedents", Workshop on Case-Based Reasoning (1988).
- [12] 労災保険 業務災害および通勤災害認定の理論と実際
- [13] J.Kim, et.al : " Parallel Classification for Knowledge Representation on SNAP", Technical Report CENG 89-34, University of Southern California
- [14] E.L.Rissland, et.al : "Interpreting Statutory Predicates", International Conference on Artificial Intelligence and Law (1989).