データベースシステム 94－4
（1993．7．21）

データベースに判決は予測できるか？

横田 一正　　柴崎 真人
(財) 新世代コンピュータ技術開発機構 (ICOT)

法的推論システムは、人工知能、自然言語処理、データベースなど計算機のさまざまな分野の技術を統合した大規模知識情報処理システムのひとつである。データベースの観点から考えれば、データ / 知識の大規模性とその分類、部分情報の扱い、高次推論を含む問合せ機能、など次世代データベースの機能として検討すべき多くの課題を抱えている。と同時に、データベースと応用の境界線についても大きな示唆を与えてくれる。本稿では、データベースの観点から、法的推論に必要とされる特徴と機能を検討し、演繹オブジェクト指向データベース言語 *Quixote* 上に作った実験システムの経験を踏まえて、次世代データベースの機能を議論する。

Can Databases Predict Legal Judgments?

Kazumasa Yokota　　Makoto Shibasaki
Institute for New Generation Computer Technology (ICOT)

A legal reasoning system is one of large-scaled knowledge information processing systems, where many technologies such as artificial intelligence, natural language processing, and databases are integrated. From a database point of view, the application has many kinds of data and knowledge, and provides many research topics to be considered for new generation databases: features of very large database and knowledge-bases, their classification, treatment of partial information, query processing containing high-level reasonings, and so on. Further, it suggests an idea about boundaries between databases and applications. In this paper, we investigate features and functions necessary for legal reasoning from a database point of view, and discuss features of next generation databases, based on our experimental legal reasoning system on a deductive object-oriented database system *Quixote*.

1 はじめに

法律の分野で計算機を利用しようという試みは、数学の問題を計算機に解かせようという定理証明の試みと同じように、人工知能が提唱される以前の、計算機のもっとも初期の時代からあった (たとえば [5])。法律は単に弁護士や裁判官など法曹界の人びとのためのものではなく、社会生活と一体のものであり、あらゆる社会的行為に深く関係している。したがってこれらに対応したシステムの必要性が強く要求されており、これまでにさまざまな種類の法的エキスパートシステムが数多く構築されてきている。たとえば

- 判決予測
- 節税計画
- 損害賠償の交渉
- 契約文書作成
- 立法支援

などを目的としたものがある。

この中で、判決予測 (あるいは法的推論) は、データベース、知識ベース、高次推論、知識表現、自然言語処理など多くの分野の技術を統合した大規模知識情報システムであり、数多くの新しい問題を提起している。ICOT でも、第5世代コンピュータプロジェクトの中で、判決予測のために HELIC-II [8] と TRIAL [11, 9, 13] という2種類のシステムを試作した。データベースの観点からこの経験を踏まえると、法律のデータ / 知識は大規模データベースあるいは大規模知識ベースそのものであるし、それらを表現するためにはさまざまな革新的技術が必要で、さらに問合せ処理は演繹データベースの能力をはるかに上回ったものが要求されている。つまりこの応用は次世代データベースのための多くの問題を提起していると考えられる。本稿では、データベースの観点から、法的エキスパートシステム、その中でも判決予測システムを中心に分析し、次世代データベースが備えるべき能力と特徴について議論を行う。これらの多くは法的推論という応用に留まらず、科学データベースを始めとする新しい応用にも適用できるだろう。

本稿ではまず 2 節で法的推論のための知識と推論の整理を行う。次に 3 節では ICOT で演繹オブジェクト指向データベース (DOOD) システム *Quixote* [13] 上に試作した判決予測システム TRIAL の概要とその反省点を述べる。最後に 4節で、上記を踏まえた、法的推論システムのためのデータベース管理システム、および法的推論システムの経験から得られた次世代データベースシステムのための課題を議論する。

2 法律の知識と推論

2.1 法律の知識 — 法源

裁判官が判決を行う際に拠り所とする規範を法源という。これを [4] を中心にして考えてみよう。法源は国によって大きく異なるので、以下では日本の場合に限定する。法源は以下の2つに分類することができる[1]。

- 制定法 (あるいは成文法)
- (広義の) 不文法

制定法とは立法機関により制定された法規範であり、明確に条文として記述されたもので、不文法とは明確には法規範として成文化されていないものである。

制定法には (分類法にはいくつかあるが)

- 憲法
- 法令 — 国会で制定
- 命令 — 国の行政機関が制定
- 条例 — 地方公共団体が制定
- 規則 — 地方公共団体の執行機関が制定

がある。法令には、刑法、民法、商法、刑事訴訟法、民事訴訟法の5つの基本法があり、憲法と合わせて基本6法と呼ばれる。これら制定法は1つの論理体系のように無矛盾なものではない。そこで制定法間の矛盾を避けるために、制定法間には

- (上の分類の) 上のものが下のものに優先
- (時間的な設定順序での) 後法は前法に優先
- 特別法は一般法に優先 (たとえば民法と商法)

という優先順位があり、さらにこの適用は上のものが下のものに優先される。

(広義の) 不文法には、主要なものとして

- 慣習法
- 判例法

がある。慣習法とは、法令の適用方法を規定した法律である法例の2条に

[1] 「法源」の分類にはいくつかあるが、ここでは法源を広く解釈する立場に立っている。

公ノ秩序又ハ善良ノ風俗ニ反セサル慣習ハ法令ノ規定ニ依リ認メタルモノ及ヒ法令ニ規定ナキ事項ニ関スルモノニ限リ法律ト同一ノ効力ヲ有ス

と規定されている。判例法とは、判決文の中に記述されている判決理由が他の事例にも効力をもつことで、「先例拘束性の原則」を取っていない日本では法源とはならないはずであるが、実際には重要視されている。先例に反する場合には判例変更の手続きが必要となっている。罪刑法定主義という原則を持っている刑法においても、刑事訴訟法405条2号と3号で「最高裁判所の判例と相反する判断をしたこと」が上告理由になると規定されている。

この他に、裁判(裁判官の価値観)に影響を与える不文法として

- 学説
- 社会的規範
- 産業政策

などが挙げられている。学説は、法令の体系化を計ったり罪の軽重のバランスをとったりする学者の説であり、社会的規範や産業政策も時代と共に変化するものであり、これらは法源性は規定されてはいないが、判決への重要な要素のひとつであるといわれている。

これらの中でもっともデータ量が膨大なのが判例である。内容は2.3節で検討するが、個々の判例は長大な自然言語で記述されており、データ量の点からも文献データベースに匹敵している。このような法源の種々のデータ/知識は人手に負えるものではなくなっており、データベースの側からの強力なサポートが強く求められている。

2.2 いかに判決を行うか?

まず例を考えてみよう。民法90条[公序良俗違反]には

公ノ秩序又ハ善良ノ風俗ニ反スル事項ヲ目的トスル法律行為ハ無効トスル

と書かれている。単純化して考えれば、この条文は

$$\text{法律行為}(X) \wedge \text{公序良俗違反}(X) \supset \text{無効}(X)$$

と論理式で書くことができる。しかし、法令がいかに単純でも、これを実際に適用するには大きな問題点がある。つまり、何が公序良俗違反かが法令中に明記されていないのである。たとえより具体的な記述であったとしても(たとえばPKO法案の「パリ和平協定の破綻」)同様で、条文の解釈が重要になってくる。また「公序良俗」という概念は時代と共に変化するものでもあるので、法令は抽象的であるのが望ましく、(法令に記載されていない)解釈を時代に合わせていくべきだという説もある。

そのために、判決に必要なのは、

- 事例の事実認定
- 法令の解釈
- 法令の適用

の3つであると考えられる。

事実認定は判決予測の出発点として必要不可欠のものであるが、すべてを計算機で処理するわけにはいかない。したがって考えられるのは、立証された事実の集まりに対して、どれを重要視するか、どれを軽視するかの価値判断が重要で、これを判決予測のために計算機でシミュレーションすることが考えられる。

法令を解釈するためには2.1節の不文法が必要となる。たとえば上の「公序良俗」の条文に続いて、[2]には判例(を抽象化したもの)として以下のようなものが19例掲載されている。

「賭博の用に供されることを知ってする金銭消費賃貸契約は公序良俗に反し、無効である。」(最判昭61・9・4判時1215-47)

「金地金の先物取引の委託が著しく不公平な方法によって行われたときは、商品取引法違反かどうかを論じるまでもなく、公序良俗に反し、無効である。」(最判昭61・5・29判時1196-102)

事例は多様であるので、このような判例から公序良俗違反の事実を集めても限界がある。そのために

- 判例からの類推を行う。つまり概念階層が与えられていると、その階層内の近さにより判断する。たとえば、「賭博用と知ってする家屋賃貸契約は無効か?」に対し、金銭消費賃貸契約と家屋賃貸契約が概念上近ければ、法令が適用可能と判断し適用する。
- 公序良俗違反を規定するルールを追加する。つまり判例から判決理由を判例ルールとして取り出す。たとえば上の判例の場合

$$\text{賭博用}(X) \wedge \text{金銭消費賃貸契約}(X) \supset \text{公序良俗違反}(X)$$
$$\text{不公平}(X) \wedge \text{金地金先物取引}(X) \supset \text{公序良俗違反}(X)$$

が取り出せる。この判例ルールをいかに取り出すかは2.3節で議論するが、この場合には上で述べたように適用範囲が限られている場合と過剰な適用になる場合があるという問題がある。そこでルールの抽象化あるいは特殊化が必要となる。

- ルールの抽象化 (下線部分が抽象化されている)

  賭博用 (X)∧契約 (X)⊃ 公序良俗違反 (X)
  不公平 (X)∧取引 (X)⊃ 公序良俗違反 (X)

  これによって契約あるいは取引の下位概念に入っているものがすべて公序良俗違反となる。
- ルールの特殊化 (下線部分によって特殊化されている)

  賭博用 (X)∧ 金銭消費貸借契約 (X)∧ 高額 (X)
  ⊃ 公序良俗違反 (X)

  これによって高額でないものは公序良俗違反とはならない。

このような抽象化・特殊化は自動的に行うことは難しく、対象領域のヒューリスティクスの利用、人との対話などの他の価値判断が必要となる。

法令の解釈の抽象化の例としては以下のものがある [6]。共同正犯を実行者として規定している刑法60条 (二人以上共同シテ犯罪ヲ実行シタル者ハ皆正犯トス) の場合、条文からは読めない、共同意思主体説による共謀共同正犯とか、さらに暗黙の共謀とか順次共謀といったように、判例によって共同正犯の概念が拡張され定着していっている。さらに判例の重要さとしては、裁判官の自由裁量がかなり許されている民法において考えることができる。たとえば著作権の問題を始め社会的変化が大きく、立法がそれに対応できない場合などに、司法が判例によって適宜対応しようとすることも多くなっている。このように法令の解釈においては、文理解釈や体系解釈、目的解釈 (「立法目的」や「司法政策的考慮」[2]) があるが、すでに見てきたようにその解釈において判例が (実質的に) 法令以上に大きな役割を持っている場合がある。

適用可能な判例が期待する結果と異なっている場合には、その判例が不当であることを訴え判例変更を求めたり、事実関係がその判例と異なっているのでその判例は意味がないことを主張しなければならない。

[2] たとえば1993年6月23日には東京高裁が、対応法令の不成立を受けて、民法900条4項 (非嫡出子の区別) の違憲判決を出している。

法令の適用は、(判例ルールを含む) 法令解釈によって新しい事例に対し法令の適用の可能性が出てきたときに行われる。この場合は、すべてを奇麗な演繹推論でできることは稀で、仮説推論や仮説生成 (アブダクション) といった推論が必要となる。

判決のためには、このように法源を複雑に用いた推論が必要となるので、予測される判決 (矛盾するものが複数出力されるかもしれない) を検証する手段も必要となろう。

2.3 法源の表現

法令は比較的簡潔な自然言語で記述されているので、それを法的知識として計算機用の表現言語で表現するのは (他の法源に比べれば) 比較的容易だろう。慣習 (社会的規範) については、常識ベースを構築するのと同様な、本質的な困難さがあるが、実際の判決で使用される頻度はそれほど多くない。一方判例は、2.2節でみたように非常に重要な役割を持っている。そこで本節では判例の表現について考えたい。

判例は自然言語で具体的事実等を記述した長文のものであって、そのままでは法規としては使用できない。一般的に判例は、

- 判決理由
- それ以外の傍論

とからなっており、その判決理由を抽象化した判例ルールが制定法のように機能する。ここで問題なのは、

- 判決理由と傍論の区別が一意的ではない
- 判決理由の抽象化の基準が不明確である

ことである。前者が異なれば違った判例ルールが出されるし、後者が異なれば適用範囲の違った判例ルールとなる。いずれにしても一意的な判例ルールの出力は難しいので、判決予測時の判例ルールの動的な抽出が必要となろう。

自然言語で記述された判例を一次情報とすれば、上記の判例ルールが実際に判決予測に使用される2次情報 (知識) となる。判決文自体はCD-ROM化されており、簡単に入手することができ、また多くの情報検索サービスも提供されている [10]。しかし情報の質とサービスの機能を考えれば、手作業で判例を探す方が正確で迅速だという意見も根強くある [3]。この情報検索と判決予測の関係については、本稿では4節で触れるに留めたい。

問題は2次情報をいかに表現するかである。一般的な言い方をすれば、元の情報が自然言語で記述されているので、それと同等な表現が要求される。しかし計算機で処理する必要性から、述語論理 (の部分クラス) の拡張に基づいた言語を用いる場合が多い。その理由は、

- 対象領域が論理と親和性がある (判決は「論理」的)
- 推論過程を理解しやすい (判決の理由づけと論理の証明過程の類似)

などからである。一般的に表現方法を論じるのは難しいので、本稿では3節での *Quixote* を用いた記述、システムの試作を踏まえて、判例の表現の問題と今後の拡張点を4節で議論することにする。

3 判決予測システム TRIAL

3.1 TRIAL の設計方針

法的推論は 2.2 節で述べたように、事実認定、法令解釈、そして法令適用の3つの段階から構成されている。

事実認定自体はすでに述べたように本質的に難かし過ぎる。そこで、新規の事案に対し必要な事実はすでに認定されており、そのどれを重視するか、あるいはどれを選択するかが実験できる環境を提供することを検討することとする。

法令解釈は、人工知能の観点からは最も興味深いテーマのひとつである。判決予測システム TRIAL では、法令適用のみならず法令解釈に焦点を合わせた。法令解釈のためには多くのアプローチがあるが、TRIAL では下記のステップを採用している。

- 類比検出
  新規の事案が与えられると、既に存在する判例データベースからその事案に対して類似の判例を検索する。
- ルール変換
  類比検出によって抽出した判例 (判例ルール) を、新規の事案がそれらに適用できるようになるまで抽象化する。これを抽象判例ルールと呼ぶ。
- 演繹推論
  ルール変換によって変換した抽象判例ルールに、新規の事案を演繹的方法で適用する。この段階は、同じ方法が使用される法令適用を含んでいる。
- 事案修正
  上のステップで導かれた結論 (判決) に不満があれば、事案中の事実を変更し、再度上のステップを繰り返す。

これらのステップの中で、類比検出のための戦略はより良い判例のより効率的な検索のために法的推論に欠くことのできないものであり、それは法的推論の結果の質を決定する。TRIAL の主要な目的は、この分野での *Quixote* の可能性を調査することと、プロトタイプ・システムを開発することであったので、この中の小さなターゲットにのみ焦点を絞った。すなわち、どの範囲まで、ひとつの新しい事案のために法令ルールが抽象化されるべきかということである。それはもっともらしい説明を伴った回答を得るためであって、(本質的に困難な) 一般的な抽象化機構を得るためではない。また一般的に判例や事案は必ずしも完全な情報ではないので、演繹推論のステップでは *Quixote* の仮説推論や仮説生成の機能を使い、試行錯誤的に判決予測の実験ができる環境を提供する。

その他に TRIAL システムを *Quixote* 上に構築するに当たって下記の方針を採った。

- 使用する法律の知識はすべて *Quixote* で記述する。
- 抽象化と特殊化は *Quixote* の包摂関係に対応させ、包摂制約中の変数の値域を変化させることによって抽象化と特殊化を実現する。
- 抽象化と特殊化の制御は TRIAL 側で実行し、この結果を *Quixote* の親言語インタフェースに渡す。
- ユーザ・インタフェースは判決予測に特化したものとし TRIAL 独自で構築するが、束表示等は *Quixote* のインタフェースを利用する。

3.2 *Quixote* の特徴

TRIAL のシステムを説明する前に、*Quixote* 言語/システムの主要な特徴をまとめておこう。詳細は [13, 14] などを参照されたい。

- (純) Prolog に対し上位互換性をもっている。
- oid は明示的に表現され、その集合は包摂関係に基づく束構造 (つまり概念階層) を構成している。
- 集合 - 要素関係に基づくクラス - インスタンス関係は導入しない。

- 包摂制約によってプロパティを表現し、またこれによって論理とオブジェクト指向の統合を行う。
- メソッドはプロパティとして表現され、その実現はルール本体に記述する。
- 大規模データ / 知識の分類機構として (パラメータ付) モジュールを導入する。
- 問合せ機能として、仮説推論、包摂制約の仮説生成 (アブダクション)、説明機能を付加する。仮説推論を制御するために入れ子トランザクションを導入する。
- データベース機能として、動的更新、永続性管理、排他制御機能を持たせる。

従来の Prolog によっては書けなかったり、書きにくかった点が、これらの特徴により効率的に表現できている (3.5 節参照)。

3.3 TRIAL システム

TRIAL のすべてのデータと知識は *Quixote* 言語で記述されているが、TRIAL システム自体は KL1 で書かれ、*Quixote* システム上に構築されている。図 1 に、全体的な構成を示す。この図の中で、波線が *Quixote* と

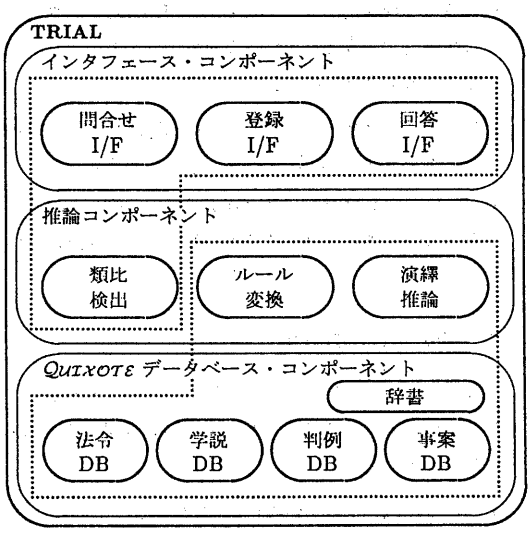

図 1: TRIAL のシステム構成

TRIAL の役割分担を示している。つまり *Quixote* がサポートしているのは、データベース・コンポーネントの他に、基本的な機能としてルール変換と演繹推論である。一方 TRIAL は、インタフェース・コンポーネントに加え、類比検出の機能をサポートしている。*Quixote* はサーバ、TRIAL はクライアントの位置を占めている。なお図中で "DB" と書かれているものは *Quixote* のモジュールに対応している。

3.4 実験例 — 労働法

TRIAL では労働法を対象として実験を行った。ここでは類比検出について論じるために、"過労死" に関連する、以下の簡素化した例 (事案) を考える。

> 運転手の花子は、"S" という会社に雇用されていた。仕事の合間に仮眠をとっていた際、心臓麻痺により死亡した。この事件に労働法が適用されうるか?

これは、下記のように *Quixote* の、モジュール "事案" として表現される。

```
事案::{{事案 /[対象者=花子,
              状況=仮眠,
              結果=心臓麻痺];;
       関係 [状態=雇用, 従業員=花子]
            /[所属=組織 [名前="S"],
              業務→運転手]}}
```

ここで ";;" はルール間の区切り文字である。このモジュールは、事案データベースに格納される。ここでは、業務遂行性と業務起因性の 2 つの抽象化された判例があると仮定する[3]。

```
事件1 :: 判決[事件=X]/[判決→業務起因性]
          ⇐関係 [状態=Y, 従業員=Z]/[原因=X]
           ||{X ⊑parm. 事件,
              Y ⊑parm. 状態,
              Z ⊑parm. 従業員 };;
事件2 :: 判決 [事件=X]/[判決→業務遂行性]
          ⇐ X/[状況 = Y, 結果 = Z]
           ||{ Y ⊑業務
              X ⊑parm. 事件,
              Y ⊑parm. 状況,
              Z ⊑parm. 結果 }.
```

[3]本稿では、ルール変換の段階は省略し、抽象判例ルールは与えられているものと仮定している。

両方のルールの中の、変数 X,Y そして Z がオブジェクト parm のプロパティによって制限されていることに注目されたい。すなわち、これらは既に parm によって抽象化されていて、これらの抽象化の程度は、parm のプロパティによって制御されている。このような判例が、類比検出によって判例データベースから検索され、またルール変換によって抽象化されている。下記のような、(法令データベースの中の) 労働法と (学説データベースの中の) 学説を考察しなければならない。

労働法:: 組織[名前=X]/[責任→保証 [対象=Y,
金銭=給与分]]
⇐事案 /[対象者=Y, 結果→病気],
判決 [事件→事件]/[判決→保険],
関係[状態=Z, 従業員=Y]
/[所属=組織 [名前=X]].

学説:: 判決 [事件=X]/[判決→保険]
⇐判決 [事件=X]/[判決→業務起因性],
判決 [事件=X]/[判決→業務遂行性]
||{$X \sqsubseteq$ 事件 }.

さらにまた、下記の parm オブジェクトを定義しなければならない。

$parm$ :: parm/[事件 = 事件,
状態 = 関係,
状況 = 業務,
結果 = 病気,
従業員 = 人].

事件$_1$ と事件$_2$ に parm を使うために、サブモジュール関係を下記のように定義する。

$$parm \sqsupseteq_S \text{事件}_1 \cup \text{事件}_2.$$

この情報は、ルール変換の際に動的に定義される。その上に、包摂関係を定義しなければならない。

事件 ⊒ 事案	関係 ⊒ 雇用
業務 ⊒ 仮眠	病気 ⊒ 心臓麻痺
人 ⊒ 花子	業務起因性 ⊒ 保険
	業務遂行性 ⊒ 保険

このような定義は、辞書の中に前もって格納される。

これで準備ができたので、上記のデータベースに対して仮説を伴う質問をすることができる。

- もし "事案" が "$parm$" と "学説" を継承したら、どのような種類の判決を受けとることができるか?

?-事案: 判決 [事件=事案]/[判決=X]
if 事案 $\sqsupseteq_S$ $parm$∪ 学説.

この問合せに対し 3 つの回答が得られる。

- X = 業務遂行性
- もし "事案:判決 [事件=事案]" が、"判決→業務起因性" というプロパティを持っていたら、"$X \sqsubseteq$保険"
- もし "事案:関係 [状態=雇用, 従業員 = 花子]" が、"原因=事案" というプロパティを持っていたら、"$X \sqsubseteq$保険"

これらのうち 2 つは仮定を含む回答であり、仮説生成機能がなければ返ってこない答である。

- もし "事案" が "労働法" と "$parm$" を継承すれば、花子が所属している組織はどのような責任を負うべきか?

?-事案: 組織 [名前="S"]/[責任=X]
if 事案 $\sqsupseteq_S$ $parm$∪ 労働法 ∪ 学説.

次に 2 つの回答を得る。

- もし、"事案:判決 [事件=事案]" が "判決→業務起因性" というプロパティを持っていたら、"$X \sqsubseteq$ 保証 [対象=花子, 金銭=給与分]"
- もし "事案:関係 [状態=雇用, 従業員=花子]" が "原因=事案" というプロパティを持っていたら、"$X \sqsubseteq$ 保証 [対象=花子, 金銭=給与分]"

いずれの答えも仮説を持っている。

類比検出のために、parm オブジェクトは、

- "事件$_1$" と "事件$_2$" にあるように、いかにルールを抽象化するか、
- どのプロパティを parm の中で抽象化するか、
- parm のプロパティの中にどんな値をセットするか

を決定する上で本質的な役割を果たしている。TRIAL の試作システムでは、TRIAL 側で類比検出を行い、parm によって *Quixote* に抽象化を行わせるという実験をすることができた。

TRIAL のユーザ・インタフェースのために、*Quixote* は、もし必要ならば、回答と共に説明 (導出グラフ) を返している。TRIAL のインタフェースは、ユーザの要求に従ってこれをグラフとして表示する。仮定と対応する説明の妥当性から回答を判断することによって、ユーザはデータベースの更新や抽象化の戦略を変更することができる。

3.5 TRIAL から見た *Quixote* の特徴

労働法を対象にした TRIAL の実験で有効に働いた *Quixote* の特徴としては以下のものがある。

- オブジェクト指向と論理の両方の性質をもった DOOD の表現力が、3.4 に見られるように DOOD の狙い [12] 通りに、複雑な法源の記述に有効であった。
- モジュールにより大規模知識ベースの法源の分類が簡単に行えた。
- 包摂制約によりルールの抽象判例ルールを表現でき、それを簡単に制御することが可能だった。
- 仮説推論によりモジュールを動的に結合・分離することが可能で、柔軟な知識ベースが構築できた。
- 仮説生成と (本稿では述べていないが) 説明機能により判決予測を試行錯誤的に行う思考実験の場が構築できた。

一方、TRIAL を *Quixote* 上で実現する際に直面したいくつかの問題点としては以下のものがあった。

- 設計論
  表現能力の向上と設計の容易性は必ずしも両立しない。Prolog の場合もそうであったが、とにかく書けば動くために、効率の悪いプログラム (たとえば変数がヘッドのルール) が多く見られた。
- 意味論
  Quixote は (純) Prolog との上位互換性はもっているが、オブジェクト指向概念をもった言語であるために (商用) Prolog とは手続き的意味論が異なっている。ユーザは当初、それらを混同し (たとえば backtrack) 意図しない答えが戻ることが多かった。*Quixote* プログラマ用デバッガが必要である。
- 表現能力
  判例はもともと自然言語で記述されているために高度の表現能力が要求される。たとえば否定 (negation-as-failure と classical negation) とか選言の導入への要求が多かった。*Quixote* は包摂関係の制約解消系を使っているために、それらの導入は制約の可解性に影響を及ぼすために導入していないが、オブジェクト自体への否定または選言の導入は検討されるべきだろう。
- 説明機能
  判決予測で必要とする論証 (説明) と *Quixote* が返す説明 (証明木でデバッガ用にも使える) が一致していないことが問題とされた。これには応用側とのさらなる詰めが必要だろう。
- 発火条件
  選択されたルールをすべて発火させるのではなく、エキスパートシステムによく見られるように事前に発火条件を評価することが求められた。HELIC-II で導入している確信度は静的評価が可能であるので必ずしも本質的ではないが、処理効率を考えれば処理系の対応を考えなければならない。
- 冗長な解
  Quixote の現在の実装では効率の向上のために、極小解を必ずしも求めていない。このために冗長な解を出力していた。
- 効率
  演繹データベースを含め問合せの最適化は処理効率に劇的な変化をもたらす。ところが *Quixote* では制約の存在のためにこれが不完全である (有限領域の制約解消系の場合を除けばこれはまだ未解決の問題である)。

全般的には 3.3、3.4 節で見たように、従来の表現言語やシステムに比べかなり効率的な環境を与えたと考えている。TRIAL システムの開発期間は、データの入力を除けば 3 人月程度であり、この面でも *Quixote* 上へのシステム構築は有効だった。

4 法的推論システムとデータベース

今後さらに本格的な判決予測システムを構築するために、また他の応用のための次世代データベースを展望するときには、さらに多くの検討点が必要である。TRIAL の

経験から得られたいくつかの検討点の中で、大きなものを挙げる。

- データや知識の貯蔵庫から思考実験の場へ
  Quixote の仮説推論、仮説生成をもった問合せ機能は、判決予測を試行錯誤的に実験する場を提供した。これは、判例のような法源データベースが、実世界の明解な部分だけを切り出した従来のデータベースと違って、不確定な部分情報のデータベースであるために、本質的な機能である。これは、多くのノイズや実験データを含んだ科学データベースなどの新しい応用に有効となるだろう。*Quixote* では、仮説推論については仮説情報は静的評価し入れ子トランザクションで制御し、仮説生成は包摂制約に関するアブダクションによって実現したが、それらの効率的実装についてはさらに検討が必要と思われる。

- 大規模データベース / 大規模知識ベースの管理
  実験で使った TRIAL データベースは比較的小規模のものだったが、質的には大規模のものと変わりはないので、大規模データベース / 大規模知識ベースとして設計を行った。ひとつのデータベース中にさまざまな (ときには互いに矛盾した) データが混在しているのでそれらの分類・継承機能が必要となる。このために *Quixote* のモジュール機能を利用した。さらに本格的なデータベース / 知識ベースを構築するためにはオントロジーの局所性 / 大域性が問題となるだろう。それをいかに整理するかが今後の検討課題として必要になるだろう。

- 異種データベース / 知識ベースの管理 (図 2)
  判例データベースは 2.3 節で見たように一次情報、二次情報、等から構成されている。一次情報に対する操作は高度な情報検索機能であり、二次情報に対しては類推を含む類似検索であった。一次情報は唯ひとつしか存在しないが、二次情報は複数存在しうる。これは異種データベースの一種ではあるが互いの間に制約があるという点で異なっている。問合せについても包括的なものが考えられる。これらに無理やり統一的な意味論を付与するのは KIF [1] の二の舞いになる危険性があるだろう。今後の拡張として、図 2 のような基本アーキテクチャのデータベースを現在 (問合せを含め) 検討中である。これは単に法的データベースのみならず、自然言語データベース、科学データベース、マルチメディア・データベースにも共通するアーキテクチャである。

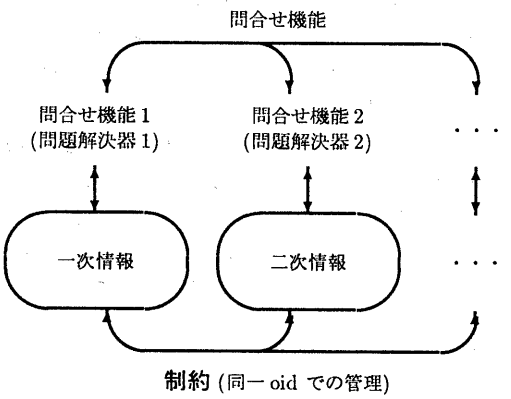

図 2: 異種情報の統一的管理

- 知識発見
  判例から判例ルールを抽出 (図 2 の二次情報を生成) する作業をすべて人手で行うには無理がある。そこでこれを実現するために欠かすことのできない技術として大規模データベースからの知識発見 [7] がある。これは、法的データベースに限らず、今後のデータベースの中心技術のひとつになるのは間違いないだろう。

- 他分野の技術との統合化
  法的データベースのような大規模データベースを考えると、複数の分野の技術の統合が必要となってくる。たとえば図 2 で一次情報から二次情報を生成するには自然言語処理は不可欠であるし、問合せ機能にも情報検索機能はもちろん類推を始めとする人工知能技術が必要である。拡張可能データベースのアーキテクチャにはこのような共同作業が不可欠となろう。

- 応用とデータベースの境界
  データベースの高度化はある意味では応用機能のデータベースへの取り込みでもあった。TRIAL ではルールの抽象化・特殊化の制御をユーザ・プログラムで行ったが、従来のような区分けが必ずしもうまくいくとは限らないと考えられる。単純な応用でも、Kappa-P [13] に見られるように、並列処理の効果を発揮するためにはデータベース管理システム中にユー

ザ・ルーチンを取り込むことが必須である。さらに *Quixote* の応用のひとつとして検討している数式データベースでは、データベースからの (基本機能の) データ検索自体に ACI マッチングが必要とされるが、これらすべてをデータベース機能として抱え込むのは不可能である。応用が高度化していくにしたがって (オブジェクト指向プログラミング言語のライブラリのような) 開放型の拡張可能なシステムで、かつ多言語インタフェースをもつものが必要となるだろう。

5 おわりに

本稿の標題「データベースに判決は予測できるか?」に対する答えはすでに 4 節に暗示されている。つまり、従来のデータベースの概念に固執する限りこの答えは「否」であるが、データベースをさらに他分野と融合発展させた基本技術のひとつと考えるならば答えは「可」であると考える。法的データベースの他にも、遺伝子データベース、地理データベース、マルチメディアデータベースなど、従来のデータベースとは性質を異にした、新しいデータベースを必要とする応用が増えている。それら必要機能を備えた、*Quixote* の次のデータベースの設計を現在 ICOT で開始している。

謝辞

法的推論一般および HELIC-II について辛抱強く教示して頂いた ICOT の新田克己氏、孤軍奮闘しながら TRIAL の設計開発を担当した日立製作所の山本晃一郎氏、また日頃 *Quixote* について議論している *Quixote* グループに謝意を表する。

参考文献

[1] M.R. Genesereth and R.E. Fikes, "Knowledge Interchange Format Version 3.0 Reference Manual", *Logic Group Report Logic-92-1*, June, 1992.

[2] 判例六法編修委員会 (編)、『模範六法』、三省堂、1989。

[3] 細見利明、"データベースと弁護士実務"、『自由と正義』、vol.41、no.1、pp.62-67、1990。

[4] 五十嵐清、法学入門、一粒社、1979。

[5] L.O. Kelso, "Does the Law Need a Technological Revolution?", *Rocky Mt. Law Rev.*, vol.18, pp.378-392, 1946.

[6] 村井敏邦、"刑法の最新判例を読む"、『法学セミナー』、vol.41、no.1、1989。

[7] 西尾章治郎、"大規模データベースにおける知識獲得"、情報処理、vo.33、no.3、1993。

[8] K. Nitta, Y. Ono, T. Chino, T. Ukita, and S. Amano, "HELIC-II: A Legal Reasoning System on the Parallel Inference Machine", *Proc. Int. Conf. on Fifth Generation Computer Systems*, ICOT, Tokyo, June 1-5, 1992.

[9] 柴崎真人、山本晃一郎、近藤秀文、横田一正、"法的推論システム TRIAL の開発"、情報処理学会全国大会、3 月、1993。

[10] 田島裕、法律情報のオンライン検索、丸善、1992。

[11] N. Yamamoto, "TRIAL: A Legal Reasoning System (Extended Abstract)", *Joint French-Japanese Workshop on Logic Programming*, Renne, France, July, 1991.

[12] 横田一正、"演繹オブジェクト指向データベースについて"、コンピュータソフトウェア、vol.5、no.4、1992。

[13] K. Yokota and H. Yasukawa, "Towards an Integrated Knowledge-Base Management System", *Proc. Int. Conf. on Fifth Generation Computer Systems*, ICOT, Tokyo, June 1-5, 1992.

[14] K. Yokota, H. Tsuda, and Y. Morita, "Specific Features of a Deductive Object-Oriented Database Language *Quixote*", *Proc. SIGMOD Workshop on Combining Declarative and Object-Oriented Databases*, Washington DC, May 29, 1993.