

演繹・オブジェクト指向データベースによる
判例データベースの構築

山本 展一郎

新世代コンピュータ技術開発機構

演繹・オブジェクト指向データベース言語 *QUIXOTE* による応用システムとして、判例データベースを構築することを検討している。この判例データベースは、モジュール化して蓄積した既存の判例群を新しい事案に対しても類推適用することにより、その事案の法的判断を予測するための判例類推適用システムである。

QUIXOTE の演繹推論機能を拡張して類推を導入することにより、また *QUIXOTE* のオブジェクト指向機能を利用して判例群をモジュール化することにより、この判例の類推適用機能を実現する方法を提案する。

A Legal Precedent Database
as a Deductive and Object-Oriented Database

Nobuichiro YAMAMOTO

Institute for New Generation Computer Technology(ICOT)

1-4-28,Mita,Minato-ku,Tokyo 108,JAPAN

e-mail: nyamamot@icot.or.jp

This paper presents a legal precedent database which has been developed in the deductive and object-oriented database language *QUIXOTE*. This legal precedent database is a system which predicts a legal judgement for a new case by analogically applying existing legal precedents.

This function is realized by analogical reasoning which is an extension of the deductive feature of *QUIXOTE*, and by modularization which is an object-oriented feature of *QUIXOTE*.

1 はじめに

1.1 法的推論

演繹・オブジェクト指向データベース言語 *QUIXOTE*[1] による応用システムとして、法的推論システムの要素技術の一つとしての判例データベースを構築することを検討している。

法的推論とは、法律家が法的問題を解決するための法的判断に至る過程で行う推論であり、その法律家の活動を支援するシステムが法的推論システムである。一口に法的推論といってもその対象とする法的情報は多様であり、また、法的推論システム全体が必要とする基本的機能も多岐に渡る [7]。本稿では、考察の対象を判例、特に業務災害の認定に関する労働判例の周辺に絞り、より柔軟な法的推論システムの実現に向けて、演繹・オブジェクト指向データベースによるアプローチを検討する。なお、このモチベーションは [2] と同様であるが、アプローチが異なっている。

1.2 解釈と判例

法律の条文は抽象的な表現がなされている。この抽象的な条文の意味内容を具体的に明らかにすることを解釈という。

例えば、労働基準法第 75 条には、

「労働者が業務上負傷し、又は疾病にかかった場合においては、使用者は、その費用で必要な療養を行い、又は必要な療養の費用を負担しなければならない。

前項に規定する業務上の疾病及び療養の範囲は、命令で定める。」

とあり、労働者が業務上負傷したり、疾病にかかったりした場合の使用者の療養に関する補償責任について規定している。これに関連して、労働基準法施行規則第 35 条において、業務上の疾病の範囲が典型的に列挙されているが、個々の事例に即して判断する途も残されており、その疾病が業務上か業務外かの判断、いいかえれば業務上外の認定については、解釈に委ねられている [8]。

裁判官は、持ち込まれた訴訟に即して、その解釈を行う。その解釈は裁判官の裁量に任されているが、全く自由に行えるわけではなく、裁判官は通常は既存の判例を無視することはできない。また、裁判官に対し自己の期待する裁判を求める立場にある弁護士なども、どういふ裁判がなされるかの見通しを立てるのについて、裁判官を拘束している判例に間接的には支配されているということができる [6]。

したがって、判例は将来の裁判を予測する上で重要な意味を持ち、既存の判例の類推適用という方法で将来の裁判が予測され得る。判例の類推適用とは、既存の判例に類似している事案に、その既存の判例の判断を“準用”することであり [6]、判例の類推適用の機能を備えた判例データベースの構築は、法的推論システムに向けての重要な要素技術の一つとなる。

2 判例データベース

2.1 演繹・オブジェクト指向データベースとしての判例データベース

本稿で考察する、演繹・オブジェクト指向データベース言語 *QUIXOTE* による判例データベースとは、単にキーワード等により判例を探し出すための判例検索システムではなく、既存の判例を利用して新しい事案の法的判断を予測するための判例類推適用システムである。この判例データベースは、既存の判例から抽出した規則を、新しい事案の事実に対しても類推適用することにより、その事案に対する法的判断を予測し、その特別な場合として、判決も導出し、しかも、その法的判断に至った論理過程がその根拠とともに提示されるといふものである。これにより、判例を利用した法的思考実験のための環境を提供することが目標である。

この判例データベースの類推適用機能は、演繹推論システム (演繹データベース) としての *QUIXOTE* システムを、原口、有川の類推理論 [4] と同様にして、類推システムに拡張することで実現する。このとき、その類推システムの証明とそ

の証明時に得られる類比が、その法的判断に至った論理過程とその根拠を与えることになる。

さらに、この判例データベースは、その類推適用による法的思考実験が柔軟に行えるように、判例群をモジュール(部分データベース)化して蓄積する機能を持つ。柔軟であるとは、どの判例群を使用した類推適用を行うのかがユーザに選択可能であるということである。例えば、業務上外の認定を予測する場合、あるときは裁判所の判決のみから、またあるときは労働保険審査会の裁決や労働基準監督署の行政解釈等も含めて類推適用を行いたいということがあるだろう。この判例群のモジュール化機能は、*QUIXOTE* システムのオブジェクト指向データベースとしての機能を利用して、判例群をモジュール化して蓄積することで実現する [1],[3]。

2.2 判例として抽出する情報

何を判例として抽出するかに関しては専門的な議論がある [5],[6]。判例として予めある程度一般的な・抽象的な情報を抽出する場合、その抽出時にどこまで一般化・抽象化するかを事前に決定しておかなければならず、その時点で解釈を一義的に決めてしまうことになる [5]。“裁判官の行方であろう解釈”だけを生成するのが理想的であろうが、現段階では、可能性のある解釈を生成することにより、判例を利用したユーザの法的思考実験を支援することが現実的である。そこで、判例として抽出する情報は、具体的事実関係に基づいた具象的な情報のままに留めておき、推論により可能性のある解釈が生成できるようにしておく方が望ましい。

本稿では、[2]と同様に、具象的な認定事実および具象的な判断規則を判例として抽出することにする。認定事実とは、原告、被告双方の主張事実の中からその裁判において認定された事実のことである。判断規則とは、その裁判において示された、その認定事実から法的判断に至る論理過程のことである。また、これから法的判断を予測しようとする新しい事案に対しては、判断規則はな

く、また認定事実ではなく主張事実のみがある。

3 判例群のモジュール化

3.1 モジュール構成

判例データベースのモジュール(部分データベース)構成について述べる。

判例データベースは、マクロに見ると、各判例に関するそれぞれのデータベースと新しい事案に関するデータベースから構成される。少しミクロに見ると、各判例に関するデータベースは、各判例に関する認定事実データベースと判断規則データベースに分割される。これらの各判例に関する認定事実データベースと判断規則データベースは、共にその判例に関する概念辞書データベースをモジュールとして含んでいる。そして、新しい事案に関するデータベースは、その事案の概念辞書データベースをモジュールとして含んでいる、その事案の主張事実データベースのみから成る。このモジュール構成を図示したのが、図1である。このように分割しておく理由は、モジュール化して構成することで、様々な判例の認定事実や判断規則を取り換えた柔軟な類推が行えるようになるからである(第4節)。

さて、このモジュール構成は、*QUIXOTE* では、モジュール識別子間の包摂関係を定義することで表せる：

判例 db \supseteq 事案 db, 判例 db \supseteq 判例 1db,
判例 db \supseteq 判例 2db, ...,
事案 db \supseteq 事案主張事実 db,
事案主張事実 db \supseteq 事案概念辞書 db,
判例 1db \supseteq 判例 1 認定事実 db,
判例 1db \supseteq 判例 1 判断規則 db,
判例 1 認定事実 db \supseteq 判例 1 概念辞書 db,
判例 1 判断規則 db \supseteq 判例 1 概念辞書 db,
.....

例えば、“判例 db \supseteq 判例 1db”は、

“判例 db”は“判例 1db”をモジュール(部分データベース)として持つ

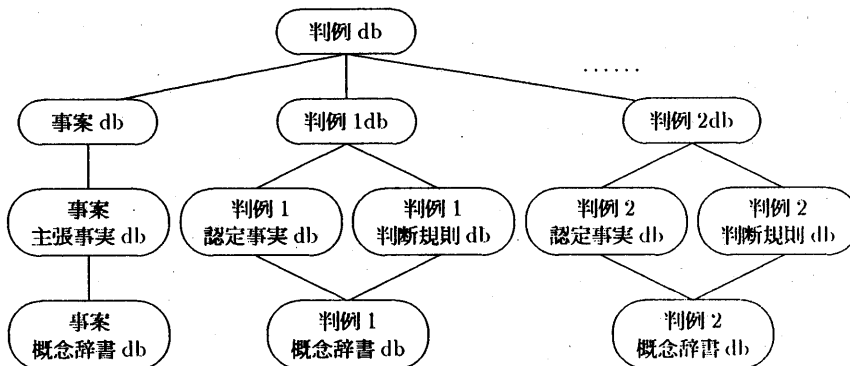


図 1: モジュール構成

ということの意味しており、“判例 1db”に含まれる認定事実や判断規則は“判例 db”にも含まれることになる。

また、上記のほかにすべての判例の判断規則を一つのモジュールにまとめておくと質問時に便利である(第4節):

判断規則 db ⊇ 判例 1 判断規則 db,
 判断規則 db ⊇ 判例 2 判断規則 db,
 ...

3.2 判例の記述例

各モジュール内のデータが, *QUICKOTE* により如何に記述されるかについて述べる。

図 2 に判例および事案の記述例を示す。ここでは、業務災害(負傷、疾病、障害または死亡)の業務上外の認定基準として、業務遂行性(業務遂行中に発生した災害であること)と業務起因性(業務と災害との間に相当因果関係があること)の二つの要件が充足されるか否かが判断されている部分を取り上げる [9]。そして判例 1 として、その業務遂行性に関しては、「事業主の支配下にあり、かつ管理下において業務に従事している場合」、すなわち「労働者が事業場施設内で業務行為中に発生した業務災害」という典型的な場合を取り扱う。また、新しい事案としては、「労働者“山本 0”が事業場施設“工場 0”内で業務行為“製造準備行為 0”中に発生した業務災害“打撲

0”」という場合を考える。

各判例および新しい事案の概念辞書データベースは、そこに現れる基本オブジェクト間の包摂関係の定義の集合となる。例えば、“判例 1 概念辞書 db”の 1 番目の式は、

“業務災害 1”は業務災害である

ということを表している。*QUICKOTE* の構文に従えば、実際には、“判例 1 概念辞書 db ::”をその先頭に書くことで、そのモジュールに含まれることを表す。

各判例の認定事実データベースおよび新しい事案の主張事実データベースは、それら基本オブジェクトの具象的な属性値の定義の集合となる。例えば、“判例 1 認定事実 db”の 1 番目の式は、

“業務災害 1”は、“労働者 1”がその被災者であり、“事業場施設 1”がその発生場所であり、そして“担当業務行為 1”がその発生状態である

ということを表している。“判例 1 認定事実 db ::”に関しては上記と同様である。

各判例の判断規則データベースは、ある基本オブジェクトの属性がある値をとるときのある基本オブジェクトの属性値を定める具象的な(変数を含まない)規則の集合となる。例えば、“判例 1 判断規則 db”の 2 番目の式は、

判例1 概念辞書 db

業務災害 1 ⊆ 業務災害
 業務遂行性 1 ⊆ 業務遂行性
 業務起因性 1 ⊆ 業務起因性
 労働者 1 ⊆ 労働者
 事業場施設 1 ⊆ 事業場施設
 担当業務行為 1 ⊆ 担当業務行為
 担当業務行為 ⊆ 業務行為
 業務行為 ⊆ 行為
 部署 1 ⊆ 部署
 事業主 1 ⊆ 事業主
 ...

事案概念辞書 db

打撲 0 ⊆ 業務災害
 業務遂行性 0 ⊆ 業務遂行性
 業務起因性 0 ⊆ 業務起因性
 工場 0 ⊆ 事業場施設
 会社 0 ⊆ 事業主
 計算機 0 ⊆ 製品
 製造準備行為 0 ⊆ 業務準備行為
 業務準備行為 ⊆ 業務行為
 山本 0 ⊆ 労働者
 ...

判例1 認定事実 db

業務災害 1/[被災者 = 労働者 1,
 発生場所 = 事業場施設 1,
 発生状態 = 担当業務行為 1]
 労働者 1/[業務 = 担当業務行為 1,
 所属 = 部署 1]
 事業場施設 1/[管理者 = 事業主 1]
 担当業務行為 1/[支配者 = 事業主 1]
 ...

事案主張事実 db

打撲 0/[発生場所 = 工場 0,
 発生状態 = 製造準備行為 0,
 被災者 = 山本 0]
 工場 0/[管理者 = 会社 0,
 製造物 = 計算機 0]
 製造準備行為 0/[支配者 = 会社 0]
 山本 0/[業務 = 製造準備行為 0]
 ...

判例1 判断規則 db

業務災害 1/[認定 = 業務上]
 ⇐ 業務災害 1 ⊆ 業務災害,
 業務遂行性 1 ⊆ 業務遂行性/[災害 = 業務災害 1, 業務 = 担当業務行為 1],
 業務起因性 1 ⊆ 業務起因性/[災害 = 業務災害 1, 業務 = 担当業務行為 1],
 担当業務行為 1 ⊆ 業務行為 .

業務遂行性 1/[災害 = 業務災害 1, 業務 = 担当業務行為 1]
 ⇐ 業務遂行性 1 ⊆ 業務遂行性,
 業務災害 1 ⊆ 業務災害/[被災者 = 労働者 1, 発生場所 = 事業場施設 1,
 発生状態 = 担当業務行為 1],
 担当業務行為 1 ⊆ 業務行為/[支配者 = 事業主 1],
 労働者 1 ⊆ 労働者/[業務 = 担当業務行為 1],
 事業場施設 1 ⊆ 事業場施設/[管理者 = 事業主 1],
 事業主 1 ⊆ 事業主 .

図 2: 判例および事案の記述例

“業務遂行性1”が業務遂行性であり、そして“業務災害1”が業務災害であり、その被災者、発生場所および発生状態が“労働者1”、“事業場施設1”および“担当業務行為1”であり、……、そして“事業主1”が事業主であるならば、

“業務遂行性1”は、その災害および業務が“業務災害1”および“担当業務行為1”である

ということを表している。ここに、“業務遂行性”、…、“事業主”は類推適用する場合の上限を示している(第4節)。“判例1判断規則db”に関しては上記と同様である。

4 判例の類推適用

4.1 質問の記述例

既存の判例を類推適用させる質問が、*QUINOTE*により如何に記述されるかについて述べる。

表1に質問の記述例を示す。*QUINOTE*では、質問時の指定により各モジュールの結合は動的にも行われる。これにより、判例の類推適用を行うのに、どの判例を使用し、あるいはどの判例を使用しないかが質問時に指定可能となる。

質問1は、判例1に関するすべての事実を知りたい場合の質問である。演繹推論のみを行う場合、裁判での判断も含めたすべての事実が導出されることになる。これは、特定のモジュール“判例1db”に対して、質問“ $\Leftarrow X$ ”を表している。

質問2はいずれも、事案が“業務上”災害と認定されるか否かを、指定した判断規則を類推適用して予測したい場合の質問である。類推適用される判断規則は、(1)はすべての判例、(2)は判例1のみ、(3)は判例1と判例2、(4)は判例1を除いた残りの判例のものである。ここに、“ \supset ”は質問時に動的に結合されるモジュール間の包摂関係を、また“ \cup ”、“ \setminus ”はそれぞれモジュール間の“和”、

“差”を表している。これらはいずれも、モジュール“事案主張事実db”に対して、指定したモジュールをその質問時だけ動的に継承させた上で、質問“ \Leftarrow 打撲0/[認定=業務上]”をすることを表している。動的に継承されるモジュールは、(1)は“判断規則db”、(2)は“判例1判断規則db”、(3)は“判例1判断規則db”と“判例2判断規則db”の和、(4)は“判断規則db”から“判例1判断規則db”を引いた差であることを表している。

質問3は、事案が“業務上”災害と認定されるか否かを、業務起因性は仮定した上で判例1の判断規則を類推適用して予測したい場合の質問である。これも、モジュール“事案主張事実db”に対して、モジュール“判例1判断規則db”を動的に継承させているが、さらに作業用のモジュール“db0”を用いて、事実“業務起因性0/[災害=打撲0,業務=製造準備行為0]”をその質問時だけ追加した上で、質問“ \Leftarrow 打撲0/[認定=業務上]”をすることを表している。

質問4は、“業務上”災害と認定される判例を知りたい場合の質問である。これは、不特定のモジュール“X”に対して、質問“ $\Leftarrow Y \sqsubseteq$ 業務災害/[認定=業務上]”をすることを表しており、モジュールを求めるための質問である。

4.2 類推戦略

実際にどのように類推適用を実現するのかについて述べる。判例としてどのような情報を抽出したかに応じて以下の方式が考えられるが、完成版において方式1を実装する前に、試作版において方式2を実装し、*QUINOTE*システムの有効性を確認するという方針で現在その詳細を検討している:

1. 具象表現 - 類推方式

判例からは具象的な規則のままを抽出しておいた場合、新しい事案にそれを類推により適用する。

2. 抽象表現 - 演繹推論方式

1		判例 1db :: $\Leftarrow X$.
2	(1)	事案主張事実 db \supseteq 判断規則 db :: \Leftarrow 打撲 0/[認定 = 業務上] .
	(2)	事案主張事実 db \supseteq 判例 1 判断規則 db :: \Leftarrow 打撲 0/[認定 = 業務上] .
	(3)	事案主張事実 db \supseteq (判例 1 判断規則 db \cup 判例 2 判断規則 db) :: \Leftarrow 打撲 0/[認定 = 業務上] .
	(4)	事案主張事実 db \supseteq (判断規則 db \setminus 判例 1 判断規則 db) :: \Leftarrow 打撲 0/[認定 = 業務上] .
3	db0 :: 業務起因性 0/[災害 = 打撲 0, 業務 = 製造準備行為 0] . 事案主張事実 db \supseteq (判例 1 判断規則 db \cup db0) :: \Leftarrow 打撲 0/[認定 = 業務上] .	
4	$X :: \Leftarrow Y \sqsubseteq$ 業務災害 / [認定 = 業務上] .	

表 1: 質問の記述例

判例から予め抽象的な規則を抽出しておいた場合、新しい事案にそれを演繹推論により適用する。

4.2.1 具象表現 - 類推方式

この方式は、抽出した規則が具象的なままであるため、新しい事案への適用のためには、通常の演繹を超えた推論が必要となる。

HELIC-II[2]では、事例ベース推論の立場から基本的にこの方式であり、以下のように実現している:

1. ユーザは具象的なままの判断規則を抽出するが、規則の条件部の各事実には、その規則におけるその事実の重要度を示す点数を与えておき、その合計をその規則の点数とする。
2. ユーザは質問時に規則を発火させるための点数の下限値を指定して質問する。
3. 推論エンジンは、以下のようにして、発火するための規則の点数を計算し、点数がユーザの指定した下限値以上であれば規則を発火させ、ユーザの質問に回答する:
 - 規則の条件部の事実とちょうど一致する事実があれば、その事実の最初の点数をそのまま与える。

- ちょうど一致していなくても、上位概念が一致している事実があれば、その上位概念を辿る段数分ほど、その事実の最初の点数から減算した点数を与える。
- 規則の条件部の各事実に対して、それらの点数の合計を規則を発火させるための点数とする。

得点の状況によっては、規則の条件部の各事実に対応する事実がなくても規則を発火させている。抽象化の上限を指定しておくこともできる。

判例データベースの完成版では、この方式を以下のように改良して実現することを検討している:

判例の類推適用は、既存の判例と新しい事案間に類似性を検出し、その類似性を用いて既存の判例での判断規則を変換することにより、新しい事案の事実にも適用しようとするものであった。この方式は、原口、有川の類推理論[4]を演繹推論システム(演繹データベース)である *QUINOTE* システムに拡張することにより実現可能である。その詳細は現在考察中の段階であるが、[4]には、「類推は通常の演繹が失敗した時点で起動される」と、「類比を事前に求める必要はない」という特徴があり、その証明が論理過程として、その類比が根拠として得られる点が有効である

と考えている。しかし、この方式でも次節で詳しく述べる抽象化の程度の問題は起こり、類比的の生成時に類推を制御する仕組みが必要である。

例として、図2における事案に対して、

事案主張事実 db ⊆ 判例1判断規則 db ::
⇐ 業務遂行性 0/[災害 = 打撲 0,
業務 = 製造準備行為 0]

すなわち、判例1から類推して新しい事案には業務遂行性があるか否かという質問をした場合を考える。

このとき、

{(業務遂行性 1, 業務遂行性 0),
(業務災害 1, 打撲 0),
(担当業務行為 1, 製造準備行為 0),
(労働者 1, 山本 0),
(事業場施設 1, 工場 0),
(事業主 1, 会社 0)}

の類比により、判例1の2番目の規則が

業務遂行性 0/[災害 = 打撲 0,
業務 = 製造準備行為 0]
⇐ 業務遂行性 0 ⊆ 業務遂行性,
打撲 0 ⊆ 業務災害
/[被災者 = 山本 0, 発生場所 = 工場 0,
発生状態 = 製造準備行為 0],
製造準備行為 0 ⊆ 業務行為
/[支配者 = 会社 0],
山本 0 ⊆ 労働者
/[業務 = 製造準備行為 0],
工場 0 ⊆ 事業場施設
/[管理者 = 会社 0],
会社 0 ⊆ 事業主

に変換され、その事案の主張事実(概念辞書を含んでいる)からこの規則が発火し、“yes”が回答されることが期待される。ただし、“打撲0”の属性(発生場所, 発生状態, 被災者)の順序, および“工場0”の余分な属性(製造物)とは無関係に、この規則は発火することがPROLOGの場合と異なる。また、この規則が発火するための類比的

候補は、置換可能なオブジェクトの上限(“業務災害”, “業務行為”, “労働者”, “事業場施設”, “事業主”)が与えられているので、それら上限との包摂関係で小さいものしかありえないことになる。

4.2.2 抽象表現 - 演繹推論方式

この方式は、抽出した規則の方に抽象性があるため、新しい事案への適用のためには、基本的には演繹推論があれば良いことになり、結局のところ推論エンジンの方で類推を行うのではない。

しかし、この方式ではどこまで抽象化するかを規則の抽出時に事前に決定しておかなければならず、この時点で解釈を一義的に決めてしまうことになる[5]。すなわち、抽象化の程度の問題を規則の抽出時にユーザに任せることになる。抽象化の程度が大きいほど、可能性のあるいろいろな解釈を生成することになるが、大き過ぎると不適切な解釈も多く生成してしまう。また、この方式では、抽出した判例の個別的事実関係を捨象しているので、適用した判例における事実と新しい事案における事実との間の類似性を示す情報は直接には得られないことになる。

例えば、判例1から予め抽象化した判断規則として、

X/[災害=Y, 業務=Z]
⇐ X ⊆ 業務遂行性,
Y ⊆ 業務災害/[被災者=A, 発生場所=B,
発生状態=Z],
Z ⊆ 業務行為/[支配者=C],
A ⊆ 労働者/[業務=Z],
B ⊆ 事業場施設/[管理者=C],
C ⊆ 事業主

を抽出しておいた場合、これを新しい事案(図2)の主張事実(概念辞書を含んでいる)に演繹的に適用すれば、その事案の業務遂行性は導かれることになる。しかし、例えば“担当業務行為1”と“製造準備行為0”との類似性を示す対応関係は直接には得られない。

一方、この規則の条件部の3番目の項は、“Z ⊆ 担当業務行為/[支配者=C]”に抽象化を留めておくこともでき、その場合は新しい事案の業務遂行性は導かれぬ。また、“Z ⊆ 行為/[支配者=C]”まで抽象化することもできてしまうので、それをユーザが規則の抽出時に選択しておかなければならない。

この方式に基づく判例データベースの試作版では、HELIC-IIでの点数による規則発火の方式を、判例データベースが規則を抽象化することでシミュレートし、以下のように類推適用を実現する：

1. ユーザは判断規則を抽象化せず、具象的な判断規則のまま、判例データベースに蓄積する(図2)。
2. ユーザは質問時に抽象化の程度を示す値(HELIC-IIでの、規則を発火させるための点数の下限値に相当するものであるが、それとは逆に、この値が大きければ大きいほど規則は発火し易くなる)を指定して質問する。
3. ユーザの指定した値に応じて、判例データベースが抽象化された判断規則を生成し、それをを用いて演繹推論によりユーザの質問に回答する。

抽象化の程度の問題は、ユーザの責任とする代わりに、抽象化の程度を示す値の指定により、抽象化の程度に応じた実験ができるようにする。また、行き過ぎた抽象化がされないように、ユーザは各基本オブジェクトの抽象化の上限を指定しておくこともできる。

5 おわりに

5.1 法的推論システムへの拡張

判例データベースが組み込まれた法的推論システム全体を一つの *QUIXOTE* システムの拡張として実現する場合の構成について触れる(図3)。

法的推論システムは、法的推論データベース、法的推論エンジンおよび法的推論インタフェースから構成される。法的推論データベースは全体をマクロに見ると、判例データベース、法律データベース、学説データベース等や辞書データベースがそれぞれ一つのモジュール(部分データベース)を作る。また、法的推論エンジンは演繹推論エンジンを基本として、類推エンジンや各種の推論エンジンがそれに加わる。

HELIC-IIとは違い、法律データベースとそのための演繹推論エンジンとのサブシステムに融合した形で判例データベースとそのための類推エンジンが付加される構成をとる。このうち、法的推論データベースと演繹推論エンジンは *QUIXOTE* システムとして実現される。

5.2 今後の課題

以下に今後の課題、将来の研究について述べる：

- *QUIXOTE* における原口、有川流の類推理論の詳細検討

QUIXOTE における類推と類推和の概念を厳密に定義し、その類推システムにおける推論が類推和からの演繹推論と同値であることを証明する。

- 判例の標準的・統一的な記述方式の設定

判例としてどのような情報を抽出し、如何に記述するかに応じて、同じ判例群を使用しても違う判例データベースができてしまう。判例の類推適用のためには何らかの標準的・統一的な記述方式を設定し、それによる試作・評価を行う必要がある。

- 判例理論の発見とその適用方式の検討

将来の裁判を予測する方法には、既存の判例の類推適用のほかに、判例理論の発見とその適用という方法もあるが[6]、これも興味ある課題である。

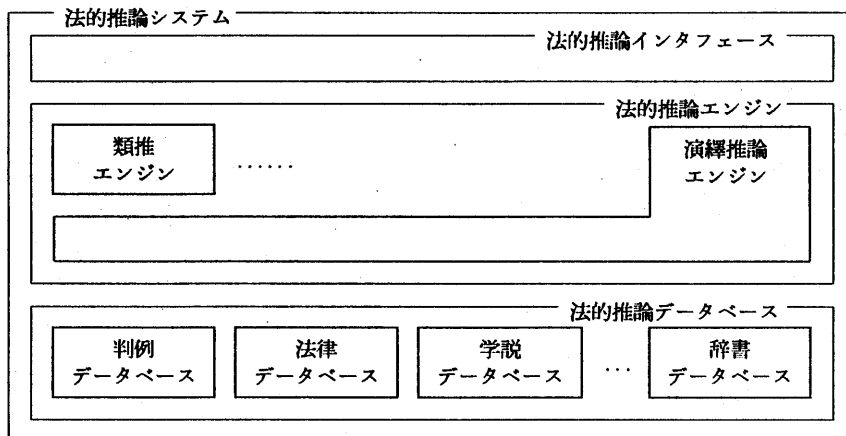


図 3: 法的推論システムの構成

謝辞

新田克己氏をはじめ、ICOT 法的推論勉強会のメンバの方からは、法的推論に関して教えていただきました。HELIC-II 関係者の方からは、判例に関する貴重なデータをいただきました。

また、横田一正氏をはじめ、QUOTE 関係者の方からは多くの示唆を与えていただきました。阿比留幸展氏からも有益なコメントをいただきました。

これらの方々に深く感謝致します。

参考文献

- [1] Hideki YASUKAWA, Kazumasa YOKOTA: Labeled Graphs as Semantics of Objects, 情報処理学会 第 80 回データベース・システム研究会, 1990
- [2] 新田克己, 星田昌紀: 事例を用いた法的推論とその並列化, 情報処理学会 第 69 回知識工学と人工知能研究会, 1990
- [3] 高橋千恵, 横田一正: 演繹データベースへの階層概念の導入, 情報処理学会 第 78 回データベース・システム研究会, 1990

- [4] 原口誠, 有川節夫: 類推の定式化とその実現, 人工知能学会誌, Vol.1, No.1, 1986
- [5] “特集 / 判例ってなんだ?”, 『法学セミナー』, 日本評論社, No.411, 1989
- [6] 中野次雄編著: 『判例とその読み方』, 有斐閣, 1986
- [7] 吉野一編著: 『法律エキスパートシステムの基礎』, ぎょうせい, 1986
- [8] 労働省労働基準局編著: 『労働基準法』(全訂版), 労務行政研究所, 1988
- [9] 労働省労働基準局編著: 『労災保険 業務災害及び通勤災害認定の理論と実際』, 労働法令協会, 1984