

著作権に関する法的推論システム  
Expert System of Copy-right Law

池田光生 田中穂積 (東京工業大学 工学部)

著者らの研究室で開発した知識表現形式、DCKRを応用した著作権法に関する推論システムについて述べ、DCKRの有効性を評価した。それによれば推論の中核をなす部分のほぼ全てをPrologに組み込みの機能で代用することができる。

1. はじめに

エキスパートシステムを構築する場合、知識の表現形式をどのようにするか、また、その知識をどのように用いて問題解決を行うかが大きな問題となる。本研究では知識表現形式として、筆者らの研究室で開発したDCKR(Definite Clause Knowledge Representation)を用いて著作権に関する法的推論システムを作成しその有効性について検討している。DCKRでは知識をPrologのホーン節で表わし、知識の継承、if-needed型の知識表現や、それらを用いた推論をPrologに組み込みの機能を用いて扱うことができる。知識は、Prologプログラムで書かれているため、それに対する推論を行なうためのインタープリタを別に作成する必要がないだけでなく、知識を直接実行できるため効率の点でも改善されている。

[小山 85] 次章以降では、まず著作権法における概念の階層構造を示しながらDCKRの基本メカニズムについて説明する。次に、本システムの概要を示し、その基本メカニズムとその知識の表現法を具体的に説明し、さらに今後の問題について述べる。

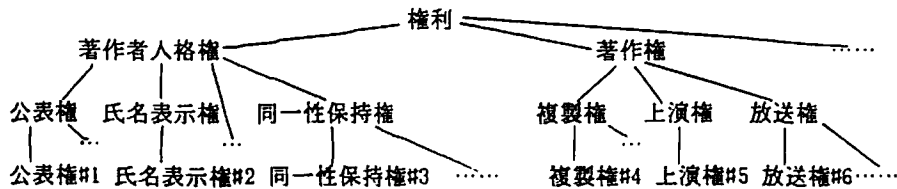
2. 著作権法における概念の階層構造とDCKR

著作権法で用いる概念には、複製権、公表権などの権利に関するもの、複製、発行、譲渡などの行為に関するもの、あるいは本、レコード、映画などの著作物に関するものなどがある。このうち著作権に関係する概念の階層構造を図1に示す。図1は、公表権、氏名表示権、同一性保持権は著作者人格権であり、複製権、上演権、放送権などは、著作権であり、著作者人格権と著作権は権利であるということを示している。

これはDCKRによって容易に記述することができる。DCKRの基本述語semは次のような形式をしている。

sem( Unitname, Slotname:Filler)

例えば sem(複製権#1, owner:池田#2)は、複製権#1の持ち主が池田#2であることを表している。ここで複製権#1のように'#'を持つものは'個体(individual)'を表す。'#'を持たないものは'プロトタイプ(prototype)'を表す。プロトタイプ複製権は、集合'複製権'に属する個体に共通した性質を持つ典型的な(抽象的な)複製権を表すものである。図1の階層構造の一部をDCKRを用いて表現したものを図2に示す。



<図1>

- (a) `sem(公表権#1, Prop) :-`  
`isa(公表権 ^ 公表権#1, Prop).`
- (b) `sem(公表権^R, Prop) :-`  
`isa(著作者人格権^R, Prop).`  
`sem(同一性保持権^R, Prop) :-`  
`isa(著作者人格権^R, Prop).`
- (c) `sem(氏名表示権^R, owner:N) :-`  
`sem(N, 著作物:O),`  
`sem(O, 複製物:C),`  
`asserta(sem(R, object:C)),`  
`isa(著作者人格権^R, Prop).`
- (d) `sem(著作者人格権^R, owner:N) :-`  
`sem(N, 著作物:O),`  
`asserta(sem(R, object:O)),`  
`asserta(sem(R, owner:N)).`
- (e) `isa(Upper^N, Prop) :-`  
`(e1)       PROP=isa:Upper ;`  
`(e2)       sem(Upper^N, Prop).`

<図 2>

階層構造における、知識の継承は、(e)に示す isa 述語を用いて行う。

知識の表現形式としてフレーム形式が使われる。これは幾つかのスロットの集合からなる structured object であるとみなせる。DCKR による知識表現では、一つの節が、一つのスロットに対応させる。従ってヘッドにある sem 述語の第一引数が同一の節の集合が一つの structured object (フレーム) を表すと考えることができる。

さて、すべての著作権に共通の手続きは著作権フレームに記述しておき、知識の継承を用いて、著作権フレームの下位の複製権、公表権フレームからそれらを利用することは、以下のようにして行われる。次のようなゴールの実行を考えてみよう。

?- sem(公表権#1, owner: 池田#2).

まず、(a) によってプロトタイプの公表権が呼ばれる。その際、個体である公表権#1は isa 述語の (e2) に書かれた sem(Upper^N, Prop) によって、上位へ運ばれる。上位のフレームにおいて何かの手続きを行なう場合に、手続きの方法はわかっていても、実行時に必要となる情報は、個体のフレームに書かれている。このようにして、上位のプロトタイプである (公表権(b)) へ移行する際に呼び出しもとの情報を持っていく。(b) のプロトタイプ公

権には、著作者人格権の持ち主が決まったときに行なわれる手続きが記述されている。ここでは、権利の対象となる著作物を調べて、権利の持ち主が池田#2であるという事実とともに、データベースに記入(assert)する。

ここで、(c) の知識について説明しておく。(c) の氏名表示権は公表権のように著作物の原本を対象とするものと違い、その複製物に対する手続きという氏名表示権に固有の手続きが必要になる。これも、(c) のような記述によって実現される。

次章では、具体的なルールの記述を示しながらシステムの概要について述べる。

### 3. システムの概要

DCKR の知識はすべて sem 述語を用いて書かれている。このことによって、概念間の階層関係が自然に実現され、さらに知識自体がプログラムであるため interpreter が要らないという利点が生じるが、カットオペレータの使用は危険を伴う、ループに陥る恐れがある、などの問題もある。このように sem の制御には注意が必要なのであるが、ここでは、sem の制御は sem 自身にさせる、という発想でシステムを作製する。

著作権における概念の中で主たるものは、'著作者'、'著作者の権利'、'著作物'、'行為'である。事実の記述の大部分は、'著作者は著作物について著作者の権利を持つ' ということや、'著作者はある行為をする権利がある'、といったものである。これらの主要なフレーム間の関係を表すのに次のようなスロットを用意した。

- sem(著作者, right: 著作者の権利)
- sem(著作者, actDid: 行為)
- sem(著作者, work: 著作物)
- sem(著作者の権利, object: 著作物)
- sem(著作物, actDone: 行為)

著作権法では、これらの四つのプロトタイプ間の関係を考えることが重要である。実際、著作権法の条文は、「...する権利を専有する」「...のとき著作物は...されたものとする」「...をすることができる」などの記述が多い。これらのプロトタイプのフレームの例を図 3 に示す。

sem<著作者#1,  
sem<著作者#1,r1,  
sem<著作者#1,actDia.

sem<複製権#2,owner:著作者#1),  
sem<複製権#2,object:著作物#5).

sem<複製#4,orig:著作物#5),  
sem<複製#4,agent:著作者#1),  
sem<複製#4,object:複製物#6).

### <図3>

図3は著作者#1が複製権#2と公表権#3をも  
っていて、複製#4を行ったことを示している。

このシステムは、大きく四つの部分に分か  
れる。その第一は、if-then型のルールベ  
ースである。この部分の記述はDCKR、すなわち  
sem述語によって書かれており、また一つの  
ホーン節が、一つの実際の法律の条文もしく  
はその一部と対応している。第3条第1項の  
例を図4に示す。

#### 第3条第1項 著作物の発行

著作物は、その性質に応じ公衆の要求を満  
たすことができる相当程度の部数の複製物  
が、第21条に規定する権利を有する者又は  
その許諾を得たもの若しくは第79条の出版  
権の設定を受けた者によって作成され、頒  
布された場合において発行されたものとする。

#### 上記のDCKRによる表現

```
sem<著作物^N,actDone:発行#_) :-  
  (sem(N,actDone:複製#P);  
   ask(N,actDone:複製#P) ),  
  sem<複製#P,agent:A),  
  sem<複製#P,条件:公衆の要求を満たす),  
  sem(N,actDone:頒布#Q),  
  sem<頒布#Q,agent:A),  
  sem(B,right:複製権#_),  
  ( A=B;  
    sem(B,actDid:許諾#R),  
    sem<許諾#R,beneficiary:A);  
    sem(B,actDid:出版権の設定#S),  
    sem<出版権の設定#S,beneficiary:A)),  
  action<発行,agent:N).
```

### <図4>

図4のホーン節はまず、ある著作物Nが発行  
されたかどうか確かめるために、ある具体的  
な特定の行為である、複製#P、頒布#Qなどの  
フレームを検索する。複製#Pがデータベース

に記述されていないならば、次のようにして複製  
をチェックする。

1. sem(N,actDone:複製#P)のとき
  - 1.1 データベースを見にいき、sem(N,actDo  
ne:複製#P)という事実があるか調べる。
  - 1.2 なければ、Nの上位でsem<著作者^N,act  
Done:複製#P)を帰結としてもつ、ルール  
を適用する。
2. ask(B)のとき
  - 2.1 以前にユーザーに問いかけたことがあ  
り、ユーザーの答が残っていたらそれを  
利用する。
  - 2.2 そうでなければ、ユーザーに質問する。  
ユーザーの答は、同じ質問を繰り返さな  
いために残しておく。

このようにして、すべての条件が満たされ  
たならば、著作物Nが発行されたことがわか  
ったのでaction実行する。actionは、新たな  
個体'発行#T'を作り出して次に述べるシス  
テムの第二の部分起動する。

第二の部分には、著作権法に現われる概念  
が、階層構造によって表現されている。スロ  
ットに対応するホーン節のボディには、フィ  
ラーに対する制約が書かれている。第一と第  
二の部分のルールは、協調して問題を解決す  
る。第一の部分は、仮説のための条件のチェ  
ック、第二の部分はそれに対するアクション  
であると解釈できる。第二の部分での'複製'  
の記述を図5に示す。

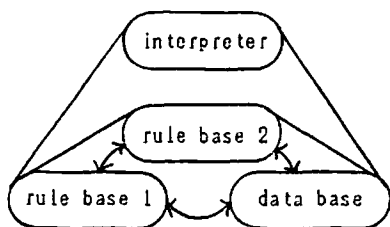
```
(a)sem<複製^A,orig:0) :-  
  asserta(sem(A,orig:0)),  
(b) ask(A,agent:X),  
(c) sem(X,right:複製権#_),  
  new_indiv<複製物#P),  
  isa<行為^A,action:(0,X,複製物#P))
```

### <図5>

第三の部分は、データベース、すなわち、  
事実や中間仮説を書き込んでおくための黒板  
である。この部分もsem述語で表現されてい  
る。この部分は、実際に発生したある特定の  
概念の記述である。以上の三つの部分は、互  
いに参照しあい、呼びあって一つの環を形成  
している。

第四の部分は、推論エンジンとなるinterp  
reter、それにユーザーインタフェース、ユ  
ーティリティなどである。interpreterの中  
核はprolog組み込みの機能で代用できるため、  
interpreterとして作成したプログラムは高  
々20行程度である。これは、三つのsemから  
なる環よりも一段上のレベルにあってルール  
ベース、データベースを支配する。以上の、

四つの部分の関係を図6に示す。



<図6>

ここで、ルールの適用順序と適用回数について考えてみる。図5で、もしかすると(c)が失敗するかもしれない。(c)に失敗するとバックトラックが行なわれ、(a)は失敗したことになる。しかし、他のルールを適用してデータベースの内容が変われば(c)が成功するかもしれない。最後まで失敗に終わったなら、それはユーザーが法に反することをを行ったか、システムの不備のいずれかである。失敗の原因をユーザーに教えるなどの処理が必要になる。残念ながら、そういった処理を行なうまでには至っていない。ルールの適用の順序と回数については、さらに深い追及が必要であろう。現段階ではif-then型のルール(条文)は一度成功したら、二度と適用されないという方針を採っている。

さて、エキスパートシステム持つべき機能に説明機能がある。なぜそのような結果が出たのか、どういう過程を経たのかわからなければならぬ。そこで、ルールに番号(条文の番号)をつけて、その履歴を取ることにした。履歴は、差分リストの考え方を使い、semからsemへ常に手渡される。

### 3. まとめと今後の課題

知識表現形式DCKRを応用した著作権法に関する推論システムについて述べた。DCKRの特徴をいかし、知識の継承をスムーズに行い、実行効率のよい推論システムを構築することができた。DCKRにより、著作権法に関する知識を見通しよく記述可能なことを示した。

現在のシステムは、知識そのものが大変小さくあまりにも簡単な問題しか扱えないので、もっと多くの条文を入れる必要がある。ルール、概念の数が増えるに従って新たな問題が発生するだろう。

そのひとつとして、時間の問題がある。現在は時間については全く考えていない。著作権法は比較的時間を考えずに済むことが多いのだが、それでも問題が複雑になるにつれ出

来事の前後関係を考えなくてはならないことがある。実現の方法としては1個のフレーム全体を時間軸につないでやり、あるフレームの成立する時間を決めてやるという方法が考えられる。[新田 他 85] フレームごとに時間が決まるのでスロット間で時間が違う場合は扱いが困難であるが、多くの問題はこれで解決できる。

次に、条文に現われない知識がある。判例、慣例、常識、他の法律などである。レコード、ソフトウェアの扱いや、各種の契約などはほとんど他の法律へ広がってしまう。できるだけ閉じたものを扱う、ユーザーの判断に依存するなどの対策が必要である。

以上のようなことを踏まえて、DCKRをベースにしてより複雑な問題に対処できるシステムを作りたいと考えている。

### 謝辞

研究のきっかけを与えてくださり、数々の有益な助言をしていただいた明治学院大学の吉野一教授、及び法律エキスパートシステム研究会の皆様にご感謝いたします。また、日頃協力していただいた東工大田中研究室の皆様にも感謝いたします。

### 参考文献

- [小山 85] Definite Clause Knowledge Representation -Prologによるstructured objectの表現形式と推論- Proc. of ロジックプログラミングコンファレンス '85
- [新田 他 85] 工業所有権法エキスパートシステムの事例問題解決機能 ibid.
- [吉野 84] 法の論理構造分析と実験的システム作成による法適用への電算機応用の可能性の検討 -民法における売買契約の成立と効力の領域に限定して- 昭和58年度科学研究費補助金研究成果報告書
- [Hayes-Roth 他 85] エキスパートシステム AIUEO訳 産業図書 (1985)
- [Hayes, P. J. 80] The Logic of Frame, in Metzger, D. ed: Frame Conceptions and Text Understanding, Walter de Gruyter, Berlin, New York, (1980), 46-61
- [Clark, K. L. 他 82] PROLOG: a language for implementing expert systems, Machine Intelligence vol. 10, pp. 455-476, Ellis Horwood (1980)
- [嶋谷 84] ソフトウェアと著作権法 講談社 (1984)
- [半田 85] 著作権のノウハウ 改訂版 有斐閣選書(1985)
- [中島 85] 知識表現とProlog/KR 産業図書 (1985)
- [山本 84] Prologによるプロダクションシステムに関する一考察 Proc. of ソフトウェア学会第一回大会 (1984)