

複合述語論理式 (CPF) による法的知識の表現 S6-1

Representation of Legal Knowledge by Compound Predicate Formulas(CPF)

吉野 一

Yoshino Hajime

明治学院大学法学部

Meiji Gakuin University*

The essential points in developing any method to represent legal knowledge are : (1) the method is easy for lawyers to understand and use, (2) it has the sufficient ability to express legal knowledge in detail, and (3) it is applicable to formalizing legal reasoning. For these three reasons I have designed Compound Predicate Formulas (CPF), which is a conservative extension of the first order predicate logic, as a method of the representation of legal knowledge and developed Legal Expert Systems (LES-2 and LES-3) on its basis. In this paper I explain the method, present some examples, and also give the rigorous logical foundation of CPF i.e. the establishment of its syntax and semantics.

1 はじめに

法的知識の表現方法にとって重要な要素は、第一に、法律家に理解しやすく書きやすいものであること、第二に法の実際に対応した詳細な表現力があること、そして第三に法的推論の形式化に適合していることである。筆者は、この三つの理由から複合的述語論理式(CPF)を提案し、用いてきた。そして、これをもとにして、法律エキスパートシステム(LES-2およびLES3)を発展させた¹。本稿はその表現方法を実例を用いて説明するとともに、その論理学的基礎付けを行なう、すなわち、そのシンタックスとセマンティックスを確立する。

2 なぜ CPF か

われわれがCPFを導入した理由は、一言で言うと、法的知識を適切で分かりやすい形に表現するためである。法律の中でよく使われる文を例にとって考えてみよう。

例: 太郎が花子に申込をして、それが承諾された。これを従来の一階の言語で表現しようとする

般にはうまくいかない。それは、まさに「それが」という部分が表現しようとする対象にある。上の文で、「太郎が花子に申込をする」を「offer(太郎, 花子)」で記号化するとすれば、「それが承諾された」はどのように記号化されようであろうか。指示代名詞「それ」が指すものは、太郎の花子に対する申込である。しかし、従来の一階の言語で名指すことができるのは、個物だけであって、太郎の花子に対する申込という事態を指すことができない。われわれが従来の一階の言語の内部にとどまる限り、上の例文を一つの述語「 $p(x_1, x_2)$ 」を使って「 $p(\text{太郎}, \text{花子})$ 」で表現するしかないであろう。しかし、これでは、この例文が持つ内部構造が適切に表現できない。これは法律的推論の知識表現にとって致命的である。従来の一階の言語には「あの申込」や「1994.2.3.15:00に札幌で行なわれた売買」のような個々の法的行為に言及するための装置が存在しない。さらに、「あの申込」という表現に出てくるような指示代名詞の表現に相当するような言語的装置が従来の一階の言語には存在しない。しかし、法律の言語を考える場合には、これらの表現を扱えることが当然必要になってくる。

*108 東京都 港区白金台 1-2-37; e-mail: hyoshino@tansei.cc.u-tokyo.ac.jp; TEL: 03-5421-5310; FAX: 03-5421-7317.

¹LES-2におけるCPFの利用については、参照:[6]。LES-3については、参照:[7]。

3 いくつかの先行アプローチ

今述べた目的のために通常の述語論理の言語でこれまで行われてきた工夫としては、例えばラムダオペレータを用いた表記法、クラスによる表記法等が考えられるであろう。以下その方法について簡単に述べ、その欠点を指摘する。

3.1 λ 記号による方法

例えば、「XがYに申込み」を「offer(X,Y)」で記号化すると、申込みという関係は、 λ 表現を使って、下のように書き表わせる。

$$\lambda X \lambda Y (offer(X,Y))$$

一般に、 λ 演算子は与えられた述語からその述語が表わす概念に言及する表現を形成する。しかし、これでは「あの時、太郎と花子の間に取り交わされた申込み」のような個々の申込に言及することができない。われわれが求めているのは(法律的な)関係の具体的な事例(インスタンス)の名前であって、抽象的な概念の名前ではない。

3.2 クラスによる表現

それでは、クラスで表現したらどうであろうか。この試みは、関係をクラスによって表現し、個々の事例をその要素という形で表わそうというものである。例えば、「ある申込が承諾された」という文を記号化してみると、

$$A = \{X : accept(X)\}$$

$O = \{Z : \exists X \exists Y (offer(X,Y) \wedge Z = \langle X,Y \rangle)\}$ とくと、A、Oはそれぞれ、承諾という概念、申込みという概念(の外延)を表わす。そこで、

$$\exists Z (Z \in O \wedge Z \in A)$$

によって、上の文の記号化を与える。これは、最終的には次のように書き換えられる。

$$(1) \exists Z \exists X \exists Y (offer(X,Y))$$

$$\wedge Z = \langle X,Y \rangle \wedge accept(Z))$$

しかし、この方法では、「申込の承諾」というような表現を構成するのが大変困難である。いずれにしても、上の方法は、例えば、「申込」という概念そのものを指示するには適しているが、個々の申込を指示するには適しているとはいえない。

4 ID-記号の導入

「XがYに申込して、その申込が承諾された」という例文で、その代名詞が指すものを表現するには、クラスで表現することはほとんど不可能である。し

かし、自然言語での指示代名詞の役割を果たすような表記法をいれるとこれを簡単に表現することが出来る。それがID-記号である。複合述語論理式の特徴はID-記号の導入にある。

そこで、従来の一階の言語の限界を越えて、個々の(法律的)行為に言及するような表現を以下に述べるような仕方で導入する。

offer(X,Y): XはYに対する申込である。

acceptance(W,X): WはXの承諾である。

とすると、「XのYに対する申込が承諾された」は次のように記号化される。

$$(2) \exists Z (offer(X,Y,Z) \wedge acceptance(Z,W))$$

(1)と(2)を比較すると、(2)の方がより適切で簡明である。これは個別的な関係や行為に言及する表現を導入したからである。さらに進んで、われわれは個々の申込や承諾により明確に言及するような表現ID-記号を導入する。一般に、述語 $p(X_1, \dots, X_n)$ に対して、

$$ID_p(X_1, \dots, X_n)$$

は当の述語のID-記号である。より具体的に言うと、例えば、述語offer(X,Y)に対して、

$$ID_offer(X,Y)$$

はX,Y間の申込を表わす。別の言い方をすれば、それは当の申込の名前である。(2)はID-記号を使ってこのように簡潔に書ける。

$$(3) acceptance(ID_acceptance, ID_offer)$$

ID-記号のこの名詞化 nominalization の機能は特に強調されねばならない。ラムダ演算子やクラスによる表記法が、ある概念そのものの名前を形成するのに対し、ID-記号のほうは、ある概念の事例の名前を形成する。例えば、

例: 「あの申込の承諾」

これは、ID-記号を使って

$$(4) ID_acceptance(-, ID_offer(-, -))$$

と書き表わせる。ID-記号においてはこのような複雑な名詞化も可能である。

4.1 CPF のシンタックス

これから、CPF のシンタックスの定義を行なう。大部分従来の一階の言語と同じであるが、CPF の言語が従来のそれと大きく異なるのは、格記号²やID-記号を導入した点である。CPF のシンタックスは以下の通りである:

1. アルファベット:

²格記号は述語の内部構造を明確に表現するために考案された表記法である。格記号の構文論的および意味論的な身分については別の機会に論じることとする。

1.1. 個体変項: $X_1, X_2, \dots, T_1, T_2, \dots$

1.2. 個体定項: $a_1, a_2, \dots$

1.3. 格 case 記号: $agl: obj: goa: tim: \dots$

1.4. 述語記号: $p_1, p_2, \dots$

1.5. リスト記号: $[]$

1.6. 論理定項: $\neg, \wedge, \vee$

1.7. 補助記号: $(), (コンマ)$

2. 言語 CPF の項と式の定義:

2.1 個体定項と個体変項と ID-記号は項である。

2.2 l が個体定項か個体変項か ID-記号であれば、 $agl: l, obj: l, tim: l, goa: l$ は項である。($c_1, c_2, \dots$ で格記号を表わすことにする。)

2.3. $[l_1, \dots, l_n] (l_i (1 \leq i \leq n) \text{ は項})$ はリストである。

2.4. 述語 p について、 $p([l_1, \dots, l_n])$ は式である。

2.5. A, B が式であれば $\neg A, A \wedge B$ も式である。

2.6. $A(X)$ が式であれば $\forall X A(X)$ も式である。

2.7. 述語 ID 記号の定義: 述語 $p([l_1, \dots, l_n])$ について、 $ID_p([l_1, \dots, l_n])$ は当の述語の ID-記号である³。

2.8. 式 $p([l_1, \dots, l_n])$ について、

$$p(ID_p, [l_1, \dots, l_n]) \text{ も式である}^4。$$

2.9 項と式は 2.1-2.8 でできるすべてである。

他の論理定項は、定義によって導入する:

$$A \wedge B \triangleq \neg(\neg A \vee \neg B)$$

$$A \vee B \triangleq \neg A \wedge \neg B$$

$$\exists X A \triangleq \neg(\forall X \neg A)$$

4.2 CPF による法的知識表現

ここでは法律の条文を一つ例にとって、それが CPF の言語にどのように翻訳されるかを具体的に見ていくことにする。

CISG 23 条: A contract is concluded when an acceptance of an offer becomes effective.

邦訳: 契約は、申込に対する承諾が効力を生じたときに成立する。

まず、上の条文に現われる述語の一つ一つを CPF の述語に翻訳する。

1. contract($ID_co, [agl: [X,Y], obj: C]$): X と Y の間に C という内容の契約が成り立つ

2. acceptance($ID_ac, [agl: X, obj: ID_of, goa: Y]$): X が Y に対して ID_of を承諾する

3. offer($ID_of, [agl: X, goa: Y, obj: C]$): X が

³誤解が生じない限り、ID-記号の中の引き数は適宜省略することにする。またわれわれは派生的に、述語記号 p についても ID-記号 ID_p を定義しておく。

⁴前者の形の式を ID-記号抜ききの式、後者の形の式を ID-記号付ききの式と呼び、区別して考えるがそれらには内容上の相違はない。

Y に対して C という内容の申込を行なう

4. be - concluded($ID_be, [obj: ID_co, tim: T]$): ID_co が時点 c T において成立する

5. become - effective($ID_be, [obj: ID_ac, tim: T]$): ID_ac が時点 T において効力を生じる

これに基づいて、上の条文は下の形の式に翻訳される。

$$be - concluded(ID_be, [obj: ID_co, tim: T_1]) \wedge contract(ID_co, [agl: [X,Y], obj: C]) \rightarrow$$

$$become - effective(ID_be, [obj: ID_ac, tim: T_1])$$

$$\wedge acceptance(ID_ac, [obj: ID_of])$$

$$\wedge offer(ID_of, [agl: X, goa: Y, obj: C])$$

このような形の式をフラット化された CPF の式 (FCPF) と呼び、これは下記の形の式と等値である:

$$be - concluded(ID_be, [obj: contract(ID_co, [agl: [X,Y], obj: C], tim: T_1)]) \rightarrow$$

$$become - effective(ID_be, [obj: acceptance(ID_ac, [agl: X, goa: Y, obj: offer(ID_of, [agl: X, goa: Y, obj: C])], tim: T_1)])$$

この式は上記 FCPF の短縮表記である。知識ベース

にはこの形で記述されて入力される。推論実行のためには、上記 FCPF にコンパイル (フラット化) される⁵。

CISG 23 条、(そしてそれを適用した際に生じる法的推論) を従来の標準的な一階の言語で形式化しようとしたら困難に直面するであろう。その困難は「その申込」、「その承諾」そして「その契約」を指示する道具がそこには存在しないからである。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

推論規則に関して言えば、われわれは一階の論理の推論規則を拡張しないで済むことができる。CPF の言語は、一階の論理の保存的拡張にとどまっている。

述べる。

Definition 5.1 $M = \langle D (= D_1 \cup D_2 \cup \{\perp\}), I \rangle$ は割り当て $g (= g_1 \cup g_2)$ に関する CPFL のモデルである。 $\iff$

1. D_1 、 D_2 はそれぞれ個体の集合、時間の集合である。また、 $D_1 \cap D_2 = \emptyset$ と規定しておく。 $\perp$ は $\perp \in D_1 \cup D_2$ であるようななんらかの対象である。
2. $g_1: INDVAR \rightarrow D_1$ かつ $g_2: TIMVAR \rightarrow D_2$ 。
3. I が個体定項のとき、 $I(t) \in D_1$ 。
4. n 項述語 p について、 $I(p) \subseteq D^n$ ここで、 D^n はデカルト積である。
5. (ID-記号の解釈) 述語記号 $p(t_1, \dots, t_n)$ が与えられたとき、その ID-記号の解釈 $I_g(ID_p(t_1, \dots, t_n))$ は以下のように ID-記号のレベル⁷ に関する帰納法で定義される⁸。 $I(ID_p)$ は D 上に定義される関数である。すなわち、任意の n -項述語 p 、任意の $e_1, \dots, e_n \in D$ について、 $I(ID_p)(\langle e_1, \dots, e_n \rangle) \in D$ 。そこで $LEVEL(ID_p(t_1, \dots, t_n)) = n \geq 2$ であり、レベル $m (m < n)$ の ID-記号に対して解釈が定義されているとすると、 $I_g(ID_p(t_1, \dots, t_n))$ は次のように規定される。
 1. もし、 $(I(ID_p)(\langle I_g(t_1), \dots, I_g(t_n) \rangle)) = a$ であれば、
 $I_g(ID_p(t_1, \dots, t_n)) = a$ とおく。
 2. 値が定義されていないければ、
 $I_g(ID_p(t_1, \dots, t_n)) = \perp$ とおく⁹。

6 結論

これまで、CPF について、そのシンタックスとセマンティクスを中心に、詳細に述べてきたが、最後に、CPF の導入、特に ID-記号の導入のメリットについて簡単にまとめておくことにする。

1. 表現力が豊富になる。契約、売買等の個々の法的行為に言及することができる。
2. 自然言語により近い形での法的知識表現が可能になる。

このように ID-記号の導入のメリットには大きなものがある。何よりも、CPF に対して論理学的基礎付

⁷ INDVAR, TIMVAR はそれぞれ個体変項、時間変項全体の集合を表す。

⁸ ID-記号 $ID_p(t_1, \dots, t_n)$ が与えられたとき、 $ID_p(t_1, \dots, t_n)$ の中の ID-記号全体の個数をその ID-記号のレベルと言い、 $LEVEL(ID_p(t_1, \dots, t_n))$ で表す。

⁹ I_g は、変項 x に対しては $I_g(x) = g(x)$ 、その他の種類の項 t に対しては $I_g(t) = I(t)$ を与えるような関数である。

⁹ 式の充足は従来の通りにして定義できるので、ここではその定義は省略する。

けを与えたことが本論文の最大の意義であると、いえよう。

参考文献

- [1] Ebbinghaus, H.D., J. Flum and W. Thomas, *Mathematical Logic*, Springer, 1984.
- [2] Gupta, A., *The Logic of Common Noun*, Yale University Press, 1980.
- [3] Sakurai, Seichiro and Yoshino, Hajime, Identification of Implicit Legal Requirements with Legal Abstract Knowledge, in: *Proc. The Fourth International Conference on Artificial Intelligence and Law*, ACM (The Association for Computing Machinery), 1993, pp. 298-305.
- [4] Yoshino, Hajime, Über die Notwendigkeit einer besonderen Normenlogik als Methode der juristischen Logik, in: Klug, U. et al. (Hrsg.), *Gesetzgebungstheorie, Juristische Logik, Zivil- und Prozeßrecht (Gedächtnisschrift für Jürgen Rödig)*, Springer-Verlag Berlin Heidelberg, 1978, S. 140-161.
- [5] Yoshino, Hajime, Zur Anwendbarkeit der Regeln der Logik auf Rechtsnormen, in: *Die Reine Rechtslehre in wissenschaftlicher Diskussion (Schriftenreihe des Hans Kelsen-Instituts Band 7)*, Wien (Manz-Verlag), S. 142-164.
- [6] Yoshino, Hajime, et al., Legal Expert System Les-2, in: Wada, E. (Ed.), *Logic Programming '86 (Lecture Notes in Computer Science 264)*, 1987, p. 36ff.
- [7] Yoshino, Hajime, *Research Report on Explication of Legal Knowledge Structure and Development of Legal Knowledge-base*, Association for Machine System Promotion, 1990, pp. 27-32, pp. 41-55 (in Japanese).
- [8] Yoshino, Hajime et al., Towards a Legal Analogical Reasoning System Knowledge Representation and Reasoning Methods, in: *Proc. The Fourth International Conference on Artificial Intelligence and Law*, ACM (Association for Computing Machinery), 1993, pp. 110-116.