Artificial Intelligence and Law **5:** 77–96, 1997.
© 1997 *Kluwer Academic Publishers. Printed in the Netherlands.*

On the Logical Foundations of Compound Predicate Formulae for Legal Knowledge Representation

HAJIME YOSHINO
Meiji Gakuin University, 1-2-37, Shirokanedai, Minato-ku, Tokyo, Japan
E-mail: yoshino@mh.meijigakuin.ac.jp

Abstract. In order to represent legal knowledge adequately, it is vital to create a formal device that can freely construct an individual concept directly from a predicate expression. For this purpose, a Compound Predicate Formula (CPF) is formulated for use in legal expert systems. In this paper, we will attempt to explain the nature of CPFs by rigorous logical foundation, i.e., establishing their syntax and semantics precisely through the use of appropriate examples. We note the advantages of our system over other such systems and discuss the significance of CPFs with regard to the formalization of legal reasonings using examples from the United Nations Convention for the International Sale of Goods.

Key words: legal reasoning, CISG, knowledge representation, logic, compound predicate formula

1. Introduction

The most important factors in developing any method to represent legal knowledge are: (1) ease for lawyers to understand and use the method; (2) the ability to express legal knowledge in detail; and (3) its applicability to formalizing legal reasoning. In order to obtain these three objectives, we have developed the Compound Predicate Formulas (CPFs). The CPFs, a conservative extension of first order predicate logic, are devices used to represent legal knowledge in our efforts to develop Legal Expert Systems (e.g., LES-2,[⋆] LES-3,[⋆⋆] and LES-4[‡]).

Given the goal of formalizing ordinary knowledge as expressed in a natural language, CPFs and sorted logic languages seem to be incomplete sublanguages of a Montague Grammar, which has been used in attempts to formalize our natural language. But for computational efficiency, CPFs and sorted logic work well, unlike a Montague Grammar. CPFs and sorted logic are very similar in their structures. However, sorted logic implies a strong metaphysical claim that sorts and predicates belong to different ontological categories. If we look at many examples of legal reasoning, we find that it is meaningless to distinguish such metaphysical differences. Just as we use nouns and predicates without any such metaphysical

[⋆] (Yoshino, 1987) and (Yoshino 1988).
[⋆⋆] (Yoshino and Kakuta 1992).
[‡] This system is now under construction.

constraints, we do not have to express predicates and sorts separately. Therefore, it may be useful to construct a system in which we can freely use individual concepts derived from a predicate expression.

McCarty invented a device which, at first glance, is similar to our ID-symbols, e.g. (McCarty, 1989), but the author is the first to introduce a method to refer to a particular instance of a given individual concept within a knowledge representation system. Thus, it is necessary to explain in this paper our method and its advantages over others.

In this paper, I will explain the methodology of CPF by use of various examples establishing its syntax and semantics, thus providing it with a rigorous logical foundation.

2. Why Use a CPF?

In this section let me explain why we have chosen the CPFs, as opposed to other possible methodologies. In a nutshell, I chose the CPFs in order to represent legal knowledge adequately and in a simplistic fashion. In order to clarify this point, we will use a simple example (i.e., “a legal sentence”) which will illustrate the difficulty of representing such a sentence by standard first order logic. Consider the following:

EXAMPLE. John made an offer to Mary and it was accepted.

It is difficult to express the whole sentence in standard first order language, for the standard first order language does not contain any device for representing the referential expression “it”. We can symbolize “John made an offer to Mary” in the above sentence, as: “offer (John, Mary)”, but how can we symbolize “it was accepted?” We all agree that the referential pronoun “it”, which is part of the above sentence, refers to “an offer of John to Mary”. Unfortunately, first order language has the ability to refer only to individual entities and not to any state of affairs, such as an offer made by John to Mary. If we are to use standard first order language, we must content ourselves with symbolizing the above example as a predicate $p(X_1, X_2)$, thus:

$$\mathit{offer}(\mathit{John}, \mathit{Mary}).$$

Even though such an approach will result in a more useful way to deal with the problem, such symbolizing does not adequately reflect the inner structure of the sentence. This fact implies that the standard first order language is not robust enough to adequately represent legal sentences, and therefore to describe legal reasoning.

In short, the standard first order language lacks the means to refer to each legal act, which involves “that contract” or “the trade at 15:00 on Feb. 3rd, 1994”. Moreover, the standard language has no device to represent referential pronouns like the word "it". What we really need is a more robust language that enables us to deal with these types of expressions.

3. Some Precedent Approaches

One might attempt to express various legal sentences using class notation by allowing relations to correspond to classes and each instance to its member. For example:

$$A = \{X : \textit{acceptance}(X)\},$$

$$O = \{Z : \exists X \exists Y(\textit{offer}(X,Y) \wedge Z = \langle X,Y \rangle\},$$

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

A and O show (the extension of) the concept of "acceptance" and the concept of an "offer", respectively. Below, (1) is derived as the translation of the legal sentence: "X made an offer to Y and it was accepted":

$$\exists Z(\textit{offer}(X,Y) \wedge Z = \langle X,Y \rangle \wedge \textit{acceptance}(Z)). \tag{1}$$

As (1) shows, the class notation is adequate for denoting a concept by itself. However, this method is clearly not satisfactory for denoting its particular instance. Indeed, each instance is a certain element of a given class, as already noted. Many expressions in natural languages, however, involve ample pragmatic, i.e. contextual information, and it is hard to specify a purposed set, which can handle much contextual information. Even though the specification is acquired, the formula is often too complex to understand. Also, the class notation does not have any type of apparatus for representing referential pronouns. In legal sentences and legal reasoning, one does meet with referential expressions frequently.

4. ID-Symbols: A New Device

In the previous section, we noted that there are certain problems in utilizing the standard first order language because it is inadequate for properly expressing each individual legal act as well as for describing sentences with referentials in a form which properly reflects their inner structures. We can, however, formulate a new device for coping with these problems, namely, ID-symbols, but before doing so, we need to clarify the notion from which they were derived. Let us assume that:

$\textit{offer}(Z,X,Y)$: Z *is an offer of* X *to* Y,

$\textit{acceptance}(W,Y,Z)$: W *is the acceptance of* Z *by* Y.

If we assume these formulae, then a sentence of the form "An offer of X to Y was accepted" would be formalized as follows[*]:

$$\exists Z(\textit{offer}(Z,X,Y) \wedge \textit{acceptance}(W,Y,Z)). \tag{2}$$

[*] In (Davidson, 1980), Davidson employed essentially the same idea in order to analyze the logical form of action sentences. Our approach should be considered an extension to his. In another

Compared to (1), (2) is simpler and more understandable. Example (2) contains a way for referring to a certain specified individual legal act or relation. In general, for any predicate $p(X_1, \ldots, X_n)$,

$$\mathit{id\_p}(X_1, \ldots, X_n)$$

is the ID-symbol in question. For instance, for a predicate "*contract*(*Mary*, *John*)", we use:

$$\mathit{id\_contract}(\mathit{Mary}, \mathit{John}).$$

The above notation expresses a contract between Mary and John. In other words, it is the name of that contract. Using ID-symbols, formula (2) is more concisely rewritten as:

$$\mathit{acceptance}(\mathit{id\_ac}, [Y, \mathit{id\_offer}]). \tag{3}$$

Here, I would like to emphasize that ID-symbols, a kind of nominalization, i.e. the fact that an ID-symbol forms the name of *a particular instance* in a given concept. (Recall that notations using lambda operators or classes form the name of a concept itself.) As one more example, "the rejection of that offer by Y" is, in ID-symbols, formalized as:

$$\mathit{id\_rejection}([Y, \mathit{id\_offer}]). \tag{4}$$

Using ID-symbols, we can now easily deal with such a relatively complex case.

I should note, in passing, if ID-symbols function as names, it is necessary that the obvious identity criterion for the referents of ID-symbols be given. This means that for particular instances of a concept, the condition of continuity through time must be defined. To define that condition, we must define every legal concept precisely. It should also be noted that this problem relates to law – not logic.

5. Similar Symbols

Before going into the details of the technical description of CPFs, let us consider similar devices already used in order to understand ID-symbols better. For example, Hilbert's ε-symbol and Gupta's logic of the common noun should be considered. They are syntactic devices used to build a term from a given formula. CPFs and these two systems share certain characteristic. First, let us look at the common elements they share.

5.1. HILBERT'S ε-SYMBOL

Consider Hilbert's ε-symbol. The formation rule of an ε-term is as follows:

If A is a formula, then $\varepsilon x A$ is a term.

paper (Yoshino, 1978), the author used a symbolism to refer to actions. E.g. see (Yoshino, 1978, pp. 152–153).

Intuitively, $\varepsilon x A$ denotes an unspecified object satisfying the property expressed by the formula A, if such an object exists, otherwise $\varepsilon x A$ refers to an arbitrary object in the domain.

The ε-calculus is a formal system obtained from the first-order calculus with equality by adjoining the ε-symbol to represent an additional logical constant. The following axiom is then added:

$$A(t) \longrightarrow A(\varepsilon x A),$$

where t is any term.

Originally, the ε-symbol was introduced by Hilbert and others in order to provide explicit definitions for the quantifiers $\forall$ and $\exists$. They are given by the following formulae:

$$\exists x A \longleftrightarrow A(\varepsilon x A),$$

$$\forall x A \longleftrightarrow A(\varepsilon x \neg A).$$

These are provable in the ε-calculus.

Moreover, this system has another axiom,

$$\forall x(A \leftrightarrow B) \longrightarrow \varepsilon x A = \varepsilon x B.$$

The ε-symbol is specified by the axioms only to the extent that any non-empty set has at least one representative and this representative is an element of the set. For example, if A is the formula $x = x$, then the term $\varepsilon x A$ must denote some object. However, we have no way of knowing what that object is.

It is often desirable to have within a formal theory some way of designating "the unique x *s.t. As*". The ε -symbol was introduced for just this purpose. Using the ε-symbol, we adjoin all formulae of the following forms as additional axioms of the theory, thus

$$\exists ! x A \longrightarrow A(\iota x A),$$

$$\neg \exists ! x A \longrightarrow \iota x A = t,$$

where t is some specified term of the language, such as 0. These axioms say that if there exists a unique x such that A, then $\iota x A$ designates that unique object, and if not, then it is the designated term t. That is:

$$\iota x A = \varepsilon x((\exists ! x A \wedge A) \vee (\neg \exists ! x A \wedge x = t)).$$

Now we can say the ε-symbol and the ID-symbol function essentially in the same manner in their logical representations.[⋆]

[⋆] For more information on ε-symbol, the reader should consult e.g. (Leisenring, 1969).

5.2. Common Nouns

The logic of common nouns, according to Gupta (1980), was used to analyze the logical and semantic behavior of common nouns. His main concern was how the intensions of common nouns are sorts. The common nouns are added to a language, so a formula is defined by recursion with terms and common nouns as follows:

- If K is a common noun, x is a variable, and A is a formula, then $(\forall K, x)A$ is a formula.
- If K is a common noun, x is a variable, and A is a formula, then $(K, x)A$ is a common noun.
- If K is a common noun, then ιK is a term.

Intuitively, a symbol $(K, x)A$ stands for the subsort of K each member of which satisfies A. For example, If M stands for the common noun "man", L for "likes" and m for Margaret, then "man who likes Margaret" is expressed as

$$(M, x)L(x, m).$$

On the other hand, the symbol ιK stands for the unique object that belongs to the sort K if such an object exists. For example, "the man who likes Margaret" is expressed as

$$\iota(M, x)L(x, m).$$

Hilbert's and Gupta's systems use devices to express the usual noun clauses that are similar to ours, but their motivations are very different. As previously noted, our motivation for using the ID-symbol is different from the one above. But the technical devices we adopt closely resemble each other. That is, I use similar devices but my motivation for using them is different. My motivation for using ID-symbols stems from the idea that we need to use the symbol to refer to the object which was represented previously by using a noun phrase or a sentence.

6. Outline of the Syntax of CPFs

What follows are attempts to define the syntax of CPFs. In this respect, CPFs are largely the same as the standard first order language, but CPFs differ from first order language in that CPFs utilize new devices such as case symbols and ID-symbols.[⋆] We must eventually define and describe the syntactic behavior of ID-symbols in a more complete fashion. But our concern here is restricted only to program clauses. Thus, we will think only of the quantifier-free part, the so-called "the Horn clause":

$$B \longleftarrow A_1, \dots, A_n,$$

where $B, A_1, \dots, A_n$ are literals.

The syntax of CPFs is as follows:

[⋆] Case symbols are a notation contrived to express the inner structure of predicates explicitly. On the syntactic and semantic status of case symbols, we may leave this to another occasion.

1. **Basic Vocabulary**:
1.1 **individual variables**: $X, Y, \ldots, X_1, X_2, \ldots$
1.2 **individual constants**: $a, b, \ldots, a_1, a_2, \ldots$
1.3 **case symbols**: *agt, obj, goa, tim*, ...
1.4 **predicate letters**: $p, q, \ldots, p_1, p_2, \ldots$ In addition to these we shall introduce a special two-place predicate letter $=$, i.e., the equality symbol, into our language.
1.5 **list symbols**: [,].
1.6 **ID-operator**: *ID_, id_*.[⋆]
1.7 **logical constants**: $\neg, \longleftarrow, \forall$.
1.8 **commas, parentheses:** (,),.
2. **terms and formulae**:
2.1 Variables and individual constants are terms.
2.2 If t is an individual constant, an individual variable or an ID-symbol and c is a case symbol, then $c : t$ is a term.
2.3 $[c_1 : t_1, \ldots, c_n : t_n]$ is a list.
2.4 $p([c_1 : t_1, \ldots, c_n : t_n])$ is a formula.
2.5 If A and B are formulae, then $\neg A$, $A \longleftarrow B$ are formulae.
2.6 If $A(X)$ is a formula, then $\forall X A(X)$ is a formula.
2.7 **The definition of ID-symbols**: For the predicate representing a legal concept $p([c_1 : t_1, \ldots, c_n : t_n])$, *id_p*$([c_1 : t_1, \ldots, c_n : t_n])$ is a term called ID-symbol of its predicate.[⋆⋆]
2.8 The following, $p(t, [c_1 : t_1, \ldots, c_n : t_n])$, is a formula as well where t is either a variable or an ID-symbol. The position in the formula $p(X, [t_1, \ldots, t_n])$ occupied by the variable X we shall call "the ID-position" of its formula.[‡] For the sake of convenience, we will stipulate that the ID-position of a given predicate p be occupied only by the corresponding ID-symbol *id_p* or the variables.
2.9 An expression is a formula only if it can be shown to be a formula on the basis of conditions 2.4–2.6, 2.8, above.

Other logical constants are introduced in an obvious manner.

At this juncture, I would like to make a few comments on the syntax just defined. In the above, we simply stipulate that the ID-position of a given predicate p be occupied only by the corresponding ID-symbol *id_p* or the variables. Hence, we

[⋆] We shall provide two kinds of ID-operator. We shall use *ID_p* as a variable. On the other hand, *id_p* is a function symbol.

[⋆⋆] We will omit the arguments in ID-symbols unless this leads to misunderstanding. Derivatively we will define ID-symbols about predicate symbols as well.

[‡] The former is called *the formula without ID-position*, the latter, *the formula with ID-position* as a matter of convenience. As can be plainly observed, we need a formula to assure $p([t_1, \ldots, t_n]) \longleftrightarrow p(\textit{id\_p}, [t_1, \ldots, t_n])$, if we construct an axiomatic system for a CPF.

shall admit only the following kinds of expressions:

$$p(\textit{id\_p}, [c_1 : t_1, \ldots, c_n : t_n]),$$
$$p(X, [c_1 : t_1, \ldots, c_n : t_n]).$$

On the other hand, the type of expression below cannot be allowed by our formalism:

$$p(\textit{id\_q}, [c_1 : t_1, \ldots, c_n : t_n]),$$

where p and q are predicate symbols which denote different concepts. Of course, the following type of an expression is quite possible in our formalism.

$$p([s_1, \ldots, s_n, \textit{id\_q}, t_1, \ldots, t_m]).$$

In other words, the ID-symbol *id_q* can appear in other predicates than q. Before moving on, we would like to make the following point in order to clarify the significance of CPFs here.

Since McCarty has also invented a device similar to ID-symbols (McCarty, 1989), it is necessary to mention and compare that device with our device thus emphasizing the advantages of our system to his.

As far as we know, this author is the first to introduce ID-symbols into a legal knowledge representation system. Originally, ID-symbols were used as variables ranging over the instances of the legal relations, e.g. offer, acceptance. This idea of reification of the instances of relations has guided our development of the legal expert system LES-2 since then.*

Motivated by essentially the same idea, McCarty employed similar symbolisms (McCarty, 1989).** In his paper, he used the following notation:

$$(\textit{Own}O1(\textit{Actor}A)(\textit{Property}P)).$$

This is interpreted to mean that an actor A owns a property P, and $O1$ denotes this very ownership. Thus, the symbol $O1$ roughly corresponds to our ID-symbol *ID_own*. However, such symbols seem to play a minor role in his language LLD. The use of such terms are very restrictive from a legal knowledge representation point of view.

Regardless, we must take into consideration that the name of an instance of a given relation itself can appear in argument places of the predicate denoting another relation. In light of this fact, we should freely and extensively use an ID-symbol as a term denoting an instance of a given relationship. Therefore, we allow an ID-symbol *ID_p* to appear in predicates other than p, as noted earlier. For example,

* The author reported progress on LES-2 including CPF at the Logic Programming Conference 86 on 23–26 June 1986 in Tokyo. The paper was published in (Yoshino, 1987).

** The author reported also about CPF at the Symposium on Legal System and Knowledge Representation held at Meiji Gakuin University on 27 October 1986. To the symposium Professor T. McCarty contributed as well. As the organizer of this symposium, I was very grateful for his kind contribution.

the sentence “an acceptance of an offer becomes effective” can be expressed by the following CPF:

$$\textit{become\_effective}(\textit{ID\_bea}, [\textit{obj} : \textit{acceptance}(\textit{ID\_ac},$$
$$[\textit{agt} : Y, \textit{goa} : X, \textit{obj} : \textit{offer}(\textit{ID\_of}, [\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C])]), \textit{tim} : T]).$$

On the other hand, such a representation cannot be expressed using McCarty’s formalism (McCarty, 1989; McCarty and van der Meyden, 1992). This is one advantage of our formalism over his.

7. Legal Knowledge Representation in Terms of CPFs

Having defined the syntax of CPFs, we can now state and represent legal knowledge. Consider now a rule based on an article of the United Nations Convention for the International Sale of Goods (CISG) and how it can be translated into CPFs.

Rule 23 (cf. Article 23). A contract is concluded if and only if an offer is effective and an acceptance of the offer becomes effective.

1. $\textit{contract}(\textit{ID\_co}, [\textit{agt} : [X, Y], \textit{obj} : C])$: *ID_co* is a contract concerning C between X and Y.
2. $\textit{acceptance}(\textit{ID\_ac}, [\textit{agt} : X, \textit{obj} : \textit{ID\_of}, \textit{goa} : Y])$: *ID_ac* is an acceptance of *ID_of* by X to Y.
3. $\textit{offer}(\textit{ID\_of}, [\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C])$: *ID_of* is an offer of C by X to Y.
4. $\textit{be\_concluded}(\textit{ID\_bc}, [\textit{obj} : \textit{ID\_co}, \textit{tim} : T])$: *ID_co* is concluded at time T.
5. $\textit{become\_effective}(\textit{ID\_bea}, [\textit{obj} : \textit{ID\_ac}, \textit{tim} : T])$: *ID_ac* becomes effective at time T.
6. $\textit{is\_effective}(\textit{ID\_ie}, [\textit{obj} : \textit{ID\_of}, \textit{tim} : T])$: *ID_of* is effective at time T.

Translating the CISG into a CPF below, we get

$$\textit{be\_concluded}(\textit{ID\_bc}, [\textit{obj} : \textit{ID\_co}, \textit{tim} : T])$$
$$\wedge \textit{contract}(\textit{ID\_co}, [\textit{agt:}\ [X, Y], \textit{obj} : C]) \longleftrightarrow$$
$$\textit{is\_effective}(\textit{ID\_effective ie}, [\textit{obj} : \textit{ID\_of}, \textit{tim} : T])$$
$$\wedge \textit{offer}(\textit{ID\_of}, [\textit{agt:}\ X, \textit{goa:}\ Y, \textit{obj} : C])$$
$$\wedge \textit{become\_effective}(\textit{ID\_bea}, [\textit{obj} : \textit{ID\_ac}, \textit{tim} : T])$$
$$\wedge \textit{acceptance}(\textit{ID\_ac}, [\textit{agt:}\ Y, \textit{goa:}\ X, \textit{obj} : \textit{ID\_of}]).$$

Such a formula, called *a Flatted CPF formula (FCPF)*, is an abbreviation of the CPF formula below:

$$\textit{be\_concluded}(\textit{ID\_bc}, [\textit{obj} : \textit{contract}(\textit{ID\_co}, [\textit{agt:}\ [X, Y], \textit{obj} : C]), \textit{tim} : T]) \longleftrightarrow$$

$$\textit{is\_effective}(\textit{ID\_ie}, [\textit{obj} : \textit{offer}(\textit{ID\_of}, [\textit{agt} : X, \textit{goa}: Y, \textit{obj} : C]), \textit{tim} : T])$$
$$\wedge \textit{become\_effective}(\textit{ID\_bea}, [\textit{obj} : \textit{acceptance}(\textit{ID\_ac},$$
$$[\textit{agt}: Y, \textit{goa}: X, \textit{obj} : \textit{offer}(\textit{ID\_of}, [\textit{agt}: X, \textit{goa}: Y, \textit{obj} : C])]), \textit{tim} : T]).$$

Legal sentences are described and stored in a knowledge base in this form. To execute the predicational reasoning, these formulas are compiled (flatted) into FCPF above.*

Any CPF A can be flatted into an FCPF formula according to the following procedure:

1. if A contains no formulae which have the form of $p(\textit{ID\_p}, [c_1 : t_1, \ldots, c_i : q(\textit{ID\_q}, [\,]), \ldots, c_n : tn])$ $(1 \leq i \leq n)$ in A, the formula is not flatted.
2. if A contains any formulae described in 1 above, choose the left-most one, replace $c : q(\textit{ID\_q}, [\,])$ with $c : \textit{ID\_q}$, and replace the original formula with the one below,

$$p(\textit{ID\_p}, [\ldots, c : \textit{ID\_q}, \ldots]) \wedge q(\textit{ID\_q}, [\,]).$$

3 Repeat the procedure in 2 above until it can no longer be applied.

We shall need other rules based on articles of CISG which we will we use below. Each rule is followed by its representation in CPF.

Rule 15(1) (cf. Article 15(1)). An offer becomes effective if and only if it reaches the offeree and a withdrawal of the offer does not become effective.

$$\textit{become\_effective}(\textit{ID\_beo}, [\textit{obj} : \textit{offer}(\textit{ID\_of},$$
$$[\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C]), \textit{tim} : T_1]) \longleftrightarrow$$
$$\textit{offer}(\textit{ID\_of}, [\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C])$$
$$\wedge \textit{reach}(\textit{ID\_re}, [\textit{goa} : Y, \textit{obj} : \textit{ID\_of}, \textit{tim} : T])$$
$$\wedge \neg(\textit{become\_effective}(\textit{ID\_bew}, [\textit{obj} : \textit{withdrawal}(\textit{ID\_wd},$$
$$[\textit{agt} : X, \textit{goa} : Y, \textit{obj} : \textit{ID\_of}]),$$
$$\textit{tim} : \textit{before}(T_1, [\textit{tim} : T_1, \textit{tto} : T])])).$$

Rule 15(2) (cf. Article 15(2)). A withdrawal of the offer becomes effective if and only if the withdrawal reaches the offeree unless the offer reaches the offeree before

* Some observers may think that we have become too involved in the so-called "Russell Paradox" when attempting to formalize FCPFs. However, this fear is not justified. A CPF formula is only an abbreviation of its corresponding FCPF formula. And this flatting is, substantially, the procedure for converting a many-sorted formula into a one-sorted one. If we apply the order sorted logic, we could deal with CPFs directly in inference.

the withdrawal does.

$$\begin{aligned}&\textit{become\_effective}(\textit{ID\_bew}, [\textit{obj} : \textit{withdrawal}(\textit{ID\_wd}, [\textit{agt} : X, \textit{goa} : Y,\\ &\textit{obj} : \textit{offer}(\textit{ID\_of}, [\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C])]), \textit{tim} : T]) \longleftrightarrow\\ &\textit{reach}(\textit{ID\_rew}, [\textit{goa} : Y, \textit{obj} : \textit{withdrawal}(\textit{ID\_wd},\\ &[\textit{agt} : X, \textit{goa} : Y, \textit{obj} : \textit{ID\_of}]), \textit{tim} : T])\\ &\wedge\neg(\textit{reach}(\textit{ID\_reo}, [\textit{goa} : Y, \textit{obj} : \textit{offer}(\textit{ID\_of}, [\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C]),\\ &\textit{tim} : \textit{before}(T_1, [\textit{tim} : T_1, \textit{tto} : T]))).\end{aligned}$$

Rule 16(1) (cf. Article 16(1)). A revocation of an offer becomes effective if and only if the revocation reaches the offeree before he has dispatched an acceptance of the offer.

$$\begin{aligned}&\textit{become\_effective}(\textit{ID\_berv}, [\textit{obj} : \textit{revocation}(\textit{ID\_isre},\\ &[\textit{agt} : X, \textit{goa} : Y, \textit{obj} : \textit{offer}(\textit{ID\_of},\\ &[\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C])]), \textit{tim} : T]) \longleftrightarrow\\ &\textit{reach}(\textit{ID\_rer}, [\textit{goa} : Y, \textit{obj} : \textit{revocation}(\textit{ID\_isre},\\ &[\textit{agt} : X, \textit{goa} : Y, \textit{obj} : \textit{ID\_of}]), \textit{tim} : T])\\ &\wedge\neg(\textit{dispatch}(\textit{ID\_dpa}, [\textit{agt} : Y, \textit{goa} : X, \textit{obj} : \textit{acceptance}(\textit{ID\_ac},\\ &[\textit{agt} : Y, \textit{goa} : X, \textit{obj} : \textit{offer}(\textit{ID\_of},\\ &[\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C])]), \textit{tim} : \textit{before}(T_1, [\textit{tim} : T_1, \textit{tto} : T])])).\end{aligned}$$

Rule 17 (cf. Article 17). A rejection of an offer becomes effective when a rejection reaches the offeror.

$$\begin{aligned}&\textit{become\_effective}(\textit{ID\_berj}, [\textit{obj} : \textit{rejection}(\textit{ID\_rj}, [\textit{agt} : Y, \textit{goa} : X,\\ &\textit{obj} : \textit{offer}(\textit{ID\_of}, [\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C])]), \textit{tim} : T]) \longleftarrow\\ &\textit{reach}(\textit{ID\_rej}, [\textit{goa} : X, \textit{obj} : \textit{rejection}(\textit{ID\_rj},\\ &[\textit{agt} : Y, \textit{goa} : X, \textit{obj} : \textit{offer}(\textit{ID\_of}, [\textit{agt} : X, \textit{goa} : Y, \textit{obj} : C])]), \textit{tim} : T]).\end{aligned}$$

Rule 18(2) (cf Article 18(2)). An acceptance of an offer becomes effective at the moment T when the acceptance reaches the offeror.

$$\textit{become\_effective}(\textit{ID\_bea}, [\textit{obj} : \textit{acceptance}(\textit{ID\_ac},$$

$[agt : Y, goa : X, obj : ID\_of]), tim : T]) \longleftarrow$

$reach(ID\_rea, [goa : X, obj : acceptance(ID\_ac, [agt : Y, goa : X,$

$obj : offer(ID\_of, [agt : X, goa : Y, obj : C,])]), tim : T]).$

8. Application of CPF to Legal Reasoning

The ID-symbols play an important role in CPF. We have already noted some advantages of ID-symbols. In this section, we will attempt to show another advantage from introducing ID-symbols into legal reasoning. Since our CPF has its basis in the standard first order language, we can utilize its inference rules. Moreover, ID-symbols increase the power of our language so that we can also deal with some legal reasoning cases that have been difficult to cope with up to this point. For example, consider the following common situation:

On October 1st, A in Budapest dispatched a letter of an offer to B in Hamburg, the content of which is that A sells a construction machine to B. The letter reached B on October 8th.

In the following two cases, have contracts been concluded?

Case a: On October 7th A telephoned to say “I withdraw my offer”. B said immediately after that “I accept your offer”.

Case b: On October 7th A dispatched a revocation of offer, which reached B October 11th. On October 10th B dispatched the acceptance of the offer, which reached A on October 12th.

Legal regulations presuppose the following principle:

Pr1: Everything is effective at T if and only if it becomes effective at time T_1 before T and it is not the case that it becomes null at any time between T_1 and T.

The CPF formalization is as follows:

$is\_effective(ID\_ie,([obj : X, tim : T])) \longleftrightarrow$

$become\_effective(ID\_bee, [obj : X, tim : before(T_1, [tim : T_1, tto : T])])$

$\wedge\neg(become\_null(ID\_ben, [obj : X, tim : between(T_2, [tfr : T_1, tto : T])])).$

In relation to **Pr1**, **Rule 16(1)** and **Rule 17**, legal regulations presuppose the following principle:

Pr2: An offer becomes null at T if and only if a revocation of the offer becomes effective or a rejection of the offer becomes effective at T.

The relevant legal facts of the above situations can be formalized by means of CPF as follows:

The common situation(C):
C-1: On October 1st, A in Budapest dispatched an offer to B in Hamburg, the content of which is that A sells a construction machine to B.

dispatch(*id_dpo1* [*agt :* ‘A’, *goa :* ‘B’, *obj : offer*(*id_of1*, [*agt :* ‘A’, *goa :* ‘B’, *obj :* “A sells a construction machine to B”]), *tim : 10_1*]).

C-2 : It reached B on October 8th.

reach(*id_reo1*, [*obj : id_of1*, *goa :* ‘B’, *tim : 10_8*]).

Case a:
Case a-1: A’s withdrawal of the offer reached B on October 7th.

reach(*id_rew*, [*goa :* ‘B’, *obj : withdrawal*(*id_wd1*, [*agt :* ‘A’, *goa :* ‘B’, *obj : offer*(*id_of 1*, [*agt :* ‘A’, *goa :* ‘B’, *obj :* “A sells a construction machine to B”])]), *tim : 10_7*]).

Case a-2: B’s acceptance of the offer reached A on October 7th.

reach(id_rea, [*goa: A*, *obj : acceptance* ([*id_ac*, [*agt :* ‘B’, *goa :* ‘A’, *ob: id_of1*]), *tim :’10_7*]).

Below is a statement which can be proven from the common situation and **Case a**[⋆]:

fa-1: It is not the case that the offer reached B before October 7th.

¬(*reach*(*id_reo1*, [*obj : id_of1*, *goa :* ‘B’, *tim : before*(*10_8*, [*tim : 10_8*, *tto : 10_7*])])).

Case b:
Case b-1: B dispatched the acceptance of the offer on October 10th.

dispatch(*id_dpa2*, [*agt :* ‘A’, *goa :* ‘B’, *obj : acceptance*(*id_aco1*, [*agt :* ‘A’, *goa :* ‘B’, *obj : offer* (*id_of1*, [*agt :* ‘A’, *goa :* ‘B’, *obj :* “A sells a construction machine to B”])]), *tim : 10_10*]).

[⋆] We omit the formalization of the following facts because of the limitation of space. In a legal reasoning system, these propositions can be proven through “negation as failure” under the closed world assumption.

Case b-2: B's acceptance of the offer reached A on October 12th.

reach(*id_rea*, [*obj : acceptance*(*id_aco1*,

[*agt :* 'B', *goa :* 'A', *obj : offer* (*id_of1*, [*agt :* 'A', *goa :* 'B', *obj :*

"`A sells a construction machine to B`"])]), *tim : 10_12*]).

Case b-3 : A's revocation of the offer reached B on October 11th.

reach (*id_rerv*, [*obj : revocation*(*id_revo1*,

[*agt :* 'A', *goa :* 'B', *obj : offer* (*id_of1*, [*agt :* 'A', *goa :* 'B', *obj :*

"`A sells a construction machine to B`"])]), *tim : 10_11*]).

Some of the statements which can be proven from the situation and **Case b** are:

fb-1: Any withdrawal of the offer did not become effective.
fb-2: Any rejection of the offer did not become effective.

Now, let us check the respective cases.

In Case a: We get the answer "The contract was not concluded between A and B" to the question "is a contract concluded?" as the withdrawal of the offer becomes effective. **Rule 15(2)** applies to this case. Formalization of this conclusion is as follows:

$\neg\exists ID\_bc \exists ID\_co \exists ID\_of \exists T(be\_concluded(ID\_bc,$

$[obj : contract(ID\_co, [agt : [A, B], obj : ID\_of]), tim : T])).$

In Case b: We get the answer "yes", based on the following reasoning. We conclude that the offer becomes effective, based on **Rule 15(1)**. The letter of acceptance of the offer reached A on October 12. Thus, we conclude that the acceptance of the offer becomes effective, according to **Rules 18(2)**. In turn, we conclude that the contract is consummated by virtue of **Rule 23**. In this case, the revocation does not become effective since the requirements of **Rule 16(1)** are not met. Formalization of this conclusion is as follows:

be_concluded(*id_bc*, [*obj : contract*(*id_co*,

[*agt :* $[A, B]$, *obj : id_of1*]), *tim : 10_12*]).

We can give the formalization of the above informal reasoning using CPF in the following manner:

Deduction in Case a

1 Formalization of Pr1 in CPF
2 Formalization of Rule 15(1) in CPF
3 Formalization of Rule 15(2) in CPF
4 Formalization of Rule 23 in CPF
5 Formalization of Case a-1 in CPF
6 Formalization of fa-1 in CPF

---

7 Formalization of the conclusion in CPF.

Lines 5 and 6 fulfill the requirements of line 3, **Rule 15(2)**. Therefore, it is proven that the withdrawal of the offer became effective on October 7th. Therefore, it is proven through line 2, **Rule 15(1)**, that it is not the case that the offer does not become effective. Therefore, it is proven through line 1, **Pr1** that the first requirement of line 4, **Rule 23** is not fulfilled, i.e. the offer is not effective at any time. Therefore, it is proven that no contract has been concluded.[⋆]

Now, consider

Deduction in Case b

1 Formalization of Pr1 in CPF
2 Formalization of Pr2 in CPF
3 Formalization of Rule 15(1) in CPF
4 Formalization of Rule 15(2) in CPF
5 Formalization of Rule 16(1) in CPF
6 Formalization of Rule 18(2) in CPF
7 Formalization of Rule 23 in CPF
8 Formalization of C-2 in CPF
9 Formalization of Case b-1 in CPF
10 Formalization of Case b-2 in CPF
11 Formalization of Case b-3 in CPF
12 Formalization of fb-1 in CPF
13 Formalization of fb-2 in CPF

---

14 Formalization of the conclusion in CPF.

[⋆] Strictly speaking, some principles governing temporal reasoning might be needed in order to sanction these deductions. But they should be obvious. So we do not mention them explicitly.

From line 3 **Rule 15(1)**, line 8, together with line 12, we can infer that the offer becomes effective on October 8th. (We call this conclusion (a).) From line 6, **Rule 18(2)** and line 10 we deduce that the acceptance of the offer becomes effective on October (b). From lines 9 and 11 we infer that the acceptance of the offer is dispatched on October 10th before October 11th, i.e. the date when A's revocation reaches B. Therefore, the second requirement of line 5, **Rule 16(1)** is not fulfilled, thus proving that the revocation does not become effective, even if it reaches the offeree B(c). From line 2, **Pr2** and (c) together with line 13 it is proven that: it is not the case that the offer becomes null (d). From line 1, **Pr1** and (a) together with (d), we can infer that the offer is effective on October 12th (e). From line 7, **Rule 23** and (e) as well as (b), we conclude therefore that the contract is concluded between A and B on October 12th.

We have applied the CPFs above as examples of legal reasoning in the field of the contracts for the international sales of goods. We would like to now explain a merit of ID-symbols with respect to legal reasoning. In the above rules, we have expressions which refer to an instance of a given concept, withdrawal of an offer, revocation of an offer, an acceptance of an offer, etc. Hence, in order to formalize the above rules and reason using them, we need such terms to denote a particular instance of a concept. As a result, we have introduced ID-symbols in CPFs. CPFs could formalize relevant rules, facts and legal reasoning adequately. Such representation of legal reasoning would be difficult only by means of pure standard first order language. As is shown, we can enrich the representation power of first order language by introducing ID-symbols to a considerable degree. We have, thus, a new formal system enabling us to reify relations themselves.

9. The Semantics of CPF

We can define the semantics of CPFs in the usual manner. The only difference between the usual first order language and CPF is the introduction of ID-symbols in the latter. Our basic idea in defining the interpretation of ID-symbols is that an ID-symbol *ID_p* stands for an instance of the given predicate p. The semantics of CPF reflects this point. A model for CPF is of the form $M = \langle D^1, D^2, I, \Phi\rangle$, where D^1 and D^2 are non-empty sets. Intuitively, D^1 is a set of possible state of affairs or instances of relations. D^2 is a set of individuals in the usual sense. I is an interpretation function defined on D. If t is a individual constant, then $I(t) \in D^1 \cup D^2$. For an n-place predicate symbol p with the ID-position, we define its extension $I(p)$ as follows:

$$I(p) \subseteq D^1 \times D_1 \ldots \times D_n,$$

where D_i is either D^1 or D^2. Φ is a choice function on D^1, i.e. for any non-empty subset E of D^1, $\Phi(E) \in E$. Next, we introduce an assignment function g such that

g : *INDVAR*$^\star \longmapsto D^1 \cup D^2$. The following function I_g (it is called an interpretation relative to the assignment function g) is to be defined:

1. If t is an individual variable, $I_g(t) = g(t)$.
2. if t is an individual constant, $I_g(t) = I(t) \in D^1 \cup D^2$.

With these preliminaries completed, we can define the satisfaction of the formulae with respect to a given assignment and interpretation of the ID-symbols in the following manner. First, a satisfaction of an atomic formula with the ID-position is defined as follows:

$$M \models_g p(t, [t_1, \ldots, t_n]) \Longleftrightarrow (\langle I(t), I(t_1), \ldots, I(t_n)\rangle) \in I(p).$$

Then, derivatively, we can define a satisfaction of an atomic formula without the ID-position as follows:

$$M \models_g p([t_1, \ldots, t_n]) \Longleftrightarrow \exists \alpha \in D^1(\langle \alpha, I(t_1), \ldots, I(t_n)\rangle \in I(p)).$$

As to the equality predicate, it would appear as

$$M \models_g \; s = t \Longleftrightarrow I_g(s) = I_g(t).$$

Having determined the truth value of $p([t_1, \ldots, t_n])$ in the model $M = \langle D^1, D^2, I, \Phi\rangle$, we are ready to define the interpretation of an ID-symbol *id_p*$([t_1, \ldots, t_n])$: If $\{I(X) : M \models_g p(X, [t_1, \ldots, t_n])\} \neq \emptyset$, then

$$I_g(\textit{id\_p}) = \Phi\{I(X) : M \models_g p(X, [t_1, \ldots, t_n])\}.$$

Otherwise, we simply determine:

$$I_g(\textit{id\_p}) = \textit{an arbitrary element of } D^1.$$

As for the other types of expressions, we shall attach the usual interpretations.

9.1. Basic Properties of Our Semantics

In this subsection we shall formalize some of the basic properties of our semantics. First, consider

$$M \models_g \exists X p(X, [t_1, \ldots, t_n]) \Longleftrightarrow p(\textit{id\_p}, [t_1, \ldots, t_n]).$$

Secondly, consider

$$M \models_g \exists X p(X, [t_1, \ldots, t_n]) \Longleftrightarrow M \models_g p([t_1, \ldots, t_n]).$$

These are desirable properties of our semantics. The reader can easily verify that our models will validate all of the axioms listed below. From these observations the soundness of the system below follows immediately.

* *INDVAR* is the set of the individual variables of our language.

10. The Axiomatic System of CPFs

We have explained the ID-symbols and CPFs. It was pointed out that ID-symbols and ε-symbols have the same logical roles. Hence, it is not difficult to construct an axiomatic system for which we can use the system of ε-symbols to construct our system. As a matter of fact, it is just a simple adaptation of the system of ε-symbols. Of course, there are several minor differences.

The axioms of CPF are as follows.

1. All tautological formulae of the syntax of CPFs.
2. If φ, ψ are formulae of the syntax of CPF and X is an individual variable (therefore, X is a X_i or T_j) not free in φ, then the formula

   $$(\forall X\psi \longleftarrow \varphi) \longleftarrow \forall X(\psi \longleftarrow \varphi)$$

   is an axiom.
3. If φ, ψ are formulae and ψ is obtained from φ by freely substituting each free occurrence of v in φ by the term t (i.e., no variable X in t shall occur bound in ψ at the place where it is introduced), then the formula

   $$\psi \longleftarrow \forall X\psi$$

   is an axiom.
4. (*Identity Axioms*) If X, Y are variables, $t(X_1, \ldots, X_n)$ is a term and $\varphi(X_1, \ldots, X_n)$ is an atomic formula, then the formulas
   1. $X = X$
   2. $t(X_1, \ldots, X_{i-1}, X, X_{i+1}, \ldots, X_n) =$
      $t(X_1, \ldots, X_{i-1}, Y, X_{i+1}, \ldots, X_n) \longleftarrow X = Y$
   3. $(\varphi(X_1, \ldots, X_{i-1}, Y, X_{i+1}, \ldots, X_n) \longleftarrow$
      $\varphi(X_1, \ldots, X_{i-1}, X, X_{i+1}, \ldots, X_n)) \longleftarrow X = Y$

   are axioms.
5. If $p(X, [t_1, \ldots, t_n])$ is an atomic formula with the ID-position occupied by X, then

   $$p(\mathit{id\_p}, [t_1, \ldots, t_n]) \longleftarrow \exists X p(X, [t_1, \ldots, t_n])$$

   is an axiom.
6. $\mathit{id\_p}([t1, \ldots, t_n]) = \mathit{id\_q}([t_1, \ldots t_n]) \longleftarrow \forall X(p(X, [t_1, \ldots, t_n]) \longleftrightarrow$
   $q(X, [t_1, \ldots, t_n]))$
   is an axiom.

   Rule 1: Rule of Detachment: From φ and $\psi \longleftarrow \Phi$, we can infer ψ.

   Rule 2: Rule of Generalization: From φ, we can infer $\forall X\varphi$.

Given the axioms and the rules of inference, we can define the usual notions of proof, theorem, etc. Following standard usage, $\vdash \varphi$ means that φ is a CPF theorem and $\sum \vdash \varphi$ means that there is a proof of φ from the axioms and $\sum$, a set of formulae.

We should mention briefly concerning the completeness of the axiomatic system of CPF that the proof itself is almost clear, for the basic idea is in the spirit of Henkin's proof of the completeness of first order predicate calculus. That is, what we prove is that every consistent set of sentences T has a model. For a more precise proof, see Leisenring (1969, Ch. 4).

11. Conclusion

Typically, natural languages are used for expressing legal knowledge. Hence, if we want to formalize inferences with legal knowledge, we believe it would be better to use a formal language which has the ability to translate sentences expressed by a natural language correctly. We already possess such a formal language, called "Montague grammar". If we focus on the logical aspect of Montague grammar, we find there the theory of types, which is an extension of higher-order logic. Of course, Montague grammar has another component, namely, a categorial grammar. Because of these, a Montague grammar possesses a characteristic which we cannot find in the first-order language. This characteristic is the quantification over properties. But it is too large as a system to be treated computationally. Therefore, introducing CPF, which is easy to use computationally, helps to resolve this problem. It is also very useful in expressing individual concepts. Moreover, as to the latter, CPF is less complex than sorted logic. We have so far been stating the quantifier-free part of CPF (syntax, legal knowledge representation using it, semantics, and so on). Finally, by way of summary I give the merits of introducing CPF, in particular ID-symbols.

- CPFs have great expressive capacity.
- CPFs make legal knowledge representation whose form is close to natural language.

In this way CPFs have significant advantages. The most important significance of this paper is that it gives a logical basis for CPF.

In order to avoid the focus of the argument becoming obscured, I omitted to offer an explanation of case symbols here. Of course they are an important tool. Case symbols are a device for clarifying what roles terms play in a predicate. It is interesting to give semantics for such a category of grammar.

I have often mentioned the characteristics of ID-symbols in terms of demonstrative pronouns. We must also consider the other kinds of demonstrative pronoun (indexicals such as "I", demonstratives such as "the book I have" and so on) from a wider point of view. The following are two points to consider in the future.

- How to formally identify objects that are represented by making some extension of first order language.
- How to combine ID-symbols and many sorted language.

References

Davidson, D. (1980). The Logical Form of Action Sentences. In Davidson, D. (ed.) *Essays on Actions and Events*, 105–122. Oxford University Press: New York.

Ebbinghaus, H.D., Flum, J. & Thomas, W. (1984). *Mathematical Logic*. Springer-Verlag: New York.

Gupta, A. (1980). *The Logic of Common Noun.* Yale University Press: New Haven.

Leisenring, A.C. (1969). *Mathematical Logic and Hilbert's ε-symbol*. Gordon and Breach Science Publishers: New York.

Lloyd, J. (1987). *Foundations of Logic Programming.* Springer-Verlag: Berlin.

McCarty, T. (1989). A Language For Legal Discourse I. Basic Features. In Proceedings of *The Second International Conference on Artificial Intelligence and Law*, 180–189. The Association for Computing Machinery.

McCarty, T. & van der Meyden, R. (1992). Reasoning about Indefinite Actions. In *Principles of Knowledge Representation and Reasoning. Proceedings of the Third International Conference (KR92)*, 59–70. Morgan Kaufmann: Los Altos.

Rödig, J. (1972). Über die Notwendigkeit einer besonderen Logik der Normen. In Albert, H. et al. (eds.) *Rechtstheorie als Grundlagenwissenschaft der Rechtwissenschaft, Jahrbuch für Rechtssoziologie und Rechtstheorie* Bd. 2, 163–185. Bertelmann Universitätsverlag: Düsseldorf.

Sakurai, S. & Yoshino, H. (1993). Identification of Implicit Legal Requirements with Legal Abstract Knowledge. In Proceedings of *The Fourth International Conference on Artificial Intelligence and Law*, 298–305. Association for Computing Machinery: Amsterdam.

Yoshino, H. (1978). Über die Notwendigkeit einer besonderen Normenlogik als Methode der juristischen Logik. In Klug, U. et al. (eds.) *Gesetzgebungstheorie, Jurisitische Logik, Zivil- und Prozeßrecht (Gedächtnisschricht für Jürgen Rödig)*, 140–161. Springer-Verlag: Berlin.

Yoshino, H. (1994). Representation of Legal Knowledge by Compound Predicate Formula. In *Pre-Proceedings of the Workshop on Legal Application of Logic Programming (ICLP '94)*, 128–137.

Yoshino, H. (1995). Systematization of Legal Meta-inference. In *Proceedings of The Fifth International Conference on Artificial Intelligence and Law*, 266–275. Association for Computing Machinery: Maryland.

Yoshino, H. & Kitahara M. (1988). LES-Project. In Fielder, H. et al. (eds.) *Expert Systems in Law (Neue Methoden im Recht Band 4)*, 47–65. Attempto Verlag: Tübingen.

Yoshino, H. & Kakuta, T. (1992). The Knowledge Representation of Legal Expert System LES-3.3 with Legal Meta-inference. In Yoshino, H. (ed.) *Legal Knowledge and Legal Reasoning Systems (Proceedings of the 6th International Symposium of Legal Expert System Association), LESA*, 1–9. Legal Expert System Association: Tokyo.

Yoshino, H. et al. (1987). Legal Expert System Les-2. In Wada, E. (ed.) *Logic Programming '86*, Lecture Notes in Computer Science, Vol. 264, 36ff, Springer-Verlag: Berlin.

Yoshino, H. et al. (1993). Towards a Legal Analogical Reasoning System Knowledge Representation and Reasoning Methods. In Proceedings of *The Fourth International Conference on Artificial Intelligence and Law*, 110–116. Association for Computing Machinery: Amsterdam.