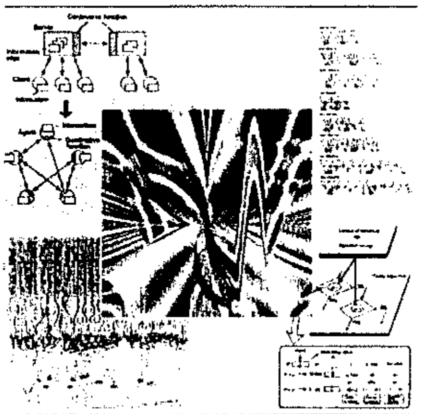
Journal of Advanced Computational Intelligence

Vol.2 No.1
Feb. 1998



Cover Pictures:

Upper left:
A function of Agent

Upper right
Examples of leg's motion based on data received for both waking and running during constant velocity trial
(Dr. Takashi Yokoi, National Institute of Bioscience and Human Technology)

Center
2D CG image drawn with "SBART," a simulated breeding tool.
(Prof. Tatsuo Unemi, Dept. of Information Systems Science, Soka University)

Lower left
The structure of retina

Lower right
The threshold control method of fuzzy algorithm in FRASH2
(Prof. Yoichiro Maeda, Faculty of Engineering, Osaka Electro-Communication University)

Publishing Staff:

Editor	Keiji Hayashi
Assistant Editor	Yasushi Inoue
	Kiyoe Kojima
Art Director	Prof. Yuji Isa
Publisher	Keiji Hayashi

Published bimonthly by
Fuji Technology Press Ltd.

7F Daini Bunsei Bldg.
11-7,Toranomon 1-chome
Minato-ku, Tokyo, 105 Japan
Tel: +813-3508-0051
Fax: +813-3592-0648
E-mail: TAE00762@niftyserve.or.jp

One year subscription

Institutional rate	(Air speed)	$660
	(Vol.1	$220)
Personl rate	(Air speed)	$140
	(Vol.1	$40)
IFSA menber	(Air speed)	$110

Contents

Editorial for Special Issue:

AI and Law

Hajime Yoshino

Faculty of Law, Meiji Gakuin University
1-2-37 Shirokanedai, Minatoku, Tokyo 108, Japan
E-mail: yoshino@mh.meijigakuin.ac.jp

Katsumi Nitta

Department of Computational Intelligence and Systems Science,
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226, Japan
E-mail: nitta@dis.titech.ac.jp

In the last issue (Vol.1, No.2), we introduced the Legal Expert System (LES) project led by Hajime Yoshino of Meiji Gakuin University, presenting six papers on the LES project. Those papers were mainly related to higher order reasoning systems such as case-based reasoning, abductive and inductive logic programming, nonmonotonic reasoning, and analogical reasoning.

The objective of the LES project was to develop a legal expert system effective for use by lawyers, so the project covers inference mechanisms, analysis of legal knowledge, and user interfaces.

In this second special issue on the LES project, we present five more papers, mainly related to the analysis of legal knowledge, legal knowledge representation language, and legal reasoning system user interfaces.

Hajime Yoshino analyzes the logical structure of contract law. To develop a knowledge base for the United Nations Convention on Contracts for the International Sale of Goods (CISG), he proposes a clear logical model of the contract law system, which treats relations between events and legal status such as rights and obligations. Yoshino demonstrates that legal metarules are effective in constructing deductive legal reasoning systems, and are appropriate from the viewpoint of jurisprudence.

Seiichiro Sakurai discusses the logical features of the legal knowledge representation language, CPF, developed by Hajime Yoshino. CPF is a logic programming language that enhances the representation of complex data structures. CPF is a convenient tool for representing legal knowledge, yet lawyers often attempt to describe nonexecutable forms of CPF rules.Sakurai introduces a way to construct an executable knowledge base from lawyers' CPF rules.

Masato Shibasaki and Katsumi Nitta introduce a new framework to formalize nonmonotonic reasoning with dynamic priorities. The several frameworks proposed thus far to model relationships among arguments do not treat complex arguments, composed of strict rules and default rules. They show that the new framework represents such relationships and analyze these relationships for this framework and others.

Takashi Miyata and Yuji Matsumoto introduce LES natural language generation using a user interface for lawyers rather than computer scientists. They describe a sentence generation system that translates logical forms provided from an inference engine into natural-language sentences, and present the unification grammar, generation algorithm and graphical debugging tool.

To develop a knowledge base, the lawyers of the LES project analyze and represents the relationships between requirements (actions or events) and consequences (legal status) of legal rules in the form of logical flowcharts. Once the appropriateness of a flowchart is confirmed, they convert it to a CPF rule in their knowledge base. Koji Miyagi, Motoki Miura and Jiro Tanaka developed a flowchart editor that makes legal flowcharting easier. To make it easier to decide where to locate flowchart components and draw linens between the components, the editor possesses several algorithms.

Paper:

Logical Structure of Contract Law System – For Constructing a Knowledge Base of the United Nations Convention on Contracts for the International Sale of Goods –

Hajime Yoshino

Faculty of Law, Meiji Gakuin University
1-2-37 Shirokanedai, Minato-ku, 108 Tokyo
E-mail: yoshino@mh.meijigakuin.ac.jp
[Received January 20, 1998; accepted February 10, 1998]

In order to construct a deductive legal knowledge base, it is necessary first to clarify the structure of the law as a deductive system from which a legal judgement can be justified as a conclusion of logical deduction together with relevant facts. As the legal state of affairs changes according to the time progress of an event, a clarified logical model of law is necessary to enable us to deduce the changes among legal relationships over time from the beginning to the end of a case. This study presents such a model based on Logical Jurisprudence, in which the relationship between legal sentences and the legal meta sentences regulating the validity of legal sentences plays a definitive role. The model is applied to the United Nations Convention on Contracts for the International Sale of Goods (CISG) to develop a deductive knowledge base. The deductive structure of the contract law is clarified so that appropriate answers are deduced to questions about legal state of affairs at any time point as a result of the application of CISG provisions to a concrete case.

Keywords: Contract, CISG, Expert system, AI, Legal knowledge, Logical structure

1. Introduction

In the "Legal Expert" Project, we have developed a knowledge base of the United Nations Convention on Contracts for the International Sale of Goods (CISG). For a legal knowledge base of the CISG, it has been necessary for us first to clarify the logical structure of the contract law system as a whole because, to show (justify) a legal judgement as a conclusion of logical deduction from a legal system of the CISG, together with a given fact by means of a legal expert system, we must make a deductive knowledge base of the CISG and, for such construction, we must to have a clear logical model of the contract law system to which the CISG belongs and upon which it is based, thus making it possible to justify the judgement as a result of logical deduction.

The legal state of affairs, which refers to the status of legal relations, changes according to time progress of an event. We therefore must clarify such a logical model of law that enables us to deduce changes of legal relation according to time, regardless of any time point in given events from the beginning to the end: for example, before or after the contract conclusion; before or after fulfillment or non-fulfillment of an obligation on contract; before or after remedies for breach of contract; before or after cancellation of a contract; before or after fulfillment or non-fulfillment of restitution, and so on. The present work contributes to this clarification.

The systematization of law, i.e., to present the law as a deductive system, has long been a central theme of legal theories, but remains illusive.[1] Modern mathematical logic and the construction of a knowledge base system of law gives us the opportunity to systematize this properly, succinctly and explicitly and demonstrate that the proposed systematization is correct.

I believe we have already clarified the logical structure of the contract law system in the above sense and have developed a knowledge base that demonstrates it appropriately. Our aim is to present the essence of that clarification of the logical structure of contract law system by focusing on the United Nations Convention on Contracts for the International Sale of Goods (CISG).

The study is based on Logical Jurisprudence. This paper demonstrates the basic structure of law from the point of Logical Jurisprudence. In accordance with such a framework, this study clarifies and demonstrates the structure of contract law as a deductive system from which a legal decision may be justified as a logical deduction when the CISG is applied to a concrete case. This report considers the relationship between legal sentences and legal meta sentences that provide the validity of legal sentences as the starting point for legal knowledge analysis and modeling. From this point, a deductive model of the contract law system is presented and applied to the CISG. The legitimacy of the model is demonstrated in an example of the CISG application to a concrete case.

1 The systematization of law has been endeavored especially in continental law countries. Scholars of modern natural law, such as H. Grotiius, S. F .v. Pufendorf, and B. de. Spinoza have tried to present the natural law system as a deductive system such as geometry. Legal scholars of general theory of law in Germany, such as F. R. Bierling and K. Bergbohm, have tried to explicate positive law as a deductive system. From a strictly logical point of view, however, they did not succeed in presenting a legal system as deductive. Cf. Ref.9).

2. Logical Jurisprudence

Logical Jurisprudence ("ronri hogaku" "Logishe Rechtslehre") is a legal theoretically developed form of a discipline in Jurisprudence that we call "legal logic" or "Juristishe Logik."

Logical Jurisprudence tries to constitute the world of legal discourse in terms of smallest unit of primitives. It starts from three primitives: "sentence," "validity" of sentence, and "inference rule." Logical Jurisprudence attempts to explain or model the law using these three notions.

Logical Jurisprudence does not support in the existence of "legal norms as a meaning," which has traditionally been admitted or presupposed in legal studies and praxis. Logical Jurisprudence presupposes the notion "**sentence**." Sentences exist, as a form of written or spoken sign, cognizable or perceptible and therefore communicable. In our opinion, legal norms as a meaning belong to the world of images. It is what is imagined when legal sentences are thought of. To communicate images to other persons, they must be put them into sentential form perceptible by others. Logical Jurisprudence considers sentences in the field of law as the direct and sound object of legal recognition.[2] The second basic conception in Logical Jurisprudence is "**validity**" of a legal sentence. The validity of a legal sentence is viewed by Logical Jurisprudence as a "truth in the logical sense." That a legal sentence is valid means that the sentence is true in the world of legal discourse, i.e., legally true. Logical Jurisprudence represents this legal truth by means of a predicate (e.g., "is_valid(sentence1, goal1, time1)," which could be read as follows: "a sentence1 is valid for a goal1 at time1." The representation of the validity concept by a predicate is characteristic of the knowledge representation of Logical Jurisprudence that corresponds to the natural language representation of knowledge in the real legal world.

The third basic concept in Logical Jurisprudence is the "**inference rule**." The logical correct reasoning is based on inference rules. The main inference rule is *Modus Ponens* which is represented in the following schema where A and B express propositions:

$$(A \rightarrow B), A \Rightarrow B$$

This formula is read: If 'if A then B' is true and A is true, then follows: B is true. *Modus Ponens* is the basic reasoning schema legal justification, as discussed later.

In Logical Jurisprudence, legal reasoning is a process of the development of legal sentences. In other words, legal sentences are developed in the process of legal reasoning.

Logical Jurisprudence divides legal reasoning into reasoning of justification and reasoning of discovery. **Reasoning of legal justification** is reasoning through which a judgement is justified from already justified legal knowledge. Logical deduction is type of reasoning in legal justification. The logical structure of this reasoning is *Modus Ponens*. Judgement may not be deduced from statutes and facts alone, but may be shown to be deduced from the whole body of legal knowledge, including statutes, facts, and additional legal sentences to the former as implicit legal common sense or as a result of the reasoning of legal discovery. Logical Jurisprudence makes this implicit or discovered knowledge clear and identifies it to make it explicit. Following are such additional legal sentences: principles of law that unify statutory legal sentences; common sense about legal terms, especially hierarchical relations between legal concepts; and the proposition of interpretation of statutes that are produced by the reasoning of legal discovery. Logical Jurisprudence analyzes legal knowledge in detail, recognizes and demonstrates the implicit knowledge of legal experts, and legal sentences created by the reasoning of legal discovery, such that the reasoning of legal justification is formed as logical deduction.

Reasoning of legal discovery is reasoning through which judgements themselves or additional legal sentences are discovered or created. This reasoning is based on logical deduction because discovered legal sentences are to be set so that the whole reasoning process including these additional sentences can be presented as a logical deduction on the one hand and the reasoning of discovery is to be performed through a falsification inference on the other.[3] Falsification has the logical structure of *Modus Tollens*:

$$(A \Rightarrow B), \neg B \Rightarrow \neg A$$

This formula is read as follows. If 'if one sets a hypothesis A (together with theorems accepted already) then B follows' and it is proven that B is not true, then it follows that hypothesis A is not true. (The legal hypothesis cannot be proven as just but only falsified as unjust.)

The reasoning of legal discovery, however, requires something more than deduction. To get hypothesis A in the schema above, abductive or inductive reasoning are needed. Reasoning to get a hypothetical fact sentence is abduction and reasoning to generate a rule is induction. Logical Jurisprudence analyzes the legal reasoning process in two directions: (1) concretization (putting in concrete terms) and (2) systematization. This is also true for legal reasoning of discovery. The study of legal interpretation or analogy is important to concretization. In systematization, it is important to make legal principle sentences clear which will enable us to bring mere collections of legal sentences into a system, on the one hand, and to analyze how legal principle sentences are to be found as hypotheses, on the other.

The structure of legal reasoning in the application of law, where both reasoning of justification and discovery interact to a concrete case is shown in **Fig.1**.

The study of legal discovery reasoning is important to the theory of legal reasoning, both in concretization (Cf. Ref.15)) and systematization.[7)] Few engineers, however, study legal knowledge systematization itself, i.e., showing laws as a deductive system. This is: because engineers assume that a theory of science has a deductive system, they are not interested in finding the deductive structure of law and, furthermore, legal knowledge is too specialized and complicated for engineers to deduce the structure. To construct a legal expert system, however, the deductive struc-

2 The difference between conventional and legal sentences and how these differ is discussed in section 5.2.1.

3 We proved the quasi-deductive structure of such legal (theoretical) systems in modern mathematical logic and clarified the logical structure of reasoning of legal discovery as an falsification inference to reasoning of justice through which one gets more just legal rules, applying K. Popper's Theory of Falsification.[7)] We applied this result to analyze the logical structure of legal decision, where judges get a certain legal decision reasoning, as falsification inference of *Modus Tollens*.[11)]

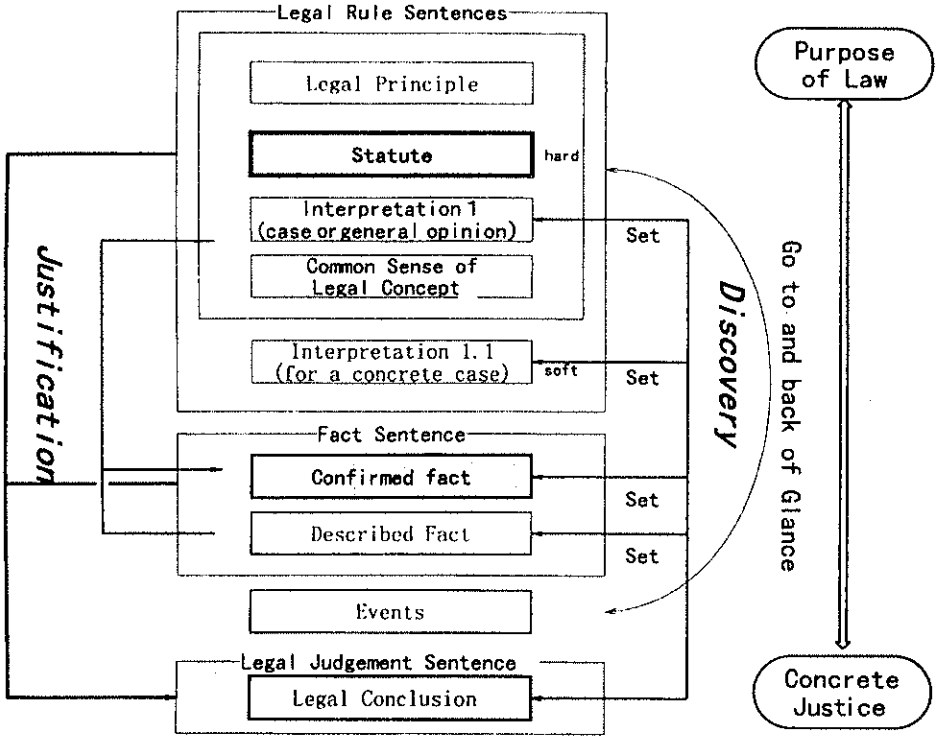


Fig. 1. Legal Reasoning Structure.

ture of law must be clarified to make a deductive knowledge base. It has long been desired in legal studies to clarify the deductive system of law and to systematize legal knowledge.[4)] We focus on how to systematize the law of contracts as a logical deductive system,[4] leaving the reasoning of legal discovery in CISG to another time.[5]

3. Basic Concepts and Structures of Legal Sentences

Sentences in the legal field, referred to here as **legal sentences,** are starting points. We introduce basic legal sentence concepts, according to which legal sentences are classified so that laws can be systematized as a deductive system of legal sentences.

First, it is important to distinguish between legal rule and fact sentences. Legal sentences consist of two types: **Legal rule sentences** have the following syntactic form: "$\forall X\{a(X) \leftarrow b(X)\}$." This formula is read: "For all X, X is a, if X is b." In legal sentences, the consequence of the sentence, which is the formula at left in the implication, is called a "legal consequence" and the antecedent, which is the formula at right, is called a "legal requirement." **Legal fact sentences** have the following syntactic form: "b(x1)," read: "x1 is b." Note that the difference between legal rule and fact sentences is, in Logical Jurisprudence, purely syntactic, as mentioned above.

Second, legal sentences are to be further classified in terms of elementary and complex legal sentences. An **elementary legal sentence** is the smallest unit of legal sentences. Statutes or contracts are composed of elementary legal sentences, e.g., "one must drive a car under 100 km /hour on a highway" or "A may require B to pay the price of $10000." A **complex legal sentence** is a group of legal sentences, e.g., "the United Nations Convention on Contracts for the International Sale of Goods," or "a contract for sale of a farming machine between A and B on October 8, 1997." Code and parts or sections or an article of a statute is a complex legal sentence. In most cases, the fact that a certain legal sentence belongs to a complex legal sentence is represented by the place and space where they are printed. The relationship is represented in Logical Jurisprudence by a sentence describing the united relationship of grouped sentences. The concept of a complex legal sentence enables us to treat the validity of legal sentences at once. Namely, if one has described the validity of a complex legal sentence, then all legal sentences that belong to it have been regulated. The advantage of the complex legal sentence is that it contributes to producing economical description.

It is also important for the deductive systematization of legal knowledge to distinguish between legal *object sentences* and legal *meta sentences*. A **legal object sentence** describes the object itself . In the legal domain, the object is an "obligation." Legal object sentences prescribe the obligations of a person. The sentence "one must drive a car under 100 km /hour on a highway" or "B must pay A the price of $10,000" is a legal object sentence. A **legal meta sentence** prescribes legal sentences. More precisely, it de-

4 Interesting books on law and legal reasoning modeling have been published.[2,3,6)] Our study developed independently of them. Our approach is different from Kralingen's approach, for example, in that it is not conceptual or frame-based, but purely logical, especially in that we analyze and reconstruct the law intensively in "legal sentences," "their validity," and "logical deduction."

5 We have already done this to a certain extent, i.e., Ref.15).

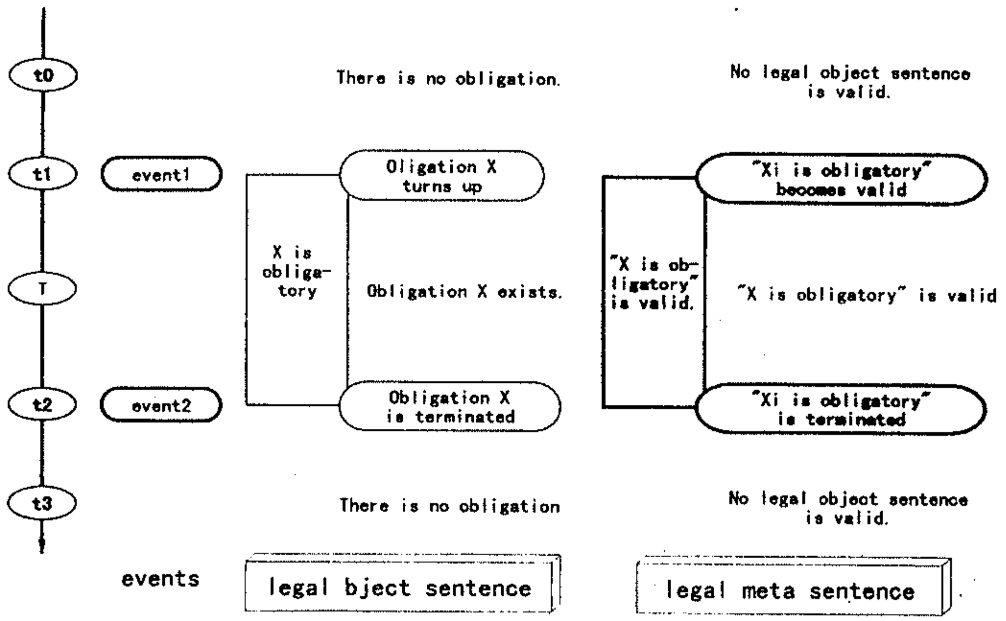


Fig. 2. Existence of Obligation and Validity of a Legal Sentence.

scribes the validity of a legal sentence. Some legal meta sentences describe the validity of legal meta sentences. An example of a legal meta sentence is: "A law is enforced 20 days after the day of its promulgation" (Article 1 of the law governing the application of laws (HOUREI)) or "(1) This Convention applies to contracts of the sale of goods between parties whose places of business are in different states: (a) when states are contracting states; or ... " (Article 1 of the CISG).

Law ultimately prescribes the obligation of persons. In other words, people's conduct is ultimately regulated by obligations given them by law. What legal obligations exist depend on the legal sentences that describe obligations, or more precisely, on the validity of legal object sentences. The validity of legal object sentences is prescribed by legal meta sentences. In Logical Jurisprudence, the existence of A's obligation to do Z means that "A has an obligation to do Z" or "It is obligatory for A to do Z" is valid. The relation of the existence of an obligation and the validity of a legal object sentence describing the obligation are shown in **Fig.2.**

The validity of legal meta sentences that prescribe legal object sentences is prescribed by other legal meta sentences. A legal meta sentence that prescribes the validity of a legal meta sentence is called a higher or upper level legal meta sentence. The validity of each legal meta sentence is prescribed by a higher level of legal meta sentences. The highest, final level of legal meta sentence is called a "basic" or "fundamental" legal sentence. The validity of the final, highest legal meta sentence is set as fact.[6]

In legal sentences describing rights, note that they are not legal object sentences, which describe obligations. They do not belong to an object level of legal language but to a meta level. Logical Jurisprudence takes sentences that describe rights as a legal meta rule sentence, which make it possible to set forth a new legal object rule. This is discussed later.

4. Case and Solution

This section describe an example of CISG and questions on the example, and introduces legal solutions to questions so that the deductive knowledge structure of contract law by which solutions may be deduced are clarified.

[Case7h]

(1) On April 3, 1997 A, a farming machine maker in New York sent a letter to the branch office in Hamburg of B, a Japanese trading company. The letter indicated that A was to sell B a set of farming machines for $50,000, and that A was to deliver the machine to B by May 10 and that B was to pay the price to A by May 20.

(2) On April 8, the letter reached B, the branch office in Hamburg.

(3) On April 9, B made a telephone call to A. "The offer is accepted." Then, B said to A, "I would like to withdraw my offer."

(4) On May 1, A finally handed the farming machine over to a Japanese container ship at the port of New York.

(5) On May 31, the machine was delivered to the branch office in Hamburg.

(6) On June 5, B examined the machine.

(7) On May 10, B paid the price of $50,000 to A.

(8) On August 10, the machine proved to out of order because of a faulty connection gear. B immediately notified A specifying the nature of the problem.

(9) On September 1, B asked A to repair the problem within one month. A did not repair it until October 1.

(10)On October 10, B declared the contract void.

(11)On December 10, A recovered damages and B restituted the machine delivered by A.

(12)On December 20, A estitute the price paid by B.

6 Ref.4), p.109, proposed the concept of "basic norm." Note that my basic legal rule sentence does not always coincide with Kelsen's concept. They differ in that Kelsen starts on legal norms as a meaning, while I start on legal rule sentences; Kelsen's basic norm is conceived of as a norm that gives the ground of the validity of constitution or convention as a given positive law, while my theory presents both such a basic legal rule sentence and fundamental rules always applied where the validity of a legal sentence is to be decided. This has become the case of our logical analysis of legal systems and legal reasoning.

The following questions are set as examples:

[Question]

At each of the points in time below, what is the legal relation between A and B?

1: April 5
2: April 15
3: May 5
4: August 15
5: September 15
6: October 5
7: November 15
8: December 15
9: December 25

The following CISG articles apply:

Article 15

(1) An offer becomes effective when it reaches the offeree.
(2) An offer, even if it is irrevocable, may be withdrawn if the withdrawal reaches the offeree before or at the same time as the offer.

Article 16

(1) Until a contract is concluded an offer may be revoked if the revocation reaches the offeree before he has dispatched an acceptance.

Article 18

(2) An acceptance of an offer becomes effective at the moment the indication of assent reaches the offeror.

Article 23

A contract is concluded at the moment an acceptance of an offer becomes effective in accordance with the provisions of this Convention.

Article 31

If the seller is not bound to deliver the goods at any other particular place, his obligation to deliver consists:

(a) if the contract of sale involves carriage of the goods - in handing the goods over to the first carrier for transmission to the buyer;

Article 38

(1) The buyer must examine the goods, or cause them to be examined, within as short a period as is practicable in the circumstances.

Article 39

(1) The buyer loses the right to rely on a lack of conformity of the goods if he does not give notice to the seller specifying the nature of the lack of conformity within a reasonable time after he has discovered it or ought to have discovered it.

Article 45

(1) If the seller fails to perform any of his obligations under the contract or this Convention, the buyer may:
   (a) exercise the rights provided in articles 46 to 52;
   (b) claim damages as provided in articles 74 to 77.
(2) The buyer is not deprived of any right he may have to claim damages by exercising his right to other remedies.

Article 46

(1) The buyer may require performance by the seller of his obligations unless the buyer has resorted to a remedy which is inconsistent with this requirement.
(2) If the goods do not conform with the contract, the buyer may require delivery of substitute goods only if the lack of conformity constitutes a fundamental breach of contract and a request for substitute goods is made either in conjunction with notice given under article 39 or within a reasonable time thereafter.
(3) If the goods do not conform with the contract, the buyer may require the seller to remedy the lack of conformity by repair, unless this is unreasonable having regard to all the circumstances. A request for repair must be made either in conjunction with notice given under article 39 or within a reasonable time thereafter.

Article 47

(1) The buyer may fix an additional period of reasonable length for performance by the seller of obligations.

Article 49

(1) The buyer may declare the contract avoided:
   (a) if the failure by the seller to perform any of his obligations under the contract or this Convention amounts to a fundamental breach of contract; or
   (b) in case of non-delivery, if the seller does not deliver the goods within the additional period of time fixed by the buyer in accordance with paragraph (1) of article 47 or declares that he will not deliver within the period so fixed.

[Solution]

1) On April 5, there is no legal relation between seller A and buyer B.
2) On April 15, A has a duty to deliver the farming machine to B by May 10 and B has a duty to pay the price of $50,000 to A by May 20, while B has the right to require A to deliver goods to B and A has the right to require B to pay the price to A by May 10.
3) On May 5, B has a duty to pay the price of $50,000 to A by May 20, while A has the right to require B to pay the price to A by May 10.
4) On August 15, A has the duty to recover the damage, while B has the right to claim A for damage and B has the right to require A to repair the machine.
5) On September 15, A has the duty to recover the damage and a duty to repair the machine, while B has the right to claim damage from A and B has the right to require that A repair the machine, restricted to exercise.
6) On October 5, A has the duty to recover the damage and to repair the machine, while B has the right to claim damage from A, B has the right to require A to repair the machine and B has the right to declare the contract void.
7) On November 15, A has the duty to recover the damage and the duty to restitute the price paid by B, and B has the duty to restitute the machine delivered by A, while B has the right to claim damage for A and the right to require A to restitute the price, and A has the right to require B to restitute the machine.
8) On December 15, A has the duty to restitute the price paid by B, while B has the right to require A to restitute the price.

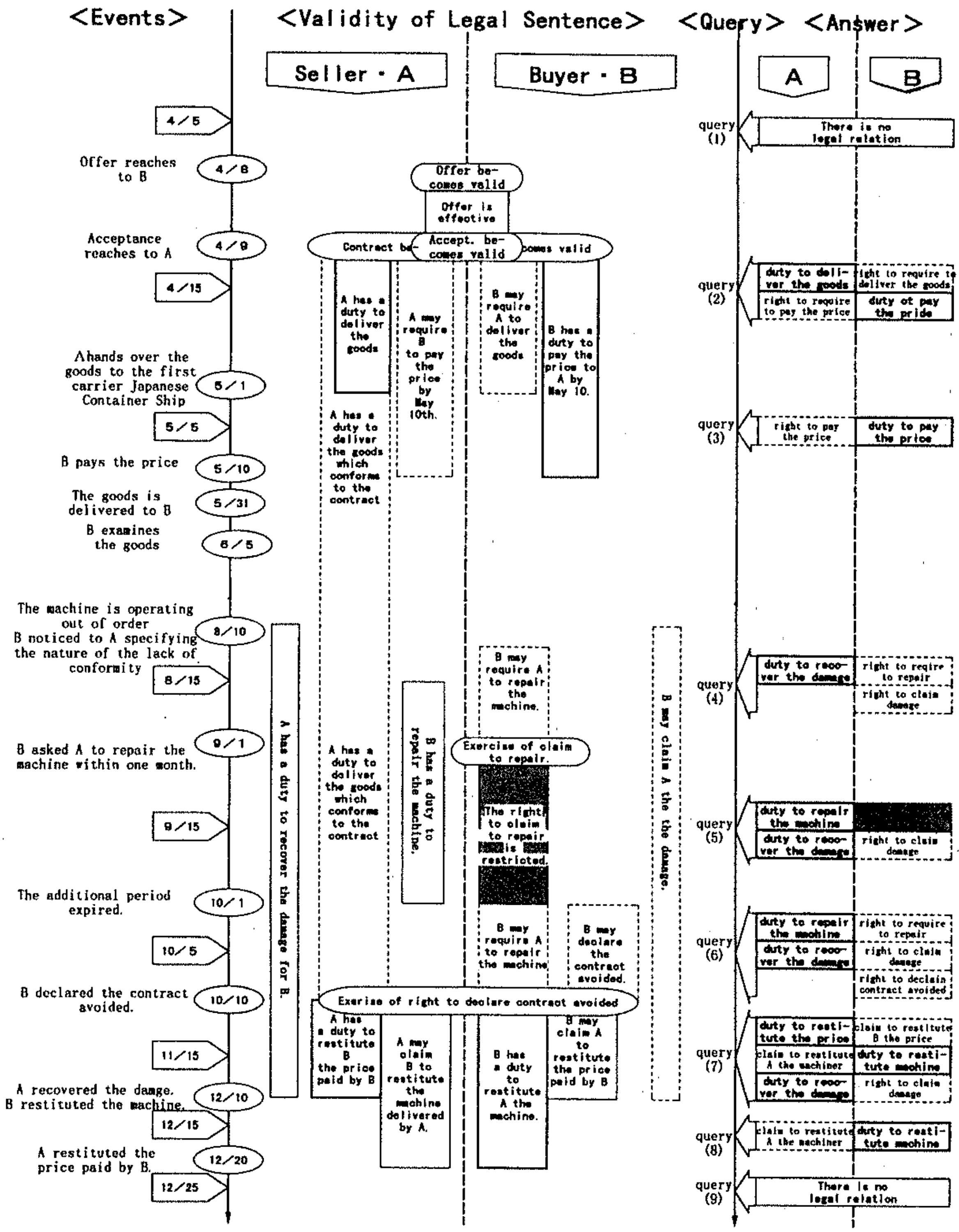


Fig. 3. Changes in Legal Relations.

The above solutions correspond to obligation and right. In this chart, the existence of legal relations is indicated by the rectangle zones of the validity of legal sentences which describe obligations and rights.

9) On December 25, there is no legal relation between A and B on the contract.

The changes of legal relation according to the time progress in case 7h are shown in **Fig.3**.

The knowledge structure enabling deduction of the above solutions, or enabling the formation of rectangle zones of legal relations is to be clarified below.

5. Logical Structure of Contract Law Regulating Changes in Legal Relations

In Logical Jurisprudence, the existence of an obligation means that a legal object sentence describing the obligation is valid as mentioned above. The existence of A's obligation to deliver a farming machine to B means that "A has an obligation to deliver a farming machine to B'' or ''It is obligatory for A to deliver a farming machine to B'' is valid. If the parties have an obligation to deliver a farming machine to B based on a contract, it is so because the sentences in the contract describing the obligation (that is, legal object sentences) are valid as proved. The contract law is a set of legal meta rule sentences that regulates the validity of the legal object sentences of the contract. Below, We show what legal meta rule sentences work to prove the validity of legal object sentences related to contracts and how they do so.

5.1. Legal Rule Sentences Deciding that Legal Sentences are Valid.

The following fundamental legal meta rule sentence is valid for confirming that legal sentences are valid:[7]

> *(mrl) ''A legal sentence S is valid at the time T if and only if S becomes valid at time T1 before T, and S is not terminated until T.''*

This legal rule sentence cannot be found as a statutory text in the CISG or other regulations. This is a fundamental legal meta rule sentence implicitly taken for granted by the CISG and all other regulations. Without this rule, no statutory legal sentence works when it comes to application. This rule is the most fundamental among legal meta rules enabling us to put mere collection of legal sentences into a legal system. This rule applies to every case where the validity of legal sentences is considered.

In deciding, for example, whether legal sentence ''A has an obligation to deliver the machine to B on April 15'' is valid, we apply this rule and examine its two specified requirements: '''A has an obligation to deliver the machine to B' becomes valid before April 15'' and '''A has an obligation to deliver the machine to B' is not terminated until April 15.'' If both requirements are satisfied, then the legal object sentence is valid in April 15. Therefore, A's obligation to deliver the machine exists in the prevailing usage of legal language; if not, it is not valid, and therefore the obligation does not exist.

How are legal sentences to be systematized under this fundamental legal meta rule sentence? All other legal meta rule sentences are systematized as subrules of this sentence, as rules to decide whether the two different requirements of this fundamental meta rule sentence, i.e. ''the legal sentence becomes valid'' and ''the legal sentence is not terminated,'' are satisfied.[8]

Now, we shall clarify the structure of legal knowledge deciding these two factors, i.e. ''the legal sentence becomes valid'' and ''the legal sentence is not terminated'' focusing on the validity of legal object sentences to make the logical structure of legal knowledge regulating changes of legal obligation clear. Here, note the following: ''The legal sentence is not terminated'' means ''it is not the case that the legal sentence is terminated.'' In the real legal world, there is no rule that decides directly ''a legal sentence is not terminated,'' but there exist many legal rule sentences that decide ''a legal sentence is terminated.'' The negation of the sentence ''a legal sentence is terminated'' is conceived of as proven in fact if the later sentence fails to be proven.

5.2. Logical Structure of Contract Law Deciding Accrual of Obligation

Legal obligations accrue because legal object rule sentences become valid, as mentioned above.

5.2.1. Accrual of validity of elementary legal sentences with accrual of contract validity

The accrual of validity of a complex legal sentence follows the accrual of validity of elementary legal sentences belonging to it. The following legal meta rule sentence is presupposed:

> *(r01) become_valid(ES,G,T) ←*
> *element_complex_sentence(ES,CS) &*
> *become_valid(CS,G,T)*

This rule is read: A legal sentence ES becomes valid for goal G at time T, if ES is an element sentence of complex sentence CS and CS becomes valid for G at T.

Consider, for example, the change in the legal relation on April 9 in Fig.3. As the contract as a complex legal sentence has become valid, the following two obligation sentences (legal object sentences) as elementary legal sentences of the contract, become valid: ''A has an obligation to deliver the machine to B'' and ''B has an obligation to pay the price A by May 20.'' The main part of contract law is legal meta rule sentences regulating changes of validity of the contract itself as a complex legal sentence, i.e., the accrual and termination of its validity.

Figure 4 is a logical flowchart of the legal rule sentence that decides the accrual of validity of contract. 3AA1BA in Fig.4[9] means that the contract is concluded. The ''conclusion'' of the contract means that it is formed as a legal sentences named contract. Legal sentences differ from conventional sentences because legal sentences satisfy the requirements of legal meta rules prescribing the formation of the relevant legal sentences such as contracts, judgements, statutes, constitutions, and conventions.

Part 2 of CISG regulates in detail the conclusion of contracts from Articles 14 through 24. To bring them into a unified system, however, we need a legal rule sentence such as that in **Fig.5**.

This rule is related to Article 23, but is not the same. The article does not refer to the effectiveness of an offer directly. For Articles 14 through 17 to be systematized, the first requirement must be met. This legal rule sentence therefore [2A] (Fig.5) is a legal principle of contract law.[10] (This rule would be valid for CISG and also for other contract laws.) Articles 14 through 17 and 24 in part 2 are to systematized as a subrule of the first requirement [2AA] of this legal rule sentence. Articles 18 through 22 and 24 in part 2 are systematized as a subrule of the second requirement [2AB].

7 The validity of this fundamental legal meta rule is a fact, or is presupposed always valid. In our knowledge base, a sentence that describes this *mrl* is valid is set as a legal fact sentence.

8 Thus, all legal meta rules in this sense contribute to regulating the validity of legal sentences.

9 For knowledge representation of law by logical flowcharts, refer to Refs.16) and 17).

10 This legal requirement is defined and the inference process of the discovery formalized in Ref.7).

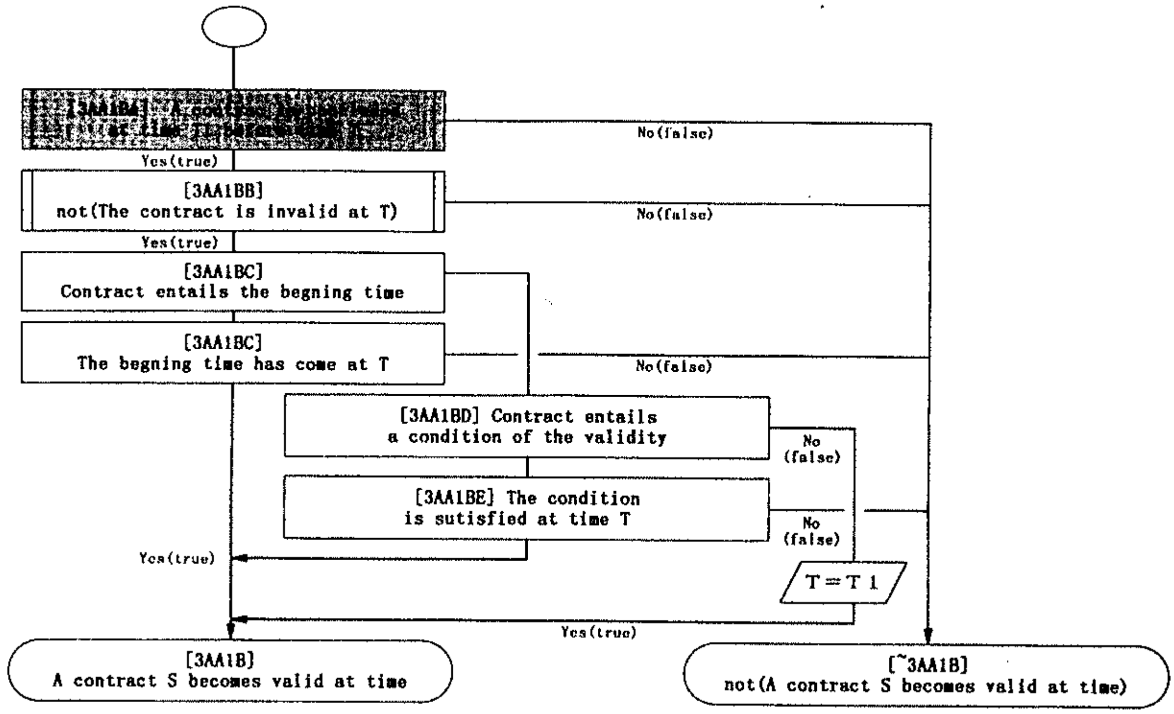


Fig. 4. 3AA1B A contract becomes valid.

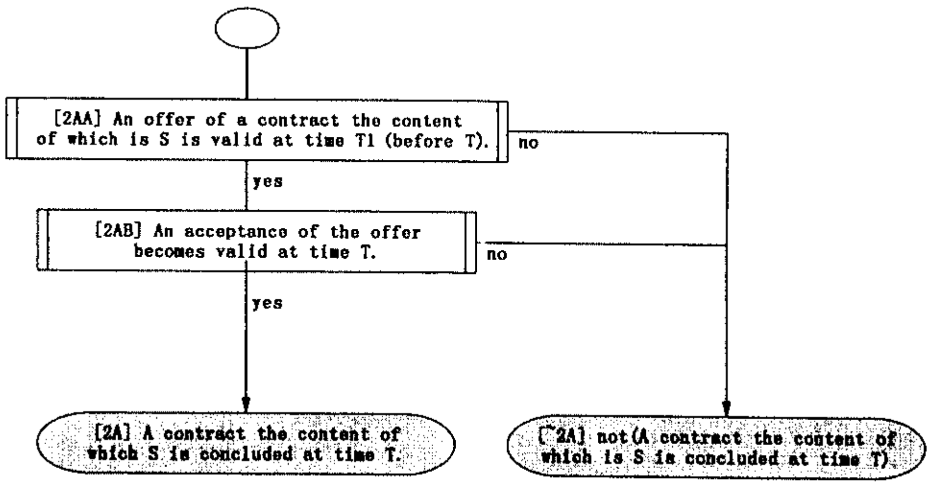


Fig. 5. [2A] Contract is concluded.

5.2.2. Accrual of validity of a legal sentence by exercising rights

In some cases, the accrual of validity of the elementary legal sentence by itself, not as a result of the accrual of contract validity, is regulated. An obligation accrues, for example, along with exercise of the relevant right. In Figure 3, the legal sentence "B has an obligation to repair the machine for A" becomes valid because A exercised the right to require the repair of the machine on September 1.

Logical Jurisprudence does not consider sentences describing rights as a legal object sentence as in the prevailing opinion in legal theories, but as a legal meta rule sentence, as described above. That a person has a right to require another person to do Z, for example, means, in our opinion, that the person may arrive at a legal object sentence concluding that the other person is obligated to do Z.

The legal meta rule sentence below must be valid.

(3AA2) "A legal sentence 'X has an obligation to do Z' becomes valid at time T, if a legal sentence 'Y has a right to require X to do Z' is valid at time T, and Y exercises the right to require X to do Z at time T."

The accrual of seller A's concrete obligation to repair the machine on September 1. For example, in Fig.3, for the present case is deduced by the application of this rule. The proof is as follows: The second requirement of the rule "*Y exercises the right to require X to do Z at time T*" is satisfied by buyer B's exercise of the right to require seller A to remedy the problem by repair on September 1. The instantiated first requirement, "*'Buyer B has a right to require seller A to remedy the lack of conformity by repair on September 1' is valid," is proved by applying the fundamental meta rule mr1*. The instantiated first condition of the latter rule "'*Buyer B has a right to require seller A to remedy the lack of conformity by repair' becomes valid on August 10*" is proved by applying the following legal rule sentence representing Article 46 of CISG:

> (*rCISG46*): "*The buyer has a right to require the seller to remedy the lack of conformity by repair*" *becomes valid, if the goods do not conform to the contract.*

The requirement of rule *rCISG46* is satisfied by fact (8) on August 10. The instantiated second requirement of the applied mr1 "'*B has a right to repair the machine' is not terminated until September 1*" *is proven because the proof of* "'*B has a right to repair the machine' is terminated until September 1st*'" is false.

The deductive system of legal knowledge to deduce an accrual of the validity of an legal object sentence by exercising a right of claim is explicated in an example of the claim to repair goods delivered. Legal meta rule sentence *3AA2* applies to many other cases such as accruals of the seller's duty to perform obligations (Article 46(1)), to deliver substitute goods (46(2)), and so on.

Many statutory legal rule sentences regulate the accrual of validity an legal object directly. In such a case, one need apply the relevant statutory legal rule sentences, not *3AA2*.

5.3. Logical structure of contract law deciding termination of obligations

The termination of obligations means that the validity of legal object sentences describing obligations is terminated. There are two ways to terminate the validity of elementary legal object sentences: the termination of their validity along with the termination of the complex legal sentence and the termination of their validity by themselves.

5.3.1. Termination of elementary legal sentence validity and contract termination

The validity of elementary legal sentences is terminated if the complex legal sentence to which they belong is terminated. The validity elementary sentences of a contract are terminated if the validity of the contract as a complex legal sentence is terminated.

Complex legal sentences lose their validity on the day when a fixed term expires, when the termination condition is met or when contract avoidance becomes effective. Regulations concerned with these factors can be integrated as a legal rule sentence, which makes concrete the second requirement of the fundamental legal meta rule sentence *mrl* as its subrule sentence.

In Fig.3, two legal object rule sentences, "A has an obligation to B that the machine delivered conform to the contract" and "A has an obligation to B to repair the machine" is terminated on October 1, because the validity of the contract as a complex legal sentence was terminated owing to B's exercise of the right to declare the contract avoided when B has the right, i.e. 'B has the right to declare the contract avoided' is valid. The right to declare the contract void resulted from the fact that the seller had not fulfill an obligation to repair the machine within the additional period of time (one month) fixed by the buyer.[11]

5.3.2. Termination of validity elementary legal object sentences with fulfillment of obligation

In some cases, the validity of one article of a contract is terminated independently of the validity of the whole contract. The following legal meta rule sentence is valid:

> (*mr4b*) "*The validity of elementary legal object sentences is terminated when the obligation is fulfilled.*"

Because of the delivery by A on May 1, for example, the validity of the legal object sentence "A has an obligation to deliver the machine to B" is terminated May 1, and because of payment by B on May 20, the validity of legal sentence "B has an obligation to pay the price by May 20" is terminated May 20. These terminations of obligations are deduced by applying the above legal meta rule sentence *mr4b*.

6. Conclusion

This research comfirmed the structure of contract law by taking up CISG as an example and focusing on the systematization of law from the view of Logical Jurisprudence. By using three standards of legal sentences -- that is, legal fact sentences and legal rule sentences, complex legal sentences and elementary legal sentences, and legal object sentences and legal meta sentences -- we explicated the basic structure of legal knowledge enabling us to systematize contract law. Applying the frame to cases (case 7h here), we formalized the change of legal relation as a change of the validity of legal sentences that describe obligations. On formalization, we found the fundamental legal meta rule sentence under which every other legal meta rules are systematized. We thus clarified the logical structure of a contract law system that deductively proves the change of legal relations along with the progress of events in a concrete example.

The results of this study have been introduced to the knowledge base of the CISG. We have developed a knowledge base system by which solutions about legal states of affairs can be deduced at any time as a result of applying the CISG to a given international trade case.

Acknowledgments:

This report is part of the research result of the joint research project "Development of the Legal Expert System - Clarification of Legal Knowledge Structure System and Realization of Legal Reasoning" (Legal Expert), funded by the Japanese Ministry of Science, Education and Culture. We thank the joint research team of the above project and the research team of civil law at Meiji Gakuin University.

References:

1) Ashley, K.D., "Modeling Legal Argument: Reasoning with Cases and Hypotheticals," MA: MIT Press, Cambridge, (1990).
2) Hage, J.C., "Reasoning with Rules - an Essay on Legal Reasoning and Its Underlying Logic," Kluwer Academic Publishers, Dordrecht/Boston/London, (1997).
3) Hart, H.L.A., "The Concept of Law," Oxford University Press, London, (1961).
4) Kelsen, H., "Pure Theory of Law," Translation of 'Reine Rechsle-

11 This reasoning is made through the analogical application of article 49 (1)(b), discussed elsewhere.

hre'(second revised edition. 1960) by Knight, First published in 1934. University of California Press, Berkeley, (1978).
5) Kralingen. R.W.van, "Frame-based Conceptual Models of Statute Law," The Hague/London /Boston: Kluwer Law International, (1995).
6) Prakken, H., "Logical Tools for Modelling Legal Argument - A Study of Defeasible Reeasoning in Law," The Hague/London/Boston: Kluwer Law International, (1997).
7) Sakurai, S. & Yoshino, H., "Identification of Implicit Legal Requirements with Legal Abstract Knowledge," In Proceedings of the Fourth International Conference on Artificial Intelligence and Law, 298-305. Amsterdam: ACM Press, (1993).
8) Sergot, M. J., Sadri, F., Kowalski, R. A., Kriwaczek, F., Hammond, P. and Cory, H. T., "The British Nationality Act as a Logic Program," Communications of the ACM 19-5, 370-386, (1986).
9) Yoshino, H., "Justice and Logic - Role of Deductive Methods in Reasoning on Justice," In: Justice. Annual of Legal Philosophy 1974 (Jajpanese) 38ff.. Tokyo: Yuhikaku Publisher, (1975).
10) Yoshino, H., "Ueber die Notwendigkeit einer besonderen Normenlogik als Methode der juristischen Logik," in Albert, H. et al., (hrsg.), Juristishe Logik, Zivil-und Prozessrecht, 140ff. Springer Verlag, Berlin-Heidelberg-New York, (1978).
11) Yoshino, H., "Die Logische Struktuer der Argumentation bei der Juristischen Entscheidung," in: Aarnio, Niiniluoto, Uusitalo(Hrsg.), Methodoligie und Erkentnistheorie der Argu- mentation, 235ff. Duncker Humblot Verlag, Berlin, (1981).
12) Yoshino, H, "Zur Anwendbarkeit der Regeln der Logik auf Rechtsnormen," in Walter(hrsg.), Die Reine Rechtslehre in wissenschaftlicher Disskussion, 142ff. Manz Verlag, Wien, (1982).
13) Yoshino, H. et. al., "Legal Expert System LES-2," in Wada(ed.), Logic Programming '86, 107-108, Springer Verlag, Berlin-Heidelberg-New York-London-Paris-Tokyo, (1987).
14) Yoshino, H., "About the Applicability of the Principles of Logic to Legal Norm," in: Keio Law Journal, 62-12, 512-472, (1993).
15) Yoshino, H. Haraguchi, M., Kagayama, S. and Sakurai, S., "Towards a Legal Analogical Reasoning System Knowledge Representation and Reasoning Methods," in Proceedings of the Fourth International Conference on Artificial Intelligence and Law, ACM (Associaton for Computing Machinery), 110-116, (1993).
16) Yoshino, H., "Representation of Legal Knowledge by Logic Flowchart and CPF," In K.Nitta. H.Tsuda & K.Yokota (eds.) ICOT Technical Memoradum:TM-1298 ' Workshop on Knowledge Representation for Legal Reasoning .143ff. (1994).
17) Yoshino, H., "Logical Flow Charts of Part 2 of the CISG," In Yoshino, H. (ed.). Development of the Legal Expert System - Clarification of Legal and Realization of Legal Reasoning, the Reports of the Result of the Research of the Legal Expert Project in 1993 (Japanese), 129-151, (1994).
18) Yoshino, H. and Wada, S., "The representation of Part 2 'FORMATION OF THE CONTRACT of the CISG by Logical Flow Charts' the CISG," In Yoshino, H. (ed.), Development of the Legal Expert System - Clarification of Legal and Realization of Legal Reasoning, the Reports of the Result of the Research of the Legal Expert Project in 1994 (Japanese), 165-180, (1995).
19) Yoshino H., "The Knowledge Structure of Contract Law," In Yoshino, H.(ed.), Development of the Legal Expert System - Clarification of Legal and Realization of Legal Reasoning, the Reports of the Result of the Research of the Legal Expert Project in 1994 (Japanese), 98-112, (1995).
20) Yoshino, H., "The Systematization of Legal Meta-inference," in: Proceedings of the Fifth International Conference of Artificial Intelligence and Law, 266-275, ACM, Amsterdam, Netherlands, (1995).
21) Yoshino H., "Logical Jurisprudence as a Basic Theory for Clarifying the Structure of Legal Knowledge," In Yoshino, H.(ed.), Development of the Legal Expert System - Clarification of Legal and Realization of Legal Reasoning, the Reports of the Result of the Research of the Legal Expert Project in 1995 (Japanese), 83-93, (1996).
22) Yoshino H., "The Structure of Changes of Legal Relation on the CISG," In Yoshino, H.(ed.), Development of the Legal Expert System - Clarification of Legal and Realization of Legal Reasoning, the Reports of the Result of the Research of the Legal Expert Project in 1995 (Japanese), 94-105, (1996).
23) Yoshino H., "The Structure of Contract Law and the Knowledge Base - on an Example of the Application of the CISG to Case 7f," In Yoshino, H.(ed.), Development of the Legal Expert System - Clarification of Legal and Realization of Legal Reasoning, the Reports of the Result of the Research of the Legal Expert Project in 1996 (Japanese), 94-108, (1997).
24) Yoshino H., "On the Logical Structure of Contract Law - Focusing on the CISG," In Ashley, K.D & Yoshino, H.(eds.) Proceedings of the Fourth International Workshop on a Legal Expert System for the CISG, 43-53, LESA, Melbourne, Australia, (1997).
25) WPaper

Name:
Hajime Yoshino

Affiliation:
Professor of law, Meiji Gakuin University Faculty of law

Address:
1-2-37 Shirokanedai, Minato-ku, Tokyo 108, Japan

Brief Biographical History:

1972- Assoc. Professor at Meiji Gakuin University Faculty of law, Tokyo
1975- Professor at Meiji Gakuin University Faculty of law, Tokyo

Main works:

- Foundation of Legal Expert Systems, Gyosei Press (1986), (Japanese).
- "Towards a Legal Analogical Reasoning System: Knowledge Representation and Reasoning Methods," in Proc. The Fourth International Conference on Artificial Intelligence and Law, ACM, pp.110-116, (1993).
- "The Systematization of Legal Meta-inference," in Proc. The Fifth International Conference on Artificial Intelligence and Law, ACM, pp. 266-275, (1995).
- "On the Logical Foundation of Compound Predicate Formulae for Legal Knowledge Representation," in Artificial Intelligence and Law, vol.5, Nos.1-2, pp.77-96 (1997).

Paper:

A Framework for Nonmonotonic Reasoning with Rule Priorities

Masato Shibasaki and Katsumi Nitta

Department of Computational Intelligence and Systems Science,
Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku, Yokohama 226, Japan
E-mail: nitta@dis.titech.ac.jp

[Received November 30, 1997; accepted January 10, 1998]

In the 1990, a number of studies was made on nonmonotonic reasoning with rule priorities. Little is known, however, about relationships among these semantics because there is no framework in which these semantics can be compared. In this paper, we give the basis of this framework, which is a special form of Dung's argumentation framework, although not covering all semantics of past studies in this category.
To be concrete, we provide rule-based framework (RF) for extended logic programs (ELPs), clarify semantics of default rules and rule priorities, and extend it to RF for prioritized extended default logic programs (EDLPs). By means of RF for prioritized EDLPs, we reformulate several semantics of past studies , indicate their improvements, and give new prioritized EDLPs semantics.

Keywords: Argumentation, Defeasible reasoning, Extended logic programming, Nonmonotonic reasoning, Rule prioritization

1. Introduction

In the 1990, a number of studies was made on nonmonotonic reasoning with rule priorities . They may be divided into four types. The first is adding rule priorities to extended logic programs (ELPs).[6)] The second is adding rule priorities to default theories.[2,4,5,7)] The third is adding rule priorities to circumscription.[11)] The fourth is proposing individual semantics for default rule sets with rule priorities.[1,8,10,12,13,15)]

Little is known, however, about relationships among these semantics because there is no framework in which these semantics can be compared.

In this paper, we give the basis of this framework, which is a special form of Dung's argumentation framework, which has not covered all semantics in the above literature.

This paper is organized as follows: In section 2, we explain Dung's argumentation framework, then give rule-based framework (RF) for ELPs. Then, we extend RF by defining semantics of default rules and rule priorities. In section 3, we reformulate several semantics of nonmonotonic reasoning with rule priorities on RF, then discuss improvements of these semantics and finally propose new semantics. In section 4, we state related work, and in section 5, we give concluding remarks.

2. RF for Prioritized EDLP

2.1. Argumentation Framework

We explain argumentation framework and its semantics according to Ref.9). First, we show the definition of argumentation framework, conflict-free, acceptable, and admissible.

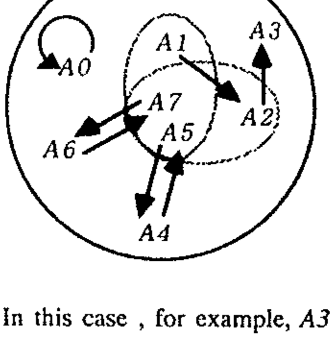
Definition 2.1.1 (*Argumentation Framework*)
An argumentation framework is a pair $AF = (AR, AR\text{-}attacks)$ where AR is a set of arguments, and $AR\text{-}attacks$ is a binary relation on AR, i.e., $AR\text{-}attacks \subseteq AR \times AR$.

Definition 2.1.2 (*Conflict-free*)
A set AS of arguments is *conflict-free*
iff $\forall A \in AS\ \forall A' \in AS\ \ (A, A') \notin AR\text{-}attacks$

Definition 2.1.3 (*Acceptable*)
An argument A is acceptable w.r.t. a set AS of arguments
iff $\forall A' \in AR\ \exists A1 \in AS$ if $(A', A) \in AR\text{-}attacks$
then $(A1, A') \in AR\text{-}attacks$.

Definition 2.1.4 (*Admissible*)
A set AS of arguments is *admissible*
iff AS is conflict-free and each argument in AS is acceptable w.r.t. AS.

Example 2.1.1
An argumentation framework $(AR, AR\text{-}attacks)$ is as follows :
$AR = \{A0, A1, A2, A3, A4, A5, A6, A7\}$ and
AR-attacks $= \{(A0, A0), (A1, A2), (A2, A3), (A4, A5), (A5, A4), (A6, A7), (A7, A6)\}$.



In this case , for example, $A3$ is acceptable w.r.t. $\{A1,$

A5, A7}, {*A1, A5, A7*} is admissible and {*A2, A5, A7*} is, not admissible , conflict-free.

Argumentation framework has four kinds of semantics. We show the definition of these semantics and their relationships below.

Definition 2.1.5 (*Preferred Extension*)
A set *AS* of arguments is a *preferred extension* of *AF* iff *AS* is a maximal (w.r.t. set inclusion) admissible set of *AF*.

Corollary 2.1.1
Every argumentation framework possesses at least one preferred extension.

Definition 2.1.6 (*Stable Extension*)
A set *AS* of arguments is a *stable extension* of *AF* iff *AS* is conflict-free and $\forall A' \in AR \setminus AS\ \exists A \in AS\ (A, A') \in AR\text{-}attacks$.

Lemma 2.1.2
Every stable extension is a preferred extension, but not vice versa.

Definition 2.1.7 (*Grounded Extension*)
A set *AS* of arguments is a *grounded extension* of *AF* iff *AS* is the least fix point of F_{AF}.
F_{AF} is defined as follows:

$F_{AF}: 2^{AR} \rightarrow 2^{AR}$

$F_{AF}(S) = \{A \in AR \mid A \text{ is acceptable w.r.t. } S\}$

Definition 2.1.8 (*Complete Extension*)
A set *AS* of arguments is a *complete extension* of *AF* iff *AS* is admissible and each argument, which is acceptable w.r.t. *AS*, belongs to *AS*.

Theorem 2.1.3
(1) *Each preferred extension is a complete extension , but not vice versa.*
(2) *The grounded extension is the least (w.r.t. set inclusion) complete extension.*
(3) *The complete extensions form a complete semilattice w.r.t. set inclusion.*

Example 2.1.1
This argumentation framework possesses nine complete extensions ({*A1,A3,A4,A6*}, {*A1,A3,A4,A7*}, {*A1,A3, A5,A6*}, {*A1,A3,A5,A7*}, {*A1,A3,A4*}, {*A1,A3,A5*}, {*A1, A3, A6*}, {*A1, A3, A7*}, {*A1, A3*}), four preferred extensions ({*A1,A3,A4,A6*}, {*A1,A3,A4,A7*}, {*A1,A3, A5, A6*}, {*A1, A3, A5, A7*}), one grounded extension ({*A1,A3*}), and no stable extension. If *A0* is not included in *AR*, each preferred extension is stable.

2.2. RF for ELPs

An ELP *P* is a finite set of rules of the form

$$L_0 \leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots, \sim L_n.$$

where L_i is a literal. G_P denotes the set of all ground instances of rules in *P*. We call $\sim L_i$ in rule body assumption. $De\text{-}rule_P$ denotes the set of all defeasible rules, which are rules including assumption, in G_P and $St\text{-}rule_P$ denotes the set of all strict rules, which are rules not including assumption, in G_P. *As*(*R*), Head(R) and Body(R) denotes the set of assumptions, the head literal and the body literals of rule *R*.

First, we define Cl^P, Cn^P and $G\text{-}attacks_P$, preparing for defining RF for ELPs.

Definition 2.2.1 ($Cl^P(S)$, $Cn^P(S)$)
Let *P* be an ELP and *S* be a subset of $De\text{-}rule_P$. $Cl^P(S)$ (resp. $Cn^P(S)$) is a minimal set of literals that is (resp. logically) closed under the rule set, which is gained by deleting rules, which has assumptions other than assumptions of *S* from *P*, and removing assumptions from remaining rule set.

We write *Cl*(*S*) for $Cl^P(S)$ and *Cn*(*S*) for $Cn^P(S)$, when *P* is obvious.

$S \vdash L$ means $L \in Cl(S)$.
$S \vdash_{min} L$ means *S* is a minimal set such that $S \vdash L$.
$S \models L$ means $L \in Cn(S)$.
$S \models_{min} L$ means *S* is a minimal set such that $S \models L$.

Definition 2.2.2 ($G\text{-}attacks_P$)
Let *S* be a subset of $De\text{-}rule_P$ and *R* belong to $De\text{-}rule_P$.
$(S, R) \in G\text{-}attacks_P$ iff $\exists \sim L \in As(R)\ S \vdash_{min} L$.

"$(S, R) \in G\text{-}attacks_P$" means that *S* is a minimal set of defeasible rules such that if all assumptions of *S* are true, then some assumption of *R* is false.

We define RF for ELPs, which represents a special form of argumentation framework. We regard a set of defeasible rules as an argument so as to define semantics of rule priorities in argumentation framework.

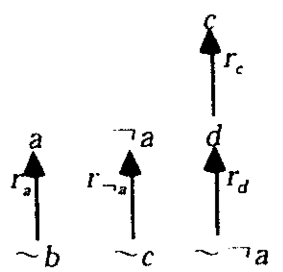
Definition 2.2.3 (*RF for ELPs*)
A RF for an ELP *P* is a pair

$RF(P) = (De\text{-}rule_P, attacks_P)$

where $De\text{-}rule_P$ is the set of all defeasible rules in G_P and $attacks_P$ equals $G\text{-}attacks_P$.

We write RF for ELPs as RF_P.

Example 2.2.1
$P = \{r_a: a \leftarrow \sim b.,\ r_{\neg a}: \neg a \leftarrow \sim c.,\ r_c: c \leftarrow d.,\ r_d: d \leftarrow \sim\neg a.\}$



$RF(P) = (\{r_a, r_{\neg a}, r_d\},$
$\{(\{r_{\neg a}\}, r_d), (\{r_d\}, r_{\neg a})\})$

Note: In examples of this paper we append a rule name to each rule to indica te the rule easily.

The above RF_P represents an argumentation framework $AF(P) = (AR_P, AR\text{-}attacks_P)$, where AR_P is the set of all arguments, argument is a set of defeasible rules, and $AR\text{-}attacks_P$ equals $\{(S, S')\in AR_P\times AR_P \mid \exists R'\in S'\ \exists S1 \subseteq S\ (S1, R')\in attacks_P\}$.

We define conflict-free, acceptable, and admissible in RF_P, so that they are equivalent to these definitions in subsection 2.1. These equivalences are expressed by Theorem 2.2.1~2.2.3.

Definition 2.2.4 (*Conflict-free in RF_P*)
A set S of defeasible rules is *conflict-free in RF(P)* iff $\forall S1\subseteq S\ \forall R\in S\ (S1, R)\notin attacks_P$.

Definition 2.2.5 (*Acceptable in RF_P*)
A defeasible rule R is *acceptable* w.r.t. a set S of defeasible rules *in RF(P)*
iff $\forall S'\in De\text{-}rule_P$ if $(S', R)\in attacks_P$ then
$\exists R'\in S'\ \exists S1\subseteq S\quad (S1, R')\in attacks_P$

Definition 2.2.6 (*Admissible in RF_P*)
A set S of defeasible rules is *admissible in RF(P)*
iff S is conflict-free and each argument in S is acceptable w.r.t. S in $RF(P)$.

Example 2.2.1
For example $\{r_{\neg a}, r_d\}$ is not conflict-free, r_d is acceptable w.r.t. $\{r_a, r_d\}$ and $\{r_a, r_d\}$ is admissible, in $RF(P)$.

We define preferred extension, stable extension, grounded extension, and complete extension in RF_P, so that they are equivalent to these definitions in subsection 2.1. These equivalences are expressed by Theorem 2.2.4.

Definition 2.2.7 (*Preferred Extension of RF_P*)
A set S of defeasible rules is a *preferred extension of RF(P)*
iff S is a maximal admissible set of RF(P).

Definition 2.2.8 (*Stable Extension of RF_P*)
A set S of defeasible rules is a *stable extension of RF(P)*
iff S is conflict-free in RF(P) and
$\forall R'\in De\text{-}rule_P\backslash S\ \exists S1\subseteq S\ (S1, R')\in attacks_P$.

Definition 2.2.9 (*Grounded Extension of RF_P*)
A set S of defeasible rules is a *grounded extension of RF(P)*
iff S is the least fix point of $F_{RF(P)}$.
$F_{RF(P)}$ is defined as follows.

$$F_{RF(P)} : 2^{De\text{-}rulep} \rightarrow 2^{De\text{-}rulep}$$
$$F_{RF(P)}(S) = \{R\in De\text{-}rule_P \mid R \text{ is acceptable w.r.t. } S \text{ in } RF(P)\}$$

Definition 2.2.10 (*Complete Extension of RF_P*)
A set S of defeasible rules is a *complete extension of RF(P)*
iff S is admissible in $RF(P)$ and each defeasible rule, which is acceptable w.r.t. S in $RF(P)$, belongs to S.

Assuming that P is an ELP and S is a subset of $De\text{-}rule_P$, the theorems below can be concluded. Due to spaces limitation, their proofs are omitted here.

Theorem 2.2.1
(1) $\exists AS\subseteq 2^{De\text{-}rulep}$ *s.t.* $\cup AS=S$ $AS\cup S$ *is conflict-free in AF(P) iff 2^S is conflict-free in AF(P).*
(2) *2^S is conflict-free in AF(P) iff S is conflict-free in RF(P).*

Theorem 2.2.2
(1) *Let AS be a set of arguments, which are sets of defeasible rules, and $\cup AS$ equal S. If an argument S0 is acceptable w.r.t. AS on AF(P) then S0 is acceptable w.r.t. 2^S in AF(P).*
(2) *An argument S0 is acceptable w.r.t. 2^S in AF(P) iff $\forall R\in S0$ R is acceptable w.r.t. S in RF(P).*

Theorem 2.2.3
(1) *Let AS be a set of arguments, which are sets of defeasible rules, and $\cup AS$ equal S. If AS is admissible in AF(P) then 2^S is admissible in AF(P).*
(2) *2^S is admissible in AF(P) iff S is admissible in RF(P).*

Theorem 2.2.4
(1) *Let AS be a set of arguments, which are sets of defeasible rules, and $\cup AS$ equal S. If AS is a preferred (stable, grounded, or complete) extension of AF(P) then AS = 2^S.*
(2) *2^S is a preferred (resp. stable, grounded, and complete) extension of AF(P) iff S is a preferred (resp. stable, grounded, and complete) extension of RF(P).*

Next, we consider the definitions of well-founded semantics and answer sets in RF_P.

Theorem 2.2.5
Let P be an ELP, WFS well-founded semantics[6] of P and GE a grounded extension of RF(P).*
In this case, WFS equals Cn(GE).*

Proof.
Let $f_\uparrow(\phi)$ be the least fix point of function f.
• Definition of *WFS**
Let X be a set of literals and P_X be the program obtained from P by deleting each rule having assumption $\sim L$ s.t. $L\in X$.
$\gamma^*_P(X) := Cl(P_X\backslash St\text{-}rule_P)$ $\gamma_P(X) := Cn(P_X\backslash St\text{-}rule_P)$
$\Gamma^*_P(X) := \gamma_P(\gamma^*_P(X))$ $WFS^* := \Gamma^*_{P\uparrow}(\phi)$
• Proof of $WFS^*=Cn(GE)$
Assume that $S\subseteq De\text{-}rule_P$ and S is admissible in $RF(P)$.
$\gamma^*_P(Cl(S)) = Cl(\{R\in De\text{-}rule_P \mid \forall S1\subseteq S\ (S1, R)\notin attacks_P\})$
$\Gamma^*_P(Cl(S)) = \gamma_P(\gamma^*_P(Cl(S)))$
$= Cn(\{R\in De\text{-}rule_P \mid \forall S'\in De\text{-}rule_P$ if $(S', R)\in attacks_P$ then $\exists R'\in S'\ \exists S1\subseteq S\ (S1, R')\in attacks_P\})$
$= Cn(F_{RF(P)}(S))$
$\Gamma^*{}_P^2(Cl(S)) = Cn(F_{RF(P)}{}^2(S))$
$\because$ if $Cn(F_{RF(P)}(S)) = Cl(F_{RF(P)}(S))$
then $\Gamma^*{}_P^2(Cl(S)) = Cn(F_{RF(P)}{}^2(S))$
otherwise $Cn(F_{RF(P)}(S)) = Lit$ then
$\Gamma^*{}_P^2(Cl(S)) = \Gamma^*_P(Lit) = Lit$ $Cn(F_{RF(P)}{}^2(S)) = Lit$

Similarly $\Gamma^*{}_P{}^n(Cl(S)) = Cn(F^n{}_{RF(P)}(S))$.
Since $\phi \subseteq Cl(\phi) \subseteq \Gamma^*{}_P(\phi)$
then $\Gamma^*{}_{P\uparrow}(\phi)=Cn(F_{RF(P)\uparrow}(\phi))$. □

Note: We write preferred extension as PE, stable extension SE, grounded extension GE, and complete extension CE.

Example 2.2.1
GE = $\{r_a\}$ Cn(GE) = a
PE = SE = $\{r_a, r_{\neg a}\}$, $\{r_a, r_d\}$
Cn(PE) = Cn(SE) = Lit, $\{a, c, d\}$
CE = $\{r_a\}$, $\{r_a, r_{\neg a}\}$, $\{r_a, r_d\}$
Cn(CE) = a, Lit, $\{a, c, d\}$

Cn(SE) is not an answer set. If we want to define answer sets in RF_P, $G\text{-}attacks_P$ must be replaced by $G\text{-}attacks^*{}_P$.

Definition 2.2.12 ($G\text{-}attacks^*{}_P$)
Let S be a subset of $De\text{-}rule_P$ and R belong to $De\text{-}rule_P$.
$(S, R) \in G\text{-}attacks^*{}_P$
iff $\exists \sim L \in As(R)\ S \models_{\min} L$.

When $G\text{-}attacks_P$ is replaced by $G\text{-}attacks^*{}_P$, $RF(P)$ is denoted by $RF^*(P)$.

Theorem 2.2.6
Ans is an answer set of an ELP P iff there is a stable extension SE of RF(P) such that Ans = Cn(SE).*

Proof.
$\Rightarrow$: $Ans = \gamma_P(Ans)$. Therefore, $P_{Ans} \backslash St\text{-}rule_P$ is conflict-free and $\forall R \in De\text{-}rule_P \backslash (P_{Ans} \backslash St\text{-}rule_P)\ \exists S \subseteq P_{Ans} \backslash St\text{-}rule_P\ (S, R) \in attacks_P$.
$\Leftarrow$: SE is conflict-free and $\forall R \in De\text{-}rule_P \backslash SE\ \exists S \subseteq SE\ (S, R) \in attacks_P$.
Therefore, $SE = P_{Cn(SE)} \backslash St\text{-}rule_P$. Then $Ans = \gamma_P(Ans)$. □

2.3. RF for EDLPs

We add to an ELP P default rules such that

$$L_0 \Leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots, \sim L_n.$$

where L_i is a literal. We call rules other than default rules exact rules, and call rules such that $L_0 \leftarrow L_1, \ldots, L_m$ strict rules and others defeasible rules. $Dt\text{-}rule_P$ denotes the set of all default rules in G_P.

We named finite sets of rules as follows:

(1) DP (Default Program)
:a finite set of rules of the form $L_0 \Leftarrow L_1, \ldots, L_m$.

(2) DPwN (Default Program with Negation as failure)
:a finite set of rules of the form $L_0 \Leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots, \sim L_n$.

(3) DLP (Default Logic Program)
:a finite set of rules of the form $L_0 \Leftarrow L_1, \ldots, L_m$. and $L_0 \leftarrow L_1, \ldots, L_m$.

(4) DLPwN (Default Logic Program with Negation as failure)
:a finite set of rules of the form $L_0 \Leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots, \sim L_n$. and $L_0 \leftarrow L_1, \ldots, L_m$.

(5) EDLP (Extended Default Logic Program)
:a finite set of rules of the form $L_0 \Leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots, \sim L_n$. and $L_0 \leftarrow L_1, \ldots, L_m, \sim L_{m+1}, \ldots, \sim L_n$.

We next discuss default rule semantics. We can consider that a default rule has an implicit assumption in its rule body. Clarifying the conditions in which implicit assumption come to be false yeilds semantics of a set of rules that has default rules. Therefore, assuming that S is a subset of $De\text{-}rule_P$ and R belongs to $Dt\text{-}rule_P$, we define $(S, R) \in D\text{-}attacks_P$ so that "S is a minimal set of defeasible rules such that if every explicit and implicit assumptions of S are true, then the implicit assumtion of R is false." Up to now, we have used assumption denoting an explicit assumption.
We define Cl^P, Cn^P, and $D\text{-}attacks_P$, preparing for defining RF for EDLPs.

Definition 2.3.1 ($Cl^P(S)$ $Cn^P(S)$)
Let P be an EDLP and S be a subset of $De\text{-}rule_P$. $Cl^P(S)$ (resp. $Cn^P(S)$) is a minimal set of literals that is (resp. logically) closed under the rule set, which is gained by deleting default rules other than S and exact rules, which has explicit assumptions other than explicit assumptions of S from P and removing (explicit and implicit) assumptions from remaining rule sets.

We write $Cl(S)$ shortly for $Cl^P(S)$, and $Cn(S)$ shortly for $Cn^P(S)$ when P is obvious.
$CR\text{-}set_P$ is the set of all closed rule sets of P and denotes $\{S \subseteq De\text{-}rule_P \mid$ There is no $S' \subset S$ such that $Cl(S) = Cl(S')\}$

Definition 2.3.2 ($D\text{-}attacks_P$)
Let S belong to $CR\text{-}set_P$ and R belong to $Dt\text{-}Rule_P$ $(S, R) \in D\text{-}attacks_P$
iff (assuming that $R1$ is the default rule, into which R is removed body literals, and P is replaced by $P \cup \{R1\}$, $\exists L\ S \cup \{R1\} \vdash_{\min} \neg L \wedge L$)
or $\exists L$ ($S \vdash_{\min} L$ and $S \vdash \neg L$ and R is used last among default rules to derive $\neg L$)

The latter part of this definition indicates that L is derived from a set of literals including $\neg L$. If P is a DPwN, the definitions of $D\text{-}attacks_P$ can be simplified such that "$(S, R) \in D\text{-}attacks_P$ iff $S \vdash_{\min} L$ and $Head(R) = \neg L$."
Next, we define RF for EDLPs.

Definition 2.3.3 (RF for EDLPs)
A RF for an EDLP P is a pair

$$RF(P) = (De\text{-}rule_P, attacks_P)$$

where $De\text{-}rule_P$ is the set of all defeasible rules in G_P and $attacks_P$ equals $G\text{-}attacks_P \cup D\text{-}attacks_P$.

The definition of $G\text{-}attacks_P$ is not changed, but the meaning of $(S, R) \in G\text{-}attacks_P$ is that S is a minimal set of defeasible rules such that, if all explicit and implicit assumptions of S are true, then some explicit assumptions of R are false.

We call $L_0 \leftarrow L_1, ..., L_m, \sim L_{m+1}, ..., \sim L_n, \sim\neg L_0$. the seminormal exact rule of $L_0 \Leftarrow L_1, ..., L_m, \sim L_{m+1}, ..., \sim L_n$.

Lemma 2.3.1

(1) *Let P be a DP and P' be an ELP into which all rules in P are replaced by their seminormal exact rules.*
(S, R)∈D-attacks$_P$ iff (S, R)∈G-attacks$_{P'}$.

(2) *Let P be a DPwN and P' be an ELP into which all rules in P are replaced by their seminormal exact rules.*
If (S, R)∈D-attacks$_P$ then (S, R)∈G-attacks$_{P'}$.

(3) *Let P be a DLP and P' be an ELP into which all default rules in P are replaced by their seminormal exact rules.*
(S, R)∈D-attacks$_P$ if (S, R)∈G-attacks$_{P'}$.

Proof.

(1)(2) ⇒ :(*S*, *R*)∈*D-attacks$_P$*.
Then $\exists \sim L \in As(R)$ $S \vdash_{min} L$ in P'.

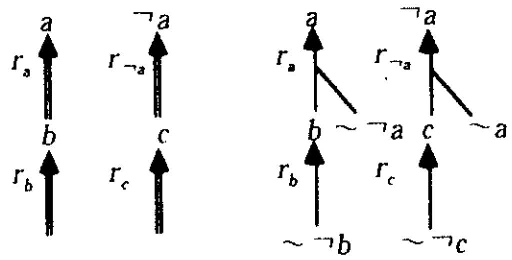
(1) ⇐ :(*S*, *R*)∈*G-attacks$_P$*.
Then $\exists L$ $S \vdash_{min} L$ and $Head(R) = \neg L$ in *P*.

(3) ⇐ :(*S*, *R*)∈*G-attacks$_P$*.
Then $\exists L$ $S \vdash_{min} L$ and $Head(R) = \neg L$ in *P*.
If $S \vdash \neg L$ in *P* then $\exists L$ ($S \vdash_{min} L$ and
R is used last among default rules to derive $\neg L$) in *P*
else assuming that *R1* is the default rule, into which *R* is removed body literals, and *P* is exchanged by *P*∪{*R1*}
$\exists L$ $S \cup \{R1\} \vdash_{min} L \wedge \neg L$ in *P* □

We next show an example of lemma 2.3.1(1).

Example 2.3.1

$P1 = \{r_a : a \Leftarrow b., r_{\neg a} : \neg a \Leftarrow c., r_b : b \Leftarrow., r_c : c \Leftarrow.\}$
$P2 = \{r_a : a \leftarrow b, \sim\neg a., r_{\neg a} : \neg a \leftarrow c, \sim a., r_b : b \leftarrow \sim\neg b., r_c : c \leftarrow \sim\neg c.\}$



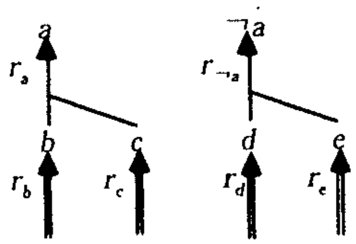
P1 *P2*

Attacks$_{P1}$ is $\{(\{r_a, r_b\}, r_{\neg a}), (\{r_{\neg a}, r_c\}, r_a)\}$, which equals *Attacks$_{P2}$*.

Definition2.2.4~2.2.10 are not changed and Theorem2.2.1~2.2.4 are hold, because *AF*(*P*) is not changed.
We show an example of DLP.

Example 2.3.2

$P = \{r_a : a \Leftarrow b, c., r_{\neg a} : \neg a \Leftarrow d, e., r_b : b \Leftarrow., r_c : c \Leftarrow., r_d : d \Leftarrow., r_e: e \Leftarrow.\}$



$attacks_P = D\text{-}attacks_P = \{(\{r_b, r_c, r_d\}, r_e), (\{r_b, r_c, r_e\}, r_d), (\{r_b, r_d, r_e\}, r_c), (\{r_c, r_d, r_e\}, r_b)\}$
GE = ϕ
PE = SE = $\{r_b, r_c, r_d\}, \{r_b, r_c, r_e\}, \{r_b, r_d, r_e\}, \{r_c, r_d, r_e\}$

2.4. RF for Prioritized EDLP

A prioritized EDLP$_\Pi$ is (*P*, <). *P* is an EDLP. < is a strict partial order on *P*.

We define *conflicts$_P$* as relations between sets of defeasible rules that attacks each other in *AF*(*P*).

Definition 2.4.1 (*Conflicts$_P$, G-conflicts$_P$, D-conflicts$_P$, DG-conflicts$_P$*)

conflicts$_P$:= {((*S*, *R*), (*S*', *R*)) | (*S*, *R*')∈*attacks$_P$*, (*S*', *R*)∈*attacks$_P$*, *R*'∈*S*' and *R*∈*S*}
G-conflicts$_P$:= {((*S*, *R*), (*S*', *R*)) | (*S*, *R*')∈*G-attacks$_P$*, (*S*', *R*)∈*G-attacks$_P$*, *R*'∈*S*' and *R*∈*S*}
D-conflicts$_P$:= {((*S*, *R*), (*S*', *R*)) | (*S*, *R*')∈*D-attacks$_P$*, (*S*', *R*)∈*D-attacks$_P$*, *R*'∈*S*' and *R*∈*S*}
DG-conflicts$_P$:= {((*S*, *R*), (*S*', *R*)) | (*S*, *R*')∈*D-attacks$_P$*, (*S*', *R*)∈*G-attacks$_P$*, *R*'∈*S*' and *R*∈*S*} ∪ {((*S*, *R*), (*S*', *R*)) | (*S*, *R*')∈*G-attacks$_P$*, (*S*', *R*)∈*D-attacks$_P$*, *R*'∈*S*' and *R*∈*S*}

We consider semantics of rule priorities in *AF*(*P*) = (*AR$_P$*, *AR-attacks$_P$*). When ((*S*, *R*), (*S*', *R*'))∈*conflicts$_P$* is formed, {(*S*, *S*'*S* ∪ {*R*}), (*S*', *S**S*' ∪ {*R*})} is a subset of *AR-attacks$_P$*. In brief, if some condition between sets of rules in *S* and *S*' is satisfied, (*S*, *S*'*S* ∪ {*R*}) or (*S*', *S**S*' ∪ {*R*}) is removed from *AR-attacks$_P$*. *attacks$_P$* denotes this removed one.

Definition 2.4.2 (*RF for prioritized EDLPs*)

A RF for a prioritized EDLP$_\Pi$ = (*P*, <) is

$$RF(\Pi) = (De\text{-}rule_P, attacks_P, attacks^-_P, AR\text{-}attacks_P)$$

where *De-rule$_P$* is the set of all defeasible rules in *G$_P$*, *attacks$_P$* is *G-attacks$_P$* ∪ *D-attacks$_P$*, *attacks^-_P* is {(*S*, *S*'*S* ∪ {*R*}) | ((*S*, *R*), (*S*', *R*'))∈*CONFLICTS*, and *RELATE*)} and *AR-attacks$_P$* equals $\{(S, S') \subseteq 2^{De\text{-}ruleP} \times 2^{De\text{-}ruleP} \mid \exists R' \in S'\ \exists S1 \subseteq S\ (S1, R') \in attacks_P$ and $\forall S1'$ s.t. $R' \in S1' \subseteq S'\ (S1, S1') \in attacks^-_P\}$. *CONFLICTS* is one of the above defined conflicts. *RELATE* is a strict partial order between sets of rules in *S* and *S*'.

CONFLICTS and *RELATE* can be variously defined. We write RF for prioritized EDLPs as RF$_\Pi$. The above RF$_\Pi$ represents an argumentation framework *AF*(Π) = (*AR$_P$*, *AR-attacks$_P$*), where *AR$_P$* is the set of all arguments, argument is a set of defeasible rules.

Lemma 2.4.1
Let S and S' be subsets of $De\text{-}rule_P$.
(1) *If (S, S')∈$AR\text{-}attacks_P$*
then ∀S1' s.t. R'∈S1'⊆S' (S0, S1')∈$AR\text{-}attacks_P$
(2) *If (S, S)∈$AR\text{-}attacks_P$*
then ∀S0⊆$De\text{-}rule_P$ s.t. S⊆S0 (S0, S0)∈$AR\text{-}attacks_P$
(3) *If ∃S1⊆S ∃R1'∈S' ((S1, R1')∈$attacks_P$ and (S, S')∉$AR\text{-}attacks_P$)*
then ∃Δ⊆S ∃Δ'⊂S ((S' ∪ Δ, S)∈$AR\text{-}attacks_P$ or (S ∪ Δ', S')∈$AR\text{-}attacks_P$)

We define conflict-free, acceptable and admissible in RF_Π, and preferred, stable, grounded and complete extension of RF_Π so that they can be equivalent to these definitions in subsection 2.1. These equivalences are expressed by Theorem 2.4.2~2.4.5.

Definition 2.4.3 (*Conflict-free in RF_Π*)
A set *S* of defeasible rules is *conflict-free in RF*(Π) iff *∀S1⊆S ∀R∈S (S1, R)∉$attacks_P$.*

Definition 2.4.4 (*Acceptable in RF_Π*)
A set S0 of defeasible rules is *acceptable* w.r.t. a set S of defeasible rules *in RF(Π)* iff ∀*S'*⊆$De\text{-}rule_P$ if (*S'*, *S0*)∈$AR\text{-}attacks_P$ then (S, S')∈$AR\text{-}attacks_P$.

Definition 2.4.5 (*Admissible in RF_Π*)
A set *S* of defeasible rules is *admissible in RF*(Π) iff *S* is conflict-free and acceptable w.r.t. S in *RF*(Π).

Definition 2.4.6 (*Preferred Extension of RF_Π*)
A set *S* of defeasible rules is a *preferred extension of RF*(Π) iff *S* is a maximal admissible set of *RF*(Π).

Definition 2.4.7 (*Stable Extension of RF_Π*)
A set *S* of defeasible rules is a *stable extension of RF*(Π) iff *S* is conflict-free in *RF*(Π) and ∀S'∈$2^{De\text{-}rule_P}\backslash 2^S$ (*S, S'*)∈$AR\text{-}attacks_P$.

Definition 2.4.8 (*Grounded Extension of RF_Π*)
A set *S* of defeasible rules is a *grounded extension of RF*(Π) iff *S* is the least fix point of $F_{RF(\Pi)}$.
$F_{RF(\Pi)}$ is defined as follows.
$F_{RF(\Pi)} : 2^{De\text{-}rule_P} \rightarrow 2^{De\text{-}rule_P}$
$F_{RF(\Pi)}(S) = \cup\{S0 \subseteq De\text{-}rule_P \mid S0$ is acceptable w.r.t. *S* in *RF*(Π)}

Definition 2.4.9 (*Complete Extension of RF_Π*)
A set *S* of defeasible rules is a *complete extension of RF*(Π) iff *S* is admissible in *RF*(Π) and each set of defeasible rules, which is acceptable w.r.t. *S* in *RF*(Π), is a subset of *S*.

Assuming that Π = (*P*, <) is a prioritized EDLP and *S* is a subset of $De\text{-}rule_P$, the theorems below can be concluded. Due to space limitation, their proofs are omitted here.

Theorem 2.4.2
(1) *Let AS be a set of arguments, which are sets of defeasible rules, and ∪AS equal S.*
AS∪{S} is conflict-free in AF(Π) iff AS∪{S} is conflict-free in AF(P).
(2) *∃AS⊆$2^{De\text{-}rule_P}$ s.t. ∪AS = S AS∪{S} is conflict-free in AF(Π) iff S is conflict-free in RF(Π).*

Theorem 2.4.3
(1) *Let AS be a set of arguments, which are sets of defeasible rules, and ∪AS equal S.*
If an argument S0 is acceptable w.r.t. AS in AF(Π) then S0 is acceptable w.r.t. AS∪{S} in AF(Π).
(2) *∃AS⊆$2^{De\text{-}rule_P}$ s.t. ∪AS = S an argument S0 is acceptable w.r.t. AS∪{S} in AF(Π) iff S0 is acceptable w.r.t. S in RF(Π).*

Theorem 2.4.4
(1) *Let AS be a set of arguments, which are sets of defeasible rules, and ∪AS equal S.*
If AS is admissible in AF(Π) then AS ∪ {S} is admissible in AF(Π).
(2) *∃AS⊆$2^{De\text{-}rule_P}$ s.t. ∪AS = S AS∪{S} is admissible in AF(Π) iff S is admissible in RF(Π).*

Theorem 2.4.5
(1) *Let AS be a set of arguments, which are sets of defeasible rules, and ∪AS equal S.*
If AS is a preferred (stable, grounded or complete) extension of AF(Π) then S∈AS.
(2) *∃AS⊆$2^{De\text{-}rule_P}$ s.t. ∪AS = S AS∪{S} is a preferred (resp. stable, grounded and complete) extension of AF(Π) iff S is a preferred (resp. stable, grounded and complete) extension of RF(Π).*

The corollary below can be concluded from Corollary2.1.1, Lemma2.1.2, Theorem2.1.3 and Theorem2.4.5.

Corollary 2.4.6
(1) *Every RF_Π possesses at least one preferred extension.*
(2) *Every stable extension of a RF_Π is a preferred extension of it, but not vice versa.*
(3) *Each preferred extension of a RF_Π is a complete extension of it, but not vice versa.*
(4) *The grounded extension of a RF_Π is the least complete extension of it.*
(5) *The complete extensions of a RF_Π form a complete semilattice.*

Theorem 2.4.7
Let Π = (P, <) be a prioritized EDLP, SE(Π) and GE(Π) be stable extension and grounded extension of RF(Π), and SE(P) and GE(P) be stable extension and grounded extension of RF(P).
(1) *SE(Π) is SE(P).*
(2) *GE(Π)⊇GE(P)*

Proof.
(1) S is a stable extension of RF(Π)
⇒ S is conflict-free in RF(Π) and

Table 1

	problem	*CONFLICTS*	*RELATE*	semantics
[Brewka '96][6]	ELP	$G\text{-}conflicts_P$	$R < R'$	Cn(GE')
[Gabbay, et.al '91][10]	DP	$D\text{-}conflicts_I$	$R < R'$ and *1	Cn(GE), Cn(PE), Cn(CE), Cn(SE)
[Dimopoulos&Kakas '95][8]	DP	$D\text{-}conflicts_P$	*2	Cn(PE)
[Kowalski&Toni '96][12]	DPwN	$D\text{-}conflicts_P$	$R < R'$ and *1	Cn(PE)
[Prakken&Sartor '96][15]	DLPwN	($D\text{-}conflicts_P$ / $DG\text{-}conflicts_P$	$R < R'$ and *3 / $(S, S') \in D\text{-}conflicts_P$)	Cn(GE)

*1 : $Head(R) = \neg Head(R')$

*2 : $\forall R1 \in S\ \forall R1' \in S'\ \neg(R1' < R1)$ and $\exists R1 \in S\ \exists R1' \in S'\ R < R1'$

*3 : there is a literal L such that $R \in Top_L(S \cup S')$ and $R' \in Top_{\neg L}(S \cup S')$

$\forall S' \in 2^{De\text{-}rule_P} \backslash 2^S\ \exists S1 \subseteq S\ \exists R' \in S'$
$((S1, R') \in attacks_P$ and $\forall S1'$ s.t. $R' \in S1' \subseteq S'$
$(S1, S1') \notin attacks^-_P)$
$\Rightarrow$ S is conflict-free in RF(Π) and
$\forall R' \in De\text{-}rule_P \backslash S\ \exists S1 \subseteq S\ (S1, R') \in attacks_P$
$\Rightarrow$ S is a stable extension of RF(P)

(2) $F_{RF(P)}(\phi) = \{R \mid \forall S' \in De\text{-}rule_P\ (S', R) \notin attacks_P\}$
$F_{RF(\Pi)}(\phi) = \cup\{S0 \mid \forall S' \in De\text{-}rule_P\ (S', S0) \notin AR\text{-}attacks_P\}$
Assuming that $S0 = F_{RF(P)}(\phi)$,
$\forall S' \in De\text{-}rule_P\ (S', S0) \in AR\text{-}attacks_P$
$\therefore F_{RF(P)}(\phi) \subseteq F_{RF(\Pi)}(\phi)$

$F_{RF(P)}{}^2(\phi) = \{R \mid \forall S' \in De\text{-}rule_P$ if $(S', R) \in attacks_P$
then $\exists R' \in S'\ \exists S1 \subseteq F_{RF(P)}(\phi)\ (S1, R') \in attacks_P\}$
$F_{RF(\Pi)}{}^2(\phi) = \cup\{S0 \mid \forall S' \in De\text{-}rule_P$
if $(S', S0) \in AR\text{-}attacks_P$
then $(F_{RF(\Pi)}(\phi), S') \in AR\text{-}attacks_P\}$
Assuming that $S0 = F_{RF(P)}{}^2(\phi)$,
$\forall S' \in De\text{-}rule_P$ if $(S', S0) \in AR\text{-}attacks_P$
then $(F_{RF(\Pi)}(\phi), S') \in AR\text{-}attacks_P$
$\therefore F_{RF(P)}{}^2(\phi) \subseteq F_{RF(\Pi)}{}^2(\phi)$

Similarly $F_{RF(P)}{}^3(\phi) \subseteq F_{RF(\Pi)}{}^3(\phi)$, $F_{RF(P)}{}^4(\phi) \subseteq F_{RF(\Pi)}{}^4(\phi)$, ...
$\therefore GE(P) \subseteq GE(\Pi)$ □

2.5. Rule Priorities within Logical Language

In Refs.5)~7) and 15), rule priorities can be expressed within logical language. In this subsection we define the semantics of this case. Named EDLP Γ is (P, *name*) where P is an EDLP, which includes (infix) predicate symbol $\prec$ that can take rule names as arguments. Rule name is a constant symbol, and P contains all ground instances of the following rules, where x, y, and z are parameters for rule names:

$$x \prec z \leftarrow x \prec y,\ y \prec z.$$
$$\neg(x \prec y) \leftarrow y \prec x.$$

name is a partial injective function such that "$De\text{-}rule_P \rightarrow N$" and N is a set of constant symbols. We next define the semantics of named EDLP.

Definition 2.5.1 (*Preferred (Stable, Grounded, Complete) Extension of named EDLP*)
A set S of defeasible rules is a preferred (resp. stable, grounded, and complete) extension of named EDLP $\Gamma = (P, name)$
iff S is a preferred (resp. stable, grounded, and complete) extension of $RF(\Pi)$.
$\Pi = (P, <)$
$< = \{n1 < n2 \mid n1 \prec n2 \in Cl(S)\}$
< is consistent

Although the above definition is recursive, the constructive definitions of stable extension and grounded extension can be formed according to theorem 2.4.7.

Definition 2.5.2 (*Stable Extension of named EDLP*)
A set S of defeasible rules is a stable extension of named EDLP $\Gamma = (P, name)$
iff S is a stable extension of $RF(P)$. And S is a stable extension of $RF(\Pi)$.
$\Pi = (P, <)$
$< = \{n1 < n2 \mid n1 \prec n2 \in Cl(S)\}$
< is consistent

Definition 2.5.3 (*Grounded Extension of named EDLP*)
A set S of defeasible rules is a *grounded extension of named EDLP* $\Gamma = (P, name)$
iff S is the least fix point of F_Π.
F_P is defined as follows:
$F_\Gamma: 2^{De\text{-}rule_P} \rightarrow 2^{De\text{-}rule_P}$
$F_\Pi(S)$ = ground extension of $RF(\Pi)$ if < is consistent
$= S$ otherwise
$\Pi = (P, <)$
$< = \{n1 < n2 \mid n1 \prec n2 \in Cl(S)\}$

3. Analysis of Semantics by RF_Π

3.1. Reformulation of Semantics

We reformulate several semantics from the literature, mentioned in Section 1 (**Table 1**).

GE' is grounded extension where $F_{RF(\Pi)}(S)$ is exchanged by $F_{RF(\Pi)}'(S)$.

$F_{RF(\Pi)}'(S) = S \cup \{R \subseteq De\text{-}rule_P \backslash S \mid \forall S' \subseteq De\text{-}rule_P$
if $(S', R) \in attacks_P$
then $\exists S2$ s.t. $R \in S2 \subseteq \{R\} \cup S\ (S', S2) \subseteq attacks^-_P$ or
$\exists S1 \subseteq S\ \exists R' \in S'\ (S1, R') \in attacks_P)\}$

GE' ⊆ GE is hold. $Top_L(S)$ stands for a set of default rules, which are used last among S to derive L. About

Ref.15) we modified its semantics. In Example 2.3.2 if $<$ equals $\{r_b < r_d, r_b < r_e\}$, then original semantics, "justified conclusions," is $\{b, c, d, e, \neg a\}$. We modified this into $\{c, d, e, \neg a\}$. If $<$ equals $\{r_b < r_d, r_c < r_e\}$, then original semantics is $\{b, c, d, e\}$. We modified this into $\{d, e, \neg a\}$. We modified the definition of *rebuts* as follows:

> *Let A1 and A2 be two arguments. Then A1 rebuts A2 iff there is a sequence S of strict rules such that A1+A2+S has conclusions L and ¬L and* $\exists R' \in R_{\neg L}(A1+A2+S) \cap A2$ $\forall R \in R_L(A1+A2+S) \cap A1$ $R \nless R'$.

Note: that the above *argument* is not used in this paper. The original definition of rebuts is as follows.

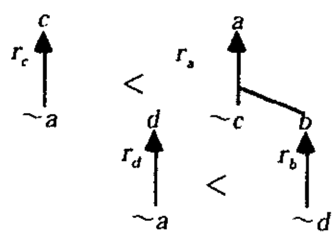
> Let A1 and A2 be two arguments. Then A1 rebuts A2 iff there are sequences S1, S2 of strict rules such that A1+S1 has a conclusion L and A2+S2 has a conclusion ¬L and $\forall R \in R_L(A1 + S1)$ $\exists R' \in R_{\neg L}(A2+S2)$ $R \nless R'$.

3.2. Improved Semantics

Ref.6)'s semantics can be improved by replacing GE' with GE.

Example 3.2.1

$\Pi = (\{r_a{:}a \leftarrow b, {\sim}c.,\ r_b{:}b \leftarrow {\sim}d.,\ r_c{:}c \leftarrow {\sim}a.,\ r_d{:}d \leftarrow {\sim}a.\},$
$\{r_c < r_a, r_d < r_b\})$

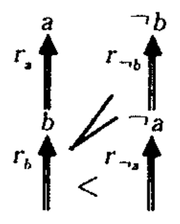


$Cn(\text{GE}')_{\text{Brewka}} = \phi$
$Cn(\text{GE})_{\text{Brewka}} = \{a, b\}$

For Refs.8), 10), 12), and 15)'s semantics, rule priorities between rules not used last to derive conflicting literals, can be considered by removing "$Head(R) = \neg Head(R')$" or "there is a literal L such that $R \in Top_L(S \cup S')$ and $R' \in Top_{\neg L}(S \cup S')$" from *RELATE*.

Example 3.2.2

$\Pi = (\{r_a{:}a \Leftarrow b.,\ r_b{:}b \Leftarrow.,\ r_{\neg a}{:}\neg a \Leftarrow \neg b.,\ r_{\neg b}{:}\neg b \Leftarrow \neg a\},$
$\{r_b < r_{\neg a}, r_b < r_{\neg b}\})$



$Cn(\text{GE})_{\text{Gabbay}} = Cn(\text{GE})_{\text{Prakken}} = \phi$
$Cn(\text{PE})_{\text{Gabbay}} = Cn(\text{PE})_{\text{Dimopoulos}} = Cn(\text{PE})_{\text{Kowalski}} = \{a, b\}, \{\neg a, \neg b\}$

By above mentioned removal, all of them become $\{\neg a, \neg b\}$.

Refs.3) and 9) suggest that in argumentation framework grounded extension, preferred extensions and complete extensions can be improved by replacing acceptable (Definition2.1.3) with the following acceptable*:

Definition 3.2.1 (*Acceptable**)
An argument A is acceptable* w.r.t. a set AS of arguments
iff $\forall A' \in AR$ $\exists A1 \in AS$ if $(A', A) \in AR\text{-}attacks$
then $(A1, A') \in AR\text{-}attacks$ or $(A', A') \in AR\text{-}attacks$

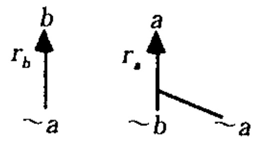
In the same way, most semantics in Table 1 can be improved by replacing acceptable in RF_Π(Definition2.4.4) with the following acceptable* in RF_Π:

Definition 3.2.2 (*Acceptable* in* RF_Π)
A set $S0$ of defeasible rules is acceptable* w.r.t. a set S of defeasible rules in $RF(\Pi)$
iff $\forall S' \subseteq De\text{-}rule_P$ if $(S', S0) \in AR\text{-}attacks_P$ then
$(S, S') \in AR\text{-}attacks_P$ or $(S', S') \in AR\text{-}attacks_P$.

We denote this improved GE, PE, and CE as GE*, PE*, and CE*.

Example 3.2.3

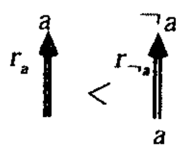
$\Pi = (\{r_a{:}a \leftarrow {\sim}b, {\sim}a.,\ r_b{:}b \leftarrow {\sim}a.\}, \{r_b < r_a\})$



$Cn(\text{GE})_{\text{Brewka}} = \phi$
$Cn(\text{GE}^*)_{\text{Brewka}} = \{b\}$

Example 3.2.4

$\Pi = (\{r_a{:}a \Leftarrow.,\ r_{\neg a}{:}\neg a \Leftarrow a.\}, \{r_a < r_{\neg a}\})$



$Cn(\text{GE})_{\text{Gabbay}} = Cn(\text{GE})_{\text{Prakken}} = Cn(\text{PE})_{\text{Gabbay}} = Cn(\text{PE})_{\text{Dimopoulos}} = Cn(\text{PE})_{\text{Kowalski}} = \phi$
$Cn(\text{GE}^*)_{\text{Gabbay}} = Cn(\text{GE}^*)_{\text{Prakken}} = Cn(\text{PE}^*)_{\text{Gabbay}} = Cn(\text{PE}^*)_{\text{Dimopoulos}} = Cn(\text{PE}^*)_{\text{Kowalski}} = \{a\}$

3.3. New Semantics

We propose new semantics (**Table 2**), based on the literature of Table 1.

We next explain the features of this semantics. When we consider the semantics of Ref.6) in a program including default rules, we assume all default rules in the program are replaced by their seminormal exact rule.

(1) We give semantics of prioritized EDLPs, which is more

Table 2

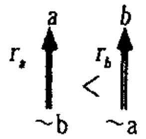
problem	CONFLICTS	RELATE	semantics
	G-conflicts$_P$	$R<R'$	
EDLP	*D-conflicts*$^*_P$*	$R<R'$	Cn(GE'), Cn(PE')
	DG-conflicts$^*_P$*	$(S, S')\in$ *D-conflicts*$^*_P$	

general than the problems of Table 1.

(2) We use rule priorities to resolve both *D-conflicts$_P$* and *G-conflicts$_P$*.

Example 3.3.1

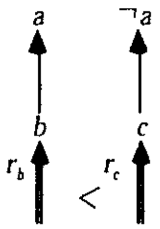
$\Pi = (\{r_a{:}a \Leftarrow \sim b.,\ r_b{:}b \Leftarrow \sim a.\},\ \{r_a < r_b\})$



$Cn(\mathrm{GE})_{\mathrm{Prakken}} = \phi$
$Cn(\mathrm{GE})_{\mathrm{Brewka}} = Cn(\mathrm{GE})_{\mathrm{New}} = \{b\}$

Example 3.3.2

$\Pi = (\{r_a{:}a \leftarrow b.,\ r_{\neg a}{:}\neg a \leftarrow c.\ r_b{:}b \Leftarrow.,\ r_c{:}c \Leftarrow.\},$
$\{r_b < r_c\})$

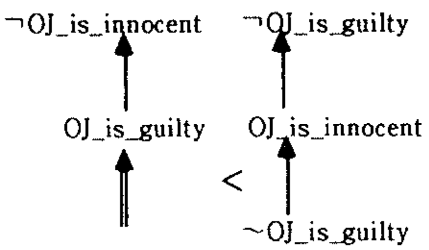


$Cn(\mathrm{GE})_{\mathrm{Brewka}} = Lit$
$Cn(\mathrm{GE})_{\mathrm{Prakken}} = Cn(\mathrm{GE})_{\mathrm{New}} = \{\neg a, c\}$

(3) For *DG-conflicts$_P$*, as in Ref.15), we give precedence to *G-attacks$_P$* over *D-attacks$_P$*. Although Refs.6) and 12) do not make out the meaning, we show an example easily understood and similar to Example2.18 of Ref.15).

Example 3.3.3

Π = ({r₁:¬OJ_is_innocent ← OJ_is_guilty., r₂:¬OJ_is_guilty ← OJ_is_innocent., r₃:OJ_is_guilty⇐., r₄:OJ_is_innocent ← ~OJ_is_guilty.}, {r₃ < r₄})



$Cn(\mathrm{GE})_{\mathrm{Brewka}}$ = {OJ_is_innocent, ¬OJ_is_guilty}
$Cn(\mathrm{GE})_{\mathrm{New}}$ = {OJ_is_guilty, ¬OJ_is_innocent}

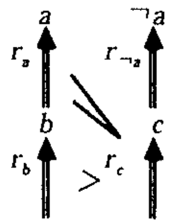
(4) We use *D-conflicts*$_P$* instead of *D-conflicts$_P$*, and *DG-conflicts*$_P$* instead of *DG-conflicts$_P$*. These are where *D-attacks$_P$*(*Definition2.3.2*) are replaced by *D-attacks*$_P$*.

Definition 3.3.1 (*D-attacks*$_P$*)

Let S be a subset of *De-rule$_P$* and R belong to *Dt-Rule$_P$*.
$(S, R)\in$*D-attacks*$_P$*
iff (assumed that $R1$ is the default rule, into which R is removed body literals, and P is replaced by $P\cup\{R1\}$, $\exists L$ $S\cup\{R1\} \vdash_{\min} \neg L \wedge L$)
or $\exists L$ ($S \vdash_{\min} L$ $S \vdash \neg L$ and R is used last among default rules to derive $\neg L$)

Example 3.3.4

$\Pi = (\{r_a{:}a \Leftarrow b.,\ r_{\neg a}{:}\neg a \Leftarrow c.\ r_b{:}b \Leftarrow.,\ r_c{:}c \Leftarrow.\ \},$
$\{\ r_c < r_b,\ r_c < r_a\})$



$attacks_P = D\text{-}attacks_P = \{(\{r_a, r_b\}, r_{\neg a}),\ (\{r_{\neg a}, r_c\}, r_a),$
$(\{r_a, r_{\neg a}, r_c\}, r_b),\ (\{r_a, r_{\neg a}, r_b\}, r_c)\}$
$attacks^-_P = \{((\{r_{\neg a}, r_c\}, \{r_a, r_b\}),\ (\{r_a, r_{\neg a}, r_c\}, \{r_b\})\}$
$\mathrm{GE_{Hew}} = \mathrm{PE_{Hew}} = \{r_a, r_b, r_c\}$
$Cn(\mathrm{GE})_{\mathrm{Hew}} = \mathrm{Cn(PE)_{Hew}} = \{a, b, c\}$
For semantics other than Ref.8), rule priorities are disregarded and Cn(GE) is $\{b, c\}$, Cn(PE) are $\{a, b, c\}$ and $\{\neg a, b, c\}\}$.

(5) We apply GE* and PE* instead of GE and PE.

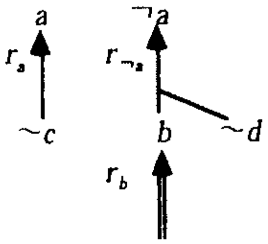
In Example3.3.4 s.t. $< = \phi$, $\mathrm{Cn(GE)_{Hew}} = \phi$ is improved to $\mathrm{Cn(GE^*)_{Hew}} = \{b, c\}$

4. Related Work

For a framework for nonmonotonic reasoning, NAF-based frameworks[3,9] are simple and easy to understand. In Ref.9), for example, *AR* of argumentation framework is the set of all arguments and argument is a set of assumptions. For the purpose of analyzing the effects of rule priorities, however NAF-based frameworks are inconvenient, because rules, from which conclusions are derived, must be identified. For semantics of nonmonotonic reasoning with rule priorities, Refs.6), 8), 10) and 12) submit to an "argument"-based approach. "Here, argument" denotes a minimal set of rules, from which a literal is derived. In an "argument-based" approach, only *AR-attacks* between arguments are considered. As we show in Example 2.3.2, however, in DLP *AR-attacks* between closed rule sets must be considered. Furthermore, in EDLP, *AR-attacks* between sets of defeasible rules must be considered, such as Example 4.1.1. If *D-conflicts$_P$* is replaced by *D-conflicts*$_P$*, even in DP, *AR-attacks* between sets of defeasible rules must be considered.

Example 4.1.1

$P = \{r_a{:}a \leftarrow \sim c.,\ r_{\neg a}{:}\neg a \leftarrow b, \sim d.\ r_b{:}b \Leftarrow\ \}$



$D\text{-}attacks_P = \{((\{r_a, r_{\neg a}\}, r_b))\}$
$Cn(\text{GE}) = Cn(\text{PE}) = \{a\}$

For semantics of default rules, except for regarding $a \Leftarrow b$. as $a \leftarrow b, \sim\neg a$, regarding $a \Leftarrow b$ as a default $:b \rightarrow a / b \rightarrow a$ is known.[14] This semantics has a problem of contraposition arising also in nonmonotonic reasoning with rule priorities.

Several symblols, concepts' names and concepts in Subsection2.2~2.4 are based on various literature. To be concrete, $\vdash_{min}$,[8] *Cl*,[6] *Cn*,[6] *defeasible rule*,[6,15] *strict rule*,[6,15] *exact rule*,[12] *default rule*,[7] *closed*[8] rule set, *seminormal*[6] rule, *prioritized*[6] EDLP and concepts of *conflicts*[6] and *named*[6] EDLP are based on their superscripts' literature.

5. Conclusion

We provided RF for ELPs, clarified semantics of default rules and rule priorities in RF, and extended it to RF for prioritized EDLPs. Using RF for prioritized EDLPs, we reformulated several semantics of past studies, indicated their improvement, and gave new prioritized EDLP semantics. This papaer is a first step in this line of research, and much work remains, e.g., to make formal results of Section 3 sufficient, to show relationships among semantics in Table 1, to extend RF to deal with the rest of the literature, and to consider proof procedures for computing these semantics.

References:

1) A.Analyti and S.Pramanik, "Reliable Semantics for Extended Logic Programs with Rule-Prioritization," J. Logic Computat., 5-3, 303-324, (1995).
2) F.Baader and B.Hollunder, "How to Prefer More Specific Defaults in Terminological Default Logic." Proc. of IJCAI-93, 669-674, (1993).
3) A.Bondarenko, F.Toni and R.A.Kowalski, "An Assumtion-based Framework for Non-monotonic Reasoning," Proc. 2nd Int. Workshop on Logic Programming and Non-Monotonic Reasoning, 171-189, (1993).
4) G.Brewka, "Prefered Subtheories: An Extended Logical Framework for Default Reasoning," Proc. of IJCAI-89, 1043-1048, (1989).
5) G.Brewka, "Reasoning about Priorities in Default Logic," Proc. of AAAI-94, 247-260, (1994).
6) G.Brewka, "Well-founded Semantics for Extended Logic Programs with Dynamic Preferences," J. of Artificial Intelligence Research, 4, 19-36, (1996).
7) G.Brewka and T.F.Gordon, "How to Buy a Porsche: An Approach to Defeasible Decision making. Preliminary Reports," Proc. AAAI-94 Workshop on Computational Dialectics, (1994).
8) Y.Dimopoulos and A.C.Kakas, "Logic Programming without Negation as Failure," Proc. of the Int. Logic Programming Symposium, Portland, 369-384, (1995).
9) P.M.Dung, "On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and N-person Games," Artificial Intelligence, 77, 321-357, (1995).
10) D.Gabbay, E.Laenens and D.Vermeir, "Credulous vs. Sceptical Semantics for Ordered Logic Programs," Proc. of the Second Int. Conf. on Principles of Knowledge Representation and Reasoning, 209-217, (1991).
11) B.Grosof, "Generalizing Prioritization," Proc. of Second Int. Conf. on Knowledge Representation and Reasoning, 289-300, (1991).
12) R.A.Kowalski and F.Toni, "Abstract Argumentation," Artificial Intelligence and Law, 4, 275-296, (1996).
13) E.Laenens and D.Vermeir, "On the Relationship between Well-Founded and Stable Partial Models," Proc. of the Mathematical Fundamentals of Database and Knowledge Base Systems, 59-73, (1991).
14) D.Poole, "A Logical Framework for Default Reasoning," Artificial Intelligence, 36, 27-47, (1988).
15) H.Prakken and G.Sartor, "Argument-based Extended Logic Programming with Defeasible Priorities," J. of Applied Non-Classical Logics, 7-1, 25-75, (1996).

Name:
Masato Shibasaki

Affiliation:
Department of Computational Intelligence and System Science, Tokyo institute of Technology

Address:
4259 Nagatsuta, Midori-ku, Yokohama 226, Japan

Brief Biographical History:
1989- Hitachi, Ltd.
1993- Institute for New Generation Computer Technology (ICOT)
1995- Hitachi, Ltd.
1996- Tokyo Institute of Technology

Main Works:
- M. Shibasaki, K. Nitta, "Defeasible Reasoning in Japanese Criminal Jurisprudence." FGCS '94 Workshop 5, pp.37-44, (1994).

Membership in Learned Societies:
- The Japanese Society for Artificial Intelligence (JSAI)

Name:
Katsumi Nitta

Affiliation:
Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology

Address:
4259 Nagatsuta, Midori-ku, Yokohama 226, Japan

Brief Biographical History:
1980- Electrotechnical Laboratory.
1989- Institute for New Generation Computer Technology (ICOT)
1995- Electrotechnical Laboratory
1996- Tokyo Institute of Technology

Main Works:
- K. Nitta, M. Shibasaki, "Defeasible Reasoning in Japanese Criminal Jurisprudence," Journal of AI and Law, Vol. 5, No. 1-2, pp.139-159, (1997).

Membership In Learned Societies:
- The Japanese Society for Artificial Intelligence (JSAI)
- The Information Processing Society of Japan (IPSJ)