Translating the Japanese Presupposed Ultimate Fact Theory into Logic Programming

Ken SATOH [a,1] Masahiro KUBOTA [a] Yoshiaki NISHIGAI [a] Chiaki TAKANO [a]
[a] *National Institute of Informatics*

Abstract. The Japanese "theory of presupposed ultimate facts" (called "Yoken-jijitsu-ron" in Japanese) for interpreting the Japanese civil code has been underway for over forty years mainly by judges in the Japanese Legal Training Institute, but not yet formalized in a mathematical way. This paper attempts to mathematically formalize this theory and presents the correspondence between the theory and logic programming with "negation as failure". It is quite surprising that Japanese judges independently developed such a theory without knowing about logic programming.

Keywords. ultimate facts, burden of proof, civil code

1. Introduction

The Japanese Presupposed Ultimate Fact Theory[2](called "*Yoken-jijitsu-ron*" in Japanese), which we call the JUF theory in this paper, has been mainly developed by judges in the Japanese Legal Training Institute in order to handle the uncertainty that sometimes occur in the court because of a lack of enough evidence. This theory has not attracted scholars from the law departments of universities in Japan for years, but thanks to the Japanese Judicial System Reform, this theory started to be taught in law schools and interest in this theory has grown.

Kitamura [7] concisely explained the JUF theory, so we refer to the article here (in the subsection of "Precision Justice in Civil Cases" in [7]).

> *Referred to as the "theory of presupposed ultimate facts" (Yoken-jijitsu-ron), it is based essentially on a procedural approach to the burden of persuasion. Civil judges should know which facts must be proved, by which party and what to do in case of failure to prove. Accordingly, advocates of this method propose to distinguish, from among the legal events prescribed as being the cause of a give effect, those facts that the plaintiff or defendant must rely on and establish. They call these facts Yoken jijitsu (presupposed ultimate facts). ... in each text stipulated in the Civil Code, facts of positive presupposition relied on by the plaintiff and facts of negative presupposition raised in the defendant's rebuttal must be distinguished precisely. .. Judges are unhappy with scholarly treaties on civil law which neglect to draw this strict distinction. As a result, the judges give a systematic character to their analysis by adding*

[1]Corresponding Author: Ken Satoh, National Institute of Informatics and Sokendai, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan; E-mail: ksatoh@nii.ac.jp

this new element of interpretation to the traditional method in order to extract a code of "judicial norms" from the Civil Code.

We explain the JUF theory in more detail using the following working example.

- Suppose that a plaintiff claims that a lease contract for his house between him and the defendant ended by his cancellation of the contract[2] because the defendant let his sister use a room in the house. According to the Japanese civil code system, the plaintiff must prove that the contract was established [3], and his house was handed over to the defendant[4], and the contract between the defendant and his sister was established [5], and a room in his house was handed over to his sister[6], and she used the room[7], and the plaintiff manifested the intention of cancellation of the contract[8].
- In turn, the defendant alleges that
  - ∗ the plaintiff approved subleasing[9], and the approval was before the cancellation[10].
  - ∗ his subleasing a room to her does not cause any abuse of confidence with the plaintiff because the time of use was very short[11].
- In turn,
  - ∗ the plaintiff denied that he approved subleasing, and
  - ∗ the plaintiff alleged that neighbors' complaints about noise from piano lessons during subleasing abused the confidence with the plaintiff[12].

To handle the above case, we took into account the Japanese Civil Code Article 612,

- item 1 states that a lessee may not assign the lessee's rights or sublease a leased thing without obtaining the approval of the lessor, and
- item 2 states that if the lessee allows any third party to make use of or take profits from a leased thing in violation of the provisions of the preceding paragraph, the lessor may cancel the contract.

However, according to previous Supreme Court cases item 2 is not applicable in exceptional situations where the sublease does not harm the confidence between a lessee and a lessor, and therefore the lessor cannot cancel the contract unless they prove the lessee's abuse of confidence.

So from these rules, the plaintiff firstly must show that the contract with the defendant was established and then that the contract ended because of the sublease to the defendant's sister based on item 2. If there were no counter-arguments, the plaintiff would

[2] In this paper, this fact is represented as `contract_end`.
[3] This fact is represented as `agreement_of_lease_contract`.
[4] This fact is represented as `handover_to_lessee`.
[5] This fact is represented as `agreement_of_sublease_contract`.
[6] This fact is represented as `handover_to_sublessee`.
[7] This fact is represented as `using_leased_thing`.
[8] This fact is represented as `manifestation_cancellation`.
[9] This fact is represented as `approval_of_sublease`.
[10] This fact is represented as `approval_before_cancellation`.
[11] This fact is represented as `fact_of_nonabuse_of_confidence`.
[12] This fact is represented as `fact_of_abuse_of_confidence`.

not have to prove anything else. However there are two kinds of counter-arguments that the defendant must prove. One counter-argument is that the plaintiff had given the defendant permission according to item 1, whereas the other counter-argument is that the defendant had not abused the confidence of the plaintiff according to the case rule above. Suppose that the defendant failed to prove the first counter-argument because of a lack of evidence against the plaintiff's denial. In this case, we cannot deductively prove either the existence of permission nor its non-existence, but because of the burden of proof that was assigned to the defendant, the judge can say that there is no reason for the existence of permission and therefore, this defense does not work. For the latter counter-argument, suppose that the plaintiff proves a counter-counter-argument that there is an abuse of confidence. Then, even if there are some facts related to a non-abuse of confidence, because of other facts related to the abuse of confidence, the defendant failed to prove exceptional circumstances according to the case rule.

These phenomena cannot be explained by using the "deductive" civil code since the "deductive" civil code is designed on the basis of complete information; any fact can be decided to be true or false and therefore the conditions of a rule can be fully evaluated so that we can decide the conclusion to be either false or true. However, in reality, the truth value of the conditions might not be decided even after full factual inquiries has been conducted by the plaintiff and the defendant because of the lack of evidence. Then, the truth value can be "non liquet" and the conclusion is deductively evaluated as "unknown". However, the decision must be made by the court, so we have to find a way to solve the problem of the "non liquet" status. In order to solve this problem, we can use the idea of *burden of proof*. However, in order to use this idea, we must decide to which party the burden of proof of each legal requisite is assigned. Therefore, there has been a high demand for deciding to assign the burden of proof in each civil law code. In Japan, a group of judges has been developing just such an assignment of the burden of proof and this theory of assignment is called the "Presupposed Ultimate Fact Theory" (called "*Yoken-jijitsu-ron*" in Japanese).

In our opinion, the original civil code can be regarded as the rules from a "God's eye view", where scholars argue the legal correctness under a complete set of information. On the other hand, the JUF theory can be regarded as the rules from an "agent's eye view" where the judges as agents argue the correctness of the legal reasoning under a set of incomplete information. This means that the JUF theory is inheritably "nonmonotonic" since the JUF theory makes a conclusion even in situations where the truth-values of some conditions are unknown, and therefore, when the truth-values of the conditions become known, the conclusion is different. Therefore, the whole reasoning process cannot be deductively formalized, and so, we need nonmonotonic formalism.

In [13], we presented a formalization of the burden of proof (precisely speaking burden of persuasion) and showed that a switch of burden of proof can be correctly expressed in the nonmonotonic formalization without any additional functions, which Prakken used in [9]. In Japanese civil procedure law, a burden of proof is a tool used for decision making by a judge when an ultimate fact is "non liquet" even after the examination of all the available pieces of evidence [16]. We formalize the status of "non liquet" as the status in which judges cannot decide its truth value. When an ultimate fact is "non liquet", the default value of a fact is used and as a matter of practice, if some default value is beneficial to one side, then the other side must prove the negation of the default value to prevent the "non liquet" status.

It is known that the JUF theory is one way of presenting a burden of proof in civil code, so we apply the above idea of [13] to mathematically formalize the JUF theory.

In this paper, we firstly give a mathematical formalization of the JUF theory and then provide a translation of the JUF theory into logic programming with "negation as failure". It is quite surprising that without knowing about logic programming, Japanese judges created the corresponding notions in legal reasoning. We then report on our project which formalizes the civil code based on the JUF theory in logic programming technology.

2. Formalizing Japanese Ultimate Fact Theory

The JUF theory divides ultimate facts into two set kinds. One set corresponds to the cause to make the conclusion effective and the other set corresponds to the cause to make the conclusion ineffective.

Let S be a set of ultimate facts. Note that this is based on the assumption that S leads to either a conclusion or the negation of a conclusion according to the civil code or case rules. We define a *critical* set of ultimate facts.

Definition 1

- *$\emptyset$ is a critical set that does not lead to the conclusion.*
- *Let S be a critical set that does not lead to a conclusion and T be a superset of S (that is $S \subset T$) that leads to a conclusion. T is a critical set if the deletion of any element of $T \backslash S$ (the difference set of S and T) from T does not lead to a conclusion. We call $T \backslash S$ an* attack set *w.r.t. S.*
- *Let S be a critical set that leads to a conclusion and T be a superset of S (that is $S \subset T$) that does not lead to a conclusion. T is a critical set if the deletion of any element of $T \backslash S$ (the difference set of S and T) from T leads to a conclusion. We call $T \backslash S$ a* defense set *w.r.t. S.*

In the viewpoint of Japanese law, attack sets generally lead to upholding a claim, and defense sets lead to dismissal of a claim.

Example 1 *Consider the working example. An empty set $\emptyset$ is critical according to the above definition. Then, $T_1 = \{$*
`agreement_of_lease_contract`, `handover_to_lessee`,
`agreement_of_sublease_contract`, `handover_to_sublessee`,
`using_leased_thing`, `manifestation_cancellation` $\}$
is a critical set that leads to `contract_end` *and if we delete any element of T_1, the resulting set does not lead to* `contract_end`*. Therefore, T_1 is also an attack set w.r.t. $\emptyset$. On the other hand, $T_2 = T_1 \cup \{$* `approval_of_sublease`, `approval_before_cancellation` *$\}$ is also a critical set that does not lead to* `contract_end` *since any deletion of an element in $T_2 \backslash T_1 = \{$* `approval_of_sublease`, `approval_before_cancellation` *leads to* `contract_end`*. Therefore, $T_2 \backslash T_1$ is a defense set w.r.t. T_1.*

Definition 2 *We define a JUF tree $\mathcal{T}$ as a finite labeled directed tree $\langle V(\mathcal{T}), E(\mathcal{T}) \rangle$, where*

1. *The label of the root is the concerned conclusion and then every child node of the root is labeled "ATTACK(*$name$*)" and is attached with an attack set w.r.t.* $\emptyset$*, where* $name$ *is the arbitrary name of the child node. The directed edge from the child node to the root is included in* $E(\mathcal{T})$.
2. *Let* $v \in V(\mathcal{T})$ *be an attack node and a union of sets attached with all nodes from the root to* v *be* T*, then every child node of* v *is labeled "DEFENSE(*$name$*)" attached with a defense set* S *w.r.t.* T*. The directed edge from the child node to* v *is included in* $E(\mathcal{T})$*. We call a node labeled with "DEFENSE" a* defense *node.*
3. *Let* $v \in V(\mathcal{T})$ *be a defense node and a union of sets attached with all nodes from the root to* v *be* T *then every child node of* v *is labeled "ATTACK(*$name$*)" and is attached with an attack set* S *w.r.t.* T*. The directed edge from the child node to* v *is included in* $E(\mathcal{T})$*. We call a node labeled with "ATTACK" an* attack *node.*

Example 2 *Consider that another condition, "expiration of term of lease", which leads to the conclusion of "contract ends" is added to the working example*[13]*. Then, the* $\mathcal{T}$ *of the augmented example is shown in Fig. 1.*

We now introduce a new notion, "alleged facts", to avoid irrelevant reasoning in the JUF tree in the following way.

We say that an ultimate fact f is *alleged* if the plaintiff or the defendant alleges the truth or the falsity of f.

Definition 3 *A block diagram* $\mathcal{B}$ *w.r.t a JUF tree* $\mathcal{T}$ *and a set of alleged facts* $\mathcal{A}$ *is a directed tree* $\langle V(\mathcal{B}), E(\mathcal{B})\rangle$ *defined as follows.*

- *The root of* $\mathcal{B}$ *is the root of* $\mathcal{T}$.
- *If* $v \in V(\mathcal{B})$ *and* $v' \in V(\mathcal{T})$ *is a child node of* v *s.t.* $label(v') \subseteq \mathcal{A}$*, then* $v' \in V(\mathcal{B})$ *and* $(v', v) \in E(\mathcal{B})$.

A block diagram filters such irrelevant nodes in the JUF tree that there is a lack of alleged facts attached to the node.

Example 3 *Consider the JUF tree* $\mathcal{T}$ *in example 2 and let* $\mathcal{A}$ *be all the facts in the JUF tree except* `expiration_of_the_term`*. Then, the tree in Fig. 2 is a block diagram w.r.t* $\mathcal{T}$ *and* $\mathcal{A}$.

We say that an ultimate fact f is *plausible* if the degree of belief of f for a judge is over the stringency level. Note that we use a JUF tree for decision making by judge so we assume that the plausibility of fact is decided in advance in a phase of fact finding so we do not care how the plausibility of fact is decided.

Definition 4 *A node* n *is* alive *in a block diagram* $\mathcal{B}$ *w.r.t.* $\mathcal{T}$ *and* $\mathcal{A}$ *under a set of plausible facts* $\mathcal{P}$ *if the following conditions are satisfied.*

- *Every set attached to* n *is in* $\mathcal{P}$.
- *There is no alive node* n' *directed to* n.

If n *is not alive, it is* dead.

[13] Precisely speaking, according to the Japanese Act on Land and Building Lease, we need to prove more ultimate facts, but we omit them here for simplicity.

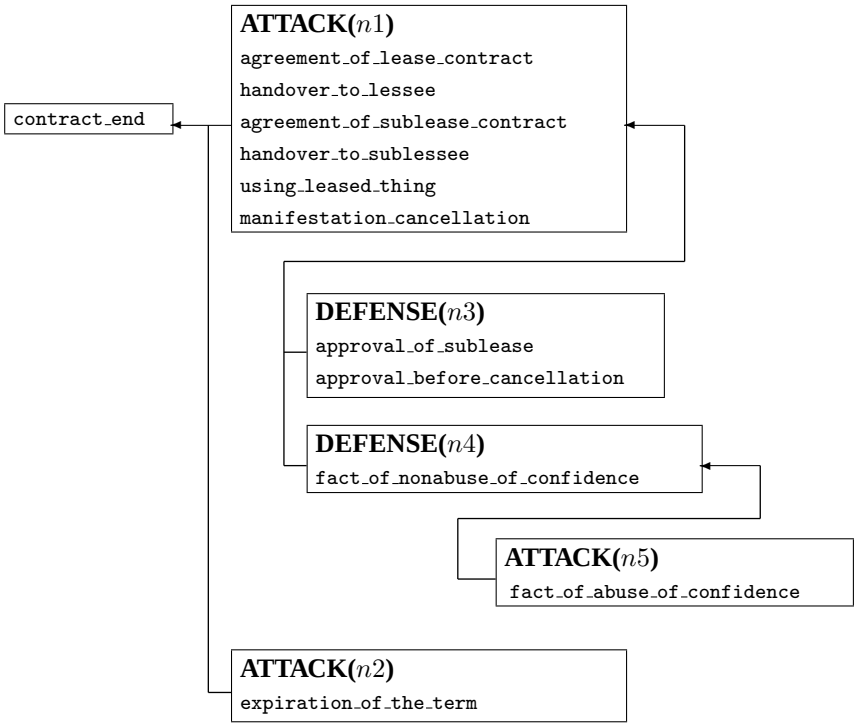

Figure 1. JUF tree

Definition 5 *A block diagram $\mathcal{B}$ w.r.t. $\mathcal{T}$ and $\mathcal{A}$ satisfies c under a set of plausible facts $\mathcal{P}$ (written as $\mathcal{B}^{\mathcal{A}}_{\mathcal{T}}, \mathcal{P} \vdash c$) if an alive child node of c exists in $\mathcal{B}^{\mathcal{A}}_{\mathcal{T}}$ w.r.t. $\mathcal{T}$ and $\mathcal{A}$ under $\mathcal{P}$.*

Example 4 *Consider the block diagram $\mathcal{B}^{\mathcal{A}}_{\mathcal{T}}$ in example 3. Suppose $\mathcal{P}$ is a set of all facts in the block diagram in Fig. 2 except* `approval_of_sublease`*. Then, the node $n3$ attached with* `approval_of_sublease` *is dead. The node $n4$ attached with* `fact_of_nonabuse_of_confidence` *is dead because the node $n5$ attached with* `fact_of_abuse_of_confidence` *is alive. Therefore, the node directed to the root is alive, so $\mathcal{B}^{\mathcal{A}}_{\mathcal{T}}, \mathcal{P} \vdash$* `contract_end`*.*

3. Translation into Logic Programming

We propose a translation of the JUF tree $\mathcal{T}$ into the following logic program $tr(\mathcal{T})$ where each node is translated into the following rules in $tr(\mathcal{T})$.

- We put a name $name_n$ for every node n except the root.
- Let c be the conclusion and $n_1, ...n_m$ be the names of its children.
  Then, the translation for the root in $tr(\mathcal{T})$ is
  $\texttt{prove}(c)\text{: } -\texttt{prove}(n_1).$
  $\vdots$

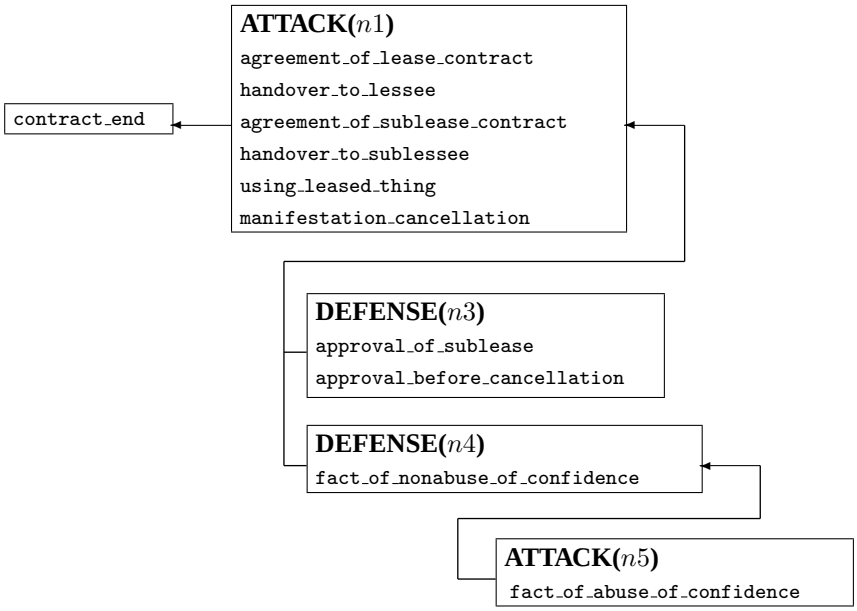

Figure 2. Block diagram without alleged fact of "term expiration"

$\texttt{prove}(c)$: $-\texttt{prove}(n_m)$.

- Let the ultimate facts attached to node n whose name is $name_n$ be $f_1^n, ..., f_{m_n}^n$ and the names put for the children of n be $d_1^n, ..., d_{k_n}^n$. Then, the translation for n in $tr(\mathcal{T})$ is the following rules

  $\texttt{prove}(name_n)$: $-\texttt{prove}(f_1^n), , ..., \texttt{prove}(f_{m_n}^n),$
  $not\ \texttt{prove}(d_1^n), ..., not\ \texttt{prove}(d_{k_n}^n)$.
  $\texttt{prove}(f_i^n)$: $-\texttt{alleged}(f_i^n), \texttt{plausible}(f_i^n).(1 \leq i \leq m_n)$

  where not expresses "Negation as Failure".

We also translate the facts in $\mathcal{A}$ and $\mathcal{P}$ into the following program $tr(\mathcal{A})$ and $tr(\mathcal{P})$ as follows.

$tr(\mathcal{A}) = \{\texttt{alleged}(f).|f \in \mathcal{A}\}$
$tr(\mathcal{P}) = \{\texttt{plausible}(f).|f \in \mathcal{P}\}$

Example 5 *Consider the JUF tree $\mathcal{T}$ in Example 2. $tr(\mathcal{T})$ becomes:*

```
prove(contract_end) :- prove(n1).
prove(contract_end) :- prove(n2).
prove(n1) :-
        prove(agreement_of_lease_contract),
        prove(handover_to_lessee),
        prove(agreement_of_sublease_contract),
        prove(handover_to_sublessee),
        prove(using_leased_thing),
        prove(manifestation_cancellation),
```

```
        not prove(n3), not prove(n4).
prove(agreement_of_lease_contract) :-
        alleged(agreement_of_lease_contract),
        plausible(agreement_of_lease_contract).
⋮
prove(n2) :- prove(expiration_of_the_term).
prove(expiration_of_the_term) :-
        alleged(expiration_of_the_term),
        plausible(expiration_of_the_term).
prove(n3) :-
        prove(approval_of_sublease),
        prove(approval_before_cancellation).
prove(approval_of_sublease) :-
        alleged(approval_of_sublease),
        plausible(approval_of_sublease).
prove(n4) :- prove(fact_of_nonabuse_of_confidence),
        not prove(n5).
prove(fact_of_nonabuse_of_confidence) :-
        alleged(fact_of_nonabuse_of_confidence),
        plausible(fact_of_nonabuse_of_confidence).
prove(n5) :- prove(fact_of_abuse_of_confidence).
prove(fact_of_abuse_of_confidence) :-
        alleged(fact_of_abuse_of_confidence),
        plausible(fact_of_abuse_of_confidence).
```

Then, we have the following theorem.

Theorem 1 *Let $\mathcal{T}$ be a JUF tree, and $\mathcal{A}$ be a set of alleged facts and $\mathcal{P}$ be a set of plausible facts. Let $tr(\mathcal{T})$, $tr(\mathcal{A})$ and $tr(\mathcal{P})$ be the translated logic programs for $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{A}$ respectively. And let $\mathcal{B}$ be a block diagram w.r.t. $\mathcal{T}$ and $\mathcal{A}$.*

Then, $\mathcal{B}^{\mathcal{A}}_{\mathcal{T}}, \mathcal{P} \vdash c$ if and only if the model M of a logic program $tr(\mathcal{T}) \cup tr(\mathcal{A}) \cup tr(\mathcal{P})$, $M \models prove(c)$.

Note that $\mathcal{T}$ is a finite tree and $tr(\mathcal{T})$ is a stratified program, we can guarantee the uniqueness of the model as well as completeness and soundness in the above theorem.

Example 6 *Consider the JUF tree $\mathcal{T}$ in example 2, a set of the alleged facts $\mathcal{A}$ in example 3 and a set of the plausible facts $\mathcal{P}$ in example 4. From $\mathcal{B}^{\mathcal{A}}_{\mathcal{T}}, \mathcal{P} \vdash$ contract_end. On the other hand, for the model of $tr(\mathcal{T}) \cup tr(\mathcal{A}) \cup tr(\mathcal{P})$, $M \models prove($contract_end$)$.*

4. PROLEG system

We have started to implement the system to reason about the JUF theory in a PROLOG-based meta-interpreter. We call the system the *PROLEG* system (PROlog based LEGal reasoning support system). By implementing the civil code into the PROLEG system, we aim at a computer-aided learning system of the JUF theory for law school students

and a support system of lawyers who choose the strategy of allegement by simulating the judges' decisions using PROLEG and scholars who seek their interpretations of the law.

The behavior of the meta-interpreter of PROLEG is the same as that of the PROLOG depth-first evaluation of the above $tr(\mathcal{T}) \cup tr(\mathcal{A}) \cup tr(\mathcal{P})$, so that if all the $\texttt{prove}(f)$ parts in a translated rule from each node are true and there is some not $\texttt{prove}(f')$ such that f' is not alleged, we just assert that the head of the rule is true without further evaluating the counter-argument. Even if f' is alleged, if the other party failed to prove f' then we can also assert that the head of the rule is true. This evaluation strategy is efficient for minimizing the fact finding task of judges.

The PROLEG system can handle the first-order terms with arguments to enhance the expressibility and produce a trace of reasoning that is useful for students or lawyers for better understanding a judge's reasoning. We will report the details of the PROLEG system elsewhere.

5. Related Work

Many works have been reported on formalizations of burden of proof[9,1,10,11,13,12].

Prakken[9] claimed that we cannot formalize a switch of burden of persuasion in nonmonotonic reasoning and introduced rather complicated functions to formalize it. We [13] challenged this claim by formalizing a switch of burden of persuasion as a decision making process of judges in stead of formalizing a switch of burden of persuasion as a turn change in dialogue game. In our work, a switch of burden is not a dynamic process during argumentation, but a tool for decision process after all the argument are already done. Since we borrowed this idea from the Japanese Civil Procedure Law and the JUF theory is a kind of realization of the idea, it is quite natural that we extended our framework to formalize the JUF theory in our current paper.

Gordon et al. handled not only a burden of persuasion but also a burden of production in their Carneades system[1]. They formalized a burden of proof as a kind of dynamic process of arguments and therefore, they needed to introduce additional mechanisms to handle these burdens. This kind of additional mechanisms were also introduced in [10,11, 12]. Although these additional mechanisms may be necessary for capturing a notion of a burden of proofs in dynamic process, we showed that a simple mechanism of "negation as failure" in logic programming is sufficient for capturing the notion of a burden of proof in decision making for judges in the Japanese Civil Code.

There are a lot of literatures in argumentation framework[3,5,4,8]. These works are general formalizations of argumentation whereas our framework is very concrete in order to formalize the JUF theory. Therefore, our formalization would be an application of these frameworks. However, more detailed analysis would be necessary for comparison and this is our future work.

In the formalization of a block diagram, we introduce "alleged facts". This allegement is actually related with "burden of production". We show a detailed analysis in [15].

6. Conclusion

We presented a formalization of the JUF theory and proposed a translation method from the JUF tree to logic programming. It is quite surprising that Japanese judges developed

the JUF theory in order to solve their problems and it coincides with the logic programming paradigm. In addition to our purpose of furthering the PROLEG project, our future work includes the application of advanced reasoning such as abductive reasoning[14] and the analogy[6] of a translated logic programming from the JUF theory, in order to show the usefulness of the abduction and analogy in the legal domain.

Acknowledgments We thank anonymous referees for their valuable comments.

References

[1] Gordon, T. F., Prakken, H., Walton, D., "The Carneades Model of Argument and Burden of Proof", Artif. Intell., Vol. 171, Nos. 10-15, pp. 875 – 896 (2007).

[2] Ito, S., "Basis of Ultimate Facts", Yuhikaku (2001) (in Japanese).

[3] Dung, P. M., "On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning and Logic Programming", Proc. of IJCAI-93, pp. 852 – 859 (1993).

[4] Dung, P. M., Dung, T. M., "A Unified Framework for Representation and Development of Dialectical Proof Procedures in Argumentation", Proc. of IJCAI-09, pp. 746 – 751 (2009).

[5] Garciá, A. J., Simari, G. R., "Defeasible Logic Programming: An Argumentative Approach", TPLP Vol. 4, Nos 1-2, pp. 95 – 138 (2004).

[6] Haraguchi, M., "A Reasoning System for Legal Analogy", Machine Intelligence, Vol.14, pp. 323 – 346 (1995).

[7] Kitamura, I., "The Judiciary in Contemporary Society: Japan", Case Western Reserve Journal of International Law, 00087254, Sprint93, Vol. 25, Issue 2 (1993).

[8] Modgil, S., Caminada, M., "Proof Theories and Algorithms for Abstract Argumentation Frameworks", Argumentation in AI (Rahwan and Simari eds.), Springer-Verlag (2009).

[9] Prakken, H., "Modelling Defeasibility in Law: Logic or Procedure?", *Fundam. Inform.*, 48(2-3), pp. 253 – 271 (2001).

[10] Prakken, H., Sartor, G., "Presumptions and Burdens of Proof", Proc. of JURIX 2006, pp. 21 – 30 (2006).

[11] Prakken, H., Sartor, G., "Formalising Arguments about the Burden of Persuasion", Proc. of ICAIL 2007, pp. 97 – 106 (2007).

[12] Prakken, H., Sartor, G., "More on Presumptions and Burdens of Proof", Proc. of JURIX 2008, pp. 176 – 185 (2008).

[13] Satoh, K., Tojo, S., Suzuki, Y., "Formalizing a Switch of Burden of Proof by Logic Programming", International Workshop on Juris-Informatics (JURISIN 2007), pp. 76 – 85 (2007). (`http://research.nii.ac.jp/~ksatoh/papers/jurisin2007.pdf`)

[14] Satoh, K., Tojo, S., Suzuki, Y., "Abductive Reasoning for Burden of Proof", International Workshop on Juris-Informatics (JURISIN 2008), pp. 93 – 102 (2008). (`http://research.nii.ac.jp/~ksatoh/papers/jurisin2008.pdf`)

[15] Satoh, K., "A Formalization for Burden of Production in Logic Programming", International Workshop on Juris-Informatics (JURISIN 2009), to appear (2009).

[16] Shindo, K., New Civil Procedure Law (Revised 3rd Edition), Kobundo publisher (in Japanese) (2005).