A Legal Reasoning System : new HELIC-II

Katsumi Nitta, Masato Shibasaki, Tsuyoshi Sakata,
Takahiro Yamaji, Wang Xianchang, Hiroshi Ohsaki (JIPDEC),
Satoshi Tojo (MRI), Iwao Kokubo (MRI) and Takayuki Suzuki (MTC)

Institute for New Generation Computer Technology
4-28, Mita 1-chome, Minato-ku, Tokyo 108, Japan
nitta@icot.or.jp

Abstract

Legal reasoning is the thinking process adopted by lawyers when they apply legal rules to a new case and draw legal conclusions. In the field of AI and Law, many legal reasoning systems have been developed. However, most of them are focused on generating arguments. Though argument is the most important function of the legal reasoning, without value judgment and debate strategy, we cannot construct a complete model of legal reasoning. The new HELIC-II is a legal reasoning system based on such a complete model. In this paper, we introduce a legal reasoning model capable of value judgment and debate strategy, and give an overview of the new HELIC-II. Especially, we show how legal knowledge is represented in the new HELIC-II illustrated by presenting the solution to an actual case.

1 Introduction

Legal reasoning is the thinking process adopted by lawyers to solve legal problems. In the field of AI and Law, many models of legal reasoning have been proposed. However, most of them are focused on the process of legal argument. Though argument process is the most important process in legal reasoning, we cannot model the lawyers' thinking process without goal generation, value judgment and debate strategy. The research target of the *new HELIC-II* is to propose a complete model of the legal reasoning, and to develop a legal reasoning system based on the model.

We have already developed a legal reasoning system HELIC-II [Nitta et al. 1992] in the FGCS project. This system (old HELIC-II) is a hybrid system which consists of two inference engines - a rule base reasoner and a case base reasoner. We showed the effectiveness of the hybrid architecture by presenting solutions to several criminal cases.

However, the old HELIC-II has the following problems.
(1) It can only generate arguments. Although it can generate many alternative arguments, there is no function to select the best one. (2) It runs only on the parallel inference machines (PIMs), and it lacks portability. (3) It is not easy to use because the user interface is poor. (4) The mechanisms which generate the arguments are too simplistic.

To resolve these problems, we started development of the *new HELIC-II* at the FGCS Follow-on project. Development embraces many topics such as knowledge representation, non-monotonic reasoning, hypothetical reasoning and analysis of legal knowledge.

In this paper, we give a brief introduction to the key concepts of the *new HELIC-II* system. In Section Two, we analyze reasoning as used by lawyers. In Section Three, we present a model of legal reasoning based on the analysis in Section Two, and in Sections Four and Five, we give an overview of the *new HELIC-II* system. In Section Four, we introduce the argument generation and selection process. In Section Five, we introduce debate strategy and discuss solutions of a ample case.

2 Analysis of Legal Reasoning

Lawyers are involved in finding solutions to many kinds of legal problems embracing legislation, consultation, fact-findings, courtroom debate and judgment. We focus on the arguments involved in applying rules to facts, and on legal judgment.

In processing a case, the prosecution focus on some important event in the problem, and generate a legal consequence (goal) which they want to achieve. The prosecution constructs an argument which supports the goal, and presents it to the defense. When the defense is presented with an argument from the prosecution, they generate another goal to rebut the argument. During courtroom debate, both parties generate (sub)goals to defeat their opponent. Their thinking process is largely goal-oriented.

On the other hand, the judge's thinking process is not always goal-oriented. Some judges observes the facts of a new case and make their decision (goal). They then formulate an argument to support it. Other judges observe the facts, list up possible goals and arguments, and select the best one.

While the actions of the prosecution and defense consist of (1) making arguments, (2) selecting arguments and (3) debate strategy, the actions of the judge consist of (1) and (2).

(1) Argument
After a goal is generated, both prosecution and defense must present arguments to support it by refering to legal rules. Legal argument differs from automated deduction because legal rules are incomplete in the following ways:

1. As it is inherently impossible to define all the necessary conditions of legal rules in sentence-form, legal rules may be incomplete. Therefore, some legal rules may conflict.

2. As some predicates that appear in legal rules are abstract, the scope of meaning of such predicates is ambiguous. Usually, facts are represented by more concrete predicates. Therefore, there exist semantic gaps between legal concepts and facts.

When lawyers apply legal rules to facts, they must interpret these rules and decide whether they apply or not, taking into account background knowledge such as social customs, industrial policy, objectives of rules and so on. This interpretation process is included in the explanation of the decision. In interpreting legal rules, lawyers often refer to legal theory or established precedents.

(2) Selecting arguments by value judgment
As legal rules conflict, we may formulate several arguments whose conclusions conflict. In such cases, we must select the most suitable one by considering some viewpoint. There are several criteria for value judgment, and some of them may conflict. When we select one argument, we must take a balanced view.

Legal norms, public opinions, social customs and ideology are examples of criteria for value judgment. Judicial precedents are important sources of value judgment.

(3) Debate strategy
In the courtroom, both parties have different knowledge (facts, precedents and theories) and different viewpoints, and they debate the case. The debate strategy consists of making arguments, extracting issues from arguments, making counter-arguments, changing goals and so on.

Debate strategy controls the components of the arguments and their selection by value judgment (Fig. 1).

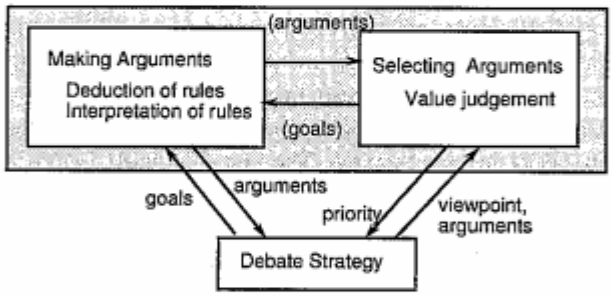

Figure 1: Basic components of legal reasoning

3 Legal Reasoning Model of the New HELIC-II

Our legal reasoning model consists of seven components.

{ Facts, Knowledge Base,
Procedure for making Argument,
Procedure for comparing arguments,
Procedure for debate, Agents }

We explain each of them below.

(1) Facts
The situation of a new case is represented as a set of facts. Facts represent agents, objects, actions, status, relation between action and status, and so on. We assume some facts may conflict because the prosecution

and the defense may present conflicting evidence.

(2) Knowledge Base
The knowledge base consists of rules, concept definitions and factors of values.

(a) Rule

Most rules take the form of "$A \leftarrow B(, notC)$." (if B then A (unless C)). There are two types of rules - absolute rules and default rules. The consequences of default rules may be denied by other rules.

Statutory rules, precedents and legal theories are rules.

(b) Concept Definition

A concept definition is a dictionary which consists of a set of concepts and subsumption relations between them. A concept definition is absolute knowledge.

(c) Criteria for Value Judgment

Some facts and rules are related with various factors (features) of value. For example, a rule from an old case may have features like "focus on social customs" and "a supreme court case".

A criteria for value judgment is defined as priority relation among factors of value. "New rules have priority over old ones" and "maintaining public order is more important than freedom of the press" are example of criteria of value. Norms and legal maxims belong to criteria of value.

(3) Procedure for making arguments
Making arguments consists of two kinds of reasoning.

(a) Deductive Reasoning

If a rule "$A \leftarrow B$" and a fact F exist and if B subsumes F, then A is drawn. It corresponds to the application of statutory rules or legal theories.

(b) Generalization

If a rule "$A \leftarrow B$" exists and if B has upper concept B_0, then the original rule is generalized as "$A \leftarrow B_0$."

Generalization occurs in two cases. In the first case, a legal rule is interpreted and its condition part is expanded. In the second case, precedents are applied to a similar new case.

Even if a rule "$A \leftarrow B$" is reliable, the generalized rule "$A \leftarrow B_0$" may not be reliable. Therefore, the generalization level B_0 must be as low as possible.

After a rule is generalized and applied to a fact F, the distance between B and F is measured. If this value is more than some threshold, we consider the rule is over-generalized.

(4) Procedure for Selecting Arguments
When two arguments are given and their conclusions conflict, then this procedure is used to compare them based on some viewpoint, and to select the best one. This procedure corresponds to a value judgment.

(5) Procedure for Debate
This procedure includes the strategy for debate and controls other procedures such as making arguments and selecting them.

It also generates a viewpoint which strengthens the priority of one argument, finds the issue, and changes (sub) goals.

(6) Agent
An agent represents the prosecution or the defense or the judge. An agent has internal states such as known facts, known rules, (sub)goal, strength of goal, and a viewpoint.
A viewpoint is defined as the priority relation among criteria for value judgment. As a criteria for value judgment is the priority relation between value factors, and as some rules are related to value factors, by selecting one viewpoint, we can decide on priorities among rules.

In the case of debate, we use two agents - the prosecution agent and the defense agent. With different knowledge and different viewpoints, they try to make arguments which defeat the opposition.

In this section, we introduced the legal reasoning model of the *new HELIC-II*. Figure 2 shows the relations among components.

4 Argumentation Function

The legal reasoning model in the previous section allows us to generate arguments, select arguments and form a debate strategy. We call generating and selecting arguments the "argumentation function." While an argumentation function is a common function of both parties and the judge, the debate function is related to only both parties. Therefore, we treat the argumentation function and debate function separately.

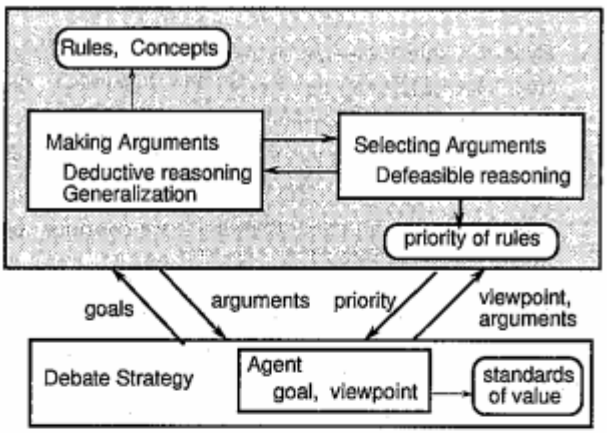

Figure 2: Legal Reasoning Model

Much research has been conducted into generating arguments [Branting 1991] [Nitta et al. 1992]. The research of [Prakken 1993] and [Sartor 1993] looked at the selection of arguments. We expanded the results of this research to develop a legal reasoning system based on the model in the previous section. To do so, we designed a new language to represent legal knowledge.

In this section, we give an overview of the argumentation function and the knowledge representation language, and in the next section, we introduce the debate function.

4.1 Overview of knowledge representation language

We describe legal knowledge in our language, and implement the generation and selection of arguments. In this chapter, we explain several important concepts of our language.

(1) Type

The primary components of knowledge representation are an "object" and an "event." We call classes of objects and events "object type" and "event type," respectively. An object type is a set of objects and it is represented by an object symbol. An event type is a set of events and it is represented by an event symbol.

Between object types, we have partial order relations ($>_o$), and between event types, we have partial order relations ($>_e$). Object types and partial order relation $>_o$, and event types and partial order relation $>_e$ construct lattices, respectively.

There are special type $\top$ and $\bot$, and for any $T \in T_o$ and $P \in T_e$, the following relations hold.

$$\top >_o T, \quad T >_o \bot, \quad \top >_e P, \quad P >_e \bot$$

For any event type, we can define another type with negation ($\neg$).

(2) ψterm and Hterm

We introduce two terms - ψterm and Hterm. A ψterm is used to define an object, and an Hterm is used to define an event or a status.

(a) **ψterm**

A ψterm is defined as an object type, or a structure constructed by a root symbol and a list of object labels as follows [Ait-Kaci *and* Nasr 1986].

$$P/person[age \Rightarrow 20, last\_name \Rightarrow X/string, \\ parent \Rightarrow person[last\_name \Rightarrow X]]$$

Here, a *root symbol* (*person*) is an object type symbol, and an *object label* consists of an object label symbol (*age*, *last_name*, *parent*) and its value (20, $X/string$, $person[last\_name \Rightarrow X]$). We can attach an object tag symbol (P, X) before an object type. To the same object tag symbols, the same ψterms are substituted.

Semantics of ψterm : Let U be a universe of ψterms and let I_o be an interpretation of ψterm, then the following relations hold.

$$I_o[\top] = U, \quad I_o[\bot] = \{\ \}$$
$$\forall S_1,\ S_2 \in T_o\ S_1 <_o S_2 \rightarrow I_o[S_1] \subset I_o[S_2]$$

A *label* is a function from U to U. Interpretation of ψterms ($\Psi_1 = P[l \Rightarrow t]$ and $\Psi_2 = P[l_1 \Rightarrow t_1, ..., l_n \Rightarrow t_n]$) is defined as follows.

$$I_o[\Psi_1] = \{x \in I_o[P] \mid \exists y \in I_o[t], I_o[l](x) = y\}$$
$$I_o[\Psi_2] = \bigcap I_o[P[l_i \Rightarrow t_i]]$$

If two ψterms ($\Psi_1 = S_1[l_1 = A_1, ...]$, $\Psi_2 = S_2[l'_1 = A'_1, ...]$) satisfy following three conditions, then Ψ_1 is called a *sub object type* of Ψ_2 and can be represented as $\Psi_1 \preceq_o \Psi_2$.

(1) $S_1 \leq_o S_2$, (2) if $l = A'$ appears in the label of Ψ_2 and if $l = A$ appears in the label of Ψ_1, then $A \leq_o A'$ holds, (3) constraints by tags of Ψ_2 are satisfied by Ψ_1.

If $\Psi_1 \preceq_o \Psi_2$ then $I_o[\Psi_1] \subseteq I_o[\Psi_2]$ holds.

(b) **Hterm**

An Hterm is defined as an event type, or a structure consisting of a root symbol and a list of event

label as follows.

$A/watch(agent = X/person,$
$\quad object = hit(agent = Y/person,$
$\quad\quad object = Z/person[sex \Rightarrow male],$
$\quad\quad cause = \#fight)$

Here, a root symbol is an event type, and an event label is a pair of an event label symbol ($agent$, $object$) and its value ($person$, $person[sex \Rightarrow male]$, $hit(agent = Y/person, object = Z/person[sex \Rightarrow male])$, $\#fight$). A label value may be a ψterm or an Hterm. "$\#fight$" is an Hterm tag, and it is replaced by another Hterm. Hterm tags appearing in facts are preceded by #, and those appearing in rules are preceded by @.

If two Hterms ($H_1 = P_1[l_1 = A_1, ...]$, $H_2 = P_2[l'_1 = A'_1, ...]$) satisfy the following four conditions, then H_1 is called a *sub event type* of H_2 and it is represented as $H_1 \preceq_e H_2$. (1) $P_1 \leq_e P_2$, (2) if $l = A'$ appears in the label of H_2, $l = A$ appears in the label of H_1 and A and A' are Hterms, then $A \preceq_e A'$ holds, (3) if $l = A'$ appears in the label of H_2, $l = A$ appears in the label of H_1 and A and A' are ψterms, then $A \succeq_o A'$ holds, (4) constraints by tags of H_2 are satisfied by H_1.

Semantics of Hterm : Interpretation of Hterm consists of interpretation of ψterm I_e and a mapping π from Hterm to $\{true, false\}$. π satisfies the following two conditions. (1) $\pi(\top) = true$, $\pi(\bot) = false$, (2) for two Hterms H_1 and H_2, if $H_1 \preceq_e H_2$, then

$\pi(H1) = true \rightarrow \pi(H2) = true$

holds.

(3) Substitution and Unification

(a) Unification of ψterm

Let X_i and Y_i be ψterms and let θ be defined as follows.

$\theta = \{X_1/Y_1, X_2/Y_2, ..., X_n/Y_n\}$

If θ satisfies the following two conditions, then θ is a *substitution* of ψterm. (1) Tags appearing in Y_i are different from tags appearing in X_j, (2) $X_i \succ_o Y_i$.

(b) Unification of Hterm

Let X_i and Y_i be event types and let η be defined as follows.

$\eta = \{X_1/Y_1, X_2/Y_2, ..., X_n/Y_n\}$

If η satisfies the following two conditions, then η is a *substitution* of Hterm. (1) Tags appearing in Y_i are different from tags appearing in X_j, (2) $X_i \preceq_e Y_i$.

(4) Rules

A rule consists of a unit name, consequence part and condition part as follows.

$U :: A \leftarrow B_1, B_2, ..., B_n$

Here, A and B_i are Hterms or Hterms preceded by *not*. "*not*" means "negation as failure".

Some rules may be generalized when applied to a new case. We distinguish such rules from others, and call them *Crules*. For Crules, we can define two kinds of information to control generalization.

(a) Limit of generalization

If a Crule is generalized without any constraint, the condition part will be meaningless because it may be generalized to $\top$. Therefore, we introduced information to restrict generalization.

For example, let the following be type definitions,

$hit <_e do\_violence,$
$person <_o animal$

and let the following Hterm have a limit of generalization.

$hit < do\_violence > (agent =$
$\quad person[name \Rightarrow" Tom"] < animal >$

This means that the original Hterm

$hit(agent = person[name \Rightarrow" Tom"])$

may be generalized as in the following examples.

$hit(agent = person)$
$do\_violence(agent = person[name \Rightarrow" Tom"])$
$do\_violence(agent = person)$

(b) Weighting of conditions

The condition part of a Crule may be consists of more than one Hterms, and each Hterm may have more than one label. Among these, some information (Hterm, label) may be essential to draw a conclusion, and others may not be important. Not important information can be generalized more than important information. To discriminate not important information from others, we introduce the notation "!".

That information is reflected to calculate the degree of generalization.

(5) SLD Resolution

Let G_i be the following goals

$\leftarrow A_1, .., A_k, ..., A_n.$

and C_i be

$A \leftarrow B_1, ..., B_m. \in Rules$

and θ be an *mgu* which satisfies $A\theta \preceq_e A_k\theta$, then the following G_{i+1} is derived from G_i and C_i.

$\leftarrow (A_1, .., A_{k-1}, B_1, .., B_m, A_{k+1}, ..., A_n)\theta.$

If C_i is a Crule, it is generalized to the upper limit, and then the generalized rule C_i' is used for resolution. In this case, the degree of generalization is evaluated after the resolution is finished. If the total degree is over the threshold, then the resolution is considered to have failed.

(6) Argument

Let F and $Conc(F)$ be a set of Hterms. IF $Conc(F)$ satisfies the following two conditions, then $Conc(F)$ is called an *answer set* of F. (1) $F \subseteq Conc(F)$, (2) if $H_1 \in Conc(F)$, and there exists an Hterm H_2 which satisfies $H_1 \preceq_e H_2$, then $H_2 \in Conc(F)$.

Let R be a set of rules, F be a set of Hterms which show facts, and $Ext(R, F)$ be a set of Hterms. If $Ext(R, F)$ is a minimum set of Hterms which satisfies following conditions, it is called a *correct answer set*. (1) $Conc(F) \subseteq Ext(R, F)$, (2) for any rule $A \leftarrow B_1, .., B_n. \in R$ and for any substitution θ, if $\{B_1\theta, ..., B_n\theta\} \in Ext(R, F)$, then $A\theta \in Ext(R, F)$ holds. Hterms preceded by *not* are not considered here. (3) The degree of generalization is less than the threshold, (4) Let H be a set of Hterms preceded by *not*. For any $A \in Ext(R, F)$ and $B \in H$, A and B are not unifiable. (5) Let A and B be Hterms. If $A, B \in Ext(R, F)$, then A and $\neg B$ is not unifiable.

Let G be an Hterm, F be a set of Hterms and R be a set of rules. If $F' \subseteq F$ and $R' \subseteq R$ are the minumum sets which satisfy $G \in Ext(R', F')$, then a set of instances of F' and R' is called the *argumentation of* H and it is represented as $Arg(G, R, F)$.

(7) Defeasible reasoning based on priority of rules

Each rule has a name called *unit*. Using units, we define the priority relation between rules.

(a) **Priority of rules**

We can define a unit name for not only a rule but a group of rules. For example, let r_1, r_2, r_3 be unit names of three rules. Then,

$r_0 := \{r_1, r_2, r_3\}$

defines a new unit name "r_0" which is defined for a group of $\{r_1, r_2, r_3\}$.

Priority of rules is defined as a priority name and a set of priority relations of unit names.

$p_1 := \{r_1 >_r r_2\}.$

$p_2 := \{r_1 >_r r_3\}.$

Moreover, we can define priority of priority. Priority of priority is defined as a name of priority of priority and a set of priority relations of priority names.

$view1 := \{p_1 >_s p_2,\ p_3 >_s p_4\}.$

$view2 := \{p_2 >_s p_1,\ p_3 >_s p_5\}.$

In the case of Crules, we must take care with the priority of generalized rules. For example, let r and r' be a Crule and its generalized rule. Then, we consider that the following priority is defined by default because generalized rules are less reliable than original rules

$r >_r r'$

And let r_1 and r_2 be Crules and let r_1' and r_2' be their generalized rules. If the following priority is defined

$r_1 >_r r_2,$

then we can conclude the following priority

$r_1 >_r r_2',$

however, we cannot decide the priority between r_1' and r_2.

(b) **Defeat relation**

Let A and B be Hterms. And let R and F be a set of rules and a set of Hterms.

If A and $\neg B$ are unifiable, or if $\neg A$ and B are unifiable, then A and B are said to be a *contradiction*. For two arguments $Arg(A, R, F)$ and $Arg(B, R, F)$, if A and B cause a contradiction, we say "$Arg(A, R, F)$ attacks $Arg(B, R, F)$" and "$Arg(B, R, F)$ attacks $Arg(A, R, F)$". For any argument $Arg(G, R, F)$, if $Arg(G, R, F) \subset Arg(A, R, F)$ and $Arg(A, R, F)$ attacks $Arg(B, R, F)$, then we say $Arg(B, R, F)$ is a *counter argument* of $Arg(G, R, F)$.

Let $Arg(B, R, F)$ be a counter argument of $Arg(G, R, F)$, and let r_2 and r_1 be top default rules including $Arg(B, R, F)$ and $Arg(G, R, F)$. If one of the following conditions holds, then we say $Arg(B, R, F)$ *defeats* $Arg(H, R, F)$ (Fig.3). (1) r_2

is only one default rule included in $Arg(B, R, F)$, (2) for any sub argument of $Arg(B, R, F)$ which does not include r_2, it is a *justified argument*.

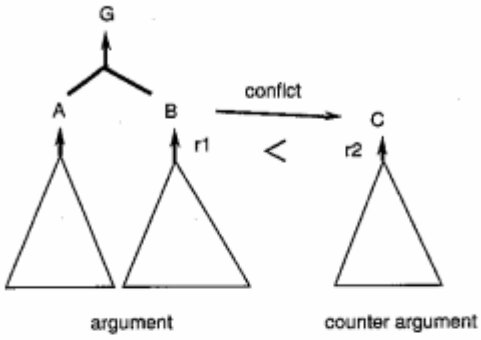

Figure 3: Defeat relation

An argument can be classified into three categories - a *defeated* argument, a *justified* argument and a *merely plausible* argument [Sartor 1993].
A defeated argument is an argument which is defeated by some counter argument. A justified argument is an argument which defeats any counter arguments. A merely plausible argument is an argument which is neither a defeated one nor a justified one. Justified arguments and merely plausible ones are called *plausible* ones.

(c) **Query Mode**

Our language has two query modes - analysis mode and normal mode. In the analysis mode, our language conducts OLDT resolution [Tamaki and Sato 1986] and outputs plausible arguments which support the given goal. OLDT resolution calculates all arguments, all counter arguments, and all counter-counter arguments and so on, and it compares them.

In the normal mode, it calculates arguments and counter arguments, but it doesn't calculate counter-counter arguments. It outputs (temporally) plausible arguments, because such arguments may be defeated by counter-counter arguments.

4.2 Description of Legal Knowledge

In this chapter, we show how legal knowledge is represented.

(1) Concepts

Common-sense knowledge about concepts is represented using type definitions. For example,"hitting other people is a violent action" and "father is a male" are represented as follows.

$$hit < do\_violence$$
$$father[sex : male]$$

We have to define which object types are related to which object labels, and which event types are related to which event labels as type definitions. However, some event labels such as "agent", "object", "a_object", "condition", "cause", "time", "place", "manner", "implement" and so on are predefined [EDR].

(2) Facts

A new case is represented as a set of facts. A fact is represented as a rule whose condition part is "*true*".
The prosecution and the defense may have different evidence which conflicts as in following example.

$$act :: hit(agent = tom, object = bill) \mid \#act.$$
$$pla :: with\_criminal\_intent(a\_object = \#act).$$
$$def :: by\_negligence(a\_object = \#act).$$
$$res :: injured(a\_object = bill, cause = \#act).$$

Such conflict is resolved if priority relations between evidence is defined.

(3) Statutory rules, Legal Theories

As most legal rules take the form of "if - then - unless rules", they are easily represented as rules. The following are examples of articles 36 and 199.

$$penal36 :: punishable(a\_object = @act) \leftarrow act(agent = X/person, object = Y/person) \mid @act, not\ self\_defense(a\_object = @act).$$
$$penal199 :: crime\_of\_homicide(a\_object = @act) \leftarrow act(agent = X/person, object = Y/person) \mid @act, kill(a\_object = @act).$$

As predicates which appear in legal rules are vague, supplementary rules are needed to make the meaning of vague concepts clear. For example, "If a person X did violence to another Y with criminal intent, and Y died, and there is legal causality between the action and the death, then X killed Y." is a supplementary rule to define the meaning of "kill." Supplementary rules are also represented by rules.
Legal theories concerning interpretation of legal rules are also supplementary rules. As different lawyers may

adopt different interpretations, there may exist rules which conflict.

(4) Judicial precedents
Precedents contain information such as facts, a final decision, arguments of both sides, and the argument of the judge. Arguments included in precedents consist of several levels of rules. While rules appearing near the conclusion consist of more general conditions, rules near facts consist of concrete conditions.

Though rules of the former level are applied as they are, rules of the latter level are applied after they are generalized. Therefore, the latter level rules are represented as Crules.

(5) Criteria for value judgment and viewpoint
When lawyers select one from conflicting interpretations of legal rules, they evaluate each interpretation based on their own viewpoint. We describe a personal viewpoint as a priority relation between criteria of value.

The following are examples of criteria of value. Legal norms belong to criteria for value judgment.

- New law has priority over old law.
- Public discipline must be protected.
- People must be treated equally.

Criteria of value are represented by a priority name and priority between factors of value.

$$focus\_on\_economy := \{economy > polution\}.$$
$$> case\_of\_localCourt\}.$$
$$focus\_on\_PublicDiscipline :=$$
$$\{morals > freedom\_of\_press\}.$$

The relation between rules and factors of value is represented as the definition of a new unit.

$$economy := \{r_1, r_4, r_7\}.$$
$$polution := \{r_2, r_6\}.$$
.....

Using two kinds of information, if we select one dimension, then this defines the priority relation between rules.

A *viewpoint* is a priority relation of criteria of value. As a criteria of value defines the priority of rules, a viewpoint is a priority of priority of rules.

$$v1 :=$$
$$\{ focus\_on\_economy > focus\_on\_PublicDiscipline,$$
$$focus\_on\_consistency\_of\_interpretation$$
$$> focus\_on\_flexibility\_of\_law \}.$$

(6) Query
After we give this knowledge to the *new HELIC-II*, we focus on an event of facts of a new case, and input a query as follows.

$$? - punish(a\_object = \#hit,$$
$$goal = crime\_of\_inflicting\_injury).$$
$$? - prove(a\_object = \#hit, goal = X/crime).$$

A former query is used to obtain arguments to support a given goal. The latter is used to obtain a goal and its arguments.

5 Debate Function

In the previous section, we showed that the defeat relation of arguments is defined by the priority between rules. However, that definition has two problems.

The first one is that it is assumed that there is only one definition of priority of rules and that both parties have the same knowledge. However, in the actual case, both parties have different evidence, precedents and viewpoints. Therefore, there may exist more than one priority relation of rules.

The second problem is that it is assumed that all arguments, counter-arguments, counter-counter arguments, and so on are calculated. In the actual case, when both parties make arguments, they don't always consider the counter arguments and the counter-counter arguments because they cannot predict the opponent's action. Furthermore, from the viewpoint of computation, it take a long time to calculate all arguments, counter-arguments, counter-counter arguments, and so on.

One promising approach which resolves these problems is the debate model. The debate model simulates the actual legal process in the court, and it generates better arguments in a shorter time.

Already research into the debate model has been conducted by [Rissland et al. 1987] [Loui 1992] [Gordon 1993]. Our model is different from theirs in that we focused on the difference of viewpoints of both sides and combined this into the argumentation function.

5.1 Overview of Debate Strategy

When both parties debate in the court, there are two kinds of issues. The first one is whether there exists any facts or not (issues of fact finding). The other is which interpretation is more suitable (issues of selecting interpretation, issues of value judgment). We model the debate strategy as follows.

1. Initially, two parties have different facts, different rules and different viewpoints. They don't know what rules and viewpoints the opposite side may have.

2. The defeat relation of arguments are extension of the definition explained in the previous section. If both parties have the same priority $r_1 > r_2$, we use this priority to decide the defeat relation of arguments. However, if the plaintiff insists $r_1 > r_2$, and the defendant insists $r_2 > r_1$, $r_1 \sim r_2$ ("$\sim$" means there is no priority relation), is used to decide the defeat relation.

3. Both parties present their claims to each other. There are several claims as follows. (1) To make a new argumentation for a given goal. (2) To find issues in the arguments posed by the opponent. (3) To select one issue and to make a counter argument to it. (4) To decide if an argument is defeated by a counter argument or not. (5) To modify one's own viewpoint to make one's argument defeat the counter argument. (6) To change the issue.

4. There are several strategies to selecting one issue from a list of candidate issues and to select one argument from possible arguments. From them, we adopted the depth first approach in which an issue moves deeply from an argument to a counter-argument, and from a counter-argument to a counter-counter argument, and so on (Fig.4). If the debate process falls into deadlock, backtracking occurs and another issue and another argument are selected.

5. During the debate process, the current viewpoint of each party may be enhanced by attaching new priority relations of criteria of value. For example, let $v_1 := \{p_1 > p_2\}$ be a current viewpoint, and let $p_1 > p_3$ be a priority by which an argument of this side defeats a counter argument of the opponents side. Then, v_1 is modified and becomes $v_2 := \{p_1 > p_2,\ p_1 > p_3\}$. If there are more than one candidate to be attached (for example, $p_1 > p_3$ and $p_2 > p_3$ are candidates), then we must select one. By selecting different candidates, the debate takes a different process (Fig.5).
If, by attaching a new priority, the priority relation of rules which held in the past in the debate process is altered, then this new change causes a contradiction. Therefore, the party has to find another priority relation to be attached, and if there is no other priority to make the argument defeat the counter argument, the party loses the issue. They must change the issue, and start another debate.

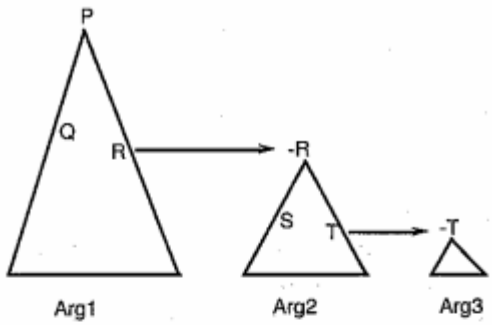

Figure 4: Depth first strategy of debate

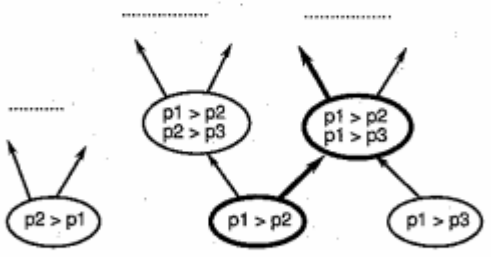

Figure 5: Changing a viewpoint

5.2 Example of Debate

We will show an example of the debate process using *new HELIC-II*. We selected this example from the lawyers.

(1) A new case (Mary's case)
Mary had hated Jane for a long time and wanted to hurt her. Mary waited Jane in the street, and hit her in the face. Jane had a bad fall, and she lost consciousness. Mary thought Jane was dead. Then Mary

took away Jane's handbag in order to make other people that Jane was killed during robbery. Mary threw the handbag in the river. Which crime should Mary be punished for?

(2) Issues in Mary's case

This case contains several hard issues concerning interpretation of the Japanese Penal Code. One issue is whether hitting Jane is the crime of inflicting an injury or the crime of violence.

The second issue is whether taking the handbag is punishable as the crime of robbery, the crime of theft, or the crime of embezzlement. How to evaluate Mary's intent affects the conclusion.

The third issue is whether abandoning a handbag is punishable as the crime of damage to property.

Concerning the above issues, there are several theories and precedents, and different lawyers support different interpretations depending on their viewpoints.

(3) The debate function

Let's solve this problem using the *new HELIC-II*. In this example, the system takes the part of the prosecution and the user takes the part of the defense.

When we start up the initial window of *new HELIC-II* and input data files, two windows appears. One is for the prosecution and the other is for the defense. From the prosecution window, we input a query "for what crime is taking a handbag punishable?" Then, the initial goal becomes "the crime of robbery", and the system generates an argument for this goal (Fig. 6).

Then, from the other window, the user lists up the issues, selects one and makes the counter-argument.

The *new HELIC-II* finds the counter argument defeats the original argument, and changes the issue.

As in these processes, we can simulate the debate process by changing claims.

(4) Example of debate process

The following is an explanation of one of the debate processes which the *new HELIC-II* can generate. In this example, (P) is the prosecution claim and (D) is the defense claim.

1. (P) According to Professor Otsuka, taking away property corresponds to the crime of theft unconditionally. Therefore, taking away a handbag is the crime of theft.

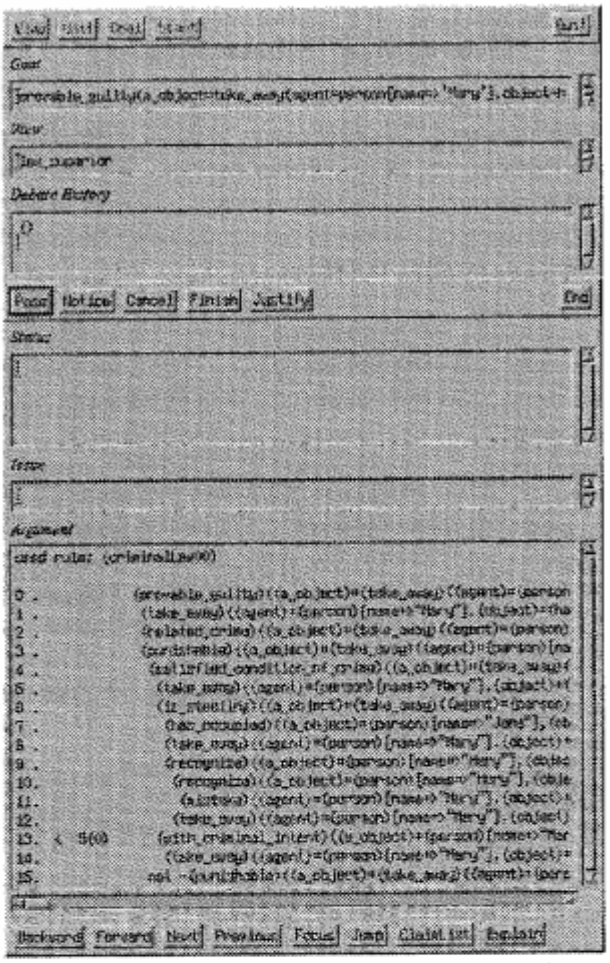

Figure 6: Initial argument

2. (D) According to Professor Dando, to punish someone for the crime of theft, criminal intent to deceive another into believing that he or she is the owner of the property must be shown. As Mary did't have this intent, her action is not the crime of theft.

3. (P) From the viewpoint of protecting property, Professor Otsuka's theory is relevant.

4. (D) From the viewpoint of distinguishing between theft and embezzlement, Professor Dando's theory is more relevant. There are many precedents to support Otsuka's theory.

5. (P) Concerning Dando's theory, there is a influential counter theory. According to Professor Hirano, to punish for the crime of theft, criminal intent

to use the property must be shown. It was a long time from taking away the handbag to throwing it in the river. So, it is reasonable to conclude that Mary intended to use it herself.

6. (D) Professor Dando's theory is relevant from the viewpoint of consistency of the system of criminal law.

7. (P) As regards the flexibility of law, Professor Hirano's theory is supported by many lawyers.

8. (D) Mary threw away the handbag in the end. So she had no intention to use it. The prosecution's opinion concerning her intent to use it is not reliable.

The relation among arguments in this debate process is shown in Fig.7.

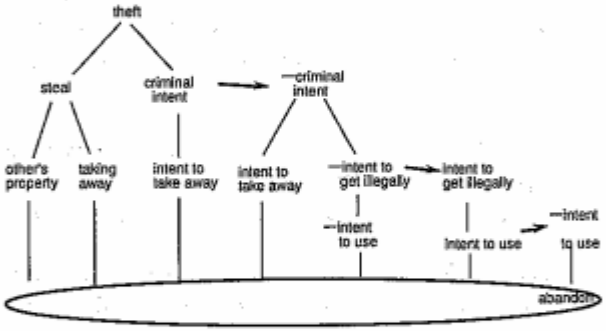

Figure 7: Relation of arguments

6 Conclusion

We have introduced the *new HELIC-II* system. Here is a summary of our evaluation of the system.

(1) As a software tool for legal reasoning
The reasoning model of the *new HELIC-II* contains important components of legal reasoning such as generating arguments, value judgment and debate strategy. It can also treat various legal knowledge such as statutes, precedents, legal theories, criteria of value evaluation, personal viewpoints and debate strategies. We can thus use the *new HELIC-II* to simulate many aspects of legal reasoning. The following are example of using of the *new HELIC-II*.

- If we give a new case and a complete viewpoint, then the *new HELIC-II* outputs the most reliable conclusion and its argument based on that viewpoint.
- If we give a new case, a desirable goal and a complete viewpoint, then it outputs an argument which achieves the goal.
- If we give a new case, all possible conclusions and their arguments are obtained.
- If we give an argument and a counter argument, a viewpoint on which counter argument defeats the original argument is obtained.
- Debate by both parties is simulated between the system and the user.

As the *new HELIC-II* is implemented on KLIC, it runs on the Unix environment. User-friendly interface has made the system easier to use.

(2) As a language for legal knowledge
Using our knowledge representation language we were able to describe a variety of legal knowledge. For example, in book one of the penal code, we described chapter 7 (Non-constitution of a Crime and Reduction or Remission of Penalty), and in book two, we described chapter 26 (Crime of Homicide), chapter 27 (Crime of Inflicting Injury), chapter 30 (Crime of Desertion), chapter 36 (Crime of Theft and Robbery), and chapter 38 (Crime of Embezzlement). We described 10 legal theories, 80 precedents, 20 criteria for value evaluation and 30 cases of common-sense knowledge. Furthermore, we constructed a conceptual dictionary for criminal cases which contains about 500 concepts.

Rules are sometimes generalized when they are applied to a new case. This mechanism is a very powerful generator of the interpretation of legal rules. However, as the current generalization mechanism is too simple, it takes a long time to apply general rules to a new case and it generates useless rules.

We need some functions to control generalization, and this is the topic of future research.

(3) As a Debate Model
One of the important features of the *new HELIC-II* is that the personal viewpoint plays an important role during debate. As both parties have different viewpoints, there two priority relations of rules. Therefore, the debate strategy becomes more complex than [Sartor 1993] [Prakken 1993] [Gordon 1993] and

[Loui 1992].

In criminal cases, interpretation of the law is not as important as fact finding. However, debate comparing interpretations is a useful tool for educating students at law school.

(4) Evaluation of the Knowledge Base

The knowledge base of the *new HELIC-II* consists of a conceptual dictionary, the penal code, theories of interpretation, precedents, and criteria for value evaluation. As our research focuses on the inference mechanism, the knowledge base is an experimental one and it is a small part of the penal code. However, we believe that the quality of represented rules is high.

And apart from the *new HELIC-II* project, Shibasaki analyzed the logical aspect of penal code in detail, classified legal rules in the penal code and represented them as rules [Shibasaki et al. 1994]. His research will help us to improve quality of a rule base.

(5) Future Projects

The *new HELIC-II* consists of several inference modules. Though each module is rather simple, the system achieved high level inference functions overall. Following is a list of future projects.

- Flexible control mechanism for generalization.
- Theoretical research concerning generalization and analogical reasoning.
- User's language to describe debate strategy.
- Extension of defeasible reasoning which can calculate defeat relation effectively.
- Extension of our language by employing temporal logic, deontic logic and logic for knowledge.
- Developing Legal knowledge base with lawyers.

Acknowledgements

In conducting this research, discussions with Professor Hajime Yoshino (Meijigakuin University) and members of his legal reasoning project was very useful. In addition, we would like to express our appreciation for their suggestions to Dr. Thomas Gordon (GMD), Dr. Ronald Loui (Washington University), and Dr. Giovanni Sartor (CIRFID).

References

[Ait-Kaci *and* Nasr 1986] H. Ait-Kaci and R. Nasr, LOGIN: A Logic Programming Language with Built-In Inheritance, In *Journal of Logic Programming*, pp. 185-215, 1986.

[Branting 1991] K. Branting : Integrating Rules and Precedents for Classification and Explanation: Automatic Legal Analysis, ph D. thesis, Univ. Texas, 1991.

[EDR] User's manual, EDR, 1984.

[Gordon 1993] T. Gordon : The Pleading Game - An Artificial Intelligence Model of Procedural Justice, ph D. Thesis, GMD,1993.

[Loui 1992] R. Loui : Computing Specificity, Research Report, WUCS-92-46, Washington Univ., 1992.

[Nitta et al. 1992] K. Nitta, et al. : HELIC-II: A Legal Reasoning System on Parallel Inference Machine, Proc. Int. Conf. on FGCS92, 1992. pp. 1115 - 1124.

[Nitta et al. 1993] K. Nitta, et al. : A Computational Model for Trial Reasoning, Proc. Int. Conf. on Artificial Intelligence and Law, 1993. pp. 20 - 29.

[Prakken 1993] H. Prakken : Logical Tools for Modeling Legal Argument, Ph D. thesis, Vrije Universiteit, 1993.

[Rissland et al. 1987] E.L. Rissland et al. : A Case-Based System for Trade Secrets Law, Proc. Int. Conf. on Artificial Intelligence and Law, 1987. pp. 60 - 66.

[Sartor 1993] G. Sartor : A Simple Computational Model for Nonmonotonic and Adversarial Legal Reasoning, Proc. Int. Conf. on AI and Law, 1993. pp. 192 - 201.

[Tamaki and Sato 1986] Tamaki,H and Sato.T : OLD Resolution with Tabulation, Proc. Int. Conf. on Logic Programming, 1986.

[Shibasaki et al. 1994] M. Shibasaki and K. Nitta : Defeasible Reasoning in Japanese Criminal Jurisprudence, FGCS workshop on Application of Logic Programming to Legal Reasoning, 1994.