A Computational Model for Trial Reasoning

Katsumi Nitta, Stephen Wong, Yoshihisa Ohtake

Institute for New Generation Computer Technology
4-28, Mita 1-chome, Minato-ku, Tokyo 108, Japan

{nitta,wong,ohtake}@icot.or.jp

Abstract

The purpose of this paper is to describe a computational model for legal reasoning in criminal law (i.e. trial reasoning). This logic-programming based model contains seven key components: facts of a new case, old cases, domain knowledge, meta rules, similarity matching relations, various implications, and two explicit agents, the plaintiff and the defendant, with opposing goals and reasoning strategies. The argumentation process in this model can be likened to a two-agent game. One agent puts forward an argument. The other agent recognizes the situation, generates candidates to refute the claim, and selects the best one for the next move. The game ends when any one agent can no longer make a move. Certain debate strategies of this model are illustrated in this paper with examples. In addition, the computational model presented has been used in the design and development of HELIC-II – a parallel knowledge-based system for trial reasoning.

1 Introduction

The primary source of legal knowledge in most countries is statutes. Since a statute often consists of a set of legal rules, the mode of legal reasoning with statutes is usually deduction. Legal rules, however, often contain vague and discretionary legal concepts and their meanings are not fixed until they are applied to actual cases in court. This is known in the literature as the *open texture* problem. To study such a problem, we at ICOT have been designing and developing a legal reasoning system, HELIC-II, in the domain of criminal law [Nitta et al. 92].

HELIC-II is a logic-programming based system implemented on the Parallel Inference Machine (PIM). It consists of two inference engines - a rule base engine and a case base engine. The rule base engine refers to the rule base, which contains legal rules in the form of logical formulas and draws legal conclusions by applying rules deductively. The case base engine refers to the case base which contains old cases in the form of case rules, and generates legal hypotheses by similarity based matching.

The output of HELIC-II is a set of arguments. Each argument is an inference tree whose root is the conclusion and whose leaves are the initial facts of a new case. Certain arguments are based on the plaintiff's (prosecutor's) opinions of old cases while other arguments are based on the defendant's opinions. In addition, no priority is assigned to the resultant arguments. HELIC-II solved several cases in the Japanese lawyer qualification examination. We have also had lawyers evaluate the practicality of HELIC-II. They drew two observations from their evaluation.

First, although relevant information, such as the referenced cases and opinions, is included in arguments, it was not easy to compare the different standpoints and argument premises. Such a comparison is important if we are to understand the weakness of each argument and to predict the refutation strategies of the parties. Second, the legal reasoning process should be goal-driven instead of data-driven (i.e., through the facts of a new case). In court, both parties are trying to present arguments to achieve individual goals. The plaintiff, for example, would initially aim to secure the most serious crime possible in court. The interpretation of legal knowledge by that agent is, thus, biased towards achieving this goal. Moreover, this goal-driven reasoning process is hierarchical and dynamic. A complex goal is usually decomposed into simpler, more manageable subgoals, and as new information is obtained during the debate, the agent may shift the initial goal to something less severe.

To take into account these two observations, we extend HELIC-II with the notion of two-agent debate.

There are many good research projects on the subject of legal debate, but, few of them involve the *explicit* modeling of legal agents in their prototypes. For example, HYPO [Rissland et al. 87] [Ashley 90] simulates the debate by comparing and contrasting old cases. Though HYPO can treat the change of focal points to a certain extent, it does not discuss the interpretation of legal concepts. Furthermore, it is sometimes difficult to select a set of good dimensions. In this paper, we introduce a computational model of HELIC-II that attempts to address multiple interpretations of legal concepts, incorporate various viewpoints and opinions into legal argumentation, and provide a set of strategies for debate.

The organization of this paper is as follows. Section 2 introduces the key components of the computational model. Section 3 explains debate strategies that build on these components, Section 4 shows an example of a debate and Section 5 presents the conclusion.

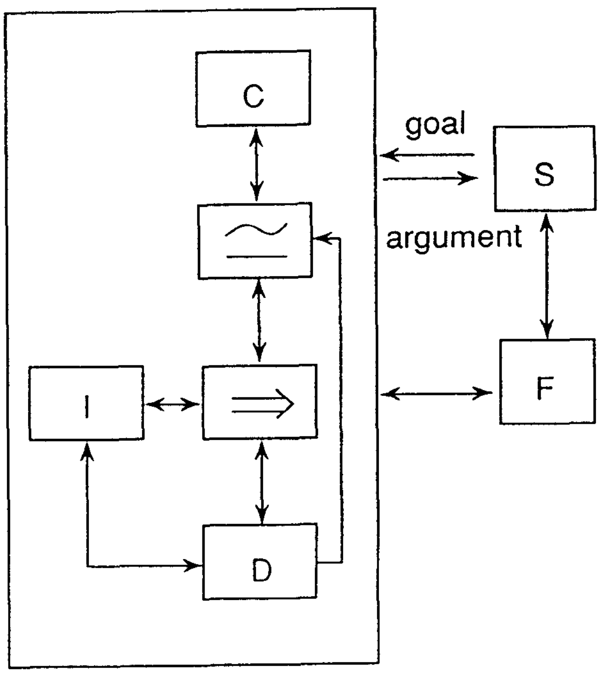

Figure 1: The Architecture of new HELIC-II

2 Computational Model

In this section, we present a computational model that guides the design and development of new HELIC-II. Figure 1 presents this architecture of HELIC-II.

The model, $\mathcal{M}$, consists of a 7-tuple:

$$\ll \mathcal{F}, \mathcal{C}, \mathcal{D}, \mathcal{I}, \simeq, \Rightarrow, \mathcal{S} \gg,$$

where $\mathcal{F}$ is a set of facts about the new case, $\mathcal{C}$ is a set of old legal cases, $\mathcal{D}$ is domain knowledge which includes entities such as domain postulates (H) and legal rules (L), $\mathcal{I}$ is meta rules such as interpretation and making hypotheses, $\simeq$ consists of a set of similarity matching relation, $\Rightarrow$ consists of a set of implication relations such as case implication ($\Rightarrow^c$) and rule implication ($\Rightarrow^r$), and $\mathcal{S}$ consists of the plaintiff / prosecutor (P) and the defendant (D) agents.

When a new case is given in $\mathcal{F}$, agent P generates a goal to be achieved and sends it to implicators ($\Rightarrow^c$ and $\Rightarrow^r$) and meta rules $\mathcal{I}$. Then, they generate an argument in the form of an inference tree, which is sent to agent D. To attack this argument, agent D generates other goal and sends it to implicators and meta rules. These processes are repeated until there is no goal to be generated.

The object language of $\mathcal{M}$ is Guarded Horn clause logic (the knowledge bases of HELIC-II are written in the parallel logic programming language KL1 [Ueda and Chikayama 90] which is based on the Flat Guarded Horn clause).

We describe each component in the following subsections.

2.1 New Case Facts

A fact in a new case is a ground predicate. Each predicate has the following object-oriented form:

$$predicate\_name(objectID, list\_of\_slots)$$

where "objectID" is an identifier which is used to refer to the instance of the predicate, and "list of slots" is a list of "attribute=variable" pairs.

For example, the predicate of name *own* is defined as follows.

$$own(objectID, [agt = X, thing = Y, price = Z])$$

where *agt* denotes the agent attibute, *thing* denotes the thing owned by that agent, and *price* indicates the price of that thing. A fact in a new case is an instance of a predicate with a unique object identifier and with *some or all* of the attributes of that predicate instantiated.

The value of an attibute can be a constant or the identifier of another object. For example, consider some facts of the following case:

$$own(own1, [agt = tom, thing = house1])$$
$$sell(sel1, [agt = tom, obj = bill, thing = house1])$$
$$own(own2, [agt = bill, thing = house1])$$
$$meets(rel1, [obj1 = own1, obj2 = sel1])$$
$$meets(rel2, [obj1 = sel1, obj2 = own2])$$

In short, these facts state that Tom owned a house and sold it to Bill, where $own1, sel1, own2, rel1$, and $rel2$ are all unique identifiers of their instances. Allen's set of temporal relations, such as *before, meets,* and *overlaps* [Allen 84], is used to represent temporal situations between objects or events in cases.

2.2 Legal Cases

Old cases contain the legal arguments from both sides. We represent each argument as a set of case rules [Branting 1989]. A *case rule* is a grounded sentence of the following form:

$$conclusion : -BG \mid predicate1 \;\&\; ... \;\&\; predicateN$$

where $predicate1, ..., predicateN$ are the antecedents of the rule, and BG is a set of background conditions that must hold before applying the rule. The background conditions include factual information of the case rule such as its identification number, the agent's opinion and goal, and legal hypotheses on which this rule is based, i.e.,

$$BG : -RuleID \;\&\; Goal \;\&\; Bcond1 \;\&\; ... \;\&\; BcondM$$

Background hypotheses are crucial in the debating process to establish the degree of certainty and the scope of applicability of that rule for a new case. In general, the only occasion when it becomes necessary to take background conditions into account and investigate what they are, is when the conclusion drawn from the case rule leads to conflict with other arguments or a change in circumstances that weakens the applicability of that rule. Thus, the involvement of background conditions make the information content of a case rule highly context-dependent. For instance, let us consider case1 below.

> **Facts:** On a cold night, Bill drank too much and fell asleep in the street. Tom ran over Bill and Bill was injured. Tom fled leaving Bill in the street, where Bill froze and died.

In this case, it is to difficult to judge Tom's liability because Bill might have frozen and died even if Tom hadn't run over him.

> **Prosecutor's Argument:** There is causality between the accident and Bill's death. The basis of this opinion is the theory that a person who injures others should give assistance even if the injury was due to negligence.

The argument of the prosecutor is represented by the following rule.

$$\begin{aligned}&caused(cause4, [cau = acc1, eff = dea1]) : -\\&\quad BG \mid\\&\quad accident(acc1, [agt = tom])\\&\quad \& \; caused(cau1, [cau = acc1, eff = inj1])\\&\quad \& \; injury(inj1, [agt = bill])\\&\quad \& \; fleed(fle1, [agt = bill])\\&\quad \& \; caused(cau2, [cau = [inj1, fle1], eff = fre1])\\&\quad \& \; freezing(fre1, [agt = bill])\\&\quad \& \; caused(cau3, [cau = [fre1, eff = dea1])\\&\quad \& \; death(dea1, [agt = bill]),\end{aligned}$$

where BG is as follows.

$$\begin{aligned}&ruleID(case1)\\&\& \; goal(causal\_relation)\\&\& \; view(prosecutor)\\&\& \; view(obligation\_of\_person\_who\_injured)\end{aligned}$$

If some condition of BG is inconsistent with the current view, this case rule isn't reffered to.

2.3 Domain Knowledge

Domain knowledge consists of domain postulates (H) and legal rules (L). An item of domain knowledge is represented by a Horn clause with a background condition similar to that of the case rule:

$$conclusion' : -BG' \mid predicate1' \;\&\; ... \;\&\; predicateN'.$$

However, the set of predicates in a domain sentence may be ungrounded. The background condition, BG', denotes its identification number, the legal goal and the view of that clause. For any ungrounded sentence A, we represent the instantiation of A as $A\theta$.

Domain Postulates: The use of domain postulates to capture various kinds of semantic relations between object sentences is an extension of Carnap's notion of meaning postulates [Carnap 58]. In $\mathcal{M}$, it covers the general knowledge found in encyclopedias and social customs as well as specific knowledge in the legal domain. For example, the common understanding of the concept of "grandmother" is denoted as follows.

$$grandmother(\_,[gm = X, obj = Y]) :- \\ view(common) \mid mother(\_,[mot = X, ch = Z]) \\ \& \ mother(\_,[mot = Z, ch = Y])$$

where view(def) is the background condition predicate indicating the type of postulate. The domain-related knowledge that the punishment for the crime of homicide is more severe than the punishment for death_by_negligence can be represented as:

$$:-view(penal\_code) \mid grave(gr1,[less = homicide, \\ more = death\_by\_negligence]).$$

Domain knowledge $\mathcal{D}$ includes the conceptual hierarchy. As an example, the following two postulates encode the "isa" relationship between the instances: person, mammal, and adult.

$$mammal(ID, Slots) :-view(isa) \mid \\ person(ID, Slots). \\ adult(ID,[age = X, Slots]) :- \\ view(isa) \mid person(ID,[age = X \mid Slots]) \\ \& \ X \geq 20$$

$\mathcal{D}$ also includes different aspects of concepts as folllows.

$$swim(ID, Slots) :-view(feature) \mid \\ whale(ID, Slots).$$

Furthermore, $\mathcal{D}$ specifies the primitive temporal relations between legal concepts (objects or events). The formulation of these temporal postulates is based on the set of axioms proposed by Allen [Allen 84]. An example is the transitivity of the *before* relation.

$$before(\_,[obj1 = X, obj2 = Z]) :- \\ view(common) \mid meets(\_,[obj1 = X, obj2 = Y]) \\ \& \ meets(\_,[obj1 = Y, obj2 = Z])$$

By applying this temporal axiom to the simple case example in Section 2.1, one can conclude that Tom owned the house before Bill.

Legal Rules: Most of the sentences contained in statutes can be represented as rules with an ungrounded form. The following is an example of criminal law in Japan.

> (Article 210:) A person who causes the death of another by negligence shall be punished for the crime of death by negligence.

Article 210 is represented in $\mathcal{D}$ as follows.

$$death\_by\_negligence(crime1,[act = Act]) :- \\ ruleID(210) \ \& \ goal(obligation\_to\_be\_attentive) \\ \mid person(A,[]) \ \& \ action(Act,[agt = A]) \\ \& \ caused(cause1,[cau = Act, eff = Death]) \\ \& \ death(Death,[agt = B]) \ \& \ person(B,[])$$

2.4 Meta Rules

Interpretation Rules: Interpretation rules are meta rules that modify object level knowledge, such as legal and case rules, to attain particular goals. An example is the following rule of "expansion."

$$(B :-BG \ \& \ goal(satisfy(G,A)) \ \& \ view(V) \mid A') \text{ if} \\ goal(satisfy(G,A)) \ \& \ (B :-BG \mid A) \in \mathcal{D} \\ \& \ \mathcal{F} \not\vdash A\theta \ \& \ \mathcal{F} \vdash G\theta \\ \& \ (A' :-view(V) \mid A) \in \mathcal{D} \\ \& \ (A' :-view(V) \mid G) \in \mathcal{D}.$$

The meaning of this meta rule is as follows. Let there be a rule $(B :-BG \mid A)$ where we wish to achieve $satisfy(G,A)$, but condition A is not satisfied. If A and G have a common upper concept A' from view V, rule $(B :-BG \mid A)$ can be expanded as $(B :-BG \ \& \ goal(satisfy(G,A)) \mid A')$ to achieve a goal.

The following simple example illustrates the function of this rule. Let the domain knowledge, $\mathcal{D}$, contain a legal rule "a person, who fails to use such care as is required in the performance of occupation and thereby kills another, shall be punished for the crime of death by negligence in the performance of work (article 211)".

$$death\_by\_negligence\_in\_work(\_,[agt = H, act = Act1]) \\ :-ruleID(211) \\ \& \ goal(attention\_to\_professional\_work) \\ \mid action(Act1,[agt = H]) \\ \& \ occupation(Act2,[agt = H]) \\ \& \ during(\_,[eve1 = Act1, eve2 = Act2]) \\ \& \ negligence(Negl,[agt = H]) \\ \& \ during(\_,[eve1 = Act1, eve2 = Negl]).$$

Let $\mathcal{F}$ contain a situation in which a student caused a traffic accident while driving a car negligently. It is not clear whether the above rule can be applied to the student.

If we wish to apply the rule to this case, the goal becomes "$goal(satisfy(driving, occupation))$."

By following the domain postulates, we know that *driving* and *occupation* have the same upper concepts of *professional work*.

$$\begin{aligned} &professional\_work(H, Sl) : -view(need\_training) \\ &\qquad \mid driving(H, Sl). \\ &professional\_work(H, Sl) : -view(need\_training) \\ &\qquad \mid occupation(H, Sl). \end{aligned}$$

Therefore, it is possible to expand "occupation" in the rule to "professional work" as follows.

$$\begin{aligned} &death\_by\_negligence\_in\_work(\_, [agt = H, act = Act1]) \\ &\quad : -ruleID(211) \\ &\quad \& \; goal(satisfy(driving, occupation)) \\ &\quad \& \; view(need\_training) \\ &\quad\quad \mid action(Act1, [agt = H]) \\ &\quad\quad \& \; professional\_work(Act2, [agt = H]) \\ &\quad\quad \& \; during(\_, [eve1 = Act1, eve2 = Act2]) \\ &\quad\quad \& \; negligence(Neg1, [agt = H]) \\ &\quad\quad \& \; during(\_, [eve1 = Act1, eve2 = Neg1]). \end{aligned}$$

Besides "expantion", meta rules include "reduction" and "analogical interpretation."

2.5 Similarity matching

Legal reasoning involves the searching of similar old cases and drawing plausible conclusions from these similar precedents. Thus, an effective and reliable means of similarity matching is crucial to the overall performance of a legal reasoning system. We first define several similarity relations between any two objects $O1$ and $O2$ in $\mathcal{M}$. Let $ID1$ and $ID2$ be objectIDs of $O1$ and $O2$, respectively.

Matching Objects:

1. (Exact matching of object identifiers)
"$ID1 \simeq_{en} ID2$" if $O1$ and $O2$ are instances of the same predicate name R. For example, if there are two objects, $person(tom, Slot1)$ and $person(bill, Slot2)$ in $\mathcal{F}$, then we denote this as

   "$tom \simeq_{en} bill$."

   This means that *tom* and *bill* are identifiers of instances of the same predicate name *person*.

2. (Similarity matching of object identifiers)
"$ID1 \simeq_{sn(V)} ID2$" if (i) $O1$ and $O2$ are instances of predicate names P and Q, and (ii) there are domain postulates which lead P and Q to the same predicate, R, from the same viewpoint, V.
For example, if $cat(cat1, Slot1)$ and $dog(dog1, Slot2)$ exist in $\mathcal{F}$, and if there is the following knowledge in $\mathcal{D}$

   $$\begin{aligned} &animal(ID, SL) : -view(isa) \mid cat(ID, SL). \\ &animal(ID, SL) : -view(isa) \mid dog(ID, SL). \end{aligned}$$

   then the following relation holds.

   "$cat1 \simeq_{sn(isa)} dog1$"

   This means that *cat*1 and *dog*1 are instanceIDs of the similar predicate names from the viewpoint of *isa* hierarchy because they are lower concept of a concept *animal*.

3. (Exact matching of objects)
"$O1 \simeq_{em(Vi)} O2$" if (i) $ID1 \simeq_{en} ID2$ or $ID1 \simeq_{sn(Vi)} ID2$, (ii) for each slot "$S = V$" in the list of slots of $O1$ there exists "$S = V'$" in the list of slots of $O2$ such that $V \simeq_{en} V'$ or $V \simeq_{sn(Vi)} V'$, and (iii) for each slot $S = V'$ in the list of slots of $O2$, there exists $S = V$ in the list of slots of $O1$ such that $V' \simeq_{en} V$ or $V' \simeq_{sn(Vi)} V$.
For example, if "$susie \simeq_{en} mary$", then

   $$\text{“}person(tom, [mother = susie]) \simeq_{em(isa)} person(bill, [mother = mary])\text{”}$$

   This means that the relation between *tom*, *mother*, and *susie* can be one-to-one mapped as a relation between *bill*, *mother*, and *mary*.

4. (Partial Matching of objects)
"$O1 \simeq_{pm(Vi)} O2$" if (i) $ID1 \simeq_{en)} ID2$ or $ID1 \simeq_{sn(Vi)} ID2$, (ii) $\neg(O1 \simeq_{em(Vi)} O2)$
For example, as there is no information concerning attribute "father" in *tom*:

   $$\text{“}person(tom, [mother = susie]) \simeq_{pm(isa)} person(bill, [mother = mary, father = dick]).\text{”}$$

This means that the relation between *tom*, *mother*, and *susie* can be partially mapped as a relation between *bill*, *mother*, *mary*, *father*, and *dick*.

Similarity matching between concepts is dependent on the particular view taken. As an example, "$whale(\_,\_) \simeq_{sm(isa)} cat(\_,\_)$" holds, but "$whale(\_,\_) \simeq_{sm(feature)} cat(\_,\_)$" does not if there isn't following rule.

$$swim(ID, Slots) : -view(feature) \mid$$
$$cat(ID, Slots).$$

Let *F1* and *F2* be two sets of predicates describing a case. We define the similarity relations between *F1* and *F2* as follows.

Matching Facts:

1. "$F1 \simeq_{ef(Vi)} F2$" iff (i) for any $O1 \in F1$, there exists $O2 \in F2$ such that "$O1 \simeq_{em(Vi)} O2$" and (ii) for any $O2 \in F2$, there exists $O1 \in F1$ such that "$A \simeq_{em(Vi)} B$".

   For example, let's consider two cases.

   F1
   $person(bill, [])$.
   $person(mary, [])$.
   $kick(kic1, [agt = bill, obj = mary])$.
   $heart\_attack(att1, [agt = mary])$.
   F2
   $person(ken, [])$.
   $person(tom, [])$.
   $knock(kno2, [agt = ken, obj = tom])$.
   $cerebral\_hermorhage(att2, [agt = tom])$.

   If domain knowledge includes the following rules

   $violence(ID, Sl) : -view(isa) \mid kick(ID, Sl)$.
   $violence(ID, Sl) : -view(isa) \mid knock(ID, Sl)$.
   $disease(ID, Sl) : -view(isa) \mid$
   $heart\_attack(ID, Sl)$.
   $disease(ID, Sl) : -view(isa) \mid$
   $cerebral\_hermorhage(att2, [agt = tom])$.

   then *bill*, *mary*, *kic*, and *att*1 can be mapped to *ken*, *tom*, *kno*2, and *att*2 respectively. Therefore, "$F1 \simeq_{ef(isa)} F2$" holds.

2. "$F1 \simeq_{pf(P,Vi)} F2$", if for some $O1 \in F1$, there exists $O2 \in F2$ such that $O1 \simeq_{em(Vi)} O2$ or $O1 \simeq_{pm(Vi)} O2$, and the rate of objects which satisfy the above condition is more than the pre-defined percentage. As an example, let us consider the following two sets of facts.

   F3
   $person(bill, [])$.
   $person(mary, [])$.
   $love(lov3, [agt = bill, obj = mary])$.
   $marry(mar3, [agt = bill, obj = mary])$.
   F4
   $person(ken, [])$.
   $person(tom, [])$.
   $hate(hat4, [agt = ken, obj = tom])$.

   Though there are seven objects in F3 and F4, *lov*3, *mar*3, and *hat*4 cannot mapped. Therefore, as the matching rate of F3 and F4 is 57% (or 4/7), "$F3 \simeq_{pf(50,isa)} F4$" holds and "$F3 \simeq_{pf(60,isa)} F4$" doesn't.

IF "$F1 \simeq_{ef(Vi)} F2$" or "$F1 \simeq_{pf(Pe,Vi)} F2$" where Pe is pre-defined constant, we consider two cases are similar from view point of Vi and denote it as "$F1 \simeq_{f(Vi)} F2$."

2.6 Implication

In this section, we briefly describe how to infer further legal arguments from cases and from domain knowledge.

Case Implication:
A legal case is represented as a set of case rules which is a grounded sentence in the form of "$(B : -BG \mid A)$." There are several ways to make use of case rules.

1. When fact $\mathcal{F}$ and legal cases $\mathcal{C}$ are given, relevant case rules are searched by similarity based matching, and a new case rule is generated as follows.

   For any $(B : -BG \mid A) \in \mathcal{C}$
   $F' \subset \mathcal{F}$ & $(F' \simeq_{f(Vi)} A)$
   $\Rightarrow^c (B\theta' : -BG \& view(Vi) \mid F')$.

   where $B\theta'$ is an operation which replaces objects in B with objects in F' according to the mapping of A to F'. The intuitive meaning of this implication is that as subset F' of a new fact is similar to condition part A of a case rule from viewpoint of Vi, a similar conclusion $B\theta'$ and its explanation $(B\theta' : -BG\&view(Vi) \mid F')$ is generated.

For example, let F' be

$person(bill, [])$.
$person(mary, [])$.
$kick(kic1, [agt = bill, obj = mary])$.
$heart\_attack(att1, [agt = mary])$.

and let a case rule be

$$caused(cau1, [cau = kno2, eff = att2]) \\ : -\ goal(liability\_of\_person\_who\_injured) \\ \&\ view(prosecutor) \mid \\ person(ken, []) \ \&\ person(tom, []) \\ \&\ knock(kno2, [agt = ken, obj = tom]) \\ \&\ cerebral\_hermorhage(att2, [agt = tom]).$$

As the condition part of this case rule can be one-to-one mapped to F' from viewpoint of "*isa*" relation, by replacing each object, the following case rule is generated.

$$caused(cau2, [cau = kic1, eff = att1]) \\ : -\ goal(liability\_of\_person\_who\_injured) \\ \&\ view(prosecutor) \mid \\ person(bill, []) \ \&\ person(mary, []) \\ \&\ kick(kic1, [agt = bill, obj = mary]) \\ \&\ heart\_attack(att1, [agt = mary]).$$

2. When fact $\mathcal{F}$, legal cases $\mathcal{C}$, goal to be achieved G and view Vi are given, implication is conducted with goal oriented and domain knowledge being restricted as follows.

$$\text{For any } (B : -BG \mid A) \in C \\ F' \subset \mathcal{F} \ \&\ (\mathcal{F}' \simeq_{f(Vi)} A) \\ \&\ (G = B\theta' \ ;\ B \in BG) \\ \Rightarrow^{c} (B\theta' : -BG\&view(Vi) \mid F').$$

The intuitive meaning of this implication is that as subset F' of a new fact is similar to condition part A of a case rule, and as the goal of case rule is the same as G, this case is used to support the certainty of conclusion G.

For example, if F' and a case rule are the same as ones presented in the privious page, and the goal is $caused(\_, [cau = kic1, eff = att1])$ and the viewpoint is $view(prosecutor)$ then by referring to the case rule, an explanation which supports the goal is generated.

$$caused(cau2, [cau = kic1, eff = att1]) \\ : -\ goal(liability\_of\_person\_who\_injured) \\ \&\ view(prosecutor) \mid \\ person(bill, []) \ \&\ person(mary, []) \\ \&\ kick(kic1, [agt = bill, obj = mary]) \\ \&\ heart\_attack(att1, [agt = mary]).$$

Note that if the viewpoint is different from view(prosecutor), this case rule isn't used.

Rule Implication:

Domain knowledge (domain postulates, legal rules) is represented as a set of non grounded rules in the form of "$(B : -BG \mid A)$."
Like case implication, there are several ways to make use of domain knowledge.

1. When fact $\mathcal{F}$ is given, relevant rules are searched by similarity based matching, and a new fact is generated as follows.

$$\text{For any } (B : -BG \mid A) \in \mathcal{D}, \\ F' \subset \mathcal{F} \ \&\ F' = A\theta \\ \Rightarrow^{r} (B\theta : -BG \mid A\theta)$$

This implication means that if the subset F' of a new fact satisfies the condition part A of rule, then a new object $B\theta$ and its expanation $(B\theta : -BG \mid A\theta)$ is generated. In other words, this implication is forward reasoning of a rule.

2. When a fact $\mathcal{F}$ and a goal G to be achieved, implication is conducted as goal oriented.

$$\text{For any } (B : -BG \mid A) \in \mathcal{D}, \\ F' \subset \mathcal{F} \ \&\ F' = A\theta \ \&\ G = B\theta \\ \Rightarrow^{r} (B\theta : -BG \mid A\theta)$$

This implication corresponds to the backward reasoning of a rule.

2.7 Agents

Every legal case involves at least two agents - a plaintiff (or a prosecutor) P and a defendant D. These agents debate about the credibility or validity of legal arguments put forward by the other in order to persuade the judges or the jury to favor their own assertion. They share the same facts about new case $\mathcal{F}$ and have similar domain knowledge $\mathcal{D}$. However, they search for old case rules $\mathcal{C}$ which support their standpoints.

Each agent has knowledge for debate strategies as rules and generates goals to be achieved.

We denote the models of a plaintiff and a defendant as $\mathcal{M}$(P) and $\mathcal{M}$(D), so that, for example, $\mathcal{M}(P) \vdash x$ means that $\mathcal{M}$(P) supports or derives x. We require that $\mathcal{M}$(P) and $\mathcal{M}$(D) do not include inconsistent predicates in them, i.e.,

$$\mathcal{M}(P) \vdash x, \;\; \mathcal{M}(P) \vdash \neg x;$$
$$\mathcal{M}(D) \vdash x, \;\; \mathcal{M}(D) \vdash \neg x.$$

But these models may derive predicates that are conflicting with one another, and this inconsistency often forms a ground for debate, i.e.,

$$\mathcal{M}(P) \vdash x, \;\; \mathcal{M}(D) \vdash \neg x;$$
$$\mathcal{M}(D) \vdash x, \;\; \mathcal{M}(P) \vdash \neg x.$$

We view the debate as a two-agent game. There are a finite set of possible moves (or actions) in the debate game. A position in a game is a finite sequence of moves. The agents (plaintiff and defendant) take turn alternatively, and each agent has only a finite number of consecutive moves each time. A strategy for either agent is a function from one position to another. A strategy for an agent is a winning strategy if the agent has successfully cornered the other agent such that there is no other position to which the latter can move. The set of moves that lead to a winning strategy (or not) can be recorded and would provide valuable insight for lawyers in actual courtroom arguments. In addition, such a game is determined if there is a winning strategy for one of the players.

In the next section, we describe our approach to making arguments in model $\mathcal{M}$ and to outlining certain debate strategies based on these approaches. The formulation and operation of the debate game will be addressed in a future paper.

3 Debate Strategies

The first move in the debate game of the model, $\mathcal{M}$, is by the plaintiff (prosecutor), who asserts an argument from one viewpoint. Such an argument is often biased (shown by "goal" predicate) and contains many hypotheses (described by the background conditions). Figure 2 shows an example inference tree which is an argument by the prosecutor. The root is the conclusion (crime) and the leaves (shaded nodes) are initial facts. Squares denote case rules. Other nodes are intermediate legal concepts.

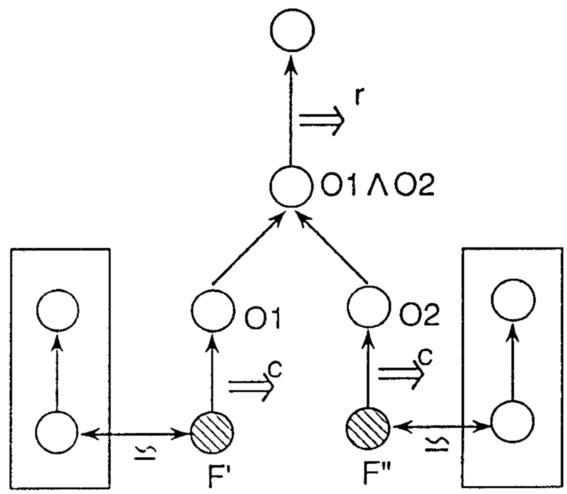

Figure 2: Inference Tree

The defendant, then, proceeds with that side's argument by scrutinizing the bias and background hypotheses of the opponent's claim. The refutation consists of three steps.

1. Listing up strategies:
   The agent determines a set of feasible strategies to refute the other side's argument. For example, there are three general types of strategies for our test cases.
   (a) To refute the similarity relation between the new case and the referenced case rule. As we have explained earlier, similarities change according to the viewpoint. So, by changing the viewpoint, we can refute the hypothesis.
   (b) To find another case rule that is closer to the new case and with a different conclusion.
   (c) To find another meta rule to refute the effect of opposite side's meta rule.

2. Evaluating strategies:
   The agent evaluates or estimates each of the feasible strategies. This involves predicting the set of possible conclusions generated by each strategy.

3. Selecting the best strategy:
   The agent selects the best refutation strategy from the domain knowledge that contains the information for comparing the appropriateness of each conclusion.

4 Example of Debate

In this section, we will show an example of debate. We will explain how each component of our model $\mathcal{M}$ contributes to the debate.
We consider the following case.

> (Example Case)
> On a winter day. Dick caused a traffic accident while driving and Jane was injured. He mistook her for dead, and he fled leaving her in the street. She froze and died.

(1) Prosecutor's argument:

1. In new fact $\mathcal{F}$, there are three actions - driving, causing accident. and fleed. Agent P finds there are two goals such as "crime of death by negligence in the performance of work" and "crime of aggravated desertations." As the latter one is more serious, $\mathcal{P}$ selects it as a first goal.

2. This goal is sent to the rule implicator. Then the rule implicator searches the rule of "crime of aggravated desertion" in the domain knowledge and tries to show that the conditions of this rule are satisfied. The implicator fails to show that the following conditions are satisfied.

   - Jane corresponds to an aged person, juvenile or deformed or sick person.
   - Dick has criminal intent to desert her.

3. For the first issue, P generates a goal "sick person" and a view "prosecutor" and sends them to the case implicator and similarity matcher. The similarity matcher finds a case rule that a person who suffered from injury by a traffic accident corresponds to "sick person" in the rule of "crime of aggravated desertion." Then, the case implicator concludes that Jane is a sick person in the rule.

4. For the second issue, P generates a goal "criminal intent" and a view "prosecutor" and sends them to the case implicator and similarity matcher again. Then, another case is found in which a person who imported opiates illegally, but he mistook the opiates as stimulants. In this case, he didn't have a criminal intent in importing the opiates because he intended to import stimulants. However, the criminal intent to import opiates is judged to exist, and the crime of "importing opiates" was applied. The case implicator concludes that Dick has criminal intent to desert her.

5. As all conditions of "crime of aggravated desertion" are met, the argument (inference tree) is sent to another agent D.

(2) Defendant's argument:

1. Agent D finds that there are two hypotheses (sick person, criminal intent) in the argument proposed by the prosecutor. Of these, D selects the latter one because referenced case rule is reliable. $\mathcal{D}$ generates a goal "$\neg$ criminal intent" and a view "defendant". These are sent to the case implicator and similarity matcher.

2. However, there is no case rule which denies criminal intent in a similar situation. Therefore, agent D tries to refute the similarity between a new case and the opiates case by changing views from "isa" to "features." Though a dead person is similar to a living person from the viewpoint of the "isa" relation, they are different from the view of "feature".

3. Agent D generates a new argument which denies Dick's criminal intent.

(3) Prosecutor's argument:

1. Agent P selects another goal "crime of death by negligence in the performance of work" and sends it to the rule implicator.

2. The rule implicator tries to prove that the conditions of this rule are satisfied. However, the rule implicator fails to show the following two conditions.

- There is the causal relation between the accident and Jane's death.
- Driving a car satisfies the concept of "occupation."

Then, the implicator generates a goal "causal relation" and sends it to the case implicator and the similarity matcher.

3. Similarity matcher finds a case whose situation is as follows.

   A woman used a gun to shoot a man and he was injured. She mistook him for dead and deserted him at the seashore, where he inhaled sand and died.

   The prosecutor's case rule is that there is a causal relation between shooting the gun and his death. By referring to this opinion, the case implicator concludes a causal relation, and sends it to the rule implicator.

4. Agent D generates goal "satisfy(driving, occupation)" to meta rules $\mathcal{I}$. $\mathcal{I}$ expand the original rule (ruleID(211)) by replacing "occupation" by "professional work," then the goal is achieved.

5. The rule implicator concludes "the crime of death by negligence in the performance of work."

(4) **Defendant's argument:**

Agent D generates a goal "$\neg$ causal relation." Agent D changes the views from "mistake" to "intent." By changing the view, the referenced case is not suitable because the referenced case includes the problem of criminal intent and Dick's case includes problems of negligence.

5 Conclusion

We have described a computational model of legal reasoning in this paper. One key distinction of this work is the explicit representation of two agents (the plaintiff and defendant) with different goals, views, and reasoning strategies. Another point is that this model has been used to design and develop a parallel logic-programming based system prototype, HELIC-II, in the domain of criminal law. This latter prototype, in turn, provides a fertile ground for experimenting and evaluating the practicality of the computational model. The current task is to incorporate the planning model used by the lawyers so that we can study and experiment more sophisticated forms of debate in HELIC-II.

References

[Allen 84] J. F. Allen, "Towards a General Theory of Action and Time," Artificial Intelligence Journal, Vol.23, No.2, 1984, pp. 123-154.

[Ashley 90] K. D. Ashley, "Modeling Legal Debate," MIT Press, 1990.

[Carnap 58] R. Carnap, Introduction to symbolic logic and its applications, Dover publications, 1958.

[Nitta et al. 92] K. Nitta et al., "HELIC-II: A Legal Reasoning System on the Parallel Inference Machine," in Proceedings of the International Conference of Fifth Generation Computer Systems, June, Tokyo, 1992, pp. 1115-1124.

[Rissland et al. 87] E.L. Rissland et al., "A Case-Based System for Trade Secrets Law," in Proceedings of the First International Conference on Artificial Intelligence and Law, Boston, MA, 1987.

[Rissland et al. 89] E.L. Rissland et al., "Interpreting Statutory Predicates," in Proceedings of the Second International Conference on Artificial Intelligence and Law, Vancourver, BC, 1989.

[Sartor 91] G. Sartor, "The structure of Norm Conditions and Nonmonotonic Reasoning in Law,", in Proceedings of the Third International Conference on Artificial Intelligence and Law, Oxford, 1991.

[Branting 1989] L.K. Branting, "Representing and Reusing Explanations of Legal Precedents," in Proceedings of the Second International Conference on Artificial Intelligence and Law, Vancouver, BC, 1989.

[Sanders 1991] K. Sanders, "Representing and reasoning about open-textured predicates," in Proceedings of the Third International Conference on Artificial Intelligence and Law, Oxford, 1991.

[Ueda and Chikayama 90] K. Ueda and T. Chikayama, "Design of the Kernel Language for the Parallel Inference Machine," The Computer Journal, Vol.33, No.6, 1990, pp. 494-500.