The Knowledge Representation of Legal Expert System LES-3.3 with Legal Meta-inference

Hajime YOSHINO 1), Tokuyasu KAKUTA 2)

1) Meiji Gakuin University Faculty of Law
2) Tokyo Institute of Technology

Abstract

A "Legal meta-inference" is an inference by which legal reasoning is controlled. This study demonstrates that the seeming incompleteness of legal knowledge is remedied by appealing to meta-inferences in legal practice. That is, it is shown that by using meta-inferences, apparent contradictions are resolved. It clarifies the logical structure of knowledge in legal meta-inferences in terms of legal meta-rules which ; stipulate legal validity for legal rules and formalizes; legal meta-inference as a type of logical reasoning. A "valid rule" for the problem to be solved is then deduced from the meta-rules together with relevant facts. In this study, a legal expert system LES-3.3 for generating legal meta-inferences has been developed on the base of the logical knowledge representation.

1 Introduction

A legal meta-inference is an inference which controls legal reasoning. It seems that knowledge about our social life contains contradictions at first glance because knowledge is cconstantly changing, increasing and decreasing, over time, place and person. Default reasoning, non-monotonic logic and so on, has tried to explore principles and methods to construct a system of inference based on incomplete knowledge like this; but these approaches don't always seem to succeeded in clarifying them sufficiently.

Contrary to these approaches we start from thesis that in the legal field knowledge about knowledge (or meta-knowledge) has been prepared so that lawyers control, willingly or unwillingly, their inference through legal meta inference by applying this meta-knowledge in order to reach reasonable conclusions according to dynamic changes of the knowledge.

In this paper, we aim to clarify knowledge structure of legal meta-inference in respect the relation between knowledge and metaknowledge, especially to the relation between rules and rules which define the validity of the rules. Moreover, our aim is also to develop an inference engine for legal meta-inference and implement a legal inference system with meta inference in this sense. The characteristics of our approach is to reconstruct legal meta inference as a logical inference.

2 An example of the Legal Meta Inference

Below, we examine an example of a legal reasoning in contract law. At first we assume certain facts and rules, namely

Case 4:
f1: An offer of "the sale of a car" by Anzai to Bernard reaches Bernard on November 7.
f2: Bernard dispatched an acceptance to Anzai on November 11.
f3: The acceptance reached Anzai on November 17.

Legal Rules:
r1: A declaration of intention made becomes effective when it reached to the other party. (Cf.: Japanese Civil Code Article 97-1)
r2: An acceptance becomes effective when it is dispatched. (Cf.: Japanese Civil Code 526-1)
r2: A contract is concluded when the notice of acceptance is dispatched. (Civil Code 526(1))

r3: An acceptance is an indication of an intention.

Let's suppose that we resolve the goal like this
Goal: "When does the acceptance become effective ?"

(a) Inference without meta inference

We get two answers by applying the above rules above. One is "the acceptance becomes effective on November 11" when rule r2 is applied, or that it becomes effective "on November 17" when rule r3 and r1 apply. These two answers contradict each other.

(b) Legal Reasoning

In the practical legal reasoning process, we get a single answer "the acceptance becomes effective on November 11th". It is the meta-inference that rejects rule r1 and applies only rule r2. We will explicate legal meta knowledge, upon which legal meta inference is based, and clarify the logical structure of legal meta inference in terms of this knowledge.

3 The Structure of Legal Meta Knowledge

3.1 Rules and Meta Rules

Legal knowledge consists of legal rules. The system of legal knowledge can be taken as a conjunction of these rules.

We can distinguish two kinds of legal rules. One is a rule that prescribes obligation, which we call an object legal rule. The other is a rule that prescribes rules, which we call "meta legal rules".

3.2 The Scope of the Validity of rules

Unlike the world of natural science, in the legal world the validity of rules is relative. The scope of their validity is limited by "time","place" and "person" and "thing".

3.3 Rules Prescribing Changes the Validity of Rules

Some meta legal rules prescribe changes in the validity of legal rules. They determine when rules become valid or null, where, to whom andto what. For instance, Article 1 of Law Concerning the Application of Laws in General of Japan determines the enforcement date of laws. Article 1 of the 1980 Vienna Sales Convention states that the convention applies to contracts for international sale of goods between parties in a certain international relation.

3.4 Rules Prescribing the Priority Relation of Rules

To avoid contradiction of legal rules, there are meta legal rules which determine the priority. For example:

pr1: An upper law derogates a lower law,
pr2: A special law derogates a general law,
pr3: A new law derogates an old law.

Meta rule pr2
It is the meta rule pr2 that settles the apparent contradiction among legal rules which are described above to solve case4. This is strictly formulated as follows:

mr4: The validity of a rule is derogated by another rule in so far as the scope of the validity of both rules overlaps when the latter rule, which is a special rule of that, and conflicts with that, becomes valid.

A rule is a special rule to the other rule, if and only if the scope of the validity of the rule in terms of persons, places and things is smaller than the other.

3.5 General Principle for the Validity of Rules

mr1: A rule R is valid for a goal G at time T, when R became valid at the time T1 before T for a goal G1 including G, and when R did not become null between time T1 and T for a goal G2 including G.

mr2: A rule R is valid for a goal G at time T, when G is included in G3 where a goal G3 is the intersection of G1 and the complement of G2, if R becomes valid for a goal G1 including G at the time T1 before T and R becomes null for a goal G2 at the time between T1 and T.
mr3: A rule becomes null when it's validity is derogated by another rule.

4. Logical Structure of Legal Meta Inference

4.1 Control the Validity of Rules and Inference

Legal reasoning is controlled by determining he validity of rules. For only valid rules can be applied to the case as premises of legal reasoning. That a legal rule "is valid" can be understood to mean that it "is true". Only valid rules, that is, true rules are applied as the premises (axioms) of an inference. Invalid rules cannot be applied. A legal rule must be valid to solve a goal at the time of the inference, at the time of the judgment, as well as the time of the event, to which legal rules are applied to get a legal judgment.

A Legal meta inference solves the meta goal the rule is valid for the goal to be solved at the time of inference as well as the event'. This inference can be formalized as an inference by first order predicate logic, in which meta rules on the validity of rules are conceived of as axioms, i.e. premises and "be_valid(rule_name,time,goal_provision)" is deduced from these axioms together with facts related to rules. In inference, a goal proposition is proved by the application of the rules proved to "be_valid" by meta inference in which meta rules are applied.

In the above quoted legal meta rules pr1, pr2 and pr3, legal rules of higher priority 'derogate' legal rules of lower priority. In my opinion for a rule to derogate another rule means that the validity of the former makes null the validity of the latter rule. In other words, the latter becomes null by the former (cf. mr3). If a rule is null, in other words, invalid or false, then it cannot be applied as a premise of the inference.

5. Legal Meta Inference System

According to the above analysis, we can construct a legal meta inference system. Our Legal Expert System LES-3.3 is an implementation of that system. This system is described with ESP (Expanded Sequential Prolog) on PSI-II, both of which are developed by ICOT (Institute for the New Generation Computer Technology).

5.1 Compound Predicate Logical Formula of the Knowledge Base on LES-3

The knowledge base on LES-3.3 contains legal knowledge about a conclusion of a contract in Japanese Civil Code as well as United Nations Convention on Contracts of International Sale of Goods. It is important to represent legal knowledge in law sentences, which is described by natural language in statutes or cases, precisely in formal language as it is in natural language. The knowledge base of LES-3 uses compoound predicate logical formula (CPF) which is invented by Yoshino. Legal knowledge -- legal rules and legal meta rules -- are represented by this CPF.

A unit of CPF is a compound predicate term, which is constructed by two terms as follows:

predicate(predicateID,CaseList)

'predicate' is a predicate name, 'predicateID' is an identifier of the predicate. 'CaseList' is a list of pairs, which represents case role of the predicate and it's filler. Each filler can be also a compound predicate term. We show an example, which represents following sentence: "Bernard's Acceptance to Anzai reaches Anzai on October 17th."

```
reach(reach1,([
   obj:acceptance(acceptance1,([
      agt:'Bernard',
      obj:OBJ_ACCEPTANCE,
      goa:'Anzai'])
   tim:11_17,
   goa:'Anzai']))
```

This formula is equivalent to the following flat formula. Compound predicate term is to be converted in to flat formula for inference.

```
reach(reach1,([obj:acceptance1,tim:11_17,goa:'Anzai'])) &
acceptance(acceptance1,([agt:'Bernard',obj:OBJ_ACCEPTANCE,goa:'Anzai'])).
```

The filler for object-case of predicate 'reach', namely 'acceptance1', is defined as a member of set 'acceptance'. Thus CPF can represent complicated situation of social events precisely.

CPFs are connected each other by propositional operator to represent a compound sentence. A legal norm sentences (legal rules) can be formalized in CPF according to it's 'legal requirement - legal effect' relation as following example,which represents art. 97-1 of Japanese Civil Law: "An indication of intention becomes effective when it reached"

```
become_effective(BE,([
   obj: indication_of_intention(IOI,([
      agt:AGT_IOI,
      onj:OBJ_IOI,
      goa:GOA_IOI])),
   tim:T]))
<-
reach(REACH,([
   obj:IOI,
   tim:T,
   goa:GOA_REACH])).
```

A Legal meta rule is represented also on the same way. A CPF for a legal meta rule has rules as a constant or a valiable for a filler of object-case of a predicate relating to the validity.

5.2 Legal Meta Inference Engine

The legal meta inference engine enhanced the inference engine in two features: First, the legal meta inference engine holds the original goal , which will be taken apart into atomic propositions, because meta rules require information about the events to be proved.
Second, it has a function to call an inference about the validity of rules. We here show a part of inference engine:

```
demo(Obj,sen(A),sen(A)):-
  demo(Obj,A,sen(A));
  ..............
demo(Obj,A1,sen(A)):-
  :get_rule(Obj!rule,rule(R,(A1<-B))),
  demo(Obj,B,B),
  ............
  :get_time_of_event(Obj!demon,A,T2),
  demo(Obj,sen(be_valid(_,[(obj:R),(tim:T2),(goa:A)])),sen(be_valid(_,[(obj:R),(tim:T2),
(goa:A)])));
  ............
```

5.3. Verification of Legal Meta Inference by Example

We shall verify the logical structure of a legal meta inference by means of following the inference process step by step. We use here the example used above.

At first, we assume as below:

If we compare rule r1 with r1, we can get:

fmrv4a: Rule r2 is a special law in comparison to rule r1.
fmrv4b: Rule r2 conflicts with the rule r1 in respect of the goal 'an acceptance comes effective'.
The time of Inference: December 31st.

According to these assumptions, our inference engine follows the steps below to prove the goal.

Rule r1 becomes a candidate to solve the goal "the acceptance became effective"(6 -- this notation refers to the line number of our inference engine listed in section 5.2). When rule r1 is applied to this case, it is proved that "the acceptance becomes effective at November 17"(7). The engine makes inference whether the rule is effective at the time of inference, December 31st (8-10). (these lines are omitted in the list we skip these steps to save space).

Next the meta inference is invoked to prove the meta goal that 'the rule r1 is effective at the time of the event November 17th (acquired from line 11) for the goal "the acceptance became effective"' (12). Rule r2 conflicts with rule r1(fmrv4b). But r2 is a special law for r1 because the former governs more specific things than the latter(fmrv4a). Therefore it is proved in the meta inference that "r2 derogates r1 on November 17th in the intersection of the their effectiveness (the part of the acceptance of the area of declarations of intentions)". Through the application of the meta rule mr3 it is proved that 'the rule r1 has no validity for the goal "the acceptance becomes effective" on November 17th '. By applying the meta rules mr1 and mr2, there is no answer to prove the current meta goal. That means that rule r1 cannot be applied to prove the goal "the acceptance became effective".

Then the engine finds the second candidate, namely rule r2(6). When the rule is applied, it can be proved that "the acceptance became effective on November 11th" (7). Thereby the meta inference is invoked to prove the new meta goal 'the rule r2 is effective at the time of event, November 11th, for the goal "the acceptance became effective"'(12). In this meta inference the engine follows the steps below. At first, by applying meta rule mr2, rule r2 became effective on November 11th to the goal "the acceptance became effective"' is proved(fv2). Second, rule r2 become null on the time November 11th is failed to prove. Then, it is proved that 'the rule r2 is valid for the goal "the acceptance became effective" on November 11th (this means that the application of the mr2 succeeds). Therefore the answer "the acceptance became effective at November 11" is to be accepted as proved.

By this process, we can get the single answer. Any steps in the process of deriving of this answer -- both object level inference and meta level inference -- are all first order predicate deductive inference. (We can ascertain this when we trace the deductive process by using the logical representations of the meta rules and the facts.)

The object level inference is made by applying the legal rules proved to 'be valid' for the goal by the meta level inference: in another words, the rules proved to 'be true'. The transition of inference from meta level to object level can be called a "transition rule".

5.4. An Example of a Running Process

We would like to present several screen hard copies of LES-3.3. Figure 1 shows that without the meta inference two contradictory answers for the goal "when the acceptance became effective" -- November 17th (when Bernard dispatched the acceptance) and November 11th (when the acceptance reached Anzai) -- can be proved. In figure 2 we chose the meta inference engine. This time, we can get only one right answer.

Figure 3 - 7 are screens of Explain Window. 'Current Goals' shows the result of the resolution of the current goal. 'Current Rule' shows the rule applied for the resolution. 'Parent Goals' shows what were the proved parent goals for the current goal. Figure 3 shows that 'the acceptance became effective on November 11th' is proved by the application of rule 2 using the engine with a meta inference. In order to see why the rule 2 was applied, we can click 'Meta(If)' and 'Meta(Ev)'. By the former we can follow the explain of the validity of the rule in terms of the time of inference and by the latter in terms of the time of the event. If we had chosen 'Meta(If)' in Figure 3 then we would come up which Figure 4, which shows that it was proved that r2 is valid for the goal 'the acceptance became effective on October 11' on December 31, the time of the inference, by the application of meta rule mr1. 'Meta level' window shows now that it is the stage of meta level 1. We can see the reason of the conclusion of the current meta inference. If we click 'Meta(If)' in the figure 4 again, then we get the figure 5. It shows that it is proved mr1 is valid for the goal 'r2 is valid for the present goal'. This kind of meta rule may be called a 'ground rule'. In order to ascertain why r1 was not applied, we can use 'LES-Tracer Window'. On Figure 6 we see that the meta goal 'r1 is valid for the goal "the acceptance became effective on October 11th" on December 31st was called and then mr1 was applied, but as a result the meta goal was failed. If we here input 'r' then we can re-try the meta goal to see why the meta goal was failed as in Figure 7. This shows that the meta goal 'r1 does not become null for the goal "the acceptance became effective on October 11" until December 12' was failed. (For the inference of this goal pr2 was applied.) On the contrary Figure 8 shows that it is proved that r2 is valid for the present goal on December 31 and also that r2 is valid for the present goal on October 11, the time of the event and therefore the present goal is proved.

6. Conclusion

We have shown we can formulate a legal meta inference as a logical inference using logical representation of legal knowledge, in which 'valid rules for the given problem to be resolved' can be deduced from legal meta rules regulating the validity of rules together with the related facts. Also we have demonstrated the systematization of this legal meta inference.

In this study, we have shown that it is possible to make a system of inferences using incomplete knowledge based on classical logical analysis and formalization. This study also proposed a method for using the relative validity of knowledge. In these points our study is meaningful not only for the development of legal expert systems but also for the development of a large scale knowledge base in general which deals with human social lives and whose knowledge is relatively valid and changes constantly.

Reference

[1] Arima & Sato: 'Non-monotonic reasoning' (Japanese), in: Fuchi (ed.) KNOWLEDGE PROGRAMMING (Japanese), (Kyoritsu Shuppan 1988) 189ff.

[2] Jackson, P., Reichgelt,H. and Harmelen, F.v. (ed.): LOGIC-BASED KNOWLEDGE REPRESENTATION (1989).

[3] McCarthy, J.: Application of circumscription to formalizing commonsense knowledge. in: ARTIFICIAL INTELLIGENCE, 28(1986), 89-116.

[4] McDermott, D.: Non-monotonic Logic II: Non-monotonic modal theories, JACM Vol.29(1982), 33-57.

[5] Poole, D.: A Logical Framework for Default Reasoning, ARTIFICIAL INTELLIGENCE 36(1988), 27-47.

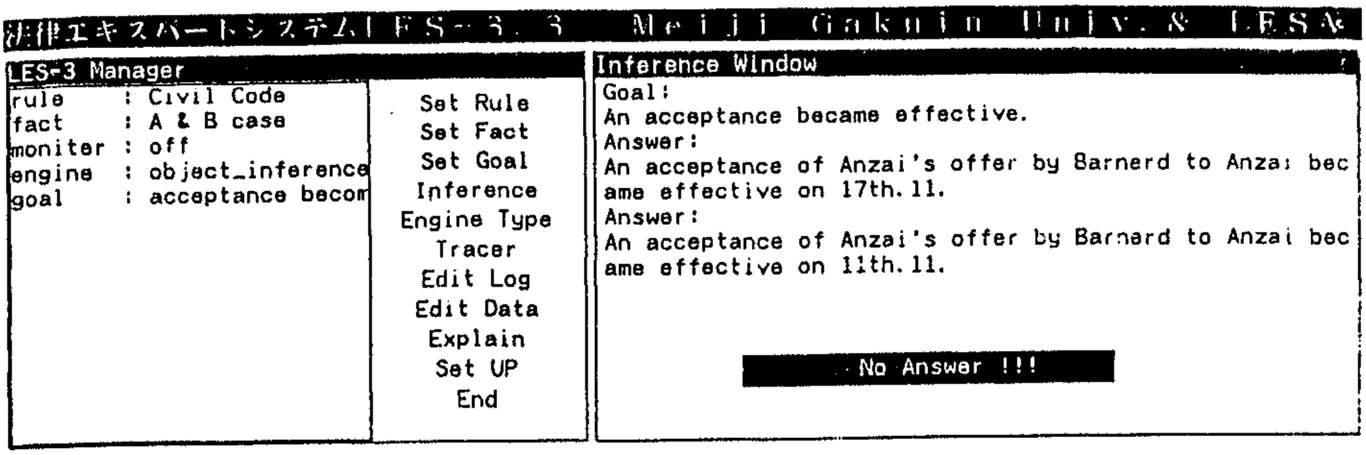
[6] Yoshino, H.(ed.): FOUNDATIONS OF LEGAL EXPERT SYSTEMS, (Japanese) Gyosei 1986.

[7] Yoshino, H.: STUDY REPORT ON LEGAL EXPERT SYSTEMS - THE PLAN I, (Japanese) Kikai System Shinko Kyokai 1988.

[8] Yoshino, H.: DEVELOPMENT OF INFRERENCE ENGINE OF LEGAL EXPERT SYSTEM, (Japanese) Kikai System Shinko Kyokai 1990.

[9] Yoshino, H.: CLARIFYING LEGAL KNOWLEDGE AND THE CONSTRUCTION OF A LEGAL KNOWLEDGE BASE, (Japanese) Kikai System Shinko Kyokai 1991.

Figure 1



法律エキスパートシステムＬＥＳ－３．３ Meiji Gakuin Univ. & LESA

LES-3 Manager

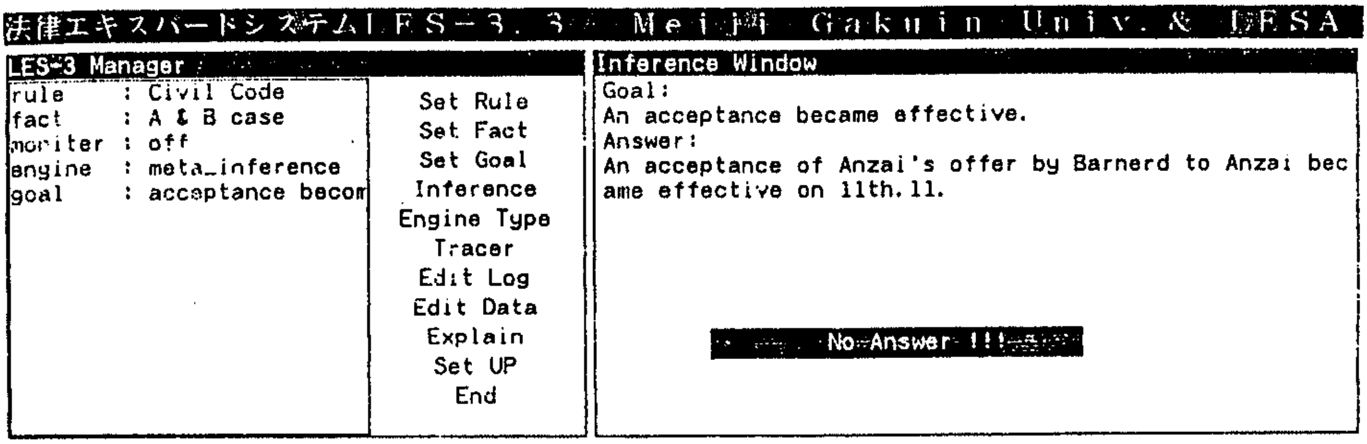
```
rule     : Civil Code
fact     : A & B case
moniter  : off
engine   : object_inference
goal     : acceptance becom
```

Inference Window

```
Goal:
An acceptance became effective.
Answer:
An acceptance of Anzai's offer by Barnerd to Anzai bec
ame effective on 17th.11.
Answer:
An acceptance of Anzai's offer by Barnerd to Anzai bec
ame effective on 11th.11.
```

No Answer !!!

Figure 2



法律エキスパートシステムＬＥＳ－３．３ Meiji Gakuin Univ. & LESA

LES-3 Manager

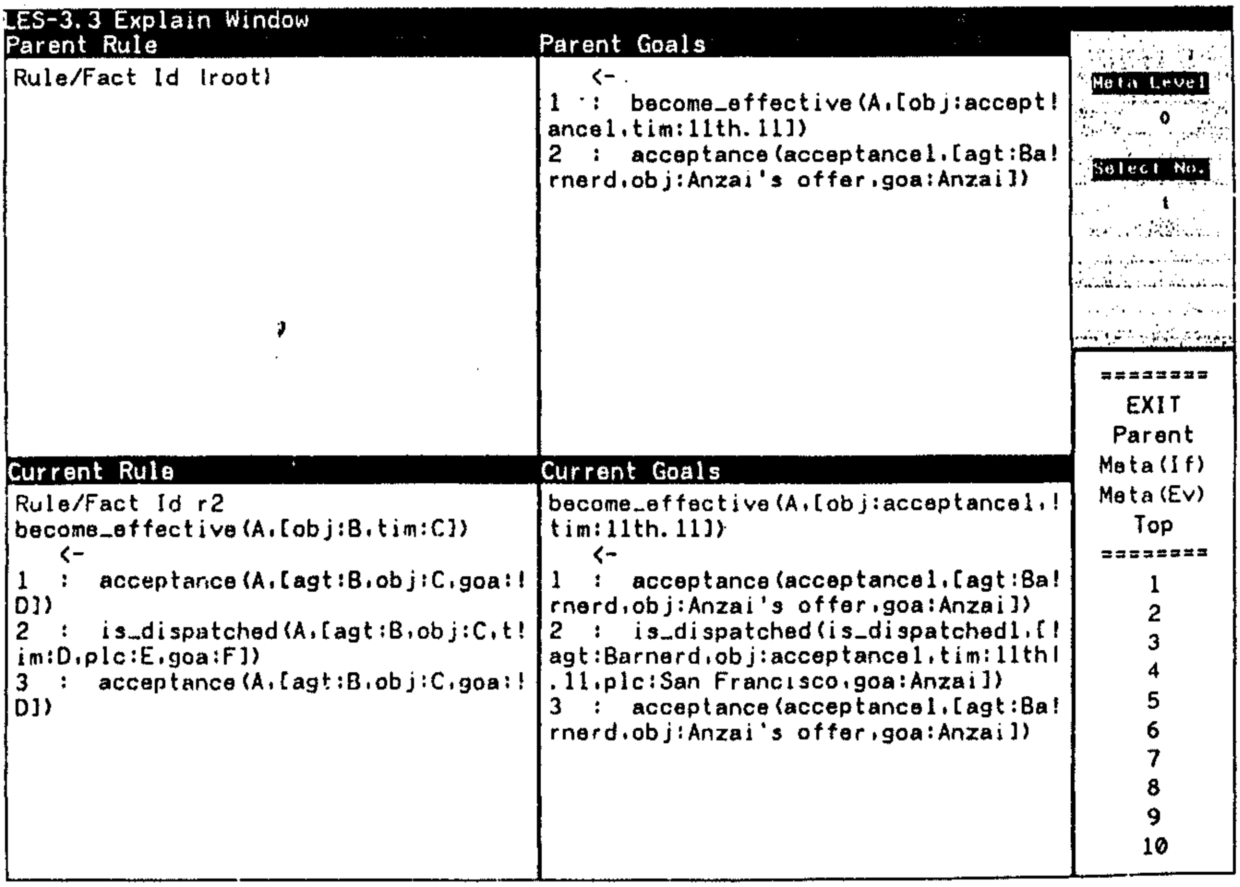
```
rule     : Civil Code
fact     : A & B case
moniter  : off
engine   : meta_inference
goal     : acceptance becom
```

Inference Window

```
Goal:
An acceptance became effective.
Answer:
An acceptance of Anzai's offer by Barnerd to Anzai bec
ame effective on 11th.11.
```

No Answer !!!

Figure 3



LES-3.3 Explain Window

Parent Rule

```
Rule/Fact Id (root)
```

Parent Goals

```
    <-
1  :  become_effective(A,[obj:accept!
ancel,tim:11th.11])
2  :  acceptance(acceptancel,[agt:Ba!
rnerd,obj:Anzai's offer,goa:Anzai])
```

Meta Level

0

Select No.

1

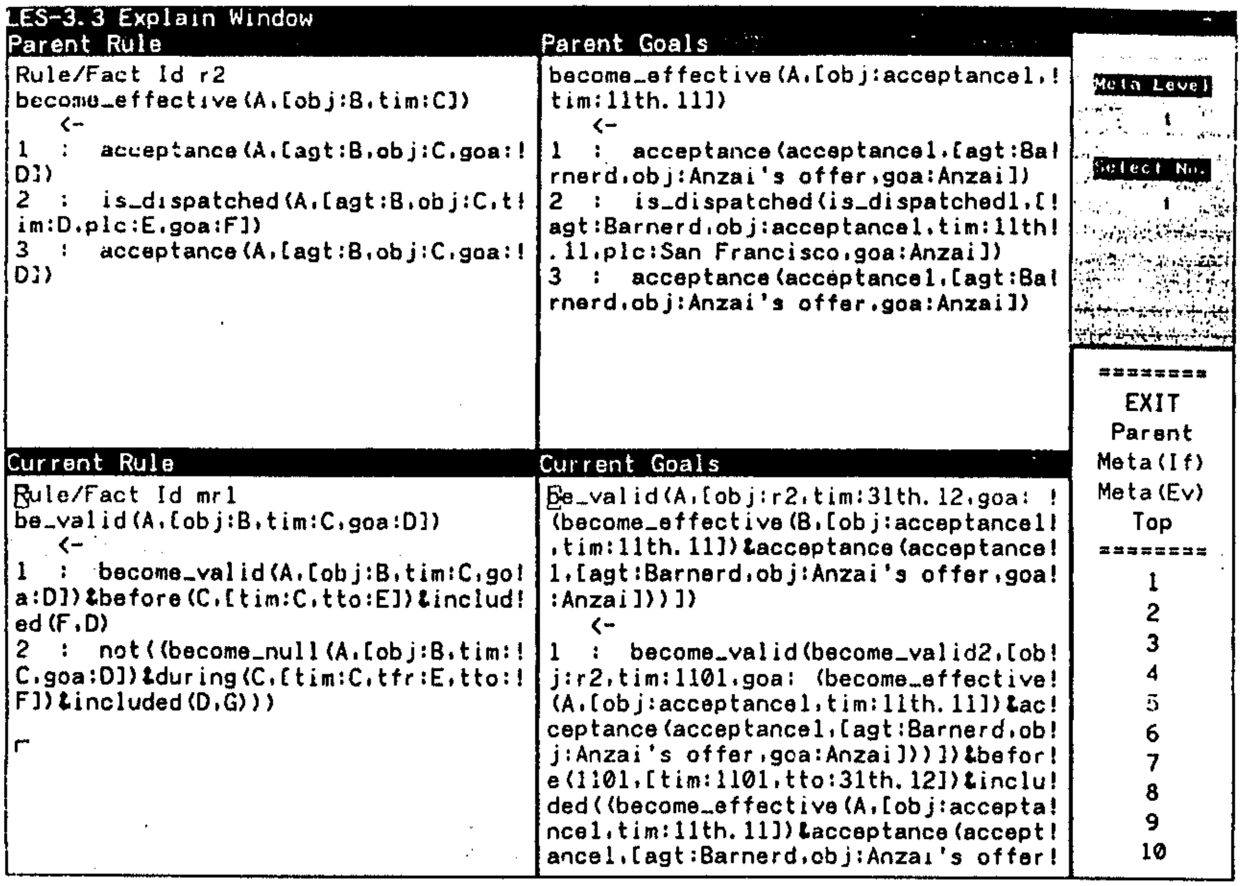
Current Rule

```
Rule/Fact Id r2
become_effective(A,[obj:B,tim:C])
   <-
1  :  acceptance(A,[agt:B,obj:C,goa:!
D])
2  :  is_dispatched(A,[agt:B,obj:C,t!
im:D,plc:E,goa:F])
3  :  acceptance(A,[agt:B,obj:C,goa:!
D])
```

Current Goals

```
become_effective(A,[obj:acceptancel,!
tim:11th.11])
    <-
1  :  acceptance(acceptancel,[agt:Ba!
rnerd,obj:Anzai's offer,goa:Anzai])
2  :  is_dispatched(is_dispatchedl,[!
agt:Barnerd,obj:acceptancel,tim:11th!
.11,plc:San Francisco,goa:Anzai])
3  :  acceptance(acceptancel,[agt:Ba!
rnerd,obj:Anzai's offer,goa:Anzai])
```

Figure 4


LES-3.3 Explain Window

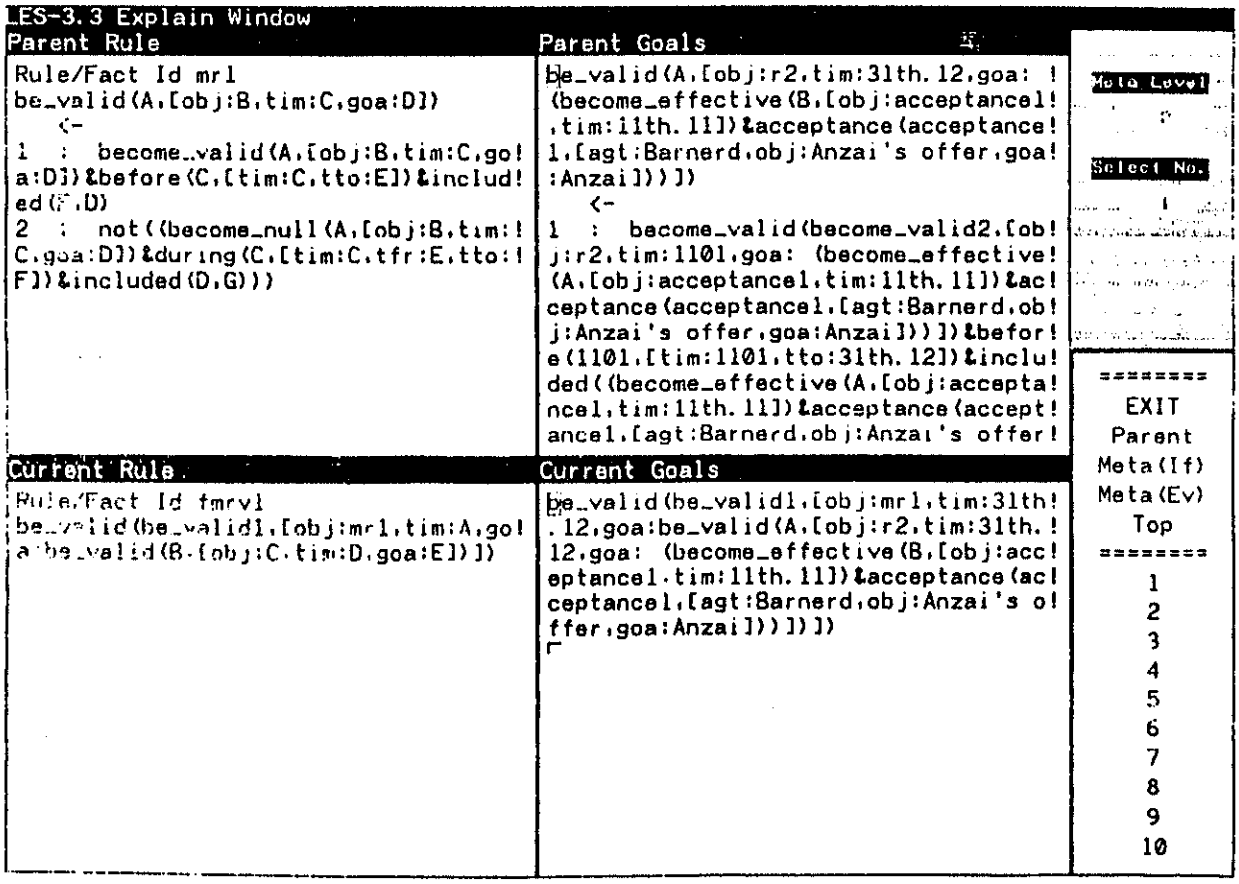
Parent Rule
```
Rule/Fact Id r2
become_effective(A,[obj:B,tim:C])
   <-
1  :  acceptance(A,[agt:B,obj:C,goa:!
D])
2  :  is_dispatched(A,[agt:B,obj:C,t!
im:D,plc:E,goa:F])
3  :  acceptance(A,[agt:B,obj:C,goa:!
D])
```

Parent Goals
```
become_effective(A,[obj:acceptance1,!
tim:11th.11])
   <-
1  :  acceptance(acceptance1,[agt:Ba!
rnerd,obj:Anzai's offer,goa:Anzai])
2  :  is_dispatched(is_dispatched1,[!
agt:Barnerd,obj:acceptance1,tim:11th!
.11,plc:San Francisco,goa:Anzai])
3  :  acceptance(acceptance1,[agt:Ba!
rnerd,obj:Anzai's offer,goa:Anzai])
```

Current Rule
```
Rule/Fact Id mr1
be_valid(A,[obj:B,tim:C,goa:D])
   <-
1  :  become_valid(A,[obj:B,tim:C,go!
a:D])&before(C,[tim:C,tto:E])&includ!
ed(F,D)
2  :  not((become_null(A,[obj:B,tim:!
C,goa:D])&during(C,[tim:C,tfr:E,tto:!
F])&included(D,G)))
```

Current Goals
```
be_valid(A,[obj:r2,tim:31th.12,goa: !
(become_effective(B,[obj:acceptance1!
,tim:11th.11])&acceptance(acceptance!
1,[agt:Barnerd,obj:Anzai's offer,goa!
:Anzai]))])
   <-
1  :  become_valid(become_valid2,[ob!
j:r2,tim:1101,goa: (become_effective!
(A,[obj:acceptance1,tim:11th.11])&ac!
ceptance(acceptance1,[agt:Barnerd,ob!
j:Anzai's offer,goa:Anzai]))])&befor!
e(1101,[tim:1101,tto:31th.12])&inclu!
ded((become_effective(A,[obj:accepta!
nce1,tim:11th.11])&acceptance(accept!
ance1,[agt:Barnerd,obj:Anzai's offer!
```

Figure 5


LES-3.3 Explain Window

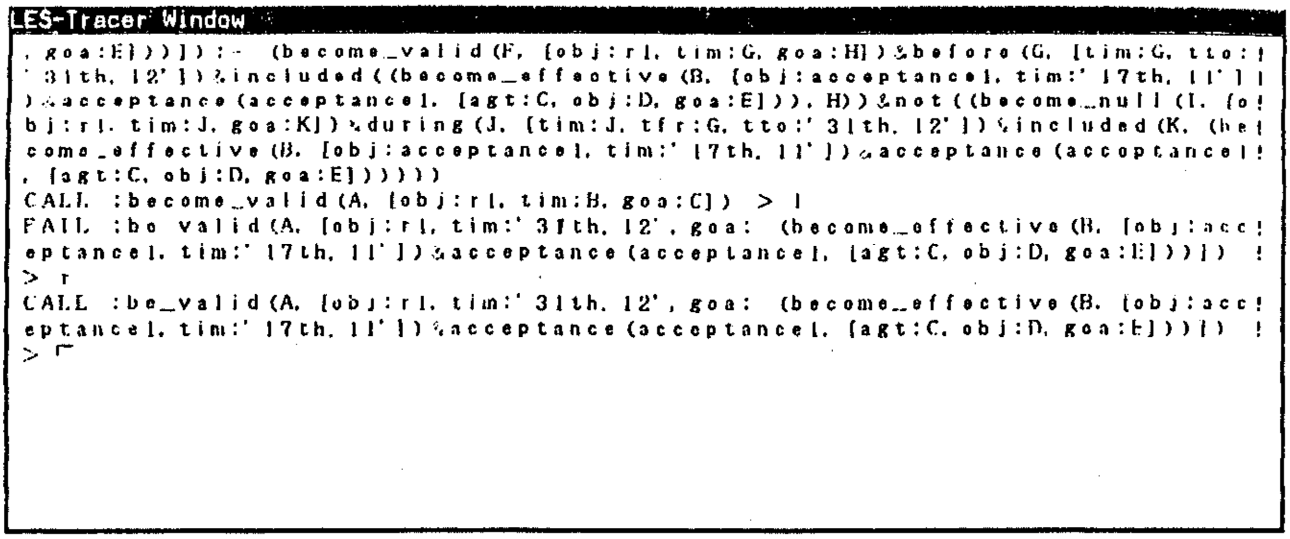
Parent Rule
```
Rule/Fact Id mr1
be_valid(A,[obj:B,tim:C,goa:D])
   <-
1  :  become_valid(A,[obj:B,tim:C,go!
a:D])&before(C,[tim:C,tto:E])&includ!
ed(F,D)
2  :  not((become_null(A,[obj:B,tim:!
C,goa:D])&during(C,[tim:C,tfr:E,tto:!
F])&included(D,G)))
```

Parent Goals
```
be_valid(A,[obj:r2,tim:31th.12,goa: !
(become_effective(B,[obj:acceptance1!
,tim:11th.11])&acceptance(acceptance!
1,[agt:Barnerd,obj:Anzai's offer,goa!
:Anzai]))])
   <-
1  :  become_valid(become_valid2,[ob!
j:r2,tim:1101,goa: (become_effective!
(A,[obj:acceptance1,tim:11th.11])&ac!
ceptance(acceptance1,[agt:Barnerd,ob!
j:Anzai's offer,goa:Anzai]))])&befor!
e(1101,[tim:1101,tto:31th.12])&inclu!
ded((become_effective(A,[obj:accepta!
nce1,tim:11th.11])&acceptance(accept!
ance1,[agt:Barnerd,obj:Anzai's offer!
```

Current Rule
```
Rule/Fact Id fmrv1
be_valid(be_valid1,[obj:mr1,tim:A,go!
a:be_valid(B,[obj:C,tim:D,goa:E])])
```

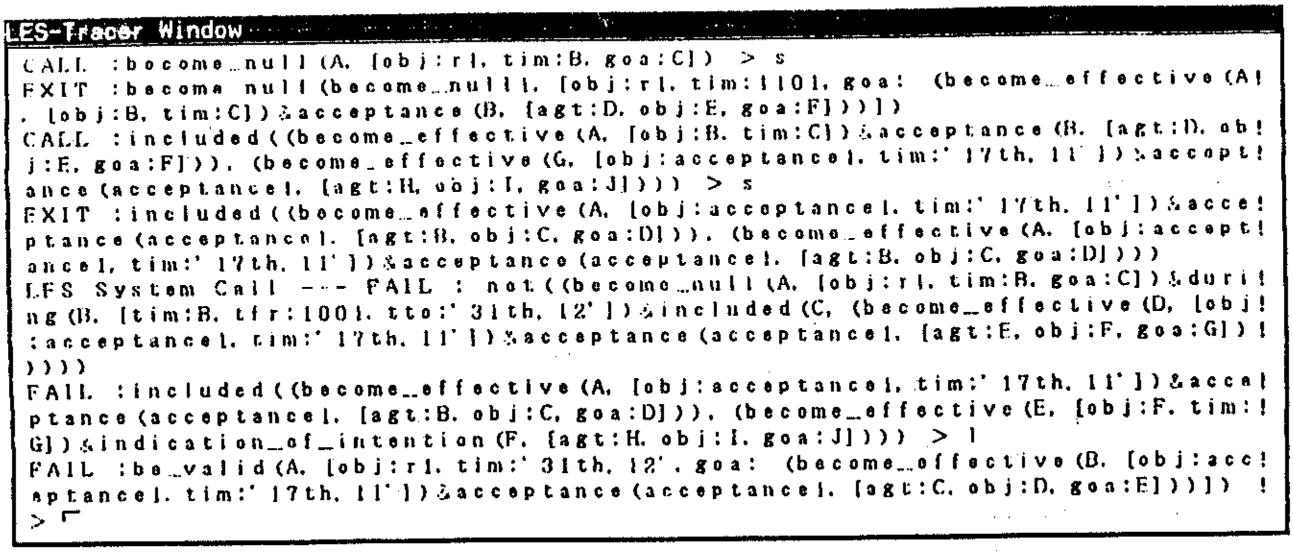
Current Goals
```
be_valid(be_valid1,[obj:mr1,tim:31th!
.12,goa:be_valid(A,[obj:r2,tim:31th.!
12,goa: (become_effective(B,[obj:acc!
eptance1,tim:11th.11])&acceptance(ac!
ceptance1,[agt:Barnerd,obj:Anzai's o!
ffer,goa:Anzai]))])])
```

Figure 6



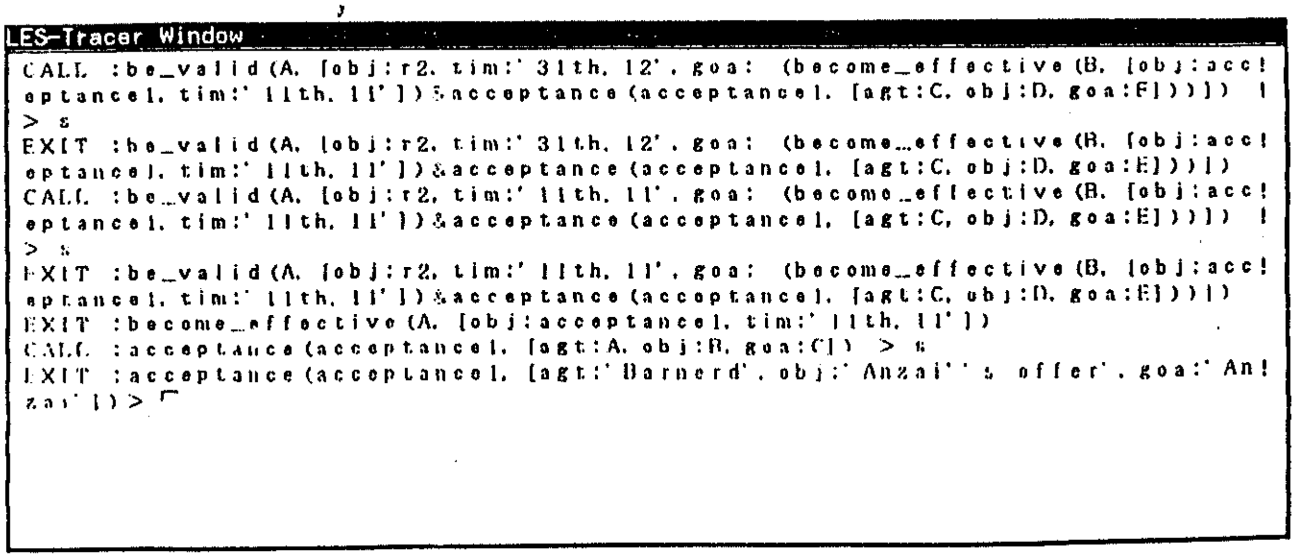
```
, goa:E]))]):- (become_valid(F, [obj:r1, tim:G, goa:H])&before(G, [tim:G, tto:!
'31th, 12'])&included((become_effective(B, [obj:acceptance1, tim:'17th, 11']!
)&acceptance(acceptance1, [agt:C, obj:D, goa:E])), H))&not((become_null(I, [o!
bj:r1, tim:J, goa:K])&during(J, [tim:J, tfr:G, tto:'31th, 12'])&included(K, (be!
come_effective(B, [obj:acceptance1, tim:'17th, 11'])&acceptance(acceptance1!
, [agt:C, obj:D, goa:E])))))
CALL :become_valid(A, [obj:r1, tim:B, goa:C]) > l
FAIL :be valid(A, [obj:r1, tim:'31th, 12', goa: (become_effective(B, [obj:acc!
eptance1, tim:'17th, 11'])&acceptance(acceptance1, [agt:C, obj:D, goa:E]))]) !
> r
CALL :be_valid(A, [obj:r1, tim:'31th, 12', goa: (become_effective(B, [obj:acc!
eptance1, tim:'17th, 11'])&acceptance(acceptance1, [agt:C, obj:D, goa:E]))]) !
> ⌐
```

Figure 7



```
CALL :become_null(A, [obj:r1, tim:B, goa:C]) > s
EXIT :become null(become_null1, [obj:r1, tim:1101, goa: (become_effective(A!
, [obj:B, tim:C])&acceptance(B, [agt:D, obj:E, goa:F]))])
CALL :included((become_effective(A, [obj:B, tim:C])&acceptance(B, [agt:D, ob!
j:E, goa:F])), (become_effective(G, [obj:acceptance1, tim:'17th, 11'])&accept!
ance(acceptance1, [agt:H, obj:I, goa:J]))) > s
EXIT :included((become_effective(A, [obj:acceptance1, tim:'17th, 11'])&acce!
ptance(acceptance1, [agt:B, obj:C, goa:D])), (become_effective(A, [obj:accept!
ance1, tim:'17th, 11'])&acceptance(acceptance1, [agt:B, obj:C, goa:D])))
LES System Call --- FAIL : not((become_null(A, [obj:r1, tim:B, goa:C])&duri!
ng(B, [tim:B, tfr:1001, tto:'31th, 12'])&included(C, (become_effective(D, [obj!
:acceptance1, tim:'17th, 11'])&acceptance(acceptance1, [agt:E, obj:F, goa:G])!
))))
FAIL :included((become_effective(A, [obj:acceptance1, tim:'17th, 11'])&acce!
ptance(acceptance1, [agt:B, obj:C, goa:D])), (become_effective(E, [obj:F, tim:!
G])&indication_of_intention(F, [agt:H, obj:I, goa:J]))) > l
FAIL :be_valid(A, [obj:r1, tim:'31th, 12', goa: (become_effective(B, [obj:acc!
eptance1, tim:'17th, 11'])&acceptance(acceptance1, [agt:C, obj:D, goa:E]))]) !
> ⌐
```

Figure 8



```
CALL :be_valid(A, [obj:r2, tim:'31th, 12', goa: (become_effective(B, [obj:acc!
eptance1, tim:'11th, 11'])&acceptance(acceptance1, [agt:C, obj:D, goa:F]))]) !
> s
EXIT :be_valid(A, [obj:r2, tim:'31th, 12', goa: (become_effective(B, [obj:acc!
eptance1, tim:'11th, 11'])&acceptance(acceptance1, [agt:C, obj:D, goa:E]))])
CALL :be_valid(A, [obj:r2, tim:'11th, 11', goa: (become_effective(B, [obj:acc!
eptance1, tim:'11th, 11'])&acceptance(acceptance1, [agt:C, obj:D, goa:E]))]) !
> s
EXIT :be_valid(A, [obj:r2, tim:'11th, 11', goa: (become_effective(B, [obj:acc!
eptance1, tim:'11th, 11'])&acceptance(acceptance1, [agt:C, obj:D, goa:E]))])
EXIT :become_effective(A, [obj:acceptance1, tim:'11th, 11'])
CALL :acceptance(acceptance1, [agt:A, obj:B, goa:C]) > s
EXIT :acceptance(acceptance1, [agt:'Barnerd', obj:'Anzai''s offer', goa:'An!
zai'])> ⌐
```

Legal Expert System Association (LESA)
The 6th International Symposium

LEGAL KNOWLEDGE AND LEGAL REASONING SYSTEMS

Proceedings of the Symposium

October 24-25, 1992
Meiji Gakuin University
Tokyo, Japan

Symposium Host
Legal Expert System Association
Sponsors
Ministry of Education, Science and Culture
(Grant-in-aid for Scientific Research)
Meiji Gakuin University
Institute for New Generation Computer Technology
Laboratory for International Fuzzy Engineering Research

Edited by
Hajime Yoshino