Defeasible Reasoning in Japanese Criminal Jurisprudence

Masato Shibasaki Katsumi Nitta

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

Abstract

To design a description language for Japanese criminal jurisprudence, defeasible reasoning based on rule priorities is a useful reasoning mechanism. But that model which have been proposed don't have sufficient functions to represent criminal rules. We expanded it by introducing absolute rules and Negation As Failure and narrowed the definition of justified argument. To add to this explanation, we show how Japanese criminal jurisprudence is represented by defeasible reasoning.

1 Introduction

Legal argumentation is one aspect of legal reasoning. It is used to justify a legal judgement by generating explanation based on legal statutes. Legal argumentation consists of three processes.

(i) Finding facts: it recognizes concrete facts.

(ii) Interpretation of law: it obtains legal norms by clarifying the meaning of statutes.

(iii) Application of law: it draws legal conclusions by applying norms to facts.

In the research of the new HELIC-II [Nitta et al. 1994], we were able to deal satisfactorily with legal argumentation by combining rule-based reasoning, case-based reasoning and debate as meta level reasoning. In parallel with constructing the new HELIC-II system, we conducted a detail analysis of the application of law, assuming that we have all the facts and have interpreted the law. Though objective of the research of new HELIC-II is to simulate the reasoning process of lawyers, objective of the research is to design a knowledge representation language for the criminal jurisprudence .

In case of criminal law, as there is the principle of the legality of crimes and punishment, the interpretation of legal statutes should be quite limited. Therefore in the case of the theory of criminal jurisprudence, the meaning of statutes is clear and consistent. In addition, criminal jurisprudence has the followin features.

• Confrontation between theories is severe.

• It is not so important as civil jurisprudenc to represent duty, possibility, permission, etc.

When legal theories contradict each other, creating a legal problem, we can draw the desired conclusion by giving preference to applying certain legal theory. To deal with these situations, defeasible reasoning based on priorities between rules [Sartor 1993][Prakken 1993] (in this paper, we say defeasible reasoning for short) is useful, but its knowledge representative power is poor.

In this paper , we focus on defeasible reasoning. In Section 2, we describe an extended version of [Sartor 1993]. In Section 3, we give an overview of the knowledge base, we are constructing, and our use of defeasible reasoning . In Section 4, we give our conclusions.

2 Defeasible Reasoning

2.1 Original Framework

We use the framework of defeasible reasoning proposed in [Sartor 1993], with rules of the form

$n{:}p0 \leftarrow p1, \ldots, pn,$

where n is the rule name, and each pi is a literal. A literal is a formula q or $\neg q$, where q is an atom, and $\neg$ is interpreted as classical negation. We also admit degenerate rules with an empty body. A rule name n is a label of the form

$r(X1, \ldots, Xn),$

where r is a new function symbol, and $X1, \ldots, Xn$ are the free variables in the rule. We write $n1 < n2$ to mean that rule $n2$ is preferred to rule $n1$. The < relation is transitive and irreflexive. Let Π be an inconsistent set of ground rules. (We consider each open rule as the set of its ground instances, i.e., those instances not containing variables.) The set of the statements derivable from a rule set Σ, i.e., the extension of Σ, denoted as $E(\Sigma)$, can be obtained by applying repeatedly the rules in Σ to the result of the previous application of those rules (starting from the empty set), until no new consequence can be obtained in this way (a fix point has been reached).

Firstly we define argument, subargument, counter argument, and attack.

Definition 2.1.1.

A is an argument for p in Π iff A is minimal among the consistent subsets A of Π such that $p \in E(A)$.

Definition 2.1.2.

An argument $A1$ is a subargument of argument A iff $A1 \subseteq A$.

An argument $A1$ is a strict subargument of argument A iff $A1 \subset A$.

Definition 2.1.3.

An argument B is a counter argument to an argument A iff B is an argument for q and A includes a subargument $A1$ for $\overline{q}$ (where $\overline{q}$ denotes the complement of q).

Definition 2.1.4.

An argument B is an attack to an argument A iff B is an argument for q and A is an argument for $\overline{q}$.
(attack is not used in [Sartor 1993], but it is used in [Prakken 1993])

ex1)

If $\Pi = \{n1(a):g(a) \leftarrow r(a),\ n2(a):r(a) \leftarrow p(a),\ n3:p(a),\ n4(a):\neg r(a) \leftarrow q(a),\ n5:q(a)\}$, then argument $B = \{n4(a), n5\}$ is a counter argument of $A = \{n1(a), n2(a), n3\}$, and an attack of $A1 = \{n2(a), n3\}$.

Secondly we define the defeat relation.

Definition 2.1.5.

An argument B defeats an argument A iff B directly defeats a subargument $A1 \subseteq A$.

Definition 2.1.6.

argument B directly defeats an argument $A1$
iff i) argument B is an attack of argument $A1$,
ii) top($A1$) < top(B) (top(A) is a rule whose head is p if A is an argument for p),
iii) if B has a strict subargument, all strict subarguments of B are justified arguments.

Defeat is an irreflexive and asymmetric relation between arguments.

Finally we define the kind of arguments.

Definition 2.1.7.

(1) Defeated argument is an argument which is defeated by any counter argument.
(2) Justified argument is an argument which defeats all attacks that are not defeated by another argument.
(3) Merely plausible argument is an argument which is not a defeated argument or a justified argument.

Plausible argument is a justified argument or a merely plausible argument.

However the definition of defeat and the definition of justified argument are mutual recursive, Definition 2.1.6., when B doesn't have a strict subargument, is a terminal condition.

ex2)

In ex1), if $n4(a) < n2(a)$, B directly defeats $A1$ and defeats A. A and $A1$ are defeated arguments. B is a justified argument.

2.2 Absolute Rule

From now on, we denote the rule name from a function to a string. The same rule name may be put on different rules. The reason that rule names were defined as functions in [Sartor 1993] is that the exception to the rule is treated by means of defeasible reasoning. But we don't treat the exception to the rule by means of defeasible reasoning. We only use a rule name to describe priority between rules.

We introduced absolute rules in addition to defeasible rules. Absolute rules are used to change an expression (this resembles meaning postulates in [Sartor 1993]). By introducing absolute rules, we can compare two arguments whose consequences don't conflict with each other. For example, consider the following case.

"John intended to kill Mary's dog, and shot it with a gun. But the bullet hit Mary, and she died as a result."

This example is described as follows.

```
f1 : person(john).
f2 : person(mary).
f3 : dog(marys_dog).
f4 : action(shot_gun, john, marys_dog).
f5 : phenomenon(hitting, mary).
f6 : phenomenon(death, mary).
f7 : causality(action(shot_gun, john, marys_dog),
                              phenomenon(death, mary)).
f8 : recognition(john, action(shot_gun, john, marys_dog),
                         action(shot_gun, john, marys_dog)).
f9 : foresight(john, causality(action(shot_gun, john, marys_dog),
                    phenomenon(death, marys_dog)).
f10 : intention(action(shot_gun, john, marys_dog),
               crime_action(killing, john, marys_dog)).
f11 : ¬ phenomenon(hitting, marys_dog).
f12 : ¬ phenomenon(death, marys_dog).
f13 : ¬ recognition(john, causality(action(shot_gun, john, marys_dog),
                              phenomenon(death, mary)).
```

If action is different from intention, and the action and the intention correspond to different crimes, then this is called a mistake at the abstract level. In this case there are two theories : a substantial interpretation theory and an abstract interpretation theory. In the former theory, if the situations of two crimes are completely different, the intended crime is treated as an attempt and the actual crime is treated as negligence. In the latter theory, the less grave crime is treated as accomplishment. If the intended crime is more grave, it is treated as an attempt. If the actual crime is less grave, it is treated as negligence. The following is a part of thsee theories.

```
substantial_interpretation_theory:
attempt(action(A1, X, W1), crime_action(CA, X, G))
  ← action(A1, X, W1),
      person(X),
      phenomenon(P1, Y),
      causality(action(A1, X, W1), phenomenon(P1, Y)),
      recognition(X, action(A1, X, W1), action(A2, X, W2)),
      foresight(X, causality(action(A2, X, W2),
                              phenomenon(P2, Z)),
      crime_result(phenomenon(P1, Y), C1),
      crime_result(phenomenon(P2, Z), C2),
```

```
        different(Y, Z),
        different(C1, C2),
        ¬recognition(X, causality(action(A2, X, W2),
                                   phenomenon(P1, Y)).,
        intention(action(A1, X, W1),
                  crime_action(CA, X, G)),
        ¬homogeneity(C2, C1).
```

```
abstract_interpretation_theory:
accomplishment(action(A1, X, W1), crime_action(CA, X, G))
    ← action(A1, X, W1),
        person(X),
        phenomenon(P1, Y),
        causality(action(A1, X, W1), phenomenon(P1, Y)),
        recognition(X, action(A1, X, W1), action(A2, X, W2)),
        foresight(X, causality(action(A2, X, W2),
                               phenomenon(P2, Z)),
        crime_result(phenomenon(P1, Y), C1),
        crime_result(phenomenon(P2, Z), C2),
        different(Y, Z),
        different(C1, C2),
        ¬recognition(X, causality(action(A2, X, W2),
                                   phenomenon(P1, Y)),
        intention(action(A1, X, W1),
                  crime_action(CA, X, G)),
        less_grave(C2, C1).
```

different is a predicate symbol, which tests identity.

We assume the following data.

```
a1 : ¬homogeneity(destruction_of_things, homicide).
a2 : less_grave(destruction_of_things, homicide).
a3 : crime_result(phenomenon(death, X), homicide)
         ←person(X).
a4 : crime_result(phenomenon(death, X),
         destruction_of_things)←dog(X).
```

We consider the following arguments.

Let arg_s_i_t be an argument for

attempt(action(shot_gun, john, marys_dog),
crime_action(killing, john, marys_dog)),

and consists of substantial_interpretation_theory, f1, f4, f6, f7, f8, f9, f10, f13, a1, a3, and a4. top(arg_s_i_t) is a substantial_interpretation_theory.

Let arg_a_i_t be an argument for

accomplishment(action(shot_gun, john, marys_dog),
crime_action(killing, john, marys_dog))

and consists of abstract_interpretation_theory, f1, f4, f6, f7, f8, f9, f10, f13, a2, a3, and a4. top(arg_a_i_t) is a abstract_interpretation_theory.

arg_s_i_t concludes that John's action is an attempt to kill Mary's dog. arg_a_i_t concludes that John's action is the accomplishmentment of killing Mary's dog.

These conclusions can turn out to be inconsistent, with the knowledge that attempt and accomplishmentment are incompatible. The following rules are absolute rules to represent that knowledge.

```
absolute1:
¬ attempt(action(A1, X, W1), crime_action(CA, X, G))
  ← accomplishment(action(A1, X, W1),
                    crime_action(CA, X, G)).
```

```
absolute2:
¬ accomplishment(action(A1,X, W1), crime_action(CA, X, G))
  ← attempt(action(A1, X, W1), crime_action(CA, X, G)).
```

arg_a_i_t', gaind by adding absolute1 to arg_a_i_t, concludes

¬ attempt(action(shot_gun, john, marys_dog),
crime_action(killing, john, marys_dog)).

arg_s_i_t', gaind by adding absolute2 to arg_s_i_t, concludes

¬ accomplishment(action(shot_gun, john, marys_dog),
crime_action(killing, john, marys_dog))

If priority is

{substantial_interpretation_theory <
abstract_interpretation_theory},

arg_a_i_t' defeats arg_s_i_t. (and arg_a_i_t defeats arg_s_i_t').

If priority is

{abstract_interpretation_theory <
substantial_interpretation_theory},

the reverse is true.

We formalize the introduction of absolute rules. Let Π be a set of ground rules, which is divided into Γ, a consistent set of absolute rules, and Δ, an inconsistent set of defeasible rules. Γ may be empty. We revise Definition 2.1.1. as below.

Definition 2.1.1'.

A is an argument for p in Π iff A is minimal such that $p \in E(A)$ and $\Gamma \cup A$ is consistent.

Because of the introduction of absolute rules, not only A must be consistent, but $\Gamma \cup A$ must also be consistent.
We revise Definition 2.1.6. as below.

Definition 2.1.6'.

argument B directly defeats an argument $A1$

iff i) argument B is an attack of argument $A1$,

ii) $\forall r_{A1} \in top(A1)\ \exists r_B \in top(B) \quad r_{A1} < r_B$,

and $\forall r_{A1} \in top(A1)\ \forall r_B \in top(B) \quad r_B \not< r_{A1}$

iii) if B has a strict subargument, all strict subarguments of B are justified arguments.

top(A) is the rule set, most closer to the consequence, except for the absolute rule.

Definition 2.2.1.

Let argument $A=F \cup D$. F is a set of absolute rules. D is a set of defeasible rules. A is an argument for p. H is a set of rule heads in top(A).

if $F=\phi$ top(A) is a set of a rule, of which the head is p.

if $D=\phi$ top(A)$=\phi$.

if $F \neq \phi$ and $D \neq \phi$ top(A) is a minimal such that top(A)$\subset D$, and $p \in E(F \cup H)$.

2.3 Narrowing of Justified Argument

In Definition 2.1.7. a justified argument may have an attack, which is not defeated by oneself, but defeated by any other argument. This is not suitable for the confrontation between legal theories. We consider the above example again. There is a nother theory, the concrete interpretation theory, as for the mistake at the abstract level. In the concrete interpretation theory, the intended crime is treated as an attempt.

```
concrete_interpretation_theory:
attempt(action(A1, X, W1), crime_action(CA, X, G))
  ← action(A1, X, W1),
      person(X),
      phenomenon(P1, Y),
      causality(action(A1, X, W1), phenomenon(P1, Y)),
      recognition(X, action(A1, X, W1),
                         action(A2, X, W2)),
      foresight(X, causality(action(A2, X, W2),
                         phenomenon(P2, Z)),
      crime_result(phenomenon(P1, Y), C1),
      crime_result(phenomenon(P2, Z), C2),
      different(Y, Z),
      different(C1, C2),
      ¬recognition(X, causality(action(A2, X, W2),
                         phenomenon(P1, Y)),
      intention(action(A1, X, W1),
                         crime_action(CA, X, G)).
```

Let arg_c_i_t be an argument for
attempt(action(shot_gun, john, marys_dog),
crime_action(killing, john, marys_dog)),
and consists of concrete_interpretation_theory, f1, f4, f6, f7, f8, f9, f10, f13, a3, and a4. top(arg_c_i_t) is a concrete_interpretation_theory.

In superior legal theories, priorities are taken as
{abstract_interpretation_theory <
substantial_interpretation_theory,
concrete_interpretation_theory <
substantial_interpretation_theory}.

We will consider the arg_c_i_t, arg_s_i_t and arg_a_i_t⁺.

Depending on Definition 2.1.7, arg_s_i_t is defeated by arg_a_i_t⁺. Therefore arg_c_i_t becomes a justified argument. We want to arg_s_i_t alone to be a justified argument, and arg_c_i_t to be a merely plausible argument.

We revise the definition of justified argument.

Definition 2.1.7

(2)' Justified argument is an argument which defeats all attacks that have no defeated strict subargument.

2.4 Negation As Failure

In [Sartor 1991], it is claimed that two kinds of negation(classical negation and Negation As Failure)are necessary for representing norms for exception, presumed facts, division of the onus of proof and incomplete information. We approve of necessity of NAF, in addition to defeasible reasoning. But from our standpoint, norms for exception should not be represented by NAF(also defeasible reasoning). Because in criminal jurisprudence, each statute is already interpreted consistently with the other statutes. Take article 199 and article35, for example.

article199 " A person, who kills a nother, shall be punished with death or penal servitude for life or not less than three years."

article35 " No person shall be punished for an act done under law or in the course of legitimate business."

If these are described faithfully in words, in the manner of [Sartor 1991], [Kowalski and Sadri 1990], the following will result.

```
article199 :
punishment(action(A, X, W), homicide)
  ← accomplishment(action(A, X, W), crime_action(killing, X, Y)),
     person(X),
     person(Y),
     not ¬ punishment(action(A, X, W), homicide)
     different(X, Y).

article35:
¬ punishment(action(A, X, W), C)
  ←under_law(action(A, X, W)).

article35:
¬ punishment(action(A, X, W), C)
  ← legitimate_business(action(A, X, W)).
```

If these are described faithfully to criminal jurisprudence, the following will result (the meaning of each rule is explained in Section3).

```
punishment:
punishment(action(A, X, W), Y)
  ↔ constitute_crime(action(A, X, W), Y),
      ¬ compound crimes(action(A, X, W)),
      ¬ derivative crimes(action(A, X, W)).

crime_constituent_theory:
constitute_crime(action(A, X, W), C, Y)
  ↔ satisfy_constituent_condition(action(A, X, W), C, Y),
      not justifiable_cause(action(A, X, W)),
      culpability(action(A, X, W), C, Y).

article199 :
satisfy_constituent_condition(action(A, X, W), homicide, Y)
  ↔ accomplishment(action(A, X, W),
     crime_action(killing, X, Y)),
     person(X),
     person(Y),
     different(X, Y).

article35:
justifiable_cause(action(A, X, W))
  ←under_law(action(A, X, W)).

article35:
justifiable_cause(action(A, X, W))
  ← legitimate_business(action(A, X, W)).
```

(*not*, used in crime_constituent_theory, correspond to division of the onus of proof in [Sartor 1991])

In order to represent controversy among legal theories and precedents, we must distinguish rules for implication from rules for equivalence . Most rules from precedents are the former, and most of statutes are the latter. Legal theory includes both types. For example, such that the substantial interpretation theory and the abstract interpretation theory means implication. Because in other cases of mistake at the abstract level, accomplishment or attempt must be defined by the other rules.

Take the theory of conditional causality and the theory of rational causality, which are concerned with causality, for examples of rules meaning equivalence. The theory of conditional causality says that if and only ifconditional causality between an action and an phenomenon is approved , causality is approved. On the other hand, the theory of rational causality says that if and only if rational causality between an action and a phenomenon is approved , causality is approved. We describe these rules as below.

theory_of_conditional_causality:
causality(action(A, X, W), phenomenon(P1, Y))
↔ conditional_causality(action(A, X, W),
phenomenon(P1, Y)).

theory_of_rational_causality:
causality(action(A, X, W), phenomenon(P1, Y))
↔ rational_causality(action(A, X, W),
phenomenon(P1, Y)).

"↔" means , "if and only if". We regard

$n{:}p(\overline{X}) \leftarrow p1, \ldots, pn.$

as syntax sugar of

$n{:}\ p(\overline{X}) \leftarrow p'(\overline{X}),$

$n{:}\ \neg p(\overline{X}) \leftarrow \mathit{not}\ p'(\overline{X})$ and

$n : p'(\overline{X}) \leftarrow p1, \ldots, pn.$

$\overline{X}$ is strings of variables. p' is a new predicate symbol,which does not appear in rules.

For example, if conditional causality is approved but rational causality is not, between " john strike bob" and " bob is died",

causality(action(strike, john, bob), phenomenon(death, bob))
is concluded by the theory of conditional causality , and

¬ causality(action(strike, john, bob),
phenomenon(death, bob))

is concluded by the theory of rational causality . We want to establish the defeat relation between these two arguments, depending on the priority between the theory of conditional causality and the theory of rational causality.

We will introduce NAF into the framework of defeasible reasoning. We extend rule form in Δ as below. (We don't consider NAF in absolute rules)

$$n{:}p0 \leftarrow p1, \ldots, pm,\ \mathit{not}\ pm{+}1, \ldots, \mathit{not}\ pn.(m<n)$$

E(*A*)is defined as follows.

Definition 2.4.1.

Let *A'* be a rule set obtained from *A* by deleting *not p* in the bodies of the rules in *A*.

Let *Lit* be the set of ground literals in the language of *A*. E(*A*), the argument answer set of *A*, is the smallest set such that

(1) for any rule $n{:}p0 \leftarrow p1, \ldots, pn.$ from *A'*, if $p1, \ldots, pn \in E(A)$, then $p0 \in E(A)$

(2) if E(*A*) contains a pair of complementary literals, then E(*A*)=*Lit*

(3) if *not p* has been deleted from Π, and E(*A*) contains literal *p*, then E(*A*)=*Lit*.

The relation between this argument answer set and answer set in [Gelfond andLifschitz 1990] is as follows.

If (i) *A* has no answer set, (ii) *A* has consistent answer sets , or (iii) *A* has an inconsistent answer set Lit, *A* has inconsistent argument answer set *Lit*. If (iv) *A*has a consistent answer set, *A* has the same argument answer set. For example, *p* and *q* are literals, $\Pi 1=\{p \leftarrow \mathit{notp}\}$, $\Pi 2=\{p \leftarrow \mathit{notq}, q \leftarrow \mathit{notp}\}$. Π 1has no answer set and Π 2 has two answer sets. Both's argument answer set is Lit (each one is inconsistent as argument.)

Definition of argument is not changed.

A argument for *p* in Π is still a consistent minimal rule set such that deriable for *p*, but may include "*not q*" in rule body. If so, that argument can't derive *q* because the answer set of that argument doesn't include *q*. But there may be other arguments derivable for *q*. We revise Definition 2.1.3.

Definition 2.1.3'

An argument *B* is a counter argument to an argument *A* iff *B* is an argument for *q* and *A* includes a subargument *A1* for $\overline{q}$ (where $\overline{q}$ denotes the complement of *q*) or includes rule that has *not*q in the body.

We divide all ground literals into 4 groups.

Justified consequence : there is a justified argument in arguments for *p*.

Defeated consequence : all arguments for *p* are defeated

Failed consequence : there is no argument for *p*

Merely plausible consequence : *p* is not included in above groups.

We revise Definition 2.1.6' ii), Definition 2.1.7(1), and Definition 2.1.7(2)'.

Definition 2.1.6".

argument *B* directly defeats an argument *A1*

iff i) argument *B* is an attack of argument *A1*,

ii) $\forall r_{A1} \in top(A1)\ \exists r_B \in top(B) \quad r_{A1} < r_B$,
and $\forall r_{A1} \in top(A1)\ \forall r_B \in top(B) \quad r_B \not< r_{A1}$

iii) if *B* has a strict subargument, all strict subarguments of *B* are justified arguments. If *B* has *not q* in the rule body, q is a failed consequence or defeated consequence.

Definition 2.1.7

(1)' Defeated argument is an argument which is defeated by any counter argument, or if has *not q* in the rule body, *q* is a failed justified consequence.

(2)" Justified argument is an argument which defeats all attacks that have no defeated strict subargument, and if *B* has *not q* in the rule body, *q* is a failed consequence or defeated consequence.

The definitions ofjustified argument and defeated argument- are mutually recursuve, and has no terminal condition. Depending on Definition 2.1.7(3), an argument, which is not able to be determined justified argument or defeated argument, is a merely plausible argument.

ex) If $\Pi=\{p \leftarrow notS,\ q \leftarrow notp\}$, $\{p \leftarrow notS\}$ is a merely plausible argument.

In the above example concerning to causality, we consider the following arguments.

```
arg_t_r_c =
  { theory_of_rational_causality:
      ¬ causality(action(strike, john, bob), phenomenon(death, bob))
        ←not causality'(action(strike, john, bob),
                                        phenomenon(death, bob)).}

arg_t_c_c=
  { theory_of_conditional_causality:
      causality(action(strike, john, bob), phenomenon(death, bob))
        ←causality''(action(strike, john, bob), phenomenon(death, bob)).
    theory_of_conditional_causality:
      causality''(action(strike, john, bob), phenomenon(death, bob))
        ←conditional_causality(action(strike, john, bob),
                                  phenomenon(death, bob))..
    fact:conditional_causality(action(strike, john, bob),
                                        phenomenon(death, bob))}

arg_t_c_c' =
  { theory_of_conditional_causality:
      causality(action(strike, john, bob), phenomenon(death, bob))
        ←not causality''(action(strike, john, bob),
                                        phenomenon(death, bob)).}
```

arg_t_c_c' is a defeated argument, because causality''(action(strike, john, bob), phenomenon(death, bob)) is a justified consequence. If priority is

theory_of_conditional_causality < theory_of_rational_causality,

arg_t_r_c defeats arg_t_c_c, and is a justified argument.

3 Overview of Knowledge Base

3.1 Statute

In Japanese penal code, crime-constituening conditions are provided from article 77 to 264. Crimes are classified into preparations, conspiracies, attempts, and accomplishments, from a viewpoint of achievement, and classified into intention and negligence from a viewpoint of mentality of criminal. In the present Japanese criminal law, crime by negligence is applied only to crime of accomplishment. For that reason, we divide crime-constituening conditions into *accomplishment*(accomplishment and intent), *attempt*(attempt and intent), *preparation*(and intent), *conspiracy*(conspiracies and intent), *negligence*(accomplishment and negligence).

On this level, we describe statute literally. Therefore rules on this level are not inconsistent except conflicting law.

For example, article 199 which is the statute for homicide is described as below.
(Contents of sentences are not described since it is out of legal argumentation to judge contents of sentences.)

```
article199 :
satisfy_constituent_condition(action(A, X, W), homicide, Y)
 ↔ accomplishment(action(A, X, W), crime_action(killing, X, Y)),
    person(X),
    person(Y),
    different(X, Y).
```

The other statutes about homicide are decsribed as below.

```
article203 :
satisfy_constituent_condition(action(A, X, W), attempt_homicide, Y)
 ↔ attempt(action(A, X, W), crime_action(killing, X, Y)),
    person(X),
    person(Y),
    different(X, Y).

article201 :
satisfy_constituent_condition(action(A, X, W),
                                        preparation_homicide, Y)
 ↔ preparation(action(A, X, W), crime_action(killing, X, Y)),
    person(X),
    person(Y),
    different(X, Y).

article210 :
satisfy_constituent_condition(action(A, X, W),
                                        negligence_homicide, Y)
 ↔ negligence(action(A, X, W), crime_action(killing, X, Y)),
    person(X),
    person(Y),
    different(X, Y).
```

We use defeasible reasoning to solve confrontation of conflicting law. Conflicting law is that though an action seemed to satisfy some crime-constituting conditions, in fact only one satisfactory is accepted and the others are not.

For example, To desert his ascendant seemed to satisfy both crime-constituting conditions of simple desertion and desertion of ascendant, in fact simple desertion is accepted and desertion of ascendant is not. Rules like that are called as Lex Specialis and this is described as below.

```
Lex Specialis1:
¬ satisfy_constituent_condition(action(A, X, W),
                                    desertion_of_ascendant, Y)
  ← satisfy_constituent_condition(action(A, X, W),
                                            simple_desertion, Y).

Lex Specialis2:
¬satisfy_constituent_condition(action(A, X, W),
                                            simple_desertion, Y)
  ← satisfy_constituent_condition(action(A, X, W),
                                    desertion_of_ascendant, Y).

article217 < article218_2
```

3.2 Interpretation of Statute

Constructing rules , of which head-predicate is accomplishment, attempt, preparation, conspiracy, or negligence corresponds to interpretting the statute.

Crimes are classified into directness, indirectness, omission, and causal liberty from a viewpoint of how to attain. That is to say, a statute providing a crime has rules of above 4 groups. Interpretation of article199, in case of directness, is described as below.

```
article199_directness_interpretation :
accomplishment(action(A, X, W), crime_action(killing, X, Y))
   ← action(A, X, W),
      intention(action(A, X, W), crime_action(killing, X, Y)),
      phenomenon(death, Y),
      causality(action(A, X, W), phenomenon(death, Y)).
```

Crimes are classified into result and behavior from a viewpoint of the necessity of certain result. Crime of homicide is a crime of result, since the death of some person ie needed for the crime-constituting . On this level, depending on the classification of crimes like that, rules are able to be described regularily, and the confrontations of legal theories are not so much. But as for legal concepts , which appeared in rule-body, such that intention, causality, and duty of action, the confrontations are much severe.

3.3 Constitution of Crime and Punishment

Even if action satisfies an crime-constituting condition, that doesn't immediately constitute the crime. There are many opposing opinions about constituting crimes. A commonly accepted theory is the crime constituent theory. This teory is that if and only if action satisfies crime-constituting condition, doesn't correspond to justifiable cause such that self defence and legitimate business, and the agent of the action has culpability, the action constitutes the crime. This rule is discribed as follows.

```
crime_constituent_theory:
constitute_crime(action(A, X, W), C, Y)
  ↔ satisfy_constituent_condition(action(A, X, W), C, Y),
     not justifiable_cause(action(A, X, W)),
     culpability(action(A, X, W), C, Y).
```

Even if one or more actions of one person constitute several crimes, he may be punished as one crime. This case is such that compound crimes or derivative crimes. Compound crimes is that an action constitutes several crimes. Derivative crimes is that an action, which is a means or a sequel of another action constituting a crime, constitutes another crime. The rule defining punishment is discribed as follows.

```
punishment:
punishment(action(A, X, W), Y)
  ↔ constitute_crime(action(A, X, W), Y),
     ¬ compound crimes(action(A, X, W)),
     ¬ derivative crimes(action(A, X, W)).
```

```
article54_1_former:
punishment(action(A, X, W), Y)
  ← compound crimes(action(A, X, W), Y).

article54_1_latter:
punishment(action(A, X, W), Y)
  ← derivative crimes(action(A, X, W), Y).
```

3.4 Defeasible Reasoning

We use defeasible reasoning to solve confrontation of legal theories or precedents and conflicting law. But other usages are mentioned in [Sartor 1993].

For example, article 199 and article 35 are described as below (c.f. 2.4) .

```
article199 :
punishment(action(A, X, W), homicide)
  ← accomplishment(action(A, X, W), crime_action(killing, X, Y)),
     person(X),
     person(Y),
     not ¬ punishment(action(A, X, W), homicide)
     different(X, Y).

article35:
¬ punishment(action(A, X, W), C)
  ←under_law(action(A, X, W)).

article35:
¬ punishment(action(A, X, W), C)
  ← legitimate_business(action(A, X, W)).

article199 < article35.
```

The reason of this difference is the same as that of difference in usage of NAF, mentioned in 2.4.

4 Conclusion

We are now in the progress of describing data and designing language for Japanese criminal jurisprudence. In our experience with data descriptions, defeasible reasoning seems to be a useful function. We expand a defeasible reasoning model by introducing absolute rule and Negation As Failure and narrowed the definition of justified argument. We use defeasible reasoning to solve confrontation between legal theories or precedents and conflicting laws.

References

[Gelfond and Lifschitz 1990] M. Gelfond and V. Lifschitz. Logic Programs with Classical Negation. In *Proc. Seventh Int. Logic Programming Conference*. 1990. pp. 579-597

[Kowalski and Sadri 1990] R. Kowalski and F. Sadri. Logic

Programs with Exceptions. In *Proc. Seventh Int. Logic Programming Conference.* 1990. 598-913

[Nitta et al. 1994] K. Nitta , M. Shibasaki, T. Sakata, T. Yamaji,W. Xianchange, H. Ohsaki, S. Tojo, I. Kokubo, and T. Suzuki. A Legal Reasoning System :new HELIC-II. In Proc. Int. Conf. on FGCS, ICOT,1994.

[Prakken 1993] H. Prakken. A logical framework for modelling legal argument. In *Proc. Int. Conf. on AI and Law*, 1993. pp. 1-9.

[Sartor 1991] G. Sartor. The Structure of Norm Conditions and Nonmonotonic Reasoning in Law. In *Proc. Int. Conf. on AI and Law*, 1991. pp. 155-164.

[Sartor 1993] G. Sartor. A Simple ComputationalModel for Nomonotonic and Adversarial Legal Reasoning. In *Proc. Int. Conf. on AI and Law*, 1993. pp. 192-201.