Defeasible Reasoning Between Conflicting Agents Based on VALPSN

Kazumi Nakamatsu
nakamatu@hept.himeji-tech.ac.jp
School of H.E.P.T.
Himeji Institute of Technology
Shinzaike 1-1-12, HIMEJI
670-0092 JAPAN

Jair Minoro Abe
jmabe@lsi.usp.br
Dept. Informatics ICET
Paulista University
Rua Dr.Bacelar 121204026-002
Sao Paulo BRAZIL

Atsuyuki Suzuki
suzuki@cs.inf.shizuoka.ac.jp
Faculty of Information
Shizuoka University
Johoku 3-5-1, HAMAMATSU
432-8011 JAPAN

Abstract

We propose a method to model agents' schedules aiming to resolve conflicts between agents based on the defeasible reasoning by VALPSN(Vector Annotated Logic Program with Strong Negation). In the method, first, agents' schedules are defined by Billington's defeasible theories. The defeasible theories are translated into VALPSNs as the next step and the VALPSNs' stable models are computed as the last step. We describe how to translate the Billington's defeasible theories into VALPSNs and how to proceed the negotiation to resolve conflicts between a meeting coordinator and agents taking an appointment schedule problem as an example.

Introduction and Motivation

Annotated logics are a family of paraconsistent logics, which are appropriate for dealing with inconsistency or conflicts (Da Costa *et al.* 1989). They were studied, from the viewpoint of logic programming, by Blair and Subrahmanian (Blair and Subrahmanian 1989). Kifer and Subrahmanian have proposed generalized annotated logic programming and its applications (Kifer and Subrahmanian 1992). Moreover, Subrahmanian has applied modified annotated logic programming to integrating knowledge from multiple sources (Subrahmanian 1994). We have also proposed annotated logic programs called ALPSN(Annotated Logic Program with Strong Negation), which have a strong negation, and shown that they provide annotated semantics for some nonmonotonic reasonings (Nakamatsu and Suzuki 1994) (Nakamatsu and Suzuki 1998). In this paper, we introduce a new version of ALPSN called VALPSN(Vector Annotated Logic Program with Strong Negation) which can deal with defeasible reasoning and has a stable model semantics (Nakamatsu and Abe 1999). Defeasible logics are well-known formalizations of defeasible reasoning, however, some of them do not have appropriate semantics. In (Nakamatsu and Abe 1999), it has been shown that a Billington's defeasible logic can be translated into VALPSN to show that VALPSN can be a semantics for the defeasible logic and deal with defeasible reasoning.

In this paper, we propose a conflict resolving method between agents by the defeasible reasoning based on VALPSN and its stable models. That is to say, it is shown that agent models are defined by VALPSNs and the semantics of the VALPSNs is given by the stable models. We take conflict resolving between agents in an appointment scheduling problem as an example. There is a coordinator who arranges each agent's schedule to resolve conflicts in the example. The following four steps are iteratively performed in order to resolve conflicts :

Step 1 each agent's schedule is defined by a defeasible theory,

Step 2 the defeasible theory in **Step 1** is translated into a VALPSN,

Step 3 the stable model for the VALPSN in **Step 2** is computed,

Step 4 if a conflict is not detected in the stable model in **Step 3**, then, END,
otherwise, update each agent's schedule and back to **Step 1**.

This paper is organized as follows : first, we introduce VALPSN and its stable model semantics, next, we introduce a Billington's defeasible logic and show the translation from the defeasible theories into VALPSNs, last, we describe how to resolve conflicts between agents taking an appointment schedule problem as an example.

Vector Annotated Logic Program with Strong Negation

We have defined formally ALPSN and its stable model semantics in (Nakamatsu and Suzuki 1994). We describe only the necessary part of VALPSN and its stable model semantics, as it is easy to define fully them by modifying the definition of ALPSN and its stable model semantics. Generally, a truth value called an annotation is explicitly attached to each atomic formula of annotated logics. For example, let p be an atomic formula, μ an annotation, then, $p:\mu$ is called an annotated atomic formula. A partially ordered relation is defined on the set of annotations and the set constitutes a complete lattice structure. The annotations in vector annotated logics are 2 dimensional vectors such that their components are nonnegative integers. Throughout this paper we assume that the complete lattice of vector annotations

$$\mathcal{T}_v = \{(x,y)|\ 0 \le x \le m, 0 \le y \le m, \\ x, y \text{ and } m \text{ are integers }\}.$$

The ordering of this lattice is denoted in the usual fashion by a symbol $\preceq$. Let $\vec{v_1} = (x_1, y_1)$ and $\vec{v_2} = (x_2, y_2)$. Then,

$$\vec{v_1} \preceq \vec{v_2} \quad \textit{iff} \quad x_1 \le x_2 \text{ and } y_1 \le y_2.$$

In a vector annotated literal $p:(i,j)$, the first component i of the vector annotation indicates the degree of positive information to support the literal p and the second one j indicates the degree of negative information. For example, a vector annotated literal $p:(3,2)$ can be informally interpreted that p is known to be true of strength 3 and false of strength 2. Annotated logics have two kinds of negations, an epistemic negation($\neg$) and an ontological negation($\sim$). The epistemic negation is a mapping between annotations and the ontological negation is a strong negation that appears in classical logics. The epistemic negation of vector annotated logics is defined as the exchange between the components of vector annotations,

$$\neg p:(i,j) = p:\neg(i,j) = p:(j,i).$$

Definition 1 (VALPSN)
If $L_0, \cdots, L_n$ are vector annotated literals, then,

$$L_1 \wedge \cdots \wedge L_i \wedge \sim L_{i+1} \wedge \cdots \wedge \sim L_n \rightarrow L_0$$

is called a *vector annotated logic program clause with strong negation*(VALPSN clause). A VALPSN is a finite set of VALPSN clauses.

In the rest of this paper, we assume that a VALPSN P is a set of ground clauses. We also assume that all interpretations of a VALPSN P have a Herbrand base B_P(the set of all variable-free atoms) under consideration as their domain of interpretation. A Herbrand interpretation can be considered to be a mapping $I : B_P \rightarrow \mathcal{T}_v$. Usually, I is denoted by the set $\{p:\sqcup\vec{v_i}|I \models p:\vec{v_1} \wedge \cdots \wedge p:\vec{v_n}\}$, where $\sqcup\vec{v_i}$ is the least upper bound of $\{\vec{v_1}, \ldots, \vec{v_n}\}$. The ordering $\preceq$ on $\mathcal{T}_v$ is extended to interpretations in the natural way. Let I_1 and I_2 be any interpretations, and A be an atomic formula.

$$I_1 \preceq_I I_2 =_{def} (\forall A \in B_P)(I_1(A) \preceq I_2(A)).$$

In order to provide the stable model semantics for VALPSN, we define a function T_P from a Herbrand interpretation to a Herbrand interpretation associated with every VALPSN P over $\mathcal{T}_v$.

$$T_P(I)(A) = \sqcup\{\vec{v}|B_1 \wedge \cdots \wedge B_m \wedge \sim C_1 \wedge \cdots \wedge \sim C_n \rightarrow \\ A:\vec{v} \text{ is a ground instance of a VALPSN} \\ \text{clause in } P \text{ and } I \models B_1 \wedge \cdots \wedge B_m \wedge \\ \sim C_1 \wedge \cdots \wedge \sim C_n\},$$

where the notation $\sqcup$ denotes the least upper bound. We define a special interpretation Δ to be an interpretation that assigns the truth value $(0,0)$ to all members of B_P. Then, the upward iteration $T_P \uparrow \lambda$ of the operator T_P is defined as :

$$T_P \uparrow 0 = \Delta$$
$$T_P \uparrow \lambda = \sqcup_{\alpha<\lambda} T_P(T_P \uparrow \alpha) \quad \text{for any ordinals } \alpha, \lambda.$$

Then, the following well-known results in terms of a VALP(Vector Annotated Logic Program with no strong negation) P and the operator T_P hold (Blair and Subrahmanian 1989).

Proposition 1

- P has a least model that is identical to the least fixed point of T_P.
- $T_P \uparrow \omega$ is identical to the least fixed point of T_P.

We extend the stable model semantics that was proposed in (Gelfond and Lifschitz 1989) for ordinary logic programs with strong negation to VALPSNs. First, we describe the Gelfond-Lifschitz transformation for VALPSNs.
Let I be any interpretation and P a VALPSN. P^I, the *Gelfond-Lifschitz transformation* of the VALPSN P with respect to I, is a VALP obtained from P by deleting
1) each clause that has a strongly negated vector annotated literal $\sim(C:\vec{v})$ in its body with $I \models (C:\vec{v})$, and
2) all strongly negated vector annotated literals in the bodies of the remaining VALPSN clauses.

Since P^I contains no strong negation, due to **Proposition 1**, it has the unique least model that is given by $T_{P^I} \uparrow \omega$

Definition 2(Stable Model for VALPSN)
If I is a Herbrand interpretation of a VALPSN P, I is called a *stable model* of P *iff* $I = T_{P^I} \uparrow \omega$.

Example 1
Let p, q, and r be literals,

$$\mathcal{T}_v = \{(i,j) | 0 \le i \le 3, 0 \le j \le 3\},$$

and a VALPSN

$$\begin{aligned} P = \{ \quad & q:(3,0), \quad && q:(2,0) \wedge \sim p:(0,3) \rightarrow p:(2,0), \\ & r:(3,0), && r:(2,0) \wedge \sim p:(3,0) \rightarrow p:(0,2)\}. \end{aligned}$$

If an interpretation

$$I = \{q:(3,0),\ r:(3,0),\ p:(2,2)\},$$

then,

$$\begin{aligned} P^I = \{ \quad & q:(3,0), \quad && q:(2,0) \rightarrow p:(2,0), \\ & r:(3,0), && r:(2,0) \rightarrow p:(0,2) \ \} \end{aligned}$$

and $T_{P^I} \uparrow \omega = I$. Therefore, I is a stable model of P. The VALPSN P has only one stable model.

Defeasible Logic and VALPSN

The alphabet of the defeasible logic(Billington 1997) is the union of the following four pairwise disjoint sets of symbols.

- A nonempty countable set of proposition symbols.
- The set $\{\neg, \rightarrow, \Rightarrow, \leadsto\}$ of connectives.
- The set $\{+, -, \Delta, \partial\}$ of positive, negative, definite, and defeasible proof symbols.
- The set of punctuation marks consisting of commas, braces and parentheses.

The negation of a proposition p is denoted by $\neg p$. The complement of the proposition p is $\neg p$ and the complement of $\neg p$ is p. If q is any literal, then, the complement of the q is denoted by $\bar{q}$. The positive proof symbol, $+$, indicates that the following literal has been proved. The negative proof symbol, $-$, indicates that the following literal has been proved to be unprovable. The definite proof symbol Δ indicates that the proof of the following literal cannot be defeated by more information. The defeasible proof symbol ∂ indicates that the proof of the following literal can be defeated by more information. A *rule* has three parts : a finite set of literals on the left, an arrow in the middle, and a literal on the right. A rule that contains the *strict arrow* $\rightarrow$, for example $\mathbf{A} \rightarrow q$, is called a *strict rule*. The intuition is that whenever all the literals in $\mathbf{A}$ are accepted then q must be accepted. A rule that contains the *defeasible arrow* $\Rightarrow$, for example, $\mathbf{A} \Rightarrow q$ is called a *defeasible rule*. If all the literals in $\mathbf{A}$ are accepted then q is accepted provided that there is an insufficient evidence against q. A rule that contains the *defeater arrow* $\leadsto$, for example, $\mathbf{A} \leadsto \bar{q}$ is called a *defeating rule* or a *defeater*. If all the literals in $\mathbf{A}$ are accepted then $\mathbf{A} \rightarrow \bar{q}$ is an evidence against q, but not for $\bar{q}$. All sets of conflicting literals are collected into a single set denoted by $\mathcal{C}$. In this paper, we assume the conflict set $\mathcal{C}$ consists of the complementary pairs $\{q, \bar{q}\}$ of literals for simplicity.

Definition 3 (Defeasible Theory)
A *defeasible theory* over $\mathcal{C}$ is a quadraple $(F, \mathcal{C}, R, >)$ such that F is a set of literals(facts), $\mathcal{C}$ is a conflict set, R is a set of rules, and $>$ is a superiority relation on R.

The defeasible logic has the four inference conditions, $+\Delta$, $-\Delta$, $+\partial$, and $-\partial$. We comment on the notations that appear in the conditions. Let q be a literal. In a proof, $+\Delta q$ indicates that q is proved definitely, $-\Delta q$ indicates that it is proved that q cannot be proved definitely, $+\partial q$ indicates that q is proved defeasibly, and $-\partial q$ indicates that it is proved that q cannot be proved defeasibly. Let R be any set of rules. The set of strict rules in R is denoted by R_s, and the union of R_s and the set of defeasible rules in R by R_{sd}. The antecedent of any rule r is denoted by $A(r)$ and its consequent is denoted by $C(r)$. $R[q] =_{def} \{r | r \in R \text{ and } q = C(r)\}$. The superiority relation on R is any symmetric binary relation $>$ on R. A finite sequence $P = (P(1), \cdots, P(|P|))$ of tagged literals$(+\Delta q, -\Delta q, +\partial q, -\partial q)$ is called a *proof*. An element of a proof is called a line of the proof. $P(i+1)$ indicates the $i+1$th line of a proof. $P(1..i)$ indicates the proof lines from the first one to the ith one. The four conditions of inference in the defeasible logic are :

$+\Delta$) If $P(i+1) = +\Delta q$, for some literal q, then either
  .1) $q \in F$; or
  .2) $\exists r \in R_s[q], \forall a \in A(r), +\Delta a \in P(1..i)$.

$-\Delta$) If $P(i+1) = -\Delta q$, for some literal q, then
  .1) $q \notin F$, and
  .2) $\forall r \in R_s[q], \exists a \in A(r), -\Delta a \in P(1..i)$.

$+\partial$) If $P(i+1) = +\partial q$, for some literal q, then either
  .1) $+\Delta q \in P(1..i)$; or

.2) All three of the following conditions hold.
  .1) $\exists r \in R_{sd}[q], \forall a \in A(r), +\partial a \in P(1..i)$,
  .2) $-\Delta \bar{q} \in P(1..i)$, and
  .3) $\forall s \in R[\bar{q}]$ either
    .1) $\exists a \in A(s)$, $-\partial a \in P(1..i)$; or
    .2) $\exists t \in R_{sd}[q]$ such that
      .1) $\forall a \in A(t)$, $+\partial a \in P(1..i)$, and
      .2) $t > s$.

$-\partial$) If $P(i+1) = -\partial q$, for some literal q, then either
.1) $-\Delta q \in P(1..i)$; or
.2) either
  .1) $\forall r \in R_{sd}[q], \exists a \in A(r), -\partial a \in P(1..i)$,
  .2) $+\Delta \bar{q} \in P(1..i)$, or
  .3) $\exists s \in R[\bar{q}]$ such that
    .1) $\forall a \in A(s)$, $+\partial a \in P(1..i)$; and
    .2) $\forall t \in R_{sd}[q]$ either
      .1) $\exists a \in A(t)$, $-\partial a \in P(1..i)$, or
      .2) not $(t > s)$.

Example 2(Genetically Altered Penguin)
This example is taken from (Billington 1997). Those are known that penguins(p) are definitely birds(b), that defeasibly birds(b) fly(f), and that defeasibly penguins(p) do not fly($\neg f$). Suppose a penguin that has large wings and flight muscles. Such a genetically altered penguin(gap) might fly(f). Then, a defeasible theory T capturing this situation is as below.
Suppose that Opus(o) is a genetically altered penguin. Let $T = (F, \mathcal{C}, R, >)$ be the defeasible theory such that $F = \{\mathbf{F1}\}$, $\mathcal{C} = \{f(o), \neg f(o)\}$, $R = \{\mathbf{R1}, \mathbf{R2}, \mathbf{R3}, \mathbf{R4}\}$, and $>$ is defined by **R5** $>$ **R4** and **R4** $>$ **R3**.

F1 ;	$gap(o)$,	**R1** ;	$gap(o) \rightarrow p(o)$,
R2 ;	$p(o) \rightarrow b(o)$,	**R3** ;	$b(o) \Rightarrow f(o)$,
R4 ;	$p(o) \Rightarrow \neg f(o)$,	**R5** ;	$gap(o) \rightsquigarrow f(o)$.

We show intuitive derivations.
From **F1**,**R1**,and **R2**, we have $+\Delta gap(o)$, $+\Delta p(o)$, and $+\Delta b(o)$. Since **R4** $>$ **R3**, the consequent $f(o)$ of **R3** is defeated by the consequent $\neg f(o)$ of **R4**. Thus, we do not have $+\partial f(o)$ and we have $-\partial f(o)$. Since **R5** $>$ **R4**, the consequent $\neg f(o)$ of **R4** is defeated by the defeater **R5**. Thus, $+\partial \neg f(o)$ cannot be derived by **R4** and we have $-\partial \neg f(o)$.

Here, we provide a translation from defeasible theories into VALPSNs. Strict rules, defeasible rules, defeaters, and facts are translated into VALPSN clauses. We have the relation between the two kinds of provability of the defeasible logic and the satisfiability of the VALPSN stable models in terms of the translation as described in Figure 1.

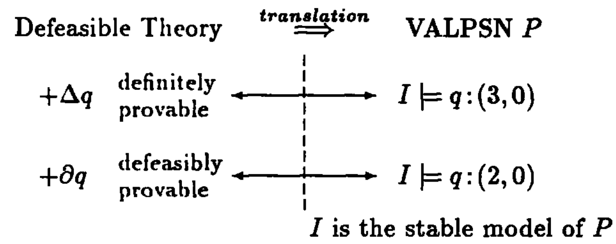

Figure 1: Defeasible Theory and VALPSN

The basic ideas of the translation are as follows. We want to represent the notions of "definite provability" and "defeasible provability" by vector annotations. Then, if we regard that "definite provability" represents stronger knowledge about provability than "defeasible provability", "a literal q is definitly provable $(+\Delta q)$" can be represented by a vector annotated literal $q:(3,0)$ and "a literal q is defeasibly provable $(+\partial q)$" can be represented by a weaker vector annotated literal $q:(2,0)$.

When we translate a defeasible theory into a VALPSN, a superiority relation between conflicting rules has to be also embedded into the VALPSN. We want to express the superiority relation by the vector annotations of consequent literals(heads) in VALPSN clauses. Let's take a simple example. Let

$$\mathbf{C1};\ p:(2,0) \rightarrow q:(1,0) \quad \text{and}$$
$$\mathbf{C2};\ r:(2,0) \rightarrow q:(0,2).$$

Roughly speaking, the strength of positive information on the literal q in the clause **C1** is regarded as 1, and in the clause **C2**, the strength of negative information on the literal q is as 2. Then, we recognize that **C2** is stronger than(superior to) **C1**.

We define the rank of a rule r, $rank(r)$, as an index of rule strength based on the superiority relation $>$ between conflicting rules.

Definition 4 ($rank(r)$)
Let $T = (F, \mathcal{C}, R, >)$ be a defeasible theory and $R = \{r_1, \ldots, r_k\}$.

- For each rule $r_i (1 \leq i \leq k)$, $rank(r_1) = 0$ or -1.
- For each superiority relation $r_i > r_j$, if there does not exist a rule r such that $r_i > r$ and $r > r_j$, $(1 \leq i, j \leq k)$, then, $rank(r_i) = 0$ and $rank(r_j) = -1 (1 \leq i, j \leq k)$.
- If a rule $r_i (1 \leq i \leq k)$ has no superiority relation to the other rules, then, $rank(r_i) = 0$.

Example 3
Suppose the same defeasible theory as **Example 2**. As the superiority relation on R is defined as **R5** $>$ **R4** and

R4 > **R3** , and there is no superiority relation between **R1**, **R2**, and **R5**,
$rank(\ \mathbf{R1}\ ) = rank(\ \mathbf{R2}\ ) = 0$,
since **R4**>**R3**, $rank(\ \mathbf{R3}\ ) = -1$ and $rank(\ \mathbf{R4}\ ) = 0$,
since **R5**>**R4**, $rank(\ \mathbf{R4}\ ) = -1$ and $rank(\ \mathbf{R5}\ ) = 0$.
$rank(r)$ is defined for each superiority relation as defined above. Therefore, the defeasible rule **R4** has two different *rank* values.

We describe the outline of the translation from facts and the three kinds of rules into VALPSN clauses based on the four inference conditions $\{+\Delta, -\Delta, +\partial, -\partial\}$ and show some examples.

[Fact]
A fact can be used to derive $+\Delta q$. If the condition $+\Delta.1$ is satisfied, there must be a literal $q \in F$ such that $+\Delta q$. Thus, the fact q is translated into a vector annotated literal $q\!:\!(3,0)$.

[Strict Rule]
A strict rule can be used to derive both $+\Delta q$ and $+\partial q$. Let $\mathbf{A} \rightarrow q$ be a strict rule, $\mathbf{A}= \{\ a_1, \ldots, a_k\ \}$ and each $a_j (1 \leq j \leq k)$ a literal. We consider the following two cases.
Case 1 If the condition $+\Delta.2$ is satisfied, there is a strict rule $\mathbf{A} \rightarrow q$ such that the antecedent $\mathbf{A}$ is definitely provable. Thus, the strict rule $\mathbf{A} \rightarrow q$ is translated into

$$a_1\!:\!(n,0) \wedge \cdots \wedge a_k\!:\!(n,0) \rightarrow q\!:\!(n,0).$$

Case 2 A strict rule can be also used to derive "defeasibly provable ($+\partial$)", if the antecedent of the strict rule is defeasibly provable. Thus, we consider the condition $+\partial.2$ as the derivation of $+\partial q$ due to the strict rule $\mathbf{A} \rightarrow q$. However, we have to consider the defeasible reasoning based on the superiority relations as well as the case of defeasible rules.

[Defeasible Rule]
Let $\mathbf{A} \Rightarrow q$ be a defeasible rule, $\mathbf{A}= \{\ a_1, \ldots, a_k\ \}$ and each $a_j (1 \leq j \leq k)$ is a literal. Since defeasible rules can be used to derive only $+\partial q$, we can translate the defeasible rule $\mathbf{A} \Rightarrow q$ into VALPSN clauses in the same way as the **Case 2** of **[Strict Rule]**.

[Defeater]
The consequents of defeaters cannot be derived by the conditions $+\Delta$ or $+\partial$. The role of defeaters is not to derive their consequents but just to defeat the derivations by the other rules. This effect is reflected in the translation of strict rules and defeasible rules, that is to say, it is reflected in **Case 2** of **[Strict Rule]**. Therefore, defeaters themselves are not translated directly into VALPSN clauses.

Although we consider only the complementary pairs of literals as conflicting literals for simplicity, it is not so difficult to modify the translation rule to allow other kinds of literals as the elements of conflict sets.

Example 4
Let a defeasible theory $T = (F, C, R, >)$, $F = \{a\}$, $C = \{q, \neg q\}$, $R = \{\mathbf{R1};\{a\} \rightarrow q,\ \mathbf{R2};\{b_1, b_2\} \Rightarrow \neg q,\ \mathbf{R3};\{c\} \rightsquigarrow \neg q\}$, and **R3** > **R1**. Then, $rank(\mathbf{R1}) = -1$, $rank(\mathbf{R2}) = 0$, $rank(\mathbf{R3}) = 0$.

The fact a is translated into $\quad a\!:\!(3,0)$.

The strict rule **R1** is translated into

$$a\!:\!(3,0) \rightarrow q\!:\!(3,0),$$
$$a\!:\!(2,0) \wedge \sim b_1\!:\!(2,0) \wedge \sim c\!:\!(2,0) \wedge \sim q\!:\!(0,3) \rightarrow q\!:\!(2,0),$$
$$a\!:\!(2,0) \wedge \sim b_2\!:\!(2,0) \wedge \sim c\!:\!(2,0) \wedge \sim q\!:\!(0,3) \rightarrow q\!:\!(2,0),$$
$$a\!:\!(2,0) \wedge \sim b_1\!:\!(2,0) \wedge c\!:\!(2,0) \wedge \sim q\!:\!(0,3) \rightarrow q\!:\!(1,0),$$
$$a\!:\!(2,0) \wedge \sim b_2\!:\!(2,0) \wedge c\!:\!(2,0) \wedge \sim q\!:\!(0,3) \rightarrow q\!:\!(1,0).$$

The defeasible rule **R2** is translated into

$$b_1\!:\!(2,0) \wedge b_2\!:\!(2,0) \wedge \sim a\!:\!(2,0) \wedge \sim q\!:\!(3,0) \rightarrow q\!:\!(0,2).$$

Since there is no superiority relation between **R2** and **R1**, we do not have to take into account the case in which the antecedent of **R1** is defeasibly provable in order to derive $+\partial\neg q$ by **R2**. The defeater **R3** is not translated into any VALPSN clause.

Example 5
We consider the same defeasible theory T as **Example 2.** Since **R5**>**R4** and **R4**>**R3**, $rank(\mathbf{R1}) = rank(\mathbf{R2}) = 0$, $rank(\mathbf{R3}) = -1$, $rank(\mathbf{R4}) = 0$ and -1, and $rank(\mathbf{R5}) = 0$.
Then, we obtain a VALPSN

$$\begin{aligned}
P = \{\ & gap(o)\!:\!(3,0), \quad gap(o)\!:\!(3,0) \rightarrow p(o)\!:\!(3,0),\\
& p(o)\!:\!(3,0) \rightarrow b(o)\!:\!(3,0),\\
& b(o)\!:\!(2,0) \wedge p(o)\!:\!(2,0) \wedge \sim f(o)\!:\!(0,3) \rightarrow f(o)\!:\!(1,0),\\
& b(o)\!:\!(2,0) \wedge \sim p(o)\!:\!(2,0) \wedge \sim f(o)\!:\!(0,3) \rightarrow f(o)\!:\!(2,0),\\
& p(o)\!:\!(2,0) \wedge b(o)\!:\!(2,0) \wedge \sim gap(o)\!:\!(2,0) \wedge \sim f(o)\!:\!(3,0)\\
& \rightarrow f(o)\!:\!(0,2),\\
& p(o)\!:\!(2,0) \wedge gap(o)\!:\!(2,0) \wedge \sim b(o)\!:\!(2,0) \wedge \sim f(o)\!:\!(3,0)\\
& \rightarrow f(o)\!:\!(0,1),\\
& p(o)\!:\!(2,0) \wedge \sim b(o)\!:\!(2,0) \wedge \sim gap(o)\!:\!(2,0) \wedge\\
& \qquad \sim f(o)\!:\!(3,0) \rightarrow f(o)\!:\!(0,2)\}
\end{aligned}$$

Let an interpretation

$$I = \{ \quad gap(o):(3,0), \qquad p(o):(3,0), \\ b(o):(3,0), \qquad f(o):(1,0)\}.$$

Then, its Gelfond-Lifschitz transformation

$$P^I = \{ \ gap(o):(3,0), \quad gap(o):(3,0) \rightarrow p(o):(3,0), \\ p(o):(3,0) \rightarrow b(o):(3,0), \\ b(o):(3,0) \wedge p(o):(3,0) \rightarrow f(o):(1,0) \ \}.$$

Since the minimal model of P^I is identified with the interpretation I, it is one of the stable models of the VALPSN P, and since $I \models gap(o):(3,0)$, $I \models p(o):(3,0)$, and $I \models b(o):(3,0)$, we have $T \vdash +\Delta gap(o)$, $T \vdash +\Delta p(o)$, and $T \vdash +\Delta b(o)$. However, $I \not\models f(o):(3,0)$, $I \not\models \neg f(o):(3,0)$, $I \not\models f(o):(2,0)$, and $I \not\models \neg f(o):(2,0)$. Therefore, we have $T \vdash -\Delta f(o)$, $T \vdash -\Delta \neg f(o)$, $T \vdash -\partial f(o)$, and $T \vdash -\partial \neg f(o)$ as well as shown in **Example 2**.

Conflict Resolving

Generally, the conflict set in the defeasible logic is defined as a pair of any literals. For example, suppose (p, q) is a pair of conflicting literals, there maybe the superiority relation between defeasible rules, $\mathbf{A} \Rightarrow p$ and $\mathbf{B} \Rightarrow q$. In such a case, a more complicated translation from the defeasible theories into VALPSNs is required. However, we have assumed the conflict set is the complementary pair of literals in the previous section. In this paper, we assume that conflicts are defined as conflict sets (the complementary pairs of literals) in defeasible theories.

We consider the following scenario as an example of conflict resolving between agents in an appointment schedule problem.

There is the set $A = \{a_1, a_2, a_3\}$ of three agents who are invitees for meetings. First, the cordinator of the meetings suggests the dates $S = \{d_\alpha, d_\beta\}$ of the meetings $M = \{m_1, m_2\}$, where $S \subseteq D = \{d_1, d_2, \cdots, d_n\}$. Each agent checks his schedule whether he can participate the meetings on the dates or not and returns the result to the coordinator. According to the result, the coordinator convinces some of the agents(invitees) to reschedule. This process is continued untill conflicts are resolved as shown in **Step 1** - **Step 4** (Introduction and Motivation).

We use the following predicate schemata :

$avail(d, a)$; an agent a is free on a date d, where $a \in A$ and $d \in D$,

$part(m, a)$; an agent a participates a meeting m, where $a \in A$ and $m \in M$,

$othermeeting(d, a)$; an agent a has other meetings on a date d, where $a \in A$ and $d \in D$,

$chair(m, a)$; an agent a is a chairperson of a meeting m, where $a \in A$ and $m \in M$,

$golfcompe(d, a)$; an agent a has a golf competition on a date d, where $a \in A$ and $d \in D$.

Each agent's schedule is represented by facts and rules in the defeasible logic. There are two kinds of provabilities, definite provability and defeasible provability in the defeasible logic as shown in the previous section. If a literal can be proved definitely in a defeasible theory, the literal can be regarded as a *definite schedule*, and if a literal can be proved defeasibly, the literal can be regarded as a *defeasible schedule*.

R1 ; $avail(d, a) \wedge d \in S \Rightarrow part(m, a)$,
R2 ; $chair(m, a) \rightarrow part(m, a)$,
R3 ; $\neg avail(d, a) \leadsto \neg part(m, a)$,
R4 ; $othermeeting(d, a) \Rightarrow \neg avail(d, a)$,
R5 ; $\{ \ \} \Rightarrow avail(d, a)$,
R6 ; $golfcompe(d, a) \Rightarrow \neg avail(d, a)$,
where $a \in A$, $m \in M$, and $d \in D$.

According to the negotiation between the cordinator and the agents, each agent's schedule is decided. First we assume that the agent a_1 has other meetings on both dates d_α and d_β, the agent a_2 has a meeting on the date d_α, and the agent a_3 has a golf competition on the date d_β, as the initial knowledge. Moreover, there is a superiority relation between the rules, **R2** > **R1**.

Each agent's initial knowledge ;

$$F_1{}^i = \{othermeeting(d_\alpha, a_1), othermeeting(d_\beta, a_1)\}, \\ F_2{}^i = \{othermeeting(d_\alpha, a_2)\}, \\ F_3{}^i = \{golfcompe(d_\beta, a_3)\}.$$

Then, if the coordinator suggests the dates $S = \{d_\alpha, d_\beta\}$ for the meetings that each agent is required to participate, each agent's knowledge has to be changed as follows. We call this situation the first stage.

$$F_j{}^1 = \{d_\alpha \in S, d_\beta \in S\} \cup F_j{}^1, \\ \text{where } j = 1, 2, 3.$$

We obtain a defeasible theory $(F, C, R, >)$, where,

$$F = F_1{}^1 \cup F_2{}^1 \cup F_3{}^1, \\ C = \{\{avail(d, a), \neg avail(d, a)\}, \\ \{part(m, a), \neg part(m, a)\}\}, \\ R = \{\mathbf{R1}, \cdots, \mathbf{R6}\},$$

$\mathbf{R2} > \mathbf{R1}$, and
$a \in A, m \in M$, and $d \in S$.

We obtain the following VALPSN P_1 and its stable model I_1.

$P_1 = \{othermeeting(d_\alpha, a_1) : (3,0),$
$othermeeting(d_\beta, a_1) : (3,0),$
$othermeeting(d_\alpha, a_2) : (3,0),$
$golfcompe(d_\beta, a_3) : (3,0),$
$d_\alpha \in S : (3,0),\ d_\beta \in S : (3,0),$
$avail(d,a):(2,2) \wedge d \in S:(2,0) \wedge \sim part(m,a):(0,3)$
$\rightarrow part(m,a):(1,0),$
$avail(d,a):(2,0) \wedge d \in S:(2,0) \wedge \sim avail(d,a):(0,2) \wedge$
$\sim part(m,a):(0,3) \rightarrow part(m,a):(2,0),$
$chair(m,a) : (3,0) \rightarrow part(m,a) : (3,0),$
$\sim othermeeting(d,a):(2,0) \wedge \sim golfcompe(d,a):(2,0)$
$\wedge \sim avail(d,a):(0,3) \rightarrow avail(d,a):(2,0)\},$
where $a \in A, m \in M$, and $d \in D$.

The stable model I_1 of the VALPSN P_1 contains

$\{avail(d_\alpha, a_1) : (0,0),$	$avail(d_\beta, a_1) : (0,0),$
$avail(d_\alpha, a_2) : (0,0),$	$avail(d_\beta, a_2) : (2,0),$
$avail(d_\alpha, a_3) : (2,0),$	$avail(d_\beta, a_3) : (0,0),$
$part(m_1, a_1) : (0,0),$	$part(m_2, a_1) : (0,0),$
$part(m_1, a_2) : (0,0),$	$part(m_2, a_2) : (2,0),$
$part(m_1, a_3) : (2,0),$	$part(m_2, a_3) : (0,0)\}$

The stable model I_1 means that the agent a_2 and the agent a_3 participate the meeting m_2 on the date d_β and the meeting m_1 on the date d_α, respectively, are defeasible schedules.

At the second stage, the coordinator negotiate with each invitee to participate the meetings from the above information. To do that, the coordinator appoints the agent a_1 and the agent a_2 to the chairpersons of the meeting m_2 and the meeting m_1, respectively. These information are added to the facts in the defeasible theory as definite schedules. Therefore, the facts are updated as follows :

$${F_1}^2 = {F_1}^1 \cup \{chair(m_2, a_1)\}$$
$${F_2}^2 = {F_2}^1 \cup \{chair(m_1, a_2)\}$$

We obtain the following VALPSN P_2 and its stable model I_2.

$P_2 = \{othermeeting(d_\alpha, a_1) : (3,0),$
$othermeeting(d_\beta, a_1) : (3,0),$
$othermeeting(d_\alpha, a_2) : (3,0),$
$golfcompe(d_\beta, a_3) : (3,0),$
$d_\alpha \in S : (3,0),\ d_\beta \in S : (3,0),$
$chair(m_2, a_1) : (3,0),$
$chair(m_1, a_2) : (3,0),$
$avail(d,a):(2,2) \wedge d \in S:(2,0) \wedge \sim part(m,a):(0,3) \rightarrow$
$part(m,a):(1,0),$
$avail(d,a):(2,0) \wedge d \in S:(2,0) \wedge \sim avail(d,a):(0,2)$
$\wedge \sim part(m,a):(0,3) \rightarrow part(m,a):(2,0),$
$chair(m,a):(3,0) \rightarrow part(m,a):(3,0),$
$\sim othermeeting(d,a):(2,0) \wedge \sim golfcompe(d,a):(2,0)$
$\wedge \sim avail(d,a):(0,3) \rightarrow avail(d,a):(2,0)\},$
where $a \in A, m \in M$, and $d \in D$.

The stable model I_2 of the VALPSN P_2 contains

$\{avail(d_\alpha, a_1) : (0,0),$	$avail(d_\beta, a_1) : (0,0),$
$avail(d_\alpha, a_2) : (0,0),$	$avail(d_\beta, a_2) : (2,0),$
$avail(d_\alpha, a_3) : (2,0),$	$avail(d_\beta, a_3) : (0,0),$
$part(m_1, a_1) : (0,0),$	$part(m_2, a_1) : (3,0),$
$part(m_1, a_2) : (3,0),$	$part(m_2, a_2) : (2,0),$
$part(m_1, a_3) : (2,0),$	$part(m_2, a_3) : (0,0)\}$

In the stable model I_2, It becomes a definite schedule for the agents a_1 and a_2 to participate the meetings m_2 and m_1. However, there is still no meeting scheduled that every invitee participates. Therefore, the coordinator negotiates with the agents again.

At the third stage, since the agent a_3 can not participate the meeting m_2 because of a golf competition on the same date d_β, the coordinator tells the priority between the appointments to each agent by adding a superiority relation, $\mathbf{R5} > \mathbf{R6}$ to the defeasible theory. Then, we obtain the following VALPSN P_3 and its stable model I_3.

$P_3 = \{othermeeting(d_\alpha, a_1) : (3,0),$
$othermeeting(d_\beta, a_1) : (3,0),$
$othermeeting(d_\alpha, a_2) : (3,0),$
$golfcompe(d_\beta, a_3) : (3,0),$
$d_\alpha \in S : (3,0),\ d_\beta \in S : (3,0),$
$chair(m_2, a_1) : (3,0),$
$chair(m_1, a_2) : (3,0),$
$avail(d,a):(2,2) \wedge d \in S:(2,0) \wedge \sim part(m,a):(0,3) \rightarrow$
$part(m,a):(1,0),$
$avail(d,a):(2,0) \wedge d \in S:(2,0) \wedge \sim avail(d,a):(0,2) \wedge$
$\sim part(m,a):(0,3) \rightarrow part(m,a):(2,0),$
$chair(m,a):(3,0) \rightarrow part(m,a):(3,0),$
$\sim othermeeting(d,a):(2,0) \wedge \sim golfcompe(d,a):(2,0)$
$\wedge \sim avail(d,a):(0,3) \rightarrow avail(d,a):(2,0),$
$golfcompe(d,a):(2,0) \wedge \sim avail(d,a):(0,3) \rightarrow$
$avail(d,a):(2,0),$
$golfcompe(d,a):(2,0) \wedge \sim avail(d,a):(3,0) \rightarrow$
$avail(d,a):(0,1)\}$
where $a \in A, m \in M$, and $d \in D$.

The stable model I_3 of the VALPSN P_3 contains

$$\begin{array}{ll} \{avail(d_\alpha, a_1) : (0,0), & avail(d_\beta, a_1) : (0,0), \\ avail(d_\alpha, a_2) : (0,0), & avail(d_\beta, a_2) : (2,0), \\ avail(d_\alpha, a_3) : (2,0), & avail(d_\beta, a_3) : (2,1), \\ part(m_1, a_1) : (0,0), & part(m_2, a_1) : (3,0), \\ part(m_1, a_2) : (3,0), & part(m_2, a_2) : (2,0), \\ part(m_1, a_3) : (2,0), & part(m_2, a_3) : (2,0)\} \end{array}$$

In the stable model I_3, It becomes a defeasible schedule for the agent a_3 to participate the meeting m_2. Therefore, it becomes a defeasible schedule or a definite schedule for every agent to participate the meeting m_2 on the date d_β.

The conflict resolving process(negotiation) described above is just one example. There may be many such processes. However, it seems difficult to decide which process is best to resolve conflicts. We would like to propose a problem to find the most efficient process to resolve conflicts as a future work.

Conclusion

We have proposed a method to resolve conflicts between agents based on the defeasible reasoning by VALPSN. We summarize the advantages and the disadvantages of the method.

Advantage

- Since there is a computing procedure for the stable models of VALPSN, it is possible to implement in computers. We have already had the translation system from defeasible theories into VALPSNs and the computing system of their stable models.
- Since VALPSN is a sort of paraconsistent logic programs, it is easy to deal with conflicts in VALPSN, although we have not emphasized it in this paper.

Disadvantage

- It is difficult to define the original scenario by defeasible theories.
- The translation from defeasible theories into VALPSNs is complicated. If it is computerized, it takes long time to translate them.
- The computation of the VALPSN stable models is not efficient.

References

Billington,D. Conflicting Literals and Defeasible Logic. In *Proc. 2nd Australian Workshop on Commonsense Reasoning*, pages 1-15, Perth, Australia, Dec. 1997.

Blair,H.A. and Subrahmanian,V.S. Paraconsistent Logic Programming. *Theoretical Computer Science*, 68:135–154, 1989.

Da Costa,N.C.A., Subrahmanian,V.S., and Vago,C. The Paraconsistent Logics $P\mathcal{T}$. *Zeitschrift für Mathematische Logic und Grundlangen der Mathematik*, 37:139-148, 1989.

Gelfond,M. and Lifschitz, V. The Stable Model Semantics for Logic Programming. *Proc. 5th International Conference and Symposium on Logic Programming*, pages 1070-1080, 1989.

Kifer,M. and Subrahmanian V.S. Theory of Generalized Annotated Logic Programming and its Applications. J.Logic Programming, Vol.12, No.4, pages 335-368, 1992.

Lloyd,J.W. *Foundations of Logic Programming* 2nd edition. Springer-Verlag, 1987.

Nakamatsu,K.and Suzuki, A. Annotated Semantics for Default Reasoning. In *Proc. 3rd Pacific Rim International Conference on Artificial Intelligence*, pages 180-186, 1994.

Nakamatsu,K.and Suzuki, A. A Nonmonotonic ATMS Based on Annotated Logic Programs. In *Agents and Multi-Agents Systems*, LNAI 1441, pages 79-93, Springer-Verlag, 1998.

Nakamatsu,K.and Abe, J.M. Reasonings Based on Vector Annotated Logic Programs. In *Computational Intelligence for Modelling, Control & Automation*, Concurrent Systems Engineering Series, Vol.55, pages 396-403, IOS Press, 1999.

Nute,D. Basic Defeasible Logic. In *Intensional Logics for Programming*, pages 125-154, Oxford University Press, 1992.

Subrahmanian,V.S. Amalgamating Knowledge Bases. ACM Transactions on Database Systems, Vol.19, No.2, pages 291-331, 1994.