Logic and Logical Philosophy
Volume 8 (2000), 181–205

Kazumi Nakamatsu

ON THE RELATION BETWEEN VECTOR ANNOTATED LOGIC PROGRAMS AND DEFEASIBLE THEORIES

Abstract. In this paper, we propose an annotated logic program called a VALPSN (Vector Annotated Logic Program with Strong Negation) to deal with defeasible reasoning. We propose a translation from Billington's defeasible theory into VALPSN and clarify the relation between them based on the translation.

Keywords: annotated logic program, defeasible logic, logic programming, nonmonotonic reasoning.

1. Introduction and motivation

We have proposed ALPSN (Annotated Logic Program with Strong Negation) and shown that it can provide the declarative semantics for default reasoning and a non-monotonic ATMS [7, 8]. However, ALPSN is not so appropriate for dealing with defeasible reasoning or decision making. In this paper, we propose a new version of ALPSN called VALPSN(Vector Annotated Logic Program with Strong Negation) to deal with defeasible reasoning in a framework of annotated logic programming. In order to clarify the relation between VALPSN and defeasible reasoning, we also provide translation rules from Billington's defeasible theory [2] into VALPSN and show that there is a correspondence between them as shown in Figure 2.

Annotated logics are a family of paraconsistent logics that were proposed by da Costa et al. [4, 11] and developed by Blair and Subrahmanian [3] from

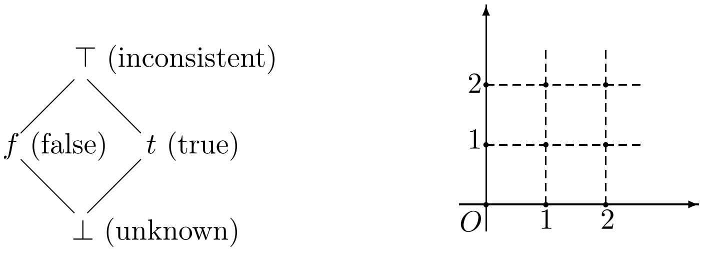

Figure 1. The Lattices $\mathcal{T}$ and $\mathcal{T}_v$

the viewpoint of logic programming. We extended the annotated logic program to ALPSN aiming for treating nonmonotonic reasoning and showed that ALPSN has stable model semantics [5]. Generally, each annotated atomic formula has a truth value called an *annotation* and the set of annotations constitutes a complete lattice. One of the most basic complete lattices of annotations is

$$\mathcal{T} = \{\, \bot \text{ (unknown)},\ f \text{ (false)},\ t \text{ (true)},\ \top \text{ (inconsistent)} \,\}$$

shown in Figure 1 and called *Lattice-4*.

Generally, annotated logics have two kinds of negations, an epistemic one and an ontological one. The epistemic negation ($\neg$) for an annotated atom is defined as a unary function from an annotation to an annotation. For example, the epistemic negation over $\mathcal{T}$ is defined as:

$$\begin{array}{ll} \neg(p\!:\!f) = p\!:\!\neg(f) = p\!:\!t\,, & \neg(p\!:\!t) = p\!:\!\neg(t) = p\!:\!f\,, \\ \neg(p\!:\!\bot) = p\!:\!\neg(\bot) = p\!:\!\bot\,, & \neg(p\!:\!\top) = p\!:\!\neg(\top) = p\!:\!\top\,. \end{array}$$

On the other hand, the ontological negation ($\sim$) is a strong negation similar to classical ones.

We compare ALPSN, VALPSN and Billington's defeasible logic [2] with taking a well-known example called **Nixon Diamond**.

[Nixon Diamond] We have two facts and two normal default rules.

F1	"Nixon is a Quaker"	$\{\, q(n) \,\}$,
F2	"Nixon is a Republican"	$\{\, r(n) \,\}$,
R1	"If Nixon is a Quaker, then he is a Pacifist"	$\{\, q(n)\!:\!p(n)/p(n) \,\}$,
R2	"If Nixon is a Republican, then he is not a Pacifist"	$\{\, r(n)\!:\!\neg p(n)/\neg p(n) \,\}$.

Then, they are translated into an ALPSN

$$\begin{aligned} P \;=\; \{\, & q(n):t,\;\; q(n):t \wedge \sim p(n):f \rightarrow p(n):t, \\ & r(n):t,\;\; r(n):t \wedge \sim p(n):t \rightarrow p(n):f \,\}, \end{aligned}$$

and the ALPSN P has two stable models, $I_1 = \{\, q(n):t,\ r(n):t,\ p(n):t \,\}$ and $I_2 = \{\, q(n):t,\ r(n):t,\ \neg p(n):t \,\}$. Intuitively, I_1 shows that "Nixon is a Pacifist" and I_2 shows that "Nixon is not a Pacifist". Clearly, they show that **Nixon Diamond** contains a conflict $\{\ p(n),\ \neg p(n)\,\}$ that may not be resolved. However, if we introduce a superiority relation between the rules **R1** and **R2**, the conflict can be resolved. For example, if we assume that the rule **R2** is more convincible than the rule **R1** , then we would be able to obtain a more convincible conclusion, "Nixon is not a Pacifist". This kind of inference can be formalized in defeasible logic. The defeasible **Nixon Diamond** is called **d-Nixon Diamond**. It is difficult to formalize **d-Nixon Diamond** in ALPSN. In order to overcome such difficulty in the framework of annotated logic programming, we propose VALPSN, a new version of ALPSN. The primary difference between ALPSN and VALPSN is in their annotations. Each annotated literal in VALPSN has a 2-dimensional vector annotation called a *vector annotation*. Generally, each component of a vector annotation is a non-negative integer, and the first component expresses the strength of true and the second one expresses the strength of false. In this paper, we assume that the non-negative integer is less than or equal to 2 to formulate Billington's defeasible theories in VALPSN. Therefore, we need the following complete lattice $\mathcal{T}_{\mathrm{v}}$ of vector annotations shown in Figure 1:

$$\mathcal{T}_{\mathrm{v}} = \{\, (i,j) \mid 0 \leq i \leq 2,\ 0 \leq j \leq 2, i \text{ and } j \text{ are non-negative integers} \,\}.$$

For example, a vector annotated literal $p:(2,1)$ can be informally interpreted as "p is known to be true of strength 2 and false of strength 1". Then, if we consider that "true of strength 2" preempts "false of strength 1", the literal p may be intuitively interpreted as relatively true. The ordering $\preceq$ on the complete lattice $\mathcal{T}_{\mathrm{v}}$ of vector annotations is defined as: let $\vec{v}_1 = (x_1, y_1)$, $\vec{v}_2 = (x_2, y_2)$, and $\vec{v}_1, \vec{v}_2 \in \mathcal{T}_{\mathrm{v}}$,

$$\vec{v}_1 \preceq \vec{v}_2 \quad \text{iff} \quad x_1 \leq x_2 \text{ and } y_1 \leq y_2\,.$$

The epistemic negation ($\neg$) in VALPSN is also defined as a unary function from a vector annotation to a vector annotation such that $\neg p:(i,j) = p:(j,i)$ and the strong negation is defined as well as ALPSN. **d-Nixon Diamond** can be formulated as a VALPSN

$$\begin{aligned} P \;=\; \{\, & q(n):(2,0), \quad r(n):(2,0), \\ & q(n):(1,0) \wedge \sim r(n):(1,0) \wedge \sim p(n):(0,1) \to p(n):(1,0), \\ & r(n):(1,0) \wedge \sim p(n):(1,0) \to p(n):(0,1)\, \}. \end{aligned}$$

VALPSN also has the stable semantics as well as ALPSN and the VALPSN P has only one stable model, $I = \{\, q(n):(2,0),\ r(n):(2,0),\ \neg p(n):(1,0)\,\}$. This stable model I can be interpreted that "Nixon is a Quaker" and "Nixon is a Republican" are facts $(q(n):(2,0), r(n):(2,0))$, and "Nixon might not be a Pacifist" $(\neg p(n):(1,0))$.

Defeasible logics are known as nonmonotonic formalisms that can deal with defeasible reasoning. A defeasible logic was first published by Nute [9]. Since then, some systems of the same kind of defeasible logics have been developed [1, 2, 10]. We introduce Billington's defeasible logic in [2], The defeasible logic has facts and three kinds of rules, a strict rule $(A \to q)$, a defeasible rule $(A \Rightarrow q)$ and a defeater $(A \rightsquigarrow q)$, and a superiority relation $>$ between those rules to carry out defeasible reasoning. There are two kinds of provability, definite(strict) provability and defeasible provability, the first one is the derivability based on facts and strict rules, and the second one is that based on defeasible reasoning. Let us take **d-Nixon Diamond** as an example for Billington's defeasible theories. It has two facts, **F1** and **F2**, and two defeasible rules, **R1** and **R2**. It also has a superiority relation, **R2** $>$ **R1**.

$$\begin{array}{llll} \textbf{F1} & q(n), & \textbf{F2} & r(n), \\ \textbf{R1} & q(n) \Rightarrow p(n), & \textbf{R2} & r(n) \Rightarrow \neg p(n). \end{array}$$

We show an informal interpretation for the defeasible theory **d-Nixon Diamond**. Since the antecedents $q(n)$ and $r(n)$ of the defeasible rules **R1** and **R2** are facts, there is the conflict $\{p(n), \neg p(n)\}$ between the consequents of those rules. However, as there is the superiority relation **R2** $>$ **R1** and there does not exist the fact $p(n)$, the consequent $\neg p(n)$ of the rule **R2** can be derived. We can formalize defeasible reasoning in VALPSN. Rules and facts are expressed in VALPSN clauses, and superiority relations between rules are represented by vector annotations. Since the notion of definite provability can be regarded as having stronger evidence than that of defeasible provability, we assume the following correspondence between those provabilities and the satisfiability for vector annotated literal p.

$$\begin{array}{lcl} p \text{ is definitely provable} & \Longleftrightarrow & \models p:(2,0) \\ p \text{ is defeasibly provable} & \Longleftrightarrow & \models p:(1,0). \end{array}$$

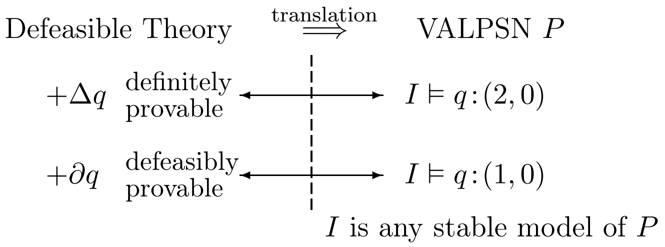

Figure 2. The Relation Between Defeasible Theory and VALPSN

Then, we have the relation between Billington's defeasible theory and VALPSN based on the translation as shown in Figure 2.

This paper organized as follows. In section 2, we define VALPSN. In section 3, we review Billington's defeasible logic in [2]. In section 4, we provide translation rules from Billington's defeasible theory into VALPSN. Then, we prove that there is a correspondence between the provability of the defeasible logic and the stable model satisfiability for VALPSN.

2. Vector annotated logic program with strong negation (VALPSN)

In this section, we define the syntax and the stable model semantics for VALPSN. We assume that the reader is familiar with the usual notions of ordinary first order logics and logic programming in Lloyd[6].

2.1. Syntax for VALPSN

Generally, the set $\mathcal{T}_{\mathrm{v}}$ of vector annotations has an arbitrary, but fixed, complete lattice structure. Throughout this paper we assume that

$$\mathcal{T}_{\mathrm{v}} = \{\,(x,y) \mid 0 \le x \le 2, 0 \le y \le 2, x \text{ and } y \text{ are integers}\,\}.$$

The ordering on the lattice $\mathcal{T}_{\mathrm{v}}$ is denoted by a symbol $\preceq$.

Definition 2.1. Let $\vec{v}_1 = (x_1, y_1)$ and $\vec{v}_2 = (x_2, y_2)$,

$$\vec{v}_1 \preceq \vec{v}_2 \quad \text{iff} \quad x_1 \le x_2 \text{ and } y_1 \le y_2\,.$$

VALPSN has two kinds of negation, an *epistemic negation* ($\neg$) and an *ontological negation* ($\sim$). The epistemic negation is a unary function from a vector annotation to a vector annotation and defined as the exchange of each component of the vector. Let p be a literal and x, y integers, $\neg(p\colon(x,y)) = p\colon(y,x)$.

Definition 2.2. If A is a literal, then, $A:\vec{v}$ is called a *vector annotated literal*, where $\vec{v} \in \mathcal{T}_{\mathrm{v}}$ and the $\vec{v}$ is called a *vector annotation.* Vector annotated formulas are defined recursively.

(1) Any vector annotated literal is a vector annotated formula,

(2) If F_1, F_2, and F are vector annotated formulas and x is an arbitrary variable, then, $\neg F$, $F_1 \vee F_2$, $F_1 \wedge F_2$, $F_1 \to F_2$, $\forall x F$, and $\exists x F$ are vector annotated formulas.

Note. Let p be a literal and $\vec{v}$ a vector annotation. $\neg p : \vec{v}$ means $\neg(p : \vec{v})$ rather than $(\neg p) : \vec{v}$ and $\neg p : \vec{v}$ is interpreted as $p : (\neg \vec{v})$. For instance, $\neg p : (2,1) = p : \neg(2,1) = p : (1,2)$. Therefore, we can eliminate all epistemic negations in VALPSN.

We call shortly the vector annotated formulas as just formulas if it does not cause confusion in the rest of this paper. The ontological negation is a strong negation similar to classical ones and defined.

Definition 2.3 (Strong Negation [4]).

$$\sim F \stackrel{\text{def}}{=} F \to ((F \to F) \wedge \neg(F \to F)),$$

where F is an arbitrary formula. $\sim$ is called *strong negation.*

Note. The epistemic negation followed by $F \to F$ is not interpreted as a mapping between vector annotations. It is interpreted as a classical negation.

Definition 2.4. Let p be an ordinary literal. $p:(m,0)$ is called a *well vector annotated literal*, where m is a non-negative integer. If $L_0, \ldots, L_n$ are well vector annotated literals,

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

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

We give intuitive interpretations for vector annotated literals before addressing the formal semantics for VALPSN. Usually, vector annotated literals can be interpreted epistemically. A vector annotated literal $A:(i,j)$ is interpreted as "A is known to be true of strength i and false of strength j". $A:(0,0)$ is interpreted as "A is known to have no information to support A".

2.2. Semantics for VALPSN

We 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 $I\colon B_P \to \mathcal{T}_{\mathrm{v}}$. Usually, I is denoted by the set $\{p\colon \bigsqcup \vec{v}_i \mid I \vDash p\colon \vec{v}_1 \wedge \cdots \wedge p : \vec{v}_n\}$, where $\bigsqcup \vec{v}_i$ is the least upper bound of $\{\vec{v}_1, \ldots, \vec{v}_n\}$. The ordering $\preceq$ on $\mathcal{T}_{\mathrm{v}}$ is extended to interpretations in the natural way. Let I_1 and I_2 be any interpretations, and A be an atom.

$$I_1 \preceq I_2 \overset{\text{def}}{\Longleftrightarrow} (\forall A \in B_P)(I_1(A) \preceq I_2(A)).$$

Definition 2.5 (Satisfaction [3]). An interpretation I is said to satisfy

(1) a vector annotated formula F iff it satisfies every closed instance of F,

(2) a vector annotated atom $A\colon \vec{v}$ iff $I(A) \succeq \vec{v}$,

(3) a vector annotated literal $\neg A\colon \vec{v}$ iff $I(A) \succeq \neg(\vec{v})$,

(4) a vector annotated formula $\neg F$ iff I does not satisfy F, where F is a compound formula,

(5) the satisfaction of the other formulas, $F_1 \wedge F_2$, $F_1 \vee F_2$, $F_1 \to F_2$, $\forall x F$, and $\exists x F$ are the same as classical logics.

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 $\mathcal{T}_{\mathrm{v}}$:

$$\begin{aligned} T_P(I)(A) \quad = \quad & \bigsqcup\{\, \vec{v} \mid B_1 \wedge \cdots \wedge B_m \wedge {\sim} C_1 \wedge \cdots \wedge {\sim} C_n \to A\colon \vec{v} \\ & \qquad \text{is a ground instance of a vasn-clause in } P \text{ and} \\ & \qquad I \vDash B_1 \wedge \cdots \wedge B_m \wedge {\sim} C_1 \wedge \cdots \wedge {\sim} C_n \,\}, \end{aligned}$$

where the notation $\bigsqcup$ denotes the least upper bound. The least upper bound always exists, since $\mathcal{T}_{\mathrm{v}}$ is a complete lattice under the ordering $\preceq$. 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:

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

The following well-known results in terms of VALP (with no strong negation) P and the operator T_P hold [3].

PROPOSITION 2.1. *If P is an VALP without strong negation,*

- *T_P is a monotonic function,*
- *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 [5] for general logic program to VALPSN. In the rest of this paper, we assume that a VALPSN P is a set of ground clauses. There is no loss of generality in making this assumption, since any logic program in the sense of [6] may be viewed as such a set of clauses by instantiating all variables occurring in clauses. First, we describe the Gelfond-Lifschitz transformation for VALPSN.

DEFINITION 2.6 (Gelfond-Lifschitz transformation [5]). Let I be any interpretation and P a VALPSN. P^I, the *Gelfond-Lifschitz transformation* of the VALPSN P with respect to I, is the VALP obtained from P by deleting

- each clause that has a strongly negated vector annotated literal $\sim(C:\vec{v})$ in its body with $I \models (C:\vec{v})$,
- all strongly negated vector annotated literals in the bodies of the remaining clauses.

Since P^I contains no strong negation, it has the unique least model that is given as $T_{P^I} \uparrow \omega$ by Proposition 2.1.

DEFINITION 2.7 (Stable Model [5]). 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 2.1. *Let p, q, and r be literals and a VALPSN*

$$\begin{aligned} P &= \{\, q:(2,0),\ \ q:(1,0) \wedge \sim p:(0,2) \rightarrow p:(1,0), \\ &\qquad r:(2,0),\ \ r:(1,0) \wedge \sim p:(2,0) \rightarrow p:(0,1)\, \}. \end{aligned}$$

If $I = \{q:(2,0),\ r:(2,0),\ p:(1,1)\}$, then,

$$P^I = \{q:(2,0), q:(1,0) \rightarrow p:(1,0), r:(2,0), r:(1,0) \rightarrow p:(0,1)\}$$

and $T_{P^I} \uparrow \omega = I$. Thus, I is the only one stable model of the VALPSN P.

3. Defeasible logic

In this section, we introduce Billington's defeasible logic in [2]. The defeasible logic contains three kinds of rules, strict rules, defeasible rules and defeaters. Conflicts between defeasible rules with incompatible consequents are resolved by using an explicit superiority relation on rules. The defeasible logic is defined as a set of conditions on nodes of proof trees.

3.1. basic notation and terminology

We begin by introducing the basic notations and terminology for the defeasible logic briefly.

Definition 3.1. The set of all integers is denoted by $\mathbf{Z}$, the set of positive integers by $\mathbf{P}$, and the set of non-negative integers by $\mathbf{N}$. If m and n are integers then $[m..n] = \{ i \in \mathbf{Z} \mid m \leq i \leq n\}$. The cardinality of any set S is denoted by $|S|$. The length of a sequence P is denoted by $|P|$. Let $P = (P(1), P(2), \dots, P(|P|))$ be a finite sequence. If $i \in [1..|P|]$, then $P(1..i) = (P(1), \dots, P(i))$; otherwise $P(1..i)$ is undefined. The notation $x \in P$ means that there exists $j \in [1..|P|]$ such that $x = P(j)$.

Definition 3.2. The *alphabet* $\mathcal{A}$ is the union of the following four pairwise disjoint sets of symbols: a nonempty countable set of proposition symbols; the set $\{ \neg, \rightarrow, \Rightarrow, \rightsquigarrow \}$ of connectives; the set $\{ +, -, \Delta, \partial \}$ of positive, negative, definite, and defeasible proof symbols; the set of punctuation marks consisting of the comma, braces and parentheses.

If q is any literal, the complement $\neg q$ 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 can not 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. The set on the left of the arrow is called the *antecedent* of the rule, and the literal on the right of the arrow is called the *consequent* of the rule. A rule containing the *strict arrow* $\rightarrow$, for example, $\boldsymbol{A} \rightarrow q$, is called a *strict rule*. The intuition is that whenever all the literals in $\boldsymbol{A}$ are accepted then q must be accepted. A rule containing the *defeasible arrow* $\Rightarrow$, for example $\boldsymbol{A} \Rightarrow q$, is called a *defeasible rule*. If all the literals in $\boldsymbol{A}$ are accepted then q is accepted provided that there is an insufficient evidence against q. A rule containing the *defeater arrow* $\rightsquigarrow$, for example, $\boldsymbol{A} \rightsquigarrow \bar{q}$, is called a *defeating rule* or a *defeater*. If all the literals in $\boldsymbol{A}$ are accepted then $\boldsymbol{A} \rightarrow \bar{q}$ is an evidence against q, but not for $\bar{q}$. It should be noted that the antecedent of a rule can be empty set. Such a rule is called a *presumption*. For example, if p is a presumption then it is written as $\{\,\} \Rightarrow p$.

Definition 3.3. Let R be any set of rules. The set of strict rules in R 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 $\mathrm{A}(r)$, and its consequent is denoted by $\mathrm{C}(r)$. The set of consequents of rules in R is denoted by $\mathrm{C}(R) = \{\mathrm{C}(r) \mid r \in R\}$. $R[q] \overset{\mathrm{def}}{=} \{r \mid r \in R \text{ and } q = \mathrm{C}(r)\}$. A *superiority relation* on R is any asymmetric binary relation $>$ on R.

All sets of conflicting literals are collected into a single conflict set denoted by $\mathcal{C}$. For each literal q, clearly $\{q, \bar{q}\}$ is a set of conflicting literals and $\{q, \bar{q}\} \in \mathcal{C}$.

Definition 3.4. $\mathcal{C}$ is a *conflict set* iff $\mathcal{C}$ is a set such that: if $C \in \mathcal{C}$, then C is a non-empty set of literals; if $C \in \mathcal{C}$, $S \subseteq C$, and $S \neq C$, then $S \notin \mathcal{C}$; and if q is a literal, then at least one subset of $\{q, \bar{q}\}$ is an element of $\mathcal{C}$.

We assume the conflict set indicates the set $\mathcal{C} = \{\{q, \neg q\} \mid q \text{ is a literal}\}\}$ hereafter, if we do not address particularly.

Definition 3.5. A *defeasible theory* over $\mathcal{C}$ is a quadruple $(F, \mathcal{C}, R, >)$ such that: F is a set of literals, often referred to as facts; $\mathcal{C}$ is a conflict set; R is a set of rules; $>$ is a superiority relation on R.

A defeasible theory together with a suitable deducibility relation $\vdash$ is called a *defeasible logic*. In the defeasible logic, notions "to be proved definitely" and "to be proved defeasibly" should be distinguished. In order to do so, a *tagged literal* is defined to be a tag followed by a literal.

Definition 3.6. A *tag* consists of two symbols; the first symbol is either $+$ or $-$, and the second symbol is either Δ or ∂.

Definition 3.7. A finite sequence $P = (P(1), \ldots, P(|P|))$ of tagged literals is called a *proof* in $(F, \mathcal{C}, R, >)$ iff for each $i \in [0..|P| - 1]$ the four inference conditions $+\Delta$, $-\Delta$, $+\partial$ and $-\partial$ all hold. If $\mathcal{C} = \{\{q, \bar{q}\} \mid q \text{ is a literal}\}$, the four conditions are;

$+\Delta$) if $P(i+1) = +\Delta q$, for some literal q, then either
- **.1)** $q \in F$; or
- **.2)** $\exists r \in R_{\mathrm{s}}[q]\ \forall a \in \mathrm{A}(r),\ +\Delta a \in P(1..i)$

$-\Delta$) if $P(i+1) = -\Delta q$, for some literal q, then
- **.1)** $q \notin F$, and
- **.2)** $\forall r \in R_{\mathrm{s}}[q]\ \exists a \in \mathrm{A}(r),\ -\Delta a \in P(1..i)$

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

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

An element of a proof is called a *line* of the proof.

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 can not be proved definitely, $+\partial q$ indicates that q is proved defeasibly, and $-\partial q$ indicates that it is proved that q can not be proved defeasibly.

Definition 3.8 ([2]). Let $T = (F, \mathcal{C}, R, >)$ be a defeasible theory. If tq is a tagged literal, then $T \vdash tq$ iff tq is a line in a proof in T, where $t \in \{+\Delta, -\Delta, +\partial, -\partial\}$.

Example 3.1 (Genetically Altered Penguin [2]). *Those are known that penguins* (p) *are definitely birds* (b)*, that defeasibly birds fly* (f)*, and that defeasibly penguins do not fly* ($\neg f$)*. Suppose a penguin that has large wings and* *flight muscles. Such a genetically altered penguin (gap) might fly. Suppose that Opus (o) is a genetically altered penguin. Then, a defeasible theory T capturing this situation is as below.*

$$\begin{array}{llllll} \mathbf{F1} & gap(o), & \mathbf{R1} & gap(o) \to p(o), & \mathbf{R2} & p(o) \to b(o), \\ \mathbf{R3} & b(o) \Rightarrow f(o), & \mathbf{R4} & p(o) \Rightarrow \neg f(o), & \mathbf{R5} & gap(o) \leadsto f(o). \end{array}$$

Let $T = (F, \mathcal{C}, R, >)$*, where* $F = \{\mathbf{F1}\}$*,* $R = \{\mathbf{R1}, \mathbf{R2}, \mathbf{R3}, \mathbf{R4}\}$*, and* $>$ *is defined by* $\mathbf{R5} > \mathbf{R4}$ *and* $\mathbf{R4} > \mathbf{R3}$*. We show an intuitive derivation. The formal one can be found in [2]. From* **F1***, we have* $+\Delta\, gap(o)$*. From* **F1** *and* **R1***, we have* $+\Delta p(o)$*. From* **R2** *and* $+\Delta p(o)$*, we have* $+\Delta b(o)$*. Since* $\mathbf{R4} > \mathbf{R3}$*, the consequent* $f(o)$ *of* **R3** *is defeated by the consequent* $\neg f(o)$ *of* **R4***. Thus,* $+\partial f(o)$ *can not be derived. Since* $\mathbf{R5} > \mathbf{R4}$*, the consequent* $\neg f(o)$ *of* **R4** *is defeated by the defeater* **R5***. Thus,* $+\partial\neg f(o)$ *can not be derived from* **R4***. In the defeasible logic, defeaters can not be used to derive its consequent. Therefore, we have neither* $+\partial f(o)$ *nor* $+\partial\neg f(o)$ *in the defeasible theory* T*.*

4. From defeasible theory into VALPSN

In this section, we provide the translation from the defeasible logic into VALPSN. Strict rules, defeasible rules, defeaters and facts in the defeasible logic are translated into vasn-clauses. Furthermore, we show the relation between definite or defeasible provabilities in the defeasible logic and VALPSN stable model satisfiability based on the translation as shown in Figure 2.

4.1. Translation

The basic idea of the translation is as follows. We want to express the two kinds of provabilities, "definite" one and "defeasible" one, based on vector annotations. We consider that if a literal is definitely provable, the literal has stronger information to support it than the case of "defeasibly provable". We use three levels, $0, 1$ and 2, to express the strength of information. Therefore, "a literal q is definitely provable ($+\Delta q$)" may be expressed by "a vector annotated literal $q:(2,0)$ is derivable" and "a literal q is defeasibly provable ($+\partial q$)" may be expressed by "a vector annotated literal $q:(1,0)$ is derivable". Moreover, the defeasible logic is nonmonotonic and we represent the nonmonotonicity by the strong negation in VALPSN. Generally, a VALPSN may have more than two stable models or no stable model. We consider the correspondence between the provability of the defeasible logic and the satisfiability of VALPSN. Let us remind the example **Nixon Diamond** for default reasoning. If we regard **Nixon Diamond** as a default theory, it can be translated into a VALPSN

$$\begin{array}{lllll} P = \{ & \mathbf{F1} & q(n):(1,0), & \mathbf{R1} & q(n):(1,0) \wedge \sim p(n):(0,1) \to p(n):(1,0) \\ & \mathbf{F2} & r(n):(1,0), & \mathbf{R2} & r(n):(1,0) \wedge \sim p(n):(1,0) \to p(n):(0,1) \,\} \end{array}$$

as well as the translation from default theory into ALPSN in [7]. Then, the VALPSN P has two stable models,

$$\begin{aligned} I_1 &= \{\, q(n):(1,0), \quad r(n):(1,0), \quad p(n):(1,0) \,\}, \\ I_2 &= \{\, q(n):(1,0), \quad r(n):(1,0), \quad p(n):(0,1) \,\}. \end{aligned}$$

The stable model I_1 satisfies $p(n):(1,0)$ and the stable model I_2 satisfies $p(n):(0,1)$, and they show only that there exist two conflicting worlds independently. On the other hand, if we formulate the example **Nixon Diamond** as a defeasible theory, we obtain a defeasible theory $T = (F, \mathcal{C}, R, >)$, where $F = \{q(n), r(n)\}$, $R = \{\, \{q(n)\} \Rightarrow p(n), \{r(n)\} \Rightarrow \neg p(n) \,\}$ and no superiority relation. Then, neither $p(n)$ nor $\neg p(n)$ are defeasibly provable in the defeasible theory T. The difference says that; if we translate defeasible theories into VALPSNs, we have to bring skeptical view into the satisfiability of VALPSN stable model semantics, that is to say, we have to regard the intersection of all the stable models as the model of the original defeasible theory.

We assume that there is a correspondence between the provability of the defeasible logic and the satisfiability of VALPSN stable model semantics.

[Assumption]
Let q be a literal, T a defeasible theory, I an interpretation of a VALPSN P, and M_P the set of all stable models for the VALPSN P. Then,

$$\begin{aligned} T \vdash +\Delta q &\iff \forall I \in M_P, I \models q:(2,0), \\ T \vdash +\partial q &\iff \forall I \in M_P, I \models q:(1,0), \\ T \vdash -\Delta q &\iff \exists I \in M_P, I \not\models q:(2,0), \\ T \vdash -\partial q &\iff \exists I \in M_P, I \not\models q:(1,0). \end{aligned}$$

Now we propose 6 translation rules **TR1**–**TR6** from facts and the three kinds of rules into vasn-clauses based on the four inference conditions, $\{+\Delta, -\Delta, +\partial, -\partial\}$. Let q be a literal, $T = (\, F, \mathcal{C}, R, > \,)$ a defeasible theory.

[Fact] A fact q can be used to derive $+\Delta q$. According to the inference condition $+\Delta.\mathbf{1}$, if there is a literal $q \in F$, $P(i+1) = +\Delta q$. Thus, we have;

Translation Rule 1 (TR1)

the fact q is translated into a vasn-clause $q:(2,0)$.

[Strict Rule] Let $\boldsymbol{A} \to q$ be a strict rule, where $\boldsymbol{A} = \{a_1, \dots, a_k\}$ and each a_j $(1 \le j \le k)$ is a literal. The strict rule can be used to derive both $+\Delta q$ and $+\partial q$. Therefore, we have to take into account the following two cases.

[Case 1] If the inference condition $+\Delta.\mathbf{2}$ is satisfied, there is a strict rule such that its antecedent is definitely provable and the consequent of the strict rule is definitely provable. Therefore, we have the following translation rule.

Translation Rule 2 (TR2)

the strict rule $\{\, a_1, \dots, a_k \,\} \to q$ is translated into a vasn-clause
$$a_1:(2,0) \wedge \cdots \wedge a_k:(2,0) \to q:(2,0).$$

The strict rule also can be used to derive $+\partial q$ when the antecedent of the strict rule is defeasibly provable. Thus, we have to consider the inference condition $+\partial.2$ as deriving $+\partial q$ by the strict rule $\boldsymbol{A} \to q$.

[Case 2] If the inference condition $+\partial.\mathbf{2}$ is satisfied;

(1) each $a_i \in \boldsymbol{A}$ $(1 \le i \le k)$ must be defeasibly provable by $+\partial.\mathbf{2.1}$;
(2) the complement $\bar{q}$ must not be definitely provable by $+\partial.\mathbf{2.2}$; and
(3) for each rule $s \in R[\bar{q}]$, either the antecedent of the rule s must not be defeasibly provable, or there must be a rule $t \in R_{\mathrm{sd}}[q]$ whose antecedent is defeasibly provable and the rule t is superior to the rule s, by $+\partial.\mathbf{2.3}$.

Basically, the defeasible reasoning is performed based on the superiority relation and if we reinterpret the above three conditions (1)–(3), we have to consider the following two cases **[I]** and **[II]** as deriving $+\partial q$ based on the superiority relation.

[I] Suppose that the strict rule $\boldsymbol{A} \to q$ has no superiority relation; or there are a rule $r \in R_{\mathrm{sd}}[\bar{q}]$, a superiority relation $\boldsymbol{A} \to q > r$ and no defeater superior to the strict rule $\boldsymbol{A} \to q$; then, in order to derive $+\partial q$ by the strict rule $\boldsymbol{A} \to q$,

(I-1) the antecedent $\boldsymbol{A}$ must be defeasibly provable,

(I-2) the conflicting literal $\bar{q}$ must not be defeasibly provable by other conflicting rules and

(I-3) the conflicting literal $\bar{q}$ must not be definitely provable.

Note. The item (I-2) is represented by a strongly negated vector annotated literal $\sim q:(0,1)$ and the item (I-3) is represented by a strongly negated vector annotated literal $\sim q:(0,2)$ independently, and "the conflicting literal $\bar{q}$ must be neither defeasibly nor definitely provable" is also represented as the strongly negated vector annotated literal $\sim q:(0,1)$.

[II] Suppose that there are a rule $r \in R[\bar{q}]$ and a superiority relation $r > \boldsymbol{A} \rightarrow q$; then, in order to derive $+\partial q$ by the strict rule $\boldsymbol{A} \rightarrow q$,

(II-1) the antecedent $\boldsymbol{A}$ must be defesibly provable,

(II-2) the antecedent of the rule r must not be defeasibly provable,

(II-3) the conflicting literal $\bar{q}$ must not be defeasibly provable by other rules than the rule r and

(II-4) the conflicting literal $\bar{q}$ must not be definitely provable.

We formulate these two cases as the following translation rule.

Translation Rule 3 (TR3)

Suppose that the strict rule $\boldsymbol{A} \rightarrow q$ has no superiority relation; or there are a rule $r \in R[\bar{q}]$, a superiority relation $\boldsymbol{A} \rightarrow q > r$ and no defeater superior to the strict rule $\boldsymbol{A} \rightarrow q$. Then, the strict rule $\{a_1, \ldots, a_k\} \rightarrow q$ is translated into a vasn-clause

$$a_1:(1,0) \wedge \cdots \wedge a_k:(1,0) \wedge \sim q:(0,1) \rightarrow q:(1,0).$$

Translation Rule 4 (TR4)

Suppose that there is a superiority relation $\boldsymbol{B} \circ \bar{q} > \boldsymbol{A} \rightarrow q$, $\circ \in \{\rightarrow, \Rightarrow, \leadsto\}$, $\boldsymbol{B} = \{b_1, \ldots, b_l\}$ and each b_i $(1 \leq i \leq l)$ is a literal. Then, the strict rule $\{a_1, \ldots, a_k\} \rightarrow q$ is translated into vasn-clauses

$$\begin{gathered}
a_1:(1,0) \wedge \cdots \wedge a_k:(1,0) \wedge \sim b_1:(1,0) \wedge \sim q:(0,1) \rightarrow q:(1,0), \\
a_1:(1,0) \wedge \cdots \wedge a_k:(1,0) \wedge \sim b_2:(1,0) \wedge \sim q:(0,1) \rightarrow q:(1,0), \\
\vdots \\
a_1:(1,0) \wedge \cdots \wedge a_k:(1,0) \wedge \sim b_l:(1,0) \wedge \sim q:(0,1) \rightarrow q:(1,0).
\end{gathered}$$

[Defeasible Rule] Let $\boldsymbol{A} \Rightarrow q$ be a defeasible rule, where $\boldsymbol{A} = \{a_1, \ldots, a_k\}$ and each a_j $(1 \leq j \leq k)$ is a literal. Since the defeasible rule can be used to derive only $+\partial q$, it is translated into vasn-clauses similarly to the **[Case 2]** of **[Strict Rule]**.

Translation Rule 5 (TR5)

Suppose that the defeasible rule $\boldsymbol{A} \Rightarrow q$ has no superiority relation; or there are a rule $r \in R[\bar{q}]$, a superiority relation $\boldsymbol{A} \Rightarrow q > r$ and no defeater superior to the defeasible rule $\boldsymbol{A} \Rightarrow q$. Then, the defeasible rule $\{a_1, \dots, a_k\} \Rightarrow q$ is translated into a vasn-clause

$$a_1 : (1, 0) \wedge \cdots \wedge a_k : (1, 0) \wedge {\sim}q : (0, 1) \rightarrow q : (1, 0).$$

Translation Rule 6 (TR6)

Suppose that there is a superiority relation $\boldsymbol{B} \circ \bar{q} > \boldsymbol{A} \Rightarrow q$, $\circ \in \{\rightarrow, \Rightarrow, \rightsquigarrow\}$, $\boldsymbol{B} = \{b_1, \dots, b_l\}$ and each b_i $(1 \le i \le l)$ is a literal. Then, the defeasible rule $\{a_1, \dots, a_k\} \Rightarrow q$ is translated into vasn-clauses

$$\begin{gathered} a_1 : (1, 0) \wedge \ldots \wedge a_k : (1, 0) \wedge \sim b_1 : (1, 0) \wedge {\sim}q : (0, 1) \rightarrow q : (1, 0), \\ a_1 : (1, 0) \wedge \ldots \wedge a_k : (1, 0) \wedge \sim b_2 : (1, 0) \wedge {\sim}q : (0, 1) \rightarrow q : (1, 0), \\ \vdots \\ a_1 : (1, 0) \wedge \ldots \wedge a_k : (1, 0) \wedge \sim b_l : (1, 0) \wedge {\sim}q : (0, 1) \rightarrow q : (1, 0). \end{gathered}$$

[Defeater] The consequent of a defeater can not be derived by the inference conditions $+\Delta$ or $+\partial$. The role of a defeater is not to derive its consequent but to prevent from deriving the conflicting literal by the rule inferior to the defeater. Therefore, a defeater itself does not have to be translated into vasn-clauses.

DEFINITION 4.1 (VALPSN-translation *tr*). Let $T = (F, \mathcal{C}, R, >)$ be a defeasible theory. Then, the VALPSN-translation $tr(T)$ of the defeasible theory T is defined as $tr(T) = tr(F) \cup tr(R)$. where $tr(F)$ is a VALPSN-translation of the set F of facts and $tr(R)$ is a VALPSN-translation of the set R of rules.

EXAMPLE 4.1. *Let a defeasible theory* $T = (\, F, \mathcal{C}, R, >\,)$, *the set of facts* $F = \{a\}$, *the set of rules* $R = \{\, \mathbf{R1}\ \{a\} \Rightarrow q,\ \mathbf{R2}\ \{b_1, b_2\} \rightarrow \neg q,\ \mathbf{R3}\ \{c\} \rightsquigarrow \neg q\}$, *and a superiority relation* $\mathbf{R3} > \mathbf{R1}$. *Then, by the translation rule* **TR1**, *the fact* a *is translated into*

$$a : (2, 0). \tag{1}$$

By the translation rule **TR6**, *the defeasible rule* **R1** *is translated into*

$$a : (1, 0) \wedge \sim c : (1, 0) \wedge \sim q : (0, 1) \rightarrow q : (1, 0). \tag{2}$$

By the translation rules **TR2** *and* **TR3**, *the strict rule* **R2** *is respectively translated into*

$$b_1 : (2,0) \wedge b_2 : (2,0) \rightarrow q : (0,2), \tag{3}$$

$$b_1 : (1,0) \wedge b_2 : (1,0) \wedge \sim q : (1,0) \rightarrow q : (0,1). \tag{4}$$

The defeater **R3** *does not have to be translated into any vasn-clause. Therefore, we have the VALPSN* $tr(T) = \{\, (1),\ (2),\ (3),\ (4)\, \}$.

The following example shows the relation in Figure 2 between the two kinds of provability of the defeasible logic and VALPSN stable model satisfiability.

Example 4.2. *Let a defeasible theory* $T = (F, \mathcal{C}, R, >)$, *the set of rules* $\mathbf{R} = \{\, \mathbf{R1}\{\, a\,\} \Rightarrow q,\ \mathbf{R2}\{\, b\,\} \Rightarrow \neg q,\ \mathbf{R3}\{\, c\,\} \Rightarrow q,\ \mathbf{R4}\{\, d\,\} \rightsquigarrow \neg q\,\}$, *and superiority relations,* $\mathbf{R1} > \mathbf{R2}$ *and* $\mathbf{R4} > \mathbf{R3}$. *By the translation rule* **TR5**, *the defeasible rule* **R1** *is translated into*

$$a : (1,0) \wedge \sim q : (0,1) \rightarrow q : (1,0). \tag{1}$$

By the translation rule **TR6**, *the defeasible rules* **R2** *and* **R3** *are respectively translated into*

$$b : (1,0) \wedge \sim a : (1,0) \wedge \sim q : (1,0) \rightarrow q : (0,1) \tag{2}$$

$$c : (1,0) \wedge \sim d : (1,0) \wedge \sim q : (0,1) \rightarrow q : (1,0). \tag{3}$$

The defeater **R4** *is not translated into vasn-clauses.*

1. *Suppose that* $F = \{a, b\}$. *Then, we have* $T \vdash +\partial q$ *and* $T \vdash -\partial\neg q$ *by the inference conditions* $+\partial.\mathbf{2.3.2}$, $-\partial.\mathbf{1}$ *and* $-\partial.\mathbf{2.3}$. *Then, the defeasible theory* T *is translated into a VALPSN* $tr(T) = \{\, a : (2,0),\ b : (2,0),\ (1), (2), (3)\,\}$, *which has only one stable model*

$$I = \{\, a : (2,0),\ \ b : (2,0),\ \ c : (0,0),\ \ d : (0,0),\ \ q : (1,0)\,\}.$$

Therefore, $I \models q : (1,0)$ *and* $I \not\models q : (0,1)$.

2. *Suppose that* $F = \{\, a, b, c, d,\ \}$. *Then, in the defeasible logic, we have* $T \vdash +\partial q$ *by the inference condition* $+\partial.\mathbf{2.3}$. *The defeasible theory* T *is translated into a VALPSN* $tr(T) = \{\, a : (2,0),\ b : (2,0),\ c : (2,0),\ d : (2,0),\ (1), (2), (3)\,\}$, *which has only one stable model*

$$I = \{\, a : (2,0),\ \ b : (2,0),\ \ c : (2,0),\ \ d : (2,0),\ \ q : (1,0)\,\}.$$

Therefore, $I \models q : (1,0)$.

3. *Suppose that* $F = \{\, a, b, c, d \,\}$ *and let* **R4** *be a defeasible rule* $\{d\} \Rightarrow \neg q$ *instead of the defeater* $\{\, d \,\} \rightsquigarrow \neg q$. *Then, we have* $T \vdash -\partial q$ *and* $T \vdash -\partial\neg q$ *by the inference condition* $-\partial.\mathbf{2.3}$. *By the translation rule* **TR5**, *the defeasible rule* **R4** *is translated into*

$$d\!:\!(1,0) \wedge \sim q\!:\!(0,1) \rightarrow q : (1,0). \tag{4}$$

Then, the defeasible theory T *is translated into a VALPSN* $tr(T) = \{\, a : (2,0),\ b : (2,0),\ c : (2,0),\ d : (2,0),\ (1),(2),(3),(4) \,\}$, *which has two stable models,*

$$\begin{aligned} I_1 &= \{\, a\!:\!(2,0),\ \ b\!:\!(2,0),\ \ c\!:\!(2,0),\ \ d\!:\!(2,0),\ \ q\!:\!(1,0) \,\},\\ I_2 &= \{\, a\!:\!(2,0),\ \ b\!:\!(2,0),\ \ c\!:\!(2,0),\ \ d\!:\!(2,0),\ \ q\!:\!(0,1) \,\}. \end{aligned}$$

Neither $q : (1,0)$ *nor* $q : (0,1)$ *can be satisfied by both the stable models* I_1 *and* I_2.

4. *Suppose that* $F = \{a, b\}$ *and there is no superiority relation between* **R1** *and* **R2**. *Then, we have* $T \vdash -\partial q$ *and* $T \vdash -\partial\neg q$ *by the inference condition* $-\partial.\mathbf{2.3}$. *By the translation rule* **TR5**, *the defeasible rules* **R1** *and* **R2** *are respectively translated into*

$$a\!:\!(1,0) \wedge \sim q\!:\!(0,1) \rightarrow q\!:\!(1,0) \tag{5}$$

$$b\!:\!(1,0) \wedge \sim q\!:\!(1,0) \rightarrow q\!:\!(0,1). \tag{6}$$

Then, the defeasible theory T *is translated into a VALPSN* $tr(T) = \{\, a : (2,0),\ b\!:\!(2,0),\ (3),(5),(6) \,\}$, *which has two stable models,*

$$\begin{aligned} I_1 &= \{\, a\!:\!(2,0),\ \ b\!:\!(2,0),\ \ c\!:\!(0,0),\ \ d\!:\!(0,0),\ \ q\!:\!(1,0) \,\},\\ I_2 &= \{\, a\!:\!(2,0),\ \ b\!:\!(2,0),\ \ c\!:\!(0,0),\ \ d\!:\!(0,0),\ \ q\!:\!(0,1) \,\}. \end{aligned}$$

Neither $q : (1,0)$ *nor* $q : (0,1)$ *can be satisfied by both the stable models* I_1 *and* I_2.

Let us take one more example. We describe how the defeasible theory in Example 3.1 is translated into a VALPSN.

EXAMPLE 4.3. *Let us remind Genetically Altered Penguin (GAP).*
F1 $gap(o)$ *is translated by the translation rule* **TR1** *into*

$$gap(o)\!:\!(2,0). \tag{1}$$

R1 *$gap(o) \to p(o)$ is translated by the translation rules* **TR2** *and* **TR3** *into*

$$gap(o):(2,0) \to p(o):(2,0), \tag{2}$$

$$gap(o):(1,0) \wedge \sim p(o):(0,1) \to p(o):(1,0). \tag{3}$$

R2 *$p(o) \to b(o)$ is translated by the translation rules* **TR2** *and* **TR3** *into*

$$p(o):(2,0) \to b(o):(2,0), \tag{4}$$

$$p(o):(1,0) \wedge \sim b(o):(0,1) \to b(o):(1,0). \tag{5}$$

R3 *$b(o) \Rightarrow f(o)$ is translated by the translation rule* **TR6** *into*

$$b(o):(1,0) \wedge \sim p(o):(1,0) \wedge \sim f(o):(0,1) \to f(o):(1,0). \tag{6}$$

R4 *$p(o) \Rightarrow \neg f(o)$ is translated by the translation rule* **TR6** *into*

$$p(o):(1,0) \wedge \sim gap(o):(1,0) \wedge \sim f(o):(1,0) \to f(o):(0,1). \tag{7}$$

Then, we have a VALPSN $P = \{(1), \ldots, (7)\}$ as the VALPSN-translation of GAP, which has only one stable model

$$I_1 = \{\, gap(o):(2,0),\ p(o):(2,0),\ b(o):(2,0),\ f(o):(0,0)\,\}.$$

Since $I_1 \not\models f(o) : (1,0)$ and $I_1 \not\models f(o) : (0,1)$, I_1 shows that $+\Delta\, gap(o)$, $+\Delta p(o)$, $+\Delta b(o)$ $-\partial f(o)$ and $-\partial \neg f(o)$.

4.2. The relation between defeasible theory and VALPSN

We provide proofs for the relation described in Figure 2, i.e., a literal q is definitely or defeasibly provable in a defeasible theory T *if and only if* for any stable model of the VALPSN $tr(T)$ satisfies vector annotated literals $q:(2,0)$ or $q:(1,0)$, respectively.

Let $T = (\,F, \mathcal{C}, R, >\,)$ be a defeasible theory, $P = tr(T)$ the VALPSN-translation of the defeasible theory T, P^I the Gelfond-Lifschitz transformation of the VALPSN P based on an interpretation I, q a literal, and M_P the set of all stable models of the VALPSN P. First, we define an interpretation $T_{P^I} \uparrow 0$ to be a special interpretation that assigns the truth value $(0,0)$ to all members of B_{P^I}, and an interpretation $T_{P^I} \uparrow 1$ to be a special interpretation that assigns the truth value $(0,0)$ to all members of $B_{P^I} \setminus \{q\}$ and the truth value $(2,0)$ to any literal $q \in F$. Moreover, let

$$T_{P^I} \uparrow n = T_{P^I}(T_{P^I} \uparrow (n-1))$$

for any integer $n \geq 2$ (refer to Definition 2.9 of T_{P^I}). Then, we have the following theorems.

THEOREM 4.1. $T \vdash +\Delta q \iff \forall I \in M_P; T_{P^I} \uparrow \omega \models q:(2,0)$.

PROOF. By induction on the number of lines in a proof of $+\Delta q$ and an integer i $(i \geq 1)$ such that $T_{P^I} \uparrow i \models q:(2,0)$.

Basis. ($\Rightarrow$ part) Suppose that $\mathrm{P}(1) = +\Delta q$. From the inference condition $+\Delta.1$, there exists a fact $q \in F$. Then, there also exists a vasn-clause $q : (2,0) \in P$ as the translation of the fact q by the translation rule **TR1**. Moreover, for any stable model I of the VALPSN P, $q:(2,0) \in P^I$. Thus, by the definition of the interpretation $T_{P^I} \uparrow 1$,

$$T_{P^I} \uparrow 1 \models q:(2,0).$$

($\Leftarrow$ part) Conversely, suppose that for any stable model I of the VALPSN P, $T_{P^I} \uparrow 1 \models q:(2,0)$. Then, by the definition of $T_{P^I} \uparrow 1$, there exists a unit vasn-clause $q:(2,0) \in P$. Then, there exists a literal $q \in F$ as the inverse translation of the unit vasn-clause $q:(2,0)$. Thus, by the inference condition $+\Delta.\mathbf{1}$,

$$\mathrm{P}(1) = +\Delta q$$

Induction Hypothesis. There is an integer $i \geq 1$ such that, for any literal q, for any stable model I of the VALPSN P,

$$+\Delta q \in \ \mathrm{P}(1..i) \iff \forall I \in M_P; T_{P^I} \uparrow i \models q:(2,0),$$

Induction Step. ($\Rightarrow$ part) Suppose that $\mathrm{P}(i{+}1) = +\Delta q \quad (i > 1)$. We consider the following two cases.

Case 1. If $+\Delta q$ is derived by the inference condition $+\Delta.\mathbf{1}$, it has been proved in the Basis.

Case 2. If $+\Delta q$ is derived by the inference condition $+\Delta.\mathbf{2}$, there exists a strict rule $\{a_1, \dots, a_k\} \to q \in R$ such that for each integer j $(1 \leq j \leq k)$, $+\Delta a_j \in \ \mathrm{P}(1..i)$. Then, there exists a vasn-clause $a_1:(2,0) \wedge \dots \wedge a_k:(2,0) \to q:(2,0) \in P$ as the translation of the strict rule $\{a_1, \dots, a_k\} \to q \in R$ by the translation rule **TR2**. By the Induction Hypothesis, for each integer j $(1 \leq j \leq k)$, $T_{P^I} \uparrow i \models a_j:(2,0)$. Thus, by the definition of $T_{P^I} \uparrow (i+1)$,

$$T_{P^I} \uparrow (i+1) \models q:(2,0).$$

($\Leftarrow$ part) Conversely, suppose that $T_{P^I} \uparrow (i+1) \models q:(2,0)$. Then, there exists a vasn-clause $a_1 : (2,0) \wedge \dots \wedge a_k : (2,0) \to q : (2,0) \in P^I$ as the translation of a strict rule $\{a_1, \dots, a_k\} \to q \in R$, and for each integer j $(1 \leq j \leq k)$, $T_{P^I} \uparrow i \models a_j : (2,0)$. Then, by the Induction Hypothesis, for each j $(1 \leq j \leq k)$, $+\Delta a_j$. Thus, we have $\mathrm{P}(i{+}1) = +\Delta q$. □

Theorem 4.2. $T \vdash -\Delta q \iff \exists I \in M_P; T_{PI} \uparrow \omega \models \sim q\colon(2,0)$.

Proof. First of all, we refer to Theorem 3.3 in Billington [2]: *Let $T_1 = (F, \mathcal{C}, R, >)$ be a defeasible theory. Suppose $d \in \{\Delta, \partial\}$. If q is a literal, then it is not the case that both $T_1 \vdash +dq$ and $T_1 \vdash -dq$.*

($\Rightarrow$ part) Suppose that $T \vdash -\Delta q$. From Theorem 3.3, we have $T \nvdash +\Delta q$ and from Theorem 4.1, we have $T \nvdash +\Delta q \iff \exists I \in M_P; T_{PI} \uparrow \omega \not\models q\colon(2,0)$. Thus, we have there exists a stable model I such that $T_{PI} \uparrow \omega \not\models q\colon(2,0)$.

($\Leftarrow$ part) Conversely, suppose that there exists a stable model I of the VALPSN P such that $T_{PI} \uparrow \omega \not\models q : (2,0)$. Then, we can trace back easily the above ($\Rightarrow$ part) proof to the conclusion $T \vdash -\Delta q$. □

Theorem 4.3. $T \vdash +\partial q \iff \forall I \in M_P; T_{PI} \uparrow \omega \models q\colon(1,0)$.

Proof. By induction on the number of lines in a proof of $+\partial q$ and an integer $i(i \geq 1)$ such that $T_{PI} \uparrow i \models q\colon(1,0)$.

Basis. ($\Rightarrow$ part) It is apparent that there can not exist $\mathrm{P}(1) = +\partial q$ by the inference conditions $+\partial.\mathbf{1}$ and $+\partial.\mathbf{2}$. Therefore, we suppose that $\mathrm{P}(2) = +\partial q$. Then, only the inference condition $+\partial.\mathbf{1}$ can be applied to the derivation of $+\partial q$ and there exists $+\Delta q \in \mathrm{P}(1)$. From the Basis in Theorem 4.1, for any stable model I of the VALPSN P, $T_{PI} \uparrow 1 \models q\colon(2,0)$. Since $(1,0) \preceq (2,0)$, $T_{PI} \uparrow 1 \models q\colon(2,0)$ implies $T_{PI} \uparrow 2 \models q\colon(1,0)$. Thus, we have

$$T_{PI} \uparrow 2 \models q\colon(1,0).$$

($\Leftarrow$ part) Conversely, suppose that for any interpretation I that can be a stable model of the VALPSN P, $T_{PI} \uparrow 2 \models q\colon(1,0)$. Then, by the definition of $T_{PI} \uparrow 1$ and the translation rule **TR1**, there exists a unit vasn-clause $q\colon(2,0) \in P^I$. Therefore, $q \in F$, $\mathrm{P}(1) = +\Delta q$ and $\mathrm{P}(2) = +\partial q$. Hence,

$$\mathrm{P}(2) = +\partial q \iff T_{PI} \uparrow 1 \models q\colon(1,0).$$

Induction Hypothesis. There is an integer $i \geq 2$ such that, for any literal q,

$$+\partial q \in \mathrm{P}(1..i) \iff T_{PI} \uparrow i \models q\colon(1,0).$$

Induction Step. We have to take the following three cases Case 1–3 into account to prove this induction step with respect to the inference conditions $+\partial$ and the translation rules **TR1**–**6**. Suppose that $\mathrm{P}(i{+}1){=}{+}\partial q$.

Case 1. In this case, $+\Delta q \in \mathrm{P}(1..i)$ and $\mathrm{P}(i{+}1) = +\partial q$. Then, by Theorem 4.1, for some integer m $(1 \leq m \leq i)$, $T_{PI} \uparrow m \models q\colon(2,0)$, and it implies,

$T_{P^I} \uparrow (i+1) \models q\!:\!(2,0)$ and $T_{P^I} \uparrow (i+1) \models q\!:\!(1,0)$, by the monotonicity of the operator T_{P^I}.

Case 2. In this case, there are a rule $\boldsymbol{A} \circ q$ that has no superiority relation; or a superiority relation $\boldsymbol{A} \circ q > \boldsymbol{B} \bullet \bar{q}$ and no defeater superior to the rule $\boldsymbol{A} \circ q$, where $\circ, \bullet \in \{\rightarrow, \Rightarrow\}$. Suppose that $+\partial q$ is derived based on the rule $\boldsymbol{A} \circ q$, where $\boldsymbol{A} = \{a_1, \dots, a_k\}$. Then, the antecedent $\boldsymbol{A}$ is defeasibly provable and $+\partial\bar{q} \notin \mathrm{P}(1..i)$. Therefore, by the translation rules **TR3** or **TR5**, there is a vasn-clause clause

$$a_1\!:\!(1,0) \wedge \ldots \wedge a_k\!:\!(1,0) \wedge {\sim}q\!:\!(0,1) \rightarrow q\!:\!(1,0) \in P.$$

Moreover, there is a vasn-clause

$$a_1\!:\!(1,0) \wedge \ldots \wedge a_k\!:\!(1,0) \rightarrow q\!:\!(1,0) \in P^I$$

such that for each j $(1 \leq j \leq k)$, $T_{P^I} \uparrow i \models a_j\!:\!(1,0)$ and $T_{P^I} \uparrow i \not\models q\!:\!(0,1)$ by the Induction Hypothesis. Thus, by the definition of the operator T_{P^I}, we have

$$T_{P^I} \uparrow (i+1) \models q\!:\!(1,0)$$

Case 3. In this case, there is a superiority relation $\boldsymbol{B} \bullet \bar{q} > \boldsymbol{A} \circ q$, where $\circ, \bullet \in \{\rightarrow, \Rightarrow\}$; $\boldsymbol{A} = \{a_1, \dots, a_k\}$ and $\boldsymbol{B} = \{b_1, \dots, b_l\}$. Moreover, $+\partial q$ is derived based on the rule $\boldsymbol{A} \circ q$. Then, the antecedent $\boldsymbol{A}$ is defeasibly provable and the literal $\bar{q}$ cannot be derived by other rules, i.e., the literal $\bar{q}$ is neither definitely nor defeasibly provable and the antecedent $\boldsymbol{B}$ is not defeasibly provable. Therefore, by the translation rules **TR4** or **TR6**, there are vasn-clauses

$$\begin{gathered}
a_1\!:\!(1,0) \wedge \ldots \wedge a_k\!:\!(1,0) \wedge \sim b_1\!:\!(1,0) \wedge {\sim}q\!:\!(0,1) \rightarrow q\!:\!(1,0), \\
a_1\!:\!(1,0) \wedge \ldots \wedge a_k\!:\!(1,0) \wedge \sim b_2\!:\!(1,0) \wedge {\sim}q\!:\!(0,1) \rightarrow q\!:\!(1,0), \\
\vdots \\
a_1\!:\!(1,0) \wedge \ldots \wedge a_k\!:\!(1,0) \wedge \sim b_l\!:\!(1,0) \wedge {\sim}q\!:\!(0,1) \rightarrow q\!:\!(1,0)
\end{gathered}$$

in the VALPSN P. Moreover, there exists a vasn-clause

$$a_1\!:\!(1,0) \wedge \ldots \wedge a_k\!:\!(1,0) \rightarrow q\!:\!(1,0) \in P^I$$

such that for each j $(1 \leq j \leq k)$, $T_{P^I} \uparrow i \models a_j : (1,0)$, for some integer $m \in \{1, \dots, l\}$, $T_{P^I} \uparrow i \not\models b_m\!:\!(1,0)$, and $T_{P^I} \uparrow i \not\models q\!:\!(0,1)$ by the Induction Hypothesis. Thus, by the definition of the operator T_{P^I}, we have

$$T_{P^I} \uparrow (i+1) \models q\!:\!(1,0)$$

Conversely, suppose that for any stable model I of the VALPSN P,

$$T_{P^I} \uparrow (i+1) \models q\!:\!(1,0).$$

We consider the following three cases Case 4–6.

Case 4. In this case, $T_{P^I} \uparrow i \models q : (2,0)$. Then, by Theorem 4.1, $+\Delta q \in \mathrm{P}(1..i)$. Therefore, by the inference condition $+\partial.\mathbf{1}$, we have

$$\mathrm{P}(i{+}1) = +\partial q.$$

Case 5. In this case, the vector annotated literal $q\!:\!(1,0)$ is the head of a vasn-clause that is translated by the translation rules **TR3** or **TR5**. Then, there is a vasn-clause

$$a_1\!:\!(1,0) \wedge \cdots \wedge a_k\!:\!(1,0) \rightarrow q\!:\!(1,0) \in P^I$$

such that for each integer $j(1 \leq j \leq k)$, $T_{P^I} \uparrow i \models a_j : (1,0)$, and $T_{P^I} \uparrow i \not\models q\!:\!(0,1)$. Then, by the Induction Hypothesis, for each integer j $(1 \leq j \leq k)$, $+\partial a_j \in \mathrm{P}(1..i)$, and $-\partial\neg q \in \mathrm{P}(1..i)$. Therefore, by the inference condition $+\partial.\mathbf{2}$,

$$\mathrm{P}(i{+}1) = +\partial q.$$

Case 6. In this case, the vector annotated literal $q\!:\!(1,0)$ is the head of a vasn-clause that is translated by the translation rules **TR4** or **TR6**. Then, there is a vasn-clause

$$a_1\!:\!(1,0) \wedge \cdots \wedge a_k\!:\!(1,0) \rightarrow q\!:\!(1,0) \in P^I$$

such that for each integer $j(1 \leq j \leq k)$, $T_{P^I} \uparrow i \models a_j : (1,0)$, and $T_{P^I} \uparrow i \not\models q\!:\!(0,1)$. Then, by the Induction Hypothesis, for each integer j $(1 \leq j \leq k)$, $+\partial a_j \in \mathrm{P}(1..i)$ and $-\partial\neg q \in \mathrm{P}(1..i)$. Therefore, by the inference condition $+\partial.\mathbf{2}$,

$$\mathrm{P}(i{+}1) = +\partial q. \qquad \square$$

Theorem 4.4. $T \vdash -\partial q \iff T_{P^I} \uparrow \omega \models \sim q : (1,0)$.

Proof. Suppose that $T \vdash -\partial q$. From Theorem 3.3, we have $T \nvdash +\partial q$. From Theorem 4.3, we also have $T \nvdash +\Delta q \longleftrightarrow T_{P^I} \uparrow \omega \not\models q : (1,0)$. Thus,

$$T_{P^I} \uparrow \omega \not\models q : (1,0), \quad (\text{i.e., } T_{P^I} \uparrow \omega \models \sim q : (1,0)\ ).$$

Conversely, suppose that $T_{P^I} \uparrow \omega \not\models q : (1,0)$. We can trace the above proof back to $T \vdash -\partial q$. □

5. Conclusion

In this paper, we have proposed VALPSN that can deal with defeasible reasoning. We have clarified the relation between VALPSN and Billington's defeasible logic. First, we proposed the translation from the defeasible logic into VALPSN, and proved that there is a correspondence between the defeasible logic provability and the VALPSN stable model satisfiability based on the translation. This correspondence shows that the defeasible logic derivation can be replaced by the stable model computation of the corresponding VALPSN, and that VALPSN can provide a semantics for the defeasible logic. However, if we replace the defeasible theory derivation by the VALPSN stable model computation, some problems in terms of computational complexity arise. Since each defeasible or strict rule in the defeasible logics are translated into more than one VALPSN clauses, there may be a case which has too many VALPSN clauses in the stable model computation. Basically, the stable model computation takes so long time when the VALPSN contains many clauses, even if it is implemented in a powerful workstation. If we want to implement a VALPSN stable model computing system, some strategies are required to speed up the computation.

References

[1] Billington,D., "Defeasible logic is stable", *J. Logic and Computation* 3, 379–400, 1993.

[2] Billington, D., "Conflicting literals and defeasible logic", *Proc. 2nd Australian Workshop on Commonsense Reasoning*, pp. 1–14, 1997.

[3] Blair, H. A., and V. S. Subrahmanian, "Paraconsistent logic programming", *Theoretical Computer Science* 68, 135–154, 1989.

[4] da Costa, N. C. A., V. S. Subrahmanian, and C. Vago, "The paraconsistent logics P$\mathcal{T}$", *Zeitschrift für Mathematische Logik und Grundlangen der Mathematik* 37, 139–148, 1989.

[5] Gelfond, M., and V. Lifschitz, "The stable model semantics for logic programming", *Proc. 5th Int'l Conf. and Symposium on Logic Programming*, pp. 1070–1080, 1988.

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

[7] Nakamatsu, K., and A. Suzuki, "Annotated semantics for default reasoning", *Proc. 3rd Pacific Rim Int'l Conf. on AI*, pp. 180–186, 1994.

[8] Nakamatsu, K. and A. Suzuki, "A nonmonotonic ATMS based on annotated logic programs", *Agents and Multi-Agents*, LNAI 1441, pp. 79–93, Springer-Verlag, 1998.

[9] Nute, D., "Defeasible reasoning", *Proc. Hawaii Int'l Conf. on System Science*, pp. 470–477, 1987.

[10] Nute, D., "Basic defeasible logic", *Intensional Logics for Programming*, pp. 125–154, Oxford Univ. Press, 1992.

[11] Subrahmanian, V. S., "On the semantics of qualitative logic programs", *Proc. 4th IEEE Symposium on Logic Programming*, pp. 178–182, 1987.

Kazumi Nakamatsu
School of Humanity for Environment Policy and Technology
Himeji Institute of Technology
Shinzaike 1-1-12, HIMEJI 670-0092, Japan
nakamatu@hept.himeji-tech.ac.jp