PAPER

A Combination of SLDNF Resolution with Narrowing for General Logic Programs with Equations with Respect to Extended Well-Founded Model

Susumu YAMASAKI†, *Member and* **Kazunori IRIYA**†, *Nonmember*

SUMMARY Negation as failure is realized to be combined with SLD resolution for general logic programs, where the combined resolution is called an SLDNF resolution. In this paper, we introduce narrowing and infinite failure to SLDNF resolution for general logic programs with equations. The combination of SLDNF resolution with narrowing and infinite failure is called an SLDNFN resolution. In Shepherdson (1992), equation theory is combined with SLDNF resolution so that the soundness may be guaranteed with respect to Clark's completion. Generalizing the method of Yamamoto (1987) for definite clause sets with equations, we formally define a least fixpoint semantics, which is an extension of Fitting (1985) and Kunen (1987) semantics, and which includes the pair of success and failure sets defined by the SLDNFN resolution. The relationship between the fixpoint semantics and the pair of sets is regarded as an extension of the relationships for general logic programs as in Marriott and et al. (1992) and in Yamasaki (1996). Instead of generalizing Clark's completion for SLDNFN resolution, we establish, as a model for general logic programs with equations, an extended well-founded model so that the SLDNFN resolution is sound and complete for non-floundering queries with respect to the extended well-founded model.

key words: *SLDNF resolution with infinite failure, narrowing, well-founded model*

1. Introduction

Although the function symbols of logic programs are usually regarded as just constructors for the terms, many theories, in which they are endowed with semantics for functions, have been presented [5], [14]. [15] treats logic programmimng endowed with equality. In [26], narrowing is combined with SLD resolution for a definite clause set with equations, and an integration of logic and equational aspects is discussed from procedural and declarative semantics views. In this paper, we are concerned with the combination of narrowing and SLDNF resolution (SLD resolution with negation as failure [18]) for general logic programs with equations. A general logic program consists of a clause containing negations in its body. Negation as failure is applied to SLD resolution based on the closed world assumption, and the combination of SLD resolution with negation as failure is SLDNF resolution, by which $\leftarrow \neg A$ succeeds if $\leftarrow A$ finitely fails for a ground atom A. The SLDNF resolution is sound with respect to the logical consequence of Clark's completion. (See [18], for example.) In [2], a class, for which the SLDNF resolution is complete, is presented, while a sound and complete semantics construction for the SLDNF resolution is given in [23]. In [24], SLDNF resolution with equality is formulated so that its soundness with respect to the Clark's completion with unification complete equation theory. Completeness of SLDNF resolution with equality is discussed for hierarchical and allowed normal programs in [25]. In these contexts, we have Clark's completion with unification, which the equational SLDNF resolution is examined to be correct with respect to.

The Clark's completion, however, does not always fit the semantics for general logic programs with equations, because of difficulty of equation theory. As regards the declarative semantics for general logic programs, the stable model is well-defined based on the 2-valued logic approach [13], [19]. On the other hand, a fixpoint semantics as in [10], [20] as well as its generalization [27], and the well-founded model as in [1], [11], [12] are discussed in the 3-valued logic approach. The well-founded model is also the semantics for SLDNF resolution with infinite failure, as long as a query is non-floundering [3], [22].

Following the extended semantics for definite clause sets with equations as in [26], and the relationship between the well-founded model and the SLDNF resolution with infinite failure, we pay attention to a combination of narrowing and SLDNF resolution with infinite failure for non-floundering queries (with respect to an extended well-founded model), which is referred to as SLDNFN resolution (SLDNF resolution with infinite failure and Narrowing). We discuss the semantics for general logic programs with equations in relation to the finite SLDNFN resolution. In general, because of narrowing, a monotonic operator associated with general logic programs with equations may not have a consistent least fixpoint, while a nonmonotonic operator is to be prepared for the consistency of the semantics. In this paper, we take a class of programs whose semantics is consistent, given by a monotonic operator. We show that the success and failure sets pair is proven to be included in a fixpoint semantics,

Manuscript received January 14, 1998.
Manuscript revised June 6, 1999.

†The authors are with the Department of Computing and Information Technology, Okayama University, Okayama-shi, 700–8530 Japan.

which can be an extension of the fixpoint semantics for general logic programs without equations by [10], [20], that is, Fitting (1985) and Kunen (1987) semantics. We also show that the SLDNFN resolution is sound with respect to an extended well-founded model for the general logic program with equations, as long as the extended well-founded model exists. The completeness of SLDNFN resolution is guaranteed for non-floundering queries with respect to the extended well-founded model.

The paper is organized as follows. In Sect. 2, we have terminologies regarding general logic programs with equations as well as the notations of term rewriting systems. In Sect. 3, narrowing is introduced to SLDNF resolution, and the denotations of success and failure sets are related with fixpoint semantics for general logic programs with equations, an extension of Fitting and Kunen semantics. In this case, infinite failure is not taken. Nor Clark's completion is examined. This section is concerned with just soundness of SLDNF resolution with narrowing and without infinite failure. In Sect. 4, SLDNFN resolution is treated as a sound and complete procedure for non-floundering queries with respect to the extended well-founded model.

2. Preliminaries

We deal with a general logic program with equations, based on the terminologies in [8], [18].

2.1 General Logic Program with Equations

A general logic program (equivalently, a normal program) is the set of clauses of the form $A \leftarrow L_1 \dots L_n$, where each L_i is a literal, that is, either a positive literal (an atom A_i) or a negative literal (its negation $\neg A_i$). A in the clause is referred to as its head, while $L_1 \dots L_n$ as its body. A normal goal is a clause of the form $\leftarrow M_1 \dots M_m$, where each M_i is either an atom B_i or its negation $\neg B_i$. A normal goal containing only positive literals is said just a goal. We call a normal goal containing no literal, the empty clause (denoted $\Box$).

The clause, the literal, the atom and the term containing no variables are referred to as the ground clause, the ground literal, the ground atom and the ground term, respectively. The clause is transformed to a homogeneous form so that the narrowing is applied to only the body of the clause [8].

The homogeneous form of the definite clause $q(t_1, \dots, t_m) \leftarrow B_1 \dots B_n$ is

$$q(x_1, \dots, x_m) \leftarrow eq(x_1, t_1) \dots eq(x_m, t_m) B_1 \dots B_n,$$

where $x_1, \dots, x_m$ are m different variables not to occur in the original clause, and $eq(x_i, t_i) \leftarrow$ is an equation as described below.

With the predicate symbol eq, the equation is a clause of the form $eq(t, s) \leftarrow$, where t and s are terms such that (i) all the variables occurring in s should occur in t, and (ii) t is not a variable, except the case that $t = s$ and t is a variable. A general logic program is supposed to involve the equation $eq(x, x) \leftarrow$ for a variable x, as an axiom. Let E_I be $\{eq(x, x) \leftarrow\}$. It is necessary to match the atom $eq(t, t)$ for any term t. Alternatively, a set of equality axioms Eq may be assumed in a given general logic program with equations, where

$$\begin{aligned} Eq = \{ & eq(x,x) \leftarrow, \\ & \quad eq(x,y) \leftarrow eq(y,x), \\ & \quad eq(x,z) \leftarrow eq(x,y) eq(y,z) \qquad \} \cup \\ \{ & eq(f(x_1,\dots,x_n), f(y_1,\dots,y_n)) \\ & \quad \leftarrow eq(x_1,y_1) \dots eq(x_n,y_n) \\ & \quad \mid f \text{ is a function symbol in } P \ \} \cup \\ \{ & eq(p(x_1,\dots,x_n), p(y_1,\dots,y_n)) \\ & \quad \leftarrow eq(x_1,y_1) \dots eq(x_n,y_n) \\ & \quad \mid p \text{ is a predicate symbol in } P \} \end{aligned}$$

for variables $x, y, z, x_1, y_1, \dots, x_n, y_n$.

Let P be a general logic program, where P' is the homogeneous form of P, that is, the set of homogeneous clauses transformed from clauses of P. Also let E be a set of equations. We call $Pr = P' \cup E \cup E_I$ to be a general logic program with equations (*glpe*, for short).

2.2 Term Rewriting and Narrowing

Before we see term rewriting and narrowing, we summarize the terminologies as to substitutions and their applications [27].

A substitution is a function from the set of variables Var to the set of terms $Term$, where the term is recursively defined: (i) a variable is a term, and (ii) if $t_1, \dots, t_n$ ($n \geq 0$) are terms and f is a n-place function symbol, then $f(t_1, \dots, t_n)$ is a term. A substitution φ may be expressed as $\{x_1 \mid t_1, \dots, x_n \mid t_n\}$, where $t_i = \varphi(x_i)$, $1 \leq i \leq n$, and $\varphi(x) = x$ if $x \neq x_i$ ($1 \leq i \leq n$). $Dom(\varphi)$ means $\{y \mid \varphi(y)$ is defined such that $\varphi(y) \neq y\}$. We assume that $Dom(\varphi)$ is finite for any substitution φ. Sub stands for the set of substitutions. The empty substitution is denoted by ε. That is, $\varepsilon(x) = x$ for any $x \in Var$. A substitution μ is said a permutation (a renaming of variables) if it is a bijection from Var to Var. For an expression F (say, a literal or a term) and a substitution θ, $F\theta$ denotes the expression obtained by substituting all the variables in F for terms according to θ, that is, by applying θ to the expression F.

For $\theta, \varphi \in Sub$, the composition of θ and φ (denoted by $\theta\varphi$) is defined by letting $(\theta\varphi)(x) = \theta(x)\varphi$ for $x \in Var$. It is easy to see $(\varphi\psi)\theta = \varphi(\psi\theta)$ for $\varphi, \psi, \theta \in Sub$. Note $\varepsilon\theta = \theta\varepsilon = \theta$. Also we see $(E\theta)\varphi = E(\theta\varphi)$ for an expression E and $\theta, \varphi \in Sub$. A relation

$\preceq$ on Sub is defined: $\theta \preceq \varphi$ iff there exists $\psi \in Sub$ such that $\varphi = \theta\psi$. θ is said to be more general than φ if $\theta \preceq \varphi$.

A relation $\sim$ on Sub is defined: $\theta \sim \varphi$ iff $\varphi \preceq \theta$ and $\theta \preceq \varphi$. It is seen that $\sim$ is an equivalence relation. Note that if $\theta \sim \varphi$ then there exist permutations (renamings of variables) ρ and σ such that $\theta\rho = \varphi$ and $\varphi\sigma = \theta$. (See [7].) For $\theta \in Sub$, let $[\theta]_\sim = \{\varphi \in Sub \mid \theta \sim \varphi\}$ and $Sub/\sim = \{[\theta]_\sim \mid \theta \in Sub\}$. It is the usual way that by letting $[\theta]_\sim \leq [\varphi]_\sim$ for $[\theta]_\sim, [\varphi]_\sim \in Sub/\sim$ if $\theta \preceq \varphi$, we have a partially ordered set $(Sub/\sim, \leq)$ [21].

Goal stands for the set of normal goals. $\ll$ is a relation on *Goal*, defined by letting

$$G_1 \ll G_2 \text{ if } \exists\sigma \in Sub\text{: } [G_2 = G_1\sigma].$$

We also denote by $F_1 \ll F_2$ for expressions F_1 and F_2 that $F_2 = F_1\sigma$ for some $\sigma \in Sub$. F_1 is said a variant of F_2 and vice versa if $F_1 \ll F_2$ and $F_2 \ll F_1$.

For $\theta \in Sub$, $\theta\mid_E$, a restriction of θ with respect to an expression E, is defined to be

$$\theta\mid_E(x) = \begin{cases} \theta(x) & \text{if } x \text{ occurs in } E, \\ x & \text{otherwise.} \end{cases}$$

The notions of unifiers and most general unifiers are used as the usual sense.

Definition 2.1: Let S be a set of terms, or a set of atoms. Let θ be a substitution. If $S\theta = \{s\theta \mid s \in S\}$ be a sigleton, then we say that θ is a unifier of S. If θ is a unifier of S, and $\theta \preceq \sigma$ for any unifier σ of S, then we say that θ is a most general unifier (mgu, for short).

Following [8], we now review the definitions of the reduction caused by the set of equations. Let F be an expression, N a term, and t be an occurrence of a term of F. (See [26] for the occurrence of the term.) $F[t \leftarrow N]$ denotes the expression obtained from F by substituting the term N for the occurrence of t.

A set E of equations is interpreted as a term rewriting system if each equation in E satisfies the condition given in Sect. 2.1. The reduction relation $\rightarrow_E$ associated with E is defined on the set of terms by: For all terms M and N,

$$M \rightarrow_E N,$$

iff there exist
(i) $eq(t,s) \leftarrow$ (an equation) in E,
(ii) an occurrence t' of a subterm of M, and
(iii) a substitution θ,
such that $t' = t\theta$ and $N = M[t' \leftarrow s\theta]$.

$\rightarrow^*_E$ is the reflexive and transitive closure of $\rightarrow_E$. $\rightarrow_E$ is terminating iff for all terms M, there is no infinite sequence $M, M_1, \ldots, M_n, \ldots$ such that $M \rightarrow_E M_1 \rightarrow_E \ldots \rightarrow_E M_n \rightarrow_E \ldots$. $\rightarrow_E$ is confluent iff for all terms M, N, N', $M \rightarrow^*_E N$ and $M \rightarrow^*_E N'$ imply $\exists M' : [N \rightarrow^*_E M'$ and $N' \rightarrow^*_E M']$. A term rewriting system E is said to be terminating and confluent iff $\rightarrow_E$ is terminating and confluent, respectively. A terminating and confluent term rewriting system is said to be canonical.

In this paper we take the following assumption:
Assumption: We deal with the canonical system (that is, the canonical set of equations).

In defining semantics over a domain based on the Herbrand base, the reduction relation $\rightarrow_E$ is used instead of narrowing. However, when we consider procedures based on resolutions with the equation set, we may use the narrowing relation $\Rightarrow_E$ based on the set of equations E as follows.

For all terms M and N,

$$M \Rightarrow_E N$$

iff there exist
(i) a variant $eq(t,s) \leftarrow$ of an equation in E such that it has no common variable with M,
(ii) an occurrence t' of a nonvariable subterm of M, and
(iii) a most general unifier θ of t' and t
such that $N = M\theta[t'\theta \leftarrow s\theta]$.

3. Combination of SLDNF Resolution and Narrowing

In this section, we formulate a combination of SLDNF resolution and narrowing for *glpe*s by generalizing the techniques for the case of logic programs with equations as in [8], [26].

3.1 Narrowing Introduced to SLDNF Resolution

We take a *glpe* $Pr = P' \cup E \cup E_I$, where P' is a homogeneous form of P, and $E_I = \{eq(x,x) \leftarrow\}$. Let $E(Pr) = E$, and $Prog(Pr) = P'$. Following the definitions regarding SLDNF resolutions in [18], we have a combination of SLDNF resolution and narrowing, as well as infinite failure, which is referred to as an SLDNFN resolution (SLDNF resolution with infinite failure and Narrowing).

Definition 3.1: Let G be a normal goal $\leftarrow L_1 \ldots L_m \ldots L_p$, and C be a clause $B \leftarrow B_1 \ldots B_q$ in $Prog(Pr) \cup E_I$. A normal goal G' is derived from G and C by using an mgu θ if
(a) L_m is an atom, called a selected atom, in G,
(b) θ is an mgu of L_m and B, and
(c) G' is the normal goal $\leftarrow L_1\theta \ldots L_{m-1}\theta\ B_1\theta \ldots B_q\theta\ L_{m+1}\theta \ldots L_p\theta$.

We recursively define an SLDNFN-refutation and a finitely failed SLDNFN-tree, by combining SLDNF resolutions and narrowings.

Definition 3.2: Let Pr be a *glpe* and G a normal goal. An SLDNFN-refutation of rank k ($k \geq 0$) of $Pr \cup \{G\}$ consists of a sequence $G \equiv G_0, G_1, \ldots, G_n \equiv$

$\square$ of normal goals, a sequence $C_1, \ldots, C_n$ of variants of clauses of Pr or ground negative literals, and a sequence $\theta_1, \ldots, \theta_n$ of substitutions, such that, for each i, either

(i) G_{i+1} is derived from G_i and C_{i+1} by using resolution with θ_{i+1}, or

(ii) $G_{i+1} \equiv G_i\theta_{i+1}[t_i\theta_{i+1} \leftarrow s\theta_{i+1}]$, where C_{i+1} in $E(Pr)$ takes a form $eq(t,s) \leftarrow$, and θ_{i+1} is an mgu of t and a term t_i occurring in G_i, or

(iii) G_i is $\leftarrow L_1 \ldots L_m L_{m+1} \ldots L_p$, where there is a failed SLDNFN-tree of rank $k-1$ for $Pr \cup \{\leftarrow A_m\}$, where $L_m \equiv \neg A_m$ (a ground literal). In this case, $G_{i+1} \equiv L_1 \ldots L_{m-1} L_m \ldots L_p$, θ_{i+1} is the identity substitution and $C_{i+1} \equiv \neg A_m$.

$(\ldots(\theta_1\theta_2)\ldots\theta_n)\mid_G$ is said a computed answer.

A failed SLDNFN-tree of rank k for $Pr \cup \{G\}$ is a tree satisfying the following conditions:

(a) Each node of the tree is a non-empty normal goal.

(b) The root node is G.

(c) Let $\leftarrow L_1 \ldots L_m \ldots L_p$ be a non-leaf node in the tree and suppose that L_m is selected. Then either

(i) L_m is an atom and, for a variant of each clause $A \leftarrow M_1 \ldots M_q$ in *Prog*(Pr) such that L_m and A are unifiable with an mgu θ, the node has a child

$$\leftarrow (L_1 \ldots L_{m-1} M_1 \ldots M_q L_{m+1} \ldots L_p)\theta,$$

or

(ii) L_m is a literal, where t_m is an occurrence in L_m of $\leftarrow L_1 \ldots L_m \ldots L_p$ such that t_m and t of $eq(t,s) \leftarrow \in E(Pr)$ are unifiable with an mgu θ, and the node has a child

$$\leftarrow L_1\theta \ldots L_m\theta[t_m\theta \leftarrow s\theta] \ldots L_p\theta,$$

or

(iii) L_m is a ground negative literal $\neg A_m$ and there is a failed SLDNFN-tree of rank $k-1$ of $Pr \cup \{\leftarrow A_m\}$ such that the node has only the child $\leftarrow L_1 \ldots L_{m-1} L_{m+1} \ldots L_p$.

(d) Let $\leftarrow L_1 \ldots L_m \ldots L_p$ be a leaf node in the tree and L_m is supposed to be selected. Then either

(i) L_m is an atom and there is no clause (nor its variant) in $P' \cup E_I$ whose head unifies with L_m, nor is there a term occurring in L_m for which narrowing is applied, or

(ii) L_m is a ground negative literal $\neg A_m$ and there is an SLDNFN-refutation of rank $k-1$ of $P \cup \{\leftarrow A_m\}$.

If the tree is finite, then the failed SLDNFN-tree is said a finitely failed SLDNFN-tree.

(Note that the refutation of rank 0 and the failed SLDNFN-tree of rank 0 can be defined by setting $k = 0$, where the cases for $k = -1$ are excluded.)

Definition 3.3: Let Pr be a *glpe* and G a normal goal. An SLDNFN-refutation of $Pr \cup \{G\}$ is an SLDNFN-refutation of rank k of $Pr \cup \{G\}$, for some $k \geq 0$. A failed SLDNFN-tree for $Pr \cup \{G\}$ is a failed SLDNFN-tree of rank k for $Pr \cup \{G\}$, for some $k \geq 0$.

By a computation of $Pr \cup \{G\}$, we mean an attempt to detect an SLDNFN refutation or a failed SLDNFN tree.

Definition 3.4: Let Pr be a *glpe* and G a normal goal. We say that a computation of $Pr \cup \{G\}$ flounders if at some point in the computation a goal is reached which contains only non-ground negative literals.

We now have a formal description of the depth of an SLDNFN-refutation and a finitely failed SLDNFN-tree. Assume that there is an SLDNFN-refutation of $Pr \cup \{G\}$ in which the sequence of normal goals is $G_0 \equiv G, G_1, \ldots, G_n \equiv \square$. The SLDNFN-refutation tree consists of the nodes G_i, $0 \leq i \leq n-1$, connected by the nodes G_{i+1}, where G_0 is a root.

Definition 3.5: Let *Tree* be the set of SLDNFN-refutation trees and finitely failed SLDNFN-trees. dp : *Tree* $\rightarrow \omega$ is defined recursively as follows:

(i) If tr consists of a root node, then $dp(tr) = 0$, and

(ii) If tr consists of a root node, connected by subtrees tr_i, $1 \leq i \leq p$, then $dp(tr) = 1 + max_{1 \leq i \leq p}\{dp(tr_i)\}$.

The depth of an SLDNFN-refutation is defined as that of the corresponding SLDNFN-refutation tree.

Example 3.6: Let

$$\begin{aligned} P = \{&C_1 : od(x_1) \leftarrow eq(x_1, s(0)),\\ &C_2 : ev(x_1) \leftarrow eq(x_1, s(s(0))),\\ &C_3 : od(x_1) \leftarrow eq(x_1, s(s(x)))od(x),\\ &C_4 : ev(x_1) \leftarrow eq(x_1, s(s(x)))ev(x),\\ &C_5 : nat(x_1) \leftarrow eq(x_1, 0),\\ &C_6 : nat(x_1) \leftarrow eq(x_1, s(x))nat(x),\\ &C_7 : ev(x_1) \leftarrow eq(x_1, x)nat(x)\neg od(x),\\ &C_8 : od(x_1) \leftarrow eq(x_1, x)nat(x)\neg ev(x)\},\\ E_I = \{&C_9 : eq(x, x) \leftarrow \qquad\qquad\qquad\qquad\}, \text{and} \end{aligned}$$

$E = \{C_{10} : eq(two\text{-}plus(x), s(s(x))) \leftarrow\}$. An SLDNFN-refutation of rank 2 of $P \cup E \cup E_I \cup \{\leftarrow od(two\text{-}plus(x))\}$ with a computed answer $\{x \mid s(0)\}$ is shown in Fig. 1. Note that $P \cup E_I \cup E$ is used for simplicity instead of $P' \cup E_I \cup E$. Without C_{10}, we have no SLDNFN-refutation of P.

We can have the following lemma as an extension modified from Lemmas 2.8 in [27].

Lemma 3.7: Let Pr be a *glpe* and $\leftarrow L_1 \ldots L_n$ a ground normal goal.

(1) If there is an SLDNFN-refutation of rank k of $Pr \cup \{\leftarrow L_1 \ldots L_n\}$ with depth d and a computed answer ε, then, for any $\{L_{i_1}, \ldots, L_{i_l}\} \subset \{L_1, \ldots, L_n\}$, there is an SLDNFN-refutation of rank k of $Pr \cup \{\leftarrow L_{i_1} \ldots L_{i_l}\}$ with depth $d' \leq d$ and a computed answer ε.

(2) If there is an finitely failed SLDNFN-tree of rank k for $Pr \cup \{\leftarrow L_1 \ldots L_n\}$ with depth d, then, there

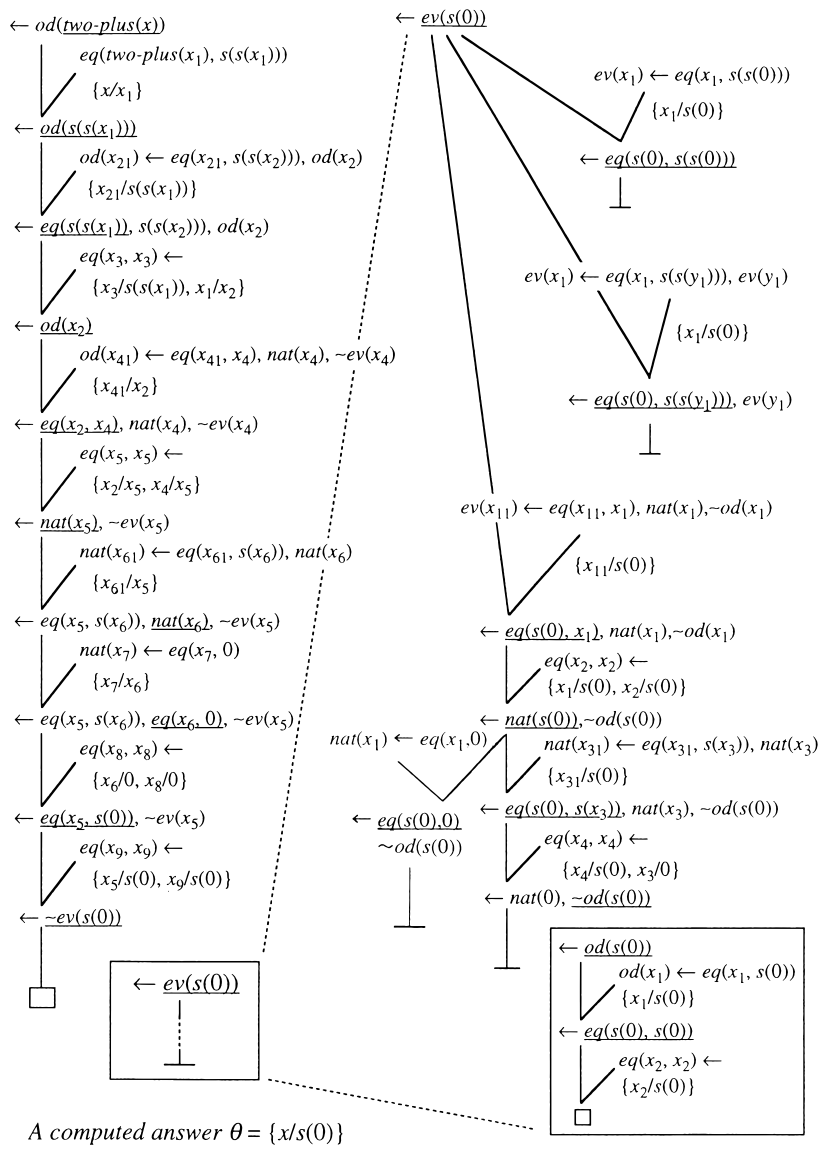

Fig. 1 SLDNFN-refutation of rank 2 of $Pr \cup \{\leftarrow od(two\text{-}plus(x))\}$.

is $\{L_{i_1}, \ldots, L_{i_l}\} \subset \{L_1, \ldots, L_n\}$ such that we have a finitely failed SLDNFN-tree of rank k for $Pr \cup \{\leftarrow L_{i_1}, \ldots, L_{i_l}\}$ with depth $d' \leq d$.

3.2 Success and Failure Sets by SLDNFN Resolutions in Relation to Fixpoint Semantics for *glpe*

We firstly generalize the semantics for general logic programs to that for *glpe*s, and next examine the denotations of success and failure sets by SLDNFN resolutions.

Combining the idea of [26] with the operator in [10], [20], we define an operator to give a fixpoint semantics for the *glpe* Pr. Let B_{Pr} be the Herbrand base of a *glpe* Pr.

Definition 3.8: Assume a *glpe* Pr, and a body $b \equiv L_1 \ldots L_n$ of a ground clause, obtained from a clause of Pr. Let $(u_s, u_f) \in 2^{B_{Pr}} \times 2^{B_{Pr}}$. We say that:

b is true in (u_s, u_f)
if $\forall i \in \{1, \ldots, n\}$, $\forall A \in B_{Pr}$:
$[(L_i = A \Rightarrow A \in u_s) \wedge (L_i = \neg A \Rightarrow A \in u_f)]$,
and b is false in (u_s, u_f)
if $\exists i \in \{1, \ldots, n\}$, $\exists A \in B_{Pr}$:
$[(L_i = A \wedge A \in u_f) \vee (L_i = \neg A \wedge A \in u_s)]$.

Definition 3.9: Given a *glpe* Pr, we define S_{Pr}: $2^{B_{Pr}} \times 2^{B_{Pr}} \rightarrow 2^{B_{Pr}} \times 2^{B_{Pr}}$ to be

$$S_{Pr}(u_s, u_f) = (u'_s, u'_f),$$

where

$$u'_s = \{A \in B_{Pr} \mid \exists (A \leftarrow b\text{: a ground clause obtained from a clause of } Pr)\text{:}$$

b is true in $(u_s, u_f)\}$
$\cup \{p(t_1, \ldots, t_n) \in B_{Pr} \mid p(t_1, \ldots, s_i, \ldots, t_n) \in u_s \wedge (t_i \rightarrow_E s_i)\}$,
$u'_f = \{A \in B_{Pr} \mid \forall(A \leftarrow b$: a ground clause obtained from a clause of Pr): b is false in $(u_s, u_f)\}$
$\cup \{p(t_1, \ldots, t_n) \in B_{Pr} \mid p(t_1, \ldots, s_i, \ldots, t_n) \in u_f \wedge (t_i \rightarrow_E s_i) \wedge \forall j$: $((t_j \rightarrow_E t'_j) \Rightarrow p(t_1, \ldots, t'_j, \ldots, t_n) \in u_f)\}$.

Since S_{Pr} is monotonic, there exists a least fixpoint of S_{Pr}, expressed as $lfp(S_{Pr})$, which is eqaul to $S_{Pr} \uparrow \beta$ for some β, where $S_{Pr} \uparrow \alpha$ is inductively defined:

$$S_{Pr} \uparrow 0 = (\emptyset, \emptyset)$$
$$S_{Pr} \uparrow \alpha = S_{Pr}(S_{Pr} \uparrow (\alpha - 1)) \text{ if } \alpha \text{ is a successor ordinal,}$$
$$S_{Pr} \uparrow \alpha = \cup\{S_{Pr} \uparrow \beta \mid \beta < \alpha\} \text{ if } \alpha \text{ is a limit ordinal.}$$

Definition 3.10: We define *cons*: $2^{B_{Pr}} \times 2^{B_{Pr}} \rightarrow \{true, false\}$ to be

$$cons(u_s, u_f) = true \text{ iff } u_s \cap u_f = \emptyset.$$

We say that (u_s, u_f) is consistent if $cons(u_s, u_f) = true$.

As we see by the following example, the least fixpoint is not always consistent.

Example 3.11: Let

$$Pr = \{p(b) \leftarrow p(b), p(a) \leftarrow, eq(a, b) \leftarrow, eq(x, x) \leftarrow\}.$$

Then

$$(\{p(a)\}, \{p(b)\}) \subset S_{Pr} \uparrow 1,$$
$$(\{p(a)\}, \{p(a), p(b)\}) \subset S_{Pr} \uparrow 2,$$

so that $S_{Pr} \uparrow 2$, which is included in the least fixpoint of S_{Pr}, is not consistent.

The operator Σ_{Pr}: $2^{B_{Pr}} \times 2^{B_{Pr}} \rightarrow 2^{B_{Pr}} \times 2^{B_{Pr}}$, for example, is nonmonotonic, but preserves the consistency.

$$\Sigma_{Pr}(v_s, v_f) = (v'_s, v'_f),$$

where

$v'_s = \{A \in B_{Pr} \mid \exists(A \leftarrow b$: a ground clause obtained from a clause of Pr): b is true in $(v_s, v_f)\}$
$\cup \{p(t_1, \ldots, t_n) \in B_{Pr} \mid p(t_1, \ldots, s_i, \ldots, t_n) \in v_s \wedge (t_i \rightarrow_E s_i) \wedge p(t_1, \ldots, t_n) \notin v'_f\}$,
$v'_f = \{A \in B_{Pr} \mid \forall(A \leftarrow b$: a ground clause obtained from a clause of Pr): b is false in $(v_s, v_f)\}$.

Note that v'_f is determined by means of (v_s, v_f). To see the relation between the model and the SLDNFN resolution, we take the *glpe* Pr for which $lfp(S_{Pr})$ is consistent.

Definition 3.12: Let Pr be a *glpe*. We define

$SS^k_{Pr} = \{(A\theta)\sigma \in B_{Pr} \mid \sigma \in Sub$, and there is an SLDNFN-refutation of rank k of $Pr \cup \{\leftarrow A\}$ with a computed answer θ, and possibly with finitely failed SLDNFN-trees$\}$,
$FS^k_{Pr} = \{A\sigma \in B_{Pr} \mid \sigma \in Sub$, and there is a finitely failed SLDNFN-tree of rank k for $Pr \cup \{\leftarrow A\}\}$,
$SS_{Pr} = \cup_{k\in\omega} SS^k_{Pr}$,
$FS_{Pr} = \cup_{k\in\omega} FS^k_{Pr}$.

The sets SS_{Pr} and FS_{Pr} are referred to as the success and finite failure sets. In the definition of the success set, we mean by the phrase “possibly, with finitely failed SLDNFN-trees” that the SLDNFN-refutation may involve finitely failed SLDNFN-trees of lower ranks, but not any infinitely failed SLDNFN-trees.

We now have a primary result on the relationship between the success and finite failure sets, and the least fixpoint of S_{Pr}. It can be regarded as an extension of the equivalence of the success set with the least fixpoint semantics for the definite clause set (see [18]), and of the relationship between the success and finite failure sets, and the least fixpoint semantics for the general logic program as in [20], [27].

Theorem 3.13: Given a *glpe* Pr, $(SS_{Pr}, FS_{Pr}) \subset lfp(S_{Pr})$.

Proof: See the Appendix. q.e.d.

Note that even if $lfp(S_{Pr})$ is inconsistent, the above theorem holds. However, we think of it as a soundness theorem for a *glpe* Pr such that $lfp(S_{Pr})$ is consistent. Unless the narrowing is applied, the logical consequence of the Clark’s completion of Pr can be contained within (SS_{Pr}, FS_{Pr}), as shown in [17]. Because it is not so easy to make the descriptions of the completion with equation theory simple, we rather take the semantics for SLDNFN resolution, by extending the well-founded model.

4. SLDNFN Resolution with Respect to Extended Well-Founded Model

We present an extended well-founded model for SLDNFN resolution. To get sound and complete procedure for non-floundering query with respect to the extended well-founded model, we need infinite failure which is not caused by infinite repetitions of narrowing but by infinite applications of resolutions and/or narrowings.

Firstly, we take success and failure sets containg infinite failure.

Definition 4.1: Let Pr be a *glpe*. We define

$$\begin{aligned}
SucS^k_{Pr} = \{(A\theta)\sigma \in B_{Pr} \mid \sigma \in Sub, &\text{ and there is an SLDNFN-refutation of rank } k \text{ of } Pr \cup \{\leftarrow A\} \text{ with a computed answer } \theta\},\\
FailS^k_{Pr} = \{A\sigma \in B_{Pr} \mid \sigma \in Sub, &\text{ and there is a failed SLDNFN-tree of rank } k \text{ of } Pr \cup \{\leftarrow A\}\},\\
SucS_{Pr} &= \cup_{k\in\omega} SucS^k_{Pr},\\
FailS_{Pr} &= \cup_{k\in\omega} FailS^k_{Pr}.
\end{aligned}$$

Note that $SS_{Pr} \subset SucS_{Pr}$, and $FS_{Pr} \subset FailS_{Pr}$.

As in [6], [11], we have the notion of unfounded sets, which are related with infinite failure.

Definition 4.2: A set S of ground atoms is an unfounded set of Pr, with respect to $(u_s, u_f) \in B_{Pr} \times B_{Pr}$ such that $u_s \cap u_f = \emptyset$, i.e., $cons(u_s, u_f) = true$, if each $A \in S$ satisfies the following condition: For each ground clause C from Pr whose head is A, at least one of the following holds:

(1) The body of C is false in (u_s, u_f).

(2) Some positive literal of the body of C occurs in S.

The greatest unfounded set of Pr with respect to (u_s, u_f) is denoted by $GU(u_s, u_f)$.

Note that the union of unfounded sets is also an unfounded set. It follows that there is always a greatest unfounded set of Pr.

Definition 4.3: Let Pr be a *glpe*. We define W_{Pr}: $2^{B_{Pr}} \times 2^{B_{Pr}} \to 2^{B_{Pr}} \times 2^{B_{Pr}}$ to be

$$W_{Pr}(u_s, u_f) = (u''_s, u''_f),$$

where

$$\begin{aligned}
u''_s = \{&A \in B_{Pr} \mid \exists A \leftarrow b \text{ (a ground clause obtained from a clause of } Pr\text{):}\\
&b \text{ is true in } (u_s, u_f)\}\\
&\cup \{p(t_1, \dots, t_n) \in B_{Pr} \mid p(t_1, \dots, s_i, \dots, t_n) \in u_s \wedge (t_i \to_E s_i)\},\\
u''_f = {}&GU(u_s, u_f) \cup \{p(t_1, \dots, t_n) \in B_{Pr} \mid p(t_1, \dots, s_i, \dots, t_n) \in u_f \wedge (t_i \to_E s_i)\\
&\wedge \forall j\text{: } ((t_j \to_E t'_j) \Rightarrow p(t_1, \dots, t'_j, \dots, t_n) \in u_f)\}.
\end{aligned}$$

$W_{Pr} \uparrow \alpha$ is inductively defined:

$$\begin{aligned}
&W_{Pr} \uparrow 0 = (\emptyset, \emptyset),\\
&W_{Pr} \uparrow \alpha = W_{Pr}(W_{Pr} \uparrow (\alpha - 1)) \text{ if } \alpha \text{ is a successor ordinal},\\
&W_{Pr} \uparrow \alpha = \cup\{W_{Pr} \uparrow \beta \mid \beta < \alpha\} \text{ if } \alpha \text{ is a limit ordinal}.
\end{aligned}$$

As in the case of S_{Pr}, we see that there is a least fixpoint of W_{Pr} because of the monotonicity of W_{Pr}, although the least fixpoint may be inconsistent. Alternatively we can take a nonmonotonic operator, which may have a consistent fixpoint. For example, we define Π_{Pr}: $2^{B_{Pr}} \times 2^{B_{Pr}} \to 2^{B_{Pr}} \times 2^{B_{Pr}}$ to be

$$\Pi_{Pr}(v_s, v_f) = (v''_s, v''_f),$$

where

$$\begin{aligned}
v''_s = \{&A \in B_{Pr} \mid \exists (A \leftarrow b\text{: a ground clause obtained from a clause of } Pr\text{):}\\
&b \text{ is true in } (v_s, v_f)\}\\
&\cup \{p(t_1, \dots, t_n) \in B_{Pr} \mid p(t_1, \dots, s_i, \dots, t_n) \in v_s\\
&\wedge (t_i \to_E s_i) \wedge p(t_1, \dots, t_n) \notin v''_f\},\\
v''_f = {}&GU(v_s, v_f).
\end{aligned}$$

Because Π_{Pr} is nonmonotonic, and its fixpoint does not seem to be well-organized, we rather deal with the case that semantics for *glpe*s are consistent:

Definition 4.4: Given a *glpe* Pr, the least fixpoint of W_{Pr} is denoted by $lfp(W_{Pr})$. If $lfp(W_{Pr})$ is consistent, it is said an extended well-founded model of Pr.

In the context of what follows, "$\subset$" may be used in the sense of componentwise subset inclusions.

Example 4.5: Let Pr be

$$\begin{aligned}
\{\, &p(x) &&\leftarrow q(x)\neg r(x),\\
&q(f(x)) &&\leftarrow q(x),\\
&q(b) &&\leftarrow,\\
&r(z) &&\leftarrow r(z),\\
&eq(f(a), b) &&\leftarrow\\
&eq(x, x) &&\leftarrow \quad \},
\end{aligned}$$

where x, y, z are variables, and f, a, b are function symbols. Omitting the extensions of eq, we see that

$$\begin{aligned}
&(\{p(f^i(b)), p(f^{i+1}(a)), q(f^i(b)), q(f^{i+1}(a)) \mid i \in \omega\},\\
&\quad \{r(f^i(a)), r(f^i(b)) \mid i \in \omega\} \cup \{p(a), q(a)\})\\
&\quad \subset lfp(W_{Pr}),
\end{aligned}$$

where f^i stands for i times application of f ($i \in \omega$). It is easy to see that $\leftarrow r(f^i(a))$ and $\leftarrow r(f^i(b))$ fail such that $r(f^i(a))$, $r(f^i(b)) \in FailS^k_{Pr}$ for any $k \in \omega$. On the other hand, there is an SLDNFN-refutation of $Pr \cup \{\leftarrow q(f^i(b))\}$. As well, by the application of narrowings, there is an SLDNFN-refutation of $Pr \cup \{\leftarrow q(f^{i+1}(a))\}$. Hence there is an SLDNFN-refutation of $Pr \cup \{\leftarrow p(f^{i+1}(a))\}$ or of $Pr \cup \{\leftarrow p(f^i(b))\}$. So far $p(f^{i+1}(a)), p(f^i(b)) \in SucS^{k+1}_{Pr}$ for $k \in \omega$.

Lemma 4.6: $lfp(S_{Pr}) \subset lfp(W_{Pr})$.

Proof: Let "$\subset$" be used as componetwise subset inclusions as well.

(i) $(\emptyset, \emptyset) \subset (\emptyset, \emptyset)$.

(ii) By the definitions of S_{Pr} and W_{Pr} and the unfounded set property, $S_{Pr}(u_s, u_f) \subset W_{Pr}(u_s, u_f)$. Using monotonicities of S_{Pr} and W_{Pr},

$$(u_s, u_f) \subset (u'_s, u'_f) \Rightarrow S_{Pr}(u_s, u_f) \subset S_{Pr}(u'_s, u'_f) \subset W_{Pr}(u'_s, u'_f).$$

(iii) For $\{(u^i_s, u^i_f)\}$ and $\{(u'^i_s, u'^i_f)\}$ such that $u^i_s \subset u'^i_s$ and $u^i_f \subset u'^i_f$,

$$(\cup_i u^i_s, \cup_i u^i_f) \subset (\cup_i u'^i_s, \cup_i u'^i_f).$$

By (i), (ii) and (iii), and fixpoint induction, we complete the proof. q.e.d.

The following theorem is regarded as a soundness theorem, when $lfp(W_{Pr})$ is consistent. However, even if $lfp(W_{Pr})$ is inconsistent, it holds.

Theorem 4.7: $(SucS_{Pr}, FailS_{Pr}) \subset lfp(W_{Pr})$:

Proof: It is sufficient to prove that for any $k \in \omega$

$$\forall A\text{: } [A \in SucS^k_{Pr} \Rightarrow A \in V_S(lfp(W_{Pr}))],$$
$$\forall A\text{: } [A \in FailS^k_{Pr} \Rightarrow A \in V_F(lfp(W_{Pr}))],$$

where $V_S(U_1, U_2)$ is the first atom set U_1 of (U_1, U_2), and $V_F(U_1, U_2)$ the second U_2. ("S" stands for the success set part, while "F" for the failure set part.)

We prove it by induction on k.

(1) In case that $k = 0$:

(i) Assume that $A \in SucS^0_{Pr}$. Then there is an SLDNFN-refutation of rank 0 of $Pr \cup \{\leftarrow A_0\}$ with a computed answer θ such that $A = (A_0\theta)\sigma$ for some substitution σ. Since the rank is 0, no failed SLDNFN-tree is involved in the refutation, and $SS^0_{Pr} = SucS^0_{Pr}$. By a similar proof as the proof of Theorem 3.13, $SucS^0_{Pr} \subset V_S(lfp(S_{Pr})$ by induction on length of refutations. By Lemma 4.6, $V_S(lfp(S_{Pr})) \subset V_S(lfp(W_{Pr}))$. It follows that $A \in V_S(lfp(W_{Pr}))$.

(ii) Assume that $A \in FailS^0_{Pr}$. Then there is a failed SLDNFN-tree of rank 0 of $Pr \cup \{\leftarrow A\}$. It follows that, for any $A_0 \leftarrow B_1 \ldots B_n \neg C_1 \ldots \neg C_m \in P$ such that $A = A_0\theta$ for some $\theta \in Sub$,

$\exists B_i\theta$: there is a failed SLDNFN-tree of rank 0 of $Pr \cup \{\leftarrow B_i\theta\}$,

or an occurrence t' in $\leftarrow A$ is selected such that

$\exists B, \exists\varphi$: $(A[t' \leftarrow s])\varphi = B$, that is, $A \Rightarrow_E B$

for a narrowing with an equation, and there is a failed SLDNFN-tree of rank 0 of $Pr \cup \{\leftarrow B\}$. It follows that $B_i\theta\sigma \in FailS^0_{Pr}$ for $B_i\theta\sigma \in B_{Pr}$, or $B\rho \in FailS^0_{Pr}$ for $B\rho \in B_{Pr}$, respectively. Let $W_{Fail} = \{A \in B_{Pr} \mid A \in FailS^0_{Pr}\}$. Since we are concerned with the canonical system, if $A' \in W_{Fail}$, then either

$$A' \in GU(lfp(W_{Pr})) \subset V_F(lfp(W_{Pr})),$$

or $A' \Rightarrow_E A''$ and $A'' \in GU(lfp(W_{Pr})) \subset V_F(lfp(W_{Pr}))$. (This is shown by induction on length of narrowing for A.) In the latter case, it follows from the definition of W_{Pr} that A' must be in $V_F(lfp(W_{Pr}))$. This completes the proof that $A \in FailS^0_{Pr} \Rightarrow A \in V_F(lfp(W_{Pr}))$.

(2) Assume that the theorem holds for rank k.

(i) Assume that $A \in SucS^{k+1}_{Pr}$. Then there is an SLDNFN-refutation of rank $k+1$ of $Pr \cup \{\leftarrow A_0\}$ with a computed answer θ such that $A \equiv (A_0\theta)\sigma$ for some σ. We can prove by induction on the depth of the refutation that $A \in V_S(lfp(W_{Pr}))$, as in the proof of Theorem 3.13.

(ii) Assume that $A \in FailS^{k+1}_{Pr}$. Then there is a failed SLDNFN-tree of rank $k+1$ of $Pr \cup \{\leftarrow A\}$. It follows that, for any $A_0 \leftarrow B_1 \ldots B_n \neg C_1 \ldots \neg C_m \in P$ such that $A = A_0\theta$ for some $\theta \in Sub$, either

$\exists B_i\theta$: there is a failed SLDNFN-tree of rank $k+1$ of $Pr \cup \{\leftarrow B_i\theta\}$,

$\exists C_j\theta$: $C_j\theta \in SucS^k_{Pr}$,

or an occurrence t' in $\leftarrow A$ is selected such that

$\exists B$: $(A[t' \leftarrow s])\theta = B$

for a narrowing with an equation, and there is a failed SLDNFN-tree of rank $k+1$ of $Pr \cup \{\leftarrow B\}$. Except the case that $C_j\theta \in SucS^k_{Pr}$, we can conclude that $A \in V_F(lfp(W_{Pr}))$ as in case of (1)(ii). Because $C_j\theta \in V_S(lfp(W_{Pr}))$ by induction hypothesis on rank k, we can conclude by the definition of W_{Pr} that $A \in V_F(lfp(W_{Pr}))$, even if there is any case that $C_j\theta \in SucS^k_{Pr}$. This concludes the induction step.

q.e.d.

The following theorem states the completeness of SLDNFN-resolution for non-floundering goals with respect to the extended well-founded model.

Theorem 4.8: Assume that $lfp(W_{Pr})$ is consistent for a given *glpePr*. Then

(1) $A \in V_S(lfp(W_{Pr})) \Rightarrow A \in SucS_{Pr}$ as long as $\leftarrow A$ does not flounder.

(2) $A \in V_F(lfp(W_{Pr})) \Rightarrow A \in FailS_{Pr}$ as long as $\leftarrow A$ does not flounder.

Proof: We take the predicates:

$Q(U)$: $[A \in U \Rightarrow A \in SucS_{Pr}$ as long as $\leftarrow A$ does not flounder. $]$

$R(U)$: $[A \in U \Rightarrow A \in FailS_{Pr}$ as long as

$\leftarrow A$ does not flounder.]

We have:

(i) $Q(\emptyset)$ and $R(\emptyset)$.

(ii) Assume that $Q(V_S((W_{Pr} \uparrow \alpha)))$ and $R(V_F((W_{Pr} \uparrow \alpha)))$.

(a) Now assume that $A \in V_S(W_{Pr} \uparrow (\alpha+1))$. It follows that

$$\exists A_0 \leftarrow B_1 \ldots B_n \neg C_1 \ldots \neg C_m \in P:$$
$$[A = A_0\theta \text{ for some } \theta \in Sub \wedge$$
$$\forall i.B_i\theta \in V_S(W_{Pr} \uparrow \alpha) \wedge \forall j.C_j\theta \in V_F(W_{Pr} \uparrow \alpha)],$$
$$\text{or } (A \Rightarrow_E A') \wedge A' \in V_S(W_{Pr} \uparrow \alpha).$$

By induction hypothesis,

$$(\forall i.B_i\theta \in SucS_{Pr} \wedge \forall j.C_j\theta \in FailS_{Pr})$$
$$\text{or } A' \in SucS_{Pr}.$$

It follows that $A \in SucS_{Pr}$. Hence $Q(V_S(W_{Pr} \uparrow (\alpha+1)))$.

(b) Now assume that $A \in V_F(W_{Pr} \uparrow (\alpha+1))$. It follows that

$$\forall A_0 \leftarrow B_1 \ldots B_n \neg C_1 \ldots \neg C_m \in P:$$
$$[A = A_0\theta \text{ for } \theta \in Sub \Rightarrow$$
$$\exists B_i\theta\sigma\text{: } B_i\theta\sigma \in GU(W_{Pr} \uparrow \alpha)$$
$$\vee\ \exists C_j\theta\sigma\text{: } C_j\theta\sigma \in V_S(W_{Pr} \uparrow \alpha)]$$
$$\text{or } (A \Rightarrow_E A') \wedge A' \in V_F(W_{Pr} \uparrow \alpha).$$

By induction hypothesis, $C_j\theta\sigma \in V_S(W_{Pr} \uparrow \alpha) \Rightarrow C_j\theta\sigma \in SucS_{Pr}$, and $A' \in V_F(W_{Pr} \uparrow \alpha) \Rightarrow A' \in FailS_{Pr}$. Now we examine the case that $B_i\theta\sigma \in GU(W_{Pr} \uparrow \alpha)$. Assume that $B \in GU(W_{Pr} \uparrow \alpha)$. It follows that

$$\forall (B \leftarrow D_1 \ldots D_n \neg E_1 \ldots \neg E_m\text{: a ground clause from } Pr)\text{:}$$
$$[\exists D_i\text{: } D_i \in GU(W_{Pr} \uparrow \alpha) \vee$$
$$[\exists D_j\text{: } D_j \in V_F(W_{Pr} \uparrow \alpha) \vee \exists E_k\text{:}$$
$$E_k \in V_S(W_{Pr} \uparrow \alpha)]].$$

For the goal $\leftarrow B$, and for any ground clause $B \leftarrow D_1 \ldots D_n \neg E_1 \ldots \neg E_m$,

$\leftarrow D_i$ behaves recursively as the same as $\leftarrow B$, or

$\leftarrow D_j \in FailS_{Pr}$ because of $D_j \in V_F(W_{Pr} \uparrow \alpha)$, or

$\leftarrow E_k \in SucS_{Pr}$ because of $E_k \in V_S(W_{Pr} \uparrow \alpha)$.

So far $B \in FailS_{Pr}$. Therefore, on the assumption that $A \in V_F(W_{Pr} \uparrow (\alpha+1))$, for any clause $A_0 \leftarrow B_1 \ldots B_n \neg C_1 \ldots \neg C_m \in P$ such that $A = A_0\theta$:

(b-1) there is $B_i\theta\sigma$ such that $B_i\theta\sigma \in FailS_{Pr}$, or

(b-2) there is $C_j\theta\sigma \in V_S(W_{Pr} \uparrow \alpha)$ and by induction hypothesis $C_j\theta\sigma \in SucS_{Pr}$, or

(b-3) there is $(A \Rightarrow_E A')$ for $A' \in V_F(W_{Pr} \uparrow \alpha)$ so that, because of the definition of W_{Pr}, $A'' \in V_F(W_{Pr} \uparrow \alpha)$ if $A \Rightarrow_E A''$, and by induction hypothesis $(A \Rightarrow_E A')$ for $A' \in FailS_{Pr}$ so that $A \in FailS_{Pr}$.

Since $lfp(W_{Pr})$ is consistent, $A \in V_F(W_{Pr} \uparrow (\alpha+1))$ means that $A \notin V_S(lfp(W_{Pr}))$. By Theorem 4.7, $A \notin SucS_{Pr}$. Therefore, there is no refutation for $\leftarrow A$ even in the case that $A \Rightarrow_E A'$ for $A' \in FailS_{Pr}$. Hence $\leftarrow A$ fails. That is, $R(V_F(W_{Pr} \uparrow (\alpha+1)))$.

(iii) Q is inclusive in the sense that if $Q(U_i)$ for any $U_0 \subset U_1 \subset \ldots$, then $Q(\cup_i U_i)$. Also we see that R is inclusive.

By (i), (ii) and (iii), and fixpoint induction, we have $Q(V_S(lfp(W_{Pr})))$ and $R(V_F(lfp(W_{Pr})))$. q.e.d.

5. Concluding Remarks

A combination of SLDNF resolution with infinite failure and Narrowing (referred to as SLDNFN resolution) is formulated so that it is sound and complete with respect to the extended well-founded model of the general logic program with equations. As regards the denotations of success and failure sets by SLDNFN resolution containing finite failures, we have a relation of them to a fixpoint semantics, which is regarded as an extension of Fitting/Kunen semantics constructed from the semantics for a general logic program without explicit equations. The relation can be regarded as a generalization of the relationship between the success and failure sets pair, and the fixpoint semantics, as in [20], [27].

The SLDNFN resolution is applicable to an abductive proof procedure in an abduction frame work based on a general logic program with equations. However, if we need the explanation as a set of (ground) atoms which fails, and if we are interested in the three-valued stable model, with respect to which the Eshghi and Kowalski abductive proof procedre [9], [16] is sound, we might have just soundness, even when narrowing is applied. However, it is not yet known whether the completeness is guaranteed.

References

[1] C.R. Baral and V.S. Subrahmanian, "Dualities between alternative semantics for logic programming and non-monotonic reasoning," J. of Automated Reasoning, vol.10, pp.399–420, 1993.
[2] S. Baratella, "A class of programs for which SLDNF resolution and NAF rule are complete," Theoret. Comput. Sci., vol.120, pp.83–99, 1993.
[3] W. Chen and D.S. Warren, "A goal-oriented approach to computing the well-founded semantics," J. of Automated Reasoning, vol.17, pp.279–300, 1993.
[4] K.L. Clark, "Negation as failure," in Logic and Data Bases, eds., H. Gallaire and J. Minker, pp.293–322, Plenum Press, 1978.
[5] D. deGroot and G. Lindstrom, eds., Logic Programming: Functions, Relations, and Equations, Prentice-Hall, 1986.
[6] P.M. Dung, "On the relation between stable and well-founded semantics of logic programs," Theoret. Comput.

Sci., vol.105, pp.7–25, 1992.
[7] E. Eder, "Properties of substitutions and unifiers," J. Symbolic Computation, vol.1, pp.31–46, 1985.
[8] M.H. van Emden and K. Yukawa, "Logic programming with equations," J. of Logic Programming, vol.4, pp.265–288, 1987.
[9] K. Eshghi and R.A. Kowalski, "Abduction compared with negation by failure," Proc. 6th ICLP, pp.234–255, 1989.
[10] M. Fitting, "Kleene-Kripke semantics for logic programs," J. of Logic Programming, vol.4, pp.295–312, 1989.
[11] A. Van Gelder, K.A. Ross, and J.S. Schlipf, "The well-founded semantics for general logic programs," J. ACM, vol.38, pp.620–650, 1990.
[12] A. Van Gelder, "The alternating fixpoint of logic programs with negation," J. of Computer and System Sciences, vol.47, pp.185–221, 1993.
[13] M. Gelfond and V. Lifschitz, "The stable model semantics for logic programs," Proc. 5th ICLP, pp.1070–1080, 1988.
[14] E. Giovannetti, G. Levi, C. Moiso, and C. Palamidessi, "Kernel LEAF: A logic plus functional language," J. of Comput. and Syst. Sci., vol.42, pp.139–185, 1991.
[15] J. Jaffar, J.-L. Lassez, and M.J. Maher, "A theory of complete logic programs with equality," J. Logic Programming, vol.1, pp.211–223, 1984.
[16] A. Kakas, R.A. Kowalski, and F. Toni, "Abductive logic programming," J. Logic and Computation, vol.2, pp.719–770, 1992.
[17] K. Kunen, "Negation in logic programming," J. Logic Programming, vol.4, pp.289–308, 1987.
[18] J.W. Lloyd, Foundations of Logic Programming, 2nd, Extended Edition, Springer-Verlag, 1993.
[19] W. Marek and V.S. Subrahmanian, "The relationship between stable, supported, default and autoepistemic semantics for general logic programs," Theoret. Comput. Sci., vol.103, pp.365–386, 1992.
[20] K. Marriott and H. Søndergaard, "Bottom-up dataflow analysis of normal logic programs," J. Logic Programming, vol.13, pp.181–204, 1992.
[21] C. Palamidessi, "Algebraic properties of idempotent substitutions," Lecture Notes in Computer Science, vol.443, pp.386–399, 1990.
[22] T.C. Przymusinski, "Every logic program has a natural stratification and an iterated least fixed point model," Proc. 8th ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems, pp.11–21, 1989.
[23] J.C. Shepherdson, "A Sound and complete semantics for a version of negation as failure," Theoret. Comput. Sci., vol.65, pp.343–371, 1989.
[24] J.C. Shepherdson, "SLDNF-resolution with equality," J. Automated Reasoning, vol.8, pp.297–306, 1992.
[25] M. Thielscher, "On the completeness of SLDENF-resolution," J. Automated Reasoning, vol.17, pp.199–214, 1996.
[26] A. Yamamoto, "A Theoretical Combination of SLD-Resolution and Narrowing," Proc. 4th ICLP, pp.470–487, 1987.
[27] S. Yamasaki, "SLDNF resolution with non-safe rule and fixpoint semantics for general logic programs," Theoret. Comput. Sci., vol.160, pp.283–303, 1996.

Appendix

Proof of Theorem 3.13: We firstly list up some notations necessary in the proof.

Definition 5.1: For a literal L, we define $atom(L)$ to be

$$atom(L) = \begin{cases} A & \text{if } L \equiv A \text{ for some atom } A, \\ B & \text{if } L \equiv \neg B \text{ for some atom } B. \end{cases}$$

Definition 5.2: Assume a *glpe* Pr. For $C \in Pr$, we define

$$ground(C) = \{C' \mid C' \text{ is a ground clause obtained from } C\}.$$

Definition 5.3: Assume a *glpe* Pr. For $A \leftarrow L_1 \ldots L_n \in Pr$, we define

$$posin(A \leftarrow L_1 \ldots L_n) = \{i_1, \ldots, i_l\},$$
$$negin(A \leftarrow L_1 \ldots L_n) = \{j_1, \ldots, j_k\},$$

where $L_{i_1}, \ldots, L_{i_l}$ are all literals containing no negation, and $L_{j_1}, \ldots, L_{j_k}$ all literals with negation. If $n = 0$, then we define

$$posin(A \leftarrow L_1 \ldots L_n) = negin(A \leftarrow L_1 \ldots L_n) = \emptyset.$$

It is sufficient to prove that for any $k \in \omega$

$$\forall A: \quad [A \in SS^k_{Pr} \Rightarrow A \in V_S(lfp(S_{Pr}))],$$
$$\forall A: \quad [A \in FS^k_{Pr} \Rightarrow A \in V_F(lfp(S_{Pr}))], \qquad (A\cdot 1)$$

where $V_S(U_1, U_2)$ is the first atom set of (U_1, U_2) as a pair of atom sets, and $V_F(U_1, U_2)$ the second.

We prove it by induction on k.

(1) In case that $k = 0$:

(i) Assume that $A \in SS^0_{Pr}$. Then there is an SLDNFN-refutation of rank 0 of $Pr \cup \{\leftarrow A_0\}$ with a computed answer θ such that $A = (A_0\theta)\sigma$ for some substitution σ. We prove by induction on the depth of the SLDNFN-refutation that $A \in V_S(lfp(S_{Pr}))$.

(a) Assume that the depth of the refutation is 1. Then there is a clause $B \leftarrow \in P' \cup E_I$ such that $A_0\theta \equiv B\theta$, and $(A_0\theta)\sigma \equiv (B\theta)\sigma \in SS^0_{Pr}$ for some σ. Because $(B\theta)\sigma$ is a ground atom, $(B\theta)\sigma \in V_S(S_{Pr} \uparrow 1)$. Hence

$$(B\theta)\sigma \equiv (A_0\theta)\sigma \equiv A \in V_S(S_{Pr} \uparrow 1).$$

(b) Now assume that if the depth of the refutation is d, then $A \in V_S(lfp(S_{Pr}))$. Assume a refutation of depth $d+1$ for $A \in SS^0_{Pr}$. As the first step, the following two cases are to be considered.

(b-1) In case that $\leftarrow B_1\theta_1 \ldots B_n\theta_1$ is derived with an mgu θ_1 from $\leftarrow A_0$ and $B \leftarrow B_1 \ldots B_n$: There is an SLDNFN-refutation of rank 0 of $Pr \cup \{\leftarrow B_1\theta_1 \ldots B_n\theta_1\}$ with depth d and a computed answer θ' such that $A_0\theta_1\theta' \equiv A_0\theta$. It follows that there is an SLDNFN-refutation of rank 0 of $Pr \cup \{\leftarrow B_1\theta_1\theta'\sigma \ldots B_n\theta_1\theta'\sigma\}$ with depth $d' \leq d$ and a computed answer ε, where $\leftarrow B_1\theta_1\theta'\sigma \ldots B_n\theta_1\theta'\sigma$ is ground. By Lemma 3.7, for each $1 \leq i \leq n$, there is an SLDNFE-refutation of rank 0 of $Pr \cup \{\leftarrow B_i\theta_1\theta'\sigma\}$ with depth d_i $(\leq d')$ and a computed answer ε. By the definition of SS^0_{Pr},

$$(B_i\theta_1)\theta'\sigma \in B_{Pr} \Rightarrow (B_i\theta_1)\theta'\sigma \in SS^0_{Pr}.$$

By using the induction hypothesis on the depth,

$$(B_i\theta_1)\theta'\sigma \in SS^0_{Pr} \Rightarrow (B_i\theta_1)\theta'\sigma \in V_S(lfp(S_{Pr})).$$

Hence $(B_1\theta_1)\theta'\sigma, \ldots, (B_n\theta_1)\theta'\sigma \in V_S(lfp(S_{Pr}))$. Note that

$$(B_1\theta_1)\theta'\sigma, \ldots, (B_n\theta_1)\theta'\sigma \in V_S(lfp(S_{Pr})) \\ \Rightarrow (B\theta_1)\theta'\sigma \in V_S(lfp(S_{Pr})).$$

It follows that

$$(B\theta_1)\theta'\varphi \in V_S(lfp(S_{Pr})).$$

Because $(B\theta_1)\theta'\sigma \equiv (A_0\theta)\sigma$, $(A_0\theta)\sigma \equiv A \in V_S(lfp(S_{Pr}))$.

(b-2) In case that an occurrence t' in $\leftarrow A_0$ is selected for a narrowing with an equation $eq(t,s) \leftarrow$ such that $\leftarrow (A_0[t' \leftarrow s])\theta_1$ is obtained by using an mgu θ_1 for t' and t: There is an SLDNFN-refutation of rank 0 of $Pr \cup \{\leftarrow (A_0[t' \leftarrow s])\theta_1\}$ with depth d and a computed answer θ', where $A_0\theta \equiv (A_0\theta_1)\theta'$. By the definition of SS^0_{Pr},

$$((A_0[t' \leftarrow s])\theta_1)\theta'\varphi \in B_{Pr} \\ \Rightarrow ((A_0[t' \leftarrow s])\theta_1)\theta'\varphi \in SS^0_{Pr}.$$

By using the induction hypothesis on the depth,

$$((A_0[t' \leftarrow s])\theta_1)\theta'\varphi \in SS^0_{Pr} \\ \Rightarrow ((A_0[t' \leftarrow s])\theta_1)\theta'\varphi \in V_S(lfp(S_{Pr})).$$

It follows that

$$\forall\varphi \in Sub\colon [((A_0[t' \leftarrow s])\theta_1)\theta'\varphi \in B_{Pr} \\ \Rightarrow ((A_0[t' \leftarrow s])\theta_1)\theta'\varphi \in V_S(lfp(S_{Pr}))].$$

If $(A_0\theta_1)\theta' = A_0\theta$, then $(A_0[t' \leftarrow s])\theta\sigma \in V_S(lfp(S_{Pr}))$. Note that $(A_0\theta)\sigma \rightarrow_E (A_0[t' \leftarrow s])\theta\sigma$. By the definition of S_{Pr}, $(A_0\theta)\sigma \equiv A \in V_S(lfp(S_{Pr}))$.

By (b-1) and (b-2), the induction step on the depth of the refutation is completed.

(ii) Assume that $A \in FS^0_{Pr}$. Then there is a finitely failed SLDNFN-tree of rank 0 for $Pr \cup \{\leftarrow A_0\}$ such that $A \equiv A_0\sigma$ for some $\sigma \in Sub$. We prove by induction on the depth of the finitely failed SLDNFN-tree that $A \in V_F(lfp(S_{Pr}))$.

(a) Assume that the depth is 0. There is no derivation for $\leftarrow A_0$, nor a narrowing. It follows from the definition of S_{Pr} that $A \equiv A_0\sigma \in V_F(S_{Pr} \uparrow 1)$.

(b) Now assume that if the depth of the finitely failed tree is d, then $A \in V_F(lfp(S_{Pr}))$. Also assume that the depth of the finitely failed tree for $\leftarrow A_0$ is $d+1$.

(b-1) In case that $\leftarrow A_0$ is used for the derivation by resolution: For each $\leftarrow B_1\theta_1 \ldots B_n\theta_1$ derived from $\leftarrow A_0$ and $B \leftarrow B_1 \ldots B_n \in P(Pr)$ with an mgu θ_1 for A_0 and B, there is a finitely failed SLDNFN-tree of rank 0 for $Pr \cup \{\leftarrow B_1\theta_1 \ldots B_n\theta_1\}$. It follows that there is a finitely failed SLDNFN-tree of rank 0 for $Pr \cup \{\leftarrow B_1\theta_1\varphi \ldots B_n\theta_1\varphi\}$ with depth d' $(\leq d)$, where $\leftarrow B_1\theta_1\varphi \ldots B_n\theta_1\varphi$ is ground. By Lemma 3.7, there is i $(1 \leq i \leq n)$ such that there is a finitely failed SLDNFN-tree of rank 0 for $Pr \cup \{\leftarrow B_i\theta_1\varphi\}$ with depth d'' $(\leq d')$. By the definition of FS^0_{Pr},

$$(B_i\theta_1)\varphi \in B_{Pr} \Rightarrow (B_i\theta_1)\varphi \in FS^0_{Pr}.$$

By using the induction hypothesis on the depth,

$$(B_i\theta_1)\varphi \in FS^0_{Pr} \Rightarrow (B_i\theta_1)\varphi \in V_F(lfp(S_{Pr})).$$

Hence $(B_i\theta_1)\varphi \in V_F(lfp(S_{Pr}))$. It follows that

$$\forall\varphi \in Sub\colon [(B\theta_1)\varphi \in B_{Pr} \\ \Rightarrow (B\theta_1)\varphi \in V_F(lfp(S_{Pr}))].$$

Therefore $A_0\sigma \equiv B\sigma \equiv A \in B_{Pr}$ is in $V_F(lfp(S_{Pr}))$.

(b-2) In case that an occurrence t' in $\leftarrow A_0$ is selected for a narrowing: For each equation $eq(t,s) \leftarrow$ such that there is an mgu θ_1 for t' and t, there is a finitely failed SLDNFN-tree of rank 0 for $Pr \cup \{\leftarrow (A_0[t' \leftarrow s])\theta_1\}$ with depth d' $(\leq d)$. By the definition of FS^0_{Pr},

$$(A_0[t' \leftarrow s])\theta_1\varphi \in B_{Pr} \Rightarrow (A_0[t' \leftarrow s])\theta_1\varphi \in FS^0_{Pr}.$$

By using the induction hypothesis on the depth,

$$(A_0[t' \leftarrow s])\theta_1\varphi \in FS^0_{Pr} \\ \Rightarrow (A_0[t' \leftarrow s])\theta_1\varphi \in V_F(lfp(S_{Pr})).$$

It follows that

$$\forall\varphi \in Sub\colon [((A_0[t' \leftarrow s])\theta_1\varphi \in B_{Pr} \\ \Rightarrow (A_0[t' \leftarrow s])\theta_1\varphi \in V_F(lfp(S_{Pr}))].$$

Therefore $(A_0[t' \leftarrow s])\sigma \in B_{Pr}$ is in $V_F(lfp(S_{Pr}))$. Note that $A_0\sigma \rightarrow_E (A_0[t' \leftarrow s])\sigma$, and that $A_0\sigma$ must be rewritten to a ground atom in $V_F(lfp(S_{Pr}))$ if $A_0\sigma$ may be rewritten. Finally we have $A \equiv A_0\sigma \in V_F(lfp(S_{Pr}))$.

By (b-1) and (b-2), the induction on the depth is completed. By (i) and (ii), we have completed the case of rank 0.

(2) Assume that the theorem holds for rank k.

(i) Assume that $A \in SS^{k+1}_{Pr}$. Then there is an SLDNFN-refutation of rank $k+1$ of $Pr \cup \{\leftarrow A_0\}$ with a computed answer θ such that $A \equiv (A_0\theta)\sigma$ for some σ. We prove by induction on the depth of the refutation that $A \in V_S(lfp(S_{Pr}))$.

(a) In case that the depth is 1: The proof is the same as the one for the refutation of rank 0.

(b) Assume that if the depth is d, then $A \in V_S(lfp(S_{Pr}))$. Now assume that the depth of the refutation for $A \in SS^{k+1}_{Pr}$ is $d+1$.

(b-1) In case that there is a derivation $\leftarrow L_1\theta_1 \ldots L_n\theta_1$ from $\leftarrow A_0$ and $B \leftarrow L_1 \ldots L_n$ with an mgu θ_1 for A_0 and B: There is an SLDNFN-refutation of rank $k+1$ of $Pr \cup \{\leftarrow L_1\theta_1 \ldots L_n\theta_1\}$

with depth d and a computed answer θ' such that $A_0\theta_1\theta' \equiv A_0\theta$. It follows that there is an SLDNFN refutation of rank $k+1$ of $Pr \cup \{\leftarrow L_1\theta_1\theta'\sigma \dots L_n\theta_1\theta'\sigma\}$ with depth d' $(\leq d)$ and a computed answer ε, where $\leftarrow L_1\theta_1\theta'\sigma \dots L_n\theta_1\theta'\sigma$ is ground. By Lemma 3.7, for each i $(1 \leq i \leq n)$, there is an SLDNFN-refutation of rank $k+1$ of $Pr \cup \{\leftarrow L_i\theta_1\theta'\sigma\}$ with depth d_i $(\leq d')$ and a computed answer ε. By the definition of SS^{k+1}_{Pr},

$$\forall j\colon [j \in \mathit{posin}(B \leftarrow L_1 \dots L_n) \wedge L_j\theta_1\theta'\sigma \in B_{Pr} \Rightarrow L_j\theta_1\theta'\sigma \in SS^{k+1}_{Pr}].$$

By the induction hypothesis on the depth,

$$L_j\theta_1\theta'\sigma \in SS^{k+1}_{Pr} \Rightarrow L_j\theta_1\theta'\sigma \in V_S(lfp(S_{Pr})),$$

for $j \in \mathit{posin}(B \leftarrow L_1 \dots L_n)$. By the definition of FS^k_{Pr},

$$\forall j\colon [j \in \mathit{negin}(B \leftarrow L_1 \dots L_n) \wedge \mathit{atom}(L_j\theta_1\theta'\sigma) \in B_{Pr} \Rightarrow \mathit{atom}(L_j\theta_1\theta'\sigma) \in FS^k_{Pr}].$$

By the induction hypothesis on the rank,

$$\mathit{atom}(L_j\theta\sigma) \in FS^k_{Pr} \Rightarrow \mathit{atom}(L_j\theta\sigma) \in V_F(lfp(S_{Pr})).$$

It follows that

$$\begin{aligned}&(B\theta_1)\theta'\sigma \leftarrow (L_1\theta_1)\theta'\sigma \dots (L_n\theta_1)\theta'\sigma\\ &\in \mathit{ground}(B \leftarrow L_1 \dots L_n)\\ &\Rightarrow \forall i\colon [[i \in \mathit{posin}(B \leftarrow L_1 \dots L_n)\\ &\Rightarrow (L_i\theta_1)\theta'\sigma \in V_S(lfp(S_{Pr}))]\\ &\vee [i \in \mathit{negin}(B \leftarrow L_1 \dots L_n)\\ &\Rightarrow \mathit{atom}(L_i\theta_1\theta'\sigma) \in V_F(lfp(S_{Pr}))]].\end{aligned}$$

By the definition of $lfp(S_{Pr})$,

$$[(B\theta_1)\theta'\sigma \in B_{Pr} \Rightarrow (B\theta_1)\theta'\sigma \in V_S(lfp(S_{Pr}))].$$

Note that $A \equiv (A_0\theta)\sigma \equiv (B\theta_1)\theta'\sigma \in B_{Pr}$. Hence $(A_0\theta)\sigma \equiv A \in V_S(lfp(S_{Pr}))$.

(b-2) In case that an occurrence t' in $\leftarrow A_0$ is selected for a narrowing: It is proved as the same way as in the case for the refutation of rank 0, by using the induction on the depth.

By (b-1) and (b-2), the induction step on the depth of the refutation is completed.

(ii) Assume that $A \in FS^{k+1}_{Pr}$. Then there is a finitely failed SLDNFN-tree of rank $k+1$ for $Pr \cup \{\leftarrow A_0\}$ such that $A \equiv A_0\sigma$ for some σ. We prove by induction on the depth of the finitely failed SLDNFN-tree that $A \in V_F(lfp(S_{Pr}))$.

(a) In case that the depth is 0: We can prove it as in the case for the finitely failed SLDNFN-tree of rank 0.

(b) Assume that if the depth is d, then $A \in V_F(lfp(S_{Pr}))$. Now assume the depth of the finitely failed SLDNFN-tree is $d+1$.

(b-1) In case that there is a derivation from $\leftarrow A_0$ by using resolution: For each $B \leftarrow L_1 \dots L_n \in P(Pr)$ with an mgu θ_1 for A_0 and B, there is a finitely failed SLDNFN-tree of rank $k+1$ for $Pr \cup \{\leftarrow L_1\theta_1 \dots L_n\theta_1\}$ with depth d' $(\leq d)$. It follows that there is a finitely failed SLDNFN-tree of rank $k+1$ for $Pr \cup \{\leftarrow L_1\theta_1\varphi \dots L_n\theta_1\varphi\}$ with depth d'' $(\leq d')$, where $\leftarrow L_1\theta_1\varphi \dots L_n\theta_1\varphi$ is a ground normal goal. By Lemma 3.7, there is i $(1 \leq i \leq n)$ such that a finitely failed SLDNFN-tree of rank $k+1$ for $Pr \cup \{\leftarrow L_i\theta_1\varphi\}$ exists. Hence

$$\begin{aligned}&\exists i \in \mathit{posin}(B \leftarrow L_1 \dots L_n)\colon L_i\theta_1\varphi \in FS^{k+1}_{Pr} \text{ or}\\ &\exists i \in \mathit{negin}(B \leftarrow L_1 \dots L_n)\colon \mathit{atom}(L_i\theta_1\varphi) \in SS^k_{Pr}.\end{aligned}$$

By the induction hypothesis on the depth,

$$L_i\theta_1\varphi \in FS^{k+1}_{Pr} \Rightarrow L_i\theta_1\varphi \in V_F(lfp(S_{Pr})).$$

By the induction hypothesis on the rank,

$$\mathit{atom}(L_i\theta_1\varphi) \in SS^k_{Pr} \Rightarrow \mathit{atom}(L_i\theta_1\varphi) \in V_S(lfp(S_{Pr})).$$

It follows that L_i is an atom and $L_i\theta_1\varphi \in V_F(lfp(S_{Pr}))$, or L_i is a negation of an atom and $\mathit{atom}(L_i\theta_1\varphi) \in V_S(lfp(S_{Pr}))$ for some i. It follows that

$$\forall \varphi \in \mathit{Sub}\colon [(B\theta_1)\varphi \in B_{Pr} \Rightarrow (B\theta_1)\varphi \in V_F(lfp(S_{Pr}))].$$

Note that $A_0\theta_1 \equiv B\theta_1$. Hence $A \equiv A_0\sigma \in B_{Pr}$ is in $V_F(lfp(S_{Pr}))$.

(b-2) In case that an occurrence t' in $\leftarrow A_0$ is selected for a narrowin: It is proved as in the case for the finitely failed SLDNFN-tree of rank 0, by using the induction on the depth.

By (b-1) and (b-2), the induction step on the depth of the finitely failed tree is completed.

By (i) and (ii) for (2), the induction step on the rank is completed.

By (1) and (2), the proof is completed. q.e.d.

Susumu Yamasaki received B.Eng. and M.Eng. in electrical engineering and Eng.D. in information science from Kyoto University, in 1970, 1972 and 1980, respectively. During 1974–1987, he had been an academic staff at Dept. of Information Science, Kyoto University. In 1985/86, he was a visiting fellow at Dept. of Computer Science, University of Warwick, U.K. In 1987, he joined Okayama University as a professor in computing and artificial intelligence. His research interests include theoretical computer science, computational logic and automated reasoning. He is a member of EATCS, AAAI, Mathematical Society of Japan, and Japanese Society for AI.

Kazunori Iriya received B.Eng. and M.Eng. in information technology from Okayama University, in 1994 and 1996, respectively. He had been engaged in reasoning. He is now with Hewlett-Packard Japan, Ltd, in HP consulting. He has been developing Operational Support System (OSS) for a telecommunication carrier.