

Completeness of Depth-Bounded Resolution for Weakly Reducing Programs

Arimura, Hiroki
Department of Information Systems, Kyushu University

<https://hdl.handle.net/2324/3128>

出版情報 : RIFIS Technical Report. 21, 1989-12-30. Research Institute of Fundamental
Information Science, Kyushu University

バージョン :

権利関係 :



RIFIS Technical Report

Completeness of Depth-Bounded Resolution for Weakly Reducing Programs

Hiroki Arimura

December 30, 1989

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan

E-mail: arim@rifis.sci.kyushu-u.ac.jp Phone: 092(641)1101 Ext.2329

Completeness of Depth-Bounded Resolution for Weakly Reducing Programs

Hiroki Arimura

Department of Information Systems
Kyushu University 39, Kasuga 816
arim@rifis1.sci.kyusyu-u.ac.jp

Abstract

SLDNF-resolution procedure is not complete with respect to the perfect model semantics for logic programs in general. In this paper, we introduce two classes of logic programs containing function symbols, reducing programs and weakly reducing programs, which are characterized by the size of atom. For these classes of programs, we prove the completeness of the derivation procedure which makes use of depth-bound. First, we introduce the local finiteness of Herbrand base of a program, and prove the finite fixpoint property for the class of programs which satisfies the local finiteness. Further, we show that the class of weakly reducing programs, a subclass of locally stratified programs, has some syntactic conditions, and prove that it has the finite fixpoint property. Using the finite fixpoint property, we prove the completeness of depth-bounded derivations for weakly reducing programs. In particular, we prove the completeness of unbounded derivations for reducing programs.

1. Introduction

The completeness of the derivation procedure is an important problem in logic programming. Any Turing machine can be simulated by a definite program[15]. Since the *termination problem* of Turing machines is undecidable, the completeness of a derivation procedure in logic programming does not hold in general. However, there are some completeness results for restricted classes of logic programs.

SLDNF-resolution is complete for the class of *allowed hierarchical programs* [10], which can not represent any recursion. QSQR/SLS query evaluation procedure[7] and OLDTNF-resolution[14] are complete for the class of *allowed stratified databases*[1], which is the class of programs containing no function symbols. The class has enough power to express deductive databases. However, many of small efficient logic programs like list-operation programs are not contained in the class, because the *function-freeness* prevents us from using complex terms representing data structures such as *List* and *Binary-Tree*. The *allowedness* rejects many of “acceptable” programs[5],[11] including the above mentioned small efficient logic

programs. For example, the program P1 below is neither function-free nor allowed.

$$P1 = \left\{ \begin{array}{l} member(x, cons(x, z)) \\ member(x, cons(y, z)) \leftarrow member(x, z) \end{array} \right\}$$

The *stratifiedness* is a condition to ensure the *freedom from recursive negation*. The following program P2 satisfies the freedom and has a clear declarative semantics, that is, even numbers represented by the successor function. However, it is not stratified.

$$P2 = \left\{ \begin{array}{l} even(s(x)) \leftarrow \neg even(x) \\ even(0) \end{array} \right\}$$

In this paper, we discuss the iterative fixpoint semantics and the procedural semantics for these natural and efficient logic programs, not for deductive databases. More precisely, we discuss three conditions, “local stratified”, “locally finite” and “local-variable-free” instead of “stratified”, “functions-free” and “allowed”, respectively.

First, we extend the notion of the finiteness of Herbrand base to the local finiteness and prove the finite fixpoint property for locally finite stratified programs. Further, we introduce two classes of logic programs, weakly reducing programs and reducing programs, which are characterized by the size of atom. These classes include many of small efficient logic programs. We give some syntactic conditions of two classes and prove that these classes are locally finite stratified and local-variable-free. Moreover, we prove the completeness of depth-bounded derivation procedures for weakly reducing programs using finite fixpoint property for locally finite programs. In particular, we prove the completeness of unbounded derivation procedures for its subclass, reducing programs.

2. Preliminaries

A *program clause* is a clause of the form $A \leftarrow L_1, \dots, L_n$. A *goal* is a clause of the form $\leftarrow L_1, \dots, L_n$, where A is an atom and L_1, \dots, L_n are literals. A *program* is a finite non-empty set of program clauses.

A *definite program clause* is a clause of the form $A \leftarrow A_1, \dots, A_n$. A *definite goal* is a clause of the form $\leftarrow A_1, \dots, A_n$, where A and A_1, \dots, A_n are atoms. A *definite program* is a finite non-empty set of definite program clauses.

A *ground term* is a term containing no variables. Similarly, a *ground atom* is an atom containing no variables. Let L be a first order language. The *Herbrand base* B_L is the set of all ground atoms. (In case L has no constants, we add some constant “a” to the first order language L .) Let P be a program. The *Herbrand instantiation* of P , denoted by $ground(P)$, is the set of all ground instances of a program P . An *Herbrand model* M is a subset of B_P which is a model of P . In this paper, we only deal with Herbrand models, so we call an Herbrand model a model.

Locally stratified programs were introduced by Przymusiński[12], who generalized the class of stratified programs. A program P is *locally stratified* if there is a partition

$$B_P = H_0 + H_1 + \cdots + H_\alpha + \cdots \quad (\alpha < \omega)$$

of an Herbrand base which satisfies the following three conditions for every program clause $A \leftarrow L_1, \dots, L_n \in \text{ground}(P)$ and any $i = 1, \dots, n$:

1. $A \in H_k$ ($k \geq 0$),
2. $D_i \in \cup\{H_j \mid j \leq k\}$, if L_i is positive literal D_i ,
3. $D_i \in \cup\{H_j \mid j < k\}$, if L_i is negative literal $\neg D_i$.

We call the partition $H_0, \dots, H_\alpha, \dots$ the *local stratification* of B_P , and the partition $P_0, \dots, P_\alpha, \dots$ of $\text{ground}(P)$ the *local stratification* of $\text{ground}(P)$, where

$$P_\alpha = \{A \leftarrow A_1, \dots, A_q \in \text{ground}(P) \mid A \in H_\alpha, q \geq 0\}$$

and $\alpha < \omega$. The *level* of a positive literal A is α if $A \in H_\alpha$, and the *level* of a negative literal $\neg A$ is $\alpha + 1$ if $A \in H_\alpha$. Every stratified program is locally stratified.

In this paper, we consider the perfect model semantics as the semantics for locally stratified programs. The perfect model is a minimal model with respect to a preference order between models. A locally stratified program has a unique perfect model independent of the stratification.

We now define the perfect model. Let P be a program, $A, B, C, D \in B_P$. Then, *priority orders* \prec and \preceq on Herbrand base B_P is given as follows:

1. (condition I.) $A \prec B$ if there exists a program clause $A \leftarrow A_1, \dots, \neg B, \dots, A_n \in \text{ground}(P)$ ($n > 0$). (We say this clause defines $A \prec B$.)
2. (condition II.) $A \preceq B$ if there exists a program clause $A \leftarrow A_1, \dots, B, \dots, A_n \in \text{ground}(P)$ ($n > 0$). (We say this clause defines $A \preceq B$.)
3. (transitivity of \preceq) $A \preceq C$ if $A \preceq B$ and $B \preceq C$.
4. (transitivity of \prec) $A \prec C$ ($D \prec B$) if $A \preceq B$ and $B \prec C$ ($D \prec A$).
5. ($\prec \Rightarrow \preceq$) If $A \prec B$ then $A \preceq B$.
6. (closure axiom) \prec and \preceq are only the relations defined above.

Let M, N be distinct models of P . M is *preferable* to N if for every $A \in M - N$ there exists a $B \in N - M$ such that $A \prec B$. We call a model M a *perfect model* of P if no other model of P is preferable to M .

The perfect model can be characterized by the fixpoint of a mapping $T_P : 2^{B_P} \rightarrow 2^{B_P}$ for logic programs containing negative literals in the bodies of clauses. We define ordinal powers of the mapping T_P by

$$T_P(I) = \left\{ A \in B(P) \mid \begin{array}{l} \text{for some literals } L_1 \wedge \dots \wedge L_n, \\ A \leftarrow L_1, \dots, L_n \in \text{ground}(P) \\ \text{and } M \models L_1 \wedge \dots \wedge L_n \end{array} \right\},$$

$$T_P \uparrow 0(I) = I,$$

$$T_P \uparrow \alpha(I) = T_P(T_P \uparrow (\alpha - 1)(I)) \cup T_P \uparrow (\alpha - 1)(I), \text{ if } \alpha \text{ is a successor ordinal,}$$

$$T_P \uparrow \alpha(I) = \bigcup \{T_P \uparrow \beta(I) \mid \beta < \alpha\}, \text{ if } \alpha \text{ is a limit ordinal.}$$

An ordinal γ_α is the *closure ordinal* if γ_α is a least ordinal such that

$$T_{P_\alpha} \uparrow \gamma_\alpha(I) = T_{P_\alpha} \uparrow (\gamma_\alpha + 1)(I).$$

For a locally stratified program, the closure ordinal $\gamma_\alpha \leq \omega$ for any $\alpha < \omega$. Now we define the fixpoint model of a locally stratified program[4].

Definition 1. Let P be a locally stratified program, $P_0, \dots, P_\alpha, \dots$ ($\alpha < \omega$) be a local stratification of $\text{ground}(P)$. A *fixpoint model* M_P is defined as follows:

$$M_0 = T_{P_0} \uparrow \omega(\phi),$$

$$M_\alpha = T_{P_\alpha} \uparrow \omega(M_{\alpha-1}) \quad (\alpha < \omega),$$

$$M_P = \bigcup \{M_\alpha \mid \alpha < \omega\}.$$

Note that ω is the closure ordinal. The following proposition characterizes the model M_P of a locally stratified program P [4].

Proposition 1 (Cavedon89). Let P be a locally stratified program. Then M_P is the unique perfect model of P .

Hereafter we call this fixpoint model M_P a perfect model. The *perfect model semantics* is a kind of closed world assumption (CWA) [13]. In the semantics, an atom is assigned *true* if it is in the perfect model M_P of a program P . Otherwise the atom is assigned *false*.

Let P be a program and M be an Herbrand model of P . A derivation procedure D is *sound* with respect to M if for all $A \in B_P$ the following (a) and (b) hold:

(a) If there exists a successful derivation sequence for $P \cup \{\leftarrow A\}$ then A is true in M .

(b) If every derivation sequence for $P \cup \{\leftarrow A\}$ is failed then A is false in M .

A derivation procedure D is *complete* with respect to M if for all $A \in B_P$ the following (c) and (d) hold:

- (c) If A is true in M then there exists a successful derivation sequence for $P \cup \{\leftarrow A\}$.
- (d) If A is false in M then every derivation sequence for $P \cup \{\leftarrow A\}$ is failed.

Note that the derivation sequence may be infinite for logic program in general.

We introduce some notations for our discussions. An *expression* is either a term, a literal, a conjunction or disjunction of literals. The *size* of an expression e , denoted by $|e|$, is the total number of occurrences of variable symbols, constant symbols, function symbols and predicate symbols in e . Note that $|\neg A| = |A|$ for negative literal $\neg A$.

Example 1. Let x and y be variable symbols, a and $[]$ be constant symbols, $cons$ be a function symbol and p be a predicate symbol. Then,

$$\begin{aligned} |x| &= 1, \quad |a| = 1, \\ |cons(a, cons(x, []))| &= 5, \\ |\neg p(x)| &= 2. \end{aligned}$$

Let $o(x, A)$ be the number of all occurrences of a variable x in an atom A , and let $v(A)$ be the set of all variables in atom A . Let $\sharp(S)$ be the number of elements in a set S , and let

$$S|_n = \{A \in S \mid |A| \leq n\},$$

where S is a set of expressions.

An order \prec is *noetherian* if there exists no infinite sequence d_1, d_2, d_3, \dots such that $d_1 \prec d_2 \prec d_3 \prec \dots$.

3. Locally Finite Stratified Programs

In this section, we consider the termination problem in logic programming. If B_P is finite then the closure ordinal γ of the mapping T_P is finite. However, B_P may be infinite if a program P contains function symbols. Thus, γ may not be infinite. Originally the notion of stratification was introduced for the consistency problem of the completed programs. On the other hand, we will use it for the termination problem of programs. Now we define the notion of the locally finite stratification.

Definition 2. Let P be a locally stratified program and $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) be the local stratification of B_P . P is *locally finite stratified* if

$$H^\alpha = \bigcup \{H_j \mid j \leq \alpha\}$$

is a finite set for any $\alpha < \omega$. We call $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) the *locally finite stratification* of B_P , and the corresponding local stratification of $ground(P)$ the *locally finite stratification* of $ground(P)$.

For a definite program P , B_P is a local stratification of B_P . Thus, definite program is locally stratified. However, B_P may not be the locally finite stratification for a definite program P if P contains function symbols.

Example 2. The definite program

$$P_3 = \left\{ \begin{array}{l} p(f(x)) \leftarrow p(x) \\ p(x) \leftarrow p(x) \\ p(a) \end{array} \right\}$$

has B_{P_3} as a trivial local stratification of Herbrand base. However, B_{P_3} is not a locally finite stratification because $B_{P_3} = \{p(f^i(a) \mid i < \omega)\}$ is not finite. On the other hand, there is another local stratification $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) of B_{P_3} such that $H_\alpha = \{p(f^\alpha(a))\}$. Since $H^\alpha = \{p(f^i(a) \mid i \leq \alpha)\}$ is finite for any $\alpha < \omega$, the local stratification $H_0, \dots, H_\alpha, \dots$ is a locally finite stratification. Hence, P_3 is a locally finite stratified program, and has countably infinite strata.

Lemma 2. Let P be a locally stratified program, $P_0, \dots, P_\alpha, \dots$ ($\alpha < \omega$) be the local stratification of $\text{ground}(P)$ and $M \subseteq B_P$. Then,

$$T_{P_\alpha} \uparrow j(M) \subseteq T_{P_\alpha} \uparrow (j+1)(M).$$

Proof. By the definition of ordinal powers of T_P , the result immediately holds. \square

Lemma 3. Let P be a locally stratified program, $P_0, \dots, P_\alpha, \dots$ ($\alpha < \omega$) be the local stratification of $\text{ground}(P)$ and $A \in B_P$. Suppose $M_{P_\alpha} = T_{P_{\alpha-1}} \uparrow \omega(M_{P_{\alpha-1}})$ and $M_{P_{-1}} = \phi$. If there exists a $k < \omega$ such that

$$T_{P_\alpha} \uparrow (k)(M_{\alpha-1}) \supseteq T_{P_\alpha} \uparrow (k+1)(M_{\alpha-1}),$$

then for any $j > k$

$$T_{P_\alpha} \uparrow (k)(M_{\alpha-1}) \supseteq T_{P_\alpha} \uparrow (j)(M_{\alpha-1}).$$

Proof. It suffices to prove only the case $j = k+2$. First suppose $A \in T_{P_\alpha} \uparrow (k+2)(M_{\alpha-1})$. Then,

$$A \in T_{P_\alpha}(T_{P_\alpha} \uparrow (k+1)(M_{\alpha-1})) \cup T_{P_\alpha} \uparrow (k+1)(M_{\alpha-1}).$$

If $A \in T_{P_\alpha} \uparrow (k+1)(M_{\alpha-1})$ then it immediately holds. Thus, we suppose

$$A \in T_{P_\alpha}(T_{P_\alpha} \uparrow (k+1)(M_{\alpha-1})).$$

There is a clause $A \leftarrow L_1, \dots, L_q \in P_\alpha$ such that

$$T_{P_\alpha} \uparrow (k+1)(M_{\alpha-1}) \models L_1 \wedge \dots \wedge L_q.$$

There are two cases to be considered for each L_i .

1. If L_i is a positive literal B then $B \in T_{P_\alpha} \uparrow (k+1)(M_{\alpha-1})$. By the assumption of lemma, $B \in T_{P_\alpha} \uparrow (k)(M_{\alpha-1})$.
2. If L_i is a negative literal $\neg B$ then $B \notin T_{P_\alpha} \uparrow (k+1)(M_{\alpha-1})$. By the definition of ordinal powers of the mapping, $B \notin T_{P_\alpha} \uparrow (k)(M_{\alpha-1})$.

By the above 1 and 2, $T_{P_\alpha} \uparrow (k)(M_{\alpha-1}) \models L_1 \wedge \dots \wedge L_q$. We obtain $A \in T_{P_\alpha} \uparrow (k+1)(M_{\alpha-1})$. By the assumption of lemma, $A \in T_{P_\alpha} \uparrow (k)(M_{\alpha-1})$. Thus,

$$T_{P_\alpha} \uparrow (k)(M_{\alpha-1}) \supseteq T_{P_\alpha} \uparrow (k+2)(M_{\alpha-1}).$$

The remainder of the proof proceeds on in the same way by an induction on j . \square

Definition 3. Let P be a locally stratified program, $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) be the local stratification of B_P , and let $M_{P_\alpha} = T_{P_{\alpha-1}} \uparrow \omega(M_{P_{\alpha-1}})$ and $M_{P_{-1}} = \phi$.

Then, we define a lattice

$$\Lambda_\alpha = \{M_{\alpha-1} \cup I \mid I \subseteq H_\alpha\}.$$

Lemma 4. Let P be a locally finite stratified program, $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) be the locally finite stratification of B_P . If

$$S_0 \subseteq S_1 \subseteq \dots \subseteq S_i \subseteq \dots \quad (S_i \in \Lambda_\alpha, i \geq 0)$$

then there exists a $k \leq \sharp(H_\alpha)$ such that $S_j = S_k$ for any $j > k$.

Proof. By the definition of Λ_α , we can assume that $S_i = M_{\alpha-1} \cup I_i$ for some $I_i \subseteq H_\alpha$ and any $i \geq 0$. Then $I_0 \subseteq I_1 \subseteq \dots \subseteq I_i \subseteq \dots$ ($i \geq 0$). Since $I_i \subseteq H_\alpha$ for any $i \geq 0$, there exists a $k \leq \sharp(H_\alpha)$ such that $I_j = I_k$ for any $j > k$. Then $S_j = S_k$ for any $j > k$. \square

Lemma 5. Let P be a locally finite stratified program, $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) be the locally finite stratification of B_P . Then, the closure ordinal $\gamma_\alpha \leq \sharp(H_\alpha)$ for any $\alpha < \omega$.

Proof. By Lemma 2, for any $j < \omega$

$$T_{P_\alpha} \uparrow 0(M_{\alpha-1}) \subseteq \dots \subseteq T_{P_\alpha} \uparrow j(M_{\alpha-1}) \subseteq T_{P_\alpha} \uparrow j+1(M_{\alpha-1}) \subseteq \dots \subseteq T_{P_\alpha} \uparrow \omega(M_{\alpha-1}).$$

By definitions of the local stratification of $\text{ground}(P)$ and ordinal powers of T_P , $T_{P_\alpha} \uparrow j(M_{\alpha-1}) \in \Lambda_\alpha$ for any $j \leq \omega$. By Lemma 3 and Lemma 4, there exists a $k \leq \sharp(H_\alpha)$ such that $T_{P_\alpha} \uparrow k(M_{\alpha-1})$ is a fixpoint $T_{P_\alpha} \uparrow \omega(M_{\alpha-1})$. \square

The closure ordinal γ_α corresponding to each H_α is finite for a locally finite stratified program. Now we show the finite fixpoint property for locally finite stratified programs.

Theorem 6. *Let P be a locally finite stratified program, $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) be the locally finite stratification of B_P and $P_0, \dots, P_\alpha, \dots$ ($\alpha < \omega$) be the locally finite stratification of $\text{ground}(P)$. If γ_α is the closure ordinal of a mapping $T_{P_\alpha} : \Lambda_\alpha \rightarrow \Lambda_\alpha$ for any $\alpha < \omega$, then*

$$\gamma_0 + \dots + \gamma_\alpha \leq \#(H^\alpha).$$

Proof. By Lemma 5, if $T_{P_\alpha} \uparrow \omega(M_{\alpha-1}) = T_{P_\alpha} \uparrow \gamma_\alpha(M_{\alpha-1})$ for any $\alpha < \omega$ then $\gamma_\alpha \leq \#(H_\alpha)$. Hence,

$$\begin{aligned} \gamma_0 + \dots + \gamma_\alpha &\leq \#(H_0) + \dots + \#(H_\alpha) \\ &= \#(H_0 \cup \dots \cup H_\alpha) \\ &= \#(H^\alpha). \end{aligned}$$

□

The above corollary asserts that we can decide whether a ground atom in H_α is true or false on the perfect model of a program P by applying operators T_{P_j} ($j = 0, \dots, \alpha$) at most $\#(H^\alpha)$ times. The following corollary is an extension of the theorem in Yamamoto[17] for elementary formal systems[2].

Corollary 7. *Let P be a definite program, and let P be locally finite stratified, $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) be the locally finite stratification of B_P and $A \in B_P$. If $A \in H_\alpha$ then*

$$A \in T \uparrow \omega(\phi) \iff A \in T \uparrow n(\phi)$$

for the finite ordinal $n = \#(H^\alpha)$.

Proof. We consider the increasing sequence

$$T_{P_0} \uparrow 0(\phi), \dots, T_{P_0} \uparrow \gamma_0(\phi), \dots, T_{P_\alpha} \uparrow \gamma_\alpha(M_{\alpha-1}).$$

Since a definite program has no negative literals in the body of a clause, we can consider the lattice 2^{B_P} whose least element is ϕ , instead of the sequence $\Lambda_0, \dots, \Lambda_\alpha$ of lattices. Thus, the increasing sequence above is reduced to the sequence

$$T_P \uparrow 0(\phi), \dots, T_P \uparrow \gamma_0(\phi), \dots, T_P \uparrow (\gamma_0 + \dots + \gamma_\alpha)(\phi).$$

Hence, the result holds by Theorem 6. □

4. Depth-Bounded BFNF-Derivation

We introduce the notion of depth-bounded BFNF-derivation to discuss the termination property of logic programs more precisely. “BF” stands for “Breadth First”, and “NF”

stands for “Negation as Failure”. A BFNF-derivation is a variant of the usual SLDNF-resolution, where in each goal all literals are selected. This computation rule ensures that every derivation is a fair[9] derivation.

We define a BFNF/ d -derivation for programs, which is an extension of the (P, E) -derivation[6]. The BFNF/ d -derivation sequence is a BFNF-derivation sequence which is bounded by the depth-bound d , where d is an ordinal. Throughout this paper, we assume $d \leq \omega$.

Definition 4. Let G be a goal $\leftarrow L_1, \dots, L_n$ (L_i is a literal) and P be a program. If there are a substitution θ and an n -tuple (C_1, \dots, C_n) (each C_i is a literal) such that the following conditions 1, 2 and 3 hold for every $i = 1, \dots, n$, then a goal $\leftarrow R_1, \dots, R_n$ is *derived from G and P using the most general unifier (mgu) θ* .

1. θ is an mgu of conjunctions of literals (L_1, \dots, L_n) and (C_1, \dots, C_n) .
2. If L_i is a positive literal A then
 - (a) C_i is a positive literal B and there is a clause $B \leftarrow M_1, \dots, M_q$ ($q \geq 0$) which is a variant of a clause in P . (We call each M_j ($j = 1, \dots, q$) a child of L_i .)
 - (b) R_i is a conjunction of literals $(M_1, \dots, M_q)\theta$.
3. If L_i is a ground negative literal $\neg A$ then
 - (a) C_i is $\neg A$ and every BFNF/ d -derivation sequence for $P \cup \{\leftarrow A\}$ is failed. (If a literal M is a child of A in the BFNF/ d -derivation sequence for $P \cup \{\leftarrow A\}$ then we call M the child of L_i . Moreover, we call derivation sequences for $P \cup \{\leftarrow A\}$ *partial derivation sequences* for $P \cup \{\leftarrow \neg A\}$.)
 - (b) R_i is empty.

Definition 5. Let P be a program, G be a goal and $d \leq \omega$ be a depth-bound. A BFNF/ d -derivation sequence for $P \cup \{G\}$ consists of a sequence $G_0 = G, G_1, \dots, G_\alpha, \dots$ of goals and a sequence $\theta_1, \theta_2, \dots, \theta_\alpha, \dots$ ($\alpha \leq d$) of mgus such that each G_{i+1} is derived from G_i and P using θ_{i+1} .

A derivation sequence G_0, G_1, \dots is *successful* if there is an n such that G_n is an empty clause. A derivation sequence G_0, G_1, \dots is *floundered* if there is an n such that G_n contains non-ground negative literals. A derivation sequence G_0, G_1, \dots, G_n is *failed* with length n if either any goal cannot be derived from G_n which contains no non-ground negative literal (in this case the sequence is *finitely failed*), or $n = d$ and a goal G_n is not empty (in this case the sequence is *depth-bound-failed*). Otherwise a derivation sequence G_0, G_1, \dots has *infinite* length.

A BFNF/ d -derivation is *successful* if there exists a successful BFNF/ d -derivation sequence for $P \cup \{G\}$. A BFNF/ d -derivation is *failed* if there exists an n such that every BFNF/ d -derivation sequence for $P \cup \{G\}$ is failed within length n .

We define a BFNF/ d -tree for measuring only the time derivation procedure requires.

Definition 6. Let P be a program, G be a goal and D be a BFNF/ d -derivation sequence G_0, G_1, \dots for $P \cup \{G\}$. A BFNF/ d -tree T for D is a tree defined as follows:

1. a root of T is a literal in G ,
2. every node in T is a literal in D or in partial derivation sequences of D ,
3. if a literal B is a child of a literal A in D or in a partial derivation sequence of D then B is a child of A in T .

The *depth* of a BFNF/ d -tree T is the maximum length of branches in T .

We define an unbounded BFNF-derivation and a depth-bounded BFNF-derivation. If a depth-bound d is the smallest limit ordinal ω then we call the derivation sequence an *unbounded* BFNF-derivation sequence. If $d < \omega$ then we call the derivation sequence a *depth-bounded* BFNF-derivation sequence with depth-bound d . We define successful, floundered, failed, infinite derivation sequences and BFNF-trees in the same way as above. Note that a failed unbounded BFNF-derivation sequence is a finitely failed unbounded BFNF-derivation sequence. Because an unbounded BFNF-derivation has no derivation sequence which is depth-bound-failed. Note also that a depth-bounded BFNF-derivation has no infinite derivation sequence.

An unbounded BFNF-derivation is sound like the usual SLDNF-resolution. We show the soundness of a successful unbounded BFNF-derivation and a finitely failed unbounded BFNF-derivation.

Proposition 8. Let P be a local-variable-free program, M_P be the perfect model of P , and L be a ground literal. Then the following (a) and (b) hold:

- (a) If there exists an unbounded BFNF-derivation for $P \cup \{\leftarrow L\}$ then $M_P \models L$.
- (b) If there exists an n such that every unbounded BFNF-derivation for $P \cup \{\leftarrow L\}$ is finitely failed within the length n then $M_P \models \neg L$.

Corollary 9. An unbounded BFNF-derivation for a local-variable-free program is sound with respect to its perfect model semantics.

A depth-bounded BFNF-tree has no infinite branch, since a depth-bound d is less than ω .

Lemma 10. Let P be a locally stratified program, L be a literal, and d be a depth-bound. If the level of L is α and a depth-bound $d < \omega$ then a depth-bounded BFNF-tree for $P \cup \{G\}$ is a finite tree whose depth is at most $(\alpha + 1) \cdot d$.

Proof. The length of each depth-bounded BFNF-derivation sequences and partial derivation sequences with depth-bound d is at most d . On the other hand, the number of strata whose level is lower than or equal to α is at most $\alpha + 1$. Hence, the length of every branch of the depth-bounded BFNF-tree is less than or equal to $(\alpha + 1) \cdot d$. \square

5. Weakly Reducing Programs and Reducing Programs

Definition 7. A program clause $A \leftarrow L_1, \dots, L_n$ is *weakly reducing* if for any substitution θ and any $i = 1, \dots, n$ the following conditions 1 and 2 hold:

1. $|A\theta| \geq |L_i\theta|$ if L_i is a positive literal.
2. $|A\theta| > |L_i\theta|$ if L_i is a negative literal.

A program clause $A \leftarrow L_1, \dots, L_n$ is *reducing* if

$$|A\theta| > |L_i\theta|$$

for any substitution θ and any $i = 1, \dots, n$.

A program P is *weakly reducing* (*reducing*) if every program clause in P is weakly reducing (reducing).

Example 3. A program P1 and a program P2 shown in Section 1 are reducing programs. Every reducing programs are also weakly reducing programs.

The proposition below provides some syntactic conditions that programs be weakly reducing and reducing.

Proposition 11. Let P be a program. Suppose P contains at least one function symbol. A clause $A \leftarrow L_1, \dots, L_n$ in P is weakly reducing (reducing) if and only if the following conditions (a), (b) and (c) hold for any variable x in the clause and any $i = 1, \dots, n$:

- (a) $o(x, A) \geq o(x, L_i)$.
- (b) $|A| \geq (>) |L_i|$, if L_i is a positive literal.
- (c) $|A| > |L_i|$, if L_i is a negative literal.

Suppose P contains no function symbols. Then a clause $A \leftarrow L_1, \dots, L_n$ in P is weakly reducing (reducing) if and only if above conditions (b) and (c) hold for any variable x in the clause and any $i = 1, \dots, n$.

Proof. Let C be a program clause $A \leftarrow L_1, \dots, L_m$ ($n \geq 0$), $v(C)$ be a set of variables $\{x_1, \dots, x_q\}$ and θ be a substitution. We assume that θ is a substitution restricted to variables in C without loss of generality. Let $\theta = \{x_j := t_j \mid j = 1, \dots, q\}$. Then, for $i = 1, \dots, n$

$$|A\theta| - |L_i\theta| = \sum_{j=1}^q (o(x_j, A) - o(x_j, L_i)) (|t_j| - |x_j|) + |A| - |L_i|.$$

(\Rightarrow part) Suppose P contains function symbols, and conditions (a),(b) and (c) hold for any $i = 1, \dots, n$. Since $|t_j| - |x_j| \geq 0$, the following conditions (d) and (e) hold for any θ :

(d) $|A\theta| \geq |L_i\theta|$, if L_i is a positive literal.

(e) $|A\theta| > |L_i\theta|$, if L_i is a negative literal.

Thus, C is weakly reducing. If there exists no function symbols then (b) and (c) imply (d) and (e). Thus, C is weakly reducing.

(\Leftarrow part) Suppose C is weakly reducing. We show that if there is an i ($1 \leq i \leq m$) such that one of the conditions (a),(b) and (c) does not hold then contradiction is derived. Suppose first there is an i ($1 \leq i \leq m$) such that neither (b) nor (c) holds. Then contradiction is immediately derived for an identity substitution θ . Therefore, conditions (b) and (c) hold for any $i = 1, \dots, n$. This is independent of existence of function symbols.

Suppose next there are k and m ($1 \leq k \leq q, 1 \leq m \leq n$) such that $o(x_k, A) < o(x_k, L_m)$. We take a substitution $\sigma = \{x_j := t_j \mid j = 1, \dots, q\}$ such that

(f) if $j \neq k$ then t_j is either a constant symbol or a variable symbol,

(g) if $j = k$ then t_k is a term that satisfies

$$|t_k| > \lceil \frac{|A| - |L_m|}{o(x_k, L_m) - o(x_k, A)} \rceil.$$

Since P contains at least one function symbol, there exists such a substitution σ . Hence, $|A\sigma| < |L_m\sigma|$ for the substitution σ . This contradicts the assumption. Thus, the condition (a) holds for any $i = 1, \dots, n$. \square

Example 4. Every propositional program not containing negation is weakly reducing. A program

$$P4 = P2 \cup \{even(x) \leftarrow even(x)\}$$

is weakly reducing. $P4$ has the same perfect model semantics as that of $P2$. $P4 \cup \{\leftarrow even(s^i(0))\}$ has no finite SLDNF-tree, while every SLDNF-tree for $P2$ and a ground goal is finite.

Throughout this paper, we only consider weakly reducing programs and reducing programs containing function symbols.

6. Freedom from Recursive Negation for Weakly Reducing Programs

The local stratifiedness ensures the freedom from recursive negation. Przymusinski[12] gave a condition:

Proposition 12 (Przymusinski88). A program P is locally stratified if and only if the priority relation \prec on B_P is noetherian.

The class of locally stratified programs is undecidable[4]. However, for weakly reducing programs the priority relation \prec on B_P is reduced to the partial order on ground atoms with respect to their sizes. Hence, we immediately obtain the following theorem.

Theorem 13. *Every weakly reducing program is locally stratified.*

Proof. Let P be a weakly reducing program, $A, B, C, D \in B_P$. It suffices to prove that the priority relation \prec on B_P is noetherian. If $A \prec B$ then the following 1, 2 or 3 hold by the definition of \prec :

1. There exists a clause which defines $A \prec B$.
2. There exist C such that $A \prec C \preceq B$ and a clause in $\text{ground}(P)$ which defines $A \prec C$.
3. There exist D such that $A \preceq D \prec B$ and a clause in $\text{ground}(P)$ which defines $D \prec B$.

We consider the case 1. Let $A \leftarrow A_1, \dots, \neg B, \dots, A_n$ ($n > 0$) be a clause to define $A \prec B$. Then $|A| > |B|$ by the definition of weakly reducing programs. In cases 2 and 3, we also prove that if $A \prec B$ then $|A| > |B|$ in the same way. Thus, the relation \prec can be reduced to the relation $>$. Therefore an increasing sequence with respect to \prec is reduced to a decreasing sequence of natural numbers. Since $>$ is noetherian, so is \prec . \square

Combining the result in Przymusinski[12] and the theorem, we can conclude that a weakly reducing program has a unique perfect model.

7. Safeness for Negation of Weakly Reducing Programs

A variable in a program clause is *local* if it appears only in the body of the clause. The condition “local-variable-free” is an alternative to the condition “allowed” with respect to occurrences of variables. Goals in a derivation sequence turn ground in a top-down manner for local-variable-free programs, while goals in a derivation sequence turn ground in a bottom-up manner for allowed programs.

Definition 8. A program clause $A \leftarrow L_1, \dots, L_n$ is *local-variable-free* if

$$v(A) \supseteq v(L_i) \ (i = 1, \dots, n).$$

A program P is *local-variable-free* if every program clause in P is local-variable-free.

Example 5. The program P_1 is a local-variable-free program.

Local-variable-free programs are also acceptable programs[5],[11].

Definition 9. Let P be a program, G be a goal and d be an ordinal. $P \cup \{G\}$ is *safe for negation* if any BFNF/ d -derivation and any partial derivation for $P \cup \{G\}$ do not flounder.

We give a condition under which $P \cup \{G\}$ is safe for negation. First we give a lemma for local-variable-free programs.

Lemma 14. *Let P be a local-variable-free program and G be a ground goal. Then, every goal in a BFNF/ d -tree for $P \cup \{G\}$ is ground.*

Proof. Since P is a local-variable-free program, a goal G_{i+1} which is derived from a ground goal G_i is ground. Thus, the result holds. \square

Hence, the next lemma follows immediately from Lemma 14.

Lemma 15. *Let P be a local-variable-free program and G be a ground goal. Then $P \cup \{G\}$ is safe for negation.*

By the condition (a) in Proposition 11, a weakly reducing program containing function symbols is local-variable-free. Thus, we have:

Lemma 16. *Let P be a weakly reducing program contains function symbols and G be a ground goal. Then $P \cup \{G\}$ is safe for negation.*

8. Termination Properties of Reducing Programs

In Section 6 and Section 7, we have shown that both a weakly reducing program and a reducing program are locally stratified and safe for negation. In this section, we show that the termination property of unbounded BFNF-derivations for reducing programs, which is a subclass of weakly reducing program. First we have:

Lemma 17. *Let P be a program. Then, a partial order $|A| > |B|$ ($A, B \in B_P$) is noetherian, and the length of the decreasing sequence $|A_1| > |A_2| > \dots$ is at most $|A_1|$.*

Lemma 18. *Let P be a reducing program containing function symbols and L be a ground literal. Then, every BFNF/ d -tree for $P \cup \{\leftarrow L\}$ is finite, and the depth of that tree is at most $|L|$.*

Proof. Let L be ground. Since a reducing program containing function symbols is local-variable-free, every goal in a BFNF/ d -tree for $P \cup \{\leftarrow L\}$ is a ground goal. We consider a branch in the BFNF/ d -tree. Since this branch is a decreasing sequence with respect to $>$, the length of it is at most $|L|$ by Lemma 17. \square

By Lemma 15 and Lemma 18, for a reducing program and a ground goal there is neither an unbounded BFNF-derivation which flounders nor an unbounded BFNF-derivation which has infinite length. By those lemmas and Proposition 9, we can prove the following theorem:

Theorem 19. *Let P be a reducing program containing function symbols, M_P be the perfect model of P and L be a ground literal. Then, the following (a) and (b) hold:*

- (a) $M_P \models L \iff$
there is a successful unbounded BFNF-derivation for $P \cup \{\leftarrow L\}$.
- (b) $M_P \models \neg L \iff$
there is an n such that every unbounded BFNF-derivation for $P \cup \{\leftarrow L\}$ is finitely failed.

Proof. By Proposition 8 in Section 4, both a successful unbounded BFNF-derivation sequence and a finitely failed unbounded BFNF-derivation sequence are sound. Hence, we only show “ \Rightarrow ” part.

(a) Let L be a ground literal. Since a reducing program is weakly reducing, $P \cup \{\leftarrow L\}$ is safe for negation. On the other hand, every unbounded BFNF-derivation sequence for a reducing program and a ground goal has a finite length by Lemma 18. Thus, every unbounded BFNF-derivation sequence for $P \cup \{\leftarrow L\}$ is either successful or finitely failed. Let $M_P \models L$. Then, there is no n such that every unbounded BFNF-derivation sequence for $P \cup \{\leftarrow L\}$ is finitely failed within length n by the soundness of finitely failed unbounded BFNF-derivation sequence. Thus, there is a successful unbounded BFNF-derivation sequence for $P \cup \{\leftarrow L\}$. Hence, part (a) holds.

(b) Combining the soundness of successful unbounded BFNF-derivation sequences and the finiteness of unbounded BFNF-derivation sequences, part (b) can be proved in the same way as in the above. \square

Corollary 20. *Let P be a reducing program containing function symbols, and L be a ground literal. Then, an unbounded BFNF-derivation for $P \cup \{\leftarrow L\}$ is complete with respect to the perfect model semantics.*

Now we consider the termination property of the usual SLDNF-resolution, instead of the BFNF-derivation. Cavedon[4] introduced the notion of locally ω -hierarchical programs, which is an extension of hierarchical programs. He showed that every fair SLDNF-resolution for locally ω -hierarchical programs terminates, and that if moreover a program and a goal are allowed then the SLDNF-resolution is complete with respect to the perfect model semantics. In fact, our reducing programs are locally ω -hierarchical, so we can also obtain the termination property of the SLDNF-resolution for reducing programs in the same way as in Cavedon[4].

9. Termination Properties of Weakly Reducing Programs

In this section, we give the solution of the termination problem of weakly reducing programs containing function symbols, and prove the correctness of depth-bounded derivations for the class.

Since a weakly reducing program is safe for negation and a depth-bounded BFNF-tree for weakly reducing programs is finite, a depth-bounded BFNF-derivation procedure returns either success or failure in finite time. Though the depth-bounded derivation may not be sound for an arbitrary depth-bound d , we can show that for a given program and a ground goal, there is a computable bounded-depth d such that the depth-bounded BFNF-derivation is complete. First we show that the subset of B_P bounded by the term size is finite.

Lemma 21. *Let L be a first order language, Σ be the set of all constant symbols and all function symbols, and Π be the set of all predicate symbols in L . If both Σ and Π are finite then $B_L|_n$ is finite and there exists a computable function $f(n)$ such that $\sharp(B_L|_n) \leq f(n)$ for any $n > 0$.*

Proof. Let $S(n)$ be the number of all ordered trees each of which has n nodes. Knuth[8] showed the following:

$$\begin{aligned} S(n) &= \frac{1}{n} \binom{2(n-1)}{n-1} \\ &= O(4^n \cdot n^{-\frac{3}{2}}). \end{aligned}$$

Now we regard a ground atom as an ordered tree whose node is labeled with the element of $(\Pi \cup \Sigma)$. Since the total number of assignments of elements of $(\Pi \cup \Sigma)$ to labels in an n -node tree is $\sharp(\Pi \cup \Sigma)^{S(n)}$, the total number of atoms of size $\leq n$ is less than or equal to

$$\sum_{k=1}^n \sharp(\Pi \cup \Sigma)^{S(k)}.$$

Therefore,

$$\sharp(B_L|_n) = O(n \cdot \sharp(\Pi \cup \Sigma)^{4^n \cdot n^{-\frac{3}{2}}}).$$

□

Thus, we show the local finiteness of weakly reducing programs.

Theorem 22. *Every weakly reducing program is locally finite stratified.*

Proof. By Theorem 13, a weakly reducing program P is locally stratified. Let

$$H_\alpha = \{A \in B_P \mid |A| = \alpha + 1\} \ (\alpha < \omega).$$

Then, $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) is the local stratification of B_P . By Lemma 21,

$$\begin{aligned} H^\alpha &= \bigcup \{H_j \mid j \leq \alpha\} \\ &= \{A \in B_P \mid |A| \leq \alpha + 1\} \\ &= B_P|_{\alpha+1} \end{aligned}$$

is finite. Then P is locally finite stratified. \square

Since the perfect model of a locally stratified program is independent of its stratification, we only consider the local stratification used in the above proof. Now we define the term-size stratification.

Definition 10. Let P be a weakly reducing program. We call the local stratification $H_0, \dots, H_\alpha, \dots$ ($\alpha < \omega$) of B_P such that

$$H_\alpha = \{A \in B_P \mid |A| = \alpha + 1\}$$

the *term-size stratification* of B_P . Further we call $P_0, \dots, P_\alpha, \dots$ ($\alpha < \omega$) corresponding to this stratification the *term-size stratification* of $\text{ground}(P)$.

We have a property of depth-bounded BFNF-derivation procedures.

Lemma 23. Let P be a weakly reducing program containing function symbols, L be a ground literal, and $d < \omega$ be a depth-bound. Then a depth-bounded BFNF-derivation sequence is either successful or failed.

Proof. This follows immediately from Lemma 10 and Lemma 16. \square

Though a depth-bounded BFNF-derivation is neither sound nor complete for an arbitrary depth-bound d , we can show that there exists a depth-bound d such that a depth-bounded BFNF-derivation is sound and complete with respect to the perfect model of the program.

Theorem 24. Let P be a weakly reducing program containing function symbols, $P_0, \dots, P_\alpha, \dots$ ($\alpha < \omega$) be the term-size stratification of $\text{ground}(P)$, γ_α be the closure ordinal of T_{P_α} , $A \in B_P$, and d be a depth-bound. If A is the level $k < \omega$ and $d \geq \sharp(B_P|_{|A|})$, then the following (a), (b) and (c) hold:

- (a) There is a successful depth-bounded BFNF-derivation sequence of length n for $P \cup \{\leftarrow A\} \implies A \in T_{P_k} \uparrow n(M_{k-1})$.
- (b) $A \in T_{P_k} \uparrow n(M_{k-1})$ and $\gamma_0 + \dots + \gamma_{k-1} + n \leq d \implies$ there is a successful depth-bounded BFNF-derivation sequence of length $\leq \gamma_0 + \dots + \gamma_{k-1} + n \leq d$ for $P \cup \{\leftarrow A\}$.

- (c) $A \in T_{P_k} \uparrow n(M_{k-1})$ and $d < \gamma_0 + \dots + \gamma_{k-1} + n \implies$
there is a successful depth-bounded BFNF-derivation sequence of
length $\leq d$ for $P \cup \{\leftarrow A\}$.

Proof. Parts (a),(b) and (c) are proved simultaneously by an induction on the level k of A .

(*Base step*) Suppose the level of A is 0. Then, P_0 is a definite program. Restricting the result of Jaffer et al.[6] to the case where E is an equality relation on ordinary first order terms and P is a definite program, we can directly obtain the following (a') and (b'),

- (a') there is a successful (P, E) -derivation sequence of length n for $P \cup \{\leftarrow A\}$
 $\implies A \in T_P \uparrow n(\phi),$

- (b') $n \leq d$ and $A \in T_P \uparrow n(\phi)$
 \implies there is a successful (P, E) -derivation sequence of length $\leq n$ for $P \cup \{\leftarrow A\},$

where P is a definite program. For a definite program P , an unbounded BFNF-derivation coincides with a (P, E) -derivation[6] in case E is an equality relation on ordinary first order terms. Furthermore, a successful depth-bounded BFNF-derivation sequence of length $n \leq d$ is a successful unbounded BFNF-derivation sequence. Hence, parts (a) and (b) immediately follows from the soundness (a') and the completeness (b') of a successful (P, E) -derivation sequence above.

Now we consider part (c). Suppose $n > d$ and $A \in T_{P_0} \uparrow n(\phi)$. Since P is a weakly reducing program, P_0 is locally finite stratified. Thus, $A \in T_{P_0} \uparrow d(\phi)$ by Corollary 7. Hence, part (c) follows from part (b).

(*Induction step*) Suppose each part of the theorem holds for any atom of level $\leq k$, and the level of A is $k + 1$. Combining parts (b) and (c) of this hypothesis, for any atom A' of level $\leq k$ and any $n < \omega$ if $A' \in T_{P_k} \uparrow n(M_{k-1})$ then there is successful depth-bounded BFNF-derivation sequence of length $\leq d$ for $P \cup \{\leftarrow A'\}$. Thus, the following (d) holds for any atom A of level $\leq k$.

- (d) Every depth-bounded BFNF-derivation sequence for $P \cup \{\leftarrow A\}$ is
 failed with depth-bound $d \geq \sharp(B_P|_{|A|}) \implies$
 $A \notin T_{P_k} \uparrow \gamma_k(M_{k-1}).$

(part (a)) We assume that there is a successful BFNF/ d derivation sequence of length $n \leq d$ for $P \cup \{\leftarrow A\}$.

Suppose first $n = 1$. Then there exists a clause $A \leftarrow \neg B_1, \dots, \neg B_q \in \text{ground}(P)$ ($q \geq 0$) such that every depth-bounded BFNF-derivation sequence for $P \cup \{\leftarrow B_i\}$ is failed for any $i = 1, \dots, q$. Since P is weakly reducing, B_i is level $\leq k$ and $d \geq \sharp(B_P|_{|A|}) \geq \sharp(B_P|_{|B_i|})$. By part (d) of main induction hypothesis,

$$B_i \notin T_{P_k} \uparrow \gamma_k(M_{k-1}).$$

As γ_k is the closure ordinal of T_{P_k} , $T_{P_k} \uparrow \gamma_k(M_{k-1}) = M_k$. Therefore $M_k \models \neg B_1 \wedge \dots \wedge \neg B_q$. Hence, $T_{P_{k+1}} \uparrow 1(M_k)$.

Suppose next $n > 1$, and the result holds for all successful depth-bounded BFNF-derivation sequences of length $\leq n - 1$. Suppose $G_0 \leftarrow A, G_1, \dots, G_n$ is the successful depth-bounded BFNF-derivation sequence for $P \cup \{\leftarrow A\}$ of length n . Since P is local-variable-free, the derivation sequence G_0, \dots, G_n consists of ground goals. Let a ground goal G_1 be $\leftarrow L_1, \dots, L_q$. By the definition of derivation sequences, G_1, \dots, G_n is the successful depth-bounded BFNF-derivation sequence of length $n - 1 \leq d$ for $P \cup \{\leftarrow L_1, \dots, L_q\}$, and there is a clause $A \leftarrow L_1, \dots, L_q \in \text{ground}(P)$. Since G_1, \dots, G_n is ground and P is local-variable-free, we can construct a successful depth-bounded BFNF-derivation sequence of length $\leq n - 1$ for $P \cup \{\leftarrow L_i\}$ ($i = 1, \dots, q$). There are three possibilities for each L_i .

1. If L_i is a positive literal $B_i \in H_j$ ($j \leq k$) then there is a successful depth-bounded BFNF-derivation sequence of length $n - 1$ for $P \cup \{\leftarrow B_i\}$. By part (a) of the main induction hypothesis, $B_i \in T_{P_k} \uparrow (n - 1)(M_{k-1})$. Since P is locally stratified, $B_i \in T_{P_{k+1}} \uparrow (n - 1)(M_k)$.
2. If L_i is a positive literal $B_i \in H_{k+1}$ then there is a successful depth-bounded BFNF-derivation sequence of length $n - 1$ for $P \cup \{\leftarrow B_i\}$. By part (a) of the secondary induction hypothesis, $B_i \in T_{P_{k+1}} \uparrow (n - 1)(M_k)$.
3. If L_i is a negative literal $\neg B_i$ ($B_i \in B_P$), then B_i is level $\leq k$ and every depth-bounded BFNF-derivation sequence for $P \cup \{\leftarrow B_i\}$ is failed. Since P is weakly reducing, $d \geq \sharp(B_P|_{|A|}) \geq \sharp(B_P|_{|B_i|})$. Thus, $B_i \notin T_{P_k} \uparrow \gamma_k(M_{k-1}) = M_k$ by part (d) of the main induction hypothesis. Since P is locally stratified and the level of $B_i \leq k$, $B_i \notin T_{P_{k+1}} \uparrow (n - 1)(M_k)$.

By 1, 2 and 3 above, $T_{P_{k+1}} \uparrow (n - 1)(M_k) \models L_1 \wedge \dots \wedge L_q$. There is a clause $A \leftarrow L_1, \dots, L_q \in \text{ground}(P)$. Hence, $A \in T_{P_{k+1}} \uparrow n(M_k)$.

(part (b)) We assume that $A \in T_{P_{k+1}} \uparrow n(M_k)$ and $\gamma_0 + \dots + \gamma_k + n \leq d$. Suppose first $n = 1$. Then, there exists a clause $A \leftarrow L_1, \dots, L_q \in \text{ground}(P)$ ($q \geq 0$) such that $M_k \models L_1 \wedge \dots \wedge L_q$. There are two possibilities for each L_i .

1. If L_i is a positive literal $B_i \in H_j$ ($j \leq k$) then $B_i \in M_k = T_{P_k} \uparrow \gamma_k(M_{k-1})$ and $\gamma_0 + \dots + \gamma_k \leq d$. By part (b) of the main induction hypothesis, there is a successful depth-bounded BFNF-derivation sequence of length $\leq \gamma_0 + \dots + \gamma_k$ for $P \cup \{\leftarrow B_i\}$.
2. If L_i is a negative literal $\neg B_i$ ($B_i \in B_P$), then $B_i \notin T_{P_{k+1}} \uparrow (n - 1)(M_k)$. Since P is locally stratified, $B_i \in H_j$ for some $j \leq k$ and $B_i \notin M_k$. By part (a) of the main induction hypothesis, if for some $n \leq d$ there is a successful depth-bounded BFNF-derivation sequence of length $\leq n$ for $P \cup \{\leftarrow B_i\}$ then $B_i \in T_{P_k} \uparrow n(M_{k-1})$. Since $B_i \notin T_{P_k} \uparrow n(M_{k-1})$ for any $n < \omega$, for any $l \leq d$ there is no successful depth-bounded

BFNF-derivation sequence of length $\leq l$. Then every depth-bounded BFNFD-derivation sequence for $P \cup \{\leftarrow B_i\}$ is failed by Lemma 23.

Combining these 1 and 2, there is a successful depth-bounded BFNFD-derivation sequence of length $\leq \gamma_0 + \dots + \gamma_k$ for $P \cup \{\leftarrow L_1, \dots, L_q\}$. Since this derivation sequence is ground, there is a successful depth-bounded BFNFD-derivation sequence of length $\leq \gamma_0 + \dots + \gamma_k + 1$ for $P \cup \{\leftarrow A\}$

Suppose next $1 < n \leq d$. Then there exists a clause $A \leftarrow L_1, \dots, L_q \in \text{ground}(P)$ ($q \geq 0$) such that $T_{P_{k+1}} \uparrow (n-1)(M_k) \models L_1 \wedge \dots \wedge L_q$. There are three possibilities for each L_i .

1. If L_i is a positive literal $B_i \in H_j$ for some $j \leq k$ then $B_i \in M_k = T_{P_k} \uparrow \gamma_k(M_{k-1})$ and $\gamma_0 + \dots + \gamma_k \leq d$. By part (b) of the main induction hypothesis, there is a successful depth-bounded BFNFD-derivation sequence of length $\leq \gamma_0 + \dots + \gamma_k$ for $P \cup \{\leftarrow B_i\}$.
2. If L_i is a positive literal $B_i \in H_{k+1}$ then $B_i \in T_{P_{k+1}} \uparrow (n-1)(M_k)$ and $\gamma_0 + \dots + \gamma_k + (n-1) \leq d$. By part (b) of the secondary induction hypothesis, there is a successful depth-bounded BFNFD-derivation sequence of length $\leq \gamma_0 + \dots + \gamma_k + (n-1)$ for $P \cup \{\leftarrow B_i\}$.
3. If L_i is a negative literal $\neg B_i$ ($B_i \in B_P$), then $B_i \notin T_{P_{k+1}} \uparrow (n-1)(M_k)$. Since P is locally stratified, $B_i \in H_j$ for some $j \leq k$ and $B_i \notin M_k$. By part (a) of the main induction hypothesis, if for some $n \leq d$ there is a successful depth-bounded BFNFD-derivation sequence of length $\leq n$ for $P \cup \{\leftarrow B_i\}$ then $B_i \in T_{P_k} \uparrow n(M_{k-1})$. Since $B_i \notin T_{P_k} \uparrow n(M_{k-1})$ for any $n < \omega$, for any $l \leq d$ there is no successful depth-bounded BFNFD-derivation sequence of length $\leq l$. Then every depth-bounded BFNFD-derivation sequence for $P \cup \{\leftarrow B_i\}$ is failed by Lemma 23.

Combining these 1, 2 and 3, there is a successful depth-bounded BFNFD-derivation sequence of length $\leq \gamma_0 + \dots + \gamma_k + (n-1)$ for $P \cup \{\leftarrow L_1, \dots, L_q\}$. Since this derivation sequence is ground, there is a successful depth-bounded BFNFD-derivation sequence of length $\leq \gamma_0 + \dots + \gamma_k + n$ for $P \cup \{\leftarrow A\}$

(part (c)) We assume that $A \in T_{P_{k+1}} \uparrow n(M_k)$ and $d < \gamma_0 + \dots + \gamma_k + n$. Since P is a weakly reducing program, P is locally finite stratified by Theorem 22. By Theorem 6, $A \in T_{P_{k+1}} \uparrow \gamma_{k+1}(M_k)$ and

$$\gamma_0 + \dots + \gamma_k + \gamma_{k+1} \leq \#(H^{k+1}).$$

Since $\#(H^{k+1}) \leq d$,

$$\gamma_0 + \dots + \gamma_k + \gamma_{k+1} \leq d.$$

By part (b) of this theorem, there is a successful depth-bounded BFNFD-derivation sequence of length $\leq \gamma_0 + \dots + \gamma_k + \gamma_{k+1} \leq d$. Hence, part (c) is proved. \square

Now we show the completeness of the depth-bounded BFNFD-derivation procedure.

Corollary 25. *Let P be a weakly reducing program containing function symbols, L be a ground literal, and d be a depth-bound. If $d \geq \#(B_P|_{|L|})$ then a depth-bounded BFNF-derivation for $P \cup \{\leftarrow L\}$ is complete with respect to the perfect model semantics.*

By Lemma 21, the depth-bound $\#(B_P|_{|L|})$ is computable. Thus, the perfect model M_P for a weakly reducing program containing function symbols is computed by the depth-bounded BFNF-derivation.

Example 6. The program P_4 is a weakly reducing program containing function symbols:

$$P_4 = \left\{ \begin{array}{l} \text{even}(s(x)) \leftarrow \neg \text{even}(x) \\ \text{even}(x) \leftarrow \text{even}(x) \\ \text{even}(0) \end{array} \right\}.$$

Its Herbrand base $B_{P_4} = \{\text{even}(s^\alpha(0)) \mid \alpha < \omega\}$, and

$$B_{P_4}|_n = \begin{cases} \phi & (n < 2), \\ \{\text{even}(s^j(0)) \mid j \leq n-2\} & (n \geq 2). \end{cases}$$

Thus, $\#(B_{P_4}|_n)$ is at most $n-1$. An atom $\text{even}(s(s(0)))$ is contained in the perfect model of P_4 , and $|\text{even}(s(s(0)))| = 4$. Hence, $P_4 \cup \{\leftarrow \text{even}(s(s(0)))\}$ has a successful depth-bounded BFNF-derivation with the depth-bound $3 (= \#(B_{P_4}|_4) = 4 - 1)$.

Acknowledgements

The author is grateful to Setsuo Arikawa for his comments and encouragement, and to Takeshi Shinohara and Akihiro Yamamoto for helpful discussions and comments.

References

- [1] Apt, K.R. , Blair, H.A. , and Walker, A. , Towards a theory of declarative knowledge, in Minker, J. (ed.), *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, 1988.
- [2] Arikawa, S. , Shinohara, T. , and Yamamoto, A. , Elementary formal system as a unifying framework for language learning, in *Proc. of the Second Annual Workshop on Computational Learning Theory*, Morgan Kaufmann, 1989.
- [3] Arimura, H. , Completeness of depth-bounded resolution in logic programming, in *6th Conference Proceedings of Japan Society for Software Science and Technology*, 1989.
- [4] Cavedon, L. , Continuity, consistency, and completeness properties for logic programs, in *Proc. of the Sixth International Conference on Logic Programming*, 1989.

- [5] Fitting, M. , A Kripke-Kleene semantics for logic programs. *J.Logic Programming* 2(4) (1985).
- [6] Jaffer, F. , Lassez, J. , and Maher, M. , A theory of complete logic programs with equality. *J.Logic Programming* 1(3) (1984).
- [7] Kemp, D.B. and Topor, R.W. , Completeness of a top-down query evaluation procedure for stratified databases, in *Proc. of the Fifth International Conference on Logic Programming*, 1988.
- [8] Knuth, D.E. , *The Art of Computer Programming*, volume 1, chapter 2. Addison Wesley, second edition, 1973.
- [9] Lloyd, J.W. , *Foundations of Logic Programming*, Springer Verlag, second edition, 1987.
- [10] Lloyd, J.W. and Topor, R.W. , A bases for deductive database systems II. *J.Logic Programming* 3(1) (1986).
- [11] Maher, M.J. , Equivalence of logic programs, in Minker, J. (ed.), *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, 1988.
- [12] Przymusiński, T. , On the declarative semantics of deductive databases and logic programs, in Minker, J. (ed.), *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, 1988.
- [13] Reiter, R. , On closed world data bases, in Gallaire, H. and Minker, J. (eds.), *Logic and Data Bases*, Plenum, 1978.
- [14] Seki, H. and Itoh, H. , A query evaluation method for stratified programs under the extended CWA, in *Proc. of the Fifth International Conference on Logic Programming*, 1988.
- [15] Tärnlund, S.A. , Horn clause computability. *BIT* 17(2) (1977).
- [16] Van Gelder, A. , Negation as failure using tight derivations for genaral logic programs, in Minker, J. (ed.), *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, 1988.
- [17] Yamamoto, A. , Elementary formal system as a logic programming language, in *Proc. of the Logic Programming Conference '89*, 1989.

About the Author



Hiroki Arimura (有村博紀) was born in Fukuoka on June 7, 1965. He received the B.S. degree from Kyusyu University in 1988. Presently, he is a graduate student of Master Course in Information Systems, Kyushu University. His interests are in logic programming and deductive database.