ICOT Technical Report: TR-178

TR-178

Derivation of Logic Programs from Implicit Definition

by

Tadashi Kanamori and Machi Maeji
(Mitsubishi Electric Corp.)

May, 1986

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Derivation of Logic Programs from Implicit Definition

Tadashi KANAMORI Machi MAEJI

Mitsubishi Electric Corporation
Central Research Laboratory
Tsukaguchi-Honmachi 8-1-1
Amagasaki,Hyogo,JAPAN 661

Abstract

A framework of Prolog program transformation, which permits more implicit definition of new predicates, is presented. Definition of a new predicate is usually done by setting a new atom, i.e., atom with the new predicate symbol, equivalent to some conjunction of old atoms, i.e., atoms with already known predicate symbols. In our new framework, we permits definitions such that some conjunction of old atoms and a new atom, called generalized head, is set equivalent to some conjunction of old atoms. This is a generalization of the framework known so far and especially usefull to accommodate a heuristics called *forced folding* or *(folding driven) goal insertion* naturally. We show the transformation rules for the extended framework and prove its correctness, that is, if a usual definite clause program P for new predicates is derived from such extended definitions D and a fixed definite clause program P_{old} for old predicates, a conjunction of atoms succeeds using $D \bigcup P_{old}$ if and only if it succeeds using $P \bigcup P_{old}$, when the conjunction is an instance of the generalized heads.

Keywords : Program Transformation, Transformation Strategy, Prolog.

Contents

1. Introduction

In the unfold/fold program transformation, we need several heuristics to derive right programs from initial description of programs. Among such heuristics, it is an important one to insert goals intentionally in order to fold derived programs by the original program.

Clark and Darlington [2] and Darlington [3] called such a technique *forced folding*. For example, when

sort($L_1 \circ L_2$)=$N_1 \circ N_2$ such that

perm(L_1,M_1),perm(L_2,M_2), perm($M_1 \circ M_2$,$N_1 \circ N_2$),ordered($N_1 \circ N_2$).

was obtained from the initial description of *sort* ($\circ$ is for *append*),

sort(L)=M such that perm(L,M),ordered(M).

they inserted $ordered(M_1) \wedge ordered(M_2)$ to fold $perm(L_1, M_1), perm(L_2, M_2)$ with these inserted goals by the initial definition, and tried to synthesize *merge* program from

perm($M_1 \circ M_2$,$N_1 \circ N_2$), ordered(M_1)$\wedge$ordered(M_2) $\supset$ordered($N_1 \circ N_2$)

But they didn't complete its derivation and their discussion was rather informal.

Tamaki and Sato [11] also used such a technique called *(folding driven) goal insertion* for Prolog program transformation. For example, when

sort([X|L],M) :- perm(L,N),insert(X,N,M),ordered(M).

was obtained from the initial description of *sort*,

sort(L,M) :- perm(L,M),ordered(M).

they inserted $ordered(N)$ in order to fold $perm(L, N)$ with the inserted goal by the initial definition, because $insert(X, N, M) \wedge ordered(M) \supset ordered(N)$ is valid in the minimum Herbrand model. But, when their "goal insertion" in general is combined with the unfold/fold rules, it might loose equivalence in the sense of minimum Herbrand model semantics and needs additional and slightly complicated book keeping to guarantee it.

In this paper, we present a framework of Prolog program transformation, which permits more implicit definition of new predicates. In transformation of Prolog programs, definition of a new predicate p is usually done by setting

$p(X_1, X_2, \ldots, X_n) \equiv A_1, A_2, \ldots, A_m.$

where $A_1, A_2, \ldots, A_m$ are atoms with old, i.e., already known predicate symbols. In our new framework, we permits definitions with generalized heads

$A_1, A_2, \ldots, A_l, p(X_1, X_2, \ldots, X_n) \equiv A_{l+1}, A_{l+2}, \ldots, A_{l+r}.$

where $A_1, A_2, \ldots, A_{l+r}$ are atoms with old predicate symbols. This is a generalization of the framework known so far and especially usefull to accommodate a heuristics called *forced folding* or *folding driven goal insertion* naturally. We show the transformation rules for the framework and prove its correctness, that is, if a usual definite clause program P for new predicates is derived from such extended definitions D and a fixed definite clause program P_{old} for old predicates, a conjunction of atoms succeeds using $D \bigcup P_{old}$ if and only if it succeeds using $P \bigcup P_{old}$, when the conjunction is an instance of the generalized heads.

This paper is organized as follows. After preparing some preliminary materials in Section 2, we explain our method by using a simple example in Section 3. Then, we prove its correctness in Section 4. Lastly in Section 5, we discuss relations to other works and problems left.

In the following, we assume familiarity with the basic terminologies of first order logic such as term, atom (atomic formula), formula, substitution, most general unifier (m.g.u.) and so on. We also assume knowledge of the semantics of Prolog such as Herbrand interpretations and minimum Herbrand models. (see [1],[4],[7]). We follow the syntax of DEC-10 Prolog [8]. As syntactical variables, we use X, Y, Z for variables, boldface letters **X**,**Y**,**Z** for sequences of variables, s, t for terms, A, B for atoms and boldface letters **H**,**B**,**O** for multisets of atoms, possibly with primes and subscripts. In addition, we use σ, τ for substitutions.

2. Preliminaries

Now on, we assume about constant, function and predicate symbols as follows.

(a) The set of constant and function symbols is fixed so that we have a fixed Herbrand universe.

(b) The set of predicate symbols is divided into two disjoint sets. One is a set of predicates called *old predicates*. Another is a set of predicates called *new predicates*.

The old predicates are defined by a fixed *explicit definite clause program* S^{old} and the new predicates are defined by an *general definite clause program* S^{new}, both of which are being explained in this section.

2.1. Atoms and Molecules

Atoms with the old predicates are called *old atoms*, while those with the new predicates are called *new atoms*. Atoms containing no variable are called *ground atoms*. Finite multisets of (ground) atoms are called *(ground) atom sets*.

An existentially quantified conjunction of the form

$$\exists X_1, X_2, \ldots, X_m \ (A_1 \wedge A_2 \wedge \cdots \wedge A_n) \ (m \geq 0, n \geq 0)$$

is called a *molecule*, where $X_1, X_2, \ldots, X_m$ are distinct variables and $A_1, A_2, \ldots, A_n$ are atoms. Variables $X_1, X_2, \ldots, X_m$ are called *existential variables*, while other variables in the molecule are called *global variables*. Molecules without global variables are called *closed molecules* and those without any variables are called *ground molecules*. (These definitions are due to Tamaki and Sato [12].) By representing each existential variable X by $?X$, the existential quantifiers at the head are omitted now on and molecules are denoted by atom sets containing (possibly no) ?-annotated variables. Molecules obtained from M by instantiating all global variables in M to ground terms are called *closed instance* of M, those obtained from a closed molecule N by instantiating existential variables to terms without global variables are called *existential instance* of N.

2.2. General Definite Clause Programs

An *explicit definite clause program* is a finite set of explicit definite clauses. An *explicit definite clause* is a formula of the form $(m \geq 0)$

A_0 :- $A_1, A_2, \ldots, A_m$.

where $\{A_1, A_2, \ldots, A_m\}$ is a molecule and there is no global variable appearing in $A_1, A_2, \ldots, A_m$ and not in A_0. An explicit definite clause without body is called a *unit clause*. (This definition is also due to Tamaki and Sato [12].)

Example 2.2.1. Let *ordered*, $\leq$ and *insert-randomly* be old predicates defined by the following explicit definite clause program S^{old}.

ordered([ ]).
ordered([X]).

ordered([X,Y|L]) :- X≤Y,ordered([Y|L]).
0≤Y.
suc(X)≤suc(Y) :- X≤Y.
insert-randomly(X,N,[X|N]).
insert-randomly(X,[Y|N],[Y|M]) :- insert-randomly(X,N,M).

A *general definite clause program* is a finite set of *general definite clauses*. A *general definite clause* is a formula of the form ($l, r \geq 0$)

$A_1, A_2, \ldots, A_l, A_0$:- $A_{l+1}, A_{l+2}, \ldots, A_{l+r}$.

where $\{A_1, A_2, \ldots, A_l\}$ and $\{A_{l+1}, A_{l+2}, \ldots, A_{l+r}\}$ are molecules consisting of old atoms, A_0 is a new atom without existential variables and there is no global variable appearing in $A_{l+1}, A_{l+2}, \ldots, A_{l+r}$ and not in $A_1, A_2, \ldots, A_l, A_0$. (Of course, these two molecule have no common existential variables.)

Following the terminologies for explicit definite clauses, the left-hand side is called the *head* and the right-hand side is called the *body* of the general definite clause. A general definite clause without body is also called a *unit clause*. (General definite clauses are of the same form as explicit definite clauses when $\{A_1, A_2, \ldots, A_l\}$ is empty, i.e., the head consists of just one atom.)

Example 2.2.2. Let ***ordered***, $\leq$, ***insert-randomly*** be old predicates defined as before and ***insert-properly*** be a new predicate defined by the following general definite clause.

ordered(N),insert-properly(X,N,M) :- insert-randomly(X,N,M),ordered(M).

2.3. Semantics of General Definite Clause Programs

Let $S^{old} \bigcup S^{new}$ be a general definite clause program. A *proof* of closed molecule $G = \{A_1, A_2, \ldots, A_k\}$ in $S^{old} \bigcup S^{new}$ is a sequence T of closed molecules defined as follows.

(a) T is a proof of G in $S^{old} \bigcup S^{new}$ when it is a sequence consisting of a single molecule G and some existential instance of G is a closed instance of the head of a unit clause in $S^{old} \bigcup S^{new}$. (The unit clause is said to be *used at the root*.)

(b) Let $A_{i_1}, A_{i_2}, \ldots, A_{i_p}, A_{i_0}$ be atoms in some existential instance $\sigma(G)$. T is a proof of G in $S^{old} \bigcup S^{new}$ when its first element is G, "$A_{i_1}, A_{i_2}, \ldots, A_{i_l}, A_{i_0}$:- $B_1, B_2, \ldots, B_r$" is a closed instance of some definite clause in $S^{old} \bigcup S^{new}$ and its tail is a proof of $G' = \sigma(G) - \{A_{i_1}, A_{i_2}, \ldots, A_{i_l}, A_{i_0}\}) \uplus \{B_1, B_2, \ldots, B_r\}$. (The definite clause is said to be *used at the root* and G is said to be *reduced to* G'.)

The length of proof T is called the *size* of T.

Example 2.3.1. Suppose ***ordered***, $\leq$ and ***insert-randomly*** are defined as before and ***insert-properly*** is defined by

insert-properly(X,[ ],[X]).
ordered([Y|N]),insert-properly(X,[Y|N],[X,Y|N]) :- X≤Y,ordered([Y|N]).
ordered([Y|N]),insert-properly(X,[Y|N],[Y|M]) :- insert-randomly(X,N,M),ordered([Y|M]).

Then the following sequence

{ insert-randomly(3,[2,4],[2,3,4]),ordered([1,2,3,4]) }
{ insert-randomly(3,[4],[3,4]),ordered([1,2,3,4]) }
{ ordered([1,2,3,4]) }
{ 1≤2,ordered([2,3,4]) }

⋮

{ ordered([4]) }

is a proof of {*insert-randomly*(3, [2, 4], [2, 3, 4]), *ordered*([1, 2, 3, 4])} and

{ ordered([2,3,4]),insert-properly(1,[2,3,4],[1,2,3,4]) }
{ $1\leq 2$,ordered([2,3,4]) }

⋮

{ ordered([4]) }

is a proof of {*ordered*([2, 3, 4]), *insert-properly*(1, [2, 3, 4], [1, 2, 3, 4])}.

Let S be a program consisting of an explicit definite clause program S^{old} and a general definite clause program S^{new}. The set of all closed molecules for which a proof in S exists is denoted by $M^*(S)$. When S consists of only explicit definite clauses, any ground molecules in $M^*(S)$ is a set of ground atoms in the usual minimum Herbrand model of S.

Example 2.3.2. Let S^{old} be a program defining *ordeded*, $\leq$ and *insert-randomly* in Example 2.2.1. Then $M^*(S^{old})$ is the set of all closed molecules. Ground molecules in it consists of ground atoms in the usual Herbrand model of S^{old}.

Let S^{new} be a program defining *insert-properly* in Example 2.2.2. Then $M^*(S^{old} \bigcup S^{new})$ contains no singleton molecule of the form {*insert-properly*(X, N, M)} where N is unordered, but contains, for example, {*ordered*([2, 3, 4]), *insert-properly*(1, [2, 3, 4], [1, 2, 3, 4])}.

We sometimes restrict our attention to a class of closed molecules. Let H be a set of closed molecules. Two programs S_1 and S_2 are said to be *H-equivalent* when $M^*(S_1) \bigcap H$ is identical to $M^*(S_2) \bigcap H$. In particular, when H is the set of all closed molecules, it is simply said to be *equivalent*. Note that S_1 and S_2 are not necessarily equivalent even if S_1 and S_2 are H-equivalent. ∎

Example 2.3.3. Suppose *insert-properly* is defined by

ordered(N),insert-properly(X,N,M) :- insert-randomly(X,N,M),ordered(M).

in S_1 and by

insert-properly(X,[ ],[X]).
insert-properly(X,[Y|N],[X,Y|N]) :- X≤Y.
insert-properly(X,[Y|N],[Y|M]) :- Y≤X,insert-properly(X,N,M).

in S_2, where *ordered*, $\leq$ and *insert-randomly* are old predicates as before. Let

H = { { ordered([ ])} } ⋃
{ { ordered([t]) } | t is a ground term } ⋃
{ { ordered([s_1,s_2| t]) } | s_1, s_2, t are ground terms } ⋃
{ { 0≤t } | t is a ground term } ⋃
{ { suc(t_1)≤ suc(t_2) } | t_1, t_2 are ground terms } ⋃
{ { insert-randomly(s,t,[s|t]) } | s, t are ground terms } ⋃
{ { insert-randomly(s_1,[s_2|t_1],[s_2|t_2]) } | s_1, s_2, t_1, t_2 are ground terms } ⋃
{ { ordered(t_1),insert-properly(s,t_1,t_2) } | s, t_1, t_2 are ground terms }

Then S_1 and S_2 are H-equivalent, but obviously not equivalent, because *insert-properly* in S_2 does not fail even if its second argument is not ordered.

3. Transformation of General Definite Clause Programs

3.1. Transformation Process

The entire process of our transformation proceeds in the completely same way as Tamaki-Sato's transformation [11] as follows.

P_0 :=the initial definite clause program ; $D_0 := \{\}$;
set each counter of definite clause in P_0 to 1;
for $i := 1$ to arbitrary N such that all definite clauses in P_N are explicit
    apply any of the transformation rules to obtain P_i and D_i from P_{i-1} and D_{i-1};

Figure 1. Transformation Process

Example 3.1. Before starting, the initial definite clause program is given, e.g.,

P$_0$: C$_1$ [[1]]. append([ ],M,M).
C$_2$ [[1]]. append([X|L],M,[X|N]) :- append(L,M,N).

and D_0 is initialized to $\{\}$. The numerals in [[ ]] denote the values of the counters. This example is used to illustrate the rules of transformation.

3.2. Basic Transformation Rules

The basic part of our transformation system consists of five rules, i.e., definition, body-unfolding, head-unfolding, folding and cancellation.

Definition : Let C be a general definite clause of the form

$$A_1, A_2, \ldots, A_l, p(X_1, X_2, \ldots, X_n) \text{ :- } A_{l+1}, A_{l+2}, \ldots, A_{l+r}.$$

where

(a) p is an arbitrary predicate appearing neither in P_{i-1} nor in D_{i-1},
(b) $X_1, X_2, \ldots, X_n$ are distinct variables which includes all global variables in the head,
(c) predicates of atoms in $A_1, A_2, \ldots, A_l, A_{l+1}, A_{l+2}, \ldots, A_{l+r}$ all appears in P_0 and
(d) $\sigma(A_1 \wedge A_2 \wedge \cdots \wedge A_l)$ is provable in P_0 when $\sigma(A_{l+1} \wedge A_{l+2} \wedge \cdots \wedge A_{l+r})$ is provable in P_0 for any closed instantiation σ.

Then let P_i be $P_{i-1} \bigcup \{C\}$ and D_i be $D_{i-1} \bigcup \{C\}$. Let C have counter 1.

The predicates introduced by the definition rule are called *new predicates*, while those in P_0 are called *old predicates*.

Example 3.2.1. We define a new predicate p by C_3. (Though the new predicate p is not interesting, this example is small enough to present our transformation rules.)

C$_3$ [[1]]. append(M,N,?MN),append(L,?MN,LMN),p(L,M,N,LMN) :-
    append(L,M,?LM),append(?LM,N,LMN).

Then $P_1 = \{C_1, C_2, C_3\}$ and $D_1 = \{C_3\}$.

Body-Unfolding : Let C be a general definite clause in P_{i-1} with counter γ defining a new predicate, A be an old atom in the body not marked "inhibited" and $C_1, C_2, \ldots, C_k$ be all the definite clauses in P_{i-1} whose heads are unifiable with A, say by m.g.u.'s $\sigma_1, \sigma_2, \ldots, \sigma_k$. Let C'_i be the result of replacing $\sigma_i(A)$ in $\sigma_i(C)$ with the body of $\sigma_i(C_i)$. (New variables substituted for global variables in A are treated as fresh global variables. Other new variables are treated as fresh existential variables.) Then let P_i be $(P_{i-1} - \{C\}) \bigcup \{C'_1, C'_2, \ldots, C'_k\}$ and D_i be D_{i-1}. Let each C'_i have counter $\gamma + 1$ unless it is already in P_{i-1} with lower counter.

It is explained in the following when old atoms are marked "inhibited". In this paper, we restricted that body-unfolded atoms be marked "inhibited" in order to make the proof in Section 4 simpler.

Example 3.2.2. When C_3 is body-unfolded at its second atom $append(L, M, ?LM)$ in the body, we obtain $P_2 = \{C_1, C_2, C_4, C_5,\}$ and $D_2=\{C_3\}$ where

C_4 [[2]]. append(M,N,?MN),append([ ],?MN,LMN),p([ ],M,N,LMN) :-
append(M,N,LMN).

C_5 [[2]]. append(M,N,?MN),append([X|L],?MN,LMN),p([X|L],M,N,LMN) :-
append(L,M,?LM),append([X|?LM],N,LMN).

Then C_5 is body-unfolded further to

C_5' [[3]]. append(M,N,?MN),append([X|L],?MN,[X|LMN]),p([X|L],M,N,[X|LMN]) :-
append(L,M,?LM),append(?LM,N,LMN).

we get $P_3=\{C_1, C_2, C_4, C_5'\}$ and $D_3=\{C_3\}$.

Head-Unfolding : Let C be a general definite clause in P_{i-1} defining a new predicate with positive counter γ, A be an old atom in the head and $C_{unfolder}$ be the only one definite clause whose heads are unifiable with A, say by an m.g.u. σ. Let C' be the result of replacing $\sigma(A)$ in the head of $\sigma(C)$ with the body of $\sigma(C_{unfolder})$. (New variables substituted for global variables in A are treated as fresh global variables. Other new variables are treated as fresh existential variables.) Then let P_i be $(P_{i-1} - \{C\}) \bigcup \{C'\}$ and D_i be D_{i-1}. Let C' have counter $\gamma - 1$ unless it is already in P_{i-1} with lower counter.

The condition that head-unfolded atoms have only one definite clause with unifiable head is crucial.

Example 3.2.3. We unfold C_4 at its head atom $append([\,], ?MN, LMN)$ to obtain $P_4=\{C_1, C_2, C_5', C_6\}$ and $D_3=\{C_3\}$, where

C_6 [[1]]. append(M,N,LMN),p([ ],M,N,LMN) :- append(M,N,LMN).

Similarly, by head-unfolding C_5' at its head atom $append([X|L], ?MN, [X|LMN])$, we get $P_4=\{C_1, C_2, C_6, C_7\}$ and $D_4=\{C_3\}$,where

C_7 [[2]]. append(M,N,?MN),append(L,?MN,LMN),p([X|L],M,N,[X|LMN]) :-
append(L,M,?LM),append(?LM,N,LMN).

Folding : Let C be a general definite clause in P_{i-1} which is of the form

$A_1, A_2, \ldots, A_p, A_0$:- $A_{p+1}, A_{p+2}, \ldots, A_{p+n}$.

with counter γ and to which no folding has ever applied. Let C_{folder} be a general definite clauses in D_{i-1} which is of the form

$B_1, B_2, \ldots, B_q, B$:- $B_{q+1}, B_{q+2}, \ldots, B_{q+m}$.

Suppose there is a substitution σ and a subset $\{A_{i_{q+1}}, A_{i_{q+2}}, \ldots, A_{i_{q+m}}\}$ of the body of C such that the following conditions hold.

(a) No atom in $\{A_{i_{q+1}}, A_{i_{q+2}}, \ldots, A_{i_{q+m}}\}$ is marked "inhibited",

(b) $A_{i_j} = \sigma(B_j)$ for $j = q+1, q+2, \ldots, q+m$,

(c) σ substitutes distinct existential variables for the existential variables of C_{folder} and moreover those variables do not occur in $\{A_0, A_1, A_2, \ldots, A_{p+n}\} - \{A_{i_{q+1}}, A_{i_{q+2}}, \ldots, A_{i_{q+m}}\}$ and

(d) $m + 1 < n + \gamma$.

Then let P_i be $(P_{i-1} - \{C\}) \bigcup \{C'\}$ and D_i be D_{i-1} where C' is a general definite clause with head $A_1, A_2, \ldots, A_p, A_0$ and body $(\{A_{p+1}, A_{p+2}, \ldots, A_{p+n}\} - \{A_{i_{q+1}}, A_{i_{q+2}}, \ldots, A_{i_{q+m}}\}) \bigcup \{\sigma(B_1), \sigma(B_2), \ldots, \sigma(B_q), \sigma(B_0)\}$. Let C' have counter $\gamma - 1$. Mark $\sigma(B_1), \sigma(B_2), \ldots, \sigma(B_q)$ in the body of C' "inhibited".

In this paper, we restricted that foldings be applied at most once to each definite clause in order to make the proof in Section 4 simpler.

Example 3.2.4. By folding the body of C_7 by C_3, we obtain $P_5=\{C_1, C_2, C_6, C_7'\}$ and $D_5=\{C_3\}$ where

C_7' [[1]]. append(M,N,?MN),append(L,?MN,LMN),p([X|L],M,N,[X|LMN]) :-
append(M,N,?MN'),append(L,?MN',LMN),p(L,M,N,LMN).

Cancellation : Let C be a general definite clause in P_{i-1} of the form

$A_1,A_2,\ldots,A_l,A_0$:- $A_{l+1},A_{l+2},\ldots,A_{l+r}$.

with counter γ defining a new predicate. Suppose $A_{i_1},A_{i_2},\ldots,A_{i_p}$ in the head and $A_{j_1},A_{j_2},\ldots,A_{j_p}$ in the body are variants by renaming of existential variables and these existential variables do not appear in other old atoms. Then let P_i be $(P_{i-1}-\{C\})\bigcup\{C'\}$ and D_i be D_{i-1} where C' is a general definite clause obtained by deleting these atoms from the head and the body. Let C' inherit the counter of C unless it is already in P_{i-1} with lower counter.

Example 3.2.5. By cancelling $append(M,N,LMN)$ in C_6 and $\{append(M,N,?MN), append(L,?MN,LMN)\}$ and $\{append(M,N,?MN'), append(L,?MN',LMN)\}$ in C_7', we get $P_6=\{C_1, C_2, C_8, C_9\}$ and $D_6=\{C_3\}$ where

C_8 [[1]]. p([ ],M,N,MN).
C_9 [[1]]. p([X|L],M,N,[X|LMN]) :- p(L,M,N,LMN).

3.3. Equivalence Preservation Theorem

The definite clause program P_0 given first is called the *initial program*. When the transformation process is stopped at some N and the program is transformed to a definite clause program P_N, several definitions are accumulated in D_N. Then P_N is called the *final program* and D_N is called the *definition set* of the transformation process and sometimes denoted simply by D. The set of all closed instance of the heads of clauses in $P_0\bigcup D$ is denoted by H.

Example 3.3. If we stop the transformation process at step 6, we reach the final program and the definition set

P_6 : C_1 [[1]]. append([ ],M,M).
C_2 [[1]]. append([X|L],M,[X|N]) :- append(L,M,N).
C_8 [[1]]. p([ ],M,N,LMN).
C_9 [[1]]. p([X|L],M,N,[X|LMN]) :- p(L,M,N,LMN).
D : C_3 [[1]]. append(M,N,?MN),append(L,?MN,LMN),p(L,M,N,LMN) :-
append(L,M,?LM),append(?LM,N,LMN).

The most important property being proved in Section 4 is the following theorem.

Theorem 3.3. $P_0\bigcup D$ is H-equivalent to P_N.

But in the following discussion, it is convinient to assume that all definitions in D are given from the beginning. To pretend it, for any transformation sequence $(P_0,D_0),(P_1,D_1),\ldots,(P_N,D_N)$, a sequence $S_0,S_1,\ldots,S_N$ is defined by $S_i=P_i\bigcup(D-D_i)$ and called *virtual transformation sequence*. (This is also due to Tamaki and Sato [11].) In particular $S_0=P_0\bigcup D$ and $S_N=P_N$. Since the definition rule is the identity transformation in the virtual transformation sequence, it is ignored when treating the virtual transformation sequence. Moreover, note the following three facts.

(a) For simplicity, we have restricted application of basic transformation rules to those on old atoms in the definite clauses defining new predicates. Hence the definite clauses defining old predicates in S_i are kept fixed during the transformation process and the definite clauses defining new predicates is the only changing part. We denote the former by S^{old} and the latter by S_i^{new}.

(b) Old atoms marked "inhibited" are generated when they are old atoms in the head of a definite clause in D used in folding. We have restricted that the old atoms, to which body-unfolding and folding rules are applied, are not marked "inhibited". Hence, throughout our transformation process, new atoms in the body of any definite clause in P_i always appear as a part of an instance of the head of a definite clause in D.

(c) For any transformation sequence, we can rearrange it without changing S_N such a way that cancellations are done all at once at step $N-1$ to drop all old atoms in the heads of all general definite clauses. (Moreover, we can assume without loss of generality that all body-unfoldings are applied first, all foldings next, all head-unfoldings next and all cancellations last. But we do not use this fact in the following proof.)

4. Preservation of Equivalence

4.1. Rank and Rank Ordering of Closed Molecule

Let G be a closed molecule. Then $rank(G)$, the *rank* of G, is the minimum of the size of the proof of G in S_0. Note that $rank(G) > 0$.

Example 4.1.1. Let S_0 be the program in example 3.2.2 defining *append* and *p*. Let G_0 be

{append([2],[3],?MN),append([1],?MN,[1,2,3]), p([1],[2],[3],[1,2,3]),append([ ],[1],[1])}.

Then the rank of G_0 is 7, because the following is the proof in S_0.

{append([2],[3],?MN),append([1],?MN,[1,2,3]), p([1],[2],[3],[1,2,3]),append([ ],[1],[1])}
{append([1],[2],?LM),append(?LM,[3],[1,2,3]), append([ ],[1],[1])}
{append([ ],[2],?LM'),append([1|?LM'],[3],[1,2,3]), append([ ],[1],[1])}
{append([1,2],[3],[1,2,3]),append([ ],[1],[1])}
{append([2],[3],[2,3]),append([ ],[1],[1])}
{append([ ],[3],[3]),append([ ],[1],[1])}
{append([ ],[1],[1])}

Let G_2 be

{ append([2],[3],?MN),append([ ],?MN,[2,3]), p([ ],[2],[3],[2,3]),append([ ],[1],[1]) }.

Then the rank of G_2 is 5, because the following is the proof in S_0.

{append([2],[3],?MN),append([ ],?MN,[2,3]), p([ ],[2],[3],[2,3]),append([ ],[1],[1])}
{append([ ],[2],?LM),append(?LM,[3],[2,3]), append([ ],[1],[1])}
{append([2],[3],[2,3]),append([ ],[1],[1])}
{append([ ],[3],[3]),append([ ],[1],[1])}
{append([ ],[1],[1])}

Let G_4 be

{ append([2],[3],?MN),append([ ],[1],[1]) }.

Then the rank of G_4 is 3, because the following is the proof in S_0.

{append([2],[3],?MN),append([ ],[1],[1])}
{append([ ],[3],?MN'),append([ ],[1],[1])}
{append([ ],[1],[1])}

We would like to prove that $M^*(S_N)\bigcap H = M^*(S_0)\bigcap H$ when $S_1, S_2, \ldots, S_N$ is the virtual transformation sequence. But this is too weak as an induction hypothesis when we prove it by induction.

Let $\overline{H}$ be the set of all closed molecules satisfying either of the following conditions. (Note that $\overline{H} \supseteq H$.)

(a) It consist of only closed old atoms.
(b) It contains just one ground new atom and includes a closed molecule in $\overline{H}$. That is, it is of the form $\mathbf{H} \bigcup \mathbf{O}$, where $\mathbf{H}$ is a closed instance of the head of a general definite clause in S_0 defining a new predicate and $\mathbf{O}$ is a set of closed old atoms.

Example 4.1.2. Let S_0 be the defnite clause program in Example 3.2.2. Then
H = { {append([ ],t,t)} | t is a ground term } $\bigcup$
{ {append([s|t$_1$],t$_2$,[s|t$_3$])} | s, t_1, t_2, t_3 are ground terms } $\bigcup$
{ {append(t$_2$,t$_3$,?MN), append(t$_1$,?MN,t$_{123}$)} | t_1, t_2, t_3, t_{123} are ground terms }
and the following closed molecules G_0, G_2, G_4 are all in $\overline{H}$.
{append([2],[3],?MN),append([1],?MN,[1,2,3]), p([1],[2],[3],[2,3],[1,2,3]),append([ ],[1],[1])},
{append([2],[3],?MN),append([ ],?MN,[2,3]), p([ ],[2],[3],[2,3]),append([ ],[1],[1])},
{append([2],[3],?MN),append([ ],[1],[1])}.

The *rank ordering* is a well-founded ordering $\ll$ on the set of closed molecules in $M^*(S_0) \bigcap \overline{H}$. Let F and G be two closed molecules in $M^*(S_0) \bigcap \overline{H}$. $F \ll G$ is defined by $rank(F) < rank(G)$.

Example 4.1.3. Let F and G be
{append([2],[3],?MN),append([ ],?MN,[2,3]), p([ ],[2],[3],[2,3]),append([ ],[1],[1])},
{append([2],[3],?MN),append([1],?MN,[1,2,3]), p([1],[2],[3],[1,2,3]),append([ ],[1],[1])}.
Then $F \ll G$,because $rank(F) = 5$ and $rank(G) = 7$.

4.2. Rank-Consistent Proof

Let $S_i = S^{old} \bigcup S_i^{new}$ be a general definite clause program and G_0 be a closed molecule in $M^*(S_i) \bigcap \overline{H}$ $(0 \leq i \leq N-1)$. A proof T of G_0 in S_i is said to be *rank-consistent* when it satisfies either of the following conditions.
(a) When G_0 consists of only closed old atoms, T is a rank-consistent proof of G_0 in S_i if it is the minimum proof of G_0 in S^{old}.
(b) When G_0 contains just one closed new atom A with its predicate symbol p, let G_0 be of the form $\mathbf{H}_0 \uplus \mathbf{O}_0$, where $\mathbf{H}_0$ is a close instance of the head of C_0 in D defining p. Some definite clause defining the new predicate symbol p must be applied to some closed molecule $\mathbf{H}$ containing A eventually in T. Let G_1 be the closed molecule in T of the form $\mathbf{H} \uplus \mathbf{O}$ to which such a definite clause, say C, is applied first, T_1 be the subproof of T whose root is G_1, T_2 be the immediate subproof of T_1 and G_2 be its root of the form $\mathbf{B} \uplus \mathbf{O}$. ($\mathbf{O}$ is a set of closed old atoms and "$\mathbf{H}$:- $\mathbf{B}$" is the closed instance of C by σ used at the root of T_1.) G_2 is in $M^*(S_i)$ because T_2 is its proof in S_i. Because any closed instance $\mathbf{B}$ of the body of C is in $\overline{H}$, G_2 is in $\overline{H}$, hence G_2 is in $M^*(S_i) \bigcap \overline{H}$ (except $i = N-1$ and cancellations are done at step $N-1$). Let Γ be the sum of all the counters of definite clauses applied to the closed molecules from G_0 to G_2 except G_2. T is a rank-consistent proof of G_0 in S_i when $rank(G_0) \geq rank(G_2) + \Gamma$, $G_0 \gg G_2$ and T_2 is a rank-consistent proof of G_2.

$$
\begin{array}{ll}
G_0: & \mathbf{H}_0 \uplus \mathbf{O}_0 \\
 & \downarrow \\
 & \vdots \\
 & \downarrow \\
G_1: & \mathbf{H} \uplus \mathbf{O} \\
 & \Downarrow C \\
G_2: & \mathbf{B} \uplus \mathbf{O} \\
 & \downarrow \\
 & \vdots
\end{array}
$$

Figure 2. Proof of G_0 in S_i

Example 4.2. When S_5 is the program in Example 3.2.4 before cancellation, let G_0 be
{append([2],[3],?MN),append([1],?MN,[1,2,3]), p([1],[2],[3],[1,2,3]),append([ ],[1],[1])}
as before. Then the following sequence is a proof of G_0 in S_5.

G_0 : {append([2],[3],?MN),append([1],?MN,[1,2,3]), p([1],[2],[3],[1,2,3]),append([ ],[1],[1])}
⇓ C_2 [[1]]
G_1 : {append([2],[3],?MN),append([ ],?MN,[2,3]), p([1],[2],[3],[1,2,3]),append([ ],[1],[1])}
⇓ C_7' [[1]]
G_2 : {append([2],[3],?MN'),append([ ],?MN',[2,3]), p([ ],[2],[3],[2,3]),append([ ],[1],[1])}
⇓ C_1 [[1]]
G_3 : {append([2],[3],[2,3]),p([ ],[2],[3],[2,3]), append([ ],[1],[1])}
⇓ C_6 [[1]]
G_4 : {append([2],[3],[2,3]),append([ ],[1],[1])}
⇓ C_2 [[1]]
G_5 : {append([ ],[3],[3]),append([ ],[1],[1])}
⇓ C_1 [[1]]
G_6 : {append([ ],[1],[1])}
⇓ C_1 [[1]]

It is rank-consistent,because
$\text{rank}(G_0) = 7 \geq 5 + 2 = \text{rank}(G_2) + 2,$
$\text{rank}(G_2) = 5 \geq 3 + 2 = \text{rank}(G_4) + 2.$

4.3. Proof of the Equivalence Preservation Theorem

In this section, we prove the following strengthened theorem.

Theorem 4.3. Let $S_1, S_2, \ldots, S_N$ be the virtual transformation sequence. Then $M^*(S_N) \cap \overline{H} = M^*(S_0) \cap \overline{H}$.

The proof of the theorem has a structure similar to the one by Tamaki and Sato [11] except the additional invariant I3. It consists of showing that the following invariants hold for each i ($0 \leq i \leq N$).

I1. $M^*(S_i) \cap \overline{H} = M^*(S_0) \cap \overline{H}$.
I2. For each closed molecule G in $M^*(S_i) \cap \overline{H}$, there is a rank-consistent proof of G in S_i.
I3. For any closed instance "**H** :- **B**" of a definite clause in S_i, if **B** is in $M^*(S_i)$, all old atoms in **H** are in $M^*(S^{old})$.

Base Case :

The invariant I1 trivially holds for $i = 0$. As for the invariant I2, for any closed molecule in $M^*(S_0)$ containing a new atom A, the proof of G is only one using the definition of the new predicate in D, which is obviously rank-consistent. (Remember that $S_0 = P_0 \bigcup D$ and the counters of the definite clauses in $P_0 \bigcup D$ are 1.) The invariant I3 trivially holds from the condition of the definition rule.

Induction Step :

The preservation of the invariants is proved in the four lemmas below.

Lemma 4.3.1. If the invariants I1 and I3 hold for S_i, then $M^*(S_{i+1}) \bigcap \overline{H} \subseteq M^*(S_i) \bigcap \overline{H}$.

Proof. Let G_0 be a closed molecule in $M^*(S_{i+1}) \bigcap \overline{H}$. When G_0 consists of only closed old atoms, the lemma is trivial. When G_0 contains just one ground new atom A with its predicate symbol p, let G_0 be of the form $\mathbf{H}_0 \uplus \mathbf{O}_0$, where $\mathbf{H}_0$ is a closed instance of the head of C_0 in D defining p, and T be a proof of G_0 in S_{i+1}. We construct a proof T' of G_0 in S_i by induction on the structure of T.

Some definite clause defining the new predicate symbol p must be applied to some closed molecule $\mathbf{H}$ containing A eventually in T. Let G_1 be the closed molecule in T of the form $\mathbf{H} \uplus \mathbf{O}$ to which such a definite clause, say C, is applied first, T_1 be the subproof of T whose root is G_1, T_2 be the immediate subproof of T_1 and G_2 be its root of the form $\mathbf{B} \uplus \mathbf{O}$. ($\mathbf{O}$ is a set of closed old atoms and "$\mathbf{H}$:- $\mathbf{B}$" is the closed instance of C by σ used at the root of T_1.) G_2 is in $M^*(S_{i+1})$ because T_2 is its proof in S_{i+1}. Because any closed instance $\mathbf{B}$ of the body of C is in $\overline{H}$, G_2 is in $\overline{H}$, hence G_2 is in $M^*(S_{i+1}) \bigcap \overline{H}$ (except $i = N-1$ and cancellations are done at step $N-1$). By induction hypothesis, we can construct a proof T'_2 of G_2 in S_i (except $i = N-1$ and cancellations are done at step $N-1$ again). If C is in S_i, we can immediately construct T' from C and the proof T'_2. When C is the result of applying one of the transformation rules to C' in S_i, we prove the lemma by case analysis.

$$
\begin{array}{ll}
\mathrm{G}_0: & \mathbf{H}_0 \uplus \mathbf{O}_0 \\
 & \downarrow \\
 & \vdots \\
 & \downarrow \\
\mathrm{G}_1: & \mathbf{H} \uplus \mathbf{O} \\
 & \Downarrow \mathrm{C} \\
\mathrm{G}_2: & \mathbf{B} \uplus \mathbf{O} \\
 & \downarrow \\
 & \vdots
\end{array}
$$

Figure 3. Proof of G_0 in S_{i+1}

Suppose C is the result of body-unfolding. Then there is a closed instance "$\mathbf{H}$:- $\mathbf{B}'$" of C' in S_i and some A_j in $\mathbf{B}'$, say A_1, is a closed instance of the atom in the body of C' to which the body-unfolding was applied with a definite clause C_{unfold}. Let T'_0 be the sequence from G_0 to G_1, T'_1 be the two-step sequence from G_1 to G_2 with C' and C_{unfold}, T'_2 be the proof of G_2 in S_i and T' be the concatenation of T'_0, T'_1 and T'_2. Then T' is the proof of G_0 in S_i.

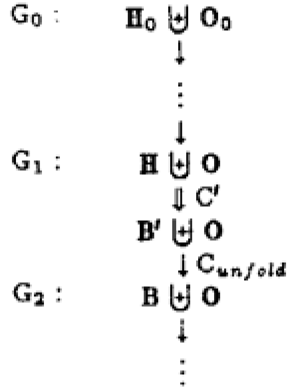

Figure 4. Proof of G_0 in S_i (Case of Body-Unfolding)

Suppose C is the result of head-unfolding. Then there is a closed instance $\mathbf{H}'$:- $\mathbf{B}$ of C' in S_i and some A_j in $\mathbf{H}'$, say A_1, is a closed instance of the atom in the head of C' to which the head-unfolding was applied with a definite clause C_{unfold}. By the invariant I1 for i, G_2 is in $M^*(S_0) \cap H$. Because all bodies of the definite clauses in S_0 contain only old atoms, $\mathbf{B}$ is in $M^*(S_0) \cap H$ by itself and old atoms in $\mathbf{O}$ are all in $M^*(S^{old})$. By the invariant I1 again, $\mathbf{B}$ is in $M^*(S_i) \cap H$. By the invariant I3 for i, all old atoms in $\mathbf{H}'$ is in $M^*(S^{old})$. By the condition of head-unfolding, all old atoms in $\mathbf{H}$ are in $M^*(S^{old})$ iff all old atoms in $\mathbf{H}'$ are in $M^*(S^{old})$. Hence all old atoms in G_1 are in $M^*(S^{old})$. Because all old atoms in G_0 are reduced to old atoms in G_1 in the sequence from G_0 to G_1, all old atoms in G_0 are in $M^*(S^{old})$ and $\mathbf{O}_0$ has a proof T'_{O_0} in S_i. Let T'_0 be the sequence from $\mathbf{H}_0$ to $\mathbf{H}'$ in which reductions are applied in the same way as head-unfoldings are applied in derivation of C from C_0 except the head-unfolding from C' to C, T'_1 be the one-step sequence from $\mathbf{H}'$ to $\mathbf{B}$ with C', T'_2 be the proof of $\mathbf{B}$ in S_i and T'_{H_0} be the concatination of T'_0, T'_1, T'_2. Then T'_{H_0} is a proof of $\mathbf{H}_0$. Let T' be any interleaving of T'_{H_0} and T'_{O_0}. Then T' is the proof of G_0 in S_i.

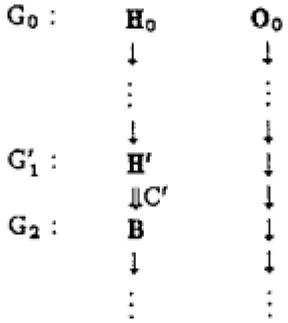

Figure 5. Proof of G_0 in S_i (Case of Head-Unfolding)

Suppose C is the result of folding. Then there is a closed instance "$\mathbf{H}$:- $\mathbf{B}'$" of C' in S_i and some closed atoms in $\mathbf{B}$, say $\mathbf{H}_{fold}$, is an instance of the atoms in the body of C introduced by the folding. By the invariant I1, G_2 is in $M^*(S_0) \cap H$. So there should be a closed instance $\mathbf{H}_{fold}$:- $\mathbf{B}_{fold}$ of some general definite clause in D such that $\mathbf{B}_{fold}$ are closed old atoms in $M^*(S^{old})$. In addition, $G_2 - \mathbf{H}_{fold}$ are all in $M^*(S^{old})$. Let T'_0 be the sequence from G_0 to G_1, T'_1 be the one-step sequence from G_1 to $(G_2 - \mathbf{H}_{fold}) \uplus \mathbf{B}_{fold}$ with C', T'_2 be the proof of $(G_2 - \mathbf{H}_{fold}) \uplus \mathbf{B}_{fold}$ and T' be the concatination of T'_0, T'_1 and T'_2. Owing to the condition of folding, T' is the proof of G_0 in S_i.

$$
\begin{array}{ll}
\mathrm{G}_0: & \mathbf{H}_0 \uplus \mathbf{O}_0 \\
 & \downarrow \\
 & \vdots \\
 & \downarrow \\
\mathrm{G}_1: & \mathbf{H} \uplus \mathbf{O} \\
 & \Downarrow C' \\
\mathrm{G}_2': & (\mathrm{G}_2 - \mathbf{H}_{fold}) \uplus \mathbf{B}_{fold} \\
 & \downarrow \\
 & \vdots
\end{array}
$$

Figure 6. Proof of G_0 in S_i (Case of Folding)

Suppose C is the result of cancellation. As was noted before, cancellations in our transformation process are applied all at once at step $N-1$. Then there is a closed instance $\mathbf{O}_2 \uplus \{A\}$:- $\mathbf{O}_2 \uplus \mathbf{B}$ of C' in S_{N-1}. Because all definite clause in S_N are explicit, all closed atoms in G_0 are provable in S_N independently. Because of the condition of head-unfolding, there is a proof of closed old atoms in $\mathbf{H}_0$ such that some closed molecule in it is of the form $\mathbf{O}_2$. Hence there is a proof of $\mathbf{H}_0$ such that some closed molecule in it is $\mathbf{O}_2 \uplus \{A\}$. Similarly, $\mathbf{O}_2 \uplus \mathbf{B}$ is provable in S_N by itself, hence provable in S_{N-1} by induction hypothesis. Let T_0 be the sequence from $\mathbf{H}_0$ to $\mathbf{O}_2 \uplus \mathbf{B}$ in which reductions are applied in the same way as head-unfoldings are applied in derivation of C from C_0, T_1 be the one-step sequence from $\mathbf{O}_2 \uplus \{A\}$ to $\mathbf{O}_2 \uplus \mathbf{B}$ with C', T_2' be the proof of $\mathbf{O}_2 \uplus \mathbf{B}$ in S_{N-1} and T'_{H_0} be the concatination of T_1', T_2' and T_3'. Then T'_{H_0} is a proof of $\mathbf{H}_0$. In addition, $\mathbf{O}_0$ has a proof T'_{O_0} in S_{N-1}. Let T' be any interleaving of T'_{H_0} and T'_{O_0}. Then T' is the proof of G_0 in S_{N-1}.

$$
\begin{array}{lcc}
\mathrm{G}_0: & \mathbf{H}_0 & \mathbf{O}_0 \\
 & \downarrow & \downarrow \\
 & \vdots & \vdots \\
 & \downarrow & \downarrow \\
\mathrm{G}_1: & \mathbf{O}_2 \uplus \{A\} & \downarrow \\
 & \Downarrow C' & \downarrow \\
\mathrm{G}_2: & \mathbf{O}_2 \uplus \mathbf{B} & \downarrow \\
 & \downarrow & \downarrow \\
 & \vdots & \vdots
\end{array}
$$

Figure 7. Proof of G_0 in S_i (Case of Cancellation)

Lemma 4.3.2. If the invariants I1 and I2 hold for S_i, then $M^*(S_i) \cap \overline{H} \subseteq M^*(S_{i+1}) \cap \overline{H}$.

Proof. Let G_0 be a closed molecule in $M^*(S_i) \cap \overline{H}$. Then by the invariant I2, there is a rank-consistent proof T of G_0 in S_i. We construct a proof T' of G_0 in S_{i+1} by induction on the well-founded ordering $\gg$.

The base case where G_0 is provable in S_0 itself and consists of only one closed old atom A obviously holds, because then A should be a closed instance of some unit clause in P_0 which should be in both S_i and S_{i+1}.

As for induction step, when G_0 consists of only closed old atoms, the lemma is trivial. When G_0 contains just one ground new atom A with its predicate symbol p, let G_0 be of the form $\mathbf{H}_0 \uplus \mathbf{O}_0$ and T be a proof of G_0 in S_i, where $\mathbf{H}_0$ is a closed instance of the head of C_0 in D defining p. Some definite clause defining p must be applied to some closed molecule $\mathbf{H}$ containing A eventually in T. Let G_1 be the closed atom in T of the form $\mathbf{H} \uplus \mathbf{O}$ to which such a definite clause, say C, is applied first and T_1 be the subproof of T whose root is G_1. ($\mathbf{O}$ is a set of closed old atoms and "$\mathbf{H}$:- $\mathbf{B}$" is the closed instance of C used at the root of T_1.) Let T_2 be the immediate subproof of T_1 and G_2 be its root of the form $\mathbf{B} \uplus \mathbf{O}$. G_2 is in $M^*(S_i)$ because T_2 is its proof in S_i. Because any closed instance $\mathbf{B}$ of the body of C is in $\overline{H}$, G_2 is in $\overline{H}$, hence G_2 is in $M^*(S_i) \bigcap \overline{H}$. By the invariant I2, $G_0 \gg G_2$ holds. So by the induction hypothesis there is a proof T_2' of G_2 in S_{i+1}. If C is in S_{i+1},the construction of T' is immediate. When C is a defnite caluse to which one of the transformation rules is applied to result in C' in S_{i+1}, we prove the lemma by case analysis.

$$
\begin{array}{ll}
G_0 : & \mathbf{H}_0 \uplus \mathbf{O}_0 \\
 & \downarrow \\
 & \vdots \\
 & \downarrow \\
G_1 : & \mathbf{H} \uplus \mathbf{O} \\
 & \Downarrow C \\
G_2 : & \mathbf{B} \uplus \mathbf{O} \\
 & \downarrow \\
 & \vdots
\end{array}
$$

Figure 8. Proof of G_0 in S_i

Suppose C is body-unfolded into $C_1', C_2', \ldots, C_k'$ in S_{i+1} and assume that some A_j in $\mathbf{B}$, say A_1, is the closed instance of the old atom at which C is unfolded. Let $\mathbf{O}'$ be the closed old atoms to which A_1 is unfolded and which is a part of the body of a closed instance of C_l'. By the invariant I1, G_2 is in $M^*(S_0) \bigcap \overline{H}$. Let G_2' be $(G_2 - \{A_1\}) \uplus \mathbf{O}'$. Because the unfolded atom is not marked "inhibited", G_2' is also in $M^*(S_0) \bigcap \overline{H}$, hence in $M^*(S_i) \bigcap \overline{H}$. In addition, because $G_0 \gg G_2'$ holds, G_2' is in $M^*(S_{i+1}) \bigcap \overline{H}$ by induction hypothesis. Let T_0' be the sequence from G_0 to G_1 in T, T_1' be the one-step sequence from G_1 to G_2' with C_l' and T_2' be the proof of G_2' in S_{i+1}. Concatenating T_0', T_1' and T_2', we obtain the proof T' of G_0 in S_{i+1}.

$$
\begin{array}{ll}
G_0 : & \mathbf{H}_0 \uplus \mathbf{O}_0 \\
 & \downarrow \\
 & \vdots \\
 & \downarrow \\
G_1 : & \mathbf{H} \uplus \mathbf{O} \\
 & \Downarrow C' \\
G_2' : & (G_2 - \{A_1\}) \uplus \mathbf{O}' \\
 & \downarrow \\
 & \vdots
\end{array}
$$

Figure 9. Proof of G_0 in S_{i+1} (Case of Body-Unfolding)

Suppose C is head-unfolded into C' in S_{i+1} and assume that some A_j in $\mathbf{H}$, say A_1, is the instance of the atom at which C is head-unfolded by using a definite clause C_{unfold}.

Let T'_0 be the sequence from G_0 to G_1 in T, T'_1 be the two-step sequence from G_1 to G_2 by using C_{unfold} and C' and T'_2 be the proof of G_2 in S_{i+1}. Concatenating T'_0, T'_1, T'_2, we obtain the proof T' of G_0 in S_{i+1}.

$$\begin{array}{ll}
G_0 : & \mathbf{H}_0 \uplus \mathbf{O}_0 \\
 & \quad\downarrow \\
 & \quad\vdots \\
 & \quad\downarrow \\
G_1 : & \mathbf{H} \uplus \mathbf{O} \\
 & \quad\downarrow C_{unfold} \\
 & \mathbf{H}' \uplus \mathbf{O} \\
 & \quad\Downarrow C' \\
G_2 : & \mathbf{B} \uplus \mathbf{O} \\
 & \quad\downarrow \\
 & \quad\vdots
\end{array}$$

Figure 10. Proof of G_0 in S_{i+1} (Case of Head-Unfolding)

Suppose C is folded into C' in S_{i+1}. Assume that k closed atoms $\mathbf{B}_{fold}$ in G_2 is the closed instance of the folded atoms in C. Let $\mathbf{H}'$ be a closed molecule such that $\mathbf{H}_{fold}$:- $\mathbf{B}_{fold}$ is a closed instance of the general definite clause in D used in the folding. Let G'_2 be $(G_2 - \mathbf{B}_{fold}) \uplus \mathbf{H}_{fold}$. By definition, $rank(G_2) + 1 \geq rank(G'_2)$. By the condition (d) of folding,

$\mathrm{rank}(G_0) \geq \mathrm{rank}(G_2) + \gamma \geq (\mathrm{rank}(G'_2) - 1) + (n - k) + \gamma > \mathrm{rank}(G'_2)$,

which means $G_0 \gg G'_2$ holds. (Actually, we can make the condition (d) much weaker. See [??].) Moreover, by the invariant I1, G'_2 is in $M^*(S_i) \bigcap \overline{H}$. Therefore by the induction hypothesis, G'_2 has a proof T'_2 in S_{i+1}. Let T'_0 be the sequence from G_0 to G_1 in T and T'_1 be the sequence from G_1 to G'_2 with C'. Concatenating T'_0, T'_1 and T'_2 with the definite clause C', we obtain the proof T' of G_0 in S_{i+1}.

$$\begin{array}{ll}
G_0 : & \mathbf{H}_0 \uplus \mathbf{O}_0 \\
 & \quad\downarrow \\
 & \quad\vdots \\
 & \quad\downarrow \\
G_1 : & \mathbf{H} \uplus \mathbf{O} \\
 & \quad\Downarrow C' \\
G'_2 : & (G_2 - \mathbf{B}_{fold}) \uplus \mathbf{H}_{fold} \\
 & \quad\downarrow \\
 & \quad\vdots
\end{array}$$

Figure 11. Proof of G_0 in S_{i+1} (Case of Folding)

Suppose C is cancelled to an explicit definite clause C'. Then we can reduce G_1 to G_2 by using C' itself. Let T'_0 be the sequence from G_0 to G_1 in T, T'_1 be the one-step sequence from G_1 to G_2 with C' and T'_2 be the proof of G_2 in $M^*(S_{i+1})$. Concatenating T'_0, T'_1 and T'_2 with the definite clause C', we obtain the proof T' of G_0 in S_{i+1}.

$$
\begin{array}{ll}
G_0 : & \mathbf{H}_0 \uplus \mathbf{O}_0 \\
 & \downarrow \\
 & \vdots \\
 & \downarrow \\
G_1 : & \mathbf{H} \uplus \mathbf{O} \\
 & \Downarrow C' \\
G_2 : & \mathbf{B} \uplus \mathbf{O} \\
 & \downarrow \\
 & \vdots
\end{array}
$$

Figure 12. Proof of G_0 in S_{i+1} (Case of Cancellation)

Lemma 4.3.3. If the invariants I1, I2 and I3 hold for S_i, then I2 holds for S_{i+1}.

Proof. We first note that in the proof of lemma 4.3.2, T' is constructed in such a way that it is rank-consistent. Thus every atom in $M^*(S_i)$ has a rank-consistent proof in S_{i+1}. Because $M^*(S_{i+1}) \subseteq M^*(S_i)$ by lemma 4.3.1, I2 holds for S_{i+1}.

Lemma 4.3.4. If the invariants I1 and I3 hold for S_i, then I3 holds for S_{i+1}.

Proof. Let "$\mathbf{H}$:- $\mathbf{B}$" be a closed instance of a definite clause C in S_{i+1}. Suppose $\mathbf{B}$ is provable in S_{i+1}. When C is in S_i, the lemma is obvious from the invariant I3 for S_i. When C is the result of applying one of the transformation rules to C' in S_i, we prove the lemma by case analysis.

Suppose C is the result of body-unfolding. Then there is a closed instance "$\mathbf{H}$:- $\mathbf{B}'$" of C'. Then obviously $\mathbf{B}'$ is provable in S_{i+1}. Moreover it is provable in S_i from lemma 4.3.1. By the invariant I3 for S_i, old atoms in $\mathbf{H}$ are in $M^*(S^{old})$.

Suppose C is the result of head-unfolding. Then there is a closed instance $\mathbf{H}'$:- $\mathbf{B}$ of C'. Because of the condition of head-unfolding, all old atoms in $\mathbf{H}$ are in $M^*(S^{old})$ iff all old atoms in $\mathbf{H}'$ are in $M^*(S^{old})$. Hence old atoms in $\mathbf{H}$ are all in $M^*(S^{old})$.

Suppose C is the result of folding. Then there is a closed instance "$\mathbf{H}$:- $\mathbf{B}'$" of C'. Because of lemma 4.3.1 and the invariant I1 for S_i, $\mathbf{B}'$ is also in $M^*(S_i)$. By the invariant I3 for S_i, old atoms in $\mathbf{H}$ are all in $M^*(S^{old})$.

Suppose C is the result of cancellation. As was noted before, cancellations in our transformation process are applied all at once at step $N-1$. Because all definite clauses in S_N are explicit, there is no old atoms in heads. Hence I3 for N is vacantly true.

This completes the proof of the theorem.

5. Discussion

Our work stemed from the work by *forced folding* by Clark and Darlington [2], Darlington [3] and *(folding driven) goal insertion* by Tamaki and Sato [11]. Though our definition style is superficially similar to *expression procedure* by Sherlis [9],[10], its use is completely different.

Our proof in Section 4 is limited in two respects.

(a) We assumed that folding is applied at most once to each definite clause defining a new predicate, hence bodies of each definite clause contains at most one new atom. This limitation has simplified the proof drastically. We conjecture that our equivalence preservation theorem still holds even if folding is applied more than once.

(b) We restricted that old atoms, to which body-unfolding and folding are applied, are not marked "inhibited", hence new atoms in the bodies of each definite clause always appear as an instance of the head of a definite clause in D. We can easily coin a counter example in which S_i and S_{i+1} are not equivalent if this condition is not observed. But, we conjecture that our equivalence preservation theorem still holds even if "inhibited" marks are ignored and the equivalence might be lost on the way $S_1, S_2, \ldots, S_{N-1}$.

6. Conclusions

We have presented a method to derive Prolog programs from implicit specifications. This method is being used in Argus/C, a system for construction of Prolog programs under development [5],[6].

Acknowledgements

This work is based on the result by Tamaki and Sato [11]. The authors would like to express deep gratitude to Prof.H.Tamaki (Ibaraki University) and Dr.T.Sato (Electrotechnical Laboratory) for their stimulating and perspicuous works.

Our construction system Argus/C under development is a subproject of the Fifth Generation Computer System(FGCS) "Intelligent Programming System". The authors would like to thank Dr.K.Fuchi (Director of ICOT) for the opportunity of doing this research and Dr.T.Yokoi (Chief of ICOT 2nd Laboratory) and Dr.K.Furukawa (Chief of ICOT 1st Laboratory) for their advice and encouragement.

References

[1] Apt,K.R. and M.H.van Emden, "Contribution to the Theory of Logic Programming", J.ACM, Vol.29, No.3, pp.841-862, 1982.

[2] Clark,K.L.and J.Darlington, "Algorithm Classification Through Synthesis", The Computer Journal, Vol.23, No.1, pp.61-65,1980.

[3] Darlington,J., "An Experimental Program Transformation and Synthesis System", Artificial Intelligence, Vol.16, pp.1-46, 1981.

[4] van Emden,M.H. and R.A.Kowalski, "The Semantics of Predicate Logic as Programing Language", J.ACM, Vol.23, No.4, pp.733-742, 1976.

[5] Kanamori,T.and K.Horiuchi, "Construction of Logic Programs Based on Generalized Unfold/Fold Rules", ICOT TR-1??, to appear, 1986.

[6] Kanamori,T.and H.Fujita, "Unfold/Fold Tramsformation of Logic Programs with Counters", ICOT TR-1??, to appear, 1986.

[7] Lloyd,J.W., "Foundations of Logic Programming", TR 82/7,Department of Computer Science, University of Melbourne, 1982.

[8] Pereira,L.M.,F.C.N.Pereira and D.H.D.Warren, "User's Guide to DECsystem-10 Prolog", Occasional Paper 15, Dept.of Artificial Intelligence, Edinburgh, 1979.

[9] Scherlis,W.L., "Expression Procedures and Program Derivation", STAN-CS-80-818,Stanford University, Computer Science Department, 1980.

[10] Scherlis,W.L., "Program Improvement by Internal Specialization", Proc.of 8th Symposium of Princilpies of Programming Languages, pp.41-49, 1981.

[11] Tamaki,H.and T.Sato, "Unfold/Fold Transformation of Logic Programs", Proc.of 2nd International Logic Programming Conference, pp.127-138,1984.

[12] Tamaki,H.and T.Sato, "Compatibility of Replacement Rules with Unfold/Fold Transformation", to appear, 1986.