The immediate consequence operator and Robinson's operator

Miguel Angel Gutiérrez Naranjo

Abstract

This paper discusses the relation between Robinson's operator (R_P^n) and van Emden and Kowalski's immediate consequence operator, in ascendant form ($T_P \uparrow$) as well as in descendant form ($T_P \downarrow$). Our main result proves that $T_P \downarrow$ and $T_{R_P^n} \downarrow$ coincide in each level k, i.e., $T_P \downarrow k = T_{R_P^n} \downarrow k$ with independence of the n we choose.

Keywords: Immediate consequence operator, Robinson's operator, Logic programming

1 Introduction

Since van Emden and Kowalski presented their immediate consequence operator [8] in 1976, it has become the paradigm of the semantic interpretation of Logic Programming. Eleven years before, J. A. Robinson [7] presented resolution as proof procedure and the so–called Robinson's operator R_P^n. Given a program P, this operator allows us to get another program P' obtained by adding to P all the binary resolvents between clauses of P. This paper discusses the relation between both operators, considering the immediate consequence operator in ascendant form as well as in descendant form. With respect to the ascendant operator, theorem 3.1 shows that if a ground atom A is a logical consequence of P, then we can find a natural number n such that for all $m \geq n$, the fact $A \leftarrow$ is a ground instance of a fact of R_P^n.
Our main result is theorem 4.8, where we prove that operators $T_P \downarrow$ and $T_{R_P^n} \downarrow$ coincide, not only in the limit as it happens with $T \uparrow$ (Th. 3.2), but in each level k, and we prove that the result holds for all n.

2 Definitions and notation

In the sequel we will consider some fixed first–order language with a finite number of function symbols (including constants) and a finite number of predicate symbols. For most definitions from logic, we refer to [1, 3, 6]. Let us recall the following:
Terms are defined inductively as follows: constants and variables are terms, and if $t_1, \ldots, t_n$ are terms and f is a function symbol of arity n, then $f(t_1, \ldots, t_n)$ is a term. An

Dpto. Ciencias de la Computación e Inteligencia Artificial, Facultad de Matemáticas, Universidad de Sevilla, C/ Tarfia s/n 41012 Sevilla, Spain Tel: +34 954556946 E–mail: magutier@cica.es

atom has the form $p(t_1, \ldots, t_n)$ where p is an n–ary predicate symbol and $t_1, \ldots, t_n$ are terms.
A *definite program clause* (a *clause* in the following) is a formula of the form $\forall x_1 \ldots \forall x_s$ $(A \vee \neg A_1 \vee \ldots \vee \neg A_n)$ where A and $A_1, \ldots, A_n$ are atoms, $n \geq 0$, and $x_1, \ldots, x_s$ are all the variables occuring in $A \vee \neg A_1 \vee \ldots \vee A_n$. We will denote that clause by $A \leftarrow A_1, \ldots, A_n$ as usual. A term or a clause is *ground* if it does not contain any variables.
A *substitution* is a finite mapping from variables to terms, and is written as $\theta = \{x_1/t_1, \ldots x_s/t_s\}$. If all $t_1, \ldots, t_s$ are ground, then θ is called *ground*. If for a substitution θ we have $A\theta = B\theta$, then θ is called a *unifier* of A and B. A unifier θ is called a *most general unifier* (or *mgu* in short) if it is more general than any other unifier of A and B. A *definite program* is a set of clauses[1]. If the clause C is the binary resolvent of the clauses C_1 and C_2, we will write $C = C_1 \cdot C_2$
The *Herbrand base* of a program P, B_P, is the set of all ground atoms which can be formed out of the symbols occuring in P. As usual, M_P is the least Herbrand model of P, i.e., the intersection of all Herbrand models for P, and 2^{B_P} is the power set of B_P.
We also recall the definitions of Robinson's operator [7] and van Emden and Kowalski's immediate consequence operator [8].

Definition 2.1 *Let P be a definite program and let $\mathcal{R}_P$ be the set*

$$\mathcal{R}_P = \{C : \exists C_1, C_2 \in P\,(C = C_1 \cdot C_2)\}$$

Robinson's operator is defined recursively as follows

$$\begin{array}{l} R_P^0 = P \\ R_P^{n+1} = R_P^n \cup \mathcal{R}_{R_P^n} \end{array}$$

Definition 2.2 *Let P be a definite program. We call the immediate consequence operator to the mapping*

$$T_P : 2^{B_P} \rightarrow 2^{B_P}$$

defined as follows: $\forall I \subseteq B_P$ $T_P(I) = \{A \in B_P : A \leftarrow A_1, \ldots, A_n$ is a ground instance of a clause in P and $\{A_1, \ldots, A_n\} \subseteq I\}$.

Obviously T_P is monotonic, i.e., if $X \subseteq Y$ then $T_P(X) \subseteq T_P(Y)$.
The next definition is adapted from [3].

Definition 2.3 *Let P be a definite program and T_P its immediate consequence operator. The following mappings are defined*

$$\begin{array}{llll} T_P\uparrow: & \mathcal{N} & \longrightarrow & 2^{B_P} \\ & n & \mapsto & T_P \uparrow n = \left\{ \begin{array}{ll} \emptyset & \text{if } n = 0 \\ T_P(T_P \uparrow (n-1)) & \text{if } n > 0 \end{array} \right. \end{array}$$

$$\begin{array}{llll} T_P\downarrow: & \mathcal{N} & \longrightarrow & 2^{B_P} \\ & n & \mapsto & T_P \downarrow n = \left\{ \begin{array}{ll} B_P & \text{if } n = 0 \\ T_P(T_P \downarrow (n-1)) & \text{if } n > 0 \end{array} \right. \end{array}$$

We also consider

$$\begin{array}{l} T_P \uparrow \omega = \cup\{T_P \uparrow k : k \geq 0\} \\ T_P \downarrow \omega = \cap\{T_P \downarrow k : k \geq 0\} \end{array}$$

[1]A definite program is usually considered as a *finite* set of clauses. We permit *infinite* sets for our purpose.

We finish this section with a classic theorem in Logic Programming (Th 6.2 and Th. 6.5 in [3]).

Theorem 2.4 *Let P be a definite program. Then* $T_P \uparrow \omega = M_P = \{A \in B_P : P \models A\}$

3 The ascendant operator

The next theorem shows that if a ground atom A is a logical consequence of P, i.e., $(\exists k)[A \in T_P \uparrow k]$ (Th. 2.4), then we can find a natural number n such that for all $m \geq n$, the fact $A \leftarrow$ is a ground instance of a fact of R_P^n.

Theorem 3.1 *Let P be a definite program and let l be the maximum number of literals in the body of a clause of P. Then, given $k \geq 1$, there is n such that*

- $T_P \uparrow k \subseteq T_{R_P^n} \uparrow 1$
- $0 \leq n \leq (k-1)l$

Proof. The proof is by induction on k
$(k = 1)$ We only must consider $n = 0$
$(k \to k+1)$ Suppose the theorem holds for k, i.e., there is n' with $0 \leq n' \leq (k-1)l$ such that $T_P \uparrow k \subseteq T_{R_P^{n'}} \uparrow 1$, i.e., for all $A \in T_P \uparrow k$, $A \leftarrow$ is a ground instance of a clause of $R_P^{n'}$. Assume $B \in T_P \uparrow (k+1)$. Then we can find a ground instance $C \equiv B \leftarrow A_1, \ldots, A_s$ of a clause of P (and hence, a clause of $R_P^{n'}$) such that $\{A_1, \ldots, A_s\} \subseteq T_P \uparrow k$. So we have that $B \leftarrow A_1, \ldots, A_s$ and $A_1 \leftarrow, \ldots, A_s \leftarrow$ are ground instances of clauses of $R_P^{n'}$ and therefore

$$B \leftarrow \equiv (((\ldots((C \cdot A_1 \leftarrow) \cdot A_2 \leftarrow) \ldots) \cdot A_s \leftarrow) \in R_P^{n'+s}$$

Then if we take $n = n' + s$, we have $B \in T_{R_P^n} \uparrow 1$ and $s \leq l$. Therefore $0 \leq n \leq kl$. □

The next theorem shows that, in the limit, we have the equality.

Theorem 3.2 *Let P be a definite program. Then, for all n,* $T_P \uparrow \omega = T_{R_P^n} \uparrow \omega$

Proof. Since P and R_P^n are logically equivalent, for all n, we have
$T_P \uparrow \omega = M_P = \cap\{I \subseteq B_P : I \models P\} = \cap\{I \subseteq B_P : I \models R_P^n\} = M_{R_P^n} = T_{R_P^n} \uparrow \omega$ □

4 The descendant operator

Our following goal is to study the relation between $T_P \downarrow$ and Robinson's operator. The main result is the theorem 4.8. It shows that for all k, $T_{R_P^n} \downarrow k = T_P \downarrow k$ with independence of the n we choose. We first prove the natural inclusion.

Theorem 4.1 *Let P be a definite program. Then* $\forall k \geq 0$, $\forall n \geq 0$, $T_P \downarrow k \subseteq T_{R_P^n} \downarrow k$

Proof. By induction on k:
$(k = 0)$ Trivial, since the Herbrand base of P and R_P^n are the same: $T_P \downarrow 0 = B_P = T_{R_P^n} \downarrow 0$
$(k \to k+1)$ Suppose the result holds for k, and assume $A \in T_P \downarrow (k+1) = T_P(T_P \downarrow k)$, i.e., there is a ground instance $A \leftarrow B_1, \ldots, B_s$ of a clause of P such that $\{B_1, \ldots, B_s\} \subseteq$

$T_P \downarrow k$. By induction hypothesis, $T_P \downarrow k \subseteq T_{R_P^n} \downarrow k$ $(\forall n \geq 0)$ and trivially $P \subseteq R_P^n$ $(\forall n \geq 0)$ Then, given $n \geq 0$, we have, on the one hand, $\{B_1, \ldots, B_s\} \subseteq T_{R_P^n} \downarrow k$ and on the other hand $A \leftarrow B_1, \ldots, B_s$ is a ground instance of a clause of R_P^n. Hence $A \in T_{R_P^n}(T_{R_P^n} \downarrow k) = T_{R_P^n} \downarrow (k+1)$ $\square$

For the proof of the other inclusion we need some previous results.

Definition 4.2 *Let P be a definite program. We denote by $[P]$ the set of all ground instances of clauses of P.*

Lemma 4.3 *Let P be a definite program. Then $\forall I \subseteq B_P$, $T_P(I) = T_{[P]}(I)$*

Proof. $A \in T_P(I)$ if and only if there is a ground instance of a clause of P, $A \leftarrow B_1, \ldots, B_s$ $(s \geq 0)$ such that $\{B_1, \ldots, B_s\} \subseteq I$ and it happens if and only if $A \in T_{[P]}(I)$ $\square$

Corollary 4.4 *Let P be a definite program. Then $\forall k \geq 0$ $\quad T_P \downarrow k = T_{[P]} \downarrow k$*

Proof. By induction on k.
$(k = 0)$ $T_P \downarrow 0 = B_P = T_{[P]} \downarrow 0$
$(k \to k+1)$ By the induction hypothesis we have $T_P \downarrow k = T_{[P]} \downarrow k$, then $T_P \downarrow (k+1) = T_P(T_P \downarrow k) = T_P(T_{[P]} \downarrow k)$. Also, by lemma 4.3 $T_P(T_{[P]} \downarrow k) = T_{[P]}(T_{[P]} \downarrow k)$. Therefore $T_P \downarrow (k+1) = T_P(T_P \downarrow k) = T_P(T_{[P]} \downarrow k) = T_{[P]}(T_{[P]} \downarrow k) = T_{[P]} \downarrow (k+1)$ $\square$

Lemma 4.5 *Let P be a definite program. Then $\forall n \geq 0$ $\quad [R_P^n] = R_{[P]}^n$*

Proof. We are going to prove the lemma by induction on n.
$(n = 0)$ $[R_P^0] = [P] = R_{[P]}^0$
$(n-1 \to n)$ We prove the equality by double inclusion
(A) We first prove $[R_P^n] \subseteq R_{[P]}^n$
Suppose $C \in [R_P^n] = [R_P^{n-1} \cup \mathcal{R}_{R_P^{n-1}}] = [R_P^{n-1}] \cup [\mathcal{R}_{R_P^{n-1}}]$ We have to prove $C \in R_{[P]}^n$. We consider two cases:
Case 1: $C \in [R_P^{n-1}]$
In this case, by induction hypothesis, we have $[R_P^{n-1}] = R_{[P]}^{n-1}$ and trivially, $R_{[P]}^{n-1} \subseteq R_{[P]}^n$, then $C \in [R_P^{n-1}] = R_{[P]}^{n-1} \subseteq R_{[P]}^n$
Case 2: $C \in [\mathcal{R}_{R_P^{n-1}}]$
Consider $C_0 \in \mathcal{R}_{R_P^{n-1}}$ such that there is a ground substitution θ such that $C_0\theta = C$. As $C_0 \in \mathcal{R}_{R_P^{n-1}}$, there are $C_1, C_2 \in R_P^{n-1}$ such that $C_0 = C_1 \cdot C_2$. Assume $C_1 \equiv A \leftarrow A_1, \ldots, A_s$ and $C_2 \equiv B \leftarrow B_1, \ldots, B_t$ and let σ be a mgu of A_i and B, i.e., $A_i\sigma = B\sigma$ and hence C_0 is the clause $A\sigma \leftarrow A_1\sigma, \ldots, A_{i-1}\sigma, B_1\sigma, \ldots, B_t\sigma, A_{i+1}\sigma, \ldots, A_s\sigma$ and C is the clause $A\sigma\theta \leftarrow A_1\sigma\theta, \ldots, A_{i-1}\sigma\theta, B_1\sigma\theta, \ldots, B_t\sigma\theta, A_{i+1}\sigma\theta, \ldots, A_s\sigma\theta$. Let γ be a substitution that make $A_i\sigma\theta$ ground, i.e., a substitution such that $A_i\sigma\theta\gamma = B\sigma\theta\gamma$ is a ground atom. Let C_1' and C_2' be the clauses

$$C_1' \equiv A\sigma\theta \leftarrow A_1\sigma\theta, \ldots, A_{i-1}\sigma\theta, A_i\sigma\theta\gamma, A_{i+1}\sigma\theta, \ldots, A_s\sigma\theta$$

and

$$C_2' \equiv B\sigma\theta\gamma \leftarrow B_1\sigma\theta, \ldots, B_t\sigma\theta$$

We have the following:

- C'_1 and C'_2 are clauses of $[R_P^{n-1}]$ since they are ground instances of C_1 and C_2, ($C'_1 \equiv C_1\sigma\theta\gamma$ and $C'_2 \equiv C_2\sigma\theta\gamma$) and C_1 and C_2 belong to R_P^{n-1}
- By induction hypothesis $[R_P^{n-1}] = R_{[P]}^{n-1}$, therefore $C'_1, C'_2 \in R_{[P]}^{n-1}$
- Finally, $C = C'_1 \cdot C'_2$, hence $C \in R_{[P]}^{n}$

(B) Now our goal is to prove $R_{[P]}^{n} \subseteq [R_P^n]$
Assume $C \in R_{[P]}^{n} = R_{[P]}^{n-1} \cup \mathcal{R}_{R_{[P]}^{n-1}}$ We have to consider the following cases:
Case 1: $C \in R_{[P]}^{n-1}$
By induction hypothesis $R_{[P]}^{n-1} = [R_P^{n-1}]$ and since $R_P^{n-1} \subseteq R_P^n$ we have that $[R_P^{n-1}] \subseteq [R_P^n]$. Therefore $C \in R_{[P]}^{n-1} = [R_P^{n-1}] \subseteq [R_P^n]$
Case 2: $C \in \mathcal{R}_{R_{[P]}^{n-1}}$
Let $C_1 \equiv A \leftarrow A_1, \ldots, A_s$ and $C_2 \equiv B \leftarrow B_1, \ldots, B_t$ be two clauses of $R_{[P]}^{n-1}$ such that $C = C_1 \cdot C_2$ with $A_i = B$, then $C \equiv A \leftarrow A_1, \ldots, A_{i-1}, B_1, \ldots, B_t, A_{i+1}, \ldots, A_s$ By induction hypothesis, $R_{[P]}^{n-1} = [R_P^{n-1}]$, hence $C_1, C_2 \in [R_P^{n-1}]$. Therefore there are two clauses $C'_1, C'_2 \in R_P^{n-1}$ (we can suppose they are standardized apart) and a ground substitution θ such that $C'_1\theta = C_1$ and $C'_2\theta = C_2$ Consider $C'_1 \equiv A' \leftarrow A'_1, \ldots, A'_s$ and $C'_2 \equiv B' \leftarrow B'_1, \ldots, B'_t$ We know that A'_i and B' are unifiable, since $A'_i\theta = A_i = B = B'\theta$. Let σ be a mgu of A'_i and B', i.e. $A'_i\sigma = B'\sigma$ and suppose $C' = C'_1 \cdot C'_2$, $C' \in R_P^n$ $C' \equiv A'\sigma \leftarrow A'_1\sigma, \ldots, A'_{i-1}, B_1\sigma, \ldots, B_t\sigma, A'_{i+1}\sigma, \ldots, A_t\sigma$. As σ is a mgu of A'_i and B', there is a substitution δ such that $\sigma\delta = \theta$ and therefore we have $C' \in R_P^n$ and $C'\delta = C$, hence $C \in [R_P^n]$ □

Theorem 4.6 *Let P be a definite program*

$$\forall n \geq 0, \qquad \forall k \geq 0 \qquad T_{R_{[P]}^n} \downarrow k \subseteq T_{[P]} \downarrow k$$

Proof. We prove the theorem by induction on n
$(n = 0)$ $T_{R_{[P]}^0} \downarrow k = T_{[R_P^0]} \downarrow k = T_{[P]} \downarrow k$
$(n \to n+1)$ Suppose

$$\forall k \geq 0 \qquad T_{R_{[P]}^n} \downarrow k \subseteq T_{[P]} \downarrow k \tag{1}$$

We have to prove

$$\forall k \geq 0 \qquad T_{R_{[P]}^{n+1}} \downarrow k \subseteq T_{[P]} \downarrow k$$

We prove this inclusion by induction on k
$(k = 0)$ $T_{R_{[P]}^{n+1}} \downarrow 0 = B_{R_{[P]}^n} = B_{[P]} = T_{[P]} \downarrow 0$
$(k \to k+1)$ In this case we suppose

$$T_{R_{[P]}^{n+1}} \downarrow k \subseteq T_{[P]} \downarrow k \tag{2}$$

and we have to prove

$$T_{R_{[P]}^{n+1}} \downarrow (k+1) \subseteq T_{[P]} \downarrow (k+1)$$

Suppose $A \in T_{R_{[P]}^{n+1}} \downarrow (k+1)$ Then there is a clause $C \in R_{[P]}^{n+1}$, $C \equiv A \leftarrow A_1, \ldots, A_s$ such that $\{A_1, \ldots, A_s\} \subseteq T_{R_{[P]}^{n+1}} \downarrow k$. Since $C \in R_{[P]}^{n+1} = R_{[P]}^{n} \cup \mathcal{R}_{R_{[P]}^n}$ we consider the following two cases:
Case 1: $C \in R_{[P]}^{n}$

On the one hand, since we assume (2), $T_{R^{n+1}_{[P]}} \downarrow k \subseteq T_{[P]} \downarrow k$, on the other hand by theorem 4.1, $T_{[P]} \downarrow k \subseteq T_{R^n_{[P]}} \downarrow k$. Therefore $T_{R^{n+1}_{[P]}} \downarrow k \subseteq T_{R^n_{[P]}} \downarrow k$. Also $\{A_1, \ldots, A_s\} \subseteq T_{R^{n+1}_{[P]}} \downarrow k$, then $\{A_1, \ldots, A_s\} \subseteq T_{R^n_{[P]}} \downarrow k$ and since $C \equiv A \leftarrow A_1, \ldots, A_s \in R^n_{[P]}$, we have

$$A \in T_{R^n_{[P]}}(T_{R^n_{[P]}} \downarrow k) = T_{R^n_{[P]}} \downarrow (k+1) \tag{3}$$

But we assume (1), then

$$T_{R^n_{[P]}} \downarrow (k+1) \subseteq T_{[P]} \downarrow (k+1) \tag{4}$$

Therefore, by (3) and (4), we have $A \in T_{[P]} \downarrow (k+1)$

Case 2: $C \in \mathcal{R}_{R^n_{[P]}}$

Let C_1 and C_2 be two clauses of $R^n_{[P]}$ such that $C = C_1 \cdot C_2$, i.e., $C_1 \equiv L \leftarrow L_1, \ldots, L_r$ and $C_2 \equiv M \leftarrow M_1, \ldots, M_t$ with $L_i = M$, hence

$$C \equiv L \leftarrow L_1, \ldots, L_{i-1}, M_1, \ldots, M_t L_{i+1}, \ldots, L_r$$

and then we have $L = A$ and

$$A_j = \begin{cases} L_j & \text{if } j \in \{1, \ldots, i-1\} \\ M_{j-i+1} & \text{if } j \in \{i, \ldots, i+t-1\} \\ L_{j-t+1} & \text{if } j \in \{i+t, \ldots, s\} \end{cases}$$

To prove $A \in T_{[P]} \downarrow (k+1)$, we will first see that $\{L_1, \ldots, L_r\} \subseteq T_{R^n_{[P]}} \downarrow k$. We will show it in two steps:

Step 1: We are going to prove $\{L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_r\} \subseteq T_{R^n_{[P]}} \downarrow k$.
We have $\{L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_r\} \subseteq \{A_1, \ldots, A_s\} \subseteq T_{R^{n+1}_{[P]}} \downarrow k$. By (2) we assume $T_{R^{n+1}_{[P]}} \downarrow k \subseteq T_{[P]} \downarrow k$ and by theorem 4.1 $T_{[P]} \downarrow k \subseteq T_{R^n_{[P]}} \downarrow k$.Hence $\{L_1, \ldots, L_{i-1}, L_{i+1}, \ldots, L_r\} \subseteq \{A1, \ldots, A_s\} \subseteq T_{R^{n+1}_{[P]}} \downarrow k \subseteq T_{[P]} \downarrow k \subseteq T_{R^n_{[P]}} \downarrow k$.

Step 2: Our goal now is to prove $L_i \in T_{R^n_{[P]}} \downarrow k$.
We have $\{M_1, \ldots, M_t\} \subseteq \{A1, \ldots, A_s\} \subseteq T_{R^{n+1}_{[P]}} \downarrow k$ and by (2) and theorem (4.1) (as above) $T_{R^{n+1}_{[P]}} \downarrow k \subseteq T_{R^n_{[P]}} \downarrow k$. And since the immediate consequence operator is monotonic and $T_{R^n_{[P]}} \downarrow 0 = B_P$, we have $T_{R^n_{[P]}} \downarrow k \subseteq T_{R^n_{[P]}} \downarrow (k-1)$. Therefore $\{M_1, \ldots, M_t\} \subseteq T_{R^{n+1}_{[P]}} \downarrow k \subseteq T_{R^n_{[P]}} \downarrow k \subseteq T_{R^n_{[P]}} \downarrow (k-1)$. So we have on the one hand $\{M_1, \ldots, M_t\} \subseteq T_{R^n_{[P]}} \downarrow (k-1)$ and, on the other hand the clause $C_2 \equiv M \leftarrow M_1, \ldots, M_t$ in $R^n_{[P]}$. Hence $M = L_i \in T_{R^n_{[P]}} \downarrow k$

So we have that $\{L_1, \ldots, L_r\} \subseteq T_{R^n_{[P]}} \downarrow k$ and $C_1 \equiv L \leftarrow L_1, \ldots, L_r$ is a clause of $R^n_{[P]}$. Hence $L = A \in T_{R^n_{[P]}} \downarrow (k+1)$. But we assume (1), so we have $T_{R^n_{[P]}} \downarrow (k+1) \subseteq T_{[P]} \downarrow (k+1)$. So we conclude $A \in T_{[P]} \downarrow (k+1)$. □

Corollary 4.7 $\forall n \geq 0, \quad \forall k \geq 0 \qquad T_{R^n_P} \downarrow k \subseteq T_P \downarrow k$

Proof. By corollary 4.4 $T_{R^n_P} \downarrow k = T_{[R^n_P]} \downarrow k$. By lemma 4.5 $T_{[R^n_P]} \downarrow k = T_{R^n_{[P]}} \downarrow k$. By theorem 4.6 $T_{R^n_{[P]}]} \downarrow k \subseteq T_{[P]} \downarrow k$. Applying corollary 4.4 again, $T_{[P]} \downarrow k = T_P \downarrow k$. So we have $T_{R^n_P} \downarrow k = T_{[R^n_P]} \downarrow k = T_{R^n_{[P]}]} \downarrow k \subseteq T_{[P]} \downarrow k = T_P \downarrow k$. □

Finally, if we put together theorem 4.1 and corollary 4.7 we can enounce our equality.

Theorem 4.8 *Let P be a definite program. Then* $\forall n \geq 0$, $\forall k \geq 0\, T_{R_P^n} \downarrow k = T_P \downarrow k$

Corollary 4.9 *Let P be a definite program. Then* $\forall n \geq 0$, $T_{R_P^n} \downarrow \omega = T_P \downarrow \omega$

Proof. Trivially, from 4.8 $T_{R_P^n} \downarrow \omega = \cap\{T_{R_P^n} \downarrow k \,:\, k \geq 0\} = \cap\{T_P \downarrow k \,:\, k \geq 0\} = T_P \downarrow \omega$ $\square$

5 Conclusion and future work

One of the research fields having received more attention in recent years has been Inductive Logic Programming [4, 5, 6]. But, despite its encouraging success, we think that a deeper research in its semantics is necessary.
The results we present, although interesting by themselves, can be seen as a new step in this line.
If we consider Robinson's operator as a generalization of the method called *unfolding*, introduced by Boström and Idestam–Almquist in [2] to solve the specialization problem of ILP, this paper offers a semantic point of view to approach this problem.

References

[1] K.R. Apt. *Handbook of Theoretical Computer Science*, chapter 10 Logic Programming, pages 493 – 573. Elsevier Science Publishers B.V., 1990.

[2] H. Boström and P. Idestam-Almquist. Specialization of logic programs by pruning sld–trees. In Stefan Wrobel, editor, *Proc. of the 4th Int. Workshop on Inductive Logic Programming (ILP–94)*, pages 31–48. Bad Honnef/Bonn, 1994.

[3] J.W.Lloyd. *Foundations of Logic Programming.* Springer–Verlag Berlin, second edition, 1987.

[4] S. Muggleton. Inductive logic programming. In *First Conference on Algorithmic Learning Theory, Tokio.* Ohmsha, 1990.

[5] S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19–29:629–679, 1994.

[6] S.H Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming.* Number 1228 in LNAI Tutorial. Springer–Verlag, May 1997.

[7] J.A. Robinson. A machine–oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, Jan. 1965.

[8] M.H. van Emden and R.A. Kowalski. The semantics of predicate logic as a programming language. *J. ACM.*, 23(4):733–742, Oct. 1976.