UNIVERSIDADE DE SÃO PAULO

Learning restricted Horn clauses:
Some considerations on the ij-determination concepts

MARIA DO CARMO NICOLETTI
MARIA CAROLINA MONARD

Nº 09

NOTAS

Instituto de Ciências Matemáticas de São Carlos

Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2577

Learning restricted Horn clauses: Some considerations on the ij-determination concepts

MARIA DO CARMO NICOLETTI
MARIA CAROLINA MONARD

Nº 09

N O T A S   D O   I C M S C
Série Computação

São Carlos
apr. / 1994

Learning Restricted Horn Clauses: Some Considerations on the ij-Determination Concept

Maria do Carmo Nicoletti[1]
Universidade Federal de São Carlos/ILTC/IFQSC-USP
Departamento de Computação
Via Washington Luiz, km 235
Caixa Postal 676, 13565-905 - São Carlos, SP
e-mail:carmo@icmsc.sc.usp.br

Maria Carolina Monard[2]
Universidade de São Paulo/ILTC
Instituto de Ciências Matemáticas de São Carlos
Departamento de Ciências de Computação e Estatística
Caixa Postal 668, 13560-970 - São Carlos, SP
e-mail: mcmonard@icmsc.sc.usp.br

Abstract

Inductive Logic Programming — ILP — is a relatively new approach to machine learning which attempts to overcome some of the limitations of earlier methods by

- allowing the use of background knowledge (domain theory) and
- providing a more powerful concept description language, namely, the language of logic programs

The adoption of the language of logic programs as instances and hypotheses description languages gives rise to many difficulties that need to be overcome — learning logical definitions requires the exploration of very large space of hypotheses descriptions and, consequentely, restrictions should be imposed on the hypotheses space so to make learning a feasible task. Among them is the ij-determination restriction on Horn Clauses. The concept of ij-determination is important since it is often used to compare the behaviour of different ILP learning systems based on ij-determination restriction. However, in the available literature, some slightly different interpretations are associated with this concept as well as to concepts which subsidize the definition of ij-determination.

In this work we present and discuss some interpretations given to ij-determination and associated concepts attempting to reach an agreement among them, aiming an unified approach.

[1]Work partially supported by State Research Council FAPESP Proc. Nº 92/2151-8.

[2]Work partially supported by National and State Research Council — CNPq and FAPESP Proc. Nº 92/2151-8.

Contents

1 Introduction

The concept of *ij-determination* is used in Inductive Logic Programming — ILP — in order to restrict the hypotheses language, which is done through limiting the introduction of new terms in the body of the clause being constructed. In the available literature related to *ij-determination* it can be evidenced some slightly different interpretations associated with this concept. This happens as well to concepts which subsidize the definition of *ij-determination.*

The main concern in this work is to present, discuss and analyse the different interpretations given to *ij-determination* and associated concepts, trying to evidence what each of them mean when used as a bias for restricting the hypotheses language in a framework of inductive logic programming. Underneath this discussion there is the intention of attempting to reach an agreement among the different interpretations, aiming an unified approach.

This work is organized as follows: Section 2 formulates the task of empirical, single predicate learning in ILP. Section 3 discusses ij-determination restriction for function-free ordered clauses. Section 4 treats the same restriction but for ordered Horn clauses with function symbols, discussing several problems related to ij-determination and suggesting possible modifications. Our conclusions are presented in Section 5.

2 Learning in Inductive Logic Programming

The task of empirical, single predicate learning in ILP can be formulated as follows [Lavrač 92] [Muggleton 91]

Given

- a set of training examples $\mathcal{E}$ described in a language $\mathcal{L}_\mathcal{E}$ and consisting of
  - positive examples, $\mathcal{E}^+$
  - negative examples, $\mathcal{E}^-$

  of an unknown predicate p — target relation
- a description language $\mathcal{L}_C$, specifying syntatic restrictions on the definition of predicate p
- background knowledge $\mathcal{B}$, described in language $\mathcal{L}_\mathcal{B}$, defining predicates q_i (other than p) which may be used in the definition of p and which provide additional information about the arguments of the examples of predicate p
- a matching operator between $\mathcal{L}_\mathcal{E}$ and $\mathcal{L}_C$ with respect to (wrt) $\mathcal{L}_\mathcal{B}$ that determines whether an example is covered by a clause expressed in $\mathcal{L}_C$

Find

- a definition $\mathcal{H}$ for p, expressed in $\mathcal{L}_C$, such that
  - $\mathcal{H}$ is complete, i.e., $\mathcal{B} \wedge \mathcal{H} \models \mathcal{E}^+$,
  - $\mathcal{H}$ is consistent , i.e., $\mathcal{B} \wedge \mathcal{H} \not\models \mathcal{E}^-$

  with respect to the examples

The complexity of a learning task depends on the expressiveness of the hypotheses language $\mathcal{L}_C$ [Nicoletti 93]. Therefore, to reduce this complexity, various restrictions can be applied to the language itself and to clauses expressed in $\mathcal{L}_C$. In practice, ILP systems work within various syntactic and semantic restrictions in order to limit the complexity of a learning problem. The most used restriction employed by ILP systems is the one which confines the expression of a concept to ordered Horn clauses only. In this work, restricting hypotheses language is meant to be restricting ordered Horn clauses, which will be approached considering functions-free and not function-free ordered Horn clauses.

Definition 2.1 *A Horn clause is ordered if there exists a total ordering over the literals in the body of the clause.*

3 Function-free Hypotheses Language

Assuming a function-free hypotheses language $\mathcal{L}_C$, the next concepts can be used to restrict the learned hypotheses — function-free Horn clauses — expressed in $\mathcal{L}_C$.

Definition 3.1 *A clause is function-free if it has no function symbols.*

Definition 3.2 *(input and output variables) [Cohen 93] If* $A \leftarrow L_1, \ldots, L_{i-1}, L_i, \ldots, L_m$ *is an ordered Horn clause, then the input variables of the literal* L_i *are those variables appearing in* L_i *which also appear in the clause* $A \leftarrow L_1, \ldots, L_{i-1}$*; all other variables appearing in* L_i *are called output variables.*

Table 1 gives some examples of input and output variables of literals in a clause.

3.1 Depth of Variables and Clauses

Some ILP systems restrict the hypotheses language by only allowing Horn clauses with variables up to a certain given depth. It follows two equivalent definitions of *depth of variables* in a Horn clause.

Definition 3.3 *(depth of variable) [Cohen 93] Let the clause*

$$A \leftarrow L_1, L_2, \ldots, L_i, \ldots, L_m$$

A		L_1	L_2	L_3	L_4
$p(X_1, X_2)$	$\leftarrow$	$q(X_1, X_2)$.			
		$I = X_1$ and X_2			
$p(X_1, X_2)$	$\leftarrow$	$q(X_1, X_3)$,	$p(X_3, X_2)$,	$r(X_2)$.	
		$I = X_1$ $O = X_3$	$I = X_2$ and X_3	$I = X_2$	
$p(X_1, X_2, X_3)$	$\leftarrow$	$q(X_1, Y_1, Z_3)$,	$r(X_2, Y_2, Z_3)$,	$s(X_3, Y_2, Z_2)$,	$t(Z_2, Z_3, Z_1)$.
		$I = X_1$ $O = Y_1$ and Z_3	$I = X_2$ and Z_3 $O = Y_2$	$I = X_3$ and Y_2 $O = Z_2$	$I = Z_2$ and Z_3 $O = Z_1$
$p(X_1, X_2)$	$\leftarrow$	$q(X_1, Y_1)$,	$r(Z_1, Z_2)$,	$s(Z_1, X_2, Z)$,	$t(Z, Z_2, W)$.
		$I = X_1$ $O = Y_1$	$O = Z_1$ and $Z2$	$I = Z_1$ and X_2 $O = Z$	$I = Z$ and Z_2 $O = W$

Table 1: Input/Output Variables of Literals in a Clause

Variables appearing in the head of a clause have depth zero. Otherwise, let L_i be the first literal containing the variable V and let d be the maximal depth of the input variables of L_i; then the depth of V is $d + 1$.

Definition 3.4 *(depth of variable) [Lavrač 94] Let the clause*

$$p(X_1, X_2, \ldots, X_n) \leftarrow L_1, L_2, \ldots, L_i, \ldots, L_m$$

The variables $X_1, \ldots, X_n$ that appear in the head of the clause have depth zero.

Let V be a variable appearing for the first time in literal L_i. Let d be the maximum depth of any of the other variables in L_i that appear in the clause

$$p(X_1, X_2, \ldots, X_n) \leftarrow L_1, L_2, \ldots, L_{i-1}$$

Then the depth of variable V is $d + 1$.

The following example justifies the need for extending Definition 3.3 and Definition 3.4, in order to deal with the situation of a literal without any input variable.

Example 3.1 *Let the clause*

$$p(X,Y) \leftarrow q1(X,Z), q2(X1,Y1), q3(X1,Z)$$

In this clause the variables X1 and Y1 appear for the first time in literal q2(X1,Y1). Since there are no variables in q2(X1,Y1) which have previously appeared in p(X,Y) ← q1(X,Z) (i.e. q2(X1,Y1) has no input variables), according to both original definitions, depth(X1) and depth(Y1) will be undefined.

Still related to the two previous definitions of depth of variables, it is important to notice that top-down ILP systems would not *especialize* a clause adding to it a literal with no input variables — those variables would be *unbound* at that point of the clause construction and consequently, useless to the system [Lavrač 93]. However, a bottom-up ILP system could eventually construct a clause having a literal without any input variables, like the literal $q2(X1, Y1)$ in Example 3.1. A suggestion to avoid this problem is to add to Definition 3.3 the following

If a literal has no input variables, all variables appearing in the literal have depth 1

and to Definition 3.4

If any other variable of L_i does not appear in the clause

$$p(X_1, X_2, \ldots, X_n) \leftarrow L_1, L_2, \ldots, L_{i-1}$$

then depth of V is 1

Definition 3.5 *The depth of a clause is the maximal depth of any variable in the clause.*

Table 2 shows some examples of depth of variables and clauses.

clause			X_1	X_2	X_3	Y_1	Y_2	Z_1	Z_2	Z_3	depth of clause
$p(X_1, X_2)$	$\leftarrow$	$q(X_1, X_2)$.	0	0							0
$p(X_1, X_2)$	$\leftarrow$	$q(X_1, X_3)$, $p(X_3, X_2)$, $r(X_2)$.	0	0	1						1
$p(X_1, X_2, X_3)$	$\leftarrow$	$q(X_1, Y_1, Z_3)$, $r(X_2, Y_2, Z_3)$, $s(X_3, Y_2, Z_2)$, $t(Z_2, Z_3, Z_1)$.	0	0	0	1	2	4	3	1	4
$p(X_1, X_2, X_3)$	$\leftarrow$	$q(Y_1, X_2)$, $r(X_3, Z_1, X_2, Y_1)$, $s(Y_2, X_1, Z_3)$, $t(Y_2, Z_3, Z_2, Z_1)$.	0	0	0	1	1	2	3	1	3
$p(X_1, X_2, X_3)$	$\leftarrow$	$q(X_1, Z_2)$, $r(Z_2, X_3, Y_1)$, $s(Y_1, Z_1, X_1)$, $t(X_2, Z_1, Z_2, Y_2)$, $u(Z_1, Y_1, Y_2, Z_3)$.	0	0	0	2	4	3	1	5	5
$p(X_1, X_2, X_3)$	$\leftarrow$	$q(Y_1, X_2, Y_2)$, $r(X_1, Z_3, Y_1, Z_2)$, $s(X_3, Z_1)$.	0	0	0	1	1	1	3	2	3

Table 2: Depth of Variables and Clauses

The dependencies among variables established by their depths impose a partial ordering over the literals within the clause, as illustrated in Figures 1 and 2.

3.2 ij-Determination

By imposing a limit on the maximum variable depth i, the syntatic complexity of clauses expressed in function-free hypotheses languages can be restricted. The following definition of *determination*, has been adapted to the case of function-free clauses.

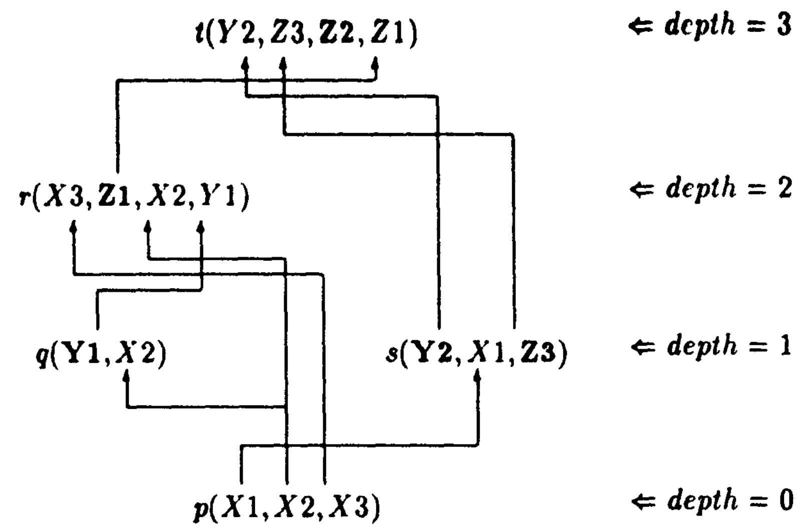

Figure 1: Partial Ordering Imposed by Depth of Variables in $p(X1, X2, X3) \leftarrow q(X1, X2), s(Y2, X1, Z3), r(X3, Z1, X2, Y1), t(X2, Z3, Z2, Z1)$

Definition 3.6 *(determination) [Džeroski 92]*

- *a predicate definition is determinate if all of its clauses are determinate*
- *a clause is determinate if each of its literals is determinate*
- *a literal in a clause is determinate if each of its variables that do not appear in preceding literals has only one possible binding given the bindings of its variables that appear in preceding literals.*

Definition 3.7 *(i-determination) [Lavrač 94] The determination restriction is called i-determination for a given maximum variable depth i.*

Definition 3.8 *(ij-determination) [Lavrač 94] Let j be an integer constant so that only the class of ILP problems where all predicates in the background knowledge $\mathcal{K}$ are of arity at most j will be considered (j is known as arity bound). Given the arity bound j then i-determination implies ij-determination.*

The following definition is equivalent to Definition 3.8.

Definition 3.9 *(ij-determination) [Cohen 93] A determinate clause of depth bounded by a constant i over a ground background theory $K \in j\text{-}K$[3] is called ij-determinate.*

It should be noted that in a function-free hypotheses language restricted to ordered Horn clauses, restricting the learning task to learning *ij-determinate* clauses is a synonym for construction Horn clauses where

[3] j-K set of ground unit clauses of arity j or less, for some fixed j.

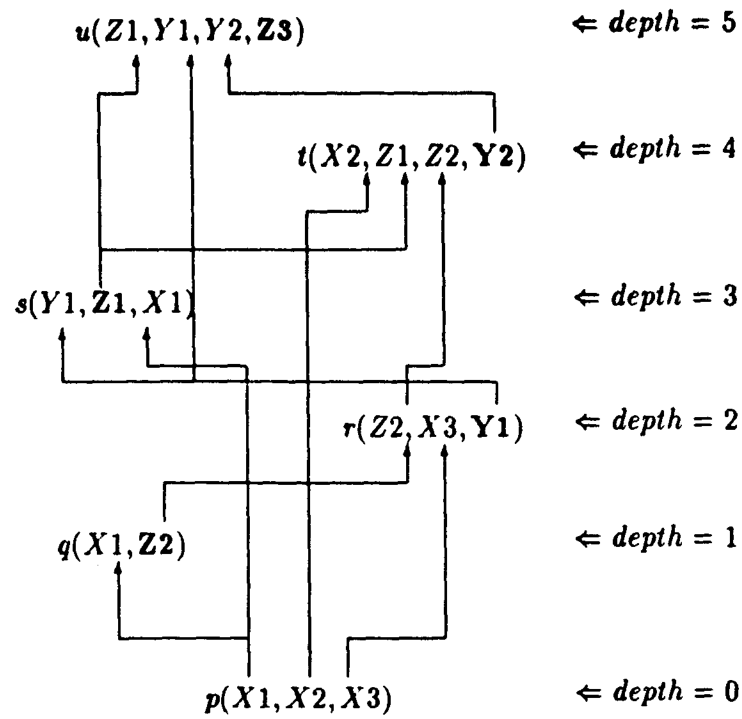

Figure 2: Partial Ordering Imposed by Depth of Variables in $p(X1,X2,X3) \leftarrow q(X1,Z2), r(Z2,X3,Y1), s(Y1,Z1,X1), t(X2,Z1,Z2,Y2), u(Z1,Y1,Y2,Z3)$

- the maximum variable depth is not greater than i
- the maximum arity of predicates given as background knowledge is not greater than j

4 Hypotheses Language with Function Symbols

Hypotheses language with function symbols allows the definition of terms other than just variables and constants and gives rise to the definition of *determinate term.* This concept is latter used to define ij-determination for hypotheses languages with function symbols.

4.1 Determinate Term

We shall discuss next two definitions of determinate term found in the literature as well as point out some problems related to them and suggest possible modifications.

Definition 4.1 *(determinate term) [Muggleton 90] Let*

- *$\mathcal{K}$ be a logic program*
- *$\mathcal{E}$ be a set of examples given as ground atoms*
- *$A \leftarrow L_1, \ldots, L_m, L_{m+1}, \ldots, L_n$ be an ordered Horn clause*
- *t be a term found in L_{m+1}*

The term t is a determinate term wrt L_{m+1} iff for every substitution θ such that

- *$A\theta \in \mathcal{E}$*
- *$\{L_1, L_2, \ldots, L_m\}\theta \subset \mathcal{M}(\mathcal{K})$*

there is a unique atom $L_{m+1}\theta\delta \in \mathcal{M}(\mathcal{K})$, i.e., for any such θ there is a unique valid ground substitution δ whose domain is the variables in t.

Notice that the valid substitutions for a determinate term t in any literal L_{m+1} are a function of the substitutions applied to a certain number of other terms $s_1, \ldots, s_d$ within L_{m+1}. In this case we say that literal L_{m+1} has degree d with respect to t. We can define a restricted hypotheses language by placing a limit j on the maximum *degree of any literal with respect to any term.* Again we can define a restricted hypotheses language by placing a limit i on the maximum *depth of any literal*[4] in an hypothesized clause [Muggleton 90].

There are some points to consider regarding Definition 4.1. One of them is that nothing has been stated how the "determination of a term" should be examined. Should the criteria of determination of a term be applied to terms in the literals in a left-to-right order of their appearance in the literal? Or should it be applied to terms within L_{m+1} disregarding its position and disregarding the examination of any previous term? According to Definition 4.1, the pre-requisites for a term t be determinate wrt a literal L_{m+1} is that for every substitution θ such that

- $A\theta \in \mathcal{E}$
- $\{L_1, L_2, \ldots, L_m\} \subset \mathcal{M}(\mathcal{K})$

there is a unique atom $L_{m+1}\theta\delta \in \mathcal{M}(\mathcal{K})$, i.e., for any such θ there is a unique valid ground substitution δ whose domain is the variables in t. Some situations to consider are discussed next.

- When the term t in literal L_{m+1} has some variables which also appear in other terms in any previous literals of the clause then, since one of the pre-conditions states that the substitution θ should be such that $\{L_1, L_2, \ldots, L_m\} \subset \mathcal{M}(\mathcal{K})$, the domain of δ should be *a subset of variables in t* instead of *the variables in t.* In other words, the domain of δ should be the unbound variables in t

[4]In [Muggleton 90] the concept of depth of a literal is not formally defined; it is introduced through examples considering as terms only variables.

- If the variables which appear in t also appear in other terms of L_{m+1}, Definition 4.1 does not state clearly which substitution should be applied to those variables: θ or δ (see Example 4.1)

- Clearly some θ which verify the conditions $A\theta \in \mathcal{E}$ and $\{L_1, L_2, \ldots, L_m\} \subset \mathcal{M}(\mathcal{K})$ can specify only variables already in $\{L_1, L_2, \ldots, L_m\}$. So, how to verify if $L_{m+1}\theta\delta \in \mathcal{M}(\mathcal{K})$ if δ only applies to variables in t and L_{m+1} has other terms with variables which do not appear neither in t nor in $L_1, L_2, \ldots, L_m$?

  One way of dealing with this situation is considering θ as an *extended* θ, so to include variables of L_{m+1} (except those that appear in t). That could be implicitly stated in the definition by stating *a unique ground atom* $L_{m+1}\theta\delta$. By saying so, the definition would implicitly be stating that the substitution θ would have in its domain the variables of L_{m+1} (except those which appear in t) in order to obtain a ground atom. If such an *extention* of θ cannot be considered, evéntually there will be variables left in L_{m+1} (others that those in t), after applying θ. In this case there is no way of saying anything about the uniqueness of δ, which is what matters for a term be determinate wrt a literal (see Example 4.2)

- Another point to consider in Definition 4.1 is related to the fact of allowing (or not) the δ substitution be the empty substitution. That is the case when, after applying θ, $L_{m+1}\theta \in M(K)$; this fact obviously makes the unique δ be the empty substitution ϵ

Example 4.1 *Consider the clause*

$$p(X,Y) \leftarrow q(X), r(Y), s(M, f(M,N,X))$$

and supose the term $f(M,N,X)$ should be examined for its determinacy wrt the literal $s(M, f(M,N,X))$. It is not clearly stated which substitution should make the variable M be ground.

Example 4.2 *Consider the clause*

$$p(X) \leftarrow q(X,Y), r(Z, f(X,W))$$

where the term $f(X,W)$ should be analysed in order to verify if it is a determinate term wrt the literal $r(Z, f(X,W))$. Consider that $\mathcal{E} = p(1)$ and $\mathcal{M}(\mathcal{K})$ is

$$q(1,2)$$
$$r(a, f(1,3))$$

For $\theta = \{X/1, Y/2\}$,

$$p(1) \leftarrow q(1,2), r(Z, f(1,W))$$

which verifies

$$1.\ \ p(X)\theta \in \mathcal{E}$$
$$2.\ \ q(X,Y)\theta \subset \mathcal{M}(\mathcal{K})$$

By definition, the substitution δ *should only consider* W *(since* X *has already been substituted using* θ*),* Z *remains unbound, so* $r(Z, f(X,W))\theta\delta$ *will never be in a ground model. Now, if* θ *is* extended *to be*

$$\theta = \{X/1, Y/2, Z/a\}$$

then (1.) and (2.) are valid and for $\delta = \{W/3\}$, $r(Z, f(X,W))\theta\delta \in \mathcal{M}(\mathcal{K})$. *As nothing has been said about the domain of substitution* θ*, each of those* θ *which specify only variables in* A *and* $L_1, L_2, \ldots, L_m$ *could eventually be extended.*

Our suggestion is to modify Definition 4.1 to take those cases into account such that instead of having

there is a unique atom $L_{m+1}\theta\delta \in \mathcal{M}(\mathcal{K})$*, i.e., for any such* θ *there is a unique valid ground substitution* δ *whose domain is the variables in* t.

have

there is a unique ground atom $L_{m+1}\theta\delta \in \mathcal{M}(\mathcal{K})$*, i.e., for any such* θ *there is at most one valid ground substitution* δ *whose domain is the variables in* t.

The following example takes into account these extensions.

Example 4.3 *Consider the clause*

$$p(X,Y) \leftarrow q(X), r(Y), s(W, f(M,X))$$

and let us suppose we are trying to find out if the term $f(M,X)$ *is determinate or not with respect to* $s(W, f(M,X))$*. This clause has the form*

$$A \leftarrow L_1, L_2, L_3$$

where the term $f(M,X)$ *appears in* L_3.

Paraphrasing Definition 4.1, informally we can say that to verify whether the term $f(M,X)$ *in literal* L_3 *is determinate with respect to that literal, the variables in the term, namely —* M, X *— must have a unique binding, given the binding of terms in* A*,* L_1 *and* L_2*. Since determination is a semantic notion it should be verified wrt examples and background knowledge. Let us suppose that*

$\mathcal{E}$	$\mathcal{K}$
$p(1,2)$	$q(1)$
	$r(2)$
	$s(6, f(0,1))$

According to Definition 4.1, $f(M,X)$ is determinate wrt $L_3 = s(W, f(M,X))$ iff for every substitution θ such that $p(X,Y)\theta \in \mathcal{E}$ and $\{L_1, L_2\}\theta \subseteq M(K)$, there is a unique atom $s(W, f(M,X))\theta\delta \in M(K)$. It should be observed that

- *δ is a substitution whose domain is the variables in $f(M,X)$*
- *in order to have $p(X,Y)\theta \in \mathcal{E}$, X should be taken into account by the substitution θ*
- *in order to have $s(W, f(M,X))\theta\delta \in \mathcal{M}(\mathcal{K})$, W should be taken into account by the substitution θ*

Let the substitution $\theta = \{X/1, Y/2, W/6\}$. We have

1. $p(X,Y)\theta = p(1,2) \in \mathcal{E}$
2. $L_1\theta = q(1) \subset \mathcal{M}(\mathcal{K})$
3. $L_2\theta = r(2) \subset \mathcal{M}(\mathcal{K})$ *and*
4. $L_3\theta = s(6, f(M,1))$

then, for $\theta = \{X/1, Y/2, W/6\}$ there exists a unique ground substitution $\delta = \{M/0\}$, such that $s(W, f(M,X))\theta\delta \in \mathcal{M}(\mathcal{K})$ so the term $f(M,X)$ is determinate wrt L_3.

If for instance the examples and background knowledge were

$\mathcal{E}$	$\mathcal{K}$
$p(1,2)$	$q(1)$
	$r(2)$
	$s(6, f(0,1))$
	$s(6, f(2,1))$

then the term $f(M,X)$ would not be determinate wrt L_3, since for the same θ as above, there would exist two δ verifying $s(W, f(M,X))\theta\delta \in \mathcal{M}(\mathcal{K})$, namely $\delta = \{M/0\}$ and $\delta = \{M/2\}$.

Determinate term has also been defined by J.R.Quinlan as follows

Definition 4.2 *(determinate term) [Quinlan 91] Let the clause*

$$A \leftarrow L_1, \ldots, L_m$$

and suppose t is a term in L_m. Let

- *$\{Y\}$ be the set of variables that occur only in t and*
- *$\{Z\}$ be the set of other variables[5] in the clause*

Term t is determinate wrt L_m if for every ground substitution for variables in $\{Z\}$ there is at most one ground substitution for variables in $\{Y\}$ so that the clause is satisfied. Intuitively, the values of variables in $\{Y\}$ are determined by the values of variables that appear earlier in the clause.

It can be seen that this last sentence does not follow from the definition. Observe that the values of variables in $\{Y\}$ can be determined not only by the values of variables that appear earlier in the clause, but by values of variables that appear in the same literal as t.

Since $\{Z\}$ is the set of variables that occur in the clause (except for those variables occuring only in t), eventually variables from terms that appear latter in the literal L_m — after t has appeared — are in the set $\{Z\}$. As there is no restriction about t being the last term of the literal L_m, eventually the values of variables in $\{Y\}$ can be determined by the values of variables in terms that appear in the literal L_m after t and not only by values of variables that appear earlier in a clause, as illustrated by next example.

Example 4.4 *Let the clause*

$$p(M,N) \leftarrow q1(f(M,Q),g(M)),q2(h(M),i(R,S),j(M,T))$$

Considering the literal $q2(h(M),i(R,S),j(M,T))$, term $t = i(R,S)$ and sets

$$\begin{aligned} \{Y\} &= \{R,S\} \\ \{Z\} &= \{M,N,Q,T\} \end{aligned}$$

the term t is determinate wrt $q2(h(M),i(R,S),j(M,T))$ if for every ground substitution for variables in $\{Z\}$ there is at most one ground substitution for variables in $\{Y\}$ so that the clause is satisfied. The values of variables R and S are determined by the value of T as well, a variable which appear after t in the considered literal.

The definition of *determinate term* presented next is a slightly modified version of Definition 4.2.

Definition 4.3 *(determinate term) [Quinlan 91a] Let the clause*

[5] The set of variables that occur in other terms in the clause.

$$A \leftarrow L_1, \ldots, L_n$$

and suppose t is a term in L_n. Let

- *$\{Y\}$ be the set of variables that occur for the first time in t and*
- *$\{Z\}$ be the set of variables that occur earlier in the clause*

Term t is determinate wrt L_m iff for every ground substitution for variables in $\{Z\}$, there is at most one ground substitution for variables in $\{Y\}$ so that the clause is satisfied. Intuitively, the values of variables in $\{Y\}$ are determined by the values of variables that appear earlier in the clause.

In this definition, the literal considered is not necessarily the last and the set $\{Z\}$ is considered as the set of variables that occur earlier than t in the clause. Nevertheless the problem evidenced through Example 4.4 still remains.

4.2 ij-Determination

Combining the parameters of depth of literals — informally defined through examples — and degree of literals and using as basis Definition 4.1, the following definition for restricting the hypotheses language is stated

Definition 4.4 *(ij-determination) [Muggleton 90] Let $\mathcal{K}$ be a logic program and the examples $\mathcal{E}$ be a set of ground facts.*

1. *Every unit clause is 0j-determinate*
2. *An ordered clause $A \leftarrow L_1, \ldots, L_m, L_{m+1}, \ldots, L_n$ is ij-determinate iff*
   - *(a) $A \leftarrow L_1, \ldots, L_m$ is (i-1)j-determinate*
   - *(b) every literal L_k in $L_{m+1}, \ldots, L_n$ contains only determinate terms and has degree at most j.*

Although in the previous definition j is related to the maximum degree of literals in a clause, some authors have given another interpretation to this parameter. In [Rouveirol 92] it can be found that

ij-determination uses the property that for some predicates in the body of the generalized clause, given one instantiation of their input arguments, there is only one possible instantiation of their output arguments. ij-determination also includes a bound on the depth of existential quantification for a variable and a bound on the number of literals needed for one literal in the body of the generalized clause to be determined.

In the above statement the parameter j is associated with the number of literals (satisfying a certain condition) and not with the maximum degree of literals.

4.3 Another Notion of Depth and the Notion of Level of a Term

The notion of *depth of term* introduced in [Muggleton 93] is an entirely different notion than those introduced by earlier definitions of depth, in the sense that it does not consider a term as part of a literal which in turn is part of a clause.

Definition 4.5 *(depth of term) [Muggleton 93] The depth*

- *$d(V)$ of a variable V is 0*
- *$d(c)$ of a constant c is 1*
- *$d(f(t_1,\ldots,t_n))$ of a term $f(t_1,\ldots,t_n)$ is $1 + max\ d(t_i)$*

Limiting the depth of terms in hypotheses to 1 corresponds to working with function free clauses. This definition of *depth of term* treats a term on its own, which means, it does not consider a term as part of a literal which in turn is part of a clause.

Example 4.5 *Considering the term*

$$f(g(X,Y),h(i(M,X,N),K),j(r(s(K)),M),O)$$

its depth is given by

$$d(f(g(X,Y),h(i(M,X,N),K),j(r(s(K)),M),O)) =$$
$$1 + max(d(g(X,Y)),d(h(i(M,X,N),K,L)),d(j(r(s(K)),M)),d(O))) =$$
$$1 + max(1,3,3) = 4$$

since

$d(g(X,Y)) = 1 + max(d(X),d(Y)) = 1 + 0 = 1$
$d(h(i(M,X,N),K,L)) = 1 + max(d(h(i(M,X,N)),d(K),d(L)) =$ $1 + max(1 + max(d(i(M,X,N))),0,0)) = 1 + max(1 + (1 + max(0,0,0)),0,0)) =$ $1 + max(1 + 1,0,0) = 3$
$d(j(r(s(K)),M)) = 1 + max(d(r(s(K))),d(M)) = 1 + max(1 + max(d(s(K))),0) =$ $1 + max(1 + 1,0) = 3$

A pictorial representation of level of the term considered in Example 4.5 is shown in Figure 4.6.

It is important to note that there are two distinct approaches to *depth of term* in the literature. The first one, given by Definition 3.3 and 3.4, restricted to variables, takes into account the first occurence of the variable in the clause. The second one, given by Definition 4.5 defines depth of a term by only taking into consideration the

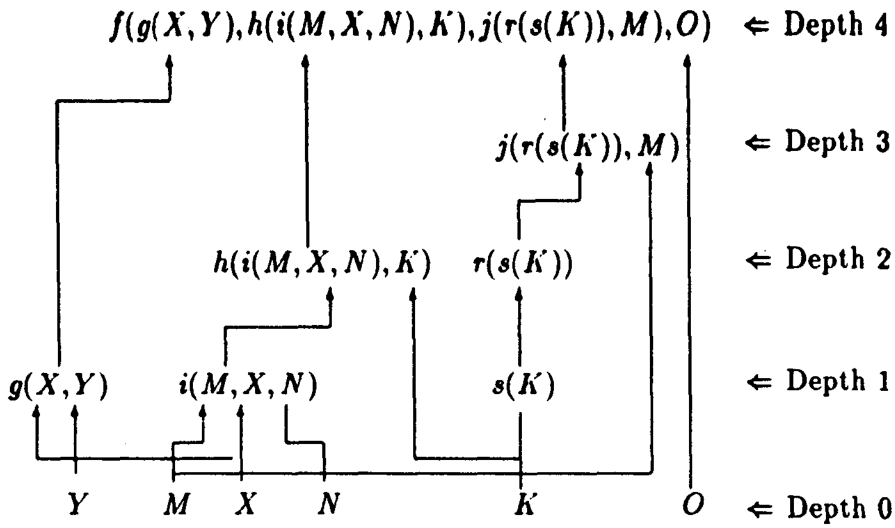

Figure 3: Depth of Term

term. Although having the same name, the concept of *depth of term* as established in Definition 3.3 and 3.4[6] has an entirely different meaning than the one established in Definition 4.5.

Muggleton defines level of a term as follows

Definition 4.6 *(level of a term) [Muggleton 93] The level l(t) of a term t in a clause C is 0 if t occurs as an argument in the head of C; and 1 + min l(s) where s and t occur as arguments in the same literal of C*

The definition of *level of a term* does not impose any order among the terms in a literal for evaluating their level. For example, considering the clause

$$p(X_1, X_2, X_3) \leftarrow q(X_1, Z_1, Z_2)$$

If we choose to evaluate the level of term Z_2, which is defined as $1 + min\ l(s)$ where $s \in \{X_1, Z_1, Z_2\}$, then $l(Z_1)$ is undefined. However, if level of Z_1 is evaluated first, then $l(Z_1) = l(Z_2) = 1$

We suggest the following two conditions to be added

1. consider the order given by the order of appearance of the literals in the clause as well as the terms in the literal

2. s and t occur as arguments in the same literal of C and s has occured in any previous literal

[6]Even being restricted to variables.

If these two conditions are considered and the chosen operation is *max* instead of *min*, then *depth of variables* (Definitions 3.4 and 3.3) will be a particular case of *level of a term*.

Furthermore, in [Muggleton 93] it is found the following comment

> The level of a term corresponds to Muggleton and Feng's *i* parameter and De Raedt's level of existential quantification

although the *i* parameter is defined in [Muggleton 90] as *depth*. If the definition of level corresponds to the *i* parameter, there is some mismatch because we do not believe that the *depth of term* as proposed in Definition 4.5 can be mapped into the *j* parameter so to justify the following statement (related to the ILP system GOLEM) found in [Muggleton 93] as well

> Both the level and depth of terms are frequently employed by ILP learners to define language restrictions. The two notions are especially important in the context of specific to general ILP systems such as ITOU, GOLEM, CLINT and PGA.

5 Conclusions

In this work we have discussed some basic concepts which support the definition of ij-determination restriction on Horn clauses as well the ij-determination concept itself, for function-free hypotheses languages and for hypotheses languages with function symbols.

It has been evidenced that for hypotheses languages without function symbols there is an agreement in the literature about ij-determination restriction. Still, the extension of the ij-determination concept for hypotheses languages with function symbols gives rise to different interpretations, which were presented and discussed. Some sugestions, attempting to reach an agreement among those interpretations aiming an unified approach, are also presented.

References

[Cohen 93] Cohen, W.W. *Learnability of Restricted Logic Programs.* Proceedings of the Third International Workshop on Inductive Logic Programming — ILP'93, Bled, Slovenia, April, 1993, pp 1-3.

[Džeroski 92] Džeroski, S.; Muggleton, H.S.; Russell, S. *PAC - Learnability of Determinate Logic Programs.* In Proceedings of ISSEK Workshop'92, Slovenia, September, 1992.

[Lavrač 92] Lavrač, N.; Džeroski, S. *Background Knowledge and Declarative Bias in Inductive Concept Learning.* Lectures Notes on Artificial Intelligence 462, 1992, pp. 51-71.

[Lavrač 93] Lavrač, N. *Personal Communication.* December 1993.

[Lavrač 94] Lavrač, N.; Džeroski, S. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, London, 1994.

[Muggleton 90] Muggleton, S.H.; Feng, C. *Efficient Induction of Logic Programs.* TIRM-90-044, Turing Institute Press, Glasgow, October 1990.

[Muggleton 91] Muggleton, S.H. *Inductive Logic Programming.* New Generation Computing, Vol. 8, 1991, pp 295-318.

[Muggleton 93] Muggleton, S.H.; De Raedt, L. *Inductive Logic Programming: Theory and Methods.* Report CW 178, Department of Computing Science, K.U.Leuven, May 1993.

[Nicoletti 93] Nicoletti,M.C.; Monard,M.C. *The Role of Description Languages in Inductive Concept Learning.* I Simpósio Brasileiro de Automação Inteligente, Rio Claro, SP, 8-10 setembro 1993, pp 403-412.

[Quinlan 91] Quinlan, J.R. *Determinate Literals in Inductive Logic Programming.* In: Proceedings of IJCAI 91, 1991, pp 746-750.

[Quinlan 91a] Quinlan, J.R. *Konwledge Acquisition from Structured Data.* IEEE Expert, December 1991, pp 32-37.

[Rouveirol 92] Rouveirol C. *Extensions of Inversion of Resolution Applied to Theory Completion.* In S. Muggleton (ed.), Inductive Logic Programming, Academic Press, 1992, pp 63-86.

NOTAS DO ICMSC

Serie computaçao

Nº 009/94 NICOLETTI, M.C.; MONARD, M.C. Learning restricted Horn clauses: some considerations on the ij-determination concept

Nº 008/94 TOME, M.F.; DUFFY, B. GENSMAC: a numerical method for solving unsteady non-newtonian free surface flows

Nº 007/94 TOME, M.F.; MCKEE, S. Numerical simulation of viscous fluid: buckling of planar jets

Nº 006/94 MASIERO, P.C.; OLIVEIRA, M.C.F.; GERMANO, F.S.R; PIERRI,G. Authoring and searching in dynamically growing hypertext data bases

Nº 005/94 NICOLETTI, M.C.; MONARD, M.C. Limiting the background knowledge in inductive logic programming

Nº 004/93 NICOLETTI, M.C.; MONARD, M.C. Learning horn clauses using the ILP system GOLEM

Nº 003/93 ARENALES, M.N.; MORABITO, R.N. An and/or graph approach to the solution of two-dimensional non-guillotine cutting problems

Nº 002/93 NICOLETTI, M.C.; MONARD, M.C. Herbrand interpretation model and least model within the framework of logic programming

Nº 001/93 MORABITO, R.; ARENALES, M.N. An and/or graph approach to the container loading problem