Term Partitions and Minimal Generalizations of Clauses

Shan-Hwei Nienhuys-Cheng
Department of Computer Science
Erasmus University, The Netherlands

Abstract Term occurrences of any clause C are determined by their positions. The set of all term partitions defined on subsets of term occurrences of C form a partially ordered set. This poset is isomorphic to the set of all generalizations of C. The structure of this poset can be inferred from the term occurrences in C alone. We can apply these constructions in this poset in machine learning.

1 Introduction

In first order logic a clause which is more general than a given clause C induces a partition on a subset of term occurrences of C. For example, given C=P(f(g(x)),g(x),g(x)) and C_1=P(f(u),g(u),v). The variable u in C_1 corresponds with the set B_u containing the first and the second g(x) in C and the variable v corresponds with the set B_v containing the third g(x) in C. Thus the two disjoint sets B_u, B_v form a partition on $B_u \cup B_v$. When the author ([3],[4]) tries to improve the algorithm of the Absorption in [2] for inverse resolution in machine learning, she observes the following:

- We should at the begining forget resolution or inverse resolution which concerns three clauses at the same time. Instead we should develope a general theory for finding all clauses more general than a given clause.
- An inverse substitution from a clause induces a partition on its term occurrences. Such partitions should be defined formally and independently from inverse substitutions. To study more general clauses than C means to study partitions on C. We should also order the partitions so it induces the order relation in clauses.
- To make the proofs easier and clearer, substitutions and inverse substitutions should be generalized to new mappings (consistent term mappings).

These questions are dealt with in [3],[4], but the algorithm there is still inefficient and undirected. In inverse resolution or other related topics we should generalize a clause to clauses which are not too general. For example, given a clause, what are the minimal generalizations? Given a few clauses what is their supremum ([5],[6])? Similiar questions can also be asked for partitions on a given clause C. This article tries to solve these questions. Consider a given clause C and all the partitions on C, we can construct the least higher partitions for a given partition and the supremum of some partitions. The constructions and proofs use only the term occurrences in the original clause(s).

To make this article self contained, we include also some of the results in [3],[4] here (theorem 1,3,4). Some theorems there are corollaries (corollary 1,2) of more general theorem here. Furthermore, the proofs of the theorems in this article can be found in [5].

2 Consistent Term Mappings

In this article we use a language of first order logic. The *constants* are denoted by a, b, c, etc. The *predicates* are denoted by P, Q, R, etc. and the *variables* are denoted by u, v, x, u_1, etc. The letters f, g, h, etc. are used to denote *functions*. A term is either *simple*, i.e. a constant or variable, or *compound*, i.e. it has the form $f(t_1,...,t_n)$ where t_i is a term and f is n-ary. An *atom* has the form $P(t_1,\ldots,t_n)$ where t_i are terms and P is n-ary. The *negation* of an atom M is denoted by ~M. An atom or the negation of an atom is called a *literal*. A clause has the form $L_1 \vee L_2 ... \vee L_n$ where L_i 's are literals.

2.1 Term occurrences

Definition. A *position* ([1],[5]) is a sequence of $<n_1,...,n_j>$ of positive integers. Let X be a term or a clause. We use <> to denote the position of X with respect to itself. If $X=L_1 \vee L_2 ... \vee L_n$, $n \geq 2$ the position of the i-th literal is denoted by <i>. If Y is a function or a predicate and $Y(t_1,..,t_n)$ has position $<p_1,...,p_k>$, then the position of t_i is $<p_1,....,p_k,i>$. A *term occurrence* is a pair (t,p) which represents the term t found in position p.

For example, y has position <1,2,1> in P(x,f(y,z))∨~Q(x) and has position <2,1> in P(x,f(y,z)). Notice that in a clause or a term the position determines the occurrence completely. If $p=<p_1,...,p_k>$ and $q=<p_1,..,p_k,q_1,..,q_j>$, then we say p is a *subsequence* of q or q *contains* p, denoted by $q \supset p$. This relation can also be denoted by pq'=q or q'=q-p where $q'=<q_1,... ,q_j>$. If (t,p) and (s,q) are two term occurrences in a clause or term X and q contains p, then we can find (s,q-p) in t. We say then (s,q) is a *subterm occurrence* of (t,p), denoted by $(t,p) \geq (s,q)$. Thus (t,p) *contains* (s,q) iff q contains p. If p=q, then (t,p)=(s,q); if q is longer than p, then we write $(t,p)>(s,q)$. Notice if $(t,p) \geq (s,q)$ and $(t',p') \geq (s,q)$, then $p \supset p'$ or $p' \supset p$. Notice also that a constant or a variable has the longest position specification. The set of term occurrences of C is denoted by T(C). Although we do not use the language of trees, some properties are intuitively clearer by thinking of clauses as trees.

2.2 Consistent term mappings

Given a clause C=P(f(g(x))). We can construct a new clause P(f(h(z))) by replacing (g(x),<1,1>) by (h(z),<1,1>). This replacement can be described by a mapping {(g(x)/h(z),<1,1>)}. Notice that changing g(x) to h(z) has the effect of changing f(g(x)) to f(h(z)). {(f(g(x))/f(h(z)),<1>)} and {(g(x)/h(z)),<1,1>),f(g(x))/f(h(z)),<1>)} are also allowed. The last replaces two nested occurrences *simultaneously*.

Definition. An *abstract term occurrence* is a pair of term and position (t,p) which is not yet associated with a clause. For a given clause C, a mapping θ from a subset T of T(C) to a set of abstract term occurrences is called a *consistent term mapping* (*CTM*) if the following two conditions are satisfied:

1) If (t,p) is in T, then (t,p)θ has also position p, i.e. θ preserves positions.
2) If (t,p) and (s,q) are both in T and $(t,p) \geq (s,q)$, then $(t,p)\theta \geq (s,q)\theta$, i.e. if (t,p)θ=(t',p) and (s,q)θ=(s',q), then in position q-p of t' we find s'.

A subset T of T(C) is called *minimal* if for every two (t,p), (s,q) in T, neither $p \supset q$ nor $q \supset p$. If a CTM θ is defined on a minimal subset T, then we can construct a new clause by replacing (t_i,p_i) in C by $(t_i,p_i)\theta$. Notice that the replacings do not interfer each other.

Theorem 1. Let θ be a CTM defined on a subset T of T(C). Let Tθ be the set of images of θ. There is a minimal subset S of T such that θ restricted to S induces a clause C' with $T(C') \supseteq T\theta$.
Proof. Let S be the subset of T which contains the shortest position specifications, i.e. $(t,p) \in S$ iff there is no $(t',p') \in T$ such that $(t',p')>(t,p)$. We use S to construct C' by replacing every (t,p) in S by its image.

Remark. The clause C' defined in the proof above is called the clause *induced by* θ and we denote it by Cθ. Notice that the inverse of a CTM θ is also a CTM. Let T be the domain of θ. The positions and terms in Cθ can be characterized by 1)-4) as follows.

1) If $(s,q) \in T(C)$ and if there exists no $(t,p) \in T$ such that $(t,p) \geq (s,q)$ or $(s,q) \geq (t,p)$, then (s,q) is in Cθ.

2) If (s,q)∈T(C) and there is a (t,p)∈T such that (s,q)≥(t,p) and there is no (t’,p’)∈T such that (t’,p’)>(s,q), then the q-th term is defined in Cθ and it can be found by replacing all subterm occurrences of (s,q) from T simultaneously by their images.
3) If (s,q)∈T(C) and (t,p),(t’,p’)∈T such that (t,p)≥(s,q)≥(t’,p’), then q is defined on Cθ. The q-th term in Cθ is the subterm occurrence of (t,p)θ in position q-p.
4) Suppose (t,p)∈T and (t,p)θ=(t’,p) and there exists no (s,q)∈T such that (t,p)>(s,q). If t’ has a subterm s' in q', then (s',pq') is defined in Cθ. The position pq’ may not be defined on C.

Notice also if (s,q)∈T(C) and there is a (t,p)∈T such that (t,p)>(s,q) and there is no (t’,p’)∈T such that (s,q)≥(t’,p’), then the position q may not be defined in Cθ.

The following theorem gives a criterium when two clauses are the same.
Theorem 2. If θ is a CTM defined on T, a subset of T(C), and if T' is a subset of T(C) that satisfies the following conditions:
1) for any (t,p)∈T, there is a (s,q) in T' such that (s,q)≥(t,p).
2) every (s,q)∈T' has one of the following two properties:
   - it contains a subterm occurrence from T, i.e. there is a (t,p)∈T such that (s,q)≥(t,p);
   - there is no (t,p)∈T such that (s,q)≥(t,p) or (t,p)≥(s,q).

Then q is a position defined in Cθ. If we define a CTM $\underline{\theta}$ with domain T' by (s,q)$\underline{\theta}$=the q-th term of Cθ, then C$\underline{\theta}$=Cθ.

2.3 Substitutions and inverse substitutions

Let C be a clause. A *substitution* θ from C is a CTM defined on the set of all variable occurrences of C such that if (v,p)θ=(t,p) and (v,q)θ=(s,q), then t=s. Notice that a substitution has a minimal domain. A substitution induces a mapping defined on all variables, i.e. if (t,p)=(v,p)θ, then t is defined as the image of v. We use also θ to denote this mapping. If there is a substitution θ from C to C', i.e if C'=Cθ, then we write C≥C'. If there is a substitution from C to C' and there is no substitution from C' to C, then we say C>C' and C is a *generalization of* C'. A substitution θ can be denoted by $\{v_1/t_1,\ldots,v_n/t_n\}$if $v_i\theta=t_i$. We may omit v_i/v_i. If θ is a substitution, then its inverse θ^{-1} is called an *inverse substitution*. We can also define inverse substitution without first having a substitution. A CTM μ defined on a subset T of T(C) is an *inverse substitution* iff the following conditions are satisfied: the domain T is minimal; the images are variable occurrences; if (t,p)μ=(v,p) and (s,q)μ=(v,q), then s=t; for every variable occurrence (v,q) of C, there is a (t,p) in T such that (t,p)≥(v,q). The last condition guarantees that the inverse of μ is a substitution. A substitution θ can be extended to a CTM $\underline{\theta}$ which has all T(C) as domain. Such a $\underline{\theta}$ is called *maximal extension* of θ. For an inverse substitution μ defined on a subset T of T(C), there is also a *maximal extension* $\underline{\mu}$. If (t,p)≥(s,q) where (s,q) is in the domain of μ, we define (t,p)$\underline{\mu}$=(t',p) by replacing all subterm occurrences of (t,p) which are in the domain of μ by their images. A term occurrence (t,p), which does not contain any subterm occurrence from the domain of μ and is not a subterm occurrence of any element in the domain of μ, is also in the domain of the extention $\underline{\mu}$ and (t,p)$\underline{\mu}$=(t,p). Notice that both $\underline{\theta}$ and $\underline{\mu}$ satisfy the conditions of theorem 2, thus they induce the same clause as the original CTM. There are also other CTM's with the same property.

Corollary 1. Let μ be a substitution from C and $\underline{\mu}$ be the maximal extension of μ defined on T(C). If θ is the restriction of $\underline{\mu}$ to a subset T of T(C) and for every variable occurrence (v,q) in T(C), there is a (t,p) in T such that (t,p)≥(v,q), then θ induces also Cμ, i.e. Cμ=Cθ.

Corollary 2. Let μ be an inverse substitution from C and $\underline{\mu}$ be the maximal extension of μ. Let θ be a CTM which is the restriction of $\underline{\mu}$ to a subset T of T(C) and for every (s,q) in the domain of μ there is a (t,p) in T such that (t,p)≥(s,q), then Cμ=Cθ.

3 Term partitions and comparisons of partitions

Definition. Let C be a given clause. A subset T of T(C) is *admissible* if it satisfies the following conditions:

1) T is minimal.
2) If (w,q) is an variable occurrence in C, then there is a (t,p) in T such that $(t,p)\geq(w,q)$.

A *term partition* $\prod$ (or simply a partition on C) is a partition $\{B_1,B_2,...,B_k\}$ of an admissible set T such that every *block* B_i contains only occurrences of one term. T is also denoted by dom($\prod$).

If we have an inverse substitution μ then we can consider a term partition $\prod$ which is defined on the domain of μ: a block B_v in $\prod$ is defined as the set of inverse images of a variable v. Notice that a term partition induces also a inverse substitution. It is possible to define more inverse substitutions from one term partition, then the two clauses induced by the same partition are equivalent, i.e. they differ only by variable names. Thus we use C($\prod$) to denote one of such induced clauses. We want to define a partial ordering $\geq$ in partitions on C such that $\prod\geq\Omega$ iff $C(\prod)\geq C(\Omega)$. If $C_1\geq C_2$, then for every (w,q) variable in C_2, there must be an (v,p) in C_1 such that $(v,p)\sigma$ contains (w,q) as subterm. In this situation w has relative position q-p in $(v,p)\sigma$. If there is also (v,p') in C_1, then $(v,p')\sigma$ contains also a variable w in the position q-p(=q'-p'). We try to translate such concepts to relations between partitions. For example,

$C=P(f(g(h(x),y))),g(h(x),y),k(a,h(x)))$
$C_2=P(f(g(w,y)),g(w,y),k(a,w))$
$C_1=P(f(u),u,v)$

To find C_2, we need the following partition Ω:

$D_1=\{(h(x),<1,1,1>),(h(x),<2,1>),(h(x),<3,2>)\}$, D_1 / w;
$D_2=\{(y,<1,1,2>),(y,<2,2>)\}$,$D_2$ / y

To find C_1, we need the following partition $\prod$:

$B_1=\{(g(h(x),y),<1,1>),(g(h(x),y),<2>)\}$, B_1 / u;
$B_2=\{(k(a,h(x)),<3>)\}$, B_2 / v.

Define $(u,<1,1>)\sigma=(g(w,y),<1,1>)$ and $(u,<2>)\sigma=(g(w,y),<2>)$ and $(v,<3>)\sigma=(k(a,w),<3>)$. The first two elements in D_1 are related to B_1. For (h(x),<1,1,1>) in D_1 there is (g(h(x),y),<1,1>) in B_1 containing it as subterm in position <1> and for (h(x),<2,1>) in D_1 there is (g(h(x),y),<2>) in B_1 containing it as subterm in position <1>. This is also the position of w in $v\sigma$. For (h(x),<3,2>) in D_1 there is (k(a,h(x)),<3>) in B_2 containing it as subterm in position <2>. This is also the position of w in $v\sigma$.

Definition. Let C be given clause and $\prod$, Ω be partitions defined on T, S, subsets of T(C), respectively. We say $\prod\geq\Omega$, $\prod$ is *higher than* Ω, if

1) For every (s,q) in S, there is (t,p) in T such that $(t,p)\geq(s,q)$.
2) Let (t,p) be in block B of $\prod$ and (s,q) be in block D of Ω. If $(t,p)\geq(s,q)$ and
   $B=\{(t,p_1),..,(t,p_n)\}$, $D=\{(s,q_1),...,(s,q_m)\}$
   then $m\geq n$ and by reordering van indices, we have $p_1=p$, $q_1=q$ and q_i-p_i=q-p for all i=1,...,n.

Condition 2) can be interpreted as: D contains the set of all subterm occurrences of occurrences in B at position q-p. If two subsets B and D of T(C) have this relation, we say $B\geq D$.

Theorem 3. Let $\prod$ and Ω be two term partitions on C. Then $\prod\geq\Omega$ iff every D of Ω is a disjoint union of D_1,... , D_k such that for every D_j there is a unique B_j in $\prod$ such that $B_j\geq D_j$ and $|B_j|=|D_j|$, i.e. the number of elements in B_j and D_j are the same. The way to divide a block D which satisfies this condition is also unique.

Proof. Let (s,q_1) be in D. There is a (t,p_1) in T such that $(t,p_1)\geq(s,q_1)$ according to 1). According to 2) there is a relationship between the the elements in $B_1=\{(t,p_1),\ldots,(t,p_n)\}$ and a subset $D_1=\{(s,q_1),\ldots,(s,q_n)\}$ of D which satisfy q_i-p_i=q_1-p_1 for i=1,...,n.

Remark. Let $\prod\geq\Omega$ be two partitions on C. Then

1) If $(s,q)\in$ dom(Ω), then there is a unique $(t,p)\in$ dom($\prod$) such that $(t,p)\geq(s,q)$.

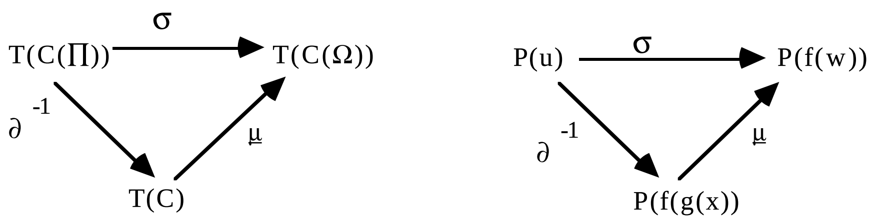
2) If $(t,p)\in dom(\Pi)$ and $(s,q)\in dom(\Omega)$, then $(s,q)>(t,p)$ is not possible.
3) If B contains term occurrences without variable subterms, then there might not exist a D in Ω such that $B\geq D$.
4) To compensate 3) and also for the convenience of the following section, we can define for a constant a, the *constant block* D_a w.r.t. Ω as follows:

$$D_a=\{(a,q)\in T(C) \mid \text{there exists no } (t,p)\in dom(\Omega) \text{ such that } (t,p)\geq(a,q)\}$$

Let us call *extension of* Ω as Ω extended by such non-empty constant blocks. In fact, every constant or variable occurrence of C is a subterm of an occurrence in a block in the extension. For every block B in Π, there is a block D in the extension of Ω such that $B\geq D$. If we replace Ω and Π in the above theorem by the extensions of Ω and Π, respectively, then the theorem holds too.
5) Notice that if $\Pi\geq\Omega$ and $\Omega\geq\Pi$, then $\Pi=\Omega$. Thus we say $\Pi>\Omega$ if $\Pi\geq\Omega$ and $\Pi\neq\Omega$.

Theorem 4. Let C be a given clause. Let ∂ and μ be two inverse substitutions which induce term partitions Π and Ω on C. If there is a substitution θ from $C\partial$ to $C\mu$, then $\Pi\geq\Omega$.

We use an example to show how the proof of the following theorem works. Let $C=P(f(g(x)))$, $C(\Pi)=P(u)$ and ∂ be the inverse substitution from C to $C(\Pi)$. Let $C(\Omega)=P(f(w))$ and let μ be the inverse substitution from C to $C(\Omega)$. Let $\underline{\mu}$ be also the maximal extension of μ and ∂^{-1} be the substitution from $C(\Pi)$ to C which is the inverse of ∂. The composition of ∂^{-1}: $u \rightarrow f(g(x))$ and $\underline{\mu}$: $f(g(x)) \rightarrow f(w)$ defines σ: $u \rightarrow f(w)$. This CTM induces a clause and we can prove it is just $C(\Omega)$ by using corollary 2. That means σ is the substitution which we are looking for.



Theorem 5. Let Π and Ω be two term partitionson C. If $\Pi\geq\Omega$, then $C(\Pi)\geq C(\Omega)$.

4 More about ordering relations among partitions

4.1 Partitions induced by partitions

Let Π and Ω be partitions on a clause C and $\Pi\geq\Omega$. Then there is a partition Π' on $C(\Omega)$ which induces also $C(\Pi)$ because $C(\Pi)\geq C(\Omega)$. If μ is the inverse substitution from C to $C(\Omega)$ and $\underline{\mu}$ is the maximal extention of μ, then a block is in Π' iff it is the image set of a block in Π under $\underline{\mu}$.

Theorem 6. Let Ω be a partition on a clause C. There is a one to one correspondence which preserves order relations between the partitions on $C(\Omega)$ and the partitions on C which are higher than or equal to Ω. If Π' is on $C(\Omega)$ and it corresponds with Π on C, then $C(\Pi)=C(\Pi')$.

4.2 Minimal generalizations and least higher partitions

Definition. Let C_1, C_2 be given clauses and $C_2>C_1$. We call C_2 is a *minimal generalization* of C_1 if for any other C_3 with $C_3>C_1$ we have $C_3\geq C_2$ or C_3 and C_2 are *incomparable*. Consider the set of all partitions on a clause C. We can define the *least higher partition* of a partition in analogous way.

If $C_1=C(\Pi_1)$, then finding a minimal generalization C_2 correspond with a partition Π_2 which is least higher than Π_1. Reynolds[6] and Plotkin[5] have proved some properties about the lattice structure of atomic formulas and Reynolds[6] has claimed the existence of a total chain of atomic formulas from C' to C when C'≥C. He has stated only two kinds of minimal generalizations without mentioning the minimal generalizations by changing constants into variables. This corresponds to the least higher partitions of the first kind below. Thus his total chain $C'>C_1>C_2>\ldots>C_n>C$ is not complete. Furthermore, he builds the chain from C' to C instead of from C to C' and he needs always the concrete clause C_i to build C_{i+1}. We build an ascending chain of partitions on a given clause and we do not have to construct the C_i 's or the partitions on C_i 's concretely. All the proofs can also be done by considering C alone.

Least higher partition of the first kind

The follwing theorem can be translated to the language of clauses as follows: by replacing some constants in C by a new variable we obtain a minimal generalization of C. For example, P(f(x),g(x),y,f(a)), P(f(x),g(x),y,f(y)) are minimal generalization of P(f(x),g(x),a,f(a)). .

Theorem 7. Let $B=\{(a,q_1),..,(a,q_n)\}\neq\phi$ be a set of a-occurrences in a clause C and Ω be the trivial partition defined by variables in C. Then the partition ∏ defined by $\Pi=\Omega\cup\{B\}$ is least higher than Ω.

Least higher partition of the second kind

Minimal generalization can be obtained by changing some occurrences of a variable to a new variable name. For example, $P(x_1,f(y),g(x),g(x))$ is a minimal generalization of P(x,f(y),g(x),g(x)).

Theorem 8. Let Ω be the trivial partition on C and D be the block in Ω defining a variable v. If B_1 is a proper subset of D and $B_2=D-B_1$, then $\Pi=(\Omega-\{D\})\cup\{B_1,B_2\}$ is least higher than Ω.

Least higher partition of the third kind

We can construct a minimal generalization of a clause by replacing some kind of compound term occurrences by a new variable. For example, P(f(z),h(z),a) is a minimal generalization of P(f(g(x,y),h(g(x,y)),a). .

Theorem 9. Let Ω be the trivial partition on C and $t=f(v_1,\ldots,v_m)$ such that $v_i\neq v_j$ if i≠j. Let $B=\{(t,p_1),...,(t,p_n)\}$ be the set of all t-occurrences and let $D_1,\ldots,D_m$ be the blocks defined by $v_1,...,v_m$. If every v_i-occurrence is always a subterm occurrence of such a (t,p_i), then the partition $\Pi=(\Omega-\{D_1,...,D_m\})\cup B$ is least higher than Ω.

According to the theorems above we can construct a sequence of clauses C, C_1, $C_2,\ldots$, C_n such that C_{i+1} is a minimal generalization of C_i. This process has the disadvantage that every step is based on a new clause which we have just constructed. We need thus the following corollary and theorem 10.

Corollary 3. Consider a partition Ω on C.

(*a*) Let B_a be a non-empty subset of a constant block w.r.t. Ω. Then $\Omega_1=\Omega\cup\{B_a\}$ is least higher than Ω.

(*b*) Let D be a block in Ω and $D=B_1\cup B_2$ where $B_1,B_2\neq\phi$. Then $\Omega_1=(\Omega-\{D\})\cup\{B_1,B_2\}$ is least higher than Ω.

(*c*) Suppose $D_1,\ldots, D_k$ are blocks in Ω which satisfiy the following conditions:

1) If $(t,p),(t,q)\in D_i$, where $p=(p_1,\ldots,p_{n-1},p_n)$ and $q=(q_1,\ldots,q_m)$, then $p_n=q_m$. Furthermore, if $p'=(p_1,\ldots,p_{n-1})$ and $q'=(q_1,\ldots,q_{m-1})$, then p’ and q’ are positions of the same term. If B_i is the set of all such p’,q’, etc. , these conditions imply that D_i is the set of all subterm occurrences of elements in B_i in the same position.

2) $B_i=B_j$ for all i, j=1,…,k. Let B be used to denote these B_i's, then the term in B has only k arguments.

Then $\Omega_1=(\Omega-\{D_1,\ldots,D_k\})\cup\{B\}$ is least higher than Ω.

Theorem 10. Let C be a given clause and Ω, $\prod$ be partitions on C such that $\prod > \Omega$. We can find a finite sequence of partitions $\Omega_0, \Omega_1, \ldots, \Omega_n$ such that $\Omega_0 = \Omega$, $\Omega_n = \prod$ and Ω_{i+1} is least higher than Ω_i for every i.

Construction. We choose first a block $E = \{(t,p_1),\ldots,(t,p_n)\}$ in $\prod$ and then construct Ω_i's in the following way:

(1) If $(t,p_1) \geq (a,q_1)$ where (a,q_1) is from a constant block D_a w.r.t. Ω, then there are $(a,q_2),\ldots,(a,q_n)$ in D_a such that $(t,p_i) \geq (a,q_i)$ and $q_i - p_i = q_1 - p_1$ for $i=2,\ldots,n$. Define $B_a = \{(a,q_1),\ldots,(a,q_n)\}$ and $\Omega_1 = \Omega \cup B_a$. Ω_1 is then least higher than Ω and a constant block w.r.t Ω_1 is either $D_a - B_a$ or some other constant block w.r.t Ω. We continue this process until every constant in t has been considered and we come to Ω_k.

(2) Compare the blocks in Ω_k and E. Suppose $D = \{(s,q_1),\ldots,(s,q_n),(s,q_{n+1})\ldots\}$ in Ω_k, $E \geq D$ such that $q_i - p_i = q_j - p_j$ for i, j=1,…, n. Let $B_1 = \{(s,q_1),\ldots,(s,q_n)\}$ and $B_2 = D - B_1$. Define $\Omega_{k+1} = (\Omega_k - \{D\}) \cup \{B_1, B_2\}$, then Ω_{k+1} is least higher than Ω_k. We can continue this process until in Ω_m where there is no block D, $E \geq D$ such that D has more elements than E.

(3) Let $D_1,\ldots,D_k$ be all the blocks in Ω_m such that $E \geq D_i$. D_i is actually the set of all subterm occurrences on a position, i.e. there is a r_i for every i, such that

$D_i = \{(s_i, p_1 r_i),\ldots,(s_i, p_i r_i)\}$.

We choose one i such that r_i is the longest. Suppose r_1 is the longest and $r_1 \neq \langle\rangle$. Let r_1' be the subsequence of r_1 which is one number less than r_1, then $p_1 r_1', .., p_n r_1'$ are all occurrences of the same term. We define a new block

$B = \{(s_1', p_1 r_1'),\ldots,(s_1', p_n r_1')\}$.

Then our new partition is $\Omega_{m+1} = (\Omega_m - \{D_1,\ldots,D_j) \cup B$ if $B \geq D_1, D_2, \ldots, D_j$ (i.e. $r_1' = r_2' = .. = r_j'$). We go on with same process until we we notice E self is a block in such Ω_k. Now we choose another block than E in $\prod$ and repeat (1), (2), (3). We are ready when every block is processed in the way above.

Example. Let C, Ω and $\prod$ be given as below, we want to construct a sequence of partitions as in the theorem. For simplicity, we write p instead of (t,p).

$C = P(f(g(u,a,a),u)),\ g(u,a,a))$

$\prod$: $E_1 = \{\langle 1,1\rangle, \langle 2\rangle\}$, $E_2 = \{\langle 1,2\rangle\}$.

$\Omega_0 = \Omega$: $D_1 = \{\langle 1,1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle\}$, $D_a = \{\langle 1,1,2\rangle, \langle 1,1,3\rangle, \langle 2,2\rangle, \langle 2,3\rangle\}$. We consider first E_1.

Ω_1: Because $E_1 \geq D_a$, let $B_1 = \{\langle 1,1,2\rangle, \langle 2,2\rangle\}$. Then $\Omega_1 = \Omega \cup \{B_1\} = \{B_1, D_1\}$ from (1). The constant block of Ω_1 is $B_a = D_a - B_1 = \{\langle 1,1,3\rangle, \langle 2,3\rangle\}$.

Ω_2: Because $E_1 \geq B_a$, let $B_2 = \{\langle 1,1,3\rangle, \langle 2,3\rangle\}$. Then $\Omega_2 = \Omega_1 \cup \{B_2\} = \{B_1, B_2, D_1\}$ from (1).

Ω_3: Compare E_1 with D_1, we have $D_1 = B_3 \cup B_4$ where $B_3 = \{\langle 1,1,1\rangle, \langle 2,1\rangle\}$ and $B_4 = D_1 - B_3 = \{\langle 1,2\rangle\}$. Thus $\Omega_3 = (\Omega_2 - \{D_1\}) \cup \{B_3, B_4\} = \{B_1, B_2, B_3, B_4\}$ from (2).

Ω_4: $E_1 \geq B_1, B_2, B_3$. The corresponding r_1, r_2, r_3 are $\langle 2\rangle, \langle 3\rangle, \langle 1\rangle$.

Because $r_1' = r_2' = r_3' = \langle\rangle$, we have constructed a $B_5 = \{\langle 1,1\rangle, \langle 2\rangle\}$ which is E_1.

Thus $\Omega_4 = \Omega_3 - \{B_1, B_2, B_3\} \cup \{B_5\} = \{B_4, B_5\} = \prod$.

4.3 Supremum of partitions

Definition. Consider all the partitions on C. A partition $\prod$ is called *supremum* of partitions Ω_1 and Ω_2 if $\prod \geq \Omega_1$, $\prod \geq \Omega_2$ and for any partition Σ which satisfies $\Sigma \geq \Omega_1, \Sigma \geq \Omega_2$, we have $\Sigma \geq \prod$.

Theorem 11. Let C be a given clause and Ω_1 and Ω_2 be partitions on C, we can construct the supremum $\prod$ of Ω_1 and Ω_2 on C.

Construction. We perform the following steps:

(1) Construct $V = dom(\Sigma)$ as follows.

$V = \{p \in dom(\Omega_1) \cup dom(\Omega_2) \mid$ there exists no $q \in dom(\Omega_1) \cup dom(\Omega_2)$ such that q is shorter than p$\}$

(2) If $p \in V \cap B$ where B is a block in Ω_1, construct a block E for $\prod$ which contains p as follows.

To begin with E should be a subset of V∩B. Thus consider a p'∈V∩B and determine if p'∈E. If for every D, a block in Ω_2 which contains an element q∈D with q⊃p, there is also a q'∈D such that q'⊃p' and q-p=q'-p', and furthermore, for every D', a block in Ω_2 which contains an element q'∈D' with q'⊃p', there is a q∈D' such that q⊃p and q-p=q'-p, then we add p' to E. If we have considered all elements in V∩B, then we have a new block E for ∑. The same way can be used for constructing a block containing an element in V∩dom(Ω_2).

5 Application

Muggleton and Buntine[2] have introduced absorption of inverse resolution for machine learning: given a literal C_1 and a clause C, how to find a C_2 such that C becomes the resolvent of C_1 and C_2. The improved algorithm in [3],[4] constructs a substitution θ_1 from C_1 and then all partitions on $C \vee \sim C_1\theta_1$. They induce C_2's such that $C_2\theta_2 = C \vee \sim C_1\theta_1$ for some θ_2. We can even prove that two C_2's which are based on different θ_1 are incomparable (under certain not too restricted conditions). Thus first investigating partitions w.r.t. one θ_1 and then another is a systematic approach. However, these constructions of partitions on $C \vee \sim C_1\theta_1$ is not efficient and directed. We want for example to generalize C_2 one step every time or we want to find a supremum for two different C_2's. The constructions in this article help us to solve such problems.

References

1 Stephen Muggleton. Inductive Logic Programming. First Conference on Algorithmic Learning Theory, Ohmsha, Tokyo, October 1990.

2 Stephen Muggleton & Wray Buntine. Machine Invention of First-order Predicates by Inverting Resolution. Proceedings of the 5th International Conference on Machine Learning, Morgan Kaufmann, pp. 339-351, 1988.

3 Shan-Hwei Nienhuys-Cheng. Consequent Functions and Inverse Resolutions. Report Eur-CS-90-03, Erasmus University, Rotterdam, Netherlands, May 1990.

4 Shan-Hwei Nienhuys-Cheng, Peter Flach. Consistent Term Mappings, Term Partitions and Inverse Resolutions, to appear in proceedings of EWSL-91, March, 1991, Portugal.

5 Shan-Hwei Nienhuys-Cheng. Term partitions and minimal generalizations of clauses Report, Erasmus University, Rotterdam, Netherlands, January, 1991.

6 Gordon D. Plotkin. A Note on Inductive Generalisation. *Machine Intelligence 5*, B. Meltzer & D. Michie (eds.), Edinburgh University Press, 1970.

7 John C. Reynolds. Transformational Systems and the Algebraic Structure of Atomic Formulas. *Machine Intelligence 5*, B. Meltzer & D. Michie (eds.), Edinburgh University Press, 1970.