A topological study of the upward refinement operators in ILP

M.A. Gutiérrez-Naranjo, J.A. Alonso-Jiménez ⋆, and J. Borrego-Díaz ⋆

Dpto. Ciencias de la Computación e Inteligencia Artificial – Universidad de Sevilla
{magutier,jalonso,jborrego}@cica.es
WWW home page: http://www-cs.us.es/{∼naranjo, ∼jalonso, ∼jborrego}

Abstract. In this paper, General Topology and ILP are brought together. The closeness between clauses is formalized using a non-metric topology which can be seen as the topological translation of the subsumption order: the Alexandrov topology. The definitions of the upward refinement operators in ILP are formalized as mappings $\varphi : \mathbb{C}_R/\!\sim\, \to \mathbb{C}/\!\sim$ where $\mathbb{C}/\!\sim$ and $\mathbb{C}_R/\!\sim$ are the quotients obtained from the set of clauses $\mathbb{C}$ and the set of reduced clauses $\mathbb{C}_R$ via the equivalence relation based on subsumption. Our main result is the proof of the continuity of these operators on the Alexandrov topology.

1 Introduction

In the last years a growing interest in the study of topological properties of the Logic Programming operators has arisen ([2, 12–14]). However, to the best of our knowledge, no study of the Inductive Logic Programming (ILP) operators in the General Topology frame has been done.

In this paper we present a first approximation in ILP to the concept of closeness among clauses when this *closeness* is settled by a non–metric topology. Usually, the intuition leads us to a notion of closeness based on a spatial representation. So, two points A and B in the plane are closer than A and C if the length of the segment which joins A and B is lesser than the length of the segment which joins A and C. This idea of closeness is associated to the concept of *distance* as a function mapping a pair of points on a real number.

It was M. Fréchet in 1906 [3] who gave the conditions to consider a mapping $d : X \times X \to \mathbb{R}$ as a distance

- $(\forall x \in X)\, d(x, x) = 0$
- $(\forall x, y \in X)\, d(x, y) = 0 \Rightarrow x = y$
- $(\forall x, y \in X)\, d(x, y) = d(y, x)$ *(the condition of symmetry)*
- $(\forall x, y, z \in X)\, d(x, z) \leq d(x, y) + d(y, z)$ *(the triangle inequality)*

since then, that definition has been the base to the development of the theory of metric spaces.

⋆ Work partially supported by DGES (Spain), projects PB96–0098-C04–04 and PB96–1345

But in General Topology it is well known that endowing a set with a distance mapping is not the unique way to formalize the closeness in a set. We can easily find situations where the symmetry condition is not verified, as in problems of nets based on flows or costs. In this cases Fréchet's conditions are not appropriate to formalize the notion of closeness.

If our unique interest is to endow the set with a distance mapping without considering any relation between the elements, we can always consider the *discrete distance*

$$d_d(A,B) = \begin{cases} 0 & \text{if } A = B \\ 1 & \text{if } A \neq B \end{cases}$$

which satisfies the Fréchet's conditions to be a distance, but it would hardly have a practical usefulness.

In general, if we consider a non–symmetric relation on a set (as flows or costs) it will be very difficult to find a distance mapping compatible with that relation. In this paper we consider the set of clauses $\mathbb{C}$ with the subsumption relation and show that the natural formalization of the idea of closeness is given by the Alexandrov topology[1].

Technically, this topology is the finest topology T_0 whose natural order (the so–called specialization order) is equal to the order which generates it, in this case, the subsumption order, and it allows properties related to subsumption to be translated into a purely topological context.

The main result is the continuity of the upward refinement operators on this topology. Roughly speaking, this means that if two clauses are close, then their refined clauses are close as well. For that we not only need to say what means closeness but also to give a strict formalization of the operator which allows to handle it as a mathematical object.

The operators will be formalized as mappings

$$\varphi : \mathbb{C}_R/\!\sim\, \to \mathbb{C}/\!\sim$$

where $\mathbb{C}/\!\sim$ and $\mathbb{C}_R/\!\sim$ are the quotients obtained from the set of clauses $\mathbb{C}$ and the set of reduced clauses $\mathbb{C}_R$ via the equivalence relation based on subsumption.

The paper is organized as follows: In section 2 we give a brief exposition of some basic definitions about logic and tree theory. In section 3, the specialization order and the Alexandrov topology are studied and the narrow relation between partial orders and the Alexandrov topology is shown. In section 4 our main result is stated and proved. First, we present a study of substitution and grafts and taking it as base, we give a new formalization of the upward refinement operators and prove our main result: these operators are continuous on the Alexandrov topology. Section 5 completes this result: the operator based on deleting a literal from a clause is continuous on that topology, as well. In section 6 the conclusions are presented and the paper is finished with the bibliography and an appendix containing several technical results used in the main theorems.

---

[1] A good introduction to General Topology can be found in [19].

The principal references have been [1, 10] for the foundations of Logic Programming and the subsumption order, [4, 5] for the tree theory, [6, 17] for the Alexandrov topology and [19] for General Topology.

2 Preliminaries

From now on, we will consider some fixed countable first-order language $\mathcal{L}$ with at least one function symbol. Var, $Pred$, $Term$ and Lit are, respectively, the sets of variables, predicate symbols, terms and literals of $\mathcal{L}$. A *clause* is a finite set of literals and $\mathbb{C}$ is the set of all clauses. If A is a set, then $|A|$ is the cardinal of A, $\mathcal{P}A$ its power set, and if f is a mapping, $f\restriction_A$ is the restriction of f to A.

Let $S \subseteq Var$ be a finite set of variables. A *substitution* is a mapping $\theta : S \to Term$ such that $(\forall x \in S)[x \neq \theta(x)]$. We will use the usual notation $\theta = \{x/t : x \in S\}$, (where $t = \theta(x)$ or $x\theta = t$) and $Dom(\theta)$ for the set S. If C is a clause and $Dom(\theta) \subseteq Var(C)$, then θ is called *applicable* to C. Obviously $\theta\restriction_{Var(C)}$ is applicable to C.

A *renaming* is a 1–1 substitution θ such that $(\forall x \in Dom(\theta))\,[x\theta \in Var]$. If θ is a renaming and L is a literal, then L and $L\theta$ are called *variants.* Analogously, if C is a clause and θ is a renaming, C and $C\theta = \{L\theta \mid L \in C\}$ are *variants.*

Let L_1, L_2 be literals. L_1 is called an *instance* of L_2, and it is denoted by $L_1 \geq L_2$, if there exists a substitution θ such that $L_1 = L_2\theta$. Let L_1, L_2 be literals, then L_1 and L_2 are variants iff L_1 is an instance of L_2 and L_2 is an instance of L_1.

We follow Gallier [4] and Huet [5] for the issues related to positions and grafts. A *position* is a finite sequence of n positive integers with $n \geq 1$. Let $\mathbb{N}^+$ denote the set of all positions. Two positions u and v are *independent* if u is not a prefix of v and v is not a prefix of u. A set of positions P is *independent* if $(\forall u, v \in P)[u \neq v \Rightarrow\ u$ and v are independent].

Let $Pos(L)$ denote the set of all positions in L. If $u \in Pos(L)$, L/u denotes the term rooted at u in L. $Pos(L, t)$ denotes the set of all positions of the term t in L. If t does not occur in L, then $Pos(L, t) = \emptyset$. Finally, if $P \subseteq Pos(L)$, then $L/P = \{L/u \mid u \in P\}$. Any other notation has the usual sense.

We finish this section recalling the definitions of a graft at a position and at a set of positions.

Definition 1. *Let $\{t_1, \ldots, t_n\}$ be a set of terms, p a n–ary function symbol or predicate symbol, $s \in Term$ and u a position. The* graft *of s at position u in $p(t_1, \ldots, t_n)$, denoted by $p(t_1, \ldots, t_n)[u \leftarrow s]$, is defined as follows:*

- *If $length(u) = 1$, then*

$$p(t_1, \ldots, t_n)[u \leftarrow s] = p(t_1, \ldots, t_{u-1}, s, t_{u+1}, \ldots, t_n)$$

- *If $u = i\hat{\ }v$ with $i \geq 1$ and $v \in \mathbb{N}^+$, then*

$$p(t_1, \ldots, t_n)[u \leftarrow s] = p(t_1, \ldots, t_{i-1}, t_i[v \leftarrow s], t_{i+1}, \ldots, t_n)$$

Note that if $u, v \in \mathbb{N}^+$ are independent and $s \in Term$, then

$$(L[u \leftarrow s])[v \leftarrow s] = (L[v \leftarrow s])[u \leftarrow s]$$

Definition 2. *Let L be a literal, s a term and $P \subseteq Pos(L)$ an independent set of positions. The graft of s at P in L is defined as follows:*

- *if $P = \emptyset$, then $L[P \leftarrow s] = L$*
- *if $u \in P$ and $P' = P \backslash \{u\}$, then $L[P \leftarrow s] = (L[P' \leftarrow s])[u \leftarrow s]$.*

3 The specialization order and the Alexandrov topology

This section gives a brief description of the Alexandrov topology. This topology appears in a natural way in other fields of computer science ([15, 17]) and, in a certain sense, it is the natural way to see a partial order as a topological space. We begin with some basic definitions from General Topology.

A topology on a set X is a collection $\mathcal{T}$ of subsets of X, called the open sets, satisfying: (a) Any union of elements of $\mathcal{T}$ belongs to $\mathcal{T}$ (b) any finite intersection of elements of $\mathcal{T}$ belongs to $\mathcal{T}$ and (c) $\emptyset$ and X belong to $\mathcal{T}$. If $\mathcal{T}$ is a topology on X, the pair $\langle X, \mathcal{T} \rangle$ is called a topological space. If $\langle X, \mathcal{T} \rangle$ is a topological space and $x \in X$, a neighborhood of x is a set U which contains an open set containing x.

A topology on the set X is T_0 iff whenever x and y are distinct points in X, there is an open set containing one and not the other.

Let $\langle X, \mathcal{T}_X \rangle$ and $\langle Y, \mathcal{T}_Y \rangle$ be topological spaces and let $f : X \to Y$. Then f is continuous at $x_0 \in X$ iff for each neighborhood V of $f(x_0)$ in Y, there is a neighborhood U of x_0 in X such that $f(U) \subseteq V$. We say f is continuous iff f is continuous at each $x_0 \in X$.

Definition 3. *Let A and B be two sets and R_A and R_B two binary relations on A and B. The mapping $f : \langle A, R_A \rangle \to \langle B, R_B \rangle$ is called a* morphism *if $(\forall x, y \in A)[x R_A y \to f(x) R_B f(y)]$.*

The next equivalence is trivial. It will be used in the proof of Theorem 2.

Lemma 1. *Let $f : X \to Y$ be a mapping between two partial orders. Then f is a morphism iff $(\forall x \in X)[f(\uparrow x) \subseteq \uparrow f(x)]$, where $\uparrow c = \{z \mid c \leq z\}$.*

Definition 4. *Let $\langle X, R \rangle$ be a partial order. A set $S \subseteq X$ is called a* final segment *if $(\forall x, y)[x \in S \wedge x R y \to y \in S]$.*

Definition 5. *Given a topological space $\langle X, \mathcal{T} \rangle$, the* specialization order *based on the topology $\mathcal{T}$ is defined as*

$$(\forall x, y \in X)\, [x R y \Leftrightarrow (\forall O \in \mathcal{T})(x \in O \to y \in O)]$$

Proposition 1 (4.1.3.(1) in [17]). *A topological space $\langle X, \mathcal{T} \rangle$ is T_0 iff the specialization order is a partial order.*

From the above it follows that every T_0 topological space has associated a partial order in a natural way: the specialization order. Next, the other way round is shown: Given a partial order we can find a topological space based on it.

Definition 6. *Let $\langle X, R\rangle$ be a partial order. The Alexandrov topology based on this order is the set $\Upsilon(X, R) = \{S \subseteq X \mid S$ is a final segment of $\langle X, R\rangle\}$.*

Note that the Alexandrov topology is actually a topology and it is T_0.

Some basic examples of this topology can be easily obtained by taking $X = \mathbb{N}$, the set of natural numbers and R the natural order "$\leq$" or the partial order *"be divisible by"*.

In order to see that the Alexandrov topology $\Upsilon(X, \leq)$ is the topological translation of the partial order $\langle X, \leq\rangle$, we have to prove that the specialization order based on the topology $\Upsilon(X, \leq)$ is again[2] the order $\leq$.

Theorem 1. *Let $\Upsilon(X, \leq)$ be the Alexandrov topology based on the partial order $\langle X, \leq\rangle$ and let R be the specialization order based on that topology. Then $(\forall x, y \in X)(xRy \Leftrightarrow x \leq y)$.*

Proof. ($\Rightarrow$) Since $\uparrow x$ is a final segment and $x \in\uparrow x$, by hypothesis $y \in\uparrow x$. Then $x \leq y$. ($\Leftarrow$) Suppose $x \leq y$ and consider $G \in \Upsilon(X, \leq)$ such that $x \in G$. Then, by definition of final segment, $y \in G$.

The section is finished with a result which shows the narrow relation between morphisms and continuous mappings from a set on itself, viewed as partial order or Alexandrov topology.

Proposition 2 (4.2.4 in [17]). *Suppose that $f : \langle X, \mathcal{T}_X\rangle \to \langle Y, \mathcal{T}_Y\rangle$ is a continuous mapping between two topological spaces. Then f is a morphism with respect to the specialization orders of $\mathcal{T}_X$ and $\mathcal{T}_Y$.*

Theorem 2. *Let $f : \langle X, \mathcal{T}_X\rangle \to \langle Y, \mathcal{T}_Y\rangle$ be a mapping between two topological spaces. If f is a morphism with respect to the induced specialization orders R_X y R_Y, $\Upsilon(X, R_X) \subseteq \mathcal{T}_X$ and $\mathcal{T}_Y \subseteq \Upsilon(Y, R_Y)$, then f is continuous.*

Proof. Suppose $x \in X$ and M a neighborhood of $f(x)$. Then there exists $N \in \mathcal{T}_Y$ such that $f(x) \in N \subseteq M$. Since $\mathcal{T}_Y \subseteq \Upsilon(Y, R_Y)$, N is a final segment in $\langle Y, R_Y\rangle$ and $\uparrow f(x) \subseteq N$. But $\uparrow x \in \Upsilon(X, R_X) \subseteq \mathcal{T}_X$ and $x \in\uparrow x$, hence $\uparrow x$ is a neighborhood of x in X, and by Lemma 1 $f(\uparrow x) \subseteq\uparrow f(x)$. Consequently, f is continuous in x.

Corollary 1. *Let $\langle X, \leq\rangle$ be a partial order. Then $f : X \to X$ is continuous on the Alexandrov topology $\Upsilon(X, \leq)$ iff f is a morphism with respect to $\leq$.*

---

[2] Actually, the Alexandrov topology is not the unique topology with this property, but it is the finest one (see [6]).

4 The subsumption order

The application of General Topology to ILP shown in this paper is the continuity of the upward refinement operators on the Alexandrov topology seen as mappings

$$\varphi : \mathbb{C}_R/\!\sim\, \to \mathbb{C}/\!\sim$$

where $\mathbb{C}/\!\sim$ and $\mathbb{C}_R/\!\sim$ are the quotients obtained from the set of clauses $\mathbb{C}$ and the set of reduced clauses $\mathbb{C}_R$ via the equivalence relation based on subsumption.

In the generalization process, when a program P is too specific, we replace it by P' with the hope that P' covers the examples better than P. The step from P to P' is usually done applying a refinement operator to some clause C of P. These operators can be seen as mappings $\varphi : \mathbb{C} \to \mathbb{C}$ where $\mathbb{C}$ is the set of clauses of the language.

Our purpose is to consider such operators as mathematical mappings and study their continuity on the Alexandrov topology, which, as we have shown, can be considered the natural way of formalizing the relation of closeness on partial orders.

Since the subsumption relation between clauses is a quasi–order, but not a partial order, instead of dealing with $\mathbb{C}$, we formalize our operators as mappings between the partial orders[3] $\langle \mathbb{C}_R/\!\sim, \succeq \rangle$ and $\langle \mathbb{C}/\!\sim, \succeq \rangle$ endowed with their respective Alexandrov topologies in the natural way.

We begin our description with the basic relations. The only point to be noted here is that, because of technical reasons, θ has to be applicable to C in the definition of subsumption.

Definition 7. *Let C and D be clauses. C* subsumes *D, $C \succeq D$, iff there exists a substitution applicable to C such that $C\theta \subseteq D$. If $C \succeq D$ and $D \succeq C$ then C and D are equivalent and we will write it as $C \sim D$. A* reduced clause *is a clause C such that there is no proper subset D of C verifying $D \sim C$. We will denote by $\mathbb{C}_R$ the set of all reduced clauses.*

The next well–known result is due to Plotkin [11].

Proposition 3. *Let C and D be reduced clauses. If $C \sim D$ then C and D are variants.*

Since $\sim$ is an equivalence relation, we will denote by $\mathbb{C}/\!\sim$ and $\mathbb{C}_R/\!\sim$ the respective quotients. Moreover, if $C \in \mathbb{C}$, $[C] = \{D \in \mathbb{C} \,|\, C \sim D\}$ and if $C \in \mathbb{C}_R$, $[C]_R = \{D \in \mathbb{C}_R \,|\, C \sim D\}$. It is trivial to prove that the mapping $i : \mathbb{C}_R/\!\sim\, \to \mathbb{C}/\!\sim$ defined by $i([C]_R) = [C]$ is a bijection.

Definition 8. *The partial order $\succeq^*$ on $\mathbb{C}/\!\sim$ is defined as follows:*

$$(\forall [C], [D] \in \mathbb{C}/\!\sim) \, ([C] \succeq^* [D] \Leftrightarrow C \succeq D)$$

The order $\succeq^*$ is well–defined and it will cause no confusion if we use $\succeq$ instead of $\succeq^*$.

[3] The use of $\mathbb{C}_R$ to deal with refinement operators is usual (see [18]).

4.1 Upward refinement operators (I)

Downward refinement operators were introduced by Shapiro in [16], later Laird described in [7] a general framework for upward and downward refinement operators. In this paper, we follow Nienhuys-Cheng and de Wolf in [10].

The reader is expected to be familiar with these operators so we will leave out their well-known description. Instead of it, we split our formal definition in several steps.

Consider the one–literal clause $C_1 = \{L\}$ with $L = p(f(x), a, f(x))$. In order to generalize[4] it, we have to obtain a new clause C_2 such that there exists a substitution applicable to C_2 verifying $C_2\theta \subseteq C_1$. For that, (a) we choose a term t in L, say $t = f(x)$, (b) we choose several subsets of $Pos(L, t)$, e.g. $P_1 = \{1\}, P_2 = \emptyset, P_3 = \{1, 3\}$, (c) we choose a variable which does not occur in L, say z_1, and (d) we build the clause $C_2 = \{L[P_i \leftarrow z_1] \,|\, i = 1, 2, 3\} = \{p(z_1, a, f(x)), p(f(x), a, f(x)), p(z_1, a, z_1)\}$. Obviously $\theta = \{z_1/f(x)\}$ is applicable to C_2 and $C_2\theta \subseteq C_1$.

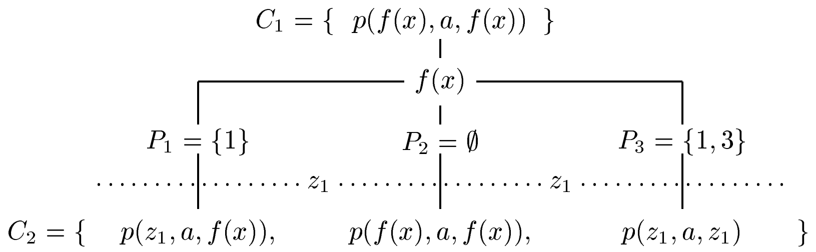

Fig. 1

If the clause has several literals, for example, $C_1 = \{L_1, L_2, L_3\}$, with $L_1 = p(f(x))$, $L_2 = q(b, f(x))$ and $L_3 = r(f(x), h(f(x)))$, the operation is done with all literals simultaneously. First, the same term is chosen in every literal of C_1, say $t = f(x)$. Then, for each literal $L_1 \in C_1$, some subsets of $Pos(L_i, t)$ are chosen, e.g.,

$$\begin{aligned} P_1^* &= \{\,\{1\}\,\} \subseteq \mathcal{P}Pos(L_1, t) \\ P_2^* &= \{\,\{2\}, \emptyset\,\} \subseteq \mathcal{P}Pos(L_2, t) \\ P_3^* &= \{\,\{1, 2{\cdot}1\}, \{1\}\,\} \subseteq \mathcal{P}Pos(L_3, t) \end{aligned}$$

After taking a variable which does not occur in C_1, say z_1, the following sets are built

$$\begin{aligned} L_1 &\xrightarrow{P_1^*} \{p(z_1)\} \\ L_2 &\xrightarrow{P_2^*} \{q(b, z_1), q(b, f(x))\} \\ L_3 &\xrightarrow{P_3^*} \{r(z_1, h(z_1)), r(z_1, h(f(x)))\} \end{aligned}$$

Finally, the clause C_2 is the union of these sets

$$C_2 = \{p(z_1), q(b, z_1), q(b, f(x)), r(z_1, h(z_1)), r(z_1, h(f(x)))\}$$

[4] The generalization based on deleting a literal from the clause will be considered in section 5.

In our general description, we will begin with a study of the relations between substitutions and grafts.

4.2 Substitution vs. graft

The next definitions show how a term can be settled in a clause by a set of positions. This construction will be useful to define the operator on $\mathbb{C}_R/\!\sim$. If $C_1, C_2 \in \mathbb{C}_R$ and they are variants, then we can think about both simultaneously by making abstraction of the terms and only handling sets of positions.

An equivalence class $[C] \in \mathbb{C}_R/\!\sim$ is the set of all variants of the reduced clause C. Thus, we cannot refer directly to a term in $[C]$ because each representative will have distinct variables, but we can do abstraction from the terms and settle them by their sets of positions.

The idea of abstracting term occurrences also appears in [9], but we study it in a different way, since we consider clauses as sets, not as sequences. This is slightly more general, since in sets the order of the elements is not important and in sequences it is.

Definition 9. *Let L be a literal. The set of positions P is called* compatible *with L if $P \subseteq Pos(L)$ and all terms rooted at positions of P are the same.*

For example, if $L = p(f(x), a, f(x))$, then $P_1 = \{2\}$, $P_2 = \emptyset$ and $P_3 = \{1, 3\}$ are compatible with L and $P_4 = \{1{\cdot}1, 2\}$ and $P_5 = \{1, 4{\cdot}3\}$ are not. Moreover, note that (a) if P is compatible with L, then P is independent, (b) if $P \subseteq Pos(L)$, then $|L/P| = 0 \Leftrightarrow P = \emptyset$ and (c) $P = \emptyset$ is compatible with any literal L.

It is not necessary that which term is occurring at positions of P be known to study its compatibility with L, since we can refer to it in an abstract way, by using L and P.

Definition 10. *Let $P \in \mathcal{P}\mathbb{N}^+$ be compatible with the literal L. If $u \in P$, then $Comp(L, P) = L/u$. If $P = \emptyset$, we define $Comp(L, P) = \emptyset$.*

Note that if $P \neq \emptyset$ is compatible with L, P *settles* a term in L.

The next definition extends the compatibility to sets of sets of positions.

Definition 11. *Let $P^* \in \mathcal{P}\mathcal{P}\mathbb{N}^+$ be finite and not empty. P^* is called* compatible *with the literal L if*

- *$(\forall P \in P^*)[P$ is compatible with $L]$ and*
- *$(\forall P, P' \in P^*)\,[u \in P \wedge v \in P' \Rightarrow L/u = L/v]$.*

When this holds, we define

$$Comp(L, P^*) = \begin{cases} \emptyset, & \text{if } P^* = \{\emptyset\} \\ Comp(L, P), & \text{if } P \in P^* \text{ and } P \neq \emptyset \end{cases}$$

For example, if $L = p(f(x), a, f(x))$, consider $P_1 = \{1, 3\}$, $P_2 = \{1\}$, $P_3 = \emptyset$ and $P_4 = \{2\}$. Suppose $P_1^* = \{P_1, P_2, P_3\}$ and $P_2^* = \{P_2, P_4\}$. Then P_1^* is compatible with L and P_2^* is not. Moreover, $Comp(L, P_1^*) = Comp(L, P_1) = f(x)$.

The next definition is basic in our formalization of the upward refinement operators. Given $L \in Lit$ and $P^* \in \mathcal{P}\mathcal{P}\mathbb{N}^+$ compatible with L, for each $P' \in P^*$ a literal L is generated by grafting a term t in every position of P.

Definition 12. *Consider a literal L, $t \in Term$ and $P^* \in \mathcal{PP}\mathbb{N}^+$ compatible with L. We will define $L[P^* \leftarrow t] = \{L[P \leftarrow t] \mid P \in P^*\}$.*

For example, if $L = p(f(x), a, f(x))$, $t \in Term$ and P_1^* is taken from the last example,

$$\begin{aligned} L[P_1^* \leftarrow t] &= \{L[P \leftarrow t] \mid P \in P_1^*\} \\ &= \{p(t, a, t), p(t, a, f(x)), p(f(x), a, f(x))\} \end{aligned}$$

Note that if $P^* = \{\emptyset\}$, then $L[P^* \leftarrow t] = \{L[\emptyset \leftarrow t]\} = \{L\}$.

Notice that if L is a literal, x a variable which occurs in L, θ a substitution such that $x\theta = t$, P be a set of positions compatible with L and $x = Comp(L, P)$, then P is compatible with $L\theta$ and $t = Comp(L\theta, P)$.

In the Appendix we prove several technical lemmas dealing with grafts and substitutions which will be used in the proof of the main results.

4.3 Some previous mappings

Above we have seen how a term can be chosen from a literal with the help of a compatible set of positions P. In this way, if a literal L_1 is replaced by a variant L_2, the analogous term is chosen, because this choice only depends on P.

Therefore, a refinement operator can be defined on $[C] \in \mathbb{C}_R/\sim$, since the set of positions P will be which settles the term to be replaced in each variant of C.

In next steps, we will define two mappings $\Delta : Lit \to \mathcal{PP}\mathbb{N}^+$ and $\Delta^* : \mathbb{C} \to \mathcal{PPP}\mathbb{N}^+$ and some constraints will be imposed to make them *compatible*. The idea behind this formalization is to guarantee that the *same* term is chosen in every literal of the clause.

Definition 13. *A mapping $\Delta : Lit \to \mathcal{PP}\mathbb{N}^+$ will be called an* assignment of positions for literals *(L–assignment, for short). An L–assignment is called*

- compatible with the literal L *iff $\Delta(L)$ is compatible with L.*
- compatible in Lit *iff $(\forall L \in Lit)\,[\Delta$ is compatible with $L]$.*
- *a* compatible morphism *iff Δ is compatible in Lit and*

$$\Delta : \langle Lit, \geq \rangle \to \langle \mathcal{PP}\mathbb{N}^+, \subseteq \rangle$$

  is a morphism.

For example, if a is a constant, then $\Delta : Lit \to \mathcal{PP}\mathbb{N}^+$ such that $\Delta(L) = \mathcal{P}Pos(L, a)$ is a compatible morphism. The next Lemma is immediate.

Lemma 2. *Let $\Delta : \langle Lit, \geq \rangle \to \langle \mathcal{PP}\mathbb{N}^+, \subseteq \rangle$ be a morphism and $L_1, L_2 \in Lit$. If L_1 and L_2 are variants, then $\Delta(L_1) = \Delta(L_2)$.*

The next function Δ^* can be seen as a preliminary version of the operator which will be defined below. Given a clause C, $\Delta^*(C)$ can be seen as a scheme of how the generalization of C will be. Notice that Δ^* is independent not only of the term replaced in C, but also of the term which will be grafted in the generalization.

Definition 14. *Let $\Delta : Lit \to \mathcal{P}\mathcal{P}\mathbb{N}^+$ be an L–assignment. The assignment of positions for clauses (C-assignment, for short) based on Δ, which will be denoted by Δ^*, is the mapping*

$$\begin{array}{rl} \Delta^* : \mathbb{C} & \longrightarrow \mathcal{P}\mathcal{P}\mathcal{P}\mathbb{N}^+ \\ C & \mapsto \ \Delta^*(C) = \{\Delta(L) \mid L \in C\} \end{array}$$

Informally, Δ^* maps a clause $\{L_1, \ldots, L_n\}$ into the set $\{\Delta(L_1), \ldots, \Delta(L_n)\}$, where $\Delta(L_i)$ settles the positions of L_i at which a term will be grafted.

Definition 15. *Let Δ be a compatible morphism and C a clause. The C–assignment Δ^* is* compatible *with C if for all $L, L' \in C$ such that $Comp(L, \Delta(L)) \neq \emptyset$ and $Comp(L', \Delta(L')) \neq \emptyset$ we have $Comp(L, \Delta(L)) = Comp(L', \Delta(L'))$. When this holds, we will define*

$$Comp(C, \Delta) = \begin{cases} \emptyset, & \text{if } C = \emptyset \\ Comp(L, \Delta(L)), & \text{if } L \in C \end{cases}$$

Formally, $Comp(C, \Delta) = \bigcup \{Comp(L, \Delta(L)) \mid L \in C\}$. The C–assignment Δ^* is called compatible with $\mathbb{C}$ *if for all $C \in \mathbb{C}$, Δ^* is compatible with C.*

The following one is a key transformation. In Definition 12 we have seen how to get a set of literals from one of them by grafting a term. If the compatibility of Δ^* is linked to the former, a new clause can be built by putting together every set of literals obtained from their individual transformations.

Definition 16. *Let Δ^* be a C–assignment compatible with $\mathbb{C}$ and $t \in Term$. We define*

$$C[\Delta \leftarrow t] = \bigcup \{L[\Delta(L) \leftarrow t] \mid L \in C\}$$

Note that if C is the empty clause then $C[\Delta \leftarrow t] = \emptyset$.

In this step $C[\Delta \leftarrow t]$ is the clause obtained by grafting the term t in each literal L of C in accordance with $\Delta(L)$. Turning back to a previous example, if $C_1 = \{L_1, L_2, L_3\}$, with $L_1 = p(f(x))$, $L_2 = q(b, f(x))$, $L_3 = r(f(x), h(f(x)))$, and

$$\begin{array}{l} \Delta(L_1) = P_1^* = \{\,\{1\}\,\} \\ \Delta(L_2) = P_2^* = \{\,\{2\}, \emptyset\,\} \\ \Delta(L_3) = P_3^* = \{\,\{1, 2{\cdot}1\}, \{1\}\,\} \end{array}$$

and considering the constant a as the term to be grafted, then

$$C[\Delta \leftarrow a] = \{p(a), q(b, a), q(b, f(x)), r(a, h(a)), r(a, h(f(x)))\}$$

Notice that if Δ^* is a C–assignment compatible with $\mathbb{C}$, C_1 and C_2 are two equivalent reduced clauses and a is a constant, then the clauses $C_1[\Delta \leftarrow a]$ and $C_2[\Delta \leftarrow a]$ are equivalent as well. This is a key point in our definition of the operator. Next we see the details.

4.4 The operators

The last definition allows us to get C_2 from C_1 grafting a term t at a set of positions of some literals. The graft only depends on the set of positions fixed by Δ and its compatibility guarantees that the same term is replaced in each literal.

Our aim is defining the upward refinement operators as mappings

$$\varphi : \mathbb{C}_R/\!\sim\, \to \mathbb{C}/\!\sim$$

In the last subsection we have shown how the clause $C[\Delta \leftarrow t]$ can be obtained by grafting the term t at the positions settled by Δ. The step to the quotient needs more attention. On the one hand, in the domain $\mathbb{C}_R$, there is no problem since two clauses are equivalent if and only if they are variants, and by Lemma 2, if two literals are variant, Δ chooses the same $P^* \in \mathcal{PPN}^+$ for both. On the other hand, the problem arises if the term grafted contains any variable z, since in every $[C] = \{C' \in \mathbb{C} \,|\, C' \sim C\}$ there exist clauses containing z.

We split the solution into several steps. Firstly, we study the grafts of constants. The next Proposition shows that grafts of constants and renamings are commutative for C–assignments compatible with $\mathbb{C}$.

Proposition 4. *Let a be a constant, θ a renaming and Δ^* a C–assignment compatible with $\mathbb{C}$. Then $(C[\Delta \leftarrow a])\theta = C\theta[\Delta \leftarrow a]$*

Proof. If $C = \emptyset$ the result holds trivially. Consider $C_1 = \{L_1^1, \ldots, L_n^1\}$ and $C_2 = \{L_1^2, \ldots, L_n^2\}$ such that there exists a renaming θ with $L_i^2 = L_i^1\theta$ for all $i \in \{1, \ldots, n\}$. We have the following

$$\begin{array}{rll} C_2[\Delta \leftarrow a] & = \cup\{L_i^2[\Delta(L_i^2) \leftarrow a] \,|\, i \in \{1, \ldots, n\}\} & \\ & = \cup\{L_i^1\theta[\Delta(L_i^1) \leftarrow a] \,|\, i \in \{1, \ldots, n\}\} & \text{by Lemma 2} \\ & = \cup\{(L_i^1[\Delta(L_i^1) \leftarrow a])\theta \,|\, i \in \{1, \ldots, n\}\} & \text{by Corollary 3} \\ & = (\cup\{L_i^1[\Delta(L_i^1) \leftarrow a] \,|\, i \in \{1, \ldots, n\}\})\theta & \\ & = (C_1[\Delta \leftarrow a])\theta & \end{array}$$

From last Proposition we have that if a constant is grafted in two equivalent reduced clauses, two variants are obtained. So, we can use this result to graft in C a variable which does not occur in C with help of a new constant and a *choice function*.

The next steps are: (a) the selected term in C is replaced by a new constant at positions fixed by Δ^* and (b) all occurrences of that constant are replaced by a variable which does not occur in C. That variable is determined by a choice function.

Definition 17. *A mapping $f : \mathbb{C} \to Var$ is called a* choice function *if $f(C)$ is a variable which does not occur in C. Let $f : \mathbb{C} \to Var$ be a choice function and a a constant. We will define:*

$$\begin{array}{lcl} \phi_f^a : \mathbb{C}_R & \longrightarrow & \mathbb{C} \\ \quad C & \mapsto & \phi_f^a(C) = \{L[Pos(L, a) \leftarrow f(C)] \,:\, L \in C\} \end{array}$$

Informally, if $f(C) = z \in Var$, $\phi_f^a(C)$ is the clause obtained by replacing in C every occurrence of the constant a by the variable z.

Lemma 3. *Let $f : \mathbb{C} \to Var$ be a choice function and a a constant. If $C_1, C_2 \in \mathbb{C}_R$ are variants, then $\phi_f^a(C_1)$ and $\phi_f^a(C_2)$ are also variants.*

Proof. The result holds trivially for the empty clause. If $C_1 = \{L_1^1, \ldots, L_n^1\}$ and $C_2 = \{L_1^2, \ldots, L_n^2\}$ are variants, by Corollary 2, $L_i^1[Pos(L_i^1, a) \leftarrow f(C_1)]$ and $L_i^2[Pos(L_i^2, a) \leftarrow f(C_2)]$ are instances of each other.

Next, we give our formal definition of upward refinement operator.

Definition 18. *Let Δ^* be a C–assignment compatible with $\mathbb{C}$. Let a be a constant which does not belong to the language $\mathcal{L}$ and let $\mathbb{C}_a$ be the set of all clauses of $\mathcal{L} \cup \{a\}$. Let $f : \mathbb{C}_a \to Var$ be a choice function. We will define the mapping*

$$\begin{array}{rcl} \widetilde{\Gamma}_{\Delta,f} : \mathbb{C}_R & \longrightarrow & \mathbb{C} \\ C & \mapsto & \widetilde{\Gamma}_{\Delta,f}(C) = \phi_f^a(C[\Delta \leftarrow a]) \end{array}$$

The upward refinement operator is defined as

$$\begin{array}{rcl} \Gamma_{\Delta,f} : \mathbb{C}_R/\!\sim & \longrightarrow & \mathbb{C}/\!\sim \\ [C] & \mapsto & [\widetilde{\Gamma}_{\Delta,f}(C)] \end{array}$$

From Proposition 4 and Lemma 3, we have that $\Gamma_{\Delta,f}$ does not depend on the representative. Furthermore, since $\Gamma_{\Delta,f}$ does not depend on f or a, (only on Δ), we will write Γ_Δ instead of $\Gamma_{\Delta,f}$. Next we see an example.

Consider $C_1 = \{p(x, f(b)), q(x)\}$ and $C_2 = \{p(y, f(b)), q(y)\}$. They are two reduced clauses and $[C_1] \sim [C_2]$. Let Δ^* be a C–assignment compatible with $\mathbb{C}$ such that

$$\begin{array}{lll} \Delta(p(x, f(b))) & = \Delta(p(y, f(b))) & = \{\ \{2\}\ \} \\ \Delta(q(x)) & = \Delta(q(y)) & = \{\emptyset\} \end{array}$$

Let a be a constant which does not belong to the language. Then

$$\begin{array}{l} C_1[\Delta \leftarrow a] = \{p(x, a), q(x)\} \\ C_2[\Delta \leftarrow a] = \{p(y, a), q(y)\} \end{array}$$

given the choice function f, $f(C_1[\Delta \leftarrow a])$ is a variable which does not occur in $C_1[\Delta \leftarrow a]$, say $f(C_1[\Delta \leftarrow a]) = y$. Analogously, say $f(C_2[\Delta \leftarrow a]) = z$. Then we have

$$\begin{array}{l} \widetilde{\Gamma}_{\Delta,f}(C_1) = \phi_f^a(C_1[\Delta \leftarrow a]) = \{p(x, y), q(x)\} \\ \widetilde{\Gamma}_{\Delta,f}(C_2) = \phi_f^a(C_2[\Delta \leftarrow a]) = \{p(y, z), q(y)\} \end{array}$$

hence, $\Gamma_\Delta([C_1]) = \Gamma_\Delta([C_2]) = [\{p(x, y), q(x)\}] = [\{p(y, z), q(y)\}]$.

As the example shows, these operators are not elementary (in the sense of [10]), since there are no constraints on the nature of the terms settled by Δ.

4.5 Continuity

Next, the application of General Topology to ILP we present in this paper is proved. Till now, we have the sets $\mathbb{C}_R/\sim$ and $\mathbb{C}/\sim$ and, for each C-assignment compatible with $\mathbb{C}$, we have a mapping Γ_Δ between them.

The natural structure of $\mathbb{C}_R/\sim$ and $\mathbb{C}/\sim$ is the partial order of subsumption but, as we have seen in section 3, these sets can be seen as topological spaces with help of the Alexandrov topology. We prove now that these operators are continuous on that topology.

Theorem 3. *The mapping Γ_Δ is continuous on the Alexandrov topology.*

Proof. Let f be a function choice and a a constant which does not belong to $\mathcal{L}$. By Corollary 1, it is sufficient to prove that if $C_1, C_2 \in \mathbb{C}_R$ and θ is a substitution applicable to C_1 such that $C_1\theta \subseteq C_2$ then there exists a substitution θ_0 applicable to $\widetilde{\Gamma}_{\Delta,f}(C_1)$ verifying $\widetilde{\Gamma}_{\Delta,f}(C_1)\theta_0 \subseteq \widetilde{\Gamma}_{\Delta,f}(C_2)$.
Let C_1, C_2 and θ such clauses and substitution and consider $L \in \widetilde{\Gamma}_{\Delta,f}(C_1) = \phi^a_f(C_1[\Delta \leftarrow a])$. Then there exists $L_0 \in C_1$ and $P_0 \in \Delta(L_0)$ such that $L_0[P_0 \leftarrow a] \in C_1[\Delta \leftarrow a]$ and $L = L_0[P_0 \leftarrow f(C_1)]$. On the other hand, $L_0\theta \in C_2$ and since $\Delta : \langle Lit, \geq\rangle \rightarrow \langle \mathcal{PP}\mathbb{N}^+, \subseteq\rangle$ is a morphism, $P_0 \in \Delta(L_0\theta)$, hence $L_0\theta[P_0 \leftarrow a] \in C_2[\Delta \leftarrow a]$ and $L_0\theta[P_0 \leftarrow f(C_2)] \in \widetilde{\Gamma}_{\Delta,f}(C_2)$.
Let $\theta_0 = \theta \restriction_{Var(C_1[\Delta \leftarrow a])} \cup \{f(C_1)/f(C_2)\}$. Since θ_0 is applicable to $\widetilde{\Gamma}_{\Delta,f}(C_1)$, it only remains to prove that $\widetilde{\Gamma}_{\Delta,f}(C_1)\theta_0 \subseteq \widetilde{\Gamma}_{\Delta,f}(C_2)$, but $(L_0[P_0 \leftarrow f(C_1)])\theta_0 = (L_0[P_0 \leftarrow f(C_1)])(\theta \restriction_{Var(L_0[P_0 \leftarrow a])} \cup \{f(C_1)/f(C_2)\})$, therefore, by Lemma 7, we have $(L_0[P_0 \leftarrow f(C_1)])\theta_0 = L_0\theta[P_0 \leftarrow f(C_2)]$ and $L\theta_0 \in \widetilde{\Gamma}_{\Delta,f}(C_2)$.

5 Deleting a literal

All classic refinement operators can be seen as particular cases of the mapping $\Gamma_\Delta : \mathbb{C}_R/\sim \rightarrow \mathbb{C}/\sim$, except one. The fourth classic refinement operator allows us to generalize a clause C deleting a most general literal (Definition 19). In this section, this operator is formalized and its continuity is proved.

Definition 19 (17.12 in [10]). *A literal $p(x_1, \ldots, x_n)$ or $\neg p(x_1, \ldots, x_n)$ is* most general *with respect to a clause C if $x_1, \ldots, x_n$ are distinct variables not appearing in C.*

Definition 20. *Given $C \in \mathbb{C}$, $max(C) \subseteq Lit$ is the set of literals L such that L is a most general literal with respect to $C - \{L\}$.*

Note that a most general literal with respect to a clause is settled by its predicate symbol and its sign: "+" for the positive ones and "−"for the negative ones.

Definition 21. *Given $L \in Lit$ we will define $pair(L) \in Pred \times \{+, -\}$ by the pair $\langle p, s\rangle$ where p is the predicate symbol of L and s is its sign. For example, $pair(\neg p(x, f(a))) = \langle p/2, -\rangle$.*

Note that for any substitution θ, $pair(L) = pair(L\theta)$. Next lemmas are straightforward from definitions.

Lemma 4. *Let $C \cup \{L\}$ be a reduced clause where L is a most general literal with respect to C and let θ be a renaming . Then $C\theta \cup \{L\theta\}$ is a reduced clause and $L\theta$ is most general with respect to $C\theta$.*

Lemma 5. *Given $C \in \mathbb{C}_R$ and $L \in max(C)$, if $L' \in C$ and $pair(L') = pair(L)$, then L' and L are variants.*

Definition 22. *Consider $\langle p, s\rangle \in Pred \times \{+, -\}$. We define*

$$\widetilde{\beta}^{\langle p,s\rangle}(C) = \begin{cases} C - \{L\}, & \text{if } pair(L) = \langle p, s\rangle \wedge L \in max(C) \\ C & \text{otherwise} \end{cases}$$

Our formal definition of the fourth upward refinement operator is

$$\begin{array}{rcl} \beta^{\langle p,s\rangle} : \mathbb{C}_R/\!\sim & \longrightarrow & \mathbb{C}/\!\sim \\ {[C]} & \mapsto & [\widetilde{\beta}^{\langle p,s\rangle}(C)] \end{array}$$

From Lemmas 4 and 5 we have that $\beta^{\langle p,s\rangle}$ does not depend on the representative.

Finally, we prove the continuity of the operators on the Alexandrov topology.

Theorem 4. *The mapping $\beta^{\langle p,s\rangle}$ is continuous on the Alexandrov topology.*

Proof. Consider $\langle p, s\rangle \in Pred \times \{+, -\}$. By Corollary 1, it is sufficient to prove that if $C, D \in \mathbb{C}_R$ and θ is a substitution applicable to C such that $C\theta \subseteq D$ then there exists a substitution θ_0 applicable to $\widetilde{\beta}^{\langle p,s\rangle}(C)$ such that $\widetilde{\beta}^{\langle p,s\rangle}(C)\theta_0 \subseteq \widetilde{\beta}^{\langle p,s\rangle}(D)$.
Let $C, D \in \mathbb{C}_R$ and θ be such clauses and substitution and consider $\theta_0 = \theta \upharpoonright_{Var(\widetilde{\beta}^{\langle p,s\rangle}(C))}$. Obviously θ_0 is applicable to $\widetilde{\beta}^{\langle p,s\rangle}(C)$. Since $C\theta \subseteq D$, in order to prove $\widetilde{\beta}^{\langle p,s\rangle}(C)\theta_0 \subseteq \widetilde{\beta}^{\langle p,s\rangle}(D)$ we need only to consider the case when there exists $L \in C$ such that $[L \in max(C) \wedge pair(L) = \langle p, s\rangle]$.

- if $L\theta \notin max(D)$, then $\widetilde{\beta}^{\langle p,s\rangle}(C)\theta_0 \subseteq C\theta \subseteq D = \widetilde{\beta}^{\langle p,s\rangle}(D)$
- if $L\theta \in max(D)$, then $\widetilde{\beta}^{\langle p,s\rangle}(C)\theta_0 \subseteq C\theta - \{L\theta\} \subseteq D - \{L\theta\} = \widetilde{\beta}^{\langle p,s\rangle}(D)$.

6 Conclusions and future work

In this paper General Topology and ILP, separated up till now, are brought together. In the first part of the paper, we have shown that the Alexandrov topology is the natural translation of partial orders into General Topology. With this topology, we lose metric properties but it allows us to handle the subsumption order in a natural way and, in certain sense, it settles down the notion of *closeness* based on subsumption.

The application that we present of General Topology to ILP is the continuity of the upward refinement operators on the Alexandrov topology. For that we have developed a study about the relation between substitution and grafts, interesting by itself, and given a new formalization of the operators.

In this formalization we apply the operator to a set of equivalent clauses instead of a unique clause. This serves our purpose, as the domain of the operators is a partial order, but it is also interesting by itself since it avoids the problem of taking a representative. In many papers, renamings are scorned and variant clauses are considered identical. After that, a representative is taken from the set of equivalent under the assumption of that any representative would play the same role when the operators were applied.

This point is critical by dealing with upward refinement operators, since the grafted variable has to be new for the chosen clause and taking a new variable for all clauses (i.e. a variable out of the language $\mathcal{L}$) has no logical sense. By defining the operator on $\mathbb{C}_R/\!\sim$ instead of $\mathbb{C}_R$, the problem is avoided.

As main result, the continuity of these operators on the Alexandrov topology has been proved.

This paper is a first approximation between ILP and General Topology. We think that the formal foundation of basic properties of the operators used in ILP, as continuity or convergence, needs to clarify and formalize the idea of closeness among clauses. Our aim is going on with a deeper study of topological properties of ILP techniques.

References

1. K.R. Apt: From Logic Programming to Prolog. Prentice Hall, 1997
2. A. Batarekh and V.S. Subrahmanian: Topological Model Set Deformations in Logic Programming. Fundamenta Informaticae XII, 3, 357–400, 1989
3. M. Fréchet: Sur quelques points du calcul fonctionnel. Reudicont del Circulo Matematico di Palermo, vol 22, 1906.
4. J.H. Gallier: Logic for Computer Science Foundations of Automatic Theorem Proving. Harper & Row, Publishers, New York, 1986
5. G. Huet: Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems. Journal of the Association for Computing Machinery, Vol 27, No 4, pp 797–821, October 1980
6. P.T. Johnstone: Stone spaces. Cambridge University Press, 1982
7. P.D. Laird: Learning from Good and Bad Data. Kluwer Academic Publishers, 1988
8. L. Nachbin: Topology and order. D. van Nostrand Company, Inc., 1965.
9. S-H. Nienhuys-Cheng: Term partitions and minimal generalizations of clauses. Technical Report EUR–FEW–CS–91–01. Department of Computer Science, Erasmus University, the Netherlands, 1991. `http://www.few.eur.nl/few/research/pubs/cs/1991/eur-few-cs-91-01.pdf`
10. S-H. Nienhuys-Cheng and R. de Wolf: Foundations of Inductive Logic Programming. LNCS 1228. Springer, 1997
11. G.D. Plotkin: A Note on Inductive Generalization. In Machine Intelligence 5, pp.: 153–163. Edinburgh University Press, Edinburgh, 1970.
12. A.K. Seda: Some Applications of General Topology to the Semantics of Logic Programs. Bull. European Association for Theoretical Computer Science 52 279-292 1994.

13. A.K.Seda: A Topological View of the Kowalski-van Emden Theorem. Bull. European Association for Theoretical Computer Science 53, 256-263, 1994
14. A.K. Seda: Topology and the Semantics of Logic Programs. Fundamenta Informaticae 24 (4) 359-386 1995
15. S. Vickers: Topology via Logic. Cambridge University Press, 1989
16. E.Y. Shapiro: Inductive Inference of Theories from Facts. Technical Report 624, Department of Computer Science, Yale University, New Haven, CT, 1981
17. M.B. Smyth: Topology. In Handbook of Logic in Computer Science Vol 1 Background: Mathematical Structures. Edited by: S. Abramsky, Dov M. Gabbay and T.S.E. Maibaum. Oxford Science Publications, 1992. pp.: 641–761.
18. P.R.J. van der Laag, S.-H. Nienhuys-Cheng: Subsumption and refinement in model inference. Technical Report EUR–FEW–CS–92–07. Department of Computer Science, Erasmus University, the Netherlands, 1992. `http://www.few.eur.nl/few/research/pubs/cs/1997/eur-few-cs-92-07.pdf`
19. S. Willard: General Topology. Addison Wesley Publishing Company, 1970

7 Appendix

The following ones are several technical lemmas dealing with grafts and substitutions. The Lemma 7 will be used in the Theorem 3.

Lemma 6. *Let L be a literal, x a variable which does not occur in L, θ a substitution, $u \in Pos(L)$, c a constant, t a term and $Var^* = Var(L[u \leftarrow c])$. Then*

$$L[u \leftarrow x]((\theta\lceil_{Var^*}) \cup \{x/t\}) = L\theta[u \leftarrow t]$$

Proof. The proof is by induction on the length of u, but notice before that $Var(L[u \leftarrow x]) = Var^* \cup \{x\}$ and $x \notin Var^*$. Consider $L = p(t_1, \ldots, t_n)$.
Case 1: Suppose $long(u) = 1$.

$$\begin{aligned} & p(t_1, \ldots, t_n)[u \leftarrow x]((\theta\lceil_{Var^*}) \cup \{x/t\}) \\ &= p(t_1, \ldots, t_{u-1}, x, t_{u+1}, \ldots, t_n)((\theta\lceil_{Var^*}) \cup \{x/t\}) \\ &= p(t_1\theta, \ldots, t_{u-1}\theta, t, t_{u+1}\theta, \ldots, t_n\theta) \\ &= p(t_1\theta, \ldots, t_{u-1}\theta, t_u\theta, t_{u+1}\theta, \ldots, t_n\theta)[u \leftarrow t] \\ &= p(t_1, \ldots, t_n)\theta[u \leftarrow x] \end{aligned}$$

Case 2: Suppose the result holds for all positions of length m and let u be a position of length $m+1$, $u = i\hat{\ }v$ with $i \geq 1$ and $v \in \mathbb{N}^+$ with $long(v) = m$.

$$\begin{aligned} & p(t_1, \ldots, t_n)[u \leftarrow x]((\theta\lceil_{Var^*}) \cup \{x/t\}) \\ = \;& p(t_1, \ldots, t_n)[i\hat{\ }v \leftarrow x]((\theta\lceil_{Var^*}) \cup \{x/t\}) \\ = \;& p(t_1, \ldots, t_{i_1}, t_i[v \leftarrow x], t_{i+1}, \ldots, t_n)((\theta\lceil_{Var^*}) \cup \{x/t\}) \\ = \;& p(t_1\theta, \ldots, t_{i_1}\theta, t_i[v \leftarrow x]((\theta\lceil_{Var^*}) \cup \{x/t\}), t_{i+1}\theta, \ldots, t_n\theta) \\ \stackrel{i.h.}{=} \;& p(t_1\theta, \ldots, t_{i_1}\theta, t_i\theta[v \leftarrow t]\theta, t_{i+1}\theta, \ldots, t_n\theta) \\ = \;& p(t_1\theta, \ldots, t_{i_1}\theta, t_i\theta, t_{i+1}\theta, \ldots, t_n\theta)[i\hat{\ }v \leftarrow t] \\ = \;& p(t_1, \ldots, t_n)\theta[u \leftarrow t] \end{aligned}$$

Lemma 7. *Let L be a literal, x a variable which does not occur in L, θ a substitution, $P \in \mathcal{P}\mathbb{N}^+$ compatible with L, c a constant, $t \in Term$ and $Var^* = Var(L[P \leftarrow c])$. Then*

$$L[P \leftarrow x]((\theta \restriction_{Var^*}) \cup \{x/t\}) = L\theta[P \leftarrow t]$$

Proof. The result holds from Lemma 6 and the fact of being P an independent set of positions.

Lemma 8. *Let L be a literal, $u \in Pos(L)$, $s_1, s_2 \in Term$ and θ a substitution such that $s_1\theta = s_2$. Then*

$$L[u \leftarrow s_1]\theta = L\theta[u \leftarrow s_2]$$

Proof. By induction on the length of u.

- Suppose $long(u) = 1$. Then

$$\begin{aligned} & p(t_1, \dots, t_n)[u \leftarrow s_1]\theta \\ &= p(t_1, \dots, t_{u-1}, s_1, t_{u+1}, \dots, t_n)\theta \\ &= p(t_1\theta, \dots, t_{u-1}\theta, s_1\theta, t_{u+1}\theta, \dots, t_n\theta) \\ &= p(t_1\theta, \dots, t_{u-1}\theta, s_2, t_{u+1}\theta, \dots, t_n\theta) \\ &= p(t_1\theta, \dots, t_{u-1}\theta, t_u\theta, t_{u+1}\theta, \dots, t_n\theta)[u \leftarrow s_2] \\ &= p(t_1, \dots, t_n)\theta[u \leftarrow s_2]\theta \end{aligned}$$

- Suppose the result holds for all positions of length m and let u be a position of length $m+1$, $u = i\hat{\ }v$ with $i \geq 1$ and $v \in \mathbb{N}^+$ with $long(v) = m$.

$$\begin{aligned} & p(t_1, \dots, t_n)[u \leftarrow s_1]\theta \\ =\ & p(t_1, \dots, t_n)[i\hat{\ }v \leftarrow s_1]\theta \\ =\ & p(t_1, \dots, t_{i_1}, t_i[v \leftarrow s_1], t_{i+1}, \dots, t_n)\theta \\ =\ & p(t_1\theta, \dots, t_{i_1}\theta, t_i[v \leftarrow s_1]\theta, t_{i+1}\theta, \dots, t_n\theta) \\ \stackrel{i.h.}{=}\ & p(t_1\theta, \dots, t_{i_1}\theta, t_i\theta[v \leftarrow s_2]\theta, t_{i+1}\theta, \dots, t_n\theta) \\ =\ & p(t_1\theta, \dots, t_{i_1}\theta, t_i\theta, t_{i+1}\theta, \dots, t_n\theta)[i\hat{\ }v \leftarrow s_2] \\ =\ & p(t_1, \dots, t_n)\theta[u \leftarrow s_2] \end{aligned}$$

Lemma 9. *Let L be a literal, $P \in \mathcal{P}\mathbb{N}^+$ compatible with L, θ a substitution and let t_1 and t_2 two terms such that $t_1\theta = t_2$. Then*

$$(L[P \leftarrow t_1])\theta = L\theta[P \leftarrow t_2]$$

Proof. By induction on the cardinal of P.

- If $P = \emptyset$, then

$$(L[P \leftarrow t_1])\theta = L\theta = L\theta[P \leftarrow t_2]$$

- If $P = \{u\} \cup P'$ and $u \notin P'$, then

$$\begin{aligned} & (L[P \leftarrow t_1])\theta \\ =\ & (L[\{u\} \cup P' \leftarrow t_1])\theta \\ =\ & (L[P' \leftarrow t_1])[u \leftarrow t_1]\theta \\ =\ & (L[P' \leftarrow t_1])\theta[u \leftarrow t_2] \qquad \text{by Lemma 8} \\ \stackrel{i.h.}{=}\ & (L\theta[P' \leftarrow t_2])[u \leftarrow t_2] \\ =\ & L\theta[P \leftarrow t_2] \end{aligned}$$

In particular, the last Lemma holds if t_1 is a constant. The next results are two straightforward corollaries from Lemma 9.

Corollary 2. *If L is a literal, x a variable which does not occur in L, θ a substitution such that $x \notin Dom(\theta)$ and $t \in Term$, then $(L[P \leftarrow x])(\theta \cup \{x/t\}) = L\theta[P \leftarrow t]$.*

Proof. Immediate from Lemma 9 since $L\theta = L(\theta \cup \{x/t\})$ and $x(\theta \cup \{x/t\}) = t$.

Corollary 3. *Let L be a literal, $P^* \in \mathcal{PP}\mathbb{N}^+$ compatible with L, θ a substitution and $t_1, t_2 \in Term$ such that $t_1\theta = t_2$. Then $(L[P^* \leftarrow t_1])\theta = L\theta[P^* \leftarrow t_2]$.*

Proof. It holds from 9 and the definition of compatibility of P^* (Definition 11).