RC 22622 (W0201-079) January 15, 2002
Computer Science

IBM Research Report

Categories of Patterns Relevant to Relational Learning

Frank J. Oles
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Categories of Patterns Relevant to Relational Learning (Extended Abstract)

Frank J. Oles

P.O. Box 218
IBM T.J. Watson Research Center
Yorktown Heights, New York 10598
Phone: 914-945-2012
Fax: 914-945-3434
Email: oles@us.ibm.com

January 14, 2002

Abstract

Precedence-inclusion patterns generalize constituent structure trees familiar to computational linguists. By small changes in the axioms, we obtain (1) structures supporting a significant theory of pattern generalization that directly speaks to the problem of relational learning in many settings, (2) when specialized to posets, a new theory of poset generalization that may support generalization of ontologies and hierarchial classification systems, and (3) a rich extension of the category of sets.

1 Introduction

We often encounter structured sets of entities, where, after some analysis, each entity may be labeled to reflect the set of properties it inherently possesses or the properties it possesses by virtue of its presence in, or perhaps by virtue of its position in, the ensemble under consideration. Moreover, in a such a structured set of entities, the structure generally serves to relate the individual entities to one another in familiar ways. Typically one entity may precede another one or one entity may include (or contain) another one. These observations are the starting point for a mathematical theory of *precedence-inclusion patterns.* This theory delivers a computationally tractable and mathematically well-founded approach to pattern generalization that is directly relevant to a fundamental problem in artificial intelligence: *relational learning,* by which we mean the induction from examples of some number of assertions that certain elements $x_1, x_2, \cdots$ of a structure S are in some particular relation $R(x_1, x_2, \cdots)$ to one another when the structure S is a specific instance of a more general pattern.

Our practical motivation is to find new approaches to relational learning applicable to text. We also hope to apply this work in the future to the generalization of ontologies, particularly hierarchies for the classification of documents or webpages.

We set out to discover a widely applicable theory of patterns, but one applicable to text in particular, where patterns are first-class mathematical objects. And, we wanted a theory of patterns in which we could go beyond viewing generalization as simply obtaining a logical summation of what is common to a collection of examples. Superficially at least, it appeared that such a logical summation was all that an inductive logic programming approach could give us. (For that approach, see [2].) Eventually, we did find the theory of patterns we wanted, and, in it, the minimal most specific generalization of a set of patterns is another pattern, not just a logical formula. For instance, it would be nice if the generalization of two hierarchies for automatically categorizing incoming email were transparently another hierarchy for automatically categorizing incoming email.

Technically, what we will present is a new mathematical category whose objects are precedence-inclusion patterns, in which morphisms between patterns play a crucial role. Precedence-inclusion patterns generalize constituent structure trees familiar to computational linguists. By small changes in the axioms, we obtain structures supporting a significant theory of pattern generalization, and a new Cartesian closed category extending the category of sets. Beyond the introduction of the basic concepts, the aims of this paper include demonstrating that each finite set of finite patterns has a minimal most specific generalization that is unique up to isomorphism, arguing that this best possible generalization can be readily computed, and making clear how this construction directly contributes to relational learning.

Moreover, any poset gives rise to a set with a strict partial order (just subtract the identity relation), which gives rise to a precedence-inclusion pattern, in which one or the other of the relations of precedence or inclusion is trivial. In this way, a new theory of poset generalization comes out of our theory of pattern generalization. Although the point just made is important, we don't specifially study generalizations of posets in this paper.

In this paper we do not deal with pattern construction based on excluding negative examples as well as generalizing from positive examples. This may be a subject for future research.

We conjecture that that some of the results might be obtained directly from Plotkin's seminal work [5] on generalization of sets of clauses. That being said, the present work stands as a coherent whole and deals with many issues that have no counterpart in [5], such as

1. the properties of categories of patterns,

2. the various properties of pattern-preserving maps, particularly retractions,

and

3. the relevance of stripped vs. unstripped patterns for relational learning.

No proofs are given in this extended abstract due to space limitations.

We wish to thank Michael W. Mislove, David E. Johnson, Sylvie Levesque, and Thilo Goetz for useful conversations.

2 Basic Definitions

Our approach to relational learning starts with assuming the existence of a set A, a single A-ary relation R (*the relation of interest,* a relation defined over the set of objects that may appear in structures exemplifying patterns, whatever those objects, structures and patterns may be), and a complete lattice L.

The set A is a set of *argument names,* each of which names an argument for R, the single relation of interest. In relational learning, we wish to determine which consituents of a structure are related by the specific relation R when that structure is an instance of a general pattern. We propose to do this by mapping elements of A to elements of the structure, thus instantitating the relation of interest. Our approach will support that idea that, in a single piece of text or of data, the relation R may be instantiated in multiple ways.

From a very formal point of view, we could actually dispense with R for the following reasons:

1. At no point in our work do we use any information about the logical properties of the relation R.

2. Only the set A enters into the definition of a precedence-inclusion pattern.

Nonetheless, keeping R in mind serves to remind us that, in any structure containing objects associated with the various argument names, those objects are supposed to be related by the specific relation R, which is supposedly of some special interest.

The complete lattice L is called the *property lattice*, and its elements intuitively correspond to the sets of properties that an entity in a structure may possess, with the minimal element being assigned to entities to which no properties are attributed. To prove that arbitrary products exist in the category of precedence-inclusion patterns, we make essential use of the assumption that L is a complete lattice.

There is no reason to assume either L or A is finite, although finiteness may often be expected in applications.

When L is degenerate (i.e., a one-element complete lattice), then patterns will be based purely on the relations of precedence and inclusion, and no other properties of the elements of structures will be relevant to pattern generalization except those that relate to instantiating the relation of interest.

When A is empty, or, equivalently, when the relation R is 0-ary, then R is either the logical constant `true` or the logical constant `false`. But, in our current perspective where we permit exactly one relation of interest, there is no intrinsic difference between these two situations. While the problem of determining specific elements related by R becomes moot when A is empty, this case enables us to see clearly how the problem of relational learning is a generalization of the problem of binary classification, i.e., finding rules that determine when a structure should be assigned to a particular class.

We will have a lot to say about irreflexive, transitive binary relations, but that phrase does not trip lightly off the tongue, so we will term such relations *strict partial orders.* Those well-schooled in ordered structures may think our preference for strict partial orders instead of partial orders is a bit odd. Rest assured that we know what we are doing: it is these relations that must be preserved by pattern-preserving maps.

A *precedence-inclusion pattern* is a set P, equipped with two strict partial orders $\prec$, called *strict precedence,* and $\sqsupset$, called *strict inclusion,* along with a partial function $\alpha : A \dot{\to} P$, called the *argument naming function,* and a total function $\Lambda : P \to L$, called the *labeling function,* such that for all $x, y, z \in P$,

1. $x \prec y$ and $y \sqsupset z$ implies $x \prec z$, and
2. $y \sqsupset x$ and $y \prec z$ implies $x \prec z$.

When no confusion can result, we will call a precedence-inclusion pattern simply a *pattern.*

Space limitations prevent us from discussing how patterns generalize constituent structure trees from computational linguistics (see [4]).

When L is degenerate, we will call a precedence-inclusion pattern *purely positional.* When A is empty, we will say a precedence-inclusion pattern is a *classification pattern.* Clearly, for a purely positional pattern P, the labeling function Λ_P need not be given explicitly, because there is no choice in its definition. Similarly, for a classification pattern P, the argument naming function α_P need not be given explicitly.

If we forget about α and Λ for a moment – think of purely positional classification patterns – we can view the definition of a pattern as proposing a nontrivial modification of the definition of a strict partial order to fit an ordered pair of binary relations that allows the relations to interact, with the property that if either component of the ordered pair is the empty relation, then the only constraint on the other one is that it be a strict partial order. This informal view will be formalized in Section 4.

Striving for the simplest nontrivial examples of precedence-inclusion patterns, consider purely positional classification patterns. For instance, based on a string $S = \langle a, b, c, d, e \rangle$ of length five, we can rather arbitrarily define a five-element purely positional classification pattern

$$W = \{\langle a, b, c \rangle, \langle a \rangle, \langle b, c \rangle, \langle b \rangle, \langle d, e \rangle\}.$$

The order of the elements of $S = \langle a, b, c, d, e \rangle$ and the fact that each string in W has a unique occurrence as a substring of S determines both the strict precedence relation on W, which is given in detail by

$$\langle a,b,c\rangle \prec \langle d,e\rangle,$$
$$\langle a\rangle \prec \langle b,c\rangle \prec \langle d,e\rangle,$$
$$\langle a\rangle \prec \langle b\rangle \prec \langle d,e\rangle.$$

and the strict inclusion relation on W, which is given in detail by

$$\langle a,b,c\rangle \sqsupset \langle a\rangle,$$
$$\langle a,b,c\rangle \sqsupset \langle b,c\rangle \sqsupset \langle b\rangle.$$

Don't be misled into thinking all precedence-inclusion patterns are essentially textual. Geometric examples are given in Section 5.

If $f : X \dot{\rightarrow} Y$ is a partial function from a set X to a set Y, then we denote the *domain of definition* of f by $\mathrm{dom}\, f$, i.e., $\mathrm{dom}\, f = \{x \in X \mid f(x) \text{ is defined}\}$.

We say that a precedence-inclusion pattern P is a *subpattern* of a pattern Q if

1. as sets, $P \subseteq Q$,
2. $\mathrm{dom}\, \alpha_P = P \cap (\mathrm{dom}\, \alpha_Q)$, and
3. for all $x, y \in P$ and for all $a \in \mathrm{dom}\, \alpha_P$,
   (a) $x \prec_P y$ iff $x \prec_Q y$,
   (b) $x \sqsupset_P y$ iff $x \sqsupset_Q y$,
   (c) $\Lambda_P(x) = \Lambda_Q(x)$, and
   (d) $\alpha_P(a) = \alpha_Q(a)$.

3 Pattern-Preserving Maps

The concept of a morphism between patterns is the basis of understanding pattern generalization. Let Q and P be precedence-inclusion patterns. When there is a morphism from Q to P, then we may say *Q is a generalization of P.*

For the formal definition of a morphism, if $\mathrm{dom}\, \alpha_Q \not\subseteq \mathrm{dom}\, \alpha_P$, then there are no morphisms from Q to P. If $\mathrm{dom}\, \alpha_Q \subseteq \mathrm{dom}\, \alpha_P$, then a morphism $h : Q \rightarrow P$ from Q to P is a (total) function from Q to P satisfying, for all $x, y \in Q$ and for all $a \in \mathrm{dom}\, \alpha_Q$,

1. $x \prec_Q y$ implies $h(x) \prec_P h(y)$,
2. $x \sqsupset_Q y$ implies $h(x) \sqsupset_P h(y)$,
3. $\Lambda_Q(x) \leq \Lambda_P(h(x))$, and
4. $h(\alpha_Q(a)) = \alpha_P(a)$.

Thus, a morphism from a pattern Q to a pattern P describes how to find all the parts of Q within P in a structurally consistent way. A morphism between patterns is also called a *pattern-preserving map.*

A bijective pattern-preserving map is not necessarily an isomorphism. This is not really so weird, since it should remind us of the similar situation in topological spaces, where a bijective continuous map is not necessarily a homeomorphism. However, we do have the following theorem that applies to finite patterns.

Theorem 3.1 *A bijective endomorphism of a finite pattern is necessarily an automorphism.*

In trying to find similar theorems in mathematics to Theorem 3.1, we came up with the following result in general topology, which is new, as far as we know.

Theorem 3.2 *A bijective continuous self-map of a finite topological space is necessarily a homeomorphism.*

It is possible for a pattern-preserving epimorphism to fail to be surjective.

4 Categories of Patterns

Let ***Set*** denote the category of sets, and let ***SPO*** be the category of strictly partially ordered sets, with strictly monotone maps as morphisms.

Theorem 4.1 *The categories **Set** and **SPO** can both be regarded as (i.e., are isomorphic to) full, coreflective subcategories of the category **PI**(A, L) of patterns.*

Theorem 4.2 *Arbitrary products exist in the category **PI**(A, L) of precedence-inclusion patterns.*

Theorem 4.3 *If the set A of argument names is empty, so that $\boldsymbol{PI}(\emptyset, L)$ is a category of classification patterns, then arbitrary coproducts of patterns exist in $\boldsymbol{PI}(\emptyset, L)$.*

Theorem 4.4 *The category of purely positional classification patterns is Cartesian closed.*

5 An Example of Pattern Generalization

At the start of Section 3, we explained that, for patterns P and Q, saying Q is a generalization of P means there exists a pattern-preserving map

$$f : Q \to P.$$

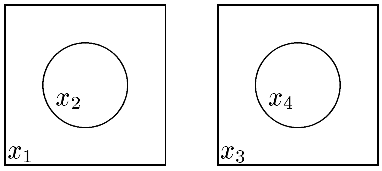

Figure 1: The pattern X, in which $R(x_2, x_4)$ holds.

Now suppose I is an index set and $\mathcal{P} = \{P_i \mid i \in I\}$ is an I-indexed set of precedence-inclusion patterns. Then we say that Q is a *generalization* of $\mathcal{P}$ if there exists an I-indexed set of pattern-preserving maps

$$\{f_i : Q \to P_i \mid i \in I\}.$$

This asserts all of the structure found in Q can be found in each of the patterns P_i. We will say that Q is a *most specific generalization* of $\mathcal{P}$ if every generalization of $\mathcal{P}$ is also a generalization of Q. That most specific generalizations always exist in categories of patterns is taken care of by the Theorem 5.1.

Theorem 5.1 *If I is an index set and $\mathcal{P} = \{P_i \mid i \in I\}$ is an I-indexed set of precedence-inclusion patterns, then the product of $\mathcal{P}$ is a most specific generalization of $\mathcal{P}$.*

While Theorem 5.1 guarantees the existence of most specific generalizations, they are by no means unique up to isomorphism. For instance, in a category of classification patterns (so that, by Theorem 4.3, coproducts exist) then a coproduct of any number of copies of a most specific generalization of a set of patterns is also a most specific generalization of that set of patterns.

One might then think that the product of a set of patterns, given that products in a category are unique up to isomorphism, is the best generalization of a set of patterns, but this would be wrong, because in general we can do much better! We can get a minimal most specific generalization:

Definition Let I be an index set and let $\mathcal{P} = \{P_i \mid i \in I\}$ be an I-indexed set of precedence-inclusion patterns. A *minimal most specific generalization* of $\mathcal{P}$ is a most specific generalization M of $\mathcal{P}$ such that no proper subpattern of M is a most specific generalization of $\mathcal{P}$.

This concept will be illustrated by an example, and will be given additional mathematical substance in Theorem 6.2.

Consider two patterns $X = \{x_1, x_2, x_3, x_4\}$ and $Y = \{y_1, y_2, y_3, y_4\}$, based on the scenes shown in Figures 1 and 2. Here precise dimensions do not matter, and each pattern consists of two rectangular elements and of two circular elements. Thus, we take our set of labels L to be the power set of the two-element set $\{\square, \circ\}$ where each of the eight picture elements is assigned by the

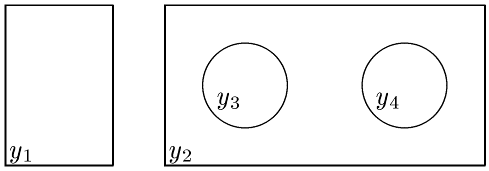

Figure 2: The pattern Y, in which $R(y_3, y_4)$ holds.

appropriate labeling function Λ either the singleton set $\{\square\}$ or the singleton set $\{\circ\}$, according to its shape. In all of the geometric patterns in this section, strict precedence $a \prec b$ means picture element a is entirely to the left of picture element b, while strict inclusion $a \sqsupset b$ means b is entirely within a. For example, $x_1 \prec_X x_3$ and $y_2 \sqsupset_Y y_3$. We could have described these particular patterns as partial parses of stings in various ways. For instance, X, to the extent that we have described it so far, might have been written

$$(x_1^\square(x_2^\circ))(x_3^\square(x_4^\circ)),$$

which can be viewed as a decorated parse of a string of length four, in which names for elements exponentiated with their labels immediately follow the appropriate leading parenthesis. Similarly Y, to the extent that we have described it so far, might have been written

$$(y_1^\square)(y_2^\square(y_3^\circ)(y_4^\circ)).$$

Of course, not every geometric pattern is susceptible to this kind of linearization. We will also use these examples to illustrate relational learning, so suppose in pattern X, $R(x_2, x_4)$ is true, and that in pattern Y, $R(y_3, y_4)$ is true. The idea here is that $R(a, b)$ holds for elements of a geometric pattern if a and b have the right properties and reside in the correct places in a pattern that is a specialization of a suitable generalization of X and Y, such as $X \times Y$, the product of X and Y. However, based on the examples, we have the strong feeling that $R(a, b)$ should imply that both a and b are labeled as circular and that a strictly precedes b. To complete the formal definition of the patterns X and Y, let's name the first and second arguments of R respectively a_1 and a_2, so that $A = \{a_1, a_2\}$. Then $\alpha_X(a_1) = x_2$ and $\alpha_X(a_2) = x_4$, while $\alpha_Y(a_1) = y_3$ and $\alpha_Y(a_2) = y_4$. If we chose to present these patterns as parses, then we could have added the argument names to the exponents, obtaining

$$(x_1^\square(x_2^{\circ,\, a_1}))(x_3^\square(x_4^{\circ,\, a_2}))$$

for X and

$$(y_1^\square)(y_2^\square(y_3^{\circ,\, a_1})(y_4^{\circ,\, a_2}))$$

for Y.

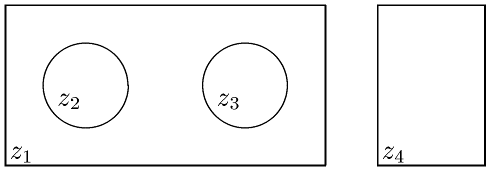

Figure 3: The stripped pattern Z. Does $R(z_2, z_3)$ hold?

By a *stripped* pattern, we mean a pattern P for which the partial function α_P is empty. Of course, when working in a category of classification patterns, all of the objects are stripped patterns. This concept proves to be more useful in categories of patterns in which the set A of argument names is nonempty. In relational learning, the examples on which the learning is based are fully analyzed and unstripped (i.e., not stripped), such as X and Y above. However, the patterns to which we want to apply what is learned are stripped, and if an instance of the relation of interest is found in one of them, then an unstripped pattern can be created. For a pattern P, let $P^\dagger$ be the stripped pattern formed by replacing the argument naming function α_P by the empty function. For example, the description of X as

$$(x_1^\square(x_2^{\circ,\, a_1}))(x_3^\square(x_4^{\circ,\, a_2}))$$

leads to the description of $X^\dagger$ as

$$(x_1^\square(x_2^\circ))(x_3^\square(x_4^\circ)).$$

Actually, what we are defining here is the *stripping functor*

$$(\cdot)^\dagger : \boldsymbol{PI}(A, L) \to \boldsymbol{PI}(\emptyset, L).$$

For a pattern-preserving map $f : P \to Q$ in $\boldsymbol{PI}(A, L)$, $f^\dagger = f$. The inclusion functor of the full subcategory $\boldsymbol{PI}(\emptyset, L)$ into $\boldsymbol{PI}(A, L)$ is left adjoint to the stripping functor, which shows that $\boldsymbol{PI}(\emptyset, L)$ is a full, coreflective subcategory of $\boldsymbol{PI}(A, L)$.

Consider the pattern Z shown in Figure 3, which has a linear representation as

$$(z_1^\square(z_2^\circ)(z_3^\circ))(z_4^\square).$$

This is a stripped pattern. A question we would like to address is whether or not $R(z_2, z_3)$ holds in Z, on the basis of the unstripped examples X and Y. (Actually, the phrase "$R(z_2, z_3)$ holds in Z" doesn't formally make sense because the argument naming function for Z is known to be empty. We should instead say "$R(z_2, z_3)$ holds in Z', where Z' satisfies $(Z')^\dagger = Z$." With this explanation, we intend to use the shorter, but technically inaccurate phrase, where the true meaning is clear.) We do see that both z_2 and z_3 are circular,

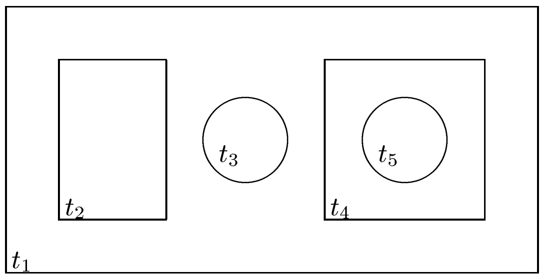

Figure 4: The stripped pattern T. Does $R(t_3, t_5)$ hold?

and z_2 certainly strictly precedes z_3. However, a careful study of X and Y shows us that in both cases the second circle is strictly preceded by a rectangle, and this does not happen in Z. So we conclude that $R(z_2, z_3)$ does not hold in Z, at least on the basis of informal generalization from our two examples.

At this point we are in a position to appreciate the inadequacy of attempting to do pattern generalization by removing elements from one pattern, say X, based on some sort of inconsistency with features of a second pattern, say Y. Just think about:

1. You can't remove either circle x_2 or circle x_4 if your goal is to learn the relation R.

2. Removing the rectangle x_1 causes loss of the property that, in both X and Y, the second circle is strictly preceded by a rectangle. This was the very fact used to resolve negatively the question of whether or not $R(z_2, z_3)$ holds in Z.

3. Removing the rectangle x_3 causes loss of the property that, in both X and Y, the second circle is strictly included in a rectangle.

Parallel reasoning indicates that you can't get away with only removing properties of elements instead of removing the element completely. The bottom line is that constructing generalizations cannot be done by removal of parts from a pattern.

A trickier situation is given by the stripped pattern T, displayed in Figure 4 and possessing a linear representation as

$$(t_1^{\Box}(t_2^{\Box})(t_3^{\circ})(t_4^{\Box}(t_5^{\circ}))).$$

Here the question is whether or not $R(t_3, t_5)$ holds in T. We claim that we should conclude that $R(t_3, t_5)$ holds in T because we will show that there is a minimal most specific generalization M of X and Y such that

1. the domain of the argument naming function α_M is all of $A = \{a_1, a_2\}$, which in conjunction with a pattern-preserving map

$$f : M^{\dagger} \to T$$

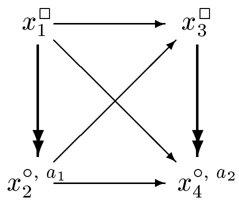

Figure 5: Directed acyclic graph representation of X.

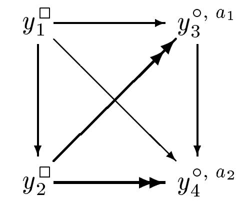

Figure 6: Directed acyclic graph representation of Y.

enables us by replacing the empty function α_T with $\alpha_{T'} = f \circ \alpha_M$ to create from T an unstripped pattern T' such that

$$(T')^\dagger = T$$

and such that

$$f : M \to T'$$

– that's the same f as before – is a pattern-preserving map, and

2. inspection of T' shows $R(t_3, t_5)$ holds in T'.

The first step in identifying M is to look closely at the product $X \times Y$ of X and Y. As we can see, it cannot be depicted pictorially in the manner of the other patterns of this section, nor does it have a linear representation via disciplined insertion of parentheses into a string. It can be represented by a edge-labeled directed acyclic graph. To construct the representation, we start by considering similar representations of X and Y, shown in Figures 5 and 6, respectively. In these graphs, we are using thin arrows to indicate strict precedence and double-headed thick arrows to indicate strict inclusion. In each of these graphs every ordered pair in both relations is depicted by an arrow. Using these two graphs, we can now depict $X \times Y$, as shown in Figure 7. In the graph, labels and argument names are placed at the vertices, with a bullet placed for those elements that have the empty set of labels and no argument names.

This all looks pretty complicated, but there is a particular kind of pattern-preserving map that helps us reduce the complexity of $X \times Y$ in small steps. If P is a pattern, a *retraction* of P is an idempotent endomorphism $r : P \to P$, i.e., for all $x \in P$, $r(r(x)) = r(x)$. A subpattern of P that is the image of a

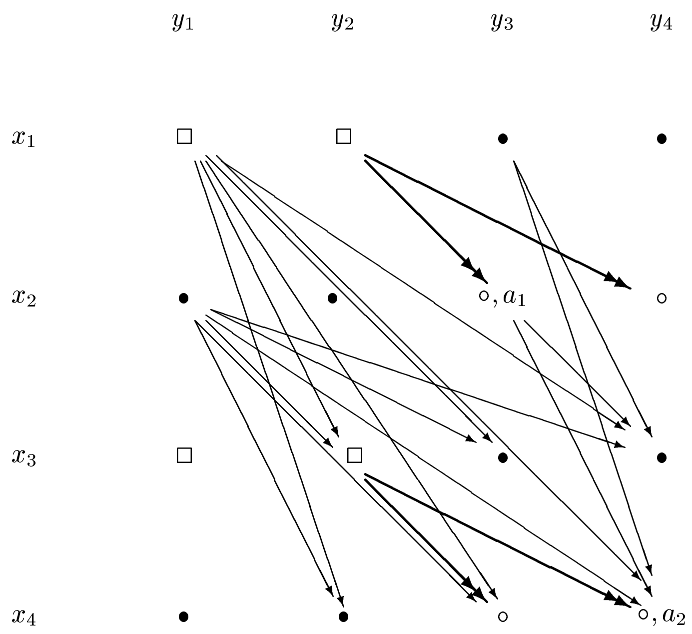

Figure 7: Directed acyclic graph representation of $X \times Y$.

retraction, which is the same as being the set of fixed points of the retraction, is known as a *retract* of P. We use the phrases "R is a retract of P" and "P has a retract R" interchangably to express the fact that there is a retraction of P whose image is R. A retract of P is a *proper retract* if it is a proper subpattern of P. This concept is modeled on the similarly named concepts found in topology (see [3], p. 216) and in domain theory (see [6] or [1], p. 344). Proposition 5.2 tells why we like retracts of patterns.

Proposition 5.2 *Let R be a retract of a precedence-inclusion pattern P.*

1. *By composing with the corestriction of a retraction that defines R, we see that every generalization of P is a generalization, too, of R.*
2. *By composing with the inclusion map from R to P, we see that every pattern that is generalized by P is also generalized by R.*
3. *If P is finite, then a proper retract R has smaller cardinality than P.*

Before constructing retracts, we need to understand some of their theory. For instance, the next proposition says being a retract is transitive.

Proposition 5.3 *If Q is a retract of a pattern P, and if R is a retract of Q, then R is a retract of P.*

It might look like the concept of a retract is overkill because perhaps the image Q of an arbitrary endomorphism $f : P \to P$ might serve the same purpose as a retract of P. However, at least for finite patterns P, Theorem 5.4 shows that each endomorphism is closely connected to a retraction. Thus, just finding an endomorphism of a finite pattern whose image is a proper subset of the pattern leads directly to a proper retract of the pattern.

Theorem 5.4 *If $f : P \to P$ is an endomorphism of a finite pattern P, then there exists a positive integer n such that $f^n : P \to P$, the composition of f with itself n times, is a retraction.*

A proper retract of the product $X \times Y$ of our two example patterns will have the two the desired generalization properties that are possessed by $X \times Y$:

1. it will generalize both X and Y, and
2. any generalization of both X and Y will be a generalization of it.

And, even better, it will be smaller! How small can it be? This is answered by Theorem 5.5, which first needs a definition: a *minimal retract* M is a pattern that has no nontrivial retraction (i.e., $1_M : M \to M$ is the only retraction of M). *Note that being a minimal retract is a property that is intrinsic to a pattern and does not depend on larger pattern.* It is entirely possible for a subpattern of a minimal retract to again be a minimal retract. When we do speak of a *minimal retract of a pattern P*, we mean a retract of P that is also a minimal retract, and hence, by Proposition 5.3 contains no proper retract of P. Note that these definitions apply to infinite patterns as well as finite patterns.

Theorem 5.5 *For every finite pattern P, there is a retraction $r : P \to P$ whose image is a minimal retract.*

Using a sequence of retractions each of which moves only one point, we can obtain the retract M of $X \times Y$ that is shown in Figure 8. Starting with the observation that the elements of M that have argument names attached to them are fixed points of every endomorphism of M, it is easy to see that M has no endomorphism except the identity function. Therefore, M is a minimal retract.

Clearly, in seeking to learn if a pattern P has all the structure common to both X and Y, it is vastly simpler to carry out a direct check for the existence of a pattern-preserving function $h : M \to P$, as opposed to a direct check for the existence of a pattern-preserving function $g : X \times Y \to P$, even though the existence of the one function is logically equivalent to the existence of the other.

It is possible to give a verbal description of the generalization M, which is a little surprising in light of the complexity of with $X \times Y$. It reads something like this:

> There is a rectangle that strictly precedes a rectangle that strictly includes a circle that is the second argument to R and which is preceded by a circle that is the first argument to R, which is strictly contained in a rectangle.

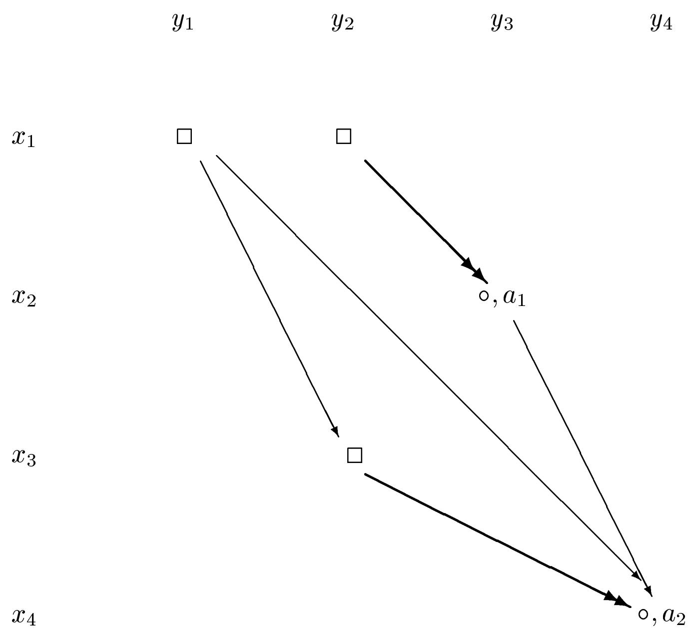

Figure 8: Directed acyclic graph representation of the retract M.

In spite of this simple recital, it can be shown that M *cannot be described as a parse structure on a string, even though X and Y could be so described.*

The pattern M has geometric representation as an arrangement of circles and rectangles, as shown in Figure 9, where we let

$$\begin{aligned} m_1 &= \langle x_1, y_1 \rangle \\ m_2 &= \langle x_1, y_2 \rangle \\ m_3 &= \langle x_2, y_3 \rangle \\ m_4 &= \langle x_3, y_2 \rangle \\ m_5 &= \langle x_4, y_4 \rangle \end{aligned}$$

Now consider the stripped pattern $M^\dagger$. The function $f : M^\dagger \to T$ given by

$$\begin{aligned} m_1 &\mapsto t_2 \\ m_2 &\mapsto t_1 \\ m_3 &\mapsto t_3 \\ m_4 &\mapsto t_4 \\ m_5 &\mapsto t_5 \end{aligned}$$

is a pattern-preserving map (and, by the way, is an example of a bijective pattern-preserving map that is not an isomorphism). Therefore, if we extend T to an unstripped pattern T' by defining the argument naming function $\alpha_{T'}$ as $\alpha_{T'} = f \circ \alpha_M$, then $f : M \to T'$ becomes a pattern-preserving map. Hence we conclude that $R(t_3, t_5)$ holds in T, as desired.

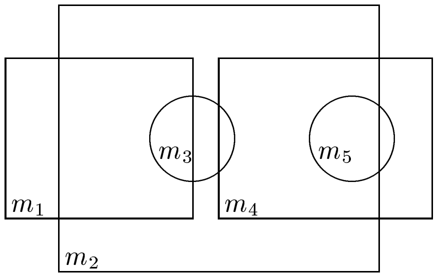

Figure 9: The minimal retract M, in which $R(m_3, m_5)$ holds.

Moreover, because there is no pattern-preserving map $h : M^\dagger \to Z$, we can say that on the basis of this formal analysis that we cannot determine that $R(z_2, z_3)$ holds in Z based on generalizing from X and Y alone.

It is easy to see M meets the definition of a minimal most specific generalization of X and Y.

6 The Theory of Generalization

Much of the theory has already been laid out in the course of presenting the example in Section 5. Here we will pull it together by providing results that show that, for finite sets of finite patterns, there is only one minimal most specific generalization, up to isomorphism, and that any finite most specific generalization, not just a product of a set of patterns, can be used as a starting point in constructing a minimal most specific generalization.

Note that in Theorem 6.1, we do not assume that the individual patterns P_i are finite, nor that the index set I is finite. It actually is possible for an infinite set of infinite patterns to have a finite most specific generalization. (Consider an infinite set $\mathcal{P}$ of ordinary infinite sets considered as patterns, i.e., with empty strict precedence and strict inclusion relations. Then a singleton set, regarded as a pattern, again with empty relations, is a minimal retract and it is a most specific generalization of $\mathcal{P}$. This should not be surprising because pattern generalization is mostly about relations that must be present, so no relations lead to trivial generalizations.)

Theorem 6.1 *Let I be an index set and let $\mathcal{P} = \{P_i \mid i \in I\}$ be an I-indexed set of precedence-inclusion patterns. Suppose there is a finite pattern Q that is a most specific generalization of $\mathcal{P}$.*

1. *There is a retract M of Q such that M is a finite minimal retract and M is a most specific generalization of $\mathcal{P}$.*

2. *If N is any minimal retract that is also is a most specific generalization of $\mathcal{P}$, then M is isomorphic to N.*

We now give the Theorem 6.2, the main goal of this section. For the finite case, it covers the existence and uniqueness of the minimal most specific generalization, and, implicitly, how to compute it.

Theorem 6.2 *Let I be a finite index set and let $\mathcal{P} = \{P_i \mid i \in I\}$ be an I-indexed set of finite precedence-inclusion patterns.*

1. *There exists a minimal most specific generalization M of $\mathcal{P}$.*
2. *M is a finite minimal retract.*
3. *Any minimal most specific generalization of $\mathcal{P}$ is isomorphic to M.*
4. *Any finite most specific generalization Q of $\mathcal{P}$ has a retraction $r : Q \to Q$ whose image is isomorphic to M.*

The last claim in Theorem 6.2 is significant. It says that, when I is finite and the patterns P_i are all finite, we do not need to start with the product of $\mathcal{P} = \{P_i \mid i \in I\}$ for computing a minimal most specific generalization. Any finite most specific generalization will suffice.

References

[1] H.P. Barendregt. Functional programming and lambda calculus. In J. van Leeuvan, editor, *Handbook of Theoetical Computer Science, Vol. B: Formal Models and Semantics*, pages 321–364. The MIT Press/Elsevier, 1990.

[2] F. Bergadano and D. Gunetti. *Inductive Logic Programming.* MIT Press, Cambridge, 1996.

[3] J.R. Munkres. *Topology: A First Course.* Prentice Hall, Inc., Engelwood Cliffs, 1975.

[4] B.H. Partee, A. ter Meulen, and R.E. Wall. *Mathematical Methods in Linguistics.* Kluwer Academic Publishers, Dordrecht, 1990.

[5] G.D. Plotkin. A note on inductive generalization. In B. Meltzer and D. Michie, editors, *Machine Intelligence 5*, pages 153–163. American Elsevier Publishing Co., Inc., 1970.

[6] D.S. Scott. Continuous lattices. In F.W. Lawvere, editor, *Toposes, Algebraic Geometry, and Logic*, volume 274 of *Lecture Notes in Mathematics*, pages 97–136. Springer-Verlag, 1972.