RC22770 (W0304-089) April 9, 2003
Computer Science

IBM Research Report

Patterns Based on Multiple Interacting Partial Orders

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

Patterns Based on Multiple Interacting Partial Orders

Frank J. Oles

IBM T.J. Watson Research Center
Yorktown Heights, New York 10598
oles@us.ibm.com

March 21, 2003

Our aim here is to outline a theory of patterns where the patterns are based on multiple interacting strict partial orders. A detailed paper on this subject is in preparation.

For instance, a pattern extracted from a video may have as its elements some picture elements occurring in individual frames of the video. There are four natural strict partial orders relating those elements: elements may be ordered by the ordering of the frames in which they occur, and elements within a single frame may be above one another, to the left of one another, and included within one another. Other examples come from parsing text, where linguistic entities in a parse tree may precede one another or they may contain one another. In fact, categories of what we call **2**-patterns include among their objects constituent structure trees, as they are normally defined in computational linguistics.

The theory is motivated by problems of relational learning, an important kind of inductive learning in which one wishes, from known training instances of related elements of structures, to create general rules for identifying elements of other structures that bear the same relation to one another. For instance, one may wish to learn from text examples (see [2]) patterns expressing the fact that a disease has a symptom (a binary relation) or that a person has a position in a company (a ternary relation). As another example, one may want to learn the properties that some nucleotide sequences have in common as well as learning out how to pick out a particular subsequence of interest (a unary relation). The general supervised learning problem of classification can be cast as learning a 0-ary relation. We call our inductive learning approach *category-theoretic inductive learning* since the notion of generalization we employ is based on morphisms between structured objects.

We say that an ordered pair $\langle \prec, \sqsupset \rangle$ of binary relations on a set P is *interactively transitive* if both $\prec$ and $\sqsupset$ are transitive and, 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 $x \prec z$.

Interactive transitivity is an extension of the concept of a transitive, binary relation to an ordered pair of binary relations. By reading $\prec$ as "precedes" and $\sqsupset$ as "includes," the intuitive content of these axioms may become clear. For instance, in two-dimensional images, the ordered pair of relations ⟨is to the left of, contains⟩ is interactively transitive, as also is the ordered pair ⟨is above, contains⟩.

Roughly speaking, a *precedence-inclusion pattern* is a set equipped with a strictly partially ordered set of strict partial orders, along with some additional structure, in which the strict partial order on the strict partial orders is taken to assert that each related pair of strict partial orders obeys the axioms of interactive transitivity. A precise definition starts in an algebraic style. A *pattern signature* an ordered triple $\Sigma = \langle O, A, L \rangle$ in which

1. O, the *order symbol set* of Σ, is a strictly partially ordered set of binary relation symbols, each of which is intended to be interpreted as a strict partial order on a set,

2. A, the *argument name set* of Σ, is a set whose elements name the arguments for some A-ary relation of interest, instances of which may be found in patterns, and

3. L, the *property poset* of Σ, is a bounded complete poset of labels that may be attached to elements of structures.

Definition Let $\Sigma = \langle O, A, L \rangle$ be a pattern signature. We will say a set P is a Σ-*pattern* when every $\sigma \in O$ has an intepretation $\prec_{\sigma,P}$ as a strict partial order on P, along with a partial function $\alpha_P : A \dot{\to} P$, called the *argument naming function,* and a total function $\Lambda_P : P \to L$, called the *labeling function,* such that $\sigma < \tau$ implies that the ordered pair of relations $\langle \prec_{\sigma,P}, \prec_{\tau,P} \rangle$ is interactively transitive.

When Σ is clear from context we call a Σ-pattern a *precedence-inclusion pattern.* Thus, when the order symbol set O is empty, Σ-patterns are just sets with some additional structure. When the order symbol set is one-element set, then Σ-patterns are strictly partially ordered sets with some additional structure. More interesting examples arise when the order symbol set is nontrivial. We can construct examples of precedence-inclusion patterns in which the order symbol set has arbitrary finite depth.

A Σ-pattern Q is a generalization of a Σ-pattern P if there is a pattern-preserving map, i.e., a morphism in the category of Σ-patterns, from Q to P. The reader can now guess at the definition of a most specific generalization (msg) of a set of patterns, which corresponds to a least general generalization (lgg) in inductive logic programming (see [3, 1]). Like lgg's, msg's are not unique, although products of patterns give (typically very large) examples of them. The problem with a large msg is that it would be computationally hard to test if another pattern is a specialization of it. A *minimal most specific generalization* of a set $\mathcal{P}$ of patterns is an msg of $\mathcal{P}$ no subpattern of which is an msg of $\mathcal{P}$. These are the kinds of generalizations we really want. A *retraction* of a precedence-inclusion pattern P is an idempotent endomorphism $r : P \to P$, and the set of fixed points of a retraction defines a pattern called a *retract* of P. A pattern having no proper retracts is said to be *fully retracted.*

Here is the main theorem. For the finite case, it covers the existence and uniqueness of the minimal most specific generalization, and, implicitly, tells how to compute it.

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

1. There exists a minimal most specific generalization M of $\mathcal{P}$.
2. M is finite and fully retracted.
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.

We end with the explicit description of a simple procedure that is guaranteed to return the minimal most specific generalization of a nonempty finite set $\{P_1, P_2, \cdots, P_n\}$ of finite Σ-patterns:

Minimal Most Specific Generalization Procedure

> $M := P_1 \times P_2 \times \cdots \times P_n$;
> while there exists a proper retract Q of M
>   do $M := Q$;
> return M;

We have an efficient implementation in Java of this procedure for pairs of finite **2**-patterns. The resulting minimal msg's can readily be used for discovering new instances of relations.

References

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

[2] C. Cardie. Empirical methods in information extraction. *AI Magazine*, 18(4):65–80, 1997.

[3] 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.