Minimal proof obligations for ordered sets

Dirk Nowotka

Turku Centre for Computer Science, 20520 Turku, Finland

Abstract

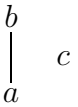
We show how the smallest set of lemmas can be generated that is sufficient to prove an assumed partial-order relation on a finite set. This idea is developed from a mathematical formulation up to an algorithm implemented in ML. The process of establishing a Hasse diagram from a set of subsets serves as an applicative example.

1 Introduction

This paper describes an algorithm that is useful for minimizing the proof effort to establish a partial order over a finite set.

Suppose, we have a finite set of objects A and a preorder relation $\preccurlyeq$ on A, and also, it is assumed to be hard to check whether or not $x \preccurlyeq y$, for some $x, y \in A$. Further, suppose that something is already known about $\preccurlyeq$, namely a partial order $\leq$ contained in $\preccurlyeq$. Now, let our goal be to show that actually $\leq = \preccurlyeq$. For that, we need to check that all elements in A not related in $\leq$ are also not related in $\preccurlyeq$. Formally, for all $x, y \in A$ we have $x \not\leq y$ implies $x \not\preccurlyeq y$. Since it is assumed to be difficult to check the membership property for $\preccurlyeq$, we look for the smallest number of pairs $(x, y) \notin \leq$ to be investigated.

The following example illustrates that often not all possible pairs have to be considered. Let $A = \{a, b, c\}$ be a set of sets and we investigate the subset relation on A. Assume, further we only know that $a \subseteq b$, so in this case $\leq = \{(a, a), (b, b), (c, c), (a, b)\}$. Proving now that $\leq = \subseteq$, means to show that the following Hasse diagram of A is correct for $\subseteq$.



For that, we need to show that for all $x, y \in A$, if $x \not\leq y$ then $x \not\subseteq y$, which means in our case to check: $b \not\subseteq a$, $b \not\subseteq c$, $c \not\subseteq b$, $a \not\subseteq c$, and $c \not\subseteq a$. But

actually, once we know that $a \not\subseteq c$, we immediately have that $b \not\subseteq c$ because we know $a \subseteq b$, and similarly, from $c \not\subseteq b$ follows that $c \not\subseteq a$. So instead of five propositions we need to prove only three.

This paper shows how the smallest set of pairs $S \subseteq \{(x, y) \mid x \not\leq y\}$, from which follows that $\leq = \preccurlyeq$, can be calculated. We give an algorithm and also show that S is unique for a given problem.

The topic of this paper was originally motivated by the practical goal to reduce work on comparing sets of formal languages, see for example [FV98] where this result is cited as a personal communication. There, given a set of sets of languages which are to be arranged in a Hasse diagram can be a nontrivial task, since comparing the expressive power of certain methods to generate a language might be difficult.

This paper is divided into two parts: Section 2, where the result is given, and Section 3, where we give an implementation of it.

2 Minimal Proof Obligations

Our terminology is chosen acording to the motivation given in the introduction. Since this is about a method to verify a finite partial-order relation on its elements, let us call a finite nonempty set *universe* which is ment to contain the objects of our interest, and let a preorder on a universe represent the actual relation the induced partial-order relation of which we would like to determine, called *goal*. Let further a partial-order relation that is contained in the goal denote a set of *assumptions* in a goal. A set of assumptions should represent the knowledge that we assume to be precise about the goal, it is defined to be a partial-order instead of a preorder by identifying elements of the universe in question that are known to be related symetrically in its goal. By proving the *validity* of our goal, we mean to show that our assumptions are precise, that is, the set of assumptions equals the goal.

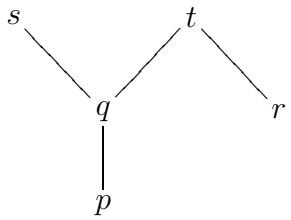
Let us fix a universe A, a goal $\preccurlyeq$ on A, and a set of assumptions $\leq$ in $\preccurlyeq$. The result we present now is how to determine the smallest set of pairs not in $\leq$ that is sufficient to show $\leq = \preccurlyeq$ and how that can be used in prove efforts such as arranging sets in a Hasse diagram.

We write $x \ \rho \ y$ for a pair (x, y) contained in ρ, where $x \in M$, $y \in N$, and $\rho \subseteq M \times N$. Our notations are taken from [BS81].

Example 1 (Hasse diagram) *Let us fix a running example. Let our universe $M = \{p, q, r, s, t\}$ be a set of sets. Suppose that the subset relation $\subseteq$*

on M is our goal with the known pairs $p \subseteq q$, $q \subseteq s$, $q \subseteq t$, and $r \subseteq t$, so, our set of assumptions $\subset$ is the transitive reflexive closure of the set $\{(p,q),(q,s),(q,t),(r,t)\}$.

We want to show that the assumed set inclusions are proper and elements which are not related by $\subseteq$ are incomparable indeed. In fact, we want to prove that



is the most general Hasse diagram for $(M,\subseteq)$. Of course, we need to show that q properly contains p, i.e., $p \not\subseteq q$, that s and r are incomparable, i.e., $r \not\subseteq s$ and $s \not\subseteq r$, and so on. Naturally, we wish to prove only those lemmas which are absolutely necessary to establish the result under the given assumption.

In order to prove $\leq = \preccurlyeq$, we have to show that any pair (x,y) not in $\leq$ is also not in $\preccurlyeq$. This is easy to see, since $\preccurlyeq \subseteq \leq$ by the above requirement, and we have $\leq \subseteq \preccurlyeq$ by definition.

Let $\sqsupset = \leq^c$ be called the *complete set of lemmas*, so $x \sqsupset y$, with $x, y \in A$, is called a *lemma*.

The complete set of lemmas is the counterpart of our assumptions. It gives us the lemmas that ought to be proven to get the complete picture, i.e., to have a statement about every possible relation between the investigated objects which does not contradict the assumptions. In some sense, the proof relation gives us *complete negative knowledge* by saying what is not known by assumption.

Example 2 (Hasse diagram—continued) *In this example, $x \sqsupset y$ translates to $x \not\subseteq y$ for $x, y \in M$. There are exactly fourteen lemmas in $\sqsupset$:*

$$\begin{array}{lllllll} \{q \not\subseteq p, & p \not\subseteq t, & s \not\subseteq q, & r \not\subseteq q, & t \not\subseteq q, & s \not\subseteq t, & s \not\subseteq r, \\ p \not\subseteq s, & p \not\subseteq r, & q \not\subseteq r, & r \not\subseteq p, & t \not\subseteq r, & t \not\subseteq s, & r \not\subseteq s\} \end{array}$$

Fortunately, most lemmas follow from others directly, and we will see which of them.

Now, we observe that some lemmas follow from others directly by using the set of assumtions, e.g., let $x_1 \sqsupset y_1$ be established, which translates to $x_1 \not\preccurlyeq y_1$, and let also $x_1 \preccurlyeq x_2$ and $y_2 \preccurlyeq y_1$ by assumption, then $x_2 \not\preccurlyeq y_2$ immediately follows, since $x_1 \preccurlyeq x_2 \preccurlyeq y_2 \preccurlyeq y_1$ would give a contradiction. This observation is expressed in the following definition.

Definition 3 *The* lemma structure $\rightarrow \subseteq \sqsupset \times \sqsupset$ *on a complete set of lemmas* $\sqsupset$ *is defined by*

$$(x_1, y_1) \rightarrow (x_2, y_2) \iff x_1 \leq x_2 \text{ and } y_2 \leq y_1$$

where $(x_1, y_1), (x_2, y_2) \in \sqsupset$.

Proposition 4 *A lemma structure* $\rightarrow$ *is a partial-order relation.*

PROOF. *Reflexivity:* For any (x, y) in $\sqsupset$, we have $(x, y) \rightarrow (x, y)$ by the reflexivity of $\leq$. *Antisymmetry:* Let $(x_1, y_1) \rightarrow (x_2, y_2)$ and $(x_2, y_2) \rightarrow (x_1, y_1)$, then $x_1 \leq x_2$, $x_2 \leq x_1$, $y_2 \leq y_1$, and $y_1 \leq y_2$, and also $x_1 = x_2$ and $y_1 = y_2$ since $\leq$ is antisymmetric, hence, $(x_1, y_1) = (x_2, y_2)$. *Transitivity:* Let $(x_1, y_1) \rightarrow (x_3, y_3)$ and $(x_3, y_3) \rightarrow (x_2, y_2)$, then $x_1 \leq x_3$, $x_3 \leq x_2$, $y_2 \leq y_3$, and $y_3 \leq y_1$, and also $x_1 \leq x_2$ and $y_2 \leq y_1$ since $\leq$ is transitive, hence, $(x_1, y_1) \rightarrow (x_2, y_2)$.

Example 5 (Hasse diagram—continued) *Reading* $\rightarrow$ *as "implies"*

$$(q \not\subseteq r) \rightarrow (s \not\subseteq r)$$

holds in our example, which follows from the assumption $q \subseteq s$.

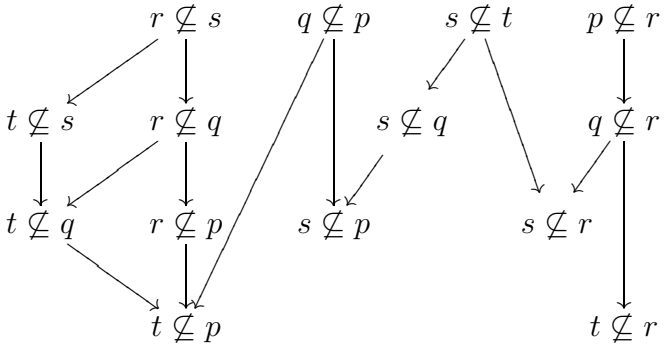
The following theorem shows that the maximal elements in $\rightarrow$ are necessary and sufficient to prove $\leq = \preccurlyeq$. Let $\rightarrow_{\max} = \{x \in \sqsupset \mid \text{if } y \rightarrow x \text{ then } x = y, \text{ for all } y \in \sqsupset\}$ denote the maximal elements in $\rightarrow$. It is clear that every nonempty finite partial-order has a nonempty set of maximal elements.

Theorem 6 $\rightarrow_{\max}$ *is the smallest set of lemmas to prove* $\leq = \preccurlyeq$.

PROOF. (i) We have to show that $\sqsupset$ follows from $\rightarrow_{\max}$. The set $\rightarrow_{\max}$ is not empty if $\sqsupset$ is not empty, since $\rightarrow$ is a partial-order by Proposition 4. So, for every lemma $x \sqsupset y$ there is a lemma $x' \sqsupset y'$ with $(p', q') \in \rightarrow_{\max}$ such that $(x', y') \rightarrow (x, y)$ by the definition of $\rightarrow_{\max}$. (ii) Every element in $\rightarrow_{\max}$ only follows from itself, which is clear from the definition of $\rightarrow_{\max}$.

Example 7 (Hasse diagram—continued) *The lemma structure of our example is indicated by the following graph, where reflexive and transitive edges*

are omitted:



One can check that the proposed Hasse diagram on page 3 follows from our set of assumptions and the four lemmas $r \not\subseteq s$, $q \not\subseteq p$, $s \not\subseteq t$, and $p \not\subseteq r$.

3 Implementation

The algorithm described in the previous section—take the complement of the set of assumptions and find the maximal elements in the corresponding lemma structure—has been imlpemented as a Haskell program which reads a file containing the describtion of a relation, calculates the transitive reflexive closure, and takes the smallest partial-order relation of that as set of assumptions. A file is generated where these steps are documented and the solution is given. The program can be found at [Now] and it requires the Glasgow Haskell Compiler [GHC] to run.

The following Haskell listing shows the core of our program. It defines a function `minlem` that takes a list of pairs defining a partial-order relation over some eqtype to a list of pairs of the same type. The argument of `minlem` represents a set of assumptions, and the result of its evaluation gives the smallest set of pairs that have to be checked to prove the assumption to be precise.

```
import List

elements :: Eq a => [(a,b)] -> [a]
elements ord = (nub . fst . unzip) ord

complement :: Eq a => [(a,a)] -> [(a,a)]
complement ord = let elems = elements ord
            in  [(x,y) | x <- elems, y <- elems,
                         (x,y) `notElem` ord]

lemma_structure :: Eq a => [(a,a)] -> [(a,a)] -> [((a,a),(a,a))]
```

```
lemma_structure ord lems =
  [((x,y),(x',y')) | (x,y) <- lems, (x',y') <- lems,
                     (x,x') `elem` ord, (y',y) `elem` ord]

maxElems :: Eq a => [(a,a)] -> [a] -> [a]
maxElems lemStruct lems =
  let pred x y = if (y,x) `elem` lemStruct then x == y else True
  in  [x | x <- lems, all (pred x) lems]

minlem :: Eq a => [(a,a)] -> [(a,a)]
minlem assume = let lems = complement assume
                in  maxElems (lemma_structure assume lems) lems
```

References

[BS81] S. Burris and H. P. Sankappanavar. *A Course in Universal Algebra*. Graduate Texts in Mathematics. Springer-Verlag, 1981.

[FV98] Z. Fülöp and H. Vogler. *Syntax-Directed Semantics*. Monographs on Theoretical Computer Science. Springer-Verlag, 1998.

[GHC] `http://www.haskell.org/ghc/`.

[Now] `http://www.cs.utu.fi/staff/nowotka/minlem.html`.