Technische Universität Berlin

FACHBEREICH 3
MATHEMATIK

Weak-Order Extensions of an Order

by

Karell Bertet[1] Jens Gustedt[2]
Michel Morvan[1]

.2ex $^\circ$ 550/1997

Weak-Order Extensions of an Order*

Karell Bertet[1], Jens Gustedt[2], and Michel Morvan[1]

[1] LIAFA/IBP - Université Denis Diderot Paris 7 - Case 7014 - 2, place Jussieu - 75256 Paris Cedex 05 - France. Email: {bertet,morvan}@litp.ibp.fr

[2] Technische Universität Berlin, Sekr. MA 6-1, D-10623 Berlin, Germany. Email: gustedt@math.tu-berlin.de. Supported by IFP *Digitale Filter*.

Abstract In this paper, at first we describe a graph representing all the weak-order extensions of a partially ordered set and an algorithm generating them. Then we present a graph representing all of the minimal weak-order extensions of a partially ordered set, and implying a generation algorithm. Finally, we prove that the number of weak-order extensions of a partially ordered set is a comparability invariant, whereas the number of minimal weak-order extensions of a partially ordered set is not a comparability invariant.

1 Introduction and Motivations

In this paper, we are interested in the algorithmic and structural study of extensions of a partially ordered set, **orders** for short. The extensions are restricted to a certain class of orders. A lot of previous works deals with studies of restricted extensions classes:

– The linear extensions (extensions which are total orders) of an order are in one-to-one correspondence with the maximal chains of the lattice of the antichains of the order [2].
– The minimal interval extensions of an order are in one-to-one correspondence with the maximal chains of the lattices of the maximal antichains of the order [8].
– The MacNeille completion of an order studied in [3, 9] is an extension of an order belonging to the class of lattices.
– Series-parallel orders are used as extensions of an order to resolve scheduling problems [11].

Among these classes, exist particular extensions which are the extensions of an order obtained by only adding some comparabilities in the order, as the linear extensions or the minimal interval extensions. We are interested in these extensions, especially the weak-order extensions of an order. Informally, a weak-order is an order composed of a set of complete bipartite orders one above an other. Weak-order extensions are suited for the scheduling of tasks [5, 6]: consider a partial order of tasks, a weak-order extension of this order is a scheduling of the

* This work was supported by the PROCOPE Program. To appear in WG'97.

tasks over processes or machine in the time. In this way, Lamport's work on time-stamping in [10] can be seen as on-line computing of a particular weak-order extension of the causal order associated to a distributed execution.

In Sect. 3 we present a one-to-one correspondence between all the weak-order extensions of an order and all the paths from the unique source to the unique sink of a certain graph. This result is related to the similar characterization of linear and minimal interval extensions cited above. We use this characterization to develop an efficient generation algorithm.

Sect. 4 deals with the minimal weak-order extensions of an order. We first present a one-to-one correspondence between all the minimal weak-order extensions of an order and all the paths from any source to any sink of a particular graph. This graph is not a suborder of the above graph since the minimal weak-order extensions of an order are not directly implied from the weak-order extensions as we illustrate with an example. We also present an efficient generation algorithm implied from this graph.

The notion of comparability invariance is fundamental in the study of orders [4, 7, 8]. It is based on the notion of a comparability graph associated to any order. A comparability graph of an order is the undirected graph obtained by deleting the direction on the edges of the order. A parameter of an order is a comparability invariant if it has the same value on any other order having the same comparability graph. Almost all classical parameters on orders are comparability invariants. For example, the number of linear extensions, the dimension, the jump number, the number of the minimal interval extensions are comparability invariants. On the other hand, the number of the interval extensions of an order is not a comparability invariant.

Surprisingly, the inverse statement than for the interval extensions holds for the weak-order extensions: the number of weak-order extensions is a comparability invariant whereas the number of minimal weak-order extensions is not a comparability invariant, as we show in Sect. 5.

2 Definitions and Notations

A *partially ordered set* $P = (X, \leq_P)$ is a reflexive, antisymmetric and transitive binary relation on a se X. Instead of partially ordered set, we often talk about an *order*. We represent an order by a diagram (Hasse diagram) where $x <_P y$ if and only if there is a sequence of connected lines moving upwards from x to y.

Two distinct elements x and y are said to be *comparable* if $x \leq_P y$ or $y \leq_P x$. Otherwise, they are said *incomparable*, denoted by $x \parallel_P y$. We say that y *covers* x, $x \prec_P y$, iff $x <_P y$ and there is no z such that $x <_P z <_P y$.

We define the following sets for P, for an element x of P, and for a subset A

of P:

$$\begin{aligned}
Max(P) &= \{x \in X \mid \text{ for all } y \in X, y \not>_P x\} \\
Min(P) &= \{x \in X \mid \text{ for all } y \in X, x \not<_P y\} \\
Ideal(A) &= \{y \in X \mid y \leq_P x, \text{ for some } x \in A\} \\
Pred(x) &= \{y \in X \mid y <_P x\} \\
Pred(A) &= \bigcup_{x \in A} Pred(x) \\
Succ(x) &= \{y \in X \mid y >_P x\} \\
Succ(A) &= \bigcup_{x \in A} Succ(x)
\end{aligned}$$

A subset A of X is called an *antichain* (resp. *chain*) of P if it contains only pairwise incomparable (resp. comparable) elements. We denote by A_P the set of all antichains of P. A subset A of X is a *maximal antichain* (resp. chain) if it is maximal under inclusion.

$A(P)$ is the order on A_P defined as follows: $A \leq_{A(P)} B$ iff for all x in A, there is y in B such that $x \leq_P y$. It is well known that $A(P)$ equipped with that order is a distributive lattice. By $AM(P)$, we denote the suborder of $A(P)$ restricted to the maximal antichains of P. $AM(P)$ is a lattice, but in general it is not distributive.

The ordering on P is a *weak-order* iff it does not contain the order $2 \bigoplus 1$ as a suborder. Here, $2 \bigoplus 1$ denotes the union of a singleton and a chain composed of two elements. An other characterization of a weak-order P is that $AM(P)$ is a total order such that for all $A \neq B$ in $AM(P)$, $A \cap B = \emptyset$. This allows us to represent a weak-order by a sequence of antichains $A_0, \ldots, A_n$ with $A_i <_{A(P)} A_{i+1}$.

A *directed graph* $G = (X, E)$ is given by a set X of elements or *nodes*, and a subset $E \subseteq X \times X$, the *arcs*. A subset $x_1, \ldots, x_n$ of X such that $(x_i, x_{i+1}) \in E$ for $i < n$ is called a *path* from x_1 to x_n. An node $x \in X$ such that for all $y \in X$ there is no arc (x, y) is called a *sink*. If there is no arc (y, x), x is called a *source*.

3 Weak-Order Extensions of an Order

In this section, we define a graph which represents all weak-orders extensions of an order P. It gives rise to a one-to-one correspondence between certain paths and all the weak-orders extensions of the P. From this graph we define an efficient generation algorithm.

An order $Q = (X, \leq_Q)$ is an *extension* of an order $P = (X, \leq_P)$ if and only if for all x and y in X, $x \leq_P y$ implies $x \leq_Q y$. Then we say that P is a *reduction* of Q. If P is not a weak-order, it clearly admits weak-order extensions.

Definition 3.1. Let $P = (X, \leq_P)$ be an order. We define the directed graph $WE(P) = (A_P, E_{we})$ as follows. For $A \neq B$ two antichains of P, $(A, B) \in E_{we}$ iff the following two conditions are satisfied:

$$A \subseteq Ideal(B) \tag{1}$$
$$B \setminus A = Ideal(B) \setminus Ideal(A). \tag{2}$$

The binary relation induced by E_{we} is an anti-reflexive and antisymmetric relation. So, the reflexo-transitive closure of $WE(P)$ is $A(P)$.

Since $A(P)$ admits a minimal element which is $\emptyset$ and a maximal element which is $Max(P)$, the same holds for $WE(P)$ which admits a unique source and a unique sink.

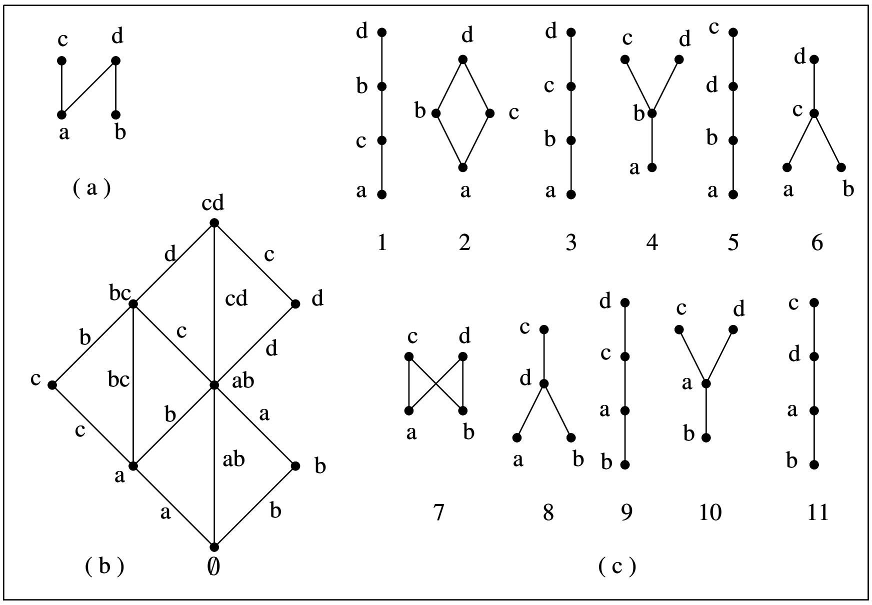

Figure 1. Weak-Order Extensions of an Order

Let P be the order in (a) in Fig. 1. The directed graph $WE(P)$ is given in (b), with arcs labeled with the difference between the two corresponding vertices (direction of the arcs is from bottom to top). The eleven weak-order extensions of P are represented in (c). All these orders are represented by their Hasse diagram. We see that there is a correspondence between the labeled arcs of $WE(P)$ and the weak-order extensions of P, and that $WE(P)$ admits a unique source, and a unique sink. This correspondence is such that $WE(P)$ represents all the weak-orders extensions of P as follows:

Theorem 3.2. *There is a one-to-one correspondence between all the paths of $WE(P)$ from the unique source to the unique sink and all the weak-orders extensions of P.*

For a sketch of a proof let us just describe the mapping. Let $A_0, \ldots, A_n$ be a path of $WE(P)$ from the source to the sink. Then $A_1 \setminus A_0, \ldots, A_i \setminus A_{i-1}, \ldots, A_n \setminus A_{n-1}$ are the maximal antichains of a weak-order extension of P.

The definition of $WE(P)$ can easily be modified as follows: For $A \neq B$ two antichains of P, $(A, B) \in E_{we}$ iff the following two conditions are satisfied:

$$A \subseteq Ideal(B) \tag{3}$$
$$B \setminus A \subseteq Min(P \setminus Ideal(A)) \tag{4}$$

This new definition gives us a way to compute all the weak-orders extensions of an order: Consider that we have $Min(P \setminus Ideal(A_i))$, for a path $A_0, \ldots, A_n$ of $WE(P)$ from the source to the sink. Then we compute $B_{i+1} = A_{i+1} \setminus A_i$ by choosing a subset of this set, and

$$Min\,(P \setminus Ideal\,(A_{i+1})) = Min\,(P \setminus (Ideal\,(A_i) \cup B_{i+1}))\,. \tag{5}$$

At the beginning, $A_0 = \emptyset$ and $Min(P \setminus Ideal(\emptyset)) = Min(P)$. So we may conclude that

$$A_{i+1} = Max\,(Ideal\,(A_i) \cup B_{i+1})\,. \tag{6}$$

Algorithm 1: Weak-Order Extensions of an Order

```
Input: The arrays Succ and Pred for an order P reduced transitively
Output: The weak-orders extensions of P and the number of weak-order
        extensions of P
begin
    let L be an inverse linear extension of P;
    for x in L such that x not visited do
        y = x;
        while |Pred(y)| = 1 and |Succ(Pred(y))| = 1 do
            y = Pred(y);
            EndChain(y) = x;
            mark y visited;
    nbext = Find1(Min(P));
    print "there are" nbext "weak-order extensions";
end
```

Algorithm 1 computes all the weak-orders extensions of an order by using the recursive function *Find1* that is presented in Algorithm 2. It distinguishes the special case that $Min(P \setminus Ideal(A_i))$ contains only one element. This allows to amortize the complexity as it is done in the following theorem. Here m is the number of comparabilities of the transitive reduction of P, n_w is the number of weak-order extensions of P, and Δ is the maximum number of immediate successors of the elements of P.

Theorem 3.3. *Algorithm 1 computes all the weak-order extensions of an order P, and requires $O(m)$ space and $O(n_w\Delta + m)$ time.*

The main idea of the proof is to amortize the work that is done for an individual extension by distributing the cost of a call to *Find1* to the subsequent recursive calls that are issued by this call.

Algorithm 2: The Function *Find1*

```
Input: X a subset
Output: The weak-orders extensions of P\(Ideal(X)∪X) and the number
        of weak-order extensions of P \ (Ideal(X) ∪ X)
begin
    nbext = 0;
    if X = ∅ then
        print "End of a weak-order extension";
        return 1;
    if |X| = 1 then
        if EndChain(X) exist then
            print "X → EndChain(X)";
            X = Succ(EndChain(X));
        else
            print "X";
            X = Succ(X);
    foreach B ⊆ X, B ≠ ∅ do
        print "B";
        X = Min((X \ B) ∪ Succ(B));
        nbext+ = Find1(X);
    return nbext;
end
```

It is also possible to obtain a better time complexity by increasing the space complexity if we explicitly compute $WE(P)$.

Algorithm 1 entirely computes all the weak-order extensions of P, but these extensions have common parts which are computed several times. With the knowledge of parts of these extensions already computed during the execution of the algorithm, we can avoid this. If we consider $WE(P)$, at each step i we can compute the corresponding node A_i of $WE(P)$ which is $Max(Ideal(A_i) \cup B_i)$. If this node already exists in $WE(P)$ then the corresponding part of the path is already computed and vice-versa.

Algorithm 3 describes the function *Find2* which is a modified version of *Find1* that enables us to compute $WE(P)$ in addition. Initially, it is called as $Find2(Min(P), \emptyset)$. Let n' be the number of elements of $WE(P)$, m' be the number of comparabilities of $WE(P)$ transitively reduced, and w be the width of P, that is the maximum size of an antichain of P. Then we have:

Theorem 3.4. *Algorithm 3 computes* $WE(P)$ *and uses a space of* $O(wn' + m')$

Algorithm 3: The Function *Find2*

Input: Y a subset, A an antichain of P
Output: The arrays *Succ* for $WE(P)$
```
begin
  if Y = ∅ then
    return ;
  foreach B ⊆ Y, B ≠ ∅ do
    A' = Max(Ideal(A) ∪ B);
    Succ(A) = Succ(A) ∪ A';
    if A' not visited then
      Y' = Min((Y \ B) ∪ Succ(B));
      Find2(Y', A');
    return;
end
```

and a time of $O(m'w \log n)$. □

Then, to compute all the weak-order extensions of P, we have to visit all the paths of $WE(P)$ from the source to the sink. So, we have:

Corollary 3.5. *By Algorithm 3, it is possible to compute all the weak-order extensions of an order* P *in* $O(m + m'w \log n + n_w)$ *time and* $O(wn' + m')$ *space.*

4 Minimal Weak-Order Extensions of an Order

Now, we characterize the minimal weak-order extensions of an order. Then we present a one-to-one correspondence between all the minimal weak-order extensions of an order and certain paths of a graph and we use this correspondence to develop an efficient generation algorithm.

A weak-order extension Q of P is a *minimal weak-order extension* of P if there is no weak-order extension Q' of P such that Q is an extension of Q'. Informally, a minimal weak-order extension of P is a weak-order extension of P which is as close as possible to P.

The main part of Algorithms 1 and 3 was to choose a subset B of Y, and to delete this subset from Y. In this way, all the weak-orders extensions of an order have been computed. We easily could add conditions to this choice, as e.g the size of the chosen subset. But if we want all the minimal weak-order extensions of P, there are no obvious local conditions that only involve the subset B chosen in Y at each step. The way to define and to compute them is not directly inherited from the general case.

We have the following characterization of a minimal weak-order extension of an order:

Lemma 4.1. *Let $P = (X, \leq_P)$ be an order. Let $Q = A_0, \ldots, A_n$ be a weak-order extension of P. The two following properties are equivalent:*

1. *Q is a minimal weak-order extension of P.*
2. *For all A_i such that $i < n$, there are x in A_i, y in A_{i+1} such that $x \prec_P y$.*

□

In the same way as for the weak-order extensions of P, our goal is now to define a directed graph such that there is a one-to-one correspondence between certain paths of this graph and the minimal weak-order extensions of P. Suppose we choose $WE(P)$ restricted to the paths from the source to the sink such that the corresponding weak-order extension verify Lemma 4.1. Let us demonstrate by Fig. 2 that this graph contains paths that do not correspond to a minimal weak-order extension of P.

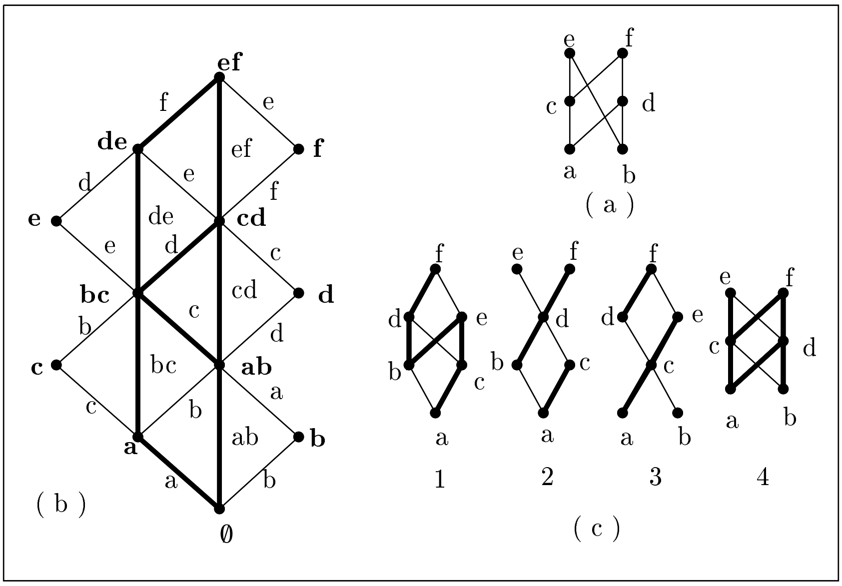

Figure 2. Creation of Wrong Paths

Let P be the order in (a) and $WE(P)$ in (b) of Fig. 2. The minimal weak-order extensions of P are represented in (c), with covering relations as required for Lemma 4.1 in bold. The corresponding paths of $WE(P)$ are given in bold, too. Indeed the subgraph induced by these paths contains 5 paths from $\emptyset$ to ef instead of 4: ab, c, d, ef does not correspond to a minimal weak-order extension of P because

$$\{c, d\} \setminus \{b, c\} \cup \{b, c\} \setminus \{a, b\} = \{c, d\} \tag{7}$$

is an antichain of P.

To avoid this, we use the linegraph transformation: We replace a node of $WE(P)$ belonging to a valid path by an arc, and an arc of $WE(P)$ belonging to a valid path by a node associated with one of the extremities of this arc. Then we differentiate nodes with the same label.

Let us now give a formal definition of this directed graph representing the minimal weak-order extensions of an order:

Definition 4.2. Let $P = (X, \leq_P)$ be an order. We define $WE_m(P) = (\mathcal{X}, \mathcal{E})$, with $\mathcal{X}$ the set of pairs of antichains of P, as follows. For $A_0, \ldots, A_n$ a path of $WE(P)$ from the source to the sink, and $B_i = A_i \setminus A_{i-1}$ with $0 < i < n$ such that $B_i \cup B_{i+1}$ is not an antichain of P include the following objects into WE_m:

$$\begin{array}{ll} (A_i, B_i) \in \mathcal{X} & \text{for all } 0 < i \leq n \\ ((A_i, B_i), (A_{i+1}, B_{i+1})) \in \mathcal{E} & \text{for all } i \text{ with } 0 < i < n \end{array}$$

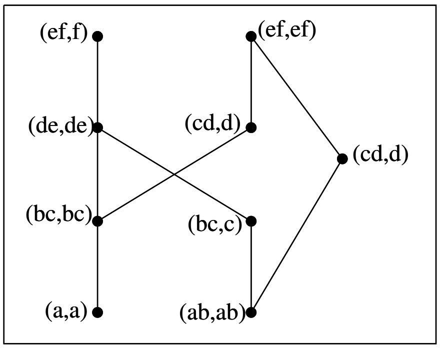

Figure 3. Minimal Weak-Order Extensions of an Order

Let P be the order in (a) of Fig. 2; the directed graph $WE_m(P)$ for P is shown in Fig. 3; the corresponding minimal weak-order extensions are given in (c) of Fig. 2.

This graph represents the minimal weak-order extensions of an order as follows:

Lemma 4.3. *There is a one-to-one mapping between the paths of $WE_m(P)$ from any source to any sink and the minimal weak-orders extensions of P.* □

Now, our goal is to compute all the minimal weak-order extensions of P. Consider the function *Find1* in Algorithm 2 and the function *Find2* in Algorithm 3.

The principal step in these functions is to choose a subset B in Y, Y being the current set of minimal elements. Suppose the last set that was chosen is called B', then computing a minimal weak-order extension of P consists in choosing a subset B in Y with respect to Lemma 4.1 which is equivalent to the two following conditions:

C1: There are $x' \in B'$ and $x \in B$ with $x' <_P x$.
C2: The deletion of B introduces a new minimal element x''.

These conditions are clearly necessary. They are also sufficient since there always is a trivial choice possible in each step: choosing the whole set of minimal elements.

Let $New = Min(Succ(Y))$. Then x'' is in New and we may choose a new subset B of Y such that

$$Z = Min((Y \setminus B) \cup Succ(B)) \cap New \neq \emptyset. \tag{8}$$

Then, to verify condition C1, we have to choose a subset B of Y such that B contains x as above. This can easily be done with a set $Mark$ containing all these x added in Y in the previous recursive call. In this way, we have to choose B such that

$$B \cap Mark \neq \emptyset. \tag{9}$$

So, conditions C1 and C2 are detailed by Eq. 9 and 8, resp.

Every valid pair (B, Z) contains a pair (i, j) where $i \in Mark \cap B$ and $j \in Z$. Then if we assume that $Mark$ and New are totally ordered, we can associate to each such pair (that is each B) the lexicographically minimal such pair (i, j).

Using that fact, it is possible to avoid the generation of invalid subsets of Y and to enumerate all valid B in time asymptotically smaller than n^2 times the number of valid B. A more subtle analysis leads to a better bound but has to be omitted for the sake of brevity. We can state the following result.

Proposition 4.4. *It is possible to compute all the minimal weak-order extensions of an order with an amortized complexity which requires $O(m)$ space and $O(n_{wm}n^2\Delta + m)$ time.* □

In a second approach we compute $WE_m(P)$ in order to avoid computing parts of minimal extensions several times as in Algorithm 3 of Sect. 3, and we compute all the minimal weak-orders extensions with a visit of $WE_m(P)$. So, we have the following complexity, where n'' is the number of elements of $WE_m(P)$ and m'' is the number of comparabilities of the transitive reduction of $WE_m(P)$:

Proposition 4.5. *It is possible to compute all the minimal weak-orders extensions of an order in space $O(wn'' + m'')$ and in time $O(m''w^2\Delta + n_{wm})$.* □

5 Comparability Invariants

Here, we will prove that the number n_w of weak-order extensions and the number n_{wm} of *minimal* weak-order extensions behave quite differently with respect to the property of being a comparability invariant. Indeed, whereas the first is such an invariant, the second is *not*, as will be shown by an example.

The *comparability graph* of an order P is the undirected graph obtained from P –seen as a directed graph– by deleting the direction of the arcs. A parameter of an order is *comparability invariant* if whenever two orders P and Q have isomorphic comparability graphs, the value of the parameter is the same for P and Q.

The *reversed order* $P^d = (X, \leq_{P^d})$ of P is defined by $x \leq_{P^d} y$ iff $y \leq_P x$.

Definition 5.1 (Substitution). Let $P = (X, \leq_P)$ and $M = (Y, \leq_M)$ be two orders such that $X \cap Y = \emptyset$. Let a be in X. $P_a^M = (X \setminus \{a\} \cup Y, \leq_{P_a^M})$, the *substitution* of a by Q, is defined by $x \leq_{P_a^M} y$ iff one of the following cases holds:

$$\begin{aligned} x, y \in X &\ \text{ and }\ x \leq_P y \\ x, y \in Y &\ \text{ and }\ x \leq_M y \\ x \in X, y \in Y &\ \text{ and }\ x \leq_P a \\ y \in X, x \in Y &\ \text{ and }\ a \leq_P y. \end{aligned}$$

Below we will use the following theorem:

Theorem 5.2. *[7] A parameter α of finite orders is a comparability invariant iff for every pair of finite orders P and M, $W = M^d$, and every element a of P,*

$$\alpha\left(P_a^M\right) = \alpha\left(P_a^W\right). \tag{10}$$

Let us first demonstrate that n_{wm} is not a comparability invariant with the counter example represented in Fig. 4. Let P and M be the orders in (a) and (b) respectively and $W = M^d$. Then $WE_m\left(P_X^M\right)$ is represented in (c) and $WE_m\left(P_X^W\right)$ in (d). Clearly, P_X^M admits seven minimal weak-order extensions, and P_X^W eleven.

Theorem 5.3. *n_w is a comparability invariant.*

Proof. We prove this by giving a one-to-one correspondence between the weak-order extensions of P_a^M and the weak-order extensions of P_a^W, for any P, M, $W = M^d$ and $a \in P$.

Let $Q = C_0, \ldots, C_m$ be a weak-order extension of P_a^M. Clearly, the suborder of Q induced by the elements of P is a weak-order extension of $P \setminus \{a\}$, and the suborder of Q induced by the elements of M is a weak-order extension of M. Let $Q_P = B_0, \ldots, B_{n'}$ and $Q_M = A_0, \ldots, A_n$ be these two suborders respectively. We define the mapping $turn_M = Q' = C'_0, \ldots, C'_m$ for the weak-order extensions of P_a^M such that for each $j < m$:

$$C'_j = \begin{cases} (C_j \setminus A_i) \cup A_{n-i} & \text{for some } i \leq n \text{ such that } A_i \subseteq C_j \\ C_j & \text{if there is no such } i. \end{cases} \tag{11}$$

□

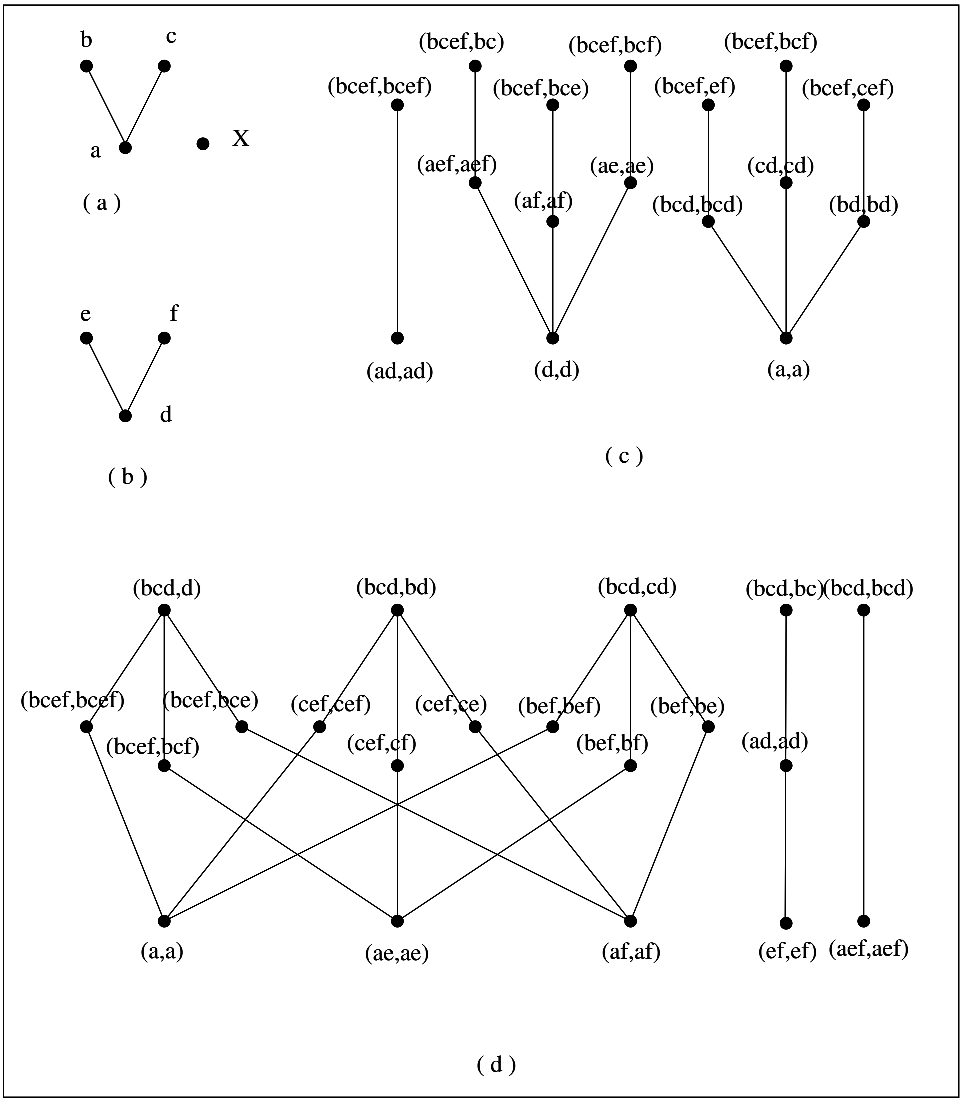

Figure 4. A Counter Example

References

1. Birkhoff, G. *Lattice Theory*, 3rd ed. American Math. Soc., Providence, RI, 1967.
2. Bonnet, R., and Pouzet, M. Extensions et stratifications d'ensembles dispersés. *C. R. Acad. Sci. 268* (1969), 1512–1515.
3. Bordat, J. Calcul pratique du treillis de gallois d'une correspondance. *Math. Sci. Hun. 96* (1986), 31–47.
4. Bouchitté, V., and Habib, M. The calculation of invariants for partially ordered sets. In *Algorithms and Order* (1989), I. Rival, Ed., Kluwer Acad. Publ., Dordrecht, pp. 231–279.
5. Chretienne, P., and Carlier, J. *Problèmes d'Ordonnancement.* Masson, 1988.
6. Conway, R., Maxwell, W., and Miller, L. *Theory of Scheduling.* Addison-Wesley, 1967.
7. Dreesen, B., Poguntke, W., and Winkler, P. Comparability invariance of the fixed point property. *Order 2* (1985), 269–274.
8. Habib, M., Morvan, M., Pouzet, M., and Rampon, J. Extensions intervallaires minimales. *C. R. Acad. Sci. 313* (1991), 893–898.
9. Jourdan, G.-V., Rampon, J.-X., and Jard, C. Computing on-line the lattice of maximal antichains of posets. *Order 11*, 3 (1994), 197–210.
10. Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM 21*, 7 (1978), 558–565.
11. Möhring, R. H., and Radermacher, F. J. The order-theoretic approach to scheduling: The deterministic case. In *Advances in Project Scheduling* (1989), R. Slowinski and J. Weglarz, Eds., Elsevier Science Publishers B.V., Amsterdam, pp. 29–66.

Reports from the group

"Algorithmic Discrete Mathematics"

of the Department of Mathematics, TU Berlin

559/1997 *Matthias Müller–Hannemann and Karsten Weihe:* Improved Approximations for Minimum Cardinality Quadrangulations of Finite Element Meshes

554/1997 *Rolf H. Möhring and Matthias Müller–Hannemann:* Complexity and Modeling Aspects of Mesh Refinement into Quadrilaterals

551/1997 *Hans Bodlaender, Jens Gustedt and Jan Arne Telle:* Linear-Time Register Allocation for a Fixed Number of Registers and no Stack Variables

550/1997 *Karell Bertet, Jens Gustedt and Michel Morvan:* Weak-Order Extensions of an Order

549/1997 *Andreas S. Schulz and Martin Skutella:* Random-Based Scheduling: New Approximations and LP Lower Bounds

542/1996 *Stephan Hartmann:* On the NP–Completeness of Channel and Switchbox Routing Problems

536/1996 *Cynthia A. Phillips, Andreas S. Schulz, David B. Shmoys, Cliff Stein, and Joel Wein:* Improved Bounds on Relaxations of a Parallel Machine Scheduling Problem

535/1996 *Rainer Schrader, Andreas S. Schulz, and Georg Wambach:* Base Polytopes of Series-Parallel Posets: Linear Description and Optimization

533/1996 *Andreas S. Schulz and Martin Skutella:* Scheduling–LPs Bear Probabilities: Randomized Approximations for Min–Sum Criteria

530/1996 *Ulrich H. Kortenkamp, Jürgen Richter-Gebert, Aravamuthan Sarangarajan, and Günter M. Ziegler:* Extremal Properties of 0/1-Polytopes

524/1996 *Elias Dahlhaus, Jens Gustedt and Ross McConnell:* Efficient and Practical Modular Decomposition

523/1996 *Jens Gustedt and Christophe Fiorio:* Memory Management for Union-Find Algorithms

520/1996 *Rolf H. Möhring, Matthias Müller-Hannemann, and Karsten Weihe:* Mesh Refinement via Bidirected Flows: Modeling, Complexity, and Computational Results

519/1996 *Matthias Müller-Hannemann and Karsten Weihe:* Minimum Strictly Convex Quadrangulations of Convex Polygons

517/1996 *Rolf H. Möhring, Markus W. Schäffter, and Andreas S. Schulz:* Scheduling Jobs with Communication Delays: Using Infeasible Solutions for Approximation

516/1996 *Leslie A. Hall, Andreas S. Schulz, David B. Shmoys, and Joel Wein:* Scheduling to Minimize Average Completion Time: Off-line and On-line Approximation Algorithms

515/1996 *Christophe Fiorio and Jens Gustedt:* Volume Segmentation of 3-dimensional Images

514/1996 *Martin Skutella:* Approximation Algorithms for the Discrete Time-Cost Tradeoff Problem

509/1996 *Soumen Chakrabarti, Cynthia A. Phillips, Andreas S. Schulz, David B. Shmoys, Cliff Stein, and Joel Wein:* Improved Scheduling Algorithms for Minsum Criteria

508/1996 *Rudolf Müller and Andreas S. Schulz:* Transitive Packing

506/1996 *Rolf H. Möhring and Markus W. Schäffter:* A Simple Approximation Algorithm for Scheduling Forests with Unit Processing Times and Zero-One Communication Delays

505/1996 *Rolf H. Möhring and Dorothea Wagner:* Combinatorial Topics in VLSI Design: An Annotated Bibliography

504/1996 *Uta Wille:* The Role of Synthetic Geometry in Representational Measurement Theory

502/1996 *Nina Amenta and Günter M.Ziegler:* Deformed Products and Maximal Shadows of Polytopes

500/1996 *Stephan Hartmann and Markus W. Schäffter and Andreas S. Schulz:* Switchbox Routing in VLSI Design: Closing the Complexity Gap

498/1996 *Ewa Malesinska, Alessandro Panconesi:* On the Hardness of Allocating Frequencies for Hybrid Networks

496/1996 *Jörg Rambau:* Triangulations of Cyclic Polytopes and higher Bruhat Orders

Reports may be requested from:

S. Marcus
Fachbereich Mathematik, MA 6–1
TU Berlin
Straße des 17. Juni 136
D-10623 Berlin – Germany

e-mail: Marcus@math.TU-Berlin.DE

Reports are available via anonymous ftp from:

ftp.math.tu-berlin.de
cd pub/Preprints/combi
file Report–<number>–<year>.ps.Z