Science of Computer Programming 9 (1987) 85–89
North-Holland


# SOME ALGORITHMS BASED ON THE DUAL OF DILWORTH'S THEOREM

Anne KALDEWAIJ
*Department of Mathematics and Computing Science, Eindhoven University of Technology, 5600 MB Eindhoven, Netherlands*

Communicated by M. Rem
Received January 1985
Revised October 1986

**Abstract.** In this paper we present a constructive proof of a theorem on minimal decompositions of partially ordered sets. The structure of this proof indicates a strategy for the development of programs that determine such minimal decompositions. The latter is illustrated by a number of examples.

## Introduction

Let $(S, <)$ be a partially ordered set. An increasing sequence of elements of $S$ is called a *chain*. A subset $X$ of $S$ is called an *antichain* if no two elements of $X$ are comparable. In particular, a sequence of one element is both a chain and an antichain.

The number of elements of a chain (antichain) is also cal'ed the length of that chain (antichain). We owe to Dilworth (cf. [1]) the following theorem:

*The minimum number of disjoint chains into which S can be decomposed equals the length of a longest antichain of S.*

If the roles of chains and antichains are interchanged we obtain the dual of Dilworth's theorem (cf. [3]).

*The minimum number of disjoint antichains into which S can be decomposed equals the length of a longest chain of S.*

In Section 1 we present a constructive proof of the dual of Dilworth's theorem. In Section 2 we show some algorithms that are inspired by this proof.

## 1. The theorem

Throughout this paper $(S, <)$ is a finite, partially ordered set.

**Theorem 1.1.** *The minimum number of disjoint antichains into which S can be decomposed equals the length of a longest chain of S.*

**Proof.** Let $k$ be the length of a longest chain of $S$. Since no two elements of a chain

belong to the same antichain, the minimum number of antichains that form a decomposition of $S$ is at least $k$.

For each element of $S$ we define its level $l$, $1 \leq l \leq k$, as the maximum length of a chain of $S$ ending in that element. According to this definition two elements of $S$ with the same level are not comparable. Hence, for each level $l$ the set of elements of level $l$ is an antichain.

Moreover, since each element of $S$ has a level, these antichains form a decomposition of $S$. Hence, the minimum number of antichains that constitute a partition of $S$ is at most $k$. □

From the proof of Theorem 1.1 we obtain a way of decomposing $S$ into a minimum number of antichains. For each element of $S$ its level is computed. The sets consisting of elements of the same level form a decomposition of $S$.

## 2. Applications

**Example 2.1.** We consider an integer sequence $X(i: 0 \leq i < N)$ of $N$, $N \geq 1$, elements. A subsequence of $X$ is obtained by removing zero or more elements from $X$. We derive an algorithm for the computation of a minimal decomposition of $X$ into descending subsequences.

Let $S$ be the set of indices of $X$. Define the partial order $\preccurlyeq$ on $S$ by

$$i \preccurlyeq j \equiv i < j \wedge X(i) < X(j)$$

where $<$ denotes the usual ordering of the integer numbers.

A chain corresponds to an increasing subsequence of $X$ and an antichain corresponds to a descending subsequence of $X$. Application of Theorem 1.1 yields

> The minimum number of descending subsequences that form a decomposition of $X$ equals the length of a longest increasing subsequence of $X$.

From [2] we know an algorithm to compute the maximum length of an increasing subsequence of $X$ in time complexity $O(N \log N)$.

Introducing an auxiliary array $level(j: 0 \leq j < N)$, we extend the algorithm, without affecting the time complexity, such that the relation

$$(\mathbf{A}j: 0 \leq j < N: level(j) = \text{the level of element } j)$$

is established as well. Arry $level(j: 0 \leq j < N)$ then represents a desired decomposition of $X$.

The program contains a repetition for which the following invariants hold:

- $0 \leq n \leq N$,
- $k =$ the length of a longest subsequence of $X(i: 0 \leq i < n)$,
- $(\mathbf{A}j: 1 \leq j \leq k: m(j) =$ the smallest value that occurs as final value of a subsequence of length $j$ of $X(i: 0 \leq i < n))$,
- $(\mathbf{A}j: 0 \leq j < n: level(j) =$ the level of element $j)$.

Notice that the level of element $j$ equals the length of a longest increasing subsequence of $X(i: 0 \leq i \leq j)$ ending in $j$.

The body of the repetition contains a binary search.

```
|[n, k: int;
  m(i: 1 ≤ i ≤ N): array of int;
  level(i: 0 ≤ i < N): array of int;
  n, k := 1, 1
; m: (1) = X(0); level: (0) = 1
; do n ≠ N
  → if X(n) ≥ m(k) → k := k + 1
                   ; m: (k) = X(n)
                   ; level: (n) = k
    □ X(n) < m(1) → m: (1) = X(n)
                   ; level: (n) = 1
    □ m(1) ≤ X(n) < m(k) → |[h, p, q: int;
                            p, q := 1, k
                          ; do p + 1 ≠ q
                            → h := (p + q) div 2
                            ;   if m(h) ≤ X(n) → p := h
                                □ m(h) > X(n) → q := h
                                fi
                            od
                            {m(p) ≤ X(n) < m(p + 1)}
                          ; m: (p + 1) = X(n); level: (n) = p + 1
                          ]|
    fi
  ; n := n + 1
  od
  {(Aj: 0 ≤ j < N: level(j) = the level of element j)}
]|
```

**Example 9.2.** Let $X$ and $S$ be as in Example 2.1. Define partial order $\preceq$ by

$$i \preceq j \equiv i > j \wedge X(i) = X(j).$$

A chain of $S$ corresponds to a constant subsequence of $X$. An antichain is a subsequence in which no two elements have the same value. Such a subsequence is called an $NE$-sequence. Applying Theorem 1.1 yields

The minimum number of $NE$-sequences that form a decomposition of $X$ equals the length of a longest constant subsequence of $X$.

The derivation of a program to determine a minimal decomposition of $X$ into $NE$-sequences is straightforward. As in Example 2.1, it establishes

$$(\mathbf{A}i: 0 \leq i < N: level(i) = \text{the level of element } i).$$

Notice that the level of element $i$ equals the number of elements in $X(j: 0 \leq j \leq i)$ that are equal to $X(i)$. The time complexity of the program is $O(N^2)$.

**Example 2.3.** Let $G$ be an acyclic directed finite graph. Let $S$ be the set of vertices of $G$ with partial order $\leq$ defined by

$$x \leq y \equiv \text{there exists a directed path from } x \text{ to } y.$$

The length of a directed path is the number of vertices of that path, including the first and the last vertex.

A chain is a subsequence of the vertices of a directed path. An antichain is a set of vertices in which no two elements are connected by a directed path. Such a set is called a *non-connect* set. Applying Theorem 1.1 yields

> The minimum number of non-connect sets that form a decomposition of $S$ equals the length of a longest directed path of $G$.

With the tripartition technique described in [4] it is easy to derive a program for the computation of the levels of the vertices. Its time complexity is linear in the number of vertices and arcs of $G$.

**Example 2.4.** Let $A$ be a set of $N$ elements. Let $S$ be the power set of $A$, i.e. the set of all subsets of $A$. Under set inclusion $S$ is a partially ordered set. A chain is a sequence of subsets of $A$ each of which is a proper subset of all its successors. An antichain is a set of subsets of $A$ in which any two elements are incomparable.

Consider a subset $B$ of $A$ with $m$ elements. Starting with the empty set, adding elements of $B$ one by one leads to a chain of $m+1$ subsets of $A$, which is evidently a longest chain ending in $B$. Hence, each subset of $A$ of $m$ elements has level $m+1$. The theorem leads to the following property:

> A minimal partition of the power set of $A$ into parts, each consisting of incomparable subsets of $A$, has $N+1$ elements. The partition obtained by choosing as parts the sets of subsets of equal size is such a minimal partition.

## 3. Conclusions

In the preceding section we showed how the dual of Dilworth's theorem can be applied to a certain class of programming problems. These problems concern computations of minimal decompositions of partially ordered sets.

Such a decomposition is represented by an array in which for each element of the set its level, i.e. the length of a longest chain ending in that element, is recorded.

This method not only yields a straightforward derivation of a program but also leads to efficient algorithms (cf. Example 2.1). The complexity depends on the efficiency with which the level of each element can be computed. This problem can be treated in isolation.

Finally, the *constructive* proof provides a guideline for solving the problems. As far as we know, there is no constructive proof for the original theorem of Dilworth.

## References

[1] R.P. Dilworth, A decomposition theorem for partially ordered sets, *Ann. of Math., (2)* **51** (1950) 161–166.
[2] D. Gries, *The Science of Programming* (Springer, Berlin, 1981) 259–262.
[3] L. Mirsky, A dual of Dilworth's decomposition theorem, *Amer. Math. Month.* **78** (1971) 876–877.
[4] M. Rem, Small programming exercises 5, *Sci. Comput. Programming* **4**(3) (1984) 323–333.