Discrete Applied Mathematics 30 (1991) 265-273
North-Holland

A generalization of the zero-one principle for sorting algorithms*

Dorothea Wagner

Fachbereich Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, W-1000 Berlin 12, FRG

Frank Wagner

Institut für Informatik, Fachbereich Mathematik, Freie Universität Berlin, Arnimallee 2-6, W-1000 Berlin 33, FRG

Received 30 August 1988
Revised 7 April 1989

Abstract

Wagner, D. and F. Wagner, A generalization of the zero-one principle for sorting algorithms, Discrete Applied Mathematics 30 (1991) 265-273.

In this paper a new general approach for the so-called "zero-one principle" for sorting algorithms is described. A theorem from propositional logic that states the connection between two-valued logic and many-valued logic is used to prove this zero-one principle. As a consequence a zero-one principle for a more general class of sorting algorithms is derived.

1. Introduction

The sorting problem is one of the mostly discussed problems in discrete mathematics and computer science. There are lots of more or less "sophisticated" sorting methods, most of them depend on comparisons of pairs of elements. To prove the correctness of such an algorithm, the zero-one principle is a very helpful tool. By applying the zero-one principle, the expense to be made is considerably reduced, i.e., instead of proving the general validity of the sorting algorithm (showing that it works correctly for arbitrary inputs), it is enough to consider only all possible inputs of zeros and ones.

* Part of this research was done while the authors were with the Lehrstuhl für angewandte Mathematik, insbesondere Informatik, RWTH Aachen. D. Wagner was supported by the Deutsche Forschungsgemeinschaft under grant Mö 446/1-1.

The correctness of many classical sorting methods can be proved very elegantly with the zero-one principle [2]. Moreover, also for a lot of new "extensive" sorting methods, for example parallel sorting algorithms [1,3], the zero-one principle turns out to be an invaluable expedient.

By showing, that the zero-one principle relies on a more general principle, we are able to prove that it is also true for sorting algorithms with input restrictions.

Beyond that, our general approach gives rise to the hope that a zero-one principle also exists for other algorithms or problems.

2. The zero-one principle for comparison trees

We consider a set M with a linear order "$<$", i.e., for any three values $x, y, z \in M$ the following conditions are satisfied:

(i) $x<y$ or $y<x$, or $x=y$ is true.
(ii) If $x<y$ and $y<z$, then $x<z$.

Methods, which sort n elements $x_1, \ldots, x_n$ of M, say into nondecreasing order, and depend only on comparisons can be represented in terms of an extended binary tree structure as shown in Fig. 1. Each internal node of the tree corresponds to a comparison of two values x_i versus x_j (write $x_i : x_j$). The left subtree of this node represents the subsequent comparisons to be made if $x_i<x_j$, and the right subtree represents the subsequent comparisons to be made if $x_i>x_j$, while for the case $x_i=x_j$ it does not matter which branch is taken below this node. Each external node of the tree contains a permutation of the input values, say $\pi(x_1), \ldots, \pi(x_n)$ for the case

$$\pi(x_1) \leq \pi(x_2) \leq \cdots \leq \pi(x_n)$$

(where "$\leq$" stands for the case "$<$" or "$=$").

If all input values are distinct, each path through the comparison tree from the

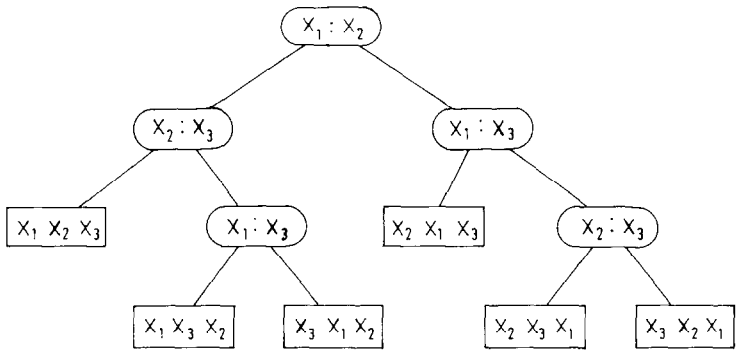

Fig. 1.

root to one external node corresponds to exactly one possible relationship between the input values.

Now, for sorting algorithms, which only depend on comparisons, we have a very helpful tool to prove their correctness [2].

Theorem 1 (Zero-one principle for comparison trees). *A comparison tree sorts every sequence of n elements of a linear ordered set into nondecreasing order if and only if it sorts every sequence of n zeros and ones into nondecreasing order.*

Theorem 1 can be proved by elementary considerations.

As a constrained type of comparison trees we can consider *homogeneous* comparison trees. Homogeneity is satisfied if, whenever we compare x_i versus x_j, the subsequent comparisons for the case $x_i < x_j$ are exactly the same as for $x_i > x_j$, but with i and j interchanged. This type of sorting algorithm is very interesting, because it corresponds to *sorting networks*. A sorting network consists of *comparator modules* which have two inputs and two outputs. The upper output is the minimum of the two inputs, while the lower output is the maximum.

For a sorting network with n inputs the first output (the upper output) is the smallest value of the inputs, the second output is the second smallest etc. A sorting network and the corresponding sorting tree is represented in Fig. 2.

For sorting networks, the zero-one principle is quite helpful, because to prove the correctness of a sorting network with n inputs is not always trivial. By the zero-one principle, it is sufficient to test all 2^n possible inputs of zeros and ones, instead of all possible inputs of n values (which might be infinitely many).

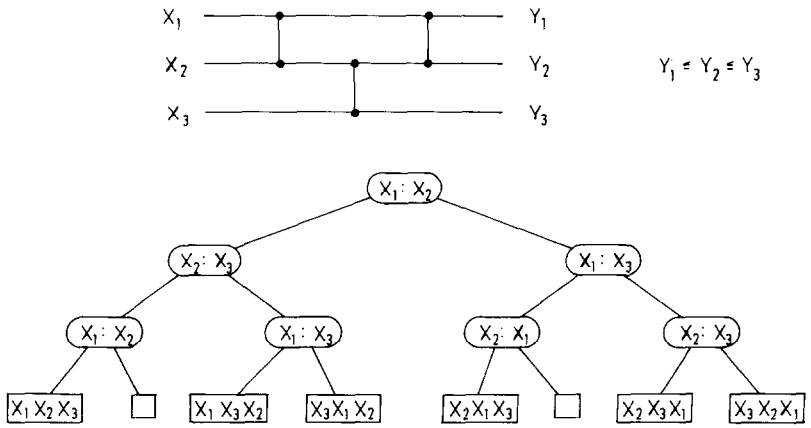

Fig. 2.

3. Fundamental definitions from the classical propositional logic

Theorem 1 can be derived as an application of a theorem about many-valued logic. To state this theorem, we need some fundamental concepts from classical propositional logic.

A *Boolean algebra* is a system $\mathscr{B} = \langle B, \min, \max, -, 0, 1\rangle$, such that min and max are binary operations on B, $-$ is a unary operation on B, $0, 1 \in B$, and the following conditions hold:

(i) min and max are commutative, associative and distributive;
and for $x, y \in B$
(ii) $\min(x, \max(x, y)) = x$, $\max(x, \min(x, y)) = x$,
(iii) $\min(x, -x) = 0$, $\max(x, -x) = 1$.

A nonempty subset I of B is an *ideal* of $\mathscr{B}$ if $\max(x, y) \in I$ whenever $x, y \in I$, and $x \in I$ whenever $\max(-x, y) = 1$ and $y \in I$.

An ideal $I^* \neq B$ of $\mathscr{B}$ is called a *prime ideal* if there is no ideal J such that $I^* \subset J \subset B$.

A fundamental existence theorem for prime ideals is the following.

Boolean prime ideal theorem. *If J is a proper ideal of $\mathscr{B}$, then there is a prime ideal I^* of $\mathscr{B}$ such that $J \subseteq I^*$.*

Let At be a countable set (set of atomic formulas). By *form* we design the set of all formulas generated by $(At, \wedge, \vee, \neg, \rightarrow, t, f)$ where t stands for the true formula and f stands for the false formula.

A *truth valuation* of *form* with respect to a class of Boolean algebras $\mathscr{C}$, is a homomorphism u from *form* to a Boolean algebra in $\mathscr{C}$, with

$$u(\neg \phi) = -u(\phi),$$
$$u(\phi \wedge \psi) = \min(u(\phi), u(\psi)),$$
$$u(\phi \vee \psi) = \max(u(\phi), u(\psi)),$$

and

$$u(\phi \rightarrow \psi) = u(\neg \phi \vee \psi) = \max(-u(\phi), u(\psi))$$

for formulas ϕ and ψ.

For a class of Boolean algebras $\mathscr{C}$, a formula ϕ is a *truth-functional consequence with respect to* $\mathscr{C}$ of a set $\Sigma \subseteq$ *form* if and only if for every truth valuation u with respect to $\mathscr{C}$ we have:

$$\text{if } u(\psi) = 1 \text{ for all } \psi \in \Sigma, \text{ then } u(\phi) = 1 \text{ (write } \Sigma \vDash_{\mathscr{C}} \phi).$$

For further notions from propositional logic we refer to [4].

The following theorem describes the connection between many-valued logic and two-valued logic [6].

Theorem 2. *If $\mathscr{C}$ is the class of all Boolean algebras and $\mathscr{C}'$ contains only the Boolean algebra over $\{0,1\}$, then*

$$\vDash_{\mathscr{C}} = \vDash_{\mathscr{C}'}.$$

Sketch of the proof. If $\Sigma \vDash_c \phi$, then $\Sigma \vDash_{\mathscr{C}'} \phi$ since $\mathscr{C}' \subseteq \mathscr{C}$. Now, let $\Sigma \vDash_{\mathscr{C}'} \phi$, and u be a homomorphism from *form* to a Boolean algebra $\mathscr{B}$, and $u(\psi) = 1$ in $\mathscr{B}$ for all $\psi \in \Sigma$. If $x = u(\phi) \neq 1$, then with the Boolean prime ideal theorem, there exists a prime ideal I^* in $\mathscr{B}$ with $x \in I^*$. We can define a homomorphism h from $\mathscr{B}$ to the Boolean algebra over $\{0,1\}$ by

$$h(z) = \begin{cases} 0, & \text{if } z \in I^*, \\ 1, & \text{if } z \notin I^*. \end{cases}$$

Then $h \circ u$ is a truth valuation from *form* to $\{0,1\}$, with $h(u(\phi)) = 0$ and $h(u(\psi)) = 1$ for all $\psi \in \Sigma$, contradicting $\Sigma \vDash_{\mathscr{C}'} \phi$.

(For details, we refer to [7].) □

4. Main result

We use Theorem 2 to prove the zero-one principle for comparison trees in the following sense. For an "appropriate" Boolean algebra $\mathscr{B}$, and an "appropriate" set of formulas Σ, and an "appropriate" formula ϕ we have:

If for every truth valuation u with respect to $\{0,1\}$ follows

$$(\text{if } u(\psi) = 1 \text{ for all } \psi \in \Sigma, \text{ then } u(\phi) = 1)$$

then for every truth valuation u' with respect to $\mathscr{B}$ follows

$$(\text{if } u'(\psi) = 1 \text{ for all } \psi \in \Sigma, \text{ then } u'(\phi) = 1).$$

In other words, for a set of elements to be sorted, we define a Boolean algebra such that all truth valuations with respect to this Boolean algebra express all possible inputs of a sorting algorithm. For a sorting algorithm that corresponds to a comparison tree, we define a formula that expresses all possible paths from the root to an external node in the comparison tree.

If there are certain input restrictions for the underlying sorting algorithm, we define a set of formulas Σ that expresses exactly these input restrictions.

Then, the zero-one principle for sorting algorithms follows from Theorem 2.

Definition of the "appropriate" set of formulas Σ. We postpone the consideration of sorting algorithms which accept only inputs satisfying certain restrictions, to Section 5. So, as the set of formulas corresponding to the input restrictions, we take the set only containing the true formula, $\Sigma := \{t\}$. In this case Theorem 2 says: A formula is a tautology with respect to the Boolean algebra over $\{0,1\}$ if and only if it is a tautology with respect to any Boolean algebra.

Definition of the "appropriate" Boolean algebra. Let M be a linear ordered set. Then $\mathscr{B}=(\mathscr{P}(M), \cap, \cup, \emptyset, M, {}^{c})$ is a Boolean algebra over the powerset of M, with $\emptyset$ being the zero and M the one in $\mathscr{B}$, and A^{c} stands for the complement of $A \in \mathscr{P}(M)$ with respect to M.

To prove the nontrivial part of the zero-one principle, we must define a formula $\varphi_{\mathscr{A}}$ corresponding to a sorting algorithm $\mathscr{A}$ such that: If $\mathscr{A}$ sorts every input of zeros and ones, then $\varphi_{\mathscr{A}}$ is a tautology with respect to the Boolean algebra over $\{0,1\}$, and if $\varphi_{\mathscr{A}}$ is a tautology with respect to $\mathscr{B}$, then $\mathscr{A}$ sorts every input of elements from the linear ordered set M.

Definition of the "appropriate" formula. Let $\mathscr{A}$ be a sorting algorithm that can be represented as a comparison tree. Consider for example the comparison tree in Fig. 1. For the case: $x_1 \leq x_2$ and $x_2 \leq x_3$, the output of the comparison tree is x_1, x_2, x_3, corresponding to $x_1 \leq x_2 \leq x_3$. Accordingly, the algorithm $\mathscr{A}$ can be expressed as follows:

If $x_1 \leq x_2$ and $x_2 \leq x_3$, then $x_1 \leq x_2 \leq x_3$, and

if $x_1 \leq x_2$ and $x_3 \leq x_2$ and $x_1 \leq x_3$, then $x_1 \leq x_3 \leq x_2$, and

if $x_1 \leq x_2$ and $x_3 \leq x_2$ and $x_3 \leq x_1$, then $x_3 \leq x_1 \leq x_2$, and

if $x_2 \leq x_1$ and $x_1 \leq x_3$, then $x_2 \leq x_1 \leq x_3$, and

if $x_2 \leq x_1$ and $x_3 \leq x_1$ and $x_2 \leq x_3$, then $x_2 \leq x_3 \leq x_1$, and

if $x_2 \leq x_1$ and $x_3 \leq x_1$ and $x_3 \leq x_2$, then $x_3 \leq x_2 \leq x_1$.

Now, as the corresponding subformulas for the leftmost path in the tree we take:

$$((\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)) \rightarrow ((\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3))$$

In the same way, for the whole comparison tree in Fig. 1, we get the formula:

$$\begin{aligned}
&(((\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)) \rightarrow ((\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3))) \\
&\wedge (((\neg x_1 \vee x_2) \wedge (\neg x_3 \vee x_2) \wedge (\neg x_1 \vee x_3)) \rightarrow ((\neg x_1 \vee x_3) \wedge (\neg x_3 \vee x_2))) \\
&\wedge (((\neg x_1 \vee x_2) \wedge (\neg x_3 \vee x_2) \wedge (\neg x_3 \vee x_1)) \rightarrow ((\neg x_3 \vee x_1) \wedge (\neg x_1 \vee x_2))) \\
&\wedge (((\neg x_2 \vee x_1) \wedge (\neg x_1 \vee x_3)) \rightarrow ((\neg x_2 \vee x_1) \wedge (\neg x_1 \vee x_3))) \\
&\wedge (((\neg x_2 \vee x_1) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_2 \vee x_3)) \rightarrow ((\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_1))) \\
&\wedge (((\neg x_2 \vee x_1) \wedge (\neg x_3 \vee x_1) \wedge (\neg x_3 \vee x_2)) \rightarrow ((\neg x_3 \vee x_2) \wedge (\neg x_2 \vee x_1)))
\end{aligned}$$

In general, for each possible path from the root to an external node of a comparison tree we choose a subformula, where the left branch after a comparison $x_i : x_j$ corresponds to $\neg x_i \vee x_j$, and the right branch to $\neg x_j \vee x_i$, and a sequence of comparisons corresponds to the conjunction of these terms. A result $\pi(x_1), \ldots, \pi(x_n)$ in an external node corresponds to the subformula

$$(\neg \pi(x_1) \vee \pi(x_2)) \wedge (\neg \pi(x_2) \vee \pi(x_3)) \wedge \cdots \wedge (\neg \pi(x_{n-1}) \vee \pi(x_n)).$$

The subformula corresponding to a path from the comparison tree to an external node, and the subformula corresponding to the content of this external node are connected by $\rightarrow$. Finally, the conjunction of all these subformulas corresponds to all possible paths through the comparison tree (which also determine the comparison tree).

To derive the zero-one principle from Theorem 2 let us first consider a sorting algorithm $\mathscr{A}$ applied to inputs of zeros and ones.

Theorem 3. *Let $\mathscr{A}$ be a sorting algorithm that sorts every input of zeros and ones, then the corresponding formula $\varphi_{\mathscr{A}}$ is a tautology with respect to the Boolean algebra over $\{0,1\}$.*

Proof. Any truth valuation u with respect to the Boolean algebra $\{0,1\}$ assigns to a variable x_i in $\varphi_{\mathscr{A}}$ 0 or 1. It is easy to see, that $x_i \leq x_j$ holds for a certain input of zeros and ones if and only if $u(\neg x_i \vee x_j) = 1$ for the corresponding truth valuation u with respect to the Boolean algebra over $\{0,1\}$. Thus, if $\mathscr{A}$ sorts any input of zeros and ones, $\varphi_{\mathscr{A}}$ is a tautology with respect to the Boolean algebra over $\{0,1\}$. □

Now, let $\mathscr{A}$ be a sorting algorithm applied to arbitrary inputs of elements from a linear ordered set M.

Theorem 4. *Let $\mathscr{A}$ be a sorting algorithm. If the corresponding formula $\varphi_{\mathscr{A}}$ is a tautology with respect to the Boolean algebra $\mathscr{B}$ over the powerset of M, then $\mathscr{A}$ sorts every input of elements from M.*

Proof. For arbitrary subsets X or Y of M we have: $X \cap Y = M$ iff $X = M$ and $Y = M$. Furthermore, if $X \rightarrow Y = M$ then (if $X = M$ then $Y = M$). For $a \in M$ let $A := \{b \in M : b \leq a\}$. Then for all $a_i, a_j \in M$: If $A_i^C \cup A_j = M$, then $a_i \leq a_j$. Since $\varphi_{\mathscr{A}}$ is a tautology we know that especially $\varphi_{\mathscr{A}}(A_1, \ldots, A_n) = M$ which using the facts above proves that $\mathscr{A}$ sorts $a_1, \ldots, a_n$. □

According to Theorem 3 and Theorem 4 the zero-one principle follows from Theorem 2.

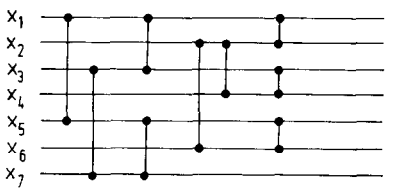

Fig. 3. Bitonic sorter of order 7.

5. Application of the zero-one principle to sorting algorithms with input restrictions

In practice, the input of a sorting algorithm often satisfies certain conditions. For example, the input consisting of two (or more) sequences of elements, which are already sorted. In these cases, it is convenient to use less powerful sorting algorithms that utilize these properties of the input, for example merge two ordered sequences into one. Algorithms that realize this are for example Batcher's odd-even merge, two-way merge or k-way merge, respectively [2]. Another algorithm, that merges an input satisfying certain restrictions to one sorted sequence is a bitonic sorter. A bitonic sorter of order p sorts any bitonic sequence of length p into nondecreasing order. (A sequence $\langle x_1, \ldots, x_p \rangle$ is *bitonic* if $x_1 \geq x_2 \geq \cdots \geq x_k \leq x_{k+1} \leq \cdots \leq x_p$ for some $k \in \{1, \ldots, p\}$.) See for example Fig. 3.

For these special types of sorting algorithms we construct a set of formulas Σ which corresponds to the input restrictions of the algorithm. Consider for example a bitonic sorter of order p, then for the input $\langle x_1, \ldots, x_p \rangle$ with $x_1 \geq x_2 \geq \cdots \geq x_k \leq x_{k+1} \leq \cdots \leq x_p$, we have

$$(\neg x_2 \vee x_1) \wedge (\neg x_3 \vee x_2) \wedge \cdots \wedge (\neg x_k \vee x_{k-1}) \wedge (\neg x_k \vee x_{k+1})$$
$$\wedge (\neg x_{k+1} \vee x_{k+2}) \wedge \cdots \wedge (\neg x_{p-1} \vee x_p).$$

The single element of Σ is then the disjunction of all analogous formulas for the possible choices of intermediate points k.

For a bitonic sorter of order p we can define a formula ϕ as in Section 4, that corresponds to all possible sequences of comparisons to be made. Let M be the linear ordered set containing all elements to be sorted, $\mathcal{B}$ the corresponding Boolean algebra. Then $\Sigma \vDash_{\mathcal{B}} \phi$ if and only if $\Sigma \vDash_{\{0,1\}} \phi$, which implies the zero-one principle for a bitonic sorter.

In general, consider a sorting algorithm $\mathcal{A}$ depending only on comparisons that works on inputs from a linear ordered set M which satisfy certain input restrictions, say $r(\mathcal{A})$. Then the zero-one principle holds for $\mathcal{A}$ if there exists a set of formulas Σ_p (corresponding to the input restrictions) such that condition i) and ii) hold.

i) If for a truth valuation u with respect to the Boolean algebra over $\{0,1\}$ we have $u(\psi) = 1$ for all $\psi \in \Sigma_p$, then the corresponding input of zeros and ones satisfies $r(\mathcal{A})$.

ii) If an input of elements from M satisfies $r(\mathcal{A})$, then for a certain truth valuation u with respect to the Boolean algebra $\mathcal{B}$ over the powerset of M $u(\psi) = M$ for all $\psi \in \Sigma_p$.

The zero-one principle for sorting algorithms with input restrictions can be very useful, for example to prove the correctness of sorting algorithms for parallel models of computation [1]. Often parallel sorting algorithms, for example for perfect shuffle computer [8] or mesh connected computer [5,9], run in different phases, where each phase realizes a sorting algorithm with input restrictions. Thus,

the correctness of the algorithm can be proved by proving the correctness of the different phases of the algorithm using the zero-one principle (which often is much easier than other correctness proofs.)

References

[1] S.G. Akl, Parallel Sorting Algorithms (Academic Press, New York, 1985).
[2] D.E. Knuth, The Art of Computer Programming, Vol. 3: Sorting and Searching (Addison-Wesley, Reading, MA, 1973).
[3] M. Kunde, A general approach to sorting on 3-dimensionally mesh-connected arrays, in: Lecture Notes in Computer Science 237 (Springer, Berlin, 1986) 329–338.
[4] J.D. Monk, Mathematical Logic (Springer, Berlin, 1976).
[5] D. Nassimi and S. Sahni, Bitonic sort on a mesh-connected parallel computer, IEEE Trans. Comput. 28 (1) (1979) 2–7.
[6] N. Rescher, Many-Valued Logic (McGraw-Hill, New York, 1969).
[7] M.M. Richter, Logikkalküle, Teubner Studienbücher (Teubner, Leipzig, 1978).
[8] H.S. Stone, Parallel processing with the perfect shuffle, IEEE Trans. Comput. 20 (2) (1971) 153–161.
[9] C.D. Thompson and H.T. Kung, Sorting on a mesh-connected parallel computer, Comm. ACM 20 (4) (1979) 263–271.