IV. Other Structured Models

Objekttyp: **Chapter**

Zeitschrift: **L'Enseignement Mathématique**

Band (Jahr): **28 (1982)**

Heft 1-2: **L'ENSEIGNEMENT MATHÉMATIQUE**

PDF erstellt am: **20.09.2024**

game has been studied extensively and is the basis for a number of results that have "TIME-SPACE tradeoffs" as part of a title. If one wishes to conserve on the number of intermediate locations (the number of pebbles) then it may be necessary to often recompute results (i.e. repebble the same node of the circuit). Tompa [78] uses the connectivity properties of the FFT problem to demonstrate a Space (number of pebbles) · Time (number of pebble moves) $= \Omega(n^2)$ lower bound. I find it interesting that, independently, Grigoryev [76] produces a similar Time-Space $= \Omega(n^2)$ tradeoff for multiplication in $Z_2[x]$ (which extends to integer multiplication) with respect to Boolean circuits (i.e. general setting) by using arguments about the range of subfunctions.

My interest stems from the fact that the same duality (between connectivity and subfunctions) again provides the basis for two results concerning VLSI design; namely, using similar models, Thompson [79] shows the product of Area (= length of wire) and Parallel Time^2 $= \Omega(n^2)$ for the FFT (structured setting) while Brent and Kung [79] show Area · Parallel Time^2 $= \Omega(n^2)$ for integer multiplication (general setting). Recently, Brent and Goldschlager have established an analogous Area · Parallel Time tradeoff result for a set recognition problem.

IV. Other Structured Models

I should use this last section to briefly indicate that many other structured models can be found in a variety of problem areas. Yet, these models are often more appropriate to particular problems rather than for a large class of problems. Hence, the real purpose of this section is to indicate a need for structured models natural to important problem areas.

Perhaps I should constrain this concluding discussion to an obvious candidate, a "model for graph-theoretic problems". But given the scope of graph theory, this seems far too ambitious. What has been done thus far? We have already discussed the use of linear comparison trees and branching programs for the study of shortest path problems. This model does seem to abstract the underlying tests and operations needed for such problems while suppressing any data structures needed for both searching and representation. The comparison tree becomes a rather uninteresting general model if we study unlabelled graph problems; since any such problem can be "solved" by looking at each entry of the input adjacency matrix. The solutions (Kirkpatrick [74], Rivest and Vuillemin [76]) to the Rosenberg-Aanderaa conjecture show that most graph problems require every entry of the adjacency matrix be probed. (Obviously, there are other ways to represent a graph. But unlabelled graph problems do again become non-trivial if we consider time-space considerations with regard to the generalized version of branching programs).

There is a class of structured models that have been developed by Savitch [73], and Cook and Rackoff [80] for studying the space complexity of the directed and undirected versions of the graph reachability (alias transitive closure) problem. The models reflect the kinds of path traversal strategies one often uses in graph theoretic algorithms. Similar models have been employed for the problem of searching mazes (see Blum and Kozen [78]). Essentially, the JAG model of Cook and Rackoff consists of a finite control of say q states plus a set of p labelled markers. The Space charge is $\log q + p \log n$, the latter term reflecting the fact that a marker is used to remember a node in the graph. The graph is oriented (i.e. edges leaving a node are numbered) and the model traverses a graph by moving markers along edges or to be coincident with other markers. Moves are determined by the state and by the presence of markers. If the model does not allow backward traversal of edges, then Cook and Rackoff can demonstrate an $\Omega(\log^2 n/\log\log n)$ Space lower bound in the directed case. For the undirected case, the result of Aleliunas *et al.* [79] shows that the reachability problem can be solved in $O(\log n)$ Space by a Monte Carlo algorithm, and hence in non-uniform $O(\log n)$ Space. Indeed, a JAG with only 2 markers and a polynomial number of states can solve the undirected reachability problem.

If the JAG model does allow backward edge traversal, then the model becomes "general for the issue of NSPACE $(\log n) \neq$ DSPACE $(\log n)$". The Aleliunas *et al.* result is based on a universal covering sequence (for all n node cubic graphs) of polynomial (n) length; it is this covering sequence which leads to a "representative set of inputs" on which a JAG with backward edge capability can simulate a general algorithm for the directed reachability problem (which is log space complete for NSPACE $(\log n)$).

Recently there has been a set of interesting results (see Bland and Las Vergnas [78], and Lovász [79]) concerning matroid properties where the model is essentially an oracle for determining the *independence* of a set of elements. Since matroid properties (and algorithms, like the Greedy algorithms) are often viewed as generalizations of graph theoretic properties, one might view the independence oracle as a structured model for graph theory. As such, this is a different kind of structured model than I have been trying to sell in this paper. The results here proceed by an adversary argument which constructs two similar looking matroids to force an exponential number of oracle calls in order to determine a certain property. But the matroids being constructed need not be, and are not, constructed from the same "domain" (e.g. graphs, with cycle free paths as independent sets). It is rather like relativized complexity theory (see Baker, *et al.* [75]) or the construction of non standard models in logic, which gives insight into what kind of arguments will not work. However, I am trying to emphasize structured models where the domain is "standard" and the structure issues hinge on the accessing and processing of such domain elements.

Given the significant progress in the field of graph theoretic algorithms, it is relatively disappointing how few structured models have been proposed for this area. In particular, we seem to have adopted the model of algebraic complexity to study graph theoretic variants of P vs NP, and to study lower bounds for graph theoretic parallel computation (see Reghbati and Corneil [78] for some upper bounds in this context). To be fair, we sometimes adopt graph theory as a means to proving lower bounds in algebraic complexity (see Valiant [77]).

Clearly, I have not nearly exhausted the variety of computational problems whose complexity has been studied from both the general and structured viewpoints. But I hope I have begun to defend my earlier conclusions that the general theory provides a standard for assessing complexity results about structured models and conversely that the structured setting gives insight for the general theory.

Note added at the end of Symposium: M. Rabin observed to me that Khachian's polynomial time solution for the Linear Programming problem provides a dramatic example of the structured vs general distinction. The present polynomial bound is based on the precision of the inputs and not just the number of inputs. Similar remarks can be made about the Transportation Problem and the Edmonds-Karp solution.

Acknowledgement. S. Cook made a number of valuable insights and suggestions to me about this topic. His ideas permeate this paper. I also want to thank, but not hold responsible, R. Aleliunas, P. Dymond, N. Pippenger, C. Rackoff, A. Rosenberg and J. von zur Gathen for their suggestions and corrections. Finally, I want to thank E. Engeler, E. Specker and V. Strassen for many enjoyable conversations about this and other topics during my stay at the E.T.H. (1976).