III. Arithmetic Models — Algebraic Complexity

$O(n^{1+1/k})$. (Pippenger [personal communication] has recently shown that a similar upper bound can also be achieved for the sorting problem in the context of ordering networks.) This same quantative behaviour has a corresponding lower bound for the (structured) simulation of linear recursion schemes by flow-chart schemes that was referred to in Section I.

III. Arithmetic Models — Algebraic Complexity

I would now like to turn attention to the complexity of arithmetic problems, and to the straight line or circuit model with operations $+$, $-$, $\times$ (and perhaps $\div$). Fortunately, I need not pursue this topic in too much detail since Valiant [80] in this conference will be addressing just this topic. Indeed, Valiant [79*a*] and [79*b*] has always provided compelling evidence for the importance of the interrelation between a structured problem setting (algebraic complexity) and the general theory. The correspondence between algebraically structured arithmetic circuits computing (say) formal polynomials in $F[x_1, ..., x_n]$ and general Boolean circuits computing Boolean functions of n variables is readily apparent. In the former, gates represent the ring operations $(\times, +, -)$ and the inputs are the (indeterminates) $\{x_i\} \cup F$, while in the latter, the gates represent some basis set of Boolean operations (say $\wedge$, $\vee$, $\neg$) and the inputs are the (Boolean variables) $\{x_i\} \cup \{0 \text{ or false}, 1 \text{ or true}\}$. Since $\vee$, $\wedge$, $\neg$ can be easily simulated by $+$, $-$, $\times$ when restricted to $\{0, 1\{$ (e.g. $x \vee y$ by $x + y - x \times y$), positive results for the arithmetic case often carry over immediately to the Boolean setting. The usual measures of complexity are SIZE (= number of gates = sequential Time complexity), DEPTH (= length of longest path in the circuit = parallel time complexity) and FORMULA SIZE (= number of gates in a circuit having fan-out one; i.e. a formula). One of the first (pair of) results that demonstrated to me the importance of keeping this correspondence in mind, is the relating of the FORMULA SIZE and DEPTH measures. Independently, Spira [71] (for the Boolean case over any basis) and Brent [74] showed how to convert any formula of size m to an equivalent formula (and hence circuit) of depth $O(\log m)$; the converse that any circuit of depth d can be converted to a formula of size 2^d is immediate. It is interesting to note that the Spira result seems to depend intrinsically on the Boolean domain, whereas the Brent result is proven in a more abstract setting using only that $\times$ (resp. $\wedge$) distributes over $+$ (resp. $\vee$). Then, here again, is a situation where a result need not

yield a direct corollary (the simulation of $x \vee y$ by $x + y - x \times y$ requires fan-out two) yet the proof technique can be applied. In this regard, it is interesting to note that whereas Pippenger [74] shows every *symmetric* Boolean function has polynomial Formula Size (i.e. can be computed in $O(\log n)$ Depth), the corresponding result is not known for the arithmetic elementary symmetric functions ($O(\log^2 n)$ Depth is easy to establish via polynomial multiplication).

While the relationship between FORMULA SIZE and DEPTH is relatively well understood, the relation between Size and Depth is a more fundamental issue (see conjectures GC4 and GC6). The relating of TIME to SIZE and SPACE to DEPTH in the general setting (for example, see Borodin [77]) also shows the relationship between conjectures GC3 and GC5 but the classes Polytimelogspace and Polytimelogdepth (in conjectures GC4 and GC6) may be quite different (see Cook [80]). In the algebraic setting, these general relationships take on added interest when combined with some important results concerning Depth. First, Csanky proved that a number of central problems (including A^{-1}, A^n) have the same depth complexity as computing det (A), and, more important, these problems can be computed in $O(\log^2 n)$ depth and simultaneously in $O(n^4)$ size. Hyafil [79] took Csanky's result further in showing that any set of multivariate polynomials of degree $\leqslant d$ and computable in Size $\leqslant t$, can be computed in Depth $= O(\log d \cdot \log t)$. Clearly $\Omega(\log d)$ is a lower bound on depth, so that Hyafil's result is a major challenge to conjecture GC4 in the algebraic (i.e. structured) setting. It is the concept of degree which gives us an opportunity to relate Size and Depth in the algebraic setting. The concept of degree (in the sense of algebraic geometry) also gave Strassen [73] the means to establish a non trivial $\Omega(n \log n)$ lower bound on Size. At present, we do not have a meaningful analogy to *degree* in the Boolean setting, and hence all the barriers and conjectures remain intact.

Hyafil's [79] result leaves open the analogue of conjecture GC6. The construction which converts from deg d, Size t to Depth $O(\log d \cdot \log t)$ results in a Size of $t^{O(\log d)}$; that is, it does not preserve polynomial size. This contrasts with Csanky's [76] result that our concrete examples (det, A^n, A^{-1}) are in Polytimelogdepth. It also remains open as to whether or not these concrete problems (or perhaps all small degree polynomials computable in polynomial Size) can be computed in smaller Depth (e.g. $O(\log n)$). In this regard, one can note the similarity between the arithmetic A^n, A^{-1} problems and the Boolean Depth requirements for integer powering and division (see Cook [80]). We also note the similarity between the arith-

metic A^n and the Boolean A^* (transitive closure). The latter problem is complete for the issue of NSPACE(S) vs DSPACE(S). A (uniform) positive result for A^n (say $O(\log^\alpha n)$ Depth with $\alpha < 2$) would directly improve Savitch's [70] deterministic simulation of nondeterministic space bounded computations. It appears to me then that a first attempt to break barriers GB2, 3 would be to try to establish a nonlogarithmic lower bound for the depth of A^n (equivalently, the det). The only known lower bound, Shamir and Snir [77], is that A^n does require depth $\Omega(\log^2 n)$ for monotone $(+, \times)$ circuits. However, Valiant [79*c*] has shown that monotone circuits can be *exponentially* inefficient.

To me, the most compelling evidence of the importance of the algebraic viewpoint for the general theory is Valiant's [79*a*] result that the *permanent* (say when restricted to integer matrices) is complete for the class $(\#P)$ of problems associated with counting the number of solutions of problems computable in nondeterministic polynomial time (NP). The difference between the determinant and permanent is thus made quite explicit in complexity terms (even though we can't yet translate this into provable lower bounds). Recently, Valiant [79*b*] uncovers the central role that the determinant and permanent problems play within algebraic complexity itself (with the result of further insights into the general theory). Valiant is directly motivated (see his introductory paragraph) by the kind of completeness results one obtains in the general theory. But rather than use complexity based reductions (as is usually done) he is able to base his reductions on purely algebraic properties. Again, following Schnorr's [76] original beliefs in this regard, I would also argue that the algebraic setting is a reasonable framework for attacking what many feel to be the fundamental issue of complexity (P vs NP), even though (or maybe because) lower bounds in the sense of algebraic complexity do not seem to have any direct corollaries to the general theory.

I want to conclude this section by considering Space complexity, and, more precisely, Time-Space tradeoffs for the algebraic setting. Space is not usually considered within algebraic complexity but I think it is quite relevant to our thesis. The Space measure here evolves naturally from the attempt to execute a circuit as a straight-line program; namely, it is the number of intermediate locations needed to store partial results. This measure has been well studied in the guise of a certain *pebble game* introduced in the schematology paper of Paterson and Hewitt [70] and used in other structured settings (e.g. Cook [73] and Cook and Sethi [76] where conjecture GC3 is formulated in a structured setting). Indeed, the pebble

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