Parallel Biomolecular Computation:
Models and Simulations

J. H. Reif

Abstract. This paper is concerned with the development of techniques for massively parallel computation at the molecular scale, which we refer to as molecular parallelism. While this may at first appear to be purely science fiction, Adleman [Ad1] has already employed molecular parallelism in the solution of the Hamiltonian path problem, and successfully tested his techniques in a lab experiment on DNA for a small graph. Lipton [L] showed that finding the satisfying inputs to a Boolean expression of size $n$ can be done in $O(n)$ lab steps using DNA of length $O(n \log n)$ base pairs. This recent work by Adleman and Lipton in molecular parallelism considered only the solution of NP search problems, and provided no way of quickly executing lengthy computations by purely molecular means; the number of lab steps depended linearly on the size of the simulated expression. See [Re3] for further recent work on molecular parallelism and see [Re4] for an extensive survey of molecular parallelism.

Our goal is to execute lengthy computations quickly by the use of molecular parallelism. We wish to execute these biomolecular computations using short DNA strands by more or less conventional biotechnology engineering techniques within a small number of lab steps. This paper describes techniques for achieving this goal, in the context of well defined abstract models of biomolecular computation. Although our results are of theoretical consequence only, due to the large amount of molecular parallelism (i.e., large test tube volume) required, we believe that our theoretical models and results may be a basis for more practical later work, just as was done in the area of parallel computing.

We propose two abstract models of biomolecular computation. The first, the Parallel Associative Memory (PAM) model, is a very high-level model which includes a Parallel Associative Matching (PA-Match) operation, that appears to improve the power of molecular parallelism beyond the operations previously considered by Lipton [L]. We give some simulations of conventional sequential and parallel computational models by our PAM model. Each of the simulations use strings of length $O(s)$ over an alphabet of size $O(s)$ (which correspond to DNA of length $O(s \log s)$ base pairs). Using $O(s \log s)$ PAM operations that are not PA-Match (or $O(s^2)$ operations assuming a ligation operation) and $t$ PA-Match operations, we can:

1. simulate a nondeterministic Turing Machine computation with space bound $s$ and time bound $2^{O(s)}$, with $t = O(s)$,
2. simulate a CREW PRAM with time bound $D$, with $M$ memory cells, and processor bound $P$, where here $s = O(\log (PM))$ and $t = O(D + s)$,
3. find the satisfying inputs to a Boolean circuit constructible in $s$ space with $n$ inputs, unbounded fan-out, and depth $D$, where here $t = O(D + s)$.

1 A preliminary version of this paper appears in Proceedings of the Seventh Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA95), Santa Barbara, CA, June 1995, pages 213–223. This research was supported by NSF/DARPA CCR-9725021, CCR-96-33567, NSFIRI-9619647, ARO Contract DAAH-04-96-1-0448, ARPA/SISTO Contracts N00014-91-J-1985, and N00014-92-C-0182 under Subcontract KI-92-01-0182. This paper, in postscript, can be found at http://www.cs.duke.edu/~reif/paper/Molecular.ps and figures can be found at http://www.cs.duke.edu/~reif/paper/mole.fig.ps.

2 Department of Computer Science, Duke University, Durham, NC 27708-0129, USA. reif@cs.duke.edu.

Received December 30, 1995; revised December 30, 1996, and January 22, 1998. Communicated by D. Gusfield and M.-Y. Kao.
We also propose a Recombinant DNA (RDNA) model which is a low-level model that allows operations that are abstractions of very well understood recombinant DNA operations and provides a representation, which we call the complex, for the relevant structural properties of DNA. The PA-Match operation for lengthy strings of length $s$ cannot be feasibly implemented by recombinant DNA techniques directly by a single step of complementary pairing in DNA; nevertheless we show this Matching operation can be simulated in the RDNA model with $O(s)$ slowdown by multiple steps of complementary pairing of substrings of length 2 (corresponding to logarithmic length DNA subsequences). Each of the other operations of the PAM model can be executed in our RDNA model, without slowdown.

We further show that, with a further $O(s)/\log(1/\varepsilon)$ slowdown, the simulations can be done correctly with probability $1/2$ even if certain recombinant DNA operations (e.g., Separation) can error with a probability $\varepsilon$. We also observe efficient simulations can be done by PRAMs and thus Turing Machines of our molecular models.

Key Words. Parallel computation, Parallel RAM, Nondeterministic computation, NP, Biomolecular computation, Biotechnology, DNA, Recombinant DNA.

1. Introduction

1.1. Solving NP Search Problems by Molecular Parallelism. Feynman [F] first proposed doing computation via molecular means, but his idea was not brought to test for a number of decades. The Hamiltonian path problem is to find a path in a graph that visits each node exactly once; it is a special case of the Traveling Salesman where the problem is to find the shortest such path in a network with positively weighted edges. Adleman’s technique [Ad1] (also see comments in [G]) to solve a Hamiltonian path problem of $n$ nodes and $m$ edges required $O(n + m)$ lab steps employing short DNA strands with $O(n \log n)$ base pairs (see Section 6.1 for definition of DNA base pairs). Adleman was the first to do an experiment demonstrating biomolecular computation, solving by his method the Hamiltonian path problem for a graph of seven nodes. This was a major milestone in biomolecular computation.

Lipton [L] showed that finding the satisfying inputs to a Boolean expression of size $n$ can be done in a linear number of lab steps. His method used DNA strands with $O(n \log n)$ base pairs, and requires a number of lab steps which depends linearly on the size (rather than the depth) of the simulated expression. Also, Beaver [Be1] made similar use of molecular parallelism in the solution of the integer factorization problem. These contributions and the work of Adleman in molecular parallelism have considered only the solution of NP search problems and provided no way of quickly executing lengthy computations by purely molecular means. They all used an exponential number of DNA strands.

Applegate and others have solved the Traveling Salesman and Hamiltonian path problems for problems of size well over 1000 cities, by the use of conventional high performance workstations; these methods avoid brute-force search and use instead sophisticated heuristics. This indicates that even for exact solutions of NP complete problems we require a more general type of biomolecular computation than simply brute-force search. For example, we would like the biomolecular computation to be general enough to implement such sophisticated heuristics in parallel. (See also Section 1.3 for a further discussion of independent work and also subsequent work since this paper was presented [Re2] at the SPAA95 conference.)