Reset reproduction of CMU Computer Science report CMU-CS-83-160. Published in *AI Magazine*, Vol. 4, No. 4, pp. 33-35 and 31 (1983). Reprinted July 1994. Current address: MIT Laboratory for Computer Science, Cambridge, Massachusetts.

What Should AI Want From The Supercomputers?

Jon Doyle

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213
U.S.A.

Abstract: While some proposals for supercomputers increase the powers of existing machines like the CDC and Cray supercomputers, others suggest radical changes of architecture to speed up non-traditional operations such as logical inference in PROLOG, recognition/action in production systems, or message passing. We examine the case of parallel PROLOG to identify several related computations which subsume those of parallel PROLOG, but which have much wider interest, and which may have roughly the same difficulty of mechanization. Similar considerations apply to some other proposed architectures as well, raising the possibility that current efforts may be limiting their aims unnecessarily.

This paper will appear in *AI Magazine*.

I thank Jaime Carbonell, Clark Glymour, John McDermott, Allen Newell, Duane Pyle, Joseph Schatz, and Richmond Thomason for helpful discussions and comments. This research was supported by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory under Contract F33615-81-K-1539. The views and conclusions contained in this document are those of the author, and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Government of the United States of America.

§**1.** Excitement has been growing over computers bigger, faster, and more capable than ever before, the so-called supercomputers. Some of the proposals for new machines amplify the strengths of existing supercomputers like the CYBER 205 and the CRAY-1, while other proposals depart more radically from traditional architectures by organizing their parallelism and speed around operations different from the traditional fetch and store. For example, HILLIS' connection machine and FAHLMAN'S NETL machine focus on message passing and semantic network computations; the CMU and Columbia production system machines speed up recognize/act cycles; and the Japanese Fifth Generation Computer (FGC) focusses on logical inferences in PROLOG. In many of these efforts, the functionality of the proposed system is gotten by "parallelizing" the structure of existing sequential programmed systems. I hope to show below that in at least one case, that of the FGC, experience with the serial predecessor has provided supercomputer designers with unnecessary blinders limiting their vision, and that a significantly more interesting functionality may be possible with relatively minor changes in organization. In fact, similar potential exists in some of the other proposals as well, but the FGC offers the simplest statement of the possibilities.

§**2.** Before proposing changes, it is worth recalling just what is the functionality of the FGC, and why it was chosen. The FGC is intended to use parallelism to enhance the speed of serial PROLOG, a programming language based on logic. Considered abstractly, PROLOG is a system for determining the deductive consequences of a set of sentences in a logical language. PROLOG accepts a set of input sentences S, a goal sentence p, and subject to computability limitations, answers the question of deducibility, *Does* $S \vdash p$*?* This is not of interest to logicians alone, for PROLOG can compute quantities as answers by extracting values from the variable bindings introduced in the proof of p from S, and so serves as a general purpose programming language. Logical programming languages attract many people in artificial intelligence because of the relative ease of stating declarative information in them, as compared with traditional programming languages. Since most knowledge-based, expert systems contain large numbers of essentially declarative statements, the designers of the FGC expect their choice of PROLOG to facilitate the construction and operation of knowledge-based systems.

Parallelism enters the picture because traditional PROLOG requires that all sentences be expressed in clausal form, and searches for proofs of its goal by examining the input clauses in a fixed linear order, and within clauses, examining literals in left-to-right order. Many of these imposed orderings have no purely logical basis, so that, as far as questions of deducibility are concerned, greater efficiency may be possible with separate deduction searches conducted concurrently. In such a reorganization of PROLOG, time of execution is ideally proportional to the depth of the proof found (the size of the answer), rather than proportional to the number of alternative proofs (the size of the search space). Ideally (though practically, only in small cases), parallel PROLOG might ameliorate some current computational limitations like the present practical inequality $P \neq NP$, since by definition problems in NP have "short" proofs. On the other hand, even an ideal parallel PROLOG need have no important impact on the provably intractable problems like real arithmetic decision procedures, since in these theories, problems may have hopelessly long shortest proofs. Nevertheless, the potential speedups are sufficiently important to make the whole project very attractive as a technological advance on current computers.

§**3.** The FGC is a very powerful machine, and restricting its use to answering only *Does* $S \vdash p$*?* may be needlessly wasteful. In fact, there are three closely related questions subsuming the deductive question which, if mechanizable via analogous techniques, could be of enormous importance. These computations are those that appear in the foundations, but rarely in the practice, of statistical decision making, namely computation of probabilities, utilities, and best alternatives from non-numerical probabilities, preferences, and logical sentences. To explain these, we must review the role of statistical decision theories in artificial intelligence. For simplicity we focus on subjective Bayesian decision theory.

Statistical decision making has played a limited role in artificial intelligence, largely because of the awkwardness of its direct use. To formulate an expert system in statistical terms, one must supply a mass of conditional probabilities, prior probabilities, and utilities. Often these are not easy to come by, and even when extracted from human experts or informants, do not appear to be very reliable indicators of solid expectations. The awkwardness of the sheer *amount* of information needed is compounded by the awkwardness of *modifying* the formulation. If one decides that two values formerly close together (e.g. .7 and .8) should in fact be further apart to accommodate a larger spread of intermediate values, one must either laboriously modify *every* value in the system by hand, or sabotage the intuitions of one's informants by telling them "By the way, 0-.7 really means 0-.4, .7-.8 really means .4-.9, and .8-1 really means .9-1," thus forcing them to write 7.2 instead of .5, and so on. Even then, one cannot be sure the informant supplies numbers with the same scale in mind on Tuesday as on Monday, so the problem is worse than simple translation of ranges.

This awkwardness is almost a cruel joke played by statistical decision theorists on their adherents in artificial intelligence, since these practical difficulties are not necessary at all from the theoretical standpoint. At the foundations of Bayesian statistics lies the notion of qualitative probabilities, "bets" or judgments that one event is more probable than another. The theory takes whatever qualitative probabilities the subject is willing to espouse, and then considers the class of numerical probability distributions compatible with the original qualitative relations. (See [Savage 1972] for the details.) The current practical awkwardness of starting with numerical probabilities is easily seen in this light. If more events need to be accommodated between two previously related points, they are just inserted in the partial order of qualitative probabilities. Since no metrical notions are involved, the simple change is effected simply, without requiring hand-revision of numerical values. If the informant supplies fewer judgments of relative likelihood, the only result is a wider range of numerical distributions that fit them. By way of analogy, no one writes the physical addresses of procedures and data into their programs any more: one just describes their structure and relations, and lets the details up to linking loaders, garbage collectors, and virtual memory systems. Similarly, we might also save ourselves much unnecessary work by specifying only essential probabilistic relations, and let the machine derive numbers whenever necessary. I suspect such derivations may be possible by adapting the techniques of backing up of values widely used in search procedures, where the stipulated qualitative order forms the tree or graph being "evaluated," but currently have no concrete algorithm to recommend.

A related computation is that of deriving utilities from stipulated preferences. A century ago, economics was in the same boat as modern expert systems, at least as concerns utilities. At that time, the foundations of economics assumed that each agent had a cardinal measure of utility, so that a potato might be valued at 5 "utils," a haircut at 20, etc. But while expert systems are still stuck with the problems of individual stipulation and manual revision of systems of utilities, economists went through two stages of re-foundation. In the first, cardinal utilities were abandoned for ordinal utilities, since the utility-maximizing behavior of interest to economists is not affected by such a change. In the second, ordinal utilities were abandoned in favor of sets of binary preferences among alternatives. With certain assumptions about the character of such preference sets, one can prove that each preference set may be represented by a family of cardinal utility functions. (See [von Neumann and Morgenstern 1944].)

Now artificial intelligence is accustomed to using both qualitative goals and numerical evaluation functions, but separately, and in different circumstances. There are even debates between proponents of "discrete" and "continuous" problem solving techniques. But these debates may be irrelevant, since we can combine the two sorts of information, stipulating and using whichever sort is more convenient at the time. For example, as with qualitative probabilities, global changes to the character of utility functions have simple expression as changes of individual preferences, thus making them more attractive for stipulation even if the application demands continuous judgments. Can we in fact unify these two approaches by basing systems on qualitative preferences and constructing compatible utility functions whenever necessary?

Of course, if both of the preceding constructions are amenable to direct architectural support in supercomputers, the next step is their obvious combination into finding, among a set of alternatives, the subset of maximal expected utility. Here the maximum-finding computation should be susceptible to the already proposed parallel search techniques.

§**4.** While the preceding constructions would aid the construction of Bayesian agents, they make no special use of the logical character of PROLOG programs. In fact, one alternative construction may be equally interesting, given the use of PROLOG, that of CARNAP'S "logical" theory of probability.

While Bayesians like SAVAGE view probabilities as constructions from the choices of individuals, CARNAP proposed an alternative notion in which probabilities are measures of the amount of ambiguity of a logical theory with respect to some question. That is, we need not simply say a theory S entails neither p nor $\neg p$; we may interpret ambiguities like this so that in some cases the theory supports p more than $\neg p$ even though it strictly entails neither. In CARNAP'S idea of probability as degree of entailment, the probability of p given S is the relative "proportion" of models of p among a class of distinguished models of S. The probabilities so constructed depend on both the range of models distinguished and on the way of measuring relative proportions. CARNAP focused on two simple measures. In one, each model receives equal weight, a Laplacian assumption of sorts. In the other, the weight accorded a model is inversely proportional to the exponential of the size of its truth set. These measures can be viewed as very abstract qualitative probability relations, where in the first, all models are assigned equal likelihood, and in the second, simpler hypotheses are more likely than more complex ones. (For details, see [CARNAP 1950] or [KYBURG 1970].) There may even be interesting combinations of subjective probability judgments and measures of logical ambiguity, for instance, using logical ambiguity measures to fill in the gaps between stipulated qualitative probabilities, that is, to refine the set of distributions compatible with the qualitative probabilities above. In terms of the search procedure suggestion above for the qualitative-quantitative construction, Carnapian measures might supply the evaluations of terminal nodes in the graph of qualitative probabilities, where the terminal nodes represent the events of minimal qualitative probability. Unfortunately, we cannot pursue such questions here. (See [DOYLE 1982] for an initial treatment. A detailed reconstruction of decision theory is in preparation.)

§**5.** Neither CARNAP'S nor SAVAGE'S constructions have been pursued practically, since at the time of their introduction adequate computers and worked-out, economically important applications were scarce. But today, we have many narrow tasks formulated in Bayesian terms, a growing set of detailed applications in the form of inferentially-oriented expert systems, and even some computational explorations of Carnap-style constructions [GLYMOUR ET AL. 1983]. These constructions may not be entirely feasible on serial computers. If not, can we use supercomputers to routinely compute degrees of entailment from PROLOG programs? This computation of course subsumes the ordinary PROLOG computation, since S entails p to degree 1 only if $S \models p$, which by the completeness of first-order logic means that $S \vdash p$. In fact, some ways of computing probabilities might rely on variants of the standard computations. For example, one might try to compute probabilities by computing artificially disambiguated deductive closures, in which each actually ambiguous disjunctive or existential statement is forced into one of its cases. Different possibilities for algorithms include (1) the straightforward probabilistic procedure of choosing individual disambiguations randomly, where probabilities of the conclusions are found by repeating the global computation several times and measuring the frequency of appearance of the conclusion in question, and (2) computing a single disambiguated closure, examining it to determine the size of each alternative, using these to compute conditional probabilities, and computing probabilities of conclusions using Bayes's formula. Even with parallelism, approximations may be necessary, but even mediocre approximations would extend the power of currently proposed supercomputers.

Conclusions

§**6.** Some theoreticians have doubts about the sensibility of statistical decision making, in light of the philosophical, informational, and computational problems it involves. Their doubts may be entirely justified. But even if so, having a constructive Bayesian machine of the sort outlined above would be a wonderful experimental tool, and may serve many limited applications extremely well. If such a machine could be constructed as a simple variant of proposed supercomputers, we might as well build one instead, since its operation subsumes that of the proposed machines. For the current crop of judgmental expert systems, a qualitative Bayesian machine may be the perfect tool.

Unfortunately, as mentioned previously, algorithms and techniques (approximate, probabilistic, or otherwise) for mechanizing these computations have not yet been worked out, and there is some chance that these computations are provably infeasible even for supercomputers. Also requiring attention is adaptation of any success with logic-based systems to the alternative non-logical production systems, for the same general ideas involved in CARNAP'S constructions apply even when logical structure is not available—see [DOYLE 1982] for suggestions.

There are also other functionalities one might desire of supercomputers in addition to those discussed above, such as the ability to supply proofs when answering deductive questions, and the ability to make non-monotonic, reasoned assumptions. The former are invaluable in explanations, the latter important in problem-solving and representation. But we cannot pursue these here, except to note that both fit well with the proposed constructional approach. (See [DOYLE 1982], [DOYLE 1983A], and [DOYLE 1983B].)

References

Carnap, R., 1950. *Logical Foundations of Probability*, Chicago: University of Chicago Press.

Doyle, J., 1982. Some theories of reasoned assumptions: an essay in rational psychology, Pittsburgh: Department of Computer Science, Carnegie-Mellon University.

Doyle, J., 1983a. Methodological simplicity in expert system construction: the case of judgments and reasoned assumptions, *AI Magazine* **3**, #2, 39-43.

Doyle, J., 1983b. A society of mind: multiple perspectives, reasoned assumptions, and virtual copies, *Eighth International Joint Conference on Artificial Intelligence*.

Fahlman, S. E., 1979. *NETL: A System for Representing and Using Real World Knowledge*, Cambridge: MIT Press.

Forgy, C. L., and Newell, A., 1983. The CMU production system machine, unpublished presentation.

Glymour, C., Kelly, K., and Scheines, R., 1983. Two programs for testing hypotheses of any logical form, *Proc. 1983 International Machine Learning Workshop*, 96-89.

Hillis, W. D., 1981. The connection machine, MIT AI Laboratory, AI Memo 646.

Kyburg, H. E., Jr., 1970. *Probability and Inductive Logic*, New York: Macmillan.

Moto-oka, T., 1982. *Fifth Generation Computer Systems*, Amsterdam: North-Holland.

Savage, L. J., 1972. *The Foundations of Statistics*, 2nd rev. ed., New York: Dover.

Stolfo, S. J., and Shaw, D. E., 1982. DADO: A tree-structured machine architecture for production systems, *Proc. AAAI-82*, 242-246.

von Neumann, J., and Morgenstern, O., 1944. *Theory of Games and Economic Behavior*, Princeton: Princeton University Press.