AI Magazine Volume 3 Number 1 (1982) (© AAAI)

HIGH-ROAD AND LOW-ROAD PROGRAMS

Donald Michie
Machine Intelligence Research Unit
University of Edinburgh, UK

Consider a class of computing problem for which all sufficiently short programs are too slow and all sufficiently fast programs are too large [1]. Most non-standard problems of this kind were left strictly alone for the first twenty-years or so of the computing era. There were two good reasons. First, the above definition rules out both the algorithmic and the database type of solution. Second, in a pinch, a human expert could usually be found who was able at least to compute acceptable approximations -- for transport scheduling, job-shop allocation, inventory optimisation, or whatever large combinatorial domain might happen to be involved.

Let us now place problem-solving by machine in the more precise mental context of evaluating two particular kinds of finite function, namely

s: *Situations* → *Actions*, and
t: *Situations* x *Actions* → *Situations*.

These expressions say that s maps from a set of situations (state-descriptions) to a set of actions, and that t maps from a set of situation-action pairs to a set of situations. The function symbol s can be thought of as standing for "strategy" and t as standing for "transform". To evaluate s is to answer the question: "What to do in this situation?". To evaluate t corresponds to: "If in this situation such-and-such were done, what situation would be the immediate result?".

If the problem-domain were bicycling, we could probably construct a serviceable lookup table of s from a frame-by-frame examination of filmed records of bicyclists in action. But t would certainly be too large for such an approach. The only way to predict the next frame of a filmed sequence would be by numerically computing t using a Newtonian physics model of the bicycle, its rider and the terrain.

Machine representations corresponding to s and t are often called *heuristic* and *causal*, respectively. Note that they model different things. The first models a problem-solving skill but says nothing about the problem-domain. The second models the domain including its causality, but in itself says nothing about how to solve problems in it.

The causal model partakes of the essence of the traditional sciences, such as physics. The school physics text has much to say about the tension in a string suspending bananas from the ceiling, about the string's breaking point under stress, the force added if a monkey of stated weight were to hang from a boat-hook of given mass and dimensions having inserted its tip into the bunch, and so forth. *How* the monkey can get the bananas is left as an exercise for the reader, or the monkey.

When it has been possible to couple causal models with various kinds and combinations of search, mathematical programming and analytic methods, then evaluation of t has been taken as the basis for "high road" procedures for evaluating s. In "low road" representations s may be represented directly in machine memory as a set of (pattern → advice) rules overseen by some more or less simple control structure. A recent pattern-directed heuristic model used for industrial monitoring and control provides for default fall-back into a (computationally costly) causal-analytic model [2]. The system thus "understands" the domain in which its skill is exercised. The pattern-based skill itself is, however, sufficiently highly tuned to short-circuit, except in rare situations, the need to refer back to that understanding.

The distinction here spelled out corresponds roughly to that made by Rouse and Hunt between S-rules and T-rules in the context of computer-aided fault-diagnosis in complex machinery [3], for example, in automobiles. Their diagram, reproduced here (Figure 1), is simple but illuminating.

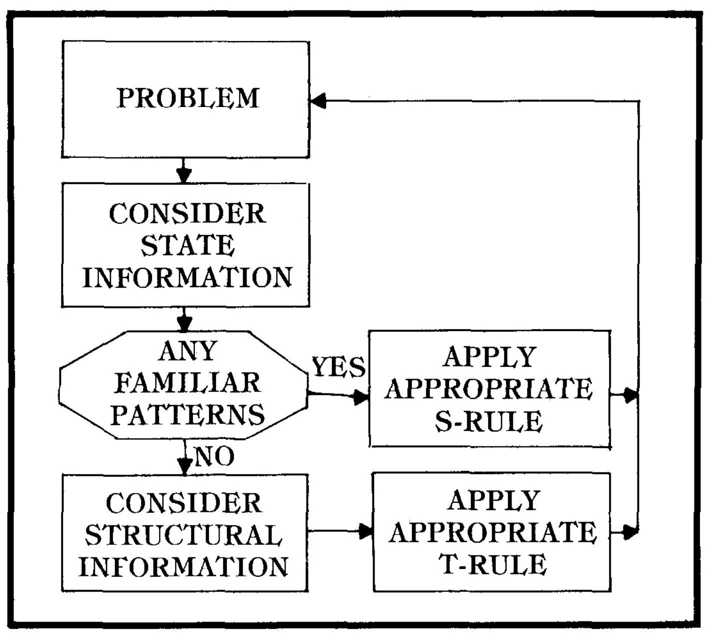

Figure 1. Overall structure of the model used by Rouse and Hunt. There are really two models, so arranged that (as in the system of Pao *et al.*) the system's "science" acts as default for its "craft".

The *s* versus *t* distinction has nothing whatsoever to do with the strange but widespread notion that problem-solving representations built from *causal* models are necessarily error-free, proved so by their implementers, and thus in some important sense "sound", while *heuristic* models are by their nature tainted with unbounded and unquantifiable error. In actuality formal proofs of correctness are no less obtainable for heuristic models [4,5] than for models of other kinds, provided that the domain is such as to sustain precise mathematical reasoning at all. The only problem-solving device yet to achieve a good and versatile record (the expert brain) has been shown to proceed at "run-time" overwhelmingly by the low road. Moreover, knowledge engineers are beginning to find in one domain after another that almost all the skill comes from the S-rules and almost all the implementational and run-time costs from the T-rules.

Perhaps this discovery should not have taken people by surprise in quite the way it seems to have done. After all it had already been noted that when a Fischer or a Karpov plays lightning chess (*S*-rules only, no time for anything else) he can still hold his own against an ordinary Master who is allowed all the time in the world for search and reasoning.

In real-world domains no more complex than chess, insistence on "high road only" has usually led to solutions which are

- opaque to the user, and
- unbelievably costly at run time.

Someone says: "I need to build an expert problem-solver, but I don't buy heuristic production-rule models. How do I know that they are *correct*, or with proved *error bounds*?"

He could equally say: "I need to make an omelet, but I don't buy eggs. How do I know that they are *not addled*?" The answer can only be: "Get your eggs certificated; or at the very least buy from a reliable farm. If you don't want to do that, then you'll have to lay them yourself." ■

REFERENCES

[1] D. Michie (1977) A theory of advice In *Machine Intelligence 8*, (eds. E. W. Elcock and D Michie), pp. 151-168. Chichester· Ellis Horwood, and New York: Halsted Press (John Wiley).

[2] Y-W. Pao, W.L. Schultz, S-Y. Oh, and R.G. Fischer (1980) A Knowledge base engineering approach to power systems monitoring and control. *Technical Report.* Cleveland: Department of Electrical Engineering, Case Western Reserve University.

[3] W.B. Rouse and R.M. Hunt (1980) A fuzzy rule-based model of human problem solving in fault diagnosis tasks. *Working paper.* Urbana: Co-ordinated Science Laboratory, University of Illinois.

[4] I. Bratko (1978) Proving correctness of strategies in the AL1 assertional language. *Inf. Proc. Letters*, *7*, 223-230.

[5] T. B. Niblett (1981) A probably correct advice strategy for the end-game king and pawn versus king. To appear in *Machine Intelligence*, *10*, (eds. D. Michie and Y-H. Pao).