CSE6390 3.0 Special Topics in AI & Interactive Systems II
Introduction to Computational Linguistics
Instructor: Nick Cercone - 3050 CSEB - nick@cse.yorku.ca
Winter Semester, 2010

---

## The Nature of Heuristics

Heuretics - the study of the informal judgmental "rules of thumb" which underlie expert know ledge-based systems.

### *1. What is the source and power of heuristics?*

The source and power of heuristics is a kind of two-dimensional *continuity.* If a heuristic H was (or would have been) useful in situation S, then it is likely that heuristics similar to H will be useful in situations similar to S. In other words, if we could compute somehow Appropriateness (Action, Situation), that function would be continuous in both variables, and would vary very slowly.

Of course the world isn't so accommodating. There are many possible measures of Appropriateness (efficiency, risk, etc.) and many possible dimensions along which Situation can vary (difficulty, time, etc.). Compounding this is the nonlinearity of the Situation space along many of these dimensions.

Nevertheless it is too attractive and too close to what human experts actually does to reject this notion out of hand. Therefore it may be useful to behave as if Appropriateness (Action, Situation) exists and is continuous.

*Example of reading paper by skipping to conclusions.*

The heuristic guidance is only as good as the generalization process you used in deciding the situation was similar (would you apply it to all articles? ...all articles written by X?)

As the world changes, a heuristic which was valid and useful may become invalid. Thus continuity is not an issue by *volatility* (or *stability*) is.

Basically we have three criteria determining which domains may adequately be modeled by heuristics: *Observability* - if data cannot be gathered, heuristics cannot be formed and evaluated; *continuity* - if the environment changes abruptly, the heuristics may never be valid; and *stability* - if the changes are continuous but rapid, the heuristics may have too short a lifespan before becoming useless. At the present time observability is the most constraining requirement. Very few fields admit automatic data acquisition. The most observable fields are those which can be completely formalized within the machine: mathematics, programming, and games.

### *2. How do new heuristics originate?*

Heuristics arise from three sources:

*Specialization* of existing, more general heuristics. This often has the form of *adapting, binding, matching* a template to observed data, producing a more specialized efficient offspring. *Compiling* and *structured programming* are two computing science analogues of this process. A second way in which specialization occurs is when an exception to a general heuristic is noted. *Debugging* and *type-checking* are the computing science analogues of such accommodation.

*Generalization* of existing more specialized heuristics. An extreme - but common - form of this is abstraction from observed data. In such a case the heuristic is a prediction about Appropriateness (Action, Situation) for a whole domain of situations and actions, based on having actually seen one or more elements of that domain. Often a powerful new theorem or technique will be proven for some domain D; it may be a useful heuristic to apply it outside D as well. For instance, the values of some infinite series were successfully guessed at by pretending they were differentiable; once the series' value is conjectured *proving* it is made much simpler.

*Analogy* to existing heuristics and to past successful acts of creating new heuristics. It is a remarkable thing that analogy works, a sign of an even deeper kind of continuity than was sought earlier. Even though two domains may appear disparate, analogous heuristics may be equally powerful in coping with them (e.g., *Look for examples of concept C before you try to prove any theories about C*). Even if heuristics for the two domains seem disparate, the paths which were followed in getting the powerful heuristics of the field may be similar.

*3. How should advice from several heuristics be combined?*

Results from building several expert systems leads to the conclusion that the details of the control structure are not critical. One can view heuristics as production rules, and then this issue becomes "What interpreter should run the rules?" This problem can itself be recursively solved by making the interpreter a production system, and so on ad infinitum, although when one tries to find such strategic rules there are few to be had, and even fewer of noticeable impact.