

AI, October 30, 2006

A. Ontology, Chapter 10, review

1. The stuff that Dr. Rohrbaugh covered
 - 10.1 ontological engineer, upper ontology
 - 10.2 (only first 2 pages) taxonomy, inheritance
2. The stuff I already covered
 - 10.4 know, believe

You may ignore the clarkKent = superman problem, but see below.
 - 10.7 closed world assumption, negation as failure in Prolog

B. Possible projects, other suggestions.

1. Think about pp. 341–343 and some references. Learn the terms reification and opaque versus transparent reference. Describe a Prolog solution to the problem that clarkKent = superman but Lois Lane doesn't know that clarkKent can fly and she does know that superman can fly. Hint: Prolog does support strings if you want to use the book's solution, and just as the book says, the same string in Prolog can be written "clark" or ['c','l','a','r','k'].
 2. On pp. 358–359 we meet the Quaker Richard Nixon situation that I introduced on the Day 1 handout. Describe a "non-monotonic logic" solution to the dilemma.
 3. Download OpenCYC from OpenCYC.org and demonstrate it with some interesting sentences.
 4. A variation of Minsky's "frames" mentioned on p. 366 is called Script Applier Mechanism (SAM), created by Roger C. Schank when he was at Yale University. Demo it. A one-page Prolog version appears in Leon Sterling and Ehud Shapiro's 1986 book, *The Art of Prolog: Advanced Programming Techniques* (Cambridge, MA: MIT Press, p.234). Our library has the book. Another small demo version of SAM appeared in *AI Expert* (October 1987, pp.19-27). I have a copy of vt_sam.pro from there, and you can ask me for it. Or you can find one on the web that has been modified and expanded from that, at www.icmc.sc.usp.br/~sandra/G9_t5/scripts.html Don't get discouraged that it is in Portuguese; Google can translate it for you. Search Google for "<tiny sam> script applier mechanism" and the first link will be to that page. Select "translate this page," and then all further links on that page (the first is to vt_sam.pro) will also be translated.

C. Neural Networks, Section 20.5

1. Special case of hill climbing via gradient ascent.
2. Terms: nodes, links, weights, activations, activation function, bias, perceptron, error
3. Parameters to set: topology, initial weights, $w[i][j]$; activation function g ; learning rate, α ; criterion for convergence (closeness, ϵ , or time trying n); training set size and whether training instances are presented cyclically or randomly, update rule.
4. History: Frank Rosenblatt (1957), Marvin Minsky & Seymour Papert (1969) *Perceptrons*.