Computer Science Department
Stanford University
Comprehensive Examination in Artificial Intelligence
Autumn 1998

October 27, 1998

PLEASE READ THIS FIRST

- You should write your answers for this part of the Comprehensive Examination in a BLUE BOOK. Be sure to write your MAGIC NUMBER on the cover of every blue book that you use.
- Be sure you have all the pages of this exam. There are three pages in addition to this.
- This exam is OPEN BOOK. You may use notes, articles, or books — but no help from people or computers.
- Show your work, since PARTIAL CREDIT will be given for incomplete answers. For example, you can get credit for making a reasonable start on a problem even if the idea does not work out. You can also get credit for realizing that certain approaches are incorrect. On a true/false question, you might get partial credit for explaining why you think something is true when we think it is false.
- Points in this exam add up to 60. Points are allocated according to the number of minutes we believe a student familiar with the material should take to answer the questions. **IF YOU ARE TAKING TOO LONG ON A QUESTION, WRITE DOWN WHATEVER YOU HAVE AND MOVE ON.**

5

1 Search (12 points)

The US Geological Survey asks you to write a program that colors any map with three colors in such a way that no two bordering countries have the same color. They give you access to a list C of countries, and to a predicate **neighbor(c, d)** that returns **true** iff countries c and d border each other.

a. (3 points) Formulate this problem as a blind tree search. In particular, state clearly what nodes and edges mean in your tree, how nodes are expanded, and which search algorithm is used. Just name the search algorithm, do not spell it out. Better solutions get more credit.

b. (3 points) When does your algorithm find out if no solution is possible?

c. (3 points) Would a heuristic-repair algorithm be appropriate for your map coloring problem? Why or why not?

d. (3 points) Independently of your solution above, does it make sense to use A^* to solve this problem? Why or why not?

2 Robotics (4 points)

a. (2 points) Draw or describe the configuration space of a screw in a bolt. Only the screw can move.

b. (2 points) Draw or describe the configuration space of a pendulum in which the bob is free to move in two dimensions (that is, *not* a planar pendulum). Make sure that the topology of your configuration space is correct, in the sense that nearby configurations are represented by nearby points.

3 Logic-Based Knowledge Representation (9 points)

a. (3 points) Represent in first-order logic the assertion **Women do not love men who hate all women.**

b. (3 points) Represent in the situation calculus the assertion **After cleaning the room, all the toys are off the floor.**

c. (3 points) Using the predicates Dog(x), Cat(x), and other predicate symbols of your choice, write a first-order theory (that is, a set of first-order sentences) in every model of which the number of cats and dogs is the same. [Hint: You may use the equality predicate =.]

4 Logic (12 points)

a. (8 points) What does a model of a first-order theory consist of?

b. (4 points) True or false:

(i) Every satisfiable formula has a model

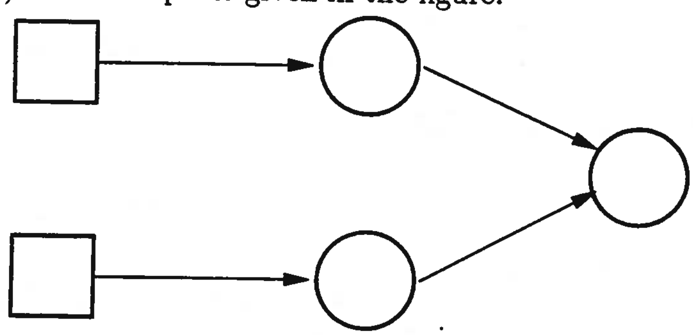
(ii) A formula is satisfiable if and only if it is not valid

(iii) Skolemizing can turn a satisfiable formula into an unsatisfiable one

(iv) If A entails B then every model of A is also a model of B

5 Learning (6 points)

Consider a neural network architecture with two input units and a single hidden layer with two units. We restrict each of the hidden units to have at most one input (but they can both have the same input). An example is given in the figure.



Assume that the inputs to all units can take on only binary values.

a. (3 points) Is the expressive power of such a unit equivalent to a single perceptron unit? Explain briefly.

b. (3 points) Is the expressive power of such a unit equivalent to a standard feed-forward neural network with two hidden units? Explain briefly.

6 Planning (17 points)

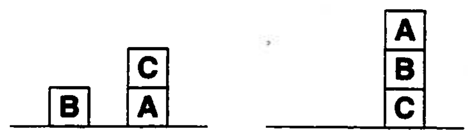
The operators used by STRIPS for standard block-stacking problems are

move(x, y, z): move block x from block y to block z

remove(x, y): move block x from block y to TABLE

stack(x, z): move block x from TABLE to block z

The Sussman anomaly, illustrated in the figure below, pointed out a basic problem with noninterleaved planners.



STRIPS will get stuck trying to either put B on top of C, or to clear A and put it on top of B. Either way, STRIPS will fail to find a plan.

a. (2 points) Using a STRIPS-like language, describe the initial state for the figure above. Your description should be detailed enough to let a complete planner find a solution.

b. (2 points) Do the same for the goal state.

c. (2 points) Write formal definitions in a STRIPS-like language for the three operators **move, remove, stack** mentioned above.

d. (3 points) Can STRIPS get lucky and find a plan for the Sussman anomaly problem above in some circumstances? If so, state which circumstances. If not, explain why not.

e. (3 points) Removing the **move** operator will force STRIPS to solve any stacking problem by first placing anything that needs to be moved onto the table, and then stacking everything back again in the required order. Does this solve the Sussman anomaly for block-stacking problems? Why or why not?

f. (5 points) Show how a partial-order planner would determine a plan for this problem. Draw a diagram illustrating a solution plan, including *all* ordering links and *all* causal links. Please use dashed lines for ordering links, and solid lines for causal links. Please also label with TR any ordering link that has been included for threat resolution. Also state a final plan as a sequence of operations.