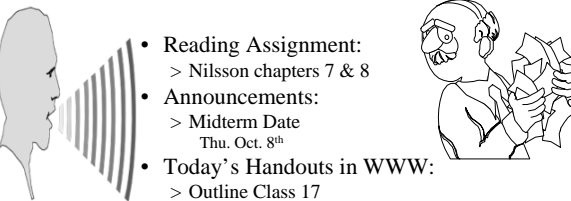


EEL5840: Elements of Machine Intelligence

Robobug

Announcements



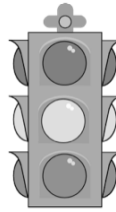
- Reading Assignment:
 - > Nilsson chapters 7 & 8
- Announcements:
 - > Midterm Date
Thu, Oct. 8th
- Today's Handouts in WWW:
 - > Outline Class 17
- Web Site
 - > www.mil.ufl.edu/eel5840

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

Today's Menu



- State Machines
 - > Blackboard Systems
 - > Example of a blackboard system for the grid-world robot
- Agents That Plan (Nilsson Chapter 7)
 - > Memory versus Computation
 - > State-Space Graphs

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

State Machines

- Blackboard Systems
 - > A *blackboard* is a data structure used in very successful AI systems [Hayes-Roth 1985]. Programs that use this architecture have been called *expert systems*—rule-based programs that exhibit the level of competence of a human expert in a specific knowledge domain. They have been used in speech understanding, medical patient-care monitoring, signal interpretation, etc.
 - > The blackboard is read and changed by programs called *knowledge sources* or KSs, which consist of condition-action rules like those in Production Systems.
 - > The condition part computes the value of a feature; it can be any condition about the blackboard data structure that evaluates to 1 or 0 {*true or false*}.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

State Machines

- Blackboard Systems
 - > The action part can either take an external action or change the blackboard data structure in some way.
 - > All KSs have access to the blackboard in parallel and when two or more KSs evaluate to 1, a *conflict resolution* program arbitrates between competing KSs.
 - > The blackboard might also be changed by perceptual subsystems that process sensory data necessitating the organizing of the blackboard in a hierarchy, with subordinate data structures occupying various levels of the hierarchy.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

State Machines

- Blackboard Systems
 - KSs are supposed to be “experts” about the part(s) of the blackboard they watch. When they detect some particular aspect of their part(s) of the blackboard, they propose changes to the blackboard, which, if selected, may evoke other KSs, and so on.
 - As computation proceeds in this manner, the blackboard ultimately becomes a data structure that contains the solution to some particular problem and/or the associated external effects change the world in a desirable way.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Armitage Armitage

EEL5840: Elements of Machine Intelligence

Robobug

State Machines

- Blackboard Systems
 - Example:
 - Suppose a robot can sense all eight cells immediately surrounding it, but the sensors sometimes give erroneous information. The robot keeps a partial map of the world in an array of cells, but because of sensor errors, this map can be incomplete and incorrect.
 - The data structure containing the map and the sensory data comprise the blackboard.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Armitage Armitage

EEL5840: Elements of Machine Intelligence

Robobug

State Machines

- Example
 - Our example has two KSs to help correct errors, namely, a *gap filler* and a *sensory filter*.
 - The gap filler looks for tight spaces in the map and either fills them in with 1's or expands them with additional 0's. It fills the top ? with a 1 since no tight spaces are allowed.
 - The sensory filter looks at both the sensory data and the map and attempts to reconcile discrepancies. In the figure s_3 is ? on the BB but the sensors indicate a strong 1 \therefore change it to 1 on the BB. The map indicates 0 for s_4 but the sensors indicate $s_4=1$, it must be a bum sensor \therefore leave the s_4 in the BB alone.

Finally, the gap filler changes the left ? to 1.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Armitage Armitage

EEL5840: Elements of Machine Intelligence

Robobug

Agents That Plan

- Memory Versus Computation
 - So far our SR agents do very little computation—they have their actions selected for them either by their designers, by learning, by evolutionary processes, or by some combination of these sources. The action functions themselves can be implemented by production rules, combinational logic, or some circuits that prescribe actions given a certain feature vector.
 - A broadly competent reactive machine, able to perform complex tasks in complex environments often require large amounts of memory. Further, its designer requires superhuman foresight in anticipating appropriate reactions for all possible situations.
 - We often trade time for space and adaptation instead of explicit design, e.g., by bringing inside the action function of the robot some of the computations that a designer of a reactive machine would have had to do.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Armitage Armitage

EEL5840: Elements of Machine Intelligence

Robobug

Agents That Plan

- Memory Versus Computation
 - > Which computations are appropriate to synthesize? One option is to consider those computations that predict the consequence of the actions possible in any given situation—surely the designer of a competent robot would have had to consider these anticipated consequences in his design.
 - > In learning processes (GAs or ANNs) the designer still has to specify what these computations are, but they will typically take much less space than would all of their results. Further specifying the computations for an ANN, for example, is MUCH easier than carrying out the computations for all possible situations. If consequence predicting computations can be learned sufficiently well (or evolved) the agent using them would be able to select appropriate actions in unforeseeable environments.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Armitage Armitage

EEL5840: Elements of Machine Intelligence

Robobug

Agents That Plan

- State-Space Graphs
 - > To predict the consequences of an action, an agent must have a model of the world it inhabits and models of the effects of its actions on its model of the world. Actual actions are postponed until the simulation shows them to be safe and effective.
 - > Example
 - Consider a grid-space world containing three toy blocks, A, B, and C, all initially on the floor. Suppose the task for our robot is to stack the blocks in ABC order, with C on the floor. How should the robot proceed?
 - Let us assume the robot is able to model the effect of each of its actions on its environment by a pair of world models—one that represents the world state before an action is taken and one that represents the world state after the action is performed.
 - The data structure in the resulting representation is called a state space graph.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Armitage Armitage

EEL5840: Elements of Machine Intelligence

Robobug

Agents That Plan

Suppose the agent is capable of moving any block, x , that has no other block on top of it, to another place y , either on the floor or on top of a clear block. We can model these actions by instances of the *schema*, $move(x,y)$ where $x=[A,B,C]$, $y=[A,B,C,floor]$, and we exclude schemas that produce no executable actions, such as $move(A,A)$. A legal instance of the schema, such as $move(A,B)$ is called an operator—operators are models of agent actions. LISP can be used to model such a graph.

Note: ((A B) (C)) and ((B C) (A)) are closer to the goal ((A B C))

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Armitage Armitage

EEL5840: Elements of Machine Intelligence

Robobug

Agents That Plan

- State-Space Graphs
 - > Thus, considering the predicted effects of a single action, the robot might prefer the actions $move(A,B)$, and $move(B,C)$ to the others—the robot has yet to move, it tentatively moves in its imaginary world...
 - > While looking ahead just one step in a simulation can often produce useful predictions, looking ahead more steps, perhaps all the way to task completion, can find shortcuts and avoid blind alleys.
 - > A useful data structure to look at the effects of alternative sequences is a directed graph. The set of worlds that an agent might produce through its actions can be represented by a directed graph whose nodes are labeled by the representations of the individual worlds and whose arcs are labeled by operators.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Armitage Armitage

EEL5840: Elements of Machine Intelligence

Robobug

Agents That Plan

- State-Space Graphs
 - > If the number of different distinguishable world situations is sufficiently small, a graph representing all the possible actions and situations can be stored explicitly. Such a graph of world models and actions is called a state-space graph. Note that by definition each action is reversible, even if we do not explicitly show this in our graphs. For example, the three toy block problem has the following graph:

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

Agents That Plan

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

Agents That Plan

From the Figure (graph) is easy to visualize and select the initial situation, mainly, ((A) (B) (C)), and if the task is to achieve the situation (the goal) described by ((A B C)), then the robot should execute the following sequence of actions: $\{move(B,C), move(A,B)\}$.

One of the advantages of representing the possible worlds in a graph structure is that any of the nodes in the graph can be taken to represent a goal situation—a flexibility in tasks should be contrasted with the single-purpose SR agents we have studied until now.

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Robobug

The End!

University of Florida
EEL 5840 - Class #17 - Fall 2009
© Dr. A. Aravamudan