CS/ENGRI 172, Fall 2002: Computation, Information, and Intelligence
9/4/02: Lecture Three Handout

Topics: Problem solving and problem spaces.
Announcements: Again, see Newell and Simon handout for the next reading assignment.

Example: River-crossing problem

(A classic dating back at least to Alcuin, 735-804 A.D.) A farmer together with a wolf, a goat, and a roughly goat-sized bale of crushed tin cans is on the left bank of a river, and wishes to get all these "goods" to the other side. The available boat can only be rowed by the farmer and can only bear the farmer and at most one of the goods. But, if the wolf and the goat are on the same shore, then the farmer must be there too to prevent the wolf from eating the goat; likewise, the goat and the bale of cans cannot be on the same shore without the farmer. Can the farmer get all the goods to the other side of the river?

Corresponding states.

1) FWGC‖ 2) FWC‖G 3) FGC‖W 4) FWG‖C 5) FG‖WC 6) FC‖WG 7) FW‖GC 8) F‖WGC

9)WC‖FG 10)C‖FWG 11)W‖FGC 12)G‖FWC 13)‖FWGC 14)WG‖FC 15)GC‖FW 16)WGC‖F

Problem space definitions

To specify a problem is to specify the *problem space*:

- Set (at least one) of *states*: distinct allowable situations
- Distinguished goal state(s): acceptable outcome(s)
- Distinguished (single) initial state: beginning situation
- Operators: distinct allowable actions, specified by indicating (perhaps implicitly) which state is reached when the action is performed in a particular state

(Translation key: Newell and Simon's *tests* are ways to describe or identify the goal state(s); their *move generators* are methods for choosing among the available operators.)

Design principles

A well-designed problem space:

1. has states that incorporate all and only relevant information, and
2. is as small as possible, but not more so.

Do not assume any constraints that aren't explicitly given; you may have different assumptions from your colleagues. Ask first.

(OVER)

Example: requirements problem

You're a student entering the Engineering school without any advanced placement credit. The roster for this semester is:

	Fall
9:00	ENGRI 111
	MATH 191
10:00	CHEM 211
11:00	MATH 191
	MATH 192
12:00	FWS 270

To remain in the program, you must have taken an ENGRI, a science class, and a math course by the end of the semester. You may not register for two class sections that are held at the same time.

(We'll skip the specification of the initial and goal states.)

1. *States* are "checklists" of the form

   [engri: x_1, science: x_2; math: x_3, fws: x_4]

   where the x_i's are either blanks or the name of all courses taken of the corresponding type. *Operators* take the form: "Take the following schedule ..." (e.g., "Take ENGRI 111 at 9, take Chem 211 at 10, and take Math 192 at 11").

2. *States* are "checklists" of the form

   [engri: x_1, science: x_2; math: x_3, fws: x_4]

   where the x_i's are either blanks or a checkmark if a course of that type was taken. *Operators* take the form: "Take the following schedule ..." (e.g., "Take ENGRI 111 at 9, take Chem 211 at 10, and take Math 192 at 11").

3. *States* are "checklists" of the form

   [engri: x_1, science: x_2; math: x_3, fws: x_4]

   where the x_i's are either blanks or a checkmark if a course of that type was taken. *Operators* take the form: "Add course C to the schedule" (e.g., "Add ENGRI 111 to the schedule").