Computation, Information, and Intelligence (ENGRI/CS/INFO/COGST 172), Spring 2007
1/29/07: **Lecture aid — Further analysis of example implicit specifications**

Agenda: Continue our discussion and analysis of design choices for implicit specifications, using the course-requirements problem as our running example of a real-life scheduling problem in microcosm.

Note: In the variants of specification #1 given below, explanations and motivations have been omitted for space. *Specifications you give in coursework must include such information.* Italics denote variables.

Course roster			
9 MTWRF	ENGRI 111, MATH 171, MATH 191	11 MTWRF	CHEM 211, MATH 191, MATH 192
10 MTWRF	CHEM 207, ENGRI 172	12 MTWRF	*ART 151*, *FWS 270*, PHYS 116

Implicit "specification" "#1 minus other".

A. States: all "advising worksheets" of the form [engri: x_{engri}; science: $x_{science}$; math: x_{math}; ~~**other:** x_{other}~~] where

- each x_i is either a blank ("—") or a list of items of the form $course(time)$ such that $course$ is a class of type i that meets at time $time$;
- (no-conflict constraint) No time appears more than once among all the x_is; and
- (ordering constraint) if x_i lists multiple courses, they are listed alphabetically and then by ascending numerical order and then by ascending course-meeting time.

B. Initial state: [engri: —; science: —; math: — ~~**; other: —**~~].

C. Goal states: those of the form [engri: x_{engri}; science: $x_{science}$; math: x_{math}; ~~**other:** x_{other}~~] such that none of x_{engri}, $x_{science}$, or x_{math} has the value "—".

D. Actions: all pairs of the form $\langle course, time \rangle$ where $course$ is a class meeting at time $time$.

An action $\langle course, time \rangle$ applies to any state [engri: x_{engri}; science: $x_{science}$; math: x_{math}; ~~**other:** x_{other}~~] such that none of the x_i's lists a course time of time $time$. If the class $course$ is of type i, then the result of applying $\langle course, time \rangle$ to an applicable state is to transition to the state in which the pair $course(time)$ has been added to the appropriate location in the list x_i as specified by the state-set definition above. (Of course, if x_i is blank, then the new state has the blank replaced by $course(time)$; **and if i is "other", then "transition" to the state the action was applied to.**)

Implicit "specification" "#1 minus ordering".

A. States: all "advising worksheets" of the form [engri: x_{engri}; science: $x_{science}$; math: x_{math}; other: x_{other}] where

- each x_i is either a blank ("—") or a list of items of the form $course(time)$ such that $course$ is a class of type i that meets at time $time$;
- (no-conflict constraint) No time appears more than once among all the x_is; and
- ~~**(ordering constraint) if x_i lists multiple courses, they are listed alphabetically and then by ascending numerical order and then by ascending course-meeting time.**~~

B, C, and D (initial state, goal states, actions) are defined as in the original specification #1.

(OVER)

Implicit specification #2. Italics denote variables. This specification exhibits the minimum level of explanations and descriptions of motivation that we require of you. There are some modifications, essentially cosmetic, from the version given in last lecture's handout.

A. The set of states consists of checklists of the form

$$[\text{engri: } x_{engri}\text{; science: } x_{science}\text{; math: } x_{math}\text{; 9: } t_9\text{; 10: } t_{10}\text{; 11: } t_{11}\text{; 12: } t_{12}]$$

where each x_i and t_i is either "—" or "✓". The intent is that $x_i = ✓$ if and only if a course of type i has been scheduled, and that $t_j = ✓$ if and only if a course has been scheduled for time j.

B. The initial state is [engri: —; science: —; math: —; 9: —; 10: —; 11: —; 12: —].

C. The set of goal states is the set of states of the form [engri: ✓; science: ✓; math: ✓; 9: t_9; 10: t_{10}; 11: t_{11}; 12: t_{12}] where each t_i may be either ✓ or —.

D. The set of actions corresponds to all pairs of the form $\langle course, time\rangle$ where $course$ is a class that meets at time $time$.

An action $\langle course, time\rangle$ applies to any state [engri: x_{engri}; science: $x_{science}$; math: x_{math}; 9: t_9; 10: t_{10}; 11: t_{11}; 12: t_{12}] such that $t_{time} =$ —; that is, we disallow time conflicts, as required. The result of applying $\langle course, time\rangle$ to such a state is to transition to the state in which all the variables have the same value as in the application state except that t_{time} has been changed from — to ✓, and, if $course$ is a class of type i and x_i is blank in the application state, then $x_i = ✓$ in the new state. The idea is that every course scheduled counts toward its type and its time period.

Implicit specification #3. Motivations and explanations omitted for lecture conciseness

A. The states consist of checklists of the form [engri: x_{engri}; science: $x_{science}$; math: x_{math}; other: x_{other}] where each x_i is either "—" or "✓".

B. The initial state is [engri: —; science: —; math: —; other: —.].

C. Goal states: those of the form [engri:✓; science:✓; math:✓; other:x], where x is ✓ or —.

D. The set of actions are all those of the following types

$\langle$9: y_9; 10: y_{10}; 11: $y_{11}\rangle$	$\langle$10: y_{10}; 11: y_{11}; 12: $y_{12}\rangle$
$\langle$9: y_9; 10: y_{10}; 12: $y_{12}\rangle$	$\langle$9: y_9; 10: y_{10}; 11: y_{11}; 12: $y_{12}\rangle$
$\langle$9: y_9; 11: y_{11}; 12: $y_{12}\rangle$	

where each y_j is a course meeting at time j.

All the operators apply only to the initial state. Applying any action $\langle j$: y_j; k: y_k; ℓ: $y_\ell\rangle$ to this state results in the state [engri: x_{engri}; science: $x_{science}$; math: x_{math}; other: x_{other}] where each x_i has the value "✓" if and only if at least one of y_j, y_k, or y_ℓ is a class of type i. Applying action $\langle$9: y_9; 10: y_{10}; 11: y_{11}; 12: $y_{12}\rangle$ to the initial state results in the state [engri: x_{engri}; science: $x_{science}$; math: x_{math}; other: x_{other}] where each x_i has the value "✓" if and only if at least one of y_9, y_{10}, y_{11}, or y_{12} is a class of type i.