

Faculty Schedule System (Version 0.1 - 2003)

Mark Sebern

September 22, 2003

1 Introduction

The faculty schedule system maintains schedule information for an academic term. It keeps track of teaching assignments (professors assigned to each course section).

The basic elements in the schedule system include professors, courses, sections, and academic departments.

$[PROF, COURSE, SECTION]$

$DEPT ::= AEBC \mid BUS \mid EECS \mid GS \mid MATH \mid ME \mid NUR \mid PC$

Note



Although it may not be necessary in this example specification, it is possible to declare variables that represent specific instances of these types. This can be done with a Z axiomatic description.

$$\left| \begin{array}{l} se381 : COURSE \\ jones, smith : PROF \end{array} \right.$$

Variable names in Z are case-sensitive. They may contain alphanumeric characters, punctuation marks, Greek letters, and some pictorial symbols. Some of the non-alphanumeric characters have conventional meanings.

At times, it may be helpful to define sets with elements of a certain type.

$$\left| \begin{array}{l} SEcourses : \mathbb{P} COURSE \\ CEcourses : \mathbb{P} COURSE \end{array} \right.$$

Sets like these can also be used in variable declarations.

$$\left| \begin{array}{l} se283 : SEcourses \\ ee210 : CEcourses \end{array} \right.$$

In the declarations just above, the type of $se283$ and $ee210$ is $COURSE$ and there are implicit predicates that require their values to be members of the sets $SEcourses$ and $CEcourses$, respectively. The declarations could be normalized by making these facts explicit.

$$\left| \begin{array}{l} se283 : COURSE \\ ee210 : COURSE \\ \hline se283 \in SEcourses \\ ee210 \in CEcourses \end{array} \right.$$

An axiomatic description with a predicate can also be used to declare a variable that represents the maximum size of a class section.

$$\frac{\text{max_section_size} : \mathbb{N}}{\text{max_section_size} \leq 100}$$

Note that this declaration of *max_section_size* means that it can take on one value between 0 and 100. (If zero is not a reasonable value, the declaration above could be modified by adding another predicate.)

2 State schema for the schedule

A schedule keeps track of the basic elements described above. A course has zero or more sections associated with it. Note that each section is associated with exactly one course. Professors are assigned to teach course sections.

Here is the state schema for the schedule system. It consists of declarations of the various state variables (including functions) and the predicates that specify the system invariants.

<p><i>Schedule</i></p> <hr/> <p><i>profs</i> : $\mathbb{P} \text{PROF}$ <i>courses</i> : $\mathbb{P} \text{COURSE}$ <i>sections</i> : $\mathbb{P} \text{SECTION}$ <i>prof_dept</i> : $\text{PROF} \leftrightarrow \text{DEPT}$ <i>course_sections</i> : $\text{SECTION} \leftrightarrow \text{COURSE}$ <i>assigned_sections</i> : $\text{PROF} \leftrightarrow \text{SECTION}$</p> <hr/> <p>$\text{dom } \text{prof_dept} = \text{profs}$ $\text{dom } \text{course_sections} = \text{sections}$ $\text{ran } \text{course_sections} \subseteq \text{courses}$ $\text{dom } \text{assigned_sections} \subseteq \text{profs}$ $\text{ran } \text{assigned_sections} \subseteq \text{sections}$</p>
--

3 Initialization schema

Initially, the schedule system has no professors, courses, or sections.

<p><i>Init</i></p> <hr/> <p><i>Schedule</i></p> <hr/> <p><i>profs</i> = \emptyset <i>courses</i> = \emptyset <i>sections</i> = \emptyset <i>prof_dept</i> = \emptyset <i>course_sections</i> = \emptyset <i>assigned_sections</i> = \emptyset</p>
--

4 Operations involving professors

Professors may be added to or deleted from the schedule.

4.1 Adding a professor

Adding a professor to the schedule is pretty simple. The schedule's courses and sections are unaffected.

AddProfessor <hr/> $\Delta\text{Schedule}$ $\text{prof?} : \text{PROF}$ $\text{dept?} : \text{DEPT}$ <hr/> $\text{prof?} \notin \text{profs}$ $\text{profs}' = \text{profs} \cup \{\text{prof?}\}$ $\text{prof_dept}' = \text{prof_dept} \cup \{\text{prof?} \mapsto \text{dept?}\}$ $\text{courses}' = \text{courses}$ $\text{sections}' = \text{sections}$ $\text{course_sections}' = \text{course_sections}$ $\text{assigned_sections}' = \text{assigned_sections}$

Note



The $\Delta\text{Schedule}$ notation above is shorthand for a schema definition that can be written in one of two ways.

$$\Delta\text{Schedule} \hat{=} \text{Schedule} \wedge \text{Schedule}'$$

$\Delta\text{Schedule}$ <hr/> Schedule $\text{Schedule}'$
--

The decoration apostrophe in $\text{Schedule}'$ is propagated to the state variables inside the Schedule schema.

4.2 Deleting a professor

Deleting a professor from the schedule removes all section assignments for that professor. The operation precondition is that the professor must be in the system. Courses and course sections are not affected.

DeleteProfessor <hr/> $\Delta\text{Schedule}$ $\text{prof?} : \text{PROF}$ <hr/> $\text{prof?} \in \text{profs}$ $\text{profs}' = \text{profs} \setminus \{\text{prof?}\}$ $\text{prof_dept}' = \{\text{prof?}\} \triangleleft \text{prof_dept}$ $\text{assigned_sections}' = \{\text{prof?}\} \triangleleft \text{assigned_sections}$ $\text{courses}' = \text{courses}$ $\text{sections}' = \text{sections}$ $\text{course_sections}' = \text{course_sections}$
--