

State Space:

Register

$enrolled : \mathbb{P} \text{ STUDENT}$

$completed : \mathbb{P} \text{ STUDENT}$

$completed \subseteq enrolled$

Initial State:

InitRegister

Register

$enrolled = \emptyset$

Enroll a new student:

EnrollStudent

$\Delta \text{Register}$

$student? : \text{STUDENT}$

$student? \notin enrolled$

$enrolled' = enrolled \cup \{student?\}$

Add an enrolled student to *completed*:

AddCompleted

$\Delta \text{Register}$

$student? : \text{STUDENT}$

$student? \notin completed$

$completed' = completed \cup \{student?\}$

Enquire if a student has completed homework:

EnquireCompleted

$\exists \text{Register}$

$student? : \text{STUDENT}$

$answer! : \{\text{Yes}, \text{No}\}$

$student? \in enrolled$

$(student? \in completed \wedge answer! = \text{Yes}) \vee$

$(student? \notin completed \wedge answer! = \text{No})$

Robust Specification

Success:

<i>Success</i>
<i>result! : REPORT</i>
<i>result! = ok</i>

Error for *AddStudent* when student being added is already enrolled:

<i>AlreadyEnrolled</i>
$\exists Register$
<i>student? : STUDENT</i>
<i>result! : REPORT</i>
<i>student? \in enrolled</i>
<i>result! = already_enrolled</i>

Error for *AddCompleted* when student being added is not enrolled:

<i>NotEnrolled</i>
$\exists Register$
<i>student? : STUDENT</i>
<i>result! : REPORT</i>
<i>student \notin enrolled</i>
<i>result! = not_enrolled</i>

RAddStudent $\triangleq (AddStudent \wedge Success) \vee AlreadyEnrolled$

RAddCompleted $\triangleq (AddCompleted \wedge Success) \vee NotEnrolled$

REnquireCompleted = (EnquireCompleted \wedge Success) \vee NotEnrolled