Formal Specification F28FS2, Lecture 4 (Up to Chapter 4.7 of Currie's book)

Jamie Gabbay

January 27, 2014

Taking stock

We've done propositions and predicates.

We've done types. We've done sets. We've done powersets.

Now we do schema.

Our first spec: the badminton club

It's a toy example.

Structure of the specification

Specify a type: $[\mathrm{STUDENT}]$.

This is a free type declaration. You can now declare variables $x : \mathrm{STUDENTS}$.

Axiomatic definitions

$$
\begin{array}{|l}
\text{maxplayers} : \mathbb{N} \\
\hline
\text{maxplayers} = 20
\end{array}
$$

This is a judgement form. It asserts:

- maxplayers is a variable ranging over $\mathbb{N}$.
- maxplayers $= 20$ does have truth-value T.

maxplayers $= 20$ is true because we asserted it to be true.

More judgement forms: Schema

$$\begin{array}{|l}
\hline
\textit{ClubState} \\
\mathsf{badminton} : \mathbb{P}\,\mathsf{STUDENT} \\
\mathsf{hall} : \mathbb{P}\,\mathsf{STUDENT} \\
\hline
\mathsf{hall} \subseteq \mathsf{badminton} \\
\#\mathsf{hall} \leq \mathsf{maxplayers} \\
\hline
\end{array}$$

badminton : $\mathbb{P}$STUDENT and hall : $\mathbb{P}$STUDENT are the state variables. They are the parameters of the model.

hall $\subseteq$ badminton and $\#$hall $\leq$ maxplayers are constraints or invariants. Values for badminton and hall are valid, if they satisfy the constraints.

More judgement forms: Schema

We could equivalently write one constraint:

$$\begin{array}{l} \mathsf{hall} \subseteq \mathsf{badminton} \\ \#\mathsf{hall} \leq \mathsf{maxplayers} \end{array} \quad \text{or} \quad \mathsf{hall} \subseteq \mathsf{badminton} \land \#\mathsf{hall} \leq \mathsf{maxplayers}$$

Values for the state variables that satisfy the constraints are valid.

State change

Now we're interested in schema that express changes to the state.

By convention, we represent a state change by making two copies of it;

- badminton and hall for before, and
- badminton′ and hall′ for after.

So now we have one state describing 'before-after'.

We annotate any other variables with $?$ for input and $!$ for output.

AddMember

AddMember

$$
\begin{array}{l}
\left.\begin{array}{l}
\mathsf{badminton} : \mathbb{P}\,\mathsf{STUDENT} \\
\mathsf{hall} : \mathbb{P}\,\mathsf{STUDENT}
\end{array}\right\} \text{— before} \\
\left.\begin{array}{l}
\mathsf{badminton}' : \mathbb{P}\,\mathsf{STUDENT} \\
\mathsf{hall}' : \mathbb{P}\,\mathsf{STUDENT}
\end{array}\right\} \text{— after} \\
\left.\mathsf{newmember?} : \mathsf{STUDENT}\right\} \text{— input} \\
\hline
\mathsf{hall} \subseteq \mathsf{badminton} \qquad \#\mathsf{hall} \leq \mathsf{maxplayers} \\
\mathsf{hall}' \subseteq \mathsf{badminton}' \qquad \#\mathsf{hall}' \leq \mathsf{maxplayers} \\
\mathsf{newmember?} \notin \mathsf{badminton} \\
\mathsf{badminton}' = \mathsf{badminton} \cup \{\mathsf{newmember?}\} \\
\mathsf{hall}' = \mathsf{hall}
\end{array}
$$

Preconditions, postconditions

A precondition is a predicate describing the state before.

A postcondition is a predicate describing the state after.

What are the preconditions and postconditions of the example in the last slide?

Precondition, postcondition

The precondition:

$$\text{newmember?} \notin \text{badminton}.$$

How about the postcondition?

Precondition, postcondition

Note how we asserted a relationship

$$\mathsf{hall}' = \mathsf{hall}.$$

As far as Z is concerned, hall' and hall are just distinct variables.

Students often forget about this.

Z is a specification language, not a programming language. There is no persistent state (unless we specify that there is).

Syntactic sugar

ClubState′ is *ClubState* with primed state variables:

$$
\begin{array}{|l}
\hline
\textit{ClubState}' \\
\text{badminton}' : \mathbb{P}\text{STUDENT} \\
\text{hall}' : \mathbb{P}\text{STUDENT} \\
\hline
\text{hall}' \subseteq \text{badminton}' \\
\#\text{hall}' \leq \text{maxplayers} \\
\hline
\end{array}
$$

Δ convention

ΔS is a copy of S, and a copy of S', put together:

$$
\begin{array}{|l}
\hline
\textit{ΔClubState} \\
\mathsf{badminton} : \mathbb{P}\mathsf{STUDENT} \\
\mathsf{hall} : \mathbb{P}\mathsf{STUDENT} \\
\mathsf{badminton}' : \mathbb{P}\mathsf{STUDENT} \\
\mathsf{hall}' : \mathbb{P}\mathsf{STUDENT} \\
\hline
\mathsf{hall} \subseteq \mathsf{badminton} \\
\#\mathsf{hall} \leq \mathsf{maxplayers} \\
\mathsf{hall}' \subseteq \mathsf{badminton}' \\
\#\mathsf{hall}' \leq \mathsf{maxplayers} \\
\hline
\end{array}
$$

Schema inclusion

$$
\begin{array}{|l}
\hline
\textit{AddMember} \\
\Delta \textit{ClubState} \\
\text{newmember?} : \text{STUDENT} \\
\hline
\text{newmember?} \notin \text{badminton} \\
\text{badminton}' = \text{badminton} \cup \{\text{newmember?}\} \\
\text{hall}' = \text{hall} \\
\hline
\end{array}
$$

That's a lot more readable.

Exercise 4.1

$$
\begin{array}{|l}
\hline
\textit{RemoveMember} \\
\Delta \textit{ClubState} \\
\mathrm{member?} : \mathrm{STUDENT} \\
\hline
\mathrm{member?} \in \mathrm{badminton} \\
\mathrm{badminton}' = \mathrm{badminton} \setminus \{\mathrm{member?}\} \\
\mathrm{hall}' = \mathrm{hall} \setminus \{\mathrm{member?}\} \\
\hline
\end{array}
$$

Precondition: $\mathrm{member?} \in \mathrm{badminton}$

Postconditions:

$\mathrm{badminton}' = \mathrm{badminton} \setminus \{\mathrm{member?}\}$ $\quad$ $\mathrm{hall}' = \mathrm{hall} \setminus \{\mathrm{member?}\}$

Entering the hall

$$
\begin{array}{|l}
\hline
\textit{EnterHall} \\
\Delta \textit{ClubState} \\
\mathsf{enterer?} : \mathsf{STUDENT} \\
\hline
\mathsf{enterer?} \in \mathsf{badminton} \\
\mathsf{enterer?} \notin \mathsf{hall} \\
\#\mathsf{hall} < \mathsf{maxplayers} \\
\mathsf{hall}' = \mathsf{hall} \cup \{\mathsf{enterer?}\} \\
\mathsf{badminton}' = \mathsf{badminton} \\
\hline
\end{array}
$$

Entering the hall

Preconditions:

$$\text{enterer?} \in \text{badminton} \quad \text{enterer?} \notin \text{hall} \quad \#\text{hall} < \text{maxplayers}$$

Postconditions:

$$\text{hall}' = \text{hall} \cup \{\text{enterer?}\} \qquad \text{badminton}' = \text{badminton}$$

Remember: $\Delta \mathit{ClubState}$ is $\mathit{ClubState}$ plus $\mathit{ClubState}'$.

Exercise 4.2: leaving the hall

LeaveHall

Δ*ClubState*
$\mathsf{leaver?} : \mathsf{STUDENT}$

$\mathsf{leaver?} \in \mathsf{hall}$
$\mathsf{hall}' = \mathsf{hall} \setminus \{\mathsf{leaver?}\}$
$\mathsf{badminton}' = \mathsf{badminton}$

Precondition: $\mathsf{leaver?} \in \mathsf{hall}$.

Postconditions: you work it out.

The Ξ schema, and queries

$$
\begin{array}{|l}
\hline
\Xi \mathit{ClubState} \\
\hline
\Delta \mathit{ClubState} \\
\hline
\mathrm{badminton}' = \mathrm{badminton} \\
\mathrm{hall}' = \mathrm{hall} \\
\hline
\end{array}
$$

The Ξ-schema is just the Δ-schema with no preconditions and postconditions meaning 'no change' or 'everything stays the same'.

The Ξ schema, and queries

We can use that to ‘output’ information, like ‘who isn’t in the hall’:

$$
\begin{array}{|l}
\hline
\textit{NotInHall} \\
\hline
\Xi \textit{ClubState} \\
\text{outside!} : \mathbb{P}\text{STUDENT} \\
\hline
\text{outside!} = \text{badminton} \setminus \text{hall} \\
\hline
\end{array}
$$

Exercise 4.3

Suppose a type MESSAGE ::= inhall | notinhall | notmember.

Specify an operation which outputs x : MESSAGE stating whether s : STUDENT is

1. In the hall.
2. Not in the hall.
3. Not a member.

Exercise 4.3

Location

$\Xi ClubState$
$s? : STUDENT$
$report! : MESSAGE$

$s? \in hall \Rightarrow report! = inhall$
$(s? \notin hall \land s? \in badminton) \Rightarrow report! = notinhall$
$(s? \notin badminton) \Rightarrow report! = notmember$