LECTURE 17: LIBRARY CASE STUDY

Software Engineering

Mike Wooldridge

1 A Library Management System

- In this lecture, we specify a simple library system.
- Operations:
  - – check out a book;
  - – return a book;
  - – add a book to library;
  - – remove book from library;
  - – get list of books by author or subject area;
  - – get list of books checked out by particular borrower;

- All books must either be checked out or available for check out.
- No book may be simultaneously checked out and available.
- There is an upper limit to number of books that may be checked out.

1.1 Types

- We need sets for:
  - – all possible books;
  - – all possible copies of books;
  - – all possible people;
  - – all possible authors;
  - – all possible subjects;
  - – the various reports that may be produced.
- So parachute in:

$$[BOOK, COPY, PERSON, AUTHOR,\\ SUBJECT, REPORT]$$

1.2 State Space

- The state space is describes in several steps. First, a schema containing information relating to books in the library.

ParaLibrary

$$\begin{array}{|l}
instance\_of : COPY \rightarrowtail\!\!\!\!\!\!+\; BOOK \\
written\_by : BOOK \nrightarrow \mathbb{P}\, AUTHOR \\
about : BOOK \nrightarrow \mathbb{P}\, SUBJECT \\
\hline
\mathrm{dom}\ written\_by \subseteq \mathrm{ran}\ instance\_of \\
\mathrm{dom}\ about \subseteq \mathrm{ran}\ instance\_of
\end{array}$$

- *instance_of* tells us what book a copy is an instance of;
- the set

  $\mathrm{ran}\ instance\_of$

  is the set of all books in the library;
- *written_by* tells us who a book is written by; there may be more than one author, hence the powerset operation; there may be no authors;
- *about* tells us the subjects a book is about; there may be no subjects;
- first invariant tells us that we only know who wrote books in the library;
- second invariant tells us that we only know subjects of books in the library.

- The database part of the schema is as follows:

LibraryDB

$$\begin{array}{|l}
borrower, staff : \mathbb{P}\, PERSON \\
available, out : \mathbb{P}\, COPY \\
borrowed\_by : COPY \nrightarrow PERSON \\
\hline
borrower \cap staff = \emptyset \\
available \cap out = \emptyset \\
\mathrm{dom}\ borrowed\_by = out \\
\mathrm{ran}\ borrowed\_by \subseteq borrower \\
\forall p : borrower \bullet \#borrowed\_by^{\sim}(\!| \{p\} |\!) \\
\quad \leq MaxCopies
\end{array}$$

- *borrower* is the set of all borrowers known to the system;
- *staff* is the set of all staff known to the system;
- *available* is the set of all available books;
- *out* is the set of borrowed books (i.e., ones that have been checked out);
- *borrowed_by* tells us who borrowed the books out on loan.

- 1st invariant tells us that a person cannot be both a borrower and a staff;
- 2nd invariant tells us that books cannot be both available and checked out;
- 3rd invariant tells us that the only books appear have been borrowed by someone are those that are out;
- 4th invariant tells us that books can only be borrowed by borrowers;
- 5th invariant tells us that a borrower can only have out up to the maximum number of books.

- The library state space is then as follows:

Library

$ParaLibrary$
$LibraryDB$

$\operatorname{dom} instance\_of = available \cup out$

- the only invariant in this schema tells us that the library does not know anything about books which are not in stock.

1.3 The Operations

- We assume initialisation operations; these are trivial.
- First we look at checking out books…
- Inputs: person name ($n?$) and copy ($c?$).

CheckOut

$\Delta Library$
$n? : PERSON$
$c? : COPY$

$n? \in borrower$
$c? \in available$
$\#borrowed\_by^{\sim}(\!|\{n?\}|\!)$
$\quad < MaxCopies$
$available' = available \setminus \{c?\}$
$out' = out \cup \{c?\}$
$borrowed\_by' = borrowed\_by \cup$
$\quad \{c? \mapsto n?\}$

- (Note that $f^{\sim}$ is the inverse of f.)
- 1st precondition is that the person trying to borrow must be a known borrower;
- 2nd precondition is that the book must be available;
- 3rd precondition is that the person trying to borrow must have out fewer than the maximum number of books available;
- the postconditions define the changes made to *available*, *out* and *borrowed_by*.

1.4 Returning a Book

- One input: the copy to be returned.

$$
\begin{array}{|l}
\textit{Return} \\
\hline
\Delta \textit{Library} \\
c? : \textit{COPY} \\
\hline
c? \in \textit{out} \\
\textit{available}' = \textit{available} \cup \{c?\} \\
\textit{out}' = \textit{out} \setminus \{c?\} \\
\textit{borrowed\_by} = \{c?\} \mathbin{\lhd\!\!\!-} \textit{borrowed\_by} \\
\hline
\end{array}
$$

- precondition states that the book can only be returned if it is out;
- 1st post-condition says that the book is available after the operation;
- 2nd post-condition says that the book is no longer out;
- 3rd post-condition uses domain subtraction to remove the correct record from the *borrowed_by* function.
- For example,

$$\textit{borrowed\_by} = \{b01 \mapsto \textit{mjw}, b02 \mapsto \textit{en}, b03 \mapsto \textit{mjw}\}$$

$$\{b01\} \mathbin{\lhd\!\!\!-} \textit{borrowed\_by} = \{b02 \mapsto \textit{en}, b03 \mapsto \textit{mjw}\}$$

1.5 Adding Books to the Library

- There are two cases to consider:
  - where the book is completely new to the library;
  - where the book is another copy of a book that is already in the library.
- We have two schemas to capture these two situations:
  - *AddNewBook*;
  - *AddAnotherCopy*.

$$
\begin{array}{|l}
\textit{AddNewBook} \\
\hline
\Delta \textit{Library} \\
c? : \textit{COPY} \\
b? : \textit{BOOK} \\
a? : \mathbb{P}\,\textit{AUTHOR} \\
s? : \mathbb{P}\,\textit{SUBJECT} \\
\hline
b? \notin \operatorname{ran} \textit{instance\_of} \\
c? \notin \textit{available} \cup \textit{out} \\
\textit{available}' = \textit{available} \cup \{c?\} \\
\textit{instance\_of}' = \textit{instance\_of} \cup \{c? \mapsto b?\} \\
\textit{written\_by}' = \textit{written\_by} \cup \{b? \mapsto a?\} \\
\textit{about}' = \textit{about} \cup \{b? \mapsto s?\} \\
\hline
\end{array}
$$

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{AddAnotherCopy}} \\
\Delta \textit{Library} \\
c? : \textit{COPY} \\
b? : \textit{BOOK} \\
\hline
c? \notin \textit{available} \cup \textit{out} \\
b? \in \operatorname{ran} \textit{instance\_of} \\
\textit{available}' = \textit{available} \cup \{c?\} \\
\textit{instance\_of}' = \textit{instance\_of} \cup \{c? \mapsto b?\} \\
\hline
\end{array}$$

1.6 Removing Books

- Removing a books from the library is similarly complicated; once again there are 2 possibilities to consider…
  - removing a book that is the only copy;
  - removing one copy of a book leaving several other copies behind.
- Two schemas:
  - *RemoveOther* to remove one of several copies;
  - *RemoveLast* to remove the last copy.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{RemoveOther}} \\
\Delta \textit{Library} \\
c? : \textit{COPY} \\
\hline
c? \in \textit{available} \\
\#(\textit{instance\_of}^{\sim}(\!| \{\textit{instance\_of}(c?)\} |\!)) > 1 \\
\textit{available}' = \textit{available} \setminus \{c?\} \\
\hline
\end{array}$$

- Note that there is no need to alter any variables in *ParaLibrary*; we only change *available*, to indicate that the book is no longer available.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{RemoveLast}} \\
\Delta \textit{Library} \\
c? : \textit{COPY} \\
\hline
c? \in \textit{available} \\
\#(\textit{instance\_of}^{\sim}(\!| \{\textit{instance\_of}(c?)\} |\!)) = 1 \\
\textit{available}' = \textit{available} \setminus \{c?\} \\
\textit{instance\_of}' = \{c?\} \mathbin{⩤} \textit{instance\_of} \\
\textit{written\_by}' = \{\textit{instance\_of}(c?)\} \mathbin{⩤} \\
\quad \textit{instance\_of} \\
\textit{about}' = \{\textit{instance\_of}(c?)\} \mathbin{⩤} \textit{about} \\
\hline
\end{array}$$

1.7 Interrogating the Database

- Two options:
  - search by author;
  - search by subject;
  - find out what copies someone has borrowed.

- *ByAuthor* takes an author name and produces the set of all books that the author appeared in the 'author' list of.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{ByAuthor}} \\
\Xi \textit{Library} \\
a? : \textit{AUTHOR} \\
out! : \mathbb{P}\,\textit{BOOK} \\
\hline
out! = \{ b : \textit{BOOK} \mid a? \in \textit{written\_by}(x) \}
\end{array}$$

- *BySubject* takes a set of subjects and produces a list of all the books which have these subjects in their 'about' list.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{BySubject}} \\
\Xi \textit{Library} \\
s? : \mathbb{P}\,\textit{SUBJECT} \\
out! : \mathbb{P}\,\textit{BOOK} \\
\hline
out! = \{ b : \textit{BOOK} \mid s? \subseteq \textit{about}(b) \}
\end{array}$$

- Finally, finding out who has borrowed what…

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{BooksBorrowedBy}} \\
\Xi \textit{Library} \\
n? : \textit{PERSON} \\
out! : \mathbb{P}\,\textit{COPY} \\
\hline
n? \in \textit{borrower} \\
out! = \textit{borrowed\_by}^{\sim}(\!| \{ n? \} |\!)
\end{array}$$