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:

  \[ \text{[BOOK, COPY, PERSON, AUTHOR, SUBJECT, REPORT]} \]
1.2 State Space

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

\[
\begin{align*}
\text{ParaLibrary} & : \text{COPY} \rightarrow \text{BOOK} \\
\text{written_by} & : \text{BOOK} \rightarrow \text{AUTHOR} \\
\text{about} & : \text{BOOK} \rightarrow \text{SUBJECT} \\
\text{dom} \ x & \subseteq \text{ran} \ y & \Rightarrow & \text{dom} \ y \subseteq \text{ran} \ x
\end{align*}
\]
• `instance_of` tells us what book a copy is an instance of;

• the set \( \text{ran} \ instance\_of \)
  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:

\[
\begin{align*}
\text{LibraryDB} & \quad \underline{borrower, staff : } \mathbb{I} \text{PERSON} \\
& \quad \underline{available, out : } \mathbb{I} \text{COPY} \\
& \quad \underline{borrowed\_by : } \text{COPY } \rightarrow \text{PERSON}
\end{align*}
\]

\[
\begin{align*}
\text{borrower } \cap \text{staff} & = \emptyset \\
\text{available } \cap \text{out} & = \emptyset \\
\text{dom } \text{borrowed\_by} & = \text{out} \\
\text{ran } \text{borrowed\_by} & \subseteq \text{borrower} \\
\forall p : \text{borrower} \bullet \#\text{borrowed\_by}\sim(\{p\}) & \leq \text{MaxCopies}
\end{align*}
\]
• 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
dom instance_of = available ∪ 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 \((c?)\).

\[
\text{CheckOut} \\
\Delta\text{Library} \\
n? : \text{PERSON} \\
c? : \text{COPY} \\
\]

\[
\begin{align*}
n? & \in \text{borrower} \\
c? & \in \text{available} \\
\#\text{borrowed\_by} \sim & \(|\{n?\}| \\
< & \text{MaxCopies} \\
\text{available'} = & \text{available} \setminus \{c?\} \\
\text{out'} = & \text{out} \cup \{c?\} \\
\text{borrowed\_by'} = & \text{borrowed\_by} \cup \\
\{c? \mapsto n?\}
\end{align*}
\]
(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.

\[
\text{Return}
\]
\[
\Delta \text{Library} \\
c? : \text{COPY}
\]
\[
c? \in \text{out}
\]
\[
\text{available}' = \text{available} \cup \{c?\}
\]
\[
\text{out}' = \text{out} \setminus \{c?\}
\]
\[
\text{borrowed_by} = \{c?\} \sqsubseteq \text{borrowed_by}
\]
• 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,

\[
\text{borrowed}_\text{by} = \{b01 \mapsto mjw, b02 \mapsto en, b03 \mapsto mjw\}
\]

\[
\{b01\} \triangleleft \text{borrowed}_\text{by} = \{b02 \mapsto en, b03 \mapsto mjw\}
\]
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.
AddNewBook

$\Delta Library$

$c? : COPY$

$b? : BOOK$

$a? : IP AUTHOR$

$s? : IP SUBJECT$

$\begin{align*}
  b? &\not\in \text{ran } \text{instance\_of} \\
  c? &\not\in \text{available} \cup \text{out}
\end{align*}$

\begin{align*}
  \text{available}' &= \text{available} \cup \{c?\} \\
  \text{instance\_of}' &= \text{instance\_of} \cup \{c? \mapsto b?\} \\
  \text{written\_by}' &= \text{written\_by} \cup \{b? \mapsto a?\} \\
  \text{about}' &= \text{about} \cup \{b? \mapsto s?\}
\end{align*}
AddAnotherCopy

$\Delta Library$

$c? : \text{COPY}$

$b? : \text{BOOK}$

$c? \not\in \text{available} \cup \text{out}$

$b? \in \text{ran instance_of}$

$\text{available}' = \text{available} \cup \{c?\}$

$\text{instance_of}' = \text{instance_of} \cup \{c? \mapsto b?\}$
1.6 Removing Books

• Removing a book 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.
\[
\text{RemoveOther} \\
\Delta \text{Library} \\
c? : \text{COPY} \\
c? \in \text{available} \\
\#(\text{instance_of}^\sim(|\{\text{instance_of}(c?)\}|)) > 1 \\
\text{available}' = \text{available} \setminus \{c?\}
\]

- Note that there is no need to alter any variables in \textit{ParaLibrary}; we only change \textit{available}, to indicate that the book is no longer available.
RemoveLast  
\[ \Delta \text{Library} \]
\[ c? : \text{COPY} \]
\[ c? \in \text{available} \]
\[ \#(\text{instance_of} \sim (\{|\{\text{instance_of}(c?)\}|))) = 1 \]
\[ \text{available}' = \text{available} \setminus \{c?\} \]
\[ \text{instance_of}' = \{c?\} \triangleleft \text{instance_of} \]
\[ \text{written_by}' = \{\text{instance_of}(c?)\} \triangleleft \text{instance_of} \]
\[ \text{about}' = \{\text{instance_of}(c?)\} \triangleleft \text{about} \]
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.

\[ \text{ByAuthor} \]
\[ \begin{array}{l}
\exists \text{Library} \\
\ a? : \text{AUTHOR} \\
\ out! : \mathbb{P} \text{BOOK} \\
\ out! = \{b : \text{BOOK} \mid a? \in \text{written_by}(x)\}
\end{array} \]

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

\[ \text{BySubject} \]
\[ \begin{array}{l}
\exists \text{Library} \\
\ s? : \mathbb{P} \text{SUBJECT} \\
\ out! : \mathbb{P} \text{BOOK} \\
\ out! = \{b : \text{BOOK} \mid s? \subseteq \text{about}(b)\}
\end{array} \]
• Finally, finding out who has borrowed what...

\[
\text{BooksBorrowedBy} \equiv \begin{cases}
\text{Library} \\
n? : \text{PERSON} \\
\text{out!} : \text{COPY}
\end{cases}
\]

\[
n? \in \text{borrower} \\
\text{out!} = \text{borrowed\_by}^\sim(\{n?\})
\]