

B Exercises 1

Sets

1. Given $X = \{a, b\}$, show the following.
 - (a) $\text{card}(X)$
 - (b) if $aa : X$, **one** possible value of aa
 - (c) $\mathbb{P}(X)$, $\text{card}(\mathbb{P}(X))$.
(The set of elements taken none at a time, one at a time, and two at a time.)
 - (d) if $aaa : \mathbb{P}(X)$, **one** possible value of aaa
 - (e) $\mathbb{P}(\mathbb{P}(X))$, $\text{card}(\mathbb{P}(\mathbb{P}(X)))$.
(The set of things taken none at a time (empty set, shown), one at a time (set containing the empty set (shown), then a set of each of the remaining elements of part b one at a time (shown), then sets of the elements of part b taken two at a time (not shown), three at a time (not shown), and finally four at a time (shown).)
 - (f) $\mathbb{P}(\mathbb{P}(\mathbb{P}(X)))$, $\text{card}(\mathbb{P}(\mathbb{P}(\mathbb{P}(X))))$.
 - (g) $X \times \{0, 1\}$, $X \times \{\}$
2. Given $X = \{a, b\}$ and $Y = \{0, 1\}$, show as sets of maplets:
 - (a) $X \leftrightarrow Y$
 - (b) $X \rightarrow Y$

What are the values of the following predicates?

 - (c) $a \mapsto 0 \in X \leftrightarrow Y$
 - (d) $\{\{a \mapsto 0, a \mapsto 1\}\} \subseteq X \rightarrow Y$
3. Students pass a subject if they gain at least 50 marks in the final examination. Given a function $results : \text{STUDENTS} \rightarrow \mathbb{N}$, that yields the examination result for a particular student, specify
 - (a) the set of students that pass;
 - (b) the set of students that fail.
4. Memory on a modern computer can be considered to be an array of bytes.
 - (a) define sets to represent memory.
 - (b) show a function application to represent memory lookup of location loc .
 - (c) show a function update (relational override) to represent assignment of value val to memory location loc

5. Assuming the following C declarations

```
int a[4] = {5,2,8,9};
char b[] = "the";
struct {int ca; char cb;} c;
```

give a set representation of the data structures *a*, *b*, *c*

6. If we were modelling a taxi fleet company we might have three variables, *drivers*, *taxis* and *assigned* constrained by

$$\begin{aligned} \text{drivers} & : \mathbb{P} \text{DRIVERS} \\ \text{taxis} & : \mathbb{P} \text{TAXIS} \\ \text{assigned} & : \text{drivers} \rightsquigarrow \text{taxis} \end{aligned}$$

where *drivers* is the set of drivers working for the company, *taxis* is the set of taxis owned by the company, and *assigned* is a function recording the assignment of drivers to taxis.

The arrow \rightsquigarrow denotes a *partial injective* function. An injective function is a one-to-one function.

- Why is *assigned* a partial function?
 - Why is *assigned* an injective function?
 - Specify the drivers who are currently assigned.
 - Specify the drivers who are currently unassigned.
 - Specify the taxis that are currently assigned.
 - Specify the taxis that are currently unassigned.
7. In the **SimpleLibrary** machine, we have three variables

$$\begin{aligned} \text{books_in_library} & \subseteq \text{BOOK} \\ \text{books_on_shelf} & \subseteq \text{books_in_library} \\ \text{books_on_loan} & \in \text{books_in_library} \rightsquigarrow \text{users} \end{aligned}$$

We want *books_on_shelf* and $\text{dom}(\text{books_on_loan})$ to *partition* the set *books_in_library*. Normally, we would specify this by:

$$\begin{aligned} \text{dom}(\text{books_on_loan}) \cup \text{books_on_shelf} & = \text{books_in_library} \wedge \\ \text{dom}(\text{books_on_loan}) \cap \text{books_on_shelf} & = \{\} \end{aligned}$$

but it was specified by

$$\text{dom}(\text{books_on_loan}) = \text{books_in_library} - \text{books_on_shelf}$$

- Show that the former follows from the latter.
 - Would $\text{dom}(\text{books_on_loan}) \cup \text{books_on_shelf} = \text{books_in_library}$ have been equally effective.
8. For the Borrow(*user*,*book*) operation of the **SimpleLibrary** machine we get a proof obligation to show that

$$\text{dom}(\text{books_on_loan} \cup \{\text{book} \mapsto \text{user}\}) = \text{books_in_library} - (\text{books_on_shelf} - \{\text{book}\})$$

Reason why this is true.