---

1. Develop a Z specification for bank accounts. Include operations to make deposits and withdrawals, open and close an account, and query the balance of an account.

2. Model the booking system for a particular flight. The aeroplane has a limited capacity and the flight should not be overbooked. The operations should include: a passenger booking onto the flight; a passenger cancelling his booking; a query of how many seats are still available; the production of a passenger list. Use the given set [PERSON].

3. A certain brokerage firm specialises in buying and selling companies. At a given time, its portfolio of investments is the set of companies it owns. Based on the State Schema

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{Portfolio}} \\
\hline
portfolio : \mathbb{P}\, COMPANY \\
portfolioSize : \mathbb{Z} \\
\hline
portfolioSize = \# portfolio \\
\hline
\end{array}
$$

write down schemas for `(i)` buying a company, `(ii)` selling a company, `(iii)` a combined operation where we buy one company and sell a different one.

4. An antique shop buys and sells items. Model the collection of antiques in the shop at a given time, and describe operations to buy and sell antiques, using the basic type $[ANTIQUE]$.

5. In the parking example discussed in class, with basic types $[PERSON, PARKING\_SLOT]$ our State Schema had the form

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{Parking}} \\
\hline
staff : \mathbb{P}\, PERSON \\
assigned\_to : PARKING\_SLOT \leftrightarrow PERSON \\
available, occupied : \mathbb{P}\, PARKING\_SLOT \\
\hline
\operatorname{ran} assigned\_to \subseteq staff \\
available \cap occupied = \emptyset \\
\hline
\end{array}
$$

Write schema (with appropriate error handling) to `(i)` find all parking slots assigned to a particular person, `(ii)` find all staff that share a particular parking slot.

6. In a certain US city, parking permission requires that the car is registered and has a parking permit. A parking permit is only issued to registered cars. Starting with the specification

$[CAR]$

$STATUS ::= noRegistration \mid registeredCar \mid permitHolder$

$$REPORT ::= success \mid alreadyRegistered \mid permitCancelledAlso \mid notRegistered \\ \mid noPermitsLeft \mid permitAlready \mid notPermitHolder$$

$$\mid \; permitLimit : \mathbb{N}$$

and the State Schema

$$\begin{array}{|l}
\hline
\textit{Reg} \\
registered, permits : \mathbb{F}\, CAR \\
\hline
permits \subseteq registered \\
\#permits \leq permitLimit \\
\hline
\end{array}$$

develop schema for each of the following ($\mathbb{F}$ above stands for a **finite subset**):

(a) Registration of a car. Write here 3 schema $RegOK$, $Success$, $AlreadyRegistered$ with the overall structure

$$Registration \mathrel{\widehat=} RegOK \wedge Success \vee AlreadyRegistered$$

(b) Deregistration of a car (4 schema: $DeregOK$, $Success$, $DeregPlusPermit$, $NotRegistered$)

$$DeRegistration \mathrel{\widehat=} DeregOK \wedge Success \vee DeregPlusPermit \vee NotRegistered$$

(c) Issuing of a permit:

$$Permit \mathrel{\widehat=} PermitOK \wedge Success \vee NotRegistered \vee PermitAlready \vee NoPermitsLeft$$

(d) Cancelling a permit:

$$CancelPermit \mathrel{\widehat=} CancelPermitOK \wedge Success \vee NotPermitHolder \vee NotRegistered$$

(e) Enquiring about the status of a car:

$$Enquire \mathrel{\widehat=} (RegisteredOnly \vee PermitHolder \vee Unregistered) \wedge Success$$