

## Schemas

**Exercise 11.1** (Dates) The schema type *Date* describes all the bindings of *day* and *month* that correspond to valid calendar dates. We may define a new schema type which includes a component *year*. If the value of *year* lies in the set *Leap*, then there are 29 days in February; otherwise, there are only 28. In leap years, Katie has a birthday party on the 29th of February; in other years, she has a party on the 28th.

- (a) using a schema called *FullDate*, describe all valid combinations of *day*, *month*, and *year*.
- (b) using a set comprehension, define the set of all dates on which Katie might hold a birthday party.

□

**Exercise 11.2** (Access control) An access control system is used to determine whether clients can connect to a server. The system is *host-based*: whether or not a client can connect will depend upon which machine, or *host*, it is associated with.

When it is switched on, the system maintains a list of permitted *hosts*: machines whose clients will be allowed to connect. This will normally include *local*: the name of the machine on which the system and server are running.

We will use a basic type *Host* to denote the set of all machines, and we will define a free type to record the status:

$$\textit{Status} ::= \textit{on} \mid \textit{o}$$

to record its status.

- (a) using a schema type called *AccessControl*, describe the set of all possible states of the access control system.
- (b) write a schema *EmptyList* to describe the states of the system in which all connections are blocked.
- (c) write a schema *NoRemoteAccess* in which connections will be accepted only from the local machine.

□

**Exercise 11.3 (Supermarket)** A supermarket monitoring system keeps track of the queues forming at the checkouts. It maintains a set *checkouts* of checkout positions in the store; this is a subset of basic type *Checkout*. If a checkout is in use, then the system records the *queue* of people standing there. The basic type *Person* represents the set of all potential customers. An invariant of the system state is that no person can be standing in two different queues at the same time.

- (a) using a schema called *SuperMarket*, describe the set of all possible states of the monitoring system.
- (b) using a schema called *NoMore*, characterise those states of the system in which there are no more checkouts to be opened
- (c) using a schema called *TooBusy*, characterise those states of the system in which every checkout that is in use has more than three people in its queue.
- (d) write a predicate which asserts that there is a possible state of the system in which all checkouts are in use and yet there are more than three people queueing at each.

□