

## Schema Operators

**Exercise 12.1** (Normalisation) In the text, we saw that normalisation may be required if we are to obtain the correct expansion of a schema negation. For example, if  $R$  is the schema defined by

$R$
$a : \mathbb{N}$
$a \leq 3$

then the expression  $\neg R$  expands to

$a : \mathbb{Z}$
$a \notin \mathbb{N} \vee a > 3$

The constraint that  $a \in \mathbb{N}$ , made explicit by normalisation, must be included in the negation.

- (a) Explain why normalisation may also be required in the expansion of a schema disjunction  $A \vee B$
- (b) Discuss whether normalisation is ever required in the expansion of a schema conjunction  $A \wedge B$ .

□

**Exercise 12.2** (Operators) The schemas  $S$ ,  $T$ ,  $U$ , and  $V$  are defined by

$$\frac{S}{\begin{array}{l} a, b : \mathbb{N} \\ \hline a \leq b \end{array}}$$

$$\frac{T}{\begin{array}{l} a, b : \mathbb{N} \\ \hline a \geq b \end{array}}$$

$$\frac{U}{\begin{array}{l} a : \mathbb{N} \\ c : \mathbb{P}\mathbb{N} \\ \hline a \in c \end{array}}$$

$$\frac{V}{\begin{array}{l} a, c : \mathbb{N} \\ \hline a = c \end{array}}$$

Expand and simplify each of the following expressions:

- (a)  $S \wedge T$
- (b)  $S \vee T$
- (c)  $S \wedge U$
- (d)  $S \vee U$
- (e)  $U \wedge V$
- (f)  $\neg S$
- (g)  $S \wedge S'$
- (h)  $[S; U \mid b \in c]$

□

**Exercise 12.3** (Conventions) Suppose that  $W$ ,  $X$ , and  $Y$  are defined by

$$\frac{\begin{array}{l} W \\ \Delta S \\ x? : \mathbb{N} \end{array}}{x? \leq a \wedge a' = x? \wedge b' = b}$$

$$\frac{\begin{array}{l} X \\ \Delta S \\ x? : \mathbb{N} \end{array}}{x? > b \wedge a' = a \wedge b' = x?}$$

$$\frac{\begin{array}{l} Y \\ \exists S \\ y! : \mathbb{N} \end{array}}{y! = a}$$

Expand and simplify the following schemas:

- (a)  $W$
- (b)  $Y$
- (c)  $W \vee X$
- (d)  $W \wedge X$
- (e)  $W \setminus (a', b')$

□

**Exercise 12.4** (Composition) Suppose that the state schema  $S$  is defined by

$$\boxed{\begin{array}{l} S \\ \hline x : \mathbb{N} \end{array}}$$

and that the operation schemas  $Op1$  and  $Op2$  are defined by

$$\boxed{\begin{array}{l} Op1 \\ \hline \Delta S \\ \hline x' = x + 1 \vee x' = 0 \end{array}}$$

$$\boxed{\begin{array}{l} Op2 \\ \hline \Delta S \\ \hline x \neq 0 \wedge x' = x - 1 \end{array}}$$

Simplify the following schema compositions:

- (a)  $Op1 \ ; \ Op2$
- (b)  $Op2 \ ; \ Op1$
- (c)  $Op1 \ ; \ Op1$
- (d)  $Op2 \ ; \ Op2$

□

**Exercise 12.5** (Implication) Apart from disjunction, conjunction, negation, and quantification, there are two other operators in our predicate calculus: implication  $\Rightarrow$  and equivalence  $\Leftrightarrow$ . Using schemas from the above examples if necessary, explain how we might interpret the following schema expressions:

- (a)  $S \Rightarrow T$
- (b)  $S \Leftrightarrow T$

□

**Exercise 12.6** (Access control) At initialisation, the access control system is switched off, and the list of hosts is empty. The system may be switched on at any time, even if it is already on; it may also be switched off at any time. Switching it on has no effect upon the list of permitted hosts, switching it off will empty the list.

Each machine is identified by a name. If *Hosts* denotes the set of all machines and *Names* denotes the set of all names, then the association between names and machines is modelled by a function

$$\mid \text{lookup} : \text{Names} \rightarrow \text{Hosts}$$

The information recorded in *lookup*, and only this information, is available to the access control system.

At any time, the user may add a set of machines to the list of permitted hosts. To do this, he or she provides a collection of names. Some of these may not correspond to any known machine: these will be ignored without comment. Wherever a name corresponds to a machine, according to *lookup*, that machine is added to the list of permitted hosts.

Similarly, the user may remove a set of machines from the list of permitted hosts. If the system is switched off when the user attempts to add or remove hosts, then it is automatically switched on: there is no need to use the *On* operation first.

- (a) write down a schema called *AccessControlInit* to represent the initialisation of the system.
- (b) define two operation schemas *On* and *Off* to represent the effects of switching the system on, and switching it off, respectively.
- (c) define two operation schemas *Add* and *Remove* that model the effects of adding and removing a collection of named hosts, respectively.

□

**Exercise 12.7** (Supermarket) A person who has finished shopping may join any checkout queue, provided that the checkout in question is currently in service. Furthermore, any person who is not currently at the head of a queue may choose to leave a queue and join another.

- (a) describe the effect upon the state of the monitoring system that occurs when a person, previously shopping, joins a queue. This operation should have two inputs: a person and a queue.
- (b) define partial operation schemas to cover the situations in which an attempt to tell the system that a person has joined a queue might fail.

- (c) describe the effect of a customer leaving one queue and joining another. This operation should have three inputs: a person and two queues.
- (d) define partial operation schemas to cover the situations in which an attempt to tell the system that a person has changed queues might fail.
- (e) define two total operations—*JoinQueue* and *ChangeQueue*—that describe the change in state that accompanies an attempt to join a queue, and an attempt to move from one queue to another.

□