4.

$u \frown v \upharpoonright \{ \text{London, Moscow, Paris, Rome} \}$
$= \langle \text{London, Paris} \rangle$

5.

$\text{tail } (u \frown v) \frown \text{front} \langle \text{Moscow, Berlin, Warsaw} \rangle$
$= \langle \text{Amsterdam, Madrid, Paris, Frankfurt, Moscow, Berlin} \rangle$

6. The stream either consists of a subsequence before, then the pattern, then a subsequence after and *pos* is the index where the pattern starts, or it does not and *pos* is zero.

7.

Delete

$\Delta TEXT$
$pat?: \quad \text{seq CHAR}$
$pos!: \quad \mathbb{N}$

$((\exists\ before, after: \text{seq CHAR} \bullet before \frown pat? \frown after = stream) \wedge pos! = \#before + 1 \wedge stream' = before \frown after)$
$\vee$
$(\neg(\exists\ before, after: \text{seq CHAR} \bullet before \frown pat? \frown after = stream) \wedge$
$pos! = 0 \wedge stream' = stream)$

Chapter 13

1.

FileSys

$file: \quad \text{seq BYTE}$

2.

Init

$\Delta FileSys$

$file' = \langle \rangle$

3.

Insert

$\Delta FileSys$
$pat?: \quad \text{seq BYTE}$
$pos!: \quad \mathbb{N}$

$1 \leq pos \leq \#file + 1$
$(\exists\ before, after: \text{seq BYTE} \bullet before \frown after = file) \wedge$
$pos! = \#before + 1 \wedge file' = before \frown pat? \frown after)$

4.

Delete

$\Delta FileSys$
$beg?, end?: \quad \mathbb{N}$

$1 \leq beg? \leq end? \leq \#file + 1$
$(\exists\ before, after, del: \text{seq BYTE} \bullet before \frown del \frown after = file) \wedge$
$beg? = \#before + 1 \wedge \#del = end? - beg?$
$file' = before \frown after)$

5.

Copy

$\Xi FileSys$
$beg?, end?: \quad \mathbb{N}$
$buf!: \quad \text{seq BYTE}$

$1 \leq beg? \leq end? \leq \#file + 1$
$(\exists\ before, after: \text{seq BYTE} \bullet before \frown buf! \frown after = file) \wedge$
$beg? = \#before + 1 \wedge \#buf! = end? - beg?)$

Chapter 14

1.

$[APARTMENT, TIMESLOT, PERSON]$

TimeShareCo

$owned: \mathbb{P}APARTMENT$
$customers: \mathbb{P}PERSON$
$booked: APARTMENT \nrightarrow (TIMESLOT \nrightarrow PERSON)$

$dom\ booked = owned$
$\forall apart: APARTMENT \mid apart \in owned \bullet$
$ran\ (booked\ a) \subseteq customers$

Init

$\Delta TimeShareCo$

$owned' = \varnothing$
$customers' = \varnothing$
$booked' = \varnothing$

2.

AddCustomer

$\Delta TimeShareCo$
$p?: PERSON$

$p? \notin customers$
$customers' = customers \cup \{p?\}$
$owned' = owned$
$booked' = booked$

3.

AddApartment

$\Delta TimeShareCo$
$ap?: APARTMENT$

$ap? \notin owned$
$owned' = owned \cup \{ap?\}$
$customers' = customers$
$booked' = booked \cup \{ap? \mapsto \varnothing\}$

4.

MakeBooking

$\Delta TimeShareCo$
$ap?: APARTMENT$
$p?: PERSON$
$t?: TIMESLOT$

$ap? \in owned$
$p? \in customers$
$t? \notin dom\ (booked\ ap?)$
$booked' = booked \oplus (ap? \mapsto (booked\ ap? \cup (t? \mapsto p?)))$
$owned' = owned$

5.

FreeApartments

$\Xi TimeShareCo$
$t?: TIMESLOT$
$free!: \mathbb{P}APARTMENT$

$ap? \in owned$
$free! = \{apart: APARTMENT \mid$
$apart \in owned \land t? \notin dom\ (booked\ apart) \bullet apart\}$

6.

(a)

- *position* maps an aircraft to its current position.
- *minsLeft* maps an aircraft to its remaining flying time (in minutes).
- *flightMins* maps each aircraft to a mapping that gives the time for the aircraft to fly to a given point (in minutes).

For each of the above the aircraft concerned are the REGA helicopters.

(b)

$ran\ position \subseteq Switzerland$

(c)

$\forall ch: POINT \mid inCH \in Switzerland \bullet$
$(\exists heli: AIRCRAFT \mid heli \in helis \bullet$
$(flightMins\ heli)\ inCH \leq 15)$

(d)

$ran\ position \subseteq ran\ base$

(e)

$position = base$

(f)

$$\begin{array}{l}
\forall \text{heli: AIRCRAFT} \mid \text{heli} \in \text{helis} \bullet \\
\quad (\exists \text{basis: POINT} \mid \text{basis} \in \text{ran base} \bullet \\
\quad\quad (\text{flightMins heli})\ \text{basis} \leq \text{minsLeft(heli)})
\end{array}$$