NUMBERS & SEQUENCES

This page is intentionally blank

☺

NUMBERS

- Our previous study has shown that *sets* may be employed to describe *unindexed* collections of objects

- A *sequence* associates (or, labels) each of the objects in a collection with a natural number - thus indexing each object

- Computing embraces sequences in many forms: *files*, *lists*, *arrays*, etc

- The integer sets (i.e. $\mathbb{Z}$ and its subsets $\mathbb{N}$ and $\mathbb{N}_1$) could, therefore, prove useful in defining such *ordered lists* or *sequences*

- Note that the integer set, $\mathbb{Z}$, is the **only** basic type automatically provided in the standard Z language

ARITHMETIC RELATIONS

- The Z specification language permits the usual (mostly *infix*) arithmetic operators to specify particular relations between integers:

Operation	Operator	Example
addition	_ + _	5 + 3 = 8
subtraction	_ - _	5 - 3 = 2
multiplication	_ * _	5 * 3 = 15
division	_ div _	5 div 3 = 1
modulo arithmetic	_ mod _	5 mod 3 = 2
negation	-	- (-2) = 2

- Note that the names of infix relations (operators) are, conventionally, prefixed and suffixed with underscores (which behave like *placeholders*)
    - Thus, the relation: $\_ + \_ : \mathbb{Z} \times \mathbb{Z}$
    permits the use of expressions such as 6+3 to mean the same as $(6, 3) \in \_ + \_$

COMPARISON RELATIONS

- The standard comparison relations (again infix) are:

Operation	Operator	Example
less than	$\_ < \_$	$3<5$, $\neg 5<3$
less than or equal	$\_ \leq \_$	$3 \leq 3$
greater than	$\_ > \_$	$a>b \Leftrightarrow b<a$
greater than or equal	$\_ \geq \_$	$a \geq b \Leftrightarrow b \leq a$
equal	$\_ = \_$	$5 - 3 = 2$

- And there are functions to yield *maximum* and *minimum* values contained in sets:

$$max\ \{3, 4, 5, 8\} = 8$$
$$min\ \{3, 4, 5, 8\} = 3$$

  - *max* and *min* should only be applied to **non-empty** sets

DEFINING OPERATORS

- The Z language permits the definition of other operators as appropriate

- For example, if *abs* is a function to return the *absolute value* of an integer then we may define *abs* by using an *axiomatic description*:

$$
\begin{array}{|l}
abs : \mathbb{Z} \rightarrow \mathbb{Z} \\
\hline
\forall n : \mathbb{Z} \bullet \\
\qquad n < 0 \Rightarrow abs\ n = -n \land \\
\qquad n \geq 0 \Rightarrow abs\ n = n
\end{array}
$$

- Note that: $\mathbf{ran}\ abs = \mathbb{N}$

SUCCESSOR FUNCTION

- When applied to a member of the set of natural numbers ($\mathbb{N}$), the *successor* function (*succ*) returns the next higher number in $\mathbb{N}$:

  i.e. $succ = \{ 0 \mapsto 1, 1 \mapsto 2, 2 \mapsto 3, ... \}$

  and $\mathbf{ran}\ succ = \mathbb{N}_1$

- Note that the element *zero* (written usually as 0) is *not* the successor of any other natural number

- According to this definition, *succ* is a *total bijection* from $\mathbb{N}$ to $\mathbb{N} \setminus \{0\}$ (i.e. from $\mathbb{N}$ to $\mathbb{N}_1$)

- Functions such as *succ* are known as *constructors*

PREDECESSOR FUNCTION

- The inverse of *succ* is the predecessor function; often written as *pred*

- The Z specification language does not have *pred* as an intrinsic function; but it is easy to derive *pred* from *succ* since we can write:

$$pred == succ^{\sim} \qquad (\text{ or } succ^{-1})$$

NUMBER RANGE

- A *number range* is a set of numbers between two cited integers

- If $a, b : \mathbb{Z}$, then $a \mathrel{..} b$ denotes the number range from a to b **inclusive**:

  i.e. $a \mathrel{..} b = \{ a, a+1, a+2, \ldots ,b-2, b-1, b \}$

  or, more formally:

  $$a \mathrel{..} b = \{ n : \mathbb{Z} \mid n \geq a \land n \leq b \}$$

- NB: if $a > b$ then $a \mathrel{..} b = \{\} = \varnothing$
  if $a = b$ then $a \mathrel{..} a = \{a\}$

CARDINALITY & FINITE SETS

- A powerset may contain infinite subsets
  - as an example, we have $\mathbb{N} \in \mathbb{P}\ \mathbb{N}$

- If we are only interested in **finite** subsets then Z allows a special 'powerset' notation to be used:
  - If S is a set, then $\mathbb{F}$ S denotes the set of all *finite* subsets of S

- We recall that the *cardinality* of a *finite* set, S, is the *number of elements* in the set (or *size* of the set) and is written #S

- For a set to be *finite*, it must be possible to map from a natural number in the range 1..*n* uniquely onto each element in that set

CARDINALITY & FINITE SETS

- Recall that $X \pfun Y$ denotes the partial function from X to Y (each element of the domain maps to, at most, one element of the range)

- More precisely,

$$X \pfun Y == \{ f : X \leftrightarrow Y \mid (\forall x : X; y_1, y_2 : Y \bullet$$
$$(x \mapsto y_1 \in f) \land (x \mapsto y_2 \in f) \Rightarrow (y_1 = y_2)\}$$

- $X \ffun Y$ denotes the corresponding set of *finite* partial functions where the domain is a finite subset of X

- We have:

$$X \ffun Y == \{ f : X \pfun Y \mid \mathbf{dom}\, f \in \mathbb{F}\, X \}$$

or:

$$X \ffun Y == (X \pfun Y) \cap \mathbb{F}\,(X \times Y)$$

SEQUENCES

- *Lists*, *arrays*, *files*, *sequences*, *trace histories* are all different names for an important data type with indexed elements where, normally, the indexes are contiguously numbered from 1

- In Z, such data types are known as *sequences*

- A sequence of elements of type T is a function from the natural numbers to the elements of T

- We define a sequence, *s*, which has elements of type T, by:

  $$s : \mathbf{seq}\ \mathrm{T}$$

- This is equivalent to declaring the function:

  $$s : \mathbb{N} \rightarrow\!\!\!\!\!+\ \mathrm{T}$$

  subject to the constraint: $\mathbf{dom}\ s = 1 \mathrel{..} \#s$

  i.e. $\mathbf{seq}\ \mathrm{T} == \{\ s : \mathbb{N} \rightarrow\!\!\!\!\!+\ \mathrm{T} \mid \mathbf{dom}\ s = 1 \mathrel{..} \#s\ \}$

SEQUENCES

- Sequences from the set of *finite* sequences defined by:

  $$\mathbf{seq}\ T == \{\ s : \mathbb{N} \twoheadrightarrow\!\!\!\!\!\!\!+\ \ T \mid \mathbf{dom}\ s = 1\ ..\ \#s\ \}$$

  are often called T-valued sequences or T sequences

- A *sequence constant* can be constructed by listing its elements in order:

  e.g. for: [TOWN] the set of possible towns
  with: *flight*: **seq** TOWN

  we might have:

  $$flight = \langle \text{London, Moscow, Tashkent, Delhi} \rangle$$

  which is equivalent to the function:

  $$flight = \{1 \mapsto \text{London}, 2 \mapsto \text{Moscow}, 3 \mapsto \text{Tashkent}, 4 \mapsto \text{Delhi}\}$$

SEQUENCES

- The **length** of a sequence is simply the number of elements it contains (i.e. its *cardinality*)

  e.g. the length of *flight* is: $\#flight$ (= 4)

- A sequence of zero length has no elements and is written as $\langle\ \rangle$, whatever its type

- If it is known that a sequence, s, will always be non-empty (i.e. have, at least, one element) it can be declared by: s: $\mathbf{seq}_1$ T

  which is equivalent to: s: **seq** T
  with the constraint $\#s > 0$

SEQUENCE DECOMPOSITION

- *Selection*

  Since a sequence is a function, a specific element may be selected by *function application*

  e.g. the **second** element of *flight* is *flight* 2 (i.e. Moscow)

- *Head*

  The first element in a sequence is known as the *head* of the sequence

  i.e. *flight* 1 is equivalent to **head** *flight*

- *Tail*

  The *tail* of a sequence is the sequence with its *head* removed

  e.g. **tail** *flight* = ⟨Moscow,Tashkent, Delhi⟩

SEQUENCE DECOMPOSITION

- *Last*

  The *last* of a sequence is its last element

  e.g. $\textbf{last}\ flight = flight\ \#flight$ (i.e. Delhi)

- *Front*

  The *front* of a sequence is that same sequence with its last element removed

  e.g. $\textbf{front}\ flight$ is $\langle$London, Moscow, Tashkent$\rangle$

SEQUENCE CONSTRUCTORS

- *Concatenation* or *Catenation*

  Concatenation chains two sequences to form a new sequence. The symbol used is $\frown$ (or, sometimes, ^ or ‖) - read as *concatenated with* or *catenated with*

  e.g. $\langle$London,Moscow,Tashkent,Delhi$\rangle \frown \langle$Bangkok,Taipei$\rangle$
  = $\langle$London,Moscow,Tashkent,Delhi,Bangkok,Taipei$\rangle$

- *Reverse*

  The *reverse* of any sequence, *s*, is written **rev** *s* and is a sequence with the elements of *s* in reverse order

  - If $s = \langle\ \rangle$ then $\mathbf{rev}\ s = \langle\ \rangle$

  - A more formal view of **rev** *s* might be:

$$
\begin{array}{|l}
rev : \mathbf{seq}\ \mathrm{T} \rightarrow \mathbf{seq}\ \mathrm{T} \\
\hline
\forall\ s : \mathbf{seq}\ \mathrm{T} \bullet \\
\qquad (s = \langle\rangle \Rightarrow rev\ s = \langle\rangle \qquad \wedge \\
\qquad s \neq \langle\rangle \Rightarrow rev\ s = \langle \mathbf{last}\ s \rangle \frown rev\ \mathbf{front}\ s)
\end{array}
$$

SEQUENCE CONSTRUCTORS

- *Squash*

  If f is a finite function and **dom** $f \in \mathbb{Z}$, then **squash** f is the sequence formed by arranging the elements of **ran** f in ascending order of their corresponding domain values

- Examples:

  - If f is the function

    $$\{ 8 \mapsto l, -7 \mapsto e, 3 \mapsto f, 1 \mapsto i, 6 \mapsto f, 7 \mapsto e\}$$

    then **squash** f is the sequence

    $$\langle e, i, f, f, e, l\rangle$$

  - Suppose, $marks : \mathbb{N} \twoheadrightarrow \text{Names}$

    where $marks ==$

    $\{43 \mapsto \text{Gary}, 37 \mapsto \text{Tom}, 76 \mapsto \text{Jane}, 55 \mapsto \text{Mike}, 81 \mapsto \text{Seth}\}$

    then **squash** *marks* is the sequence:

    $$\langle \text{Tom}, \text{Gary}, \text{Mike}, \text{Jane}, \text{Seth}\rangle$$

SEQUENCE CONSTRUCTORS

- *Extraction (Index Restriction)*

  If S is a sequence and X is a finite set of integers, then $X \upharpoonleft S$ denotes a sequence restricted to those elements that have indexes in X

  e.g. If X == {1,4} then since

  $$flight = \langle \text{London,Moscow,Tashkent,Delhi} \rangle$$

  we recover:

  $$X \upharpoonleft flight = \langle \text{London,Delhi} \rangle$$

- It should be clear that

  $$X \upharpoonleft S == \textbf{squash}\ (X \lhd S)$$

SEQUENCE CONSTRUCTORS

- *Filtering (Sequence Restriction)*

  When applied to a sequence this operation (symbol $\upharpoonright$) yields a new sequence the elements of which are all members of a specified set

  e.g. if *flight* is filtered by the set containing *Moscow*, *Delhi* and *Taipei*:

  $$\mathit{flight} \upharpoonright \{\text{Delhi, Moscow, Taipei}\} = \langle \text{Moscow, Delhi} \rangle$$

- After filtering, the sequence ordering remains that of the original sequence

- It should be clear that, if S is a sequence of elements from a set Q and X is a set of the same type as Q, then

  $$S \upharpoonright X == \textbf{squash}\ (S \rhd X)$$

GENERIC DEFINITIONS

- System models often embrace concepts which involve parameters

- For example, a *sequence* will have elements of a specific type but the functions that we can apply to a sequence are in no way dependent on that type

- The Z language, formally, defines such functions in terms of generic parameters

  e.g. the catenation operator for sequences may be formally defined by:

$$\begin{array}{|l}
\quad [X] \\
\hline
_ \frown _ : (\mathbf{seq}\ X) \times (\mathbf{seq}\ X) \\
\hline
\forall\, u,v : \mathbf{seq}\ X \bullet \\
\qquad u \frown v = u \cup \{n : \mathbf{dom} \bullet n + \#\, u \mapsto v\ n\} \\
\hline
\end{array}$$

Note that $v\ n$ symbolizes *function application*

GENERIC DEFINITIONS

- The 'box' used for *generic definitions* introduces the generic parameter(s) in the place where the schema name would normally be placed and has a *double line* at the top:

$$[X]$$

$$x : X$$

$$P$$

  The box above defines a whole family of variables x of generic type X which must satisfy some constraining predicate P

- Another example defining the *overrriding* function:

$$[X, Y]$$

$$\_\oplus\_ : (X \mathrel{\ooalign{$\rightarrow$\cr\hidewidth$\shortmid\shortmid$\hidewidth}} Y) \times (X \mathrel{\ooalign{$\rightarrow$\cr\hidewidth$\shortmid\shortmid$\hidewidth}} Y) \rightarrow (X \mathrel{\ooalign{$\rightarrow$\cr\hidewidth$\shortmid\shortmid$\hidewidth}} Y)$$

$$\forall\, u,v : X \mathrel{\ooalign{$\rightarrow$\cr\hidewidth$\shortmid\shortmid$\hidewidth}} Y \bullet u \oplus v = v \cup ((\,\mathbf{dom}\ v\,) \lhd v$$

SUMMARY OF SYMBOLS

$a \,..\, b$	*the range of consecutive integers from a to b (inclusive)*
max (min)	*function giving largest (least) element in an integer set*
succ (pred)	*function mapping integer to following (previous) integer*
$\mathbb{F}$ X	*the set of finite subsets of* X
X $\mathrel{+\!\!\!\!\twoheadrightarrow}$ Y	*the set of finite partial functions from* X *to* Y
seq X	*the set of sequences whose elements are drawn from* X == { S: $\mathbb{N} \rightarrow$ X | **dom** S = 1..#S}
$\mathbf{seq}_1$X	*the set of non-empty sequences*
$\langle$a,b,c, ...$\rangle$	*sequence of elements a,b,c, ...*= = {1 $\mapsto$ a,2 $\mapsto$ b,3 $\mapsto$ c, ...}
$\langle\ \rangle$	*the empty sequence* ({} *or* $\varnothing$)
rev S	*the sequence with elements in reverse order to those of* S
$\langle x_1, \ldots, x_n\rangle \frown \langle y_1, \ldots, y_n\rangle$	*concatenation/catenation* = = $\langle x_1, \ldots, x_n, y_1, \ldots, y_n\rangle$
head S	= = S *1* (*i.e first element of S*)
last S	= = S *#S* (*i.e last element of S*)
front S	= = S *with the element "last* S*" removed*
tail S	= = S *with element "head* S*" removed*
squash *f*	*the function f squashed into a sequence*
S $\upharpoonleft$ X	*extraction or index restriction (based on set* X*)*
S $\upharpoonright$ X	*filtration or sequence restriction (based on set* X*)*

This page is intentionally blank

☺

EXERCISES

1. Write out the *sets* of ordered pairs corresponding to the following *sequences*:

   (a) $\langle 7, 8, 9 \rangle$
   (b) $\langle 2, 2, 2 \rangle$
   (c) $\langle \{Sam\}, \{Les\} \rangle$

---

2. Which of the following sets are sequences?
   (a) $\{1 \mapsto Eddy, 2 \mapsto Eddy, 3 \mapsto Clary\}$
   (b) $\{n : \mathbb{N} \mid 0 < n \land n < 10 \bullet (n, 2n)\}$
   (c) $\{1 \mapsto 0\}$
   (d) $\{0 \mapsto Eddy, 1 \mapsto Mike\}$
   (e) $\{n : 0 \mathrel{..} 9 \bullet (n, n+1)\}$
   (f) $Chippy\_Queue \cup \{1 \mapsto Tom\}$
   where $Chippy\_Queue == \{1 \mapsto Mike, 2 \mapsto Eddy, 3 \mapsto Clary\}$
   (g) $Chippy\_Queue \oplus \{1 \mapsto Tom\}$
   where $Chippy\_Queue == \{1 \mapsto Mike, 2 \mapsto Eddy, 3 \mapsto Clary\}$
   (h) $\{4 \mapsto Eddy, 5 \mapsto Chris, 1 \mapsto Clary, 2 \mapsto Harry\}$

---

3. Write the following in sequence notation:
   (a) $\{3 \mapsto Fran, 1 \mapsto Maggie, 2 \mapsto Val, 4 \mapsto Harry\}$
   (b) $\{4 \mapsto Jan, 3 \mapsto Nick, 1 \mapsto Nick, 2 \mapsto Mary\}$
   (c) $\{ x : \mathbb{N} \mid x \leq 6 \bullet x + 1 \mapsto 6 - x \}$

---

4. If a sequence, S, is used to model a queue of customers awaiting service (seemingly forever!) at one of the *Sainsbury* supermarkets, write down an expression which represents the set of customers in the queue.

---

5. From what sort of sequence is it not possible to remove the first element?

---

6. If S : **seq** X, define the set *pal* X which is the set of all palindromes of type X (a sequence is a palindrome if it is the same backwards as it is forwards).

7. If we have a sequence S where $S == \langle u, p, c \rangle$, what are the values of the following?

(a) $\#(S \frown S)$
(b) $\mathbf{dom}\ S$
(c) $\mathbf{ran}\ (S \frown S)$
(d) $S\ (\#S)$ (this is 'function application')
(e) $\mathbf{head}\ S$
(f) $\mathbf{tail}^3\ S$ ($\mathbf{tail}^3$ is to be interpreted as 3 successive applications of *tail*)
(g) $\mathbf{front}\ (S \frown S)$
(h) $\mathbf{last}\ (\ \mathbf{front}\ (\ \mathbf{tail}\ S)\ )$
(i) $\mathbf{rev}\ (\ \mathbf{front}\ (\mathbf{tail}\ (S \frown \langle w \rangle))\ )$

---

8. Attempt to complete the following as useful laws about sequences, replacing the ? with appropriate expressions.

Assume: $s, t, u : \mathbf{seq}\ X,$

(a) $(s \frown t) \frown u = ?$
(b) $\#(s \frown t) = ?$
(c) $\mathbf{rev}\ (s \frown t) = ?$
(d) $\mathbf{rev}\ (\ \mathbf{rev}\ s\ ) = ?$
(e) $s \neq \langle\ \rangle \Rightarrow (\ \mathbf{front}\ s\ ) \frown (\ \mathbf{last}\ s\ ) = ?$
(f) $s \neq \langle\ \rangle \Rightarrow \mathbf{last}\ (\ \mathbf{rev}\ s\ ) = ?$
(g) $(\ s \frown t\ ) \upharpoonright Y = ?$
(h) $\emptyset \upharpoonleft s = ?$

---

9. Given $u, v : \mathbf{seq}\ CITY$
with $u == \langle \text{London, Amsterdam, Madrid} \rangle$
and $v == \langle \text{Paris, Frankfurt} \rangle$

write down the values of:

(a) $u \frown v$
(b) $\mathbf{rev}\ (u \frown v\ )$
(c) $\mathbf{rev}\ u$
(d) $\mathbf{rev}\ v$
(e) $\mathbf{rev}\ v \frown \mathbf{rev}\ u$
(f) $\mathbf{squash}\ (\ 2\ ..\ 4 \lhd \mathbf{rev}\ (\ u \frown v\ )\ )$
(g) $\mathbf{squash}\ (\ 4\ ..\ 2 \lhd \mathbf{rev}\ (\ u \frown v\ )\ )$
(h) $u \frown v \upharpoonright \{\text{London, Moscow, Paris, Rome}\}$
(i) $\mathbf{tail}\ (\ u \frown v\ ) \frown \mathbf{front}\ \langle \text{Moscow, Berlin, Warsaw} \rangle$

10. In general, it is possible for a sequence to have repeated elements (unlike sets). If, however, we wish there to be no duplicates in a sequence then we must restrict the function used to model the sequence.

    (a) What sort of function is needed?
    (b) Suggest a situation where such a sequence might be used for modelling.

---

11. When dealing with sets we introduced generalisations of *union* and *intersection*. For example, the *distributed intersection* of a set of sets yielded the set of those elements which were members of all sets in the set of sets.

---

12. When we consider sequences of sequences of objects then a useful operator is *distributed catenation* (*concatenation*). The application of this operator is often called *flattening*. The symbol for distributed catenation is $\frown/$. For example, $\frown/\langle\langle a,b,c\rangle,\langle d,e\rangle\rangle = \langle a,b,c,d,e\rangle$.

    Suggest what might be the results of the following:

    (a) $\frown/\langle\langle a\rangle,\langle a\rangle,\langle a\rangle\rangle$
    (b) $\frown/\langle\langle\rangle,\langle\rangle,\langle\rangle,\langle\rangle\rangle$
    (c) $\frown/\langle\langle\langle a,b,c\rangle,\langle d,e\rangle\rangle,\langle\langle f,g,h\rangle,\langle i,j,k\rangle\rangle\rangle$
