Sequences

Mairead Meagher

Waterford Institute of Technology,

2017

Sequences

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

We will look at

- the idea of and need for a sequence
- how to construct a sequence, different kinds of sequences
- operations on sequences
- examples of use of sequences

The need for sequences

- There are times when the properties of sets make them insufficient to model certain kinds of information.
- For example, sometimes order is important and it is important to include each occurrence of an element.
- In sets order is not important and duplicates are ignored i.e.

$$\{1, 2, 3\} = \{3, 2, 1\} = \{1, 3, 2, 1\}$$

- we introduce sequences to help model order (e.g. chronological, size, etc.)
- Sequences are ordered collections of objects.

- For example when talking about days of the week, their order is important. We write the sequence using the angle bracket notation.

$$daysinweek = \langle sun, mon, tue, wed, thurs, fri, sat \rangle$$

- This could be equivalently displayed without sequence notation as

$$\{1 \mapsto sun, 2 \mapsto mon, 3 \mapsto tue, 4 \mapsto wed, 5 \mapsto thu, 6 \mapsto fri, 7 \mapsto sat$$

Sequences - notation

- Thus, a sequence holds not only the elements but also the position they occupy.
- The sequence denoted

$$\langle mon, tue, wed \rangle$$

contains information as follows:

1	mon
2	tue
3	wed

- Note from this, it follows that :

$$\langle mon, tue, wed > \neq < wed, tue, mon \rangle$$
$$\langle mon, tue, wed > \neq < mon, tue, wed, mon \rangle$$

How to construct a sequence

Using the example as above:

$$DAYS ::= sun \mid mon \mid tue \mid wed \mid thurs \mid fri \mid sat$$

$$daysGoneBy : \mathrm{seq}\ DAYS$$

$$daysGoneBy = \langle mon, tue, wed \rangle$$

More generally (and formally)

$$seqX == \{ f : \mathbb{N} \twoheadrightarrow\!\!\!\!\!\!\shortmid\;\; X \mid dom f = 1 .. \# f \bullet f \}$$

Note the use of the finite function symbol $\twoheadrightarrow\!\!\!\!\!\!\shortmid\;\;$ to indicate that sequences are finite functions. Infinite sequences are not part of standard Z.

Non-empty Sequences

- A non-empty sequence is a simple specialisation of sequences.
- It represents sequences with at least one element.
- It is written seq_1 and can be defined as :

$$\mathrm{seq}_1\, X == \{s : \mathrm{seq}\, X \mid \# s > 0 \bullet s\}$$

Injective Sequences

- Sometimes it is important to have an injective (one-to-one) sequence.
- This will not allow any element in the range to be used more than once in the sequence.
- The **daysinweek** sequence seen above is an example of an injective function.
- It is written iseq and is defined as:

$$\mathrm{iseq}\, X == \mathrm{seq}\, X \cap (\mathbb{N} \rightarrowtail\!\!\!\!\!\mapsto X)$$

- It is precisely the set of finite of sequences over X which contain no repetitions.

Example of use

$$ACTIVITY ::= plan \mid code \mid test \mid review \mid deliver$$

$$\begin{array}{|l}
Project : \mathrm{seq}\ ACTIVITY \\
\hline
Project = \langle plan, code, deliver \rangle
\end{array}$$

- All normal functions operations can be used on sequences.
- dom can be used to deliver the set of index numbers for the sequence and
- ran delivers the set of items in the sequence.

$$Project(1) = plan$$
$$Project2 = code$$
$$\mathrm{ran}\ Project = \{plan, code, deliver\}$$
$$\mathrm{dom}\ Project = \{1, 2, 3\}$$
$$\#\ Project = 3$$

Operations on Sequences - Concatenation

- The main operator for constructing sequences is concatenation, written:

$$\frown$$

  where both of its arguments must be sequences.

- Note that a simple union operators between the two sets containing sequences is not sufficient.

Operations on Sequences - Concatenation

- Concatenation has the effect of joining two sequences by taking two sequences of the same type and producing a sequence numbered from one to the combined size as follows :

$$\langle mon, tue, wed\rangle \frown \langle thu, fri, sat\rangle =$$
$$\langle mon, tue, wed, thu, fri, sat\rangle$$

$$\{1 \mapsto mon, 2 \mapsto tue, 3 \mapsto wed\} \frown$$
$$\{1 \mapsto thu, 2 \mapsto fri, 3 \mapsto sat\}$$

$$= \{1 \mapsto mon, 2 \mapsto tue, 3 \mapsto wed, 4 \mapsto thu, 5 \mapsto fri, 6 \mapsto sat\}$$

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

The formal definition for the infix concatenation operation is as follows :

$$[X]$$

$$\_ \frown \_ : \mathrm{seq}\, X \times seqX \rightarrow seqX$$

$$\forall\, s, t : \mathrm{seq}\, X \bullet$$
$$\quad s \frown t = s \cup \{ n : 1 .. \# t \bullet (n + \# s \mapsto t(n)) \}$$

Reverse

The reverse operation takes a sequence and produces a sequence with the order of the elements reversed. For example,

$$rev\ \langle mon, tue, wed \rangle = \langle wed, tue, mon \rangle$$

Formally,

$$[X]$$

$$rev : \mathrm{seq}\, X \rightarrow \mathrm{seq}\, X$$

$$\forall\, s : \mathrm{seq}\, X \bullet$$
$$\quad rev\ s = \{ n : 1 \mathinner{..} \#\, s \bullet n \mapsto s((\#\, s) - n + 1) \}$$

Splitting Sequences - Head and Tail

- The head of a sequence is its first element.
- The tail of a sequence the rest of the sequence when the head is removed.
- Example

$$head(\langle mon, tue, wed \rangle) = mon$$

  Note that the head operation returns an element of the sequence, not a sequence.
- Example

$$tail(\langle mon, tue, wed \rangle) = \langle tue, wed \rangle$$

  Note that the tail operation returns a sequence.

Splitting Sequences - Front and Last

- These are similar to head and tail except that
- the last of a sequence is the last element in a sequence and
- the front of a sequence is the sequence containing everything except the last element.
- Example

$$\mathit{front}\ \langle \mathit{mon}, \mathit{tue}, \mathit{wed} \rangle = \langle \mathit{mon}, \mathit{tue} \rangle$$
$$\mathit{last}\ \langle \mathit{mon}, \mathit{tue}, \mathit{wed} \rangle = \mathit{wed}$$

Formal definitions for Splitting Sequences

More formally, the operations are defined as follows:

$$
\begin{array}{l}
[X] \\
\hline
head, last : \mathrm{seq}_1\, X \rightarrow X \\
tail, front : \mathrm{seq}_1\, X \rightarrow \mathrm{seq}\, X \\
\hline
\forall\, s : \mathrm{seq}_1\, X \bullet \\
\quad head\ s = s(1) \land \\
\quad s = \langle head\ s \rangle \frown tail\ s \land \\
\quad last\ s = s(\#\, s) \land \\
\quad front\ s = (1 \mathinner{..} \#\, s - 1) \lhd s \\
\hline
\end{array}
$$

Distributed Concatenation

- Analogous to the distributed union and intersection that we have already seen and used, there is a distributed concatenation operator.

- It takes a sequence of sequences and concatenates them all together. It is written $\frown/$

$$\frown/ \;:\mathrm{seq}(\mathrm{seq}\, X) \rightarrow \mathrm{seq}\, X$$

- Example

$$ss = \langle \langle a, b \rangle, \langle c, d \rangle, \langle e \rangle, \langle f, g, h \rangle \; \rangle$$
$$\frown/\, ss = \langle a, b, c, d, e, f, g, h \rangle$$

Squash

- Squash is a very important function. When we are working with sequences, we may be deleting elements etc. So for instance, if we wish to delete the third element of a sequence, e.g.

  $s : \operatorname{seq} \mathbb{N}$
  $s = \langle 3, 7, 10, 5, 16 \rangle$
  This can equivalently be written:
  $s = \{1 \mapsto 3, 2 \mapsto 7, 3 \mapsto 10, 4 \mapsto 5, 5 \mapsto 16\}$

  'Delete the third element'
  $s' = \{1 \mapsto 3, 2 \mapsto 7, 4 \mapsto 5, 5 \mapsto 16\}$
  s' is not a sequence, but if we apply squash to it:
  $squash(\{1 \mapsto 3, 2 \mapsto 7, 4 \mapsto 5, 5 \mapsto 16\}) = \{1 \mapsto 3, 2 \mapsto 7, 3 \mapsto 5,$
  $= \langle 3, 7, 5, 16 \rangle$ This is again a sequence.

Squash

- Squash takes a finite function defined strictly on the positive integers and compacts it into a sequence,
- i.e. there are no 'holes' left.
-

$$[X]$$

$$squash : (\mathbb{N}_1 \mathrel{+\!\!\!\!\!\twoheadrightarrow} X) \rightarrow seq\ X$$

Using squash

1. Let s' be the sequence s with the n^{th} element of it deleted. s' should be a sequence.

   $$s : \mathrm{seq}\,\mathbb{N}$$
   $$s' = squash(\{3\} \lhd s)$$

   s' is a sequence with the third element of s removed.

2. Let s' be the sequence s with all occurences of an element deleted. s' should be a sequence.

   $$s : \mathrm{seq}\,\mathbb{N}$$
   $$s' = squash(s \rhd \{23\})$$

   s' is a sequence with all the occurences of the number 23 removed from s.

- We revisit the concept of sets being disjoint and partitioning another set.
- Simply, a sequence of sets is disjoint if they have no elements in common(i.e. the distributed intersection operator yields the empty set).

$$disjoint\_ : \mathbb{P}(seq\, \mathbb{P}\, X)$$
$$disjoint\langle A, B\rangle \Leftrightarrow A \cap B = \varnothing$$

Partition/Disjoint - Example

- We model the employees in a factory.

$$management, personnel, production, employees : \mathbb{P}\, PERSON$$

management is the set of all employees in management roles.
personnel is the set of all employees in personnel roles.
production is the set of all employees in production roles.
employees is the set of all employees in the factory.

Partition/Disjoint

- We model the following thus:
  1. No employee has more than one role (e.g. they cannot be in management and production), this can be affected by:

     $$\mathit{disjoint}\ \langle \mathit{management}, \mathit{personnel}, \mathit{production} \rangle$$

  2. Furthermore, if we wish to state that each employee must in one of the three roles(sets), this can be affected by:

     $$\langle \mathit{management}, \mathit{personnel}, \mathit{production} \rangle \ \mathrm{partition}\ \mathit{employees}$$

Partition/Disjoint - Example

- Formally,

$$\_\ \mathrm{partition}\ \_ : (\mathrm{seq}(\mathbb{P}\, X)) \leftrightarrow \mathbb{P}\, X$$

- A sequence of sets partition a set S if
  1. they are disjoint and
  2. the distributed union of the sequence of sets is S.

Any questions?

Irgendwelche Fragen?

Aon céisteanna?