Formal Specification F28FS2, Lecture 8
Functions

Jamie Gabbay

March 3, 2014

Functions

Remember: a relation is a set of maplets.

An ordered pair (or maplet) looks like this: $1 \mapsto 2 : \mathbb{N} \times \mathbb{N}$.

A relation looks like this $\{1 \mapsto 2, 1 \mapsto 3\} : \mathbb{N} \leftrightarrow \mathbb{N}$ (a set of maplets).

If R is a relation then dom(R) is the set
$\{a : A \mid \exists\, b : B \bullet a \mapsto b \in R\}$ ('the set of a related to some b').

Use of functions

Every time we want to assign some information to something else (e.g. patient ID to patient; have function $\mathit{ID\_of}(\mathit{patient})$).

Represent programs that compute values deterministically given an input (or fail, if the function is partial; e.g. $2 * x$, $\sqrt{-1}$).

Indexes and arrays: map index to array value ($a[0]$, $a[1]$, $\ldots$).

Memory: $\mathbb{N} \rightarrow \langle 0..7 \rangle$ is a pretty good model of computer memory ($\mathit{contents\_of}(\mathit{cell})$).

Pointers (! is a function from a pointer l to a value $!l$).

Sequences: map natural number to a value, to model infinite lists (an infinite array is modelled as a function $a(0)$, $a(1)$, $a(2)$, $\ldots$).

Functions

A partial function $f : A \nrightarrow B$ is a relation $f : A \leftrightarrow B$ such that every element of A is related to at most one element of B. In symbols:

- $\forall\, a : A \bullet (\exists\, b : B \bullet a \mapsto b \in f) \Rightarrow (\exists_1\, b : B \bullet a \mapsto b \in f)$.
  "For every a of type A, if there is some b of type B such that $f(a) = b$ then there is exactly one such b."
- or... $\forall\, a : A \bullet (\neg \exists\, b : B \bullet a \mapsto b \in f) \vee (\exists_1\, b : B \bullet a \mapsto b \in f)$.
  "For every a of type A, either there are zero b of type B such that $f(a) = b$, or there is exactly one such b."
- or... $\forall\, a : A \bullet \#\{b : B \mid a \mapsto b \in f\} \leq 1$.
  "For every a of type A, the number of b of type B such that $f(a) = b$, is at most 1."
- or... $\forall\, a : A \bullet \#(\{a\} \lhd f) \leq 1$.
  "For every a of type A, there is at most one tuple in f whose left-hand side is a."

Total functions

A total function $f : A \rightarrow B$ is such that:

- $\forall\, a : A \bullet \exists_1\, b : B \bullet a \mapsto b \in f$.
  "For every a of type A there exists exactly one b such that $f(a) = b$."
- or... $\mathrm{dom(f)} = \mathrm{A}$.
  "The domain of f is equal to the set of elements of type A."

Write $f(a) = b$ for $a \mapsto b \in f$. Read this as f of a equals b.

If $\forall\, b : B \bullet a \mapsto b \notin f$ (i.e. $a \notin \mathrm{dom(f)}$) call f undefined on a.

Function overriding

Suppose $f, g : A \pfun B$. Define:

$$f \oplus g =$$
$$\{a \mapsto b : A \times B \mid g(a) = b \vee (a \notin \mathrm{dom(g)} \wedge \mathrm{f(a) = b}) \bullet \mathrm{a \mapsto b}\}$$

Read $f \oplus g$ as g, otherwise f. Read the predicate above in detail:

- If $g(a) = b$ then $(f \oplus g)(a) = g(a)$.
- Otherwise, if $f(a) = b$ then $(f \oplus g)(a) = f(a)$.
- Otherwise, $f \oplus g$ is undefined at a.

Note: $\mathrm{dom(f \oplus g) = dom(f) \cup dom(g)}$. Logically equivalently:

$$f \oplus g = \{a \mapsto b : A \times B \mid (a \in \mathrm{dom(g)} \Rightarrow \mathrm{g(a) = b}) \wedge$$
$$(a \in (\mathrm{dom(f)} \setminus \mathrm{dom(g)}) \Rightarrow \mathrm{f(a) = b}) \bullet \mathrm{a \mapsto b}\}$$

Injections, surjections

Call $f : A \nrightarrow B$ an injection when

- $\forall\, b : B \bullet \#\{a : A \mid f(a) = b\} \leq 1.$
  For every b of type B, there is at most one a of type A such that $f(a) = b$.
- $\forall\, a, a' : A \bullet f(a) = f(a') \Rightarrow a = a'.$
  For every a and a' of type A, if $f(a) = f(a')$ then $a = a'$.
- $\forall\, b : B \bullet \#(f \rhd \{b\}) \leq 1.$

Another way of reading this: 'no two elements of A map to the same element of B'.

$\lambda\, n : \mathbb{N}.2.n$ is injective; $2.n = 2.n'$ implies $n = n'$.

$\lambda\, n : \mathbb{N}.2$ is not injective; $2 = 2$ does not imply $n = n'$!

Think of an injection as 'losing no information'.

Injections, surjections

Call $f : A \twoheadrightarrow B$ a surjection when

- $\forall\, b : B \bullet \#\{a : A \mid f(a) = b\} \geq 1$.
  For every b of type B, there is at least one a of type A such that $f(a) = b$.
- $\forall\, b : B \bullet \exists\, a : A \bullet f(a) = b$.
  For every b of type B there is some a of type A such that $f(a) = b$.
- $range\ (f) = B$ (though you may need to define $range$).

Thus: 'every element of B is mapped to by something in A'.

$\lambda\, n : \mathbb{N}.2.n$ is not surjective; $\neg \exists\, n : \mathbb{N} \bullet 2.n = 3$.

$\lambda\, n : \mathbb{N}.n$ is surjective.

A surjection 'possibly throws away information, but captures all possible information in B'.

Sequences

Suppose T is any type (e.g. *PERSON*). Recall $N_1 = \{x : \mathbb{Z} \mid x > 0\}$.

Write *seq* T for the type populated by elements in the set

- $\{f : \mathbb{N}_1 \nrightarrow T \mid \forall\, n : \mathbb{N}_1 \bullet (n{+}1) \in \mathrm{dom(f)} \Rightarrow \mathrm{n} \in \mathrm{dom(f)}\}$.
- or. . . $\{f : \mathbb{N}_1 \nrightarrow T \mid \mathrm{dom(f)} = 1..\#\mathrm{dom(f)}\}$. (What's wrong with this?)

For example, $\{1 {\mapsto} t_1\}$ and $\{1 {\mapsto} t_1, 2 {\mapsto} t_2, 3 {\mapsto} t_3\}$ are sequences. So is $\varnothing$.

$\{2 {\mapsto} t_2\}$ and $\{2 {\mapsto} t_2, 3 {\mapsto} t_3\}$ are not sequences.

(Thanks to Ugis for his corrections.)

Nonempty sequences

Write $seq_1\ T$ for the type populated by elements in the set

- $\{f : seq\ T \mid \exists\, a : A \bullet f(a) \text{ defined}\}$.
- or. . . $\{f : seq\ T \mid \text{dom(f)} \neq \varnothing\}$.

For example $\{1 \mapsto t_1\}$ is a non-empty sequence. $\varnothing : A \nrightarrow B$ is not a non-empty sequence — it is the empty sequence.

Injective sequences

iseq T is the type populated by elements of $\mathbb{N}_1 \rightarrowtail\!\!\!\!\!\!\!\!\;+\;\, T$ which are injective; it is the set of sequences of elements of T that do not repeat.

Things to do to sequences: restrict them

$\{1,2\} \lhd f$ is the initial two elements of f (or the first element, or the empty sequence, depending on f).

$\{1,3\} \lhd f$ need not be a sequence, unless f consists of at most two elements.

For example $\{1,2\} \lhd \{1 \mapsto t_1, 2 \mapsto t_2, 3 \mapsto t_3\} = \{1 \mapsto t_1, 2 \mapsto t_2\}$.

Things to do to sequences: overwrite them

$f \oplus g$ is the sequence which starts as g, and then carries on as f (if any of f is left).

Head and tail

If $f : seq\ T$ then

$\text{head}(f) = f(1)$ ('pop f')

$\text{tail}(f) = \{i \mapsto t : \mathbb{N}_1 \times T \mid f(i+1) = t\}$ ('the stack afterwards').

Reverse a sequence

If f : $seq\ T$ then $rev\ f$ is the sequence f, reversed.

So $(revf)i = f(\#\mathrm{dom(f)} + 1 - \mathrm{i})$.

Concatenate sequences

If $f, g : seq\ T$ then $f \frown g$ is the sequence f, followed by the sequence g.

One way to specify this in Z:

$$f \frown g = f \cup \{i : \mathbb{N}_1 \mid i \leq \#g \bullet (i + \#f) \mapsto g(i)\}$$

More on sequences later.