

Formal Specification

Lecture 8

Murdoch J. Gabbay, Heriot-Watt University, Scotland

Functions

Remember

- A proposition has a truth-value.
- A variable ranges over a type.
- A type is a maximal set.
- A set has elements.
- A schema is a proposition-with-some-state.
- A relation is a set of maplets.

Functions

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

Functions

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

In symbols:

- $\forall a : A \bullet (\exists b : B \bullet a \mapsto b \in f) \Rightarrow (\exists_1 b : B \bullet a \mapsto b \in f)$.
- **or...** $\forall a : A \bullet (\neg \exists b : B \bullet a \mapsto b \in f) \vee (\exists_1 b : B \bullet a \mapsto b \in f)$.
- **or...** $\forall a : A \bullet \#\{b : B \mid a \mapsto b \in f\} \leq 1$.

Total functions

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

- $\forall a : A \bullet \exists_1 b : B \bullet a \mapsto b \in f$.
- or... $\text{dom}(f) = A$.

What does that say, in english?

Write $fa = b$ 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. if $a \notin \text{dom}(f)$ then say f is undefined on b .

Function overriding

Suppose $f, g : A \mapsto B$.

Then

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

Read $f \oplus g$ as ' g , otherwise f '. In more detail:

- $\text{dom}(f \oplus g) = \text{dom}(f) \cup \text{dom}(g)$.
- If $ga = b$ then $(f \oplus g)a = ga$.
- Otherwise, if $fa = b$ then $(f \oplus g)a = fa$.
- Otherwise, $(f \oplus g)a$ is undefined.

Injections, surjections

Call $f : A \rightarrow B$ an **injection** when

- $\forall b : B \bullet \#\{a : A \mid fa = b\} \leq 1$.
- $\forall a, a' : A \bullet fa = fa' \Rightarrow a = a'$.

In words: ‘no two elements of A map to the same element of B ’.

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

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

Think of an injection as ‘losing no information’.

Injections, surjections

Call $f : A \rightarrow B$ a **surjection** when

- $\forall b : B \bullet \#\{a : A \mid fa = b\} \geq 1.$
- $\forall b : B \bullet \exists a : A \bullet fa = b.$

In words: ‘every element of B is mapped to by something in A ’.

The function $\lambda n : \mathbb{N}.2.n$ is not surjective; $2.n \neq 3$ ever.

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

Think of a surjection as ‘possibly throwing away information in A , but not ignoring any possible information which could be expressed by B ’.

Use of functions

Assign doors to door IDs. Represent programs that compute values deterministically given an input (or fail, if the function is partial). Indexes. Memory ($\mathbb{N} \rightarrow \langle 0..7 \rangle$ is a pretty good model of computer memory) and pointers.

And sequences...

Sequences

Suppose T is any type (e.g. *PERSON*). Recall $\mathbb{N}_1 = \{x : \mathbb{Z} \mid x > 0\}$.

Write *seq* T for

- $\{f : \mathbb{N}_1 \rightarrow T \mid \forall n : \mathbb{N}_1 \bullet n \in \text{dom}(f) \Rightarrow (n-1) \in \text{dom}(f)\}$.
- **or...** $\{f : \mathbb{N}_1 \rightarrow T \mid \text{dom}(f) = 1..\#\text{dom}(f)\}$. (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 \emptyset .

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

Nonempty sequences

Write $\text{seq}_1 T$ for

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

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

Injective sequences

iseq T is the type of elements of $\mathbb{N}_1 \rightarrow T$ which are injective; it is the set of sequences of elements of T that do not repeat.

Things to do to sequences: restrict them

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

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

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

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 $f : \text{seq } T$ 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 $f : \text{seq } T$ then $\text{rev } f$ is the sequence f , reversed.

So $(\text{rev } f)_i = f(i + 1 - \#\text{dom}(f))$.

Concatenate sequences

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

That's plenty for one day. More on sequences later.