Mathematics for Computer Scientists 2 (G52MC2)

L07 : Operations on sets

Thorsten Altenkirch

School of Computer Science
University of Nottingham

October 29, 2009

Enumerations

- We construct finite sets by enumerating a list of names.
- In Coq we use `Inductive`, e.g.

```
Inductive square : Set :=
| nought : square
| cross : square
| empty : square.
```

- We use `match` to define functions on enumerations and `case` to reason about them.
- Im Maths we write $\text{square} = \{\text{nought}, \text{cross}, \text{empty}\}$
- Note that square, nought, empty are constants, not variables!
- They cannot be bound by quantifiers and different constants are never equal, e.g. $\text{nought} \neq \text{empty}$.
- An important example is $\text{bool} = \{\text{true}, \text{false}\}$.

New sets from old . . .

Given sets A, B we can construct new sets:

Cartesian product
: $A \times B$ is the set of pairs (a, b) with $a : A$ and $b : B$.

Disjoint union
: $A + B$ is the set of elements of the form $\mathrm{inl}\, a$ with $a : A$ and $\mathrm{inr}\, b$ with $b : B$.

Functions
: $A \to B$ is the set of functions from A to B. We can apply a function $f : A \to B$ to an element $a : A$ obtaining $f\, a : B$.

Cartesian Product

- $$\frac{a : A \qquad b : B}{(a, b) : A \times B}$$
- Example: Cartesian coordinates: $\mathbb{R} \times \mathbb{R}$.
- If A, B are finite sets where A has m elements and B has n elements, then $A \times B$ has mn elements.
- Use `match` in programs and `case` (or `destruct`) in proofs.
- Projections:

$$\begin{aligned} \mathrm{fst} &: A \times B \to A \\ \mathrm{snd} &: A \times B \to B \end{aligned}$$

Polymorphism in Coq

- Functions like fst and snd work for all sets.
  They are *polymorphic*.
- In Coq we can instantiate the explicitly:

$$\mathrm{fst}\,\mathrm{bool}\,\mathrm{nat} : \mathrm{bool} \times \mathrm{nat} \to \mathrm{bool}$$

- To avoid clutter, we use

```
Set Implicit Arguments.
```

- Coq *tries* to infer instantiations, e.g. we can write:

$$\mathrm{fst}\,(\mathrm{true}, 7) : \mathrm{bool}$$

Disjoint union

- Also called coproducts or sums.
- $$\frac{a : A}{\mathrm{inl}\, a : A + B} \qquad \frac{b : B}{\mathrm{inr}\, b : A + B}$$
- If A, B are finite sets where A has m elements and B has n elements, then $A + B$ has $m + n$ elements.
- inl, inr are called *injections*.
- Coq cannot infer one of the arguments, it has to be given explicitely:

$$\mathrm{inl}\ \mathrm{nat}\ \mathrm{true} : \mathrm{sum}\ \mathrm{bool}\ \mathrm{nat}$$

Functions

- $A \to B$ is the set of functions with domain A and range (or codomain) B.
- If A, B are finite sets where A has m elements and B has n elements, then $A \to B$ has n^m elements.
- Application:

$$\frac{f : A \to B \qquad a : A}{f\, a : B}$$

- λ-abstraction:

$$\frac{t : B \text{ given } x : A}{\mathrm{fun}(x : A) \Rightarrow t : A \to B}$$

Extensionality

- To show that two functions $f, g : A \to B$ are equal we need the *principle of extensionality*:

$$(\forall a : A, f\ a = g\ a) \to f = g$$

- The principle of extensionality is not provable in Coq, hence we assume it as an axiom (`ext`).
- Unlike the principle of the excluded middle, ext is accepted in intuitionistic logic.
- It reflects the idea of a function as a *black box*.

Order of a function

- The order of a function is determined by its type:

$$\begin{aligned} \text{order}\,\mathbb{N} &= 0 \\ \text{order}\,(A \to B) &= \max\,((\text{order}\,A) + 1)\,(\text{order}\,B) \end{aligned}$$

- E.g. $f : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ is a 2nd order function.

Isomorphisms

- To sets A and B are *isomorphic* ($A \simeq B$) if there are functions:

$$\begin{aligned} f &: A \to B \\ g &: B \to A \end{aligned}$$

  such that

$$\forall a : A, g\,(f\,a) = a$$
$$\forall b : B, f\,(g\,b) = b$$

- f, g is called an *isomorphism*.
- Two finite sets are isomorphic, iff (if and only if) they have the same number of elements.
- Examples of general isomorphisms, for all sets A, B, C:

$$\begin{aligned} A \times (B \times C) &\simeq (A \times B) \times C \\ A \times (B + C) &\simeq A \times B + A \times C \\ A \times B \to C &\simeq A \to (B \to C) \end{aligned}$$

The Curry-Howard correspondence

- Curry and Howard observed that operations in sets correspond to operations on sets.
- A proposition is true if the corresponding set is inhabited.
- This is an alternative to the classical correspondence of bool and Prop.

Set	Prop	bool
$\times$	$\wedge$	&&
$+$	$\vee$	||
$\emptyset$	False	false
$\rightarrow$	$\rightarrow$	implb

What about $\cup$, $\cap$ and $\subseteq$?

- In classical set theory people frequently use the following operations on sets:
  - $\cup$ union of sets
  - $\cap$ intersection of sets
  - $\subseteq$ The subset relation between sets
- These are not operations on sets in the sense of Coq.
- Every element belongs precisely to one set in Coq, hence the $\subseteq$ relation doesn't make sense.
- However, for any set A we can define $\mathcal{P}\,A = A \to \mathbf{Prop}$ and:

$$\subseteq : \mathcal{P}\,A \to \mathcal{P}\,A \to \mathbf{Prop}$$
$$P \subseteq Q = \forall a : A, P\,a \to Q\,a$$
$$\cup, \cap : \mathcal{P}\,A \to \mathcal{P}\,A \to \mathcal{P}\,A$$
$$(P \cup Q)\,a = P\,a \lor Q\,a$$
$$(P \cap Q)\,a = P\,a \land Q\,a$$