This week we're continuing to define our type theory: The last bits are: Σ-types, identity-types, and universes. (Sections 4–6 in the notes.)

General inductive types

I refer to Dybjer [4] for a general definition of inductive types and their eliminators, and also for a general definition of inductive families. The thesis by Cockx [2] gives a nice overview as well, and details how to translate definitions by pattern matching into eliminator form (in a way compatible with HoTT).

The pure logic of problems and solutions

The Brouwer-Heyting-Kolmogorov (BHK) interpretation of constructive logic suggests that we should think of types as describing problems (or intentions) and elements of types as describing solutions of problems (or fulfilments of intentions). This also describes a logic, which we'll call *pure constructive logic*, where types are propositions and elements of types are proofs. (NB We'll later have a more refined notion of proposition, where only some types are propositions.)

logical constant	type	construction
implication	$A \to B$	function
conjunction	$A \times B$	pair
(pure) disjunction	$A + B$	binary choice
universal quant.	$\prod_{(x:A)} B(x)$	function
(pure) existential quant.	$\sum_{(x:A)} B(x)$	pair
equality	$a =_A b$	identification

Identity types

There are many options for formalizing identity types:

1. Ad hoc definition for each type (maybe by induction on the presentation of the type). For instance, for $f, g : \prod_{(x:A)} B(x)$ we would take $f =_{\prod_{(x:A)} B(x)} g :\equiv \prod_{(x:A)} f\,x =_{B(x)} g\,x$. Such a proposal was *observational type theory* [1].
2. Add the (local) rule
$$\frac{\vdash p : a =_A b}{\vdash a \equiv b : A}$$
This gives *extensional type theory*. The downside is that judgmental equality becomes undecidable.
3. Define via an *interval type* and *extension types* ⇝ *cubical type theory* (see `https://ncatlab.org/nlab/show/cubical+type+theory` for references). We may discuss this further later.
4. Define as an *inductive family*: the least reflexive family (Martin-Löf, for history, see Dybjer [3]). This is what we do.
5. As above (4), but add Streicher's *axiom K*: for any type A, $a : A$, and $p : a =_A a \vdash P(p)$ type:
$$K_a^P : P(\mathrm{refl}_a) \to \prod_{(p:a=_A a)} P(p)$$
6. ...

Of these, (4), (3), and some versions of (1) are compatible with the homotopy type interpretation.

Exercises

1. Ex. 4.2: Show (i.e., construct elements of) (a) $(A + \neg A) \to \neg\neg A \to A$ and (b) $\neg\neg\neg A \to \neg A$.
2. Ex. 4.8: Lists.
3. Ex. 5.1: (a) State Goldbach's Conjecture in type theory. (b) State the Twin Prime Conjecture in type theory.
4. Ex. 5.2–5.4, 5.6: Operations on identity types.

References

[1] Thorsten Altenkirch and Conor McBride. *Towards Observational Type Theory*. Rejected from LICS 2006. 2006. URL: `http://strictlypositive.org/ott.pdf`.

[2] Jesper Cockx. "Dependent Pattern Matching and Proof-Relevant Unification". PhD. KU Leuven, 2017. URL: `https://lirias.kuleuven.be/retrieve/456787`.

[3] Peter Dybjer. *(What I Know about) the History of the Identity Type*. Workshop on Identity Types, Uppsala, 13 November, 2006. 2006. URL: `http://www.cse.chalmers.se/~peterd/papers/historyidentitytype.pdf`.

[4] Peter Dybjer. "Inductive sets and families in Martin-Löf's type theory and their set-theoretic semantics". In: *Logical frameworks. Proceedings of the first annual workshop "Logical frameworks: design, implementation and experiment", held in Sophia-Antipolis, France, May 7–11, 1990*. Cambridge University Press, 1991, pp. 280–306. URL: `http://www.cse.chalmers.se/~peterd/papers/Setsem_Inductive.pdf`.