Introduction to Theoretical Computer Science

Tutorial Sheet 7

This week's questions go beyond the course, and are intentionally difficult. If you can do them without looking things up, you should probably be teaching the course.

(1) In our definitions of polymorphic type inference, we used a $\forall$ sign: $\forall\alpha.\alpha \to \alpha$, for example. Can you think of a way for a type $\exists\alpha.\alpha \to \alpha$ to make sense? *Hint:* logically, an $\exists$ statement is saying not only that there exists a type, but it's also saying that we don't know what it is (yet). In order to demonstrate the truth of $\exists x.\phi(x)$, we have to instantiate x with some concrete value. Does this remind you of anything in programming languages?

(2) You're used to the idea of polymorphic lists in Haskell or ML. Clearly (!), the polymorphic head function has type $\forall\alpha.(\mathsf{List}\ \alpha) \to \alpha$. But what is List? It's something that takes a type, and gives back a type. It is a function on types! Similarly, $\to$ is a function that takes two types and returns a type.

So we can imagine writing a λ-calculus extended with functions on types: for example, $\mathsf{Pair} = \Lambda\alpha.\Lambda\beta.\alpha \times \beta$ might define a Pairing constructor in terms of product types.

Do we need some notion of 'type of types'? What would the rules look like for these? What would be base 'types of types' be? What's the difference between $\Lambda\alpha$ in this 'type constructor', and $\forall\alpha$ in polymorphic types?

(3) In intuitionistic logic, negation is defined as $\neg\phi \stackrel{\mathsf{def}}{=} \phi \to \bot$, where '$\to$' is logical implication, and $\bot$ is 'false'. Thus, a proof of $\neg\phi$ is a procedure that from a proof of ϕ produces a proof of $\bot$.

What does negation look like under the Curry–Howard correspondence? That is, if τ is a type, what is the type $\neg\tau$, and what is a term witnessing $\neg\tau$? (We discussed the first question in the lecture, but didn't get on to the second in any detail.)