Bar Induction and the Fan Theorem in Constructive Type Theory

Robert Constable, Abhishek Anand, Mark Bickford

October 12, 2012

1. König's Lemma° (classical 1926)

   Every finitely generated tree $\mathcal{T}$ with infinitely many points contains at least one infinite branch. Smullyan *First-Order Logic* p.32

   Every finitely generated tree $\mathcal{T}$ with an unbounded set of points has at least one unbounded path.

   The is not a constructive theorem since we can't find the path.

2. The *Fan Theorem* is its contrapositive, due to Brouwer circa 1927.

   Every finitely generated tree $\mathcal{T}$ in which every path is bounded ($\forall \alpha : Path.\ \exists n : \mathbb{N}.\ |\alpha| < n$) is finite.

   This theorem is intuitionistically true. We believe it is true constructively, but Constructive Type Theory (CTT) currently has no realizer for it. Brouwer proved this by 1927 if not earlier. He proved it from a more general principle called Bar Induction, a classically valid form of induction to which Brouwer gave a constructive justification. Indeed, he might have discovered the principle. The result is not true if the paths are given by general recursive functions—Kleene 1959. Abhishek will give new proofs of Kleene's Theorem.

3. Bar Induction on the universal spread

   Given the *spread* (infinite tree) of choice sequences of natural numbers, that is the type of all sequences $\alpha : \mathbb{N} \to \mathbb{N}$, if every sequence is "barred by a decidable predicate," e.g. is finite, then we can do induction to prove a predicate A on points of the spread. The base case is that immediately barred nodes imply A, and if for all numbers n, point a extended by n satisfies A, say $A(a{*}n)$, implies $A(a)$, then $A(nil)^{\dagger}$. Time permitting Bob and Mark will discuss Kleene's detailed justification of this principle and discuss its realizer. They will sketch Brouwer's proof of the Fan Theorem using Bar Induction.

† Note, $a * n$ is a with n adjoined at the right, and nil is the root of the tree. Nuprl notation is $a@n$, following Lisp.