Re: ILC2003: Church vs Turing Smackdown

Source: http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2003−10/1728.html

---

From: Anton van Straaten (*anton_at_appsolutions.com*)
Date: 10/17/03

```
Date: Fri, 17 Oct 2003 03:25:29 GMT
```

Kenny Tilton wrote:
> *Another thing that sailed over my head but low enough to get my interest*
> *was an exchange in which (I am going to mangle this beyond belief)*
> *McCarthy corrected something by saying it was not until Turing did*
> *something that Church (or somebody) was sure of the lambda calculus.*
>
> *This reminds me of those horrible moments when I try to sing a little*
> *bit of a song in a record store to an employee when I can't find*
something.
>
> *What was that? It happened so fast I grokked neither the assertion nor*
> *the rebuttal.*

I don't remember the assertion which prompted the response − I think it was something that Jay Sulzberger said − but I believe that the gist of McCarthy's reply was that Church wasn't sure that the lambda calculus was a complete model for computability, until Turing proposed Turing Machines as a model for computability, and Turing proved the two equivalent.

There's a summary of the relevant history at
http://en2.wikipedia.org/wiki/Church−Turing_thesis :

"In his 1936 paper On Computable Numbers, with an Application to the Entscheidungsproblem Alan Turing tried to capture this notion formally with the introduction of Turing machines. In that paper he showed that the 'Entscheidungsproblem' could not be solved. A few months earlier Alonzo Church had proven a similar result in A Note on the Entscheidungsproblem but he used the notions of recursive functions and Lambda−definable functions to formally describe effective computability. Lambda−definable functions were introduced by Alonzo Church and Stephen Kleene (Church 1932, 1936a, 1941, Kleene 1935) and recursive functions by Kurt Gödel and Jacques Herbrand (Gödel 1934, Herbrand 1932). These two formalisms describe the same set of functions, as was shown in the case of functions of positive integers by Church and Kleene (Church 1936a, Kleene 1936). When hearing of Church's proposal, Turing was quickly able to show that his Turing machines in fact describe the same set of functions (Turing 1936, 263ff)."

Anton