

Church vs. Curry

Stilrichtungen von Typsystemen in Theorie und Praxis

Baltasar Trancón y Widemann
Universität Bayreuth

Zusammenfassung

In der Theorie der Typsysteme werden zwei Stilrichtungen unterschieden, die sich jeweils auf einen der Grundväter der Disziplin berufen: Alonzo Church und Haskell Curry. Auf den ersten Blick unterscheiden sich die Ansätze durch die Art, wie Variablen eingeführt werden: Im sogenannten Church-Stil wird bei der Einführung der Typ explizit angegeben, während er im Curry-Stil implizit bleibt. Dieser nur scheinbar marginale Unterschied hat weitreichende Konsequenzen. Im Church-Stil ergibt sich natürlicherweise Typinformation von unten nach oben; die Typprüfung eines formalen Dokuments besteht daher in der Berechnung eines Typs und ggf. dem Vergleich mit einer Typdeklaration. Im Curry-Stil dagegen ist die Datenflussrichtung von Typen unklar; die Typprüfung eines formalen Dokuments muss daher Inferenz beinhalten. Typabstraktion und -instanziierung sind im Church-Stil explizit, im Curry-Stil dagegen implizit. Der Church-Stil verlangt wesentlich mehr Information und damit Schreibaufwand vom Benutzer. Im Gegenzug bleiben die zentralen Probleme auch bei sehr reichhaltigen Typsystemen entscheidbar. Typisierung im Church-Stil wird von Spezifikations- und Beweissprachen, aber auch von imperativen Sprachen mit kaum mathematisierten Typsystemen bevorzugt. Typisierung im Curry-Stil ist vor allem durch die funktionalen Programmiersprachen ML und Haskell und das diesen zugrundeliegende Hindley-Milner-Typsystem bekannt. Dieser Vortrag fasst die theoretischen Unterschiede zwischen den Stilen zusammen und zeigt Beispiele aus der Programmierpraxis in typischen Vertretern der beiden Sprachfamilien. Insbesondere werden Mechanismen vorgestellt, die den Benutzer von den Anforderungen des Curry-Stils entlasten sollen. Als Beispiele dienen die generischen Typen von Java, das Beweissystem Coq sowie die experimentelle totale funktionale Programmiersprache Tofu.

Literatur

- [1] H. P. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, pages 117–309. Oxford University Press, 1992.