

Getting Started with Haskell

Jan van Eijck

CWI & ILLC, Amsterdam

ESSLLI, Opole, August 6, 2012

A short history of Haskell



A short history of Haskell

In the 80s, efforts of researchers working on functional programming were scattered across many languages (Lisp, SASL, Miranda, ML, ...).

In 1987 a dozen functional programmers decided to meet in order to reduce unnecessary diversity in functional programming languages by **designing a common language** that is

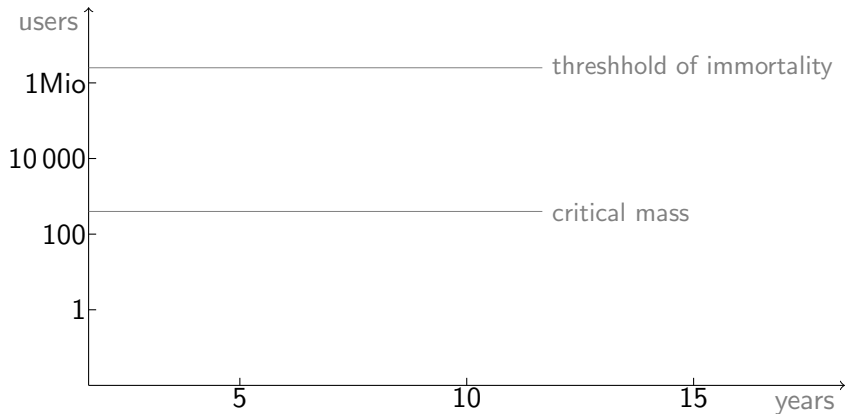
- based on ideas that enjoy a wide consensus
- suitable for further language research as well as applications, including building large systems
- freely available

A short history of Haskell

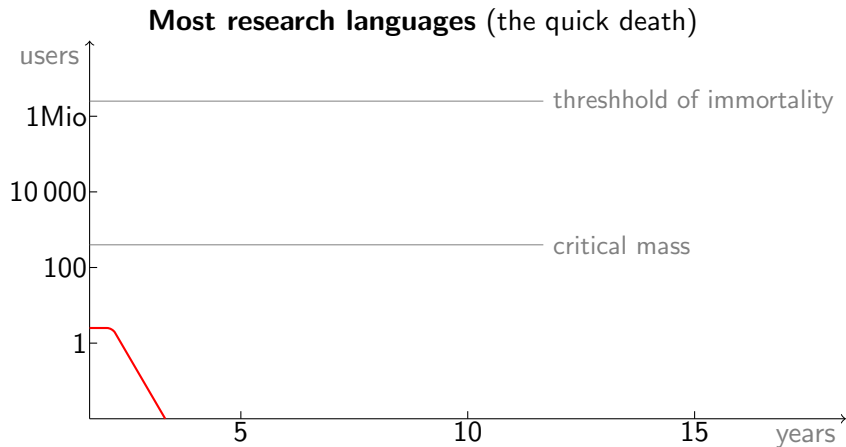
In 1990, they published the first **Haskell** specification, named after the logician and mathematician Haskell B. Curry (1900-1982).



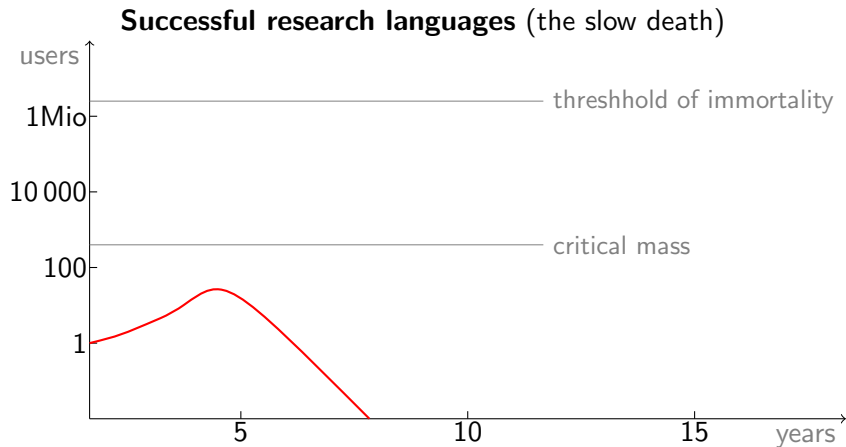
Simon Peyton-Jones: The life cycle of programming languages



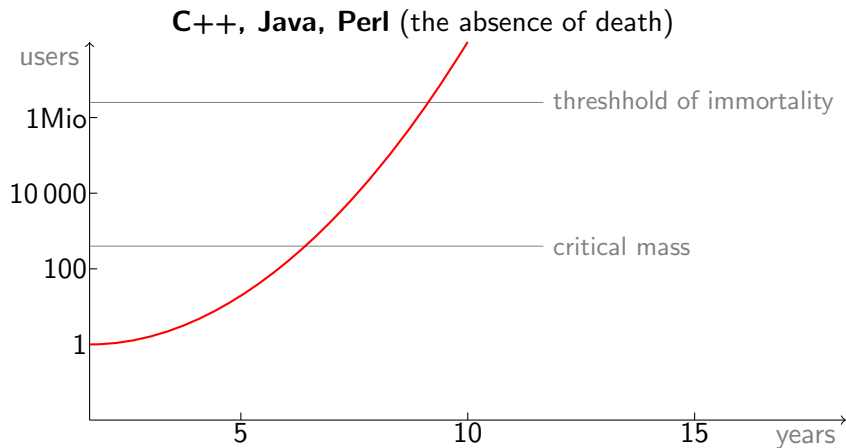
Simon Peyton-Jones: The life cycle of programming languages



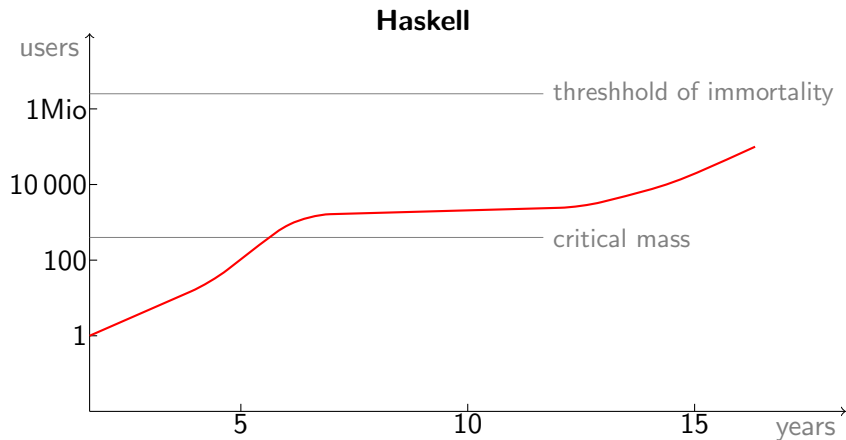
Simon Peyton-Jones: The life cycle of programming languages



Simon Peyton-Jones: The life cycle of programming languages



Simon Peyton-Jones: The life cycle of programming languages



Haskell is functional

A program consists entirely of functions.

- The main program itself is a function with the program's input as argument and the program's output as result.
- Typically the main function is defined in terms of other functions, which in turn are defined in terms of still more functions, until at the bottom level the functions are language primitives.

Running a Haskell program consists in evaluating expressions (basically functions applied to arguments).

A shift in thinking

Imperative thinking:

- Variables are pointers to storage locations whose value can be updated all the time.
- You give a sequence of commands telling the computer what to do step by step.

Examples:

- initialize a variable `examplelist` of type integer list, then add 1, then add 2, then add 3
- in order to compute the factorial of n , initialize an integer variable `f` as 1, then for all `i` from 1 to n , set `f` to `f × i`