David Greni

Report 1

CS 332

Haskell

Haskell is a functional programming language. It is a statically typed fully functional programming language. It was developed in the early 1990. Research on functional programming languages started in the early 1980s. The research was fragmented between many different groups. The name for the language came from the mathematician Haskell B. Curry whose contributions influenced the developers of the language. Haskell was created by a committee that would meet many times over multiple days to create the structure of the programming language. This structure has continued to this day with a committee representing the maintainers of the Haskell language. The committee is currently a group of 7 members who vote on any changes to the language including new syntax, packages, and any other decisions that need to be made. Haskell has many features that make it unique. It is a functional programming language that is statically typed, has native executables, garbage collection, but one of its most unique features is that it is lazy. Being lazy means that Haskell operates on a call by need system and not a call by value system. This allows for delaying execution of code until the actual value is needed. Another feature of call by need is that it ensures that values are only calculated once. One of the largest users of Haskell is Meta. They use Haskell for many projects. Most notably project Sigma uses Haskell to fight against spam, and malware. This project services more than one million requests per second. Sigma is a rule based system that checks messages and posts against their rules and filters the results. Meta uses Haskell because it is functional and strongly typed which doesn’t allow policies to overlap and interfere with one another. It is also much faster than what was previously used both in compute time and when they push the code to production allowing changes to be made more rapidly. Some of the downsides of Haskell are that there is not a wide variety of libraries for Haskell. The tools to write these libraries do exist but the off the self solution does not exist yet. Another complaint is that there is not a standard on how to write Haskell. Haskell is very flexible in the style that it is written in. This is both a pro and a con. It allows programmers to be very flexible but makes sharing code and readability much harder. An example of declaration in Haskell is data Anniversary = Birthday String Int Int Int . In Haskell there is no mutation of variables so assignment is given at variable declaration. let foo x = x + 2 would be an example of creating a function where we can assign a value to x with foo 4. In the IO module we can directly assign values with input <- getLine. An example of selection in Haskell is the following code:

select :: (t -> Bool) -> [t] -> [a] -> [a]

select p xs = map **snd** . filter (**p . fst**) . zip xs

A simple repetition example in Haskell is:

printString n =

do

putStrLn “string”

printString( n-1)

Overall Haskell is a very versatile language with many advantages. It allows for very fast data processing with fast compile times and hot swapping executables. I would be very interested in learning Haskell as I think it is interesting and I have enjoyed working through functional programming languages. I also think it is very interesting how the committee is ran and how changes are implemented.

Reference:

University, Paul Hudak Yale, et al. “A History of Haskell: Proceedings of the Third ACM SIGPLAN Conference on History of Programming Languages.” *ACM Conferences*, 1 June 2007, dl.acm.org/doi/pdf/10.1145/1238844.1238856.

*What Makes Haskell Unique*, www.snoyman.com/blog/2017/12/what-makes-haskell-unique/. Accessed 7 Dec. 2023.

Marlow, Simon, and Simon Marlow. “Fighting Spam with Haskell.” *Engineering at Meta*, 26 June 2018, engineering.fb.com/2015/06/26/security/fighting-spam-with-haskell/.