

Embedding Programming Languages : PROLOG in HASKELL

Mehul Solanki

University of Northern British Columbia

solanki@unbc.ca
230108015

April 24, 2014

Programming Languages

A programming language is an artificial language designed to communicate instructions to a machine, particularly a computer [1].

For example, C, JAVA.



Figure 1 : The Universe of Programming Languages [10]

Programming Language Paradigms

A programming paradigm is a fundamental style of computer programming, a way of building the structure and elements of computer programs [2].

For example, Object Oriented Programming.



Figure 2 : Programming Paradigms [2, 13]

Classification

Programming languages are classified into paradigms depending on their characteristics and features.

For example, JAVA is an Object Oriented Programming Language [9].

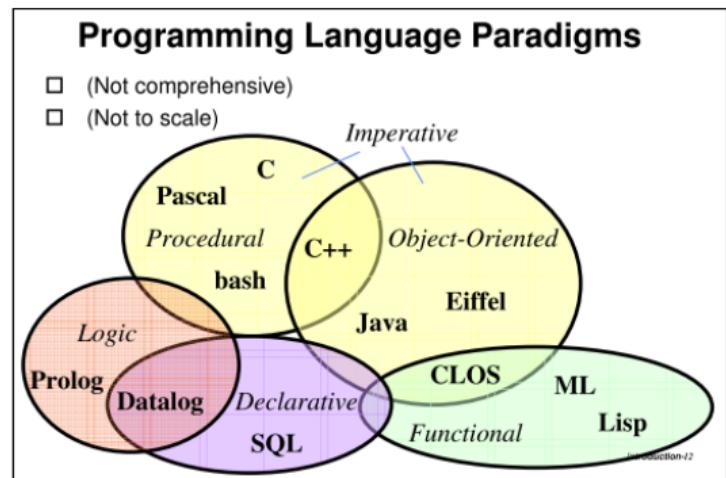


Figure 3 : Classification of Programming Languages [12]

Logic Programming

In logic programming, a program consists of a collection of statements expressed as formulas in symbolic logic. There are rules of inference from logic that allow a new formula to be derived from old ones, with the guarantee that if the old formulas are true, so is the new one [3].

For example, PROLOG.

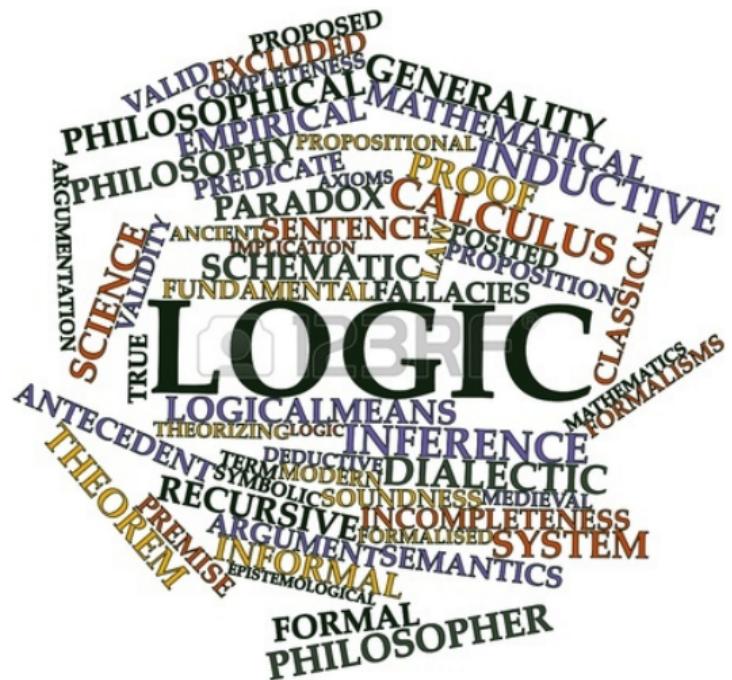


Figure 4 : Logic Programming [14]

PROLOG

General purpose logic programming language with over 20 distributions [6]. PROLOG is a programming language borrowing its basic constructs from logic. A pure PROLOG program is a logic program, in which an order is defined both clauses in the program and for goals in the body of the clause [7].



SWI Prolog

Figure 5 : SWI PROLOG Distribution [15]

Functional Programming

Programming in a functional language consists of building definitions and using the computer to evaluate expressions [4].

For example, HASKELL.

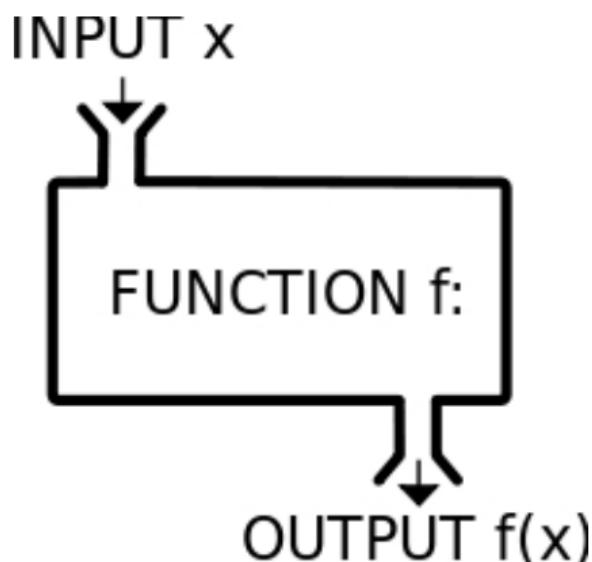


Figure 6 : Function [16]

HASKELL

HASKELL is an advanced purely-functional programming language. In particular, it is a polymorphically statically typed, lazy, purely functional language [8].



Figure 7 : HASKELL
Programming Language [17]

Alternate Classification

A programming language inherits features from a number of paradigms rather than belonging to a single paradigm[5].

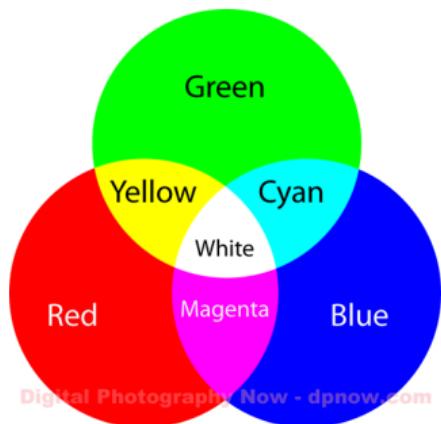


Figure 8 : Mixing Colours
[18]

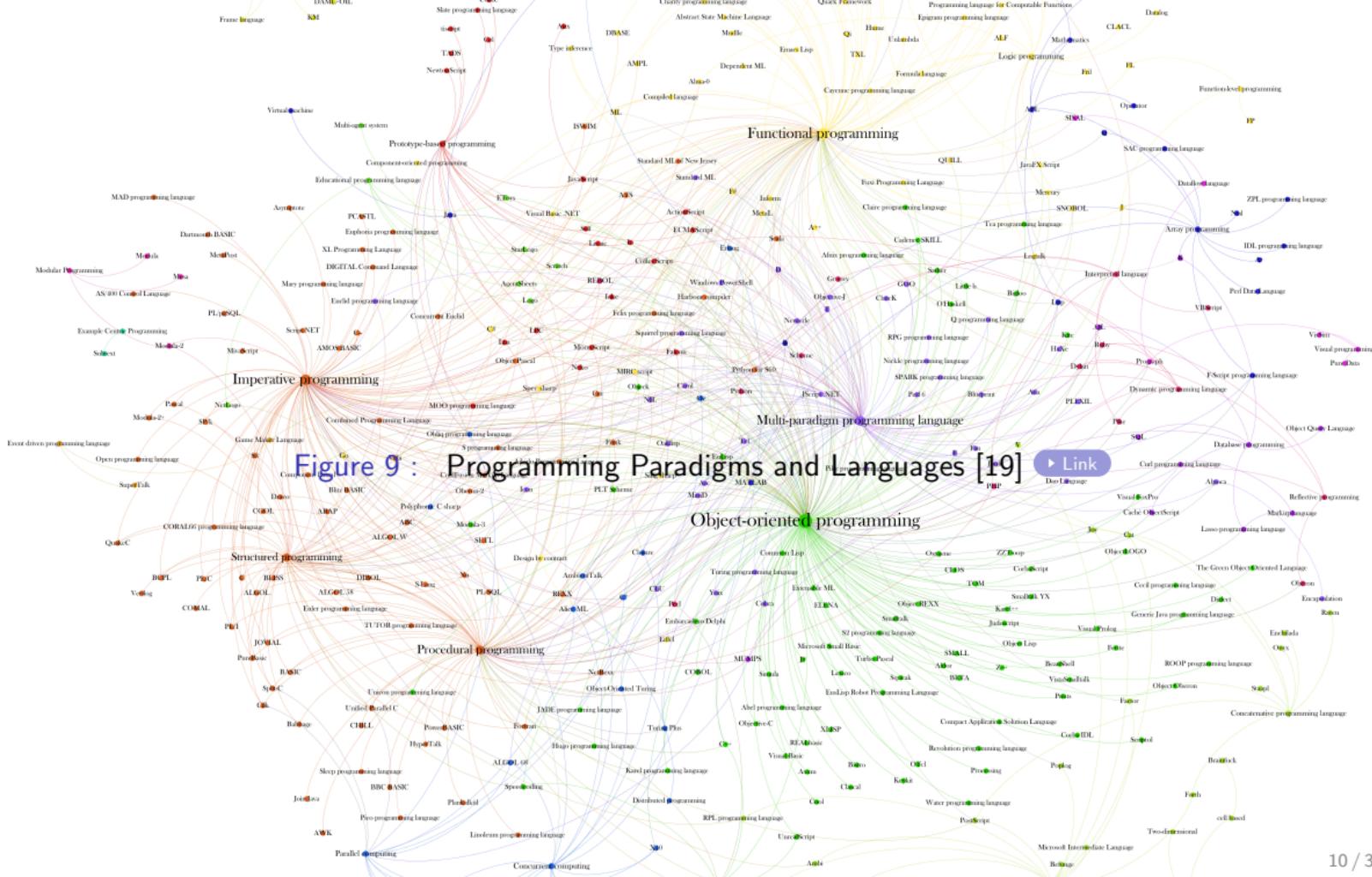


Figure 9 : Programming Paradigms and Languages [19]

Programmer's Dilemma

Increasing
number of
programming
languages.

▶ Link

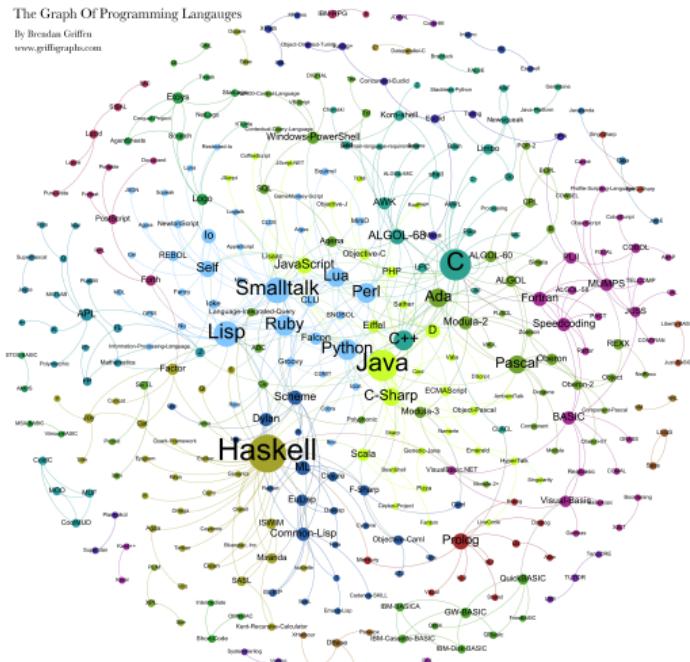


Figure 10 : The Graph of programming Languages[19]

General versus Special

General Purpose Language

Broad scope but problem needs to be moulded according to the capability of the language [20].

Special Purpose Language

Limited scope but easier to express the problem as the suitable capabilities are readily available [21].

Size Concern

Software applications have and are continuing to grow in size. On an average a software applications has hundereds of thousands of lines of code [22]. [▶ Link](#)
For example, the Windows application Paint has over 150,000 lines of code [23].

Problems with respect to PROLOG

Lacks basic features such as modules [6].

Not suitable for large programs, [24], always under 100,000 lines of code [25].

Fading usage in the industry [27, 28, 26].

As an Exercise

Conflicting characteristics.
Principle differences.

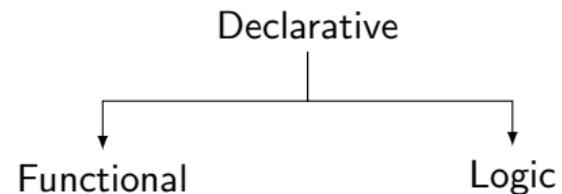


Figure 11 : Declarative Programming
Paradigms [29]

Comparison

Factor	Haskell	Prolog
Evaluation	Lazy	Strict
Type System	Strong	Weak
Working	Pattern Matching	Unification
Purity	Monadic / Pure	Ad-hoc

Table 1 : HASKELL [8] versus PROLOG [6]

For example, consider the following code,

Pattern Matching	Unification
let (x,y) = (1,2) x = 1 y = 2	(X,2) = (1,Y). X = 1. Y = 2.

Bringing programming languages closer

The language moulds itself according to the problem.

Reduce the hassle of jumping between languages.

Language helps the programmer.

Embedding Languages

Language within another language.

1. Foreign Function Interface (FFI)

A mechanism by which a program written in one programming language can make use of services written in another language [30].

For example, HASKELL provides a mechanism to embed C code in its programs [31].

2. Library or Module Extension

Replicate the features and characteristics of the target language into the host.

For example, LogLisp [32] is PROLOG library for LISP.

Merging Paradigms

Combining different programming paradigms or programming styles in one environment or in one system [33].

The idea of a multi paradigm language is to provide a framework in which programmers can work in a variety of styles, freely intermixing constructs from different paradigms [2].

SCALA

SCALA is an object functional programming language [34].

Functional	Object Oriented
Currying	Classes
Pattern Matching	Objects
Algebraic Data Types	Interfaces
Lazy Evaluation	Java Interoperation
Tail Recursion	Inheritance
Immutability	Mutability
Higher Order Functions	Dynamic Class Loading

Table 2 : SCALA Features [35]

SITREP

1. Libraries lack support for practical PROLOG features.
2. Few HASKELL based hybrid languages.
For example, CURRY [36], a functional logic programming language.
3. Literature lacks implementations.
4. Not many PROLOG environments for HASKELL.
5. No usage of *haskellian* features.

Practically speaking

1. PROLOG searches for solutions using depth first search (DFS) and may result in infinite search tree.
The *cut* operator limits the search.
2. Handling interactions with other technologies.
For example, input, output, databases among others.

Hybrid Approach

Shallow Embedding	Deep Integration
Translation	Working together
Replicate working	Merging of properties
Under utilization of host language features	Unnecessary baggage

Table 3 : Embedding versus Merging Paradigms

Hybrid Approach

Integrate core language features and embed other characteristics.

Proposed Work

1. Practical features.
2. Database capabilities.
3. Type support.
4. Handling IO.
5. Hybrid language features.

Possible Outcomes

1. Choosing a programming language made easier.
2. A complete PROLOG library with support for practical features.
3. Theoretical Model for working with IO.
4. General Embedding Scheme.

References I

- [1] Wikipedia (April 2014), Programming Languages,
http://en.wikipedia.org/wiki/Programming_language.
- [2] Wikipedia (April 2014), Programming Paradigms,
http://en.wikipedia.org/wiki/Programming_paradigm.
- [3] Spivey, Michael (1996), *An introduction to logic programming through Prolog*, Prentice-Hall, Inc.
- [4] Richard Bird, Philip Wadler (1988), *Introduction to Functional Programming*, 1st Edition, Prentice-Hall, Inc.
- [5] Krishnamurthi, Shriram (2008), Teaching programming languages in a post-linnaean age.
- [6] Wikipedia (April 2014), Prolog, <http://en.wikipedia.org/wiki/Prolog>.
- [7] Sterling, Leon (1994), *The art of Prolog: advanced programming techniques*, MIT press.
- [8] The Haskell Programming Language (September 2013), Introduction to Haskell,
<http://www.haskell.org/haskellwiki/Introduction>.

References II

- [9] The Java Programming Language (2014), Introduction to Java,
<http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>.
- [10] LurnQ, Java compared with other languages, <http://lurnq.com/lesson/Java-Introduction-and-Comparison-with-Different-Programming-Languages>.
- [11] Pseudo Random, Software engineering in the Technion semester 3,
<http://epsilonvectorplusplus.wordpress.com/2011/03/30/software-engineering-in-the-technionsemester-3/>.
- [12] US Study, Programming Languages, <http://www.ustudy.in/node/2834>.
- [13] Wordle, Creating a word cloud, <http://www.wordle.net/>.
- [14] 123rf, Abstract word cloud for Logic, http://www.123rf.com/photo_16414042_abstract-word-cloud-for-logic-with-related-tags-and-terms.html.
- [15] SWI Prolog, <http://www.swi-prolog.org/>.
- [16] Jan Kammerath (2012), The Joys of functional programming in Scala,
<http://www.kammerath.co.uk/The-Joys-of-functional-programming-in-Scala.html>.

References III

- [17] Wikimedia, Haskell Logo, <http://commons.wikimedia.org/wiki/File:Haskell-Logo.svg>.
- [18] Science by Mr.Macfarlane, Additive and Subtractive Colours, <http://mr.macfarlane.ws/index.php?/archives/14-Additive-and-Subtractive-Colours.html>.
- [19] Griff's Graphs, The Graph of Programming Languages,
<http://griffsgraphs.com/2012/07/01/programming-languages-influences/>.
- [20] Wikipedia, General Purpose Programming Languages,
http://en.wikipedia.org/wiki/General-purpose_programming_language.
- [21] Wikipedia, Special Purpose Programming Languages,
http://en.wikipedia.org/wiki/Domain-specific_programming_language.
- [22] Information is beatiful (October 2013), Codebases,
<http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>.
- [23] Wikipedia (March 2014), Paint.NET, <http://en.wikipedia.org/wiki/Paint.NET>.
- [24] Somogyi, Zoltan and Henderson, Fergus and Conway (1995), Thomas and OKeefe, Richard, Logic programming for the real world, *Proceedings of the ILPS* 83–94.

References IV

- [25] Mark Kantrowitz (August 2012), The Prolog 1000 Database,
<http://www.faqs.org/faqs/prolog/resource-guide/part1/section-9.html>.
- [26] Mark J Nelson (August 2010), Why did Prolog lose steam?,
http://www.kmjn.org/notes/prolog_lost_steam.html.
- [27] Andre Vellino, (August 2010), Prolog's Death,
<http://synthese.wordpress.com/2010/08/21/prologs-death/>.
- [28] Maarten van Emden (August 2010), Who killed Prolog?,
<http://vanemden.wordpress.com/2010/08/21/who-killed-prolog/>.
- [29] Wikipedia (March 2014), Declarative Programming,
http://en.wikipedia.org/wiki/Declarative_programming.
- [30] Wikipedia (January 2014), Foreign function interface,
http://en.wikipedia.org/wiki/Foreign_function_interface.
- [31] Haskell Wiki (August 2013), Haskell Foreign Function Interface,
http://www.haskell.org/haskellwiki/FFI_Introduction.

References V

- [32] J Alan Robinson and Ernest E Sibert (1982), Loglisp: Motivation, design, and implementation,http://aitopics.org/sites/default/files/classic/Machine_Intelligence_10/MI10-Ch20-RobinsonSiberet.pdf.
- [33] Jerinic, Ljubomir and Obradovica, Trg D (1993), A Perspective on Combining Different Programming Paradigms.
- [34] Wikipedia (April 2014), Scala Programming Language,
http://en.wikipedia.org/wiki/Scala_%28programming_language%29.
- [35] The Scala Programming Language, What is Scala,
<http://www.scala-lang.org/what-is-scala.html>.
- [36] Curry Wiki, The Curry Programming Language,
<http://www-ps.informatik.uni-kiel.de/currywiki/>.

The End

Questions?