

(AN EXTENSION TO HASKELL / CURRY) / (FUNCTIONAL LOGIC  
PROGRAMMING LANGUAGES) / (EMBEDDING PROLOG IN  
HASKELL)

---

A Thesis Proposal by  
Mehul Chandrakant Solanki  
230108015  
solanki@unbc.ca  
28 June 1990

---

Submitted to the graduate faculty of the  
MCPS  
in partial fulfillment of the requirements  
for the Thesis Proposal and  
subsequent Ph.D. in Computer Science

---

Committee Members:  
Dr. David Casperson, Committee Chair  
Dr. Alex Aravind  
Dr. Mark Shegelski

# Outline

<b>Abstract</b>	<b>iii</b>
<b>1 Embedding a Programming Language into another Programming Language</b>	<b>1</b>
1.1 The content on Blogs / Articles / Internet Discussions . . . . .	1
1.2 Related Books . . . . .	1
1.3 Related Papers . . . . .	2
1.4 Related Libraries in Haskell . . . . .	4
1.5 Possibly Related Content . . . . .	5
<b>2 Multi Paradigm Languages (Functional Logic Languages)</b>	<b>6</b>
2.1 Some Multi Paradigm Languages . . . . .	6
2.2 The content on Blogs / Articles / Internet Discussions . . . . .	6
2.3 Functional Logic Programming Languages . . . . .	7
2.4 People . . . . .	7
2.5 Functional Logic Programming Language . . . . .	7
<b>3 Introduction</b>	<b>9</b>
3.1 Problem Statement . . . . .	9
3.2 Research Approach and Contributions . . . . .	10
3.3 Thesis Statement . . . . .	10
3.4 Proposal Organization . . . . .	10
3.5 The Plan . . . . .	10
<b>4 Background</b>	<b>11</b>
<b>5 Proposed Work</b>	<b>12</b>
<b>6 Related Work</b>	<b>13</b>
6.1 Related terms . . . . .	13
6.2 Prolog Libraries in Haskell . . . . .	13
6.3 Logic Libraries in Haskell . . . . .	14
6.4 Unification Libraries in Haskell . . . . .	14
6.5 Concatenative Programming Libraries in Haskell . . . . .	14
6.6 Constraint Programming and Constraint Handling Rules . . . . .	14
6.7 Functional Logic Programming Language . . . . .	14
<b>7 Embedding a Programming Language into another Programming Language</b>	<b>16</b>
7.1 Theory . . . . .	16
7.2 Implementations . . . . .	16
7.3 Important People . . . . .	17
7.4 Miscellaneous / Possibly Related Content . . . . .	17
<b>8 Prolog in ----</b>	<b>18</b>
8.1 Theory . . . . .	18
8.2 Implementations . . . . .	18
8.3 Important People . . . . .	18
8.4 Miscellaneous / Possibly Related Content . . . . .	19
<b>9 Prolog in Haskell</b>	<b>20</b>
9.1 Theory . . . . .	20
9.2 Implementations . . . . .	21
9.3 Important People . . . . .	21
9.4 Miscellaneous / Possibly Related Content . . . . .	21

<b>10 Unifying or Marrying or Merging or Combining Programming Paradigms or Theories</b>	<b>23</b>
10.1 Theory . . . . .	23
10.2 Implementations . . . . .	23
10.3 Miscellaneous / Possibly Related Content . . . . .	23
<b>11 Functional Logic Programming Languages</b>	<b>24</b>
11.1 Theory . . . . .	24
11.2 Implementations . . . . .	24
11.3 Miscellaneous / Possibly Related Content . . . . .	24
<b>12 Quasiquotation</b>	<b>25</b>
12.1 Theory . . . . .	25
12.2 Implementations . . . . .	25
12.3 Miscellaneous / Possibly Related Content . . . . .	25
<b>13 Related Terms or Keywords</b>	<b>26</b>
<b>14 Haskell or Why Haskell ?</b>	<b>27</b>
<b>15 Prolog or Why Prolog ?</b>	<b>28</b>
<b>16 Miscellaneous or Possibly Related Content</b>	<b>29</b>
<b>17 Conclusion</b>	<b>30</b>
<b>Bibliography</b>	<b>31</b>

## Abstract

This paper proposes a different approach to improving and broadening the power, expressibility and capability of the purely functional programming language Haskell by combining and extending the methodologies of embedding of languages into one another and also marrying / merging / combining different programming paradigms. The proposal discusses the act of extending Haskell with logic programming capabilities similar to those of Prolog, a logic programming language. The embeddings and paradigm integrations are more or less towards declarative languages.

# 1 Embedding a Programming Language into another Programming Language

Embedding a programming language into another, in this we talk about embedding Prolog in Haskell.

The following are the sources or related work that can be found, ??

## 1.1 The content on Blogs / Articles / Internet Discussions

1. Lambda The Ultimate, The Programming Languages Weblog,  
<http://lambda-the-ultimate.org/node/112>
2. Takashi's Workplace (Implementation),  
<http://propella.blogspot.in/2009/04/prolog-in-haskell.html>
3. Mini Prolog for Hugs 98 (Implementation)  
<http://darcs.haskell.org/hugs98/demos/prolog/>
4. Logic Programming in Haskell,  
[http://www.haskell.org/haskellwiki/Logic\\_programming\\_example](http://www.haskell.org/haskellwiki/Logic_programming_example)
5. Haskell vs. Prolog comparison,  
<http://stackoverflow.com/questions/1932770/haskell-vs-prolog-comparison>
6. Haskell vs Prolog, or "Giving Haskell a choice"  
<http://echochamber.me/viewtopic.php?f=11&t=35369>
7. Killing Prolog and losing its steam,  
<http://vanemden.wordpress.com/2010/08/21/who-killed-prolog/>  
[http://www.kmjn.org/notes/prolog\\_lost\\_steam.html](http://www.kmjn.org/notes/prolog_lost_steam.html)

## 1.2 Related Books

1. The Reasoned Schemer, Daniel P. Friedman, William E. Byrd, Oleg Kiselyov
2. Programming Languages: Application and Interpretation, Shriram Krishnamurthi,  
Chapters 33-34 of PLAI discuss Prolog and implementing Prolog

### 1.3 Related Papers

- Papers from People

1. Type Logic Variables, K Classen,

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.2565&rep=rep1&type=pdf>

2. A Type-Safe Embedding of Constraint Handling Rules into Haskell Wei-Ngan Chin, Martin Sulzmann and Meng Wang

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.3928&rep=rep1&type=pdf>

3. Prological Features in a Functional Setting Axioms and Implementation, R Hinze

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.1016&rep=rep1&type=pdf>

4. FUNCTIONAL PEARL Combinators for breadth-first search, Micheal Spivey,

<http://journals.cambridge.org/action/displayFulltext?type=1&fid=59750&jid=JFP&volumeId=10&issueId=04&aid=59749>

5. Escape from Zurg: An Exercise in Logic Programming, Martin Erwig

<http://thelackthereof.org/docs/library/cs/functional/Erwig,%20Martin:%20Escape%20from%20Zurg%20-%20An%20Exercise%20in%20Logic%20Programming.pdf>  
<http://web.engr.oregonstate.edu/~erwig/zurg/>

- Papers from Mike Spivey and Silvija Seres

1. Embedding Prolog in Haskell / Functional Reading of Logic Programs,

[http://spivey.oriel.ox.ac.uk/mike/silvija/seres\\_haskell99.pdf](http://spivey.oriel.ox.ac.uk/mike/silvija/seres_haskell99.pdf)

This is one of the very first attempts to implement Prolog in Haskell, though there have been attempts and / or implementations of Prolog in other languages like Java(GNU Prolog, ISO Prolog as a library), Scheme(Scheme Prolog 1.2, pure Prolog interpreter, late 1980's early 1990's, 1993), Lisp (LogLisp 1982, QLog 1982) among others. There is a Hugs 98 implementation for Prolog(Mini Prolog, 1991-1996) for Hugs 1.3, but there has been no published work.

The references of this paper fall into the following categories,

- Surveys / Papers / Thesis about merging Functional and Logical Paradigms, 1,2,5,10,14,16.
- Functional Logic Languages / Embeddings, 4,6,8,9,13,17,18.
- Monads and Lazy Evaluation, 12,22,23.
- Follow up / Related Papers, 19,20,21.
- Unclassified, 14,15.

The key points from the paper,

- (a) Prolog Predicate  $\rightarrow$  Haskell Function.
- (b) Work on lazy lists, take required input produce solutions and pass it as stream.
- (c) Logical Operations  $\rightarrow$  Haskell Operations implemented using concat and map.
- (d) No extension, similar to LOGLisp(strict).
- (e) Functions to support, unification, resolution and search.
- (f) This is not a FLPL, it more of a functional language with logic capabilities, so there is no Narrowing or Residuation which are the key features of a FLPL.
- (g) The principles are general for embedding.
- (h) Only declarative features of Prolog have been implemented, no cut, assert, retract, fail(?).
- (i) Minimalistic extension, only four functions, Disjunction  $\parallel$ , Conjunction  $\&$ , Unify  $\doteq$ , Existential Quantifier (exists).
- (j) Converting a logical predicate into a pure Haskell function, bind local variables with explicit quantifiers and combining all clauses into a single equation.
- (k) Algorithm,  
Input  $\rightarrow$  Predicate + Knowledge Base  
Output  $\rightarrow$  Stream of Answers  
Done Lazily
- (l) Prolog Terms are untyped.
- (m) The function definitions are relatively simple and backtracking is naturally simulated as the evaluation is lazy.
- (n) Support for BFS is included.
- (o) The paper claims that other implementations or attempts like Babel, Kernel-LEAF, Escher, Curry **"lack semantic clarity"** (I would have to look into that).

- (p) The paper also suggests that the level of abstraction is the same as other embeddings like LOGLisp and QLog.
- (q) No implementation only Theoretical Model.
- (r) No higher order functions and nested functions.
- 2. Algebra of Logic Programming,  
[http://spivey.oriel.ox.ac.uk/mike/silviya/seres\\_iclp99.pdf](http://spivey.oriel.ox.ac.uk/mike/silviya/seres_iclp99.pdf)
- 3. The Algebra of Logic Programming,  
[http://spivey.oriel.ox.ac.uk/mike/silviya/seres\\_thesis.pdf](http://spivey.oriel.ox.ac.uk/mike/silviya/seres_thesis.pdf)
- 4. Optimisation Problems in Logic Programming : An Algebraic Approach,  
[http://spivey.oriel.ox.ac.uk/mike/silviya/seres\\_lpse00.pdf](http://spivey.oriel.ox.ac.uk/mike/silviya/seres_lpse00.pdf)
- 5. Higher Order Transformation of Logic Programs,  
[http://spivey.oriel.ox.ac.uk/mike/silviya/seres\\_lopstr00.pdf](http://spivey.oriel.ox.ac.uk/mike/silviya/seres_lopstr00.pdf)
- 6. The Algebra of Searching,  
[http://spivey.oriel.ox.ac.uk/mike/silviya/seres\\_carh99.pdf](http://spivey.oriel.ox.ac.uk/mike/silviya/seres_carh99.pdf)

## 1.4 Related Libraries in Haskell

- Prolog Libraries

- 1. Nano Prolog
- 2. Prolog
- 3. cspm-To-Prolog
- 4. prolog-graph and prolog-graph-lib
- 5. hswip,  
<https://groups.google.com/forum/#!topic/haskell-cafe/3vmCuw7NlWE>

- Logic Libraries

- 1. logict,  
<http://okmij.org/ftp/Computation/monads.html>
- 2. logic-classes
- 3. proplogic
- 4. cflp



5. logic grows on trees

- Unification Libraries

1. unification-fd

2. cmu

- Concatenative Programming Libraries

1. peg

- Constraint Programming and Constraint Handling Rules

1. monadiccp

2. monadicccp-gecode

3. csp

4. liquid fix point

## 1.5 Possibly Related Content

1. Unifying Theories of Programming, C.A.R. Hoare,

<http://www.unifyingtheories.org/>

2. Unifying Theories of Programming with Monads, Jeremy Gibbons,

<http://www.cs.ox.ac.uk/people/jeremy.gibbons/publications/utp-monads.pdf>

## 2 Multi Paradigm Languages (Functional Logic Languages)

In this section we talk about marrying or integrating the paradigms, multi paradigm programming language approach. Here we talk about combining the two most important and widely spread declarative paradigms, Functional and Logical Programming Paradigms.

### 2.1 Some Multi Paradigm Languages

Now, these days if one tries to classify programming languages according to paradigms then , a programming language will always end up being "multi paradigm"(I do not agree with this but a lot of people always tell me, including some unbc profs).

1. Scala, Object Functional Programming Language.
2. Virgil, Object Functional Programming Language.
3. CLOS, Common Lisp Object System.
4. ....???????

### 2.2 The content on Blogs / Articles / Internet Discussions

- Multi Paradigm Languages

1. Wikipedia Multiparadigm Programming Languages  
[http://en.wikipedia.org/wiki/Multi-paradigm\\_programming\\_language#Multi-paradigm](http://en.wikipedia.org/wiki/Multi-paradigm_programming_language#Multi-paradigm)  
[http://en.wikipedia.org/wiki/List\\_of\\_programming\\_languages\\_by\\_type#Multiparadigm\\_languages](http://en.wikipedia.org/wiki/List_of_programming_languages_by_type#Multiparadigm_languages)
2. Mozilla Developer Network MDN,  
<https://developer.mozilla.org/en-US/docs/multiparadigmlanguage.html>
3. Some blog called c2,  
<http://c2.com/cgi/wiki?MultiParadigmProgrammingLanguage>

- Functional Logic Programming Languages

1. FLPL Wikipedia,  
[http://en.wikipedia.org/wiki/Functional\\_logic\\_programming](http://en.wikipedia.org/wiki/Functional_logic_programming)  
[http://en.wikipedia.org/wiki/Category:Functional\\_logic\\_programming\\_](http://en.wikipedia.org/wiki/Category:Functional_logic_programming_)

languages

2. Implementation of Functional Logic Languages

<http://web.cecs.pdx.edu/~antoy/research/flp/>

3. Functional Logic Programming

<http://www.informatik.uni-kiel.de/~mh/FLP/>

## 2.3 Functional Logic Programming Languages

## 2.4 People

There are a lot of people working on this but, I found a lot of papers of two of them,

1. Michael Hanus,

<http://www.informatik.uni-kiel.de/~mh/>

2. Sergio Antoy,

<http://web.cecs.pdx.edu/~antoy/>

3. Uday S Reddy

## 2.5 Functional Logic Programming Language

1. The intergration of functions into Logic Programming : From Theory to Practice,

<http://www.informatik.uni-kiel.de/~mh/publications/papers/JLP94.html>

2. Functional Logic Programming : From theory to curry,

<http://www.informatik.uni-kiel.de/~mh/papers/GanzingerFestschrift.pdf>

3. Functional Logic Programming,

<http://dl.acm.org/citation.cfm?doid=1721654.1721675>

4. A Higher Order Rewriting Logic for FLP, <http://books.google.ca/books?hl=en&lr=&id=TSJDeaVpJyMC&oi=fnd&pg=PA153&dq=functional+logic+programming&ots=Ikp3Y-kZRV&sig=j7XQq-Hi-utdeNG54ZFkE1BeBNw#v=onepage&q=functional%20logic%20programming&f=false>

5. Toy a multiparadigm declarative system

6. A unified computation model for functional and logic programming

7. Semantics and Types in Functional Logic Programming
8. Polymorphic Types in FLP
9. A general Computation Scheme for Constraint Logic Programming

List of Functional Logic Languages

1. Mercury
2. Curry
3. Escher
4. ALF
5. Babel
6. Ciao
7. Curry
8. Life
9. LPG
10. Mercury
11. NUE-Prolog
12. Oz (Mozart Programming System)
13. TOY
14.  $\lambda$  Prolog
15. Visual Prolog (They claim it is a OOFLPL)
16. TABLOG

## 3 Introduction

The main focus of the

### 3.1 Problem Statement

The issue being discussed here is that often when a problem is to solved using a given language, it has to be moulded according to the capability the language can provide. For example a problem with a naturally recursive solution cannot take advantage of tail recursion in many imperative languages. Many problems require the system to be mutation free, but have to deal with uncontrolled side-effects and so on.

The ideal situation would be a language with a rich feature set and the ability to mould itself according to the problem. A language with ability to take the appropriate skill set and present it to the programmer will reduce the hassle of jumping between languages and / or forcibly trying to solve a problem according to a paradigm. This can be achieved in two ways,

1. Embedding

Please see Chapter 8,

[Embedding a Programming Language into another Programming Language.](#)

This approach involves, translating a complete language into the host language as an extension such as a library. The result is very shallow as all the positives as well as the negatives are brought into the host language. The negatives mentioned being, that languages from different paradigms usually have conflicting characteristics and result in inconsistent properties of the resulting embedding. Examples and further discussion on the same is provided in

2. Paradigm Integration

Please see Chapter 11,

[Unifying or Marrying or Merging or Combining Programming Paradigms or Theories](#)

This approach goes much deeper as it does not involve a direct translation. An attempt is made by taking a particular characteristic of a language and merging it with the characteristic of the host language in order to eliminate conflicts resulting in a multi paradigm language.

## **3.2 Research Approach and Contributions**

### **3.2.1 Contributions**

## **3.3 Thesis Statement**

The aim of this thesis is to add and / or extend the logical capabilities of the purely functional programming language Haskell which are derived from the logic programming language Prolog.

## **3.4 Proposal Organization**

## **3.5 The Plan**

## 4 Background

### Introduction

This section gives a broad overview of the approaches that are adopted in order to tackle the problem discussed.

## 5 Proposed Work

As discussed in the sections above, either an embedding or integration approach is taken up for the job. So there is either a very shallow approach which does not fully utilize the constructs available in the host language and just results in a mere translation of the characteristics. While the other is the fairly complex process of which results in tackling the conflicting nature of different programming paradigms, resulting in a toned down compromised language that neither takes advantages of either paradigms, sure the both the sides need to integrated but integrated languages have never really worked. Mostly the trend is to build a library for extension to replicate



## 6 Related Work

### 6.1 Related terms

1. Prolog in Haskell
2. Embedding One language into another language
3. Constraint Programming
4. Constraint Handling Rules
5. Concatenative Programming
6. Functional Logic Programming Languages
7. Residuation
8. Narrowing
9. Warren Abstraction Machine

### 6.2 Prolog Libraries in Haskell

1. Nano Prolog
2. Prolog
3. cspm-To-Prolog
4. prolog-graph and prolog-graph-lib
5. hswip
6. Embedding Prolog in Haskell, JM Spivey, `\*http://spivey.oriel.ox.ac.uk/mike/silviya/seres_haskell199.pdf`
7. Type Logic Variables, K Classen, `\*http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.37.2565&rep=rep1&type=pdf`
8. Takashi's Workplace, `\*http://propella.blogspot.in/2009/04/prolog-in-haskell.html`

### **6.3 Logic Libraries in Haskell**

1. logict
2. logic-classes
3. proplogic
4. cflp
5. logic grows on trees

### **6.4 Unification Libraries in Haskell**

1. unification-fd
2. cmu

### **6.5 Concatenative Programming Libraries in Haskell**

1. peg

### **6.6 Constraint Programming and Constraint Handling Rules**

1. monadiccp
2. monadicccp-gecode
3. csp
4. liquid fix point

### **6.7 Functional Logic Programming Language**

1. The intergration of functions into Logic Programming : From Theory to Practice,  
<http://www.informatik.uni-kiel.de/~mh/publications/papers/JLP94.html>
2. Functional Logic Programming : From theory to curry,  
<http://www.informatik.uni-kiel.de/~mh/papers/GanzingerFestschrift.pdf>
3. Functional Logic Programming,  
<http://dl.acm.org/citation.cfm?doid=1721654.1721675>

4. A Higher Order Rewriting Logic for FLP, <http://books.google.ca/books?hl=en&lr=&id=TSJDeaVpJyMC&oi=fnd&pg=PA153&dq=functional+logic+programming&ots=Ikp3Y-kZRV&sig=j7XQq-Hi-utdeNG54ZFkE1BeBNw#v=onepage&q=functional%20logic%20programming&f=false>
5. Toy a multiparadigm declarative system
6. A unified computation model for functional and logic programming
7. Semantics and Types in Functional Logic Programming
8. Polymorphic Types in FLP
9. A general Computation Scheme for Constraint Logic Programming
1. Lambda Prolog
2. Mercury
3. Curry
4. Escher

## 7 Embedding a Programming Language into another Programming Language

Embedding a language into another language,

As discussed in the

### 7.1 Theory

#### 1. Papers

- (a) Embedding an interpreted language using higher-order functions, [43]
- (b) Building domain-specific embedded languages, [25]
- (c) Embedded interpreters, [6]
- (d) Cayenne – a Language With Dependent Types, [2]
- (e) Foreign interface for PLT Scheme, [5]
- (f) Dot-Scheme: A PLT Scheme FFI for the .NET framework, [38]
- (g) Application-specific foreign-interface generation, [44]
- (h) Embedding S in other languages and environments, [34]

#### 2. Books

- (a) ?????????

#### 3. Articles / Blogs / Discussions

- (a) Embedding one language into another, [32]
- (b) Application-specific foreign-interface generation, [33]
- (c) Linguistic Abstraction, [36]
- (d) LISP, Unification and Embedded Languages, [37]

#### 4. Websites

- (a) Embedding SWI-Prolog in other applications, [15]

### 7.2 Implementations

- 1. Lots of them I guess

### **7.3 Important People**

1. ????

### **7.4 Miscellaneous / Possibly Related Content**

1. ????

## 8 Prolog in ----

Prolog in -----

### 8.1 Theory

- Papers

1. QLog, [29]
2. LogLisp Motivation, design, and implementation, [46]

- Books

1. Warrens Abstract Machine A TUTORIAL RECONSTRUCTION, [1]
2. LOGLISP: an alternative to PROLOG, [47]

- Articles / Blogs / Discussions

1. Hello

- Websites

1. Hello

### 8.2 Implementations

1. Castor : Logic paradigm for C++, [35]
2. GNU Prolog for Java, [22]
3. JLog - Prolog in Java, [26]
4. JScriptLog - Prolog in Java, [27]
5. Quintus Prolog, [39]
6. Yield Prolog, [40]
7. Racklog, [55]

### 8.3 Important People

1. ???

## 8.4 Miscellaneous / Possibly Related Content

1. ???

## 9 Prolog in Haskell

Prolog in Haskell

### 9.1 Theory

- Papers

1. Embedding Prolog in Haskell / Functional Reading of Logic Programs, [57]
2. Algebra of Logic Programming, [53]
3. The Algebra of Logic Programming, [51]
4. Optimisation Problems in Logic Programming : An Algebraic Approach, [52]
5. Higher Order Transformation of Logic Programs, [54]
6. The Algebra of Searching, [56]
7. FUNCTIONAL PEARL Combinators for breadth-first search, [58]
8. Type Logic Variables, K Classen, [9]
9. A Type-Safe Embedding of Constraint Handling Rules into Haskell Wei-Ngan Chin, Martin Sulzmann and Meng Wang, [8]
10. Prological Features in a Functional Setting Axioms and Implementation, R Hinze, [23]
11. Escape from Zurg: An Exercise in Logic Programming, [17]

- Books

1. The Reasoned Schemer, Daniel P. Friedman, William E. Byrd, Oleg Kiselyov, [13]
2. Programming Languages: Application and Interpretation, Shriram Krishnamurthi, Chapters 33-34 of PLAI discuss Prolog and implementing Prolog, [30]

- Articles / Blogs / Discussions

1. Lambda the Ultimate, Programming Languages, [31]
2. Takashi's Workplace (Implementation), [68]
3. Haskell vs. Prolog Comparison, [59]

- Websites

1. Logic Programming in Haskell, [64]



## 9.2 Implementations

1. A Prolog in Haskell, Takashi's Workplace, [68]
2. Mini Prolog for Hugs 98, [28]
3. Nano Prolog, [60]
4. Prolog, [49]
5. cspm-To-Prolog, [19]
6. prolog-graph, [4]
7. prolog-graph-lib, [48]
8. hswip, [61]

## 9.3 Important People

1. Mike Spivey
2. Silvija Seres

## 9.4 Miscellaneous / Possibly Related Content

1. Unification Libraries
  - (a) unification-fd, [62]
  - (b) cmu, [42]
2. Logic Libraries
  - (a) logicct, [11], [12]
  - (b) logic-classes, [?]
  - (c) proplogic, [20]
  - (d) cflp, [18]
  - (e) logic-grows-on-trees, [10]
3. Concatenative Programming
  - (a) peg, [14]

#### 4. Constraint Programming and Constraint Handling Rules

- (a) monadiccp, [45]
- (b) monadiccsp-gecode, [63]
- (c) csp, [3]
- (d) liquid fix point, [50]

## 10 Unifying or Marrying or Merging or Combining Programming Paradigms or Theories

Unifying / Marrying / Merging / Combining Programming Paradigms / Theories

### 10.1 Theory

- Papers

1. Unifying Theories of Programming with Monads, [21]
2. Symposium on Unifying Theories of Programming, 2006, [16].
3. Symposium on Unifying Theories of Programming, 2008, [7].
4. Symposium on Unifying Theories of Programming, 2010, [41].
5. Symposium on Unifying Theories of Programming, 2012, [67].

- Books

1. Unifying Theories of Programming, [24]

- Articles / Blogs / Discussions

1. ???

- Websites

1. ???

### 10.2 Implementations

1. Scala
2. Virgil
3. CLOS, Common Lisp Object System
4. Visual Prolog
5. ????

### 10.3 Miscellaneous / Possibly Related Content

1. ???

## 11 Functional Logic Programming Languages

### Functional Logic Programming Languages

#### 11.1 Theory

- Paper

1. FLPL Introduction Theory

- (a) Hello

2. FLPL Surveys

- (a) Hello

3. Narrowing in FLPL

- (a) Hello

4. Residuation in FLPL

- (a) Hello

5. Computation Model for FLPL

- (a) Hello

- Books

1. Hello

- Articles / Blogs / Discussions

1. Hello

- Websites

1. Hello

#### 11.2 Implementations

1. Hello

#### 11.3 Miscellaneous / Possibly Related Content

1. Hello

## 12 Quasiquotation

### 12.1 Theory

1. Papers

- (a)

2. Books

- (a)

3. Articles / Blogs / Discussions

- (a)

4. Websites

- (a) Quasiquotation Wikipedia, [66]

- (b) Quasiquotation in Haskell, [65]

### 12.2 Implementations

- 1.

### 12.3 Miscellaneous / Possibly Related Content

- 1.

## 13 Related Terms or Keywords

Related Terms / Keywords

1. Prolog in Other Languages
2. Prolog in Haskell
3. Embedding One language into another language
4. Constraint Programming
5. Constraint Handling Rules
6. Concatenative Programming
7. Functional Logic Programming Languages
8. Residuation
9. Narrowing
10. Warren Abstraction Machine
11. Foreign Function Interfaces
12. Quasiquotation
13. Programming Theory Unification

## 14 Haskell or Why Haskell ?

Haskell / Why Haskell ?

1. Hello

## 15 Prolog or Why Prolog ?

Prolog / Why Prolog ?

1. Hello



## 16 Miscellaneous or Possibly Related Content

Miscellaneous / Possibly Related Content

1. ???

## 17 Conclusion

## Bibliography

- [1] Hassan Aït-Kaci and Forêt Des Flambertins. Warrens abstract machine a tutorial reconstruction. 1999.
- [2] Lennart Augustsson. Cayenne – a language with dependent types. In *IN INTERNATIONAL CONFERENCE ON FUNCTIONAL PROGRAMMING*, pages 239–250. ACM Press, 1998.
- [3] Andrei Barbu. The csp package, August 2013. <http://hackage.haskell.org/package/csp>.
- [4] Matthias Bartsch. The prolog-graph package, September 2011. <http://hackage.haskell.org/package/prolog-graph>.
- [5] Eli Barzilay and Dmitry Orlovsky. Foreign interface for plt scheme. *on Scheme and Functional Programming*, page 63, 2004.
- [6] Nick Benton. Embedded interpreters. *Journal of Functional Programming*, 15(4):503–542, 2005.
- [7] Andrew Butterfield, editor. *Unifying Theories of Programming, Second International Symposium, UTP 2008, Dublin, Ireland, September 8-10, 2008, Revised Selected Papers*, volume 5713 of *Lecture Notes in Computer Science*. Springer, 2010.
- [8] Wei-Ngan Chin, Martin Sulzmann, and Meng Wang. A type-safe embedding of constraint handling rules into haskell. *Technical report School of Computing, National University of Singapore, Boston, MA, USA*, 2003.
- [9] Koen Claessen and Peter Ljunglöf. Typed logical variables in haskell. *Electr. Notes Theor. Comput. Sci.*, 41(1):37, 2000.
- [10] Gregory Crosswhite. The logicgrowsontrees package, September 2013. <http://hackage.haskell.org/package/LogicGrowsOnTrees>.
- [11] DanDoel. The logict package, August 2013. <http://hackage.haskell.org/package/logict>.
- [12] DanDoel. The logict package example, August 2013. <http://okmij.org/ftp/Computation/monads.html>.
- [13] Oleg Kiselyov Daniel P. Friedman, William E. Byrd. *The Reasoned Schemer*. The MIT Press, Cambridge Massachusetts, London England, 2005.
- [14] Dustin DeWeese. The peg package, April 2012. <http://hackage.haskell.org/package/peg>.
- [15] SWI Prolog Documentation. Embedding swi-prolog in other applications, June 2013. <http://www.swi-prolog.org/pldoc/man?section=embedded>.
- [16] Steve Dunne and Bill Stoddart, editors. *Unifying Theories of Programming, First International Symposium, UTP 2006, Walworth Castle, County Durham, UK, February 5-7, 2006, Revised Selected Papers*, volume 4010 of *Lecture Notes in Computer Science*. Springer, 2006.
- [17] Martin Erwig. Escape from zurg: an exercise in logic programming. *Journal of Functional Programming*, 14(03):253–261, 2004.
- [18] Sebastian Fischer. The cflp package, June 2009. <http://hackage.haskell.org/package/cflp>.
- [19] Marc Fontaine. The cspm-toprolog package, August 2013. <http://hackage.haskell.org/package/CSPM-ToProlog>.

- [20] David Fox. The proplogic package, April 2012. <http://hackage.haskell.org/package/PropLogic>.
- [21] Jeremy Gibbons. Unifying theories of programming with monads. In *Unifying Theories of Programming*, pages 23–67. Springer, 2013.
- [22] GNU. Gnu prolog for java, August 2010. <http://www.gnu.org/software/gnuprologjava/>.
- [23] Ralf Hinze et al. Prological features in a functional setting axioms and implementation. In *Fuji International Symposium on Functional and Logic Programming*, pages 98–122. Citeseer, 1998.
- [24] Charles Anthony Richard Hoare and Jifeng He. *Unifying theories of programming*, volume 14. Prentice Hall Englewood Cliffs, 1998.
- [25] Paul Hudak. Building domain-specific embedded languages. *ACM Comput. Surv.*, 28(4es):196, 1996.
- [26] JLogic. Jlog - prolog in java, September 2012. <http://jlogic.sourceforge.net/index.html>.
- [27] JLogic. Jscriptlog - prolog in javascript, September 2012. <http://jlogic.sourceforge.net/index.html>.
- [28] Mark P Jones. Mini-prolog for hugs 98, June 1996. <http://darcs.haskell.org/hugs98/demos/prolog/>.
- [29] H Jan Komorowski. Qlog: The programming environment for prolog in lisp. *Logic Programming*, pages 315–324, 1982.
- [30] Shriram Krishnamurthi. *Programming languages: Application and interpretation*, chapter 33-34, pages 295–305, 307–311. Brown Univ., 2007.
- [31] The Programming Languages Weblog Lambda The Ultimate. Embedding prolog in haskell, July 2004. <http://lambda-the-ultimate.org/node/112>.
- [32] The Programming Languages Weblog Lambda The Ultimate. Embedding one language into another, March 2005. <http://lambda-the-ultimate.org/node/578>.
- [33] The Programming Languages Weblog Lambda The Ultimate. Application-specific foreign-interface generation, October 2006. <http://lambda-the-ultimate.org/node/2304>.
- [34] Duncan Temple Lang. Embedding s in other languages and environments. In *Proceedings of DSC*, volume 2, page 1, 2001.
- [35] MPprogramming.com. Castor : Logic paradigm for c++, August 2010. <http://www.mpprogramming.com/cpp/>.
- [36] Kurt Nrmak Department of Computer Science Aalborg University Denmark. Linguistic abstraction, July 2013. [http://people.cs.aau.dk/~normark/prog3-03/html/notes/languages\\_themes-intro-sec.html#languages\\_intro-sec\\_section-title\\_1](http://people.cs.aau.dk/~normark/prog3-03/html/notes/languages_themes-intro-sec.html#languages_intro-sec_section-title_1).
- [37] University of Maryland Medical Center. Lisp, unification and embedded languages, October 2012. <http://www.cs.unm.edu/~luger/ai-final2/LISP/>.
- [38] Pedro Pinto. Dot-scheme: A plt scheme ffi for the .net framework. In *Workshop on Scheme and Functional Programming*. Citeseer, 2003.
- [39] Quintus Prolog. Embeddability, December 2003. <http://quintus.sics.se/isl/quintuswww/site/embed.html>.

- [40] Yield Prolog. Yield prolog, October 2011. <http://yieldprolog.sourceforge.net/>.
- [41] Shengchao Qin, editor. *Unifying Theories of Programming - Third International Symposium, UTP 2010, Shanghai, China, November 15-16, 2010. Proceedings*, volume 6445 of *Lecture Notes in Computer Science*. Springer, 2010.
- [42] John Ramsdell. The cmu package, February 2013. <http://hackage.haskell.org/package/cmu>.
- [43] Norman Ramsey. Embedding an interpreted language using higher-order functions and types. In *Proceedings of the 2003 workshop on Interpreters, virtual machines and emulators*, pages 6–14. ACM, 2003.
- [44] John Reppy and Chunyan Song. Application-specific foreign-interface generation. In *Proceedings of the 5th international conference on Generative programming and component engineering*, pages 49–58. ACM, 2006.
- [45] Maik Riechert. The monadiccp package, July 2013. <http://hackage.haskell.org/package/monadiccp>.
- [46] J Alan Robinson and Ernest E Sibert. Loglisp: Motivation, design, and implementation, 1982.
- [47] John Alan Robinson and EE Silbert. *LOGLISP: an alternative to PROLOG*. School of Computer and Information Science, Syracuse University, 1980.
- [48] Daniel Seidel. The prolog-graph-lib package, June 2012. <http://hackage.haskell.org/package/prolog-graph-lib>.
- [49] Daniel Seidel. The prolog package, June 2012. <http://hackage.haskell.org/package/prolog>.
- [50] Eric Seidel. The liquid-fixpoint package, September 2013. <http://hackage.haskell.org/package/liquid-fixpoint>.
- [51] Silvija Seres. *The algebra of logic programming*. PhD thesis, 2001.
- [52] Silvija Seres and Shin-Cheng Mu. Optimisation problems in logic programming: an algebraic approach. 2000.
- [53] Silvija Seres, J Michael Spivey, and CAR Hoare. Algebra of logic programming. In *ICLP*, pages 184–199, 1999.
- [54] Silvija Seres and Michael Spivey. Higher-order transformation of logic programs. In *Logic Based Program Synthesis and Transformation*, pages 57–68. Springer, 2001.
- [55] Dorai Sitaram. Racklog: Prolog-style logic programming, January 2014. <http://docs.racket-lang.org/racklog/index.html>.
- [56] JM Spivey and Silvija Seres. The algebra of searching. *Festschrift in honour of CAR Hoare*, 1999.
- [57] JM Spivey and Silvija Seres. Embedding prolog in haskell. In *Proceedings of Haskell*, volume 99, pages 1999–28. Citeseer, 1999.
- [58] Michael Spivey. Functional pearls combinators for breadth-first search. *Journal of Functional Programming*, 10(4):397–408, 2000.
- [59] Stackoverflow. Haskell vs. prolog comparison, December 2009. <http://stackoverflow.com/questions/1932770/haskell-vs-prolog-comparison>.
- [60] Jurrien Stutterheim. The nanoprolog package, December 2011. <http://hackage.haskell.org/package/NanoProlog>.

- [61] Evgeny Tarasov. The hswip package, August 2010. <http://hackage.haskell.org/package/hswip>.
- [62] Wren Thornton. The unification-fd package, July 2012. <http://hackage.haskell.org/package/unification-fd>.
- [63] Jan Tikovsky. The monadiccp-gecode package, January 2014. <http://hackage.haskell.org/package/monadiccp-gecode>.
- [64] Haskell Website. Logic programming example, February 2010. [http://www.haskell.org/haskellwiki/Logic\\_programming\\_example](http://www.haskell.org/haskellwiki/Logic_programming_example).
- [65] Haskell Website. Quasiquote in haskell, January 2014. <http://www.haskell.org/haskellwiki/Quasiquote>.
- [66] Wikipedia. Quasiquote, November 2013. <http://en.wikipedia.org/wiki/Quasiquote>.
- [67] Burkhart Wolff, Marie-Claude Gaudel, and Abderrahmane Feliachi, editors. *Unifying Theories of Programming, 4th International Symposium, UTP 2012, Paris, France, August 27-28, 2012, Revised Selected Papers*, volume 7681 of *Lecture Notes in Computer Science*. Springer, 2013.
- [68] Takashi's Workplace. A prolog in haskell, April 2009. <http://propella.blogspot.in/2009/04/prolog-in-haskell.html>.