# The Thesis To-Do-List

*Mehul C. Solanki*
solanki@unbc.ca

# Contents

# 1 Prolog Grammar

– Play around with and understand the parser and syntax of **prolog-0.2.0.1** library. ☑

– Add floating point support. ☑

– Study syntax and grammar from SICSTUS PROLOG and incorporate it into the parser above. ☐

– Incorporate those tricky PROLOG terms with single quotes along with support for ASCII codes ☐

– Go through Dr. Casperson's code for syntax, parser, print and quasi quoting. ☐

– Add arithmetic support during parsing that is, if an atom is of the form,
3 + 4 + 5
after parsing should be
12. ☐

– Other PROLOG syntax such as equality expressions, ☐
=(X,Y). or
=(1,1). or
=('a','b').

– The parser from the library parses grammars (CFG, DCG) but I do not know how well but everything works well. Moreover, there is no special syntax support for recognising clauses of the grammar form. ☐

– Do something about quasi quotation and learn more about template HASKELL. ☐

– Look at **hswip** and **Nanoprolog** for anything new. ☐

– Try and implement a small working example like *append* which was mentioned in one of the papers. ☐

# 2 Unification

– Look at unification in PROLOG-0.2.0.1. ☐

– Look at unification-fd. ☐

– Look at monad-unify. □

– Read about CURRY and how they do residuation and narrowing in their most natural form, because the evaluation works just like HASKELL when the variables are confined to the left that is when the matching has to be done just one way. □

– Look at the **Hugs98** for implementing variable search strategies. What is this "Andorra Engine" ? □

– Look at an unofficial implementaion at TAKASHI'S Workplace. □

– A new library called **compdata** which is based on Compositional Data Types and unifies terms using the "Martelli and Montanari" Unification algorithm. □

– If nothing else happens, try and implement the algorithm from above. □

# 3 IO and other practical features

– Cut, how the hell does it work ? □

– Fail, how the hell does it work ? □

– Assert, how the hell does it work ? □

– SetOf, BagOf, how the hell does it work ? □

– consult , how the hell does it work ? □

– IO operations like reading a file, how the hell does it work ? □

# 4 Publications

- Will functional programming take over the world?

  This paper ha by far the most content mostly thoughts and speculations based on experiences in Haskell collected from programming, reading and other sources. I feel this will be the last in terms of completion.

  Content is been added every day but the pace is slow.

- Prolog in Haskell : A Survey.

  Now this is the one which is expected to be finished very soon even though it does not have much content in it.

  Content is been added every week but the pace is very slow.

- Prolog in Haskell : Reloaded.

  Probably after the work is near completed.

  Content is been added once every ..... and the pace is very very slow.