

## **Relational Programming in miniKanren**



# **Relational Programming in miniKanren**

William E. Byrd

© 2013 William E. Byrd

Typeset by the author in X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X, using Tufte-Style Book from <http://www.LaTeXTemplates.com>.  
June 9, 2013 version.



This work is licensed under a Creative Commons Attribution 3.0 Unported License. (CC BY 3.0)

<http://creativecommons.org/licenses/by/3.0/>

*For my H211 students: Indiana University, Fall 2010 & 2011.*

*Learning with always trumps learning from.*

—Woodie Flowers



# *Contents*

*Preface*      ix

*Introduction*      1

*Conclusion*      3

*Bibliography*      5





# Preface

## Audience

This book is written for intermediate-to-advanced programmers, computer science students, and researchers. For this book, *intermediate* means that you are comfortable writing simple recursive procedures in a functional programming language, such as Scheme, Racket, Closure, Lisp, ML, or Haskell. I also assume you have a reading knowledge of Scheme. No knowledge of relational programming, logic programming, programming language theory, or miniKanren is required. You must, however, prepare to have your mind blown by the awesomeness that is relational programming.

If you want to learn about relational programming, but are new to programming, are you in luck! I co-wrote another book just for you: *The Reasoned Schemer*<sup>1</sup>. You might first read *The Little Schemer*<sup>2</sup>, a very gentle introduction to recursion and functional programming.

If you are an experienced programmer, but weak on recursion, you, too, might benefit from *The Little Schemer*. If you are comfortable with recursion, but not functional programming, good introductions include *Scheme and the Art of Programming*<sup>3</sup> and the classic *Structure and Interpretation of Computer Programs*<sup>4</sup>.

If you are an experienced functional programmer, but do not know Scheme, the beginning of *Structure and Interpretation of Computer Programs* should get you up to speed, while *The Scheme Programming Language, 4th Edition*<sup>5</sup> describes the language in detail.

William E. Byrd  
Salt Lake City, Utah  
June 2013

<sup>1</sup> D. P. Friedman, W. E. Byrd, and O. Kiselyov. *The Reasoned Schemer*. MIT Press, Cambridge, MA, 2005

<sup>2</sup> D. P. Friedman and M. Felleisen. *The Little Schemer (4th ed.)*. MIT Press, Cambridge, MA, USA, 1996

<sup>3</sup> G. Springer and D. P. Friedman. *Scheme and the Art of Programming*. MIT Press, Cambridge, MA, USA, 1989

<sup>4</sup> H. Abelson and G. J. Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, MA, USA, 2nd edition, 1996  
(full text at <http://mitpress.mit.edu/sicp/full-text/book/book.html>)

<sup>5</sup> R. K. Dybvig. *The Scheme Programming Language, 4th Edition*. The MIT Press, 4th edition, 2009  
(full text at <http://www.scheme.com/tspl4/>)



# *Introduction*

*gl hf!*

(Traditional greeting in the Koprulu Sector)



## *Conclusion*

G.G.

—Sean “Day[9]” Plott



# *Bibliography*

H. Abelson and G. J. Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, MA, USA, 2nd edition, 1996.

R. K. Dybvig. *The Scheme Programming Language, 4th Edition*. The MIT Press, 4th edition, 2009.

D. P. Friedman and M. Felleisen. *The Little Schemer (4th ed.)*. MIT Press, Cambridge, MA, USA, 1996.

D. P. Friedman, W. E. Byrd, and O. Kiselyov. *The Reasoned Schemer*. MIT Press, Cambridge, MA, 2005.

G. Springer and D. P. Friedman. *Scheme and the Art of Programming*. MIT Press, Cambridge, MA, USA, 1989.