

## **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 13, 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, Clojure, Lisp, ML, or Haskell. I also assume you have a reading knowledge of Scheme. No knowledge of relational programming, logic programming, or programming language theory is required.

If you want to learn about relational programming, but are new to programming, Dan Friedman, Oleg Kiselyov, and I have written a book just for you, called *The Reasoned Schemer*<sup>1</sup>. In that book we assume you are familiar with the material in *The Little Schemer*<sup>2</sup>, which is 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.

## Goals

An important goal of the book is to make other resources on miniKanren more accessible.

Another goal is to show how miniKanren relations can be derived from pure Scheme functions in a systematic manner.

One goal of this book is to present a variety of non-trivial miniKanren relations, and to show how these relations were derived. Many of these examples draw from academic papers on miniKanren, and focus on programming languages theory (interpreters, type infer-

I have attempted to deliver [these lectures] in a spirit that should be recommended to all students embarking on the writing of their PhD theses: imagine that you are explaining your ideas to your former smart, but ignorant, self, at the beginning of your studies!

—Richard P. Feynman  
*The Feynman Lectures on Computation*

<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, 1996

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

<sup>4</sup> H. Abelson and G. J. Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, MA, 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/>)

encers, etc.). Other examples, such as finite state machines, should be of interest to a wider audience of programmers.

### *Margin Notes*

This book is typeset in the style of Edward Tufte’s magnificent and beautiful *The Visual Display of Quantitative Information*<sup>6</sup>. I share Tufte’s love of margin notes, and use them in this book to help solve the problem of addressing readers with widely varying knowledge of computer science and programming. To make the book accessible as possible, in the main text I assume the reader is the hypothetical *intermediate-level* programmer or student described in the *Audience* section above. In the margin notes, however, anything goes.

William E. Byrd  
Salt Lake City, Utah  
June 2013

This book is set using the “Tufte-Style Book”  $\LaTeX$  style, freely available from <http://www.LaTeXTemplates.com>

<sup>6</sup> E. R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, 1986

Another great lover of marginalia was David Foster Wallace (1962–2008). The *Harry Ransom Center*’s DFW collection includes heavily annotated books from Wallace’s personal library: <http://www.hrc.utexas.edu/press/releases/2010/dfw/books/>. Wallace’s love of margin notes is best demonstrated by his essay “Host”, in:

D. F. Wallace. *Consider the Lobster and Other Essays*. Little, Brown and Co., 2005

# *Introduction*

*g! 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, 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, 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, 1989.

E. R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, 1986.

D. F. Wallace. *Consider the Lobster and Other Essays*. Little, Brown and Co., 2005.