The 30 Year Horizon

Manuel Bronstein  William Burge  Timothy Daly
James Davenport  Michael Dewar  Martin Dunstan
Albrecht Fortenbacher  Patrizia Gianni  Johannes Grabmeier
Jocelyn Guidry  Richard Jenks  Larry Lambe
Michael Monagan  Scott Morrison  William Sit
Jonathan Steinbach  Robert Sutor  Barry Trager
Stephen Watt  Jim Wen  Clifton Williamson

Volume 3: Axiom Programmers Guide
Portions Copyright (c) 2005 Timothy Daly

The Blue Bayou image Copyright (c) 2004 Jocelyn Guidry

Portions Copyright (c) 2004 Martin Dunstan
Portions Copyright (c) 2007 Alfredo Portes
Portions Copyright (c) 2007 Arthur Ralfs
Portions Copyright (c) 2005 Timothy Daly

Portions Copyright (c) 1991-2002, The Numerical ALgorithms Group Ltd.
All rights reserved.

This book and the Axiom software is licensed as follows:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of The Numerical ALgorithms Group Ltd. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Inclusion of names in the list of credits is based on historical information and is as accurate as possible. Inclusion of names does not in any way imply an endorsement but represents historical influence on Axiom development.

<table>
<thead>
<tr>
<th>Name</th>
<th>Name</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Michael Albaugh</td>
<td>Cyril Alberga</td>
<td>Roy Adler</td>
</tr>
<tr>
<td>Christian Aistleitner</td>
<td>Richard Anderson</td>
<td>George Andrews</td>
</tr>
<tr>
<td>S.J. Atkins</td>
<td>Henry Baker</td>
<td>Martin Baker</td>
</tr>
<tr>
<td>Stephen Balzac</td>
<td>Yuri Baransky</td>
<td>David R. Barton</td>
</tr>
<tr>
<td>Thomas Baruchel</td>
<td>Gerald Baumgartner</td>
<td>Gilbert Baumslag</td>
</tr>
<tr>
<td>Michael Becker</td>
<td>Nelson H. F. Beebe</td>
<td>Jay Belanger</td>
</tr>
<tr>
<td>David Bindel</td>
<td>Fred Blair</td>
<td>Vladimir Bondarenko</td>
</tr>
<tr>
<td>Mark Botch</td>
<td>Raoul Bourquin</td>
<td>Alexandre Bouyer</td>
</tr>
<tr>
<td>Karen Braman</td>
<td>Peter A. Broadbery</td>
<td>Martin Brock</td>
</tr>
<tr>
<td>Samuel Bronstein</td>
<td>Stephen Buchwald</td>
<td>Florian Bundschuh</td>
</tr>
<tr>
<td>Luanne Burns</td>
<td>William Burge</td>
<td>Ralph Byers</td>
</tr>
<tr>
<td>Quentin Carpent</td>
<td>Robert Caviness</td>
<td>Bruce Char</td>
</tr>
<tr>
<td>Ondrej Certik</td>
<td>Tzu-Yi Chen</td>
<td>Cheekai Chin</td>
</tr>
<tr>
<td>David V. Chudnovsky</td>
<td>Gregory V. Chudnovsky</td>
<td>Mark Clements</td>
</tr>
<tr>
<td>James Cloos</td>
<td>Jia Zhao Cong</td>
<td>Josh Cohen</td>
</tr>
<tr>
<td>Christophe Conil</td>
<td>Don Coppersmith</td>
<td>George Corliss</td>
</tr>
<tr>
<td>Robert Corless</td>
<td>Gary Cornell</td>
<td>Meino Cramer</td>
</tr>
<tr>
<td>Jeremy Du Croz</td>
<td>David Cyganski</td>
<td>Nathaniel Daly</td>
</tr>
<tr>
<td>Timothy Daly Sr.</td>
<td>Timothy Daly Jr.</td>
<td>James H. Davenport</td>
</tr>
<tr>
<td>David Day</td>
<td>James Demmel</td>
<td>Didier Deshommes</td>
</tr>
<tr>
<td>Michael Dewar</td>
<td>Jack Dongarra</td>
<td>Jean Della Dora</td>
</tr>
<tr>
<td>Gabriel Dos Reis</td>
<td>Claire DiCrescendo</td>
<td>Sam Dooley</td>
</tr>
<tr>
<td>Lionel Ducos</td>
<td>Iain Duff</td>
<td>Lee Duhem</td>
</tr>
<tr>
<td>Martin Dunstan</td>
<td>Brian Dupe</td>
<td>Dominique Duval</td>
</tr>
<tr>
<td>Robert Edwards</td>
<td>Heow Eide-Goodman</td>
<td>Lars Erickson</td>
</tr>
<tr>
<td>Richard Fateman</td>
<td>Bertfried Fauser</td>
<td>Stuart Feldman</td>
</tr>
<tr>
<td>John Fletcher</td>
<td>Brian Ford</td>
<td>Albrecht Fortenbacher</td>
</tr>
<tr>
<td>George Frances</td>
<td>Constantine Frangos</td>
<td>Timothy Freeman</td>
</tr>
<tr>
<td>Korriinn Fu</td>
<td>Marc Gaetano</td>
<td>Rudiger Gebauer</td>
</tr>
<tr>
<td>Van de Geijn</td>
<td>Kathy Gerber</td>
<td>Patricia Gianni</td>
</tr>
<tr>
<td>Gustavo Goertkin</td>
<td>Samantha Goldrich</td>
<td>Holger Gollan</td>
</tr>
<tr>
<td>Teresa Gomez-Diaz</td>
<td>Laureano Gonzalez-Vega</td>
<td>Stephen Gortler</td>
</tr>
<tr>
<td>Johannes Grabmeier</td>
<td>Matt Grayson</td>
<td>Klaus Ebbe Grue</td>
</tr>
<tr>
<td>James Griesmer</td>
<td>Vladimir Grinberg</td>
<td>Oswald Gschmitzer</td>
</tr>
<tr>
<td>Ming Gu</td>
<td>Jocelyn Guidry</td>
<td>Gaetan Hache</td>
</tr>
<tr>
<td>Steve Hague</td>
<td>Satoshi Hamaguchi</td>
<td>Sven Hammarling</td>
</tr>
<tr>
<td>Mike Hansen</td>
<td>Richard Hanson</td>
<td>Richard Harke</td>
</tr>
<tr>
<td>Bill Hart</td>
<td>Vilya Harvey</td>
<td>Martin Hassner</td>
</tr>
<tr>
<td>Arthur S. Hathaway</td>
<td>Dan Hatton</td>
<td>Waldek Hebisch</td>
</tr>
<tr>
<td>Karl Hegbloom</td>
<td>Ralf Hemmecke</td>
<td>Henderson</td>
</tr>
<tr>
<td>Antoine Hersen</td>
<td>Roger House</td>
<td>Gernot Hueber</td>
</tr>
<tr>
<td>Pietro Iglio</td>
<td>Alejandro Jakubi</td>
<td>Richard Jenks</td>
</tr>
<tr>
<td>William Kahan</td>
<td>Kyriakos Kalorkoti</td>
<td>Kai Kaminski</td>
</tr>
</tbody>
</table>
Grant Keady  Wilfrid Kendall  Tony Kennedy
Ted Kosan  Paul Kosinski  Klaus Kusche
Bernhard Kutzler  Tim Lahey  Larry Lambe
Kaj Laurson  George L. Legendre  Franz Lehner
Frederic Lehobey  Michel Levaud  Howard Levy
Ren-Cang Li  Rudiger Loos  Michael Lucks
Richard Luczak  Camm Maguire  Francois Maltey
Alasdair McAndrew  Bob McElrath  Michael McGettrick
Edi Meier  Ian Meikle  David Mentre
Victor S. Miller  Gerard Milmeister  Mohammed Mobarak
H. Michael Moeller  Michael Monagan  Marc Moreno-Maza
Scott Morrison  Joel Moses  Mark Murray
William Naylor  Patrice Naudin  C. Andrew Neff
John Nelder  Godfrey Nolan  Arthur Norman
Jinzhong Niu  Michael O’Connor  Summat Oemrawsingh
Kostas Oikonomou  Humberto Ortiz-Zuazaga  Julian A. Padget
Bill Page  David Parnas  Susan Pelzel
Michel Petitot  Didier Pinchon  Ayal Pinkus
Frederick H. Pitts  Jose Alfredo Portes  Gregorio Quintana-Orti
Claude Quitte  Arthur C. Ralfs  Norman Ramsey
Anatoly Raportirenko  Albert D. Rich  Michael Richardson
Guillerme Reis  Huan Ren  Renaud Rioboo
Jean Rivlin  Nicolas Robidoux  Simon Robinson
Raymond Rogers  Michael Rothstein  Martin Rubey
Philip Santas  Alfred Scheerhorn  William Schelter
Gerhard Schneider  Martin Schoenert  Marshall Schor
Frithjof Schulze  Fritz Schwarz  Steven Segletes
V. Sima  Nick Simicich  William Sit
Elena Smirnova  Jonathan Steinbach  Fabio Stumbo
Christine Sundaresan  Robert Sutor  Moss E. Sweedler
Eugene Surowitz  Max Tegmark  T. Doug Telford
James Thatcher  Balbir Thomas  Mike Thomas
Dylan Thurston  Steve Toleque  Barry Trager
Themos T. Tsikas  Gregory Vanuxem  Bernhard Wall
Stephen Watt  Jaap Weel  Juergen Weiss
M. Weller  Mark Wegman  James Wen
Thorsten Werther  Michael Wester  R. Clint Whaley
James T. Wheeler  John M. Wiley  Berhard Will
Clifton J. Williamson  Stephen Wilson  Shmuel Winograd
Robert Wisbauer  Sandra Wityak  Waldemar Wiwianka
Knut Wolf  Yanyang Xiao  Liu Xiaojun
Clifford Yapp  David Yun  Vadim Zhytnikov
Richard Zippel  Evelyn Zoernack  Bruno Zuercher
Dan Zwillinger
Contents

1 Details for Programmers  1
  1.1 Examining Internals ........................................ 1
  1.2 Makefile ..................................................... 4

2 Bibliography  5

3 Index  9
New Foreword

On October 1, 2001 Axiom was withdrawn from the market and ended life as a commercial product. On September 3, 2002 Axiom was released under the Modified BSD license, including this document. On August 27, 2003 Axiom was released as free and open source software available for download from the Free Software Foundation’s website, Savannah.

Work on Axiom has had the generous support of the Center for Algorithms and Interactive Scientific Computation (CAISS) at City College of New York. Special thanks go to Dr. Gilbert Baumslag for his support of the long term goal.

The online version of this documentation is roughly 1000 pages. In order to make printed versions we’ve broken it up into three volumes. The first volume is tutorial in nature. The second volume is for programmers. The third volume is reference material. We’ve also added a fourth volume for developers. All of these changes represent an experiment in print-on-demand delivery of documentation. Time will tell whether the experiment succeeded.

Axiom has been in existence for over thirty years. It is estimated to contain about three hundred man-years of research and has, as of September 3, 2003, 143 people listed in the credits. All of these people have contributed directly or indirectly to making Axiom available. Axiom is being passed to the next generation. I’m looking forward to future milestones.

With that in mind I’ve introduced the theme of the “30 year horizon”. We must invent the tools that support the Computational Mathematician working 30 years from now. How will research be done when every bit of mathematical knowledge is online and instantly available? What happens when we scale Axiom by a factor of 100, giving us 1.1 million domains? How can we integrate theory with code? How will we integrate theorems and proofs of the mathematics with space-time complexity proofs and running code? What visualization tools are needed? How do we support the conceptual structures and semantics of mathematics in effective ways? How do we support results from the sciences? How do we teach the next generation to be effective Computational Mathematicians?

The “30 year horizon” is much nearer than it appears.

Tim Daly
CAISS, City College of New York
November 10, 2003 ((iHy))
Chapter 1

Details for Programmers

Axiom maintains internal representations for domains. There are functions for examining the internals of objects of a particular domain.

1.1 Examining Internals

One useful function is `devaluate` which takes an object and returns a Lisp pair. The CAR of the pair is the Axiom type. The CDR of the pair is the object representation. For instance, consider the session where we create a list of objects using the domain `List(Any)`.

```
(1) -> w:=[1,7.2,"luanne",3*x^2+5,_,
       (3*x^2+5)::FRAC(POLY(INT)),_,
       (3*x^2+5)::POLY(FRAC(INT)),_,
       (3*x^2+5)::EXPR(INT)]$LIST(ANY)

2 2 2 2
(1) [1,7.2,"luanne",3x + 5,3x + 5,3x + 5,3x + 5]
Type: List(Any)
```

The first object, `1` is a primitive object that has the domain `PI` and uses the underlying Lisp representation for the number.

```
(2) -> devaluate(1)$Lisp

(2) 1
Type: SExpression
```

The second object, `7.2` is a primitive object that has the domain `FLOAT` and uses the underlying Lisp representation for the number, in this case, itself a pair whose CAR is the floating point base and whose CDR is the mantissa,
CHAPTER 1. DETAILS FOR PROGRAMMERS

(3) \( \rightarrow \) devaluate(7.2)$Lisp

(3) \((265633114661417543270 \cdot - 65)\)

Type: SExpression

The third object, "luanne" is from the domain STRING and uses the Lisp string representation.

(4) \( \rightarrow \) devaluate("luanne")$Lisp

(4) luanne

Type: SExpression

Now we get more complicated. We illustrate various ways to store the formula \(3x^2 + 5\) in different domains. Each domain has a chosen representation.

(5) \( \rightarrow \) devaluate(3*x^2+5)$Lisp

(5) \((1 \times (2 \ 0 \ . \ 3) \ 0 \ 0 \ . \ 5))\)

Type: SExpression

The fourth object, \(3x^2 + 5\) is from the domain POLY(INT). It is stored as the list

\((1 \times (2 \ 0 \ . \ 3) \ 0 \ 0 \ . \ 5))\)

From the domain POLY (Vol 10.3, POLY) we see that

\[
\text{Polynomial}(R: \text{Ring}): \ldots
\]
\[
\quad = \text{SparseMultivariatePolynomial}(R, \text{Symbol}) \text{ add } \ldots
\]

So objects from this domain are represented as SMP(INT,Ssymbol). From this domain we see that

\[
\text{SparseMultivariatePolynomial}(R: \text{Ring}, \text{VarSet}: \text{OrderedSet}): \ldots
\]
\[
\quad = \text{add}
\]
\[
\quad \quad \text{--representations}
\]
\[
\quad D := \text{SparseUnivariatePolynomial}(\%)
\]

So objects from this domain are represented as a SUP(INT)

\[
\text{SparseUnivariatePolynomial}(R: \text{Ring}): \ldots
\]
\[
\quad = \text{PolynomialRing}(R, \text{NonNegativeInteger}) \text{ add}
\]

So objects from this domain are represented as PR(INT,NNI)

\[
\text{PolynomialRing}(R: \text{Ring}, E: \text{OrderedAbelianMonoid}): \ldots
\]
\[
\quad \text{FreeModule}(R, E) \text{ add}
\]
\[
\quad \quad \text{--representations}
\]
\[
\quad \quad \text{Term:= Record}(k: E, c: R)
\]
\[
\quad \quad \text{Rep:= List Term}
\]
1.1. EXAMINING INTERNALS

So objects from this domain are represented as **FM(INT,NNI)**

```
FreeModule(R:Ring,S:OrderedSet): == IndexedDirectProductAbelianGroup(R,S) add
   -- representations
   Term:= Record(k:S,c:R)
   Rep:= List Term
```

So objects from this domain are represented as **IDPAG(INT,NNI)**

```
```

So objects from this domain are represented as **IDPAM(INT,NNI)**

```
   -- representations
   Term:= Record(k:S,c:A)
   Rep:= List Term
```

So objects from this domain are represented as **IDPO(INT,NNI)**

```
IndexedDirectProductObject(A:SetCategory,S:OrderedSet): == add
   -- representations
   Term:= Record(k:S,c:A)
   Rep:= List Term
```

(6) -> devaluate((3*x^2+5)::FRAC(POLY(INT)))$Lisp

(6) ((1 x (2 0 . 3) (0 0 . 5)) 0 . 1)  
Type: SExpression

(7) -> devaluate((3*x^2+5)::POLY(FRAC(INT)))$Lisp

(7) (1 x (2 0 3 . 1) (0 0 5 . 1))  
Type: SExpression

(8) -> devaluate((3*x^2+5)::EXPR(INT))$Lisp

(8) ((1 [[x,0,%symbol()()()],[NIL,1,1024] (2 0 . 3) (0 0 . 5)) 0 . 1)  
Type: SExpression

(9) -> devaluate(w)$Lisp

(9)  
(((PositiveInteger) . 1) ((Float) 265633114661417543270 . - 65)
((String) . luanne) ((Polynomial (Integer)) 1 x (2 0 . 3) (0 0 . 5))
((Fraction (Polynomial (Integer))) (1 x (2 0 . 3) (0 0 . 5)) 0 . 1)
((Polynomial (Fraction (Integer))) 1 x (2 0 3 . 1) (0 0 5 . 1))

((Expression (Integer))
(1 [[x,0,%symbol()()]],NIL,1,1024] (2 0 . 3) (0 0 . 5)) 0 . 1)
)

Type: SExpression

1.2 Makefile

This book is actually a literate program [Knut92] and can contain executable source code. In particular, the Makefile for this book is part of the source of the book and is included below.
Chapter 2

Bibliography
Bibliography

Chapter 3

Index