
Making open source textbooks, and diagrams with AIDraTex

Seth D. Bergmann

Abstract

This article describes a new paradigm for the creation of textbooks, using L^AT_EX. Macros for the automatic generation of figures and diagrams are described.

1 Introduction

Textbooks have traditionally been produced by publishing companies which provide many services, and revenue, to authors. These services include:

- Editors responsible for the acquisition and production process
- Copyeditors responsible for formatting, proof-reading the text for errors and making stylistic improvements
- Graphic artists responsible for the production of technical figures and diagrams
- Sales representatives responsible for the marketing of the book

All these services, plus other corporate employees, bricks and mortar, etc., contribute to the extremely high cost of textbooks for readers.

This article proposes a new paradigm for the creation of textbooks, based on the *open source* model of software development [2]. Open source software is developed by many developers who may be at remote locations, cooperating on the Internet. In addition to distributing the software product, the developers also distribute the source documents, making it easy for others to make corrections, enhancements, or extensions to the original product. Since there is no cost for the product, there is no revenue for the developers. Some common examples of open source software: Java, Apache, and L^AT_EX.

Developers who contribute to a well-known successful product are rewarded by the status associated with the product, and can perhaps leverage this status with job offers or promotions.

In applying the open source paradigm to textbooks, authors can collaborate on the generation of high-quality books. As the books become widely adopted, errata are found and corrected, and other authors are able to add sections and chapters to improve the book. The authors (usually teachers or professors, but also students) receive no direct compensation, but if the book is widely adopted, they receive recognition for their contributions, which can be a positive factor in applying for jobs, tenure, promotion, etc.

The most significant impediment to open-source textbooks is the production of high quality figures and diagrams. This article makes the claim that diagrams can be drawn with L^AT_EX macros, rather than traditional drawing tools. The distinction is that I am proposing the use of a markup language as opposed to WYSIWYG tools.

The development of the graphics macros can be a difficult, and time-consuming process. However, once the macros are perfected, they can be used to draw many diagrams with a professional appearance. If a minor stylistic change is needed, (e.g., arrowheads in state diagrams need to be enlarged) all that is needed is a change to the macro that draws state diagrams, and all such diagrams are regenerated automatically.

When several authors are collaborating on a textbook, it is not necessary that they all have the expertise needed to develop new macros. This could be the responsibility of one developer, with expertise in the graphics commands, to produce new macros as needed by the other authors.

This paper lists four currently available open source textbooks (and a fifth in development) and shows some examples of the diagrams that are generated automatically.

2 Some currently available open source textbooks

I have developed four open source textbooks, which are currently being distributed by the Campbell Library at Rowan University, with a total of over 40,000 downloads worldwide. They are available at rdw.rowan.edu/oer:

Title	Downloads
Compiler Design: Theory, Tools, and Examples [3]	25,843
Computer Organization with MIPS [4]	11,739
Introduction to Computer Science with Java [6]	4,757
Computer Science Principles (Java, Python, and C++ editions) [5]	1,398
Cryptology with Bitcoin and Blockchain [in development]	0

(Number of downloads as of March 2022.)

The source files for these books, along with recently updated versions, are available from the author’s web site: cs.rowan.edu/~bergmann/books.

I’d like to single out Allen Downey, at Olin College of Engineering, who has also published several open source textbooks [7]. He has been one of the first, if not the first, to promote this new paradigm, making educational resources affordable for students and school districts everywhere.

Another developer of open source textbooks is Jim Hefferon of Saint Michael’s College in Vermont, who has authored books on the Theory of Computation [10] and Linear Algebra [9]. Hefferon has made extensive use of L^AT_EX graphics packages to draw figures and diagrams.

3 Diagrams

In this section I expose some of the more complex diagrams that can be drawn with special-purpose macros. I use the graphics packages DraTex and AlDraTex, developed at Ohio State University by Eitan Gurari [8] in the 1990s; they are available in T_EX Live and on CTAN (ctan.org/pkg/dratex). These packages include fairly primitive structures such as circles, lines, rectangles, and text boxes. They also include state diagrams, trees, grids, and other structures, with parameters controlling the size and appearance.¹

As with L^AT_EX, the DraTex and AlDraTex packages are extensible: new macro commands can be defined using existing macro commands. I have used this feature in developing macros for the examples shown in this section.

3.1 Charts, trees, and other diagrams with AlDraTex

The AlDraTex macros for chart and tree diagrams support diagrams with various attributes and formats, by specifying parameter values in the macro call. AlDraTex also allows the user to define other diagrams.

3.2 Chart diagrams

AlDraTex has macros which draw charts:

Pie charts may be round or oval, with shaded or painted sections, and labeled with internal or external labels.

XY charts permit graphs of two-dimensional data on an XY grid, with labeling on the axes and points, and with a continuous graph or discrete points

¹ Other graphics packages with similar capabilities include `tikz`, available on github, and `metapost`, available from the TeX User Group.

Bar charts allow variable size charts, with various properties on the bars, including 3-dimensional bars and tailored or painted bars, and labels on the bars and axes.

3.3 Tree diagrams

Trees, in various formats, may be drawn with AlDraTex. The user simply specifies the label on each node and the number of children. Nodes may be in various shapes, including circle, oval, rectangle, and text-only. The format of the tree may allow for horizontal-vertical edges or straight edges, and optional labels on the edges. Trees may be oriented vertically or horizontally.

3.4 Other diagrams

As an extensible language, with AlDraTex it is possible to define nodes of any shape for the diagrams described above, as well as macros for drawing state diagrams, such as those needed for finite automata.

3.5 Logic diagrams

The *Computer Organization* textbook [4] makes extensive use of diagrams. The primitive drawing commands of DraTex can be used to produce diagrams of logic gates — AND, OR, XOR, NOT, etc.

These gates can then be connected with each other to form higher level components such as encoders, decoders, multiplexers, adders, and arithmetic/logic units. An example of a logic diagram is shown in Figure 1.

One commonly used logic diagram is a canonical sum-of-products diagram, corresponding to a logic expression which is the logical OR of several logical ANDs. I developed a macro to draw sum-of-products diagrams with up to four variables. An example is shown in Figure 3. The user of this macro need only specify the names of the variables and the boolean true/false values for each product in the sum-of-products expression to be diagrammed.

3.6 Karnaugh maps

Another important diagram for the *Computer Organization* textbook is the Karnaugh Map (or K-Map). It is used to reduce a logic expression to its simplest sum-of-products form. The macro which draws K-Maps needs input for the positions of 1’s (or don’t-cares) in the map, as well as the grouping of the 1’s. In theory a macro could deduce the groupings, but this feature is not currently available. The macro can draw K-Maps with three or four variables. An example of a K-Map with four variables is shown in Figure 5.

3.7 Object diagrams

The most interesting diagram macro that I have developed is for the object diagram construct commonly used in textbooks on object-oriented programming. I used this macro in the textbook *Introduction to Computer Science with Java* [6].

In an object-oriented language, such as Java or C++, the state of an object is determined by the instance variables² in the object's class. Each instance variable in an object may store either primitive data or a reference³ to another object. This is a recursive definition, and thus the object diagram, which is a visual representation of an object, is a recursive diagram. Consequently the macro which draws object diagrams is also recursive. An example of an object diagram is shown in Figure 7.

4 Examples of diagrams

Examples of diagrams which were described above are shown on the following pages. In each case the commands which generated the diagram are shown in the next figure. The full macro definitions and packages are available at cs.rowan.edu/~bergmann/books.

5 Summary

In this article I have presented a new paradigm for the production of free open-source textbooks. This paradigm has been used to produce four computer science textbooks, with a fifth book in development. The main contribution of this work is the capability of drawing figures and diagrams with \LaTeX macros. These macros are based on the extensible graphics packages `DraTex` and `AIDraTex`.

References

- [1] S.D. Bergmann. *Compiler Design: Theory, Tools, and Examples*. Wm. C. Brown Publishers, Dubuque, 1994.
- [2] S.D. Bergmann. Open source textbooks: A paradigm derived from open source software. *Publishing Research Quarterly* 30(1):1–10, March 2014.
- [3] S.D. Bergmann. *Compiler Design: Theory, Tools, and Examples*. Campbell Library at Rowan University, Glassboro, NJ, 2022. <https://rdw.rowan.edu/oer/1>
- [4] S.D. Bergmann. *Computer Organization with MIPS*. Campbell Library at Rowan University, Glassboro, NJ, 2022.

² Instance variables are also known as non-static fields in Java, or member data in C++.

³ A Java reference is similar to a C++ pointer, though there are restrictions on what can be done with a reference.

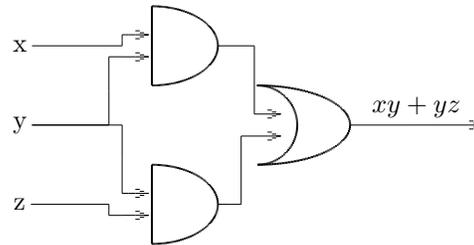


Figure 1: An implementation of the boolean function $xy + yz$ using logic gates

```
\DiagramSpec(\Inp & \Gate & \Wire)
\Diagram
  %% Inputs: Var, xLoc, yLoc
  (x,-50,30 & y,-50,0 & z,-50,-30)

  %% Gates: ID, type, inputs, xLoc, yLoc
  (And1,0,2,0,30,, &
   And2,0,2,0,-30,, & Or,1,2,50,0,,)

  %% Wires: srcID, targetID, input#
  (x, And1,1 & y, And1,2 &
   y, And2,1 & z,And2,2 &
   And1,Or,1 & And2,Or,2)
\MoveToNode(Orout,1,0.5) \Move(50,0)
\FcNode(result)
\Edge(Orout,result)
\EdgeLabel(--$xy + yz$--)
```

Figure 2: \LaTeX code used to draw Figure 1

- [5] S.D. Bergmann. *Computer Science Principles*. Campbell Library at Rowan University, Glassboro, NJ, 2022. Editions available for several programming languages.
- [6] S.D. Bergmann. *Introduction to Computer Science with Java Programming*. Campbell Library at Rowan University, Glassboro, NJ, 2022. <https://rdw.rowan.edu/oer/2>
- [7] A.B. Downey. *Python for Software Design: How to Think Like a Computer Scientist*. Cambridge University Press, New York, 2009.
- [8] E.M. Gurari. *TEX and LATEX: Drawing and Literate Programming*. McGraw-Hill, New York, 1994.
- [9] J. Hefferon. *Linear Algebra*. Orthogonal Publishing, Ann Arbor, MI, 2022. hefferon.net/linearalgebra
- [10] J. Hefferon. *Theory of Computation*. hefferon.net, 2022. hefferon.net/computation

◇ Seth D. Bergmann
Rowan University
Glassboro, NJ, USA
[bergmann \(at\) rowan dot edu](mailto:bergmann@rowan.edu)
<https://cs.rowan.edu/~bergmann/>

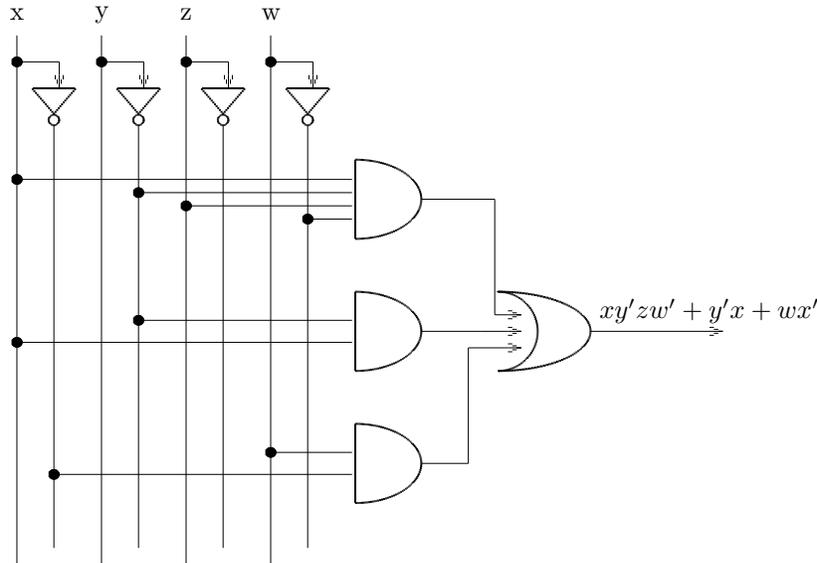


Figure 3: A sum of products logic diagram for the expression $xy'zw' + y'x + wx'$

```

\DiagramSpec(\SOP & \Ins & \Ands)
\Diagram
(4,3,xy'zw'+y'x+wx')    %% 4 variables, 3 terms
(x & y & z & w)         %% Variable names
%% Terms: #vars,
%%         var,
%%         0=negated
(4,x,1,y,0,z,1,w,0 &   %% xy'zw'
 2,y,0,x,1,0,0,0,0 &   %% y'x
 2,w,1,x,0,0,0,0,0)    %% wx'
)
    
```

Figure 4: L^AT_EX code used to draw Figure 3

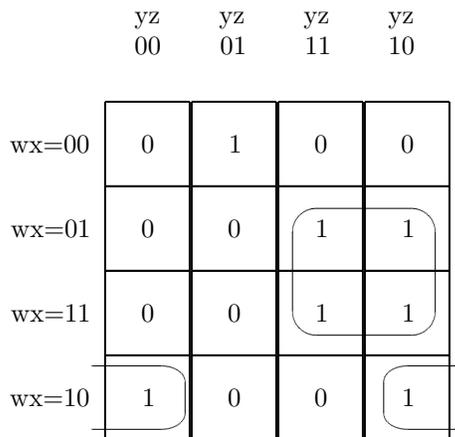


Figure 5: A K-map for the boolean expression $w'x'y'z + w'xyz + w'xy'z' + wx'yz + wx'yz' + wx'y'z' + wx'yz'$. A 1x2 group and a 2x2 group are identified. The minimized expression is $xy + wx'z' + w'x'y'z$.

```

\KmapFourTop(0,1,0,0, 0,0,1,1, 32)
\KmapFourBot(0,0,1,1, 1,0,0,1, 32)

\Group(4,7,32,)           %% xy
\Group(h,10,32,h)        %% wx'z'
    
```

Figure 6: L^AT_EX code used to draw Figure 5

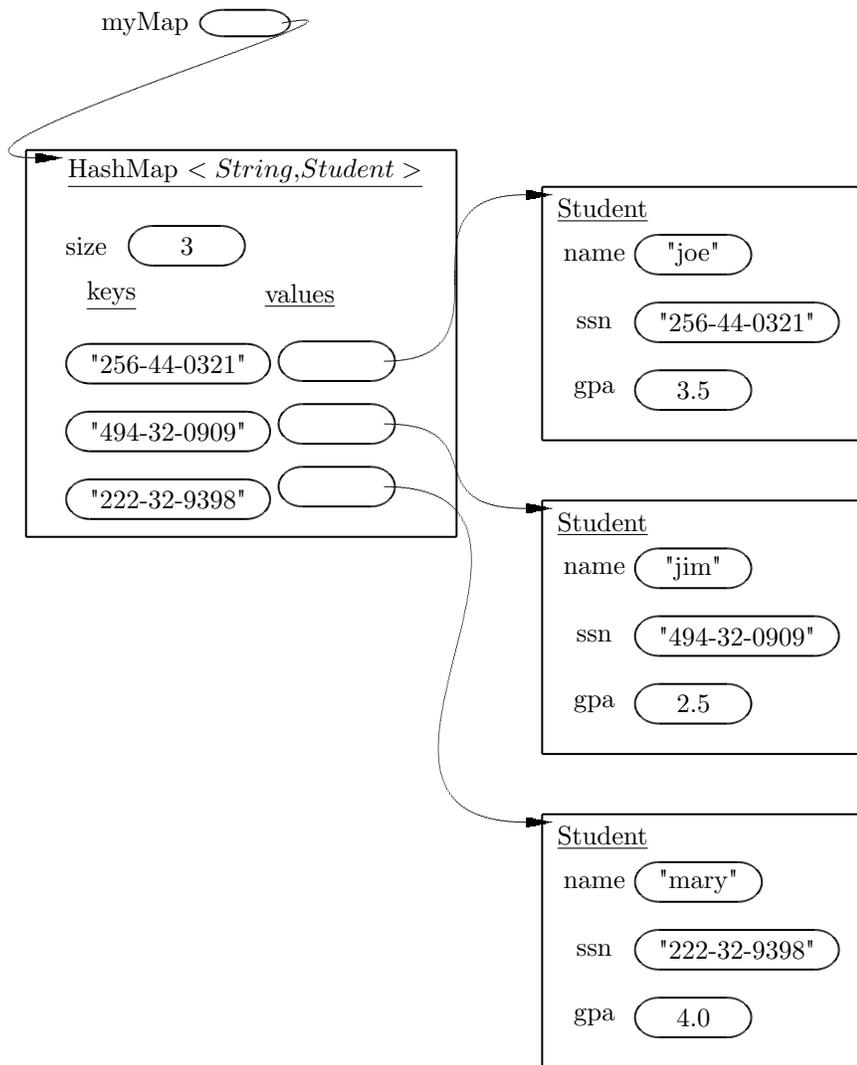


Figure 7: An object diagram showing the value of the variable `myMap` storing a reference to a map, after three entries have been added

```

\Indirect \Table <student1Prims>
{ 1, name, "joe" &
  1, ssn, "256-44-0321" &
  1, gpa, 3.5
}
\Indirect \Table <student10bjs>
{ 0, , , , , , }
\Indirect \Table <student2Prims>
{ 1, name, "jim" &
  1, ssn, "494-32-0909" &
  1, gpa, 2.5
}
\Indirect \Table <student20bjs>
{ 0, , , , , , }
\Indirect \Table <student3Prims>
{ 1, name, "mary" &
  1, ssn, "222-32-9398" &
  1, gpa, 4.0
}
\Indirect \Table <student30bjs>
{ 0, , , , , , }
\Indirect \Table <keyPrims>
{ 1, ,"256-44-0321" &
  1, ,"494-32-0909" &
  1, ,"222-32-9398" }
\Indirect \Table <keyObjs>
{ 0,0, , , , , }
\Indirect \Table <valuePrims>
{ 0, , }
\Indirect \Table <valueObjs>
{ 1, ,Student,student1Prims,
  student10bjs, , , &
  1, ,Student,student2Prims,
  student20bjs, , , &
  1, ,Student,student3Prims,
  student30bjs, , ,
}
\Obj (5, myMap,
  HashMap~$<String{,}Student>$,
  3, keyPrims, keyObjs,
  valuePrims, valueObjs)
  
```

Figure 8: L^AT_EX code used to draw Figure 7.