Functional Programming: the Glue for Introducing Computing through Data Science

Kathi Fisler
Brown University and Bootstrap
(Pyreteer and Racketeer)
joint work with Shriram Krishnamurthi
“Language Wars” have been around for some time
University Computer Science/Industry Programmers

Some pre-college education
(C++, Visual Basic, Java)

Expanded pre-college education
(Scratch, AppInventor, Javascript, ...)

Data Science!
What’s the best language for teaching data science?

STOP

What problem are you trying to solve?
A challenge for programming education

Provide computing and data science education to all students in ways that support equity and diversity in computing while working within the constraints of schools and university departments.

- Not just the “natural” programmers
- Regardless of intended major
- Without enough CS/Informatics teachers to go around

Pedagogies that support students with different skills
- Problems that resonate across cultures
- Acknowledge impacts of computing on people and society

A Data-Centric Approach to Computing via Functional Programming
I come at this from ...

Computing Education, grades 6-16

author of K-12 CS standards in multiple US states

CS department administration of a large undergrad program

Researcher in computing education, formal methods, and verification
Data Centric?

lead with data, not control operators

lead with data that students recognize and care about

lead with questions that teachers from other disciplines care about

Not just any old functional programming!

Not just the “natural” programmers

Regardless of intended major

Without enough CS/Informatics teachers to go around

Pedagogies that support students with different skills

Problems that resonate across cultures

Acknowledge impacts of computing on people and society
Atoms (nums, string, images)

Structs
Lists
List[Structs]
Trees
Mutual Recursion
Generative Recursion
State

Ooh! Datasets!
Could we do this earlier?
Rich, structured data, in a familiar format!

Recipe for preparing data for analysis:
normalize, locate suspicious data, use visualization to sanity check, analyze

As much data engineering as data science

Questions like “how many tickets sold with a student discount” let students explore problem decomposition in a concrete, physical format

Many authentic tasks that can raise impact issues

Task Planning!
R, at its heart, is a functional language. This means that it has certain technical properties, but more importantly that it lends itself to a style of problem solving centred on functions. Below I’ll give a brief overview of the technical definition of a functional language, but in this book I will primarily focus on the functional style of programming, because I think it is an extremely good fit to the types of problem you commonly encounter when doing data analysis.

Recently, functional techniques have experienced a surge in interest because they can produce efficient and elegant solutions to many modern problems. A functional style tends to create functions that can easily be analysed in isolation (i.e. using only local information), and hence is often much easier to automatically optimise or parallelise. The traditional weaknesses of functional languages, poorer performance and sometimes unpredictable memory usage, have been much reduced in recent years. Functional programming is complementary to object-oriented programming, which has been the dominant programming paradigm for the last several decades.
Data-Centric Intro to Computing (DCIC)

- Images
- Tables
- Lists
- Datatypes
- Trees
- State
- Hashtables

Task Planning

Atomics (nums, string, images)
- Structs
- Lists
- List[Structs]
- Trees
- Mutual Recursion
- Generative Recursion
- State

Socially Responsible Computing
Data-Centric Intro to Computing (DCIC)

Images
Tables
Lists
Datatypes
Trees
State
Hashtables

Structure of code follows structure of image
Data-Centric Intro to Computing (DCIC)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Name</td>
<td>Email</td>
<td>Num</td>
<td>Tickets</td>
<td>Discount Code</td>
</tr>
<tr>
<td>2</td>
<td>Josie Zhao</td>
<td><a href="mailto:jo@email.com">jo@email.com</a></td>
<td>2</td>
<td></td>
<td>BIRTHDAY</td>
</tr>
<tr>
<td>3</td>
<td>Sam Ochibe</td>
<td><a href="mailto:s@web.com">s@web.com</a></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Bart Simple</td>
<td><a href="mailto:bart@simpson.org">bart@simpson.org</a></td>
<td>5</td>
<td></td>
<td>STUDENT</td>
</tr>
<tr>
<td>5</td>
<td>Ernie O’Mailey</td>
<td>ernie.mail.com</td>
<td>0</td>
<td></td>
<td>none</td>
</tr>
<tr>
<td>6</td>
<td>Alvina Velasquez</td>
<td>alvie@schooledu</td>
<td>3</td>
<td></td>
<td>student</td>
</tr>
<tr>
<td>7</td>
<td>Zander</td>
<td>zandaman</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Shweta Chowpati</td>
<td><a href="mailto:sno@this.org">sno@this.org</a></td>
<td>three</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
tickets = table: name, email, num, discount, delivery
row: “Josie Zhao”, “jo@email.com”, 2, “BIRTHDAY”, “email”
row: “Sam Ochibe”, “s@web.com”, 1, “”, “pickup”
... end

build-column(tickets, “fee”, lam(r :: Row): r[“num”] * 10 end)
```

Or import from Google Drive
Higher-order functions (resembles those on tables) then introduce (only) structural recursion
**Data-Centric Intro to Computing (DCIC)**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Name</td>
<td>Email</td>
<td>Num</td>
<td>Tickets</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Josie Zhao</td>
<td><a href="mailto:jo@mail.com">jo@mail.com</a></td>
<td>2</td>
<td>BIRTHDAY</td>
<td>email</td>
</tr>
<tr>
<td>3</td>
<td>Sam Ochibe</td>
<td><a href="mailto:s@sweb.com">s@sweb.com</a></td>
<td>1</td>
<td></td>
<td>pickup</td>
</tr>
<tr>
<td>4</td>
<td>Bart Simple</td>
<td><a href="mailto:bart@simpson.org">bart@simpson.org</a></td>
<td>5</td>
<td>STUDENT</td>
<td>yes</td>
</tr>
<tr>
<td>5</td>
<td>Ernie O'Malley</td>
<td>ernie.mail.com</td>
<td>0</td>
<td>none</td>
<td>email</td>
</tr>
<tr>
<td>6</td>
<td>Alvina Velasquez</td>
<td>alvie@schooledu</td>
<td>3</td>
<td>student</td>
<td>email</td>
</tr>
<tr>
<td>7</td>
<td>Zander</td>
<td>zandaman</td>
<td>10</td>
<td></td>
<td>email</td>
</tr>
<tr>
<td>8</td>
<td>Shweta Chowpatti</td>
<td><a href="mailto:snc@this.org">snc@this.org</a></td>
<td>three</td>
<td></td>
<td>pickup</td>
</tr>
</tbody>
</table>

**How to represent timestamps?**

*string? number?*

Many opportunities to discuss data-design tradeoffs and connect to real-world issues (e.g., storing lists and structs in CSV files in systematic ways, “falsehoods programmers believe about names/dates”)
Tables aren’t always a useful data structure

Challenge of searching for ancestors highlights the need for data structures beyond tables (here comes CS ...)
Data-Centric Intro to Computing (DCIC)

Images
Tables
Lists
Datatypes
Trees
State
Hashtables

Back to tables! (pandas)
Tables (could easily build into Racket)

Python-esque syntax (we know, but ...)

Examples part of function definitions, separate from tests
Data-Centric Intro to Computing (DCIC)

Images
Tables
Lists
Datatypes
Trees
State
Hashtables

Back to tables! (pandas)

CSCI 0111
Computing Foundations: Data

550 students so far
non-CS majors LOVE it
many have become/added CS

Data-centricity: a challenge and opportunity for computing education

Authors: Shriram Krishnamurthi, Kathi Fisler Authors Info & Affiliations
Publication: Communications of the ACM • July 2020 • https://doi.org/10.1145/3408056

Textbook in progress (Mar 2021)
Checkpoint!

lead with data, not control operators

lead with data that students recognize and care about

lead with questions that teachers from other disciplines care about

Not just the “natural” programmers

Regardless of intended major

Without enough CS/Informatics teachers to go around

Pedagogies that support students with different skills

Problems that resonate across cultures

Acknowledge impacts of computing on people and society
Computing and Data-Science in K-12

Integrate intro computing/data science into existing pre-college classes (algebra, science, social studies)

Questions and projects that matter in the host discipline

Assess learning in the host discipline

Also backed by research ...

<table>
<thead>
<tr>
<th>Domain/Range</th>
<th>≈ Types!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>≈ Tests!</td>
</tr>
<tr>
<td>Functional Code</td>
<td>≈ Symbolic Form</td>
</tr>
</tbody>
</table>

Danger and Target Movement

Directions: Use the Design Recipe to write a function `update-danger`, which takes in the danger’s x-coordinate and produces the next x-coordinate.

Contract and Purpose Statement

Every contract has three parts...

<table>
<thead>
<tr>
<th>Function name</th>
<th>Domain</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>example1</td>
<td>input1</td>
<td>output1</td>
</tr>
<tr>
<td>example2</td>
<td>input2</td>
<td>output2</td>
</tr>
</tbody>
</table>

Examples

Write some examples, then circle and label what changes...

<table>
<thead>
<tr>
<th>Example</th>
<th>Domain</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>example3</td>
<td>input3</td>
<td>output3</td>
</tr>
</tbody>
</table>

Definition

Write the definition, giving variable names to all your input values...

```plaintext
fun (variable1) {
    function name (variable1);
}
```
Leading from data supports computing for all

- Enables authentic tasks in many fields
- Raises impacts of computing, which matters for equity
- Can accomplish a lot with small amounts of code

*Functional programming can get us there!*

*Data science gives us a new foothold*

*but the linguistic and pedagogic details matter A LOT*
Novices don’t understand these fields well enough to decide!

Little content alignment, so switching requires starting over 😞

Novices don’t understand these fields well enough to decide!
Many intro CS courses are here

Appealing to students across campus

Increasing calls for Social Responsibility

Frequently overlooked

Needs nontrivial CS

Should have Data/Stats

This is our space!!!

Questions?
kfisler@cs.brown.edu