LambdaConf 2018 – 6/05/2018

Metaphorisms: Deriving Divide-and-Conquer Recursive Programs from Relational Specifications

Presenter: William Harvey, PhD

Principal Engineer, OCLC

OCLC

José N. Oliveira, PhD

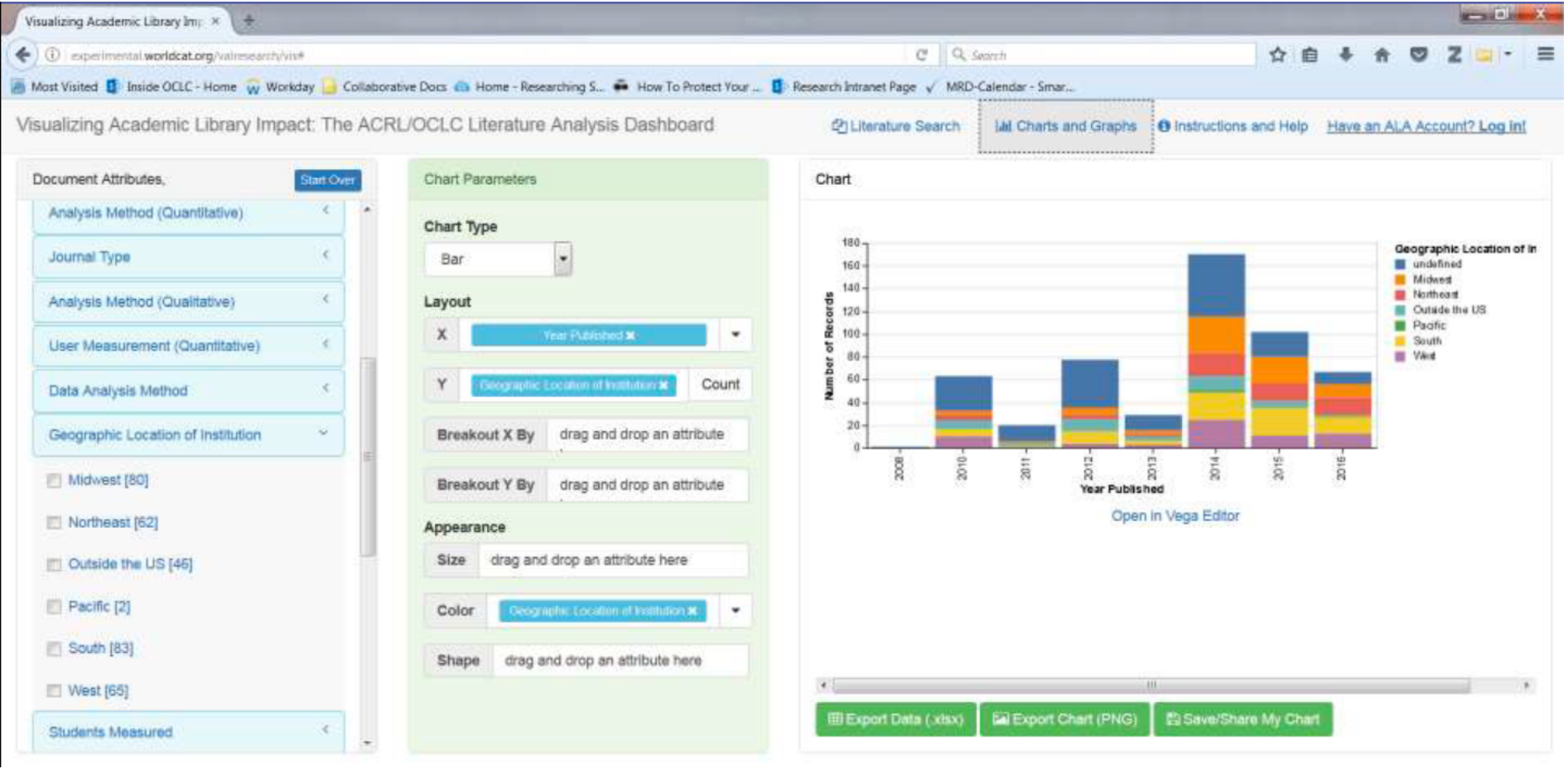
Professor, University of Minho, Portugal

Talk Overview

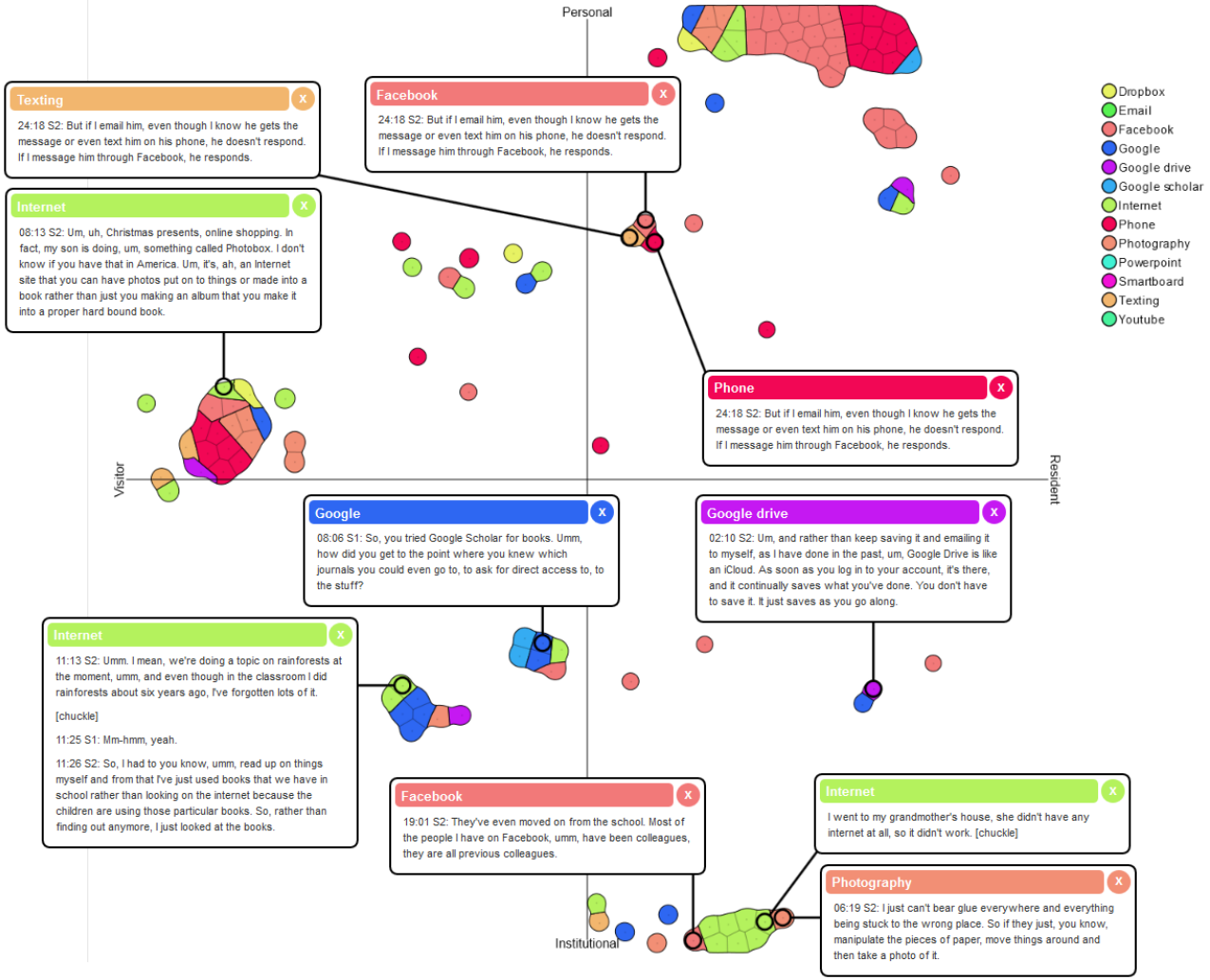
- FP and category theory at OCLC
- Our research problem
- Introduction to **metaphorisms**
- Examples

About OCLC

- We help libraries
- We’re a nonprofit global library cooperative
- Thousands of library members in more than 100 countries
- WorldCat: Largest online public access catalog in the world
- We maintain the Dewey Decimal System


Visualizing Academic Library Im
experimental.worldcat.org/valresearch/vis#
Search
Most Visited
Inside OCLC - Home
Workday
Collaborative Docs
Home - Researching S...
How To Protect Your ...
Research Intranet Page
MRD-Calendar - Smar...
Visualizing Academic Library Impact: The ACRL/OCLC Literature Analysis Dashboard
Literature Search
Charts and Graphs
Instructions and Help
Have an ALA Account? Log in!
Document Attributes.
Start Over
Analysis Method (Quantitative)
Journal Type
Analysis Method (Qualitative)
User Measurement (Quantitative)
Data Analysis Method
Geographic Location of Institution
Midwest [80]
Northeast [62]
Outside the US [46]
Pacific [2]
South [83]
West [65]
Students Measured
Chart Parameters
Chart Type
Bar
Layout
X
Year Published
Y
Geographic Location of Institution
Count
Breakout X By
drag and drop an attribute
Breakout Y By
drag and drop an attribute
Appearance
Size
drag and drop an attribute here
Color
Geographic Location of Institution
Shape
drag and drop an attribute here
Chart
Number of Records
180
160
140
120
100
80
60
40
20
0
2008
2010
2011
2012
2013
2014
2015
2016
Year Published
Geographic Location of In
undefined
Midwest
Northeast
Outside the US
Pacific
South
West
Open in Vega Editor
Export Data (.xlsx)
Export Chart (PNG)
Save/Share My Chart

UKU7 (After)


Personal
Visitor
Resident
Institutional
Dropbox
Email
Facebook
Google
Google drive
Google scholar
Internet
Phone
Photography
Powerpoint
Smartboard
Texting
Youtube
Texting
24:18 S2: But if I email him, even though I know he gets the message or even text him on his phone, he doesn't respond. If I message him through Facebook, he responds.
Facebook
24:18 S2: But if I email him, even though I know he gets the message or even text him on his phone, he doesn't respond. If I message him through Facebook, he responds.
Internet
08:13 S2: Um, uh, Christmas presents, online shopping. In fact, my son is doing, um, something called Photobox. I don't know if you have that in America. Um, it's, ah, an Internet site that you can have photos put on to things or made into a book rather than just you making an album that you make it into a proper hard bound book.
Phone
24:18 S2: But if I email him, even though I know he gets the message or even text him on his phone, he doesn't respond. If I message him through Facebook, he responds.
Google
08:06 S1: So, you tried Google Scholar for books. Umm, how did you get to the point where you knew which journals you could even go to, to ask for direct access to, to the stuff?
Google drive
02:10 S2: Um, and rather than keep saving it and emailing it to myself, as I have done in the past, um, Google Drive is like an iCloud. As soon as you log in to your account, it's there, and it continually saves what you've done. You don't have to save it. It just saves as you go along.
Internet
11:13 S2: Umm. I mean, we're doing a topic on rainforests at the moment, umm, and even though in the classroom I did rainforests about six years ago, I've forgotten lots of it.
[chuckle]
11:25 S1: Mm-hmm, yeah.
11:26 S2: So, I had to you know, umm, read up on things myself and from that I've just used books that we have in school rather than looking on the internet because the children are using those particular books. So, rather than finding out anymore, I just looked at the books.
Facebook
19:01 S2: They've even moved on from the school. Most of the people I have on Facebook, umm, have been colleagues, they are all previous colleagues.
Internet
I went to my grandmother's house, she didn't have any internet at all, so it didn't work. [chuckle]
Photography
06:19 S2: I just can't bear glue everywhere and everything being stuck to the wrong place. So if they just, you know, manipulate the pieces of paper, move things around and then take a photo of it.

Exciting Work

- Big data modeling: Homotopical patch theory, groupoids, torsors, etc.
- DSLs for data cleaning
- Information geometry
- Just starting the FP journey
- Let’s talk!

METAPHORISMS: THE MOTIVATION

Our Research Project

Use WorldCat Discovery to find more than two billion library resources

More ways to explore the resources of your library ... and the world's libraries

Our Research Project

- Behavioral (Clickstream) Analytics
  - What are users doing?
  - How can we assist?
- Terabytes of log data
- Reduced to ~100GB for our study
- **~5GB gzipped**

Succinct

Enabling Queries on Compressed Data

Rachit Agarwal, PhD

Anurag Khandelwal

Ion Stoica, PhD

OCLC

Succinct

- A compression format for unstructured text
- Query unstructured, compressed text files directly
  - No data scans
  - No secondary indexes
  - No data decompression
- http://succinct.cs.berkeley.edu/

Succinct Operations

- s.substring(offset: Nat, length: Nat): String
- s.count(str: String): Nat
- s.indicesOf(str: String): [Nat]
- s.findAllInRange(str1: String, str2: String): [Nat]
- …etc…

Unfortunately…

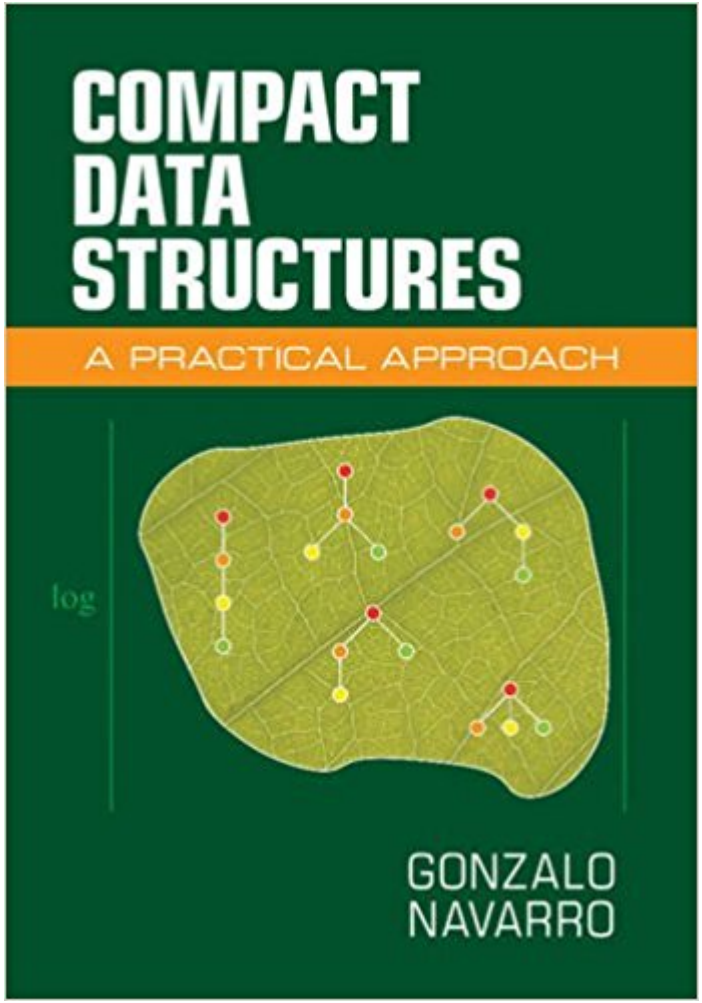
- Not implemented for our language
- Can’t call OSS version through FFI
- OSS version maxes out at 4GB (Int32)

Reinventing the Wheel

- Implement *de novo*
- Port to language X
- Copy/tweak

Build You A Succinct

- Two ingredients:
  - Compressed permutation data structure
  - Suffix array construction algorithm



On Compressing Permutations and Adaptive Sorting *

Jérémy Barbay Gonzalo Navarro

Dept. of Computer Science, University of Chile

OCLC

Suffix Array Construction

- DC3 / Skew algorithm
  - Clever algorithm!
  - Divide and conquer

There Should Be A Better Way

- Program Synthesis
  - Write a spec for program's desired behavior
  - (Semi-)mechanically synthesize a program satisfying that spec

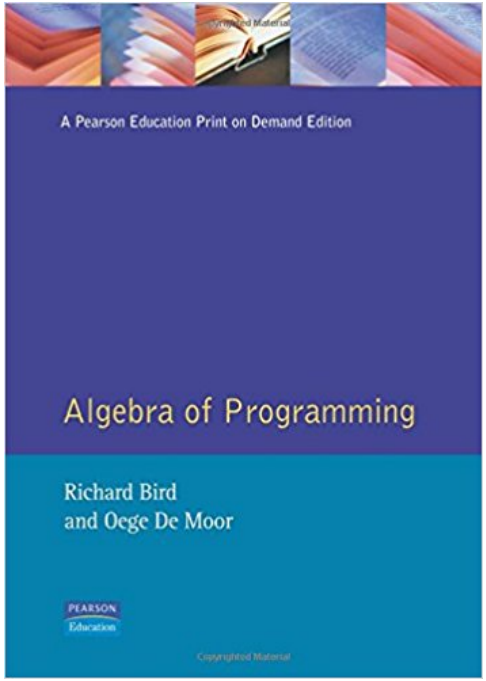
Bird-Meertens Formalism

- A categorical calculus of relations
- Category theory makes it language-agnostic
- Relations are more flexible than functions
- Calculus makes it formulaic
- Prove things!

Bird-Meertens Formalism

- Input: A **relational** spec of your algorithm
- Middle: Step-by-step derivation via algebraic rules
  - Exposes the degrees of freedom of implementation
- Output: A **functional** program

Bird-Meertens Formalism



- Dynamic programming
- Greedy algorithms
- Exhaustive search
- Divide and conquer
- And more!

Use of the Bird Meertens Formalism

Ankur Taly
(03005017)

Aditya Parameswaran
(03005015)

Ankit Jain
(03005021)

Abstract

In this report, we deal with the paradigm of Constructive Algorithmics or the science of program transformation. We examine the basic ideas of Bird Meertens Formalism and its application to segment problems. We first give the direct application of Bird Meertens Formalism to the Maximum Segment Sum Problem, and also indicate the underlying concepts involved. We then proceed to give an intuitive proof for the Sliding Tails theorem, and demonstrate how it can be applied to a problem.

Metaphorisms

- BMF-style relational program specifications…

  …defined over inductive types…

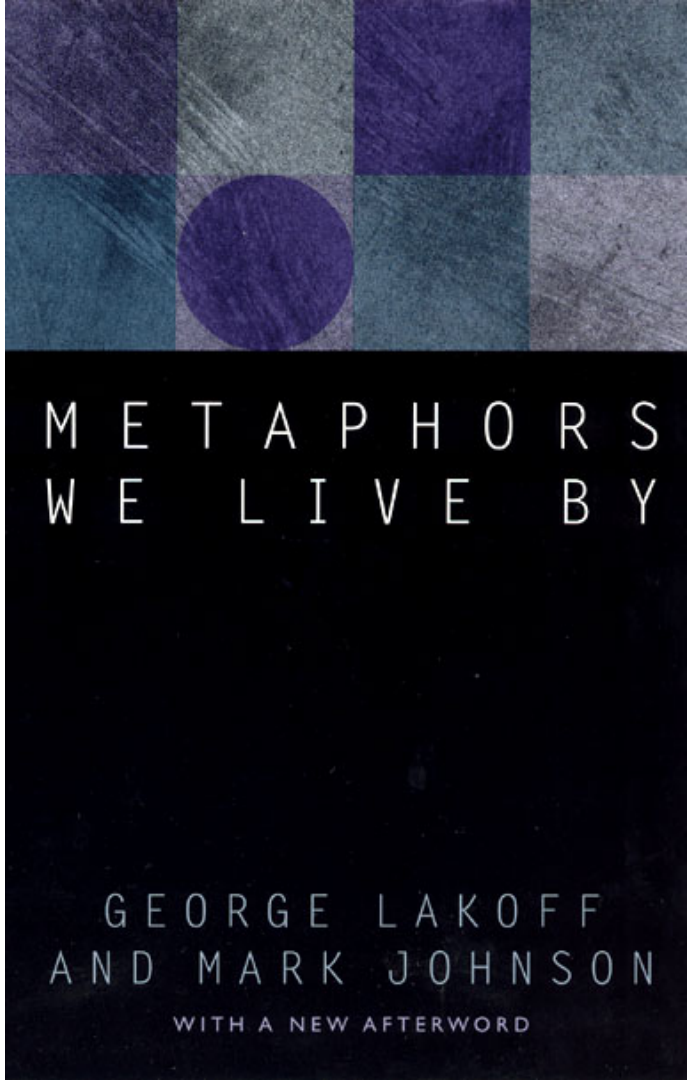
  …that mimic formal metaphors!

Metaphors

- “Time is money.”
- “All the world’s a stage.”
- “A monad is a burrito.”
- “Monads are trees with grafting.”
-  “Some say love…it is a razor…”

Metaphors in Software

- Tree
- Pipe
- Stack
- Heap
- Queue
- Spaghetti code

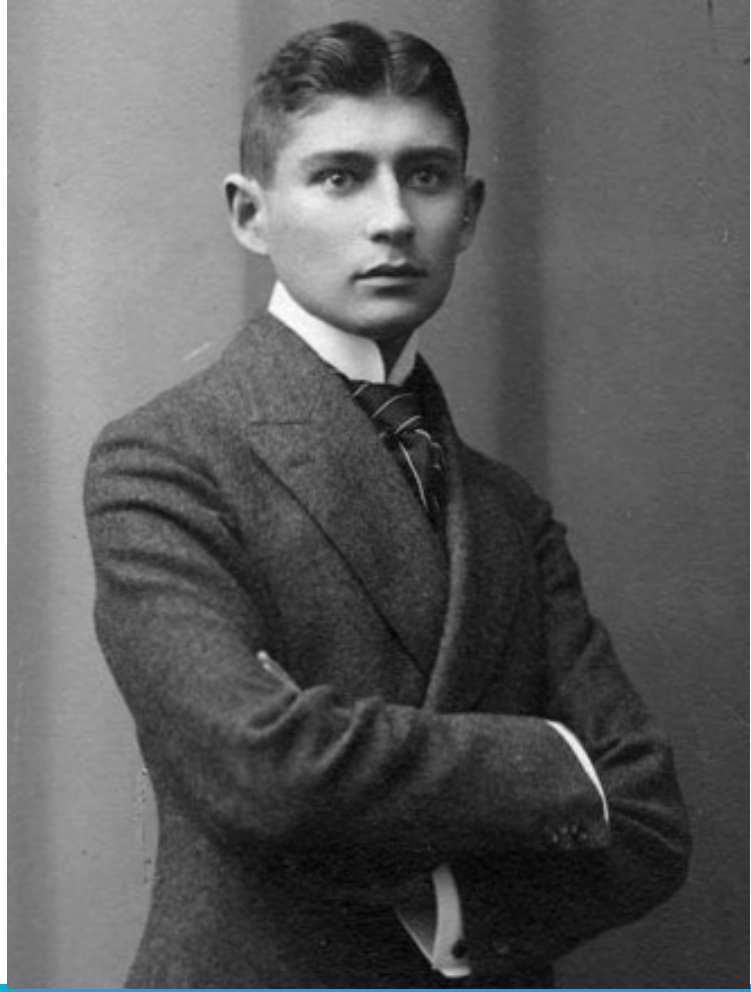

METAPHORS
WE LIVE BY
GEORGE LAKOFF
AND MARK JOHNSON
WITH A NEW AFTERWORD

Formal Metaphors

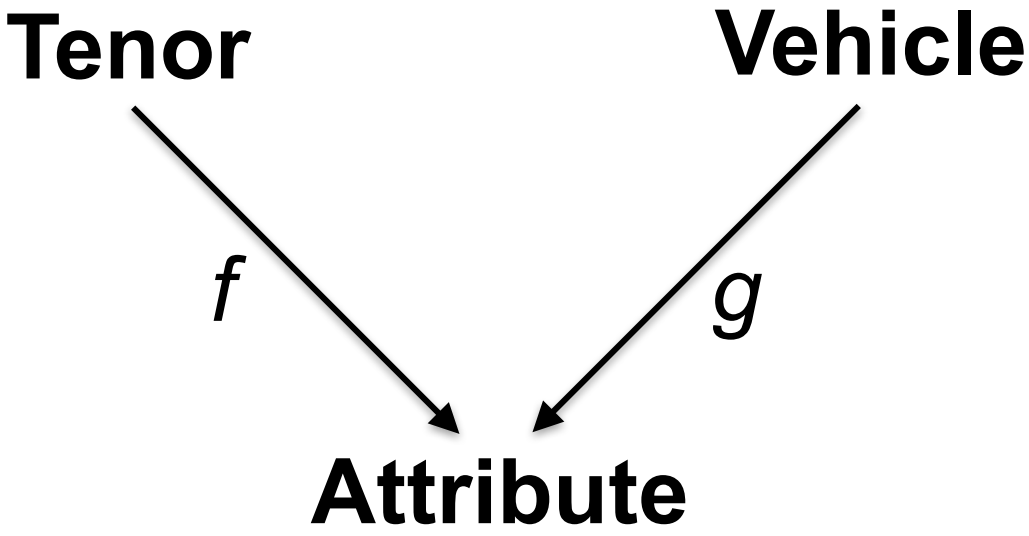
- Three components:
  1. A **tenor** (“Love”)
  2. A **vehicle** (“A Razor”)
  3. A **shared attribute** (“Ouch!”)

“Love is, that you are the knife which I plunge into myself.”

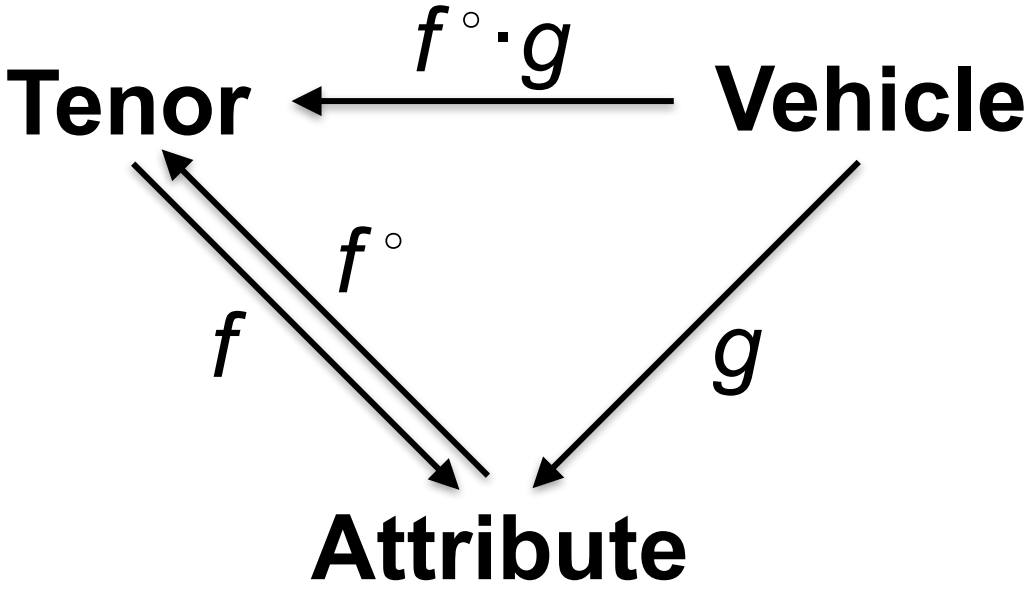
-Franz “Streaming” Kafka



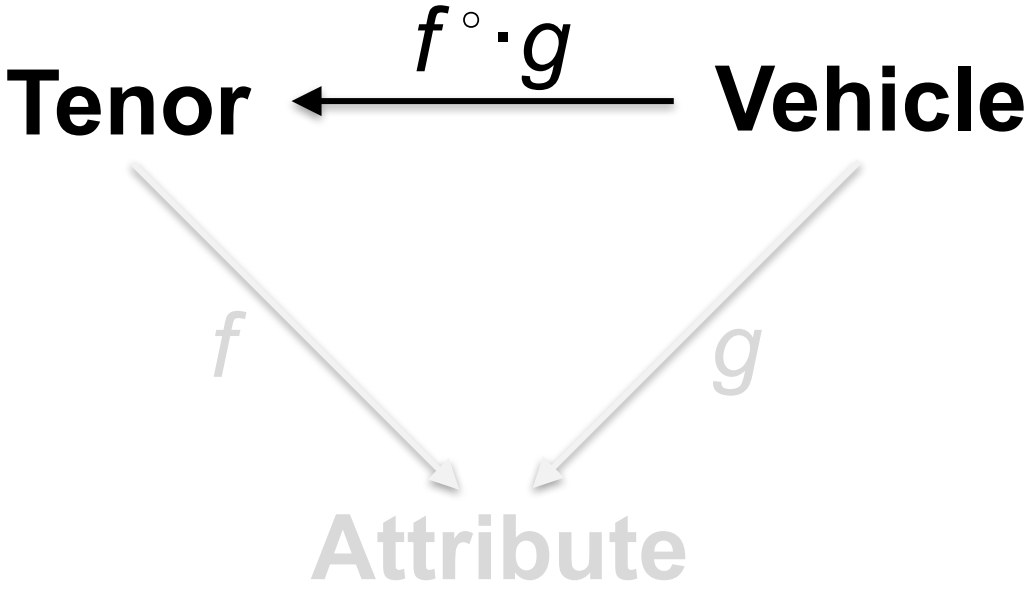
Metaphors, Categorically

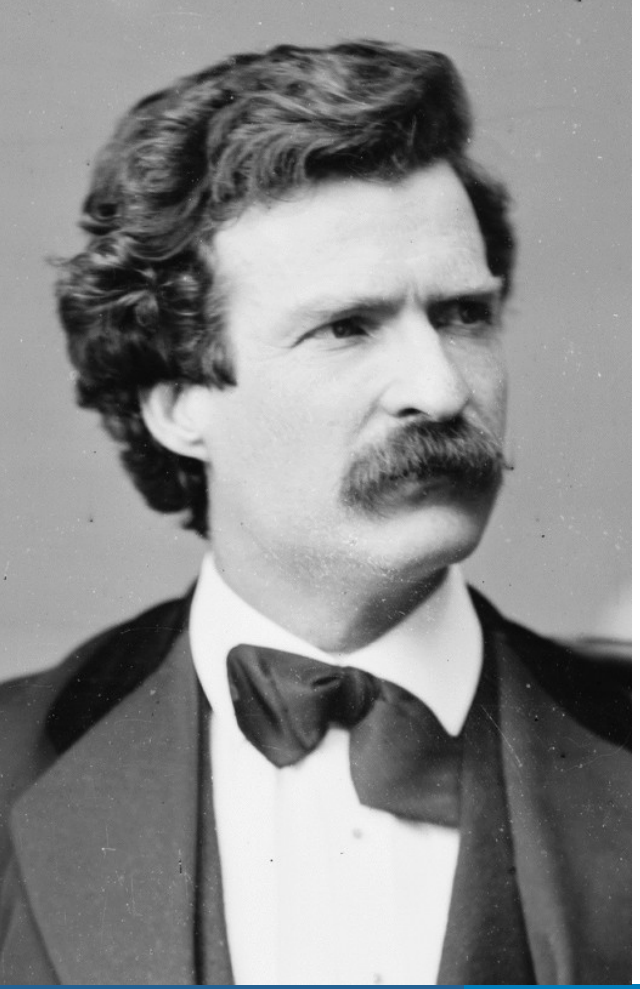


Metaphors, Categorically



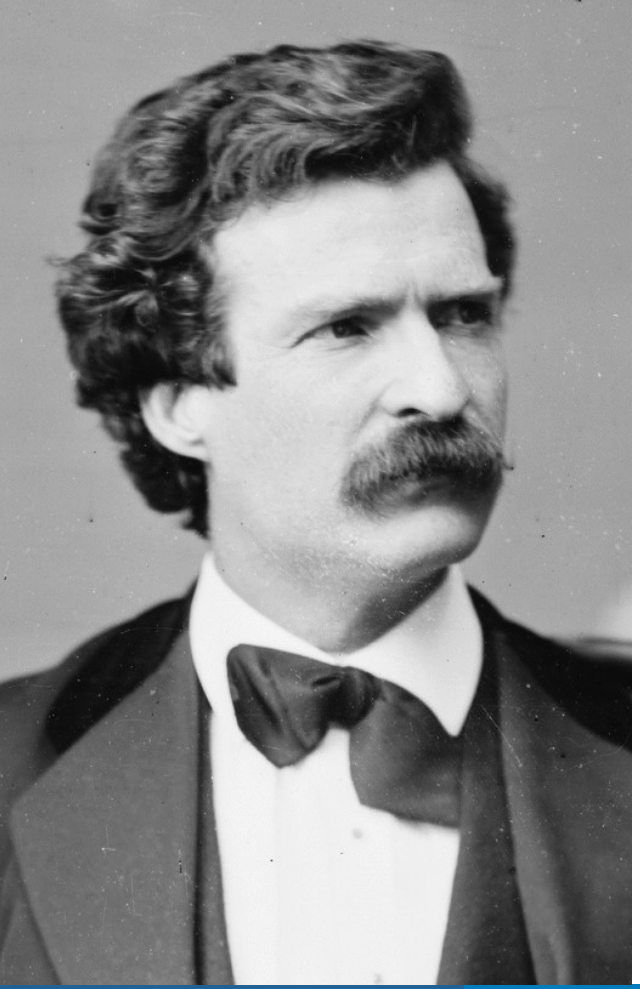
Metaphors, Categorically





“Politicians and diapers must be changed often, and for the same reason.”

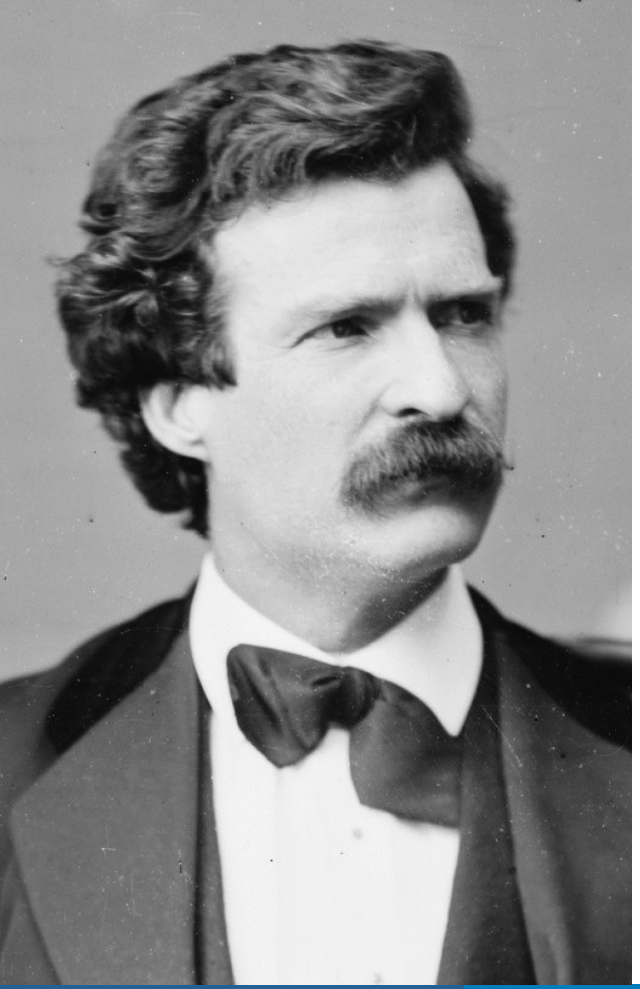
-Mark Twain



Tenor

‘Politicians and diapers must be changed often, and for the same reason.”

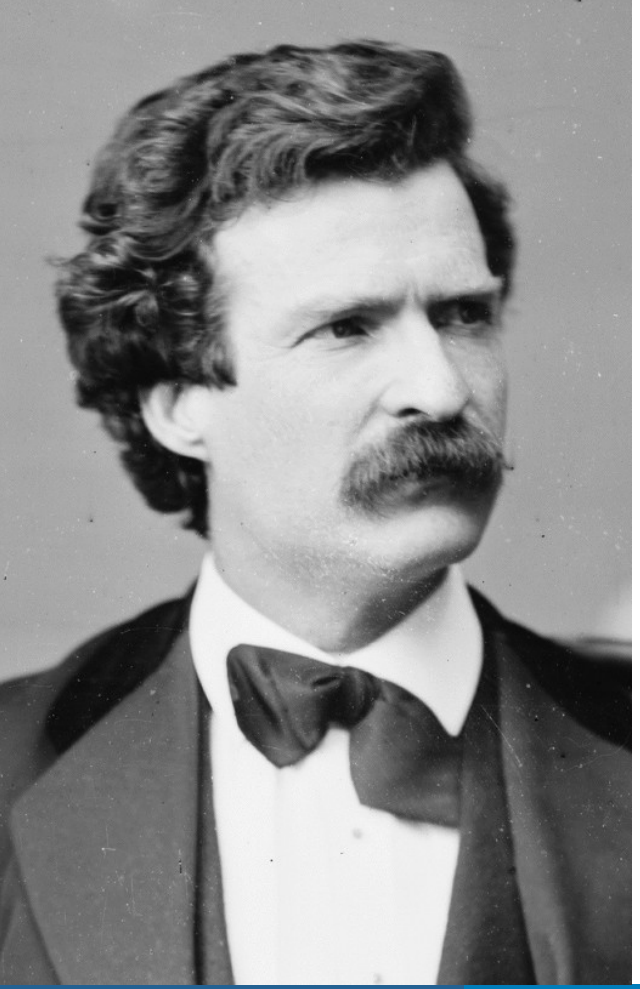
-Mark Twain



Vehicle

"Politicians and diapers must be changed often, and for the same reason."

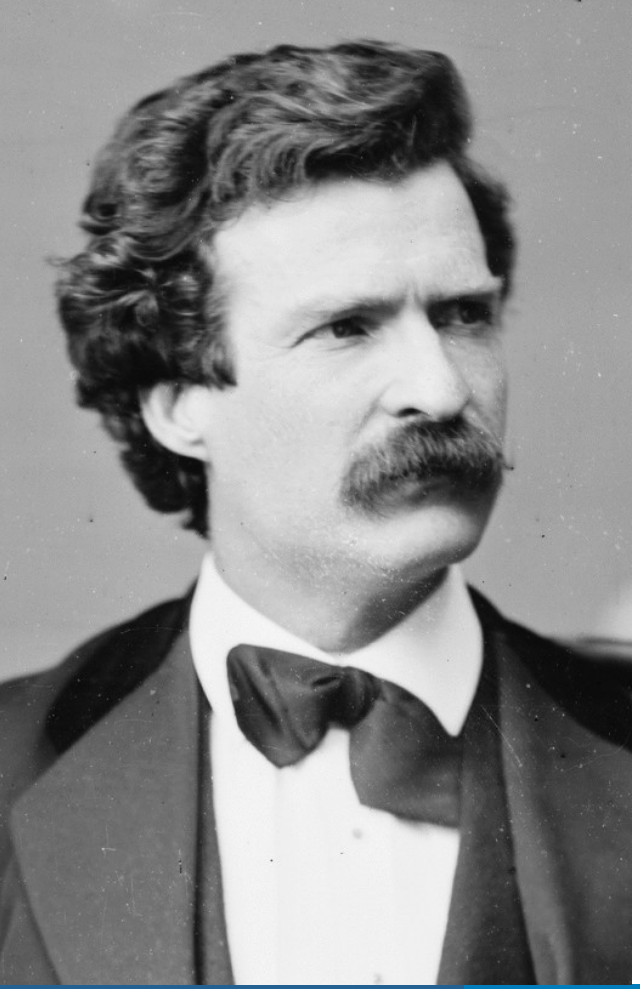
-Mark Twain



“Politicians and diapers must be changed often, and for the same reason.”

Shared attribute hint

-Mark Twain

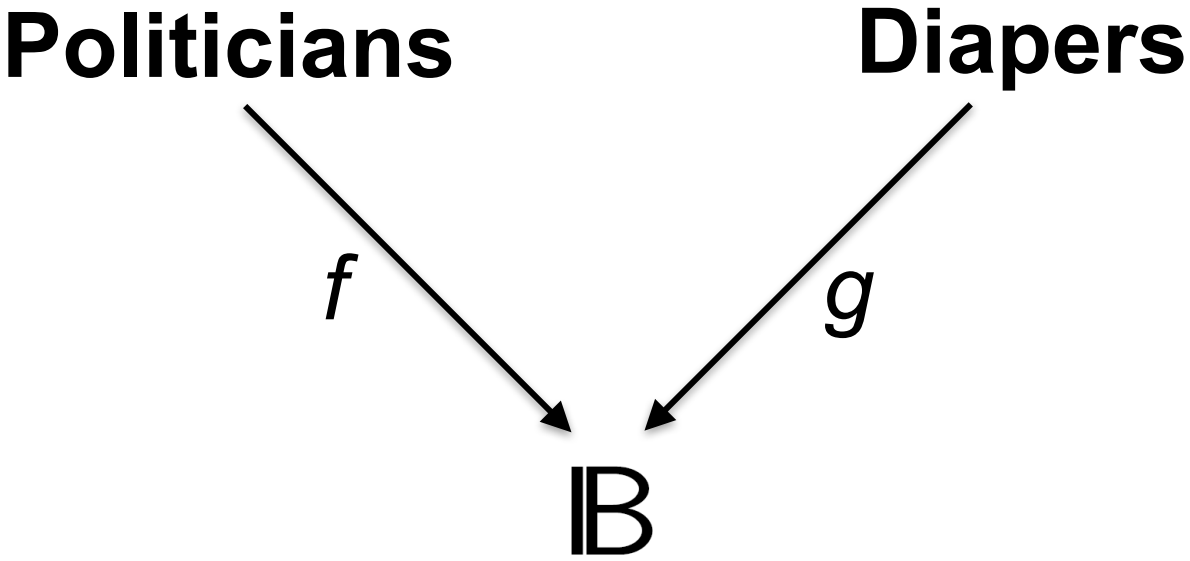


“Politicians and diapers must be changed often, and for the same reason.”

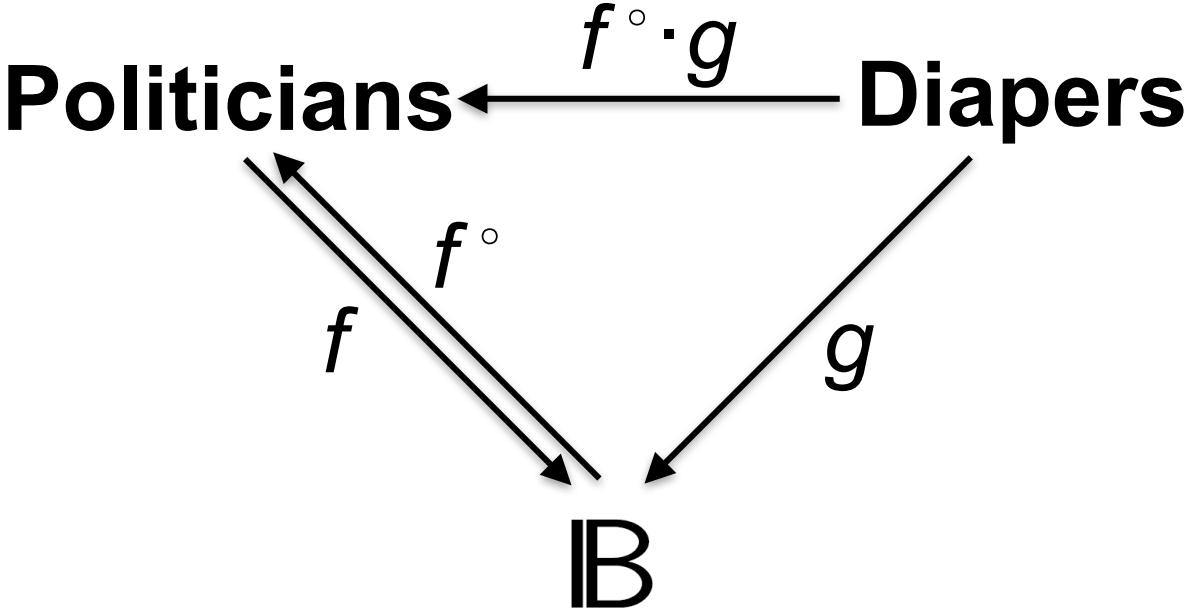
Shared attribute hint

-Mark Twain

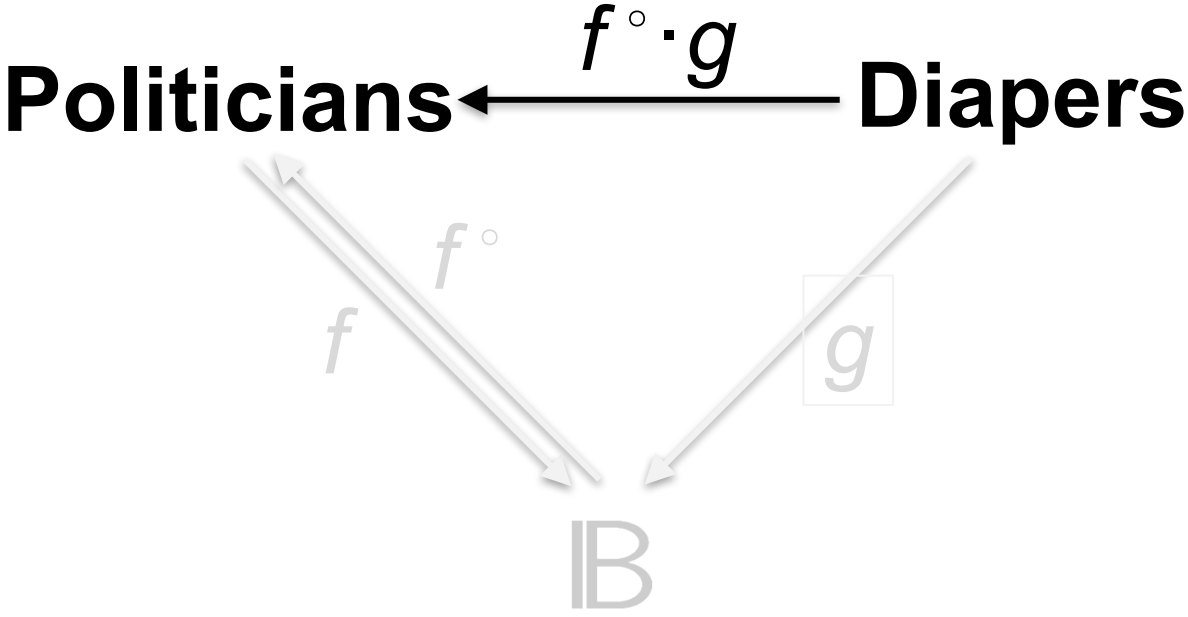
Metaphors, Categorically



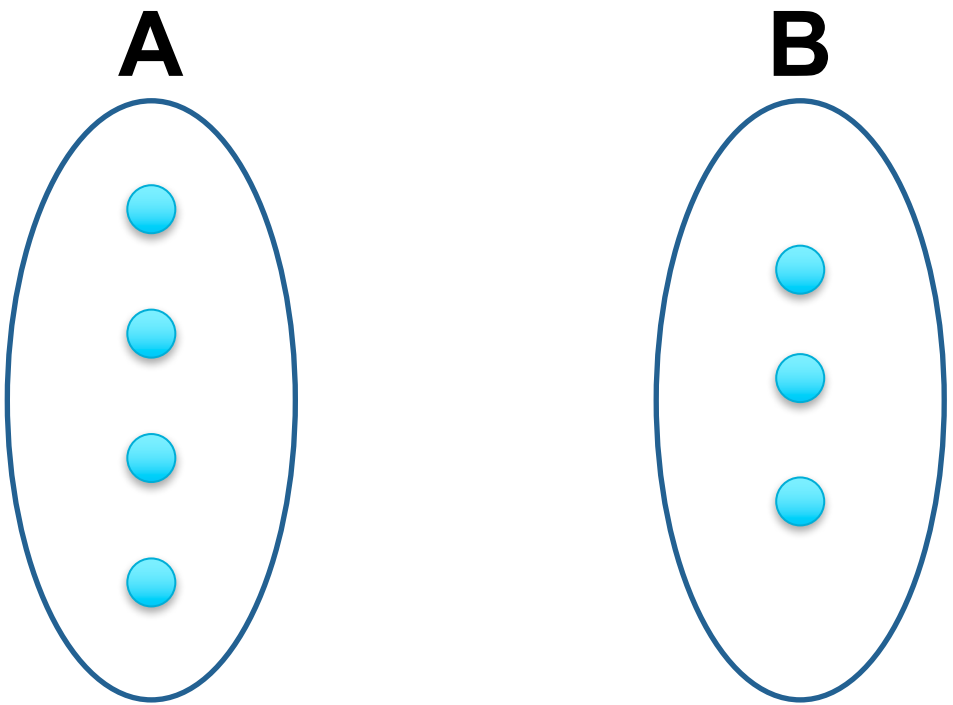
Metaphors, Categorically



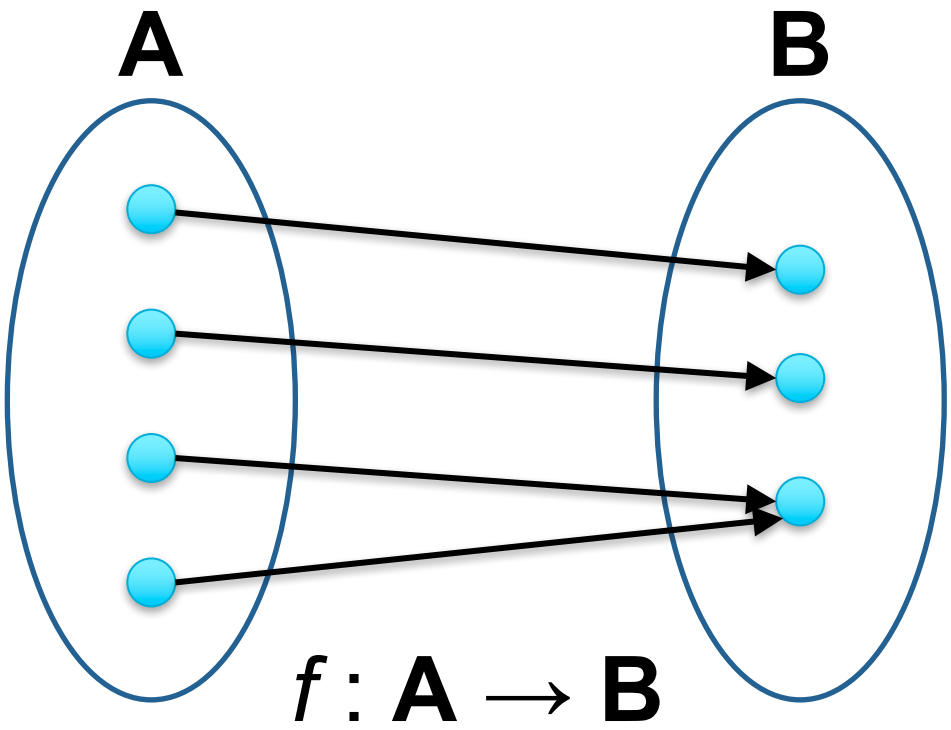
Metaphors, Categorically



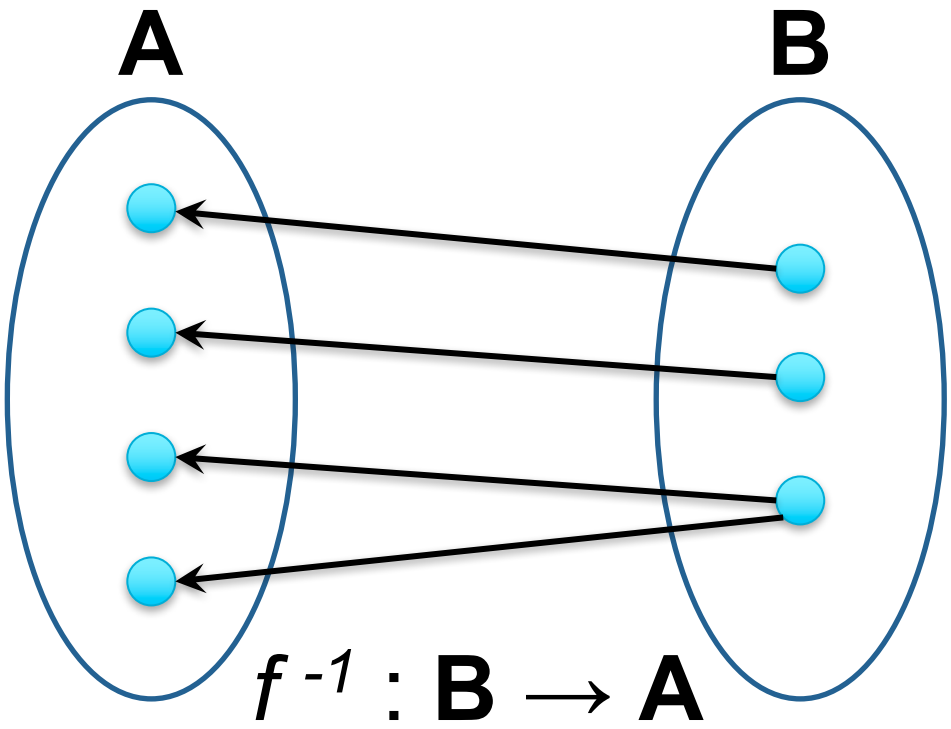
Interlude: Function Converse



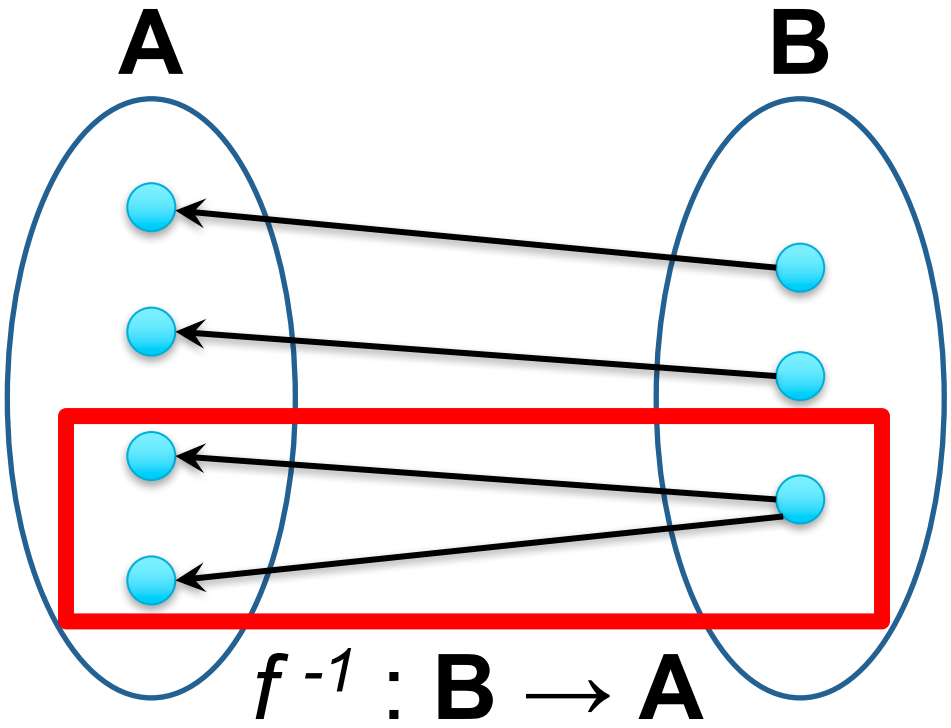
Interlude: Function Converse



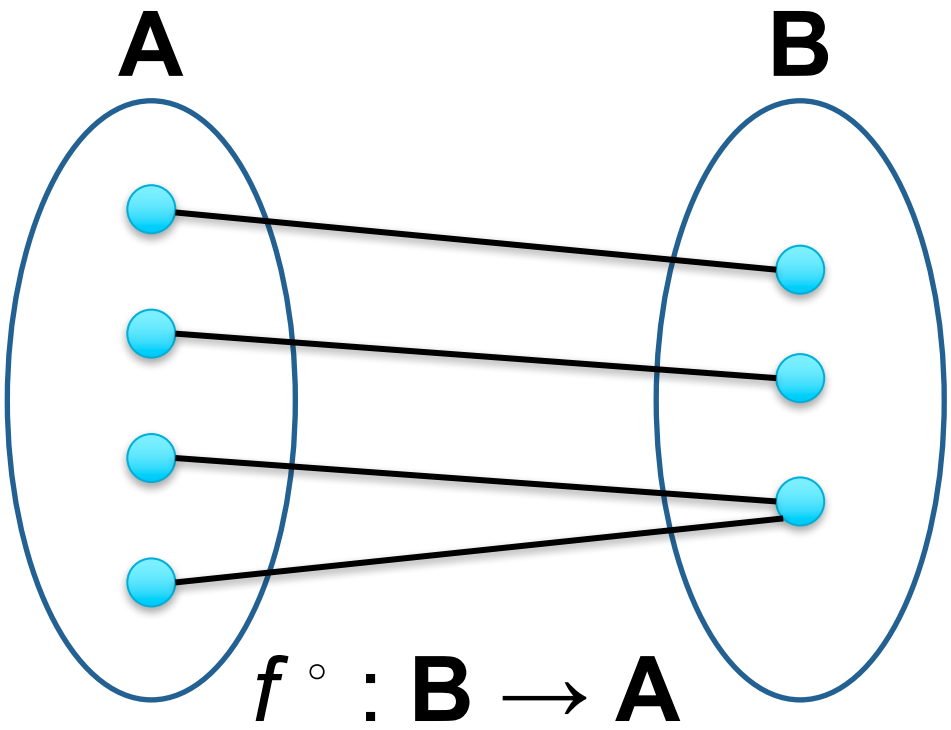
Interlude: Function Converse



Interlude: Function Converse



Interlude: Function Converse



SQL “sequel”

abbreviation for

(Computer Science) structured query language: a computer programming language used for database management

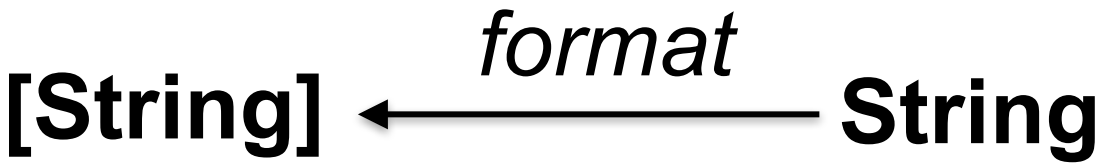
“CITE” Collins English Dictionary – Complete and Unabridged, 12th Edition 2014 © HarperCollins Publishers 1991, 1994, 1998, 2000, 2003, 2006, 2007, 2009, 2011, 2014

sequela (sē-kwē'lă) *plural.***sequelae** [L., sequel]

A condition following and resulting from a disease.

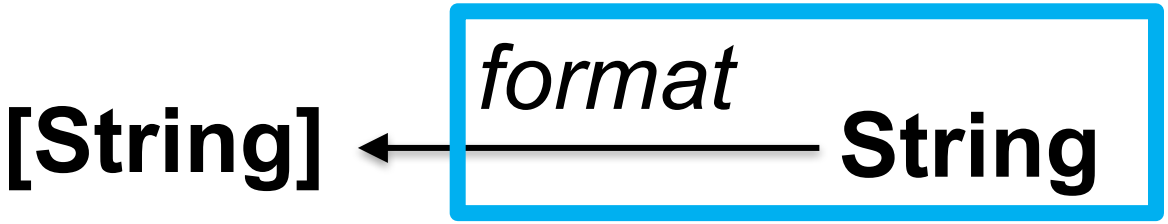
“CITE” Medical Dictionary, © 2009 Farlex and Partners

Example: Text Formatting

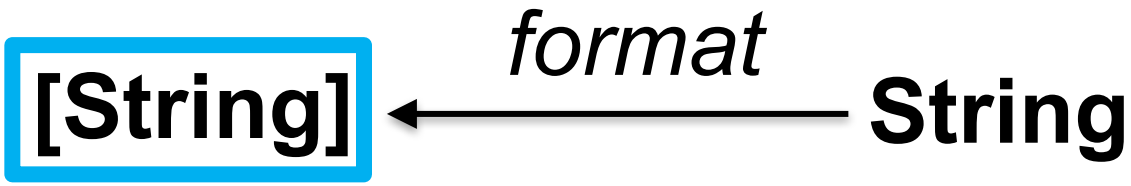


(e.g., word wrap, scalariform, gofmt, etc.)

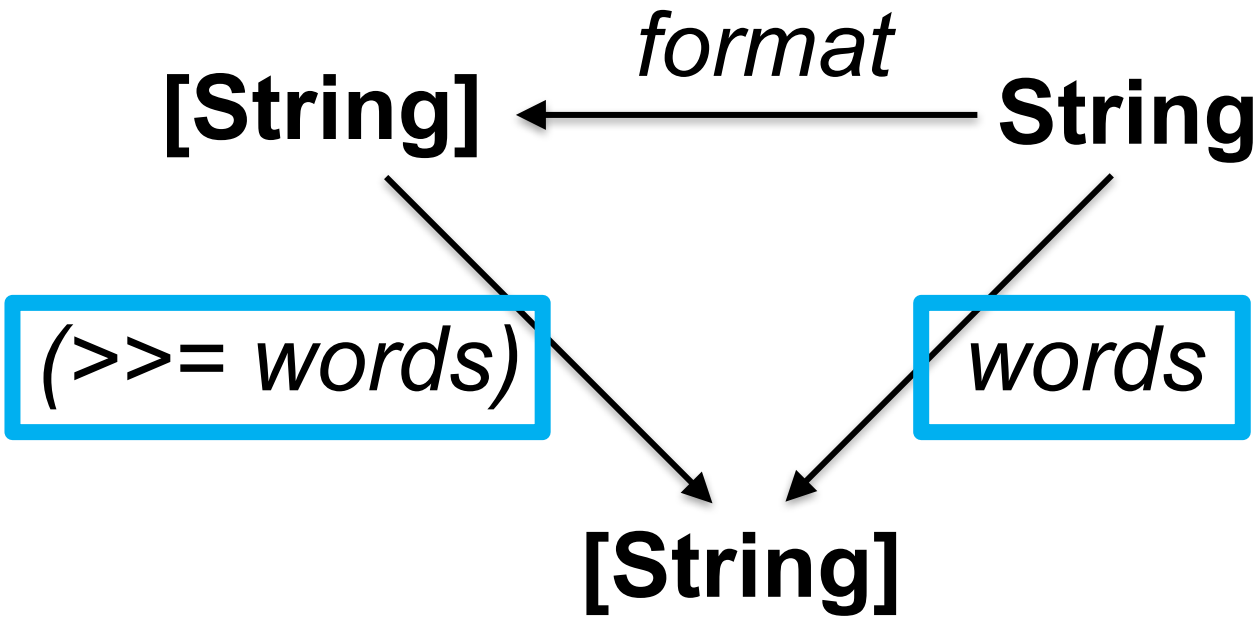
Example: Text Formatting



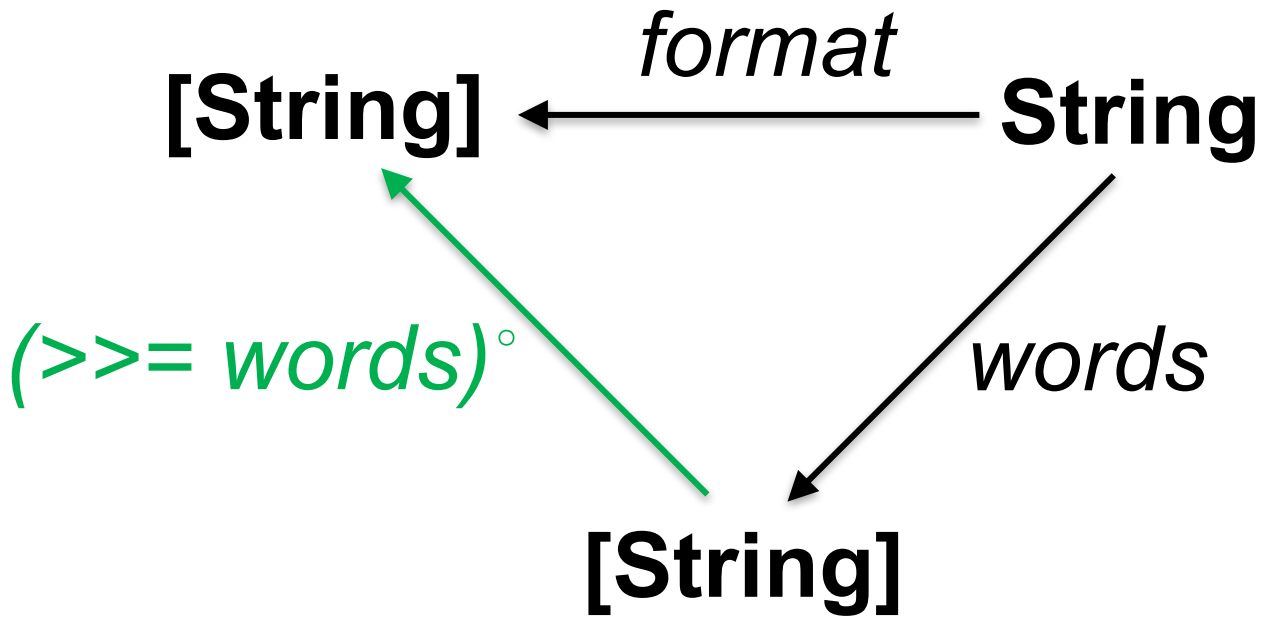
Example: Text Formatting



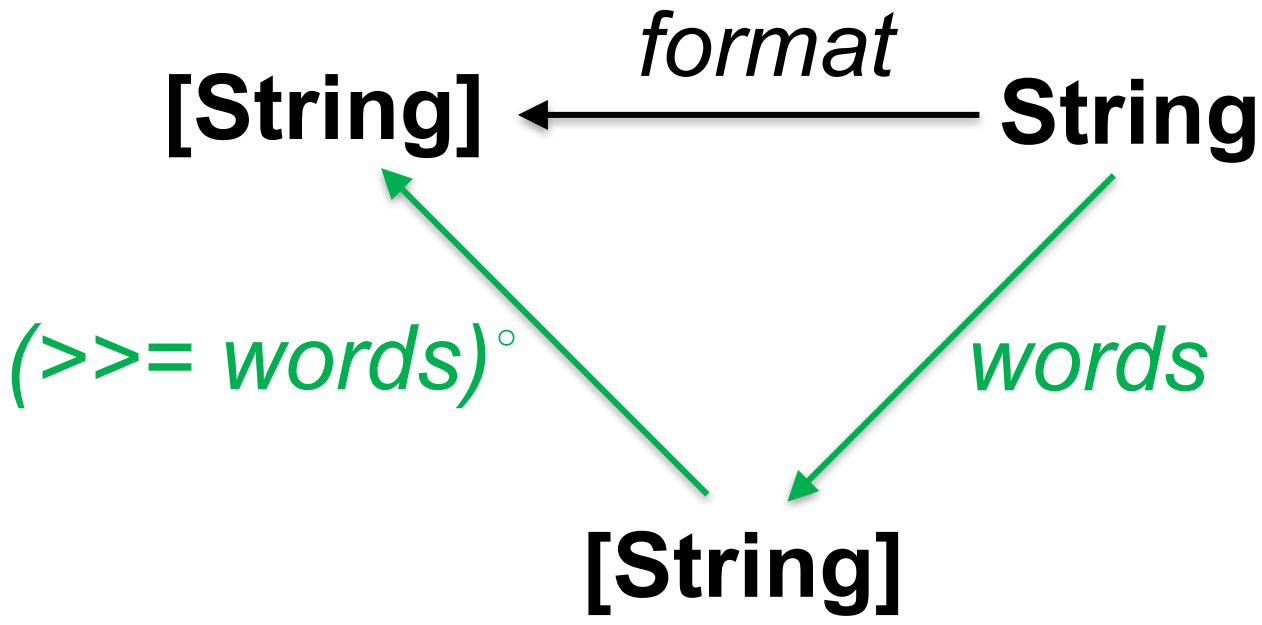
Example: Text Formatting



Example: Text Formatting



Example: Text Formatting



Example: Text Formatting

[“foo”, “bar”, “baz”] 

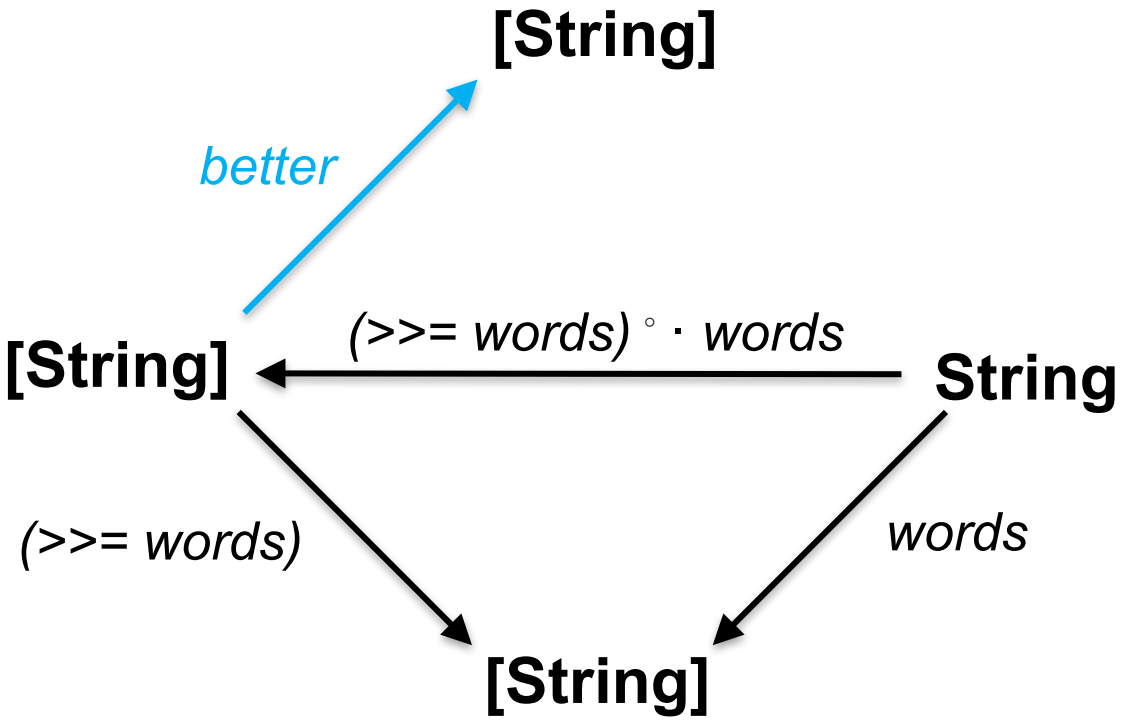
“foo bar baz”

Example: Text Formatting

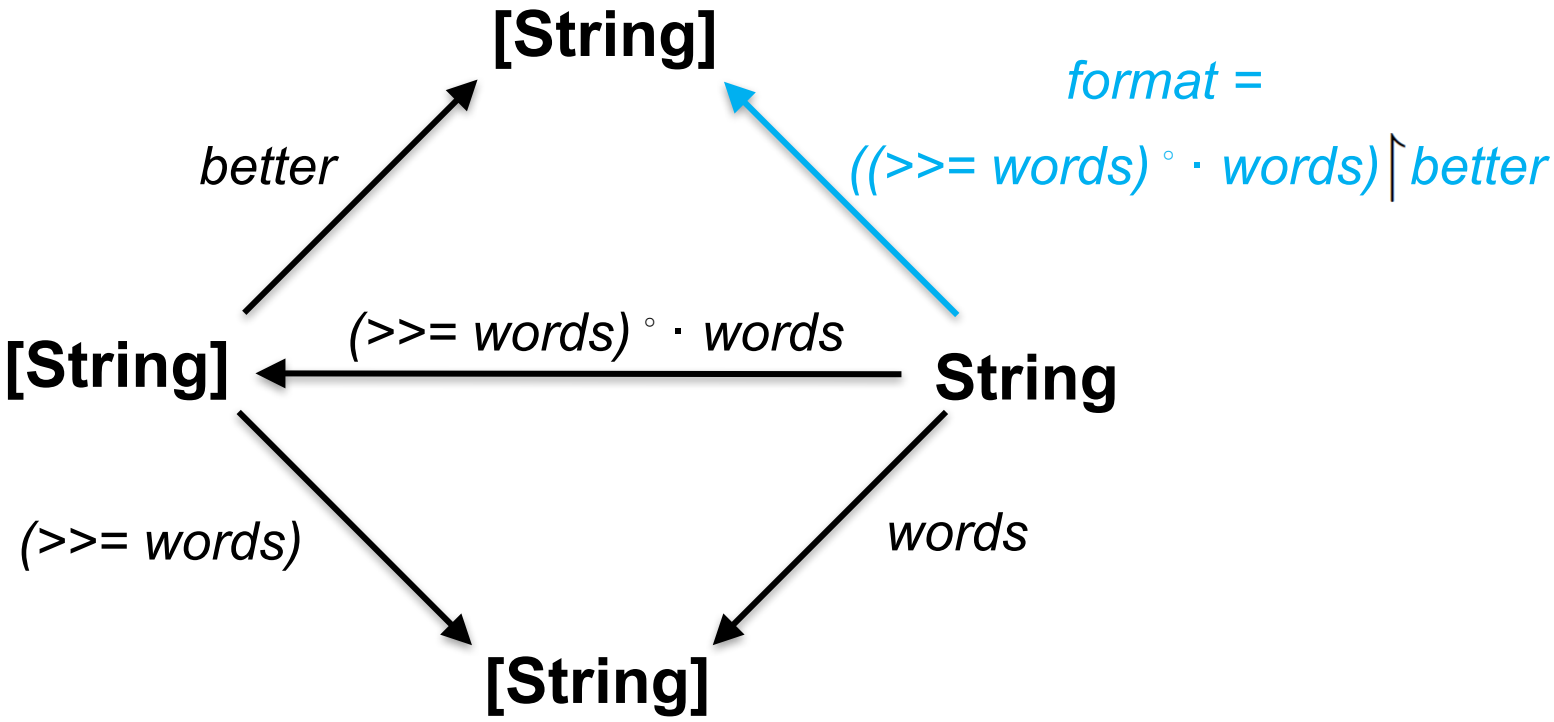
[“foo”, “bar”, “baz”],
[“foo bar”, “baz”],
[“foo”, “bar baz”],
[“ foo bar baz”]
[“ foo”, “ bar”, “ baz”]
…
…etc.

“foo bar baz”

Example: Text Formatting



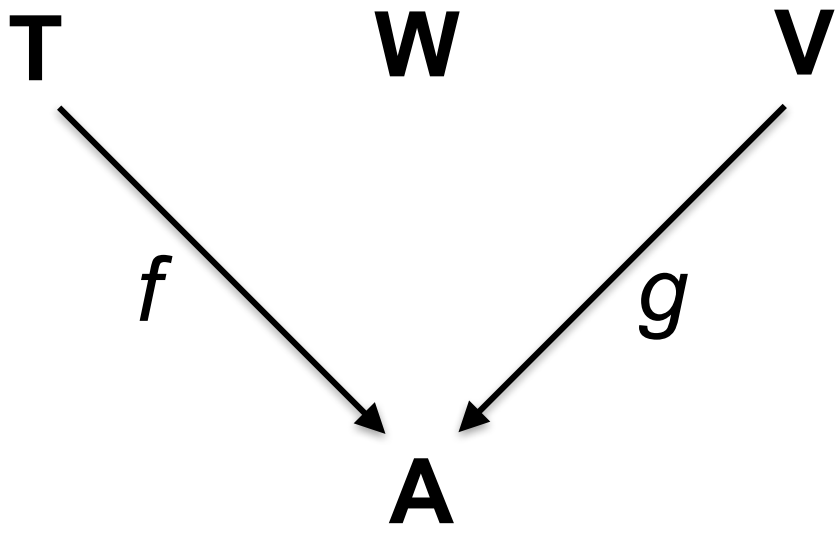
Example: Text Formatting



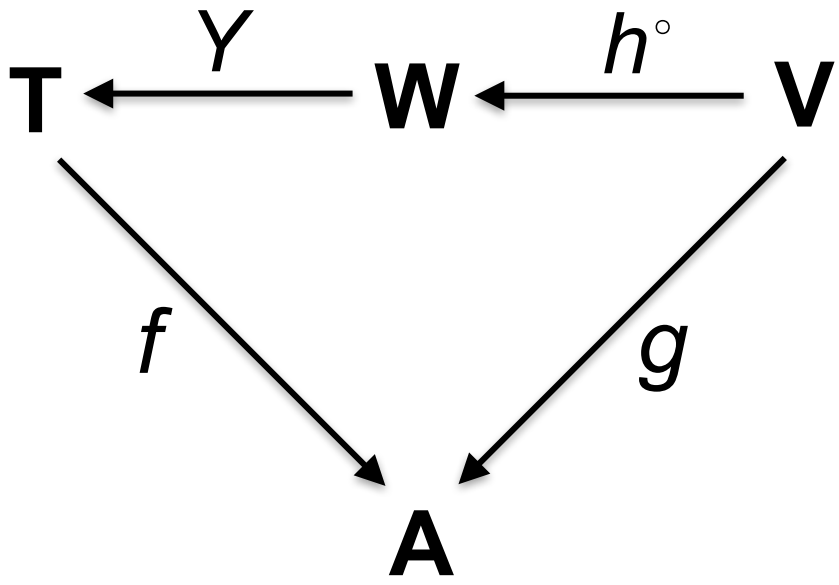
Divide and Conquer Algorithms

- Objects: Inductive types
- Arrows: Ana- and catamorphisms
- How?
  1. Introduce an intermediate inductive type
  2. Plug in a hylomorphism

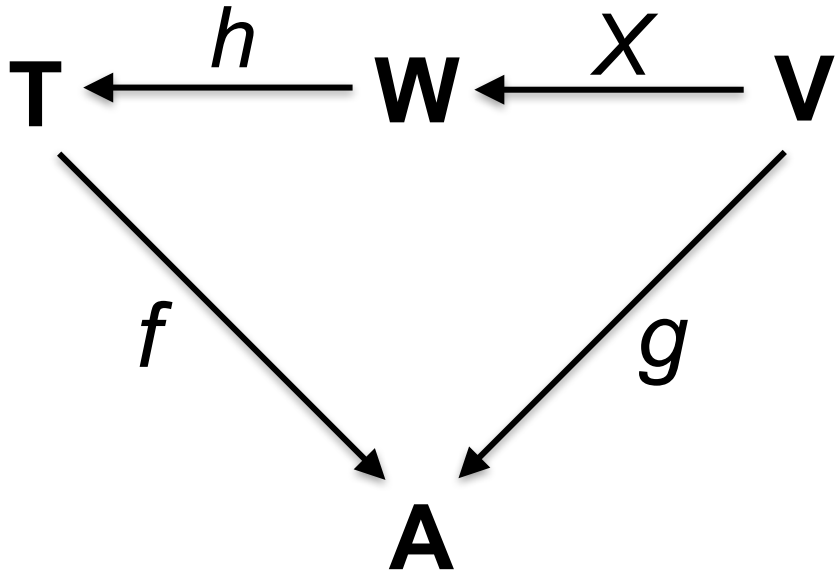
Divide and Conquer Algorithms



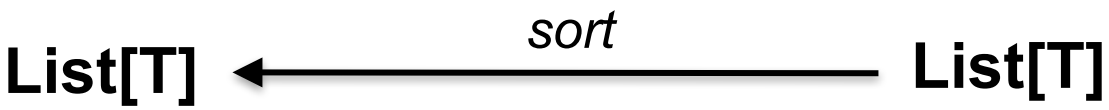
“Easy Divide, Hard Conquer”



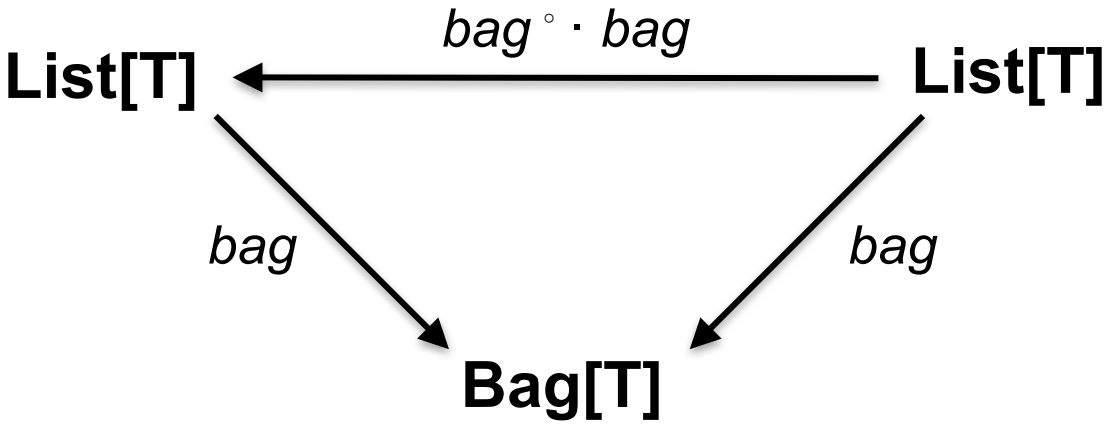
“Hard Divide, Easy Conquer”



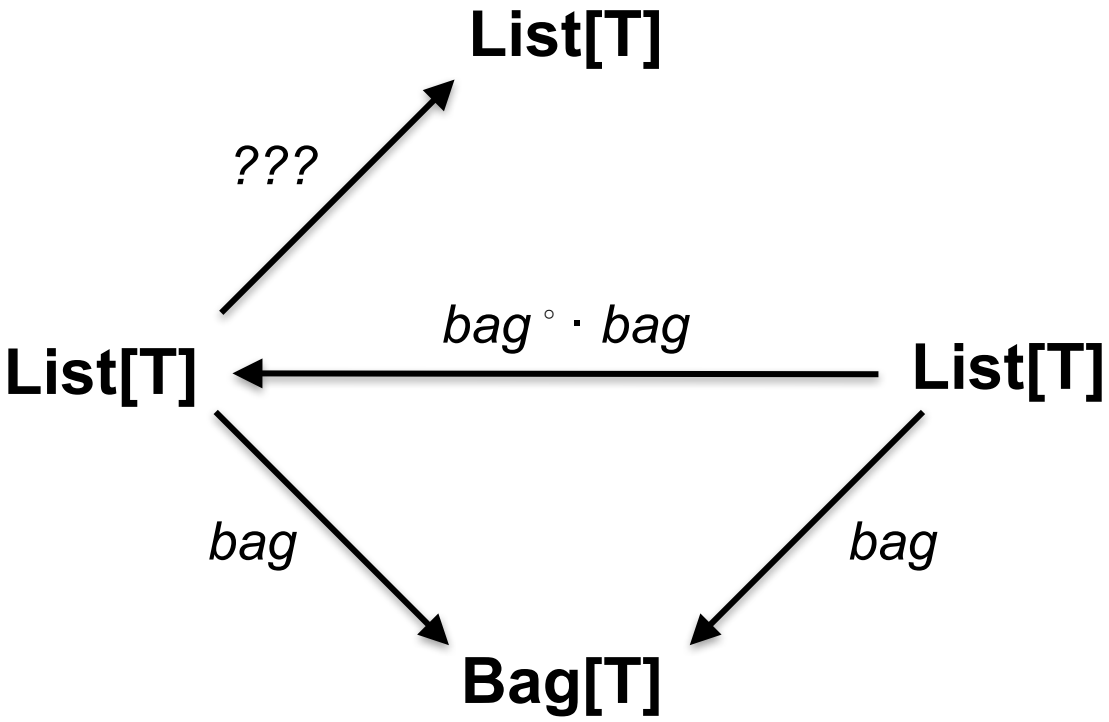
Example: Sorting a List



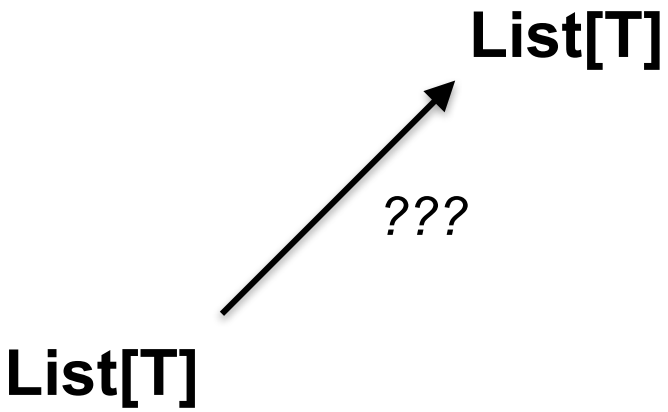
Example: Sorting a List



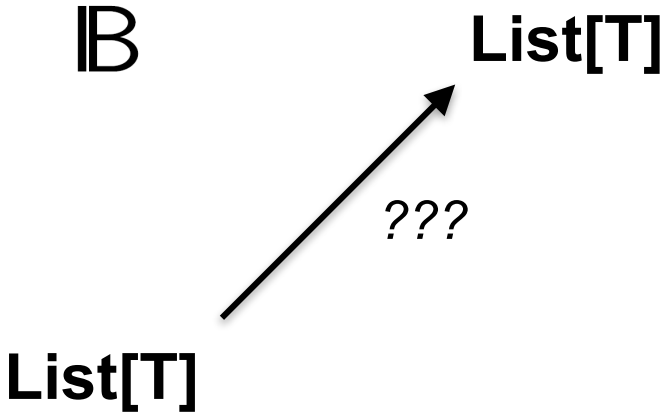
Example: Sorting a List



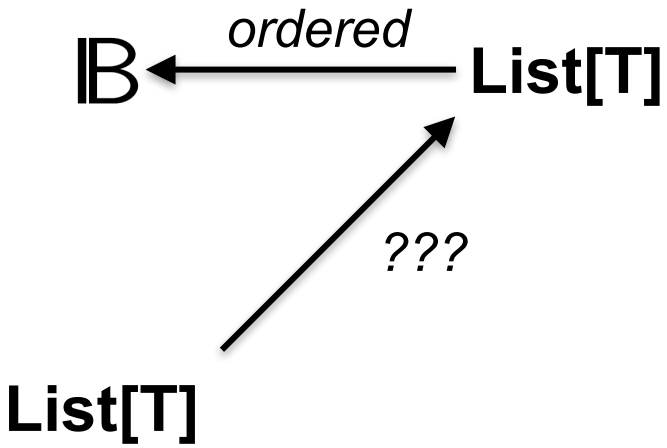
Example: Sorting a List



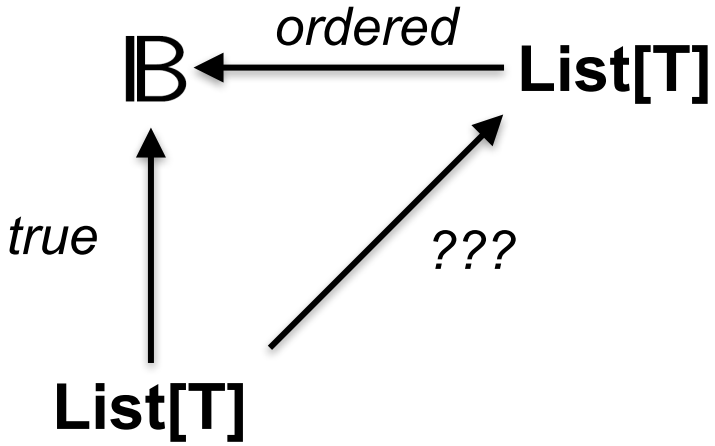
Example: Sorting a List



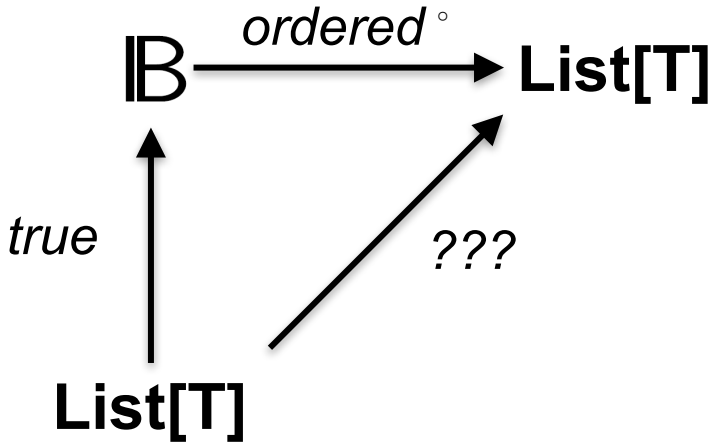
Example: Sorting a List



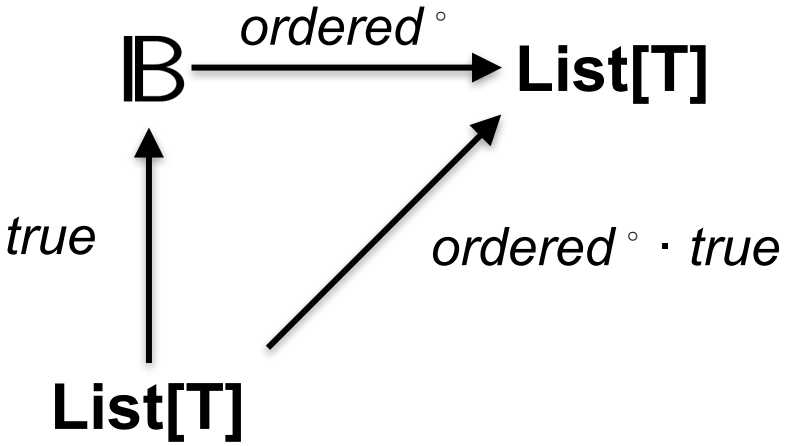
Example: Sorting a List



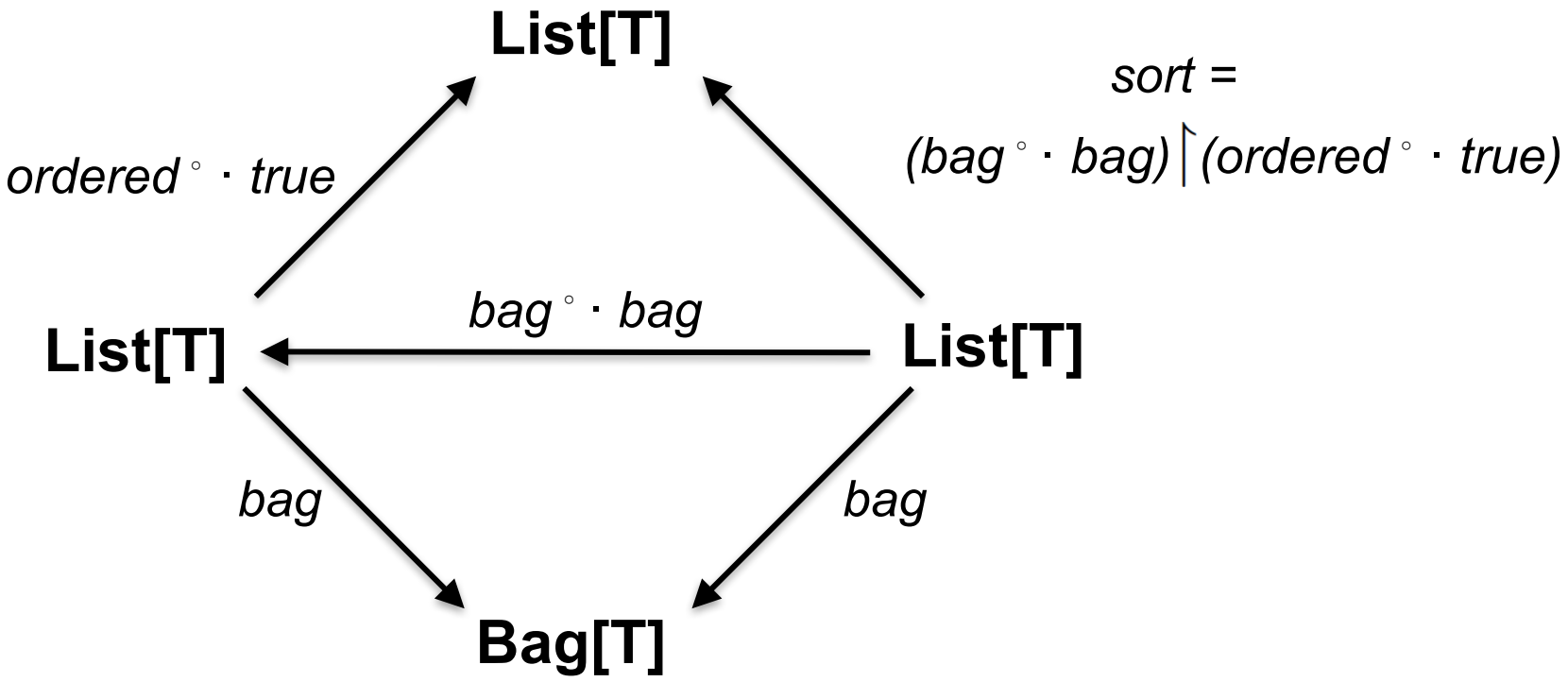
Example: Sorting a List



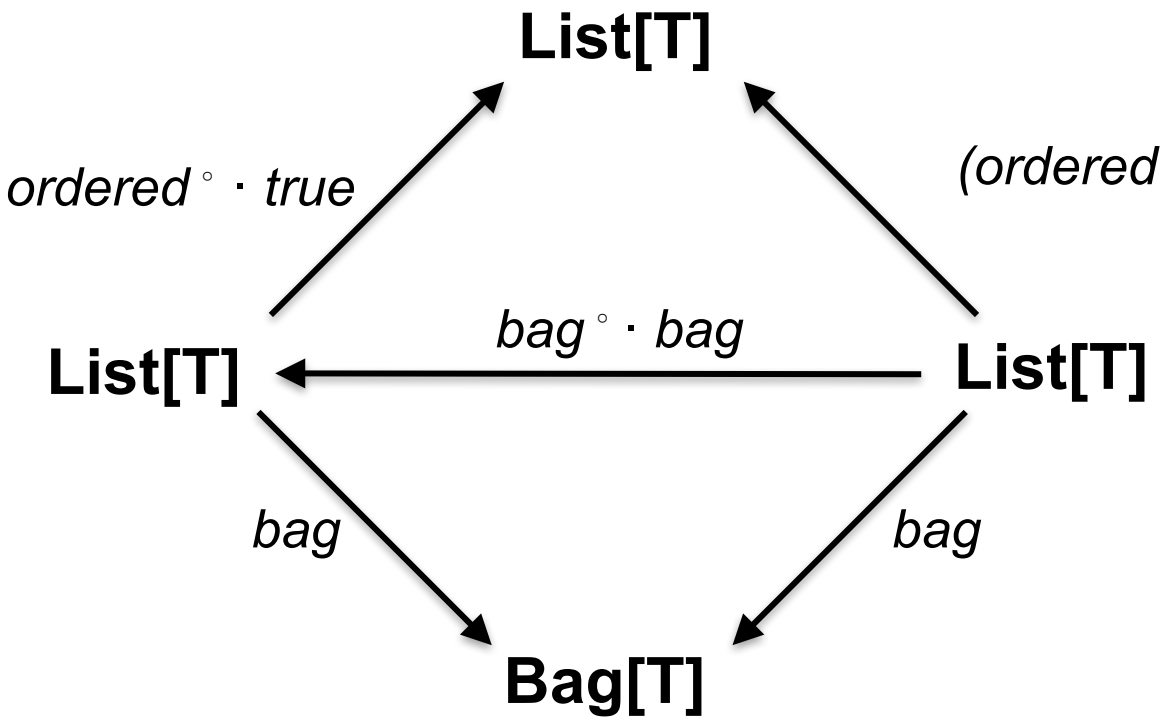
Example: Sorting a List



Example: Sorting a List



Example: Sorting a List



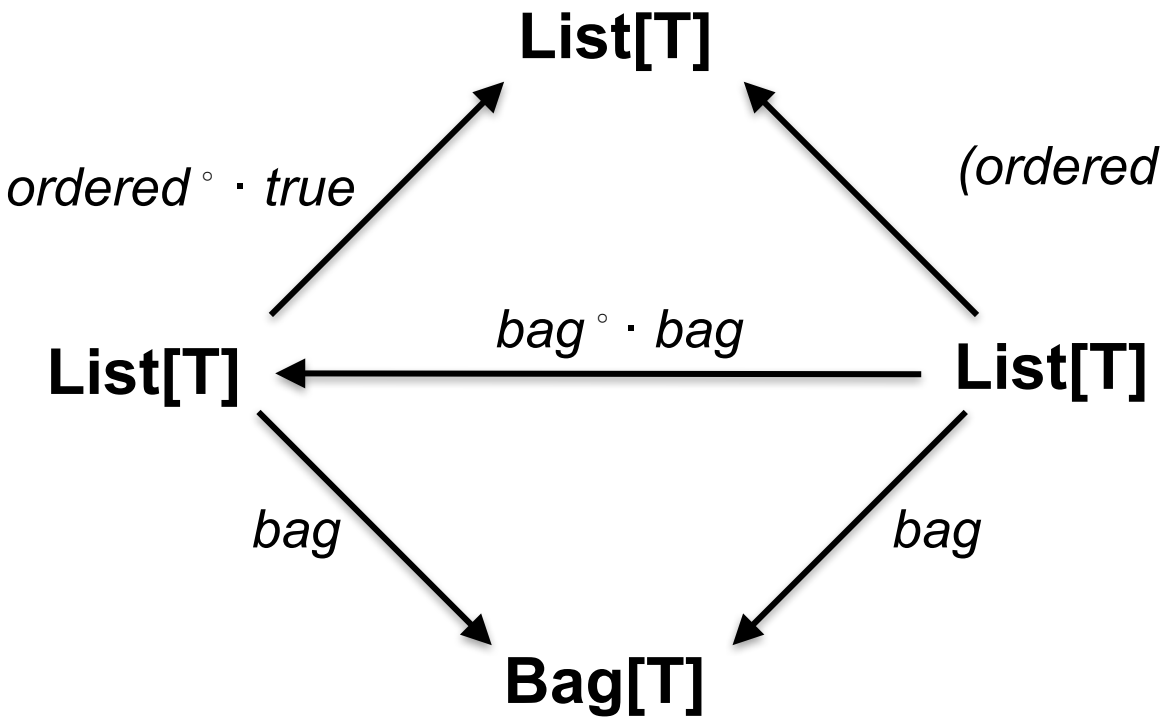
$$sort = (ordered \circ \cdot true) \cdot (bag \circ \cdot bag)$$

PROGRAM SYNTHESIS

Example: “Mystery Sort"

- Start with a “Divide and Conquer” template
- Plug in bits/pieces of our spec for *sort*
- Turn the crank!

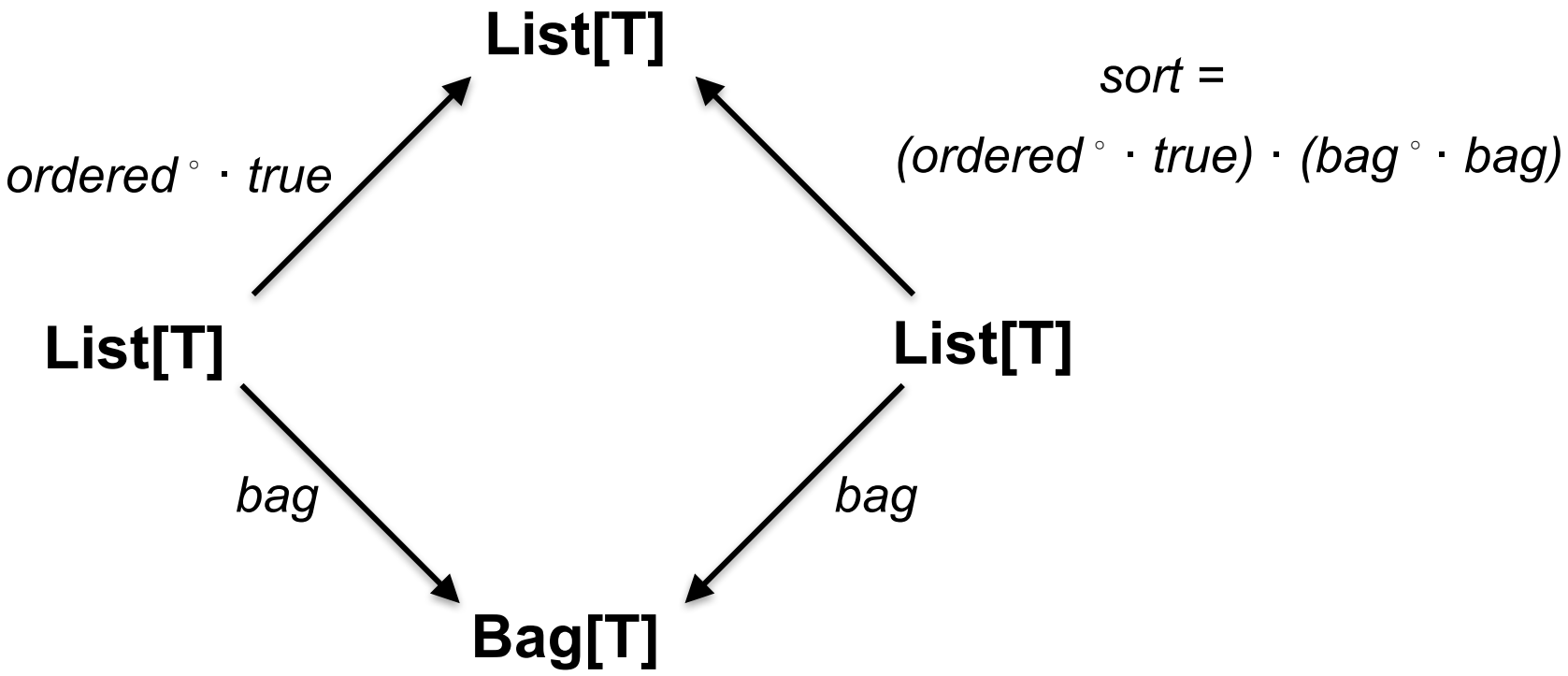
Example: “Mystery Sort”



sort =

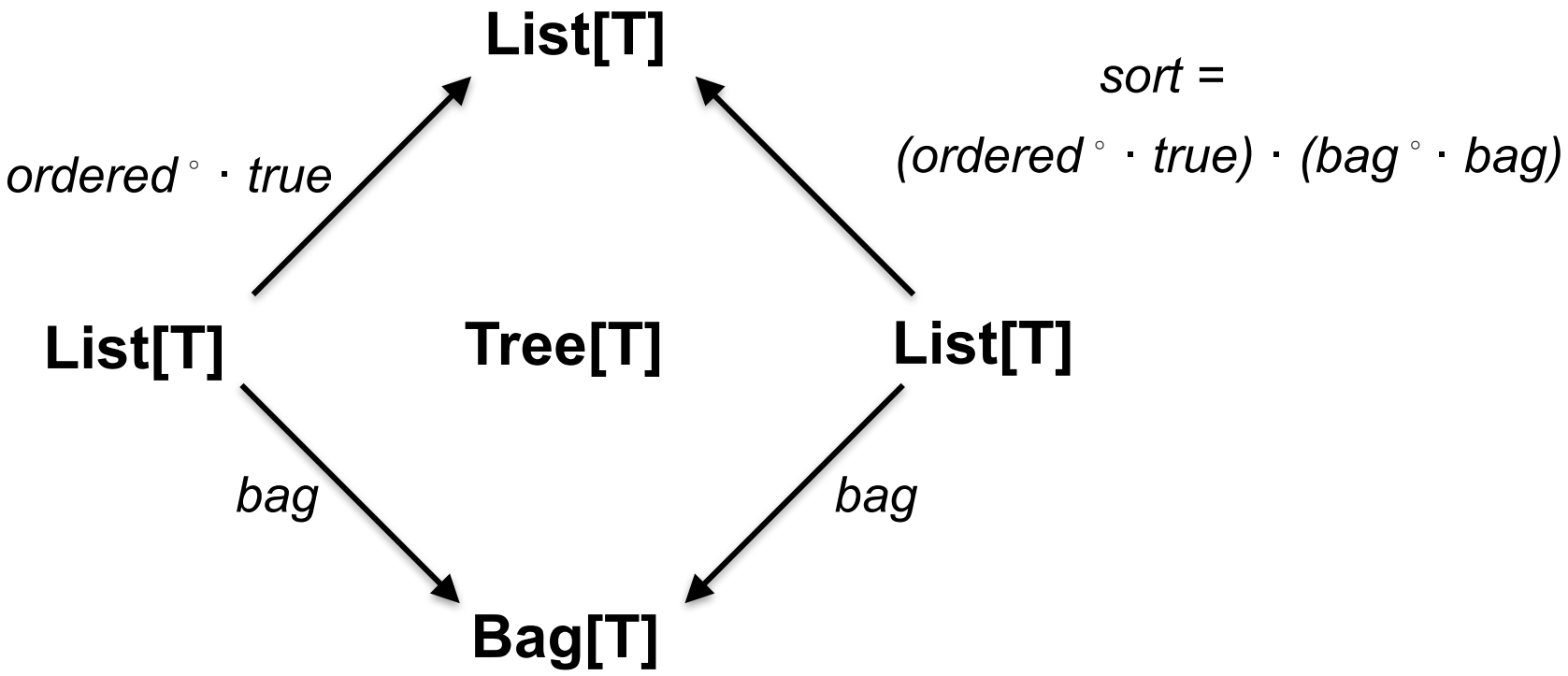
(ordered ∘ · *true)* · *(bag* ∘ · *bag)*

Example: “Mystery Sort”



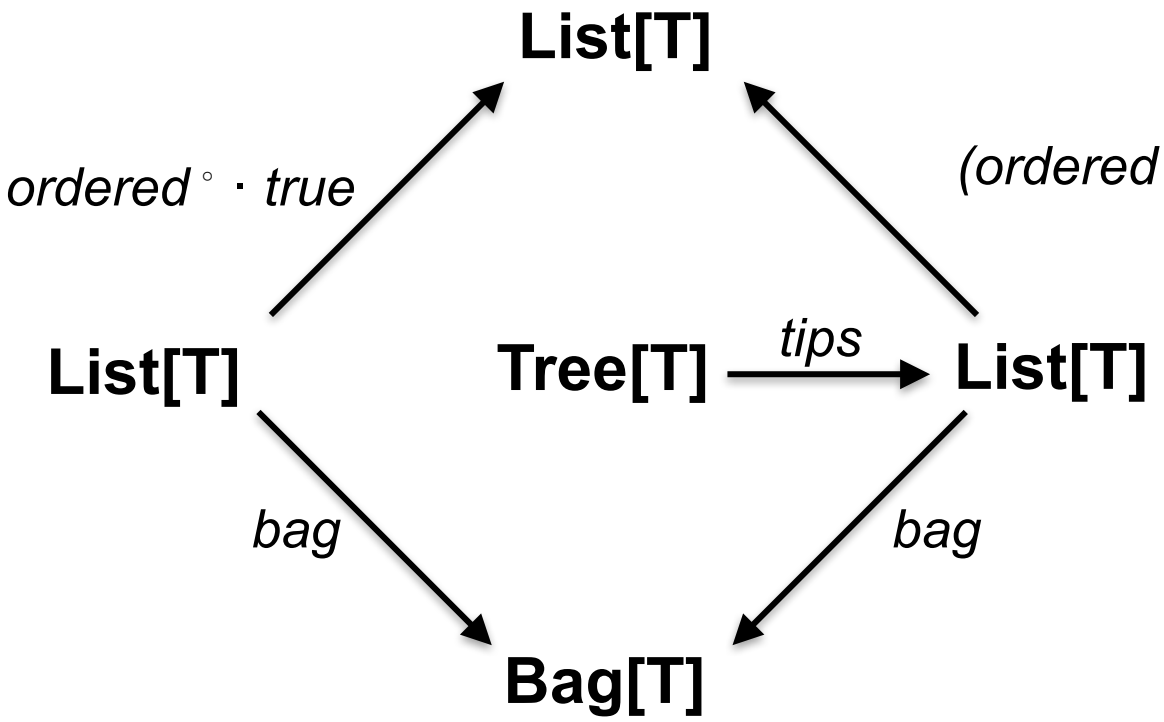
$$sort = (ordered^{\circ} \cdot true) \cdot (bag^{\circ} \cdot bag)$$

Example: “Mystery Sort”



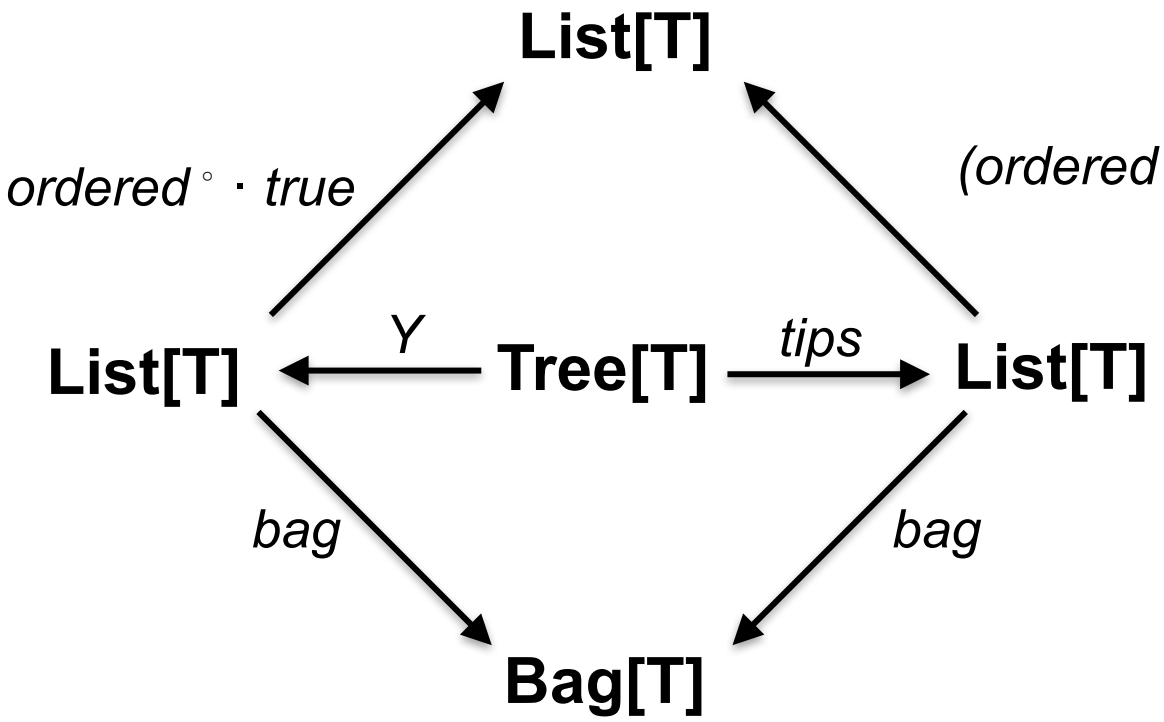
$$sort = (ordered^{\circ} \cdot true) \cdot (bag^{\circ} \cdot bag)$$

Example: “Mystery Sort”



$$sort = (ordered^{\circ} \cdot true) \cdot (bag^{\circ} \cdot bag)$$

Example: “Mystery Sort”



$$sort = (ordered^{\circ} \cdot true) \cdot (bag^{\circ} \cdot bag)$$

Example: “Mystery Sort”

- Lemma (trust me…):

$$
\begin{aligned}
& sort \\
&= (ordered^{\circ} \cdot true) \cdot (bag^{\circ} \cdot bag) \\
&= \underbrace{(ordered^{\circ} \cdot true) \cdot bag^{\circ} \cdot bag \cdot tips}_{Y} \cdot tips^{\circ}
\end{aligned}
$$

Example: “Mystery Sort”

- Punchline: A function satisfying *Y* is the “list merge” seen in merge sort.
- We have derived **Merge Sort**!
- Different choices lead to different sorting algorithms

CONCLUSION

Conclusion

- Metaphors are everywhere
- We can use them in programming
- The Bird-Meertens Formalism is cool
- Functions to relations
- Keep climbing the ladder of abstraction

Meta-metaphors?

Burning the candle at both ends
Might seem super-neat,
But I'll guarantee it's not the way
To make our two ends meet!

-Ruth M. Walsh

ACKNOWLEDGEMENTS

William Harvey, PhD

Principal Engineer, OCLC

Lynn Silipigni Connaway, PhD

Senior Research Scientist & Director of User Research, OCLC

Erin M. Hood

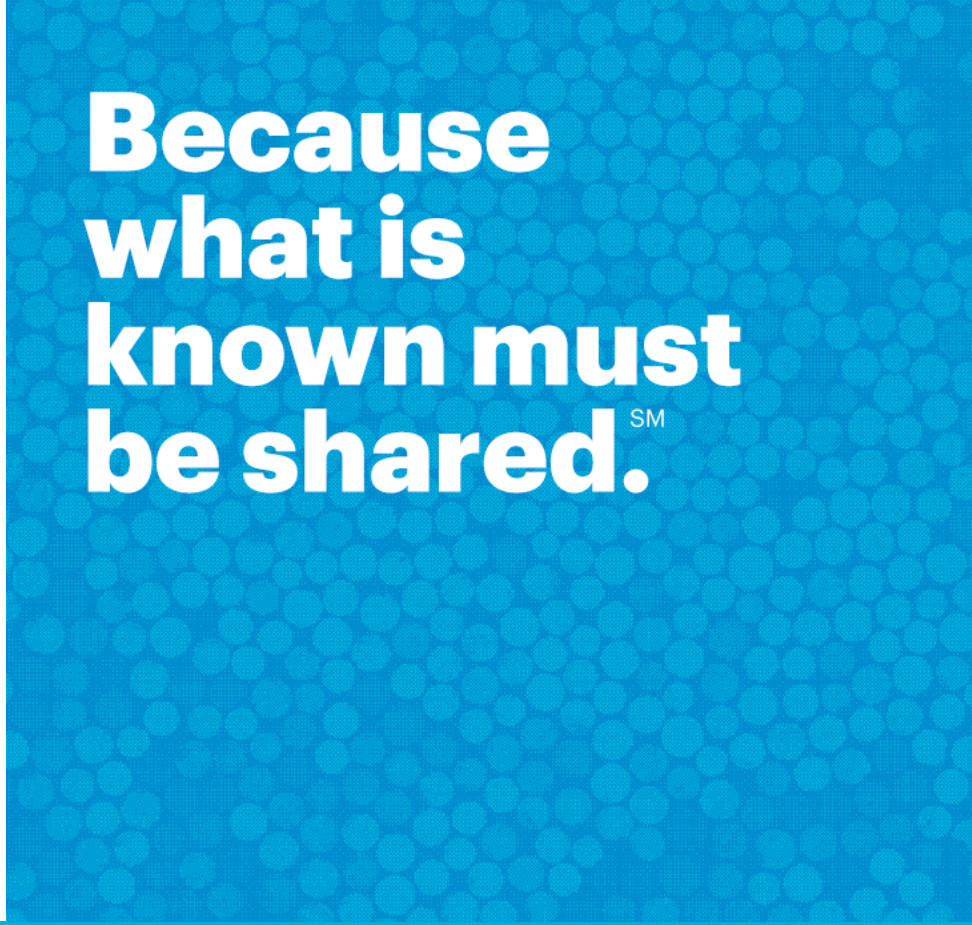
Research Support Specialist, OCLC

Brittany Brannon

Research Assistant, OCLC

Thank You!

William Harvey, PhD
Principal Engineer
harveyw@oclc.org



OCLC