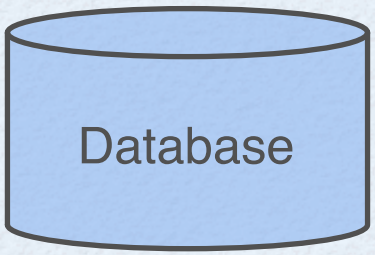
Typing Ad Hoc Data

Kathleen Fisher
AT&T Labs Research

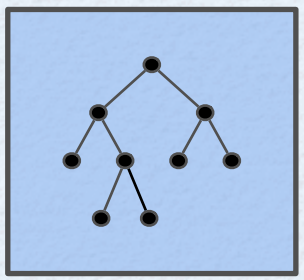
Data, Data, Everywhere!

Incredible amounts of data stored in well-behaved formats:

Databases:



XML:

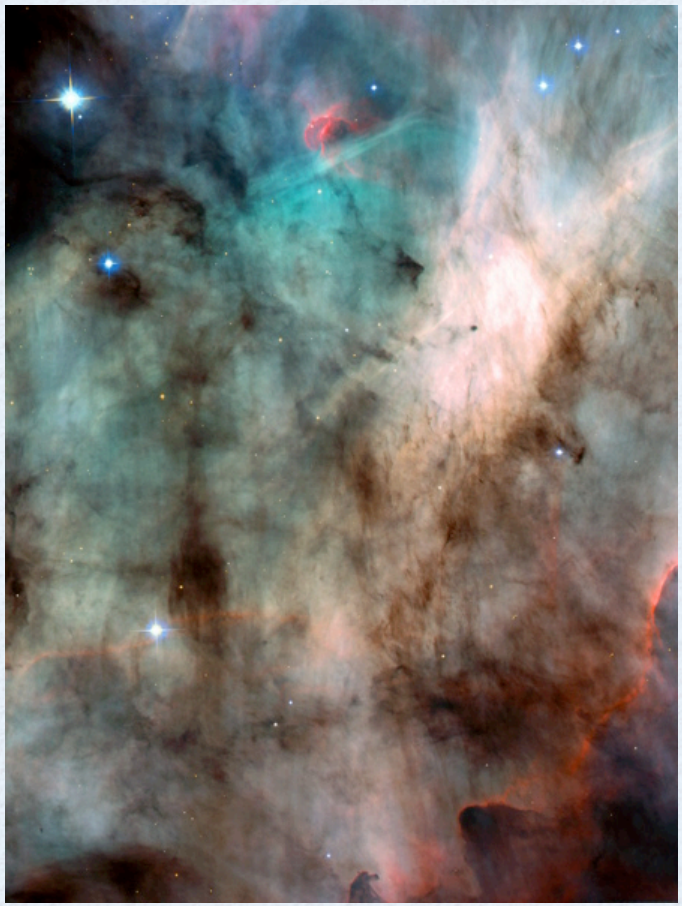


Tools

- Schema
- Browsers
- Query Languages
- Standards
- Libraries
- Books, documentation
- Training courses
- Conversion tools
- Vendor support
- Consultants...

We're not always so lucky!

Vast amounts of chaotic *ad hoc* data:



Tools

- Perl
- Awk
- C
- ...

GOVERNMENT STATS

```
"MSN","YYYYMM","Publication Value","Publication Unit","Column Order"
"TEAJBUS",197313,-0.456483,Quadrillion Btu,4
"TEAJBUS",197413,-0.482265,Quadrillion Btu,4
"TEAJBUS",197513,-1.066511,Quadrillion Btu,4
"TEAJBUS",197613,-0.177807,Quadrillion Btu,4
"TEAJBUS",197713,-1.948233,Quadrillion Btu,4
"TEAJBUS",197813,-0.336538,Quadrillion Btu,4
"TEAJBUS",197913,-1.649302,Quadrillion Btu,4
"TEAJBUS",198013,-1.0537,Quadrillion Btu,4
```

Train Stations

```
Southern California Regional Railroad Authority,"Los Angeles, CA",
U,45,46,46,47,49,51,U,45,46,46,47,49,51
Connecticut Department of Transportation ,"New Haven, CT",
U,U,U,U,U,U,8,U,U,U,U,U,U,8
Tri-County Commuter Rail Authority ,"Miami, FL",
U,U,U,U,U,U,18,U,U,U,U,U,U,18
Northeast Illinois Regional Commuter Railroad Corporation,"Chicago, IL",
226,226,226,227,227,227,227,91,104,104,111,115,125,131
Northern Indiana Commuter Transportation District,"Chicago, IL",
18,18,18,18,18,18,20,7,7,7,7,7,7,11
Massachusetts Bay Transportation Authority,"Boston, MA",
U,U,117,119,120,121,124,U,U,67,69,74,75,78
Mass Transit Administration – Maryland DOT ,"Baltimore, MD",
U,U,U,U,U,U,42,U,U,U,U,U,U,22
New Jersey Transit Corporation ,"New York, NY",
158,158,158,162,162,162,167,22,22,41,46,46,46,51
```

WEB LOGS

```
207.136.97.49 – – [15/Oct/2006:18:46:51 -0700] "GET /turkey/amnty1.gif HTTP/1.0" 200 3013
207.136.97.49 – – [15/Oct/2006:18:46:51 -0700] "GET /turkey/clear.gif HTTP/1.0" 200 76
207.136.97.49 – – [15/Oct/2006:18:46:52 -0700] "GET /turkey/back.gif HTTP/1.0" 200 224
207.136.97.49 – – [15/Oct/2006:18:46:52 -0700] "GET /turkey/women.html HTTP/1.0" 200 17534
208.196.124.26 – Dbuser [15/Oct/2006:18:46:55 -0700] "GET /candatop.html HTTP/1.0" 200 -
208.196.124.26 – – [15/Oct/2006:18:46:57 -0700] "GET /images/done.gif HTTP/1.0" 200 4785
www.att.com – – [15/Oct/2006:18:47:01 -0700] "GET /images/reddash2.gif HTTP/1.0" 200 237
208.196.124.26 – – [15/Oct/2006:18:47:02 -0700] "POST /images/refrun1.gif HTTP/1.0" 200 836
208.196.124.26 – – [15/Oct/2006:18:47:05 -0700] "GET /images/hasene2.gif HTTP/1.0" 200 8833
www.cnn.com – – [15/Oct/2006:18:47:08 -0700] "GET /images/candalog.gif HTTP/1.0" 200 -
208.196.124.26 – – [15/Oct/2006:18:47:09 -0700] "GET /images/nigpost1.gif HTTP/1.0" 200 4429
208.196.124.26 – – [15/Oct/2006:18:47:09 -0700] "GET /images/rally4.jpg HTTP/1.0" 200 7352
128.200.68.71 – – [15/Oct/2006:18:47:11 -0700] "GET /amnesty/usalinks.html HTTP/1.0" 143 10329
208.196.124.26 – – [15/Oct/2006:18:47:11 -0700] "GET /images/reyes.gif HTTP/1.0" 200 10859
```

Genetic Data

```
((raccoon:19.19959,bear:6.80041):0.84600,((sea_lion:
11.99700,seal:12.00300):7.52973,((monkey:100.85930,cat:
47.14069):20.59201,weasel:18.87953):2.09460):3.87382,dog:
25.46154); (Bovine:0.69395,(Gibbon:0.36079,(Orang:0.33636,
(Gorilla:0.17147,(Chimp:0.19268,Human:0.11927):0.08386):
0.06124):0.15057):0.54939,Mouse:1.21460):0.10; (Bovine:
0.69395,(Hylobates:0.36079,(Pongo:0.33636,(G._Gorilla:
0.17147,(P._paniscus:0.19268,H._sapiens:0.11927):0.08386):
0.06124):0.15057):0.54939,Rodent:1.21460);
```

Haskell hi files

```
00000000: 0001 face 0000 0073 0400 0000 3600 0000  .......s....6...
00000010: 3000 0000 3500 0000 3000 0000 0000 0000  0...5...0.......
00000020: 0001 0000 0000 0100 0000 0043 0001 0000  ............C....
00000030: 0002 0200 0000 0200 0000 0300 0000 0200  ................
00000040: 0000 0400 0000 4800 0100 0000 0200 0000  ......H.........
00000050: 0502 0000 0000 0006 0000 0000 0007 0000  ................
00000060: 0001 0000 0000 6800 0000 0000 006f 0000  ......h......o..
00000070: 0000 0100 0000 0800 0000 0968 6173 6b65  ............haske
00000080: 6c6c 3938 0000 0007 4350 5554 696d 6500  ll98....CPUTime.
00000090: 0000 0462 6173 6500 0000 0847 4843 2e42  ...base....GHC.B
000000a0: 6173 6500 0000 0e47 4843 2e46 6f72 6569  ase....GHC.Forei
000000b0: 676e 5074 7200 0000 0e53 7973 7465 6d2e  gnPtr....System.
000000c0: 4350 5554 696d 6500 0000 0a67 6574 4350  CPUTime....getCP
000000d0: 5554 696d 6500 0000 1063 7075 5469 6d65  UTime....cpuTime
000000e0: 5072 6563 6973 696f 6e                   Precision
```

AND MANY OTHERS...

- Gene ontology data
- Cosmology data
- Financial trading data
- Telecom billing data
- Router config files
- System logs
- Call detail data
- Netflow packets
- DNS packets
- Java JAR files
- Jazz recording info
- ...

Types to the Rescue!

Relational and XML data are relatively easy to manage (partly) because schema exist to describe the data.

Relational Data	Relational Schema
XML	XML Schema
Ad Hoc Data	???

Types to the Rescue!

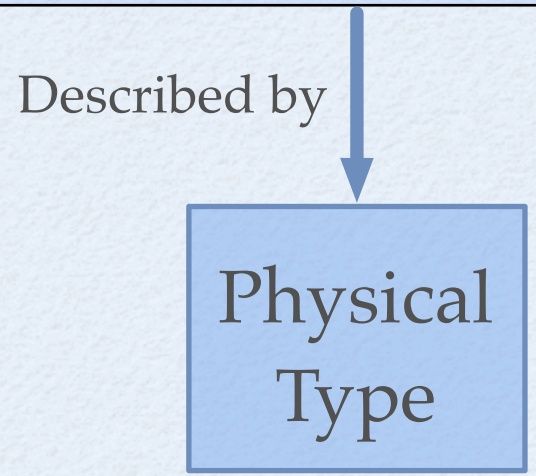
Relational and XML data are relatively easy to manage (partly) because schema exist to describe the data.

Relational Data	Relational Schema
XML	XML Schema
Ad Hoc Data	***Physical Types***

Thesis: Types can facilitate ad hoc data management, and *the types developed for in-memory values are suited to the task.*

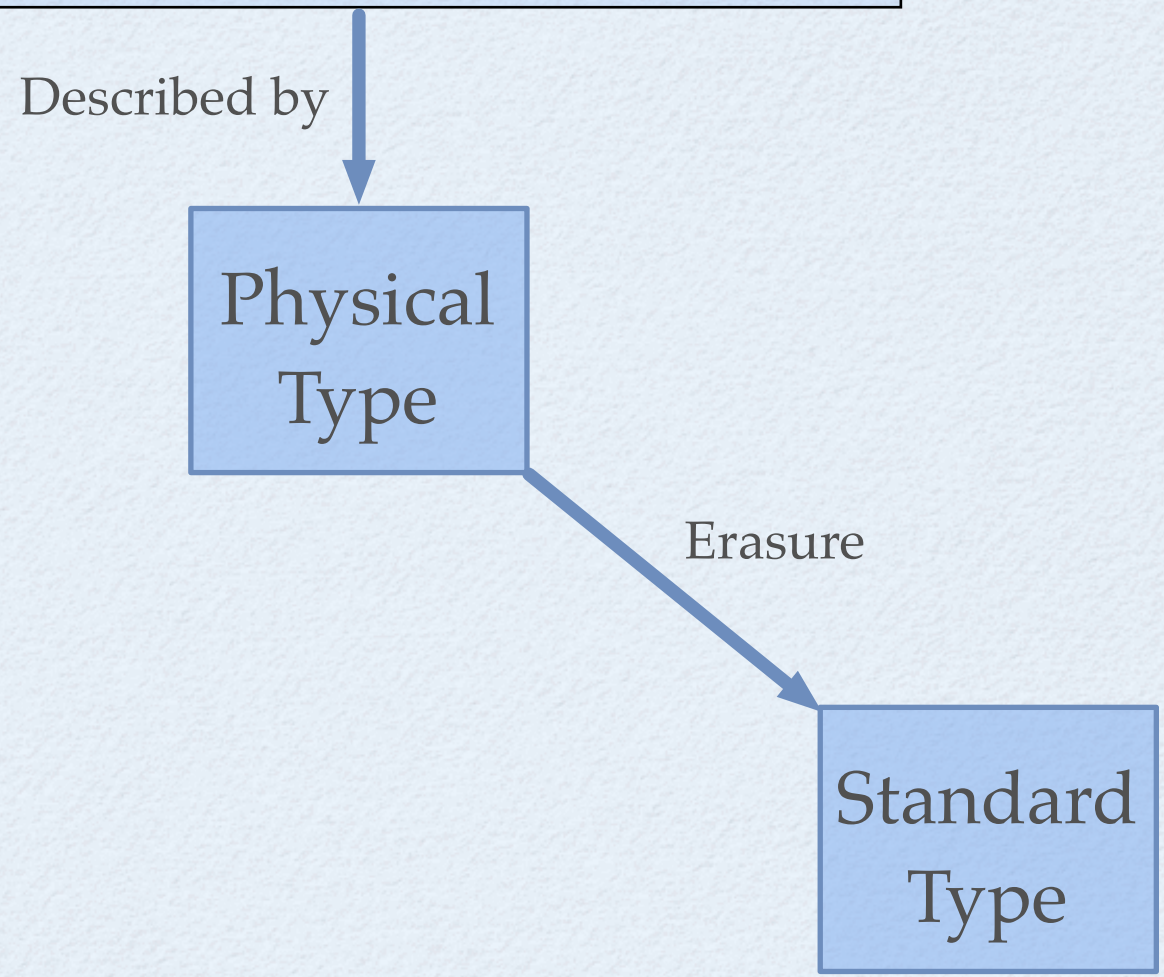
Typing Ad Hoc Data

```
"TEAJBUS",197713,-1.948233,Quadrillion Btu,4
"TEAJBUS",197813,-0.336538,Quadrillion Btu,4
"TEAJBUS",197913,-1.649302,Quadrillion Btu,4
"TEAJBUS",198013,-1.0537,Quadrillion Btu,4
```



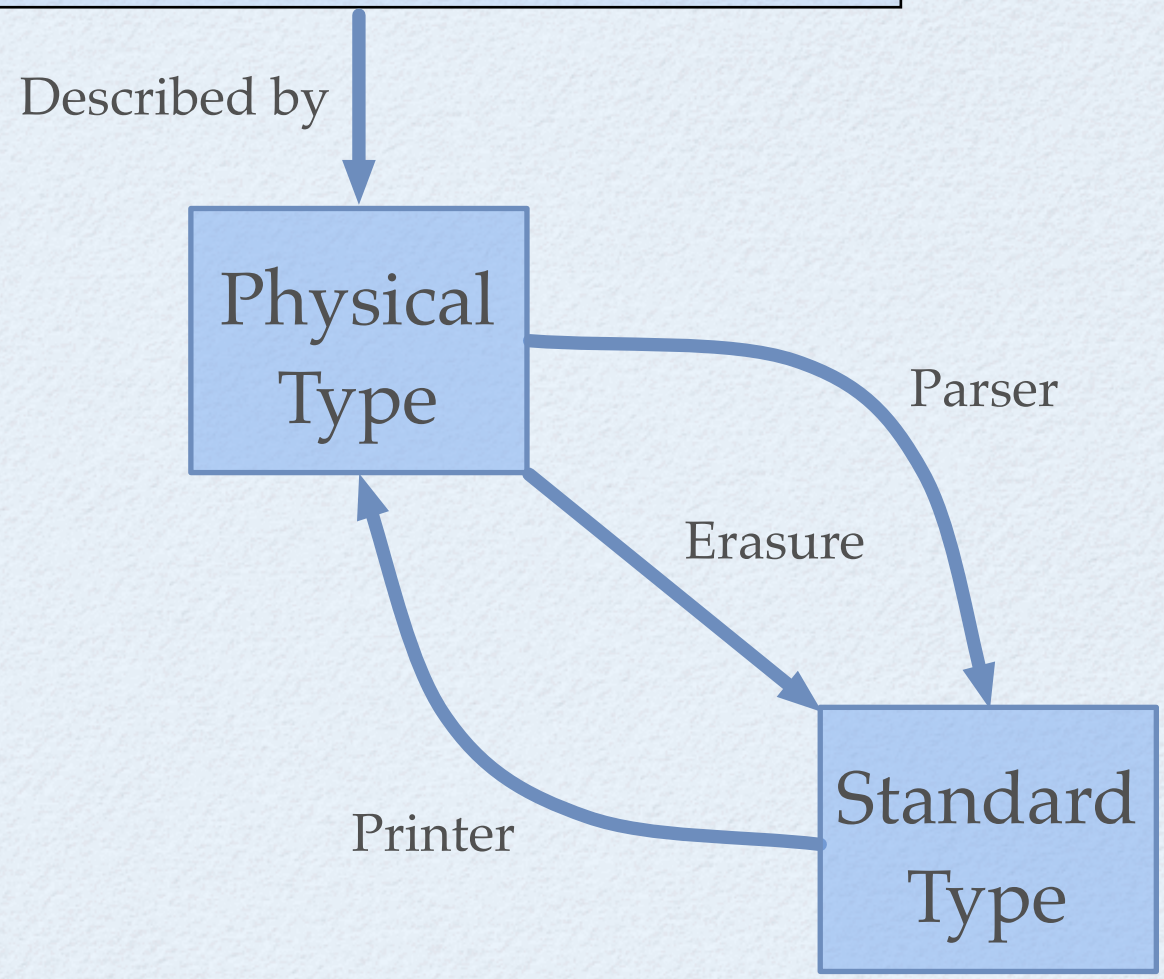
Typing Ad Hoc Data

```
"TEAJBUS",197713,-1.948233,Quadrillion Btu,4
"TEAJBUS",197813,-0.336538,Quadrillion Btu,4
"TEAJBUS",197913,-1.649302,Quadrillion Btu,4
"TEAJBUS",198013,-1.0537,Quadrillion Btu,4
```



Typing Ad Hoc Data

```
"TEAJBUS",197713,-1.948233,Quadrillion Btu,4
"TEAJBUS",197813,-0.336538,Quadrillion Btu,4
"TEAJBUS",197913,-1.649302,Quadrillion Btu,4
"TEAJBUS",198013,-1.0537,Quadrillion Btu,4
```



Roadmap

- Introduction
- Exploring how types describe physical data
- Differences
- Further connections
- Physical type inference
- Conclusion

BASE TYPES

```
"TEAJBUS",197313,-0.456483,Quadrillion Btu,4
"TEAJBUS",197413,-0.482265,Quadrillion Btu,4
```

BASE TYPES

```
"TEAJBUS",197313,-0.456483,Quadrillion Btu,4
"TEAJBUS",197413,-0.482265,Quadrillion Btu,4
```

String, Int, Float

Tuple Types

```
"TEAJBUS",197313,-0.456483,Quadrillion Btu,4
"TEAJBUS",197413,-0.482265,Quadrillion Btu,4
```

```
String * Int * Float * String * Int
```

Singleton Types

```
"TEAJBUS",197313,-0.456483,Quadrillion Btu,4
"TEAJBUS",197413,-0.482265,Quadrillion Btu,4
```

```
'\"' * String * '\"' * ','
     * Int            * ','
     * Float          * ','
     * String         * ','
     * Int
```

Where we write `','` for `S(',')`.

Simple Dependent Types

```
"TEAJBUS",197313,-0.456483,Quadrillion Btu,4
"TEAJBUS",197413,-0.482265,Quadrillion Btu,4
```

```
'\"' * String('\"') * '\"' * ','
     * Int                  * ','
     * Float                * ','
     * String(',')          * ','
     * Int
```

RECORDS

```
"TEAJBUS",197313,-0.456483,Quadrillion Btu,4
"TEAJBUS",197413,-0.482265,Quadrillion Btu,4
```

```
{                                   “\””,
  source: String(‘\”’),             “\”,”,
  date:   Int,                      “,”,
  measurement: Float,               “,”,
  units:  String(‘,’)               “,”,
  order:  Int
}
```

Unions

```
Southern California Regional Railroad Authority,"Los Angeles, CA",
U,45,46,46,47,49,51,U,45,46,46,47,49,51
Connecticut Department of Transportation ,"New Haven, CT",
U,U,U,U,U,U,8,U,U,U,U,U,U,8
Tri-County Commuter Rail Authority ,"Miami, FL",
U,U,U,U,U,U,18,U,U,U,U,U,U,18
```

Anonymous:

```
'U' + Int
```

Unions

```
Southern California Regional Railroad Authority,"Los Angeles, CA",
U,45,46,46,47,49,51,U,45,46,46,47,49,51
Connecticut Department of Transportation ,"New Haven, CT",
U,U,U,U,U,U,8,U,U,U,U,U,U,8
Tri-County Commuter Rail Authority ,"Miami, FL",
U,U,U,U,U,U,18,U,U,U,U,U,U,18
```

Anonymous:

```
'U' + Int
```

Named:

```
type OptInt = unavailable of 'U'
            |   available of Int
```

ARRAYS/LISTS

```
Southern California Regional Railroad Authority,"Los Angeles, CA",
U,45,46,46,47,49,51,U,45,46,46,47,49,51
Connecticut Department of Transportation ,"New Haven, CT",
U,U,U,U,U,U,8,U,U,U,U,U,U,8
Tri-County Commuter Rail Authority ,"Miami, FL",
U,U,U,U,U,U,18,U,U,U,U,U,U,18
```

```
type OptInt = unavailable of 'U'
            |   available of Int

type counts = OptInt[14]
```

ARRAYS/LISTS

```
Southern California Regional Railroad Authority,"Los Angeles, CA",
U,45,46,46,47,49,51,U,45,46,46,47,49,51
Connecticut Department of Transportation ,"New Haven, CT",
U,U,U,U,U,U,8,U,U,U,U,U,U,8
Tri-County Commuter Rail Authority ,"Miami, FL",
U,U,U,U,U,U,18,U,U,U,U,U,U,18
```

```
type OptInt = unavailable of 'U'
            |   available of Int

type counts = OptInt[14] sep(',')
```

ARRAYS/LISTS

```
Southern California Regional Railroad Authority,"Los Angeles, CA",
U,45,46,46,47,49,51,U,45,46,46,47,49,51
Connecticut Department of Transportation ,"New Haven, CT",
U,U,U,U,U,U,8,U,U,U,U,U,U,8
Tri-County Commuter Rail Authority ,"Miami, FL",
U,U,U,U,U,U,18,U,U,U,U,U,U,18
```

```
type OptInt = unavailable of 'U'
            |   available of Int

type counts = OptInt[] sep(',')
                       term(eor)
```

Dependent Types

```
sdw-01.ab.ca – – [16/12/06] "GET /images/fish.gif HTTP/1.0" 200 8552
sdw-01.ab.ca – DBUser [16/12/06] "GET /images/bug.gif HTTP/1.0" 200 1357
64.233.161.99 – – [16/12/26] "GET /images/plex.gif HTTP/1.0" 304 -
69.30.123.195 – – [16/12/2006] "GET /images/adjoint.gif HTTP/1.0" 304 -
```

```
type responseCode = { x : Int | 99 < x < 600}
```

DEPENDENT TYPES

```
sdw-01.ab.ca – – [16/12/06] "GET /images/fish.gif HTTP/1.0" 200 8552
sdw-01.ab.ca – DBUser [16/12/06] "GET /images/bug.gif HTTP/1.0" 200 1357
64.233.161.99 – – [16/12/26] "GET /images/plex.gif HTTP/1.0" 304 -
69.30.123.195 – – [16/12/2006] "GET /images/adjoint.gif HTTP/1.0" 304 -
```

```
type method = GET | POST | LINK | UNLINK | ...

fun  check(method, major, minor) = ...

type request =
   { method : method,          ' ',
     url    : String(' '), " HTTP/",
     major  : Int,             '.',
     minor  : Int
   } where check(method, major, minor)
```

Value Abstraction

```
sdw-01.ab.ca – DBUser [16/12/06] "GET /images/bug.gif HTTP/1.0" 200 1357
64.233.161.99 – – [16/12/26] "GET /images/plex.gif HTTP/1.0" 304 -
```

```
Connecticut Department of Transportation ,"New Haven, CT",
U,U,U,U,U,U,8,U,U,U,U,U,U,8
```

Two representations for missing integer: `'U'` and `'-'`.
We can use value abstraction to reduce redundancy:

```
type OptInt  = λx.unavailable of x
             |   available of Int
type OptIntU = OptInt 'U'
type OptIntD = OptInt '-'
```

Type Abstraction

```
sdw-01.ab.ca - DBUser [16/12/06] "GET /images/bug.gif HTTP/1.0" 200 1357
64.233.161.99 - - [16/12/26] "GET /images/plex.gif HTTP/1.0" 304 -
```

```
Connecticut Department of Transportation ,"New Haven, CT",
U,U,U,U,U,U,8,U,U,U,U,U,U,8
```

Two different types can be missing: `Int` and `String`.

We can use type abstraction to reduce redundancy:

```
type Opt = Λt.λx.unavailable of x
                 |  available of t
type OptIntU = Opt Int 'U'
type OptIntD = Opt Int '-'
type OptStringD = λy.Opt (String y)'-'
```

Recursive Types

```
(Bovine:0.69395,(Gibbon:0.36079,(Orang:0.33636,(Gorilla:
0.17147,(Chimp:0.19268,Human:0.11927):0.08386):0.06124):
0.15057):0.54939,Mouse:1.21460):0.10
```

```
type Entry = {name : String(':'), ':',
              dist : Float}
type Tree  =
   Leaf of Entry
 | Interior of
   '(' * Tree[] (',', ')') * ')' * ':' * Float
```

Recursive Types

```
(Bovine:0.69395,(Gibbon:0.36079,(Orang:0.33636,(Gorilla:
0.17147,(Chimp:0.19268,Human:0.11927):0.08386):0.06124):
0.15057):0.54939,Mouse:1.21460):0.10
```

```
type Entry = {name : String(':'), ':',
              dist : Float}
type Tree  =
   Leaf of Entry
 | Interior of
   '(' * Tree[] (',', ')') * ')' * ':' * Float
```

RECURSIVE TYPES

```
(Bovine:0.69395,(Gibbon:0.36079,(Orang:0.33636,(Gorilla:
0.17147,(Chimp:0.19268,Human:0.11927):0.08386):0.06124):
0.15057):0.54939,Mouse:1.21460):0.10
```

```
type Entry = {name : String(':'), ':',
              dist : Float}
type Tree  =
   Leaf of Entry
 | Interior of
   '(' * Tree[] (',', ')') * ')' * ':' * Float
```

POINTERS?

```
00000000: 0001 face 0000 0073 0400 0000 3600 0000  .......s....6...
00000010: 3000 0000 3500 0000 3000 0000 0000 0000  0...5...0.......
00000020: 0001 0000 0000 0100 0000 0043 0001 0000  ............C....
...
00000070: 0000 0100 0000 0800 0000 0968 6173 6b65  ............haske
00000080: 6c6c 3938 0000 0007 4350 5554 696d 6500  ll98....CPUTime.
00000090: 0000 0462 6173 6500 0000 0847 4843 2e42  ...base....GHC.B
000000a0: 6173 6500 0000 0e47 4843 2e46 6f72 6569  ase....GHC.Forei
000000b0: 676e 5074 7200 0000 0e53 7973 7465 6d2e  gnPtr....System.
000000c0: 4350 5554 696d 6500 0000 0a67 6574 4350  CPUTime....getCP
000000d0: 5554 696d 6500 0000 1063 7075 5469 6d65  UTime....cpuTime
000000e0: 5072 6563 6973 696f 6e                   Precision
```

```
type Dictionary = {count : SBH_uint32(4),
                   ids   : Hstring[count]}
...
type Hi = { id   : B_uint32(4) where checkId(id),
            dict : Dictionary Pointer(4), ...}
```

Type Summary

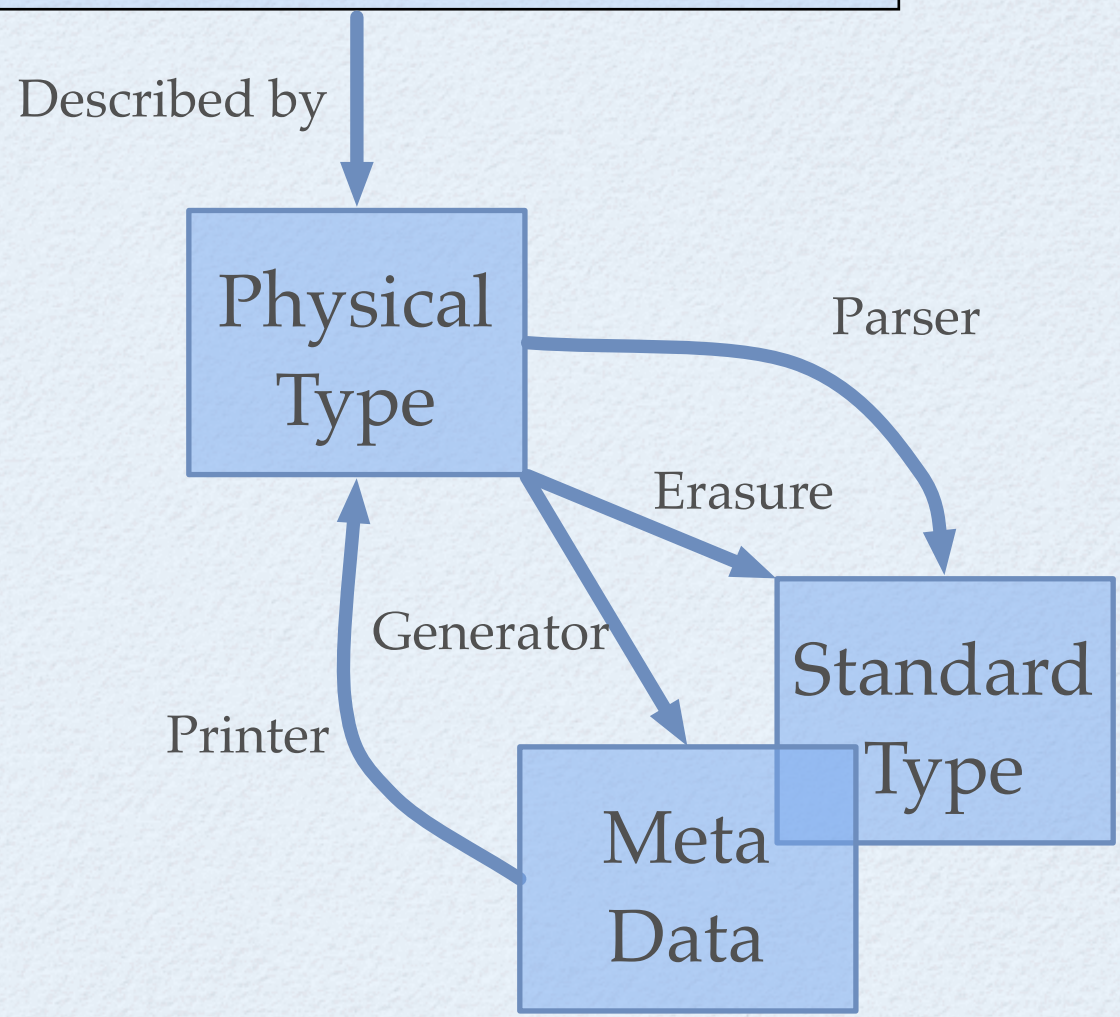
- Base types
- Tuples
- Singleton types
- Records
- Unions
- Lists/Arrays
- Value abstraction
- Type abstraction
- Dependent types
- Recursive types
- Pointers
- ???

Differences

- Data layout *is not* under the control of the type system.
  - Physical types need some extra information: separators, terminators.
  - Many physical types map to the same internal type: `String(' ')`, `String(':')`, `SBH_uint32, B_uint32 ...`
- Dependent types much more important for physical types:
  - Missing value representations, value-level constraints, embedded array lengths, union tags.
- We should not assume data conforms 100% to description.

Meta Data

```
"TEAJBUS",197713,-1.948233,Quadrillion Btu,4
"TEAJBUS",197813,-0.336538,Quadrillion Btu,4
"TEAJBUS",197913,-1.649302,Quadrillion Btu,4
"TEAJBUS",198013,-1.0537,Quadrillion Btu,4
```



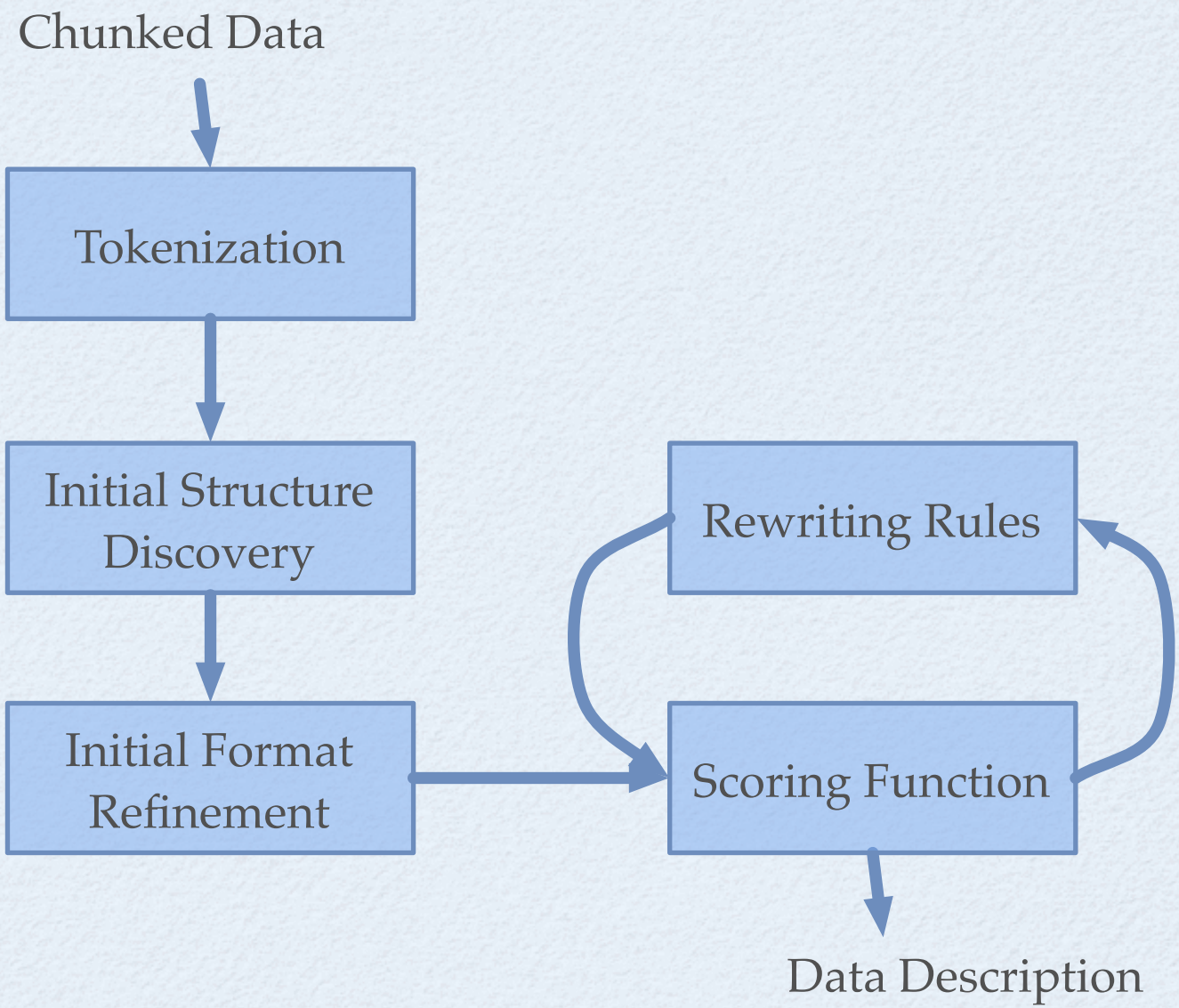
Relevant Work

- The **Data Description Calculus** (DDC) [POPL '06, Mandelbaum's thesis, 2006] formalizes these ideas.
- Some examples in practice:
  - **PADS/C** [PLDI '05] and **PADS/ML** [POPL '07]
  - **PacketTypes** [SIGCOMM '98]
  - **DataScript** [GPCE '02]
  - Erlang's bit types [ESOP '04]
  - **DFDL**

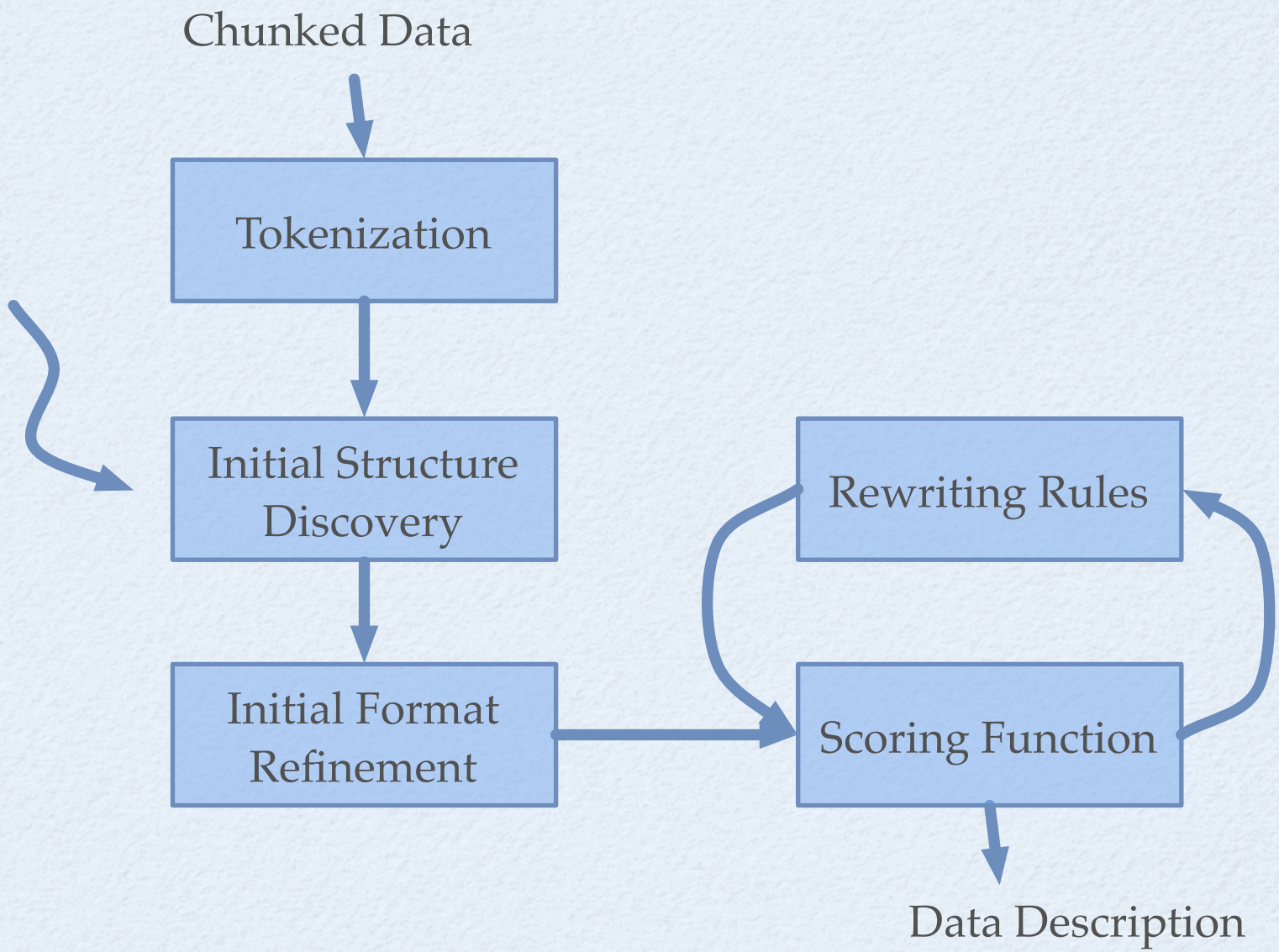
Other PL Ideas

- Analysis to determine "well-formedness" of descriptions:
  - Do union branches overlap?
  - When do printing and parsing compose [Brabrand, et al, DBPL '05]?
  - What is the on-disk size?
- Type-directed programming
  - Support user-defined tools and transformations
- Structural subtyping?
  - Generate conversion from one format to another
- Type equality?
  - Semantic basis for rewriting descriptions (simpler, shredded,...)
- Type inference?

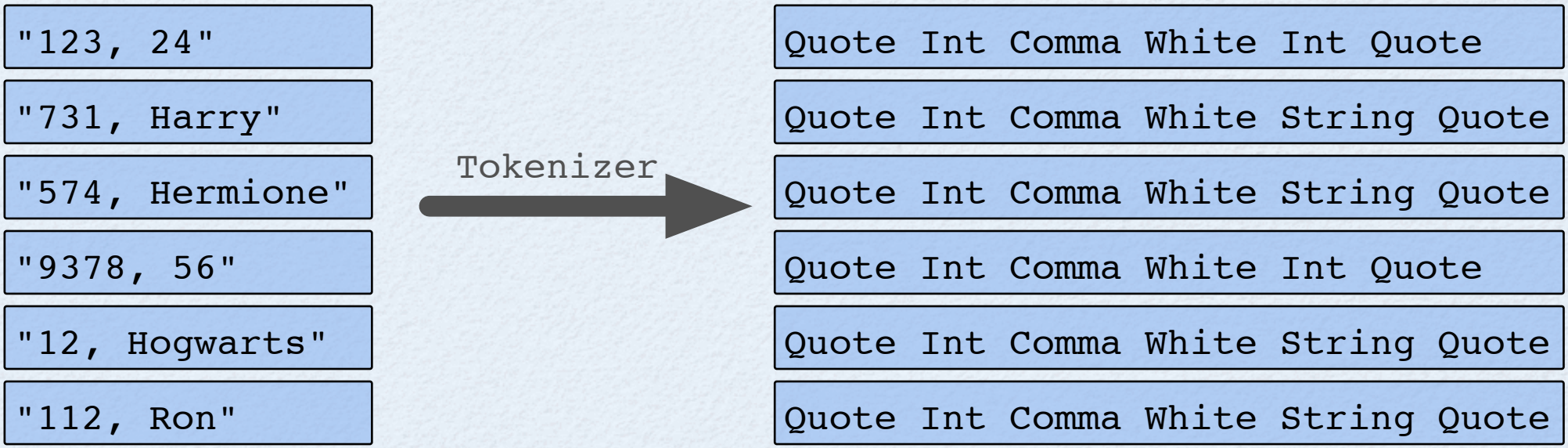
Physical Type Inference



Physical Type Inference



Tokenization

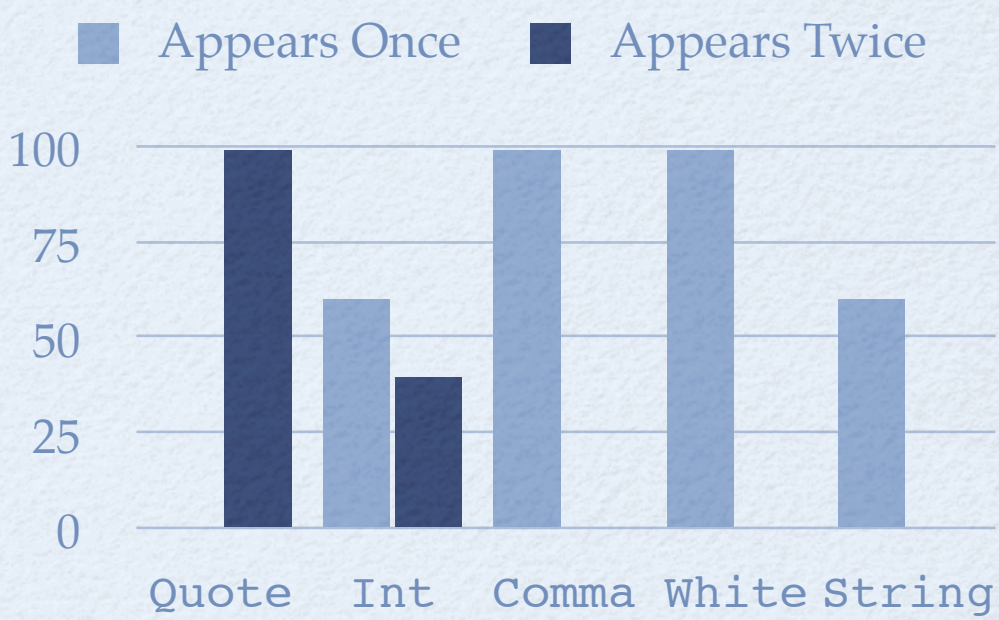


- Tokens expressed as regular expressions.
- **Basic tokens**
  - Integer, white space, punctuation, strings
- **Distinctive tokens**
  - IP addresses, dates, times, MAC addresses, ...

Histograms

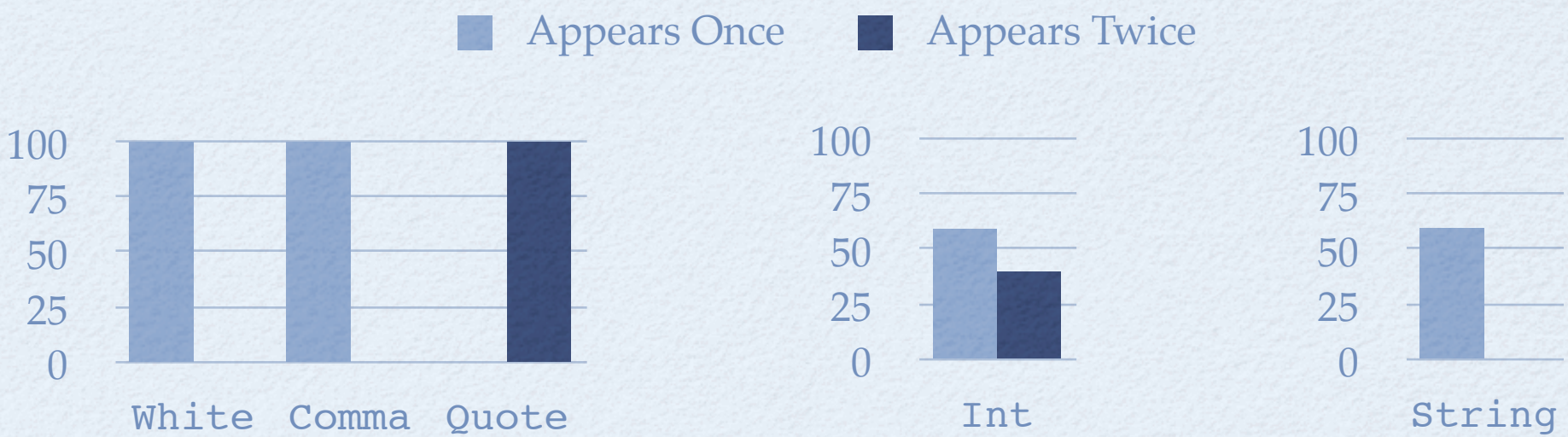
```
Quote Int Comma White Int Quote
Quote Int Comma White String Quote
Quote Int Comma White String Quote
Quote Int Comma White Int Quote
Quote Int Comma White String Quote
Quote Int Comma White String Quote
```

Frequency Analysis



CLUSTERING

Group clusters with similar frequency distributions

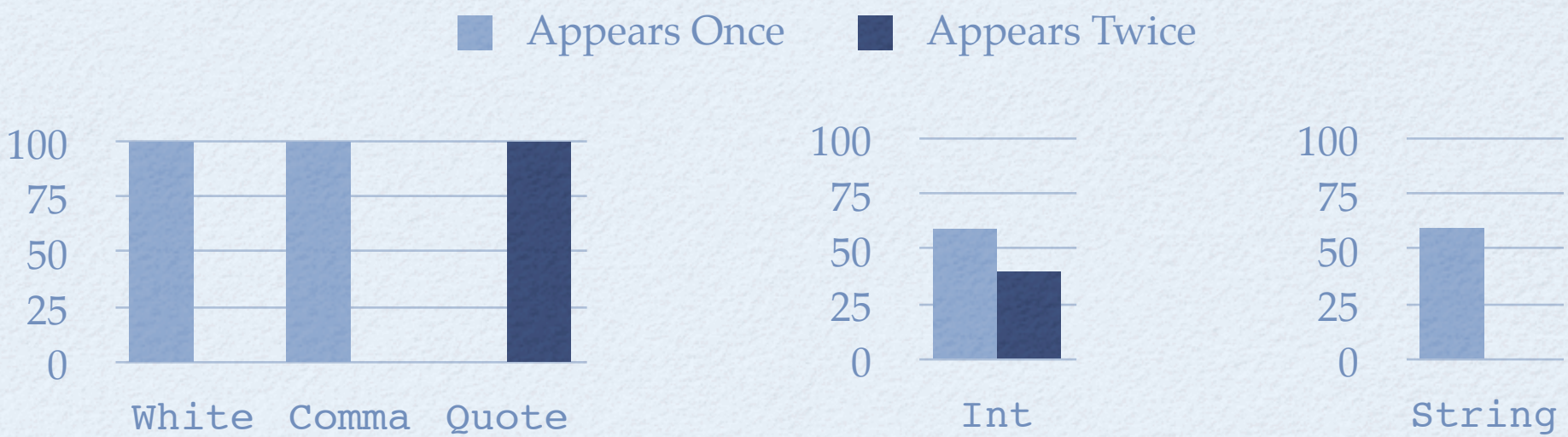


Cluster 1 Cluster 2 Cluster 3

Two frequency distributions are *similar* if they have the same shape (within some error tolerance) when the columns are sorted by height.

CLUSTERING

Group clusters with similar frequency distributions

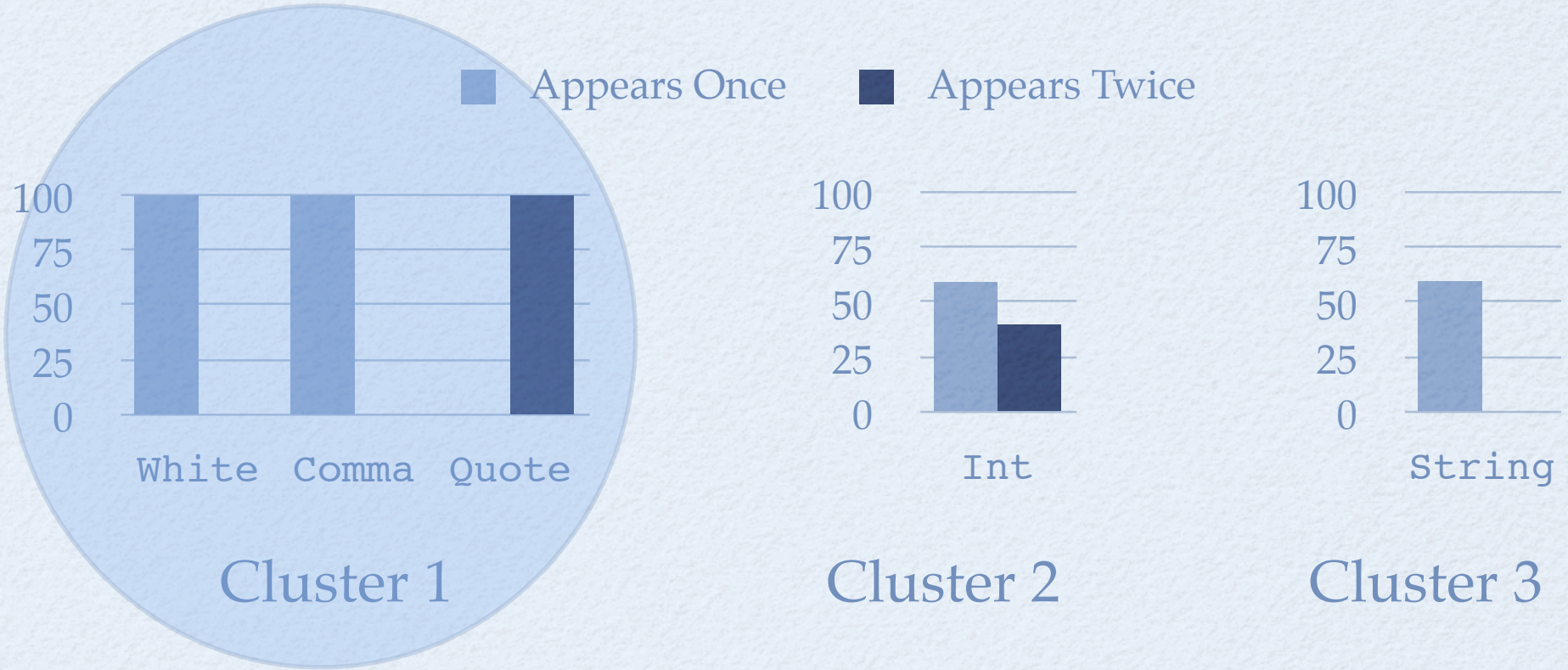


Cluster 1 Cluster 2 Cluster 3

Rank clusters by metric that rewards *high coverage* and *narrower* distributions. Chose cluster with highest score.

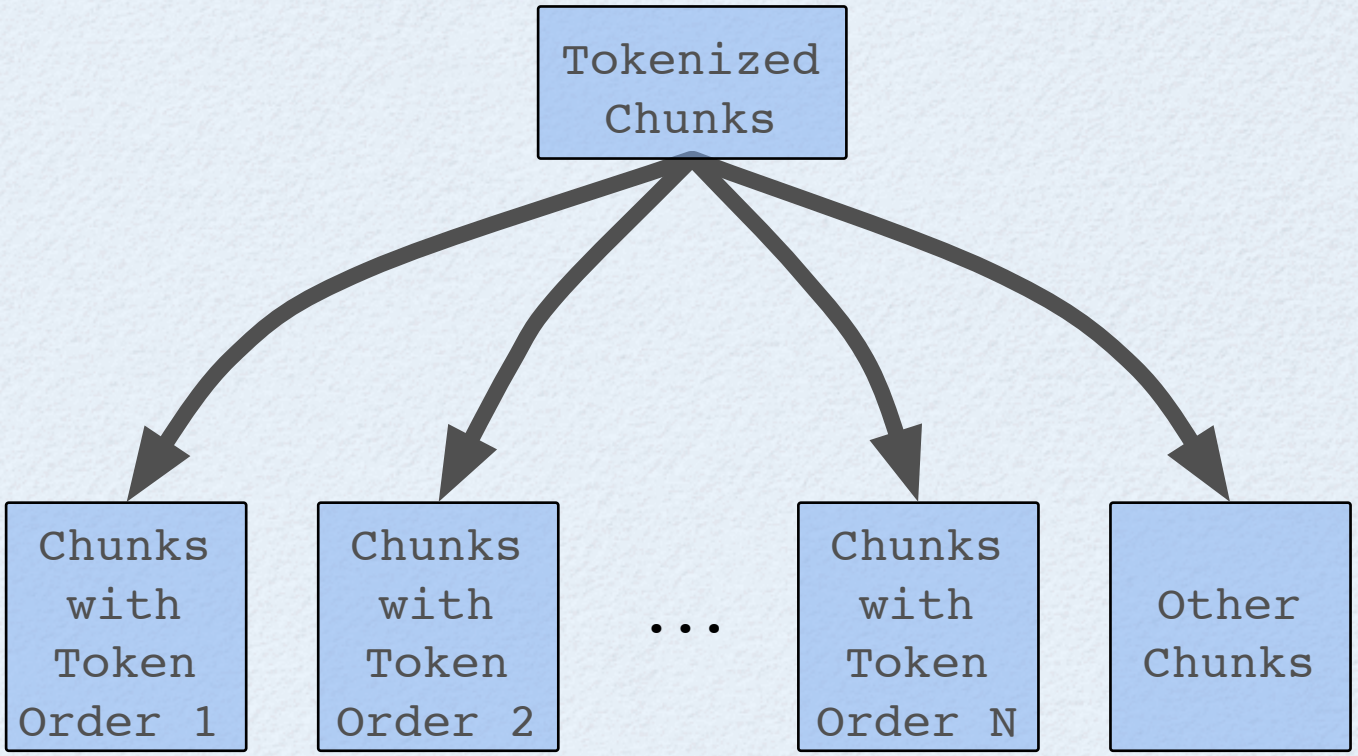
CLUSTERING

Group clusters with similar frequency distributions



Rank clusters by metric that rewards *high coverage* and *narrower* distributions. Chose cluster with highest score.

Partition Chunks



$$\tau_1 + \tau_2 \ldots + \ldots \tau_N + \tau_{other}$$

In our example, all the tokens appear in the same order in all chunks, so the union is degenerate.

FIND SUBCONTEXTS

Quote Int **Comma** **White** Int **Quote**

Quote Int **Comma** **White** String **Quote**

Quote Int **Comma** **White** String **Quote**

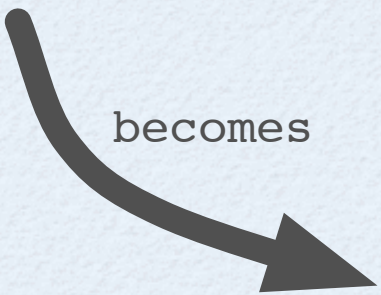
Quote Int **Comma** **White** Int **Quote**

Quote Int **Comma** **White** String **Quote**

Quote Int **Comma** **White** String **Quote**

Tokens in selected cluster:

`Quote(2) Comma White`



Quote *

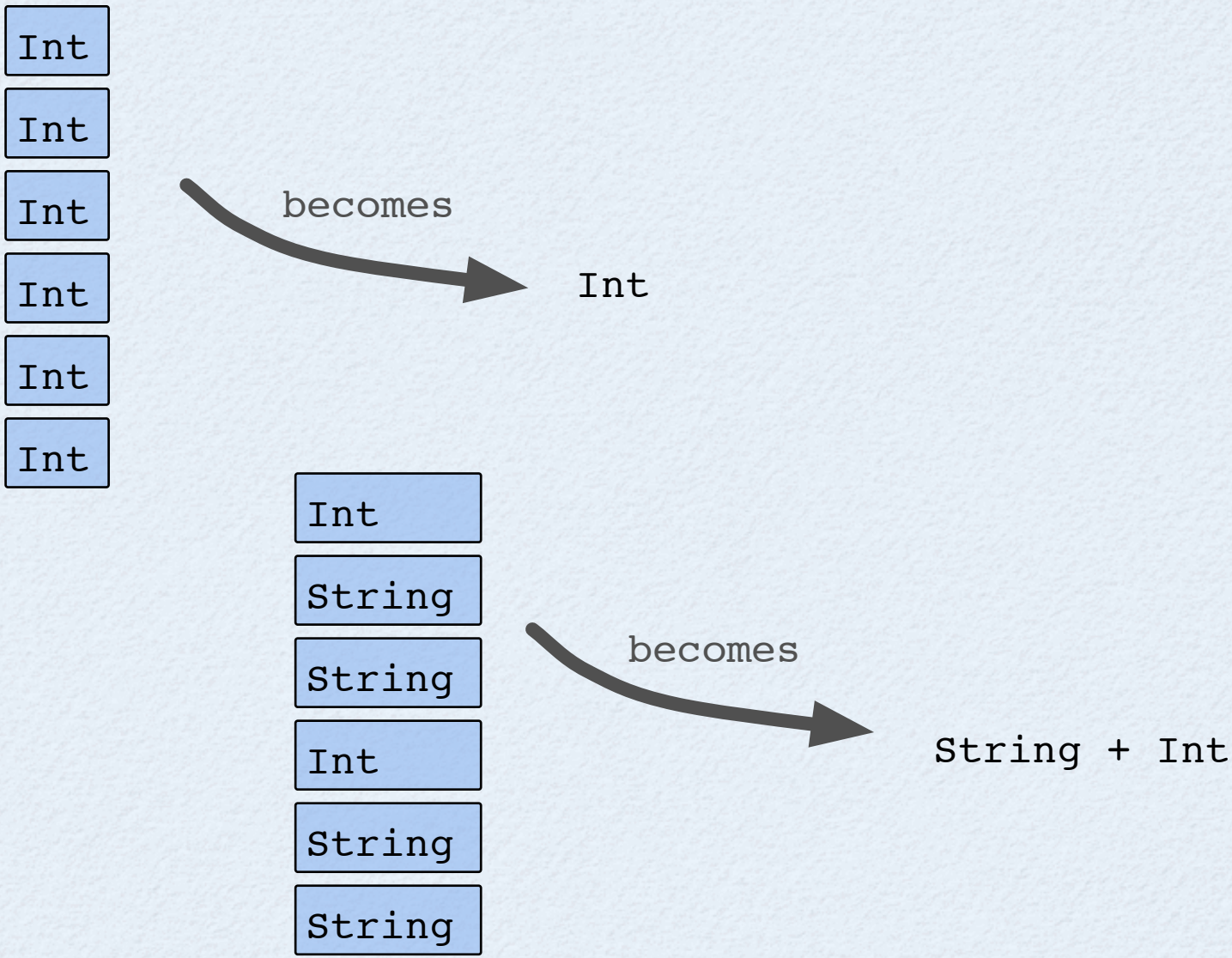
Int
Int
Int
Int
Int
Int

* Comma * White *

Int
String
String
Int
String
String

* Quote

THEN RECURSE...



INFERRED TYPE

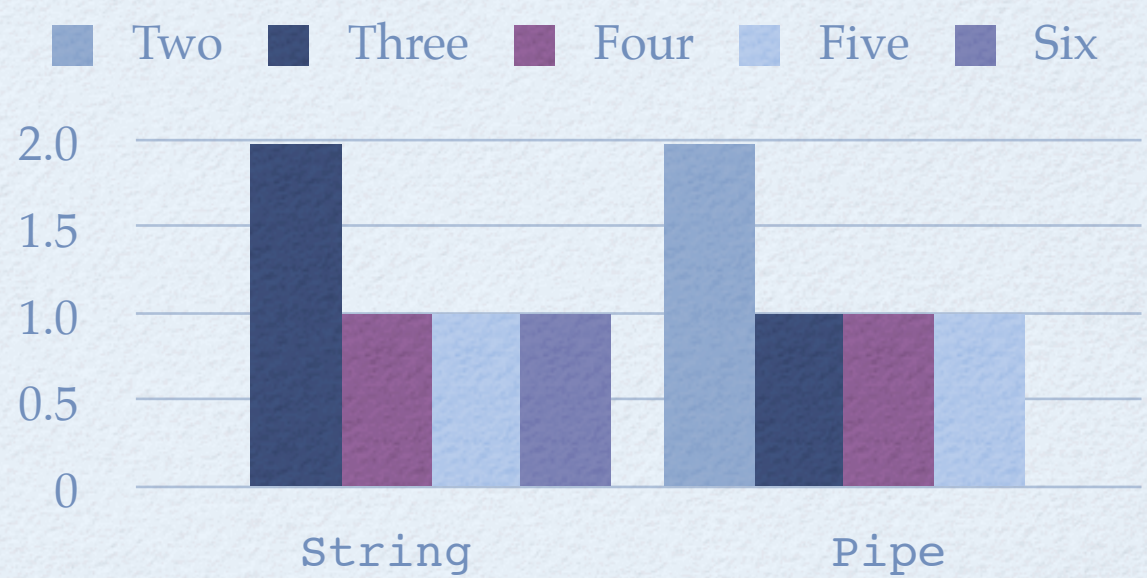
```
"123, 24"
"731, Harry"
"574, Hermione"
"9378, 56"
"12, Hogwarts"
"112, Ron"
```


```
Quote * Int * Comma * White * (String + Int) * Quote
```

FINDING ARRAYS

```
hermione|ginny|lavender
malfoy|crabbe|goyle|parkinson|bulstrode|greengrass|nott|zabini
harry|ron|nevil|george|fred
trevor|ginger|hedwig
flitwick|mcgonagall|snape|sprout
quirrell|lockhart|lupin|moody|umbridge|snape
```

Single cluster with high coverage, but *wide* distribution.



PARTITIONING

Selected tokens for array cluster: `String Pipe`

hermione| ginny| lavender

malfoy| crabbe| goyle| parkinson| bulstrode| ... zabini

harry| ron| nevil| george| fred

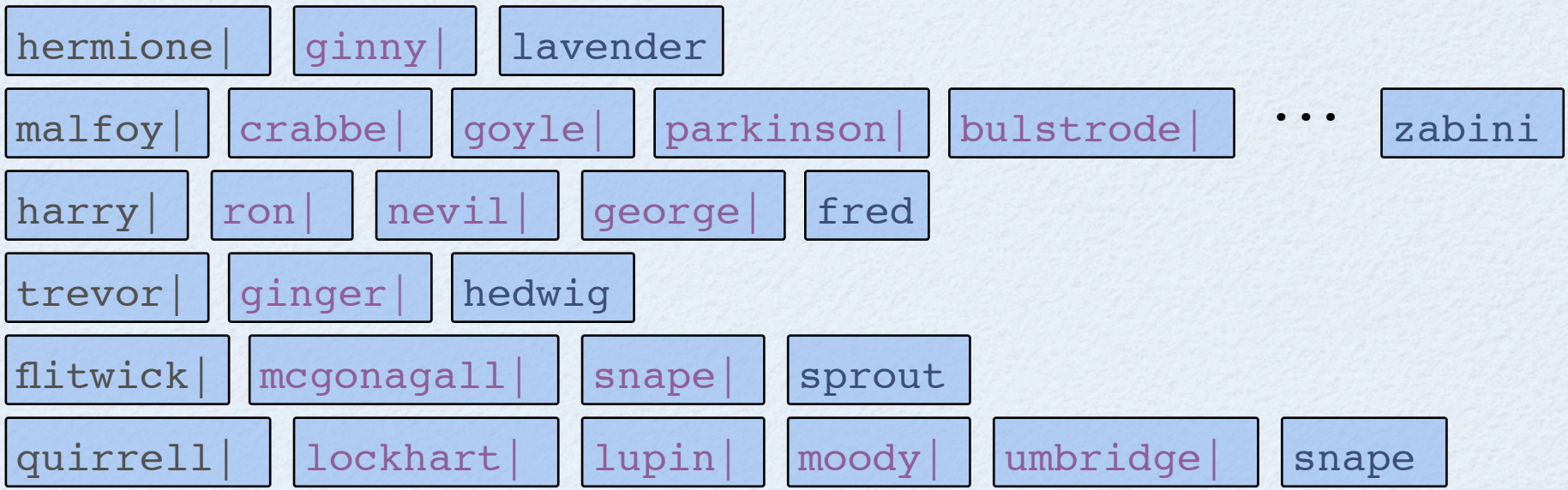
trevor| ginger| hedwig

flitwick| mcgonagall| snape| sprout

quirrell| lockhart| lupin| moody| umbridge| snape

PARTITIONING

Selected tokens for array cluster: `String Pipe`

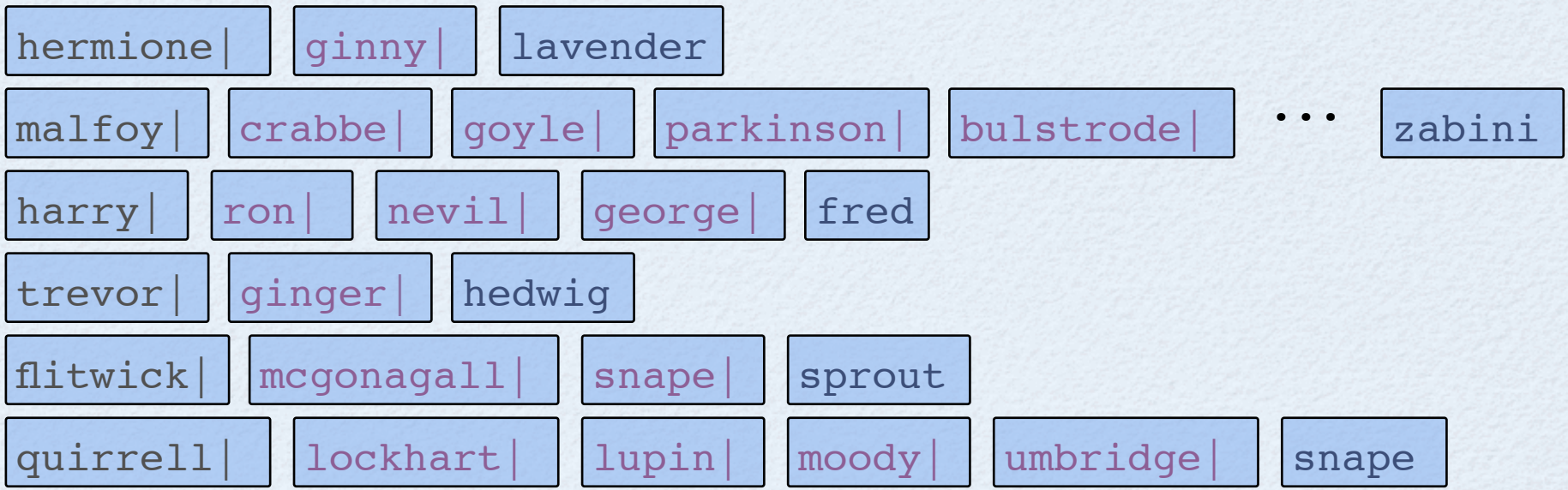


Context 1,2:

`String * Pipe`

PARTITIONING

Selected tokens for array cluster: `String Pipe`



Context 1,2:

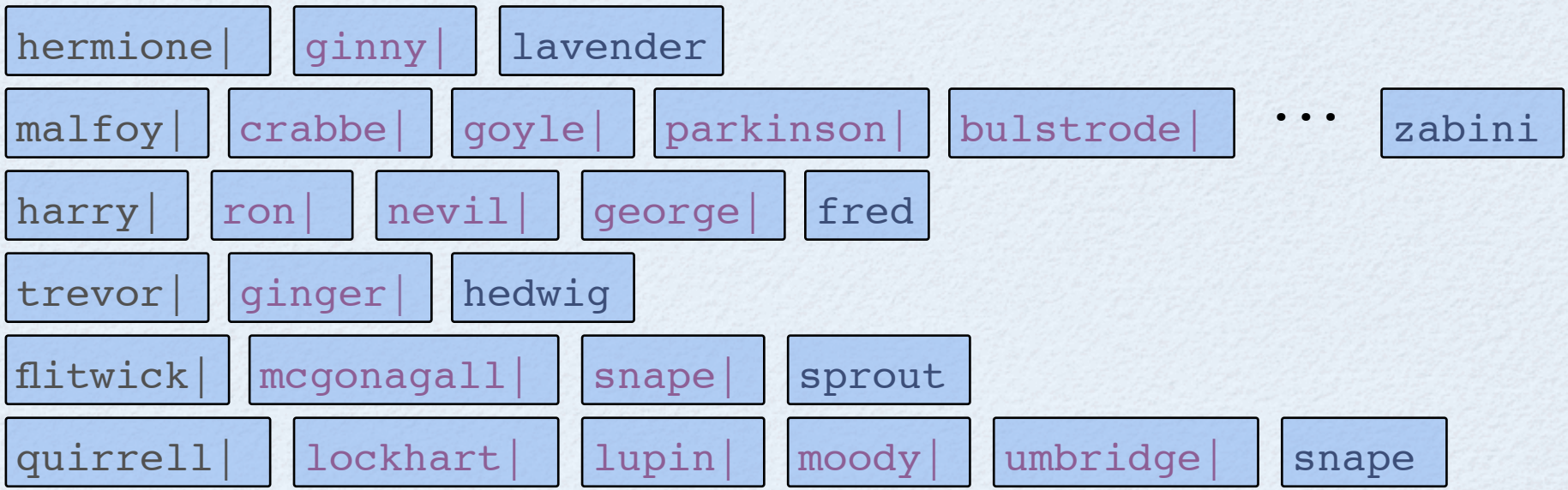
`String * Pipe`

Context 3:

`String`

PARTITIONING

Selected tokens for array cluster: `String Pipe`



Context 1,2:

`String * Pipe`

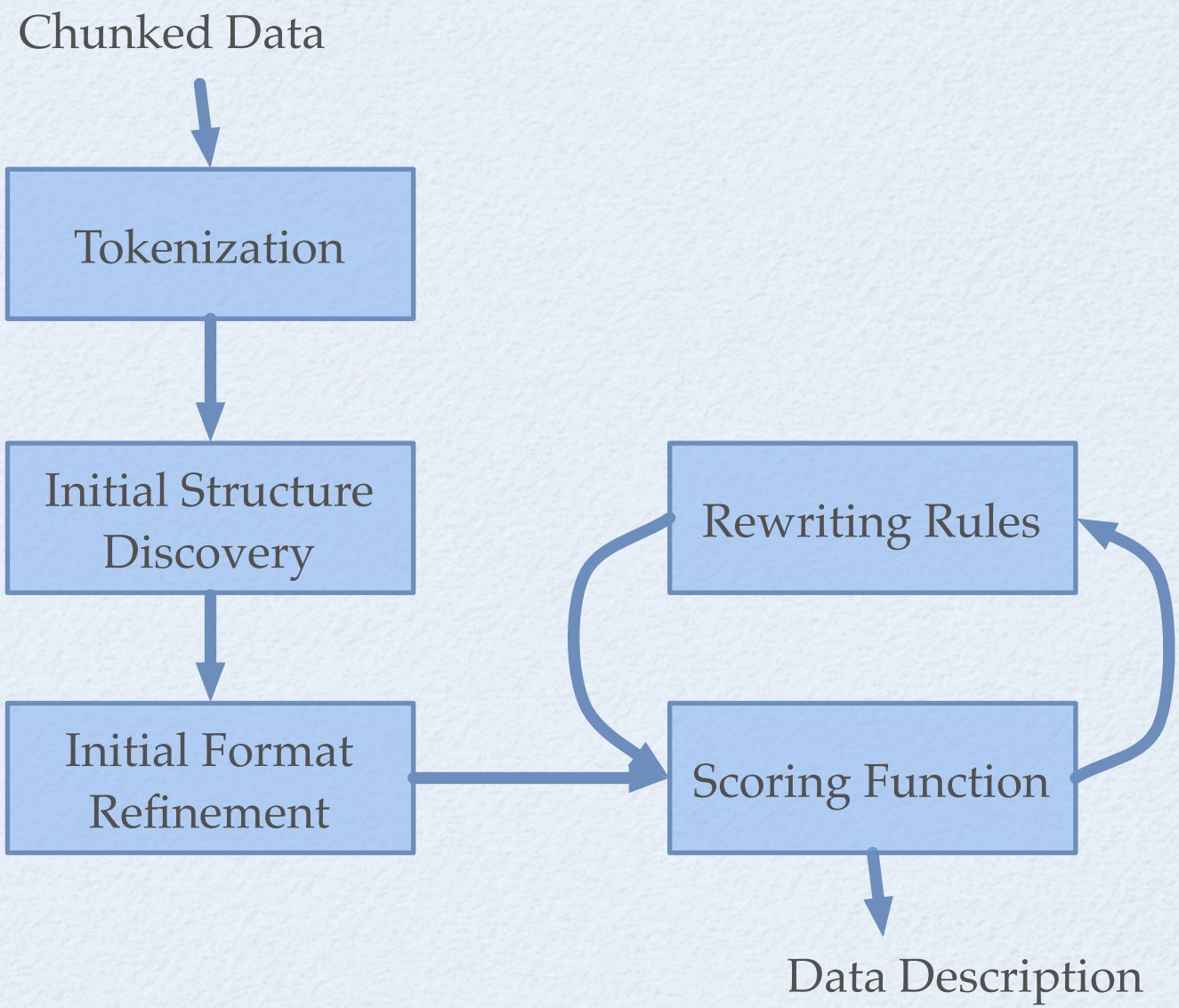
becomes

Context 3:

`String`

`String [] sep('|')`

Physical Type Inference



Related Work

- **Grammar Induction**
  - *Extracting Structure from Web Pages* [Arasu & Hector-Molena, SigMod, 2003].
  - *Language Identification in the Limit* [Gold, Information and Control, 1968].
  - *Grammatical Inference for Information Extraction and Visualization on the Web* [Hong, PhD Thesis, Imperial College, 2003].
  - *Current Trends in Grammatical Inference* [Higuera, LNCS, 2001].
- **Functional dependencies**
  - *Tane: An Efficient Algorithm for Discovering Functional and Approximate Dependencies* [Huhtal et al, Computer Journal, 1999].
- **Information Theory**
  - *Information Theory, Inference, and Learning Algorithms* [Mackay, Cambridge University Press, 2003].
  - *Advances in Minimum Description Length* [Grünwald, MIT Press, 2004].

CONCLUSIONS

- Types developed for *internal data* are relevant for *ad hoc data*.
- Other programming language ideas can facilitate ad hoc data management:
  - Type-directed programming, structural subtyping, type equality, program analysis, type inference...
- Format inference is possible for interesting and relevant formats.
- Meta-data are useful in describing errors detected during parsing.
  - How to support programming with meta-data?
  - Might meta-data be useful in other contexts?

THANK YOU

- David Walker
  Princeton
- Zach DeVito
  Princeton
- Kenny Zhu
  Princeton
- Yitzhak Mandelbaum
  AT&T
- Robert Gruber
  Google
- David Burke
  Galois
- Andy Adams-Moran
  Galois
- Alex Aiken
  Stanford
- Vikas Kedia
  Stanford

www.padsproj.org