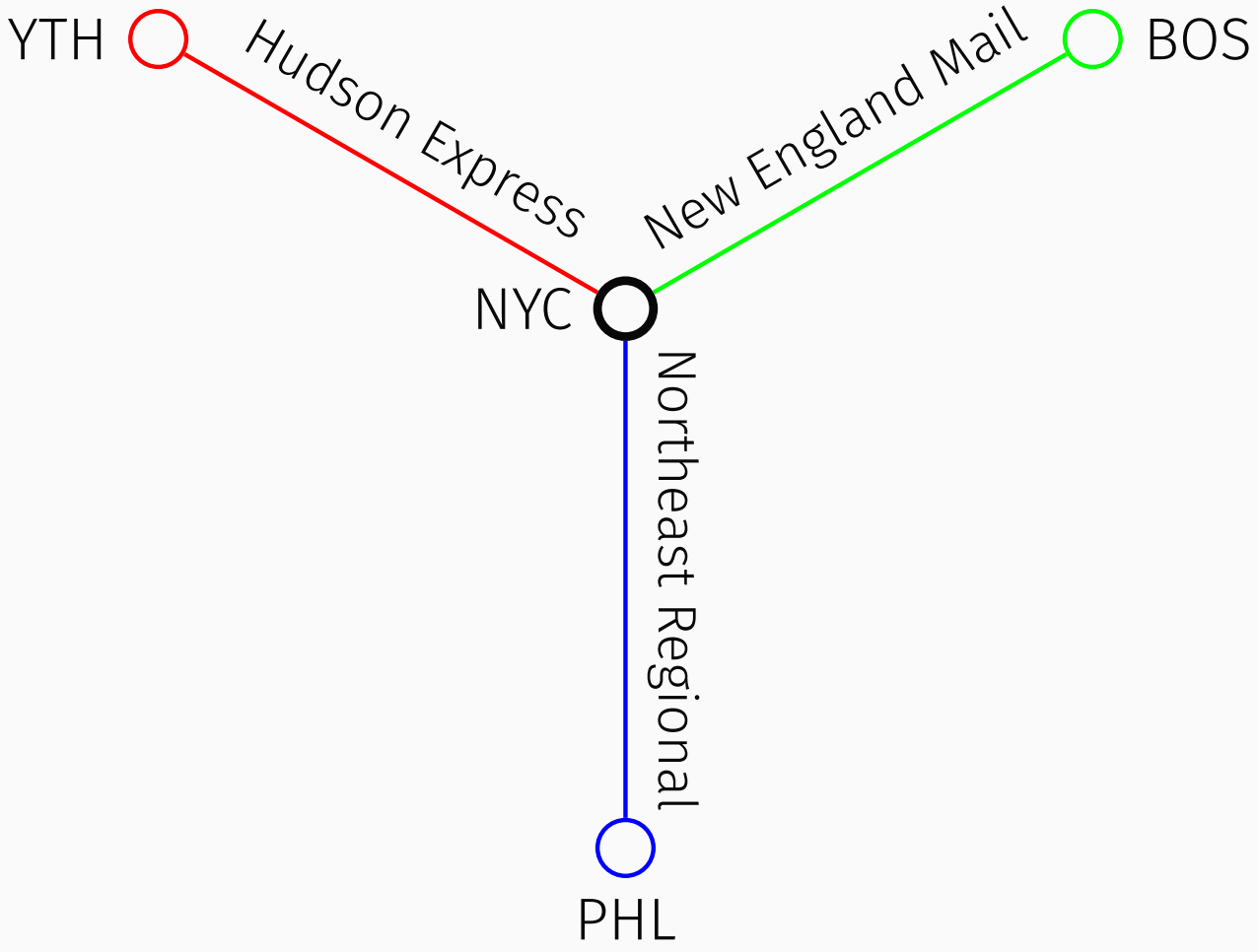
REGULAR PROGRAMMING FOR QUANTITATIVE PROPERTIES OF DATA STREAMS

Rajeev Alur Dana Fisman Mukund Raghothaman

IBM PL Day, 2015

University of Pennsylvania


YTH
Hudson Express
New England Mail
BOS
NYC
Northeast Regional
PHL

Time	Line	Station
06:00:00 AM	Hudson Express	YTH
06:27:18 AM	Northeast Regional	NYC
07:24:20 AM	Hudson Express	YTH
07:28:23 AM	Northeast Regional	PHL
08:12:46 AM	Hudson Express	YTH
09:08:54 AM	Northeast Regional	NYC
09:35:56 AM	New England Mail	NYC
09:43:10 AM	New England Mail	BOS
10:07:59 AM	New England Mail	BOS
10:45:41 AM	New England Mail	BOS
10:46:25 AM	Northeast Regional	PHL
11:32:52 AM	Northeast Regional	NYC
11:35:06 AM	Northeast Regional	PHL
11:58:00 AM	New England Mail	BOS
12:44:37 PM	Northeast Regional	PHL
12:55:55 PM	Hudson Express	YTH
01:16:32 PM	Hudson Express	NYC
01:41:55 PM	Northeast Regional	PHL
01:49:32 PM	Hudson Express	YTH
02:21:46 PM	Hudson Express	YTH
03:17:45 PM	Hudson Express	YTH
03:57:49 PM	Hudson Express	NYC
04:30:56 PM	Northeast Regional	PHL
04:58:19 PM	New England Mail	NYC
05:34:51 PM	New England Mail	BOS
06:13:03 PM	New England Mail	NYC
06:32:11 PM	Northeast Regional	NYC
07:08:09 PM	Hudson Express	YTH

THE QUERY

“What is the average travel time from PHL to YTH?”

Time	Line	Station
06:00:00 AM	Hudson Express	YTH
06:27:18 AM	Northeast Regional	NYC
07:24:20 AM	Hudson Express	YTH
07:28:23 AM	Northeast Regional	PHL
08:12:46 AM	Hudson Express	YTH
09:08:54 AM	Northeast Regional	NYC
09:35:56 AM	New England Mail	NYC
09:43:10 AM	New England Mail	BOS
10:07:59 AM	New England Mail	BOS
10:45:41 AM	New England Mail	BOS
10:46:25 AM	Northeast Regional	PHL
11:32:52 AM	Northeast Regional	NYC
11:35:06 AM	Northeast Regional	PHL
11:58:00 AM	New England Mail	BOS
12:44:37 PM	Northeast Regional	PHL
12:55:55 PM	Hudson Express	YTH
01:16:32 PM	Hudson Express	NYC
01:41:55 PM	Northeast Regional	PHL
01:49:32 PM	Hudson Express	YTH
02:21:46 PM	Hudson Express	YTH
03:17:45 PM	Hudson Express	YTH
03:57:49 PM	Hudson Express	NYC
04:30:56 PM	Northeast Regional	PHL
04:58:19 PM	New England Mail	NYC
05:34:51 PM	New England Mail	BOS
06:13:03 PM	New England Mail	NYC
06:32:11 PM	Northeast Regional	NYC
07:08:09 PM	Hudson Express	YTH

Time	Line	Station
06:00:00 AM	Hudson Express	YTH
06:27:18 AM	Northeast Regional	NYC
07:24:20 AM	Hudson Express	YTH
07:28:23 AM	Northeast Regional	PHL
08:12:46 AM	Hudson Express	YTH
09:08:54 AM	Northeast Regional	NYC
09:35:56 AM	New England Mail	NYC
09:43:10 AM	New England Mail	BOS
10:07:59 AM	New England Mail	BOS
10:45:41 AM	New England Mail	BOS
10:46:25 AM	Northeast Regional	PHL
11:32:52 AM	Northeast Regional	NYC
11:35:06 AM	Northeast Regional	PHL
11:58:00 AM	New England Mail	BOS
12:44:37 PM	Northeast Regional	PHL
12:55:55 PM	Hudson Express	YTH
01:16:32 PM	Hudson Express	NYC
01:41:55 PM	Northeast Regional	PHL
01:49:32 PM	Hudson Express	YTH
02:21:46 PM	Hudson Express	YTH
03:17:45 PM	Hudson Express	YTH
03:57:49 PM	Hudson Express	NYC
04:30:56 PM	Northeast Regional	PHL
04:58:19 PM	New England Mail	NYC
05:34:51 PM	New England Mail	BOS
06:13:03 PM	New England Mail	NYC
06:32:11 PM	Northeast Regional	NYC
07:08:09 PM	Hudson Express	YTH

Time	Line	Station
06:00:00 AM	Hudson Express	YTH
06:27:18 AM	Northeast Regional	NYC
07:24:20 AM	Hudson Express	YTH
07:28:23 AM	Northeast Regional	PHL
08:12:46 AM	Hudson Express	YTH
09:08:54 AM	Northeast Regional	NYC
09:35:56 AM	New England Mail	NYC
09:43:10 AM	New England Mail	BOS
10:07:59 AM	New England Mail	BOS
10:45:41 AM	New England Mail	BOS
10:46:25 AM	Northeast Regional	PHL
11:32:52 AM	Northeast Regional	NYC
11:35:06 AM	Northeast Regional	PHL
11:58:00 AM	New England Mail	BOS
12:44:37 PM	Northeast Regional	PHL
12:55:55 PM	Hudson Express	YTH
01:16:32 PM	Hudson Express	NYC
01:41:55 PM	Northeast Regional	PHL
01:49:32 PM	Hudson Express	YTH
02:21:46 PM	Hudson Express	YTH
03:17:45 PM	Hudson Express	YTH
03:57:49 PM	Hudson Express	NYC
04:30:56 PM	Northeast Regional	PHL
04:58:19 PM	New England Mail	NYC
05:34:51 PM	New England Mail	BOS
06:13:03 PM	New England Mail	NYC
06:32:11 PM	Northeast Regional	NYC
07:08:09 PM	Hudson Express	YTH

Time	Line	Station
06:00:00 AM	Hudson Express	YTH
06:27:18 AM	Northeast Regional	NYC
07:24:20 AM	Hudson Express	YTH
07:28:23 AM	Northeast Regional	PHL
08:12:46 AM	Hudson Express	YTH
09:08:54 AM	Northeast Regional	NYC
09:35:56 AM	New England Mail	NYC
09:43:10 AM	New England Mail	BOS
10:07:59 AM	New England Mail	BOS
10:45:41 AM	New England Mail	BOS
10:46:25 AM	Northeast Regional	PHL
11:32:52 AM	Northeast Regional	NYC
11:35:06 AM	Northeast Regional	PHL
11:58:00 AM	New England Mail	BOS
12:44:37 PM	Northeast Regional	PHL
12:55:55 PM	Hudson Express	YTH
01:16:32 PM	Hudson Express	NYC
01:41:55 PM	Northeast Regional	PHL
01:49:32 PM	Hudson Express	YTH
02:21:46 PM	Hudson Express	YTH
03:17:45 PM	Hudson Express	YTH
03:57:49 PM	Hudson Express	NYC
04:30:56 PM	Northeast Regional	PHL
04:58:19 PM	New England Mail	NYC
05:34:51 PM	New England Mail	BOS
06:13:03 PM	New England Mail	NYC
06:32:11 PM	Northeast Regional	NYC
07:08:09 PM	Hudson Express	YTH

"WHAT IS THE AVERAGE TRAVEL TIME FROM `phl` TO `yth`?"

- Function f_{trip} calculates travel time for a single trip

- $f_{avg} = \mathit{iter\text{-}avg}(f_{trip})$

"What is the average travel time from `phl` to `yth`?"

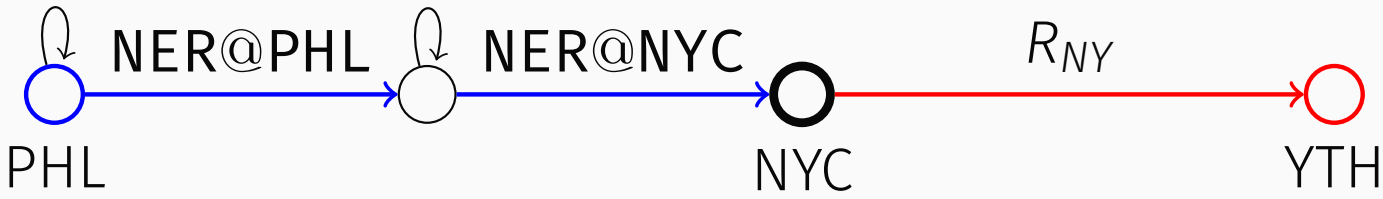
- Function f_{trip} calculates travel time for a single trip
  $f_{trip} = t_{alight} - t_{board}$

- $f_{avg} = \mathit{iter\text{-}avg}(f_{trip})$

"WHAT IS THE AVERAGE TRAVEL TIME FROM `phl` TO `yth`?"

- Function f_{trip} calculates travel time for a single trip

  $f_{trip} = t_{alight} - t_{board}$

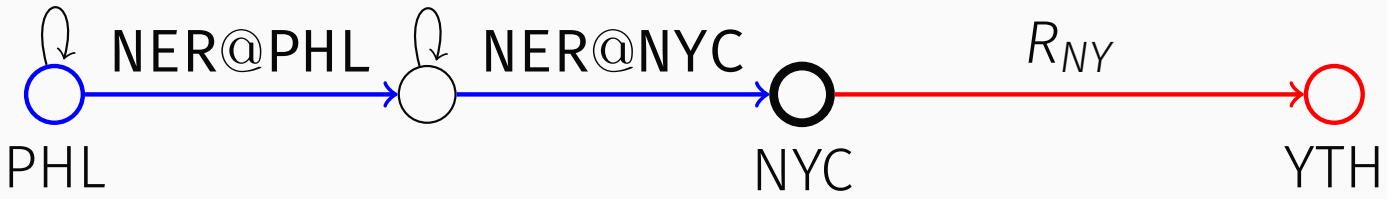


- $f_{avg} = \textit{iter-avg}(f_{trip})$

"WHAT IS THE AVERAGE TRAVEL TIME FROM phl TO yth?"

- Function f_{trip} calculates travel time for a single trip

  $f_{trip} = t_{alight} - t_{board}$



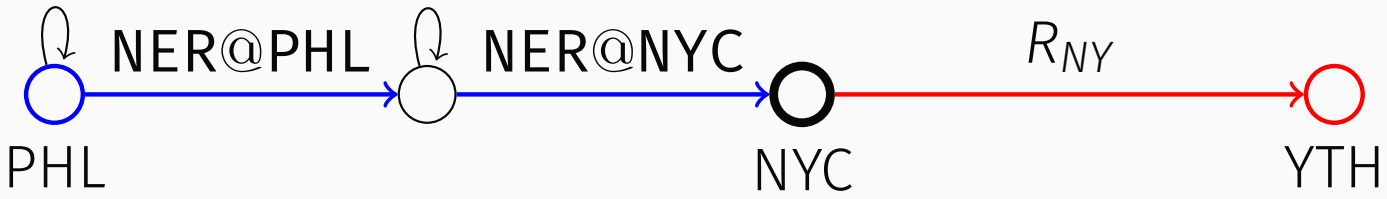
  $t_{board} =$ $\text{NER@PHL} \mapsto time$

- $f_{avg} = iter\text{-}avg(f_{trip})$

"WHAT IS THE AVERAGE TRAVEL TIME FROM phl TO yth?"

- Function f_{trip} calculates travel time for a single trip

  $f_{trip} = t_{alight} - t_{board}$



  $t_{board} = split((\neg \mathsf{NER@PHL})^*, \mathsf{NER@PHL} \mapsto time,$
  $(\neg \mathsf{NER@NYC})^*, \mathsf{NER@NYC}, R_{NY})$

- $f_{avg} = iter\text{-}avg(f_{trip})$

INTRODUCING DREX

- Language based on the idea of function combinators

INTRODUCING DREX

- Language based on the idea of function combinators
- DReX is a simple, expressive programming model for stream processing, with strong theoretical foundations and fast evaluation algorithms

Small collection of core combinators

Basic functions: $data \mapsto cost$

Conditional choice: f *else* g

Concatenation: *split-plus*(f, g), *split-min*(f, g), *split-left*(f, g), *split-right*(f, g), ...

Function iteration: *iter-plus*(f), *iter-max*(f), *iter-avg*(f), *iter-mdn*(f), ...

Cost operations: $f_1 + f_2$, $f_1 - f_2$, $\max(f_1, f_2)$, ...

Small collection of core combinators

Basic functions: $data \mapsto cost$

Conditional choice: f *else* g

Concatenation: *split-plus*(f, g), *split-min*(f, g), *split-left*(f, g), *split-right*(f, g), ...

Function iteration: *iter-plus*(f), *iter-max*(f), *iter-avg*(f), *iter-mdn*(f), ...

Cost operations: $f_1 + f_2$, $f_1 - f_2$, $\max(f_1, f_2)$, ...

Small collection of core combinators

Basic functions: $data \mapsto cost$

Conditional choice: f *else* g

Concatenation: *split-plus*(f, g), *split-min*(f, g), *split-left*(f, g), *split-right*(f, g), ...

Function iteration: *iter-plus*(f), *iter-max*(f), *iter-avg*(f), *iter-mdn*(f), ...

Cost operations: $op(f_1, f_2, \ldots, f_k)$

Small collection of core combinators

Basic functions: $data \mapsto cost$

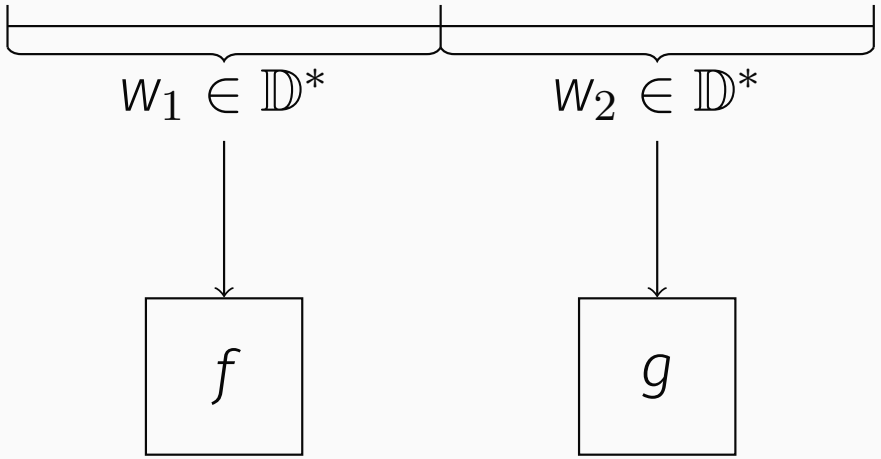
Conditional choice: f *else* g

Concatenation: *split-plus*(f, g), *split-min*(f, g), *split-left*(f, g), *split-right*(f, g), ...

Function iteration: *iter-plus*(f), *iter-max*(f), *iter-avg*(f), *iter-mdn*(f), ...

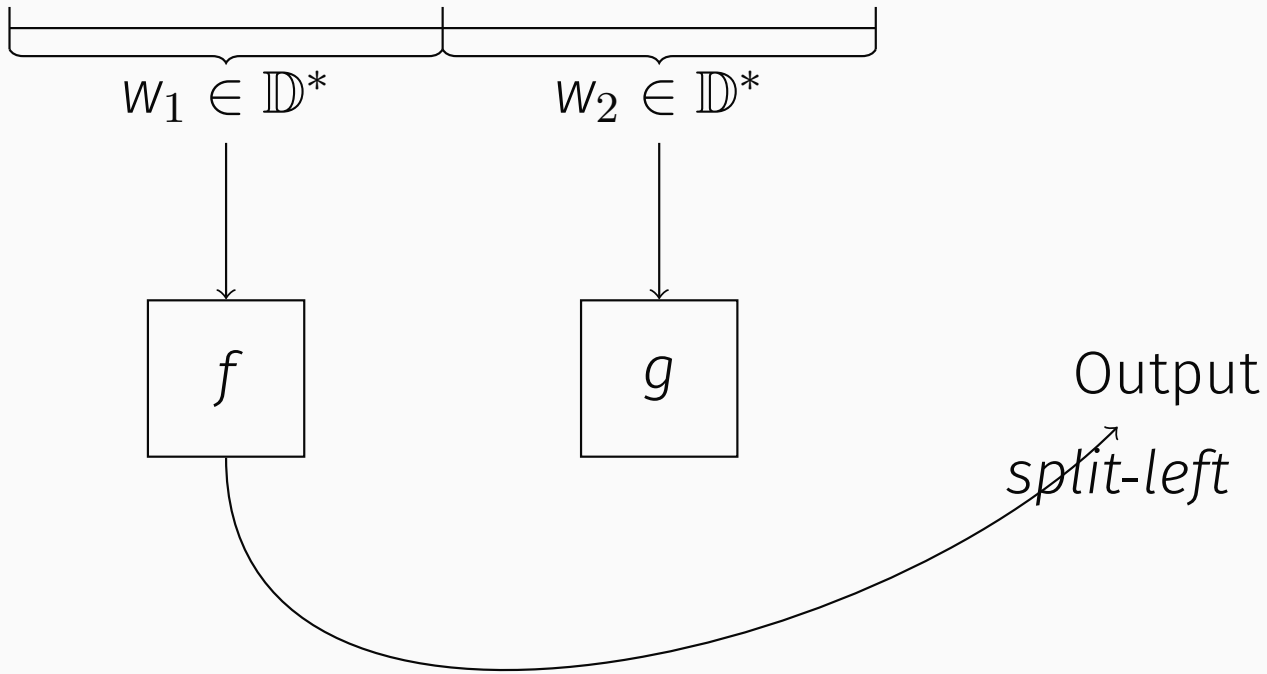
Cost operations: $op(f_1, f_2, \ldots, f_k)$

What about concatenation?

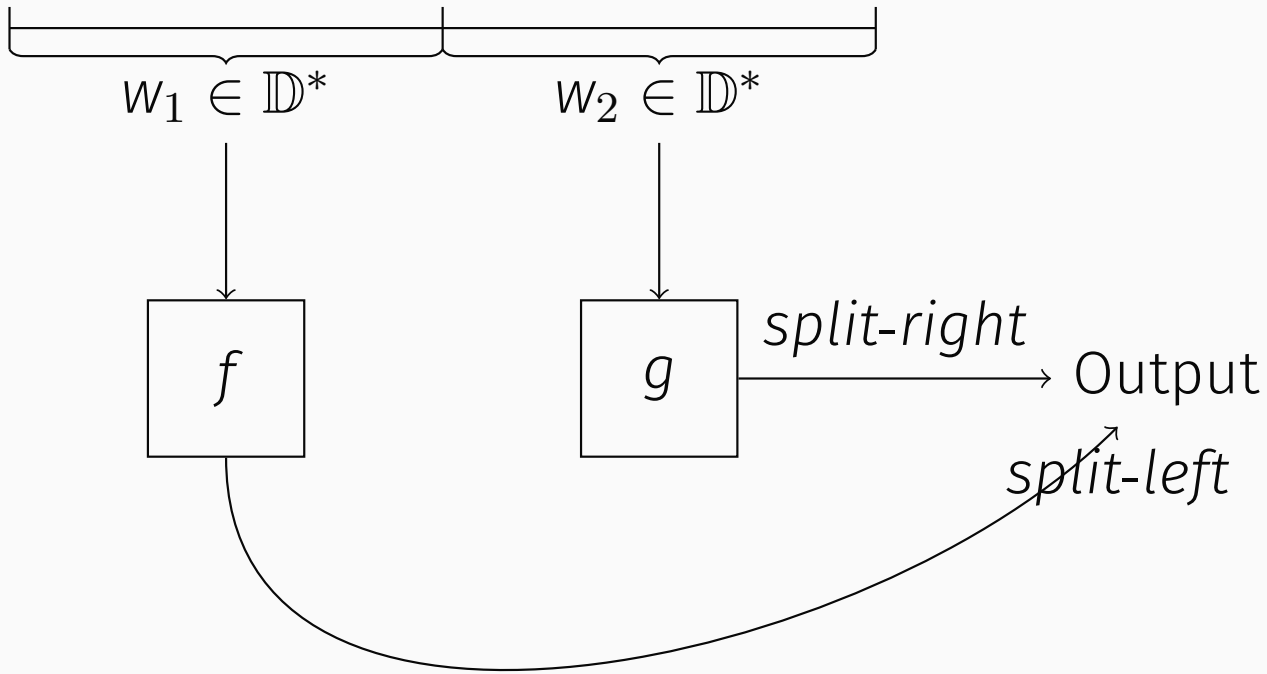


Output

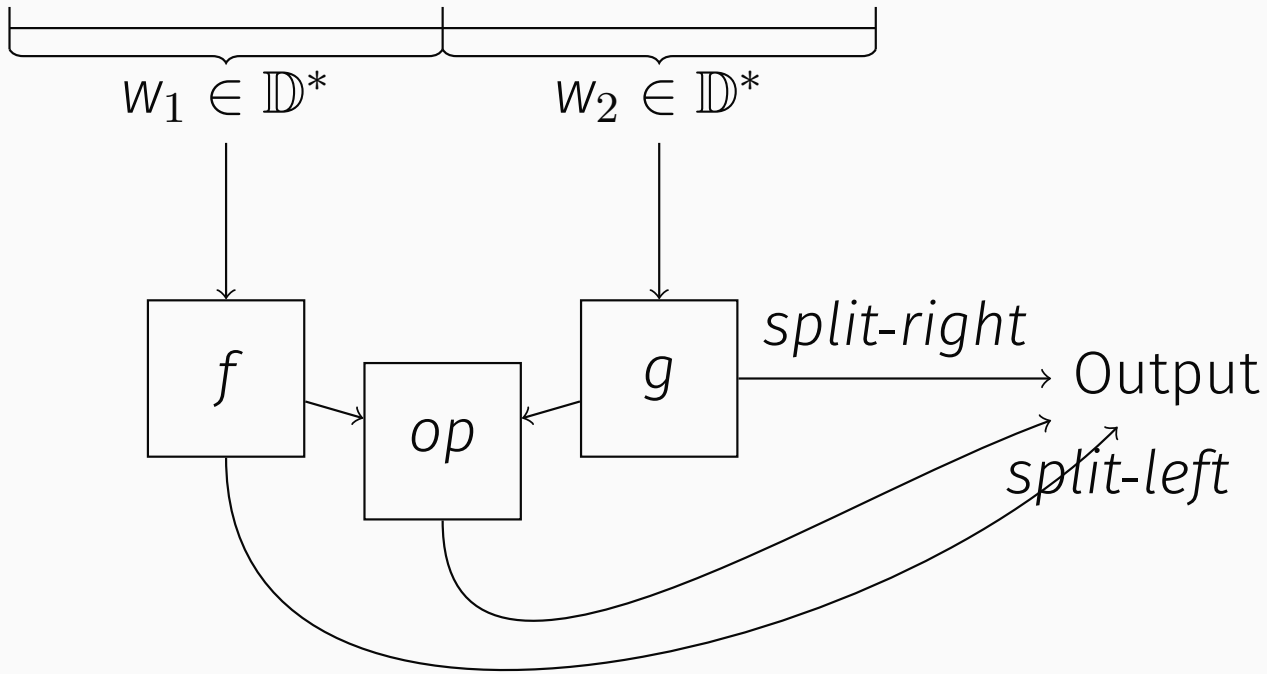
What about concatenation?



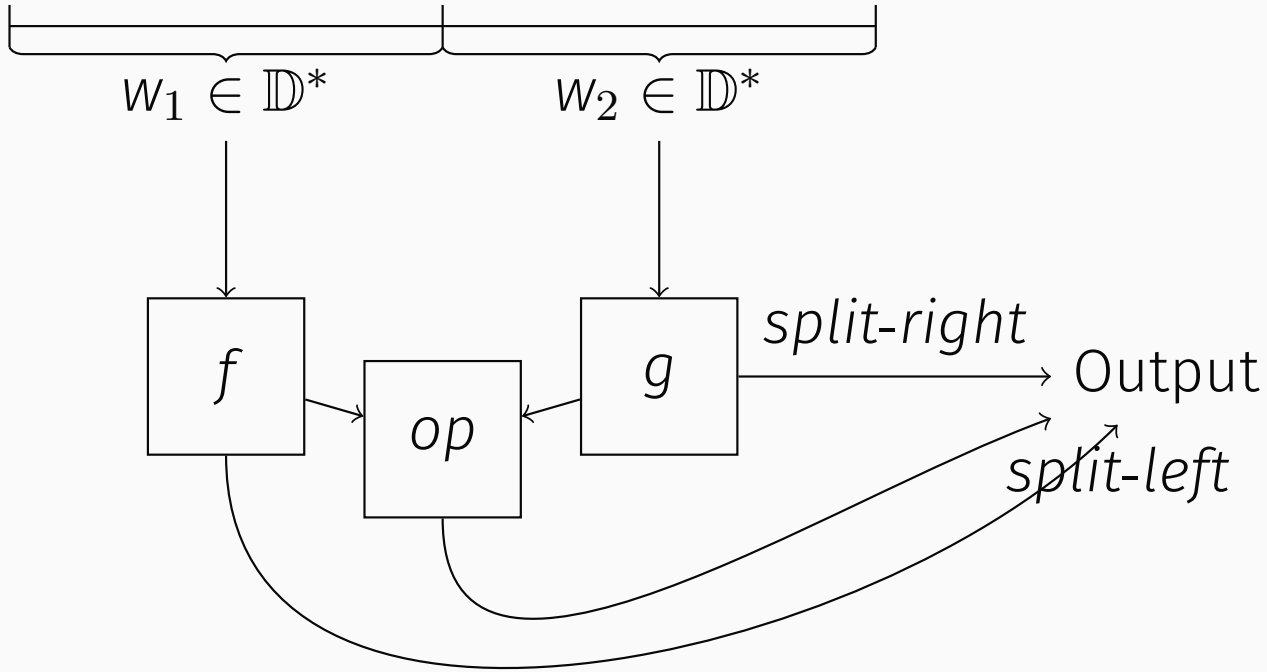
What about concatenation?



What about concatenation?

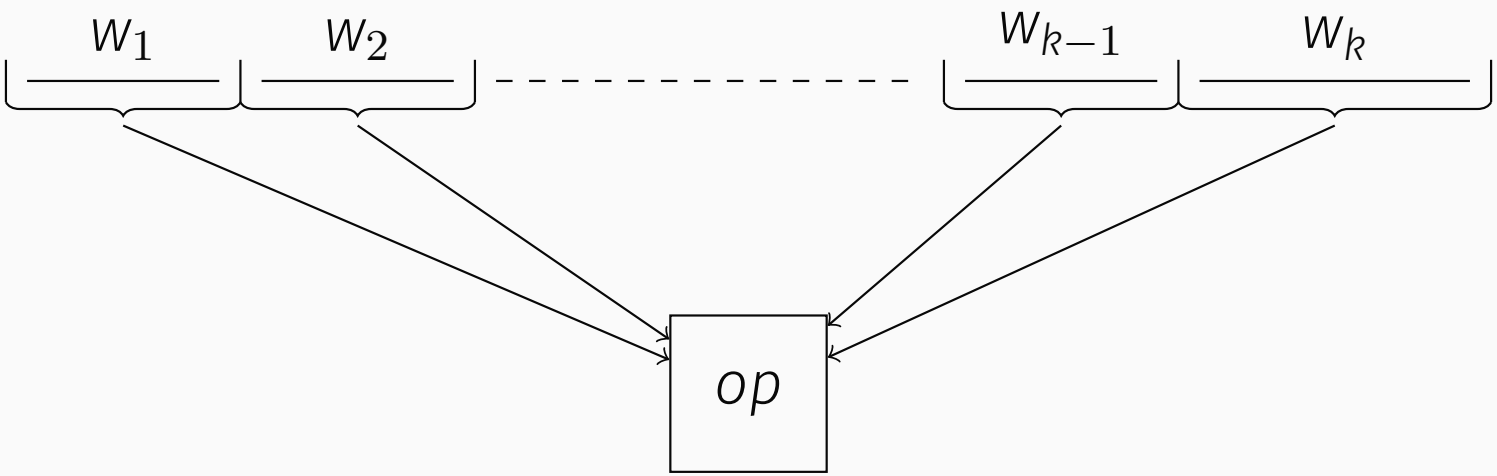


What about concatenation?

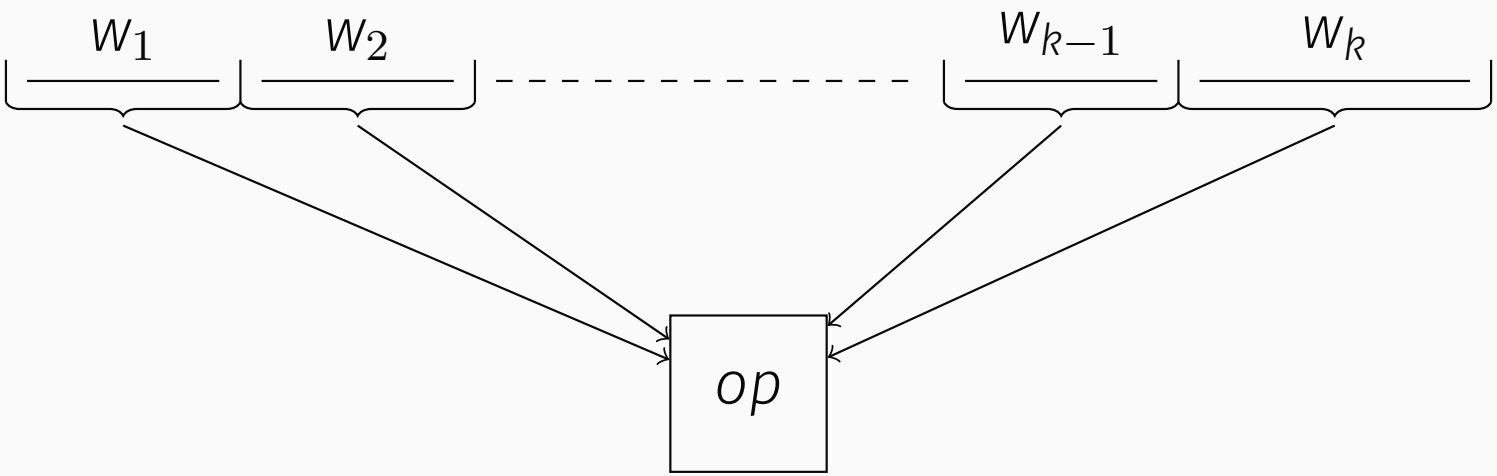


Are these combinators "sufficient"?

What about iteration?

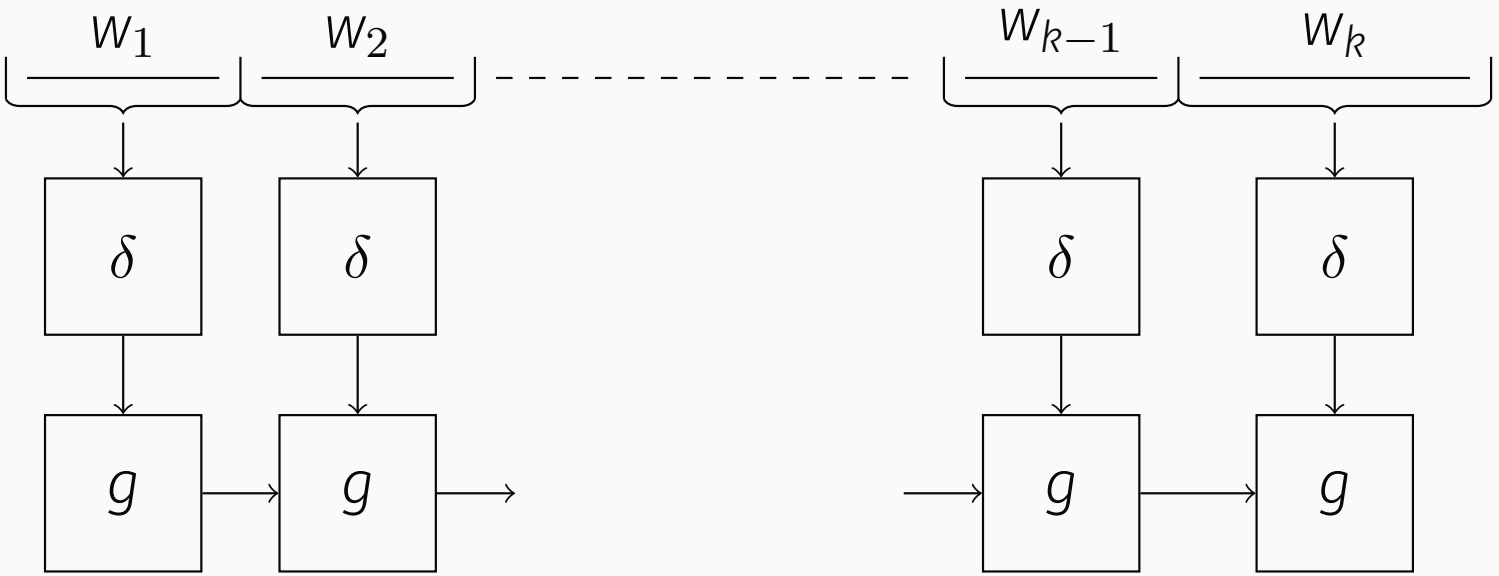


What about iteration?



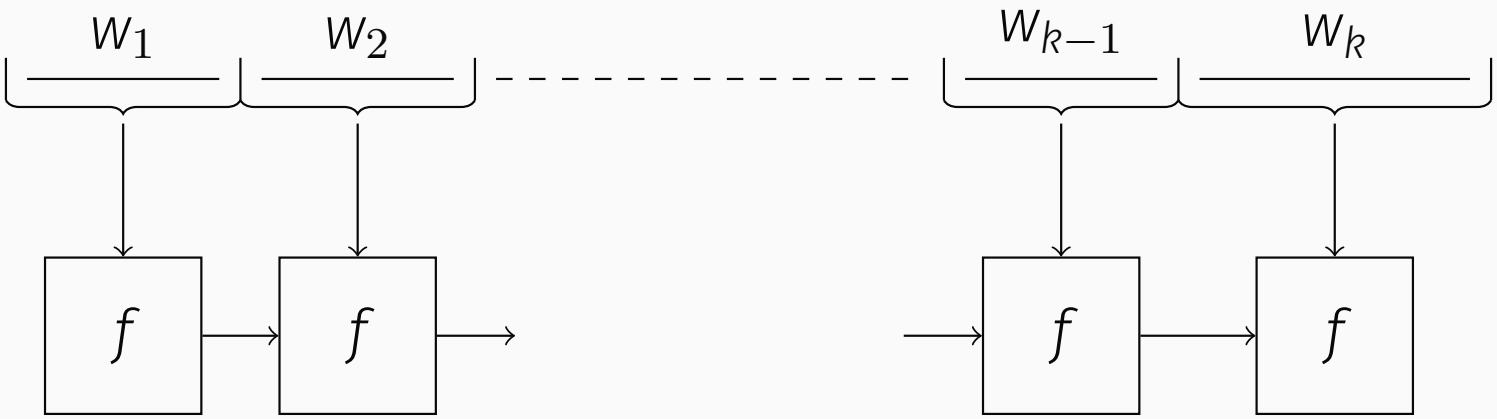
- Data stream is a sequence of bank transactions
- Interest applied at end of each month

What about iteration?



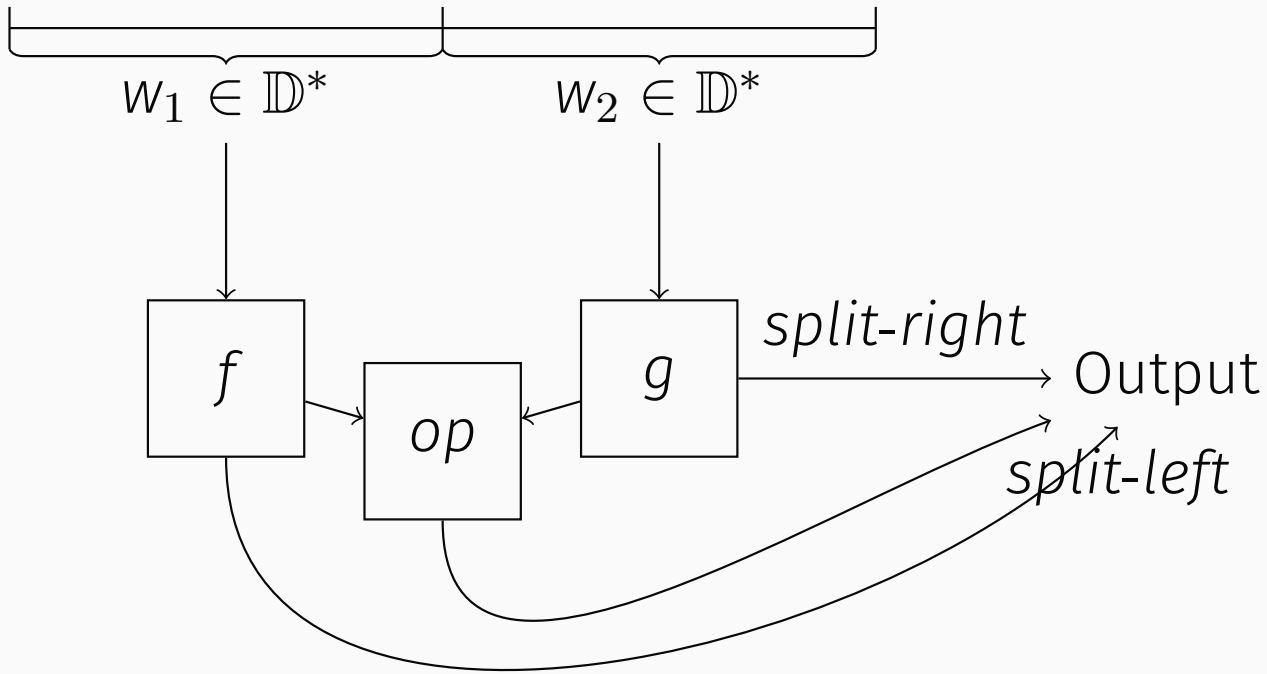
- Data stream is a sequence of bank transactions
- Interest applied at end of each month

What about iteration?



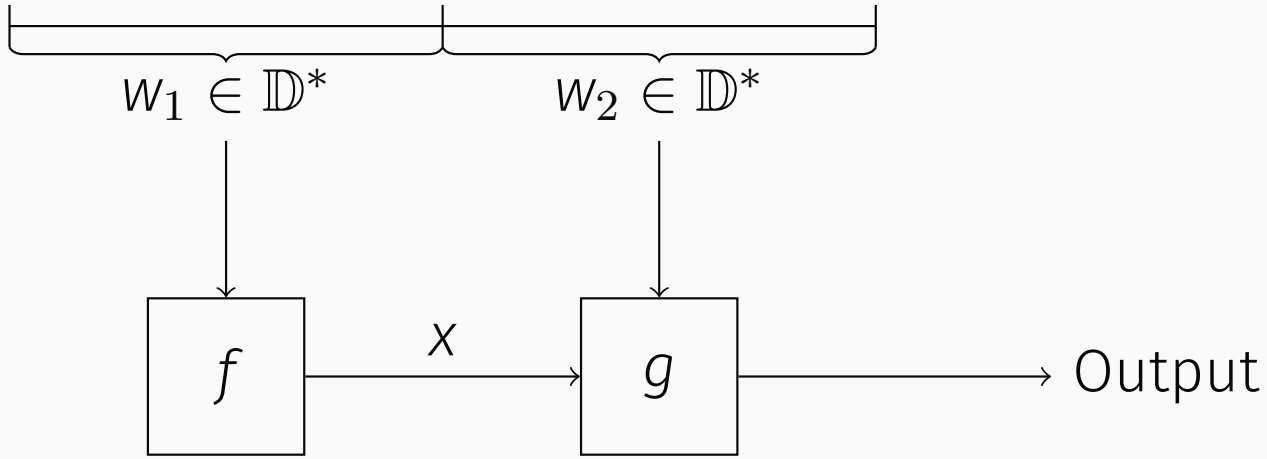
- Data stream is a sequence of bank transactions
- Interest applied at end of each month
- Map input streams to terms over the cost domain

What about concatenation?



Are these combinators "sufficient"?

What about concatenation?



Are these combinators "sufficient"?

Small collection of core combinators

Basic functions: $data \mapsto cost$

Conditional choice: f *else* g

Concatenation: *split-plus*(f, g), *split-min*(f, g), *split-left*(f, g), *split-right*(f, g), ...

Function iteration: *iter-plus*(f), *iter-max*(f), *iter-avg*(f), *iter-mdn*(f), ...

Cost operations: $op(f_1, f_2, \ldots, f_k)$

Small collection of core combinators

Basic functions: $data \mapsto cost$, $regex \mapsto term$

Conditional choice: $f\ else\ g$

Concatenation: $split(f \rightarrow^{x} g)$, $split(f \leftarrow^{x} g)$

Function iteration: $iter^{\rightarrow}(f)$, $iter^{\leftarrow}(f)$

Cost operations: $op(f_1, f_2, \ldots, f_k)$, $f[x/g]$

Small collection of core combinators

Basic functions: $data \mapsto cost$, $regex \mapsto term$

Conditional choice: $f\ else\ g$

Concatenation: $split(f \rightarrow^{x} g)$, $split(f \leftarrow^{x} g)$

Function iteration: $iter^{\rightarrow}(f)$, $iter^{\leftarrow}(f)$

Cost operations: $op(f_1, f_2, \ldots, f_k)$, $f[x/g]$

Structural operators decoupled from cost operators!

Function descriptions are modular

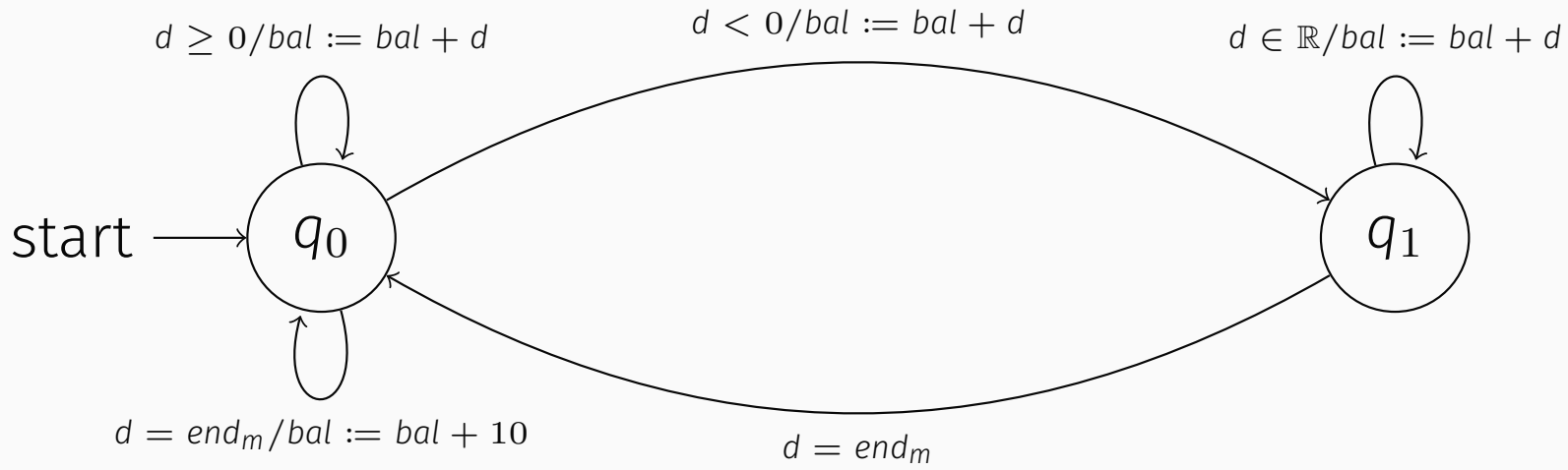
```
currLoc: {phl, nyc, yth, ...}
totalTime: int, tripCount: int
tBoard: int
while exists next event e:
  if e is (ner, phl) and currLoc = phl:
    tBoard := e.time
    currLoc := ...
  elif ...:
    ...
```

DReX allows regular parsing of the input data stream

- Unrestricted global choice: *f else g*
- Iteration patterns part of query, not of the data: *iter-avg*(f_{trip})

- Expressively equivalent to regular string-to-term transformations
- Multiple characterizations: MSO-definable graph transformations, streaming string-to-term transducers, two-way finite state transducers
- Closed under various operations: regular look-ahead, input reversal, function composition etc.

Streaming string-to-term transducers

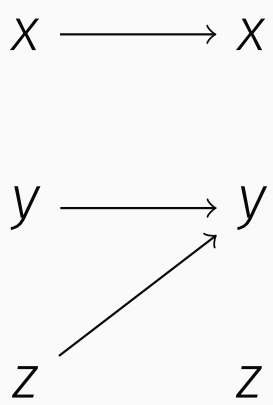


Taste of the completeness proof

- Piggy-back on DFA to regular expression translation
  Construct $R^i(q, q')$: all strings from q to q' while only traversing states less than q_i
- Construct $f^i(q, q', x)$: express final value of register x as a DReX expression

Taste of the completeness proof

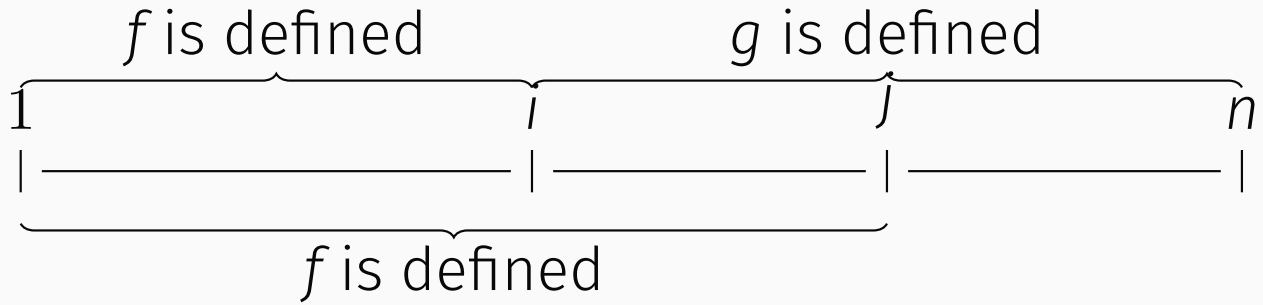
- Capture data flows using "shapes"
- Construct a partial order over shapes, and use as basis for induction

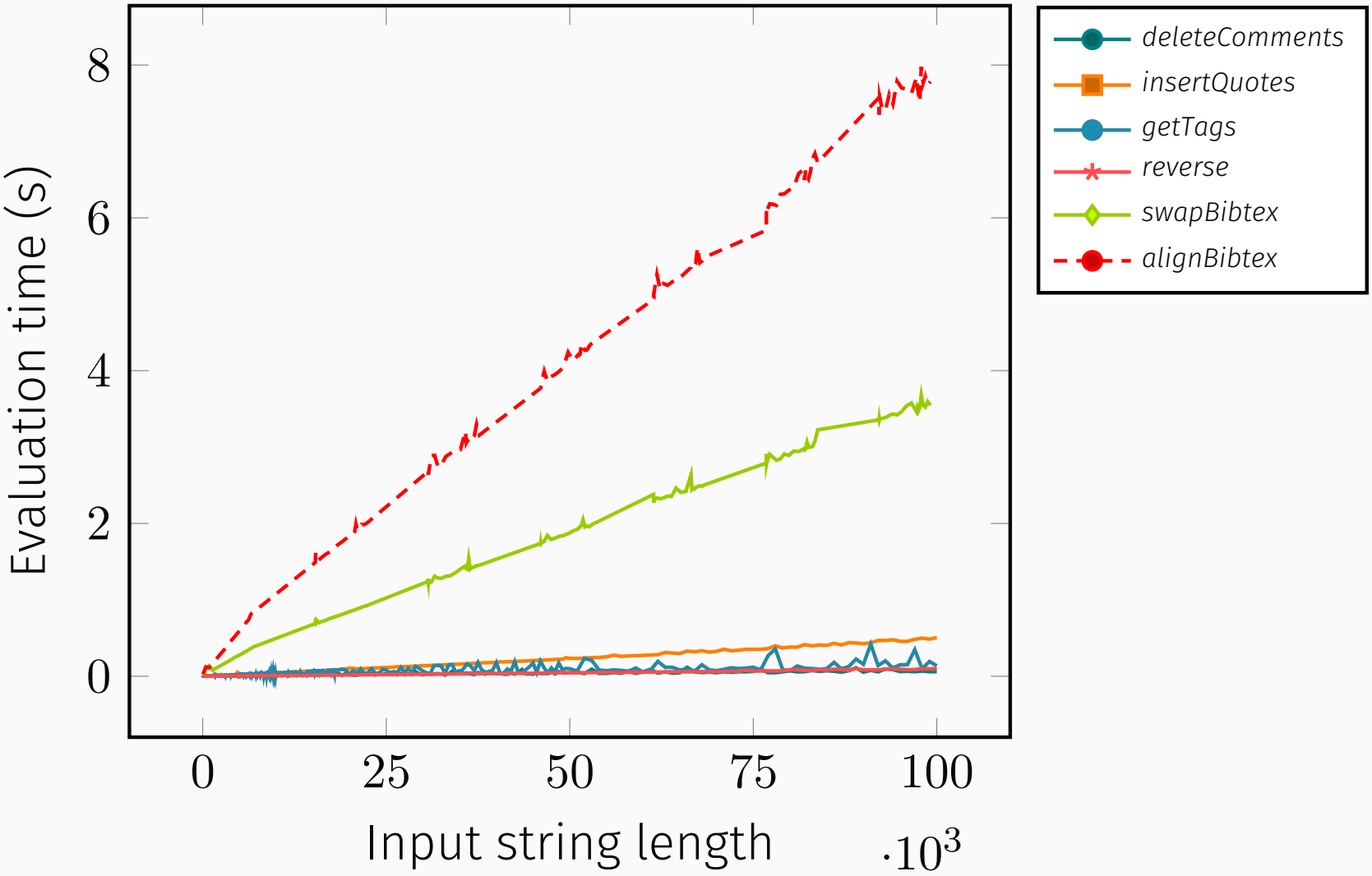


One-pass evaluation algorithm for unambiguous expressions

- $f(w)$ can be computed in time $O(|w| \cdot poly(|f|))$
- Separating intent from evaluation
- Space-efficient evaluation where possible
  $*$, $+$, $-$, min, max, *avg*
- Approximation algorithms for the masses: *iter-median*

- Require that expressions be "unambiguous"
- Informally, restricts number of alternative parse trees
- Not burdensome




Evaluation time (s)
0
2
4
6
8
0
25
50
75
100
Input string length
$\cdot 10^3$
deleteComments
insertQuotes
getTags
reverse
swapBibtex
alignBibtex

STATUS

Done: Expressiveness theorems and evaluation algorithms
Alur, Freilich, R, LICS 2014
Alur, D'Antoni, R, POPL 2015
Alur, Fisman, R, Unpublished 2015

Ongoing: Approximation algorithms, case studies, static analysis

THANK YOU!

TRY IT OUT AT HTTP://DREXONLINE.COM