

Dataflow Networks

FOCS, Fall 2020

Streaming models

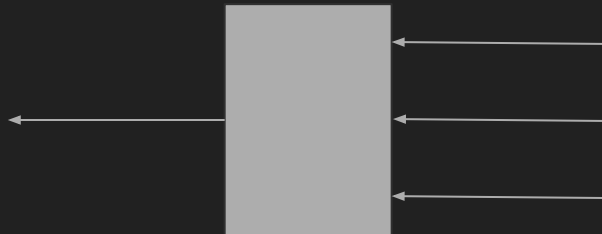
Working with infinitely streaming data

- multiple input streams
- single output stream

Process and create output stream as input comes in

- Ideally don't buffer

What goes in the box?



Dataflow networks

Dataflow networks take streams of values as inputs and produce streams of values as outputs

- type of values depends on the kind of network developed
 - floating point for approximation algorithms
 - images for streaming movies
- sequential components connected by buffered communication channels
- model assumes an underlying sequential language

Primitive components

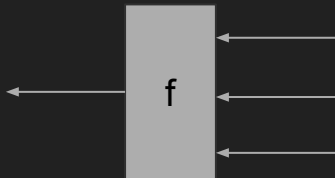
Constant k : produces an infinite stream of k



Primitive components

map f: transforms one or more streams by applying f to the inputs

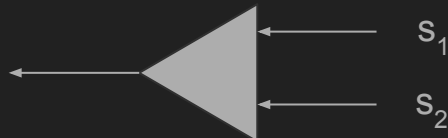
- blocks until all input streams have at least one value)
- transformation f written in underlying sequential language
- transformation f holds no state



Primitive components

followed by : produces a stream from the first element of s_1 followed by everything from s_2

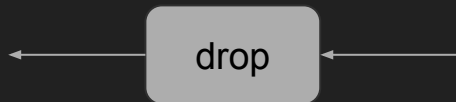
- blocks until an element of s_1 arrives
- then simply forwards values that arrive on s_2



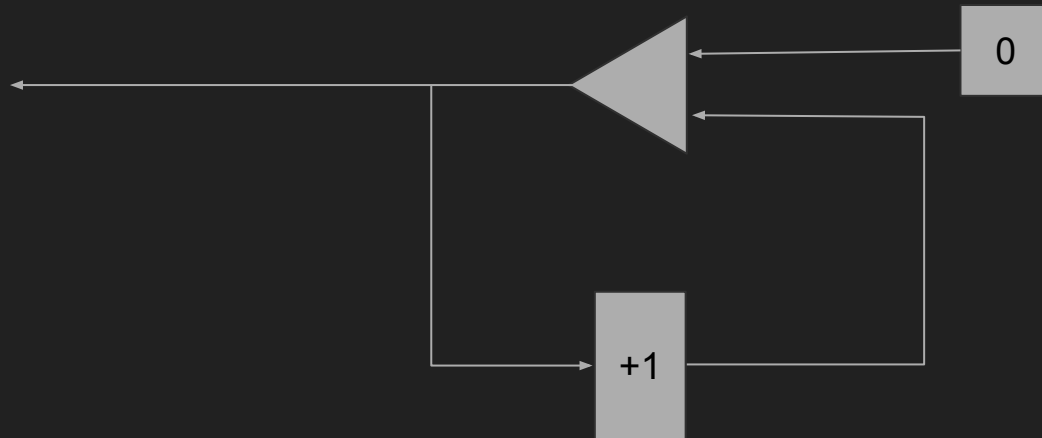
Primitive components

drop : produces a stream from the input stream by "dropping" the first element of the stream

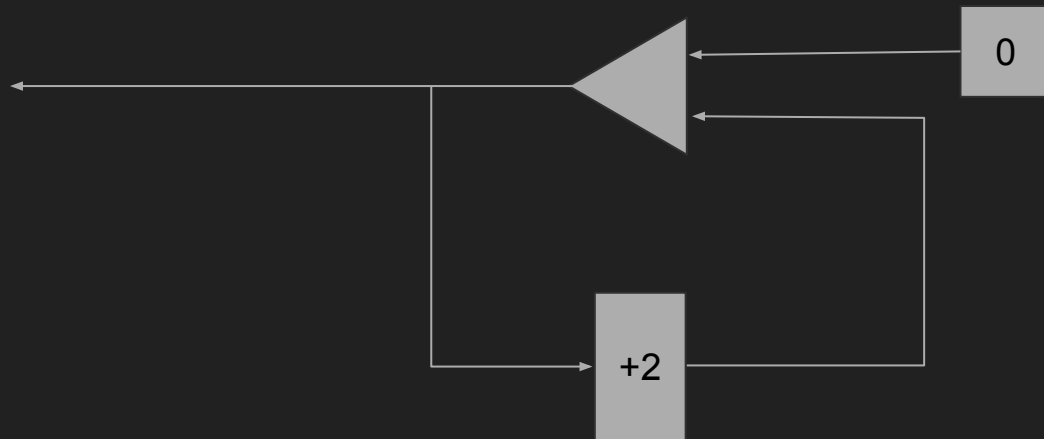
- input *a b c d e f ...* output *b c d e f ...*
- discards the first element that arrives (produce no output)
- then simply forwards everything that arrives to its output



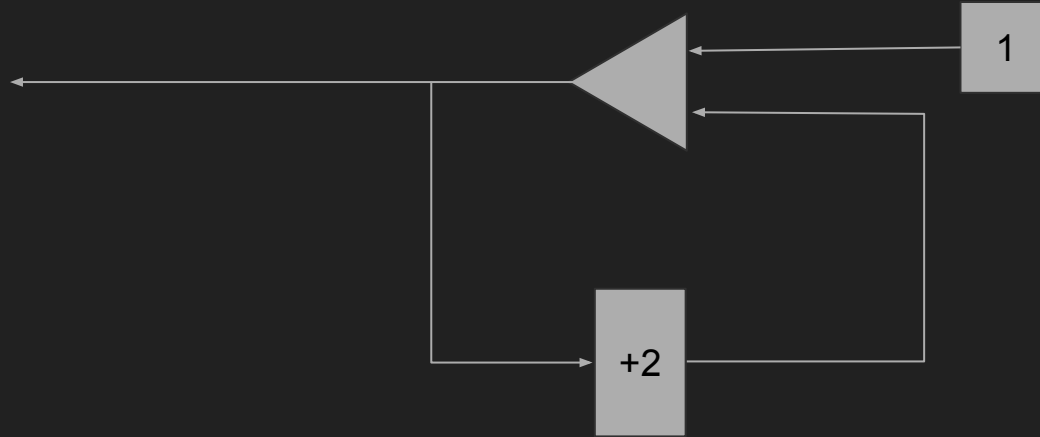
Sequences: nats



Sequences: evens



Sequences: odds



Sequences: odds



Sequences: triangular numbers

Want to create 0, 1, 3, 6, 10, 15, 21, ...

Observe:

$$0 + 1 = 1$$

$$1 + 2 = 3$$

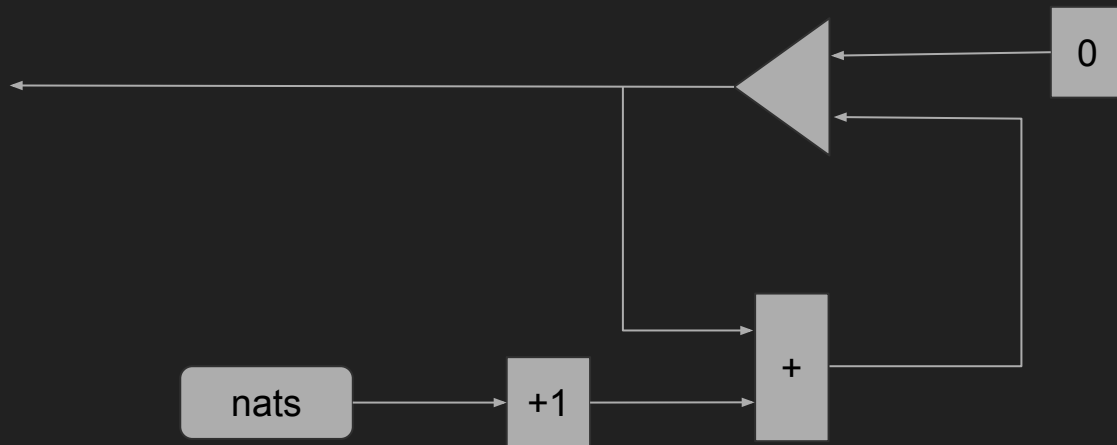
$$3 + 3 = 6$$

$$6 + 4 = 10$$

$$10 + 5 = 15$$

$$15 + 6 = 21$$

...



Sequences: square numbers

Want to create 0, 1, 4, 9, 16, 25, 36, ...

Observe:

$$0 + 1 = 1$$

$$1 + 3 = 4$$

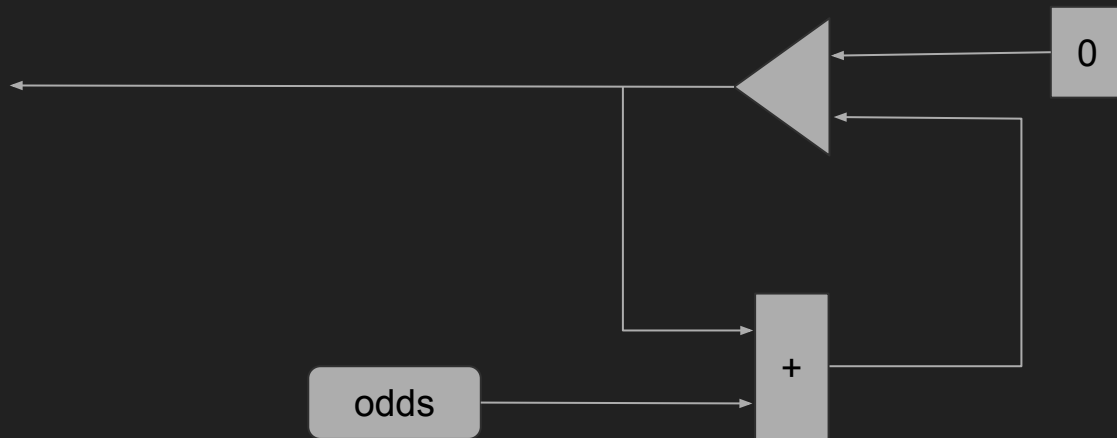
$$4 + 5 = 9$$

$$9 + 7 = 16$$

$$16 + 9 = 25$$

$$25 + 11 = 36$$

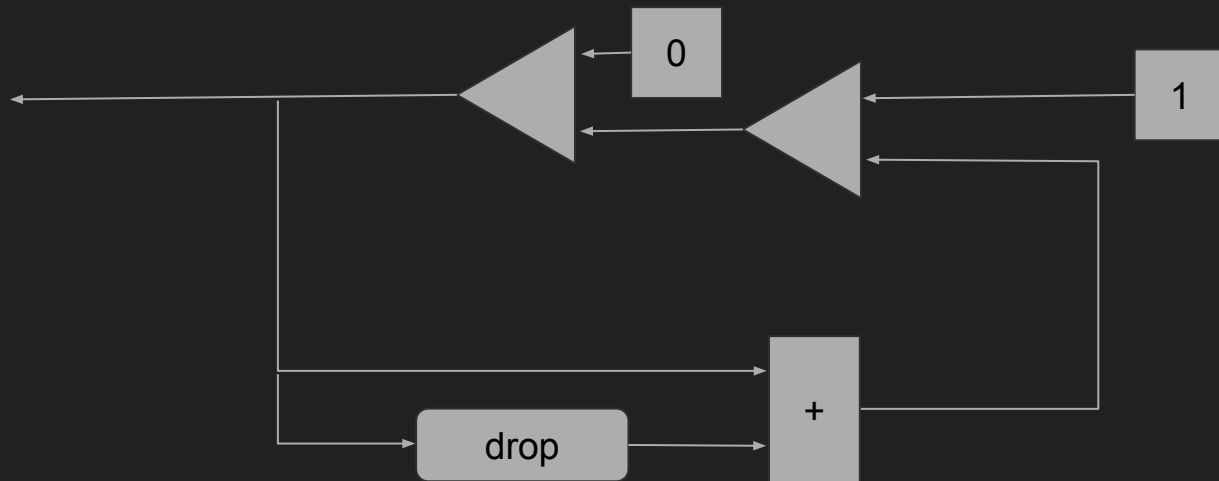
...



Sequences: Fibonacci numbers

Want to create 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

Each number in the sequence is the sum of the previous two

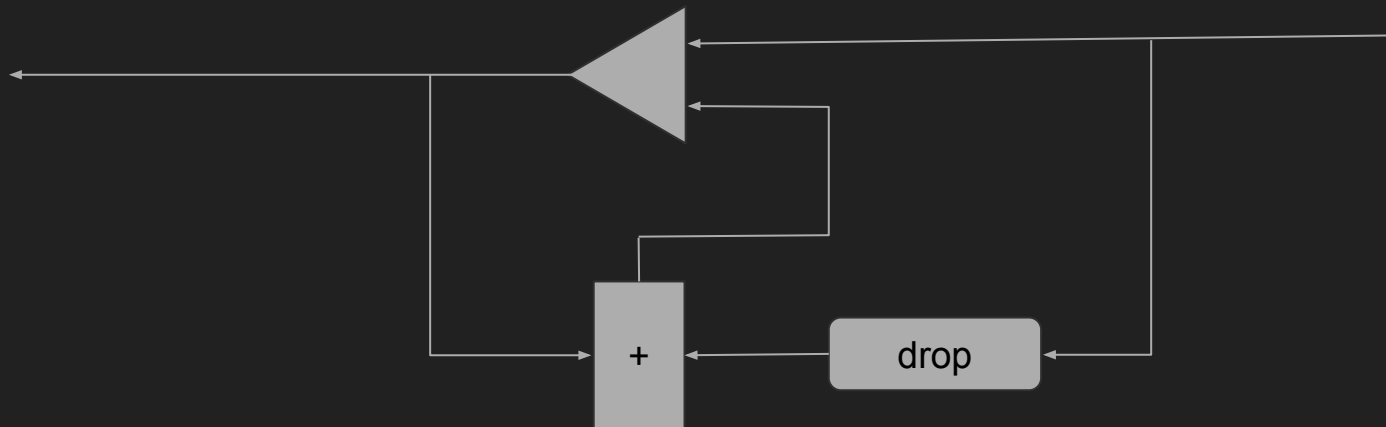


Transformation: partial sums

Input:	a	b	c	d	e	f	...
Output:	a	a+b	a+b+c	a+b+c+d	a+b+c+d+e	a+b+c+d+e+f	...

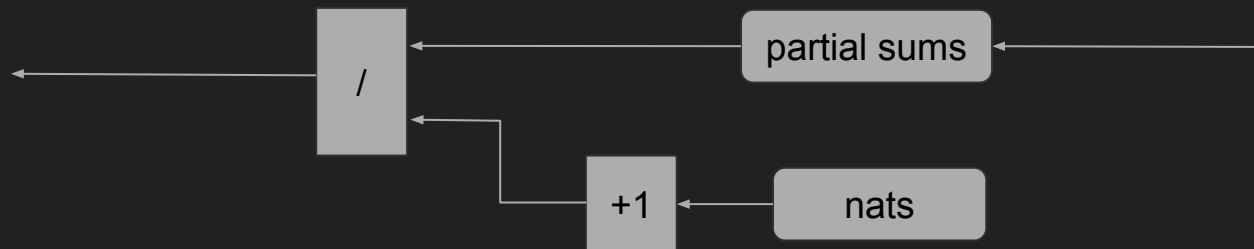
Transformation: partial sums

Input:	a	b	c	d	e	f	...
Output:	a	a+b	a+b+c	a+b+c+d	a+b+c+d+e	a+b+c+d+e+f	...



Transformations: running averages

Input:	a	b	c	d	e	...
Output:	$a/1$	$(a+b)/2$	$(a+b+c)/3$	$(a+b+c+d)/4$	$(a+b+c+d+e)/5$...



Definitions

A **dataflow network** with inputs I and outputs O is a finite network of components where:

1. every component is either a primitive component or an already defined dataflow network
2. every component's input is either in I or connected to exactly one output
3. every component's output can be connected to zero or more inputs and can also appear in O

Main theorem

A **cycle** in a dataflow network is a path from the output of some component back to an input of the same component by following links in the network

Theorem: *If every cycle in a dataflow network goes through the lower input of at least one "followed by" primitive component, then the dataflow network computes a function from its input streams to its output stream*