

Propagators: An Introduction

George Wilson

Data61/CSIRO

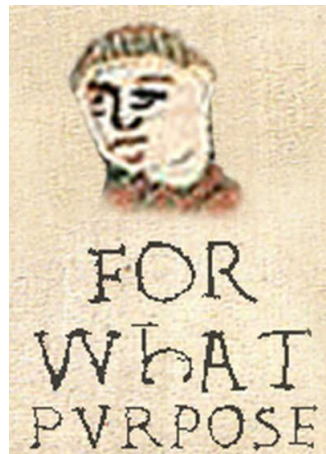
george.wilson@data61.csiro.au

November 9, 2017





What?



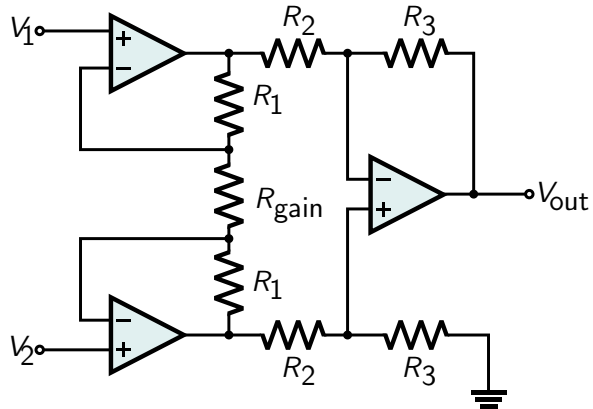
Why?

Roots as early as the 1970's at MIT

- Guy L. Steele Jr.
- Gerald J. Sussman
- Richard Stallman

More recently:

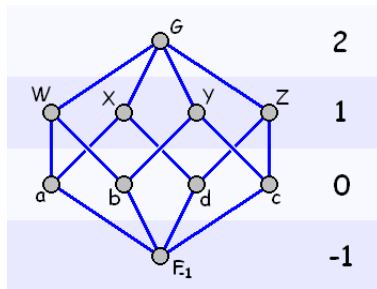
- Alexey Radul



```
(define (map f xs)
  (cond ((null? xs) '())
        (else (cons (f (car xs))
                      (map f (cdr xs)))))))
```

And then

- Edward Kmett



$$x \leq y \implies f(x) \leq f(y)$$

Propagators

The *propagator model* is a model of computation
We model computations as *propagator networks*

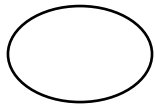
The *propagator model* is a model of computation
We model computations as *propagator networks*

Propagator networks:

- are extremely expressive
- lend themselves to parallel and distributed evaluation
- allow different strategies of problem-solving to seamlessly cooperate

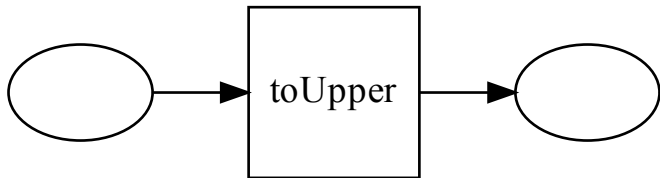
A propagator network comprises

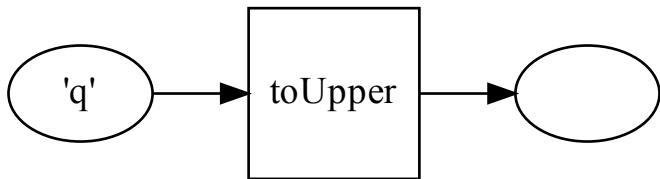
- cells
- propagators
- connections between cells and propagators

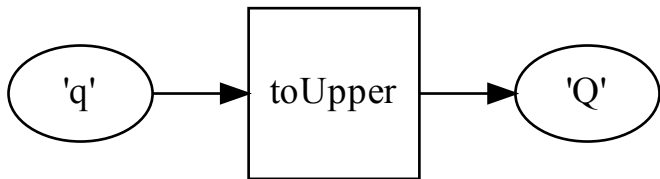


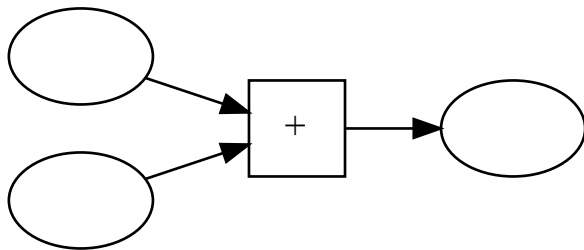
3

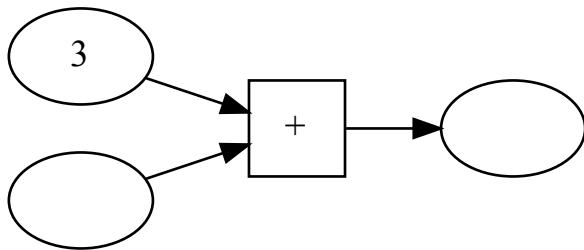
toUpper

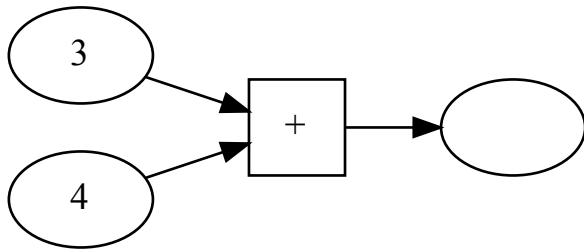


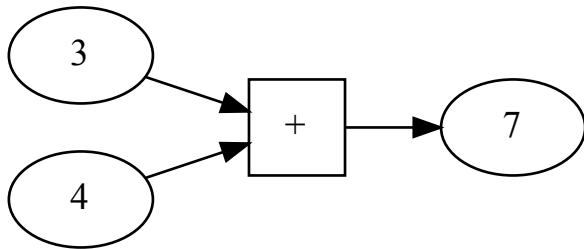












$$z \leftarrow x + y$$

$$z = x + y$$

$$7 = x + 4$$

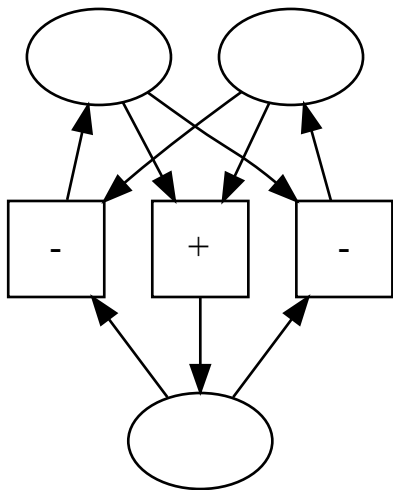
$$7 = 3 + 4$$

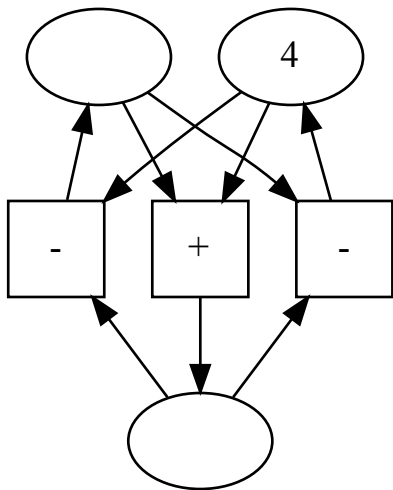
$$z = x + y$$

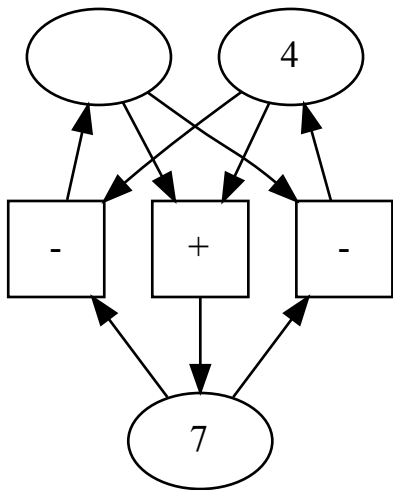
$$z \leftarrow x + y$$

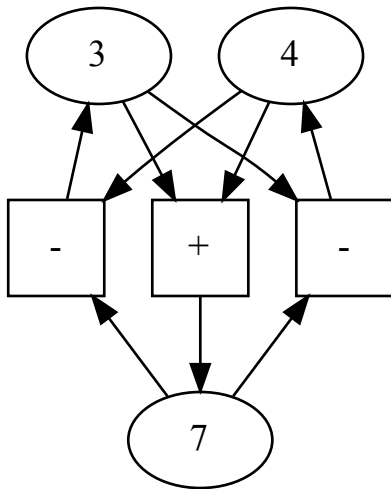
$$x \leftarrow z - y$$

$$y \leftarrow z - x$$



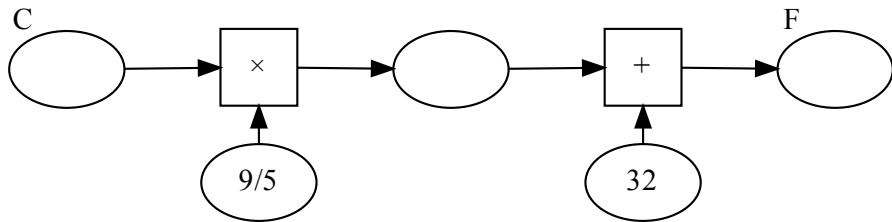




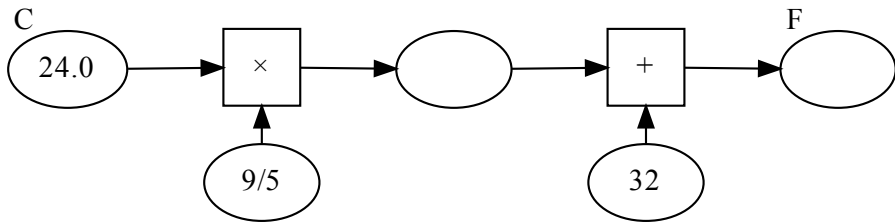


Propagators let us express multi-directional relationships!

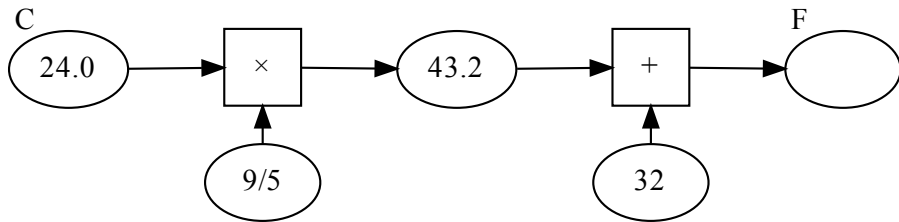
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$



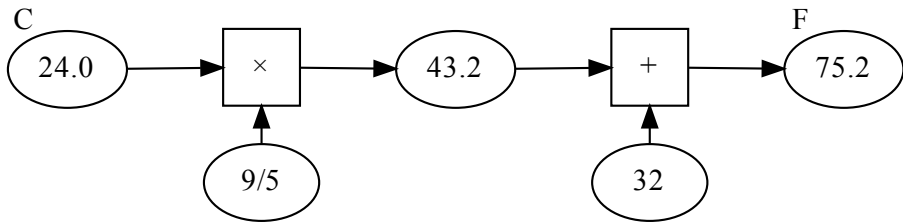
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$



$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$

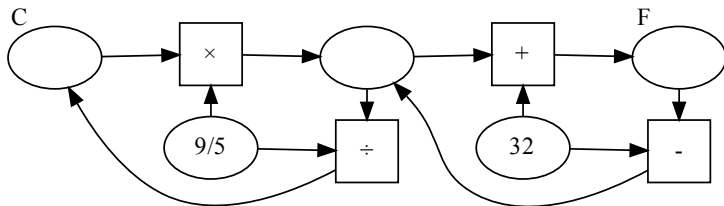


$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$



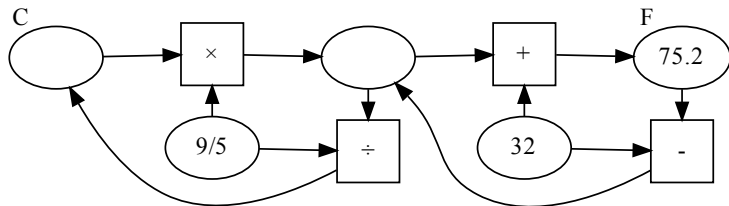
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$

$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$



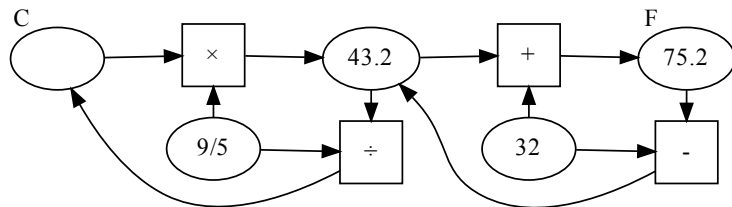
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$

$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$



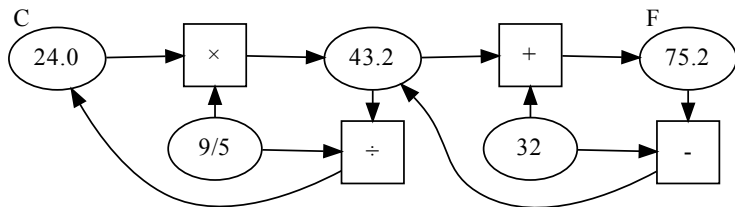
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$

$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$



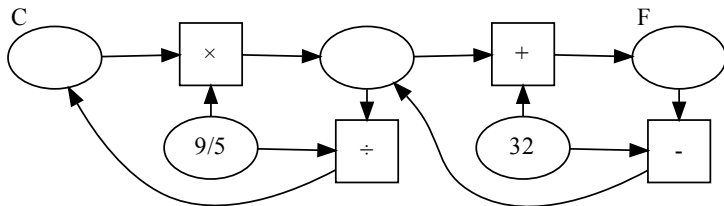
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$

$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$



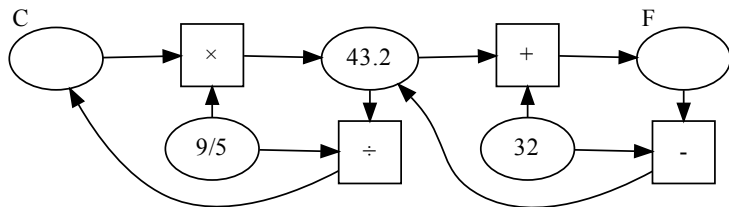
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$

$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$



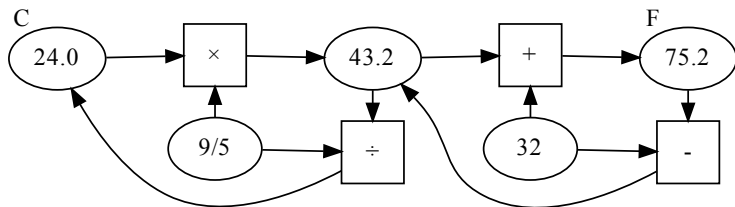
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$

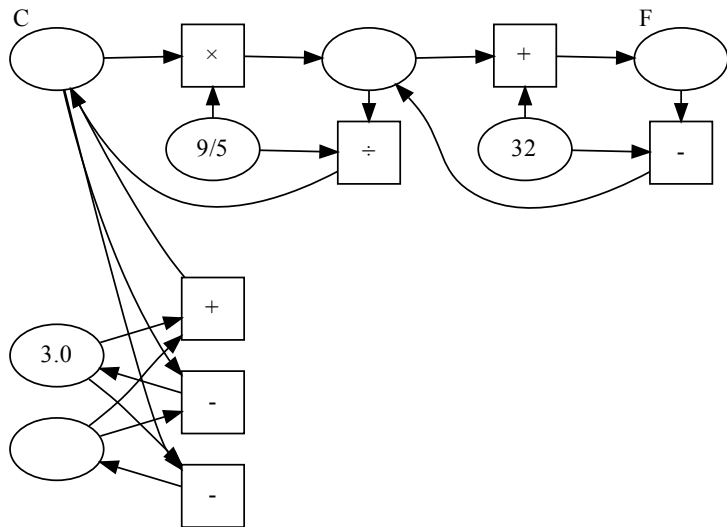
$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$

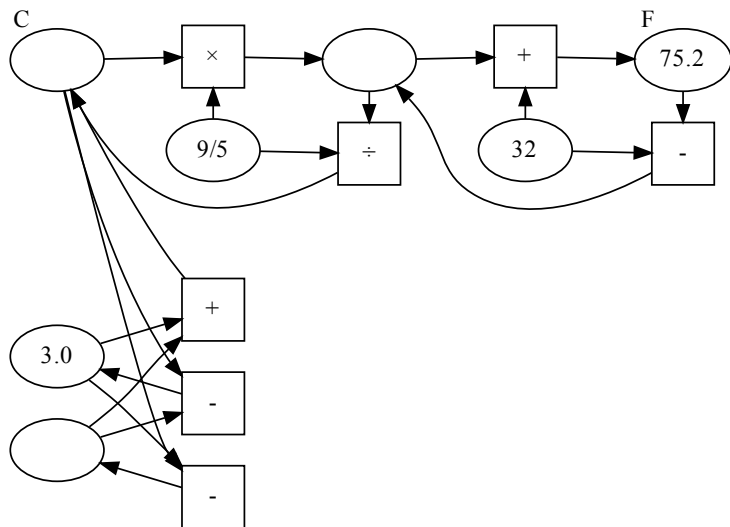


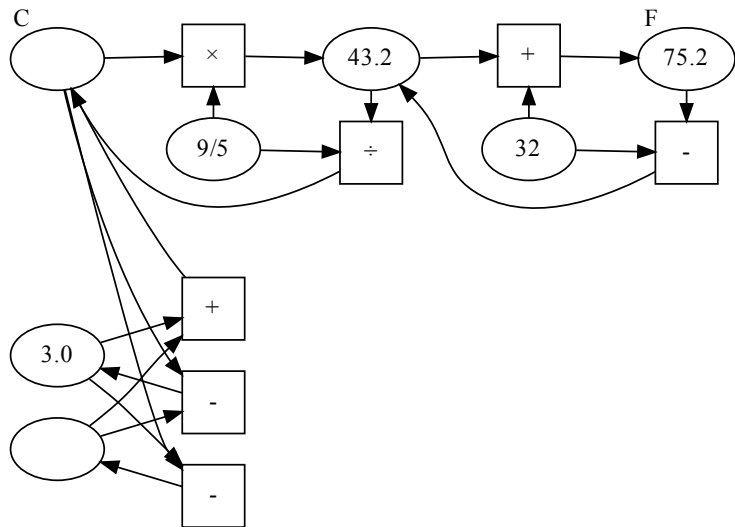
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$

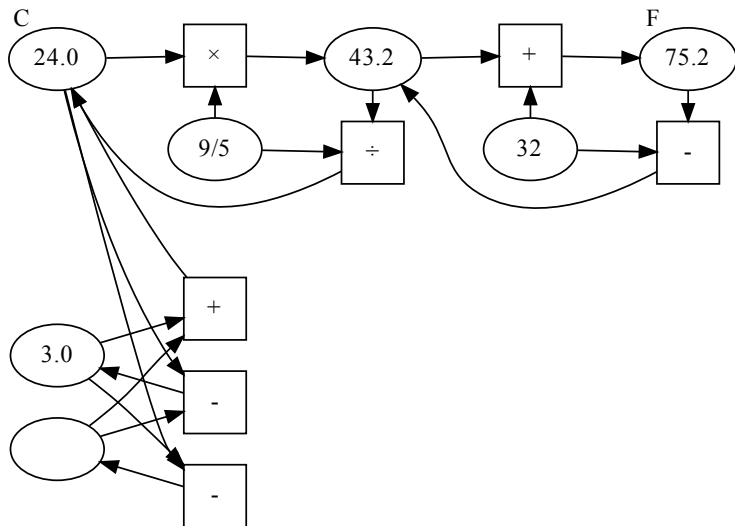
$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$

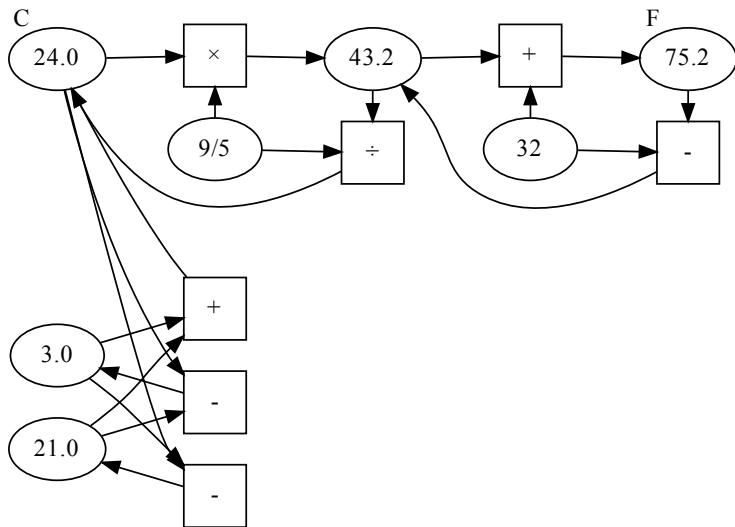












We can combine networks into larger networks!

?

Cells *accumulate information* about a value

$$[1, 5]$$

$$[1, 5] \cup [2, 7] = [2, 5]$$

$$[1, 5] \cup [2, 7] = [2, 5]$$

$$[2, 5] + [9, 10] = [11, 15]$$

TODO more interval stuff here

TV

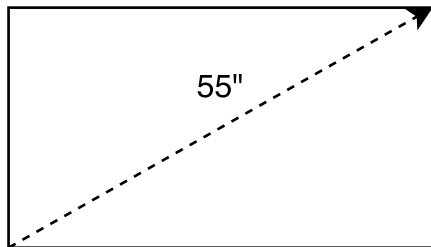
16:9

55"



TV

16:9

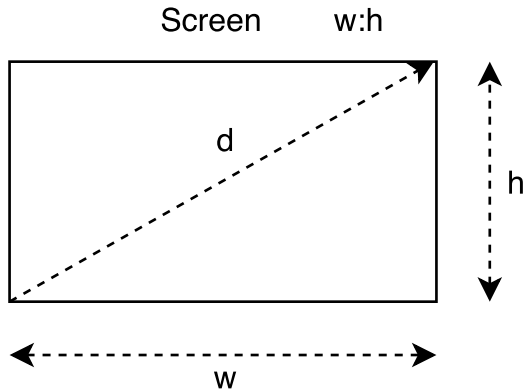


55"

Projector screen 16:9

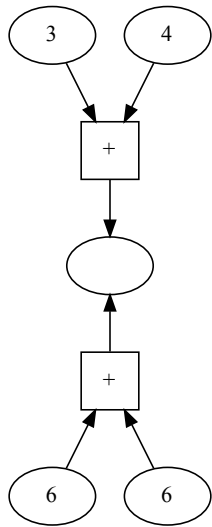


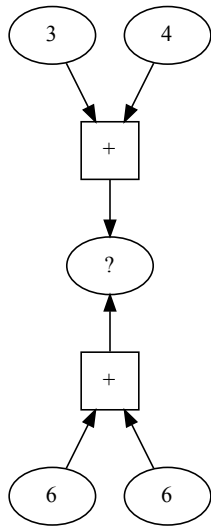
←-----→
10'

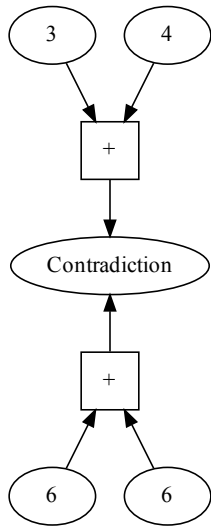


$\{True, False\}$

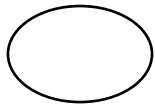
TODO set intersection examples

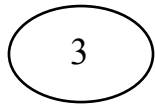






What types are the values of the cells?





'c'

Contradiction

data Perhaps a = Unknown | Known a | Contradiction

```
data Perhaps a = Unknown | Known a | Contradiction
```

```
instance Eq a => Monoid (Perhaps a) where
```

```
    mempty = Unknown
```

```
    mappend Unknown x           = x
```

```
    mappend x      Unknown      = x
```

```
    mappend Contradiction _      = Contradiction
```

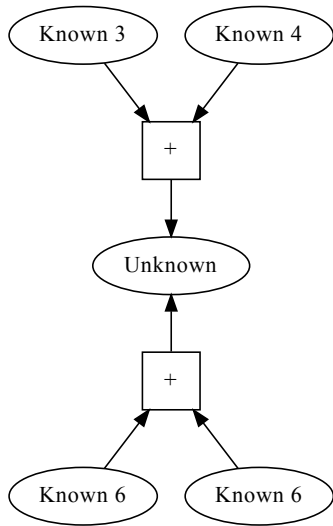
```
    mappend _      Contradiction = Contradiction
```

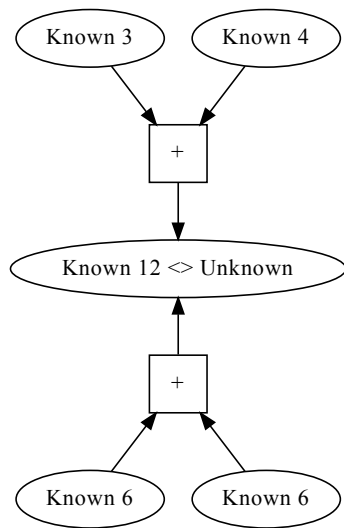
```
    mappend (Known a) (Known b) =
```

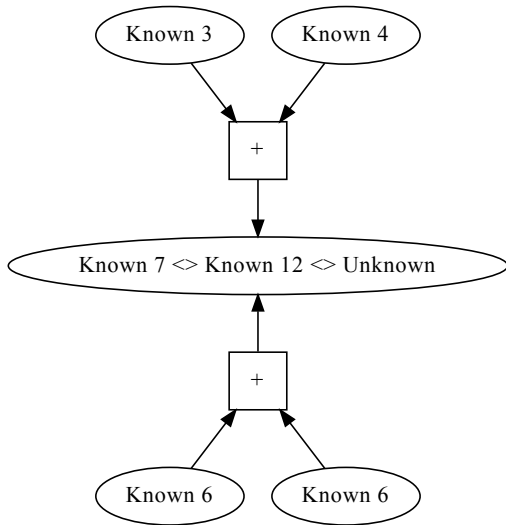
```
        if a == b
```

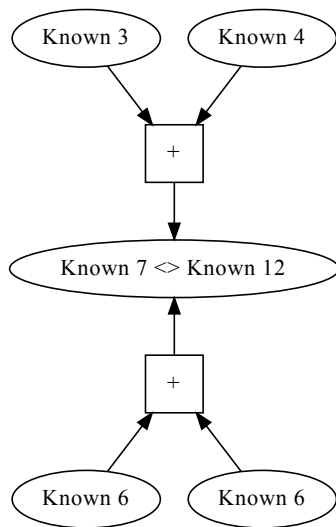
```
        then Known a
```

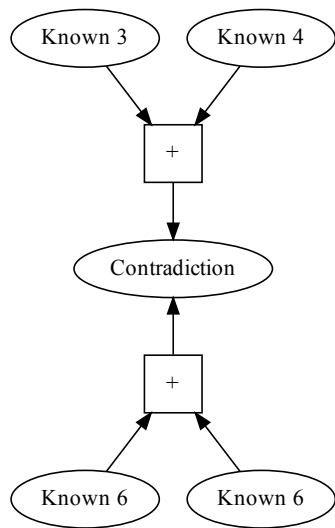
```
        else Contradiction
```







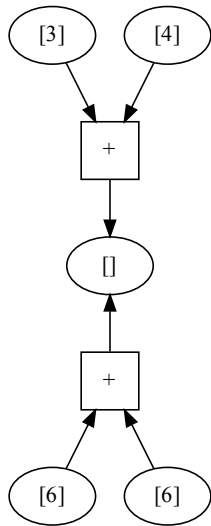


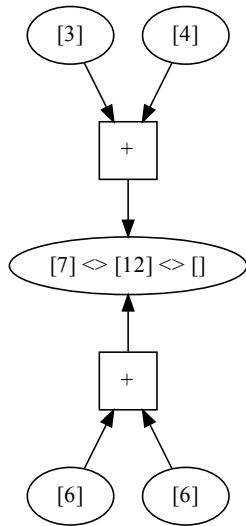


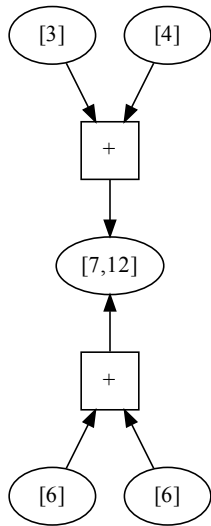
Is this the only type propagator cells can contain?
Will other monoids work?

Is this the only type propagator cells can contain?
Will other monoids work?

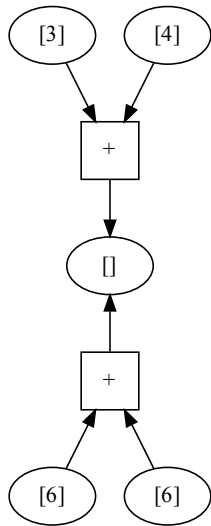
What about List?

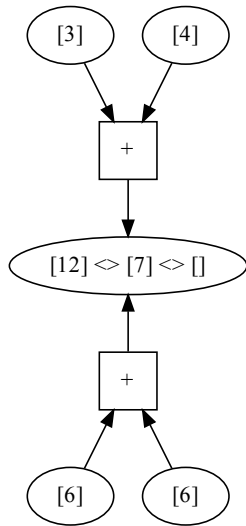


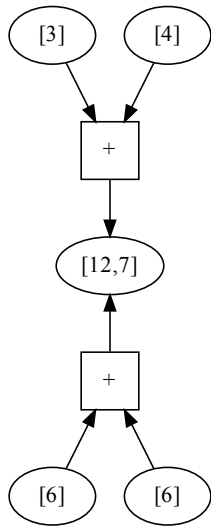


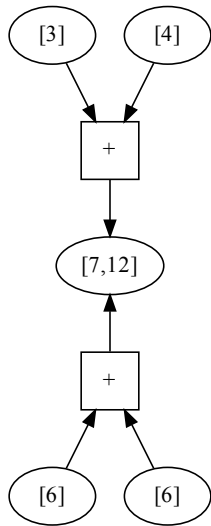


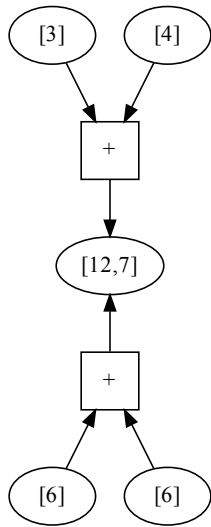
Looking good?











We need commutativity!

$$x \oplus y = y \oplus x$$

We need commutativity!

$$x \oplus y = y \oplus x$$

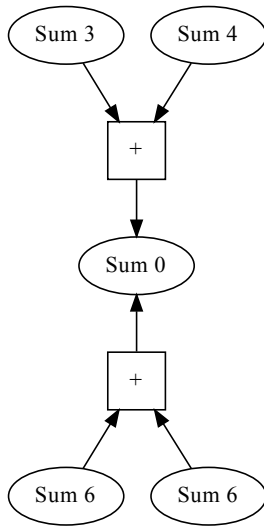
List append is not commutative!

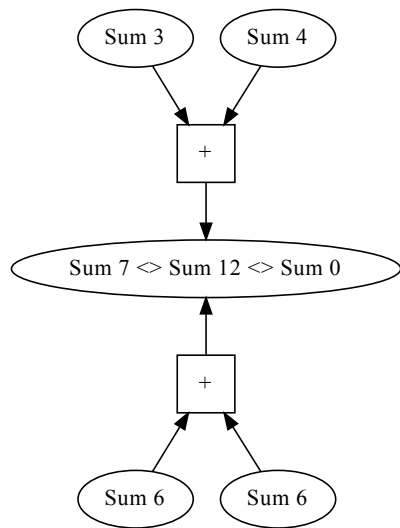
`[1, 2, 3] <> [4, 5, 6] == [1, 2, 3, 4, 5, 6]`

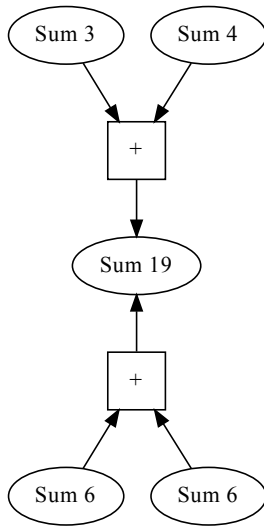
`[4, 5, 6] <> [1, 2, 3] == [4, 5, 6, 1, 2, 3]`

We need a commutative monoid
What about addition?

$$x + y = y + x$$







We need idempotence!

$$x \oplus x = x$$

We need an idempotent, commutative monoid.

This structure is called a *join-semilattice*

Associativity

$$(x \vee y) \vee z = x \vee (y \vee z)$$

Commutativity

$$x \vee y = y \vee x$$

Idempotence

$$x \vee x = x$$

Partial information that supports merging!

Other examples?