Propagators: An Introduction

George Wilson

Data61/CSIRO

george.wilson@data61.csiro.au

November 6, 2017





What?



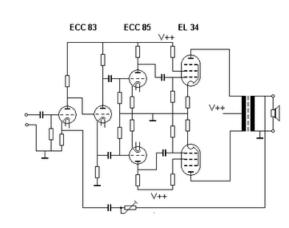
Why?

Roots as early as the 1970's at MIT

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

More recently:

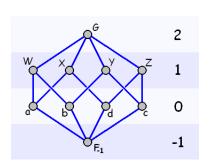
Alexey Radul



And then

• Edward Kmett





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

Propagators

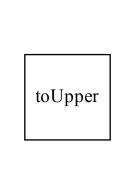
The <i>propagator model</i> is a model of computation	
We model computations as propagator networks	

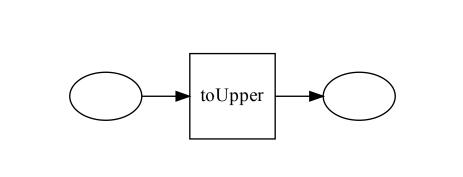
A propagator network comprises

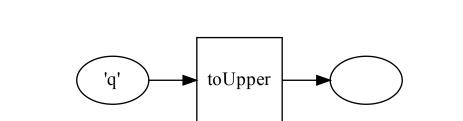
- cells
- propagators
- connections between cells and propagators

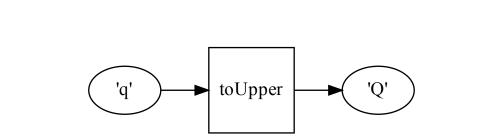


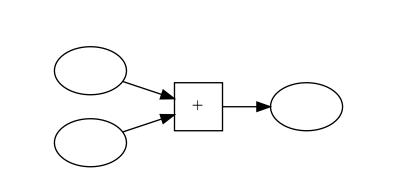


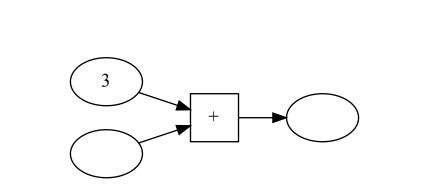


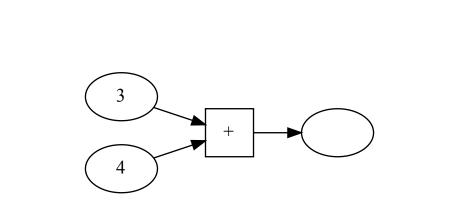


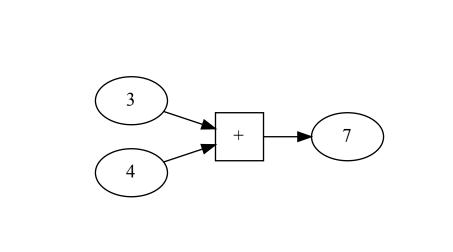


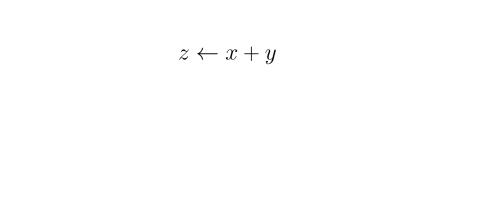


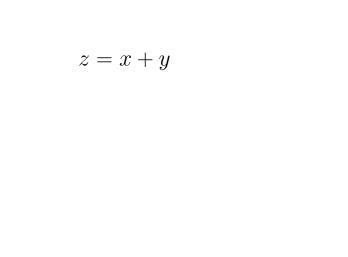


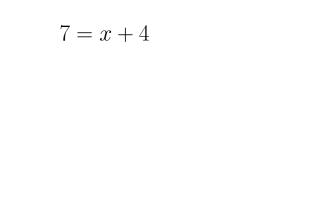


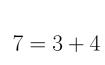


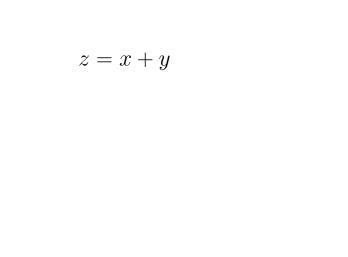


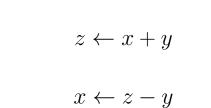




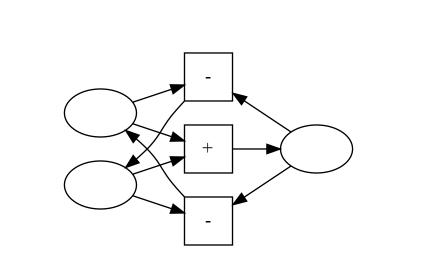


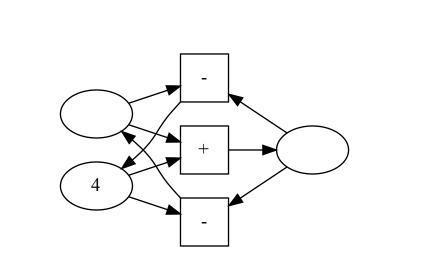


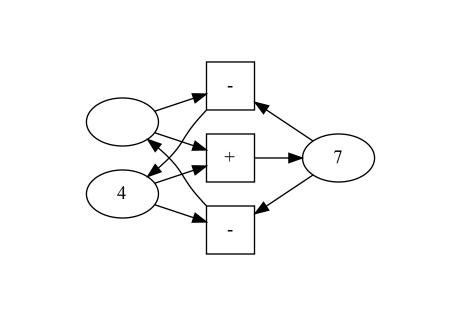


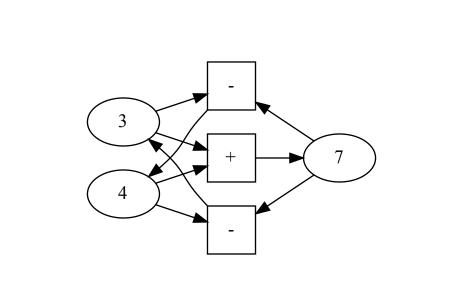


 $y \leftarrow z - x$



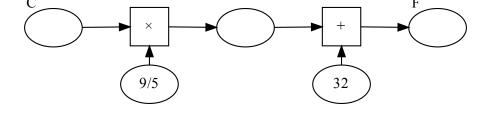




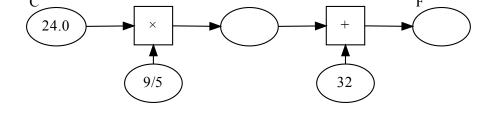


Propagators let us express multidirectional relationships!

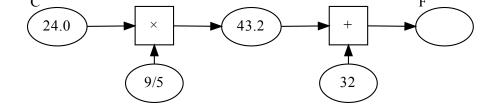
 $^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$



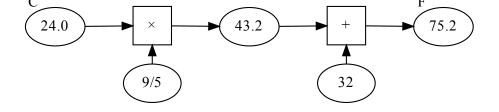
$$^{\circ}F = ^{\circ}C \times \frac{9}{5} + 32$$



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



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



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

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

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

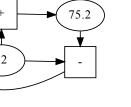
75.2

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

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

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

43.2



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

24.0

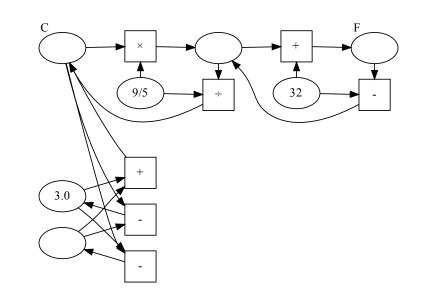
$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$

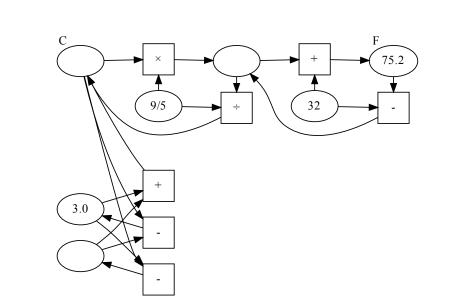
$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$

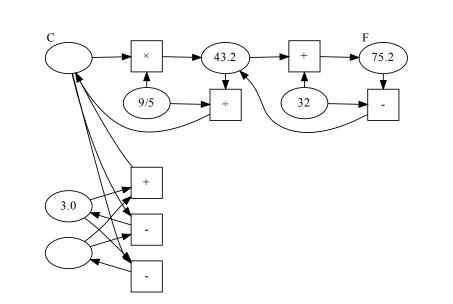
43.2

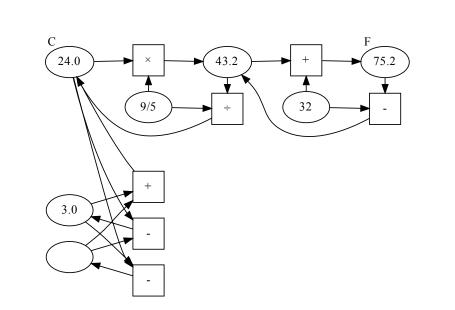
$$^{\circ}C = (^{\circ}F - 32) \div \frac{9}{5}$$

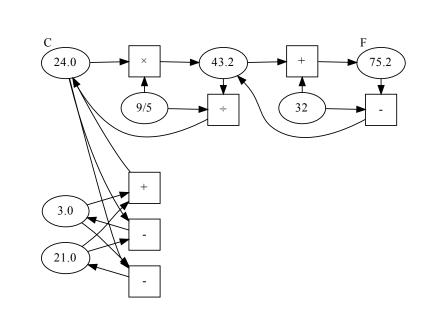
24.0





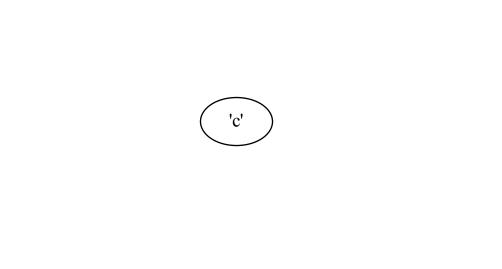






We can combine networks into larger networks!

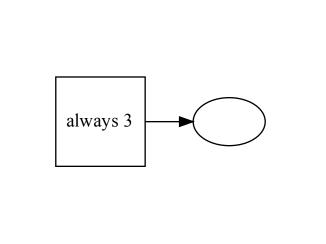
What types are the values of the cells?

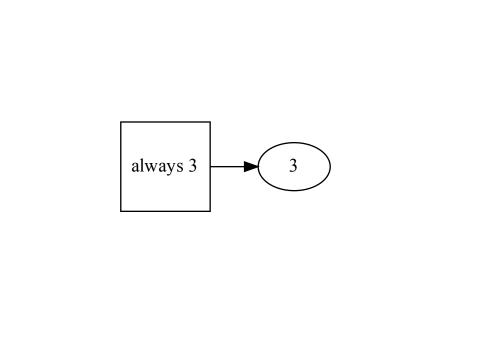


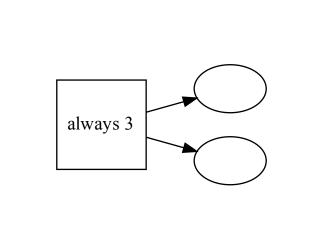


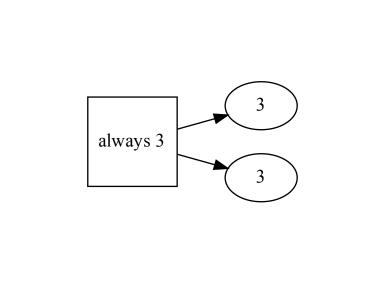


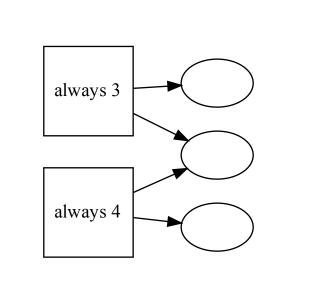
data Maybe a = Nothing | Just a

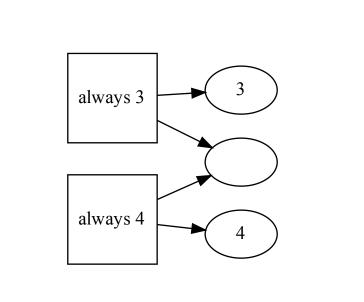


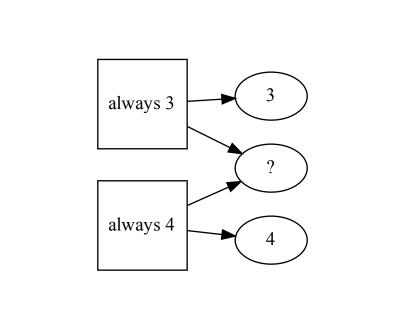












(

. /

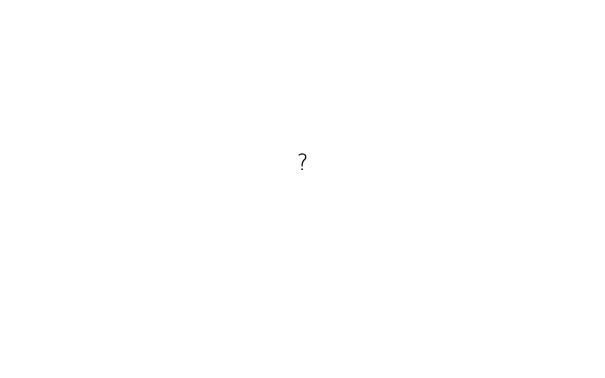
Contradiction

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

Partial information!



[1, 5]

[1,5] <> [2,7] = [2,5]

$\{True, False\}$

