Polynomial functors in Catlab

Angeline Aguinaldo[1,2], Kris Brown[3], Marco Perin[4]

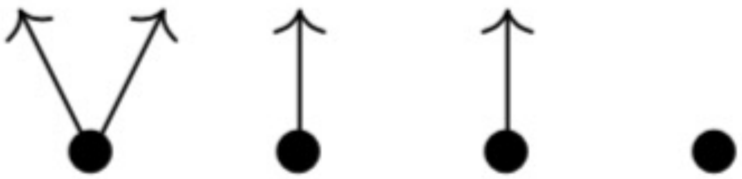
ACT Conference 2021

[1] University of Maryland, College Park, [2] Johns Hopkins University Applied Physics Laboratory, [3] Stanford University, [4]University of Nottingham

What is a polynomial functor (PF)?

$y^2 + 2y + 1$ *polynomial*

directions



forest

positions

Sum of representable functors (y^A: $\mathbf{Set} \to \mathbf{Set}$, $\mathrm{A} \in \mathbf{Set}$)

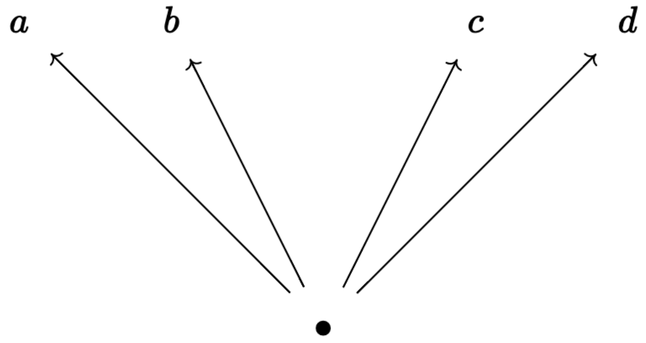
[1] Operational Semantics for Distributed Systems. Spivak and Niu. In Progress.

PF for mode-dependent dynamical systems: An Intuition

Mode 1
- Option **a**
- Option **b**
- Option **c**
- Option **d**

Mode 2
- Option **e**
- Option **c**

Mode 3
- Option **f**
- Option **g**
- Option **c**

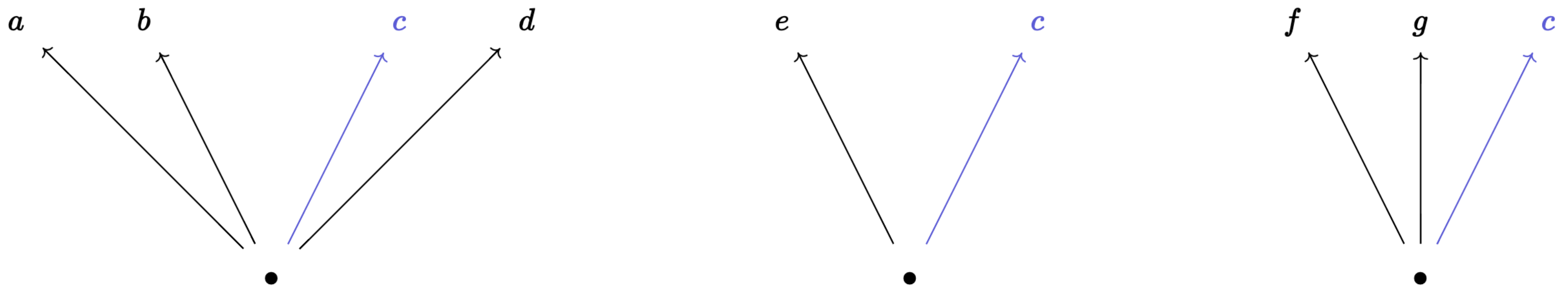


PF for mode-dependent dynamical systems: An Intuition

Mode 1
- Option **a**
- Option **b**
- Option **c**
- Option **d**

Mode 2
- Option **e**
- Option **c**

Mode 3
- Option **f**
- Option **g**
- Option **c**

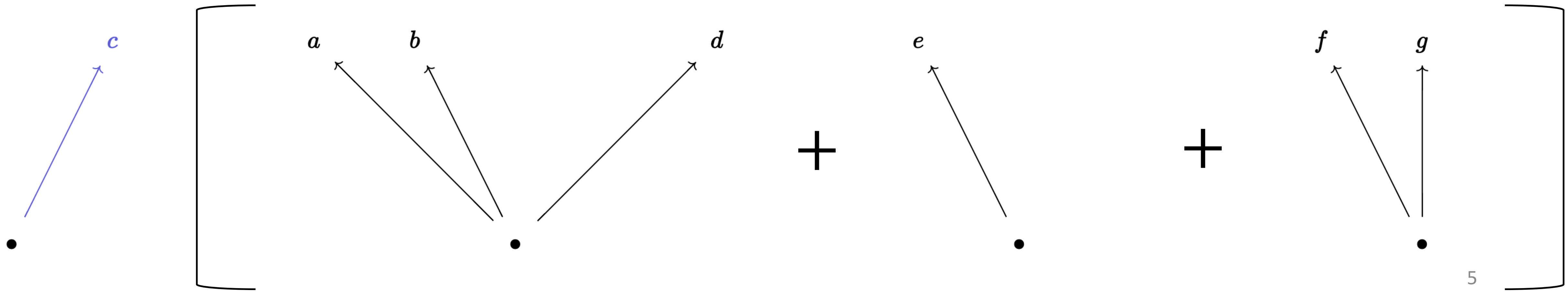


PF for mode-dependent dynamical systems: An Intuition

Mode 1
- Option **a**
- Option **b**
- Option **c**
- Option **d**

Mode 2
- Option **e**
- Option **c**

Mode 3
- Option **f**
- Option **g**
- Option **c**



Example: Happy Refrigerator

"Add" (A) Mode

- Add drink
- Don't add drink

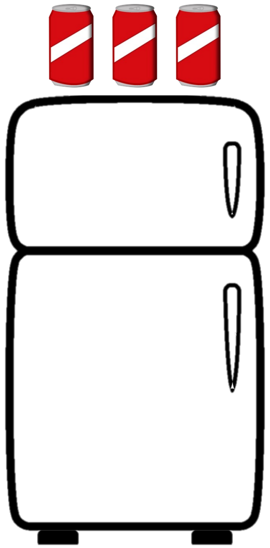
"Take" (T) Mode

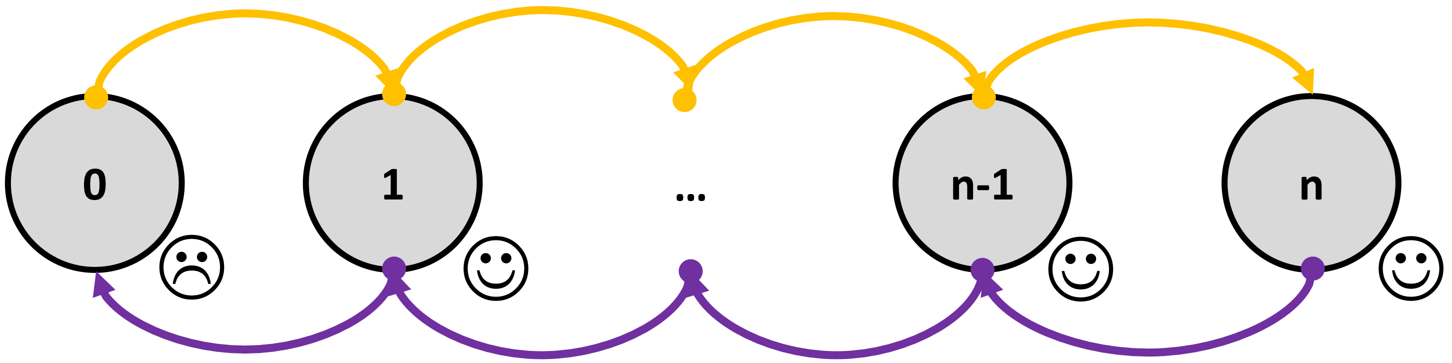
- Take drink
- Don't take drink

"Add or Take" (AT) Mode

- Add drink, don't take
- Don't add, take drink
- Add drink, take drink
- Do nothing

n := number of drinks





$☺ y^{AT} + ☹ y^{A} + ☺ y^{T}$

Finite Polynomials and Coalgebras in Catlab

```julia

mutable struct PolyDynam
    p::FinPolyLabel
    nStates::Int
    modes::AbstractVector{Int}
    behaviors::AbstractVector{Any}
    s0::Int
end

function run(D::PolyDynam)
    while true
        position = D.modes[D.s0]
        behavior = D.behaviors[D.s0]
        output = subpart(D.p, position, :pos_label)
        @printf("(State %s) %s\n\n", D.s0-1, output)

        directions = incident(D.p, position, :pos)
        options = subpart(D.p, directions, :dir_label)
        @printf("Please select your choice for '%s':\n", String(Symbol(behavior)))
```

Example: Happy Refrigerator

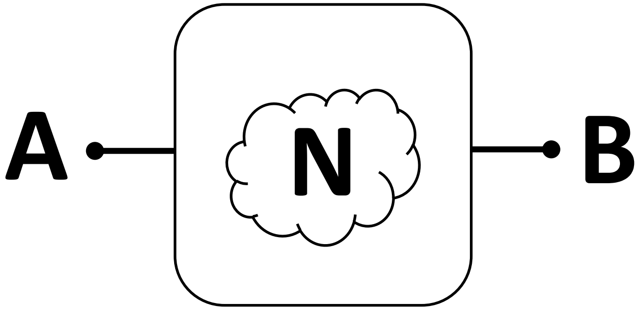
```
D = PolyDynam(p, # polynomial
              5, # number of states
              [2, 1, 1, 1, 3], # position per state
              [add, add_and_take, add_and_take, add_and_take, take], # behavior
              per state
              2) # initial state index
```

Next Steps

Universal Programable Machine and User Interface

Universal programable machine

(A,B) Moore Machine

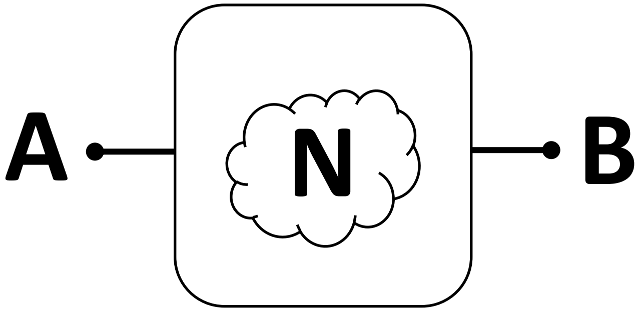


$$N \times A \xrightarrow{\textbf{update}} N$$

$$N \xrightarrow{\textbf{readout}} B$$

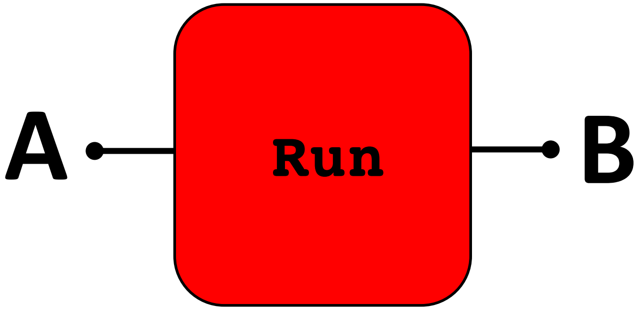
Universal programable machine

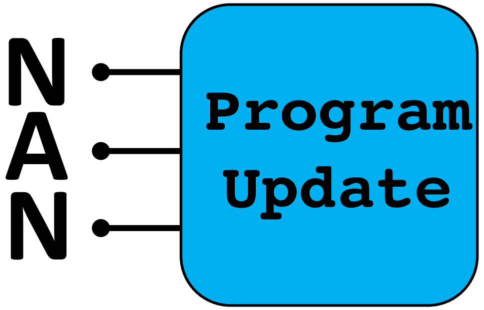
(A,B) Moore Machine

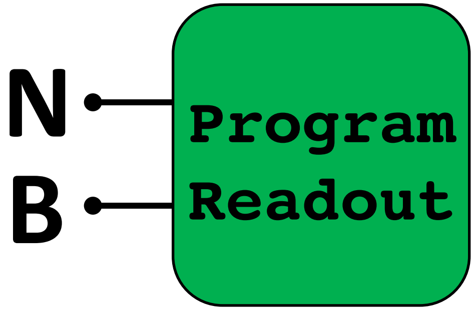


$$N \times A \xrightarrow{\textbf{update}} N$$

$$N \xrightarrow{\textbf{readout}} B$$





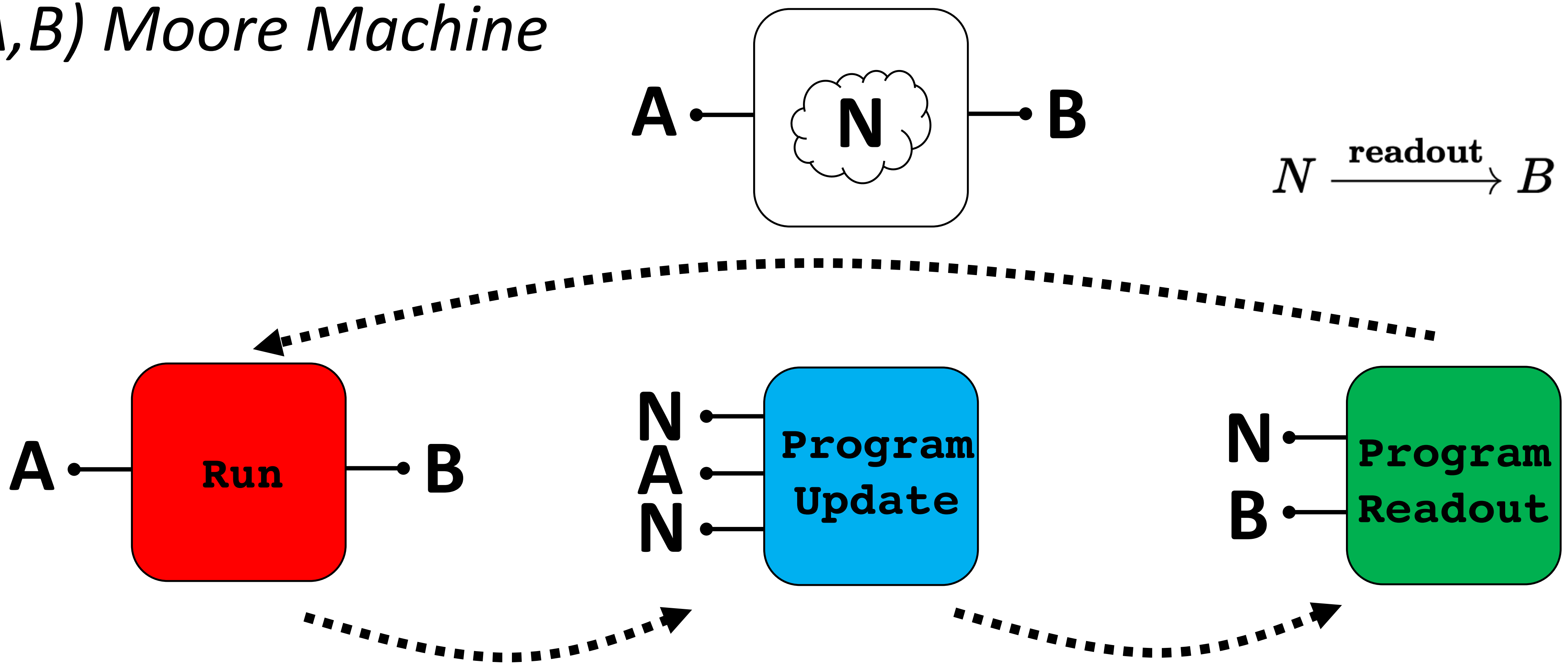


Universal programable machine

(A,B) Moore Machine

$$N \times A \xrightarrow{\text{update}} N$$

$$N \xrightarrow{\text{readout}} B$$

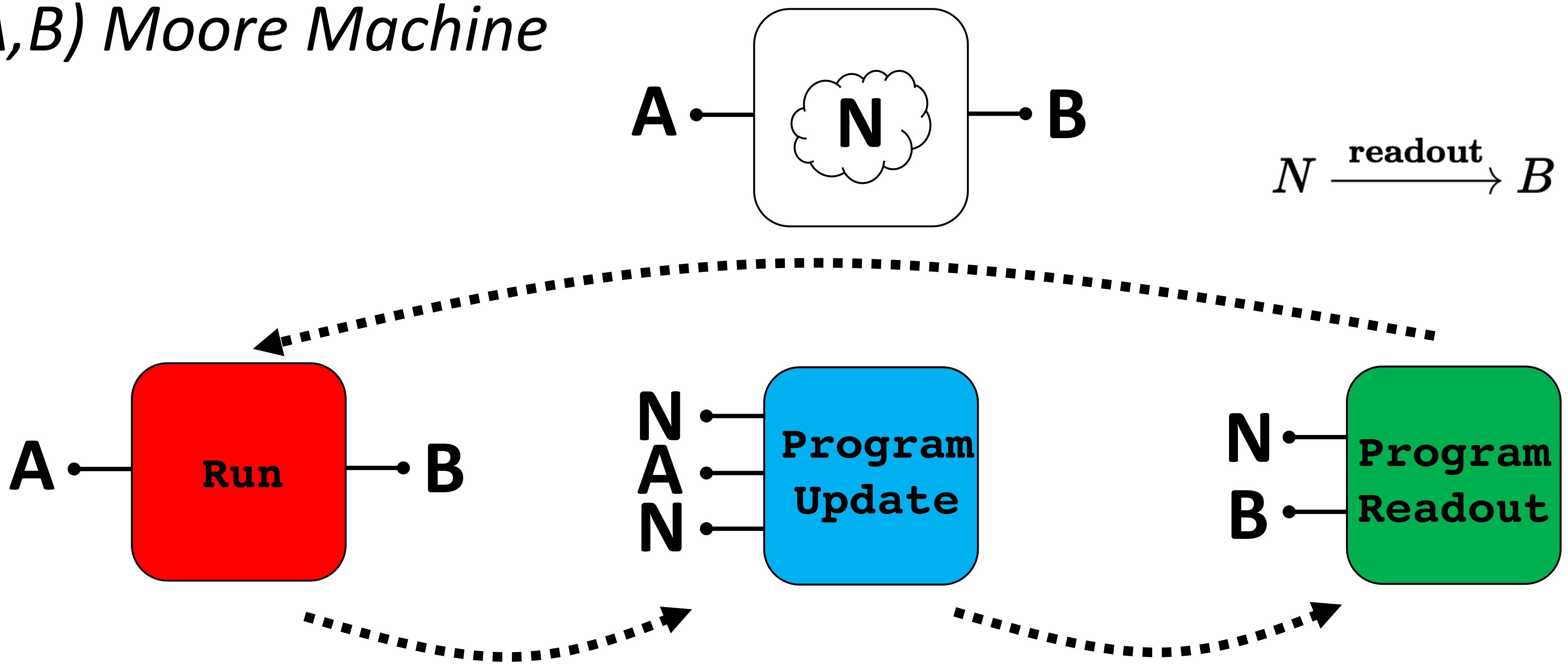


Universal programable machine

(A,B) Moore Machine

$$N \times A \xrightarrow{\textbf{update}} N$$

$$N \xrightarrow{\textbf{readout}} B$$

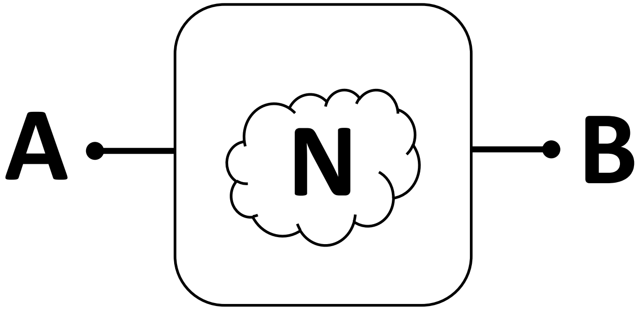


$$y(By^{A} + y^{NAN} + y^{NB})$$

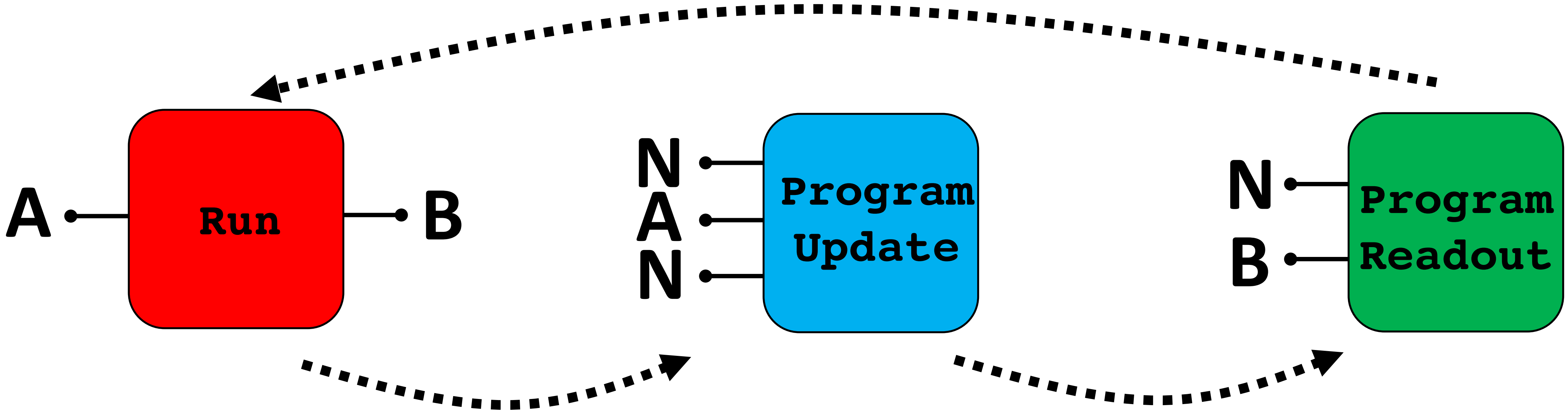
Universal programable machine

(A,B) Moore Machine

$$N \times A \xrightarrow{\textbf{update}} N$$



$$N \xrightarrow{\textbf{readout}} B$$



$$y(By^{A} + y^{NAN} + y^{NB})$$

$$By^{A+1} + y^{NAN+1} + y^{NB+1}$$

A restricted class of polynomials

$$\sum_{m \in Mode} \left(\prod_{o \in Out_m} T_o \cdot y^{\sum_{s \in Sig_m} \prod_{i \in In_s} T_i} \right)$$

A restricted class of polynomials

$$\sum_{m\in Mode}\left(\prod_{o\in Out_m} T_o \cdot y^{\sum_{s\in Sig_m}\prod_{i\in In_s} T_i}\right)$$

✓

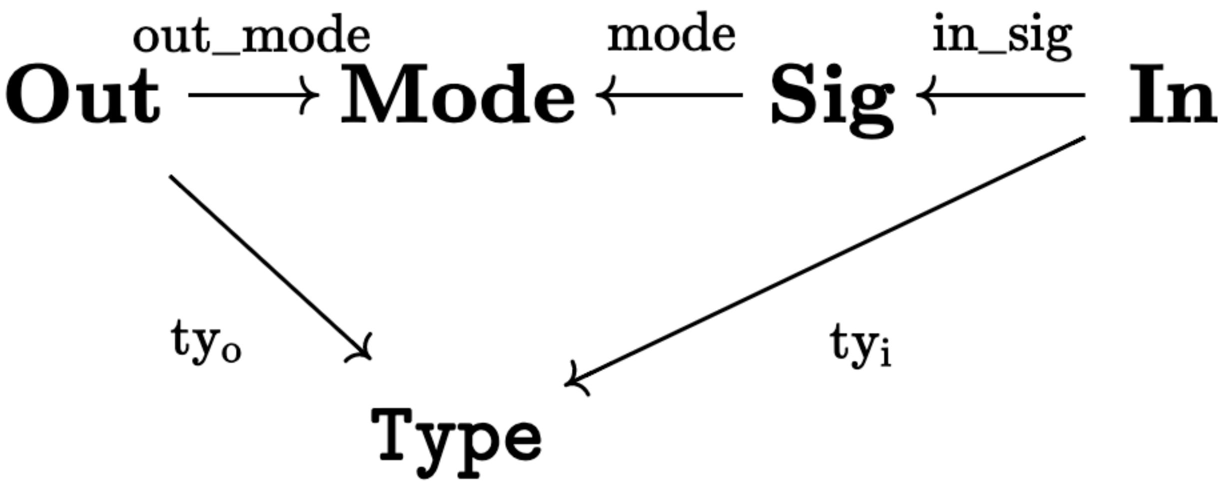
$$\mathbb{R}\cdot y^{\mathbb{R}} + \mathbb{Q}\times \mathbf{Bool}\cdot y^{\mathbb{R}+\mathbb{Z}\times\mathbb{Z}} + y + 1$$

✗

$$\mathbb{R}^{\mathbb{R}}\cdot y \qquad y^{\mathbb{Z}^{\mathbb{Z}}}$$

Attributed C-Sets

$$\sum_{m\in Mode}\left(\prod_{o\in Out_m} T_o \cdot y^{\sum_{s\in Sig_m}\prod_{i\in In_s} T_i}\right)$$



Combinatorial data

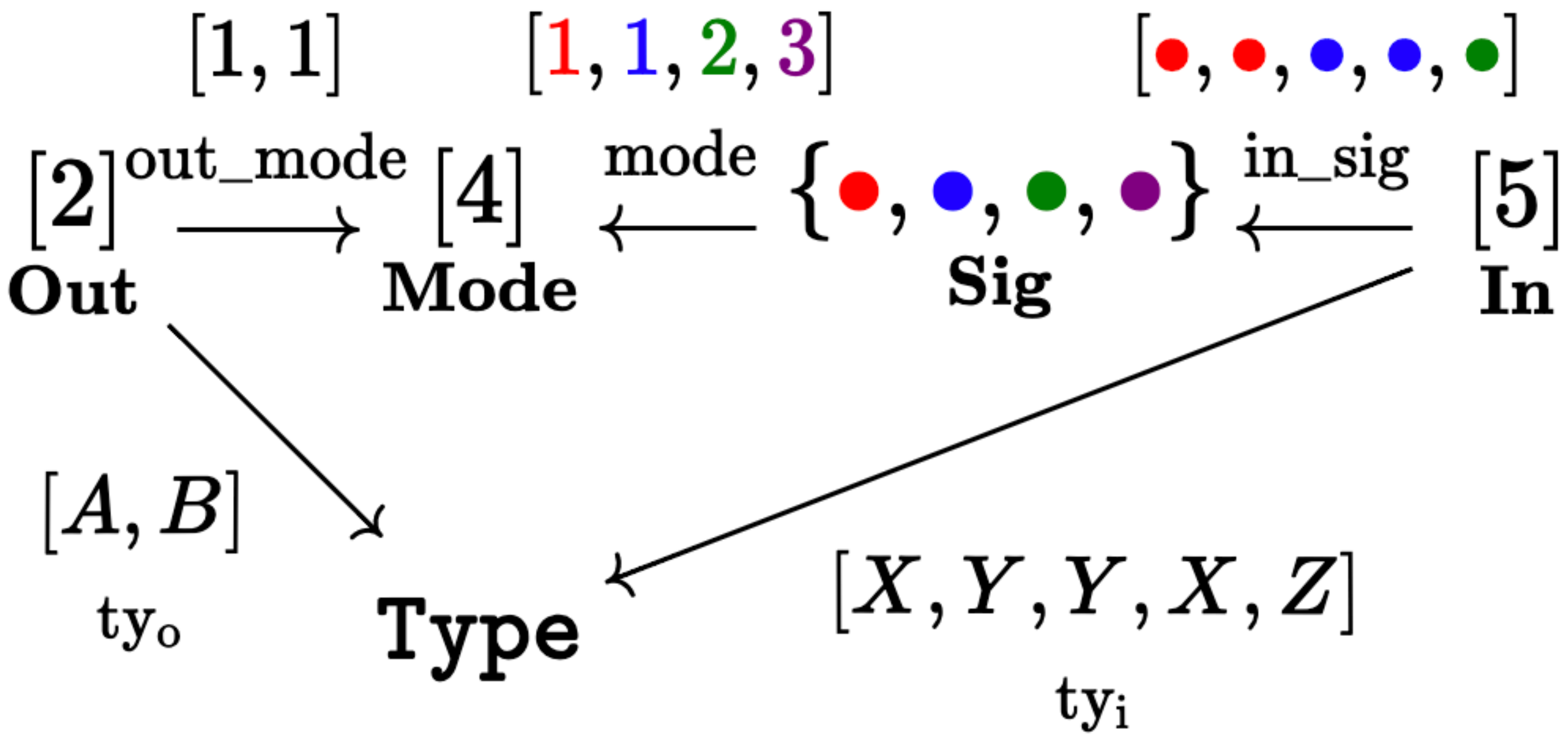
- Identifiers have no meaning
- Implementation: Skeleton of **FinSet**
- E.g. **[2]**

Non-combinatorial data

- Preserved on the nose by morphisms
- Implementation: Arbitrary Julia types
- E.g. Bool

Example instance

$$AB \cdot y^{XY+YX} + y^{Z} + y^{1} + 1$$



Back to middle school

In [2]:
```
p = SumProdPoly{Symbol}([
        [[:B]] => [:S],
        [[:S]] => [:S,:A] ])
show_(p)
```

$$Sy^B + S \cdot Ay^S$$

In [5]:
```
show_(p*p)
```

$$S \cdot Sy^{B+B} + S \cdot A \cdot Sy^{S+B} + S \cdot S \cdot Ay^{B+S} + S \cdot A \cdot S \cdot Ay^{S+S}$$

In [4]:
```
show_(p⊗p)
```

$$S \cdot Sy^{B \cdot B} + S \cdot A \cdot Sy^{S \cdot B} + S \cdot S \cdot Ay^{B \cdot S} + S \cdot A \cdot S \cdot Ay^{S \cdot S}$$

Future work: mode-dependent wiring diagrams with arbitrary data flowing along wires

Thank you

- Evan Patterson
- David Spivak
- Sophie Libkind
- Christian Williams