Transition Diagrams

EECS 20

Lecture 9 (February 5, 2001)

Tom Henzinger

Revised by Matteo Zavatteri and Tiziano Villa, Fall 2020

Discrete-Time Reactive System :

input/output function F

State-Machine Implementation :

decomposition of F into

1. memory-free part (called "Update")
2. delay part (what delay stores is called "state")

Discrete-Time Reactive System

$$F : [\, Nats_0 \rightarrow Inputs \,] \rightarrow [\, Nats_0 \rightarrow Outputs \,]$$

The Parity System

$\text{Parity} : [\,\text{Nats}_0 \to \text{Bools}\,] \to [\,\text{Nats}_0 \to \text{Bools}\,]$

such that $\forall\, x \in [\,\text{Nats}_0 \to \text{Bools}\,]$, $\forall\, y \in \text{Nats}_0$,

$$(\,\text{Parity}\,(x)\,)\,(y) = \begin{cases} \text{true} & \text{if } |\,\text{trueValues}\,(x,y)\,| \text{ is even} \\ \text{false} & \text{if } |\,\text{trueValues}\,(x,y)\,| \text{ is odd} \end{cases}$$

where $\text{trueValues}\,(x,y) = \{\, z \in \text{Nats}_0 \mid z < y \wedge x\,(z) = \text{true} \,\}$

The Count System

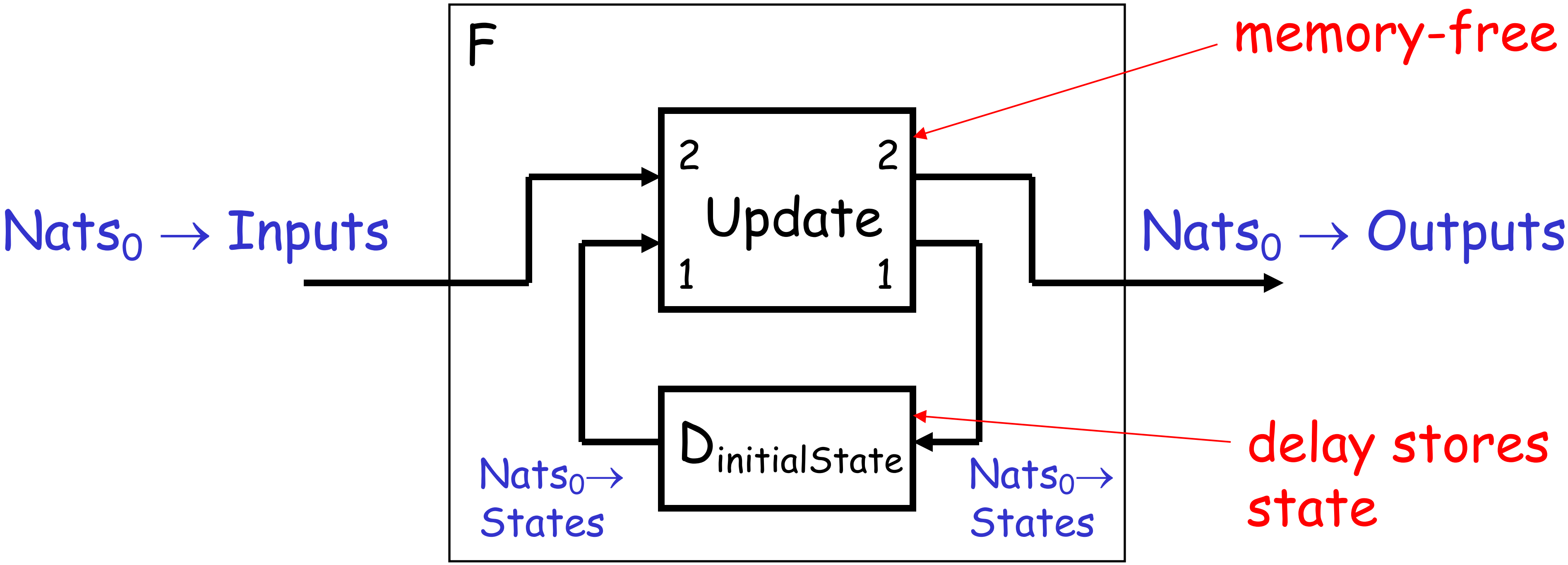
$$\text{Count} : [\,\text{Nats}_0 \rightarrow \text{Bools}\,] \rightarrow [\,\text{Nats}_0 \rightarrow \text{Bools}\,]$$

such that $\forall\, x \in [\,\text{Nats}_0 \rightarrow \text{Bools}\,]$, $\forall\, y \in \text{Nats}_0$,

$$(\,\text{Count}(x)\,)(y) = \begin{cases} \text{true} & \text{if } |\,\text{trueValues}(x,y)\,| \geq |\,\text{falseValues}(x,y)\,| \\ \text{false} & \text{if } |\,\text{trueValues}(x,y)\,| < |\,\text{falseValues}(x,y)\,| \end{cases}$$

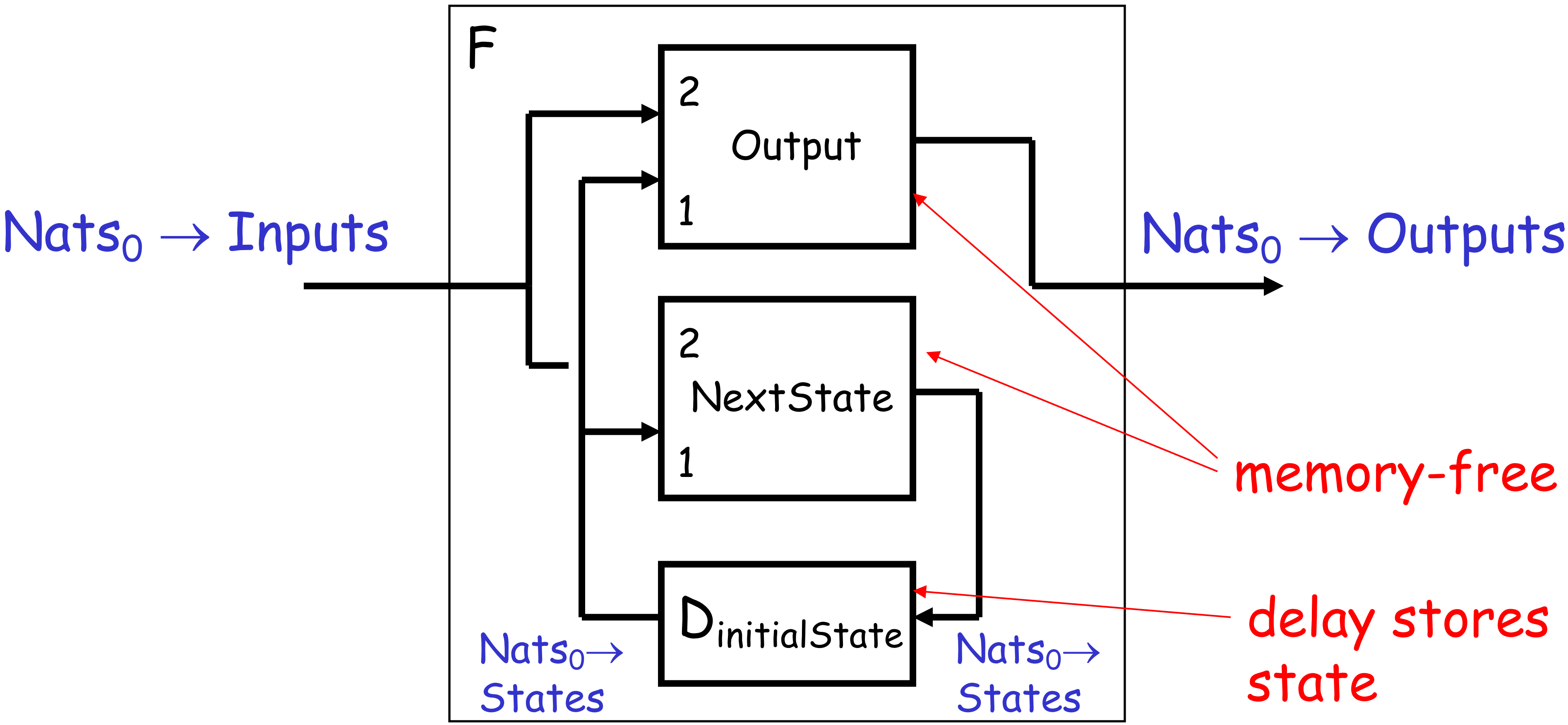
where $\text{falseValues}(x,y) = \{\, z \in \text{Nats}_0 \mid z < y \wedge x(z) = \text{false} \,\}$

State-Machine Implementation



$$update : \; States \times Inputs \;\to\; States \times Outputs$$

$$initialState \in States$$



$$\text{nextState}: \text{States} \times \text{Inputs} \to \text{States}$$

$$\text{output}: \text{States} \times \text{Inputs} \to \text{Outputs}$$

$$\text{initialState} \in \text{States}$$

State-Machine Implementation of the Parity System

State after i-th input :

true, if the first i inputs contain an even number of true values

false, if the first i inputs contain an odd number of true values

Two states

$$\text{Inputs} = \text{Bools}$$

$$\text{Outputs} = \text{Bools}$$

$$\text{States} = \text{Bools}$$

$$\text{initialState} = \text{true}$$

$$\text{nextState} : \text{States} \times \text{Inputs} \to \text{States}$$

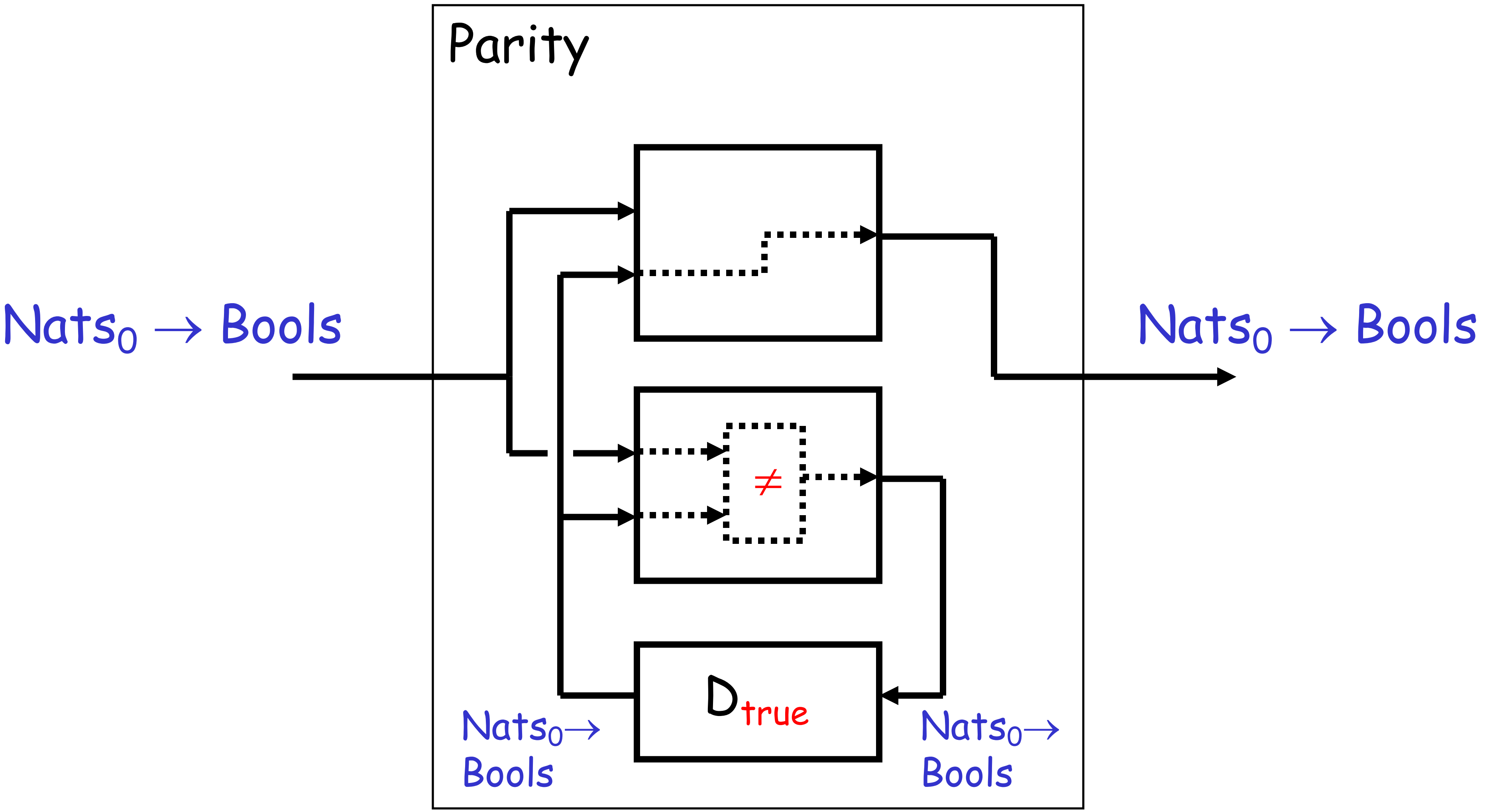
$$\text{such that } \forall\, q \in \text{States},\ \forall\, x \in \text{Inputs},$$

$$\text{nextState}(q,x) = (q \neq x)$$

$$\text{output} : \text{States} \times \text{Inputs} \to \text{Outputs}$$

$$\text{such that } \forall\, q \in \text{States},\ \forall\, x \in \text{Inputs},$$

$$\text{output}(q,x) = q$$



Parity
$Nats_0 \to Bools$
$Nats_0 \to Bools$
$\neq$
D_{true}
$Nats_0 \to Bools$
$Nats_0 \to Bools$

State-Machine Implementation of the Count System

State after i-th input :

j, if the first i inputs contain j more true values than false values

$-j$, if the first i inputs contain j more false values than true values

Infinitely many states

$$
\begin{aligned}
\text{Inputs} &= \text{Bools} \\
\text{Outputs} &= \text{Bools} \\
\text{States} &= \text{Ints} \\
\text{initialState} &= 0
\end{aligned}
$$

$$\text{nextState} : \text{States} \times \text{Inputs} \to \text{States}$$

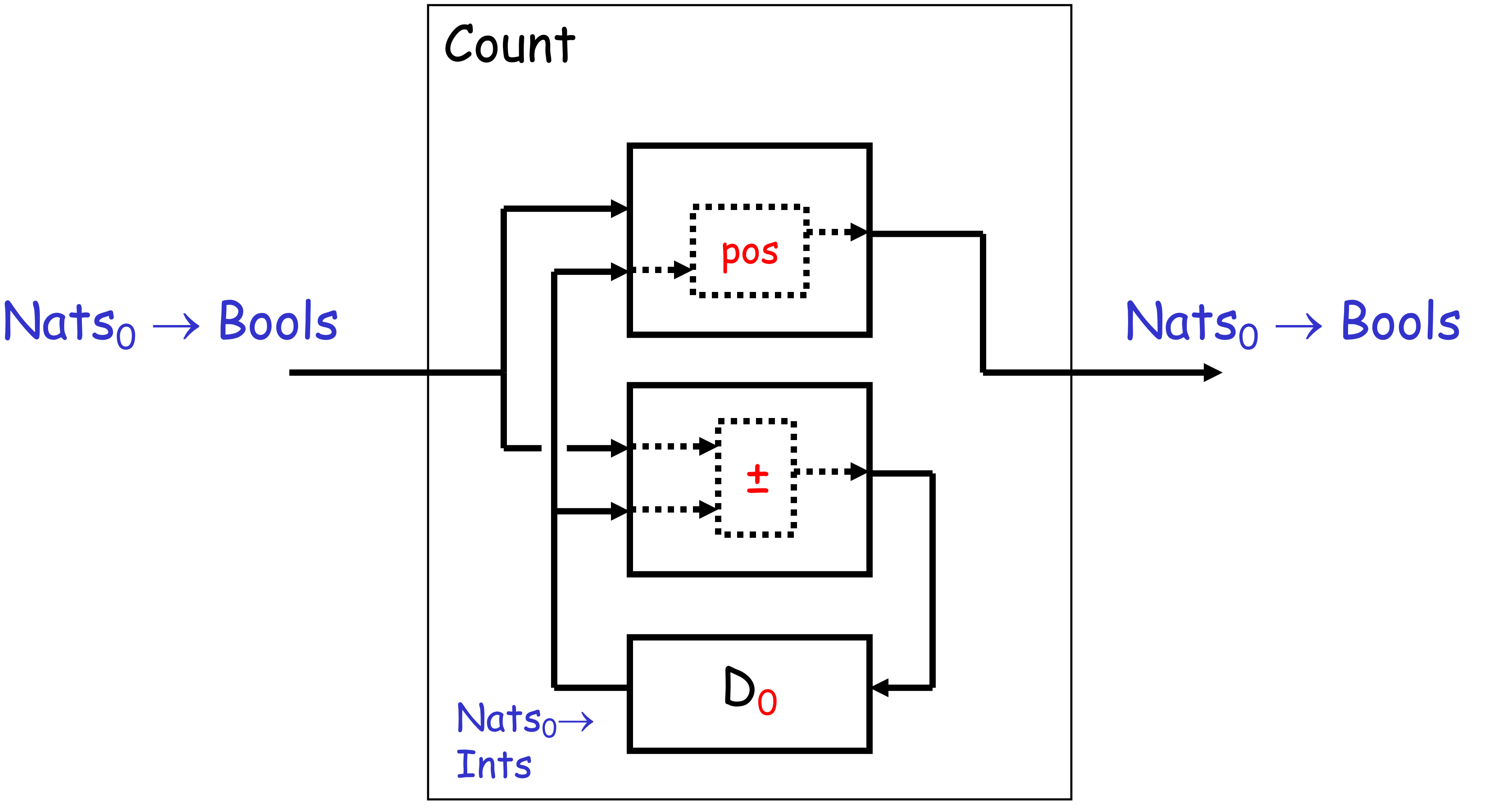
such that $\forall\, q \in \text{States},\ \forall\, x \in \text{Inputs},$

$$\text{nextState}\,(q,x) = \pm\,(x,q)$$

$$\text{output} : \text{States} \times \text{Inputs} \to \text{Outputs}$$

such that $\forall\, q \in \text{States},\ \forall\, x \in \text{Inputs},$

$$\text{output}\,(q,x) = \text{pos}\,(q)$$


Count
pos
±
D0
Nats0 → Bools
Nats0 → Bools
Nats0→
Ints

A State Machine

Inputs	(set of possible input values)
Outputs	(set of possible output values)
States	(set of states)
initialState	$\in$ States
update	$:$ States $\times$ Inputs $\to$ States $\times$ Outputs

State-Machine Implementation of the Parity System

$Inputs = Bools$

$Outputs = Bools$

$States = Bools$

$initialState = true$

$update : States \times Inputs \rightarrow States \times Outputs$

such that $\forall q \in States, \forall x \in Inputs,$

$$update(q,x)_1 = (q \neq x)$$

$$update(q,x)_2 = q$$

State-Machine Implementation of the Count System

$$Inputs = Bools$$

$$Outputs = Bools$$

$$States = Ints$$

$$initialState = 0$$

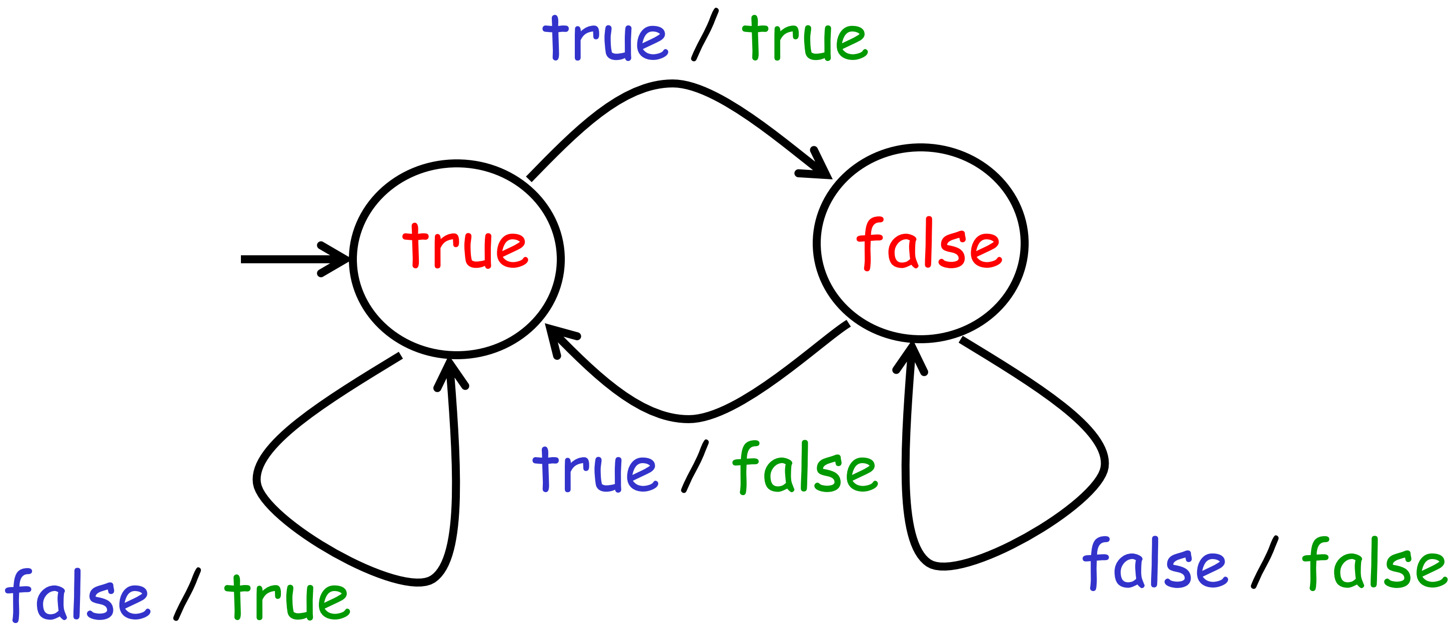
$$update : States \times Inputs \rightarrow States \times Outputs$$

such that $\forall\, q \in States,\ \forall\, x \in Inputs,$

$$update\,(q,x)_1 = \pm\,(x,q)$$

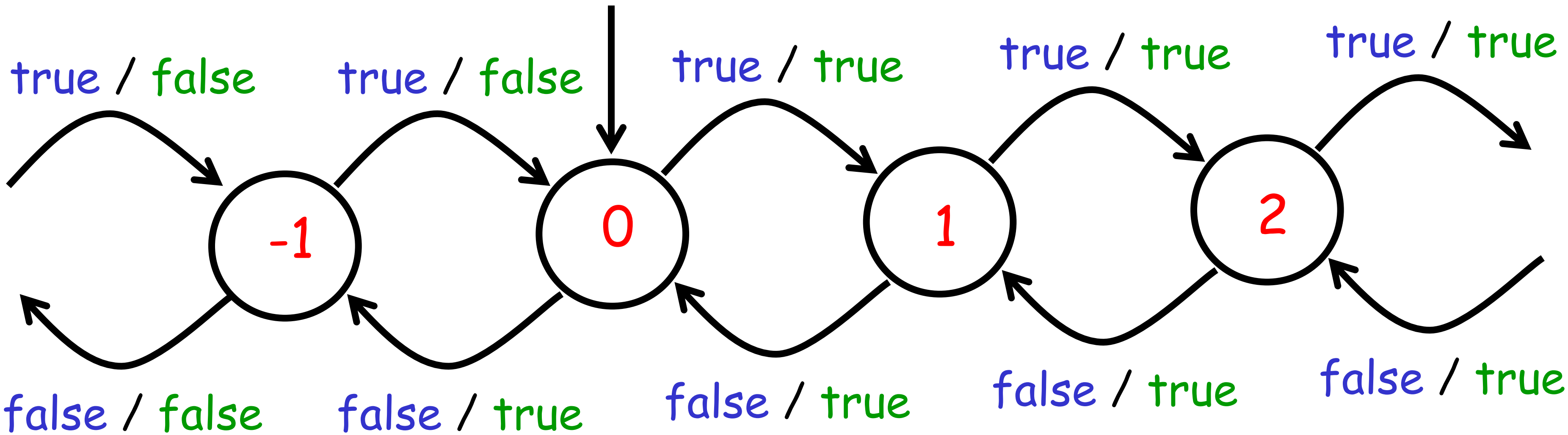
$$update\,(q,x)_2 = pos\,(q)$$

Transition Diagram of the Parity System



States = Bools
Inputs = Bools
Outputs = Bools

Transition Diagram of the Count System



States = Ints
Inputs = Bools
Outputs = Bools

Transition Diagram

Graph:

Nodes = states

Edges = transitions

Determinism : for every state and input, at most one outgoing edge

Receptiveness : for every state and input, at least one outgoing edge

(because "update" is a function)

Exercise

Draw the transition diagram of a state machine with

Inputs = Bools

Outputs = Bools

At all times t, the output is true iff the inputs at times t-2, t-1, and t are all true .

State after i-th input :

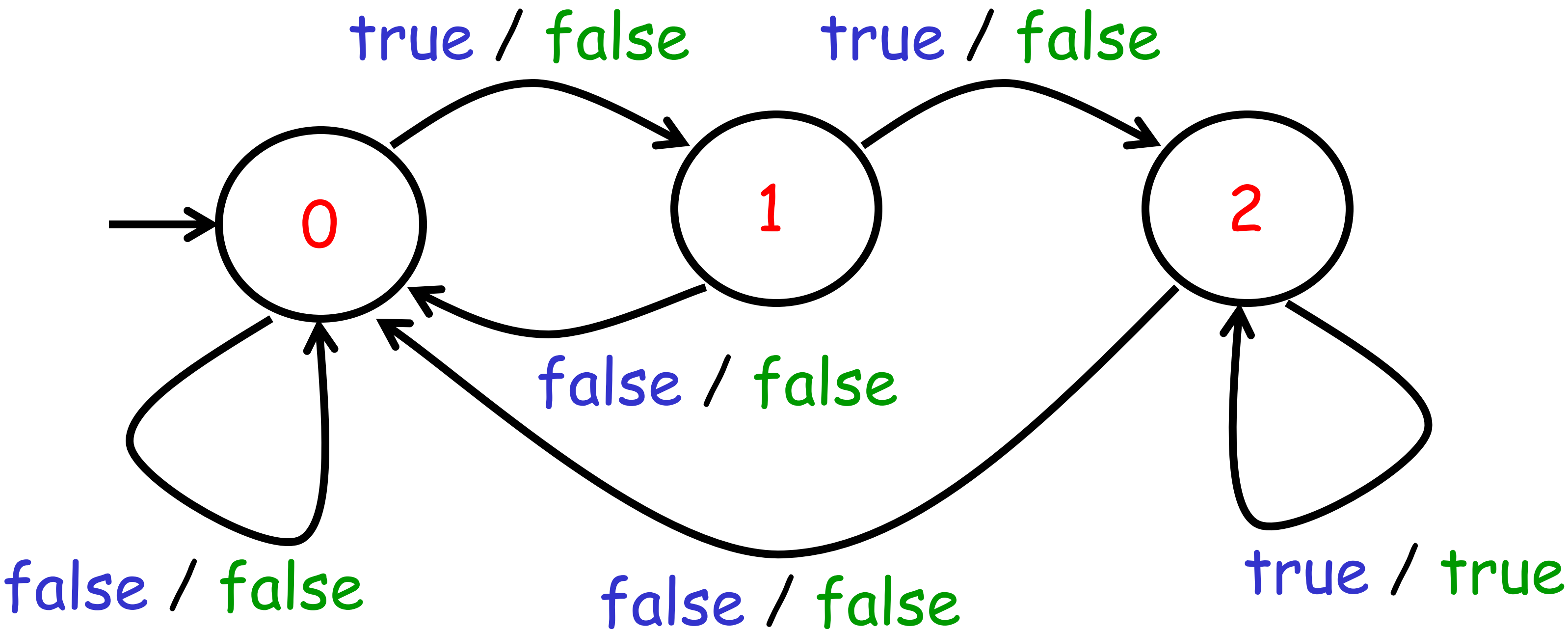
0, if i-th input is false (or i = 0)

1, if i-th input is true and (i-1)-st input is false (or i = 1)

2, if both i-th and (i-1)-st inputs are true

Three states

Transition Diagram



States = { 0, 1, 2 }
Inputs = Bools
Outputs = Bools

The Parity System :

$$\text{States}[\text{Parity}] = \{\text{true}, \text{false}\}$$

$$\text{initialState}[\text{Parity}] = \text{true}$$

$$\text{nextState}[\text{Parity}](q,x) = (q \neq x)$$

$$\text{output}[\text{Parity}](q,x) = q$$

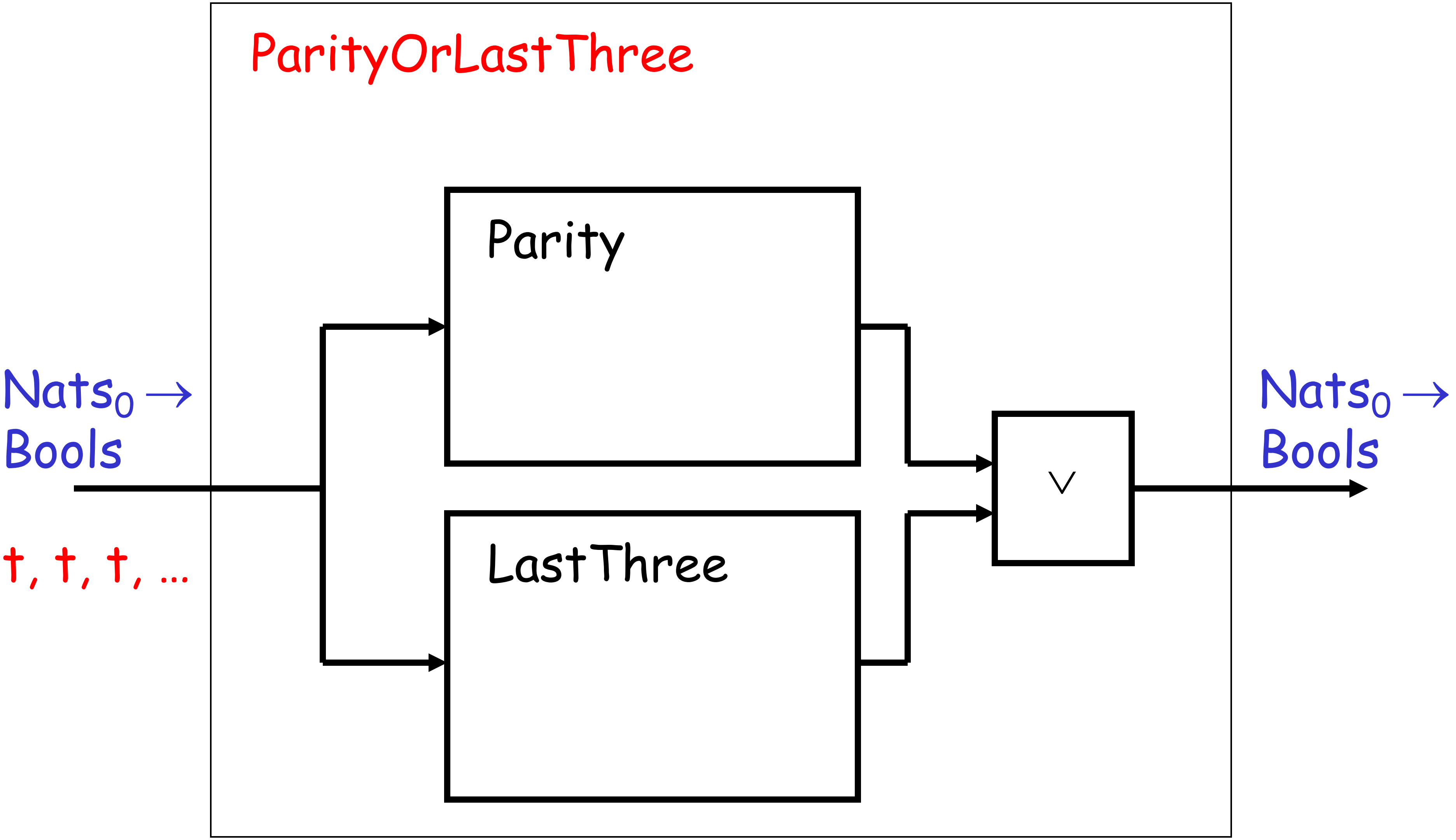
The LastThree System :

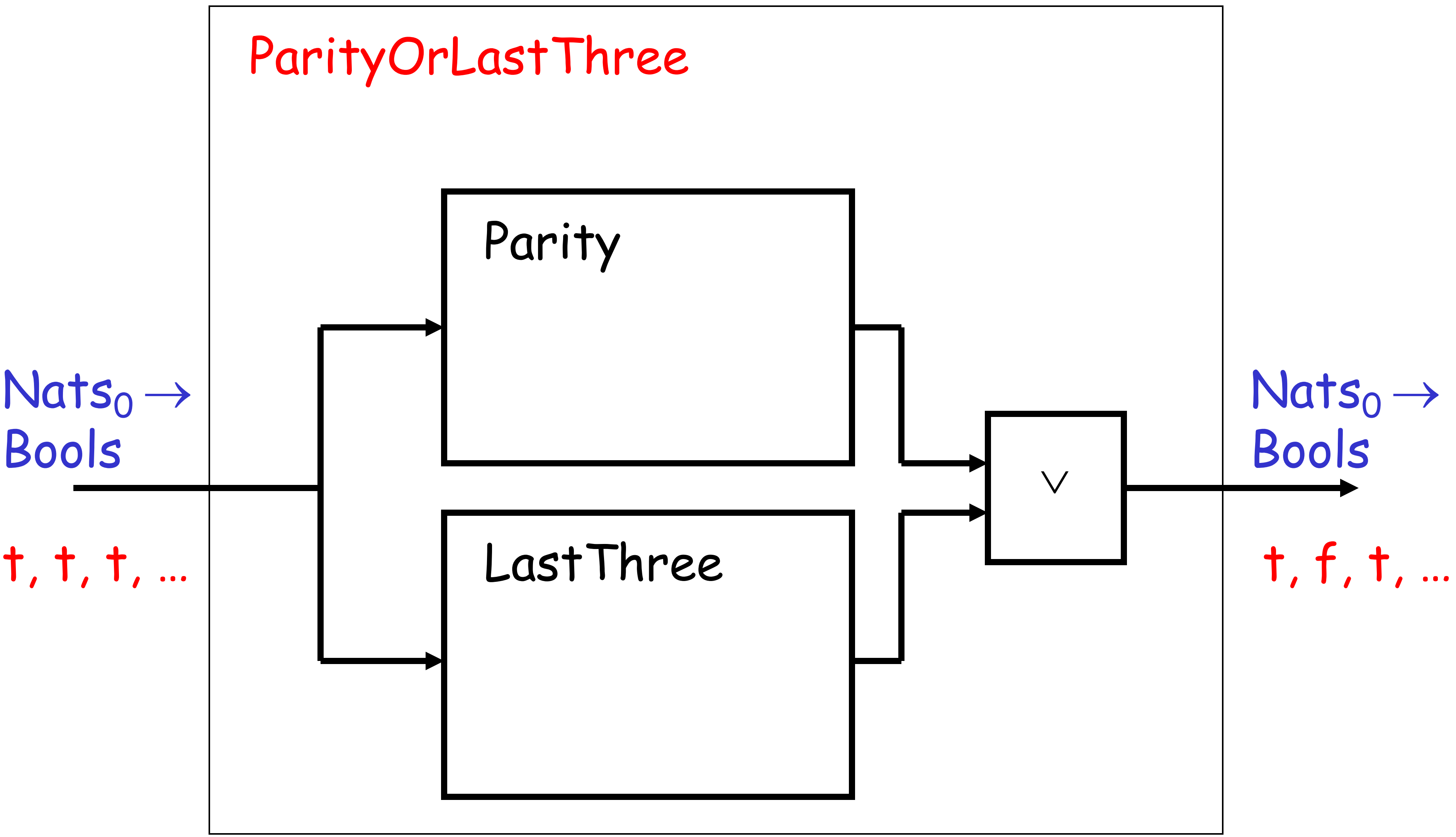
$$\text{States}[\text{LastThree}] = \{0, 1, 2\}$$

$$\text{initialState}[\text{LastThree}] = 0$$

$$\text{nextState}[\text{LastThree}](q,x) = \begin{cases} 0 & \text{if } \neg x \\ \min(q+1, 2) & \text{if } x \end{cases}$$

$$\text{output}[\text{LastThree}](q,x) = ((q = 2) \wedge x)$$


ParityOrLastThree
Parity
LastThree
$\vee$
$Nats_0 \to$ Bools
t, t, t, ...
$Nats_0 \to$ Bools

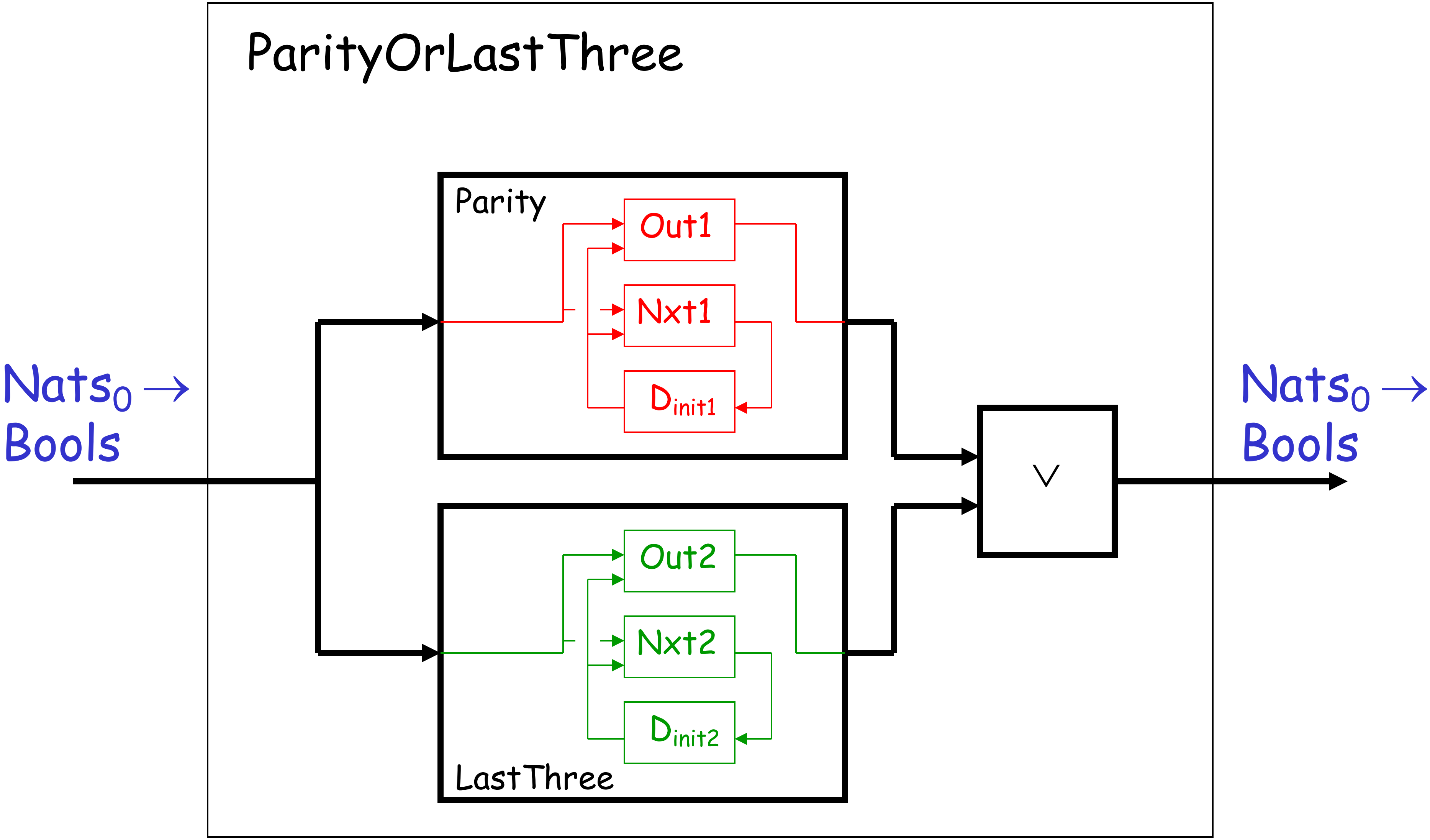

ParityOrLastThree
Parity
LastThree
∨
$Nats_0 \rightarrow Bools$
t, t, t, ...
$Nats_0 \rightarrow Bools$
t, f, t, ...

What is the state space of ParityOrLastThree ?

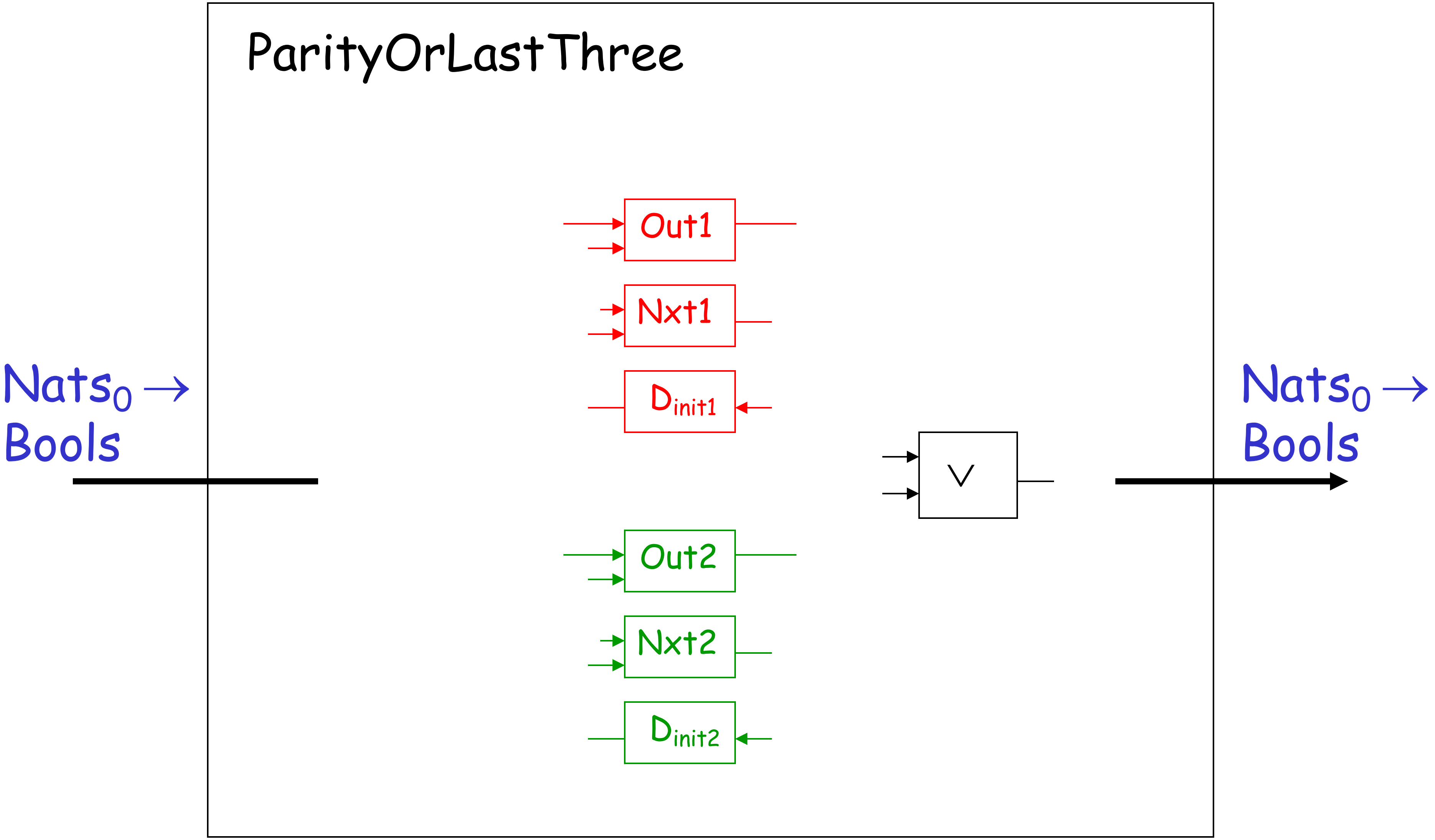
What is the initial state of ParityOrLastThree ?

What is the nextState function of ParityOrLastThree ?

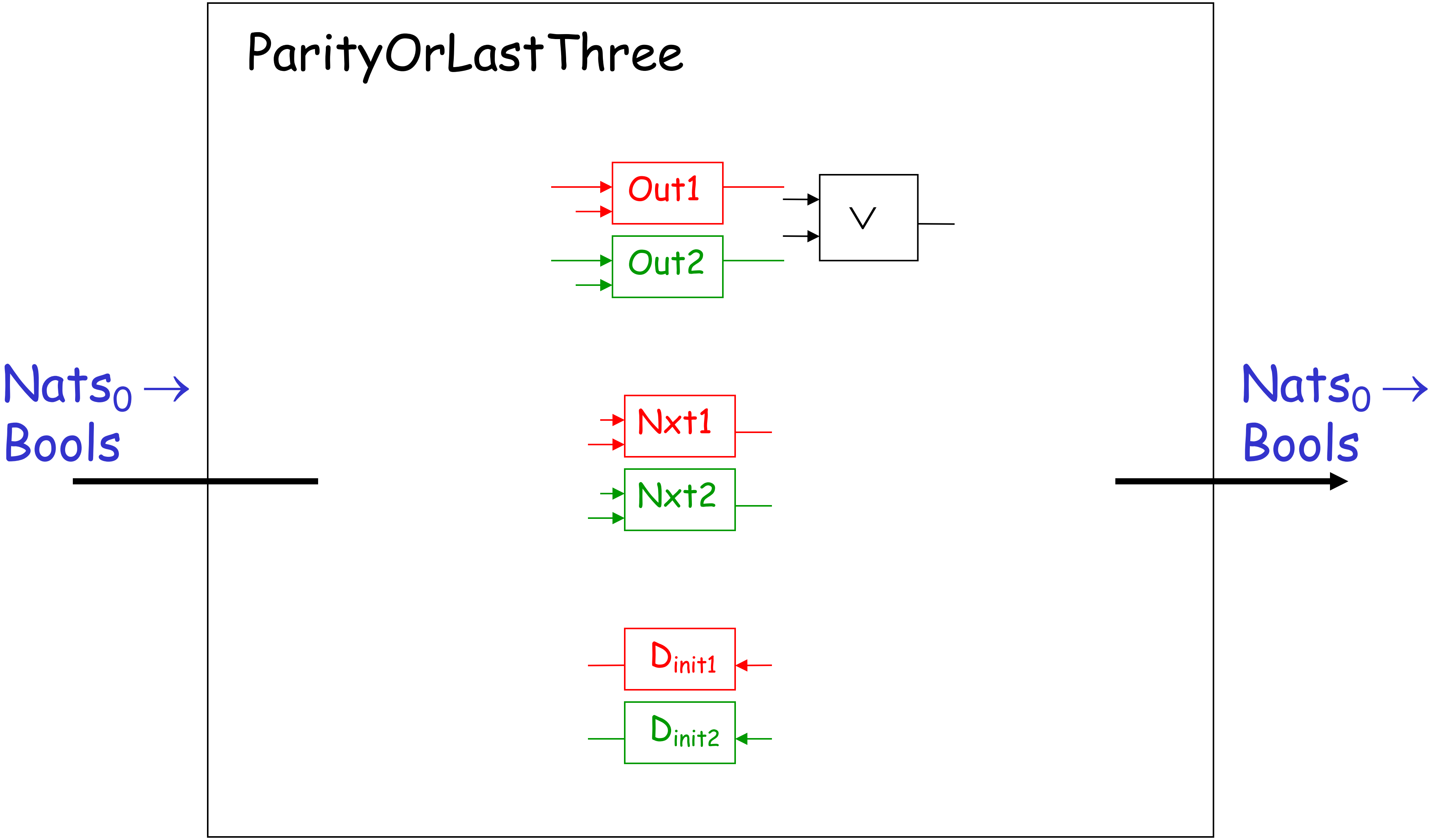
What is the output function of ParityOrLastThree ?

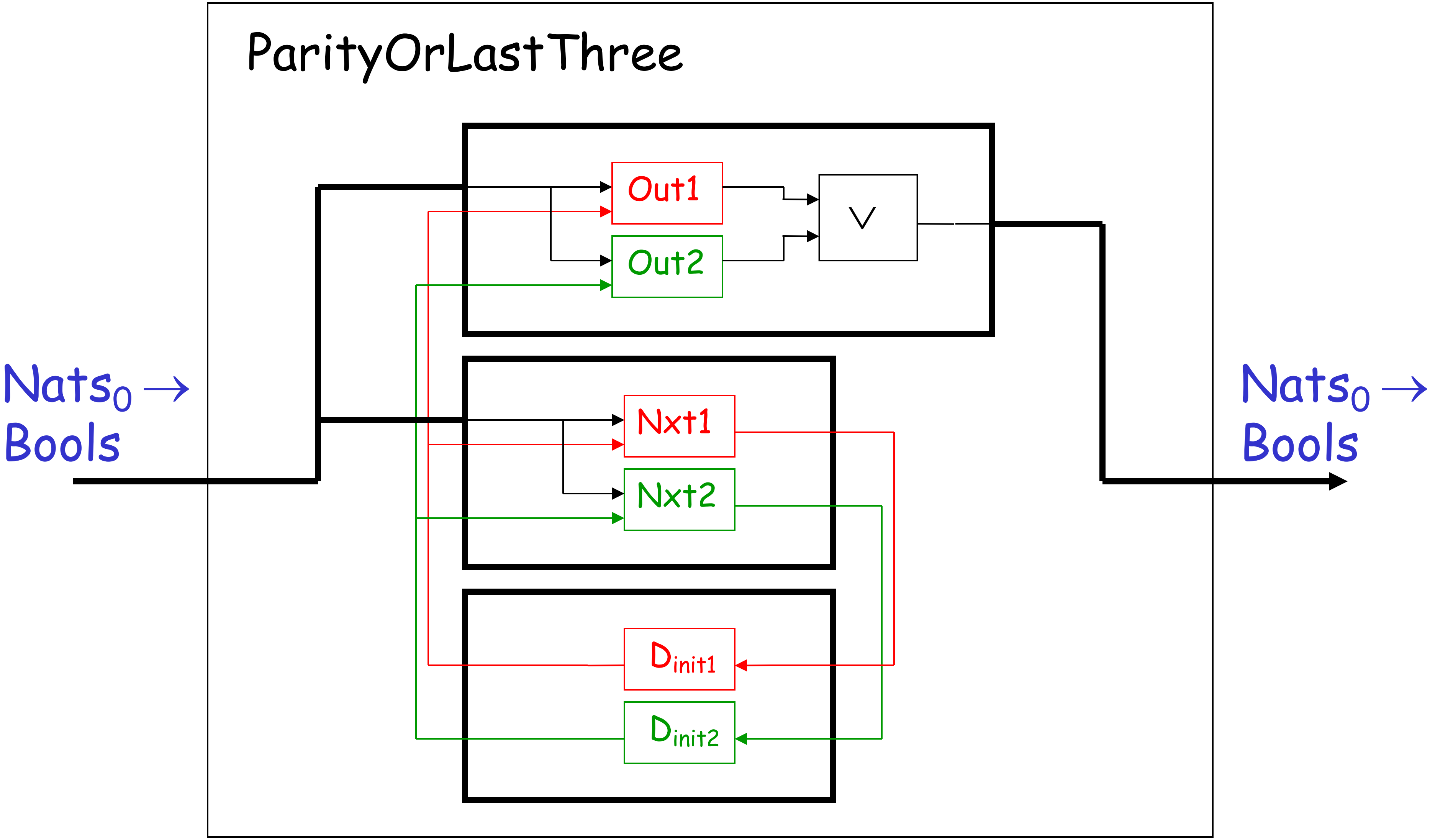


ParityOrLastThree
Parity
Out1
Nxt1
D_{init1}
LastThree
Out2
Nxt2
D_{init2}
$\vee$
$Nats_0 \to Bools$
$Nats_0 \to Bools$

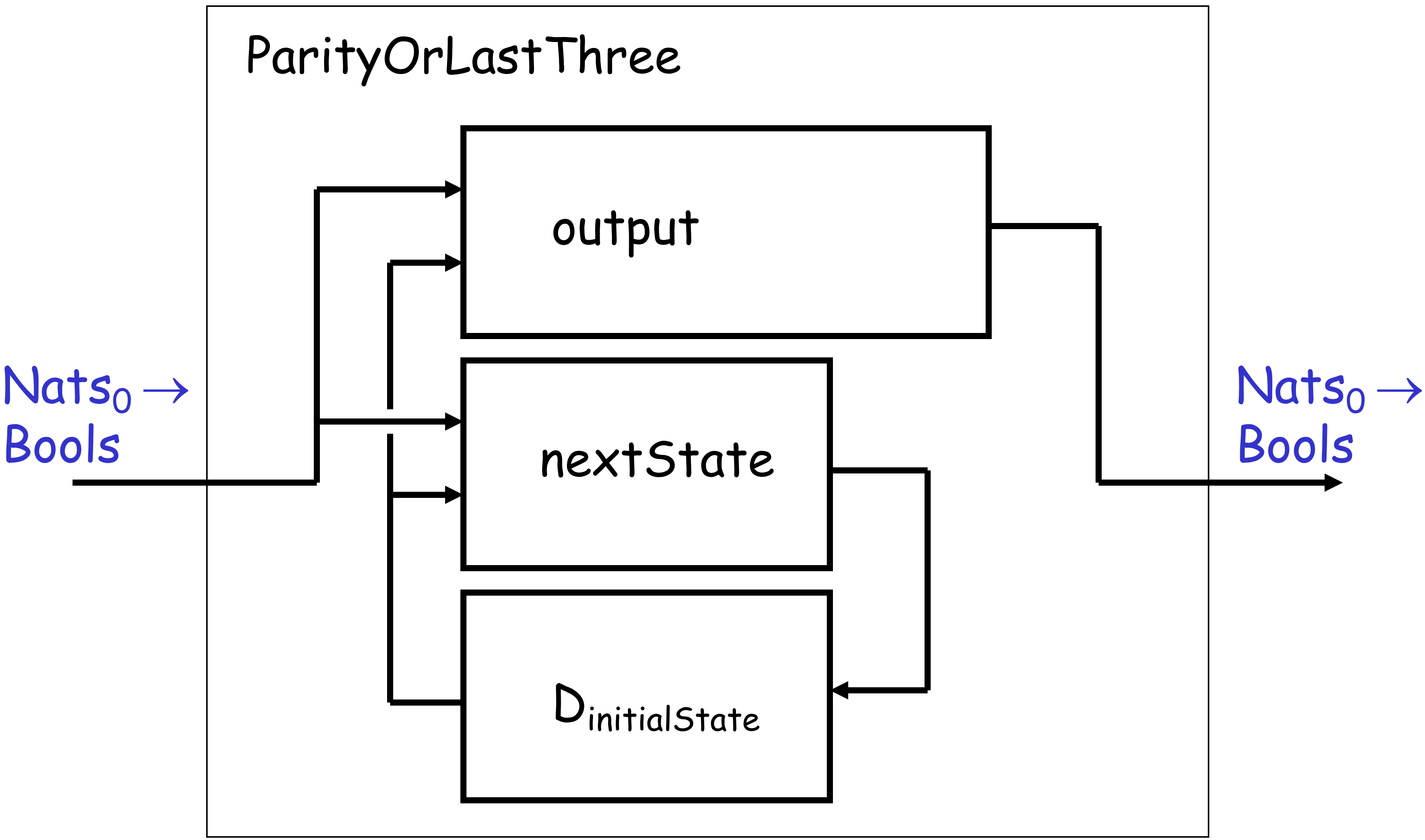


ParityOrLastThree
Out1
Nxt1
D_{init1}
$\vee$
Out2
Nxt2
D_{init2}
$Nats_0 \rightarrow$ Bools
$Nats_0 \rightarrow$ Bools


ParityOrLastThree
Out1
Out2
∨
Nxt1
Nxt2
D_{init1}
D_{init2}
$Nats_0 \to$ Bools
$Nats_0 \to$ Bools



ParityOrLastThree
Nats0 → Bools
Out1
Out2
∨
Nxt1
Nxt2
Dinit1
Dinit2
Nats0 → Bools



ParityOrLastThree
output
nextState
$D_{initialState}$
$Nats_0 \to$ Bools
$Nats_0 \to$ Bools

The ParityOrLastThree System

$Inputs$ [ParityOrLastThree] = Bools

$Outputs$ [ParityOrLastThree] = Bools

$States$ [ParityOrLastThree]

= States [Parity] $\times$ States [LastThree]

= { true, false } $\times$ { 0, 1, 2 }

initialState [ParityOrLastThree]

= (initialState [Parity], initialState [LastThree])

= (true, 0)

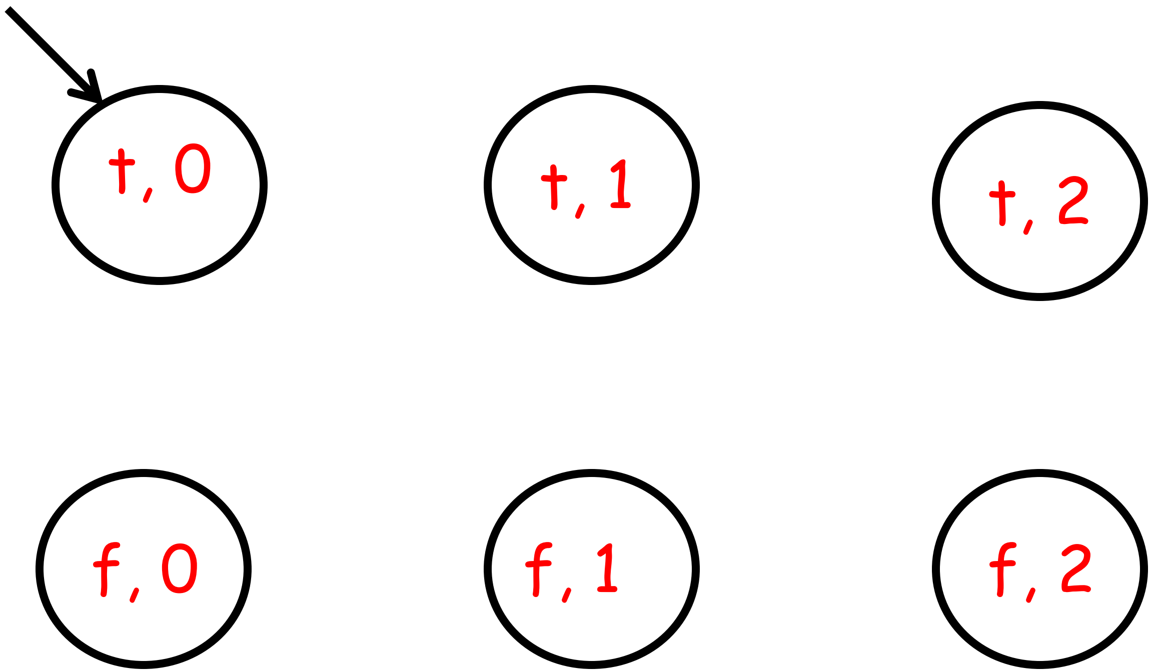
The ParityOrLastThree System, continued

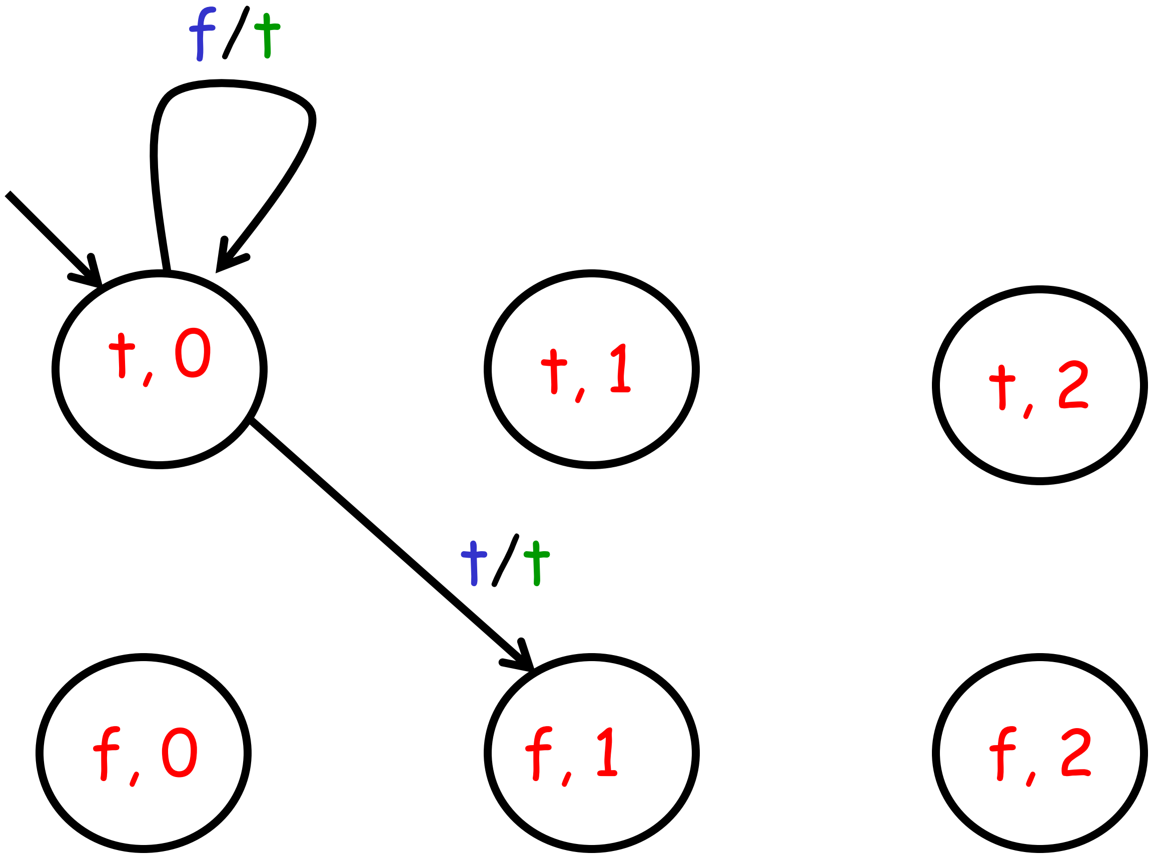
$$\text{nextState}[\text{ParityOrLastThree}]((q1, q2), x)$$

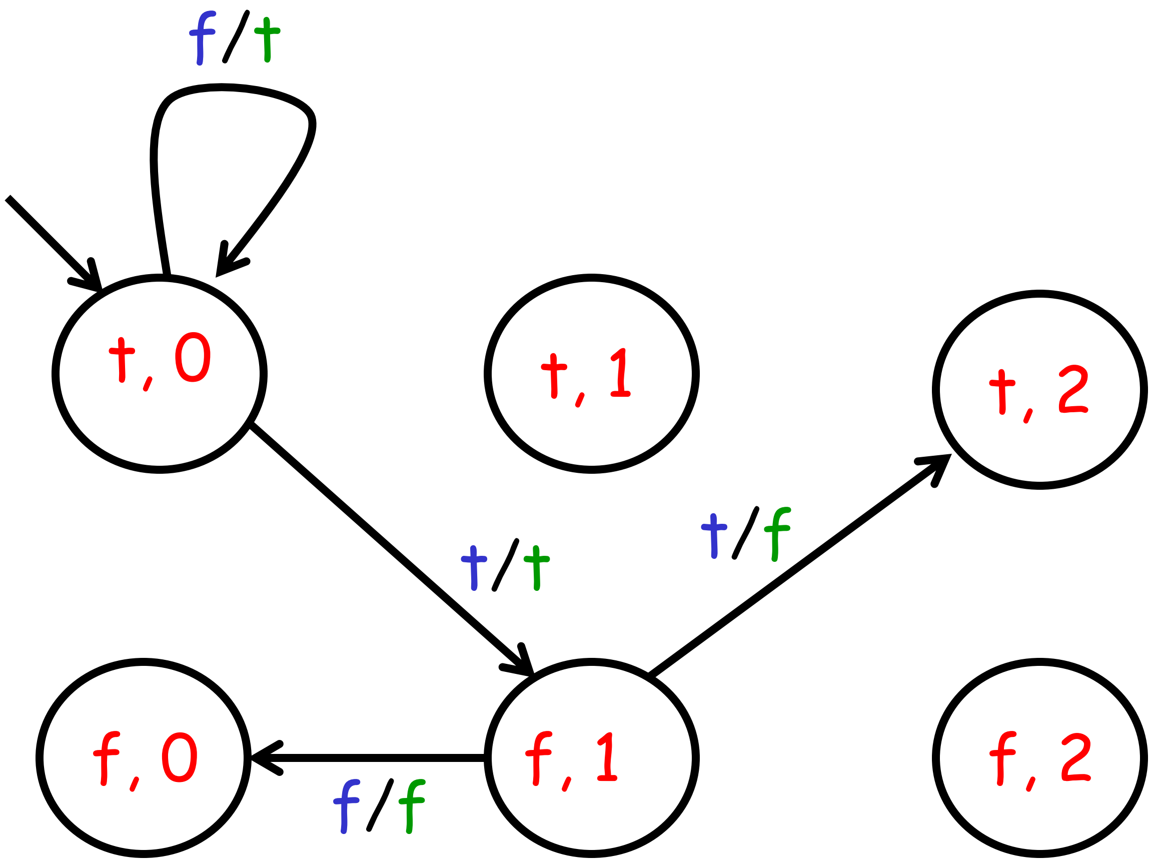
$$= (\text{nextState}[\text{Parity}](q1, x), \text{nextState}[\text{LastThree}](q2, x))$$

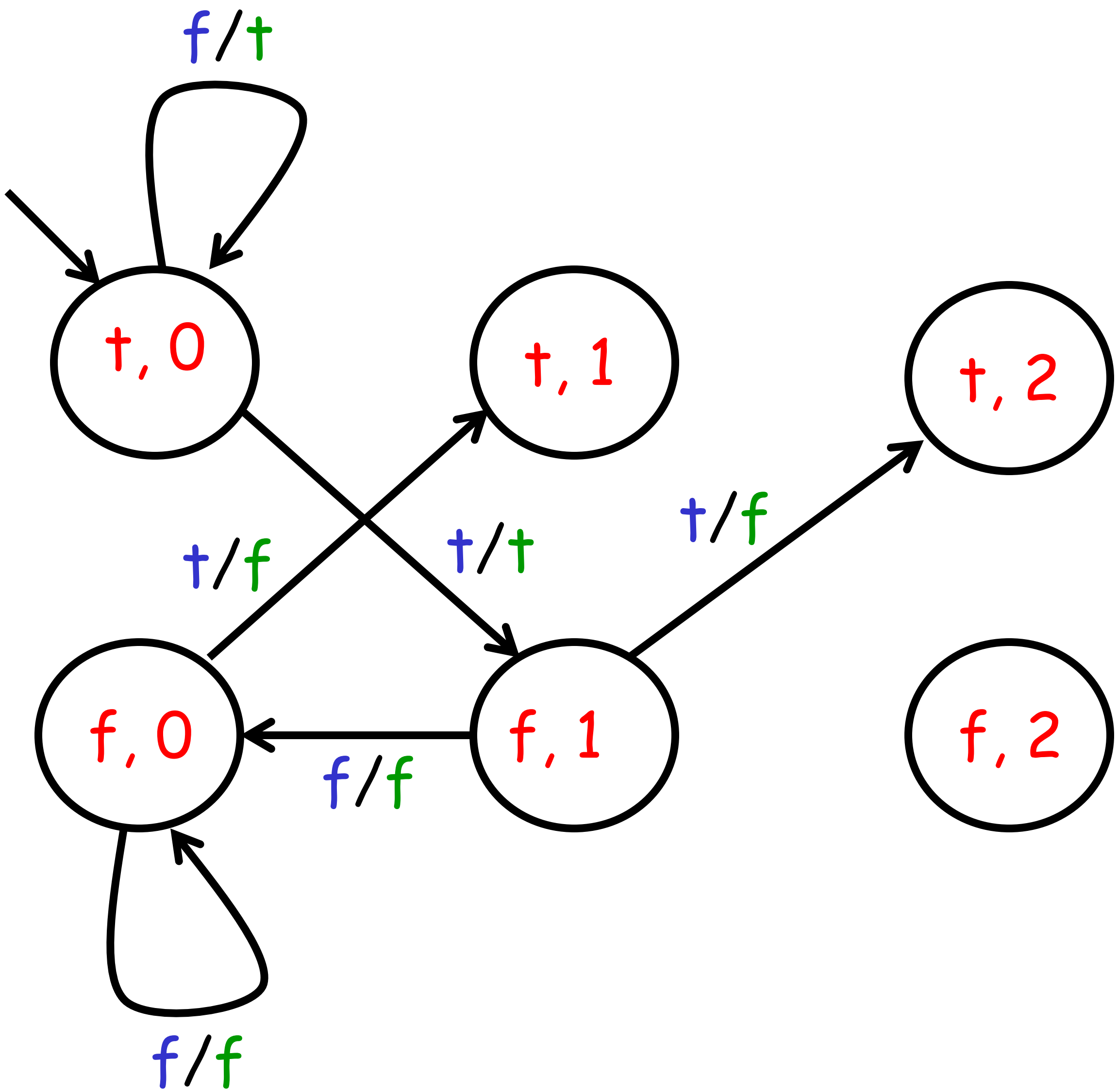
$$\text{output}[\text{ParityOrLastThree}]((q1, q2), x)$$

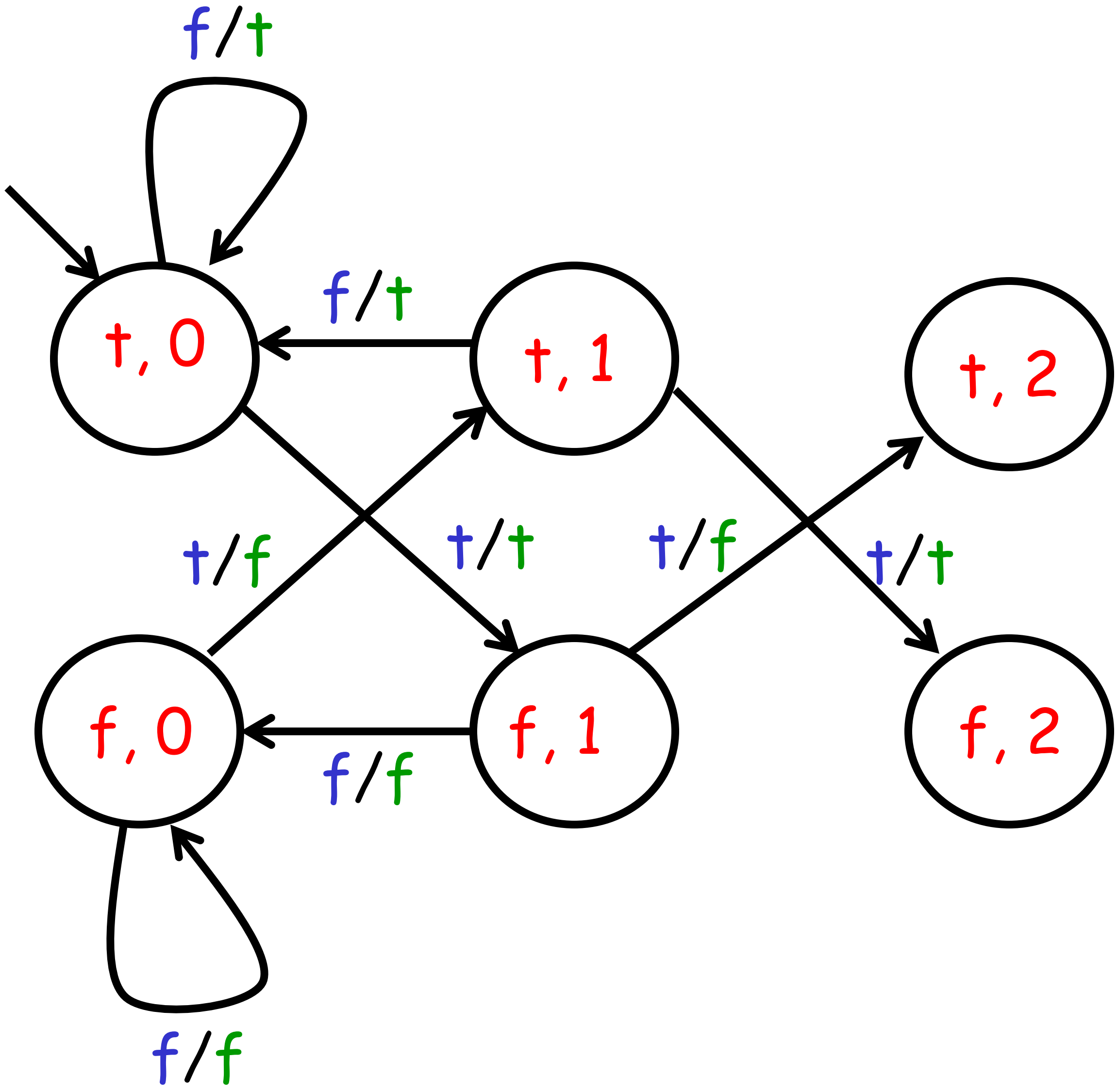
$$= \text{output}[\text{Parity}](q1, x) \vee \text{output}[\text{LastThree}](q2, x)$$


t, 0
t, 1
t, 2
f, 0
f, 1
f, 2

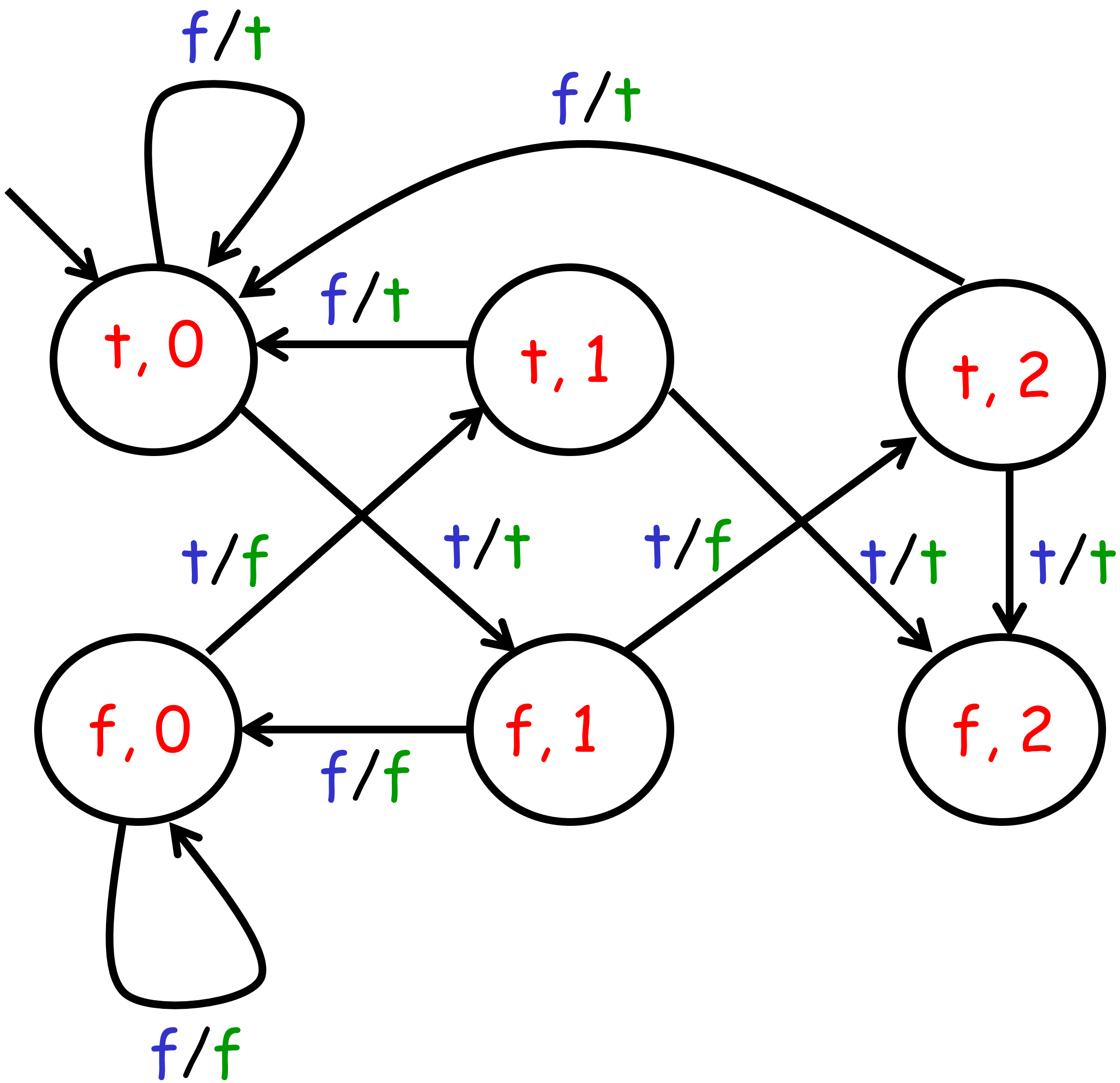

f/t
t, 0
t, 1
t, 2
t/t
f, 0
f, 1
f, 2

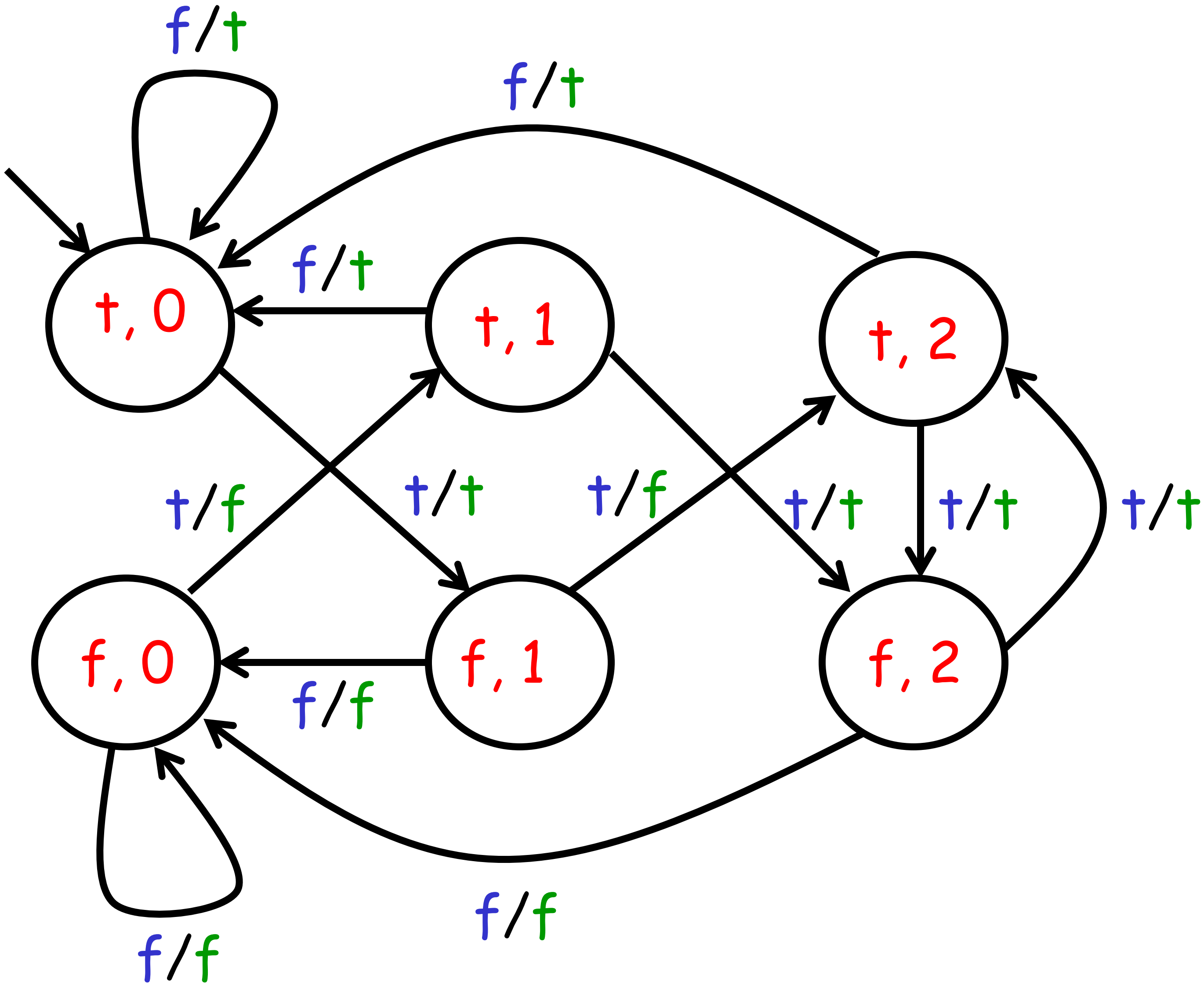

f/t
t, 0
t, 1
t, 2
t/t
t/f
f, 0
f, 1
f, 2
f/f


f/t
t, 0
t, 1
t, 2
t/f
t/t
t/f
f, 0
f, 1
f, 2
f/f
f/f



f/t
t, 0
f/t
t, 1
t, 2
t/f
t/t
t/f
t/t
f, 0
f, 1
f, 2
f/f
f/f


f/t
f/t
f/t
t, 0
t, 1
t, 2
t/f
t/t
t/f
t/t
t/t
f, 0
f, 1
f, 2
f/f
f/f



f/t
f/t
f/t
t, 0
t, 1
t, 2
t/f
t/t
t/f
t/t
t/t
t/t
f, 0
f, 1
f, 2
f/f
f/f
f/f