Product Machines

EECS 20

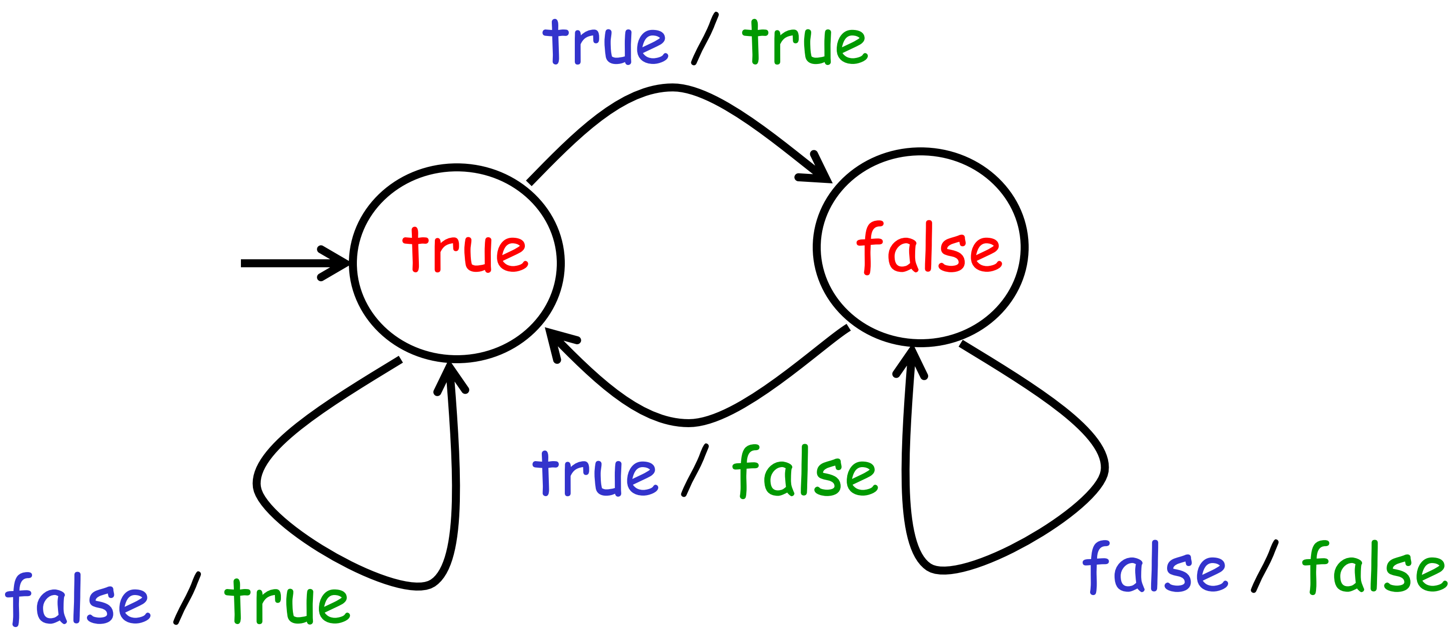
Lecture 10 (February 7, 2001)

Tom Henzinger

Revised by Matteo Zavatteri and Tiziano Villa, Fall 2020

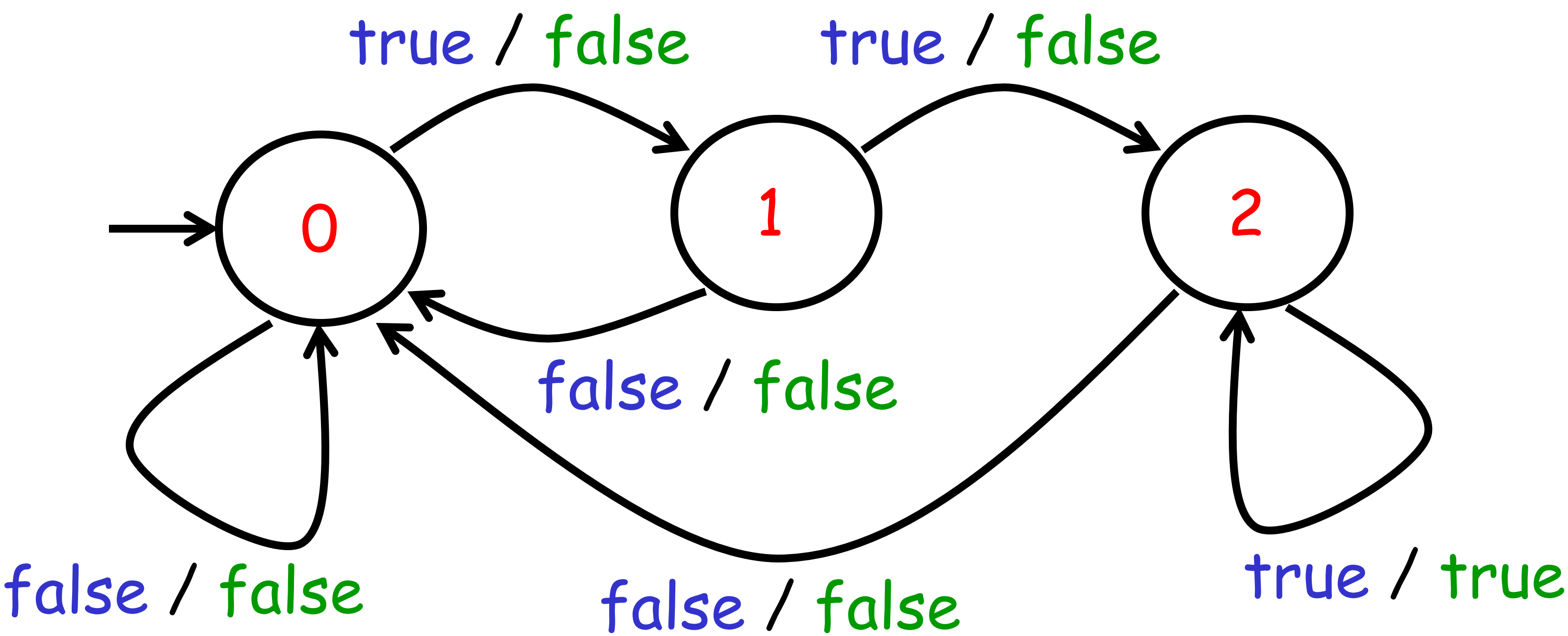
Composition of State Machines

Transition Diagram of the Parity System



States = Bools
Inputs = Bools
Outputs = Bools

Transition Diagram of the LastThree System



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 :

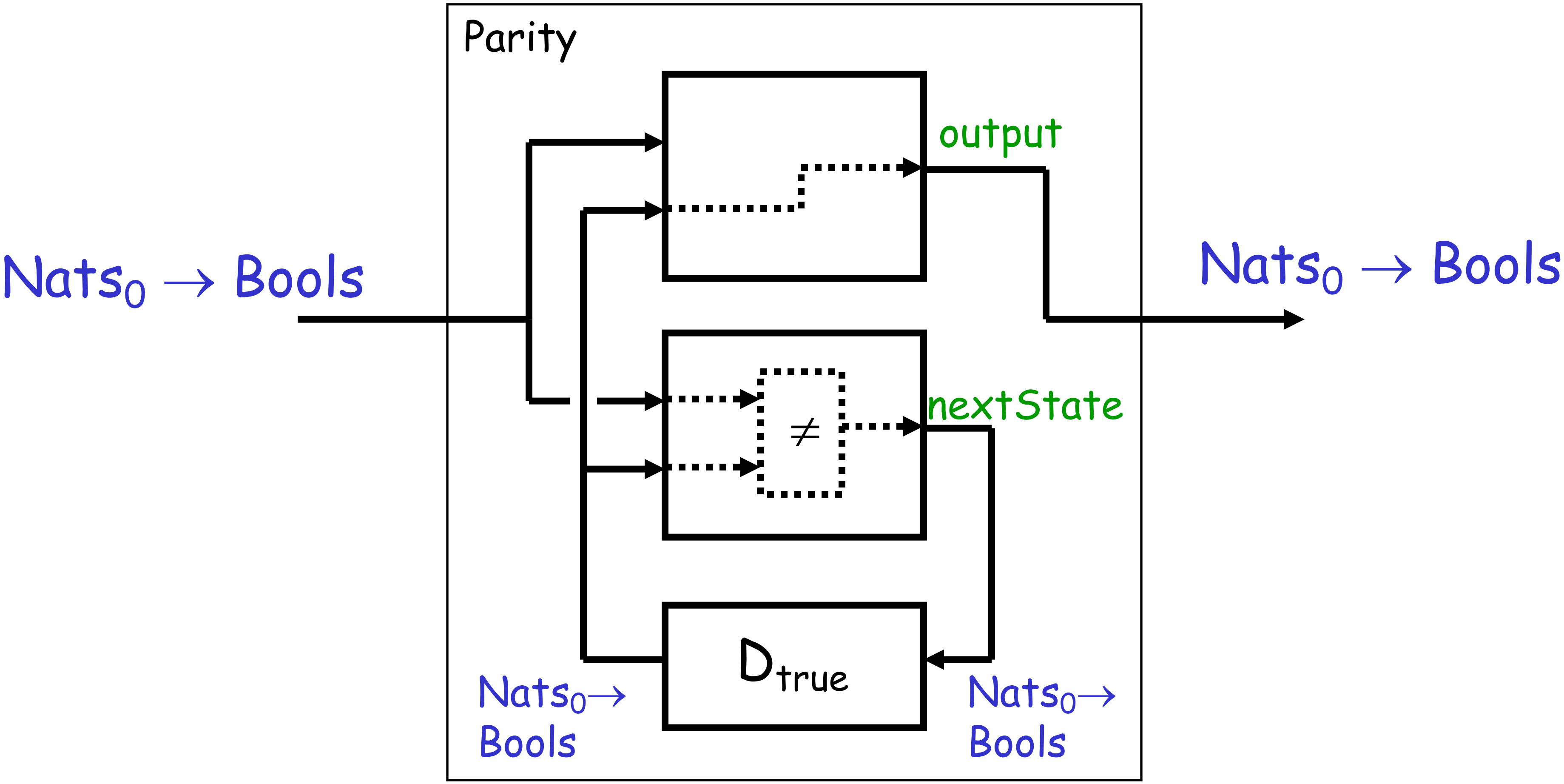
$$\text{States}[\text{LastThree}] = \{0, 1, 2\}$$

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

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

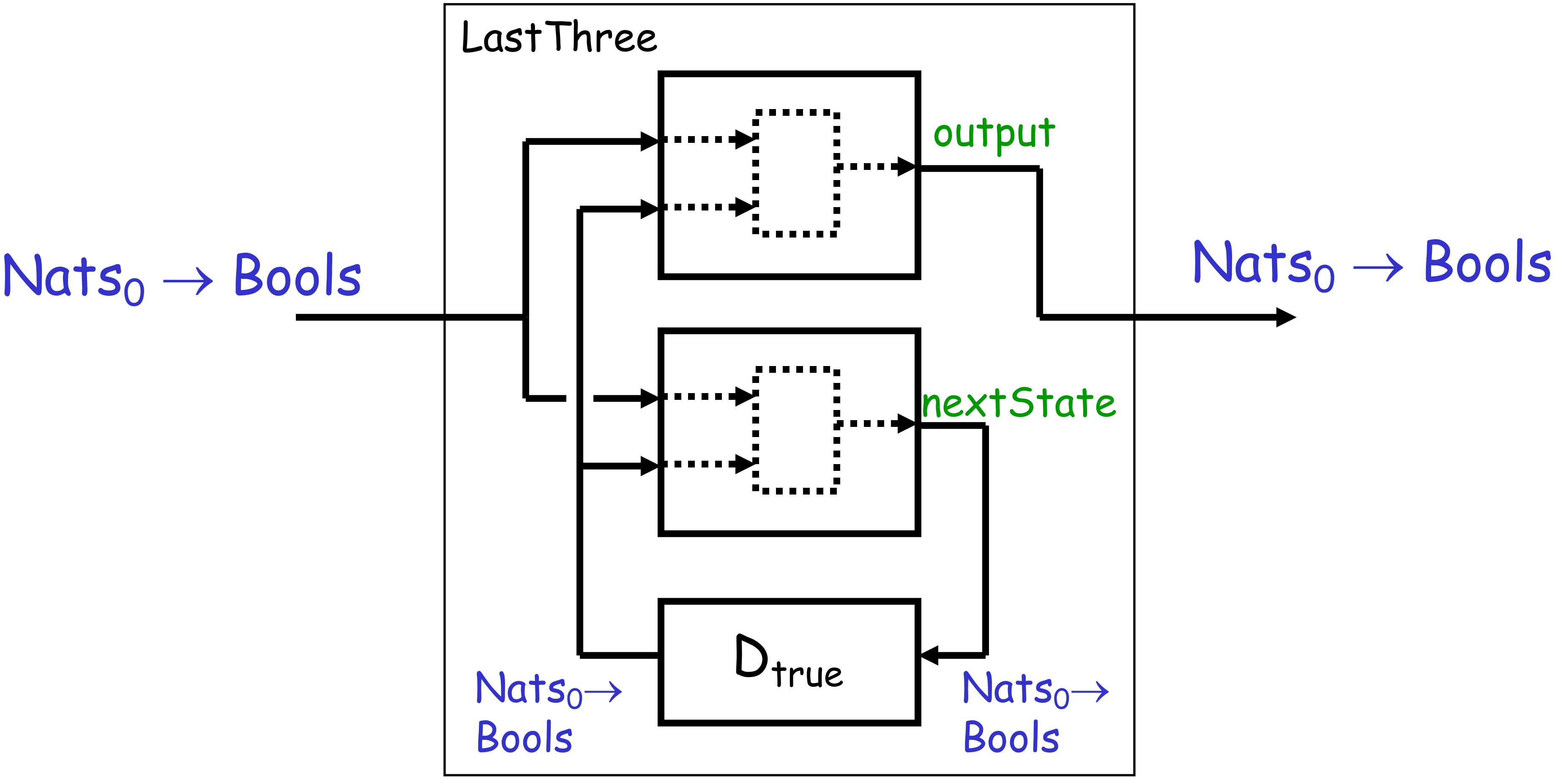
$$\text{output}[\text{LastThree}](q,x) = ((q = 2) \wedge x)$$

Block Diagram of Parity System

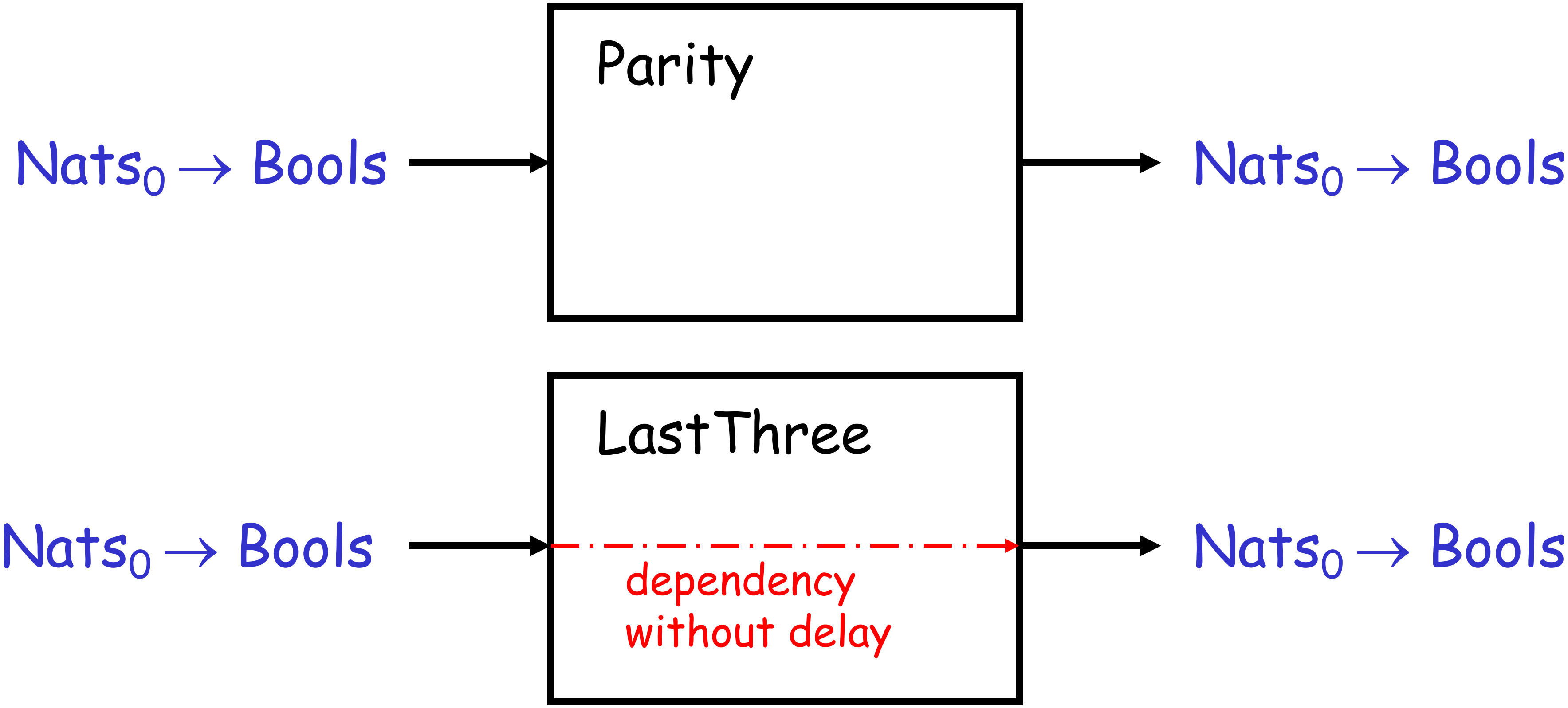


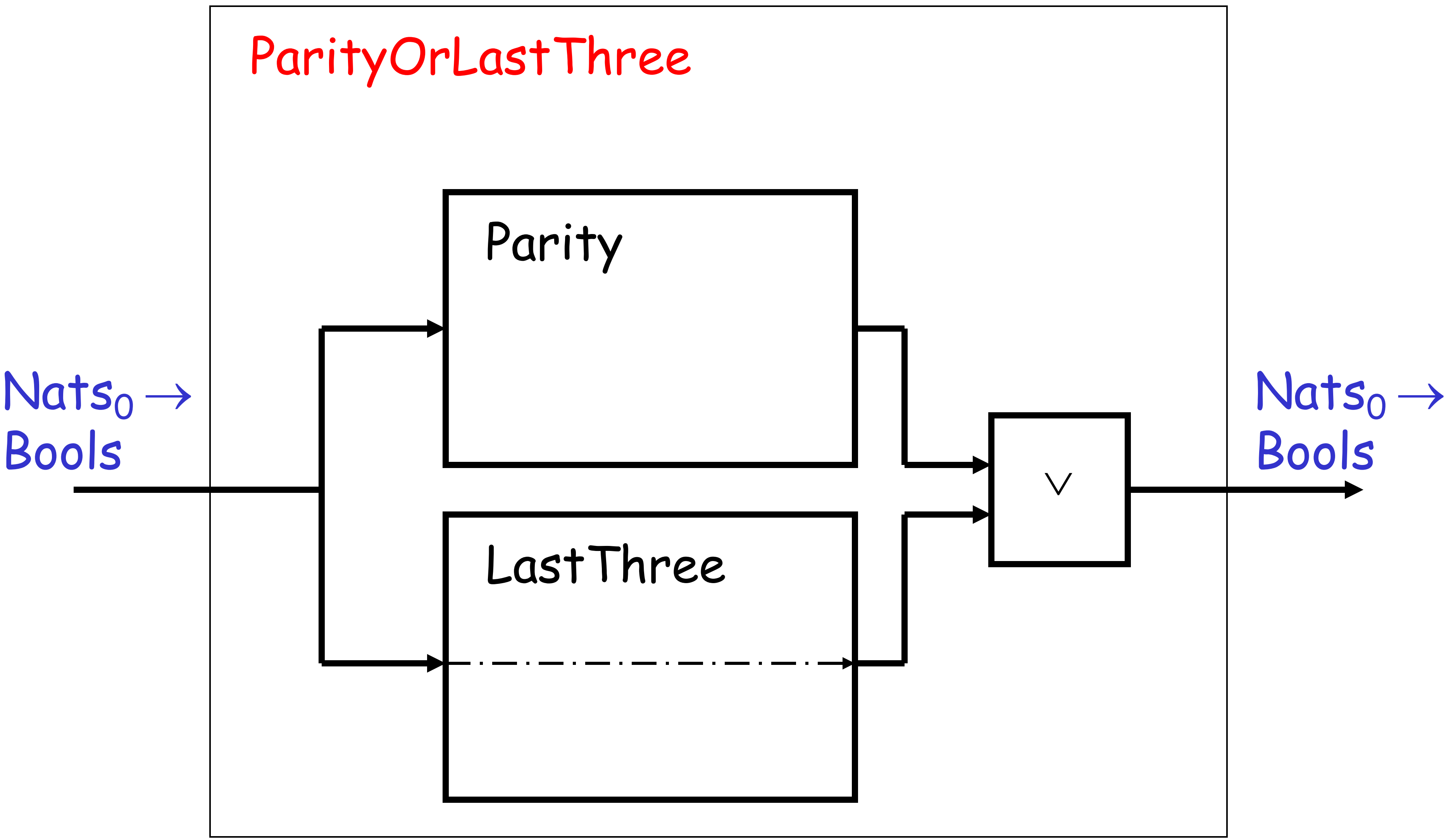
No path without delay from input to output.

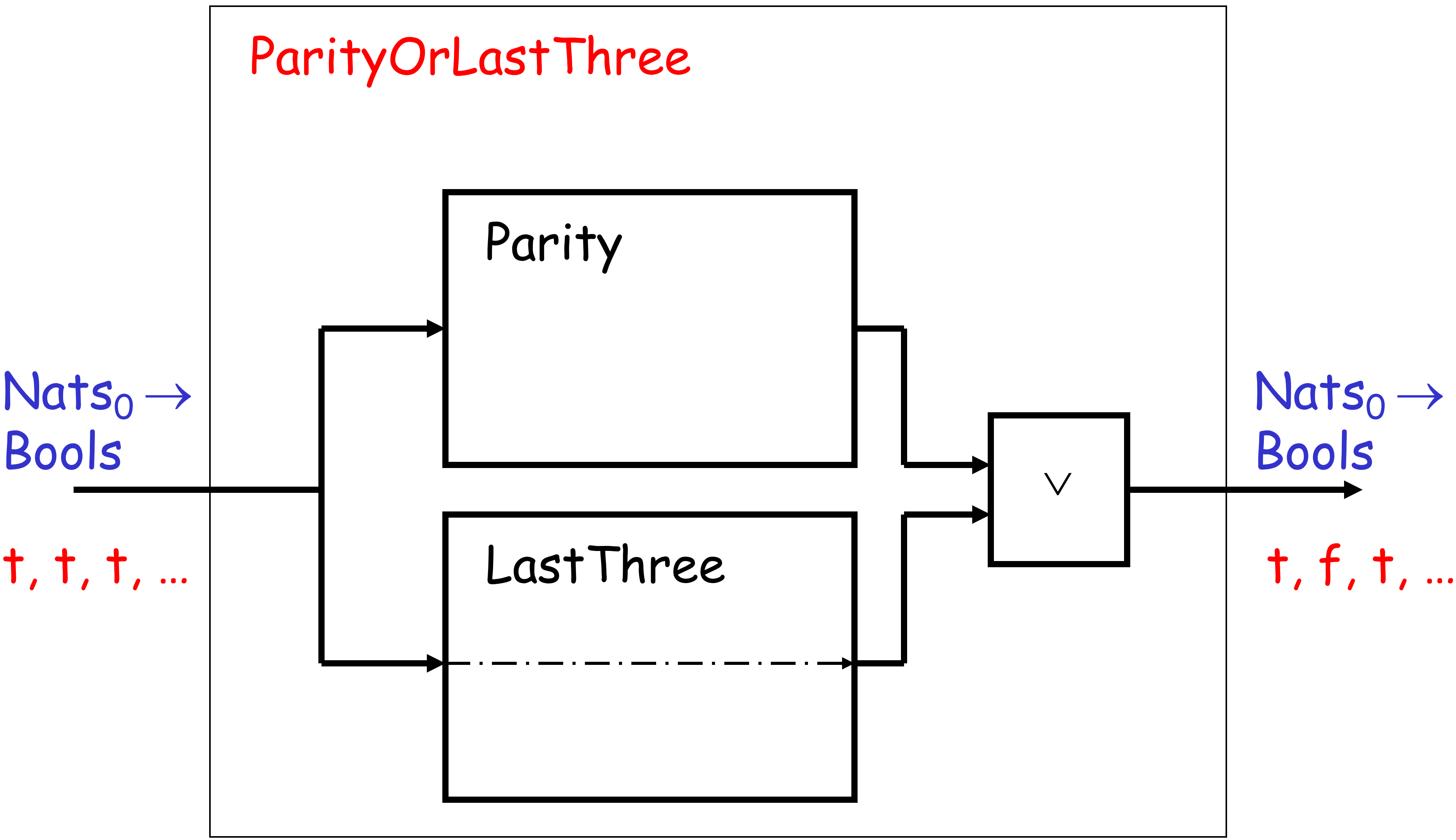
Block Diagram of LastThree System



Path without delay from input to output !


Parity
$Nats_0 \to Bools$
$Nats_0 \to Bools$
LastThree
$Nats_0 \to Bools$
$Nats_0 \to Bools$
dependency
without delay


ParityOrLastThree
Parity
LastThree
∨
$Nats_0 \to Bools$
$Nats_0 \to Bools$


ParityOrLastThree
Parity
LastThree
∨
Nats0 → Bools
t, t, t, ...
Nats0 → Bools
t, f, t, ...

The ParityOrLastThree System

Inputs [ParityOrLastThree] = Bools

Outputs [ParityOrLastThree] = Bools

States [ParityOrLastThree]

$= \text{States [Parity]} \times \text{States [LastThree]}$

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

initialState [ParityOrLastThree]

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

= (true, 0)

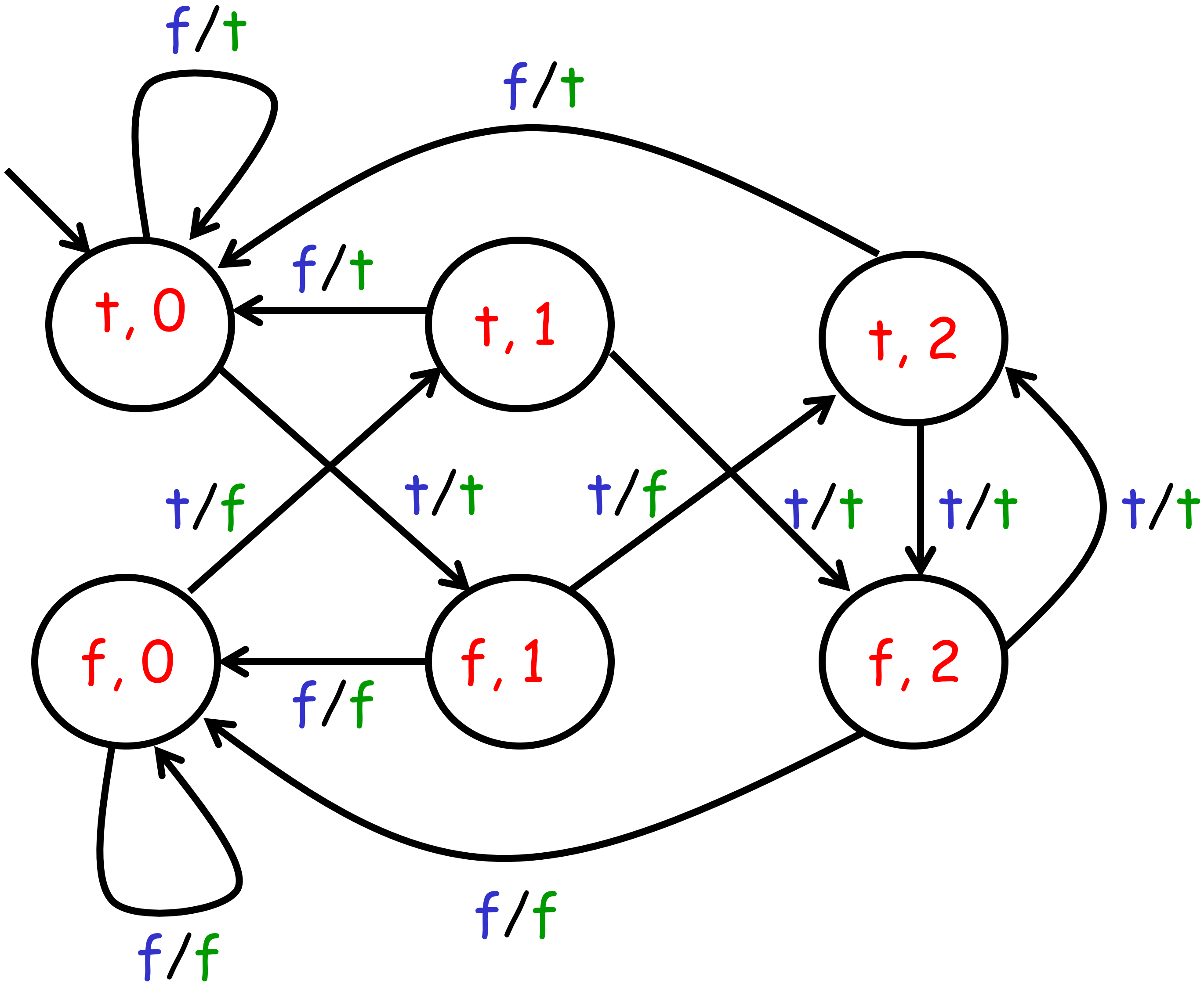
The ParityOrLastThree System, continued

nextState [ParityOrLastThree] ((q1, q2) , x)

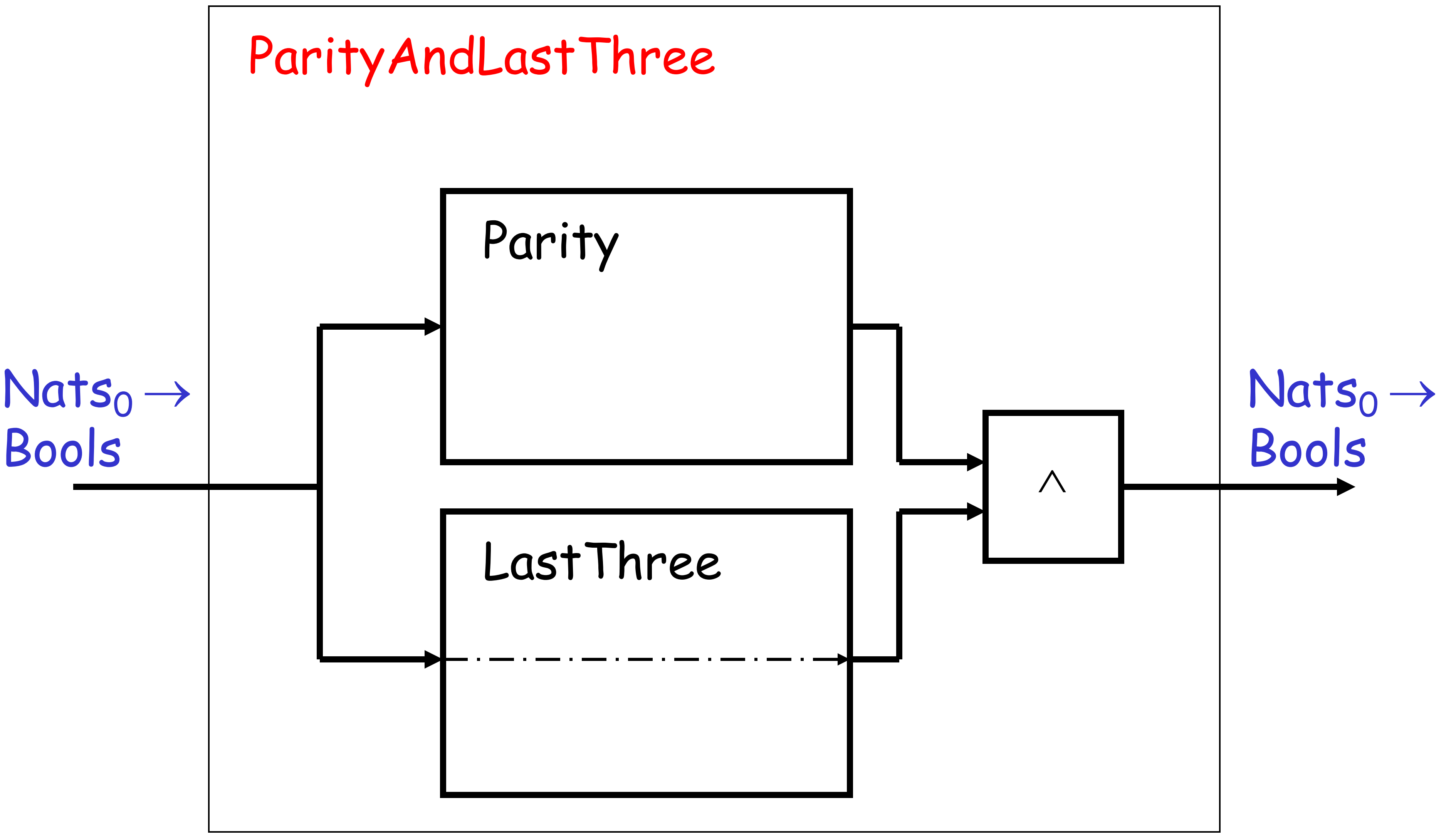
= (nextState [Parity] (q1, x) , nextState [LastThree] (q2, x))

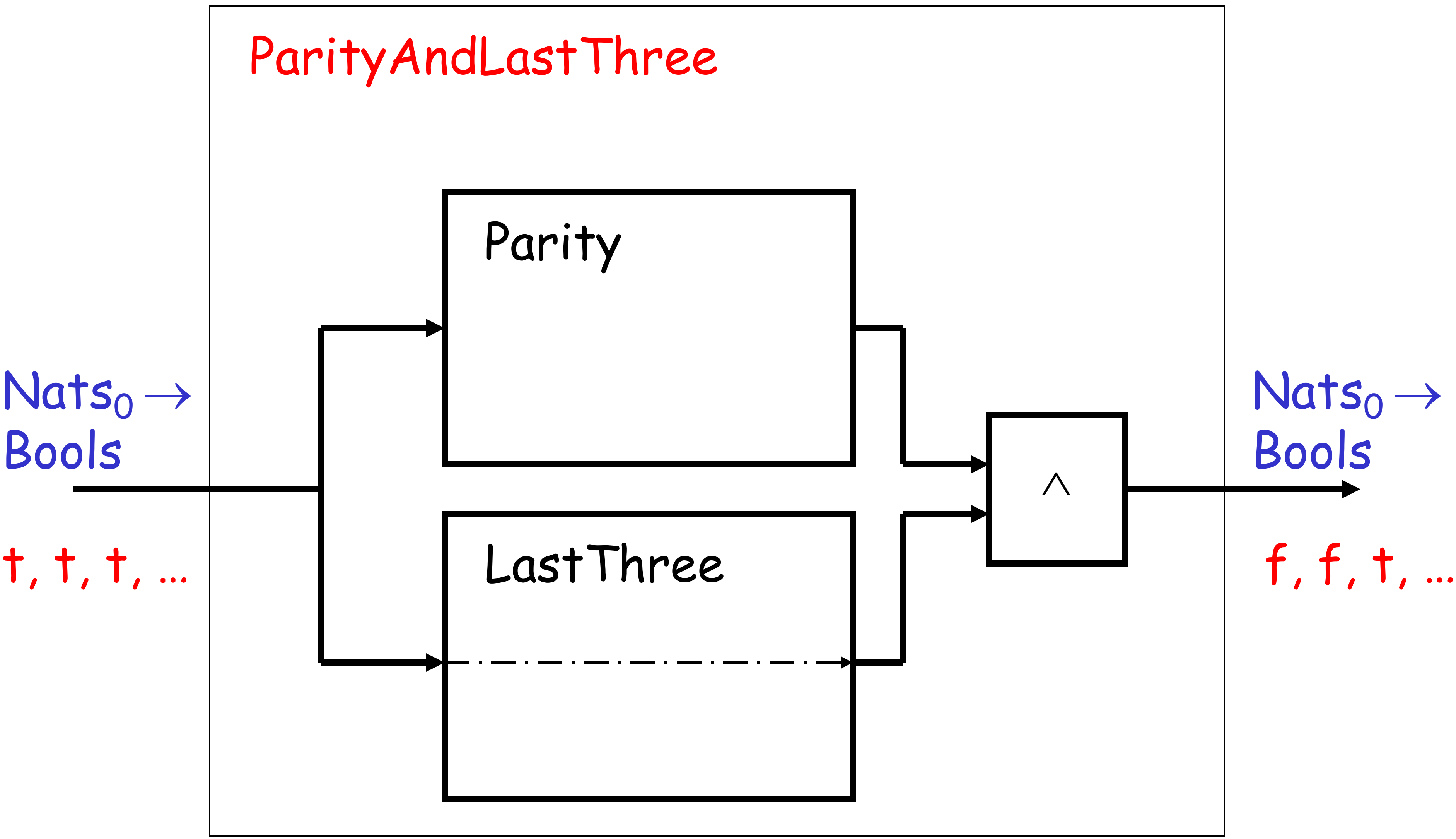
output [ParityOrLastThree] ((q1, q2) , x)

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



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


ParityAndLastThree
Parity
LastThree
∧
Nats$_0$ →
Bools
Nats$_0$ →
Bools


ParityAndLastThree
Parity
LastThree
∧
$Nats_0 \to$ Bools
t, t, t, ...
$Nats_0 \to$ Bools
f, f, t, ...

The ParityAndLastThree System

Inputs [ParityAndLastThree] = Bools

Outputs [ParityAndLastThree] = Bools

States [ParityAndLastThree]

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

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

initialState [ParityAndLastThree]

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

= (true, 0)

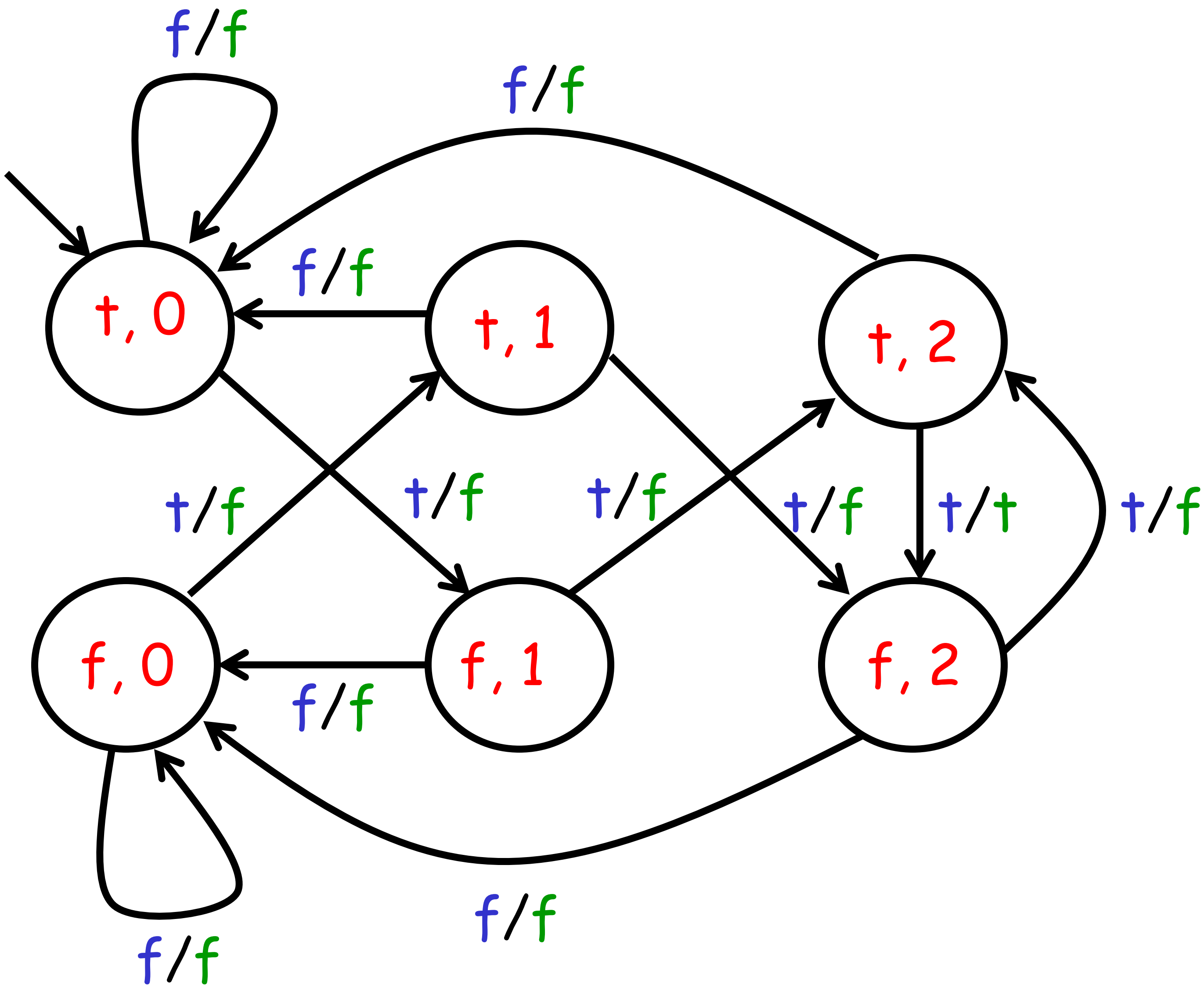
The ParityAndLastThree System, continued

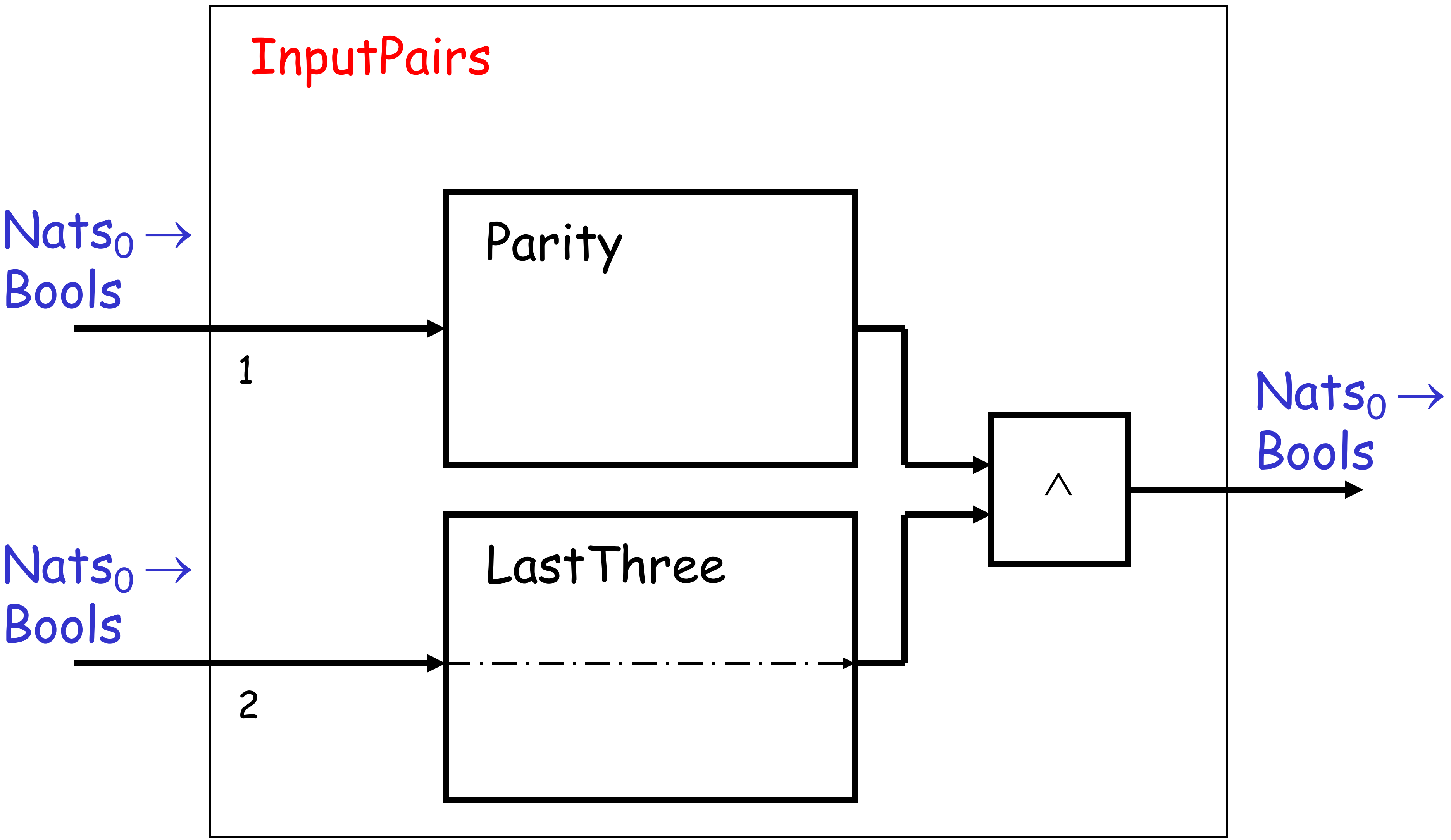
nextState [ParityAndLastThree] ((q1, q2) , x)

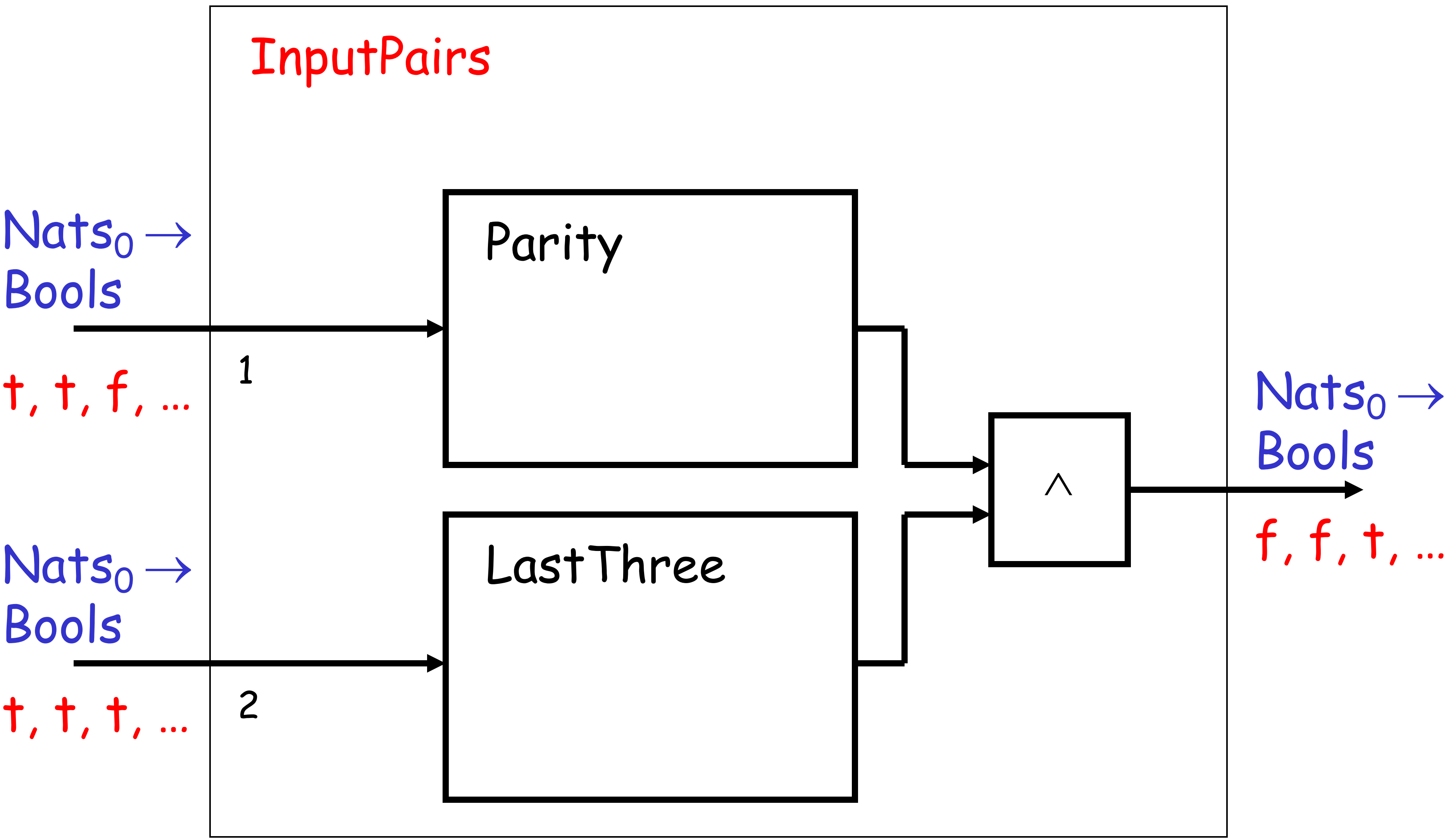
= (nextState [Parity] (q1, x) , nextState [LastThree] (q2, x))

output [ParityAndLastThree] ((q1, q2) , x)

= output [Parity] (q1, x) $\wedge$ output [LastThree] (q2, x)


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


InputPairs
Nats$_0$ → Bools
Parity
1
Nats$_0$ → Bools
LastThree
2
∧
Nats$_0$ → Bools


InputPairs
Nats$_0$ → Bools
t, t, f, ...
1
Parity
Nats$_0$ → Bools
t, t, t, ...
2
LastThree
∧
Nats$_0$ → Bools
f, f, t, ...

The InputPairs System

Inputs [InputPairs] = Bools × Bools

Outputs [InputPairs] = Bools

States [InputPairs]

= States [Parity] × States [LastThree]

= { true, false } × { 0, 1, 2 }

initialState [InputPairs]

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

= (true, 0)

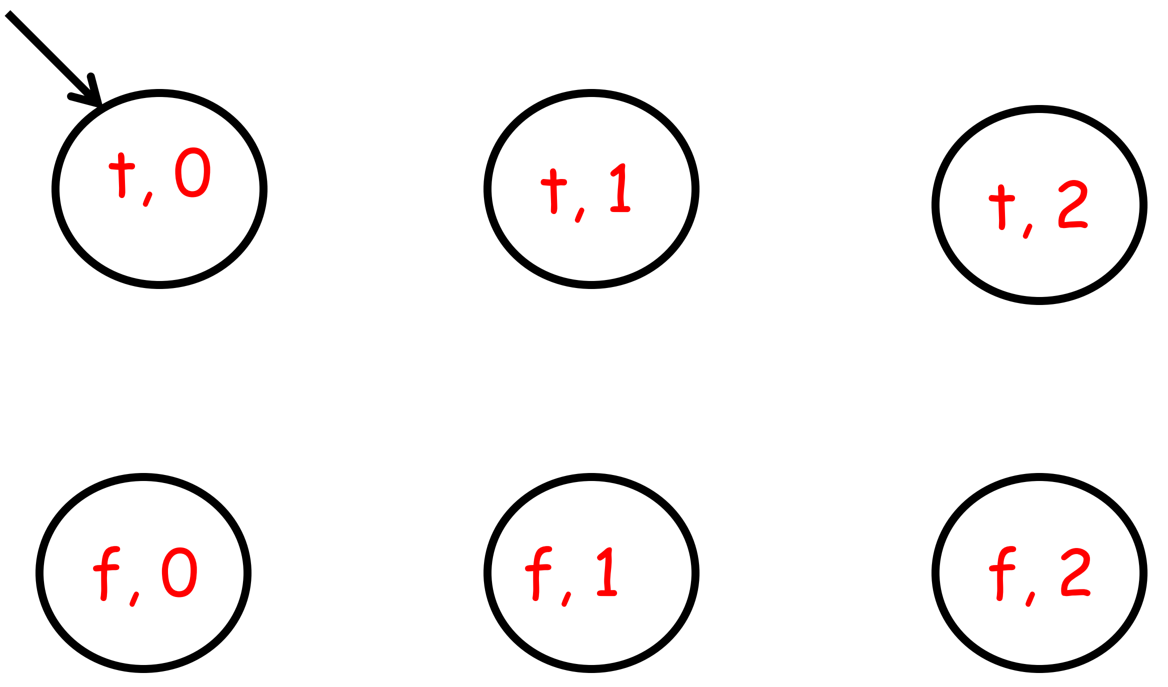
The InputPairs System, continued

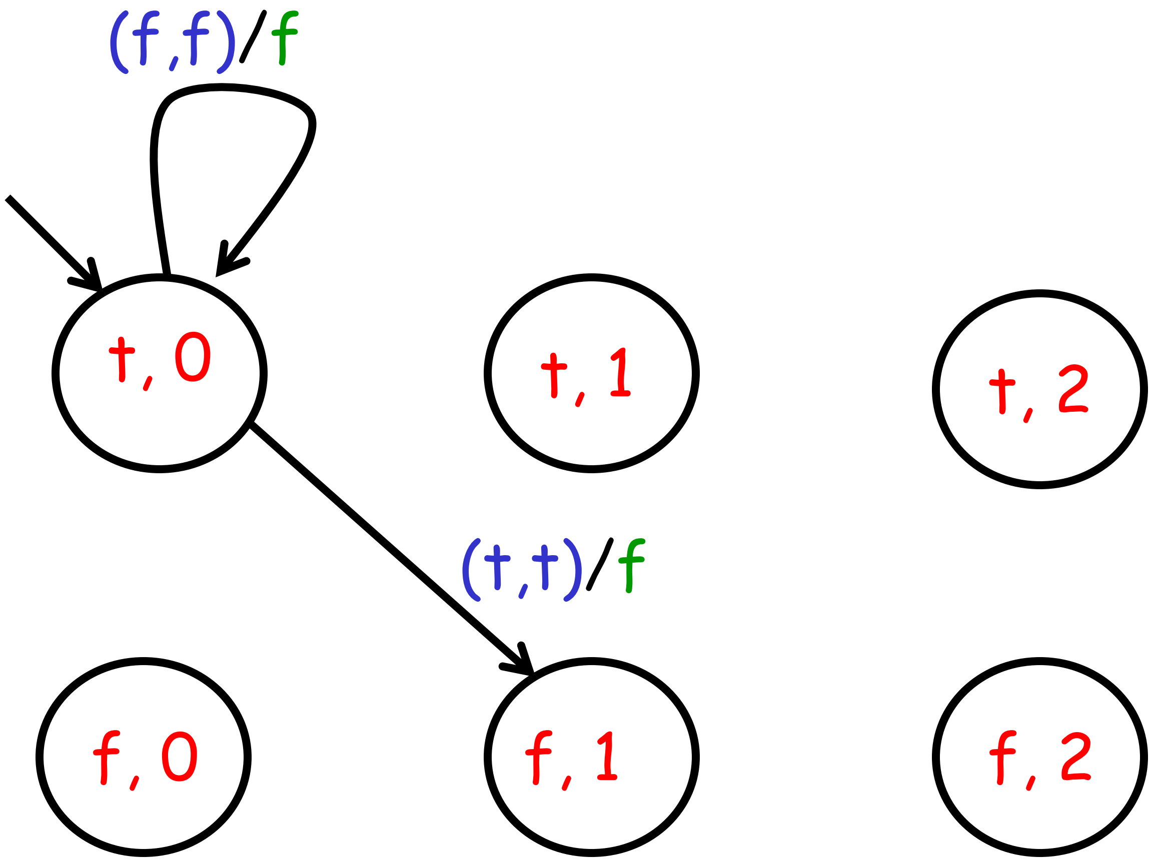
$$\text{nextState}\,[\,\text{InputPairs}\,]\,((q1, q2), (x1, x2))$$

$$= (\text{nextState}\,[\,\text{Parity}\,]\,(q1, x1), \text{nextState}\,[\,\text{LastThree}\,]\,(q2, x2))$$

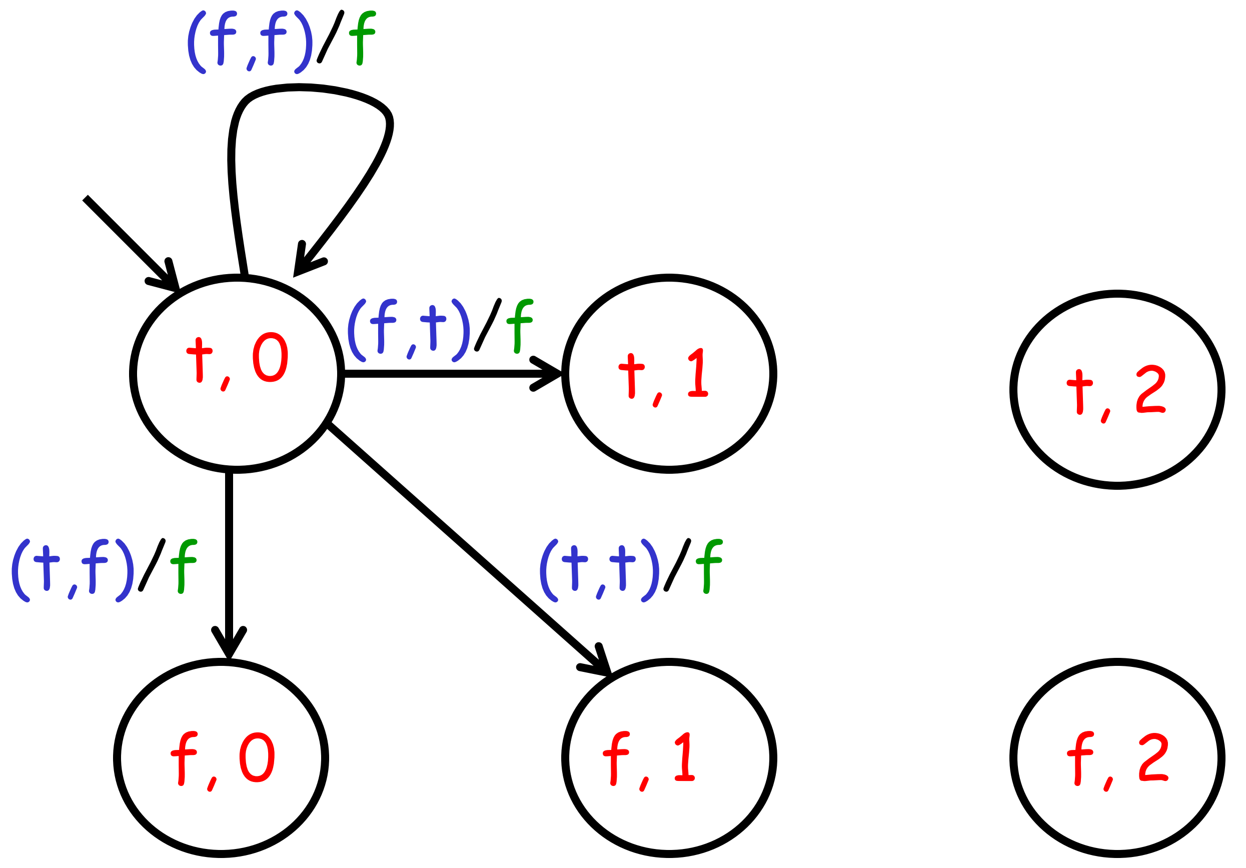
$$\text{output}\,[\,\text{InputPairs}\,]\,((q1, q2), (x1, x2))$$

$$= \text{output}\,[\,\text{Parity}\,]\,(q1, x1) \wedge \text{output}\,[\,\text{LastThree}\,]\,(q2, x2)$$

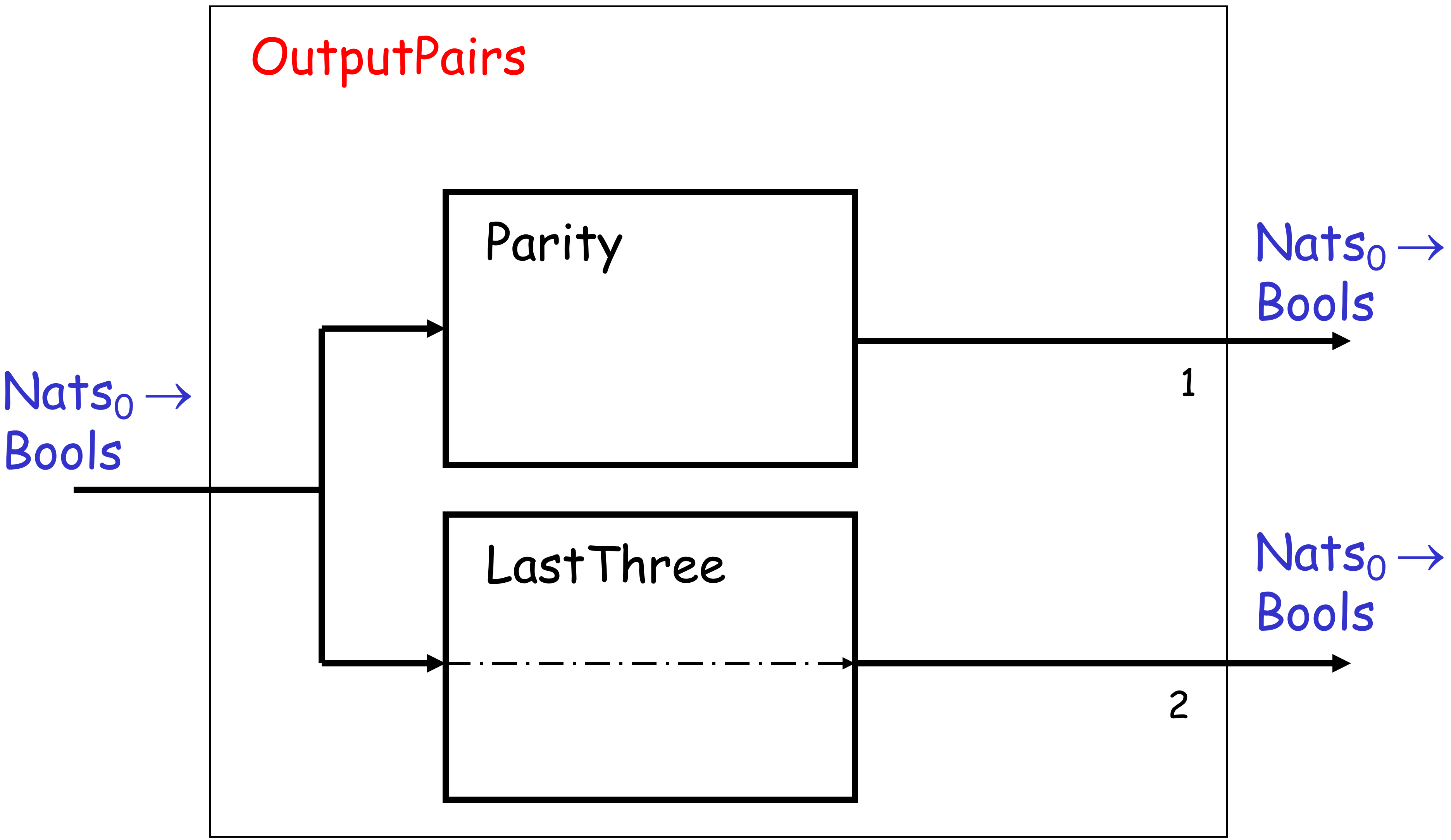

t, 0
t, 1
t, 2
f, 0
f, 1
f, 2

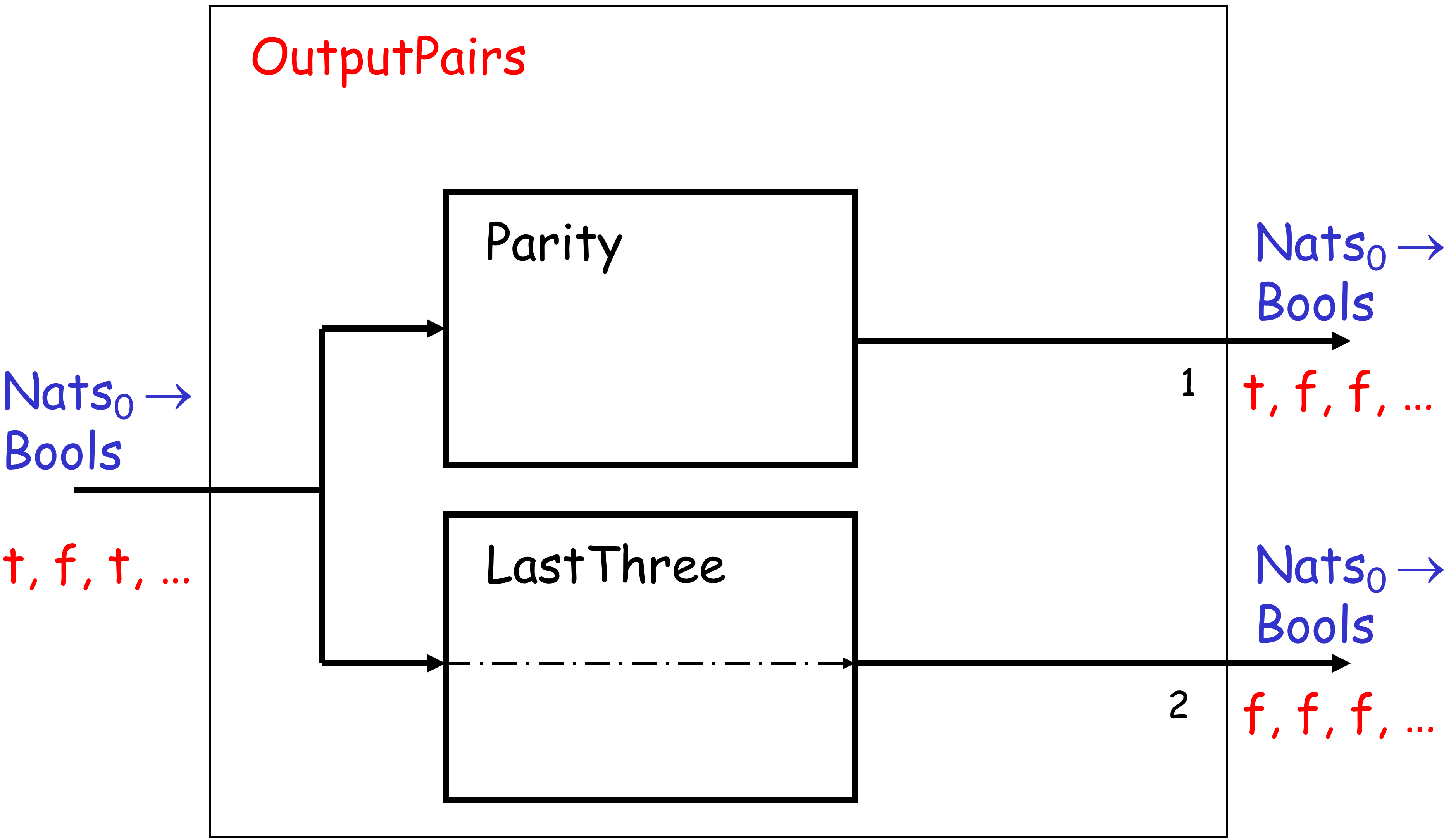


(f,f)/f
t, 0
t, 1
t, 2
(t,t)/f
f, 0
f, 1
f, 2



(f,f)/f
(f,t)/f
(t,f)/f
(t,t)/f
t, 0
t, 1
t, 2
f, 0
f, 1
f, 2


OutputPairs
Parity
LastThree
Nats0 → Bools
Nats0 → Bools
Nats0 → Bools
1
2



OutputPairs
Parity
LastThree
Nats$_0$ → Bools
t, f, t, …
Nats$_0$ → Bools
1
t, f, f, …
Nats$_0$ → Bools
2
f, f, f, …

The OutputPairs System

$$\text{Inputs}[\text{OutputPairs}] = \text{Bools}$$

$$\text{Outputs}[\text{OutputPairs}] = \text{Bools} \times \text{Bools}$$

$$\begin{aligned}\text{States}[\text{OutputPairs}] &\\ &= \text{States}[\text{Parity}] \times \text{States}[\text{LastThree}] \\ &= \{\text{true}, \text{false}\} \times \{0, 1, 2\}\end{aligned}$$

$$\begin{aligned}\text{initialState}[\text{OutputPairs}] &\\ &= (\text{initialState}[\text{Parity}], \text{initialState}[\text{LastThree}]) \\ &= (\text{true}, 0)\end{aligned}$$

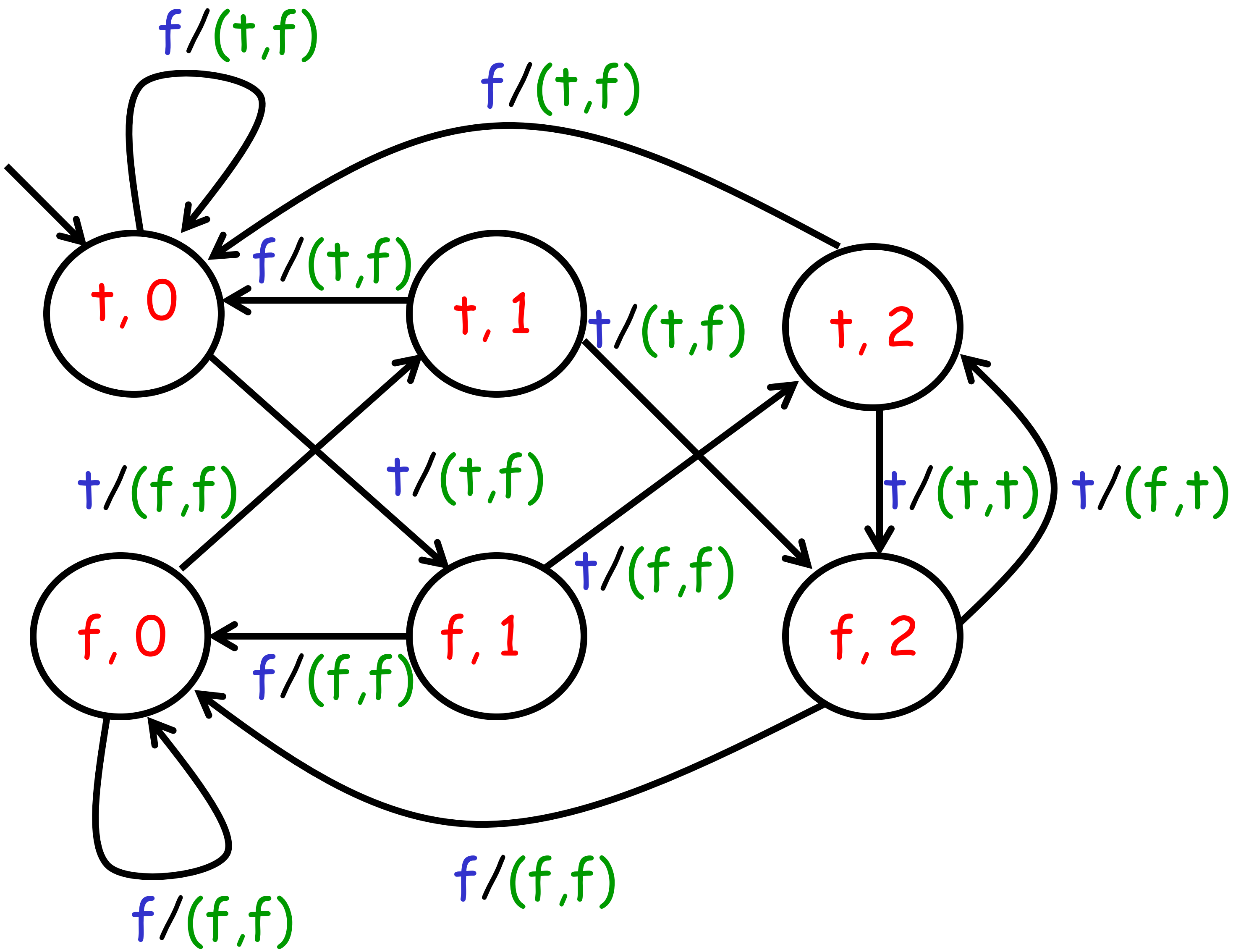
The OutputPairs System, continued

nextState [OutputPairs] ((q1, q2) , x)

= (nextState [Parity] (q1, x) , nextState [LastThree] (q2, x))

output [OutputPairs] ((q1, q2) , x)

= (output [Parity] (q1, x) , output [LastThree] (q2, x))



t, 0
t, 1
t, 2
f, 0
f, 1
f, 2
f/(t,f)
f/(t,f)
f/(t,f)
t/(t,f)
t/(f,f)
t/(t,f)
t/(f,f)
t/(t,t)
t/(f,t)
f/(f,f)
f/(f,f)
f/(f,f)

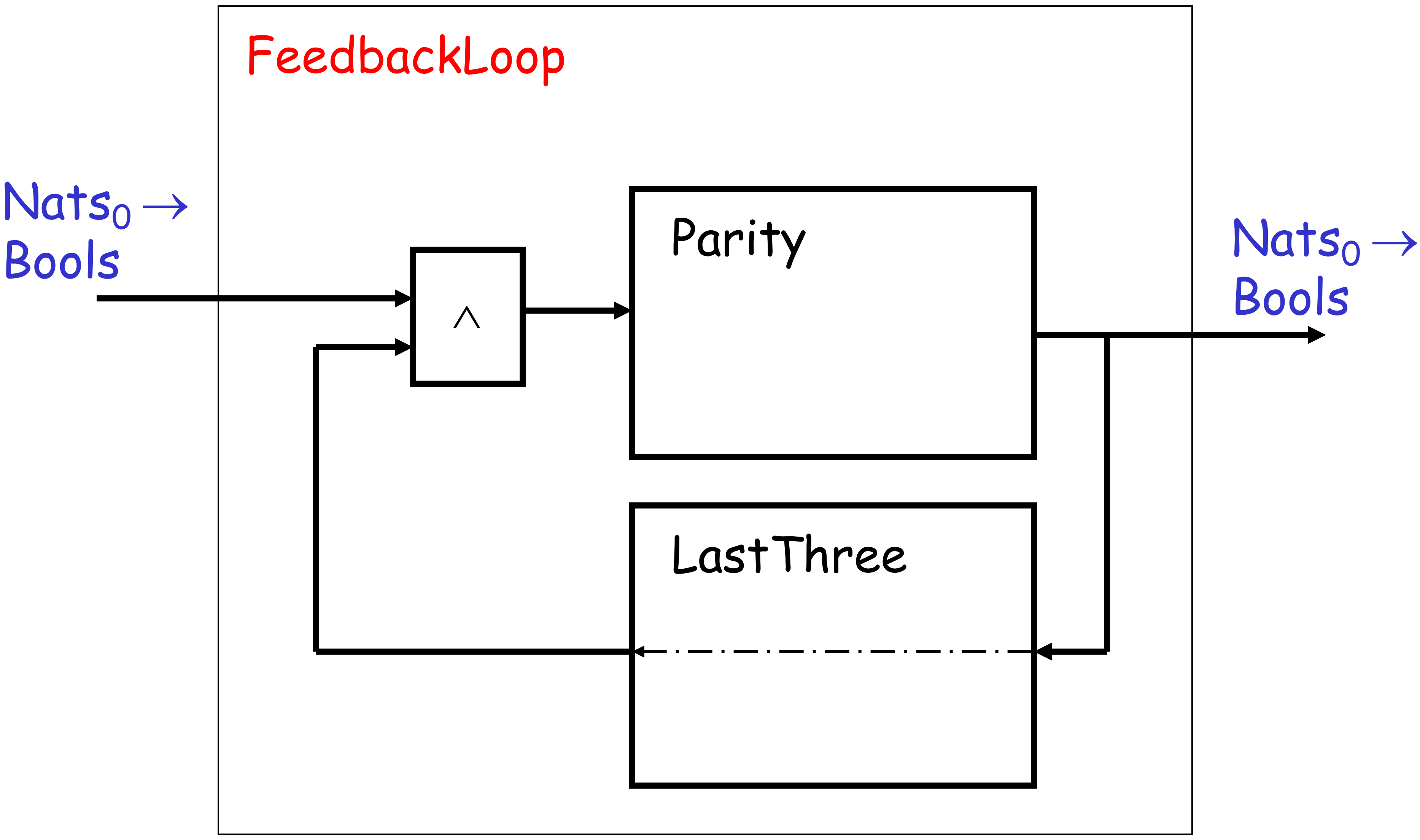
Any block diagram of N state machines with the state spaces

States1, States2, … StatesN

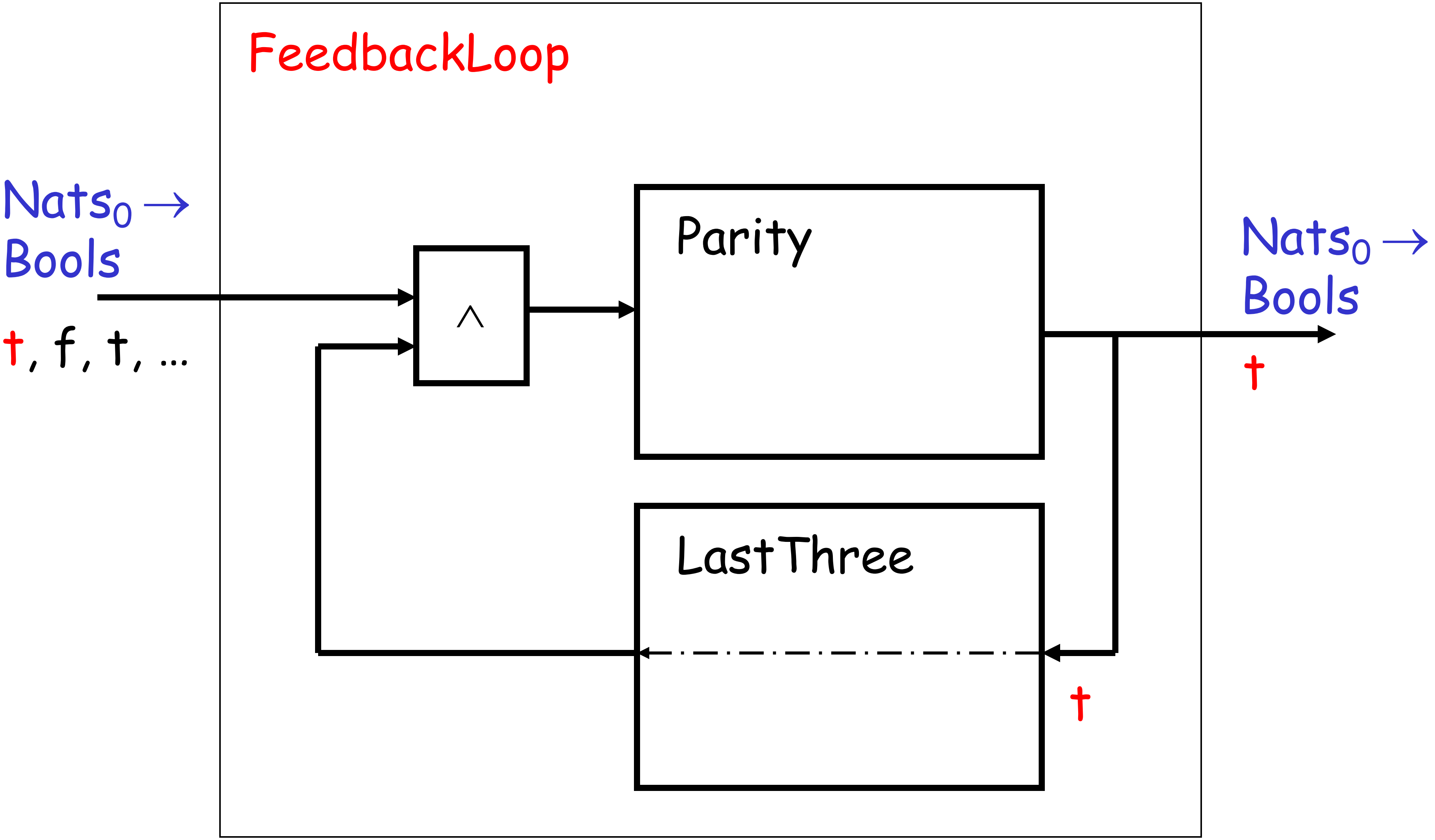
can be implemented by a single state machine with the state space

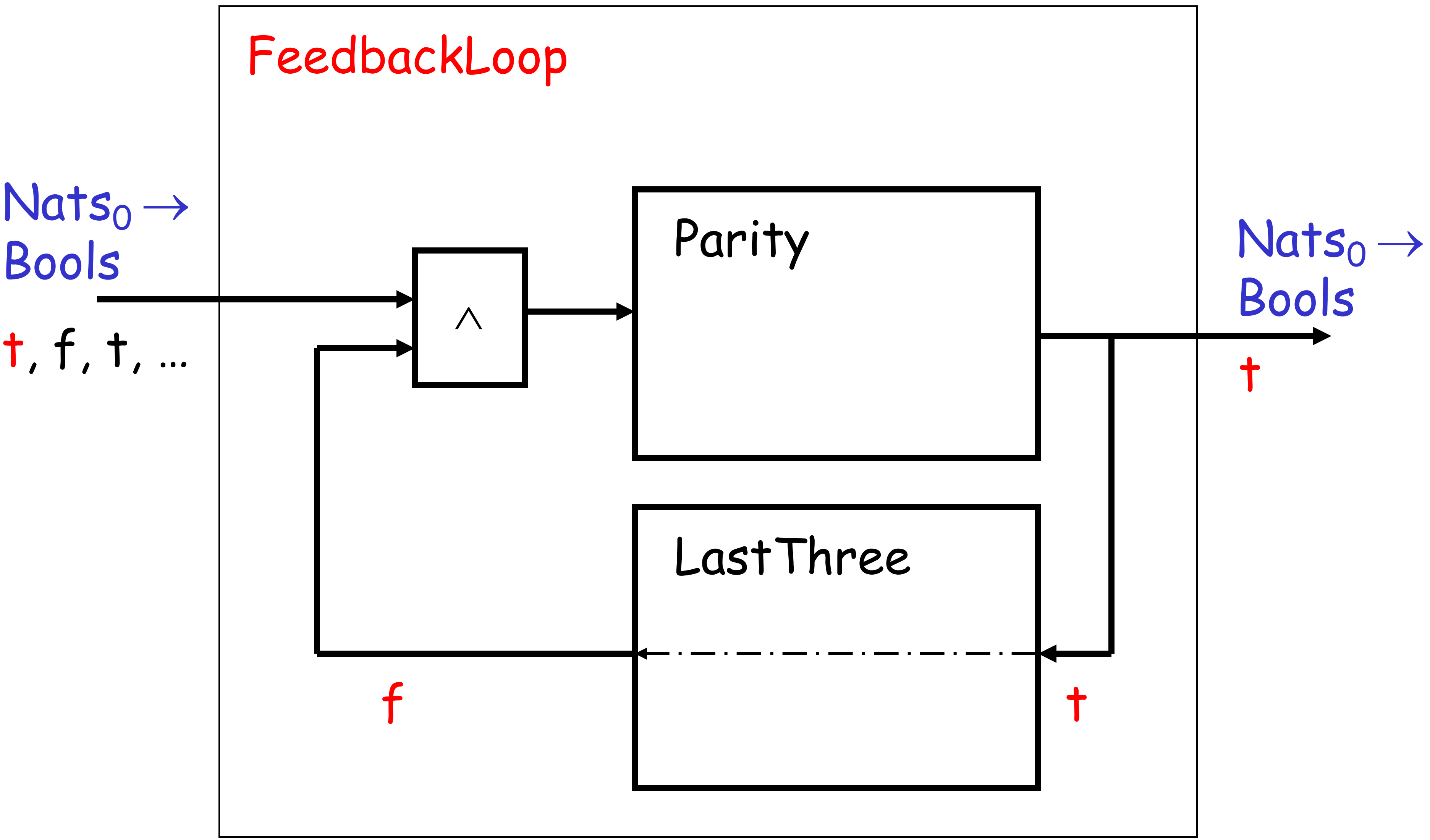
States1 $\times$ States2 $\times$ … $\times$ StatesN .

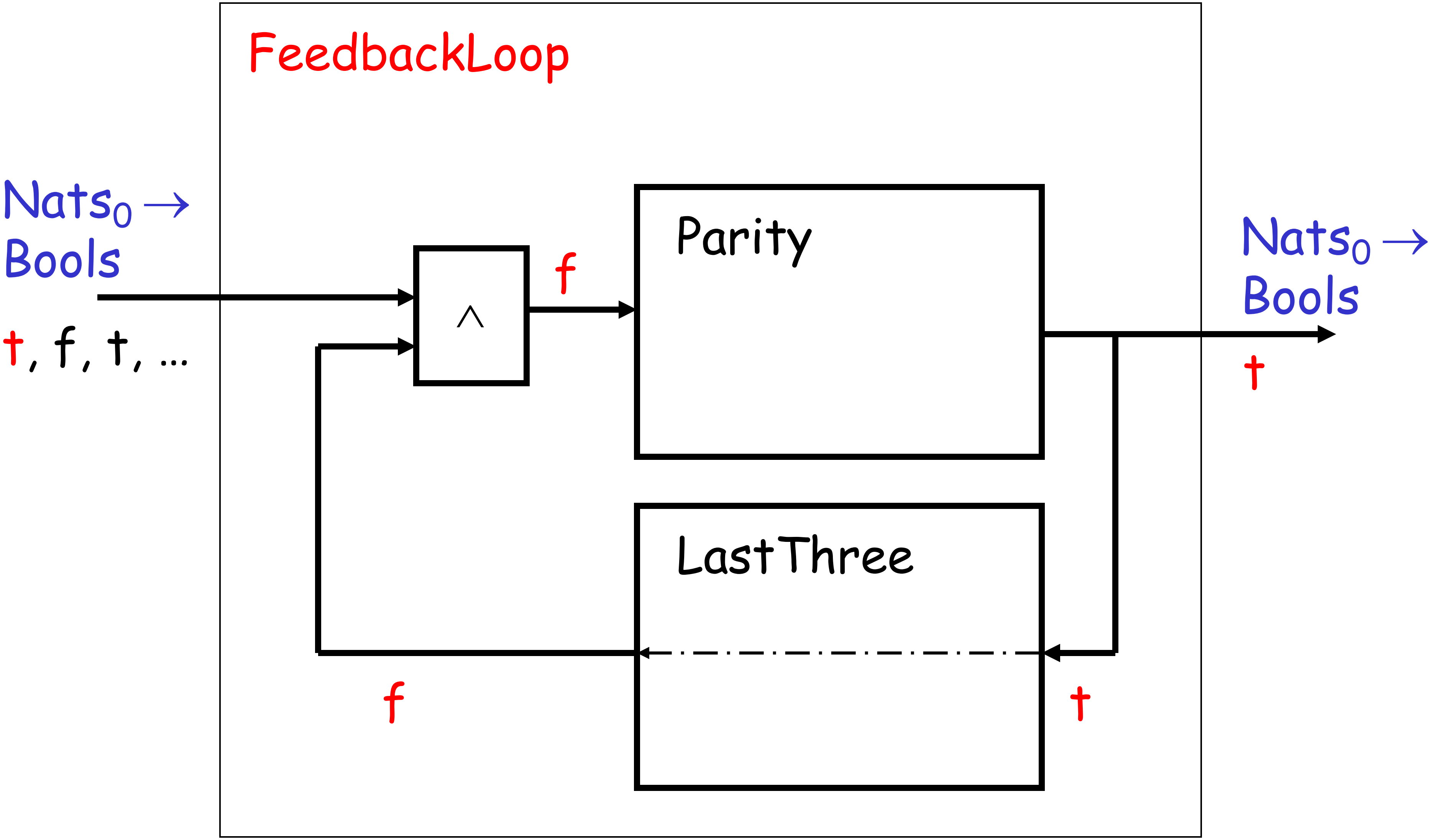
This is called a "product machine".



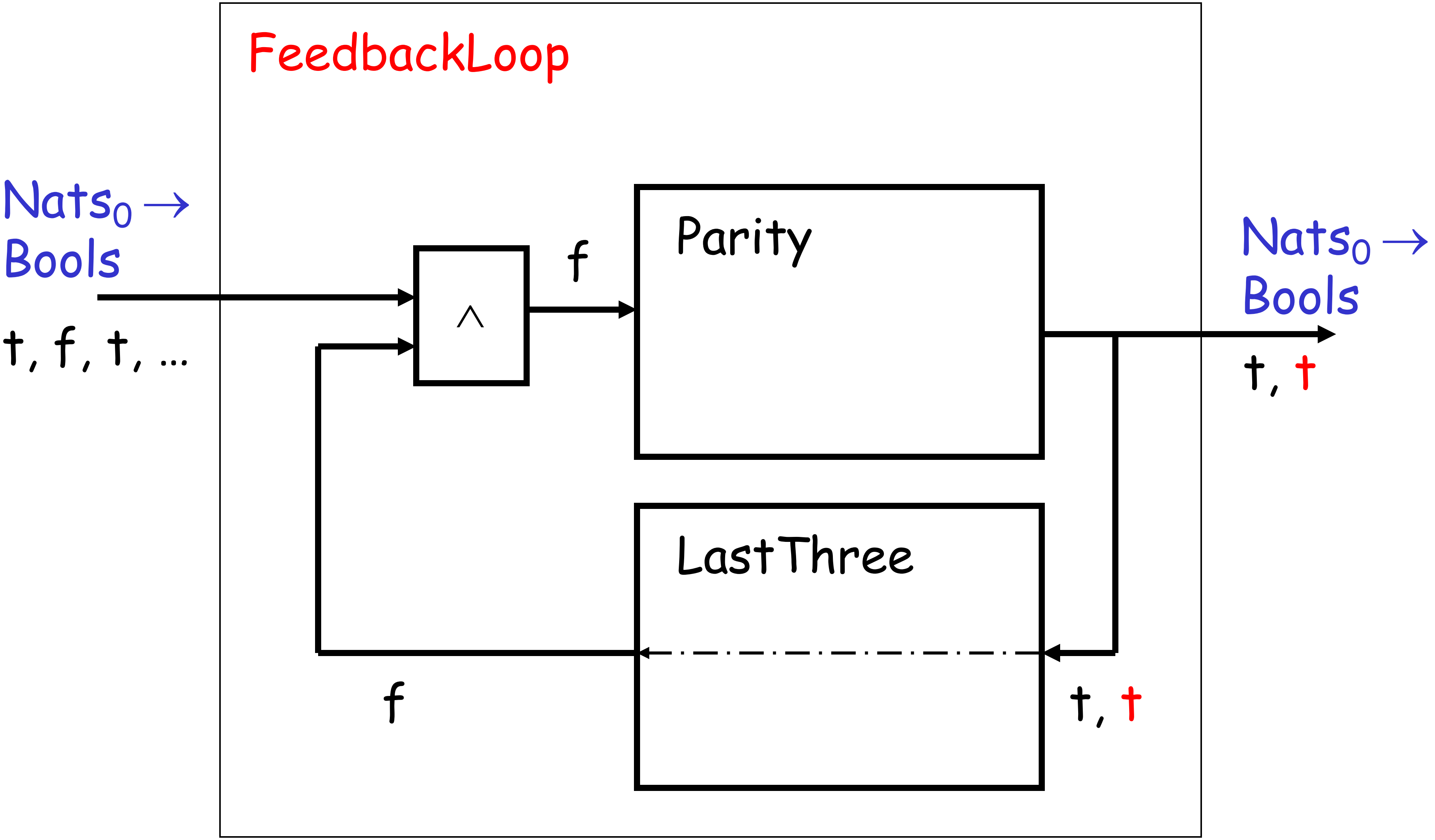
This block diagram is ok, because every cycle contains a delay.


FeedbackLoop
Nats0 → Bools
t, f, t, ...
∧
Parity
LastThree
Nats0 → Bools
t
t

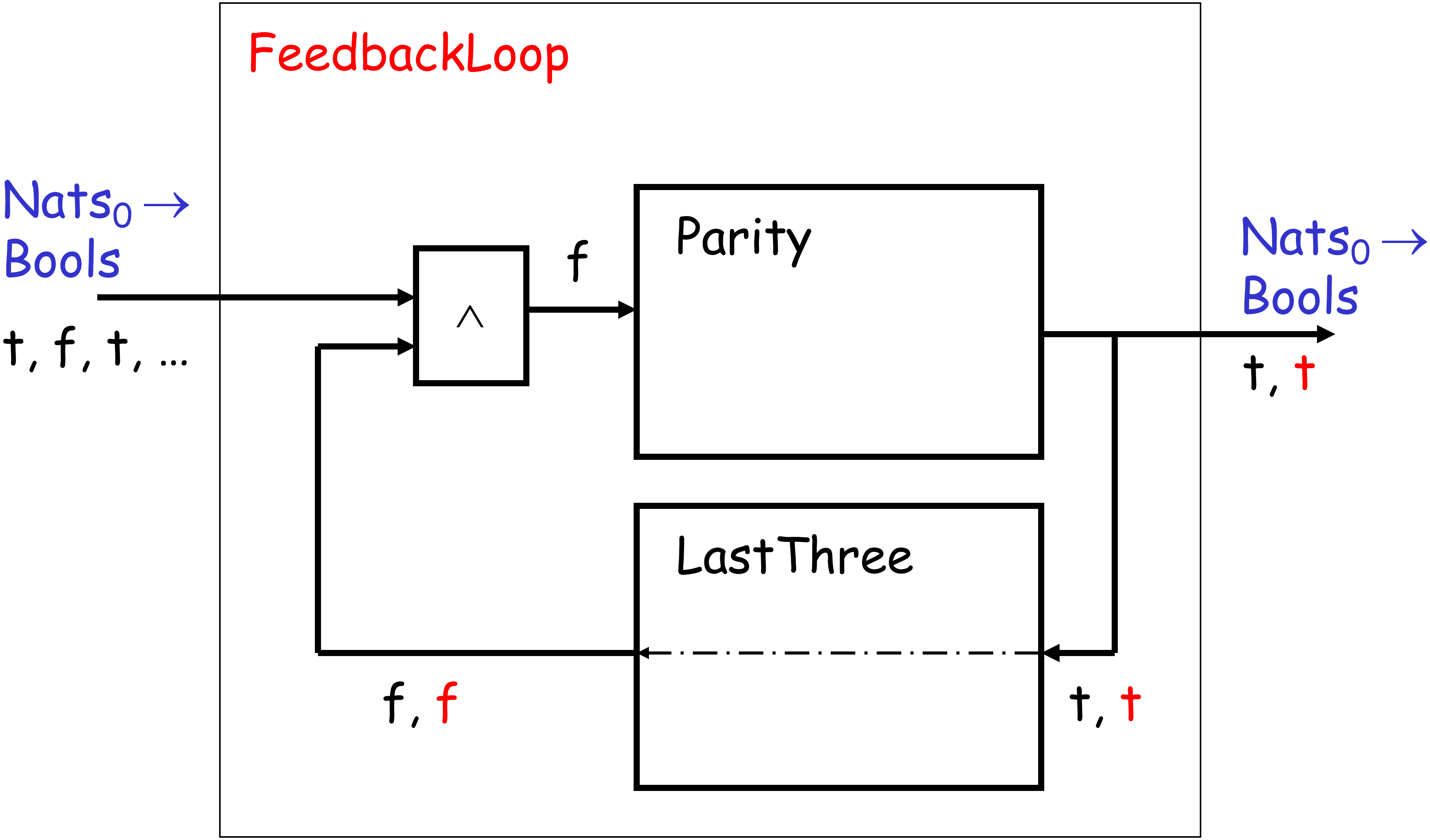

FeedbackLoop
Nats0 → Bools
t, f, t, ...
∧
Parity
Nats0 → Bools
t
LastThree
f
t

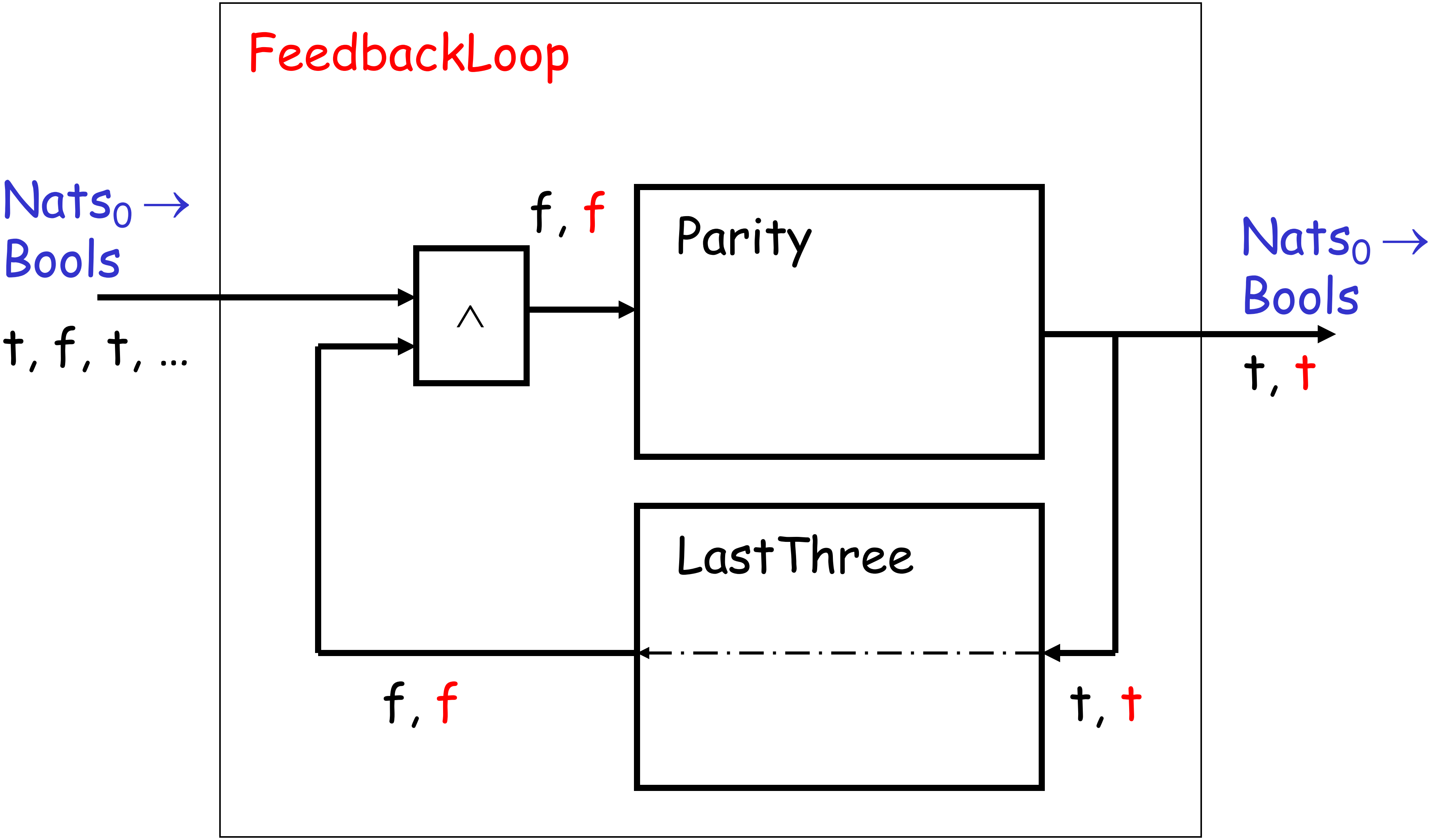


FeedbackLoop
Nats0 → Bools
t, f, t, ...
∧
f
Parity
Nats0 → Bools
t
LastThree
f
t

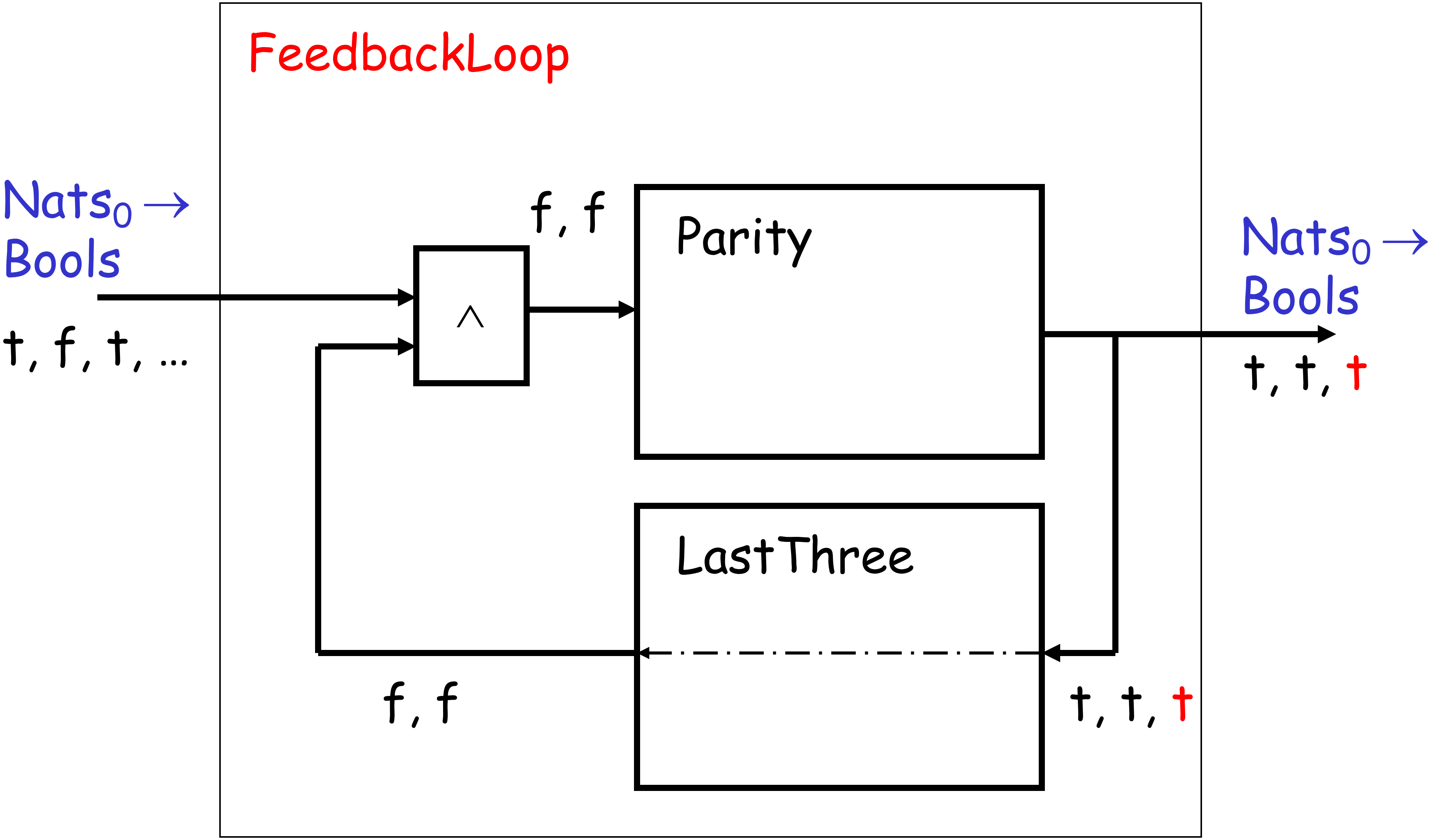


FeedbackLoop
$Nats_0 \rightarrow Bools$
t, f, t, ...
∧
f
Parity
$Nats_0 \rightarrow Bools$
t, t
LastThree
f
t, t

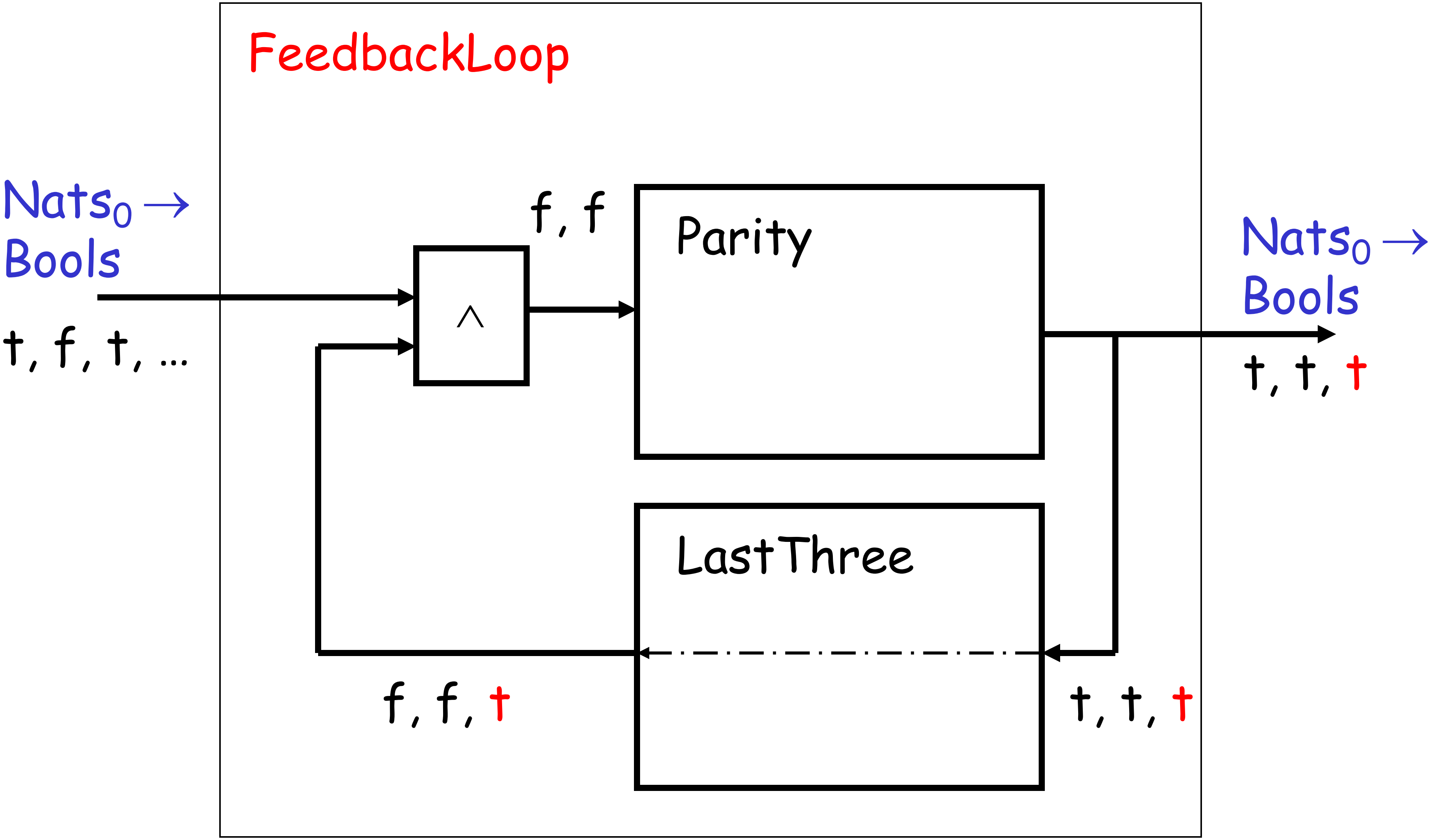

FeedbackLoop
Nats$_0$ → Bools
t, f, t, ...
∧
f
Parity
Nats$_0$ → Bools
t, t
LastThree
f, f
t, t

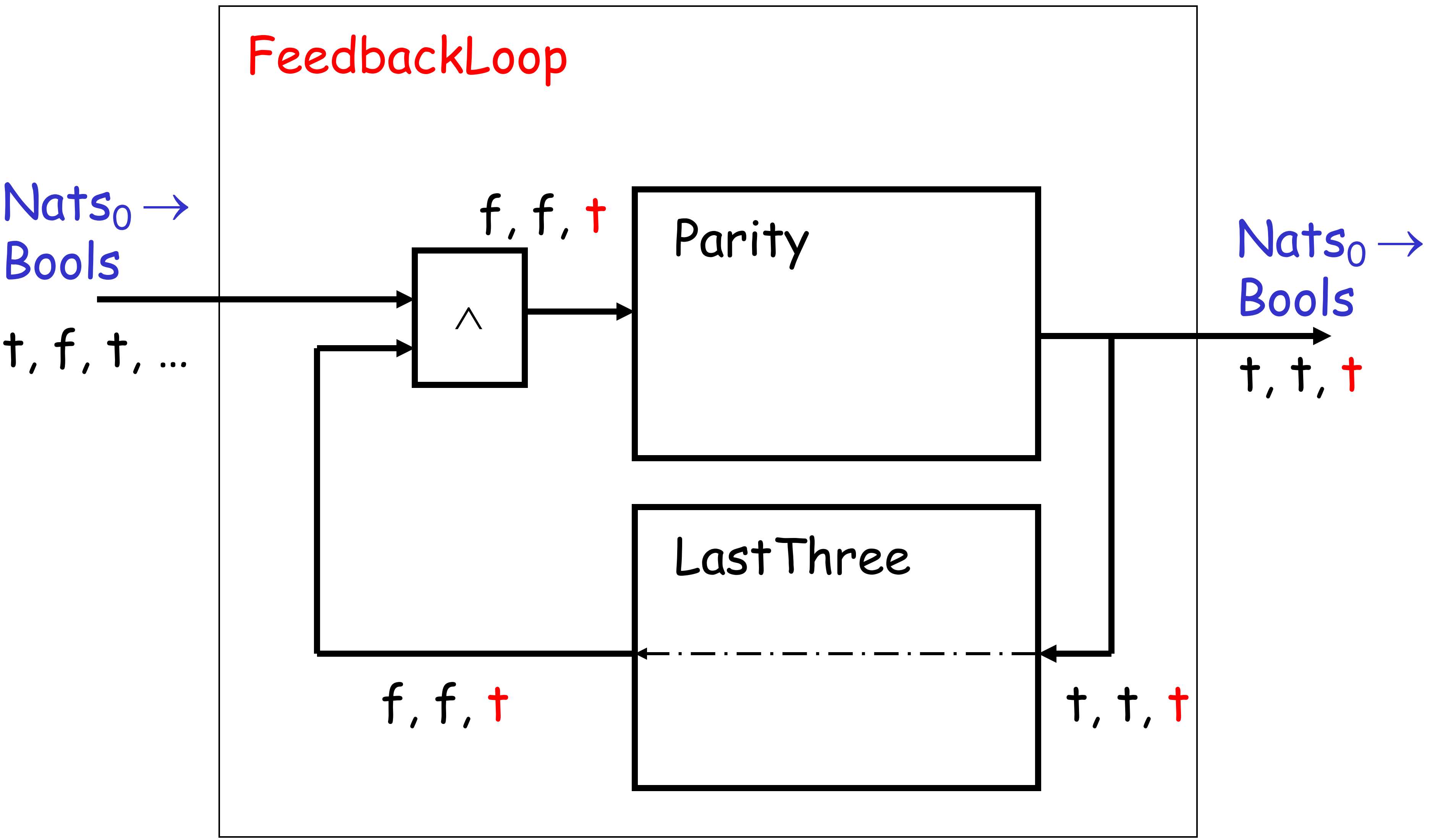


FeedbackLoop
$Nats_0 \rightarrow$ Bools
t, f, t, ...
f, f
$\wedge$
Parity
$Nats_0 \rightarrow$ Bools
t, t
LastThree
f, f
t, t

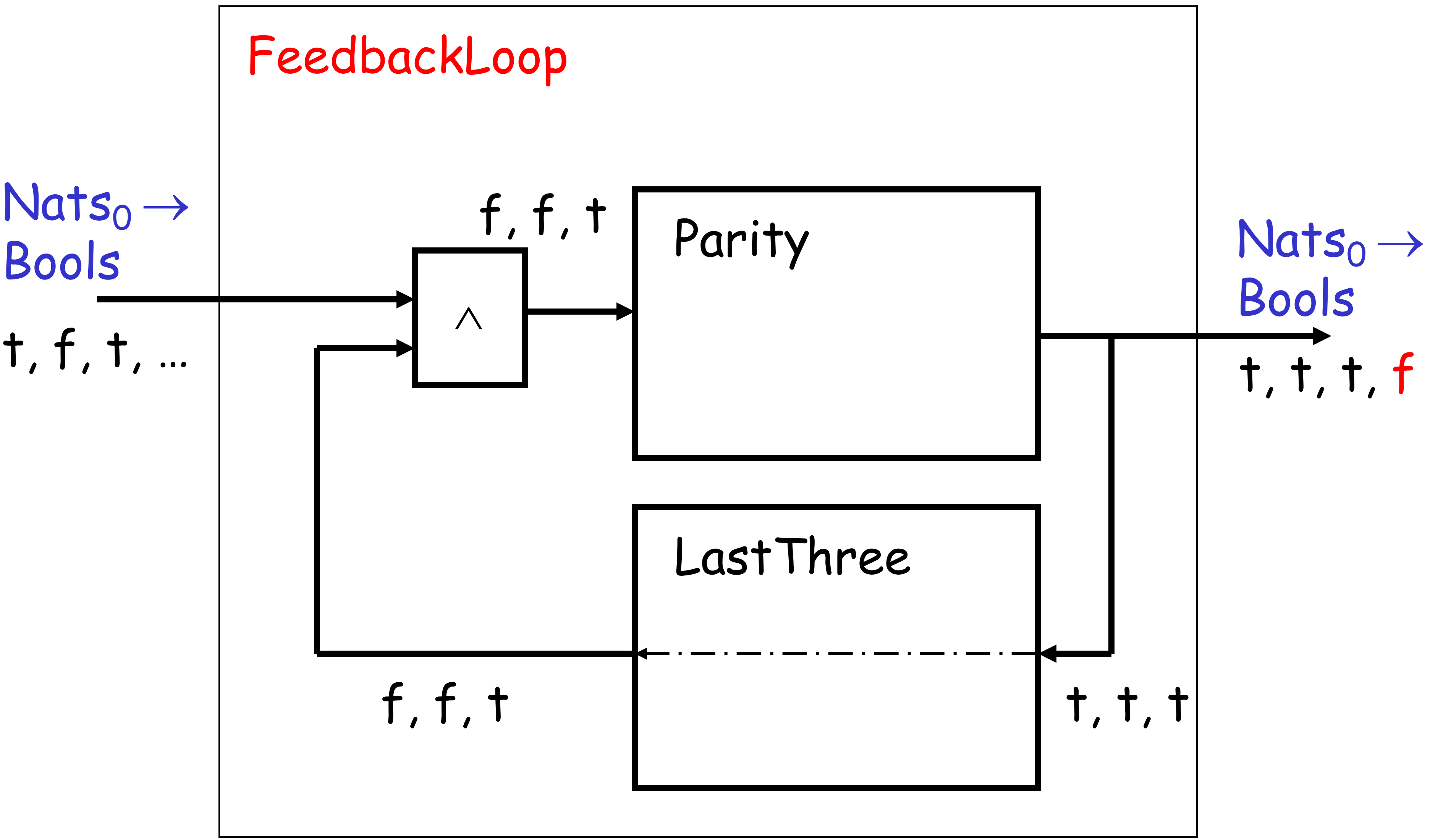


FeedbackLoop
$Nats_0 \to$ Bools
t, f, t, …
∧
f, f
Parity
$Nats_0 \to$ Bools
t, t, t
LastThree
f, f
t, t, t


FeedbackLoop
Nats0 → Bools
t, f, t, ...
f, f
∧
Parity
Nats0 → Bools
t, t, t
LastThree
f, f, t
t, t, t



FeedbackLoop
$Nats_0 \rightarrow Bools$
t, f, t, ...
∧
f, f, t
Parity
$Nats_0 \rightarrow Bools$
t, t, t
LastThree
f, f, t
t, t, t


FeedbackLoop
Nats0 → Bools
t, f, t, ...
f, f, t
∧
Parity
Nats0 → Bools
t, t, t, f
LastThree
f, f, t
t, t, t

Example: Vending Machine

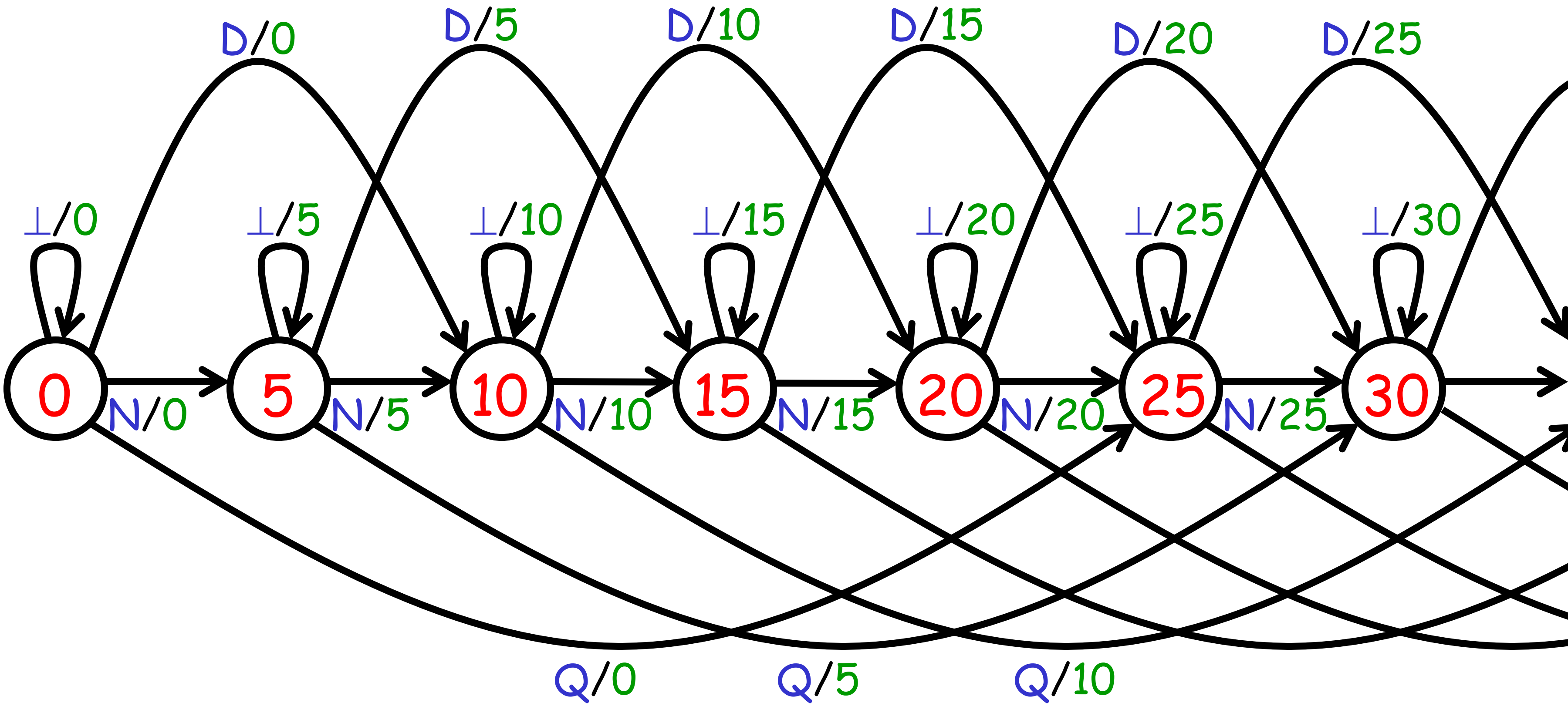
Coin Collector

$$(\mathrm{Nats}_0 \to \mathrm{Coins}_\perp) \longrightarrow \boxed{\text{Collect}} \longrightarrow (\mathrm{Nats}_0 \to \mathrm{Nats}_0)$$

Let $\mathrm{Coins} = \{ \mathrm{Nickel}, \mathrm{Dime}, \mathrm{Quarter} \}$.

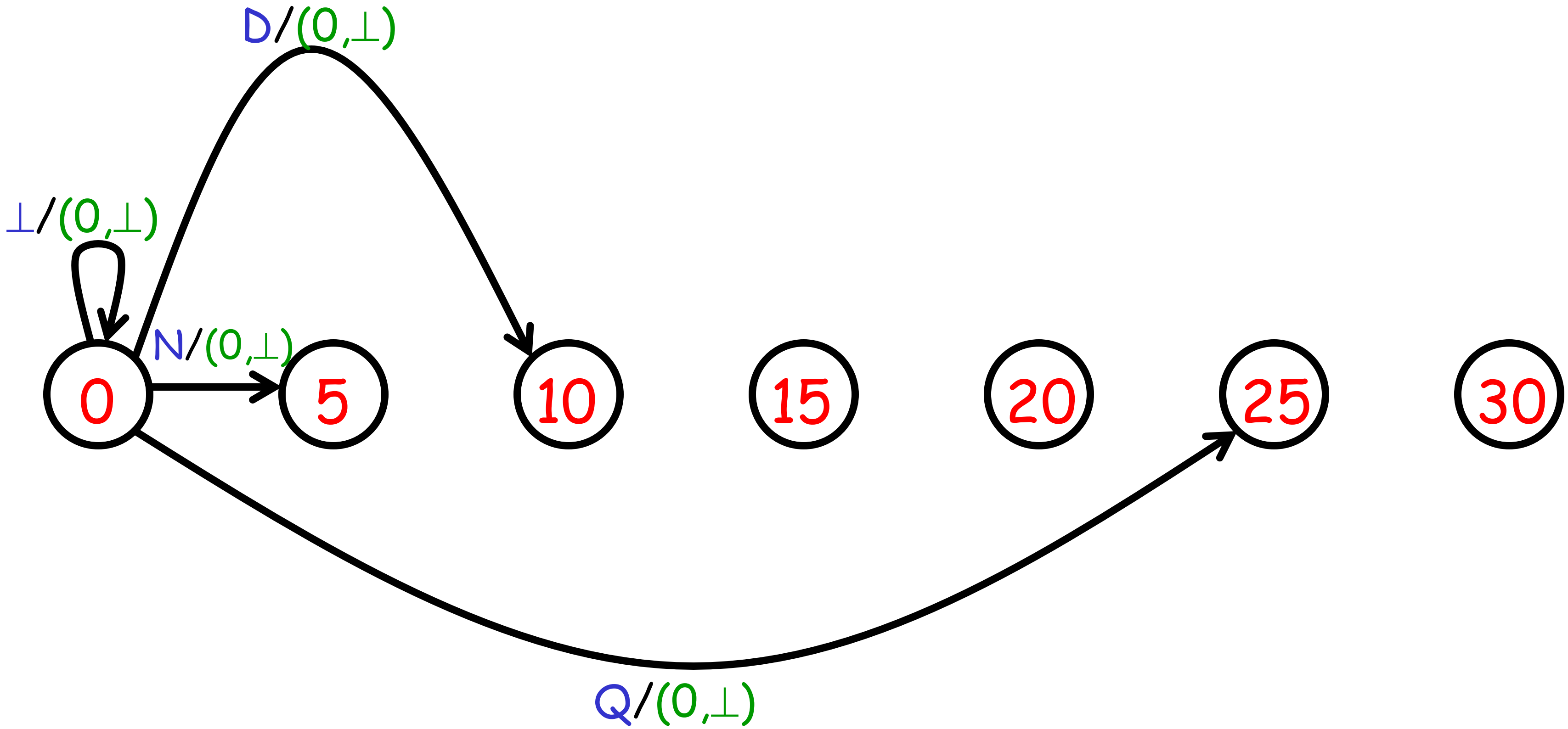
Let $\mathrm{Coins}_\perp = \mathrm{Coins} \cup \{ \perp \}$.   ($\perp$ stands for "no input.")

Coin Collector

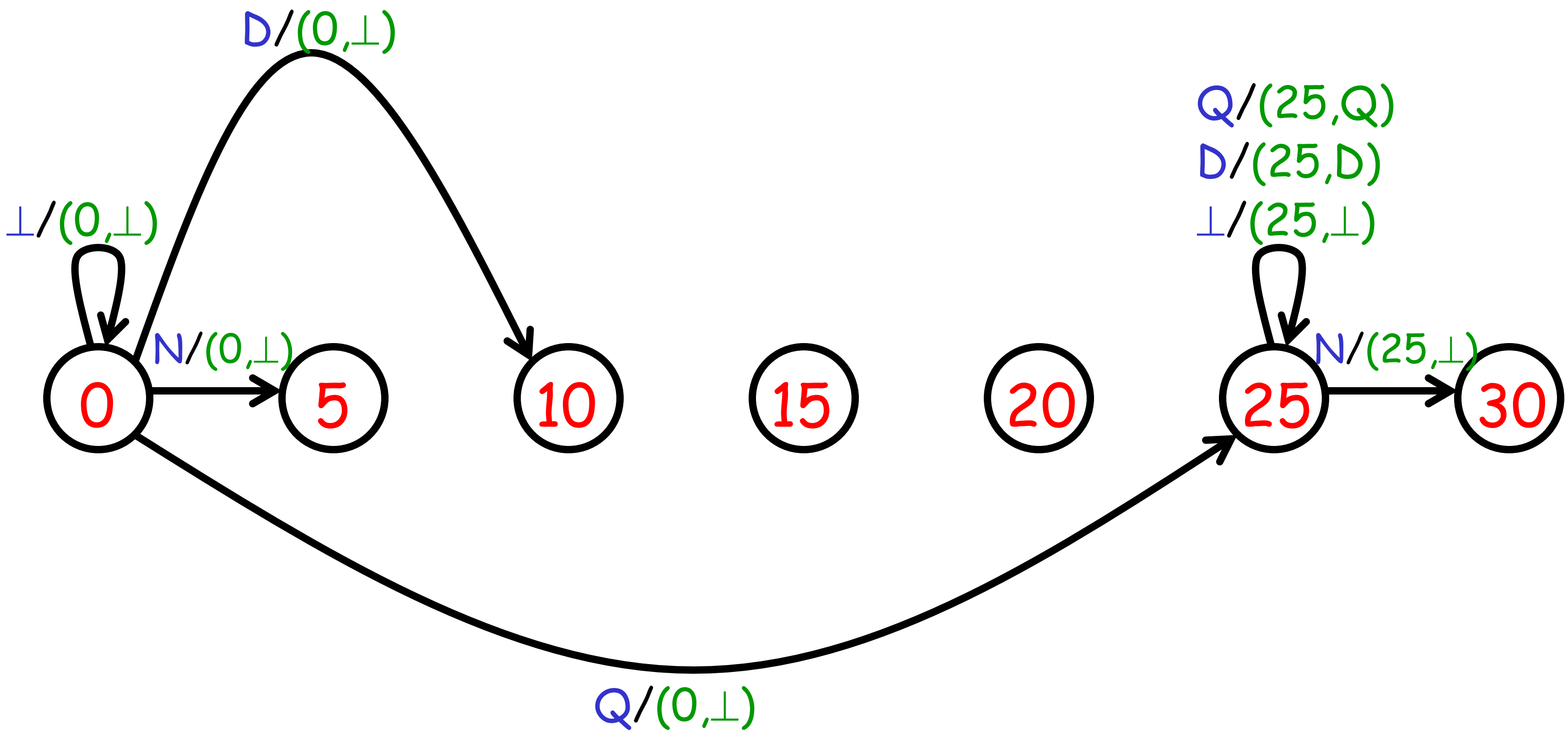


Finite-State Coin Collector

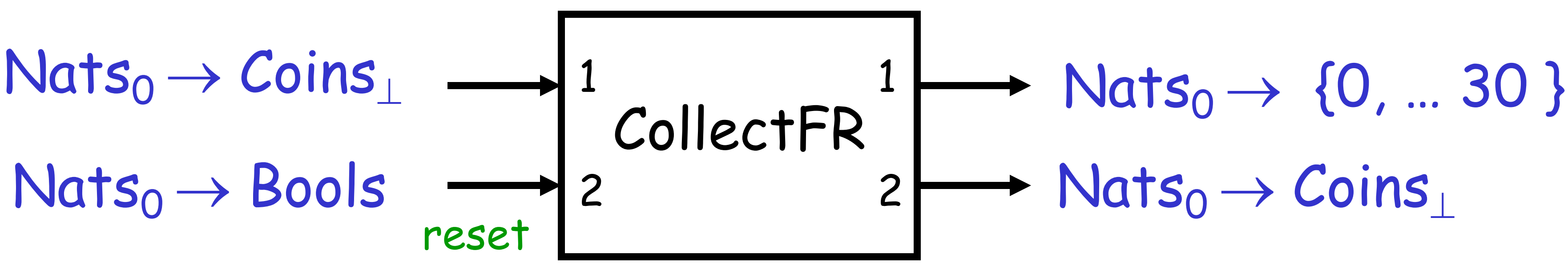
Finite-State Coin Collector



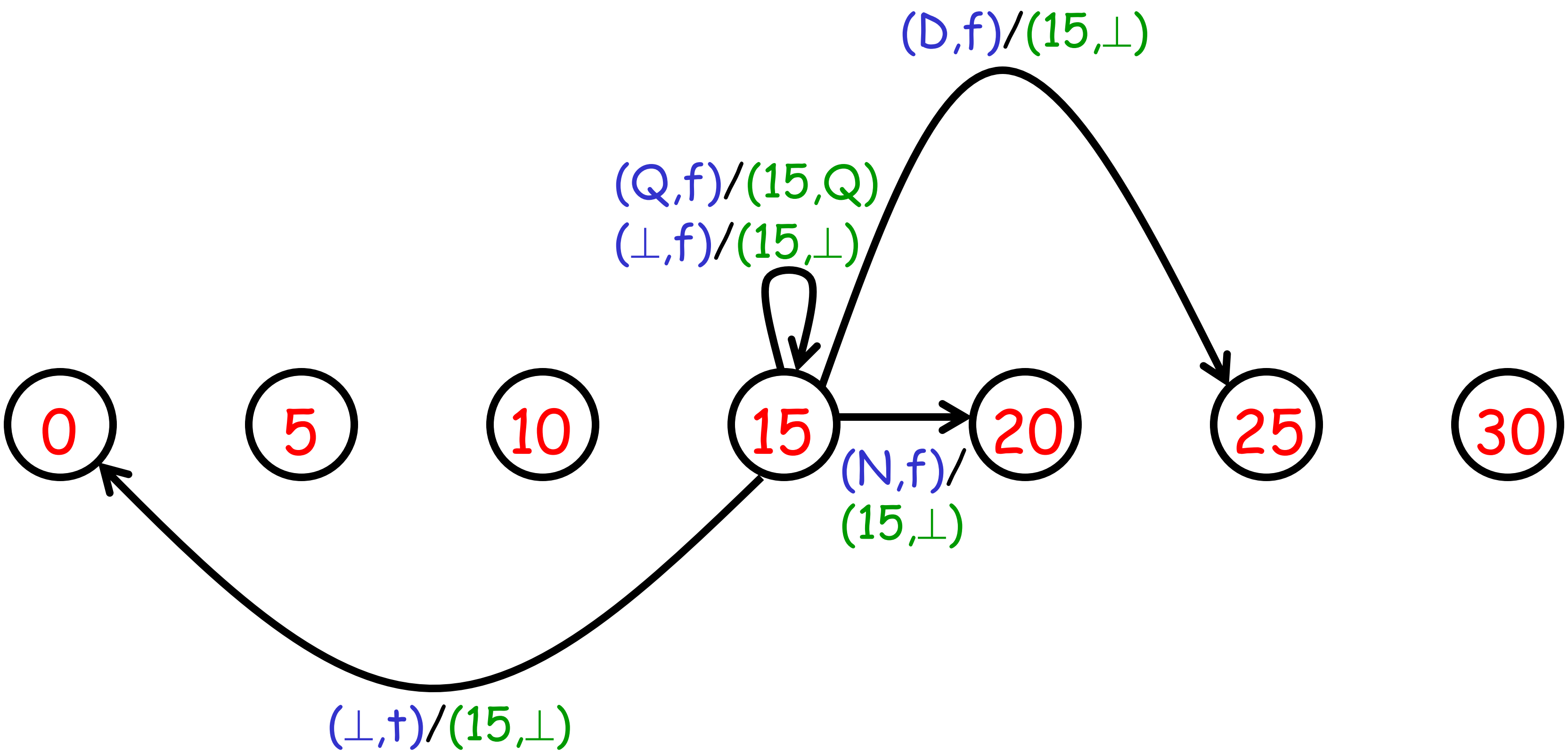
Finite-State Coin Collector



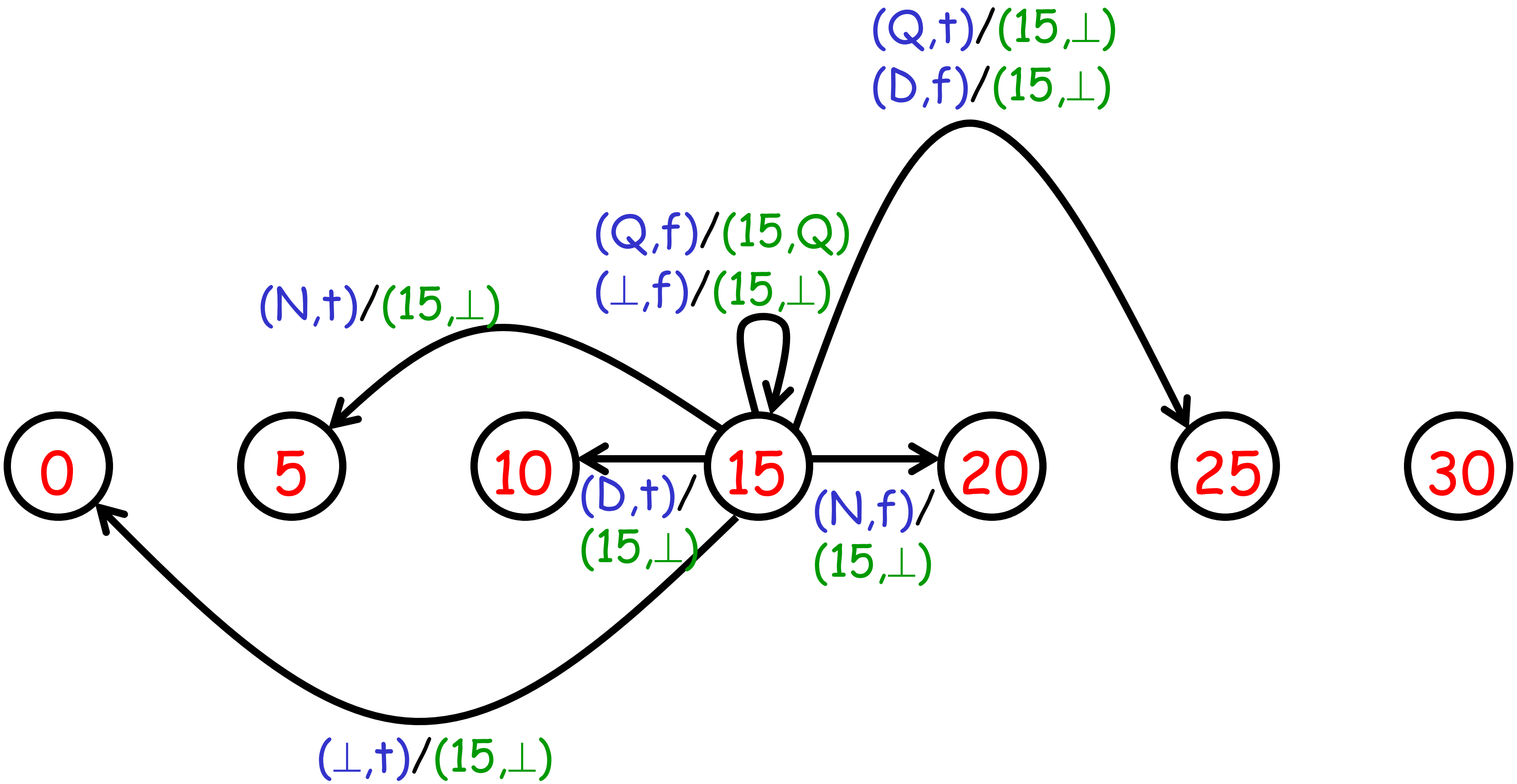
Coin Collector with Reset



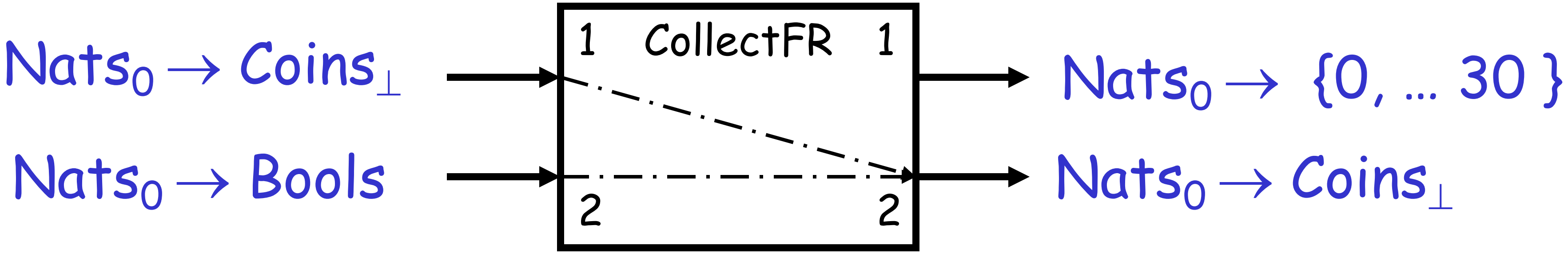
Coin Collector with Reset



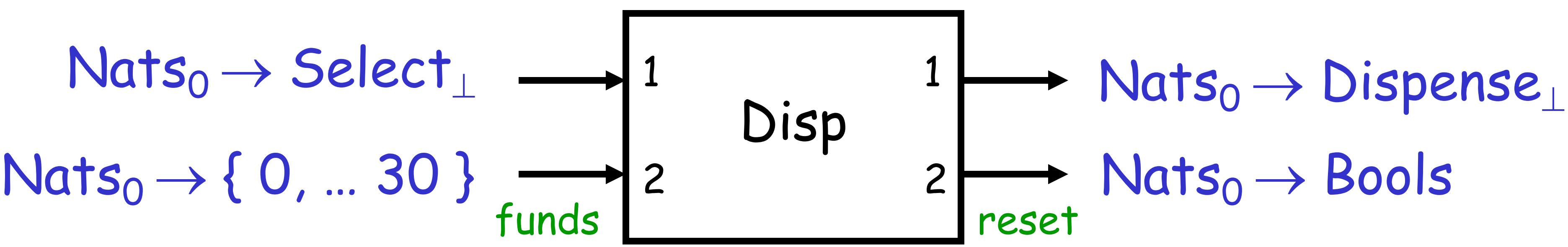
Coin Collector with Reset



Coin Collector with Reset



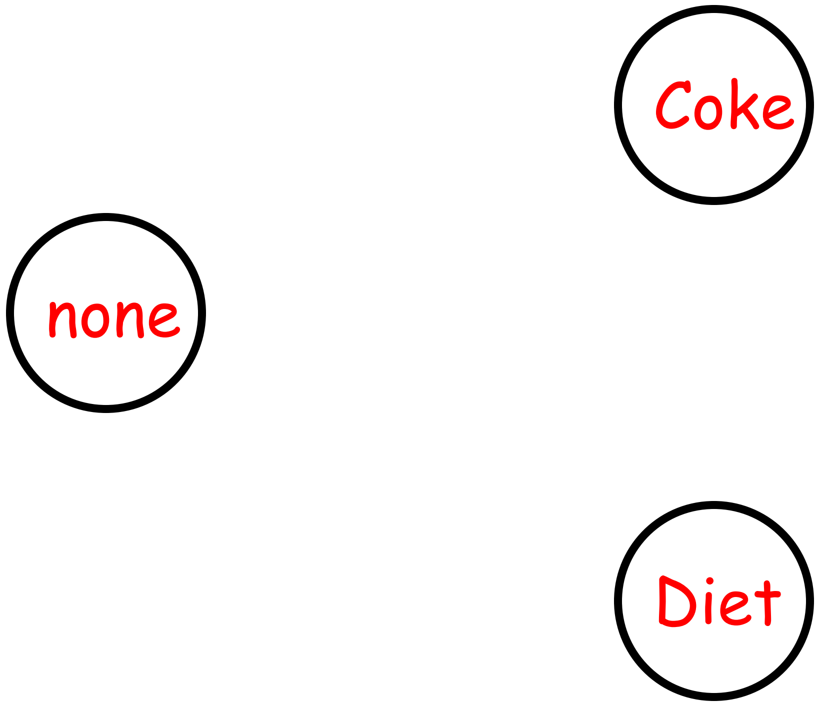
Soda Dispenser



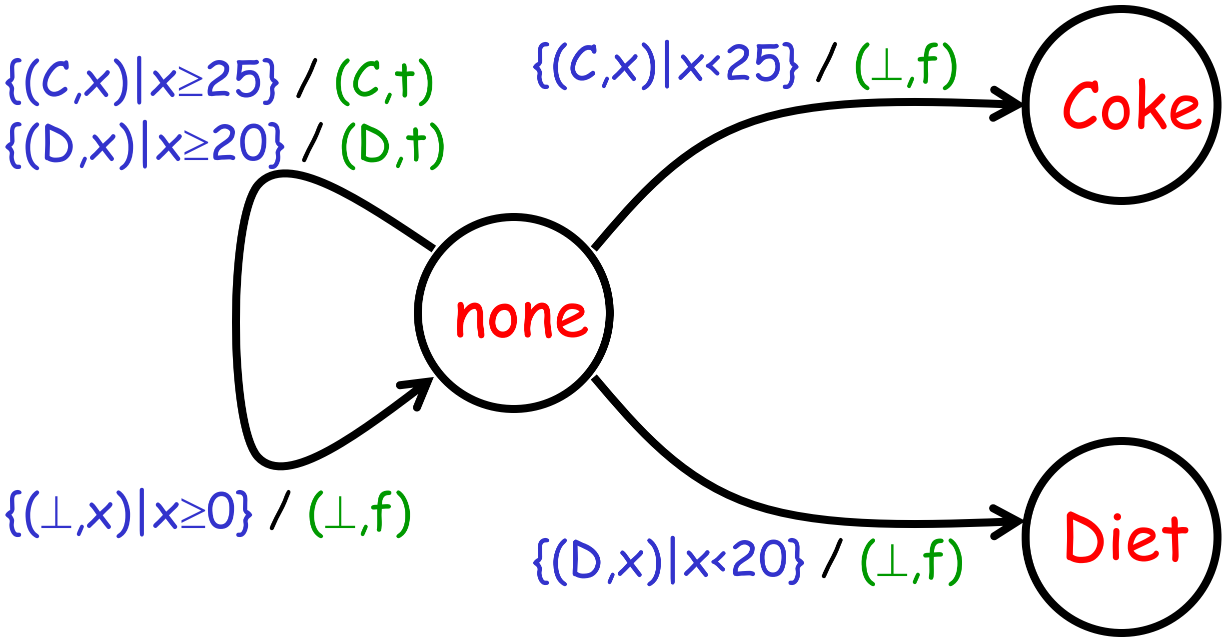
Let $Select = \{ selectCoke, selectDiet \}$.

Let $Dispense = \{ dispenseCoke, dispenseDiet \}$.

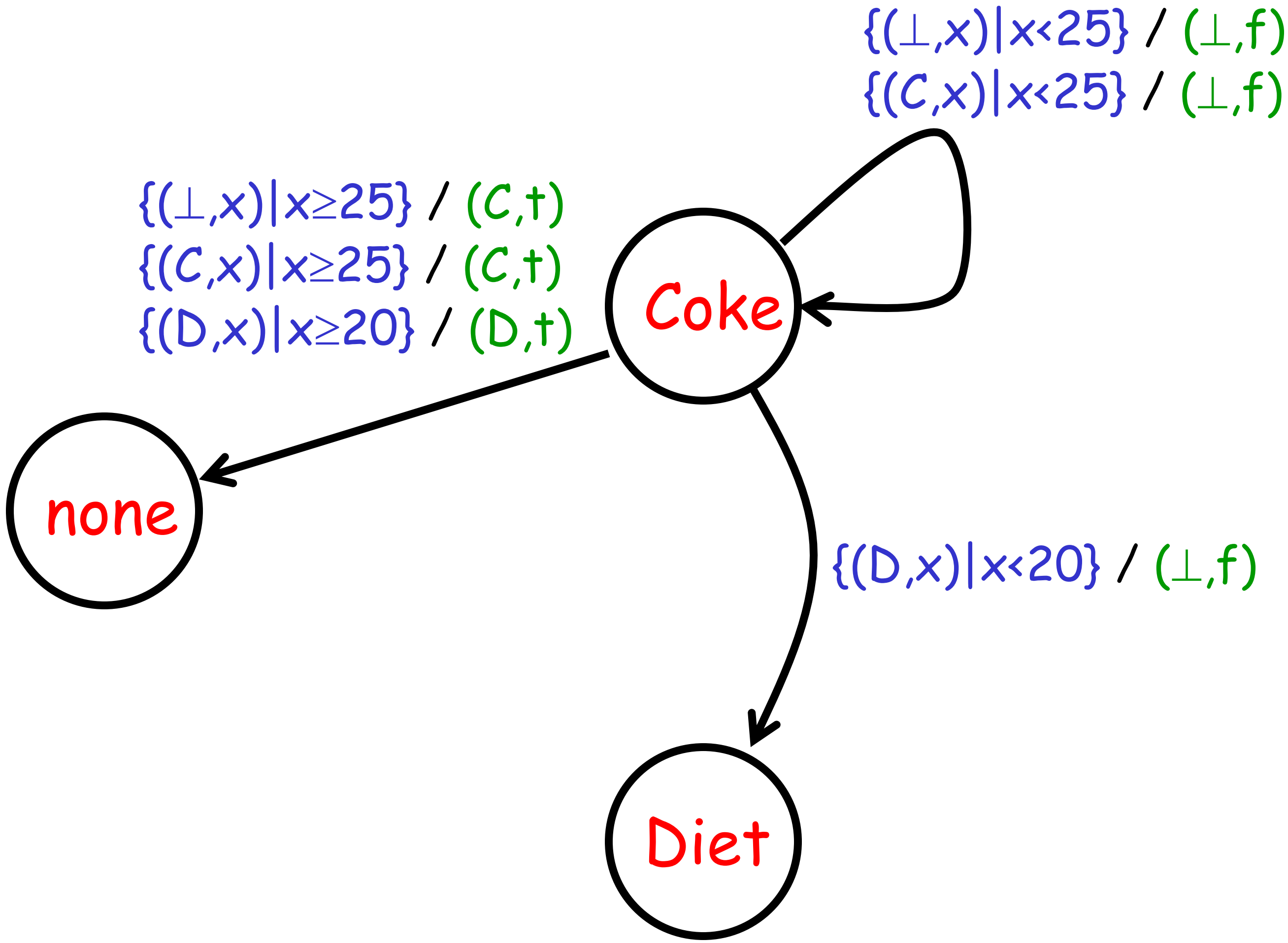
Soda Dispenser



Soda Dispenser

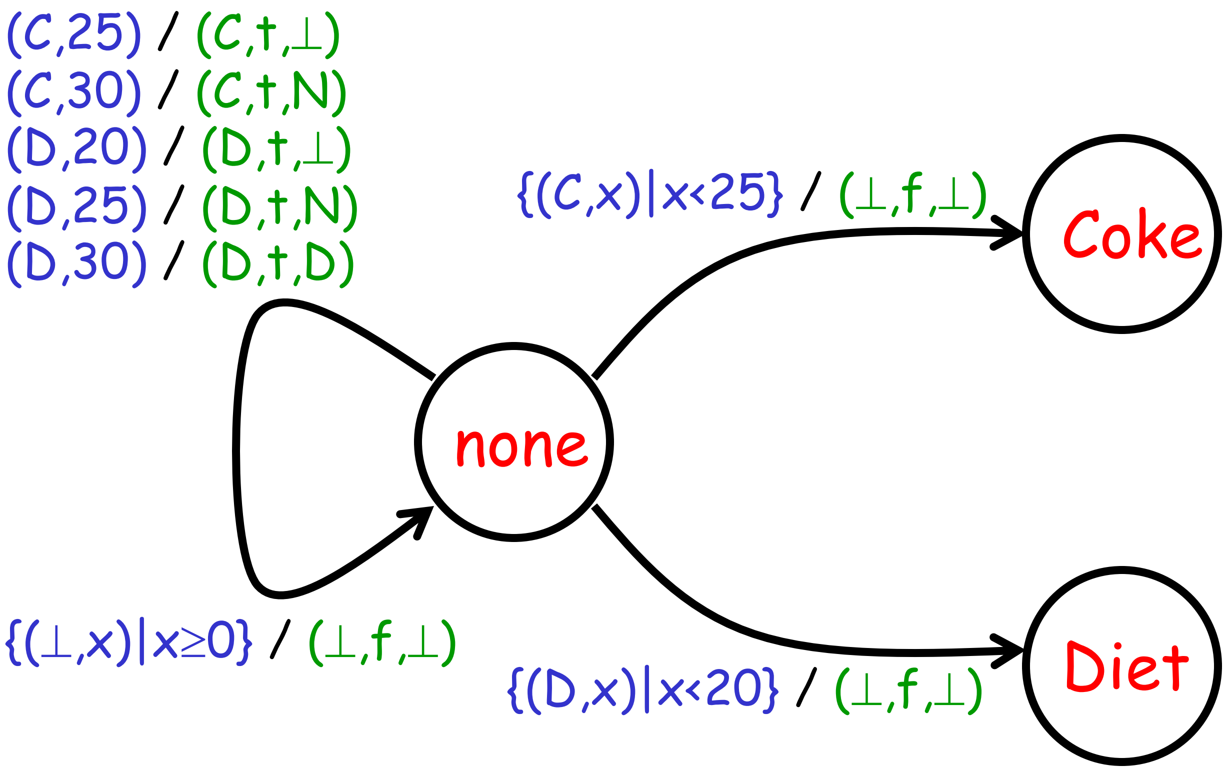


Soda Dispenser

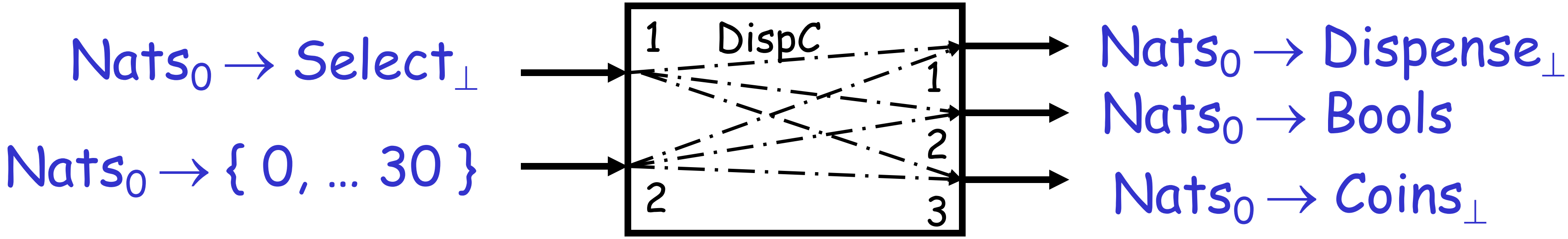


Soda Dispenser with Change

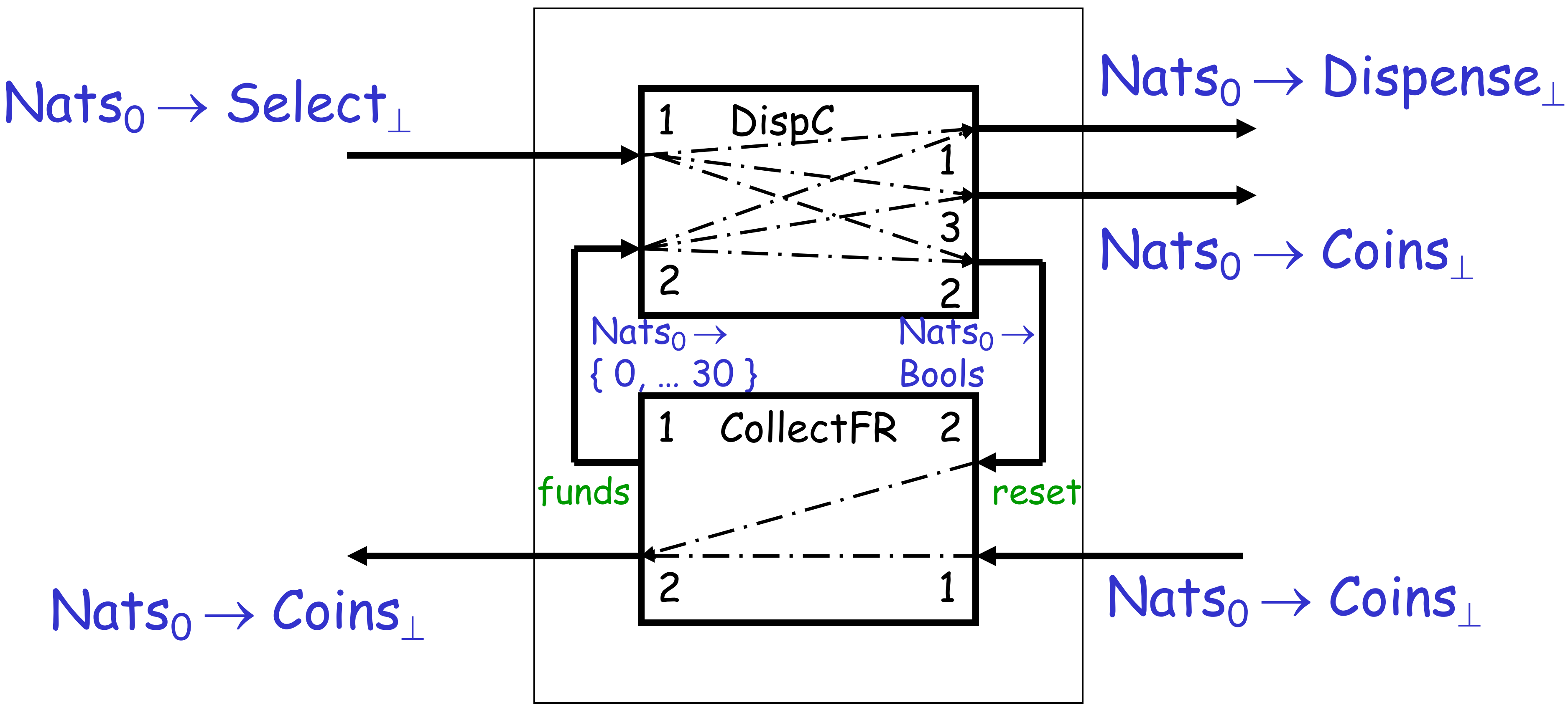
Soda Dispenser with Change



Soda Dispenser with Change



Vending Machine



State Space of Vending Machine

$$\{ 0, 5, 10, 15, 20, 25, 30 \} \times \{ \text{none}, \text{Coke}, \text{Diet} \}$$

21 states