Transition Diagrams

EECS 20

Lecture 9 (February 5, 2001)

Tom Henzinger

Discrete-Time Reactive System :

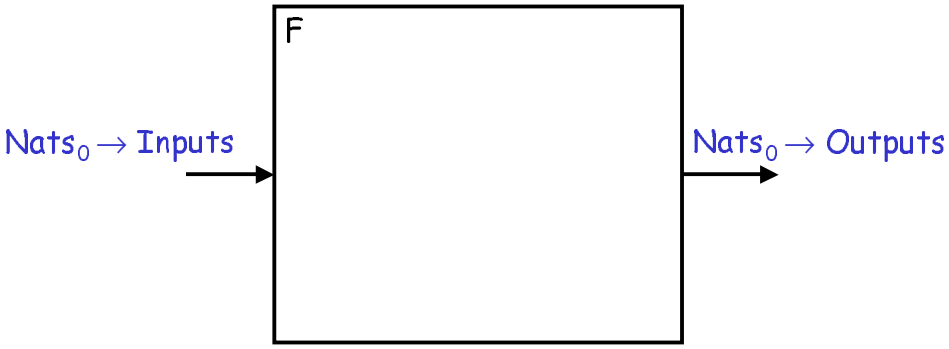
input/output function F

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

$Parity : [\ Nats_0 \rightarrow Bools\ ] \rightarrow [\ Nats_0 \rightarrow Bools\ ]$

such that $\forall\ x \in [\ Nats_0 \rightarrow Bools\ ]\ ,\ \forall\ y \in Nats_0\ ,$

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

where $trueValues\ (x,y) = \{\ z \in Nats_0 \mid z < y \wedge x\ (z) = true\ \}$

The Count System

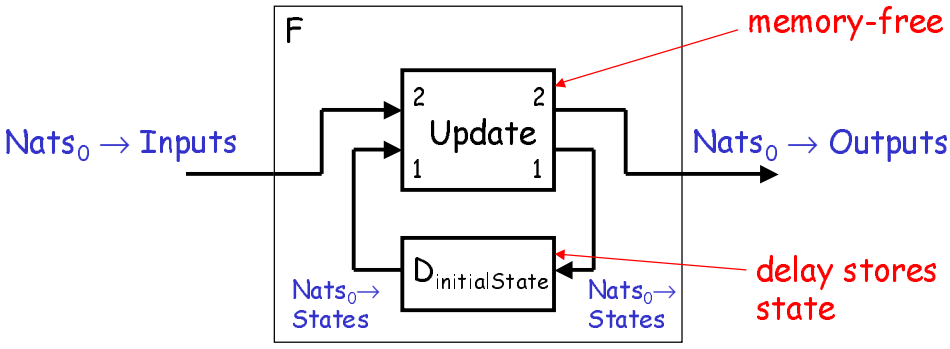
Count : $[\ Nats_0 \to Bools\ ] \to [\ Nats_0 \to Bools\ ]$

such that $\forall\ x \in [\ Nats_0 \to Bools\ ]\ ,\ \forall\ y \in Nats_0\ ,$

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

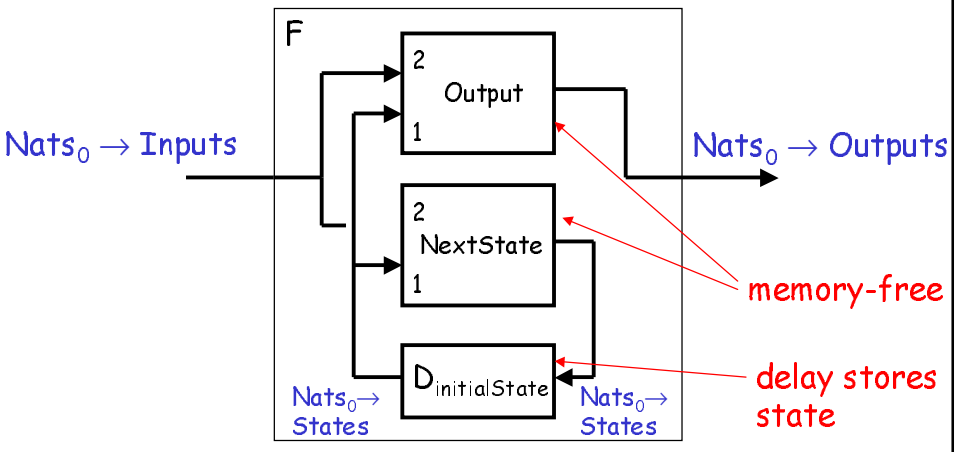
where $falseValues\ (x,y) = \{\ z \in Nats_0 \mid z < y \wedge x\ (z) = false\ \}$

State-Machine Implementation



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

$initialState \in States$



$nextState : States \times Inputs \rightarrow States$

$output : States \times Inputs \rightarrow Outputs$

$initialState \in States$

State-Machine Implementation of the Parity System

State after i-th input :

true, if first i inputs contain even number of true's

false, if first i inputs contain odd number of true's

Two states

Inputs = Bools

Outputs = Bools

States = Bools

initialState = true

nextState : States $\times$ Inputs $\rightarrow$ States

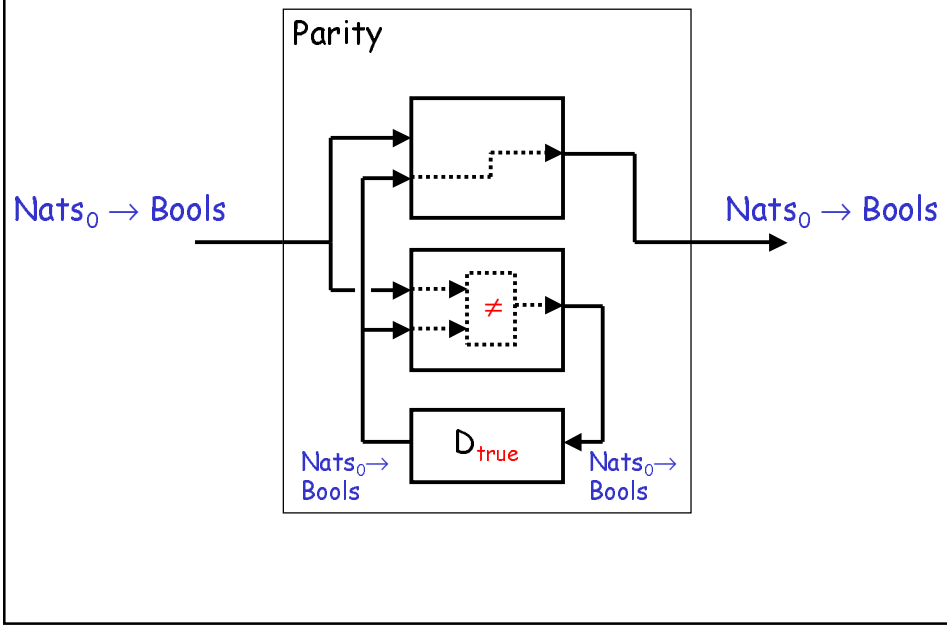
such that $\forall$ q $\in$ States, $\forall$ x $\in$ Inputs,

nextState (q,x) = (q $\neq$ x)

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

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

output (q,x) = q



State-Machine Implementation of the Count System

State after i-th input :

i, if first i inputs contain i more true's than false's

-i, if first i inputs contain i more false's than true's

Infinitely many states

Inputs = Bools

Outputs = Bools

States = Ints

initialState = 0

nextState : States $\times$ Inputs $\rightarrow$ States

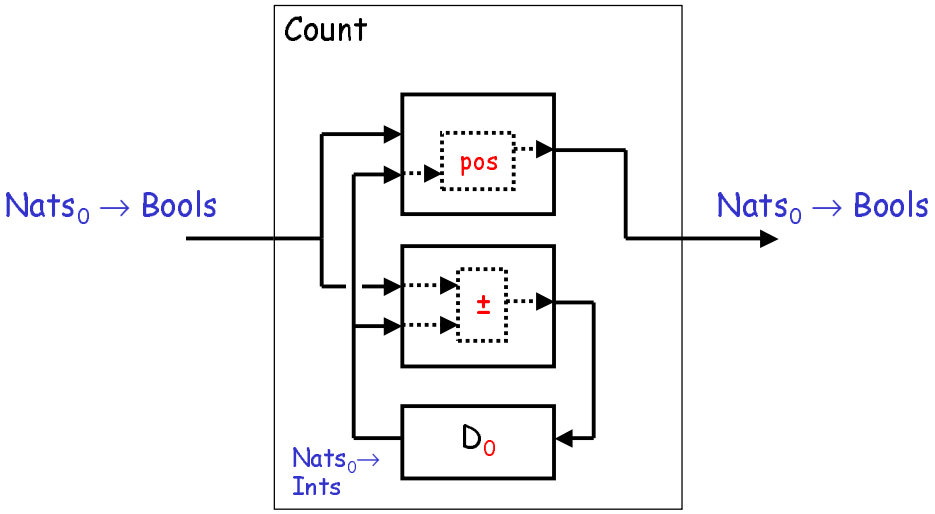
such that $\forall$ q $\in$ States, $\forall$ x $\in$ Inputs,

nextState (q,x) = $\pm$ (x,q)

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

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

output (q,x) = pos (q)



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 $\rightarrow$ 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

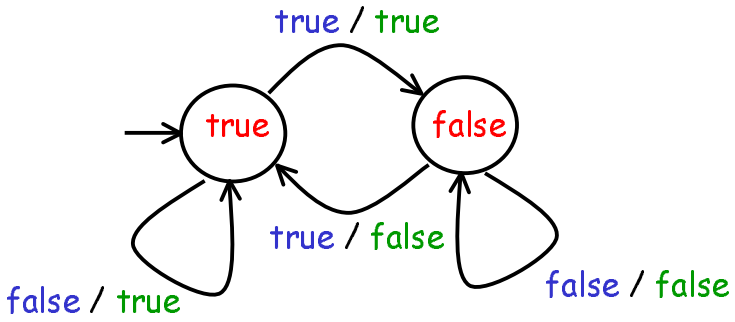
update : States $\times$ Inputs $\rightarrow$ States $\times$ Outputs

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

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

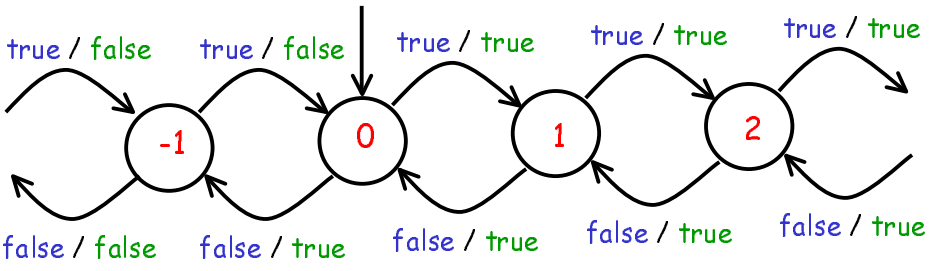
update $(q,x)_2$ = pos (q)

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 :

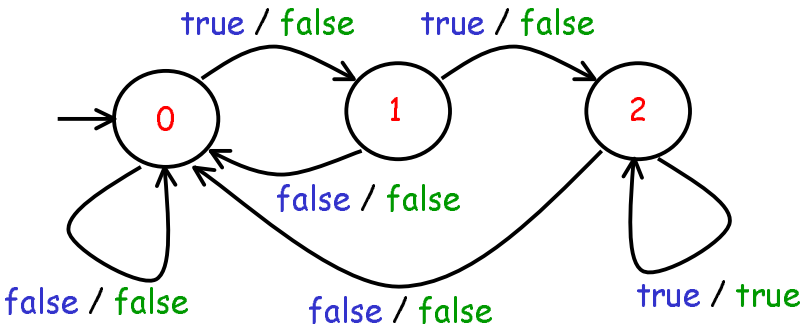
0, if i-th input is false (or i = 0)

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 :

States [Parity] = { true, false }

initialState [Parity] = true

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

output [Parity] (q,x) = q

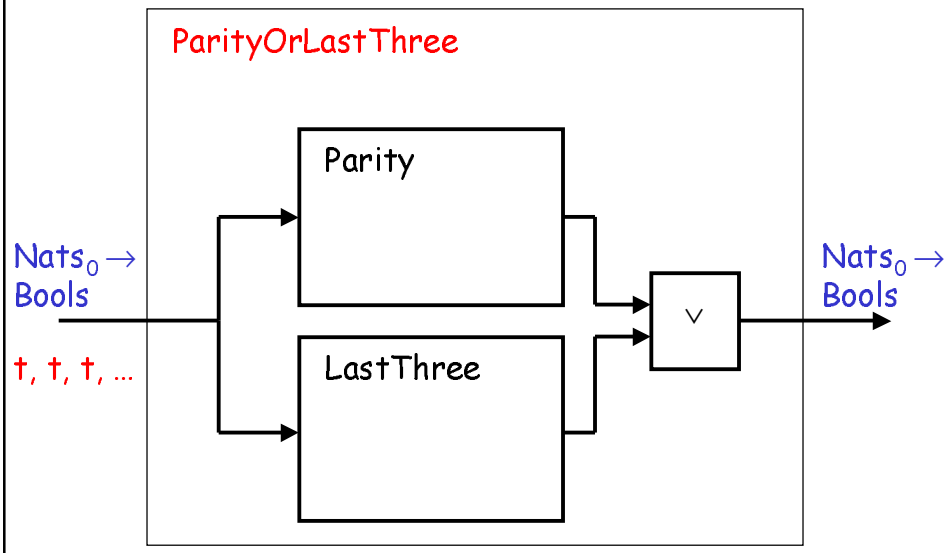
The LastThree System :

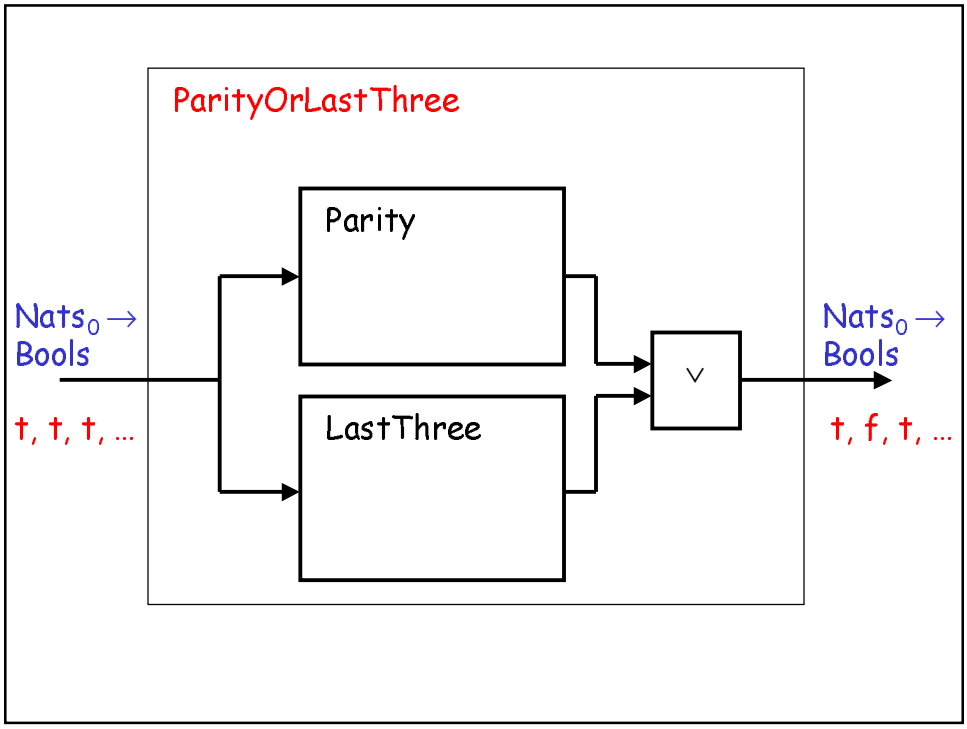
States [LastThree] = { 0, 1, 2 }

initialState [LastThree] = 0

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

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



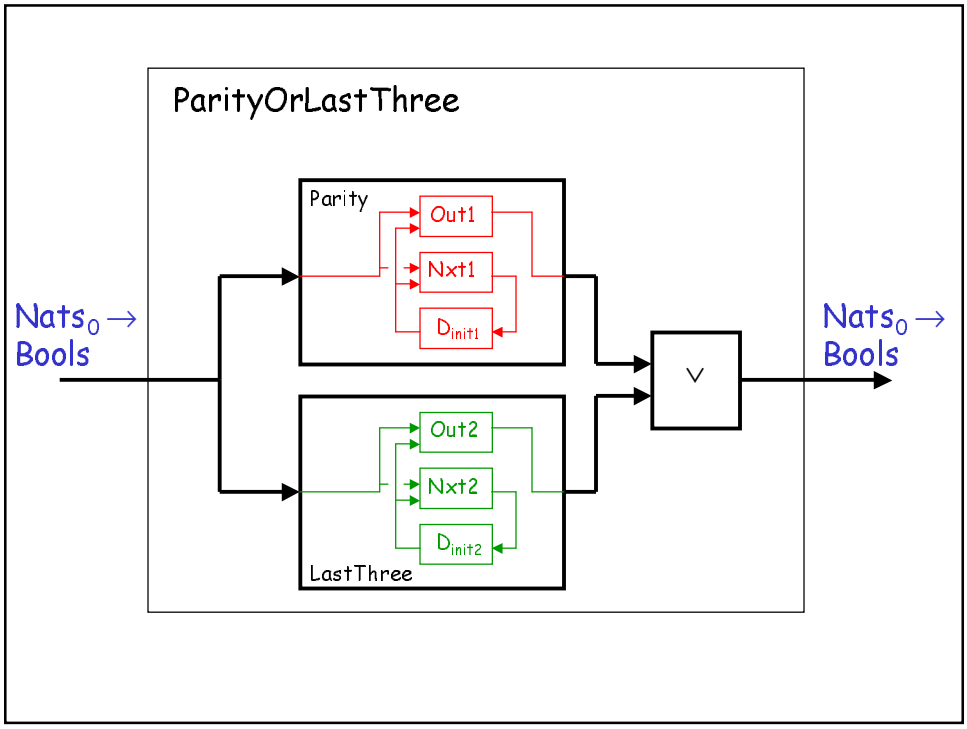


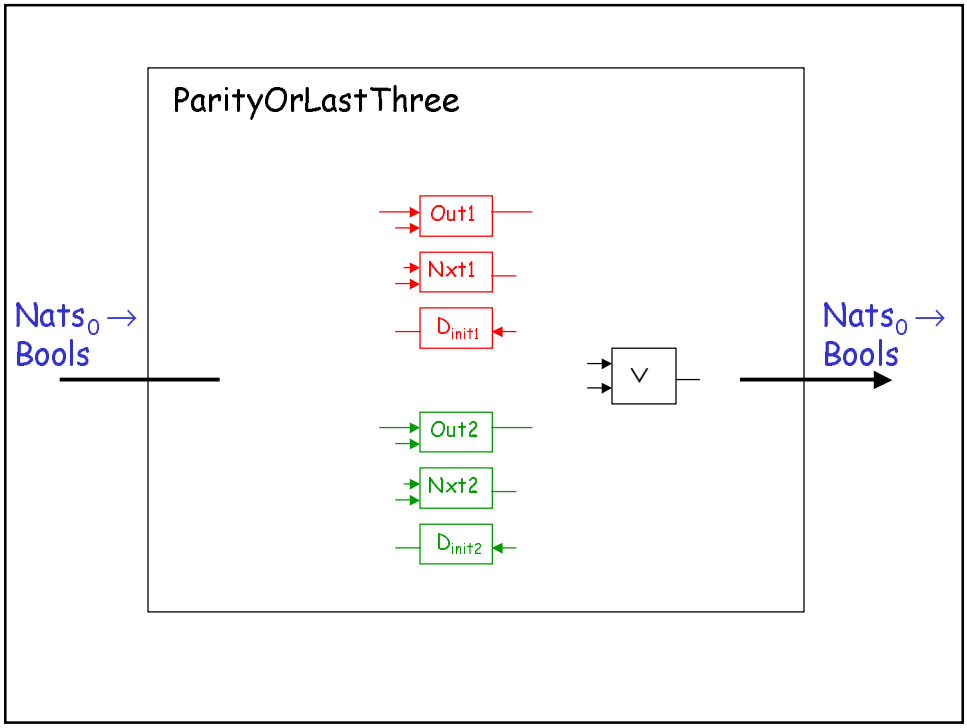
What is the state space of ParityOrLastThree ?

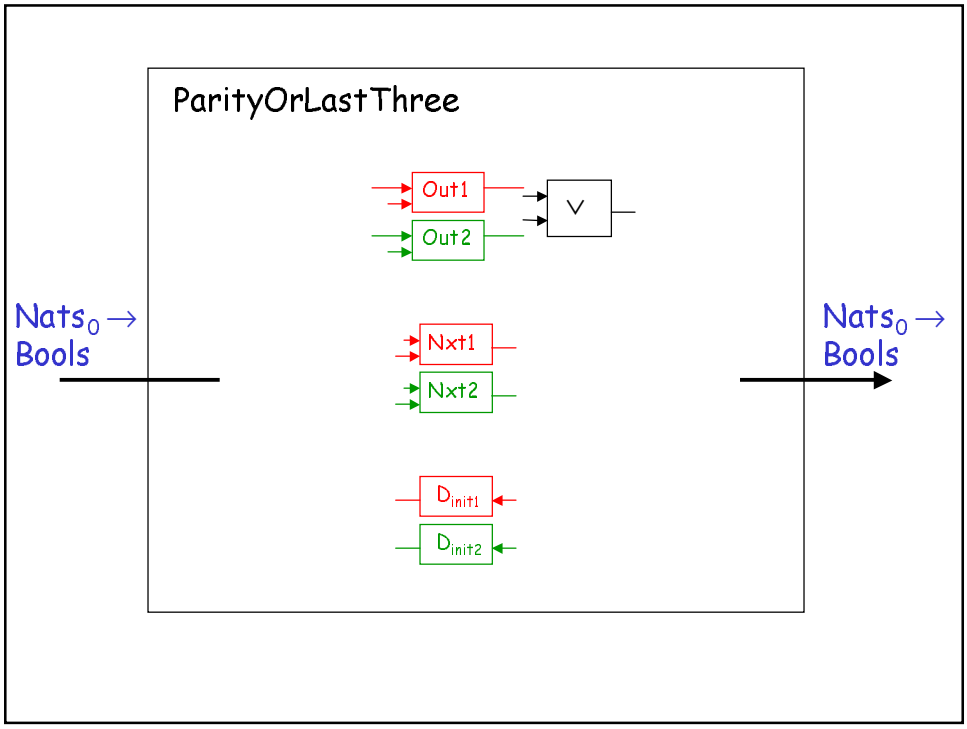
What is the initial state of ParityOrLastThree ?

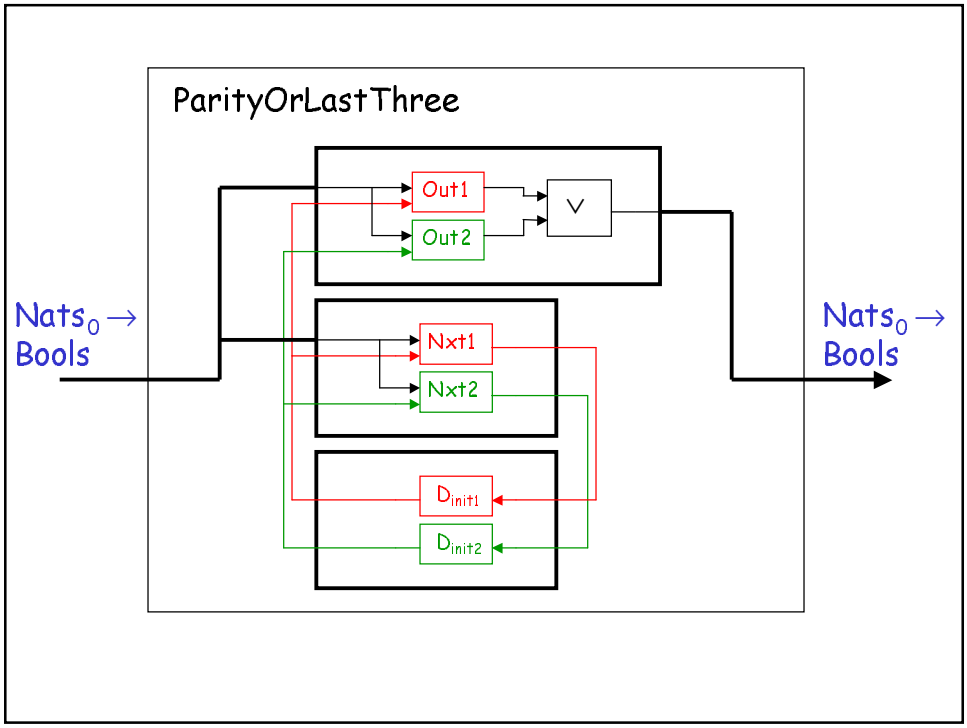
What is the nextState function of ParityOrLastThree ?

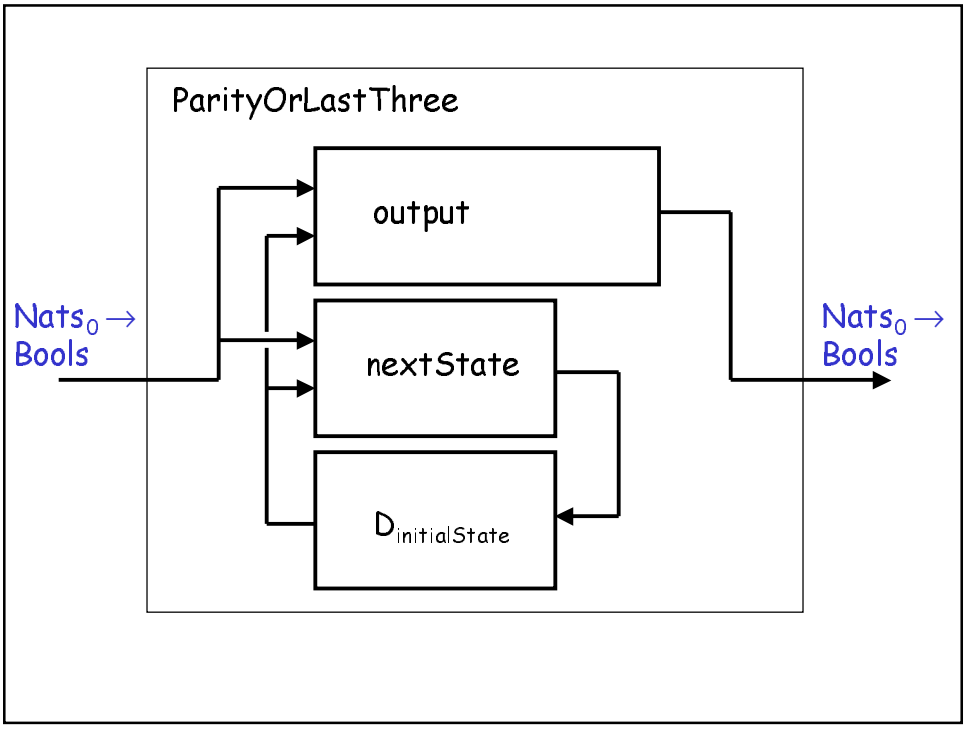
What is the output function of ParityOrLastThree ?


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


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


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


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



The ParityOrLastThree System

Inputs [ParityOrLastThree] = Bools

Outputs [ParityOrLastThree] = Bools

States [ParityOrLastThree]

= States [Parity] × States [LastThree]

= { true, false } × { 0, 1, 2 }

initialState [ParityOrLastThree]

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

= (true, 0)

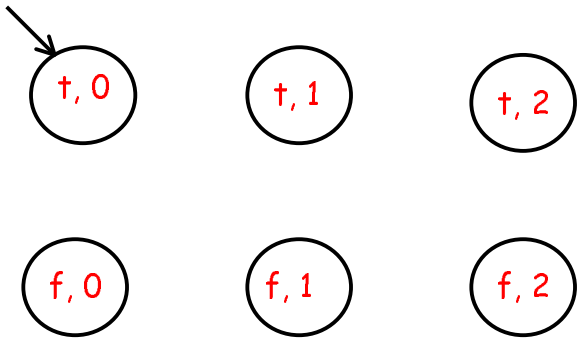
The ParityOrLastThree System, continued

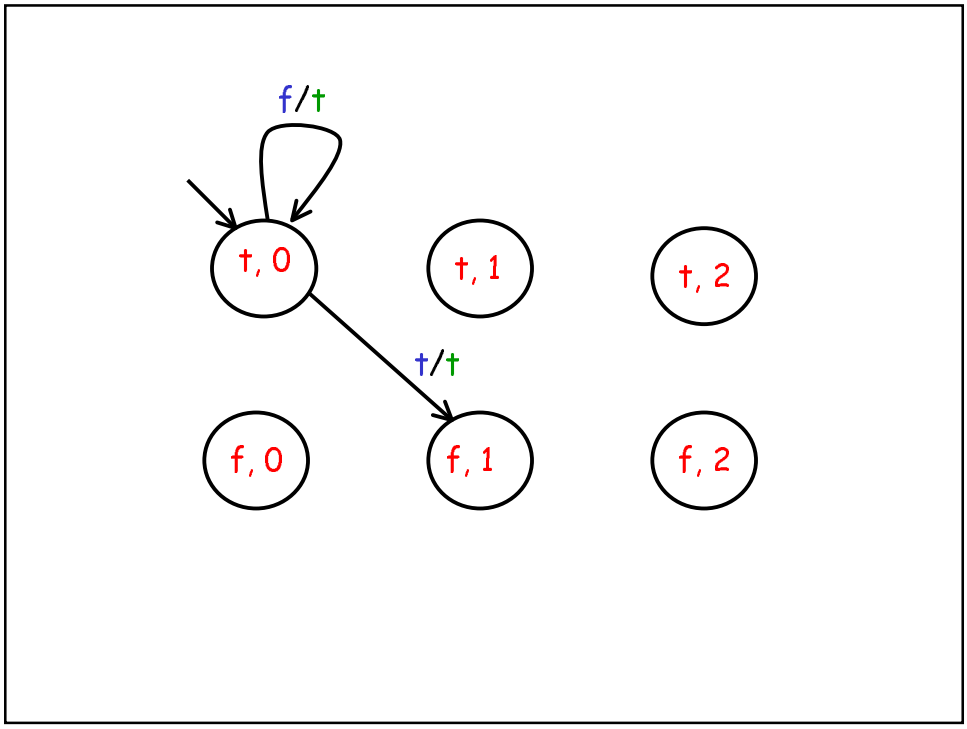
nextState [ParityOrLastThree] ((q1, q2) , x)

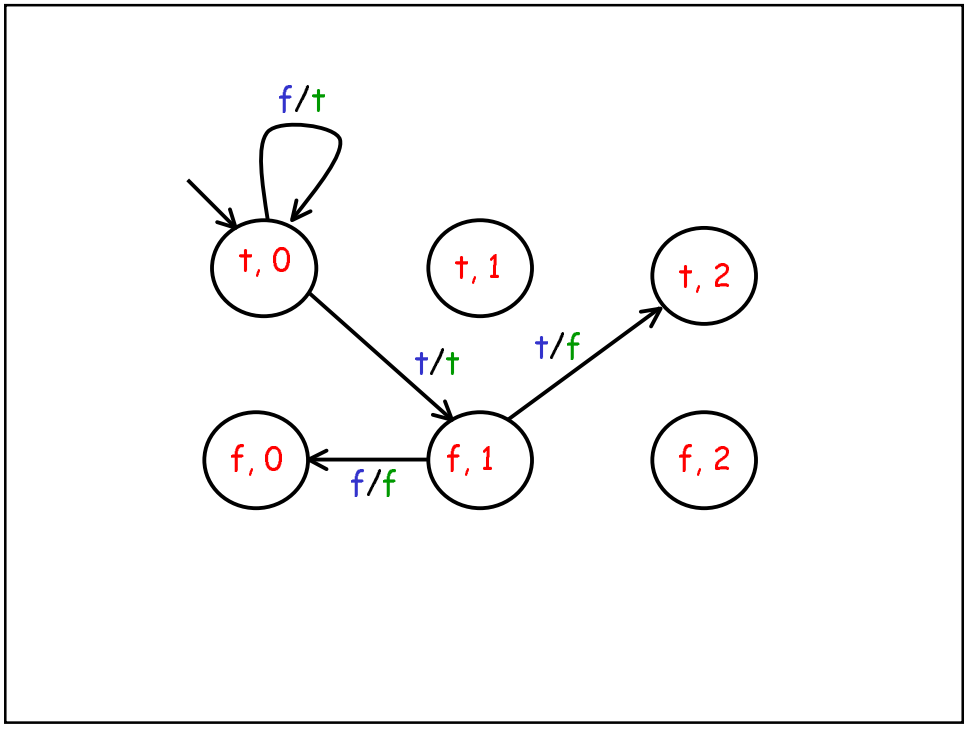
= (nextState [Parity] (q1, x) , nextState [LastThree] (q2, x))

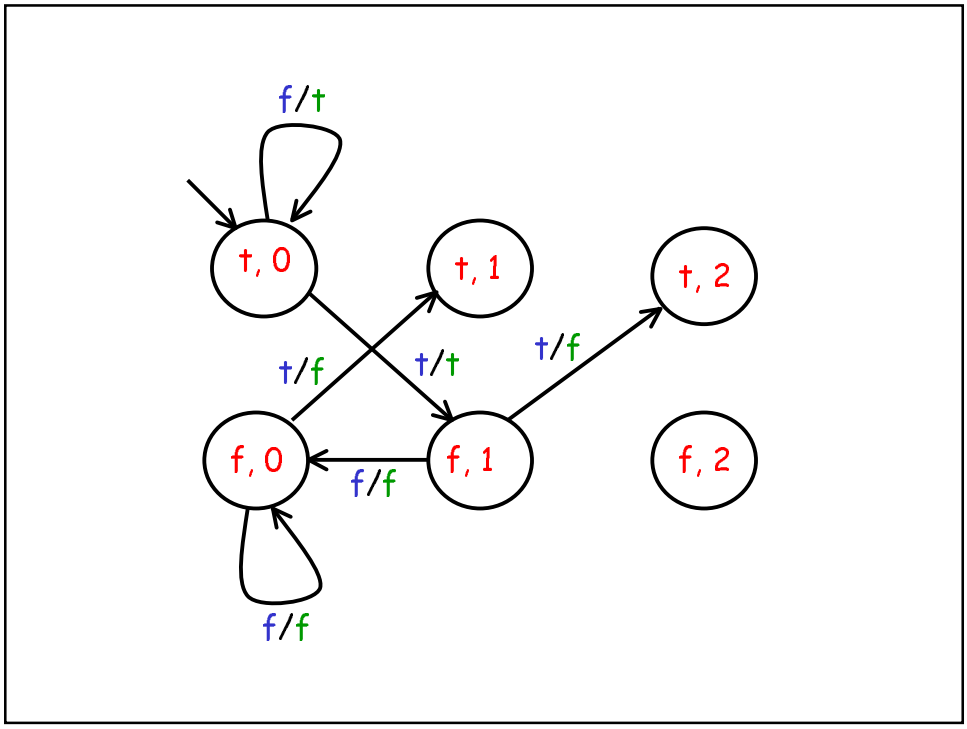
output [ParityOrLastThree] ((q1, q2) , x)

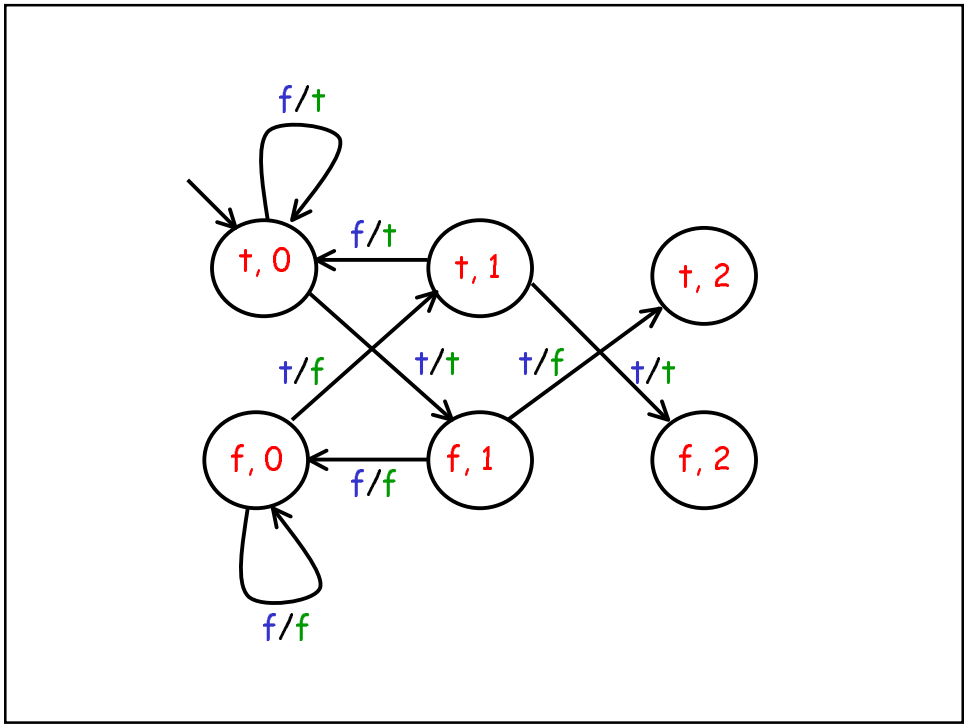
= output [Parity] (q1, x) ∨ output [LastThree] (q2, x)

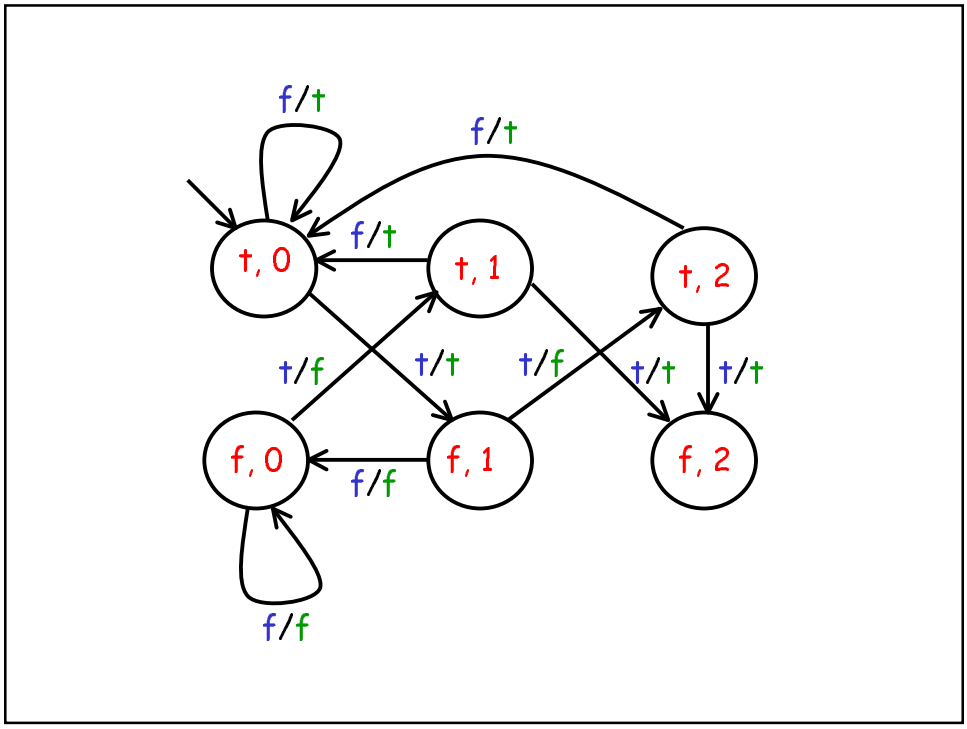


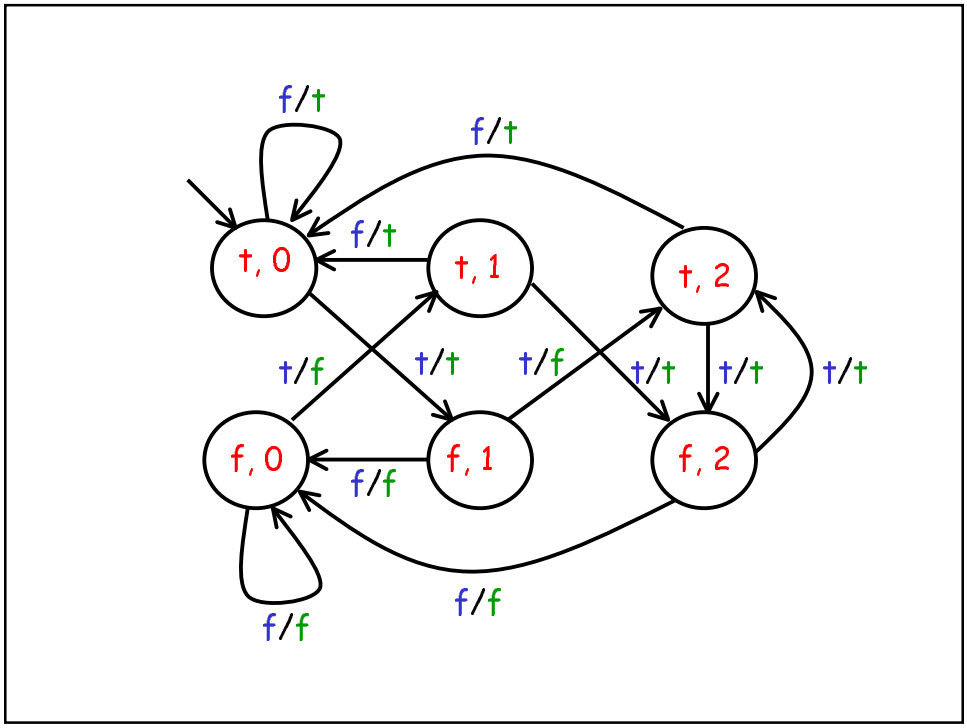

f/t
t, 0
t, 1
t, 2
t/t
f, 0
f, 1
f, 2


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
f/t
t, 0
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