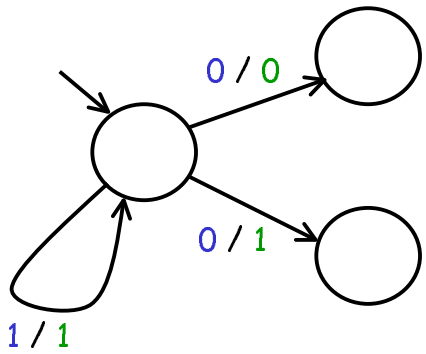
Nondeterminism

EECS 20

Lecture 14 (February 16, 2001)

Tom Henzinger

Nondeterminism

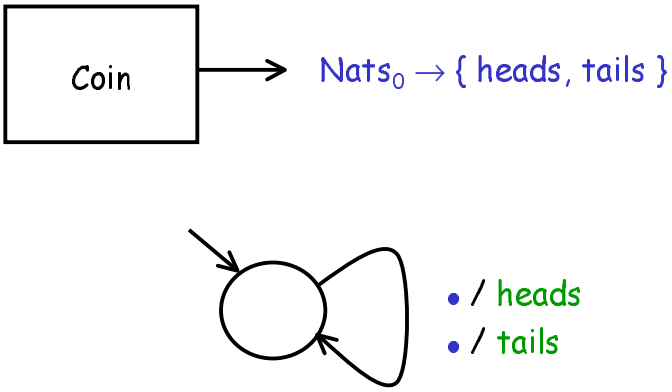


Nondeterminism

-modeling randomness
-modeling abstraction
-modeling uncertainty
-modeling properties

Modeling Randomness: Coin Tossing

Modeling Randomness: Coin Tossing



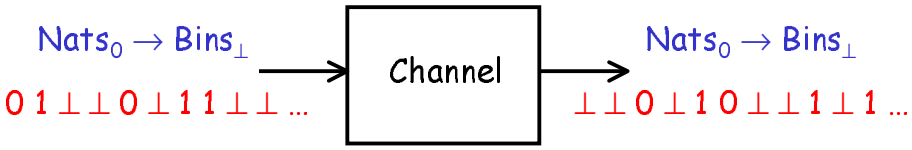
One possible behavior

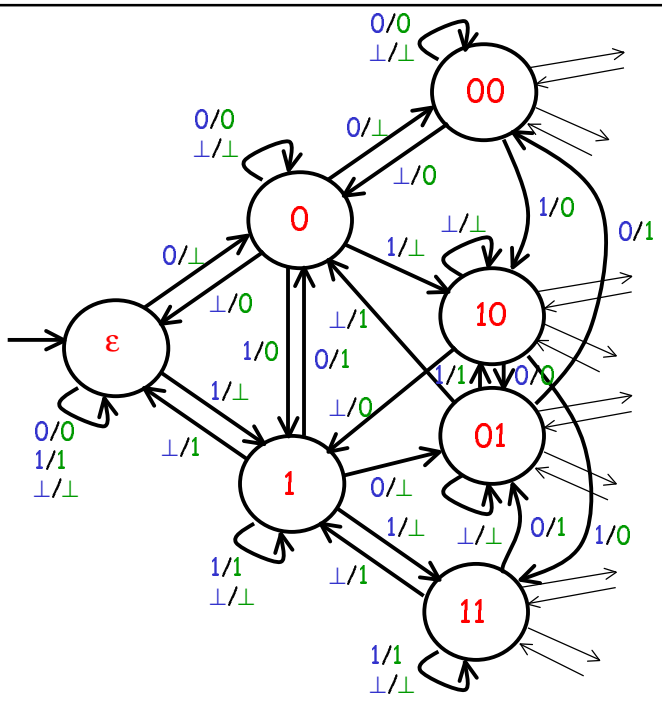
Time	0	1	2	3	4
Input	•	•	•	•	•
Output	heads	heads	tails	heads	tails

Another possible behavior

Time	0	1	2	3	4
Input	•	•	•	•	•
Output	tails	heads	tails	heads	heads

Modeling Abstraction: Channel Latency





State = channel contents

One possible run

Time	0	1	2	3	4	5
Input	0	1	⊥	0	⊥	⊥
Output	⊥	⊥	⊥	0	⊥	1
State	ε	0	10	10	01	01	0

One possible run

Time	0	1	2	3	4	5
Input	0	1	⊥	0	⊥	⊥
Output	⊥	⊥	⊥	0	⊥	1
State	ε	0	10	10	01	01	0

Corresponding behavior

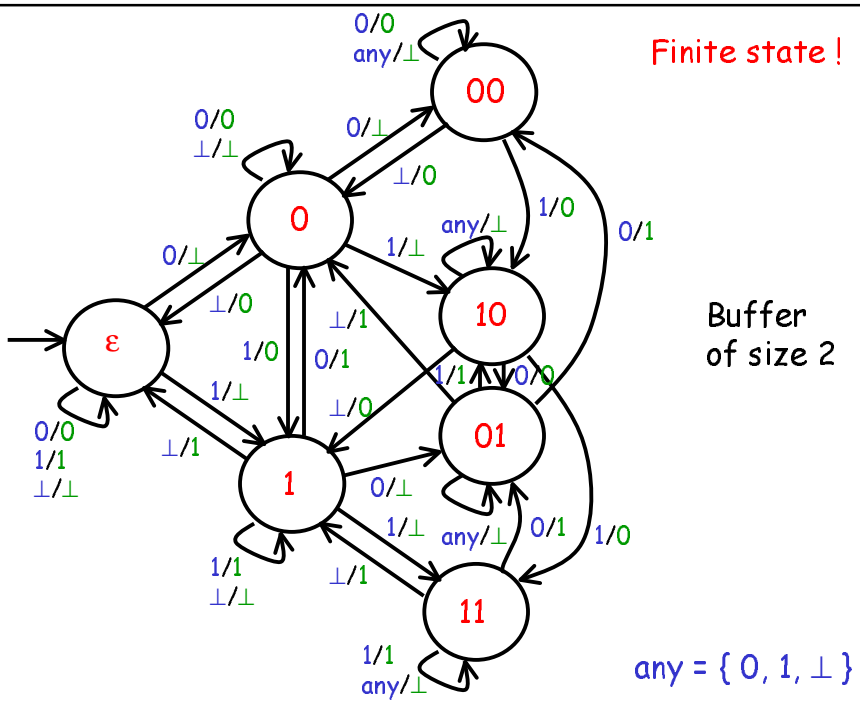
Time	0	1	2	3	4	5
Input	0	1	⊥	0	⊥	⊥
Output	⊥	⊥	⊥	0	⊥	1

One possible run

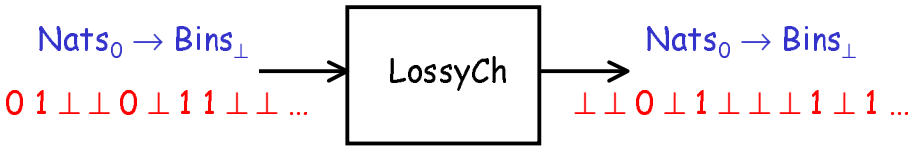
Time	0	1	2	3	4	5
Input	0	1	⊥	0	⊥	⊥
Output	⊥	⊥	⊥	0	⊥	1
State	ε	0	10	10	01	01	0

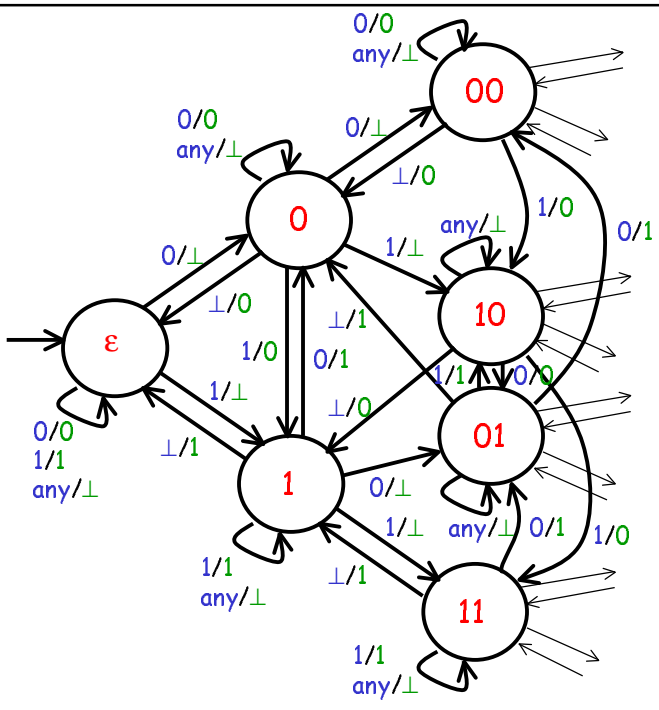
Another possible run on the same input signal

Input	0	1	⊥	0	⊥	⊥
Output	⊥	⊥	0	1	⊥	0
State	ε	0	10	1	0	0	ε



Modeling Uncertainty: Lossy Channel





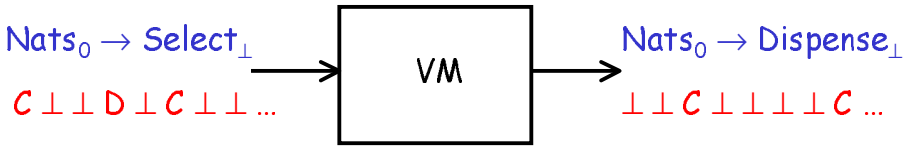
One possible run

Time	0	1	2	3	4	5	
Input	0	1	⊥	0	⊥	⊥	
Output	⊥	⊥	⊥	0	⊥	0	
State	ε	0	0	0	0	0	ε

Another possible run on the same input signal

Input	0	1	⊥	0	⊥	⊥	
Output	⊥	⊥	⊥	⊥	0	⊥	
State	ε	ε	ε	ε	0	ε	ε

Modeling Properties: Vending Machine

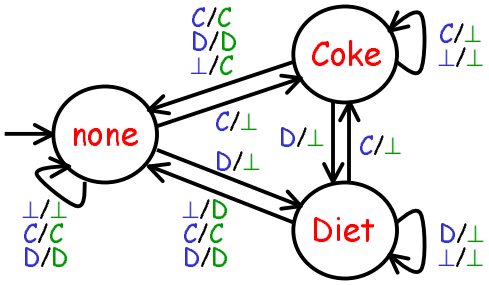


Select = Dispense = { Coke, Diet }

"No-unrequested-soda" property:

Whenever the machine dispenses Coke, then the most recent request was for Coke;

whenever the machine dispenses Diet, then the most recent request was for Diet.



One possible run

Time	0	1	2	3	4	5
Input	C	⊥	⊥	⊥	D	⊥
Output	⊥	⊥	⊥	C	⊥	D
State	n	C	C	C	n	D	n

Another possible run on the same input signal

Input	C	⊥	⊥	⊥	D	⊥
Output	⊥	⊥	⊥	⊥	⊥	D
State	n	C	C	C	C	D	n

Deterministic Reactive System:

for every input signal, there is exactly one output signal.

Nondeterministic Reactive System:

for every input signal, there is one or more output signals.

Deterministic Reactive System : function

$DetSys : [\ Time \to Inputs\ ] \to [\ Time \to Outputs\ ]$

Nondeterministic Reactive System : relation

$NondetSys \subseteq [\ Time \to Inputs\ ] \times [\ Time \to Outputs\ ]$

such that $\forall\ x \in [\ Time \to Inputs\ ]$,
$\exists\ y \in [\ Time \to Outputs\ ],\ (x,y) \in NondetSys$

Every pair $(x,y) \in$ NondetSys is called a

behavior

of the nondeterministic reactive system NondetSys .

S1 is a more detailed description of S2;

S2 is an abstraction or property of S1.

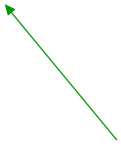
System S1 refines system S2

iff

1. Time [S1] = Time [S2] ,
2. Inputs [S1] = Inputs [S2] ,
3. Outputs [S1] = Outputs [S2] ,
4. Behaviors [S1] $\subseteq$ Behaviors [S2] .

S1 refines S2

Buffer of size 2	Arbitrary channel
Vending machine	No-unrequested-soda property
Fair coin	Nondeterministic coin



No output signal heads, heads, heads, heads, …
or tails, tails, tails, tails, …

Systems S1 and S2 are equivalent

iff

1. Time [S1] = Time [S2] ,
2. Inputs [S1] = Inputs [S2] ,
3. Outputs [S1] = Outputs [S2] ,
4. Behaviors [S1] = Behaviors [S2] .

Deterministic causal discrete-time reactive systems
can be implemented by
(deterministic) state machines.

Nondeterministic causal discrete-time reactive systems
can be implemented by
nondeterministic state machines.

Deterministic State Machine

Inputs

Outputs

States

initialState $\in$ States

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

Nondeterministic State Machine

Inputs

Outputs

States

possibleInitialStates $\subseteq$ States

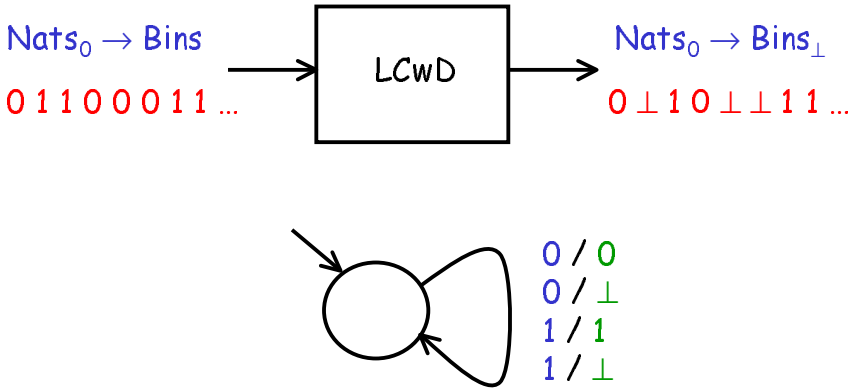
possibleUpdates :

$$\text{States} \times \text{Inputs} \rightarrow \mathcal{P}(\text{States} \times \text{Outputs}) \setminus \emptyset$$

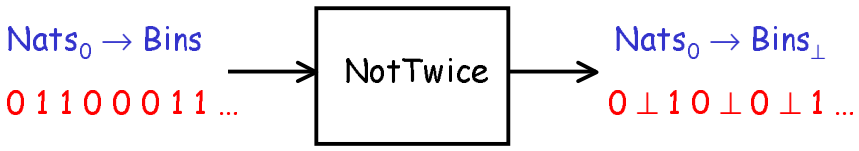
receptiveness (i.e., machine cannot prohibit an input)

Lossy Channel without Delay

Lossy Channel without Delay



Channel that never drops two in a row

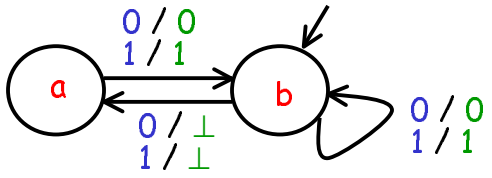


Channel that never drops two in a row

State between time t-1 and time t :

a the input at time t-1 was dropped

b the input at time t-1 was not dropped, or t = 0



Channel that never drops two in a row

Inputs = { 0, 1 }

Outputs = { 0, 1, ⊥ }

States = { a, b }

possibleInitialStates = { b }

possibleUpdates (a, 0) = { (b, 0) }
possibleUpdates (a, 1) = { (b, 1) }
possibleUpdates (b, 0) = { (b, 0) , (a, ⊥) }
possibleUpdates (b, 1) = { (b, 1) , (a, ⊥) }

Deterministic state machine:

for every input stream, there is exactly one run.

Nondeterministic state machine:

for every input stream, there is one or more runs.

Every run generates an output stream, and
therefore every run gives rise to a behavior.