Transducive Systems

EECS 20

Lecture 6 (January 29, 2001)

Tom Henzinger

Continuous-Time Signals

sound
position

Discrete-Time Signals

sampledSound
movie

ContinuousTime = { x ∈ Reals | x ≥ 0 }

DiscreteTime = { 0, 1, 2, 3, 4, … }

continuousTimeSignal : ContinuousTime → Values

discreteTimeSignal : DiscreteTime → Values

continuousTimeSignal = evolution of values

discreteTimeSignal = stream of values

More Discrete-Time Signals

text : DiscreteTime → Chars

file : DiscreteTime → Bins

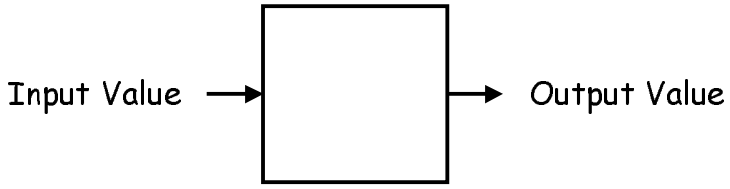
vendingMachine : DiscreteTime → Events

Chars = { a, b, c, … }

Bins = { 0, 1 }

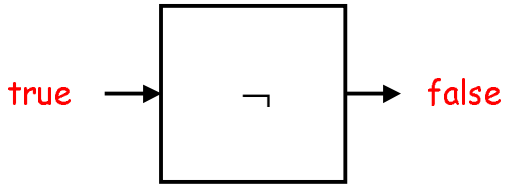
Events = { dropQuarter, requestCoke, deliverCoke, … }

Transducive or Combinational System



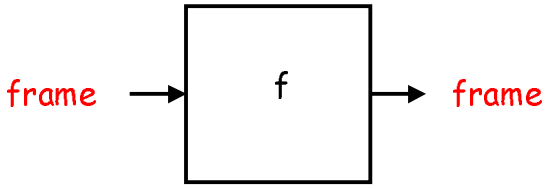
transduciveSystem : Values → Values

Transducive or Combinational System



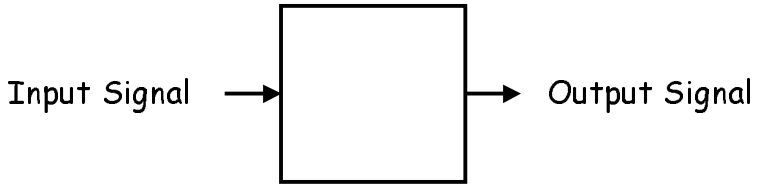
¬ : Bools → Bools

Transducive or Combinational System



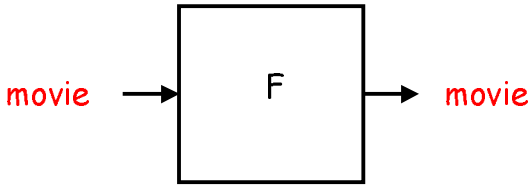
f : Frames → Frames

Reactive or Sequential System



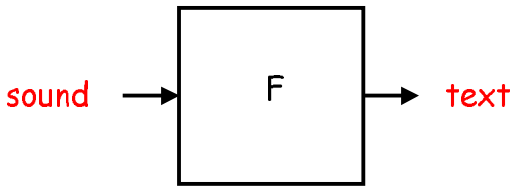
reactiveSystem : [Time → Values] → [Time → Values]

Reactive or Sequential System



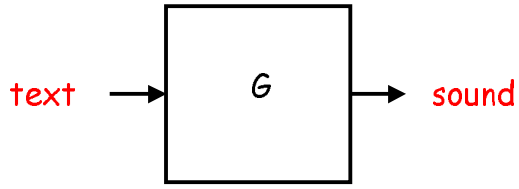
F : [Time → Frames] → [Time → Frames]

Reactive or Sequential System



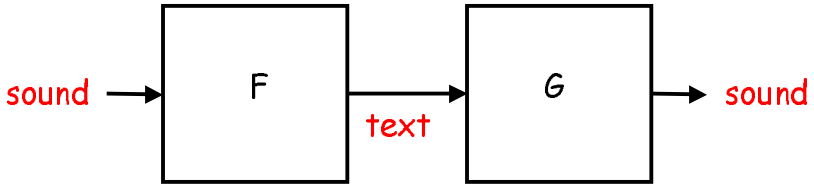
F: [ContinuousTime → Pressure] → [DiscreteTime → Chars]

Reactive or Sequential System

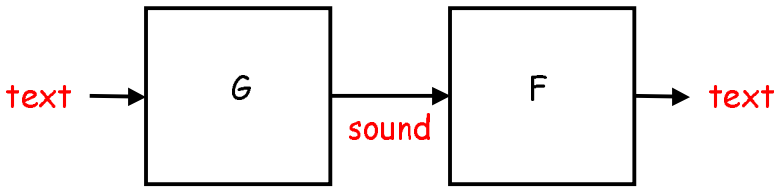


G: [DiscreteTime → Chars] → [ContinuousTime → Pressure]

Identity ?



Identity ?



Reactive Systems

1 Memory-free systems
2 Delays
3 Causality
4 Finite-memory systems
5 Infinite-memory systems

Every transducive system is a reactive system

$f : \text{Values} \rightarrow \text{Values}$

$F : [\text{Time} \rightarrow \text{Values}] \rightarrow [\text{Time} \rightarrow \text{Values}]$

such that $\forall x \in [\text{Time} \rightarrow \text{Values}], \forall y \in \text{Time},$

$(F(x))(y) = f(x(y))$

Every transducive system is a reactive system

$f : \text{Values} \rightarrow \text{Values}$

$F : [\text{Time} \rightarrow \text{Values}] \times \text{Time} \rightarrow \text{Values}$

such that $\forall x \in [\text{Time} \rightarrow \text{Values}], \forall y \in \text{Time},$

$F(x,y) = f(x(y))$

Every transducive system is a continuous-time reactive system

$$f : \text{Values} \rightarrow \text{Values}$$

$$F : [\text{ContTime} \rightarrow \text{Values}] \rightarrow [\text{ContTime} \rightarrow \text{Values}]$$

such that $\forall x \in [\text{ContTime} \rightarrow \text{Values}]$, $\forall y \in \text{ContTime}$,

$$(F(x))(y) = f(x(y))$$

$$\text{normalize} : \text{Reals} \rightarrow \text{Reals}$$

such that $\forall x \in \text{Reals}$, $\text{normalize}(x) = x - 50$

$$\text{Normalize} : [\text{Reals}_+ \rightarrow \text{Reals}] \rightarrow [\text{Reals}_+ \rightarrow \text{Reals}]$$

such that $\forall x \in [\text{Reals}_+ \rightarrow \text{Reals}]$, $\forall y \in \text{Reals}_+$,

$$(\text{Normalize}(x))(y) = \text{normalize}(x(y)) = x(y) - 50$$

Normalize (sin) : $Reals_+ \rightarrow Reals$

such that $\forall y \in Reals_+$, Normalize (sin) (y) = sin (y) - 50 .

Normalize (id) : $Reals_+ \rightarrow Reals$

such that $\forall y \in Reals_+$, Normalize (id) (y) = y - 50 .

Normalize (53) : $Reals_+ \rightarrow Reals$

such that $\forall y \in Reals_+$, Normalize (53) (y) = 3 .

Normalize (sin) : $Reals_+ \rightarrow Reals$

such that $\forall y \in Reals_+$, Normalize (sin) (y) = sin (y) - 50 .

Normalize (id) : $Reals_+ \rightarrow Reals$

such that $\forall y \in Reals_+$, Normalize (id) (y) = y - 50 .

Normalize (53) : $Reals_+ \rightarrow Reals$

such that Normalize (53) = 3 .

$$\text{trunc} : \text{Reals} \rightarrow \text{Reals}$$

$$\text{such that } \forall x \in \text{Reals}, \quad \text{trunc}(x) = \begin{cases} 256 & \text{if } x > 256 \\ x & \text{if } -256 \le x \le 256 \\ -256 & \text{if } x < -256 \end{cases}$$

$$\text{Trunc} : [\text{Reals}_+ \rightarrow \text{Reals}] \rightarrow [\text{Reals}_+ \rightarrow \text{Reals}]$$

$$\text{such that } \forall x \in [\text{Reals}_+ \rightarrow \text{Reals}], \ \forall y \in \text{Reals}_+ ,$$
$$(\text{Trunc}(x))(y) = \text{trunc}(x(y))$$

$$\text{Trunc (sin)} : \text{Reals}_+ \rightarrow \text{Reals}$$
$$\text{such that } \forall y \in \text{Reals}_+ , \quad \text{Trunc (sin)}(y) = \sin(y) .$$

$$\text{Trunc (id)} : \text{Reals}_+ \rightarrow \text{Reals}$$
$$\text{such that } \forall y \in \text{Reals}_+ , \quad \text{Trunc (id)}(y) = \begin{cases} y & \text{if } y < 256 \\ 256 & \text{if } y \ge 256 \end{cases}$$

$$\text{Trunc (500)} : \text{Reals}_+ \rightarrow \text{Reals}$$
$$\text{such that } \text{Trunc (500)} = 256 .$$

Every transducive system is a discrete-time reactive system

$$f : \text{Values} \rightarrow \text{Values}$$

$$F : [\ \text{DiscTime} \rightarrow \text{Values}\ ] \rightarrow [\ \text{DiscTime} \rightarrow \text{Values}\ ]$$

such that $\forall\, x \in [\ \text{DiscTime} \rightarrow \text{Values}\ ],\ \forall\, y \in \text{DiscTime},$

$$(\ F(x)\ )(y) = f(\ x(y)\ )$$

$$\text{quantize} : \text{Reals} \rightarrow \text{ComputerInts}$$

such that $\forall\, x \in \text{Reals},\ \text{quantize}(x) = \text{trunc}(\lfloor x \rfloor)$

$$\text{Quantize} : [\ \text{Nats}_0 \rightarrow \text{Reals}\ ] \rightarrow [\ \text{Nats}_0 \rightarrow \text{ComputerInts}\ ]$$

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

$$(\ \text{Quantize}(x)\ )(y) = \text{quantize}(\ x(y)\ ) = \text{trunc}(\lfloor x(y) \rfloor)$$

Quantize (squareroot) : $\text{Nats}_0 \rightarrow$ ComputerInts

such that Quantize (squareroot) =

{ (0,0), (1,1), (2,1), (3,1), (4,2), (5,2), … , (1000000,256), … } .

Quantize (id) : $\text{Nats}_0 \rightarrow$ ComputerInts

such that Quantize (id) =

{ (0,0), (1,1), (2,2), (3,3), … , (256,256), (257,256), … } .

Quantize (pi) : $\text{Nats}_0 \rightarrow$ ComputerInts

such that Quantize (pi) = 3 .

negate : Bools $\rightarrow$ Bools

such that $\forall x \in$ Bools, negate (x) = $\neg x$

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

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

(Negate (x)) (y) = negate (x (y))

= $\neg$ x (y)

alt : $Nats_0 \rightarrow Bools$

such that $\forall y \in Nats_0$, $alt(y) = \begin{cases} true & \text{if } y \text{ odd} \\ false & \text{if } y \text{ even} \end{cases}$

Negate (alt) : $Nats_0 \rightarrow Bools$

such that Negate (alt) =

{ (0,true), (1, false), (2, true), (3, false) , (4, true), ... } .

Negate (true) : $Nats_0 \rightarrow$ Bools

such that Negate (true) = false .

A reactive system

$F : [\ Time \rightarrow Values\ ] \rightarrow [\ Time \rightarrow Values\ ]$

is memory-free

iff

there exists a transducive system

$f : Values \rightarrow Values$

such that

$\forall x \in [\ Time \rightarrow Values\ ]$, $\forall y \in Time$,

$(\ F(x)\ )(y) = f(\ x(y)\ )$.