

Programs with time in a Discrete Runtime Environment

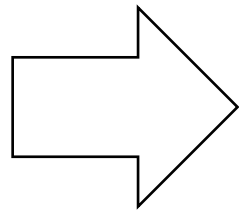
Shoji Yuen
(Nagoya University)

Yampa

- Yampa : DSL of Haskell for hybrid system
 - continuous and discrete transitions
- Signal functions : continuous behavior

$sf: SF\ a\ b$

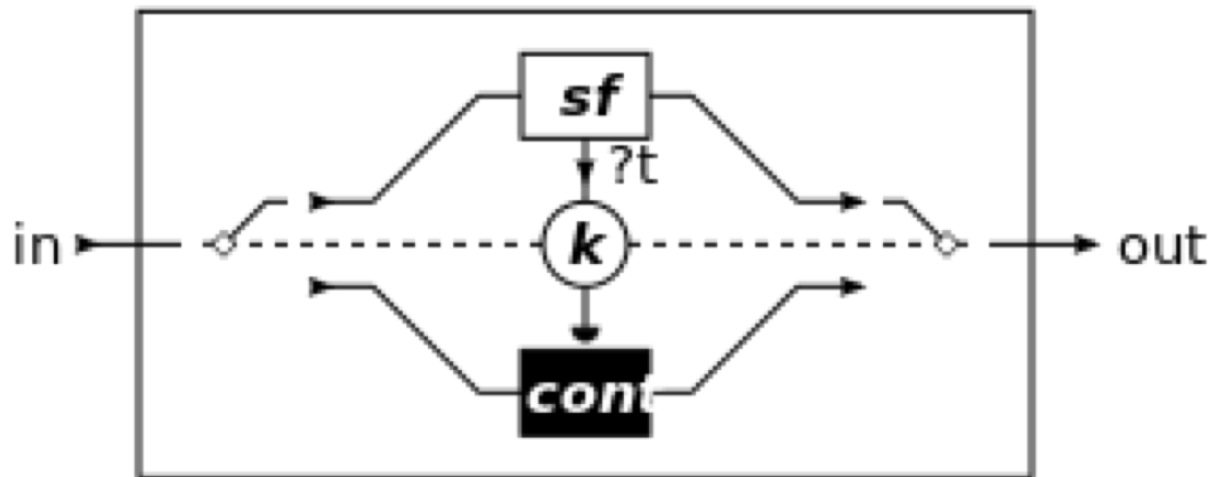
$$\left\{ \begin{array}{lcl} \frac{d^2 p}{dt^2} & = & 0 \\ \frac{dp}{dt} & = & \frac{dp}{dt} t + v_0 \\ p & = & \frac{dp}{dt} t + p_0 \end{array} \right.$$



```
v' <- integral -< 0
v <- arr (+ v0) -< v'
p' <- integral -< v
p <- arr (+ p0) -< p'
```

Yampa

- Events trigger discrete behavior
 - Event: truth change of a given predicate over the signal function
 - 'switch' function changes another signal function determined by the event



switch: $SF \text{ in } (out, \text{Event } t)$
-> $(t \rightarrow SF \text{ in } out)$
-> $SF \text{ in } out$

Yampa Program

```
1 moving p0 v0 = proc () -> do
2   v' <- integral -< 0
3   v <- arr (+ v0) -< v'
4   p' <- integral -< v
5   p <- arr (+ p0) -< p'
6   returnA -< (p,v)
7
8   Sf1      Sf2
9   hal px0 py0 vx0 vy0 = switch traj px0 py0 vx0 vy0) ssfunc
10  where traj px' py' vx' vy' = proc input -> do
11    (px,vx) <- moving px' vx' -< input
12    (py,vy) <- moving py' vy' -< input
13    event <- edge -< (px <= -10 || px >= 15 || py <= -10 || py >= 15)
14    returnA -< ((px,py,vx,vy),event 'tag' (px,py,vx,vy))
15  ssfunc (px',py',vx',vy') = if(px' <= -10 || px' >= 15)
16    then if(py' <= -10 || py' >= 15)
17      then hal px' py' (-vx') (-vy')
18      else hal px' py' (-vx') vy'
19    else if(py' <= -10 || py' >= 15)
20      then hal px' py' vx' (-vy')
21      else hal px' py' vx' vy'
```

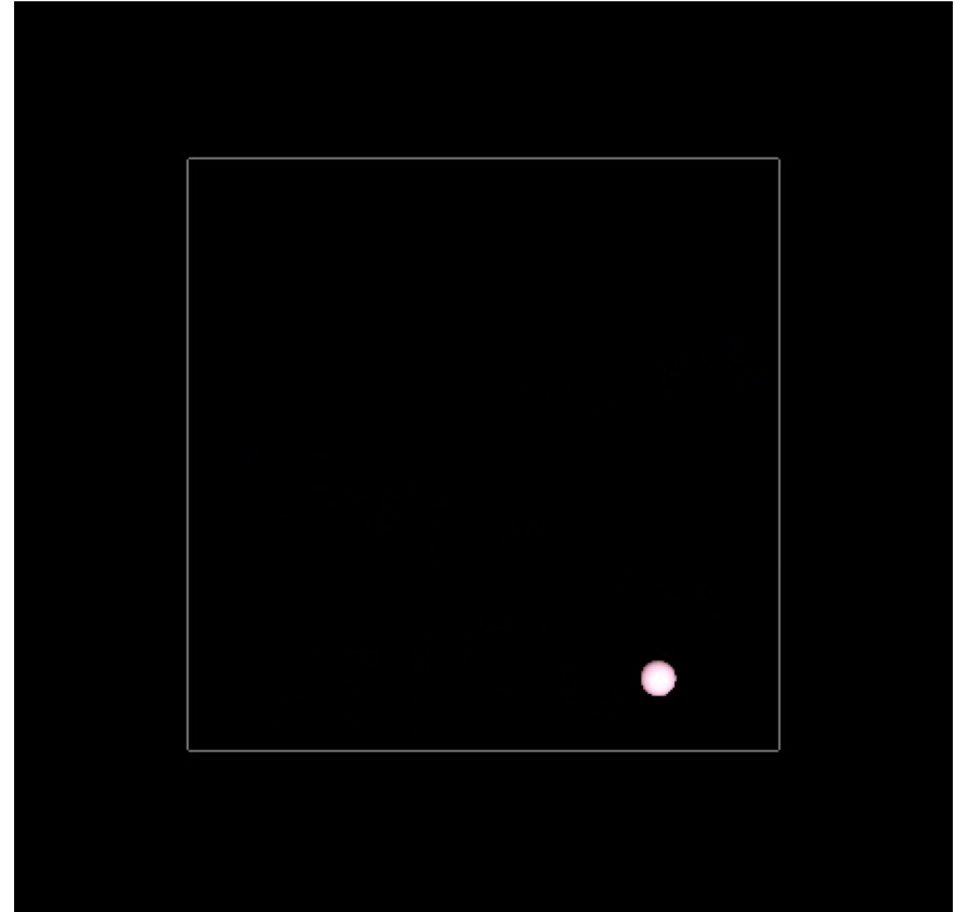
Update
Continuous Value

Event generation

New signal function

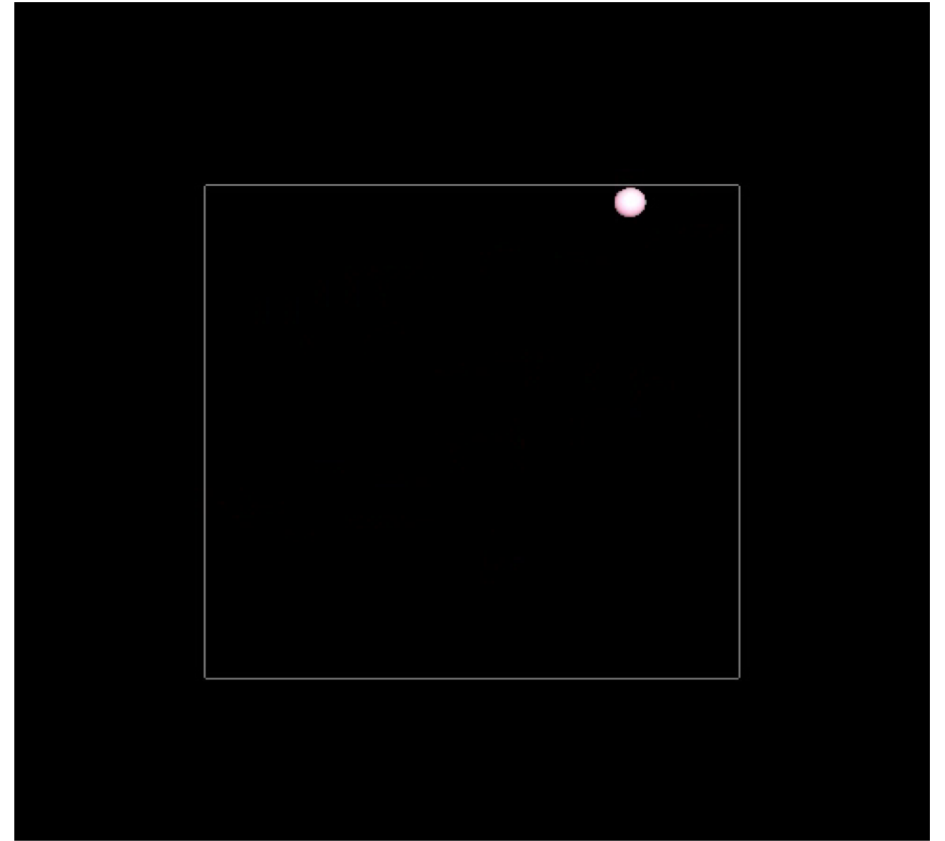
Example : Yampa Behavior

- Continuous behavior:
Ball Position
- Discrete behavior:
Bouncing on frame



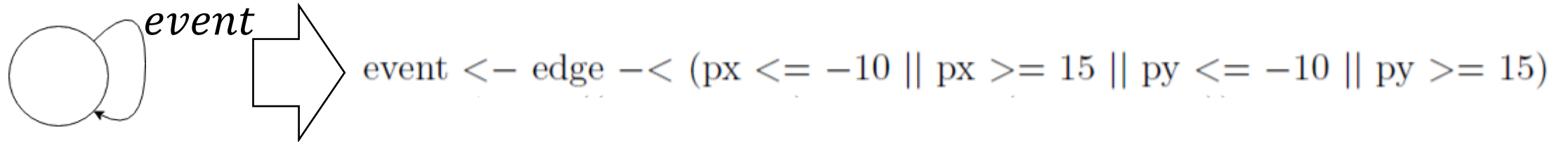
Unexpected Behavior in Yampa Program

- Ball gets out of frame
 - sampling causes this error
 - When ball reflect near the corner, this error occur
- Program can avoid the error
 - Programming technique without changing the semantics

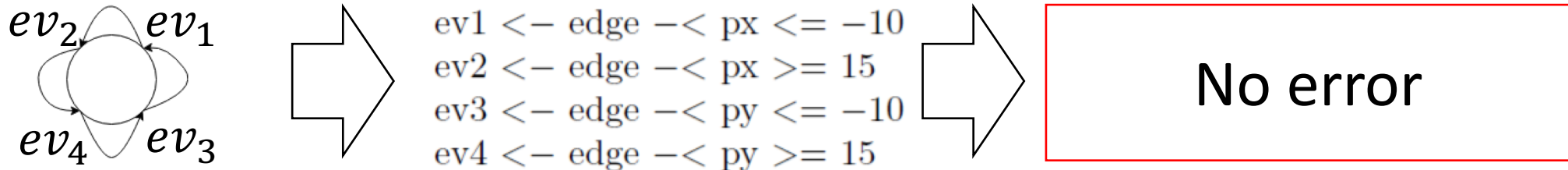


Simple Example

- HA1 : check for one frame



- HA4 : check for each frame



- HA1 = HA4

- $event = ev_1 \vee ev_2 \vee ev_3 \vee ev_4$

HA4 Program

```
1 moving p0 v0 = proc () -> do
2   v' <- integral -< 0
3   v <- arr (+ v0) -< v'
4   p' <- integral -< v
5   p <- arr (+ p0) -< p'
6   returnA -< (p,v)
```

Update
Continuous Value

```
7
8 ha4 px0 py0 vx0 vy0 = switch (traj px0 py0 vx0 vy0) ssfunc
```

Sf1

Sf2

```
9   where traj px' py' vx' vy' = proc input -> do
10     (px,vx) <- moving px' vx' -< input
11     (py,vy) <- moving py' vy' -< input
```

```
12     ev1 <- edge -< px <= -10
13     ev2 <- edge -< px >= 15
14     ev3 <- edge -< py <= -10
15     ev4 <- edge -< py >= 15
```

Event generation

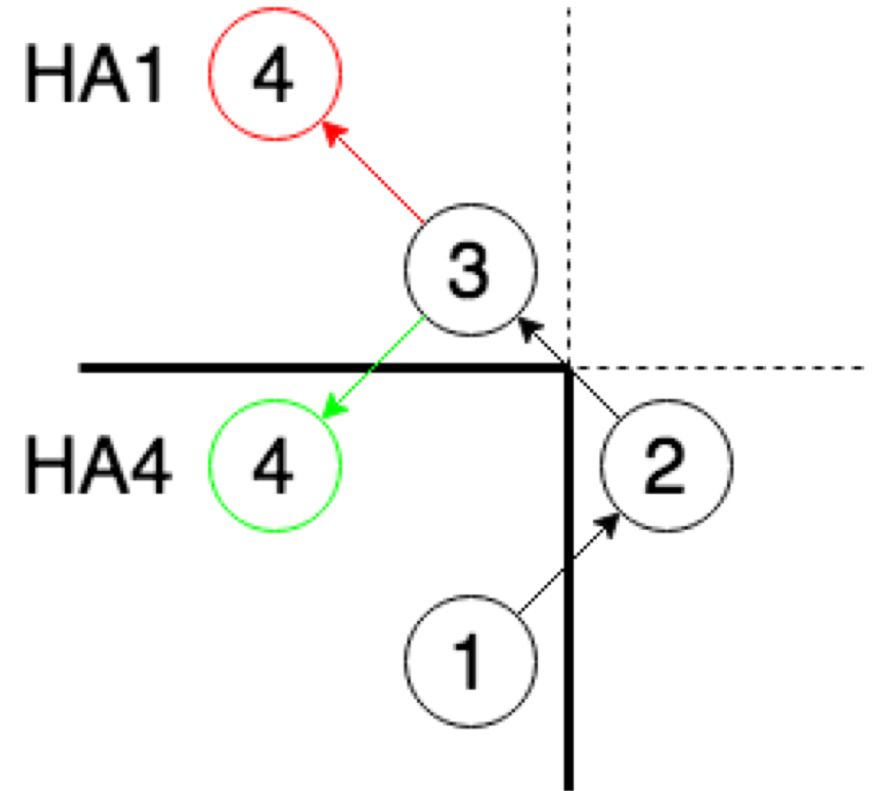
```
16     returnA -< ((px,py,vx,vy),
17       if(ev1 /= NoEvent || ev2 /= NoEvent)
18         then if(ev3 /= NoEvent || ev4 /= NoEvent)
19           then Event() 'tag' (px,py,-vx,-vy)
20           else Event() 'tag' (px,py,-vx,vy)
21         else if(ev3 /= NoEvent || ev4 /= NoEvent)
22           then Event() 'tag' (px,py,vx,-vy)
23           else NoEvent 'tag' (px,py,vx,vy))
```

Update signal
functions

```
ssfunc (px',py',vx',vy') = ha4 px' py' vx' vy'
```


Error by Clock Sampling

	1 → 2	2 → 3
HA1	Frame:in→out Can detect event	Frame:out→out Can't detect event
HA4	Left frame out Can detect event	Top frame out Can detect event



Checking Discretized behavior

- Automatic conversion from (restricted) Yampa programs to the model expressing these discrete behavior in the uppaal model
- Model checking the Yampa programs by Uppaal

Results of Uppaal

- HA1

$A \models (\text{Output1} \geq -100 \text{ and } \text{Output1} \leq 100 \text{ and } \text{Output2} \geq -100 \text{ and } \text{Output2} \leq 100)$
Verification/kernel/elapsed time used: 0s / 0s / 0.008s.
Resident/virtual memory usage peaks: 7,324KB / 26,656KB.
属性は満たされませんでした

- HA4

$A \models (\text{Output1} \geq -100 \text{ and } \text{Output1} \leq 100 \text{ and } \text{Output2} \geq -100 \text{ and } \text{Output2} \leq 100)$
Verification/kernel/elapsed time used: 0.015s / 0.016s / 0.024s.
Resident/virtual memory usage peaks: 7,644KB / 26,980KB.
属性は満たされました

Implementability

- Timed automata
 - Always implementable [Bouyer et.al 11]
 - Existence of sampling for untimed language equivalence [Abudula et.al 10]
- Analysis using pushdown system
 - DTPDA[Abudula 11], NeTA[Li,Ogawa, Y 13,15]
- Discretized Communication?