The Sparse Synchronous Model

Stephen A. Edwards and John Hui

FDL, September 15, 2020
sma = NewStateMatrix();

sma = AddState(sma, 'Name', 'ITI', ... 'Timer', S.ITI, ... 'StateChangeConditions', {'Tup', 'PreState'}, ... 'OutputActions', {});

% Pre-task states
sma = AddState(sma, 'Name', 'PreState', ... 'Timer', S.GUI.PreCue, ... 'StateChangeConditions', {'Tup', 'CueDelivery'}, ... 'OutputActions', {'BNCState', 1});

%Cue
sma = AddState(sma, 'Name', 'CueDelivery', ... 'Timer', S.GUI.CueDuration, ... 'StateChangeConditions', {'Tup', 'Delay'}, ... 'OutputActions', {'SoftCode', S.Cue});

% Delay
sma = AddState(sma, 'Name', 'Delay', ... 'Timer', S.Delay, ... 'StateChangeConditions', {'Tup', 'ExtraCueDelivery'}, ... 'OutputActions', {});

% Extra Cue for L3-SecondaryCue
sma = AddState(sma, 'Name', 'ExtraCueDelivery', ... 'Timer', S.ExtraCueDuration, ... 'StateChangeConditions', {'Tup', 'ExtraDelay'}, ... 'OutputActions', {'SoftCode', S.ExtraCue});

% Extra Delay for L3-SecondaryCue
sma = AddState(sma, 'Name', 'ExtraDelay', ... 'Timer', S.ExtraDelay, ... 'StateChangeConditions', {'Tup', 'Outcome'}, ... 'OutputActions', {});

% Reward
sma = AddState(sma, 'Name', 'Outcome', ... 'Timer', S.Outcome, ... 'StateChangeConditions', {'Tup', 'PostOutcome'}, ... 'OutputActions', {'ValveState', S.Valve});

SendStateMatrix(sma);

InitTrial
timer=inf
led center port on
enter center port

DelayStimulus
timer=stim_delay
led center port off
leave center port
timer elapsed

DeliverStimulus
timer=0.35s
deliver stimulus
leave center port
timer elapsed

TimeoutPunishment
timer=3s
stop stimulus
timer elapsed
play punishment sound

WaitForChoice
timer=3s
stop stimulus
stop stimulus

DelayReward
timer=correct_delay
led left port off
led right port off

DeliverReward
timer=reward_time
deliver reward

ITI
timer=0.5s

DelayPunishment
timer=error_delay
led left port off
led right port off

grace-period=0.4s

DelayReward
timer=correct_delay
led left port off
led right port off

DeliverReward
timer=reward_time
deliver reward

ITI
timer=0.5s

DelayPunishment
timer=error_delay
led left port off
led right port off

grace-period=0.4s

DelayReward
timer=correct_delay
led left port off
led right port off

DeliverReward
timer=reward_time
deliver reward

ITI
timer=0.5s

DelayPunishment
timer=error_delay
led left port off
led right port off

grace-period=0.4s
while \( a \neq b \)

if \( a < b \) then
\( b = b - a \)
else
\( a = a - b \)

\( r = a \)

Named routines, no return values
gcd(a, b, \
° Named routines, no return values \
° Pass-by-value (integer) arguments
gcd(a, b, &r)

Named routines, no return values
Pass-by-value (integer) arguments
Pass-by-reference arguments

while a != b
if a < b then
    b = b - a
else
    a = a - b
r = a
gcd(a, b, &r)

while a != b

Named routines, no return values
Pass-by-value (integer) arguments
Pass-by-reference arguments
Imperative while loops
gcd(a, b, &r)

while a != b
    if a < b then
        b = b − a
    else
        a = a − b
    r = a

Named routines, no return values
Pass-by-value (integer) arguments
Pass-by-reference arguments
Imperative while loops
Conditionals
gcd(a, b, &r) while a != b
   if a < b then
      b = b − a
r = a

Named routines, no return values
Pass-by-value (integer) arguments
Pass-by-reference arguments
Imperative while loops
Conditionals
Imperative assignment
gcd(a, b, &r)
  while a != b
    if a < b then
      b = b - a
    else
      a = a - b
  r = a

Named routines, no return values
Pass-by-value (integer) arguments
Pass-by-reference arguments
Imperative while loops
Conditionals
Imperative assignment
Assignment to a reference returns a value
fib(n, &r)
    var r1 = 0

Local variables
fib(n, &r)

var r1 = 0

if n < 2 then
    r = 1
else
    fork fib(n - 1, r1)
    fork fib(n - 2, r)
    r = r1 + r

Local variables

Recursive routine calls
main()
  var a = 1
  fork foo(a) bar(a)

Concurrent recursive calls

Concurrently running routines may interfere
Deterministic: execution order prescribed by call order
No true parallelism, for now
foo(&a)
  a = a + 2

bar(&a)
  a = a \ast 4

main()
  var a = 1
  fork foo(a) bar(a)

Concurrent recursive calls
Concurrently running routines may interfere

Deterministic: execution order prescribed by call order
No true parallelism, for now
foo(&a)
a = a + 2

bar(&a)
a = a * 4

main()
var a = 1
fork foo(a) bar(a)

Concurrent recursive calls

Concurrently running routines may interfere

Deterministic: execution order prescribed by call order
foo(&a)
  a = a + 2

bar(&a)
  a = a * 4

main()
var a = 1
fork foo(a) bar(a)
  // a = 12 = (1 + 2) * 4 here

Concurrent recursive calls
Concurrently running routines may interfere
Deterministic: execution order prescribed by call order
foo(&a)
a = a + 2

bar(&a)
a = a \times 4

main()
var a = 1
fork foo(a) bar(a)
   // a = 12 = (1 + 2) \times 4 here
fork bar(a) foo(a)

Concurrent recursive calls
Concurrently running routines may interfere
Deterministic: execution order prescribed by call order
foo(&a)
  
  a = a + 2

bar(&a)
  
  a = a * 4

main()
  var a = 1
  fork foo(a) bar(a)
  // a = 12 = (1 + 2) * 4 here
  fork bar(a) foo(a)
  // a = 50 = (12 * 4) + 2 here

Concurrent recursive calls

Concurrently running routines may interfere

Deterministic: execution order prescribed by call order

Concurrently running routines may interfere
foo(&a)
   a = a + 2

bar(&a)
   a = a \times 4

main()
   var a = 1
   fork foo(a) bar(a)
     // a = 12 = (1 + 2) \times 4 here
   fork bar(a) foo(a)
     // a = 50 = (12 \times 4) + 2 here

Concurrent recursive calls
Concurrently running routines may interfere
Deterministic: execution order prescribed by call order
No true parallelism, for now
blink(&led)
while 1
  after 50 ms led = 1

Delayed assignment:
future update scheduled
blink(&led)
while 1
  after 50 ms led = 1
  wait led

Delayed assignment:
future update scheduled
Blocking wait-for-write
blink(&led)
while 1
  after 50 ms led = 1
  wait led
  after 50 ms led = 0
  wait led

Delayed assignment: future update scheduled
Blocking wait-for-write
blink(&led)
while 1

  after 50 ms led = 1
  wait led
  after 50 ms led = 0
  wait led
blink(&led)
while 1
   fib(19, r)
   after 50 ms led = 1
   wait led
   after 50 ms led = 0
   wait led
blink(&led)
while 1
  fib(23, r)
  after 50 ms led = 1
  wait led
  after 50 ms led = 0
  wait led
blink(&led)
while 1
after 50 ms led = 1
wait led
after 50 ms led = 0
wait led
led
0
while 1
after 50 ms led = 1
wait led
after 50 ms led = 0
wait led
blink(led)
while 1
after 50 ms led = 1
wait led
after 50 ms led = 0
wait led
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led

led 0
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
`blink(&led)`

```plaintext
while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
```

`led 0`
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led

led 0
blink(&led)

while 1

  after 50 ms led = 1
  wait led
  after 50 ms led = 0
  wait led

led 0
blink(&led)

while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
blink(&led)
while 1
  after 50 ms led = 1
  wait led
  after 50 ms led = 0
  wait led
blink(&led)

while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
blink(&led)
    while 1
        after 50 ms led = 1
        wait led
        after 50 ms led = 0
        wait led
blink(&led)

while 1

  after 50 ms led = 1

wait led

after 50 ms led = 0

wait led

0ms 50ms 100ms 150ms

led ← 1

led
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led

0ms 50ms 100ms 150ms

led ← 1
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
blink(&led)

while 1
  after 50 ms led = 1
  wait led
  after 50 ms led = 0
  wait led
```c
blink(&led)

while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
```
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
blink(&led)

while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led

led ← 0
blink(&led)
    while 1
        after 50 ms led = 1
        wait led
        after 50 ms led = 0
        wait led
blink(&led)
while 1
  after 50 ms led = 1
  wait led
  after 50 ms led = 0
  wait led
blink(&led)

while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led

led ← 0
blink(&led)

while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
blink(&led)

while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led

led ← 1
blink(&led)

while 1

after 50 ms led = 1
wait led

after 50 ms led = 0
wait led
blink(&led)
  while 1
    after 50 ms led = 1
    wait led
    after 50 ms led = 0
    wait led
toggle(&led)
led = 1 - led
blink(&led, period)

var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e
main(&led)
fork
    blink(led, 50ms)
    blink(led, 30ms)
    blink(led, 20ms)
    led=0
main()
main(&led)

toggle(&led)
led = 1 - led

blink(&led, period)

var

e = 0

while

1

toggle(led)

after

period e = 0

wait

e

fork

blink(led, 50ms)
blink(led, 30ms)
blink(led, 20ms)

led=0

main()
main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main(&led)
   fork blink(led, 50ms) blink(led, ) blink(led, )
main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
blink(&led, period)

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
blink(&led, period)
  var e = 0

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```plaintext
blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e
main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

blink(&led, period)
  var e = 0
  while 1
    toggle(led)

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main()

PC

now = 0 ms

toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main()

PC

Caller

toggle(&led)
    led = 1 - led

blink(&led, period)
      var e = 0
      while 1
          toggle(led)
          after period e = 0
          wait e

main(&led)
      fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main()

PC
• Caller

now = 0 ms

toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main()

PC
• Caller
led

now = 0 ms

toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main()

PC
• Caller
0
led

now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
after period e = 0
wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
toggle(&led)
    led = 1 - led

blinking(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 0 ms

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
Now = 0 ms

```
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

PC
  Caller
  0
  led

led=0

PC

Ready:  
  

PC

PC

PC

PC
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
toggle(led)
after period e = 0
wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```plaintext
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
null = 0 ms

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
toggle(led)
after period e = 0
wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

now = 0 ms

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

   led = 1

   led
   period=50
   e1=0
   after period e = 0

   blink()

   led
   period=30
   e2=0
   while 1
      toggle(led)
   after period e = 0
   wait e

   blink()

   led
   period=20
   e3=0

   main()

   led
   period=50
   e1=0
   var e = 0
   while 1
      toggle(&led)
      led = 1 - led
      blink(&led, period)
      after period e = 0
      wait e

   toggle(&led)
   led = 1 - led

now = 0 ms
```
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Events: @50 e₁ ← 0

PC
     • Caller
     0

led
     • trigger
     e₁ = 0
     @50 e₁ ← 0

PC
     • Caller
     000

led
     • trigger
     e₂ = 0

PC
     • Caller
     010

led
     • trigger
     e₃ = 0
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
toggle(led)
after period e = 0
wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

PC
  • Caller
  0
  led

PC
  • Caller
  000
  led
  period=50
  • e1=0 @50 e1<->0
  trigger

PC
  • Caller
  001
  led
  period=30
  • e2=0
  trigger

PC
  • Caller
  010
  led
  period=20
  • e3=0
  trigger

Events: [0x0] @50 e1<->0

Ready: [0x0] 001 010
now = 0 ms

```c
toggle(&led)
    led = 1 - led

blink(&led, period)

var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
after period e = 0
wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

main()
  led = 0

blink()
  led
  period=50
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main()
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
toggle(led)
after period e = 0
wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```plaintext
now = 0 ms

toggle(&led)
   led = 1 - led

blink(&led, period)
   var e = 0
   while 1
      toggle(led)
      after period e = 0
      wait e

main(&led)
   fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

Events:
- @30 e2 ← 0
- @50 e1 ← 0

Ready: 010 000 001

PC
- Caller 0
- led

blink()
- PC
- Caller 000
- led period=50
- e1 = 0 @50 e1 ← 0
- trigger

blink()
- PC
- Caller 001
- led period=30
- e2 = 0 @30 e2 ← 0
- trigger

blink()
- PC
- Caller 010
- led period=20
- e3 = 0
- trigger
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

```plaintext
toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

Events:
- @30 e2←0
- @50 e1←0

Ready: [ ] [ ] [ ]

trigger
```
now = 0 ms

toggle(&led)

led = 1 - led

blink(&led, period)

var e = 0

while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 0 ms

toggle(&led)  
led = 1 - led

blink(&led, period)
var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 0 ms

toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e
    main(&led)
        fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0

while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

PC
  • Caller
  0
  led

blink()
  PC
  • Caller
  000
  led
  period=50
  • e1=0 @50 e1←0
  trigger

blink()
  PC
  • Caller
  001
  led
  period=30
  • e2=0 @30 e2←0
  trigger

blink()
  PC
  • Caller
  010
  led
  period=20
  • e3=0 @20 e3←0
  trigger
now = 0 ms

```
toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e
main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 0 ms

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 0 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 20 ms

```
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```

---

Events:

- @20 \( e_3 \leftarrow 0 \)
- @30 \( e_2 \leftarrow 0 \)
- @50 \( e_1 \leftarrow 0 \)

---

Ready:
now = 20 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Events: @30 e2←0 @50 e1←0

Ready:  🟢 🟢 🟢
main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)

PC
- Caller
0
led

PC
- Caller
0
led
period=50
- e1=0 @50 e1←0
trigger

PC
- Caller
0
led
period=30
- e2=0 @30 e2←0
trigger

PC
- Caller
0
led
period=20
- e3=0
trigger

now = 20 ms

Events:

@30 e2←0
@50 e1←0

Ready:

@50e 1 ← 0
@30e 2 ← 0
@50e 1 ← 0
@20e 3 ← 0
@30e 2 ← 0
@50e 1 ← 0
@40e 3 ← 0
@50e 1 ← 0
@60e 2 ← 0
@50e 1 ← 0
@60e 3 ← 0
@60e 3 ← 0
@100e 1 ← 0
@60e 3 ← 0
@100e 1 ← 0
@90e 2 ← 0
@100e 1 ← 0
@80e 3 ← 0
@90e 2 ← 0
@100e 1 ← 0

led=1
```
now = 20 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e
main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 20 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Events:
@30 e₂ ← 0
@50 e₁ ← 0

PC
caller
led
blink()

PC
caller
led
period=50
blink()

PC
caller
led
period=30
blink()

PC
caller
led
period=20
blink()
now = 20 ms

```
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 20 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0

while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 20 ms

shuffle(&led)
led = 1 - led

blinks(&led, period)
var e = 0
while 1
  shuffle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Events: @30 e2←0 @50 e1←0

PC
- Caller 0
- led

blink()  
PC
- Caller 000
- led
- period=50
- e1=0 @50 e1←0
- trigger

blink()  
PC
- Caller 001
- led
- period=30
- e2=0 @30 e2←0
- trigger

blink()  
PC
- Caller 010
- led
- period=20
- e3=0
- trigger

Ready: □ □ □
```
now = 20 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
toggle(led)
after period e = 0
wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 20 ms

**toggle(&led)**
led = 1 - led

**blink(&led, period)**

```c
var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e
```

**main(&led)**

```c
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 20 ms

main(&led)

fork
  blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Event:
  @30 e₂ ← 0
  @50 e₁ ← 0

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  trigger
now = 20 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
toggle(led)
    after period e = 0
wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e
  end

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 20 ms

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 20 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
toggle(led)
after period e = 0
wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 20 ms

toggle(&led)
   led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e
    main(&led)
        fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 30 ms

```
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 30 ms

```
toggle(&led)
    led = 1 - led
```

```
blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e
    end
```

```
main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
toggle(&led)
   led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Now = 40 ms

Events: @50 e₁←0  @60 e₂←0  @60 e₃←0

Ready: [ ] [ ] [ ]
toggle(\&led)
  led = 1 - led

blink(\&led, period)
  \textbf{var} \ e = 0
  \textbf{while} 1
    toggle(led)
  \textbf{after} period \ e = 0
  \textbf{wait} \ e

main(\&led)
  \textbf{fork} blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 50 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
Now = 60 ms

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Ready:  

Events:  
  @60 e2<0  @60 e3<0  @100 e1<0

trigger
toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 60 ms

```
toggle(&led)
    led = 1 - led

blink(&led, period)
    var e = 0
    while 1
        toggle(led)
        after period e = 0
        wait e

main(&led)
    fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 60 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

now = 60 ms
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e
  end

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 60 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 60 ms

toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
  toggle(led)
after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 60 ms

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 60 ms

```
toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 60 ms

toggle(&led)
led = 1 - led

blink(&led, period)

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Events:
@100 e₁ ← 0

var e = 0
while 1
  toggle(led)
after period e = 0
  wait e

main()
PC
• Caller
  0
  led
  period=50
  • e₁=0 @100 e₁ ← 0
  • trigger

led=1
PC
• Caller
  0

blink()
PC
• Caller
  000
  led
  period=50
  • e₁=0 @100 e₁ ← 0
  • trigger

blink()
PC
• Caller
  001
  led
  period=30
  • e₂=0
  • trigger

blink()
PC
• Caller
  010
  led
  period=20
  • e₃=0
  • trigger
now = 60 ms

```c
toggle(&led)
led = 1 - led

blink(&led, period)

var e = 0
while 1
toggle(led)
after period e = 0
wait e

main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```plaintext
main(&led)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
main(
  led = 0
)

blink()

var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(
  led
)

fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
now = 60 ms

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)

Events:
@90 e2←0 @100 e1←0

toggle(&led)
led = 1 - led

blink(&led, period)
var e = 0
while 1
  toggle(led)
  after period e = 0
  wait e

Ready: [ ] [ ] [ ]

PC
• Caller
0
led

PC
• Caller
000
led
period=50
• e1=0 @100 e1←0
trigger

PC
• Caller
001
led
period=30
• e2=0 @90 e2←0
trigger

PC
• Caller
010
led
period=20
• e3=0
trigger

led=0
now = 60 ms

```
toggle(&led)
    led = 1 - led

blink(&led, period)
var e = 0
while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
```
now = 60 ms

main(&led)
  led = 1 - led

toggle(&led)
  led = 1 - led

blink(&led, period)
  var e = 0
  while 1
    toggle(led)
    after period e = 0
    wait e

main(&led)
  fork blink(led, 50ms) blink(led, 30ms) blink(led, 20ms)
SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]
### SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]

<table>
<thead>
<tr>
<th></th>
<th>SSM</th>
<th>Ptides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between instants</td>
<td>Discrete-event</td>
<td>Discrete-Event</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]

<table>
<thead>
<tr>
<th></th>
<th>SSM</th>
<th>Ptides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between instants</td>
<td>Discrete-event</td>
<td>Discrete-Event</td>
</tr>
<tr>
<td>Within instants</td>
<td>Totally-ordered</td>
<td>Discrete-Event</td>
</tr>
</tbody>
</table>
## SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]

<table>
<thead>
<tr>
<th></th>
<th>SSM</th>
<th>Ptides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between instants</td>
<td>Discrete-event</td>
<td>Discrete-Event</td>
</tr>
<tr>
<td>Within instants</td>
<td>Totally-ordered</td>
<td>Discrete-Event</td>
</tr>
<tr>
<td>Topology</td>
<td>Dynamic, recursive</td>
<td>Static</td>
</tr>
</tbody>
</table>
## SSM vs. Ptides

[Zhao, Liu, and Lee, RTAS 2007]

<table>
<thead>
<tr>
<th></th>
<th>SSM</th>
<th>Ptides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between instants</td>
<td>Discrete-event</td>
<td>Discrete-Event</td>
</tr>
<tr>
<td>Within instants</td>
<td>Totally-ordered</td>
<td>Discrete-Event</td>
</tr>
<tr>
<td>Topology</td>
<td>Dynamic, recursive</td>
<td>Static</td>
</tr>
<tr>
<td>Implementation</td>
<td>Single-threaded</td>
<td>Distributed</td>
</tr>
</tbody>
</table>

[Zou Ph.D 2011]
## SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

<table>
<thead>
<tr>
<th>Feature</th>
<th>SSM</th>
<th>Esterel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Time</td>
<td>Within instants</td>
<td>Totally-ordered</td>
</tr>
<tr>
<td>Sparse/Dense</td>
<td>Sparse</td>
<td>Dense</td>
</tr>
<tr>
<td>Topology</td>
<td>Dynamic, recursive</td>
<td>Static</td>
</tr>
<tr>
<td>Compilation</td>
<td>Separate</td>
<td>Whole-program</td>
</tr>
<tr>
<td>Runtime</td>
<td>Dynamic Event Queues</td>
<td>Statically Scheduled</td>
</tr>
</tbody>
</table>
## SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

<table>
<thead>
<tr>
<th>Feature</th>
<th>SSM</th>
<th>Esterel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

<table>
<thead>
<tr>
<th></th>
<th>SSM</th>
<th>Esterel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Time</td>
<td>Sparse</td>
<td>Dense</td>
</tr>
</tbody>
</table>
### SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

<table>
<thead>
<tr>
<th></th>
<th>SSM</th>
<th>Esterel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Time</td>
<td>Sparse</td>
<td>Dense</td>
</tr>
<tr>
<td>Within instants</td>
<td>Totally-ordered</td>
<td>Constructive</td>
</tr>
</tbody>
</table>
SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

<table>
<thead>
<tr>
<th></th>
<th>SSM</th>
<th>Esterel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Time</td>
<td>Sparse</td>
<td>Dense</td>
</tr>
<tr>
<td>Within instants</td>
<td>Totally-ordered</td>
<td>Constructive</td>
</tr>
<tr>
<td>Compilation</td>
<td>Separate</td>
<td>Whole-program</td>
</tr>
</tbody>
</table>
## SSM vs. Esterel

[Berry and Gonthier, SCP 1992]

<table>
<thead>
<tr>
<th></th>
<th>SSM</th>
<th>Esterel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Time</td>
<td>Sparse</td>
<td>Dense</td>
</tr>
<tr>
<td>Within instants</td>
<td>Totally-ordered</td>
<td>Constructive</td>
</tr>
<tr>
<td>Compilation</td>
<td>Separate</td>
<td>Whole-program</td>
</tr>
<tr>
<td>Runtime</td>
<td>Dynamic Event Queues</td>
<td>Statically Scheduled</td>
</tr>
<tr>
<td></td>
<td>SSM</td>
<td>Esterel</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Time</td>
<td>Sparse</td>
<td>Dense</td>
</tr>
<tr>
<td>Within instants</td>
<td>Totally-ordered</td>
<td>Constructive</td>
</tr>
<tr>
<td>Compilation</td>
<td>Separate</td>
<td>Whole-program</td>
</tr>
<tr>
<td>Runtime</td>
<td>Dynamic Event Queues</td>
<td>Statically Scheduled</td>
</tr>
<tr>
<td>Topology</td>
<td>Dynamic, recursive</td>
<td>Static</td>
</tr>
</tbody>
</table>
https://github.com/sedwards-lab/ssm