Graphic Synchronous Formalisms

SyncCharts

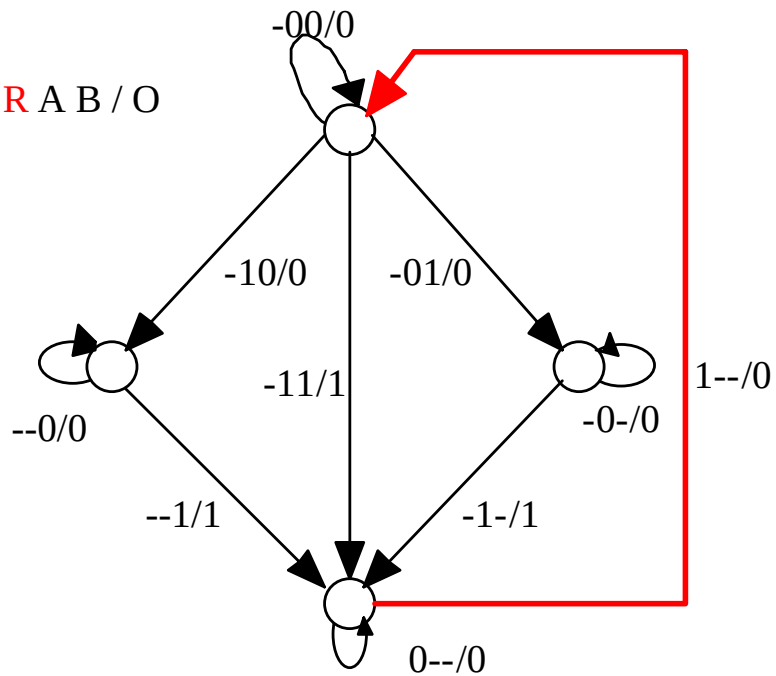
Revisiting ABRO

```
module ABRO2:
  input A,B,R;
  output O;
  loop
     [
     await A
     ||
     await B
     ];
     emit O;
     await R
  end loop
end module
```

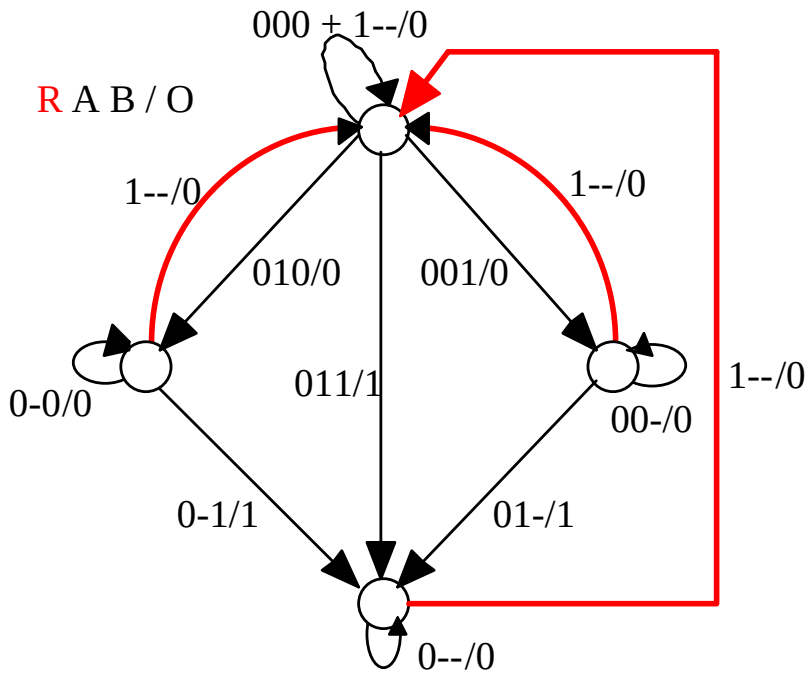
```
module ABRO:
  input A,B,R;
  output O;
  loop
     [
     await A
     ||
     await B
     ];
     emit O
  each R
end module
```

Non-synchronous models

Mealy Machine



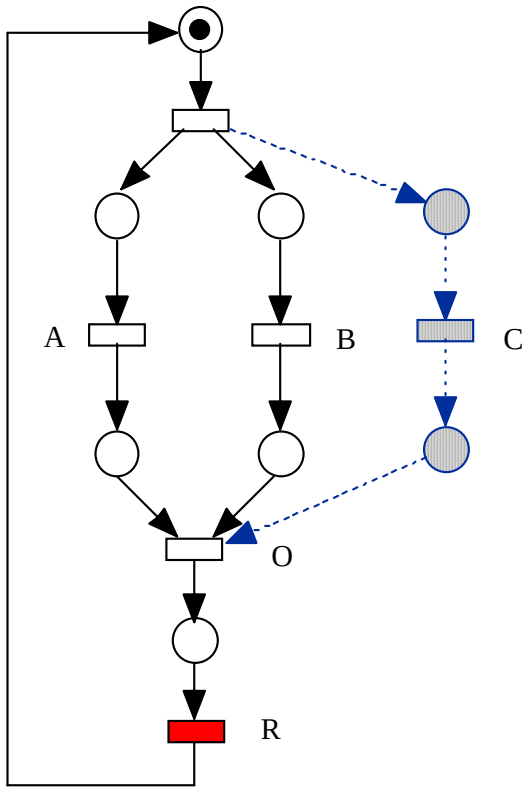
(ABRO2)



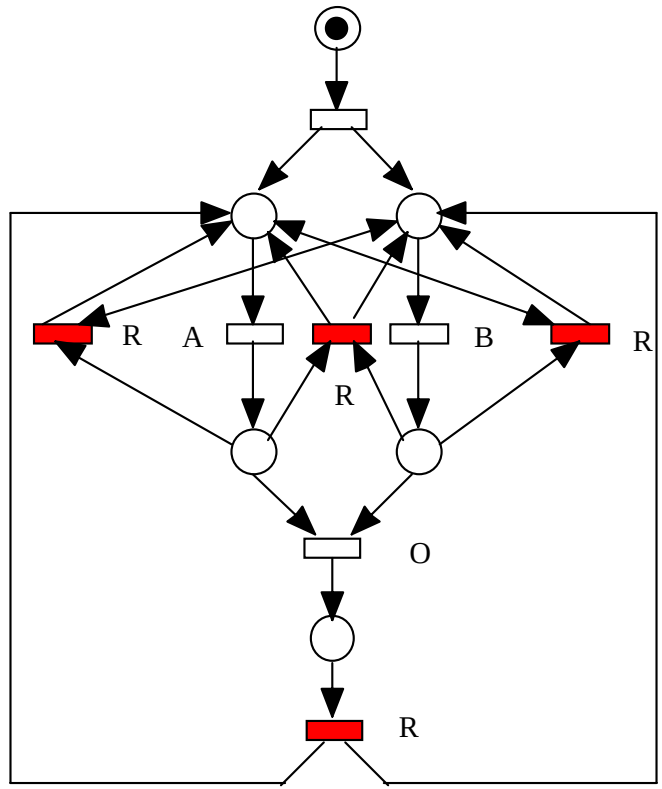
(ABRO)

Non-synchronous models

Petri Nets



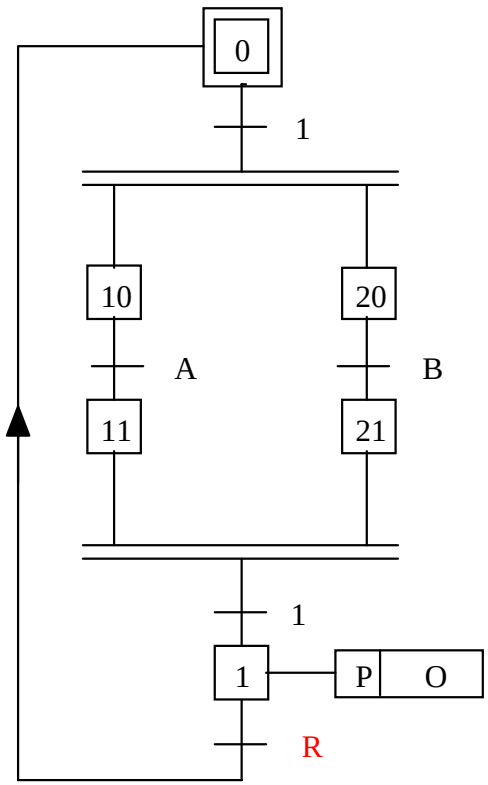
(ABRO2)



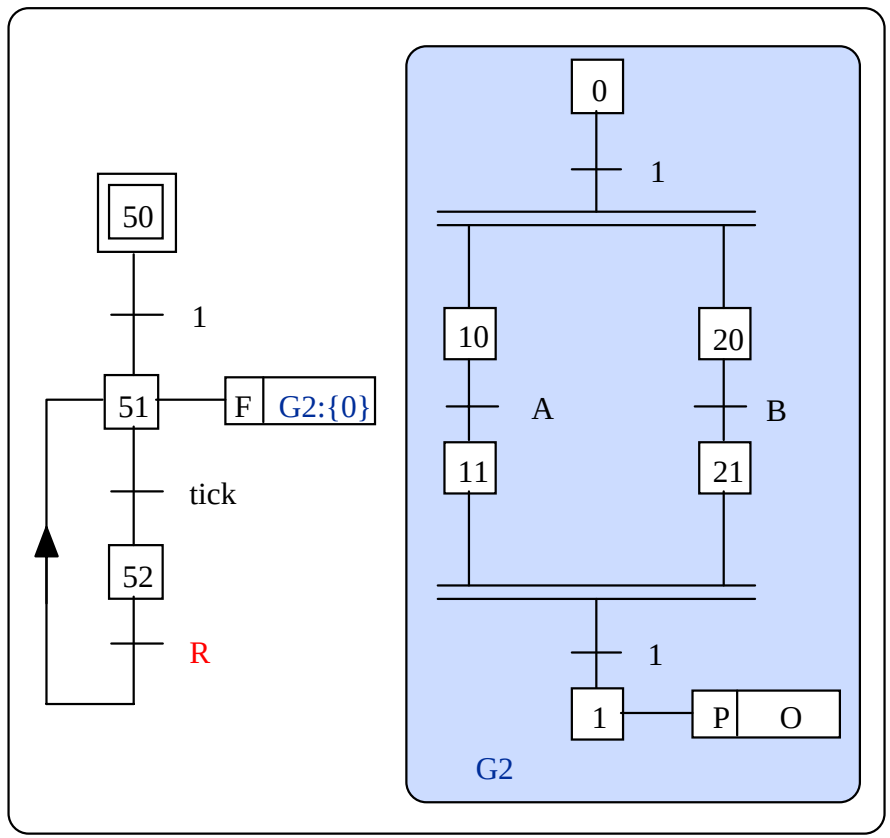
(ABRO)

« quasi » synchronous formalisms

Grafcet (Sequential Function Charts)



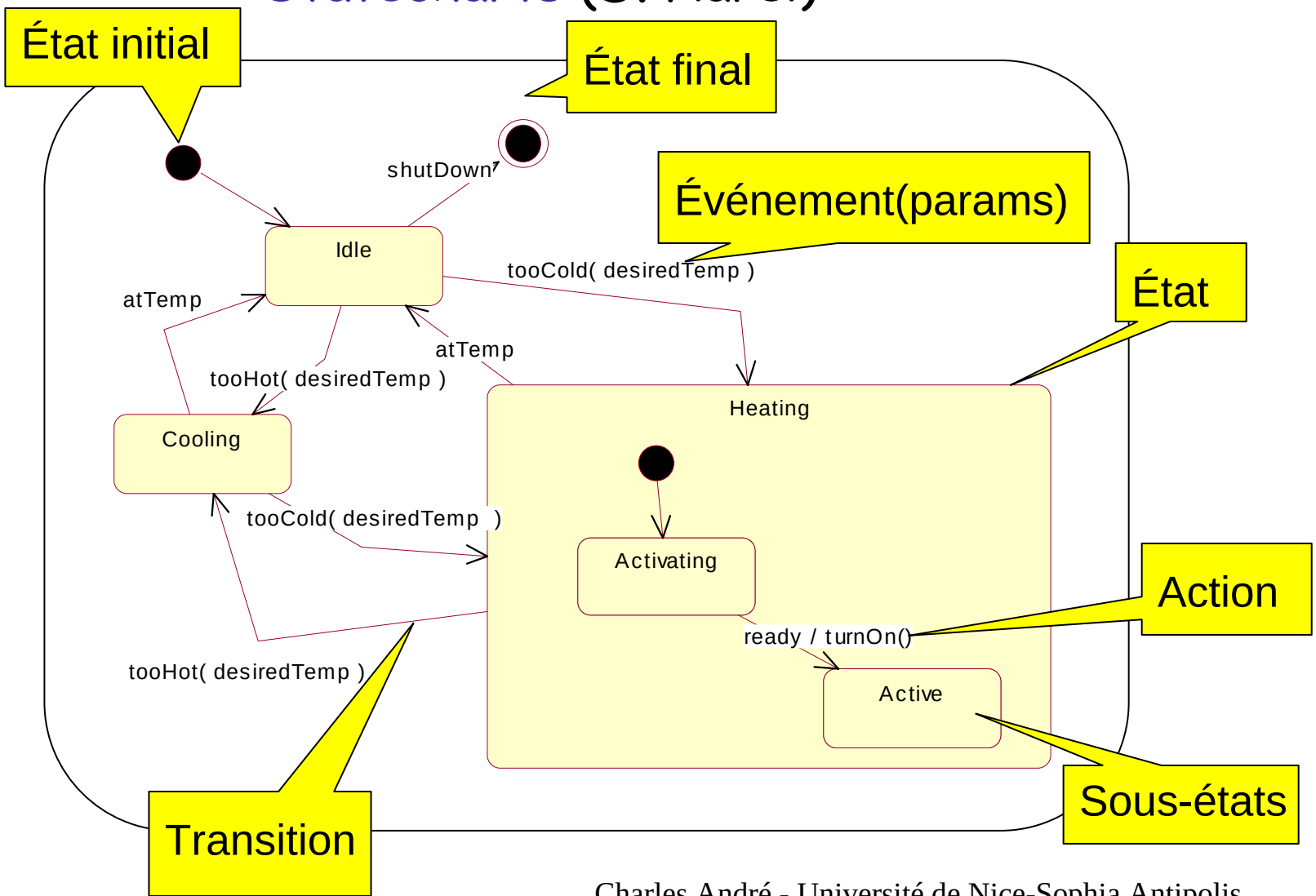
(ABRO2)



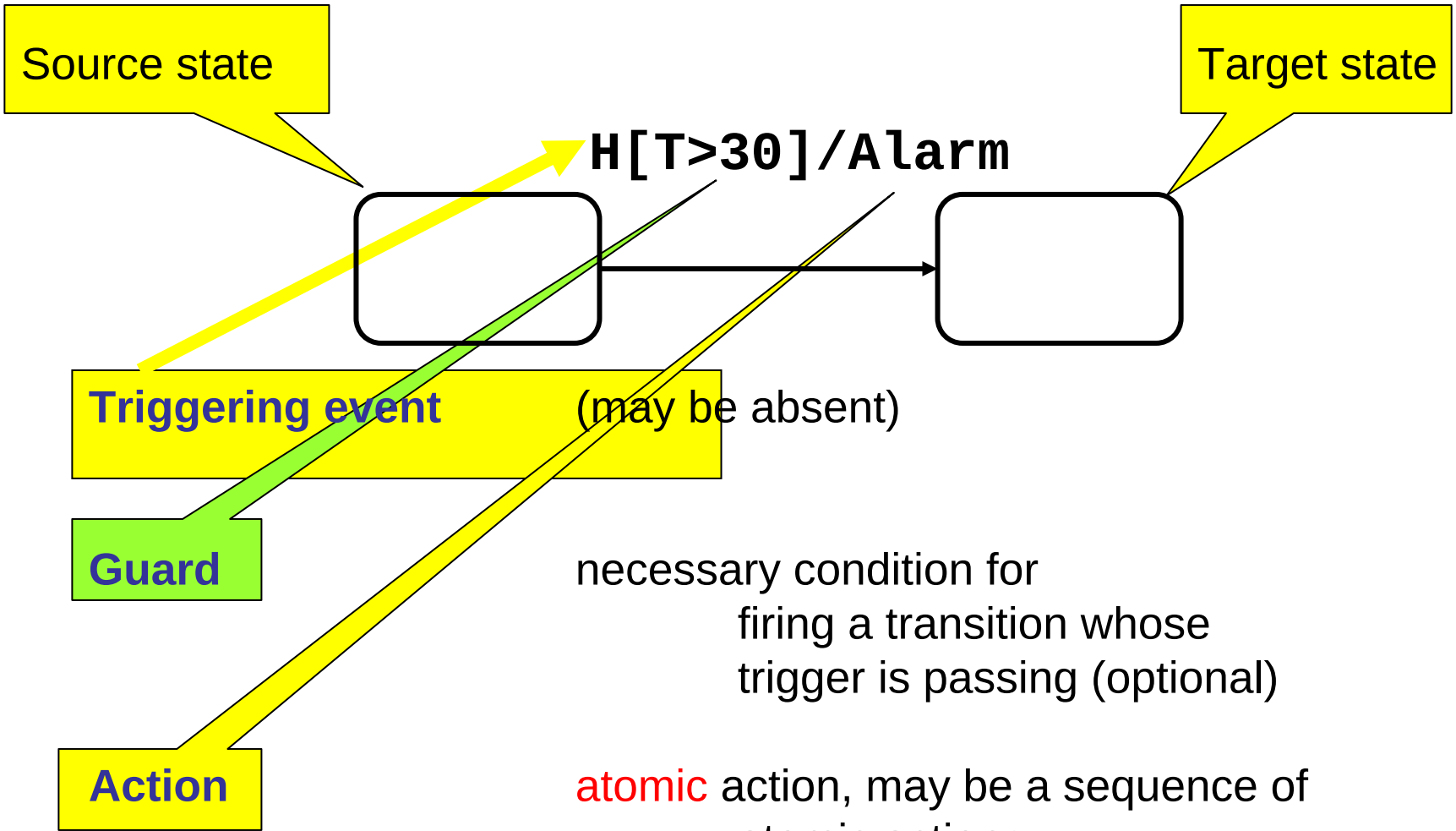
(ABRO)

« quasi » synchronous formalisms

Statecharts (D. Harel)

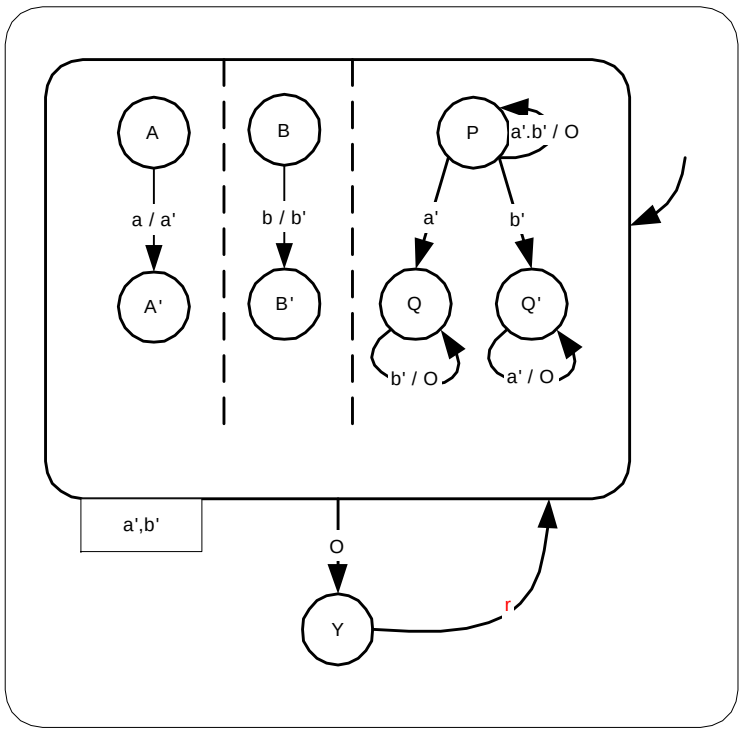


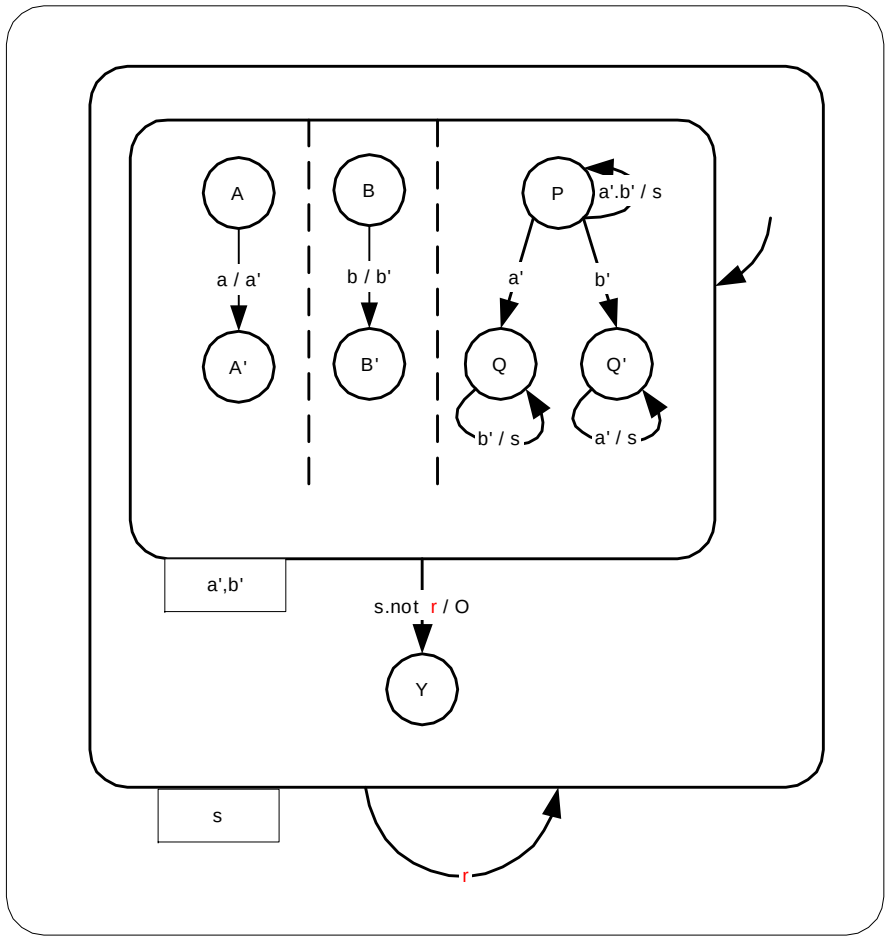
Statecharts (2)



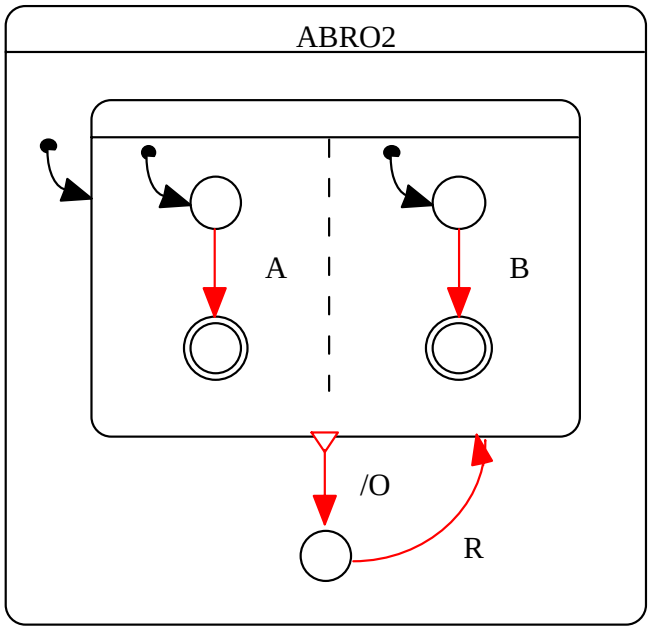
Synchronous formalisms

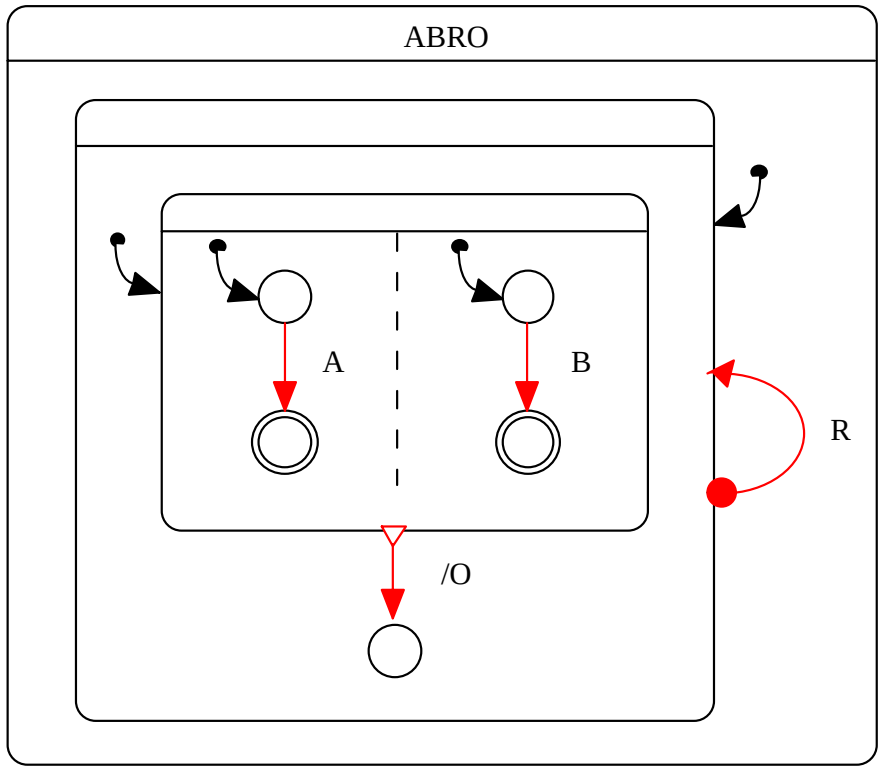
Argos (F. Maraninchi)





SyncCharts





Technical report on the model and its semantics:

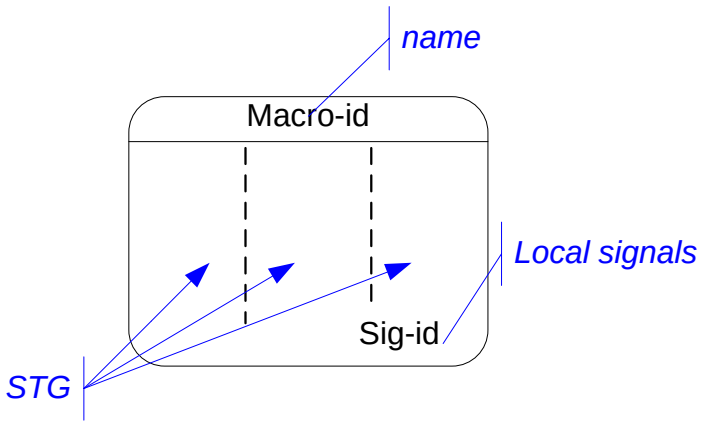
Available at //www.esterel-technologies.com,
Download>Scientific Papers>Semantics of S.S.M

SyncCharts

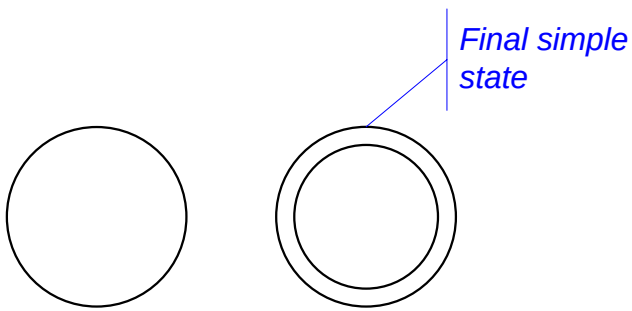
- State-transition graphs (STG)
- Concurrency (STG | STG | …)
- Hierarchy (Macrostate = state that contains STG's)
- Preemption (several ways to leave a state)
- Communication (instantaneous broadcast)

- Fully compatible with Esterel (and as complex as Esterel w.r.t. semantics!)

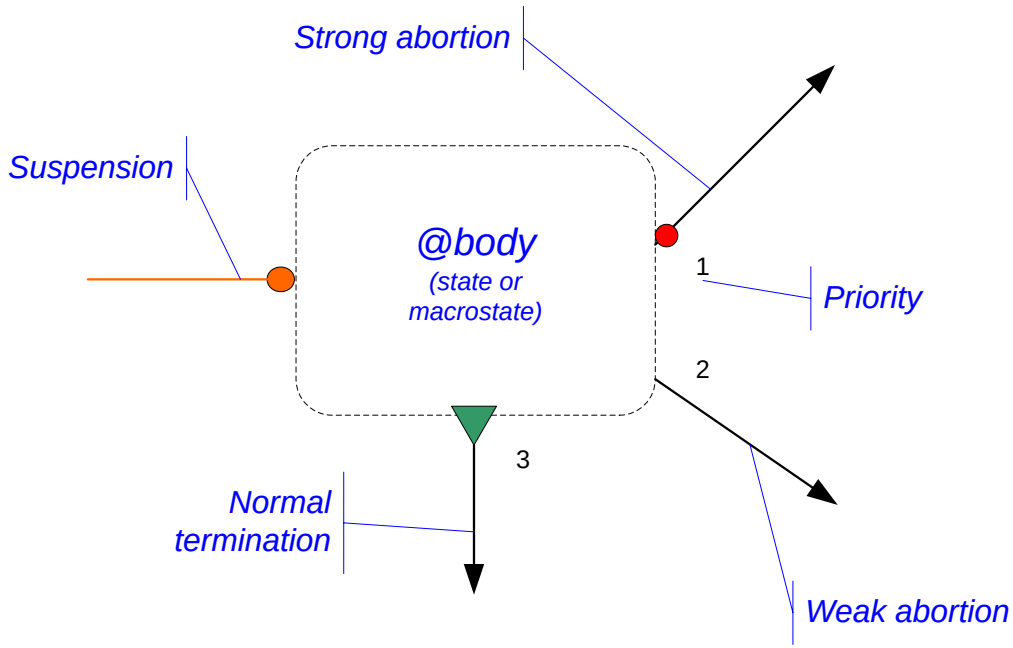
Syntax (1)



(A) Macrostate



(B) Simple states



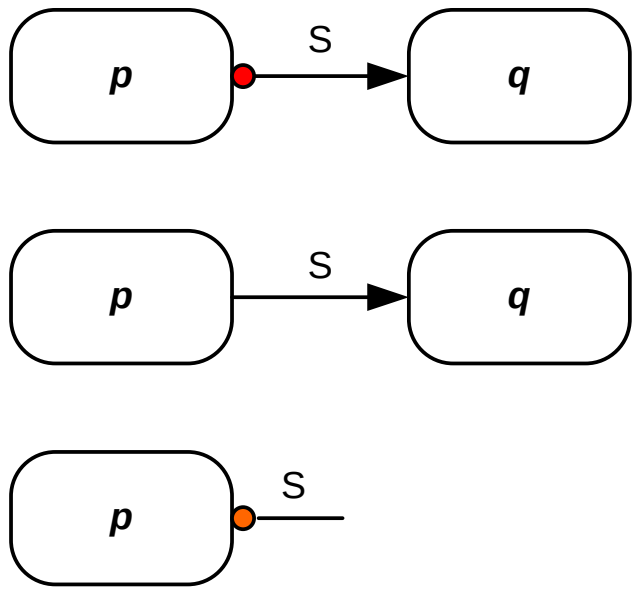
(C) Reactive Cell

Syntax (2)

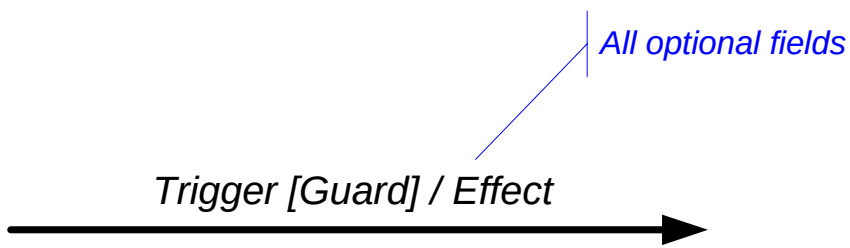
abort
p
when S;
q

weak abort
p
when S;
q

suspend
p
when S

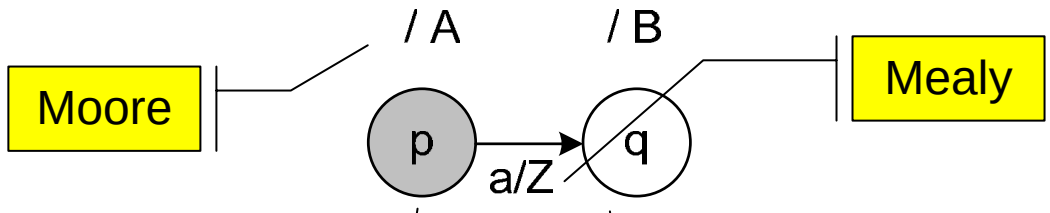


(A) Preemptions



(B) Transition

State-transition model

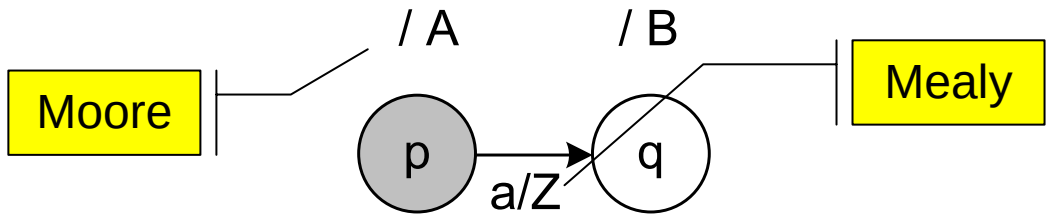



```
weak abort
      sustain A
 ||
      run p
 when a do
      emit Z
 end abort;
```

```
      sustain B
 ||
      run q
```

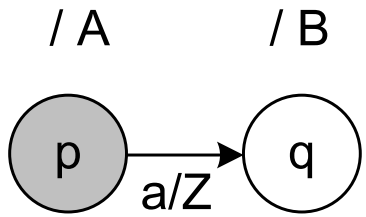
SyncCharts to Esterel in this very very simple case

State-transition model



instant		k-1	k	k+1
input			a	
Seq. Machine		p	p	q
		A	A,Z	B

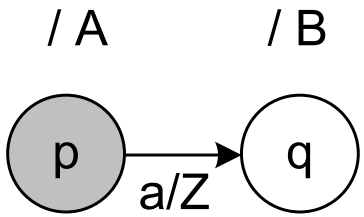
State-transition model



instant		k-1	k	k+1
input			a	
Seq. Machine		p	p	q
		A	A,Z	B
Synch. Model		p	q	q

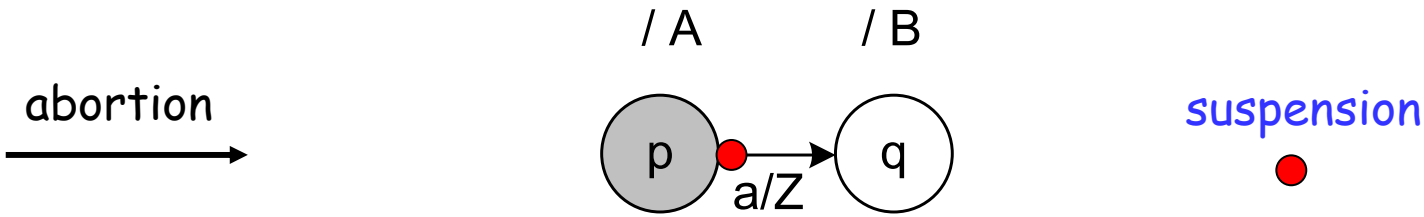
State-transition model

abortion



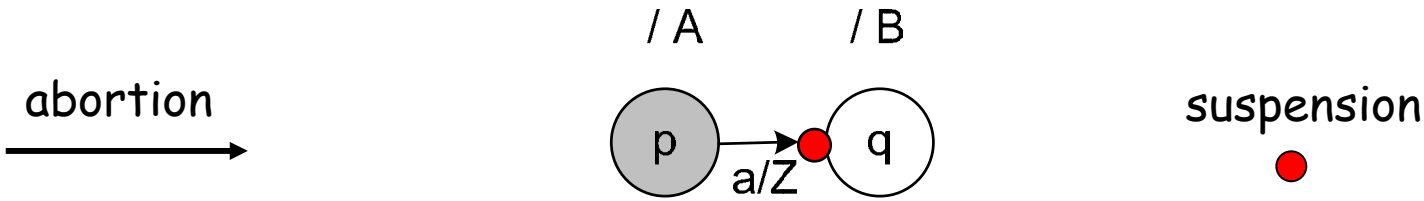
instant		k-1	k	k+1
input			a	
Seq. Machine		p	p	q
		A	A,Z	B
Synch. Model		p	q	q
Weak abortion		A	A,Z,B	B

State-transition model



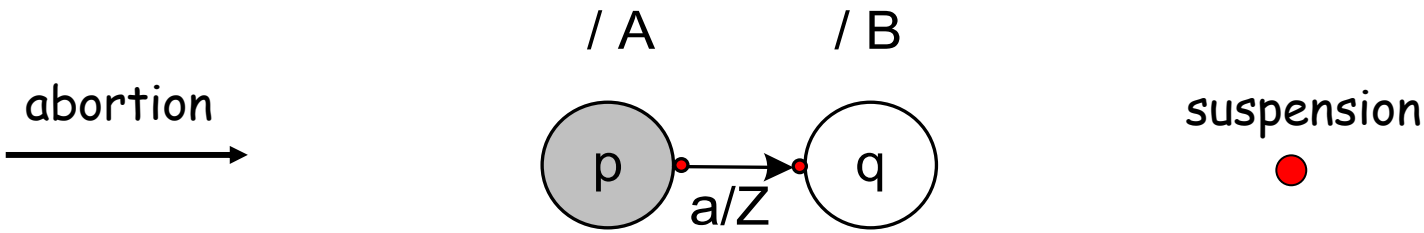
instant		k-1	k	k+1
input			a	
Seq. Machine		p	p	q
		A	A,Z	B
Synch. Model		p	q	q
Weak abortion		A	A,Z,B	B
Strong abortion		A	Z,B	B

State-transition model



instant		k-1	k	k+1
input			a	
Seq. Machine		p	p	q
		A	A,Z	B
Synch. Model		p	q	q
Weak abortion		A	A,Z,B	B
Strong abortion		A	Z,B	B
→ ●		A	A,Z	B

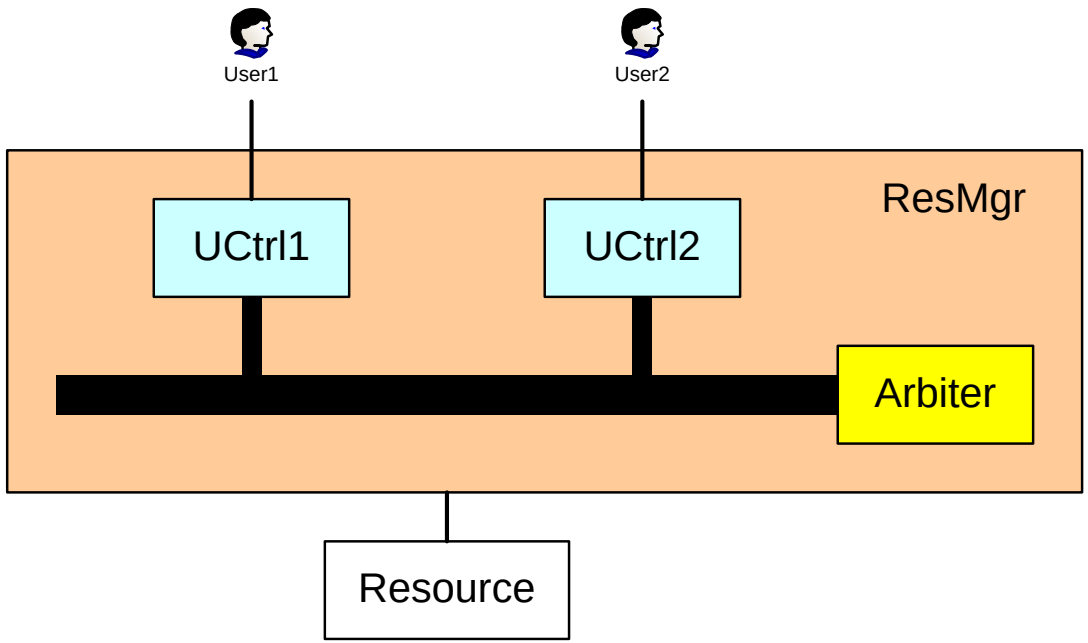
State-transition model



instant		k-1	k	k+1
input			a	
Seq. Machine		p	p	q
		A	A,Z	B
Synch. Model		p	q	q
Weak abortion		A	A,Z,B	B
Strong abortion		A	Z,B	B
→		A	A,Z	B
●→		A	Z	B

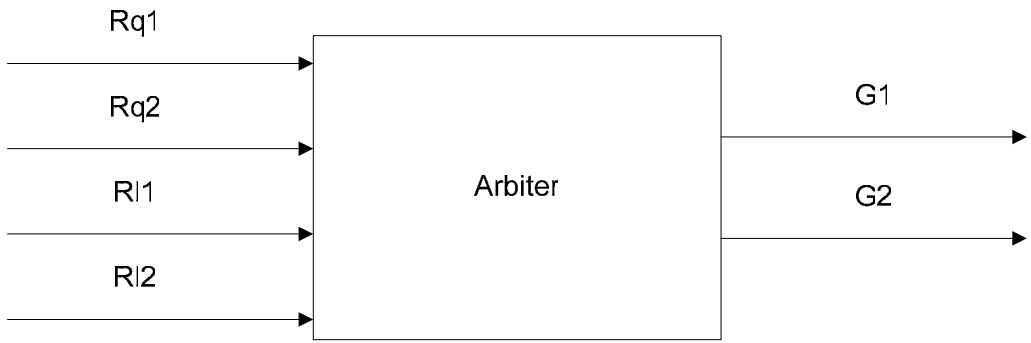
Example : A Simple System

- The system consists of
  - A shared resource
  - Two users that compete to access the resource
  - An access controller (ResMgr)

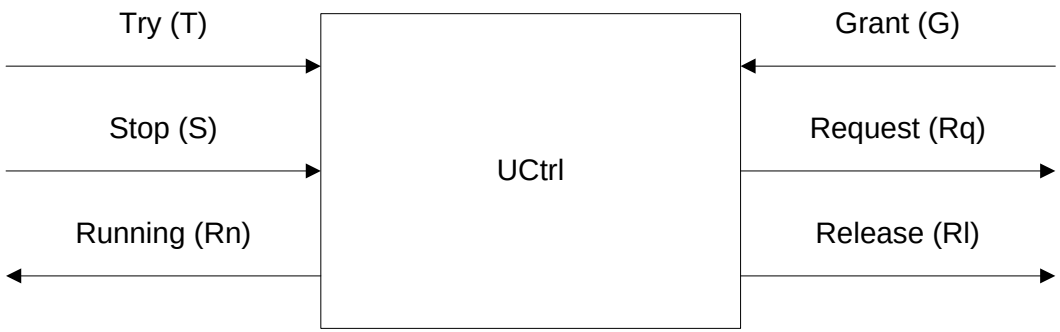


Black-Box View

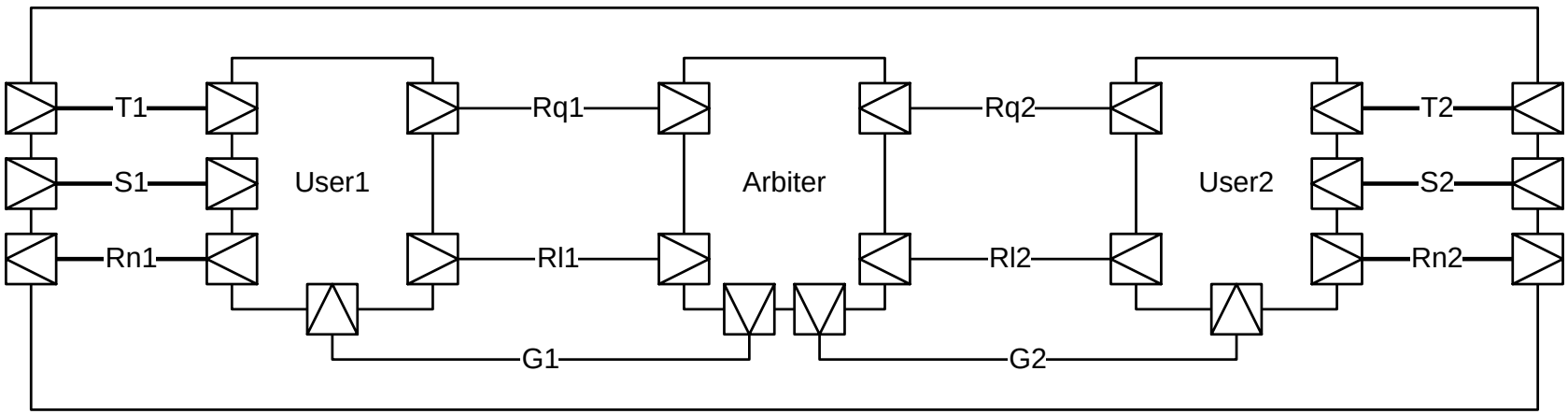
Arbiter



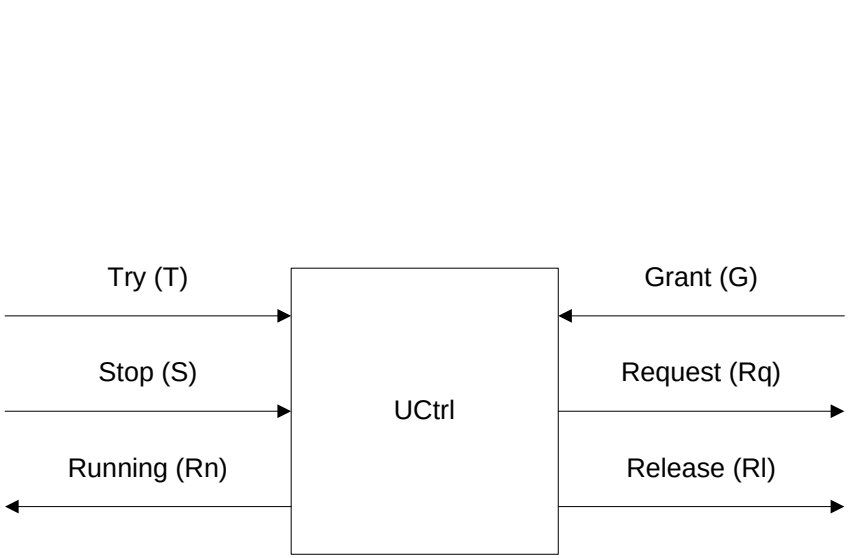
User Interface Controller



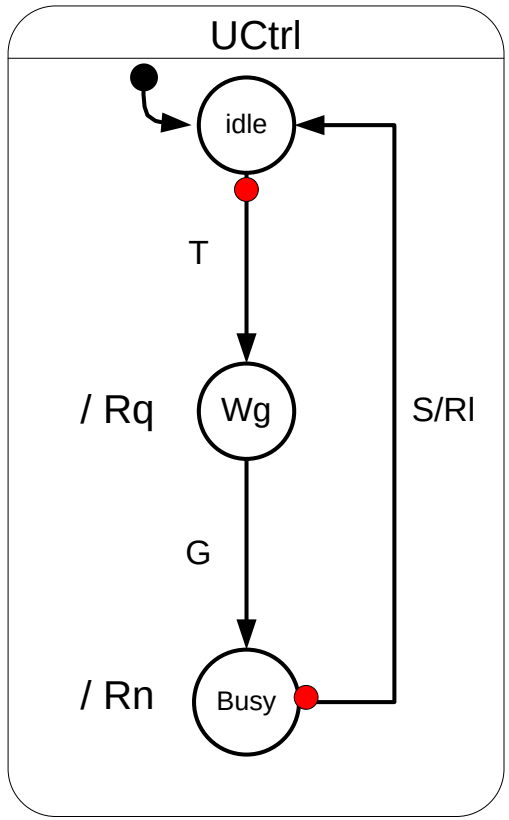
Structure



User's Dialog Controller

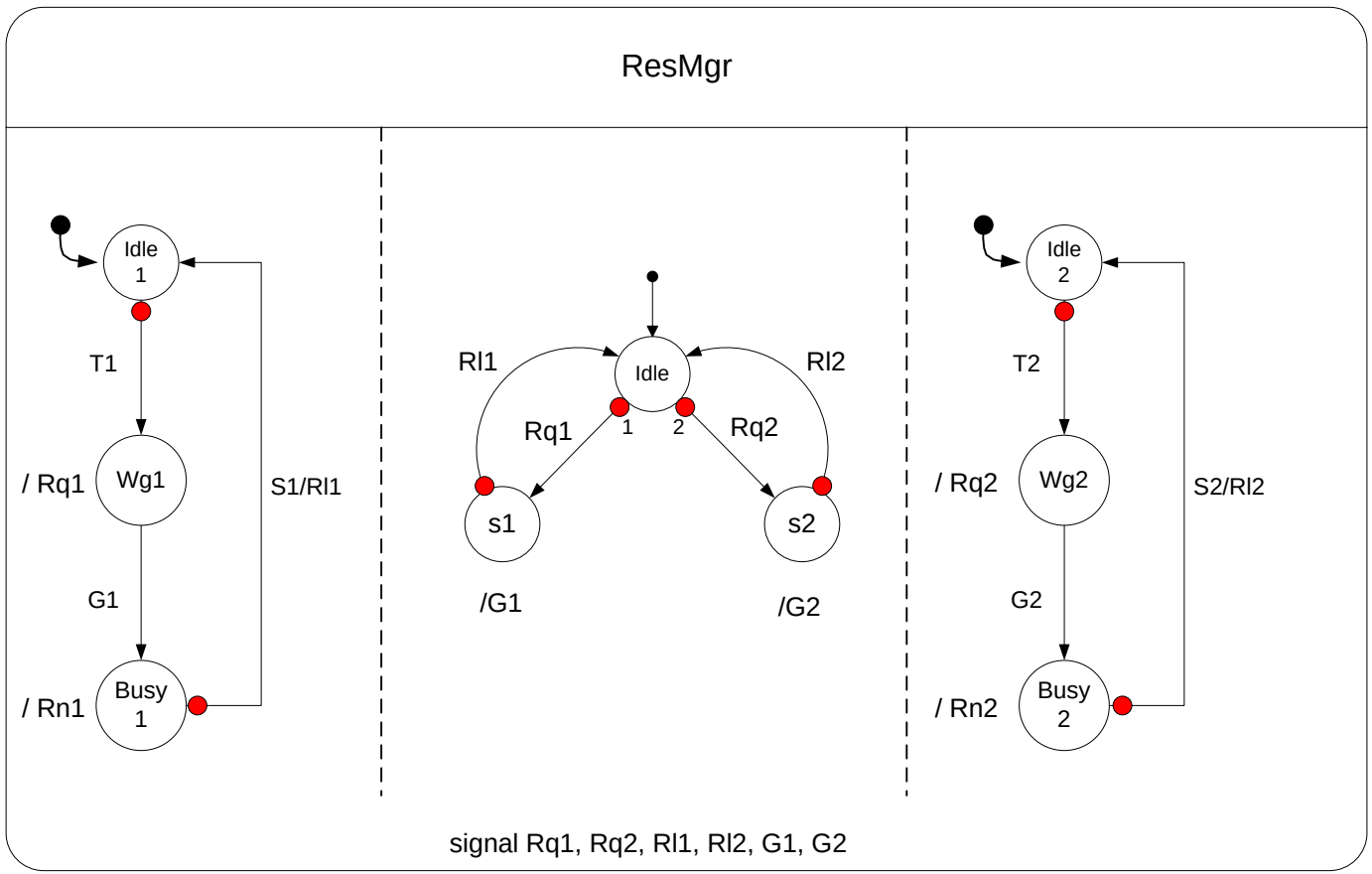


Interface

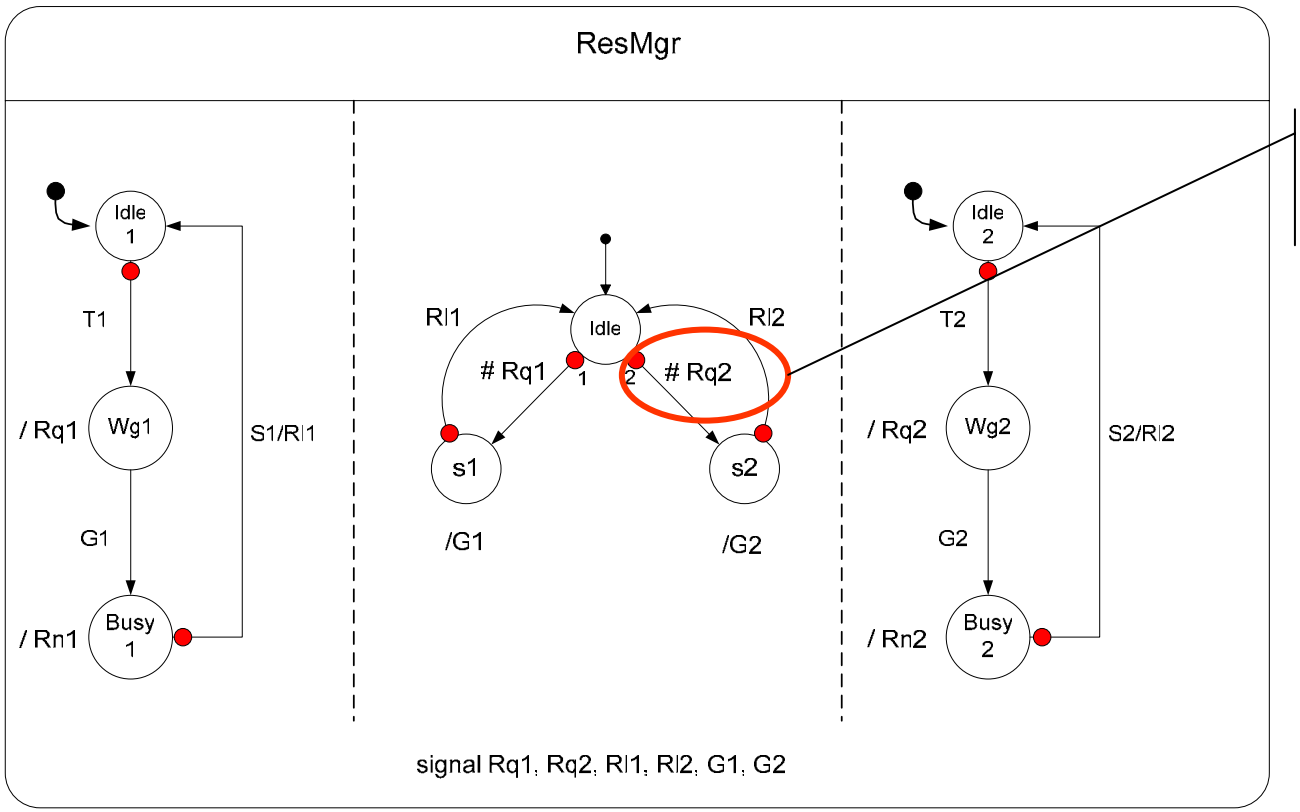


syncChart

Resource Manager Controller

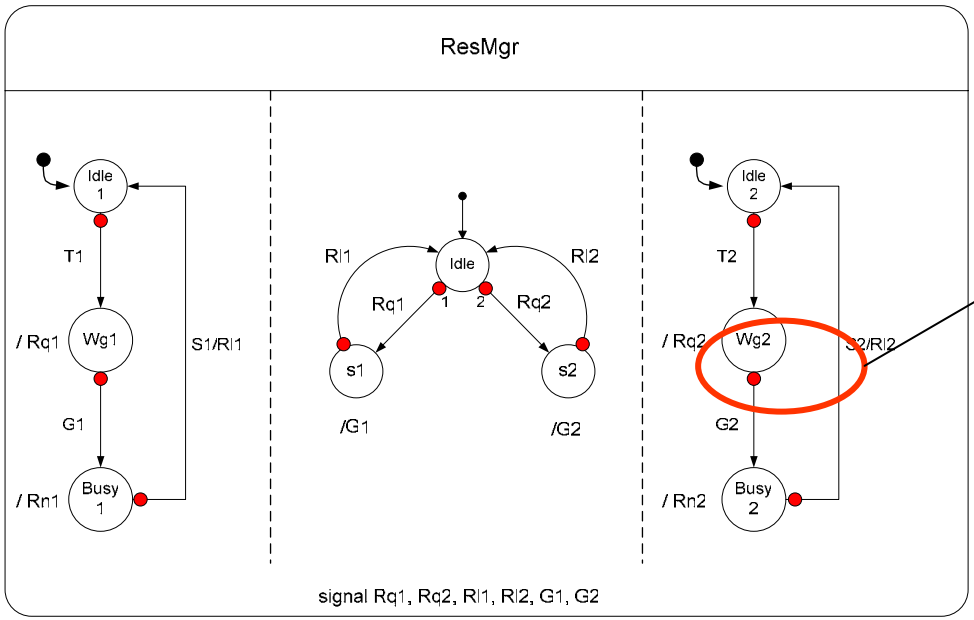


Improved ResMgr



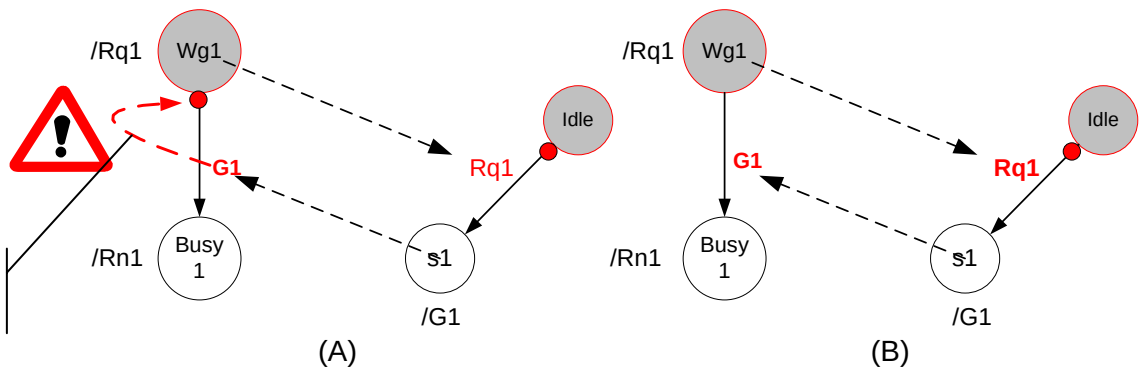
Immediate abortion

Incorrect Solution



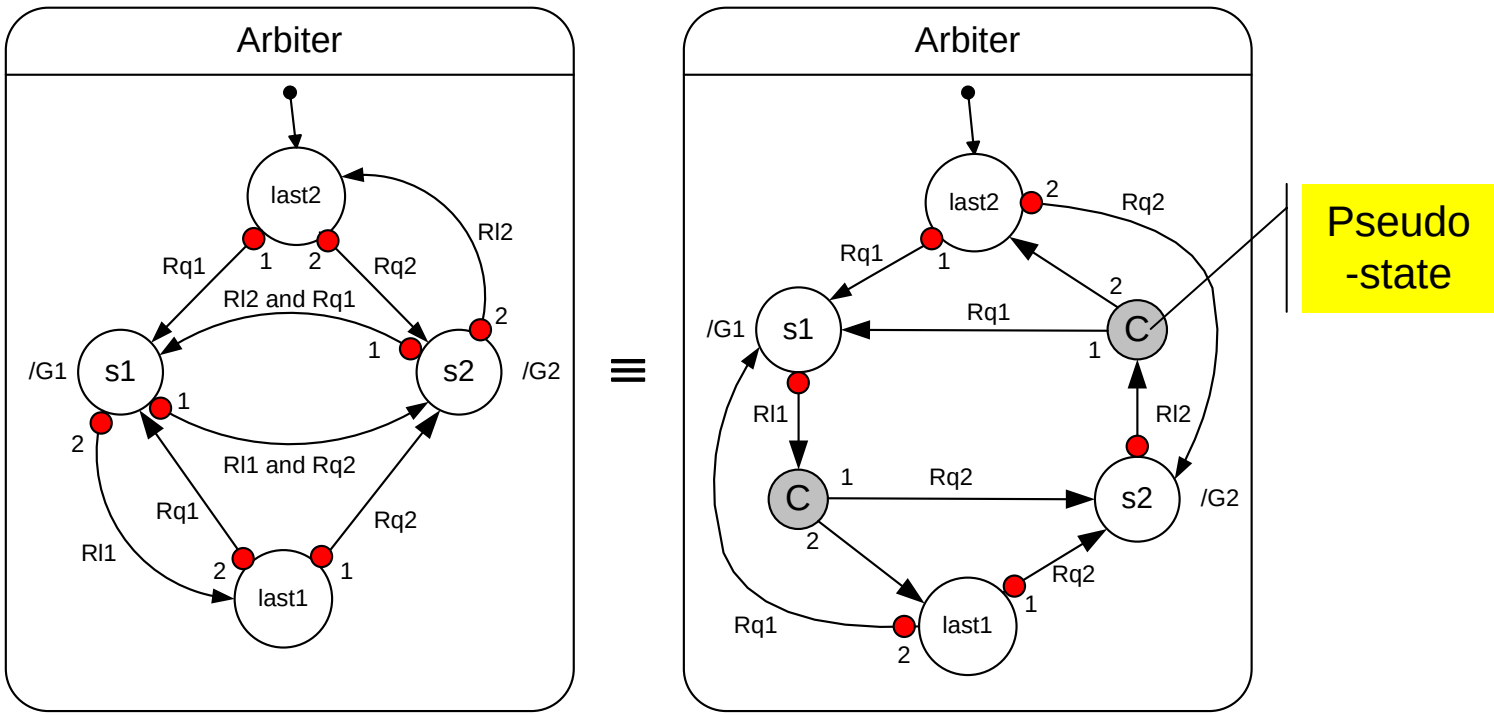
Strong abortion

Causality cycle



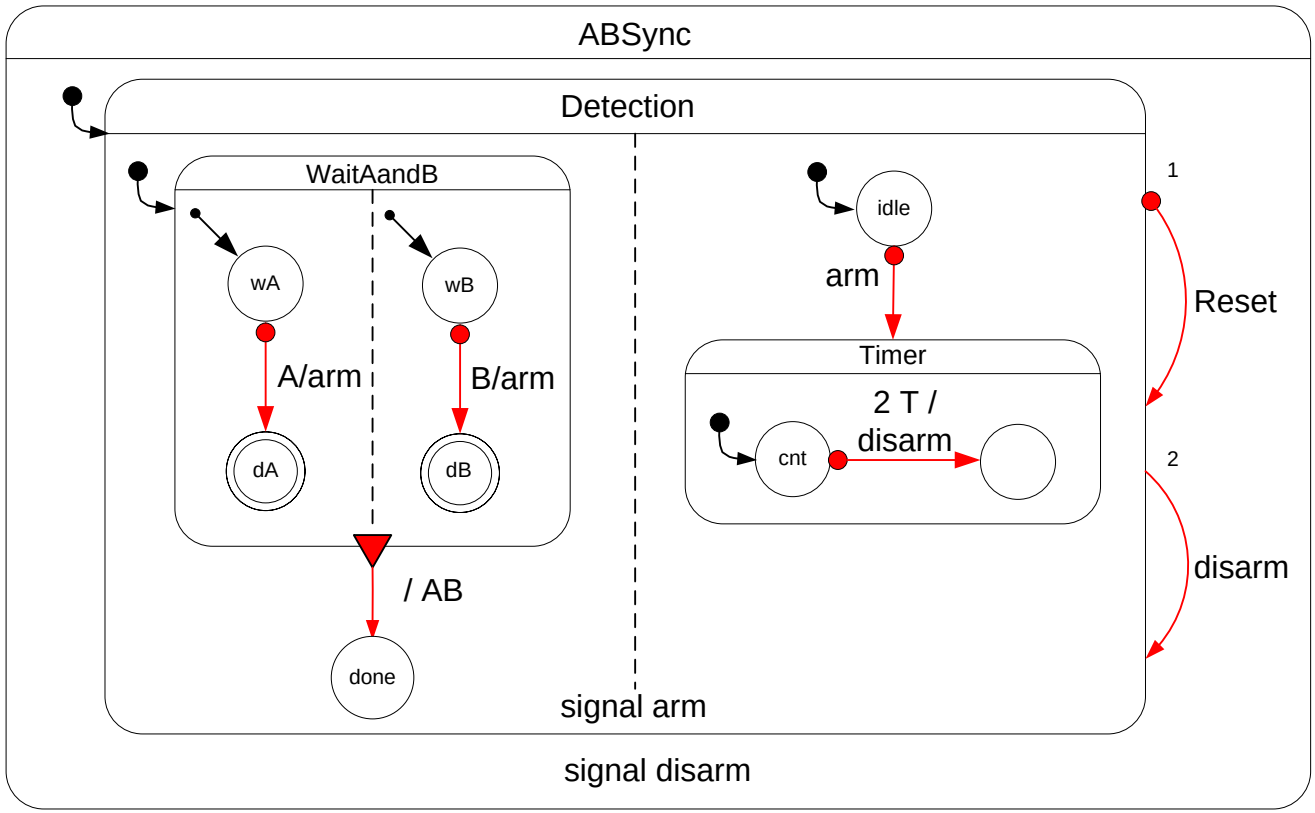
Additional Improvement

Turning Priority



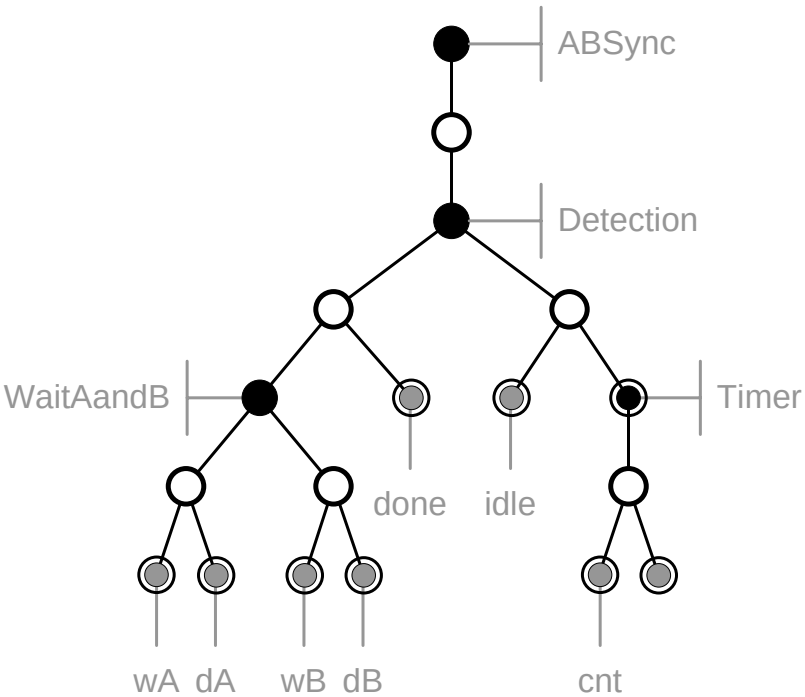
(A) (B)

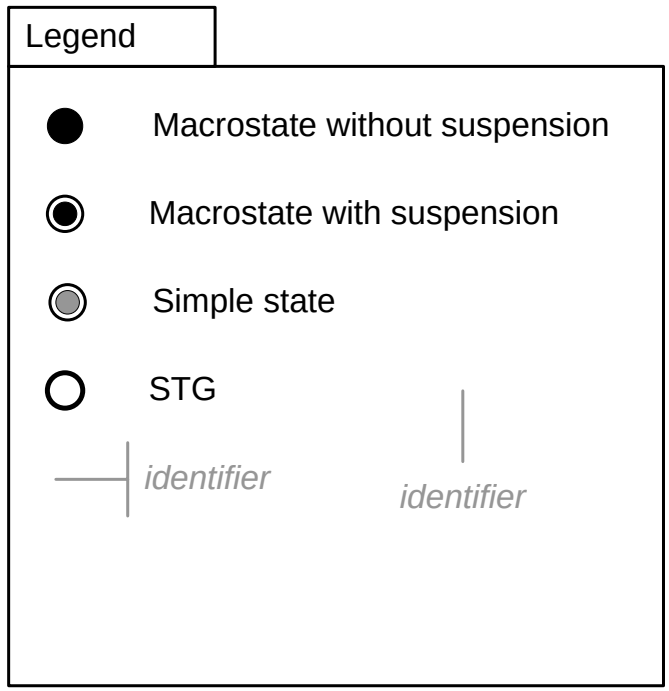
ABSync: a modified ABRO



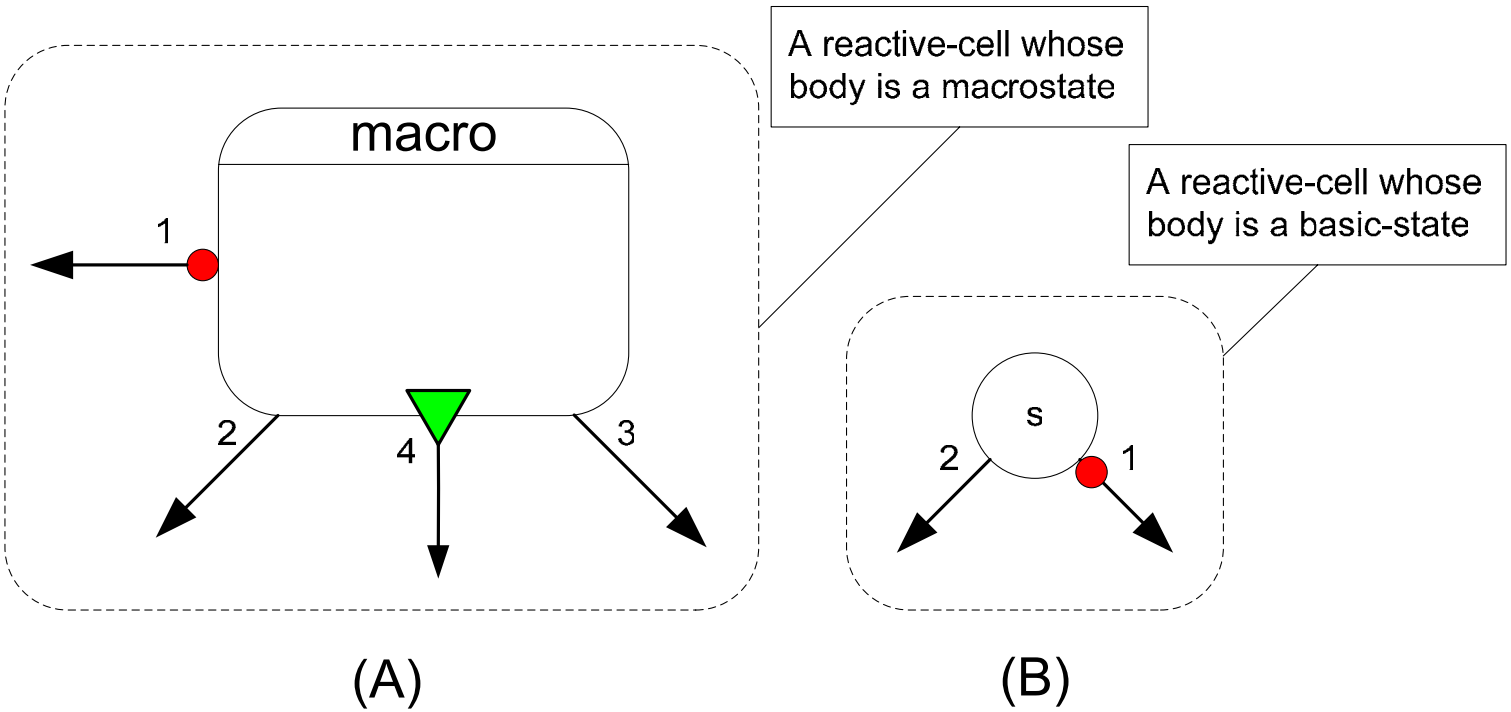
Send AB whenever A and B occur « close » enough
(no more than 2 occurrences of H in between)

ABSync: structure

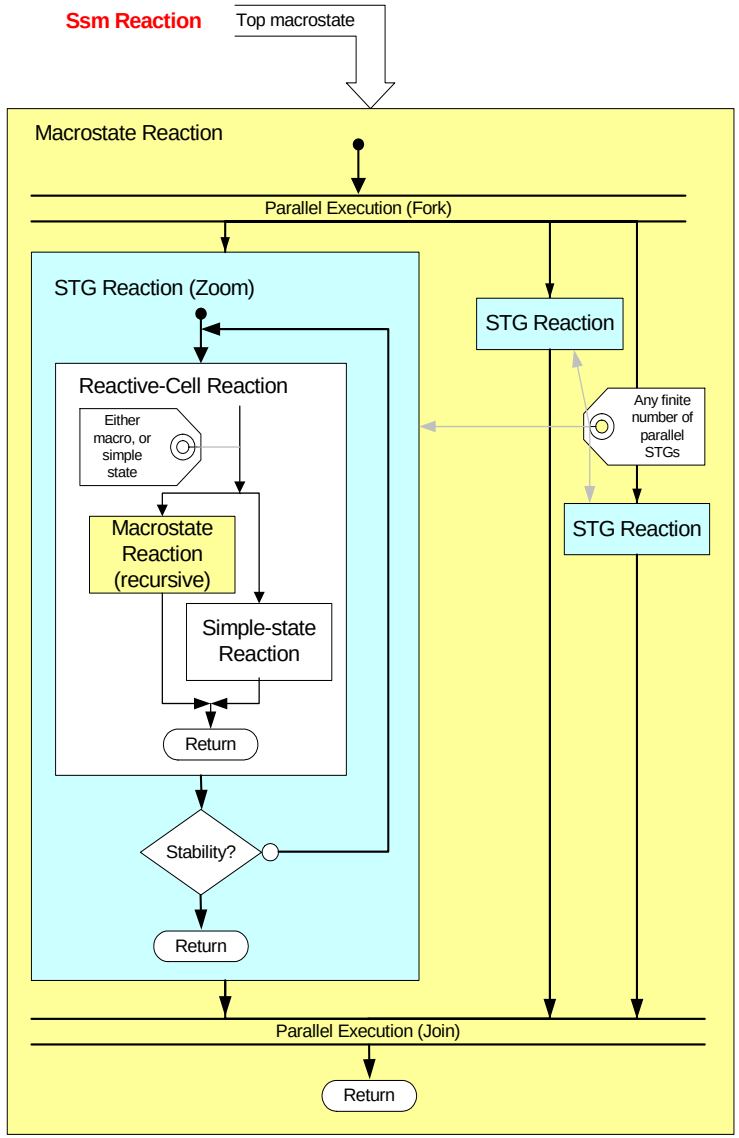




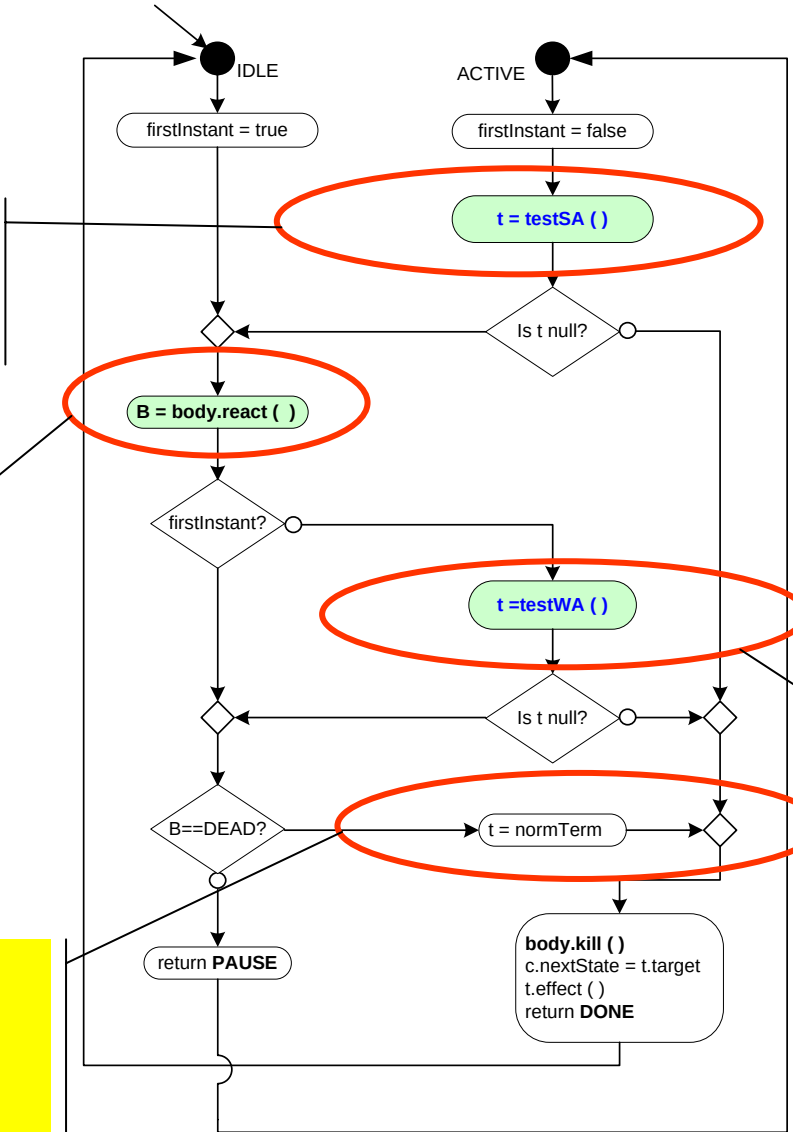
Reactive Cell



Reaction of a SyncChart



Reaction of a Reactive Cell



Test strong abortion first

Reaction of the body

Simplified version

Test weak abortion after

Test for possible normal termination