Reconciling Semantics, Implementation and Users

Gérard Berry

INRIA Sophia-Antipolis
http://www-sop.inria.fr/Gerard.Berry

Milner Symposium, Edinburgh, April 16-19, 2012
http://events.inf.ed.ac.uk/Milner2012/

My Personal Moto

You can’t protect yourself from good fortune

On n’est jamais à l’abri d’un coup de bol

My Priviledged Relation with Robin

- 1972 : Stanford AI Lab
  - proving bubble sort with LCF, with Robin as a personal mentor !
  - with a TV button on the terminal to watch football ☺
- 1976 (?) : reviewing the Full Abstraction paper
  - and devoting 7 years of my life to the problem
- 1981? : spending a week in the attic at ‘a Crescent
- 1982 : understanding that real-time is 100% different
  from Powerdomains, CCS, etc
  => Esterel

- 1983 (?) : understanding the virtues of SCCS
  - formalizing Esterel's synchrony principle
  - but SCCS was too weak for describing Esterel
  - fortunately, SuperGordon's SOS semantics came in !

- 1989 : Robin tells me he wants to come to Sophia for a sabbatical
  but I am myself going to Paris for a sabbatical !
  => Robin occupies my nice office at CMA

- 1989 : the CHAM paper is accepted at POPL but rejected by both referees for TCS
  - then accepted by the editor nevertheless ☺

- 1991 : first Milner Lecture in Old College
- 1992 : the Berry-Gonthier Esterel paper in SCP
  although no referee ever replied !
- 2000 : *The Foundations of Esterel*
  In *Proof, Language and Interaction: Essays in Honour of Robin Milner*
- 2005 : Robin elected foreign associate member
  of Académie des sciences

- 2010 : *Seven Keys to the Digital Futures*
  Informatics Forum, Edinbrrrrrrrgh
  Special thanks to Michael Fourman

Care for Users

Semantics vs. Implementation vs. users

- Domain equations
  - why such a gap between semantics and implementation?
  - make semantic as trivial as implementation !
- Synchronous concurrency
  - symmetry or asymmetry? SCCS vs. Esterel
  - why my users love asymmetric broadcasting
- CHAMonix = Chemical Abstract Machine
  - Transition rules are too intellectual, CHAM is for kids !
- Computing vs. proving
  - viewing circuits as semantic proof networks
  - proving that electrical logic is intrinsically constructive
  - the Hurry-Coward correspondence

Scott Domain Equations

- intTree $=$ int + int $*$ intTree $*$ intTree
- intTree $=$ Leaf of int

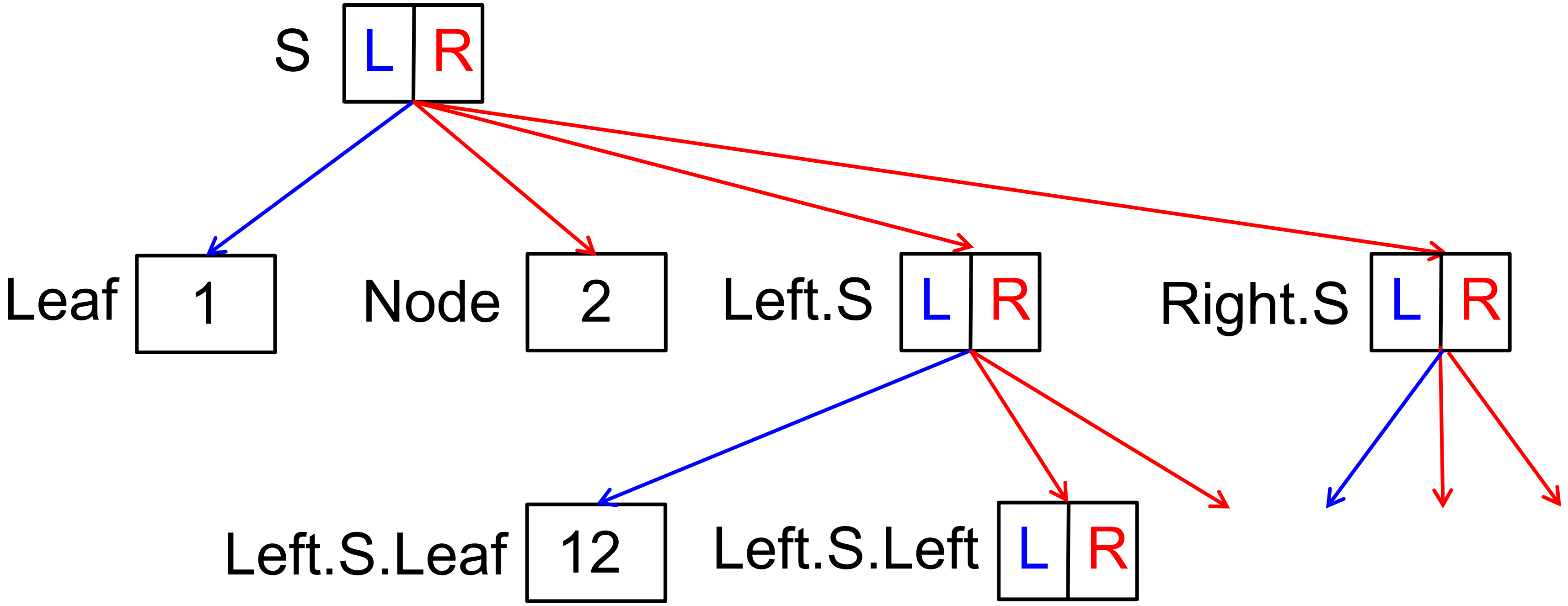
  | { Node of int;

  Left of intTree;

  Right of intTree };
- D $=$ int + (D $\to$ D)

What is `$=$' ? cpo isomorphism instead of equality !
=> take fixpoints in cpo categories instead of cpos

Concrete Data Structures (Kahn-Plotkin)



Beautiful cpo-representation theorem

CDS solve domain equations using only set equality

Function Spaces => Sequential Algorithms

$$C_1\ \boxed{tt \;\vdots\; ff}\ \ C_2\ \boxed{tt \;\vdots\; ff} \longrightarrow C\ \boxed{tt \;\vdots\; ff}$$

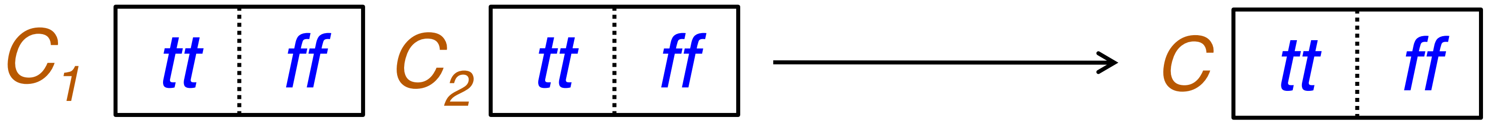
Sor :

```
$C \leftarrow C_1$ ?
        $tt \rightarrow C_2$ ?
                $tt \rightarrow$ ! $tt$
                $ff \rightarrow$ ! $tt$
        $ff \rightarrow C_2$ ?
                $tt \rightarrow$ ! $tt$
                $ff \rightarrow$ ! $ff$
```

Lor :

```
$C \leftarrow C_1$ ?
        $tt \rightarrow$ ! $tt$
        $ff \rightarrow C_2$ ?
                $tt \rightarrow$ ! $tt$
                $ff \rightarrow$ ! $ff$
```

Several Algorithms per Function

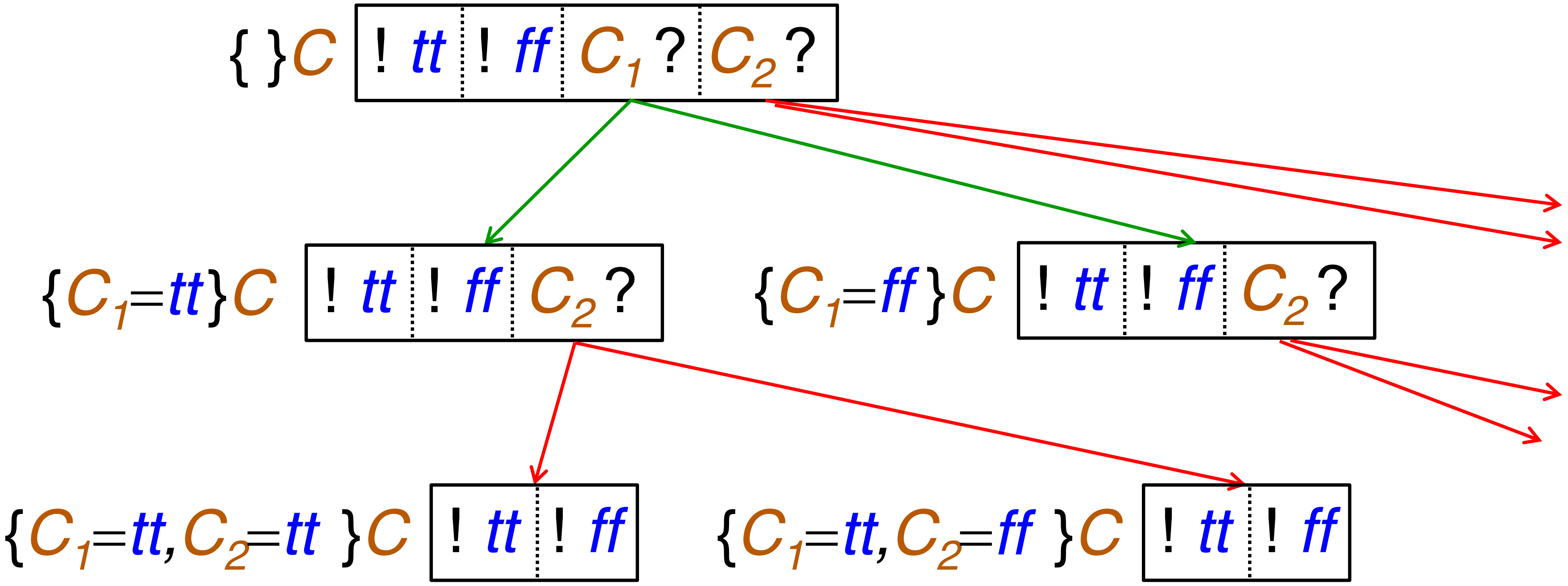


Sor-1-2 : $\neq$ Sor-2-1 :

```
Sor-1-2 :
  C ← C_1 ?
        tt → C_2 ?
               tt → ! tt
               ff → ! tt
        ff → C_2 ?
               tt → ! tt
               ff → ! ff
```

```
Sor-2-1 :
  C ← C_2 ?
        tt → C_1 ?
               tt → ! tt
               ff → ! tt
        ff → C_1 ?
               tt → ! tt
               ff → ! ff
```

Sequential Algorithms → CDS



$\{\,\}C$! *tt* ! *ff* C_1? C_2?

$\{C_1=tt\}C$! *tt* ! *ff* C_2?

$\{C_1=ff\}C$! *tt* ! *ff* C_2?

$\{C_1=tt, C_2=tt\}C$! *tt* ! *ff*

$\{C_1=tt, C_2=ff\}C$! *tt* ! *ff*

- Theorem: algorithms between CDS form a CDS

  Corollary: domain equations still solved by equality !

Parametrized Domain Equations

```
‘a Tree = Leaf of ‘a
              | { Node of ‘a;
                  Left of intTree;
                  Right of intTree};

‘a D = ‘a + ( ‘a D → ‘a D )
```

Semantics (Plotkin-Smyth) :
Much fancier category theory

Type checking : more clever (Milner)
Implementation: still nothing to do !

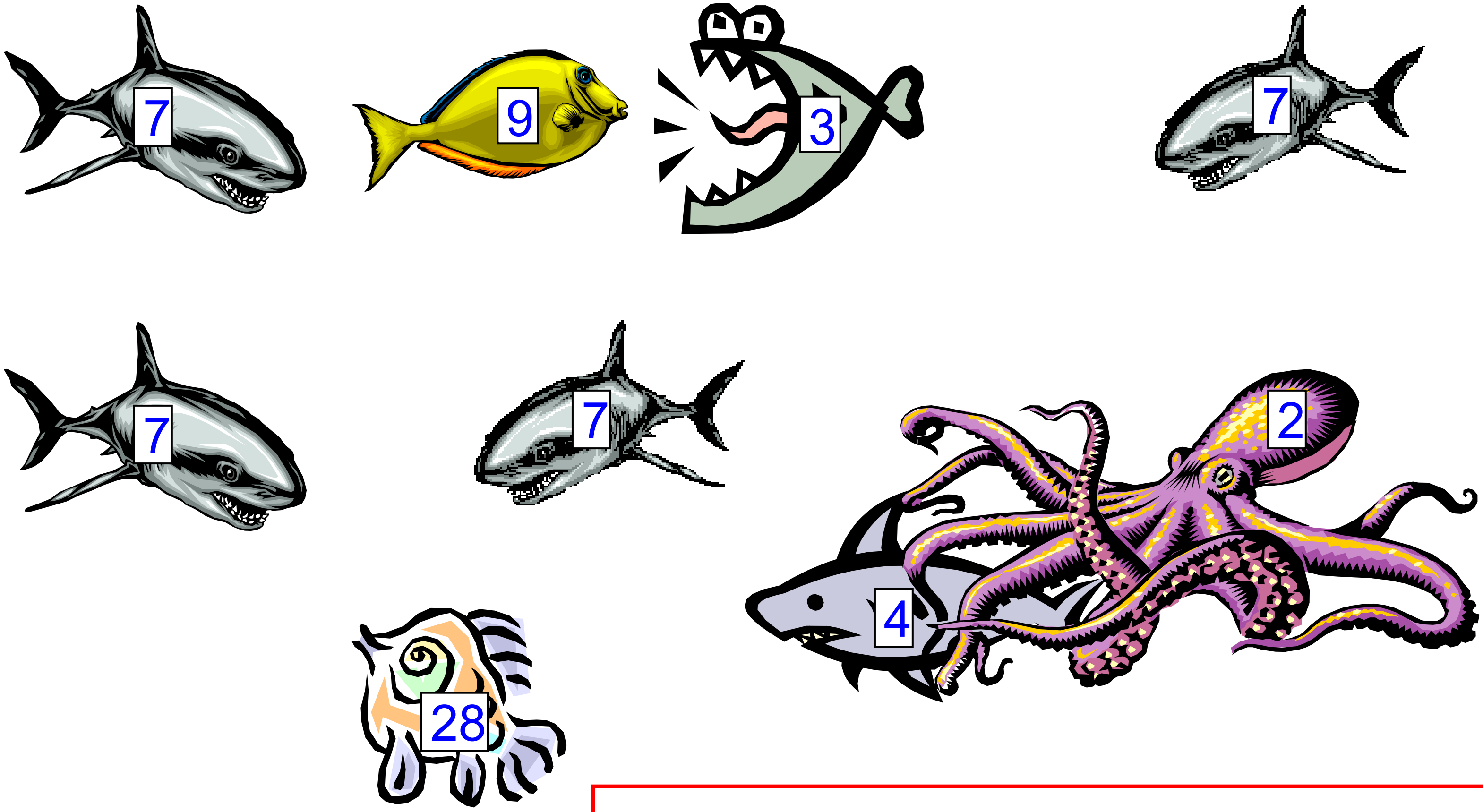
Why do you need aspirin + ibuprofen + paracetamol
to make the semantics
while there is no need for any code ?

CDS: because solution is by equality, not isomorphism!
=> standard pointer manipulation (≈ naming)

Science is sooooo conservative

- CDS led me to redefine a λ-calculus model as a CCC
- Synchronously, 3 papers proposed λ-models notions
  - Hindley-Longo
  - Barendregt
  - Meyer
- They turned out to be equivalent
- So I lost 3 to 1 !
- *But I still think I was right….*

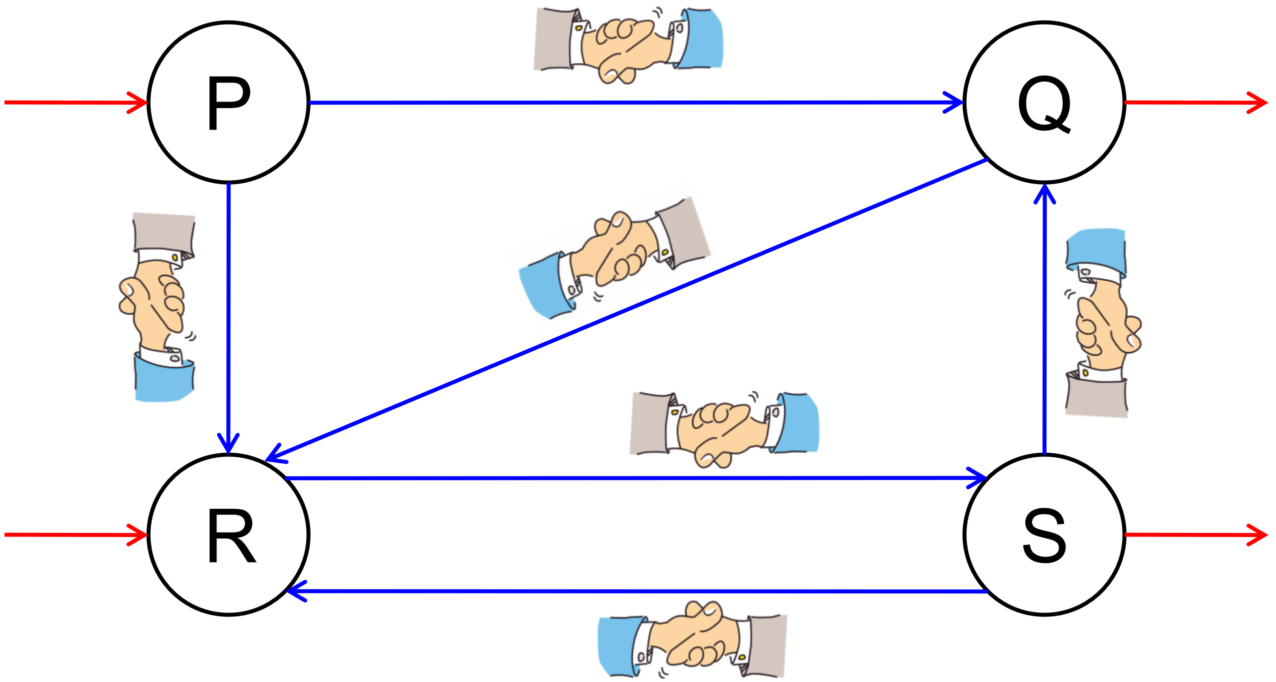
The Darwin Sieve: $p, kp \to p$



Banâtre - Le Métayer : GAMMA

Berry - Boudol : CHAM

CSP / CCS = Rendezvous



t := 10 ;

```
R :   P ? x : ...                S :   Q ! (z+1) : ...
    | Q ? y : ...                    | R ? t : ...
    | S ! 10 : ...                   | S ! 43 : ...
```

Deadlock

Lise and Laure

Starvation

Lise, Manon, and Laure

Information Propagation by Vibration

Nothing can illustrate vibration better than Bianca Castafiore, Hergé's famous prima donna. See [1] for details. The power of her voice forcibly shakes the microphone and the ears of the poor spectators.

[1] King's Ottokar Sceptre, Hergé, page 29, last drawing.

Sound, light, electrons, program counter, etc.

SCCS as a model ?

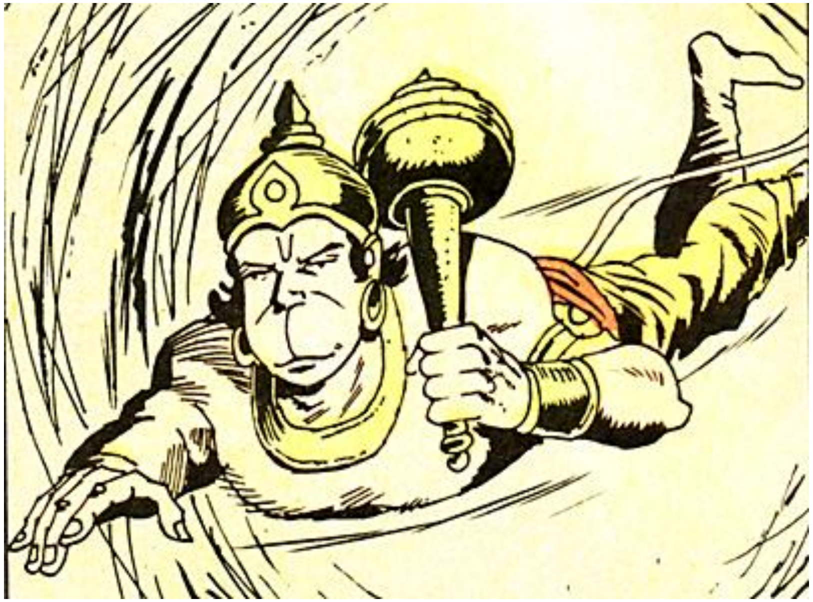
- Actions form a free group
  - $1, a, b, ab, a^{-1}$
  - $aa^{-1} = 1$
  - $abcd^{-1}$
- Parallel composition is product
- But not sufficient for Esterel!
  - X := 0; X:= X+1; ← at the same time, in the right order!

Triggering vs. Preemption

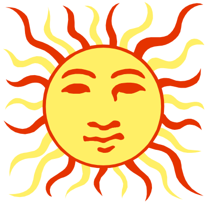
- CCS / SCCS: a.P = start P when a occurs
- Esterel : abort P when a = stop P when a occurs

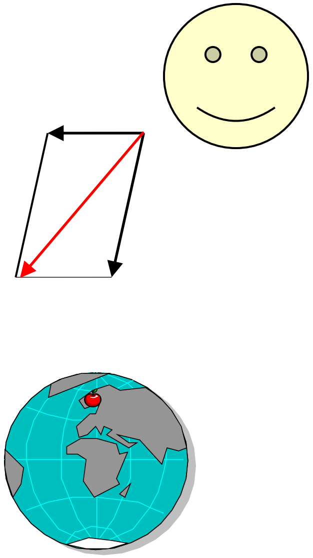
Hard and unnatural to specify in SCCS !

- SOS = SuperGordon
  to the rescue



Newtonian Mechanics =
Automatic 0-Delay Attraction Combination





SCCS: free combination
Esterel: vector addition

Concurrency + Determinism
Calculations are feasible

Zero Delay and Full Abstraction

Bianca Castafiore singing for the King Muskar XII in Klow, Syldavia. King's Ottokar Sceptre, page 38, first drawing.

Although the speed of sounds is finite, it is fast enough to look infinite. Full abstraction!

If room is small enough,
predictable delay implements zero-delay

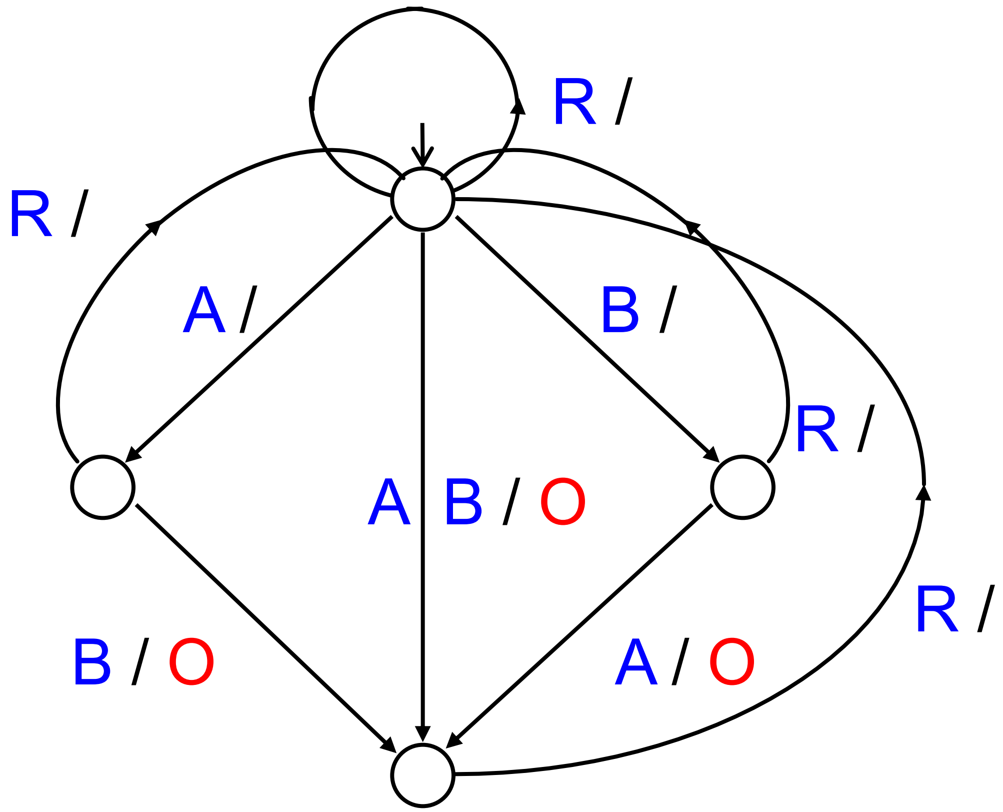
Specify with zero-delay
Implement with predictable delay
Control room size

George Heriot’s school open day, Oct. 2nd, 2010

Edinburgh is also a Synchrony City!

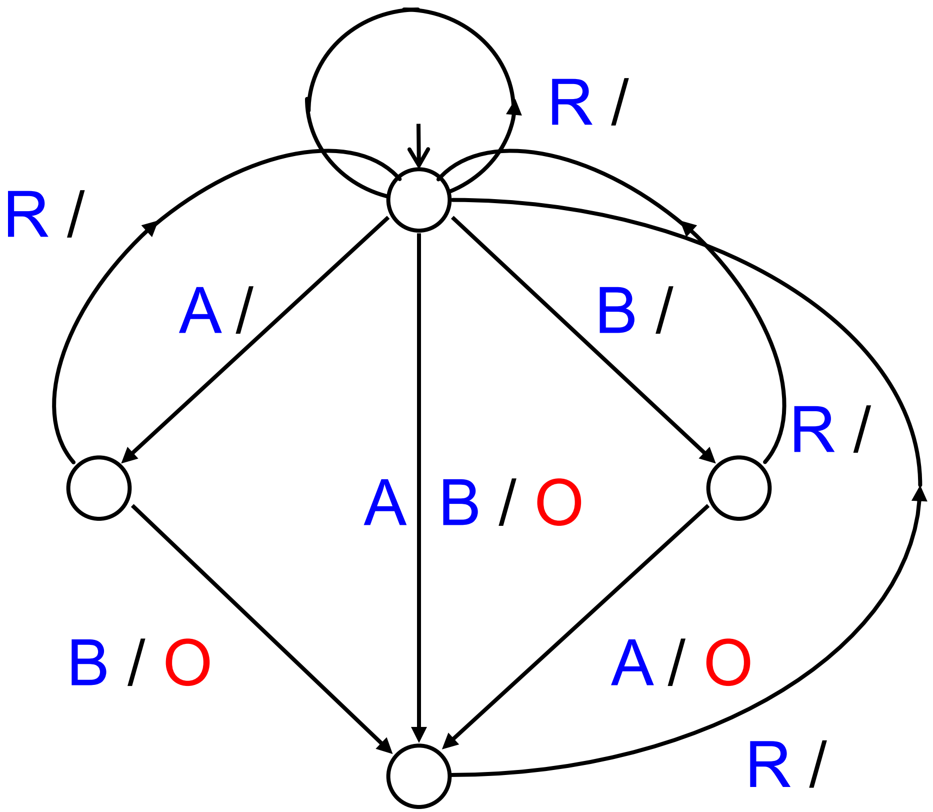
The ABRO Example

Emit O as soon as A and B have arrived
Reset behavior each time R is received



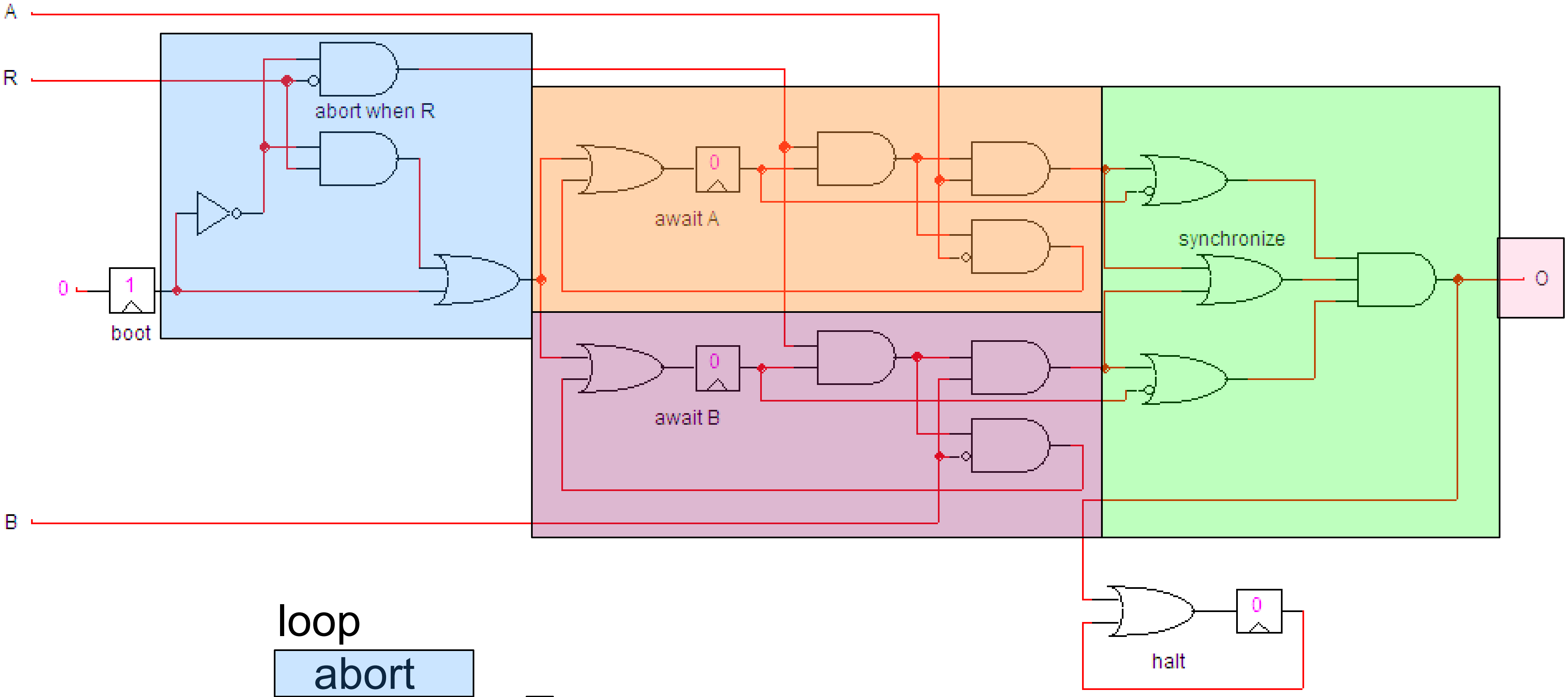
Memory write
R : Request
A : Address
B : Data
O : Write

Esterel = Linear Specification



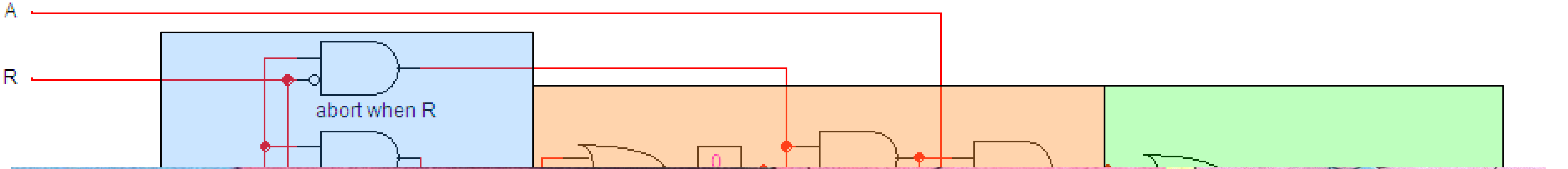
```
loop
   abort
      { await A || await B };
      emit O ;
      halt
   when R
end loop
```

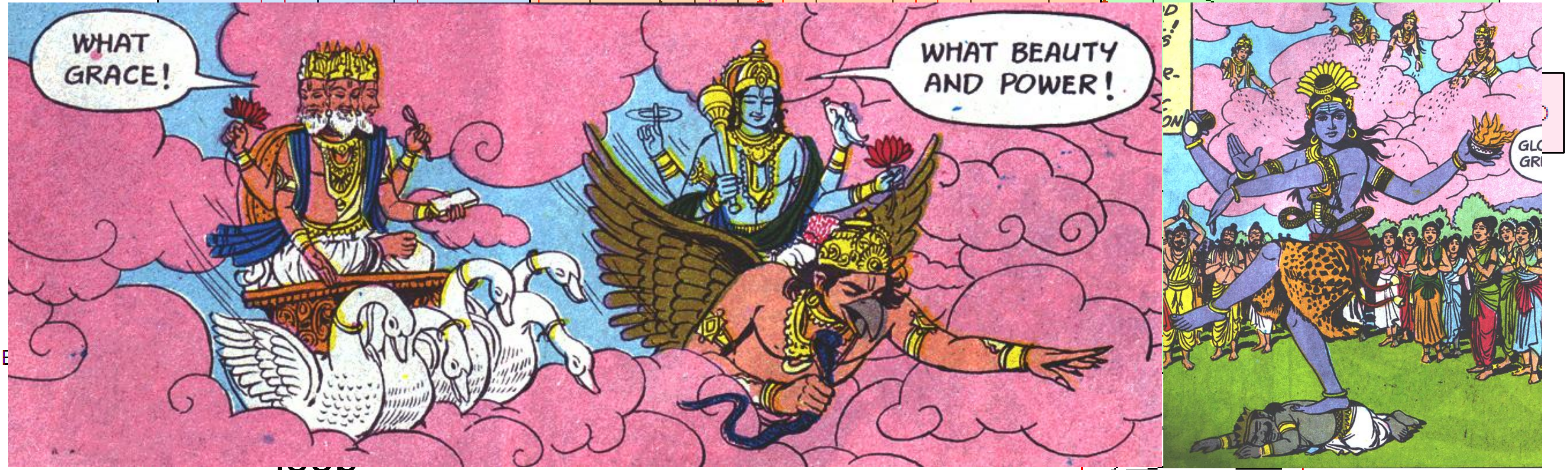
The Hierarchical ABRO Circuit

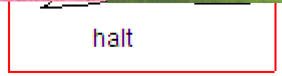


```
loop
  abort
     {await A || await B};
     emit O ;
     halt
   when R
end loop
```

The Hierarchical ABRO Circuit



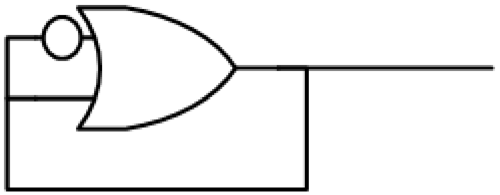




```
loop
   abort
      { await A || await B };
      emit O ;
      halt
   when R
end loop
```

Electicity is Constructive

Hamlet : ToBe = ToBe or not ToBe



ToBe

Electrically stabilizes to for some gate and wire delays but not for all delays

Question : When does a cyclic circuit stabilize for all gate and wire delays?

Theorem: if and only if equations can be solved
using only Constructive Boolean Logic
(forward propagation of 0’s and 1’s)

(Mendler, Shiple, and Berry, FMCS April 2012)