Design and Testing of Modular Systems by (FSM) Language Equation Solving

Alexandre Petrenko

LIG-CRIM

(Centre de recherche informatique de Montréal, Canada)

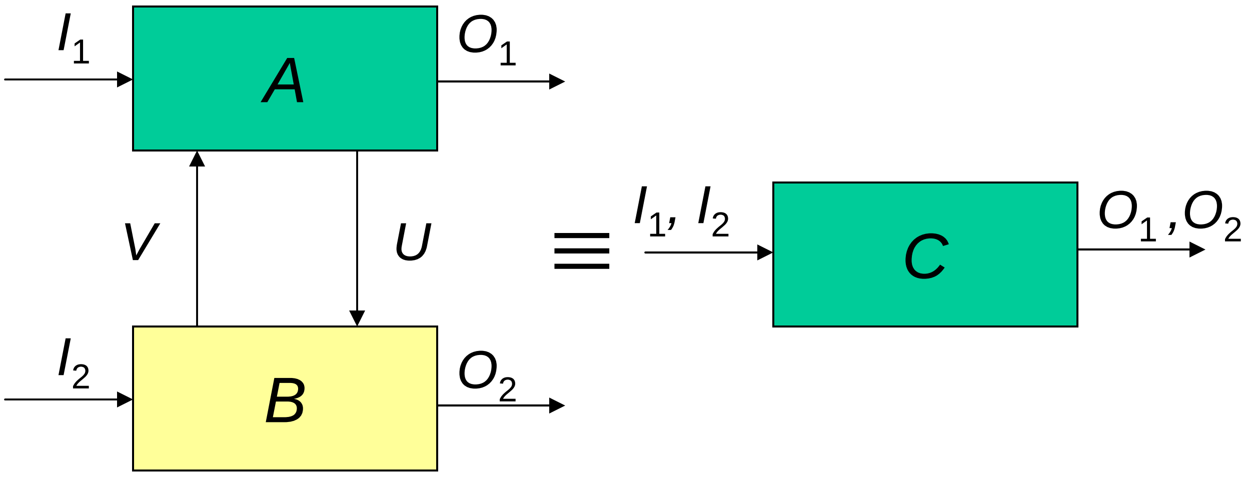
Aussois, Journées EmSoC 2007

Design and Testing of Modular Systems by (FSM) Language Equation Solving

Unknown Component Problem

Design a component that when combined with a known part of the system satisfies a given overall specification:

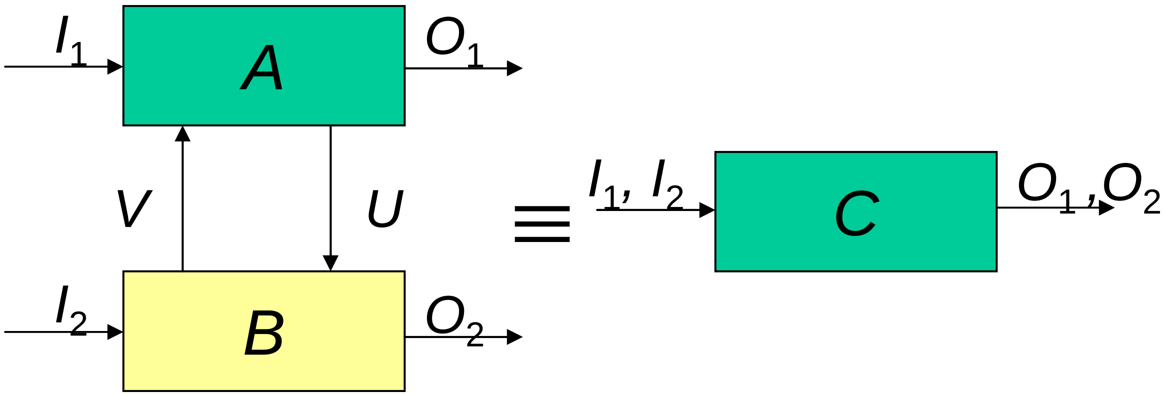
Given *A* and *C*, find *B*



Related Problems and Applications

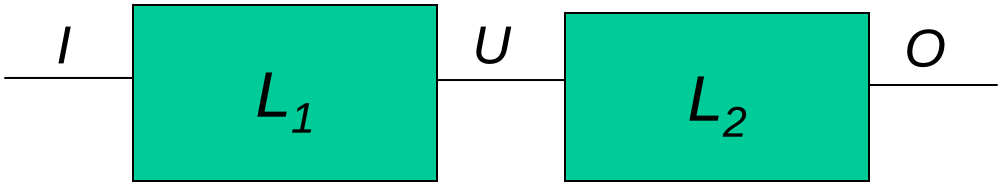
- logic synthesis and sequential logic optimization
- supervisory control and design of discrete controllers
- substitutability in compositional frameworks
- wrapper design
- protocol conversion
- testing modular systems

Making it Formal



- Model of the system and the specification
  - finite automata
  - input/output finite state machines
- Interactions between components
  - synchronous: instantaneous communication (HW)
  - asynchronous: delayed, message based (SH)
- Conformance of the system to its specification
  - language containment
  - simulation

Language Compositions



synchronous composition is the language over $I \times O$

$[(L_1)_{\uparrow O} \cap (L_2)_{\uparrow I}]_{\downarrow I \times O} = L_1 \bullet L_2$

parallel (*asynchronous*) composition - over $I \cup O$

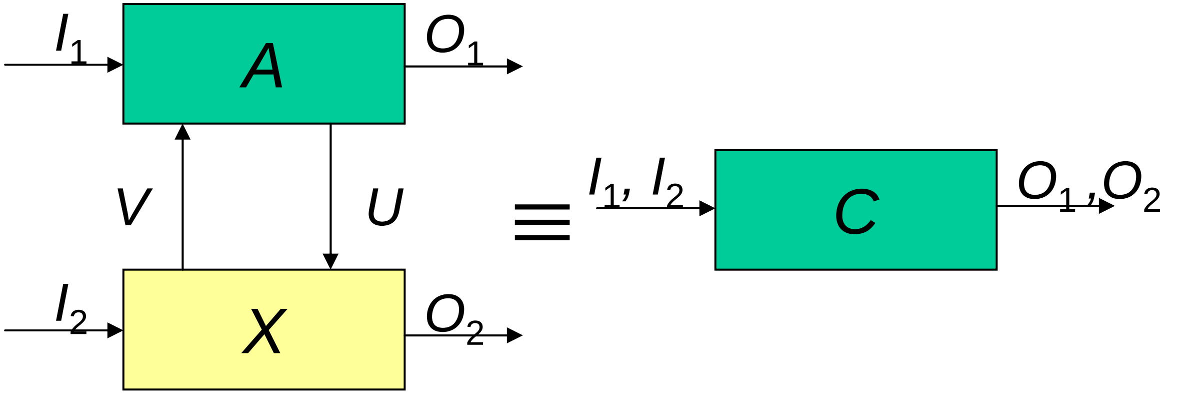
$[(L_1)_{\Uparrow O} \cap (L_2)_{\Uparrow I}]_{\Downarrow I \cup O} = L_1 \Diamond L_2$

$(L_1)_{\uparrow O}$ is the language over $I \times O$ and $(L_1)_{\Uparrow O}$ over $I \cup O$

Abstract Language Equations

- $A \bullet X \subseteq C$ (synchronous equation)
- $A \lozenge X \subseteq C$ (asynchronous equation)
- the most general solutions to the language equations are $S = \overline{A \bullet \overline{C}}$, and $S = \overline{A \lozenge \overline{C}}$
- There is an effective way to solve equations over regular languages
- Abstract language equations are specialized to languages of finite automata and finite state machines

Solving FSM Equations

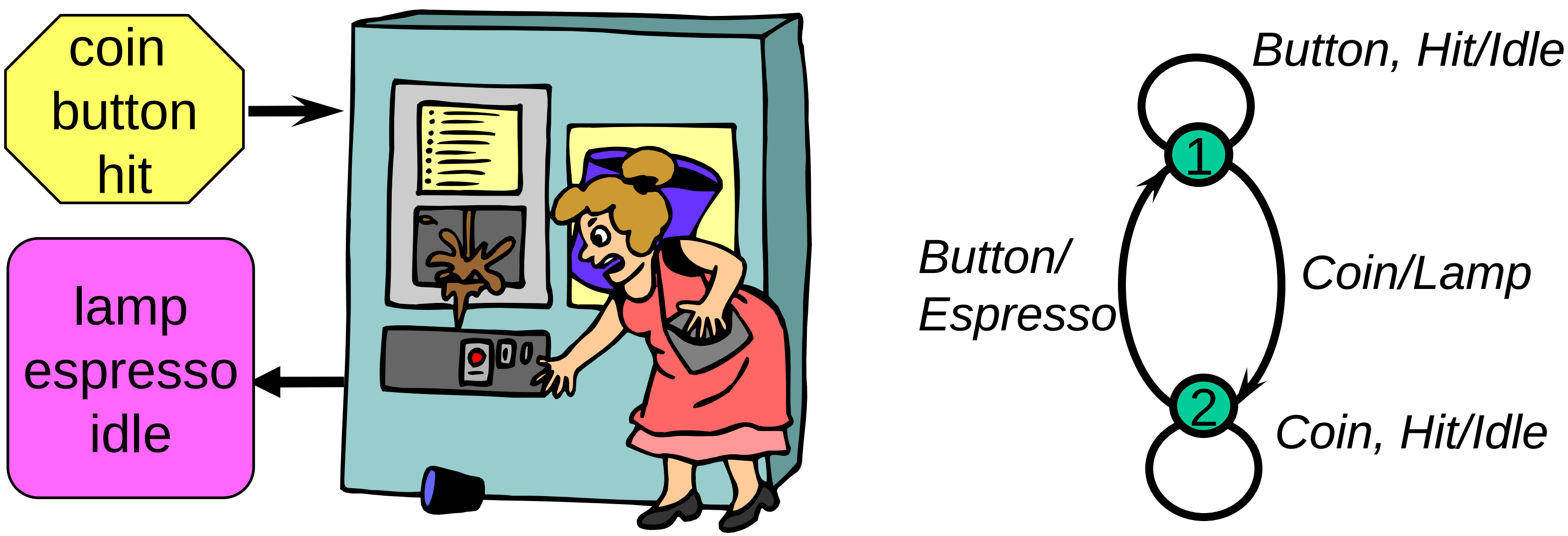


- $A \lozenge X \leq C$, $A \bullet X \leq C$
- Find a largest solution to the language equation
- Determine max prefix-closed I/O sub-language
- Find max progressive and convergent sub-language (deadlock and livelock removal, if needed)
- Build an FSM

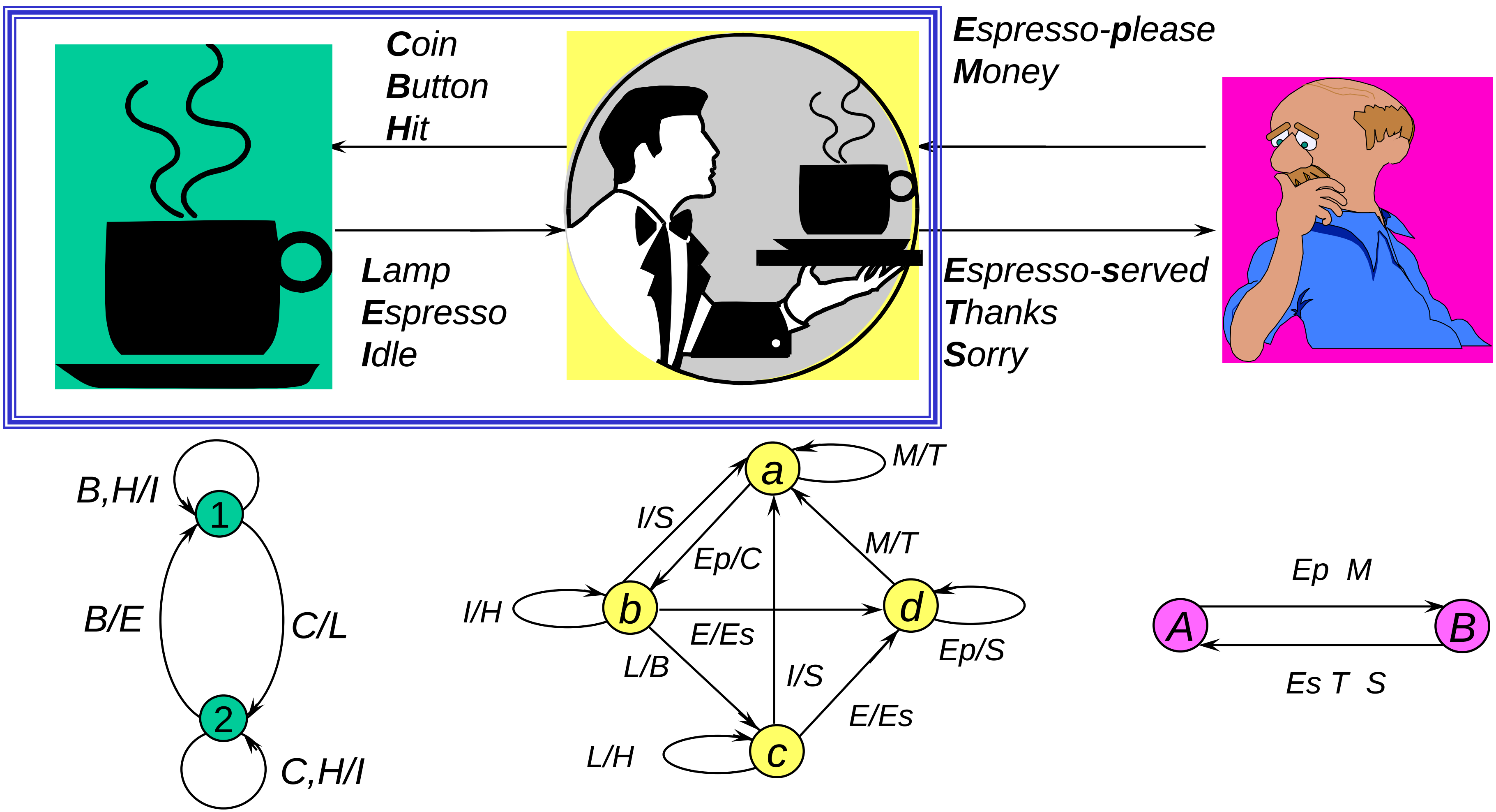
Testing Modular Systems

- Derive a test suite using external inputs and outputs
- Variety of testing problems
  - faults in
    - individual components
    - channels
  - some components are fault-free
  - distributed testing with synchronized testers

Testing in Isolation: Vending Machine



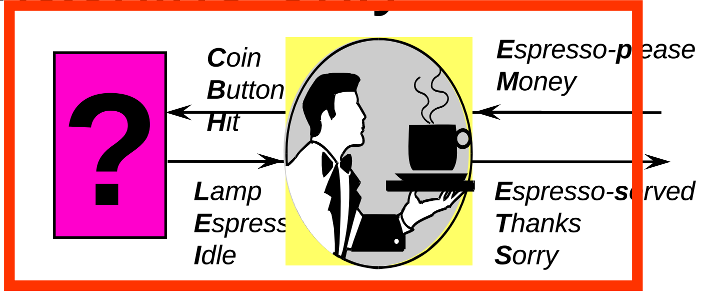
Coffee Shop



Coin
Button
Hit
Lamp
Espresso
Idle
Espresso-please
Money
Espresso-served
Thanks
Sorry
B,H/I
1
B/E
C/L
2
C,H/I
a
M/T
I/S
Ep/C
M/T
I/H
b
d
E/Es
Ep/S
L/B
I/S
E/Es
L/H
c
Ep M
A
B
Es T S

Testing Embedded Component

Assume we can test the coffee machine only via the waiter

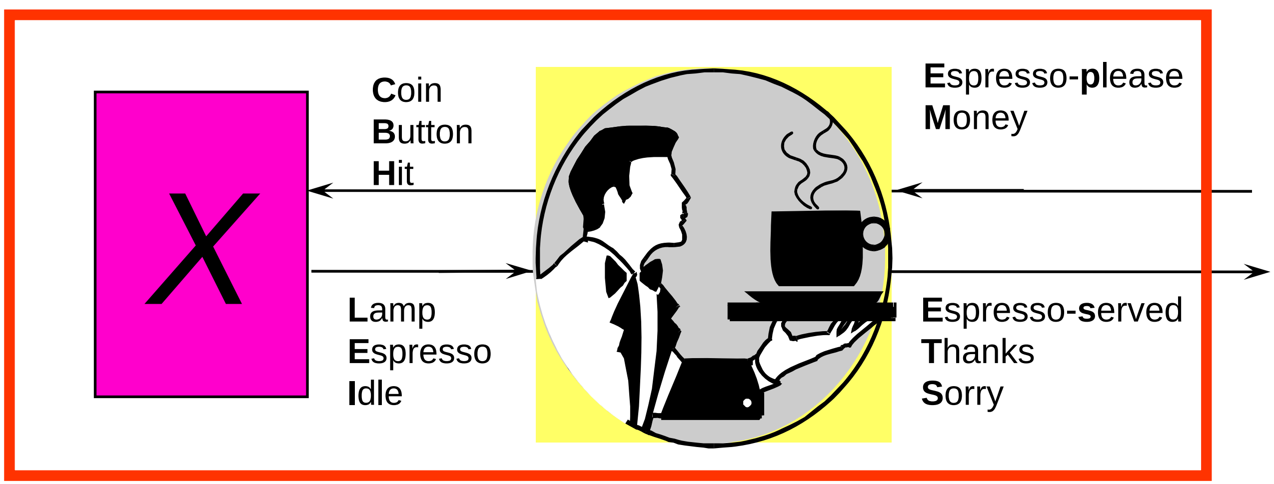


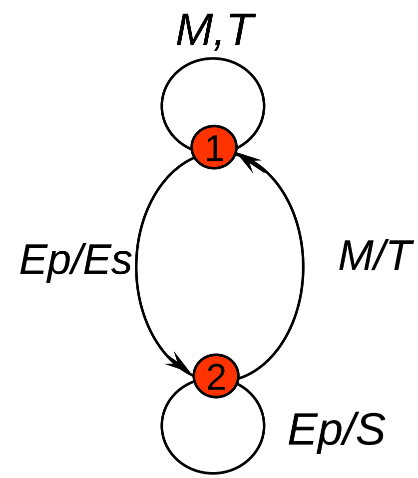
- it is neither directly *controllable*
- nor directly *observable*

- Can we characterize all undetectable faults?
- Can it be substituted by another coffee machine?
- How to generate tests for an embedded component?
- Active research topic

Undetectable Faults and FSM Equation

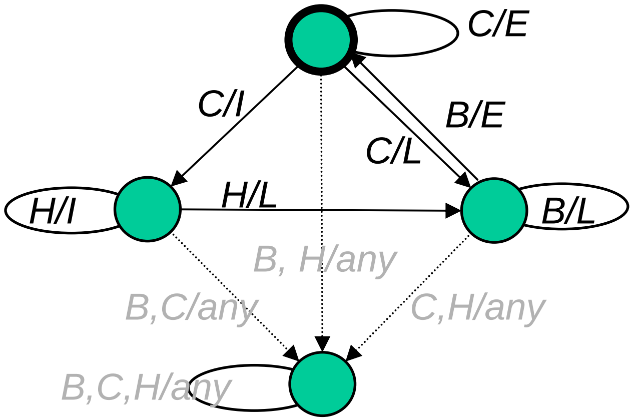
$$X \Diamond \mathit{Waiter} \cong \mathit{CoffeeShop}$$



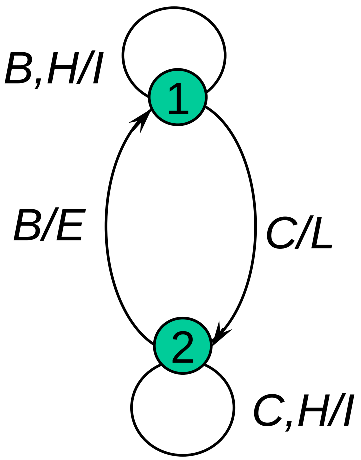


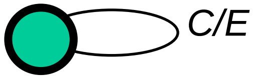
Externally Equivalent Coffee Machines Represent Undetectable Faults

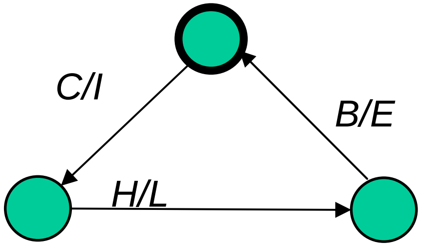
Non-deterministic largest solution



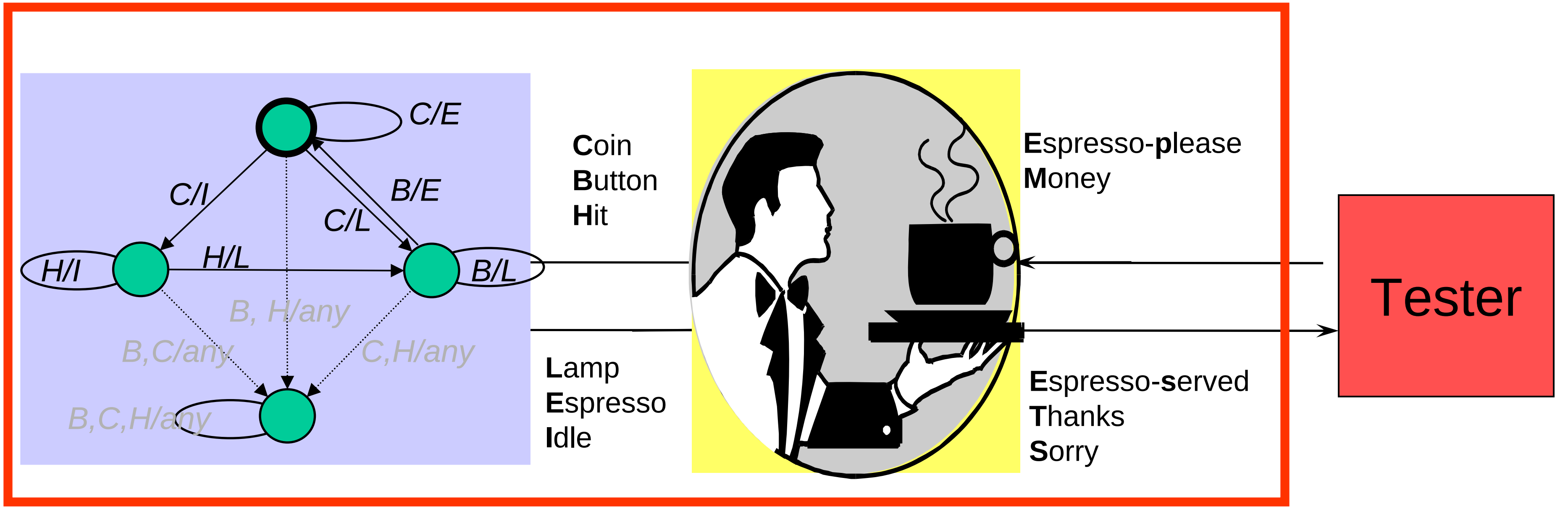
Some of the deterministic reductions are







Testing Embedded Coffee Machine



The solution becomes the specification of the component under test

Tests can be obtained in two steps:

1) Generate tests from the solution (non-deterministic FSM)

2) Translate internal tests into external

Some References

Collaborative work with

R. Brayton, T. Villa, and A. Sangiovanni-Vincentelli (Berkeley, Parades Labs)
N. Yevtushenko (Russia)

- *Fault Detection in Embedded Components*. International Workshop on Testing of Communicating Systems, 1997
- *Solving Asynchronous Equations* Formal Description Techniques for Distributed Systems and Communications Protocols, 1998
- *Solution of Parallel Language Equations for Logic Synthesis*. International Conference on Computer Aided Design, 2001
- *Sequential Synthesis by Language Equation Solving*. Technical report, Berkeley, CA, 2003
- *Compositionally Progressive Solutions of Synchronous FSM Equations*, Journal of Discrete Event Dynamic Systems, 2007, accepted
- *The Unknown Component Problem: Theory and Applications*, Springer, to appear in 2008

merci beaucoup