From Test Purposes to Asynchronous Test Cases

Adenilso Simão (Ades)*
adenilso@icmc.usp.br

Alexandre Petrenko
petrenko@crim.ca

Centre de Recherche Informatique de Montreal (CRIM), Canada
*On a leave from Universidade de São Paulo, Brazil

A-Most/ICST @ Paris, April, 2010

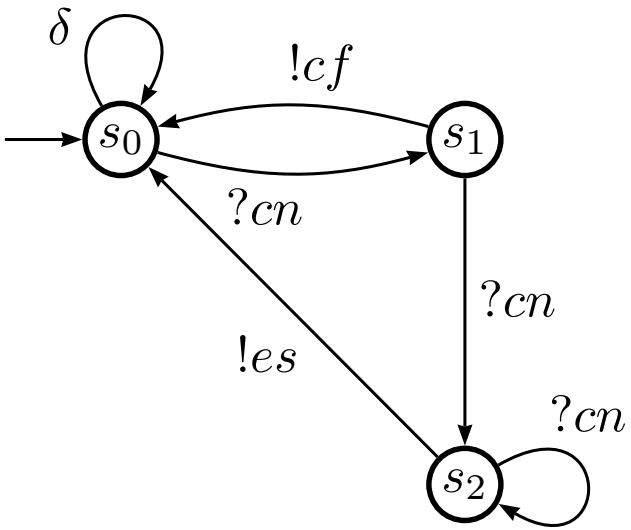
Problem Statement

- Given an Input/Output Transition System (IOTS)
  - Specification
- And a test purpose
  - Representing some functionalities
- Generate a test case
  - The communication between tester and implementation should non-blocking

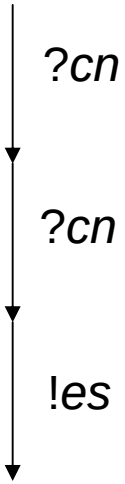
Test Purpose and Test Case

- Test Purpose is an IOTS
  - Set of finite output-branching traces to be executed/observed
  - No verdicts
- Test Case is an IOTS which should always reach a verdict state
  - **fail**, **pass** or **inc**
- A sound test case
  - No **fail** for good implementation

Example

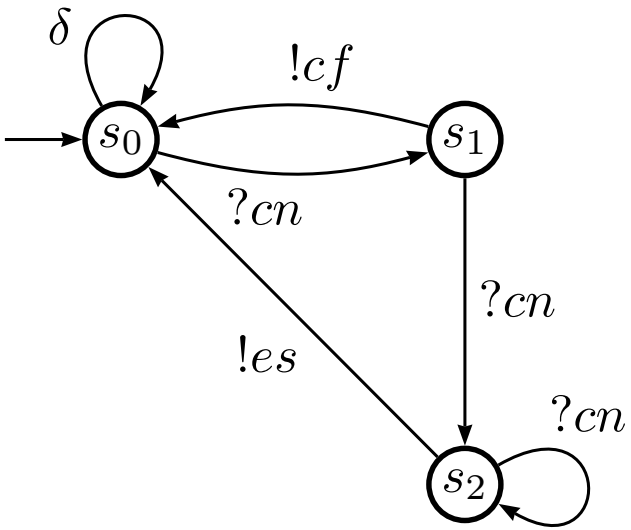


Test Purpose

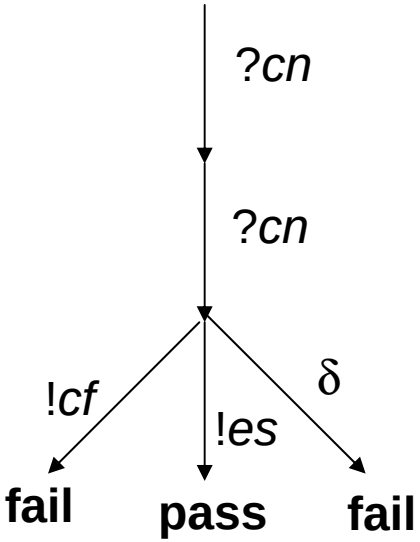


!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Example

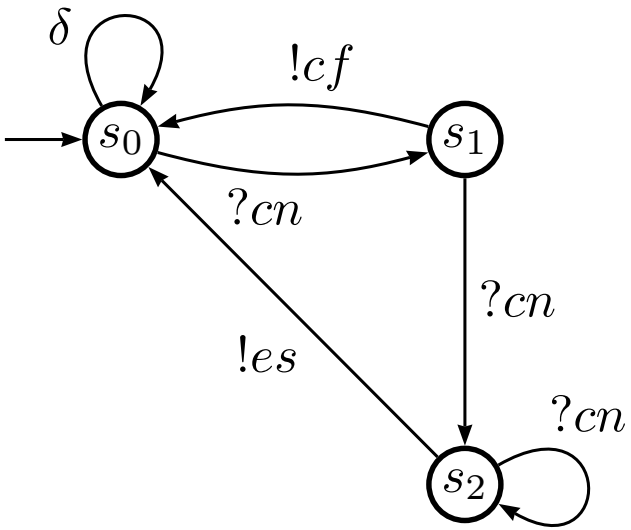


Synchronous Test Case

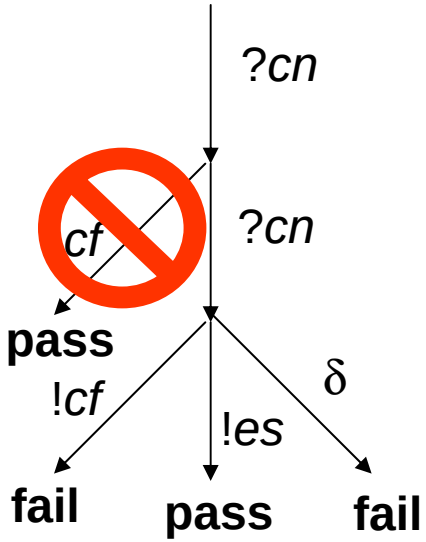


!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Example



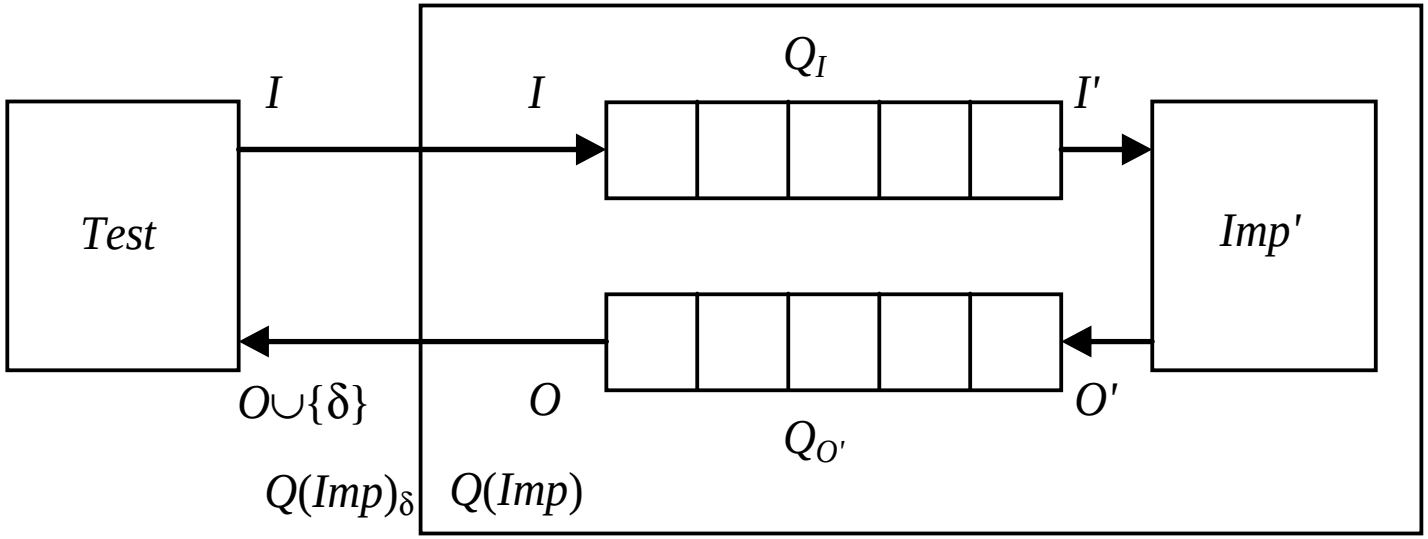
Synchronous Test Case



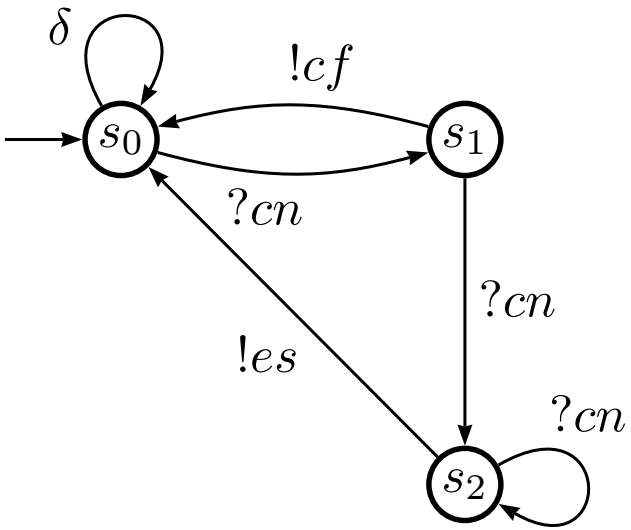
!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Test Architecture

- Asynchronous communication
  - Via queues

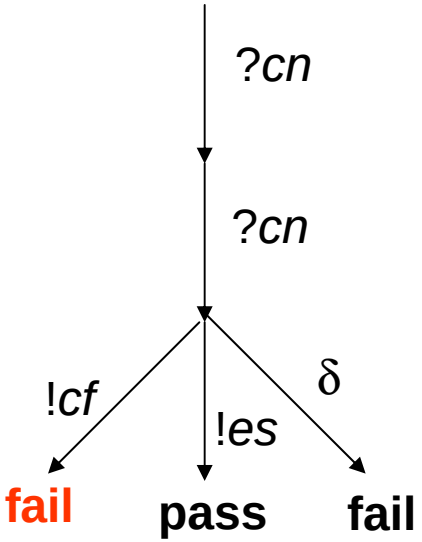


Example



!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Unsound Asynchronous Test Case



Unsoundness is due to queue distortion

Key Issue

- How to take into account the distortion due to queues?

Queue Composing Approach

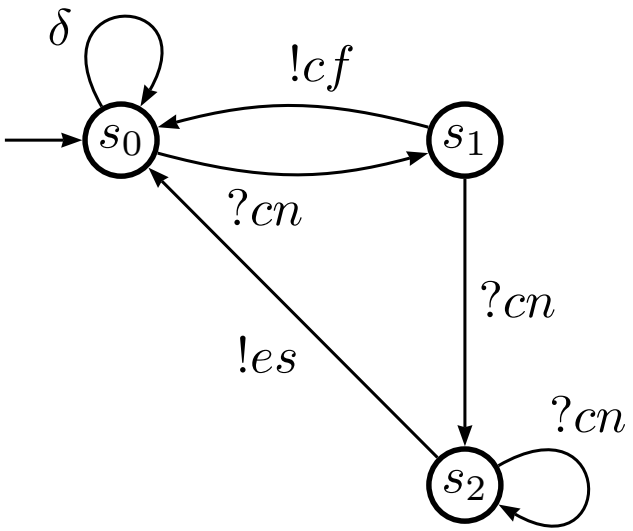
- Compose the specification with queues before test generation
  - Proposed in previous work
  - State explosion

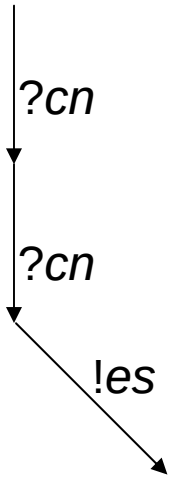
Proposed Solution

- Avoid the explicit composition of the specification, queues and the test purpose by finding an appropriate transformation of the test purpose

Building Sound Asynchronous Test Case

- The test purpose

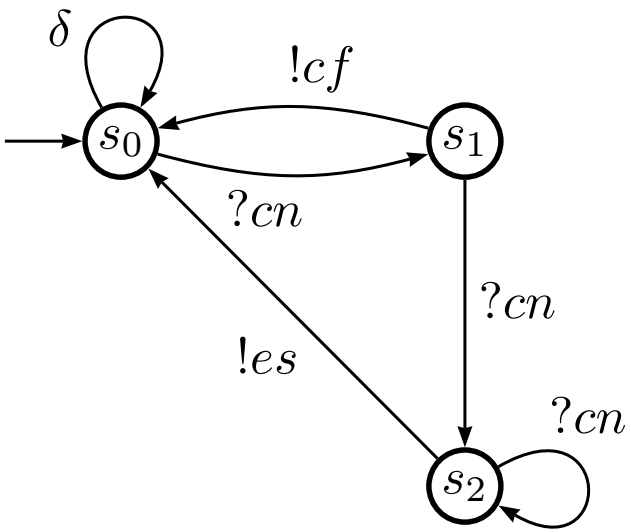


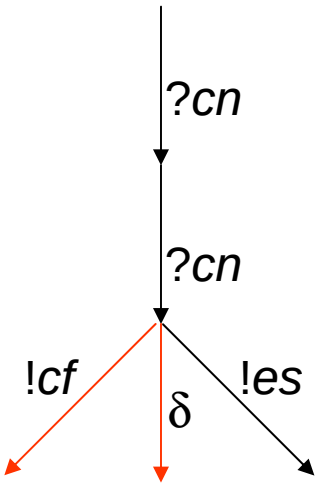


!*cf* = *coffee*
!*es* = *espresso*
?cn = *coin*
δ = *quiescence*

Building Sound Asynchronous Test Case

- Adding outputs

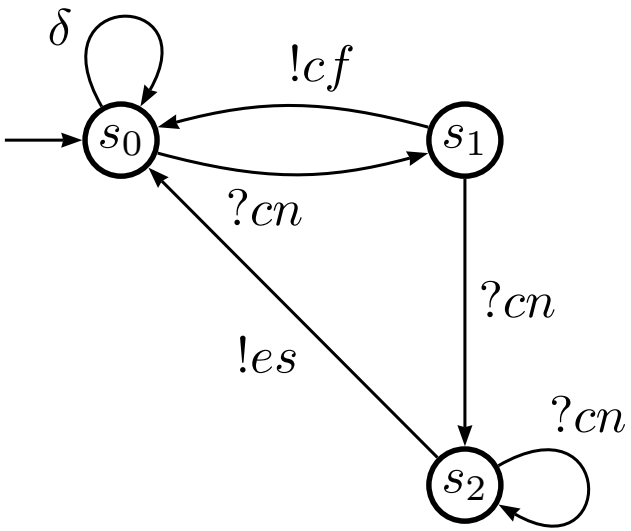


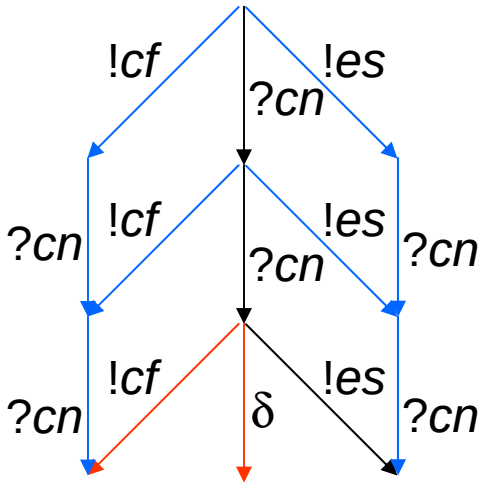


!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Building Sound Asynchronous Test Case

- Delaying inputs

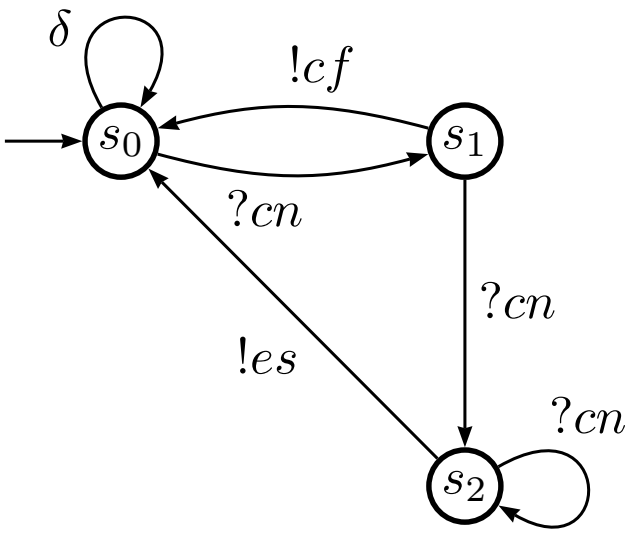


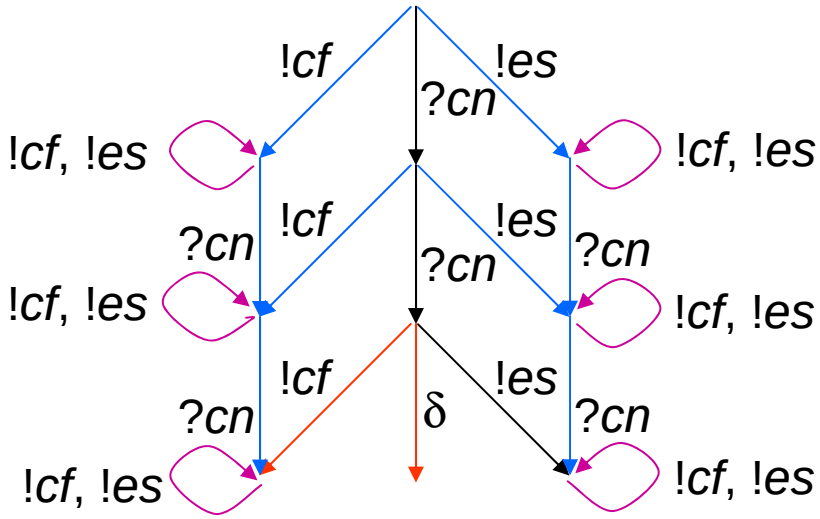


!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Building Sound Asynchronous Test Case

- Output completion

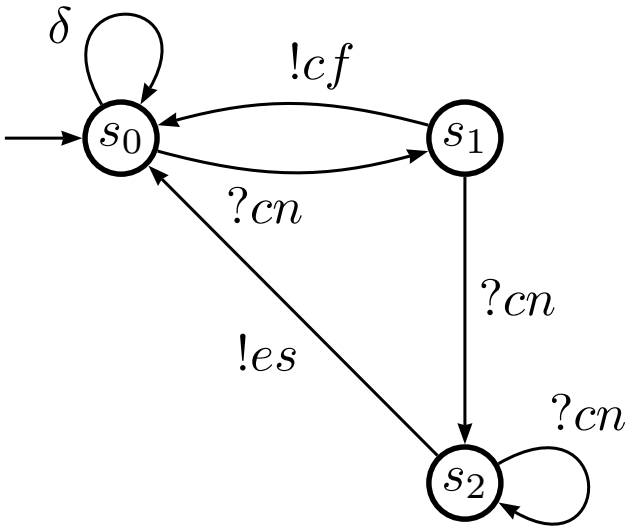


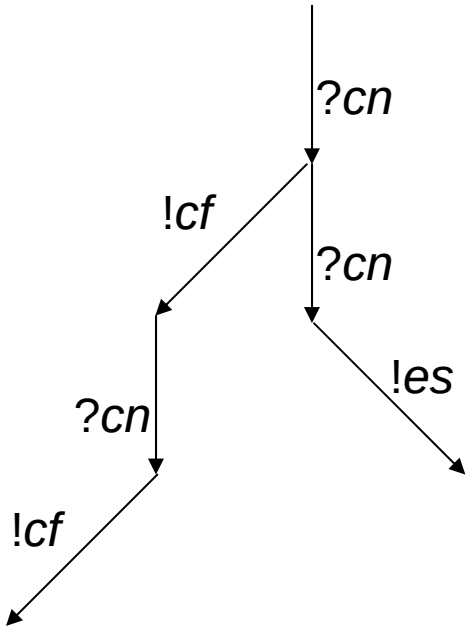


!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Building Sound Asynchronous Test Case

- Composing with Specification

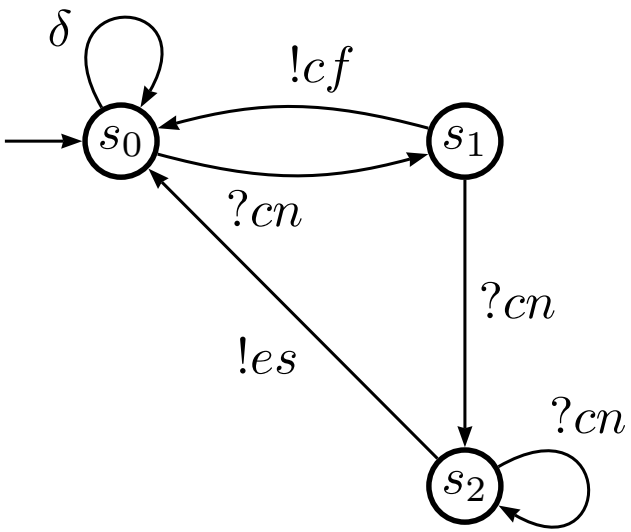




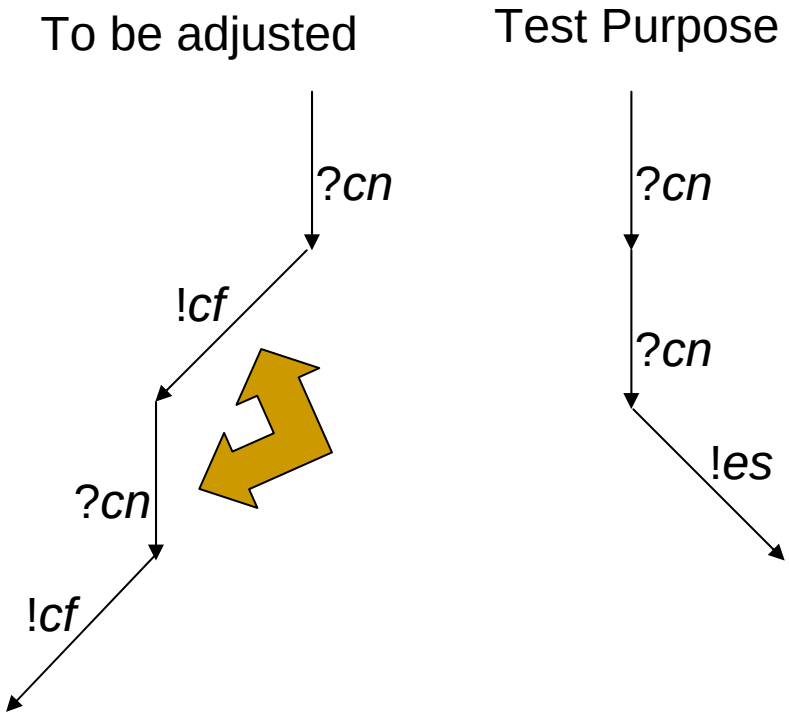
!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Building Sound Asynchronous Test Case

- Delaying Outputs

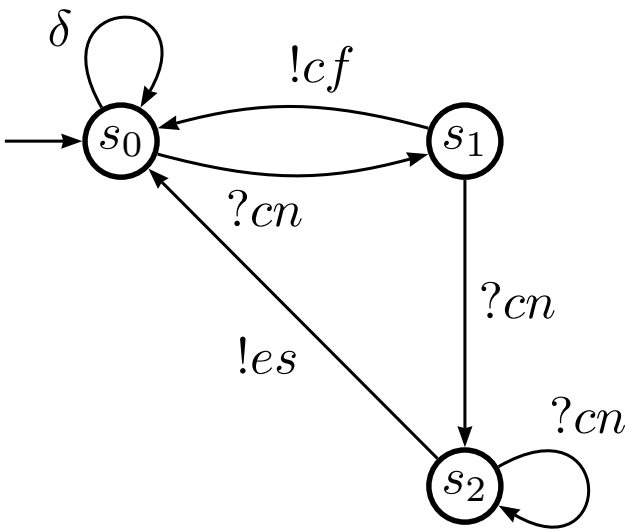


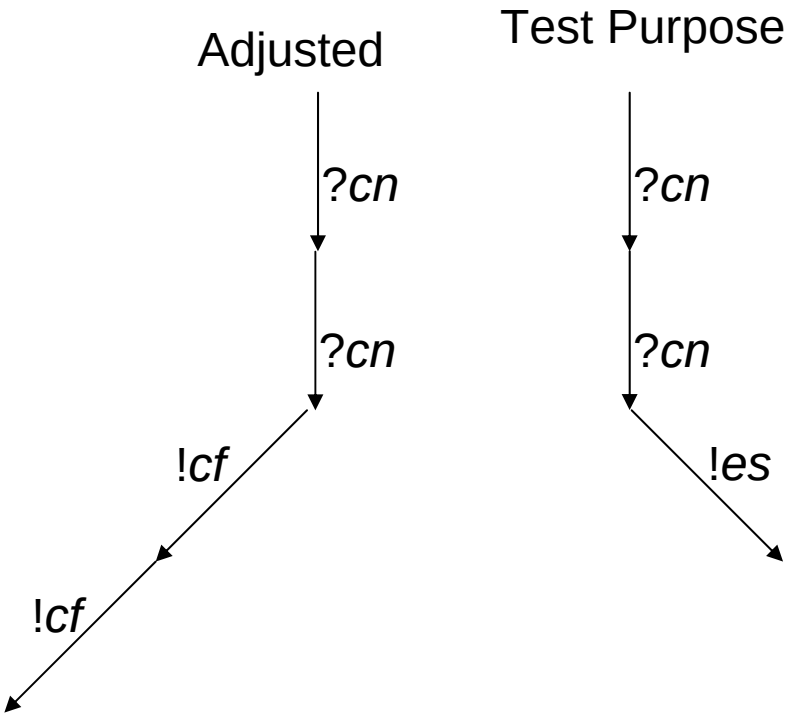
!*cf* = *coffee*
!*es* = *espresso*
?cn = *coin*
δ = *quiescence*



Building Sound Asynchronous Test Case

- Delaying Outputs

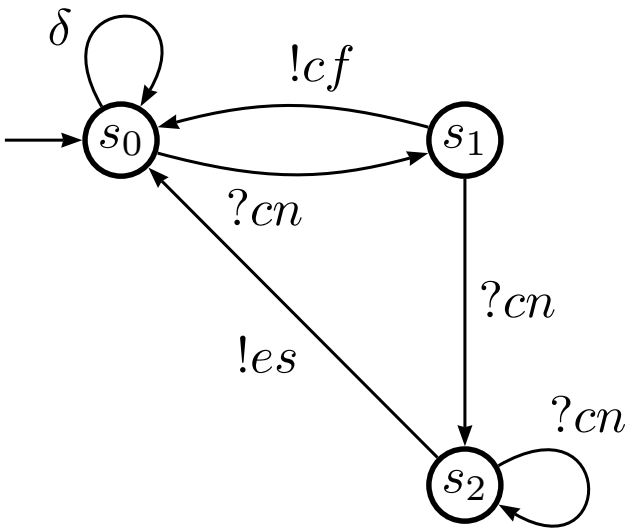


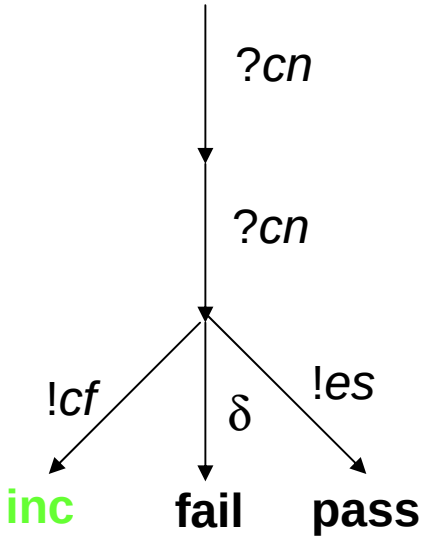


!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Building Sound Asynchronous Test Case

- Test Case Construction (possible solution)

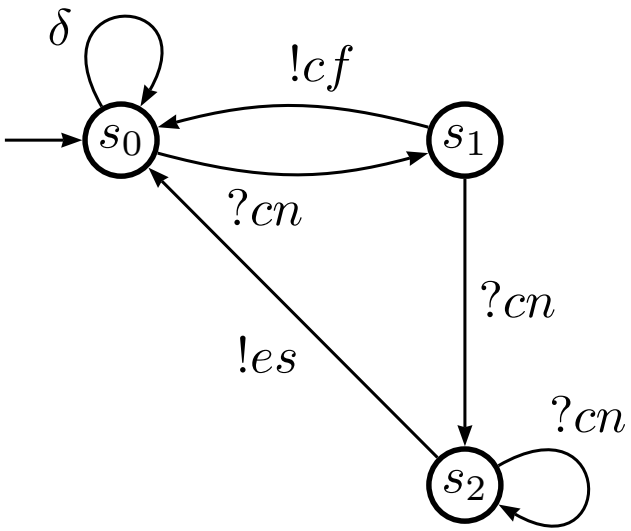


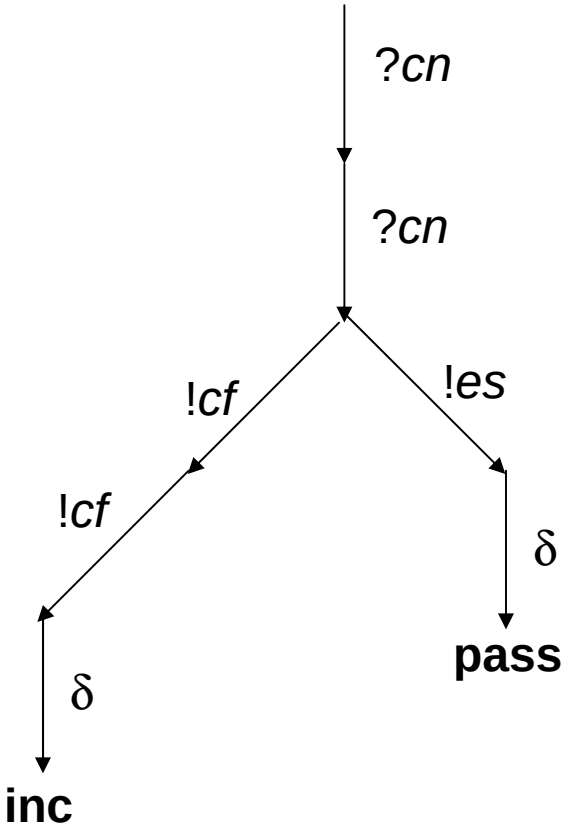


!*cf* = *coffee*
!*es* = *espresso*
?cn = *coin*
δ = *quiescence*

Building Sound Asynchronous Test Case

- Test Case Construction

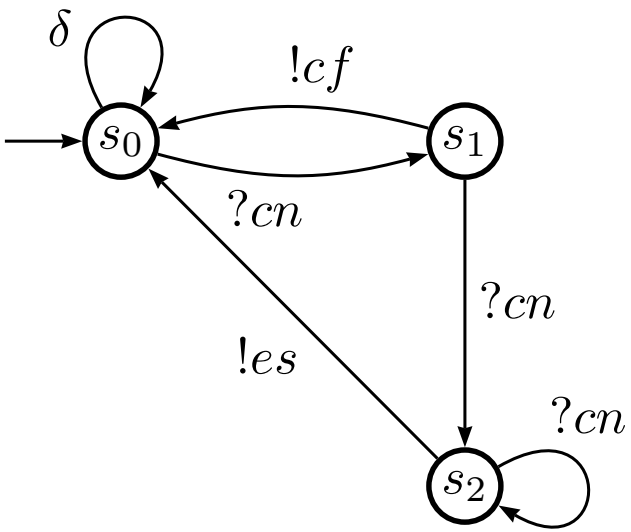


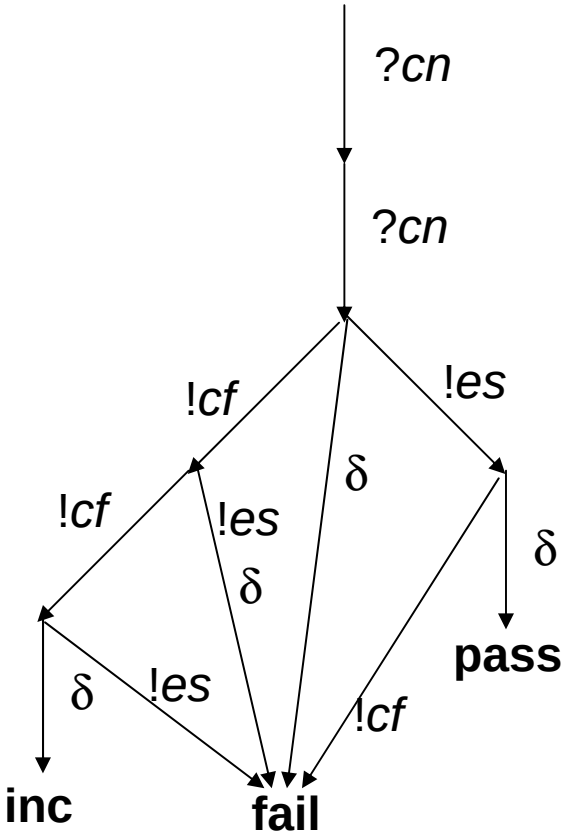


!*cf* = *coffee*
!*es* = *espresso*
?cn = *coin*
δ = *quiescence*

Building Sound Asynchronous Test Case

- Completing with Fail



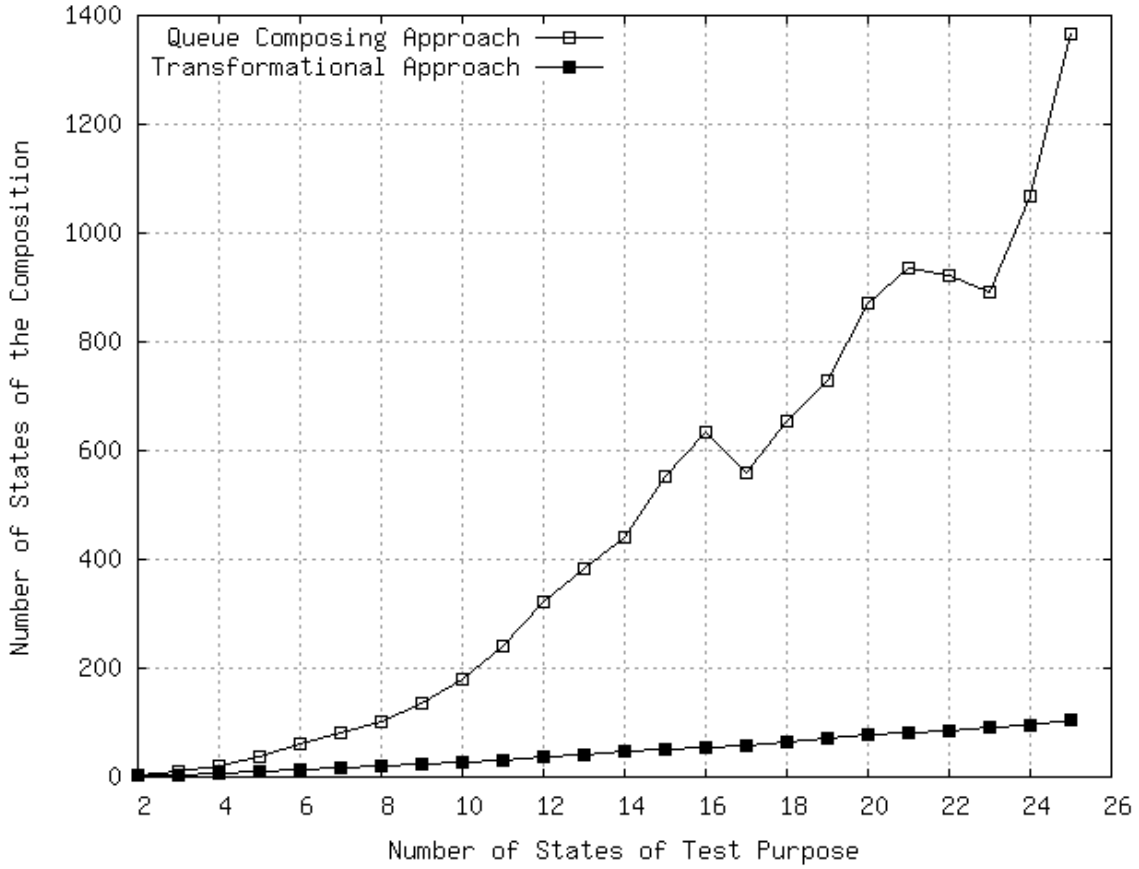


!*cf* = *coffee*
!*es* = *espresso*
?*cn* = *coin*
δ = *quiescence*

Comparing Queue Composing and Transformation Approaches

- Queue Composing Approach
  - Complexity: $\mathbf{O}(n|I|^K|O|^L t)$ states
    - I is the set of inputs
    - O is the set of outputs
    - K is the number of inputs in the test purpose
    - L is the number of outputs in the test purpose
    - t is the number of states in the test purpose
- Transformation Approach
  - Complexity: $\mathbf{O}(n|O|^2 t^2)$ states

Comparing Queue Composing and Transformation Approaches

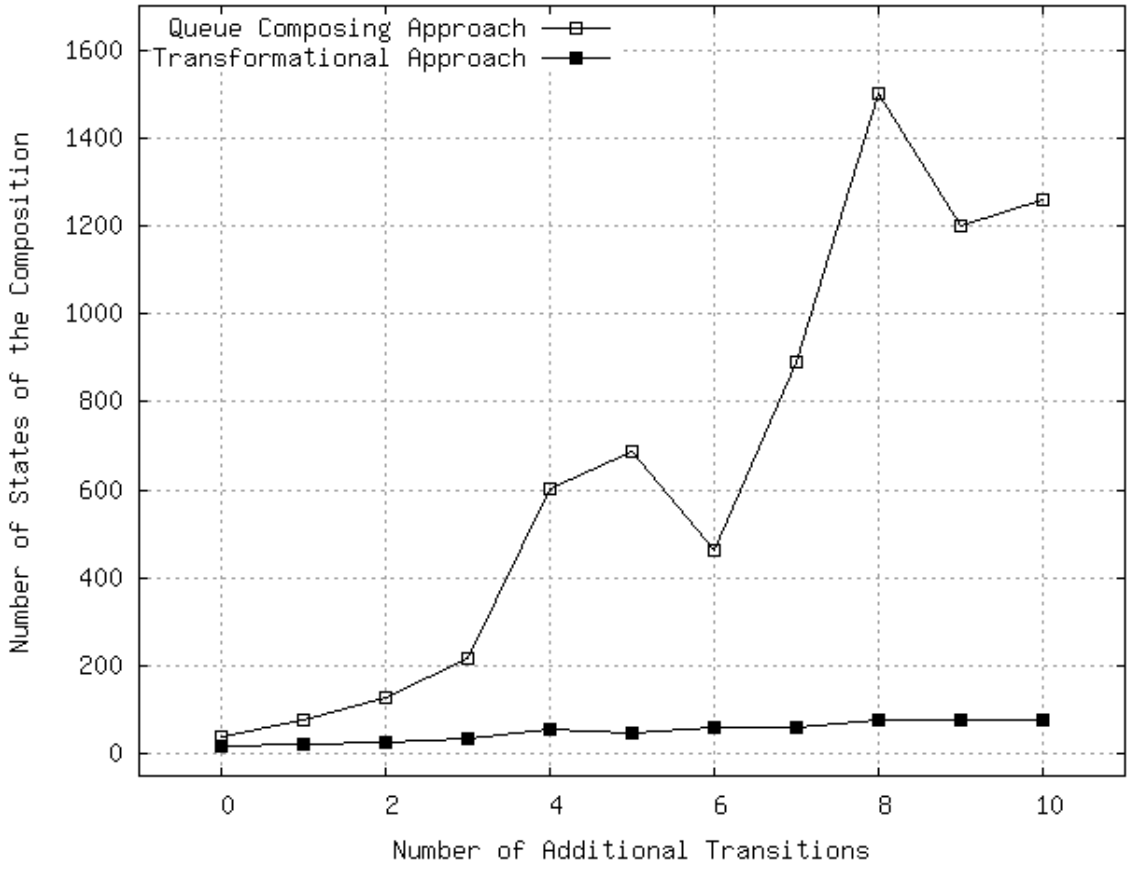


- Fixed specification
- Randomly generated test purposes

Mealy IOTS

- Stable State
  - No output, no internal transition
- If inputs are used only in stable states
  - Then synchronous test cases are sound for asynchronous testing
- Mealy IOTS
  - No input/output conflict

Comparing Queue Composing and Transformation Approaches



- Conference protocol
  - Mealy IOTS
- Adding input transitions in unstable states
  - Creating input/output conflicts

Conclusions

- The proposed solution
  - Relies on test purpose transformation
  - Better scalable, since the queues are not composed with the specification
- The more input/output conflicts in the specification, the bigger the benefits

Future Work

- Experiment with bigger specifications
- Extend the results to distributed testing (multiple queues)

Merci beaucoup!