Probabilities in the TorX test derivation algorithm

L.M.G. Feijs, N.Goga, S.Mauw

Eindhoven University of Technology

Overview of the presentation

- general context
- the CdR project and the TorX algorithm
- motivation of our research
- automatic test generation
- description of the TorX algorithm
- extension of TorX with probabilities
- computation of the probabilities
- an example of TorX with probabilities
- conclusions

General context

- More interest in the development of theoretical foundations for testing and tool support (SDL, MSC ...)
- Manual derivation of tests - sub-optimal testing process
- Automatic test generation

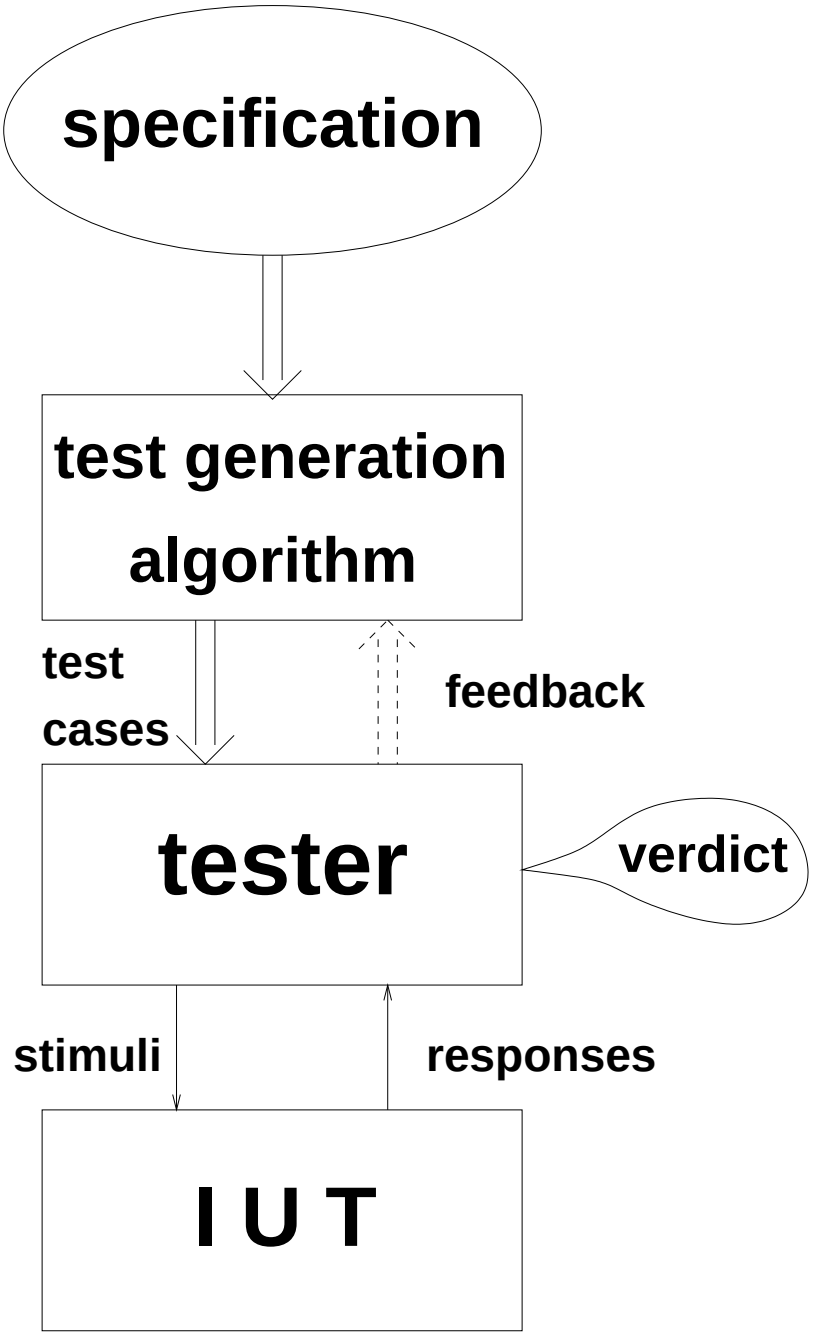
The CdR project and the TorX algorithm

- CdR (Côtes-de-Resystes): project founded by a consortium of dutch research groups from academia and industry
- CdR: formalization and automatization of the testing practice
- TorX: the CdR prototype tool for automatic test generation
- TorX: easy integration with other tools: (SDT (SDL, MSC), Lotos tools, Verilog tools)
- TorX: ioco theory (Twente University)

Motivation of our research

- TorX - should choose when a system can do an input and an output (equal distribution of choices - probabilities)
- In some cases (elevator) TorX did not produce interesting test cases
- Experiences using SDT
- Natural solution: explicit probabilities

Automatic test generation



TorX -test generation algorithm

Choice 1:

(*terminate the test case*)

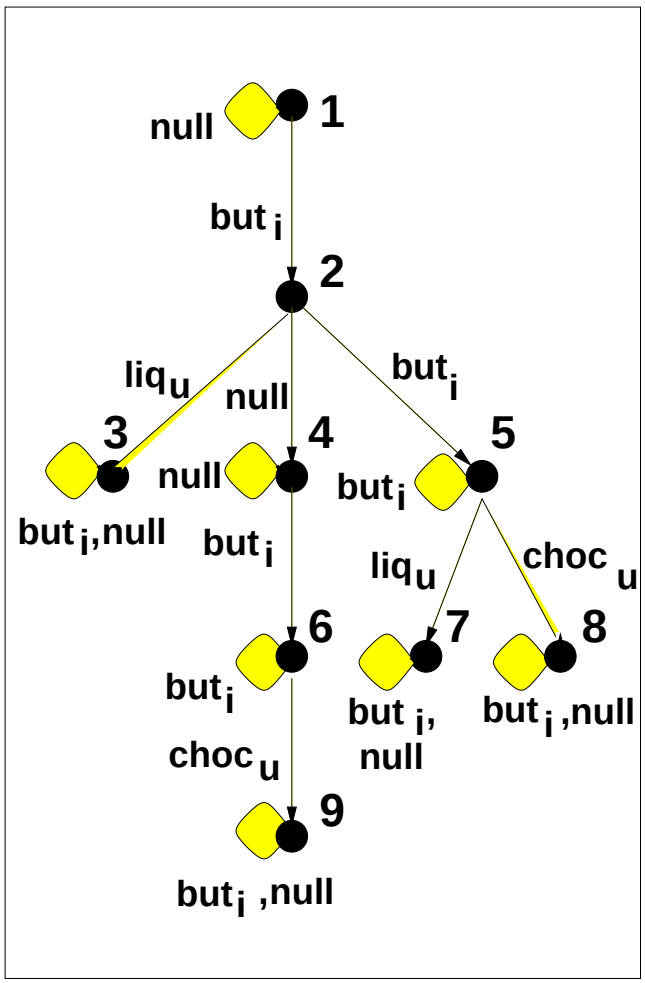
Choice 2:

(*supply an input for the implementation*)

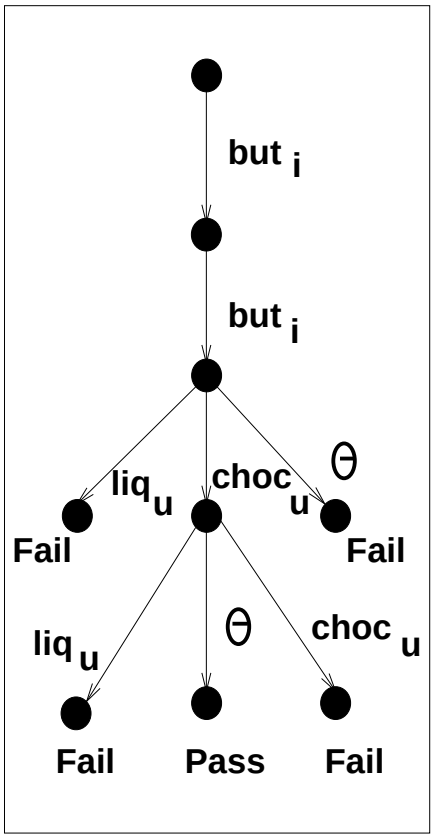
Choice 3:

(* check the next output of the implementation*)

An example of test derivation with TorX



The suspension automaton for a candy machine



The test generated for the candy machine

TorX with probabilities

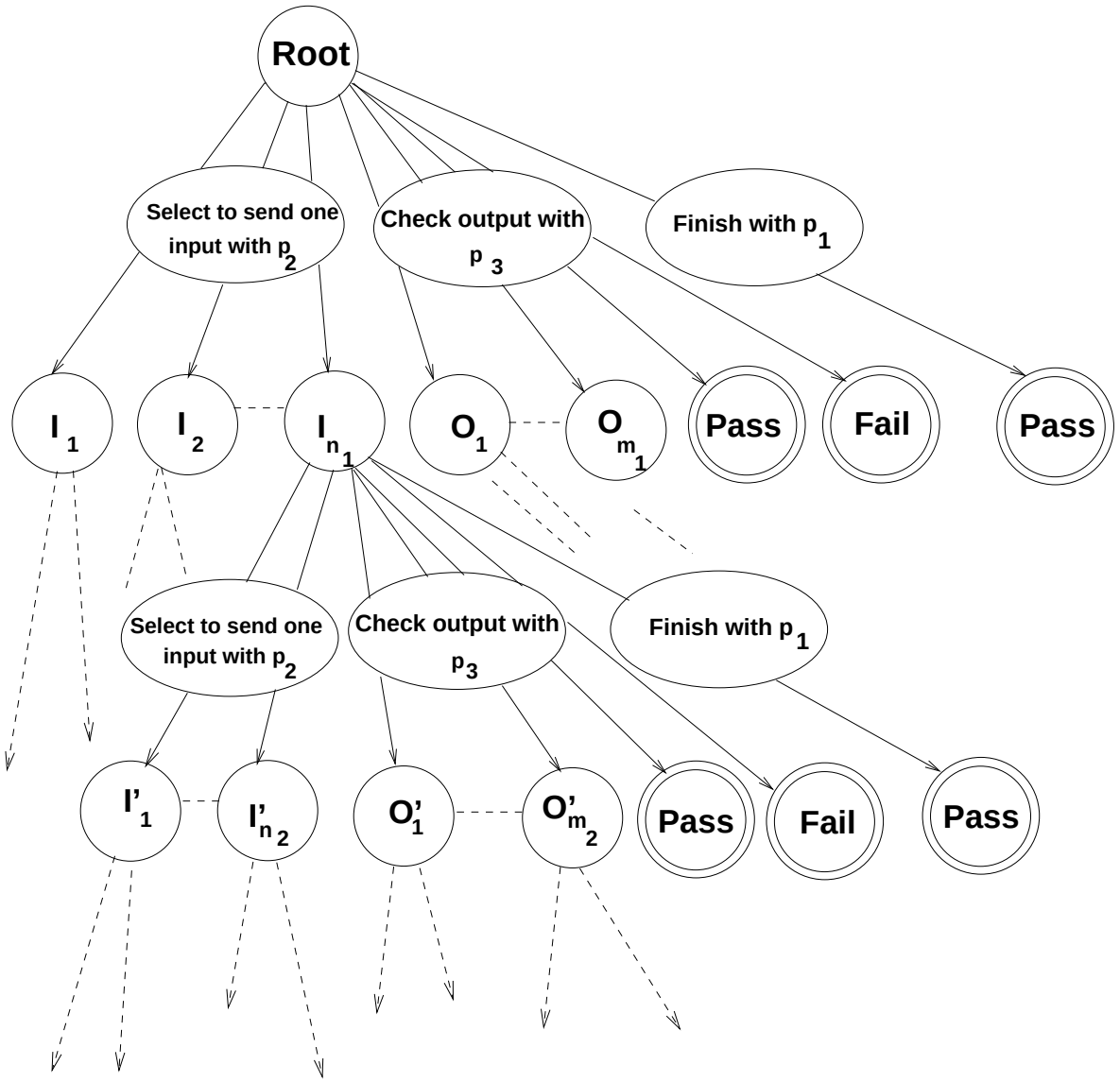
Motivation of modifying the input-ouput ratio

* Elevator case
* Experience with SDT tool set

The modification of TorX

* Choose Choice 1
  with probability p_1
* Choose Choice 2
  with probability p_2
* Choose Choice 3
  with probability p_3

The behaviour tree

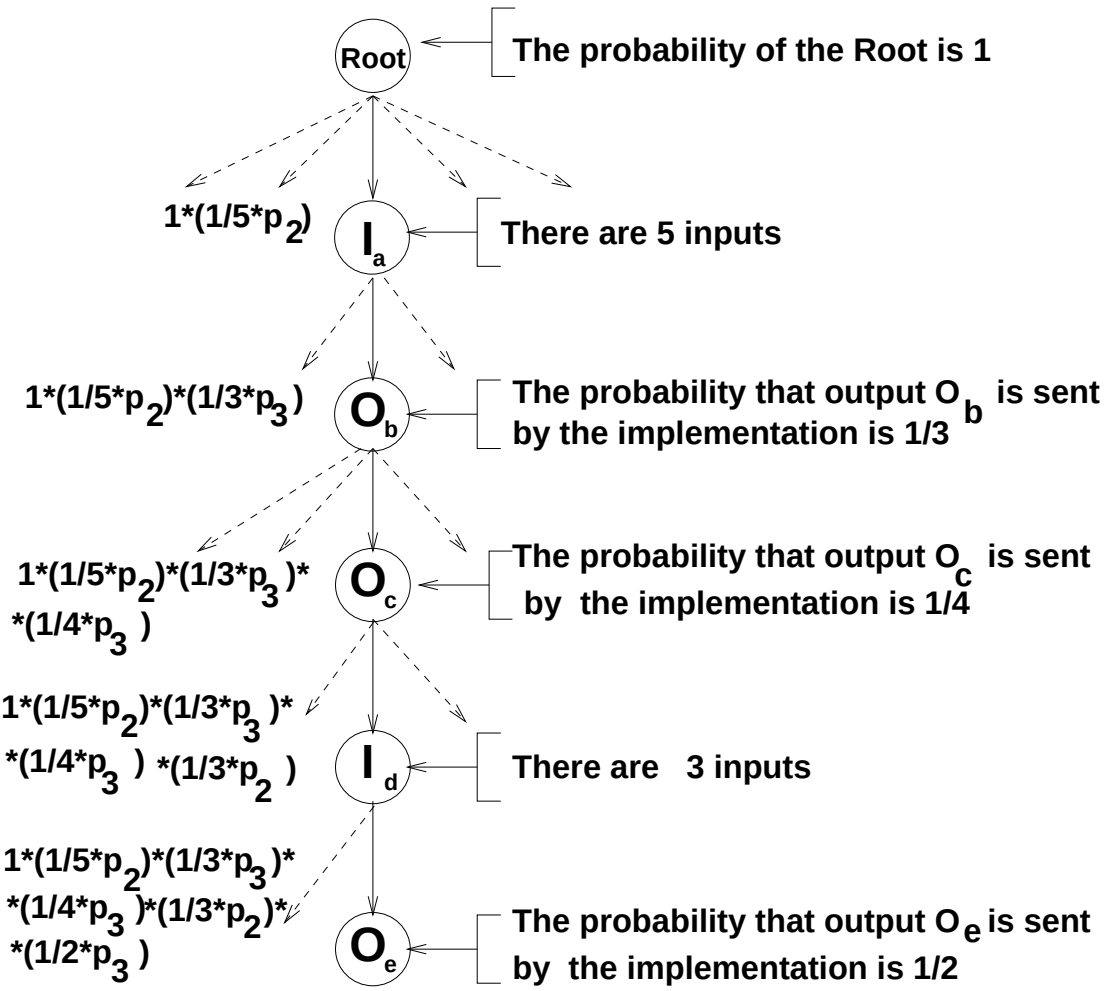


The probability to generate a given trace

$$P = C * p_2^{\,n} * p_3^{\,m}$$

n=nr. inputs on the trace

m=nr. outputs on the trace



The optimization of the global probabilities

I: r=n/m

n=nr. inputs on the trace

m=nr. outputs on the trace

O: p_2 =?, p_3 =?

$$P = C * p_2^{\,n} * p_3^{\,m}$$

$$\Downarrow$$

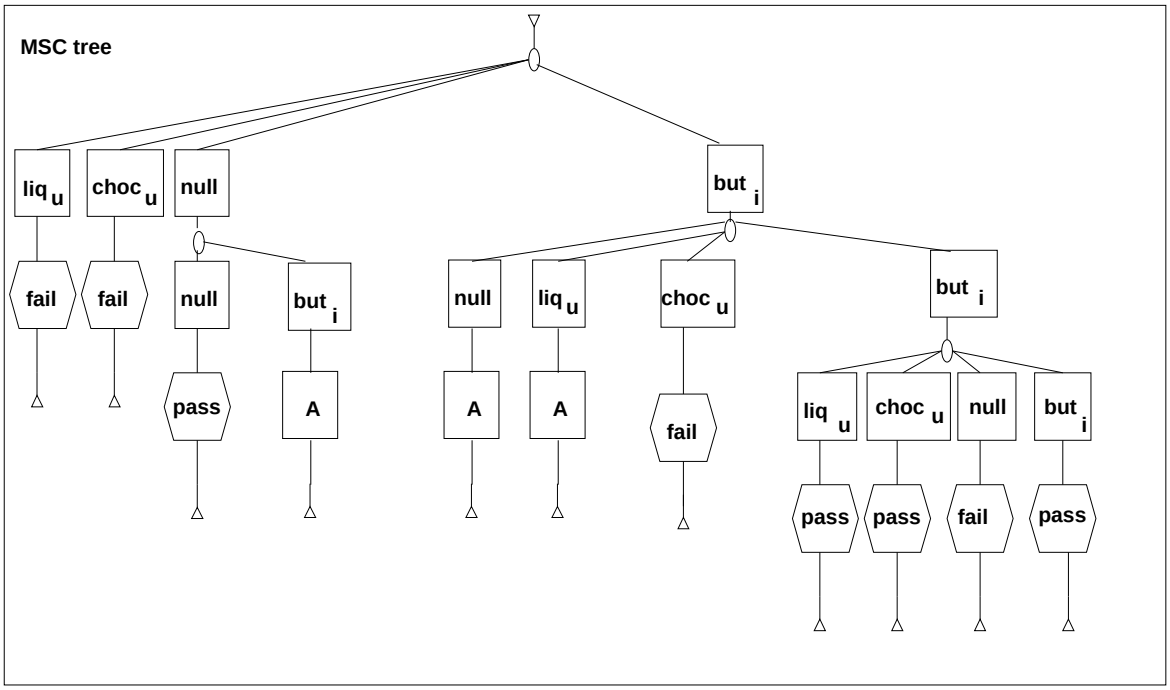
$$dP = 0$$

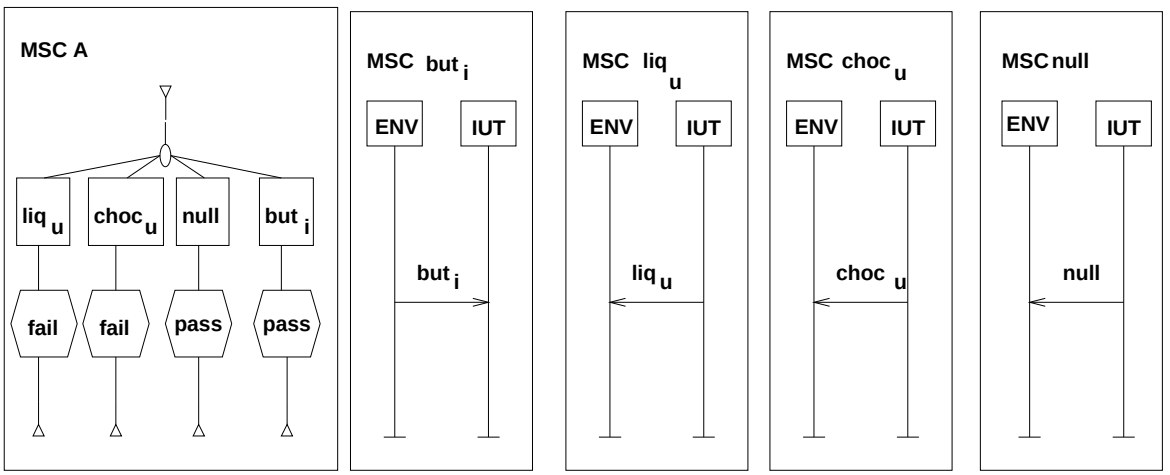
$$\Downarrow$$

$$p_2 = (r/(r+1)) * (1-p_1)$$

$$p_3 = (1/(r+1)) * (1-p_1)$$

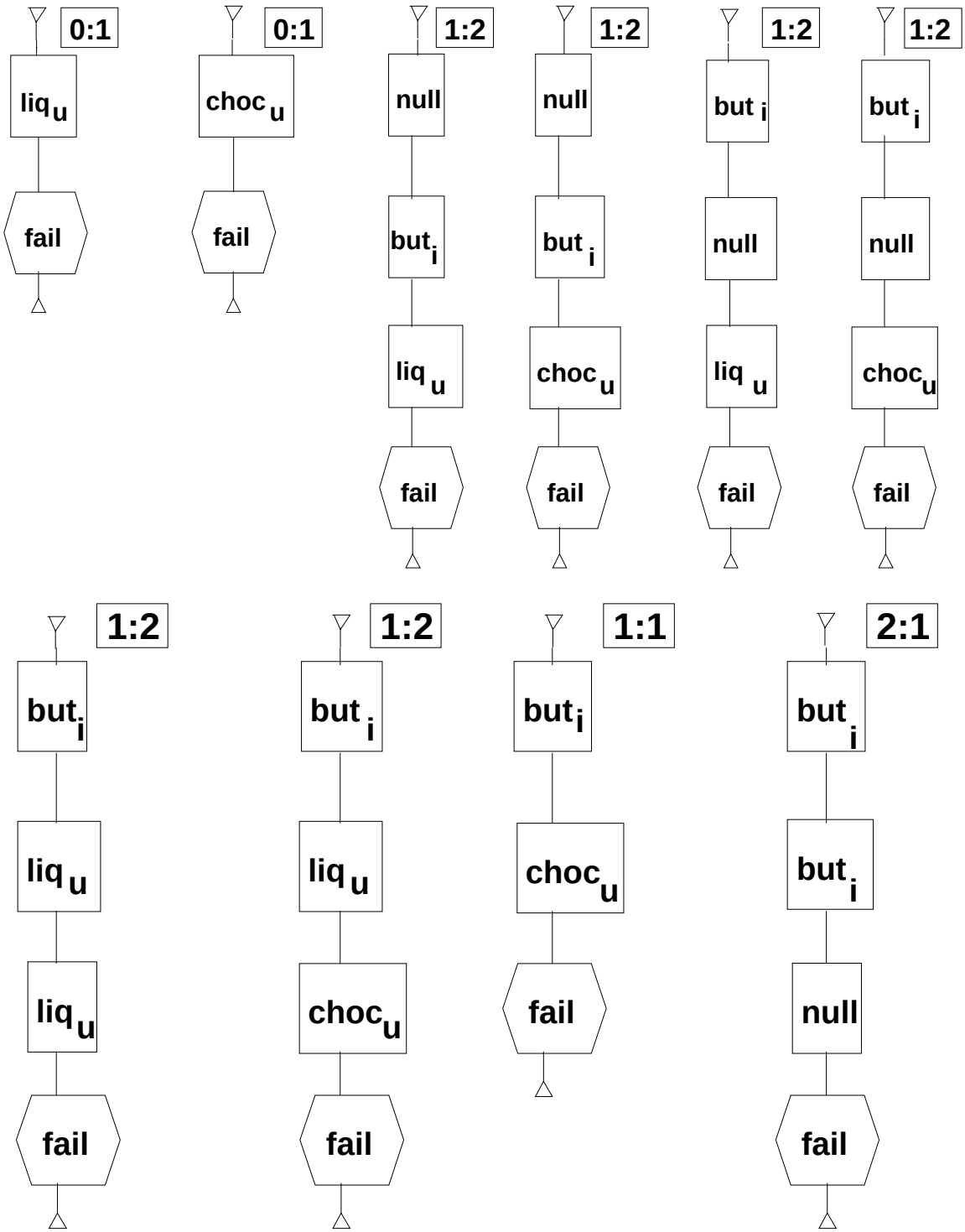
An application of TorX modified with probabilities





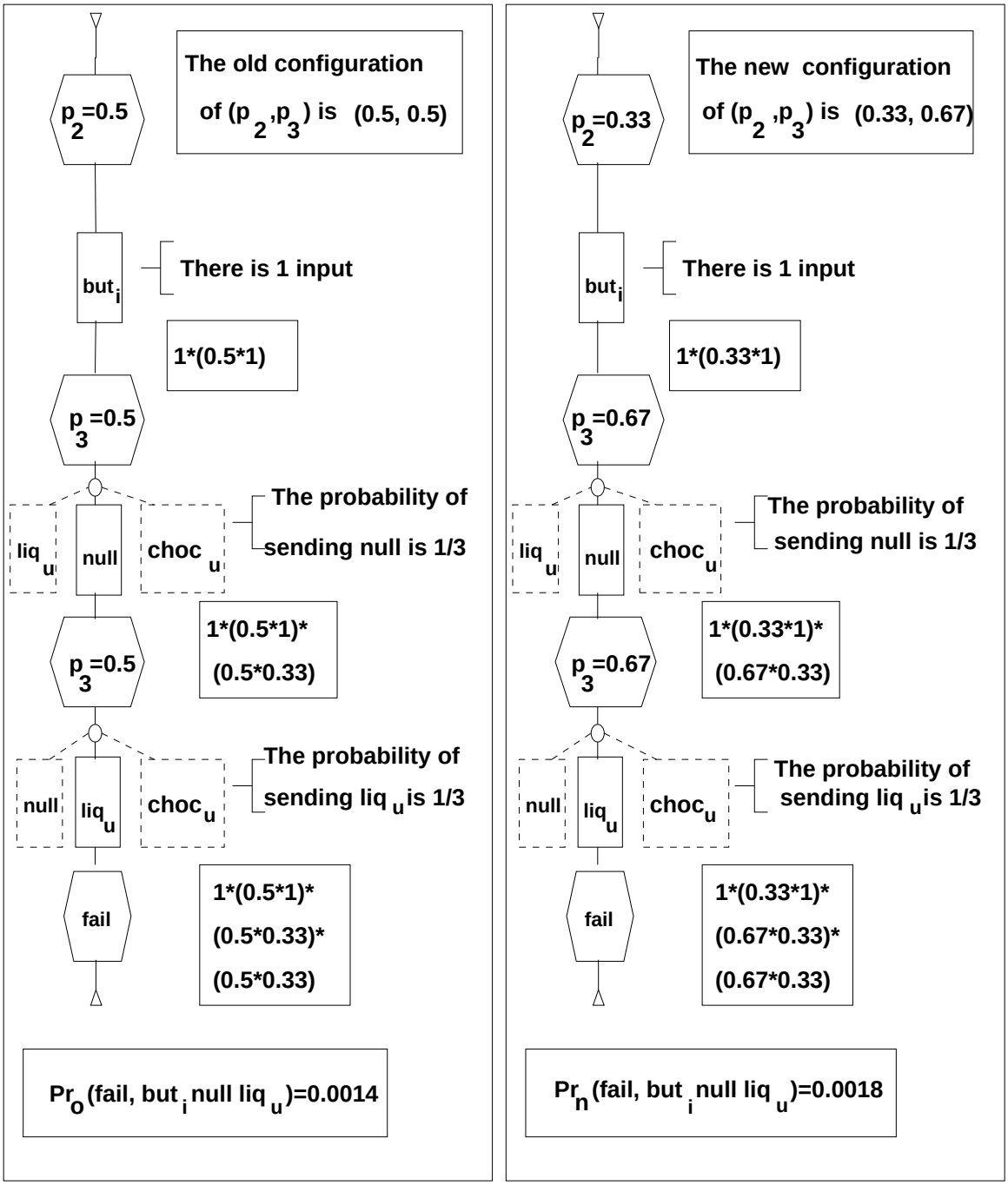
Tests derived from candy machine

Fail traces represented in HMSC

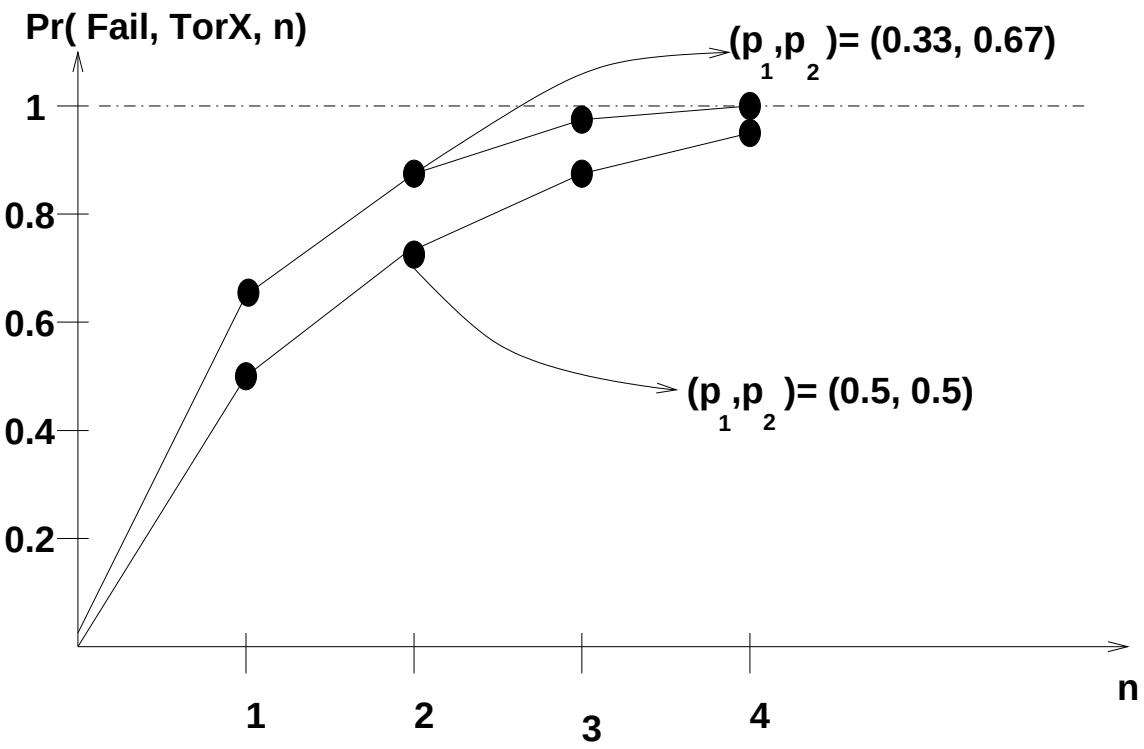


The probability of generating the trace

but_i null liq_u



The probability of getting a Fail in function of the number of tests generated



$$Pr(\text{Fail}, \text{TorX}, n) = 1-(1-Pr(\text{Fail}, \text{TorX}, 1))^{n}$$

Conclusions

* TorX extended with probabilities

* In some cases the new algorithm generates better test cases

Further work

* Probabilistic coverage