Constructing Formal Models through Automata Learning

Fides Aarts, Faranak Heidarian, Frits Vaandrager

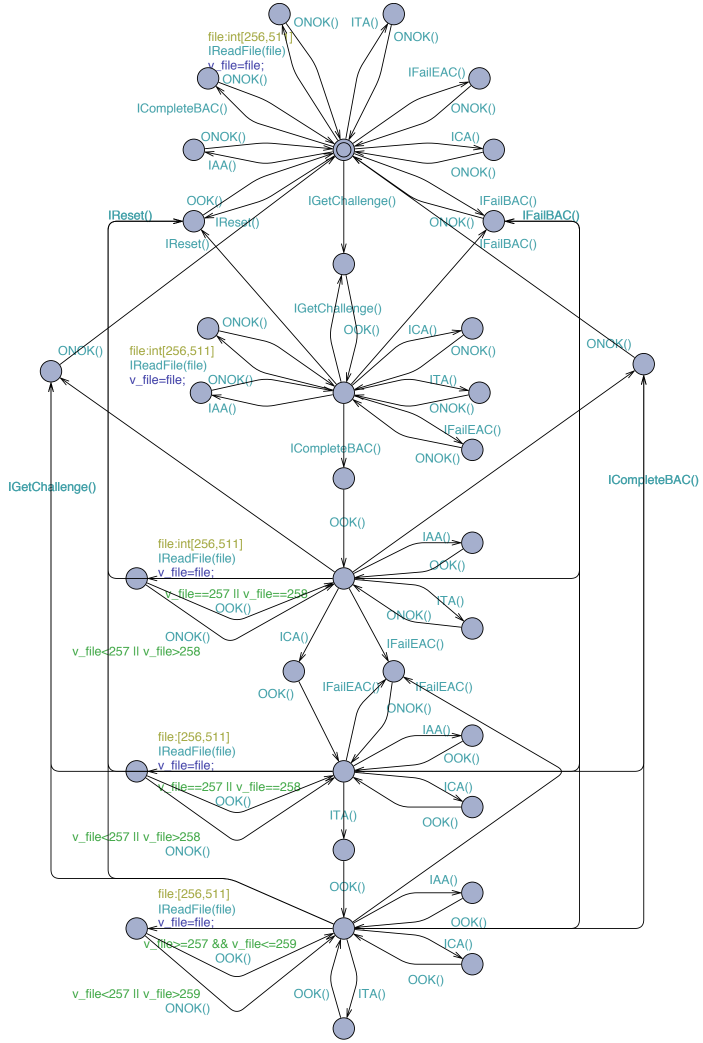
Outline

- Introduction
  - History
  - Motivation
- Idea
- Method
- Example
- Future Work

Introduction

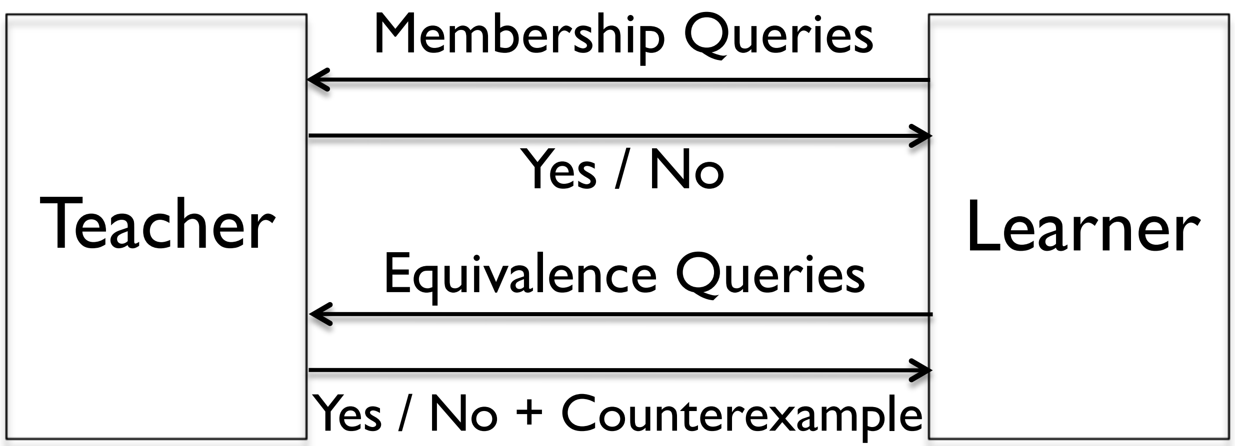
Introduction

Biometric Passport

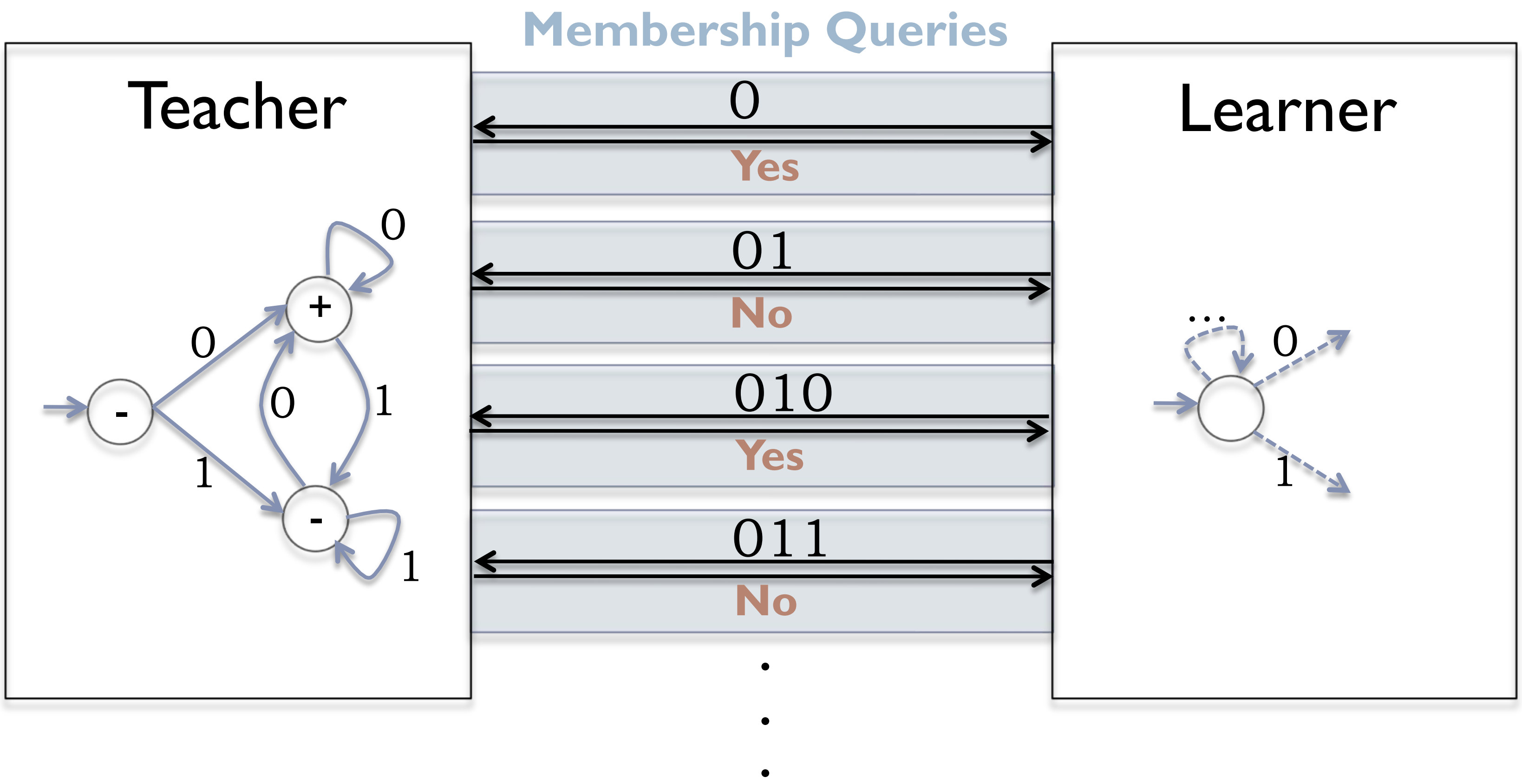


History

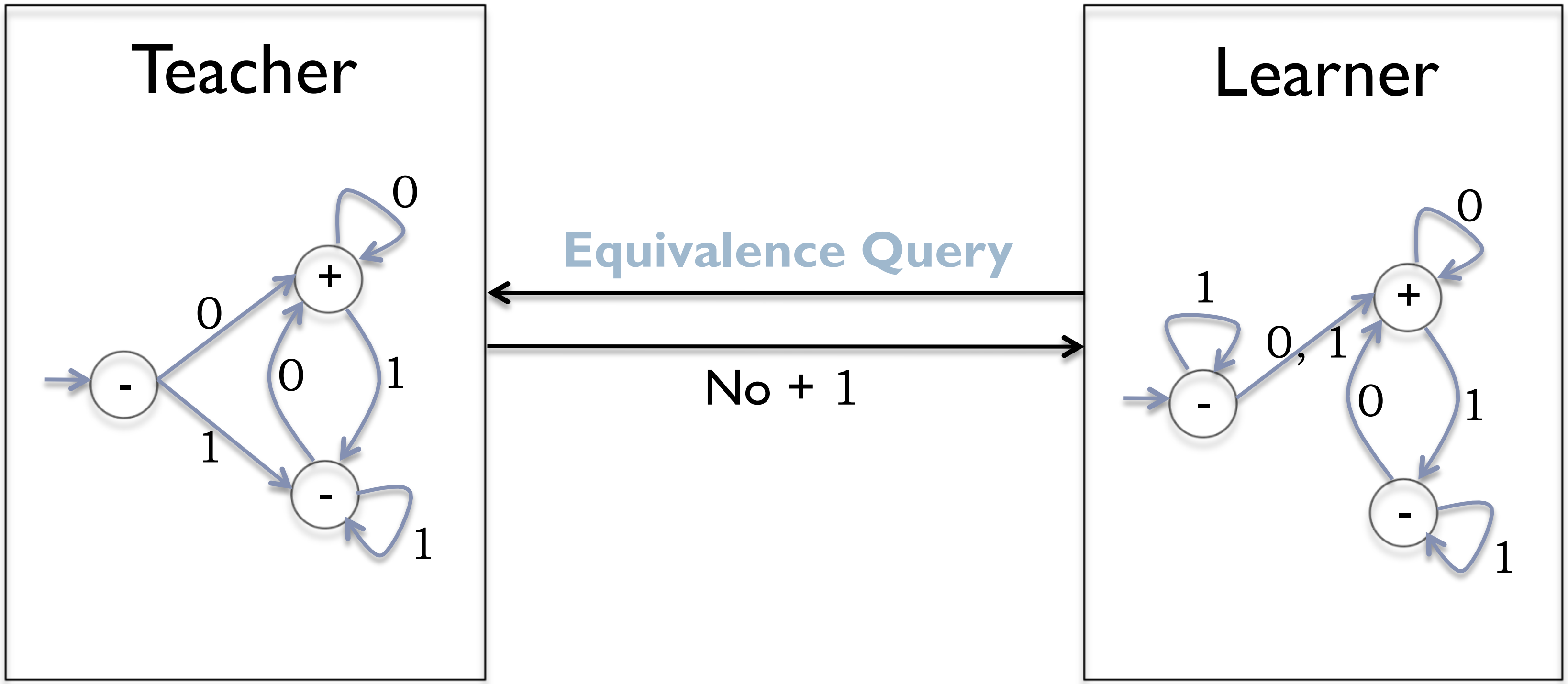
- 1987: Angluin's L* Algorithm for Learning DFA
  - Teacher knows a DFA
  - Learner knows the alphabet



Example

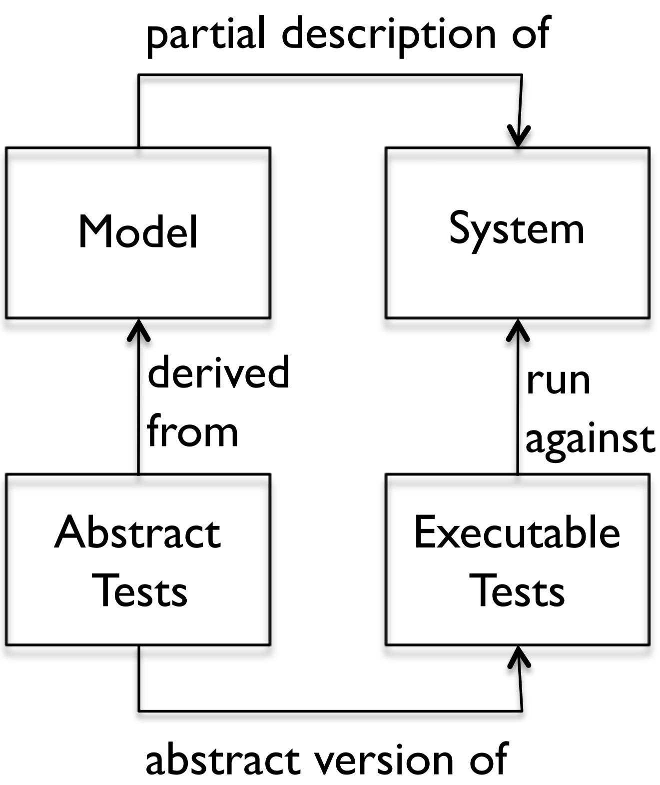

Membership Queries
Teacher
Learner
0
Yes
01
No
010
Yes
011
No
+
-
-
0
0
0
1
1
1
...
0
1

Example



Motivation

Model-based Testing



Test-based Modeling

- build models that allow effective testing
- design for testability

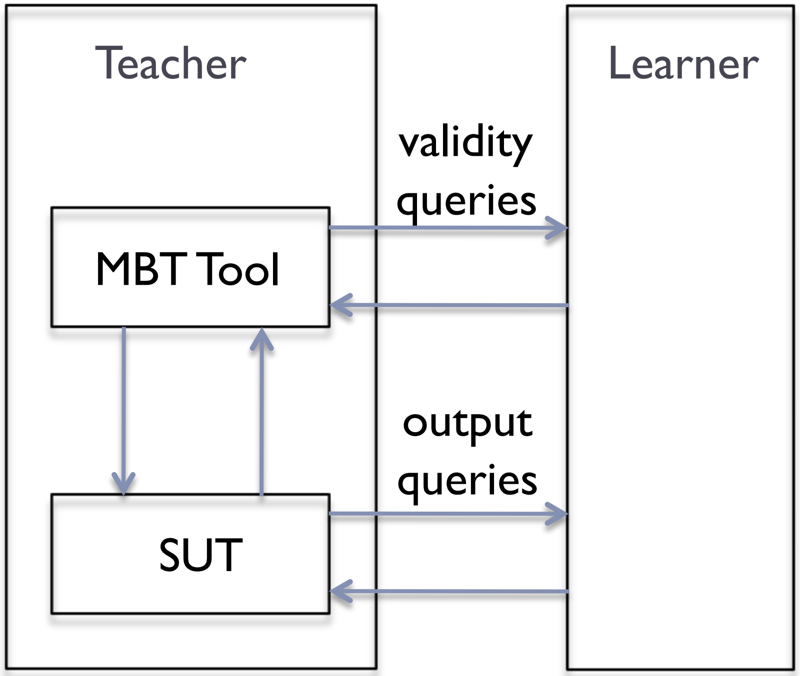
Motivation

Regression Testing

- Fix the bugs and make sure the problems are solved!

Motivation

Active Learning of Reactive Systems

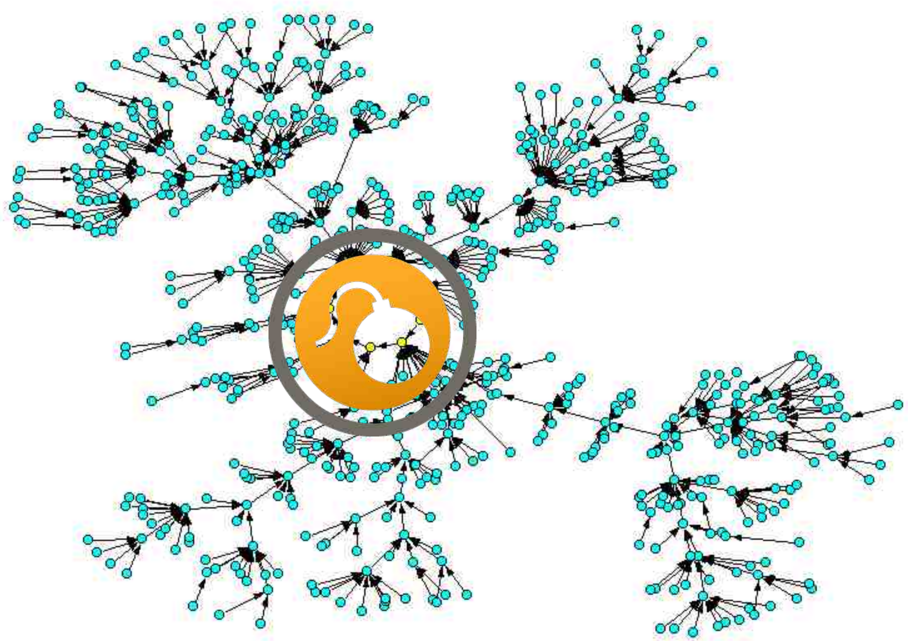


LearnLib Tool

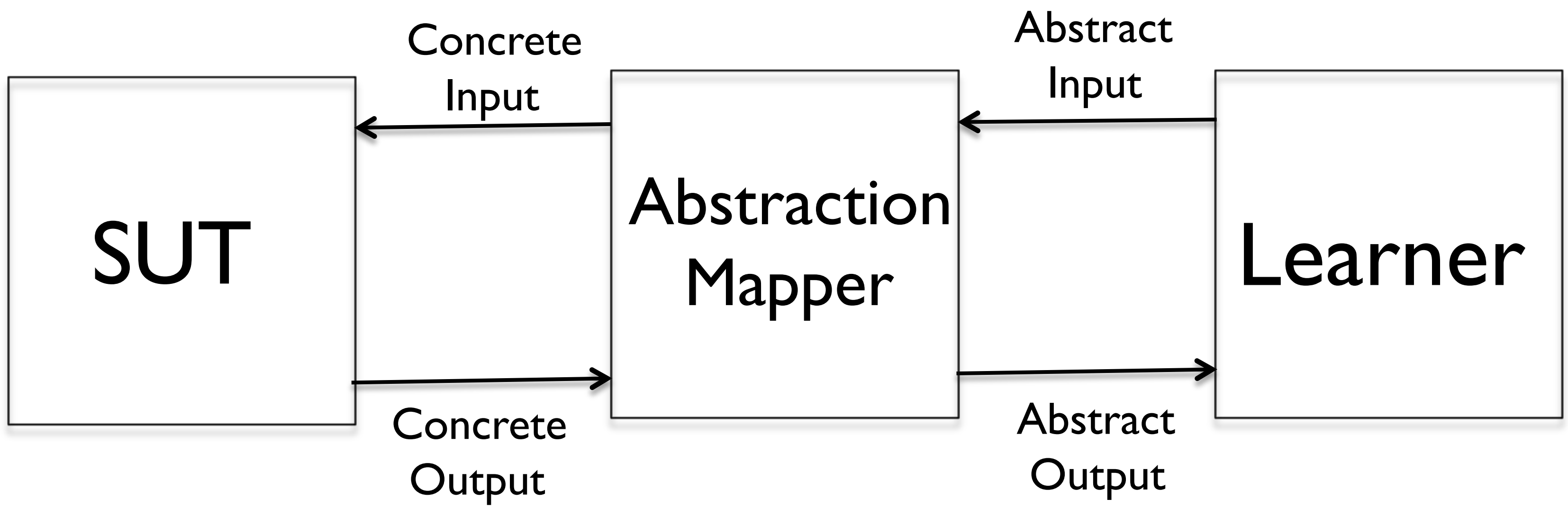
- Learns deterministic Mealy Machines
- learns state machines with up to 30000 states

Challenge

- Learning parametric systems
  - $Act(p_1, \ldots, p_n)$, $p_i \in N$
    - $Act(p_1, p_2)$
      - Act(1,1), Act(1,2), Act(2,1), Act(1,3), Act(2,2), Act(3,1), Act(1,4), …
- State space explosion



Idea



SUT
Concrete
Input
Concrete
Output
Abstraction
Mapper
Abstract
Input
Abstract
Output
Learner

Related Work

- Falk Howar, Bernhard Steffen and Maik Merten, *Automata Learning with Automated Alphabet Abstraction Refinement*, 2011
- Therese Berg, Bengt Jonsson and Harald Raffelt, *Regular Inference for State Machines Using Domains with Equality Tests*, 2008
- Therese Berg, Bengt Jonsson and Harald Raffelt, *Regular Inference for State Machines with Parameters*, 2006

Restrictions on SUT's

Actions

- Input Actions
  - $IN(p_1, \ldots, p_n)$
  - $v_i := p_j$
- Output Actions
  - $OUT(q_1, \ldots, q_m)$

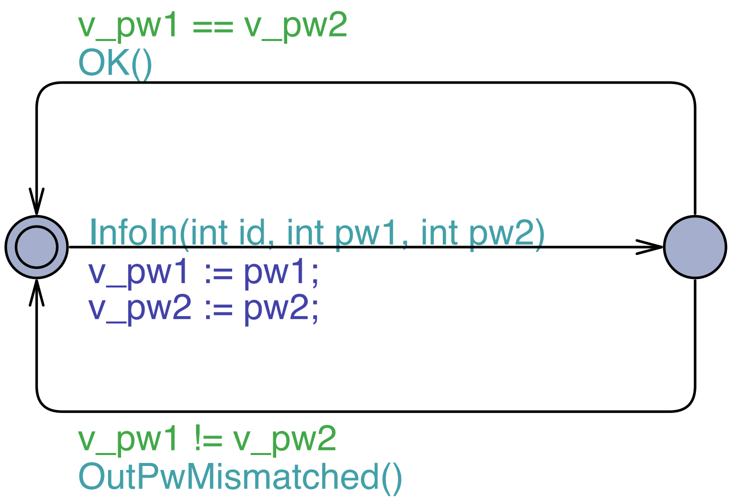
Guards

- Conjunctions of equalities and inequalities
  - Example:

$$(p_i = p_j \wedge v_k \neq p_\ell \wedge v_u = c_\upsilon)$$

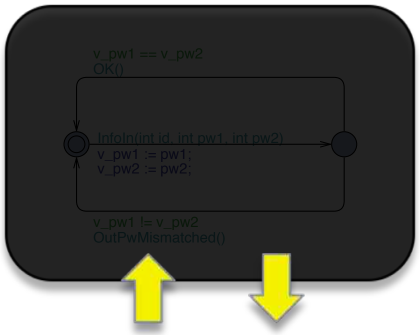
Scalarset Symbolic Interface Automata

Registration System



Registration System

SUT Model



- Input Actions
  - InfoIn(id, pw1, pw2)
- OutputActions
  - OK()
  - OutPwMismatched()

Mapper

- State Variables
  - id_f, id_ℓ
  - pw1_f, pw2_ℓ
  - pw2_f, pw2_ℓ
- Abstraction Table

	0	1	2	…
id				
pw1				
pw2				

Registration System

Traces

InfoIn	id	pw1	pw2	OK
	1	2	2	

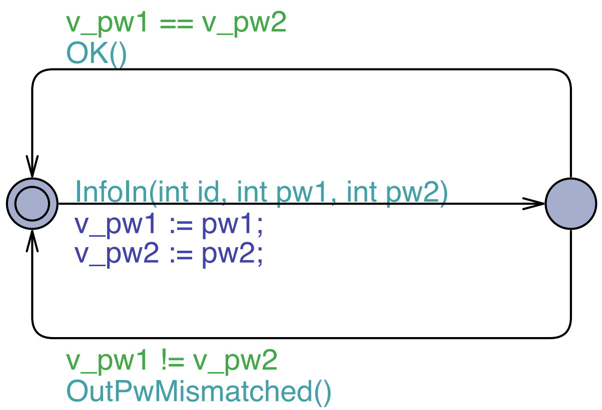
InfoIn	id	pw1	pw2	OutPwMismatched
	1	2	3	

Abstraction Table

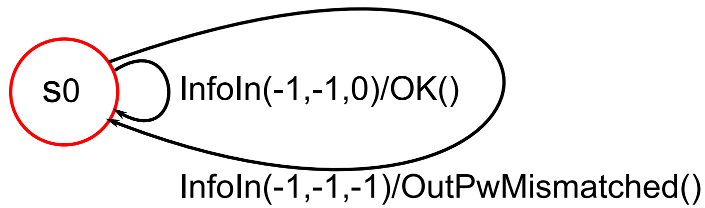
	0	1	2
id			
pw1			
pw2	pw1_f		

Registration System

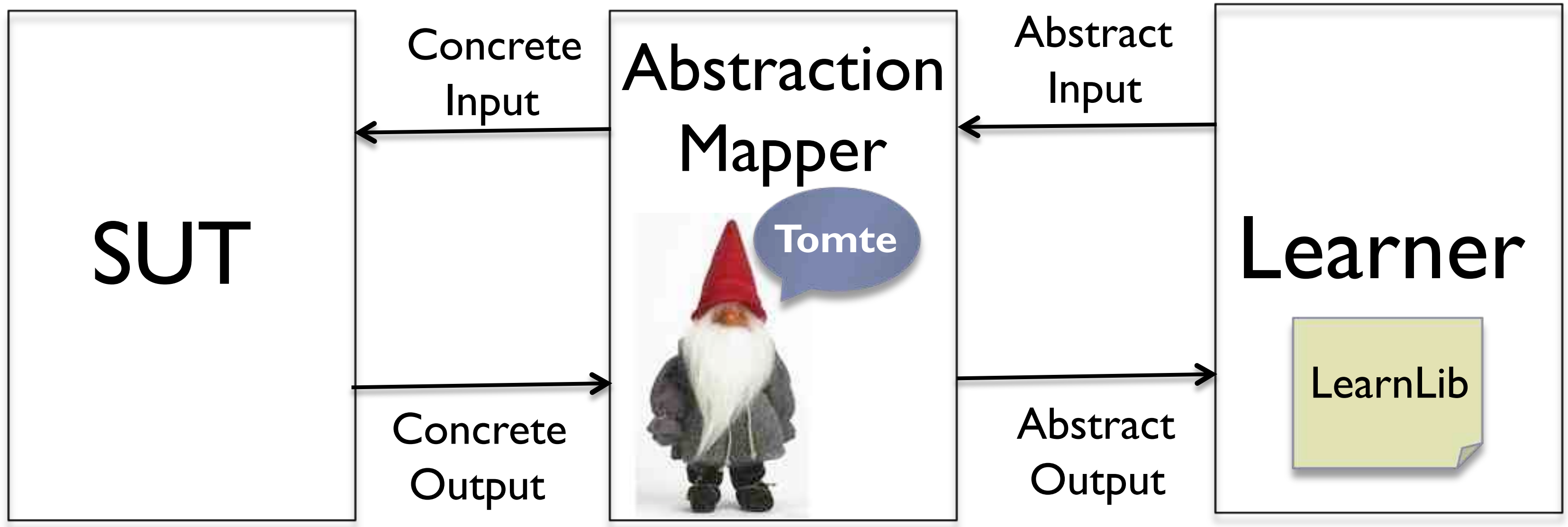
SUT Model



Learned Model



Implementation



Theory

- Under certain conditions,

$$T \ \textbf{ioco} \ A \,||\, H \Rightarrow T \,||\, A \ \textbf{ioco} \ H$$

- T: SUT
- A: Mapper
- H: Learned Automaton

Experiments Results

System Under Test	Input Refinements	States	Learning/ Testing Queries	Learning/ Testing Time (seconds)
Alternating Bit Protocol - Sender	1	7	193/3001	1.3s/104.9s
Alternating Bit Protocol - Receiver	2	4	145/3002	0.9s/134.5s
Alternating Bit Protocol - Channel	0	2	31/3000	0.3s/107.5s
Biometric Passport	3	5	2199/3582	7.7s/94.5s
Session Initiation Protocol	3	13	1755/3402	8.3s/35.9s
Login System	3	5	639/3063	2.0s/56.8s
Farmer-Wolf-Goat-Cabbage Puzzle	4	10	699/3467	4.4s/121.8s
Palindrome/Repdigit Checker	11	1	3461/3293	10.3s/256.4s

Future Work

- ITALIA project: http://www.italia.cs.ru.nl/
- Automatically learning the state variables the mapper needs to store
- Extending the memory
- Extending the class of SUT's that we can handle (in particular operations on data)
- More case studies

Thank You!

Questions or Suggestions?

faranak@cs.ru.nl