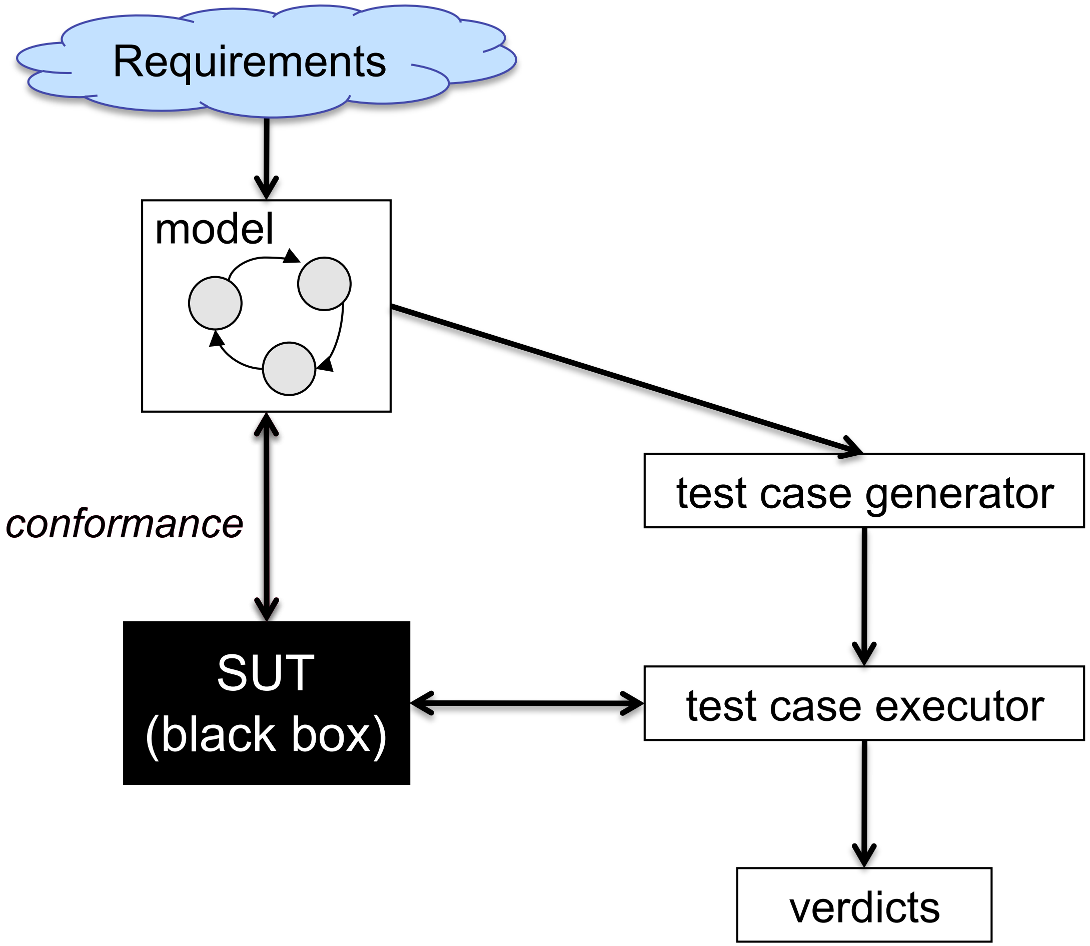
Towards Symbolic Model-Based Mutation Testing: Combining Reachability and Refinement Checking

Bernhard K. Aichernig, Elisabeth Jöbstl

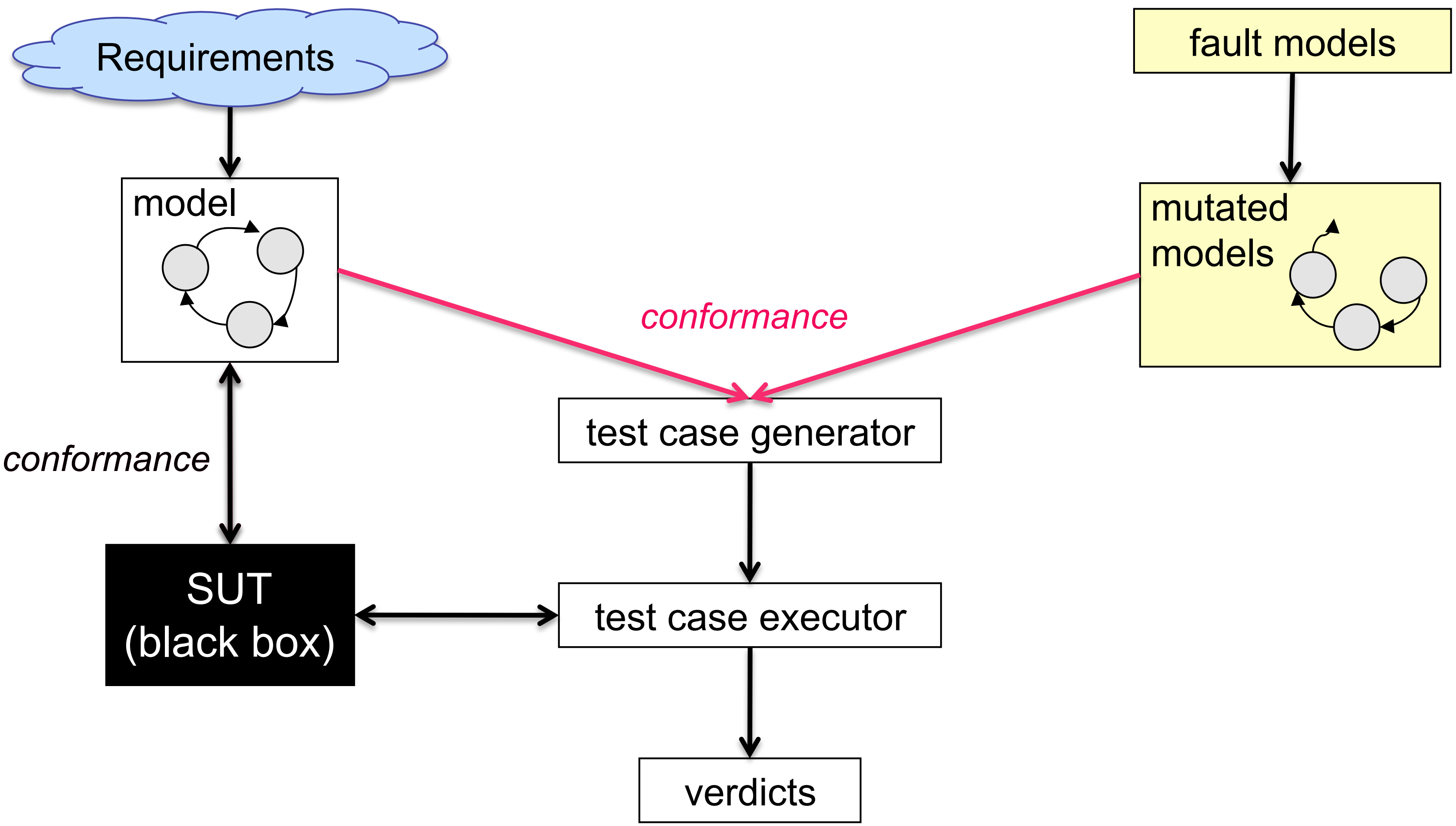
Institute for Software Technology
Graz University of Technology
ejoebstl@ist.tugraz.at
www.ist.tugraz.at

Trufal
Trust via Failed Falsification

Model-Based Testing



Model-Based **Mutation** Testing



Outlook

- Action Systems
- Conformance
- Refinement Checking
- Experimental Results
- Future Work & Conclusions

Action Systems

- Ralph-Johan Back
- Guarded commands
- Reactive systems
- Non-determinism

```
type(int, X) :- X in 0..10.
var([v1, v2], int).
state_def([v1, v2]).
init([0,0]).

as :-
  actions (
    A1 :: g1 => v1 := e1 ,
    ... ,
    Am(X) :: gm => (g => (v1 := X ; v2 := e2))
  ),
  dood (
     A1 [] ... [] Am
  ).
```

Semantics of Actions

- Predicative Semantics
- UTP (Unifying Theories of Programming)

$$l :: g => B \quad =_{df} \quad g \;\wedge\; B \;\wedge\; tr' = tr ^\frown [l]$$

$$l(\overline{X}) :: g => B \quad =_{df} \quad \exists\, \overline{X} : g \;\wedge\; B \;\wedge\; tr' = tr ^\frown [l(\overline{X})]$$

$$x := e \quad =_{df} \quad x' = e \;\wedge\; y' = y \;\wedge \ldots \wedge\; z' = z$$

$$g => B \quad =_{df} \quad g \;\wedge\; B$$

$$B(\overline{v}, \overline{v}'); B(\overline{v}, \overline{v}') \quad =_{df} \quad \exists\, \overline{v_0} : B(\overline{v}, \overline{v_0}) \;\wedge\; B(\overline{v_0}, \overline{v}')$$

$$B \;[]\; B \quad =_{df} \quad B \;\vee\; B$$

Conformance

- UTP‘s Refinement:

$$M \sqsubseteq I \;=_{df}\; \forall x, x' y, y', \dots \in \alpha \;:\; I \Rightarrow M \qquad \textit{for all } M, I \textit{ with alphabet } \alpha.$$

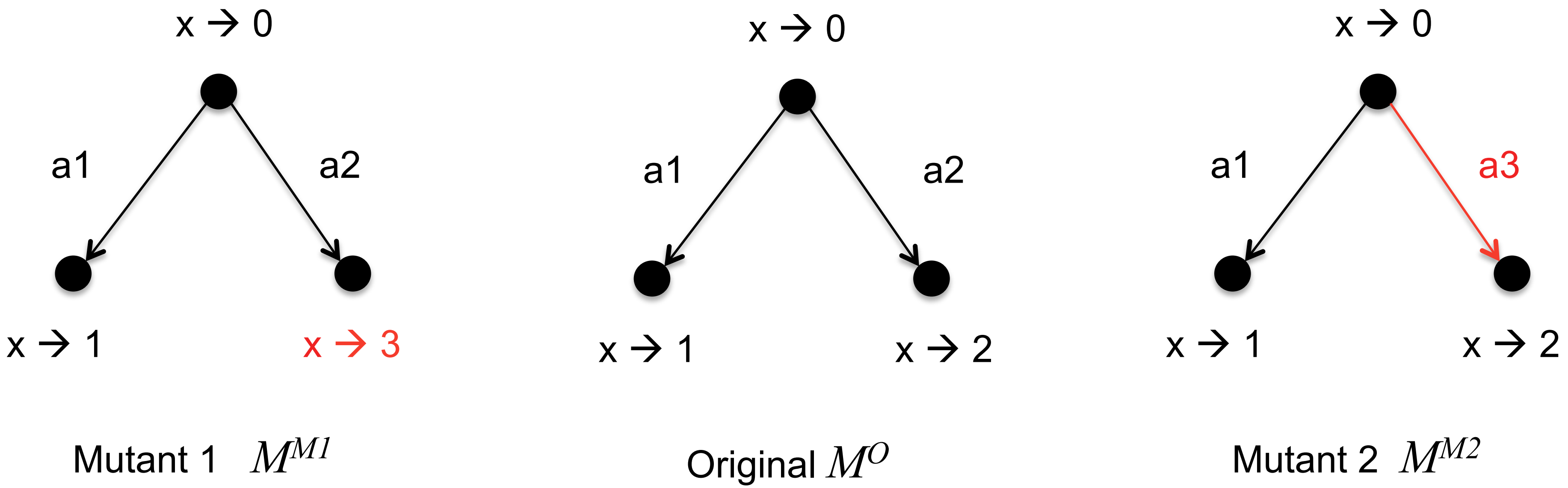
- We want to find counterexamples for refinement, i.e., cases where $M^O \not\sqsubseteq M^M$

$$\exists x, x', y, y', \dots \in \alpha \;:\; M^M \wedge \neg M^O$$

- Unsafe state:

$$u \in \{s \mid \exists\, s' : M^M(s, s') \wedge \neg M^O(s, s')\}$$

Example: Non-Refinement



$$\exists x, x', y, y', \dots \in \alpha \ : \ M^{M} \wedge \neg M^{O}$$

(Non-)Refinement of Action Systems

- Consider reachability:

Is the unsafe state reachable from the initial state?

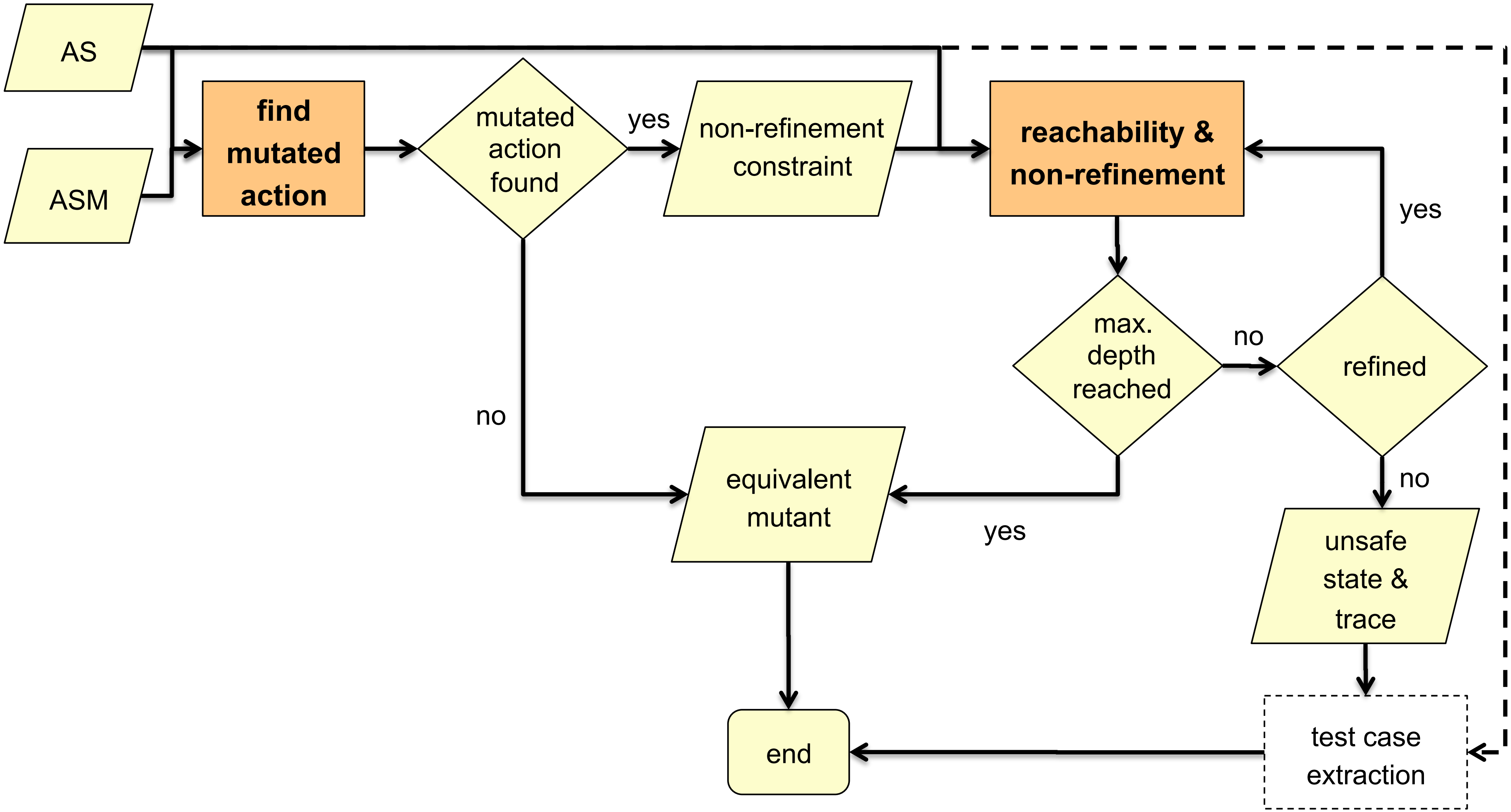
- Non-refinement:

$$\exists \overline{v}, \overline{v}', tr, tr' \ : \ (\overline{v} \in reachable(AS^O, tr) \wedge (A_1^M \vee \cdots \vee A_n^M) \wedge \neg A_1^O \wedge \cdots \wedge \neg A_m^O)$$

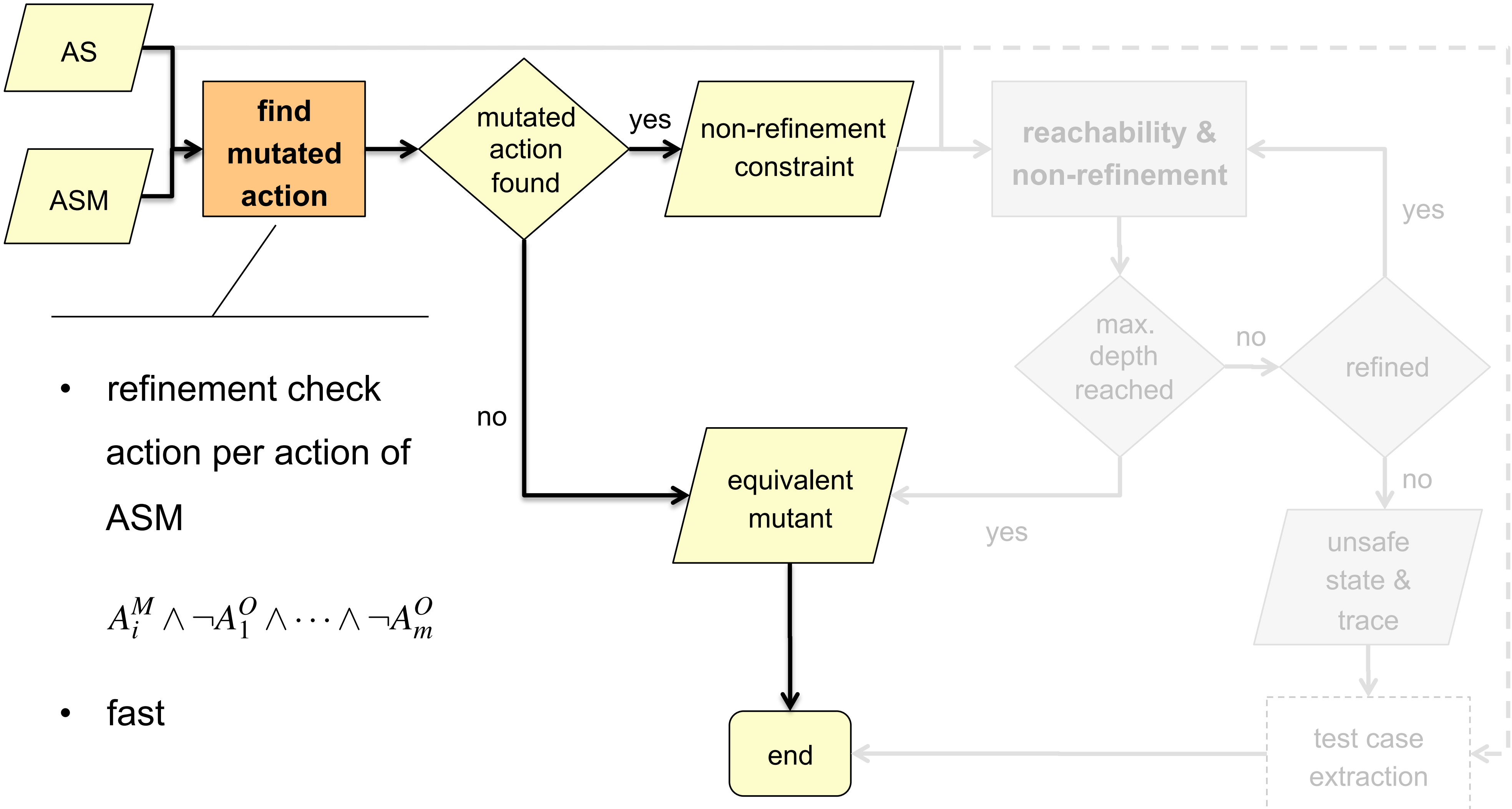
- By application of distributive law:

$$\bigvee_{i=1}^{n} \ \exists \overline{v}, \overline{v}', tr, tr' \ : \ (\overline{v} \in reachable(AS^O, tr) \wedge A_i^M \wedge \neg A_1^O \wedge \cdots \wedge \neg A_m^O)$$

Searching Unsafe States



Finding a Mutated Action

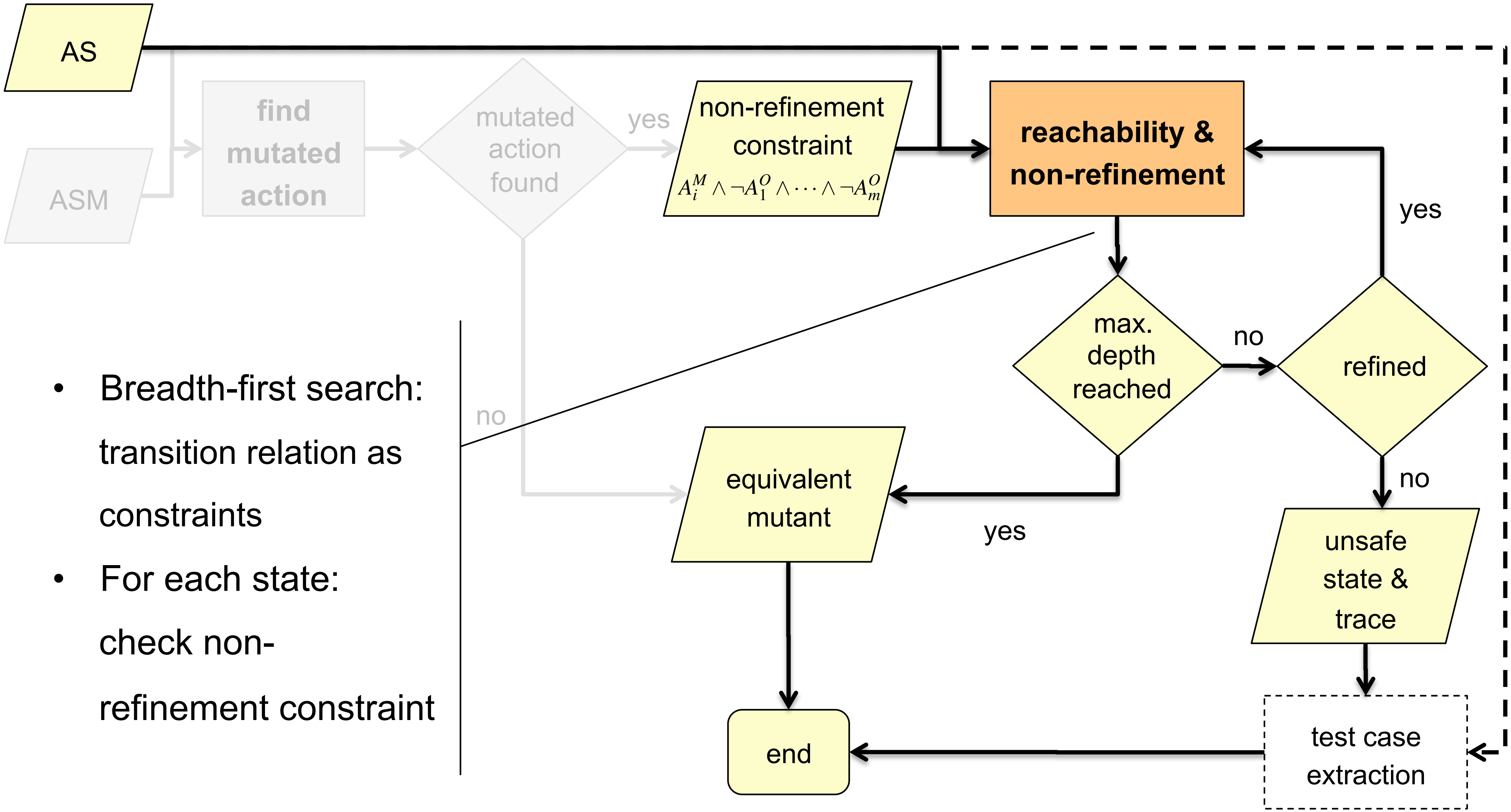


- refinement check action per action of ASM

$$A_i^M \wedge \neg A_1^O \wedge \cdots \wedge \neg A_m^O$$

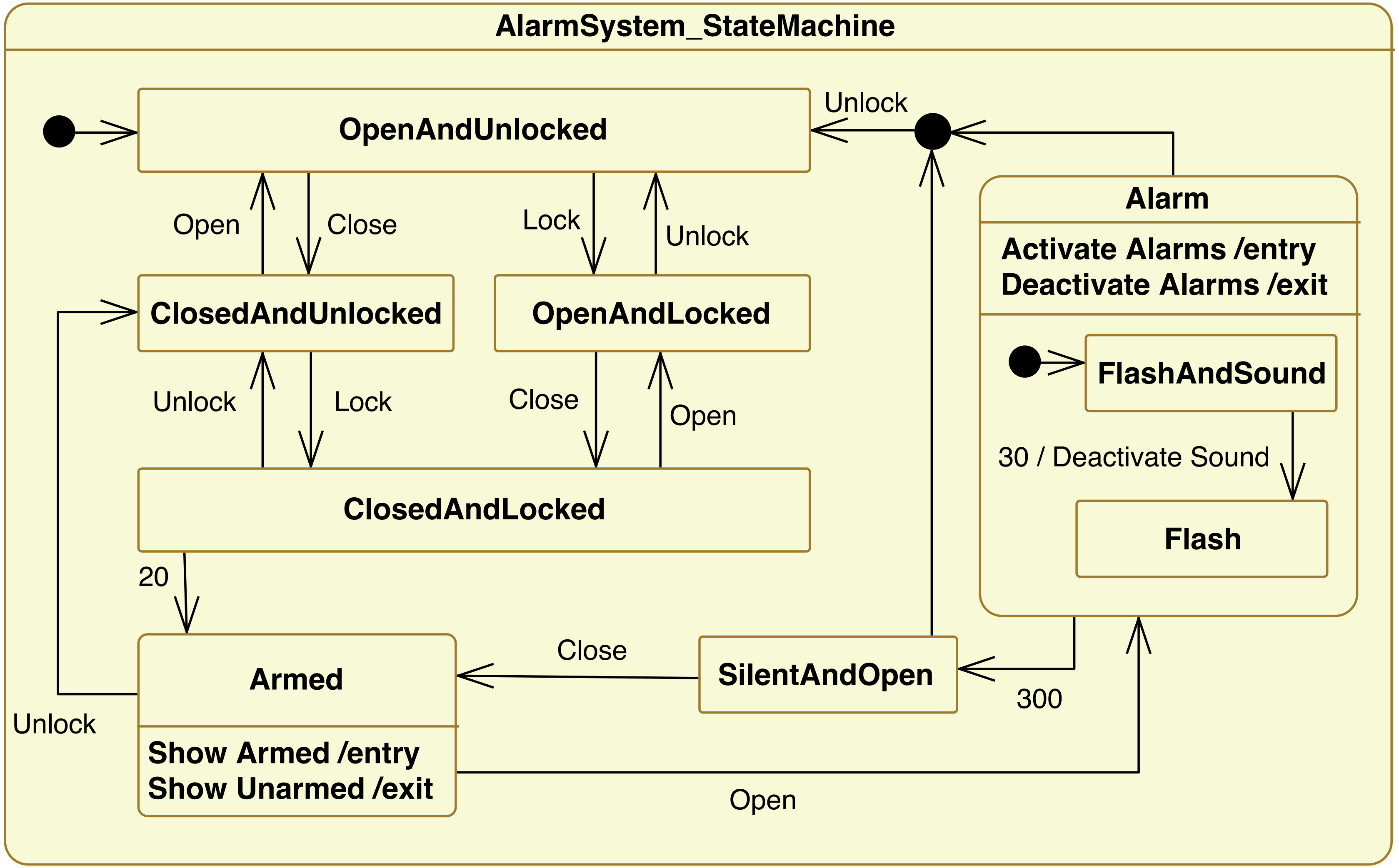
- fast

Reaching an Unsafe State



- Breadth-first search: transition relation as constraints
- For each state: check non-refinement constraint

Empirical Evaluation: Car Alarm System (CAS)



Empirical Evaluation: CAS

```
% definitions: types, variables, state, initial state
as :-
 actions (
   'after'(Wait_time)::(true) => (
     ((Wait_time #= 20 #/\ aState #= 3) =>
         (aState := 2; fromClosedAndLocked_OR_fromSilentAndOpen := 1))
   []
     ((Wait_time #= 30 #/\ aState #= 1 #/\ fromArmed #= 4) =>
         (aState := 0; fromAlarm := 4; fromArmed := 0))
   []
     ((Wait_time #= 270 #/\ aState #= 0 #/\ fromAlarm #= 2) =>
         (aState := 7; fromAlarm := 1; fromArmed := 0))
   ),
   'Lock'::(true) => (
     ((aState #= 6 #/\ fromAlarm #= 0) => (aState := 5))
   []
     ((aState #= 4 #/\ fromArmed #\= 1) => (aState := 3; fromArmed := 0))
   ), ...
 ),
 dood ( 'Lock' [] [X:int]:'after'(X) [] ... ).
```

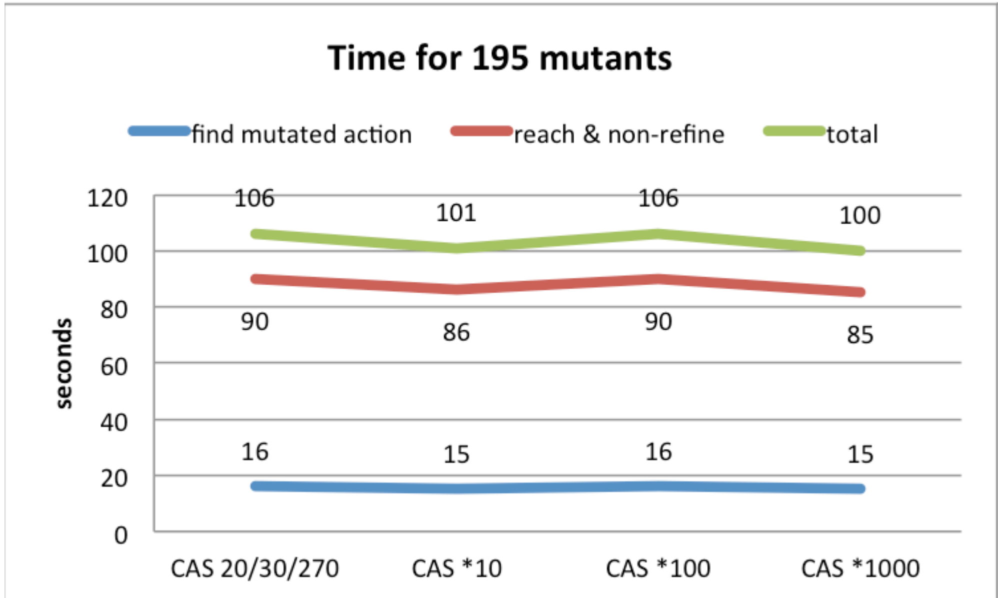
Empirical Evaluation: Mutations

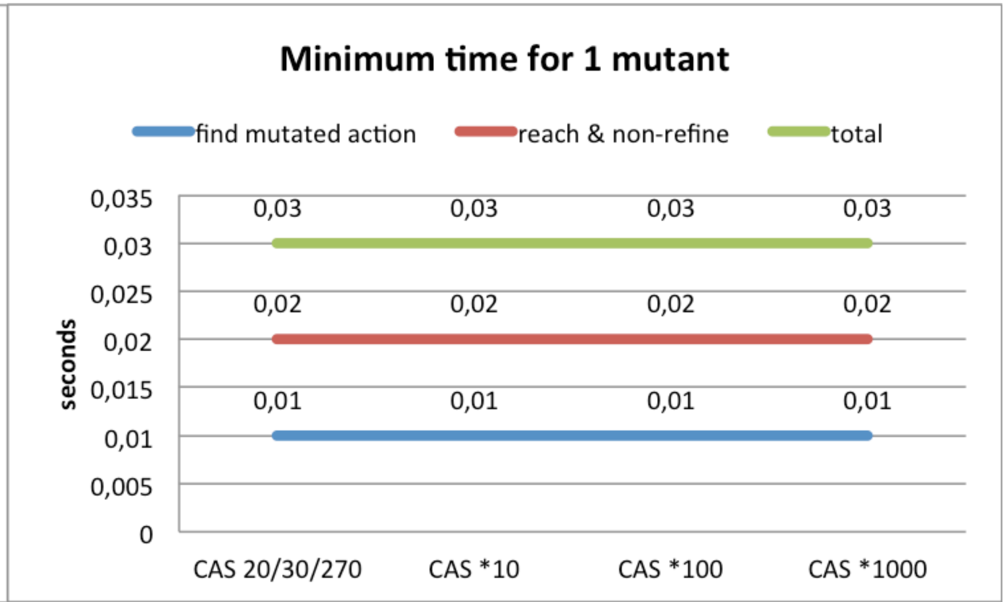
- Manual mutations:
  - guard true: 34 mutants
  - comparison operator inversion: 52 mutants
  - increment integer constant: 116 mutants

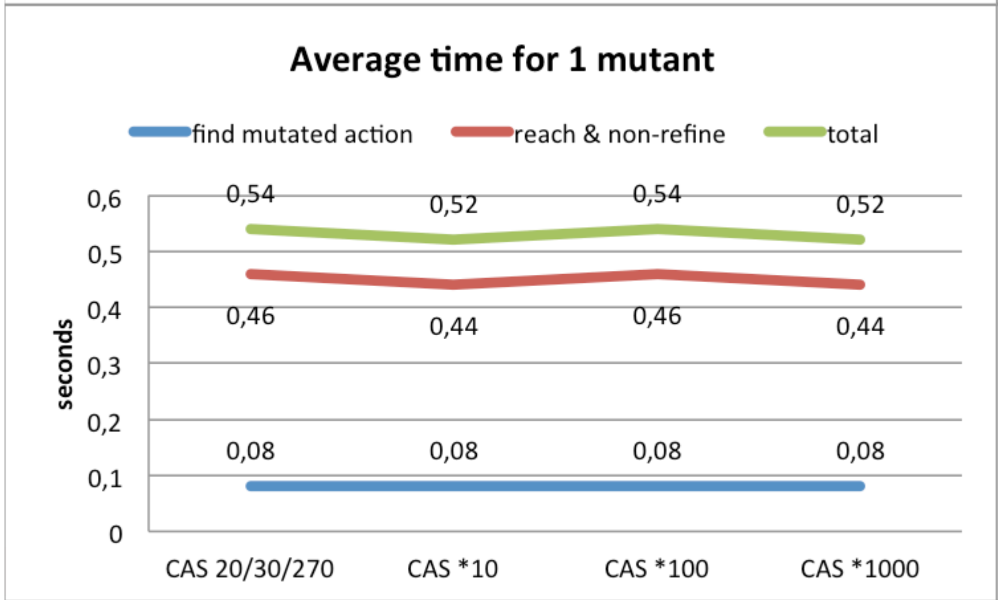
→ 206 mutants
\+ 1 unaltered (original)
= 207 mutants
\- 12 mutants (constraint solver problems)
= **195 mutants**

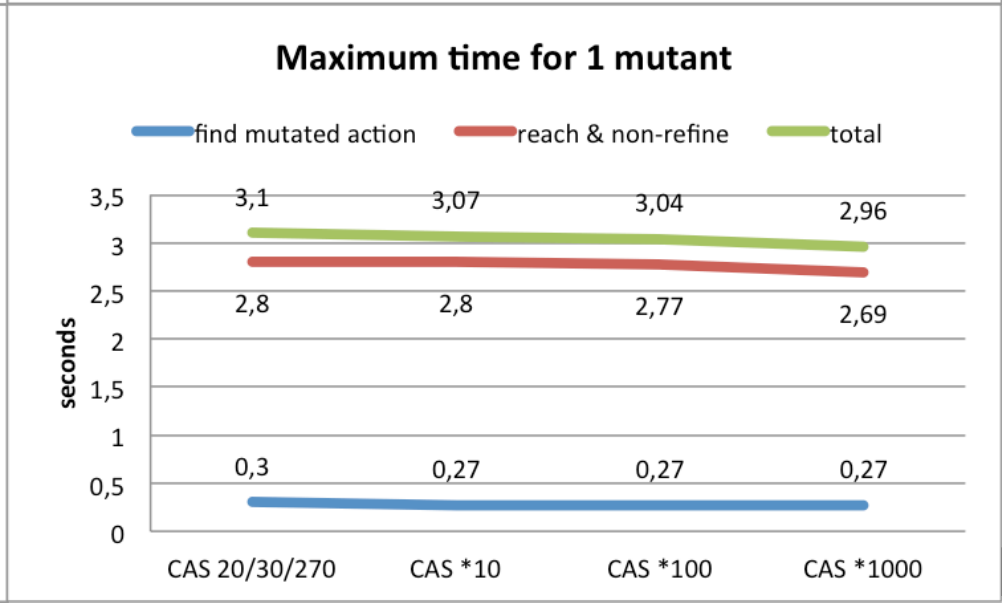
CAS version		Refinement checker: find mutated action	Refinement checker: reach & non-refine	Refinement checker: total
20/30/270	total	16	90	106
	average	0.08	0.46	0.54
	min	0.01	0.02	0.03
	max	0.30	2.80	3.10
*10	total	15	86	101
	average	0.08	0.44	0.52
	min	0.01	0.02	0.03
	max	0.27	2.80	3.07
*100	total	16	90	106
	average	0.08	0.46	0.54
	min	0.01	0.02	0.03
	max	0.27	2.77	3.04
*1000	total	15	85	100
	average	0.08	0.44	0.52
	min	0.01	0.02	0.03
	max	0.27	2.69	2.96

Refinement Checker (symbolic)

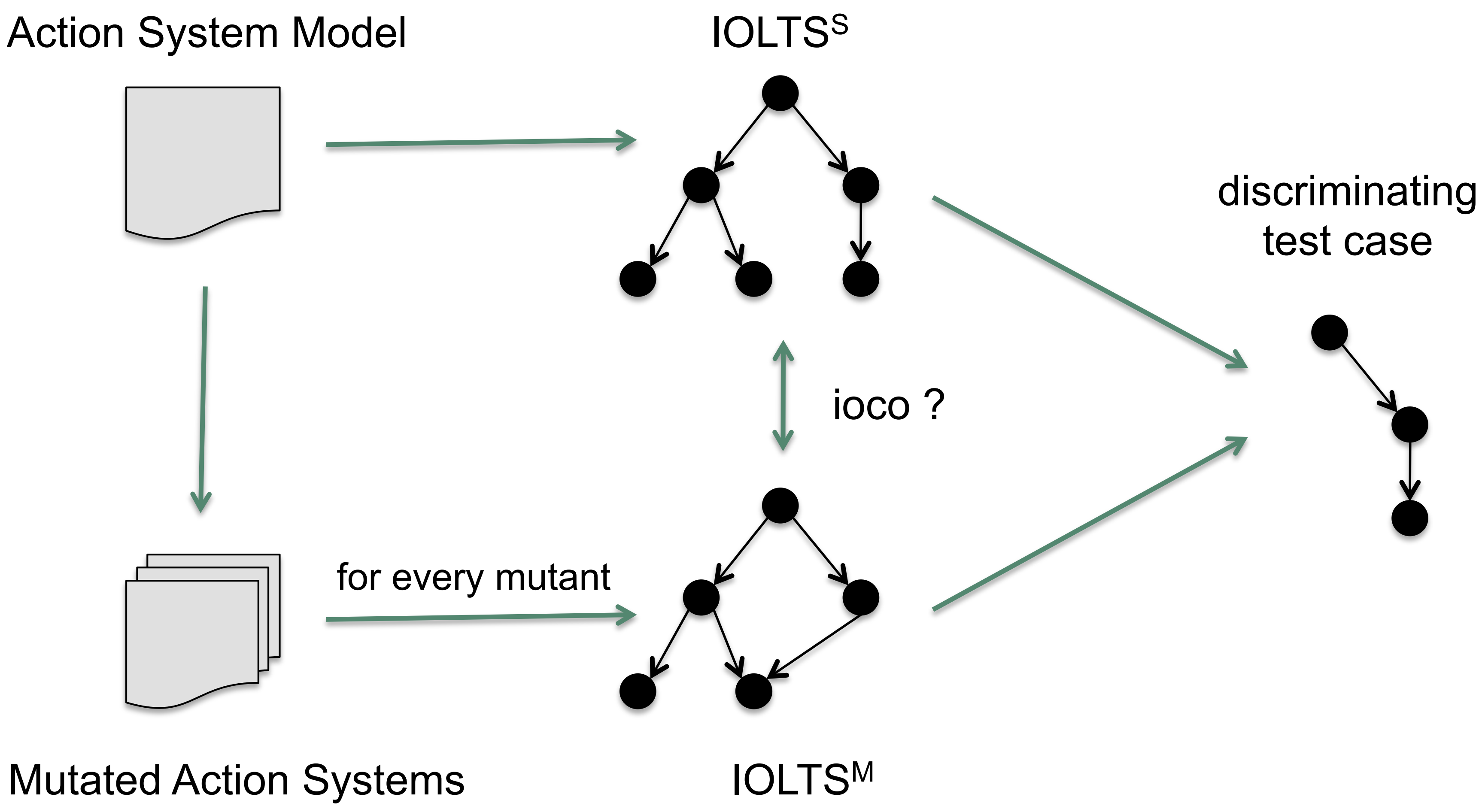






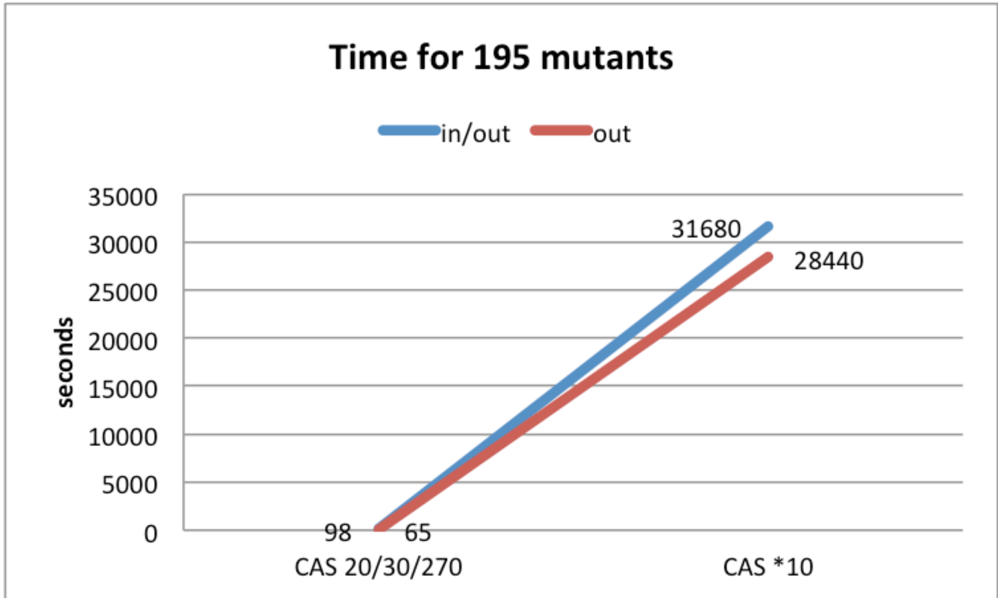


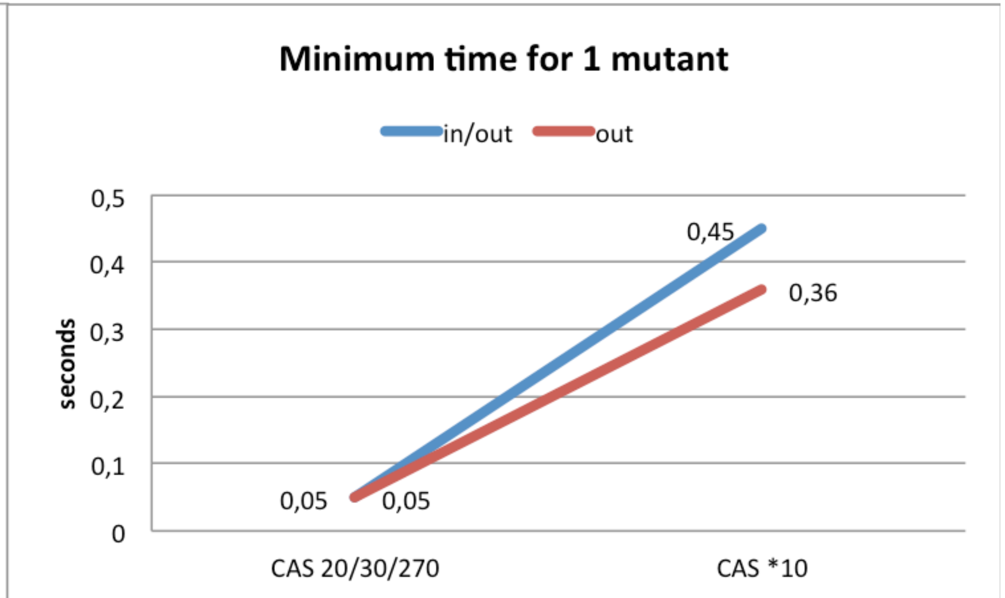
Explicit ioco checking (Ulysses)

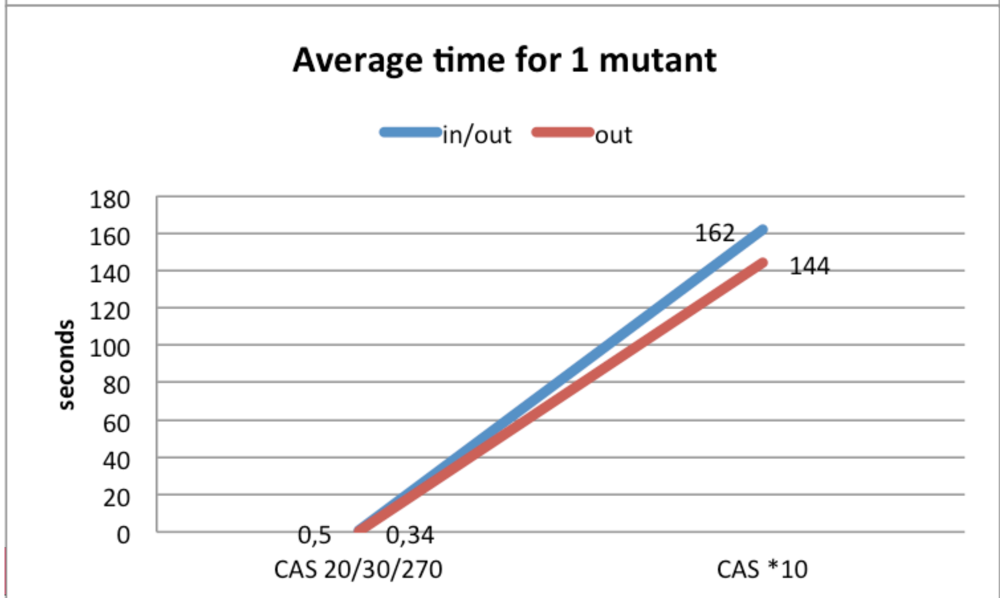


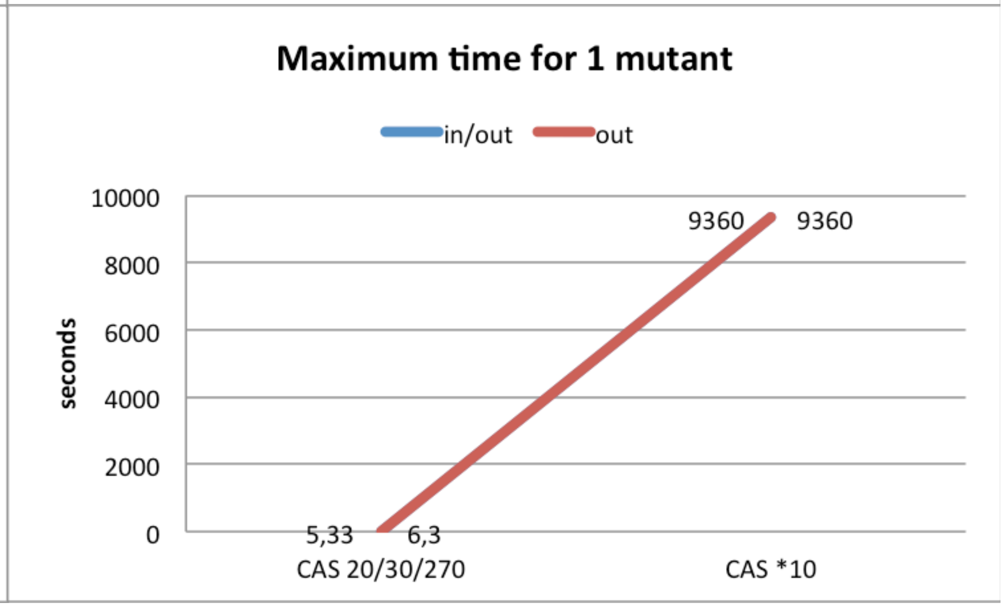
CAS version		Refinement checker			Ulysses	
		find mutated action	reach & non-refine	total	in/out	out
20/30/270	total	16	90	106	98	65
	average	0.08	0.46	0.54	0.50	0.34
	min	0.01	0.02	0.03	0.05	0.05
	max	0.30	2.80	3.10	6.30	5.33
*10	total	15	86	101	8.8 h	7.9 h
	average	0.08	0.44	0.52	2.7 min	2.4 min
	min	0.01	0.02	0.03	0.45	0.36
	max	0.27	2.80	3.07	2.6 h	2.6 h
*100	total	16	90	106	-	-
	average	0.08	0.46	0.54	-	-
	min	0.01	0.02	0.03	-	-
	max	0.27	2.77	3.04	-	-
*1000	total	15	85	100	-	-
	average	0.08	0.44	0.52	-	-
	min	0.01	0.02	0.03	-	-
	max	0.27	2.69	2.96	-	-

Ulysses (explicit)







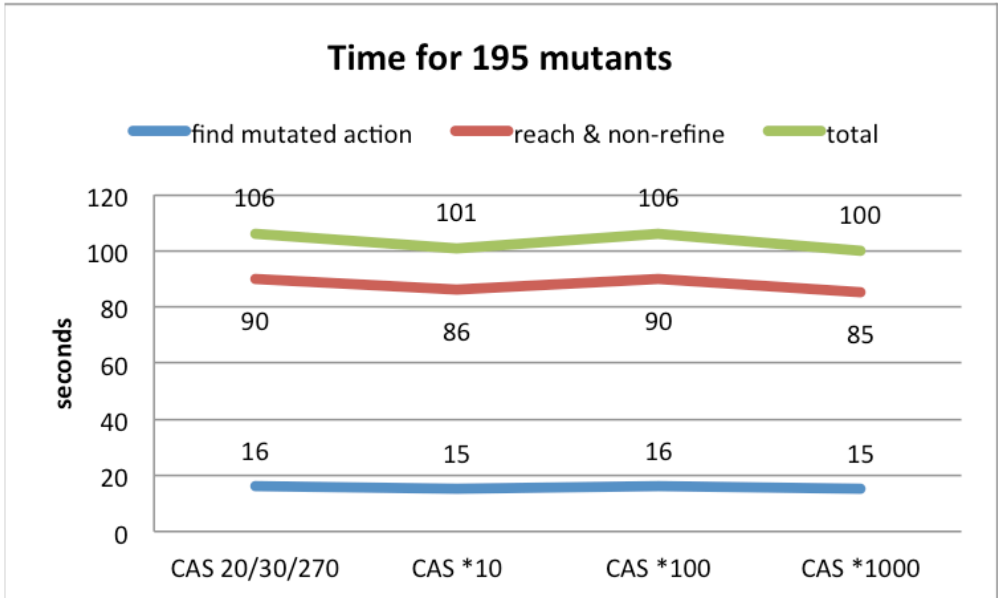


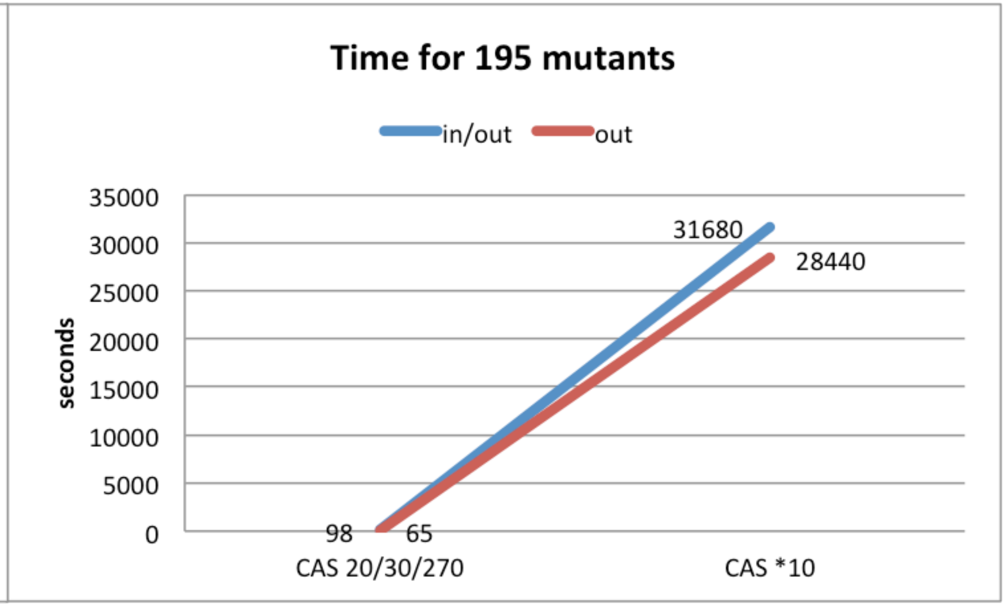
Future Work

- Other constraint solvers, e.g. MINION
- SMT solvers (ongoing diploma thesis)
- Trace to unsafe state → adaptive test case
- More experiments with different systems

Conclusions

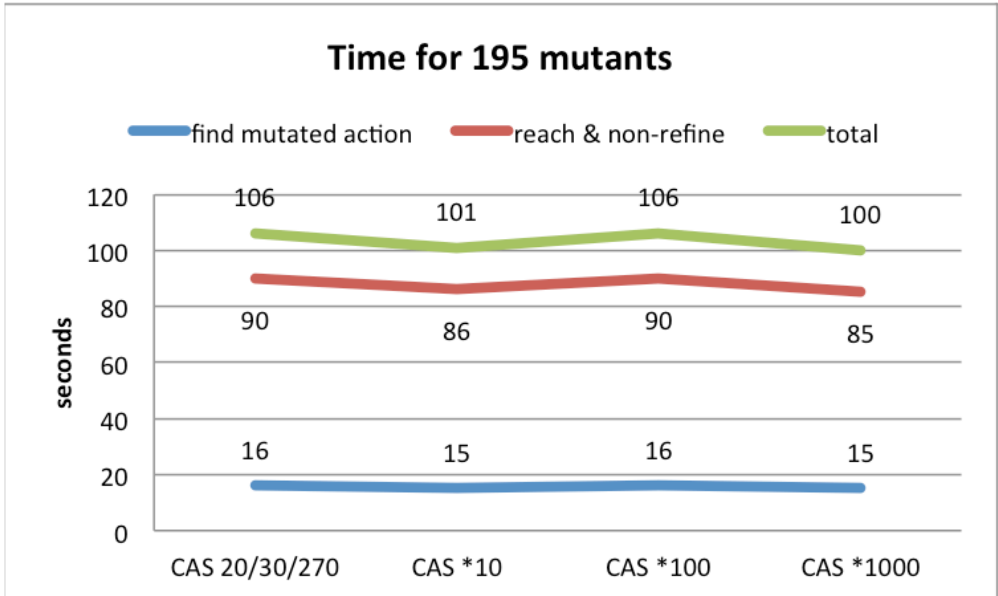
- symbolic vs. explicit

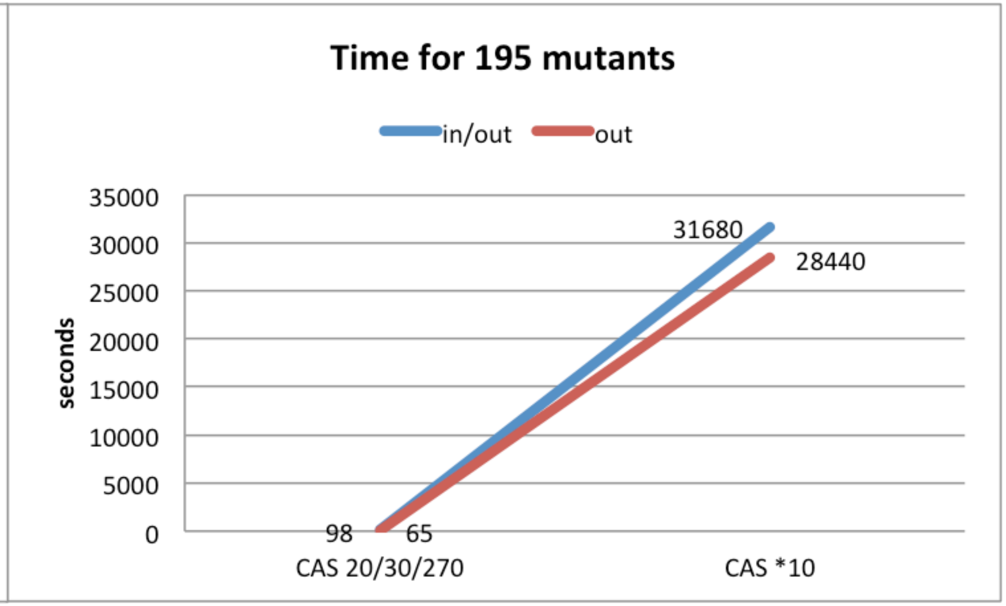




Conclusions

- symbolic vs. explicit





Thank you for your attention!