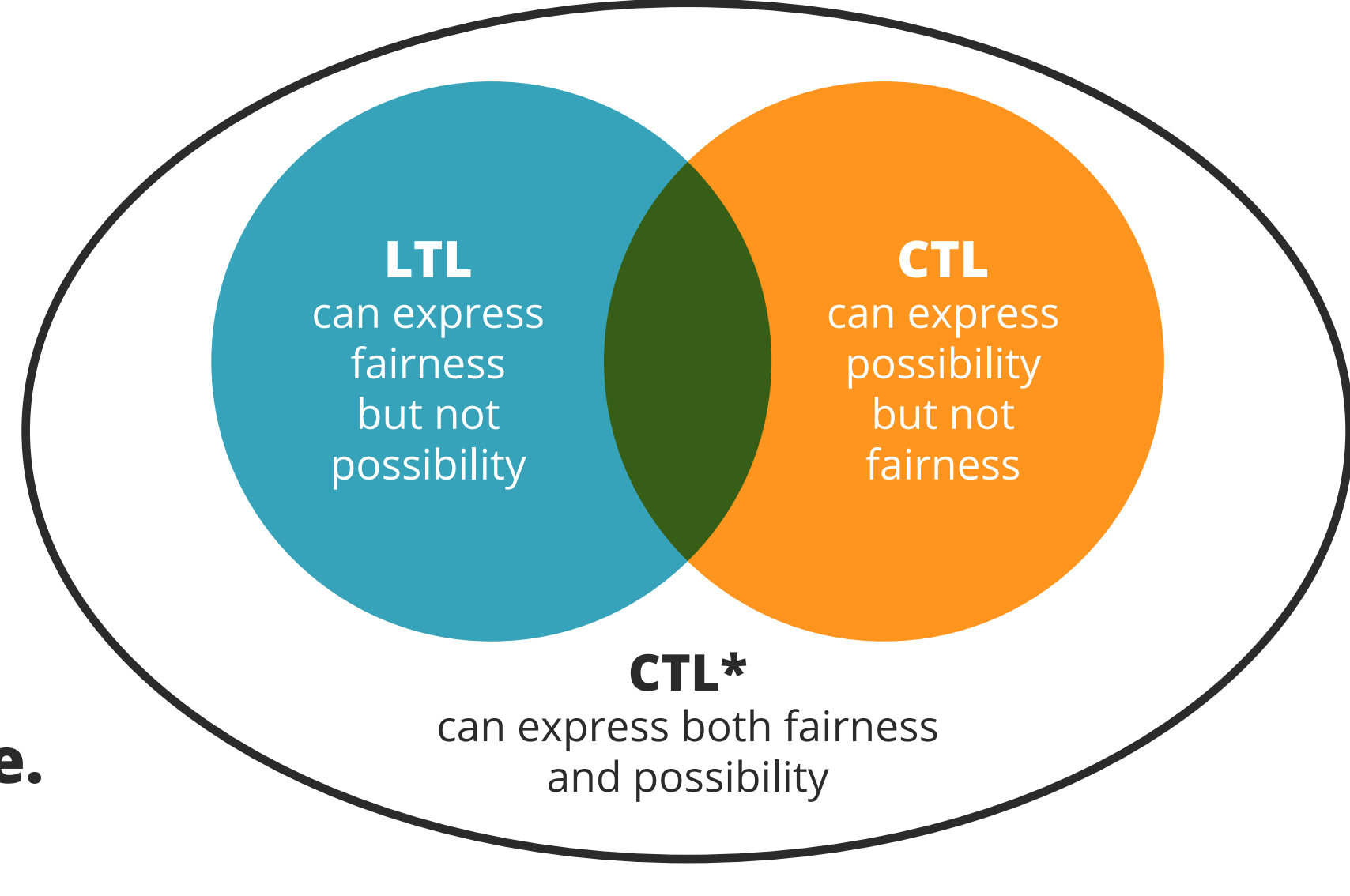


# starMC: an automata based CTL\* model checker

## MOTIVATIONS

Model-checking of temporal logic formulae is a technique used for the formal verification of Discrete Event Systems.

**CTL\* model-checking algorithms are complex, and few tools are available.**



## CONTRIBUTIONS

- the algorithms for a fully symbolic automata-based approach for CTL\*
- the open-source tool **starMC**, a CTL\* model checker for systems specified as Petri nets (to the best of our knowledge **starMC** is the only available CTL\* model-checker based on Büchi automata)
- a public, open benchmark to compare CTL\* tools.

## METHODOLOGY

**1**

Model state space is represented as a Kripke structure

**LOGICAL VIEW**

**MODEL**

**SYMBOLIC VIEW**

**ENCODED MODEL**

**RS MDD.**

**NSF MxD.**

**2**

CTL\* formula is divided into sub-formulae, each represented as a Büchi automaton

**AUTOMATON**

$F\beta$

**SYNCHRONIZATION PROCEDURE**

Model =  $\langle RS^M, NSF^M \rangle$   
Automaton =  $\langle Q, AP, \delta, Q_0, \mathcal{F} \rangle$

```

1: procedure BUILDSYNCHPRODUCT(M, a)
2: // Build the MDD of the initial states.
3: Z0 ← MDD()
4: for each location q0 ∈ Q0 do
5:   for each edge e = q0 → q' in δ: do
6:     Z0 ← Z0 ∪ AddLoc(Sar^M(a), q')
7:   end for
8: end for
9: // Build the Next State Function MxD
10: NSF ← MxD()
11: for each edge e = q → q' in δ: do
12:   nsf^e ← AddLocX(NSF^M ∩ (PS^M × Sar^M(a)), q, q')
13:   NSF ← NSF ∪ nsf^e
14: end for
15: // S_β: array of MDDs, one entry per accept. set F_i ∈ F
16: for each accepting set F_i ∈ F: do
17:   MDD_{F_i} ← [edgeForVar(q) | for each q ∈ F_i]
18:   S_β[i] ← {MDD_{F_i}}
19: end for
20: return (Z0, NSF, S_β)
21: end procedure
    
```

**3**

Synchronized product between model and Büchi automaton

**PRODUCT**

**ENCODED PRODUCT**

**NSF of M ⊗ A**

**RS of M ⊗ A**

**4**

Emerson-Lei algorithm takes the synchronized product and computes the set of satisfying states

**RESULT**

$Sat \exists LTL(NS, F\beta) = \{m_1, m_2, m_4\}$

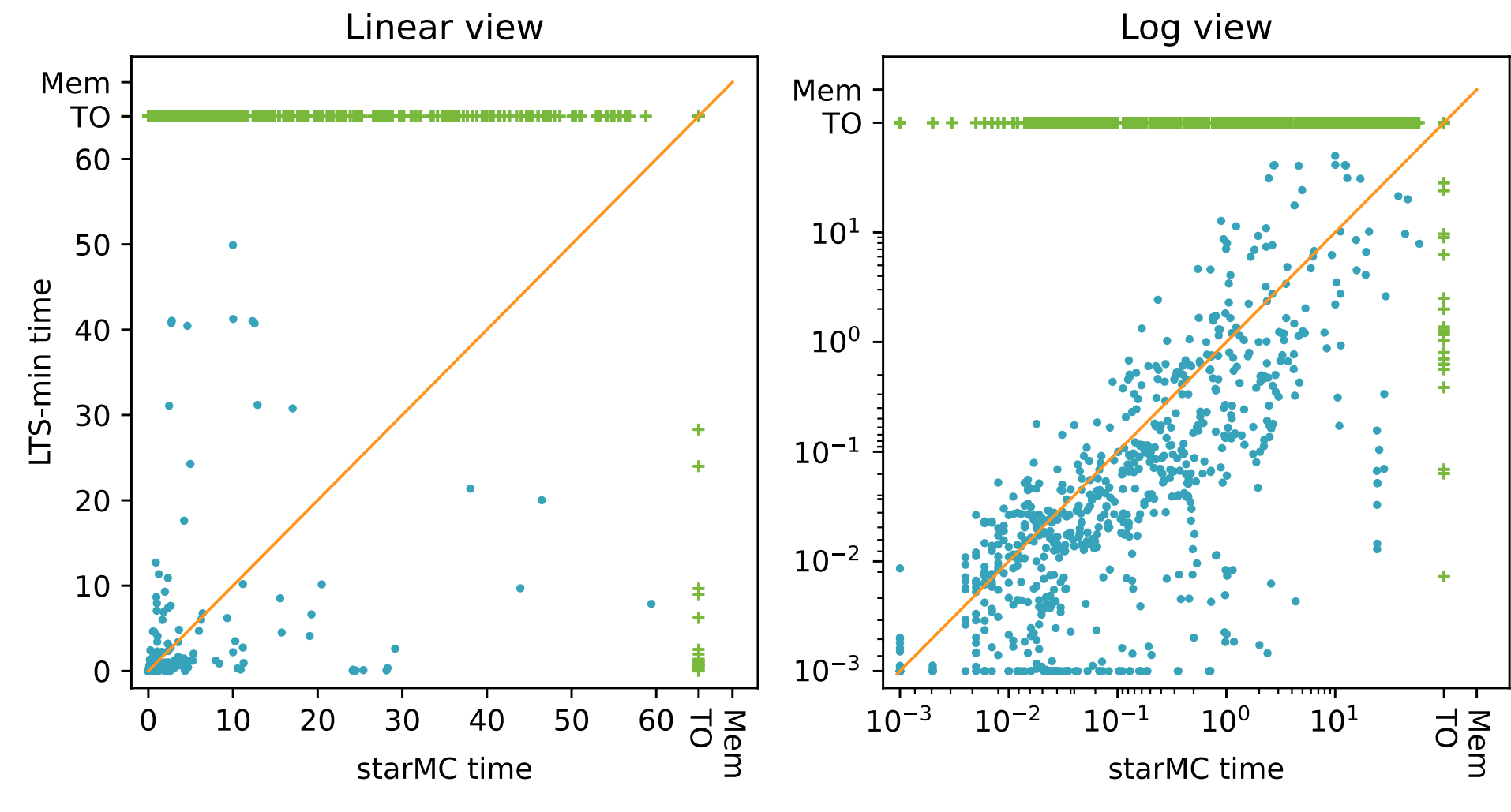
**ENCODED RESULT**

$Sat(EF\beta)$

**EFFICIENCY!**  
Everything is symbolic (decision diagrams)

## RESULTS

In the benchmark, **starMC solves the highest number of queries** (76%) w.r.t. LTS-min (45%); LTSmin is based on  $\mu$ -calculus, not on Büchi automata.



**Benchmark and virtual machines:**  
<https://zenodo.org/record/5752419>

## THE STAR MC TOOL

The starMC tool allows you to **specify the model graphically as a Petri net. Logical formulae are then verified** using the symbolic algorithm for CTL\*.

**MODEL EDITOR**

**FORMULAE EDITOR**

**starMC is user friendly and open-source:**  
<https://github.com/greatspn/SOURCES>

