

Pattern-based Abstractions for Parameterized Model Checking of Distributed Algorithms

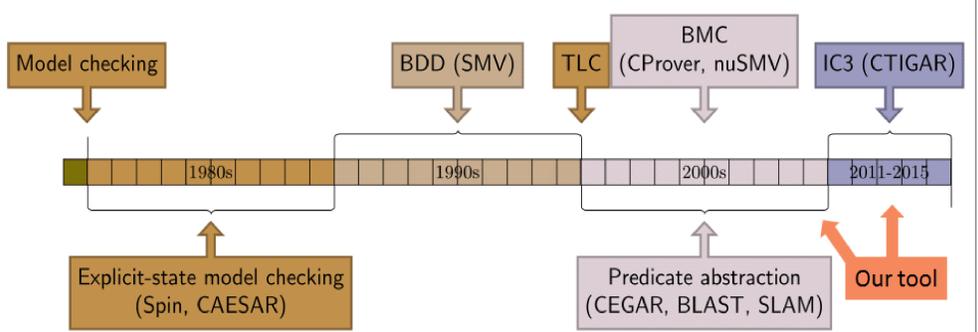
Thanh Hai Tran, Jure Kukovec

TLA+

- Versatile specification language for distributed and concurrent systems
- Based on:
 - Unsorted first-order logic
 - Set theory
 - Temporal logic
- States, transitions: logical formulas
- Supporting tools:
 - TLAPS: the interactive proof system
 - TLC: the explicit-state model checker

Automated Verification with TLC

- Explicit-state** model checker
 - Used at Amazon, Microsoft...
 - Bug missing with low probability
 - Fixed parameters, finite domains
- Goal:** automatically constructs good pattern-based abstractions to verify safety properties of TLA+ specifications



TLA+ Patterns in Srikanth and Toueg's Asynchronous Reliable Broadcast Algorithm

- N, T, F : parameters
- pc : program counters
- $sent$: message channel
- $rcvd$: received messages
- $Proc$: all processes
- $Corr$: correct processes
- $Faulty$: faulty processes
- $Step$: transitions (Receive, UponV1...)

$$\text{ASSUME } \bigwedge N \in \text{Nat} \wedge T \in \text{Nat} \wedge F \in \text{Nat} \\ \bigwedge N > 3 * T \wedge T \geq F \wedge F \geq 0$$

$$\text{Receive}(self) \triangleq \\ \bigwedge newMsgs' \in \text{SUBSET}(sent \cup \text{ByzMsgs}) \\ \bigwedge rcvd' = [i \in Proc \mapsto \text{IF } i \neq self \text{ THEN } rcvd[i] \\ \text{ELSE } rcvd[self] \cup newMsgs']$$

$$\text{UponAcceptSentBefore}(self) \triangleq \\ \bigwedge pc[self] = \text{"SE"} \\ \bigwedge \text{Cardinality}(rcvd'[self]) \geq N - T \\ \bigwedge pc' = [pc \text{ EXCEPT } ![self] = \text{"AC"}] \\ \bigwedge sent' = sent \\ \bigwedge \text{UNCHANGED } \langle Corr, Faulty \rangle$$

$$\text{TypeOK} \triangleq \\ \bigwedge pc \in [Proc \rightarrow \{\text{"V0"}, \text{"V1"}, \text{"SE"}, \text{"AC"}\}] \\ \bigwedge Corr \subseteq Proc \\ \bigwedge Faulty \subseteq Proc \\ \bigwedge sent \subseteq Proc \times M \\ \bigwedge rcvd \in [Proc \rightarrow \text{SUBSET}(sent \cup \text{ByzMsgs})]$$

Initial states

$$\text{Init} \triangleq \\ \bigwedge sent = \{\} \\ \bigwedge pc \in [Proc \rightarrow \{\text{"V0"}, \text{"V1"}\}] \\ \bigwedge rcvd = [i \in Proc \mapsto \{\}]$$

Next steps

$$\text{Next} \triangleq \\ \bigvee \exists self \in Corr : \text{Step}(self) \\ \bigvee \text{UNCHANGED } vars$$

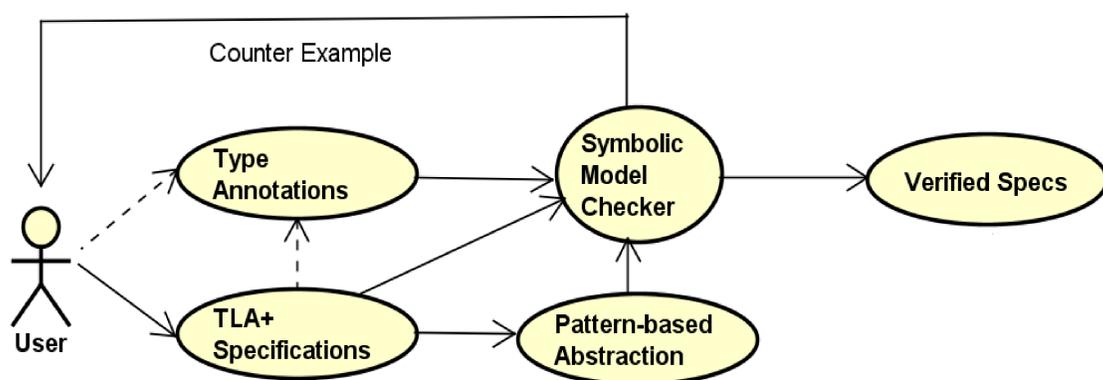
Resilience condition

Many-point update

Threshold-guarded conditions

Type invariant

Our toolchain



Challenges

- TLA+ features: sequences, set cardinality, CHOOSE...
- Type systems
- The classification and extraction of TLA+ patterns
- Pattern-based abstractions
- Quantifier elimination

References

- Lamport, Leslie. "Specifying systems." (2002).
- Newcombe, Chris. "Why amazon chose TLA+." International Conference on Abstract State Machines, Alloy, B, TLA, VDM, and Z. Springer, Berlin, 2014.
- Vanzetto, Hernán. Proof automation and type synthesis for set theory in the context of TLA+. Diss. Université de Lorraine, 2014.