Pattern-based Abstractions for Parameterized Model Checking of Distributed Algorithms

Thanh Hai Tran, Jure Kukovec
TLA+

- Underlying theories
  - Set theory (ZFC)
  - Linear-time logic

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

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

- Tools: TLC (model checker), TLAPS (theorem prover)
- Industrial projects: Paxos, Alpha EV7/EV8…
Goals: Pattern-based Abstractions

- TLA+ patterns
- Pattern-based abstractions
- Fault-tolerant distributed algorithms
Challenges

- TLA+ features: sequences, set cardinality, CHOOSE...
- Type systems
- TLA+ patterns
- Industrial-strength algorithms
Questions?