Automata in Verification

Barbora Šmahlíková

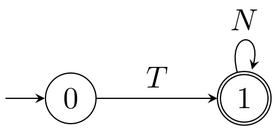
Supervisor: Ing. Ondřej Lengál, Ph.D., Consultant: Ing. Vojtěch Havlena, Ph.D.

Excel@FIT 2024

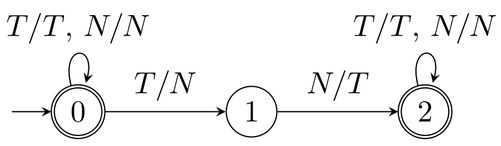
BRNO UNIVERSITY OF TECHNOLOGY — FACULTY OF INFORMATION TECHNOLOGY

Regular Model Checking

- Technique for verification of **infinite-state systems**
- System representation = initial configurations + transitions
- **Initial configurations** = finite automaton
  - Token passing protocol: linear array of processes, each process either owns a token (T) or does not own a token (N)
  - Initial configurations: only the first process owns a token



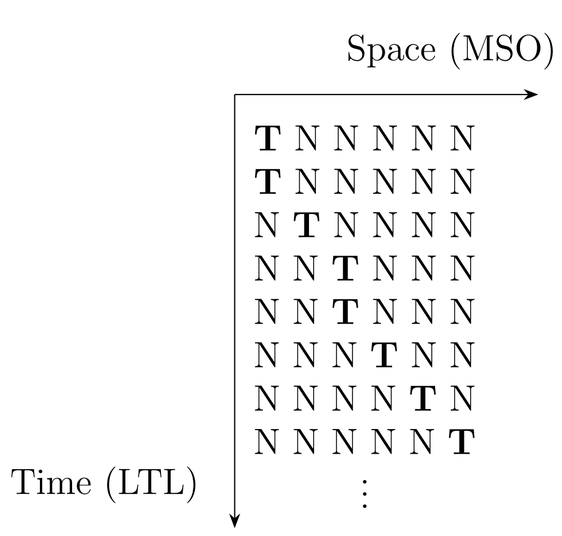
- **Transitions** = finite length-preserving transducer
  - In every step, the token can either stay at the same process, or it can be passed to its right neighbour



- Computing all reachable configurations in general does not have to terminate

LTL(MSO)

- Two-dimensional logic for specifying properties to be verified
- **MSO** for properties of system configurations
- **LTL** for temporal properties
- Models = infinite sequences of constant-length words



Syntax:

$$\psi ::= \forall i.\, \psi \mid \forall I.\, \psi \mid \exists i.\, \psi \mid \exists I.\, \psi \mid \mathsf{G}\psi \mid \mathsf{F}\psi \mid \psi \mathsf{W} \psi \mid \mathsf{X}\psi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \neg\psi \mid \varphi$$

$$\varphi ::= i \in I \mid I \subseteq J \mid i = j + 1 \mid a[i]$$

- Temporal operators G, F, W and X
- First-order process variables i, j, second-order process variables I, J
- Atomic proposition a

Formula for initial configurations of token passing protocol:

$$\forall i.\, T[i] \leftrightarrow (\neg\exists j.\, i = j + 1)$$

Contribution

- Extension of regular model checking for LTL(MSO) to **hyperproperties**
- Implementation of this algorithm in the tool ParaHyper

Hyperproperties

- LTL implicitly quantifies over only one execution trace
- HyperLTL allows to quantify explicitly over **multiple traces**
- Observational determinism: identical output across all traces

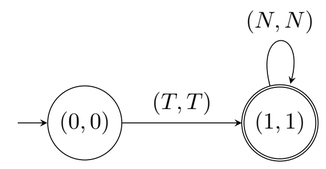
$$\forall \pi_1.\, \forall \pi_2.\, \mathsf{G}(o_{\pi_1} \leftrightarrow o_{\pi_2})$$

Verifying Hyperproperties

- Extension of LTL(MSO) to **HyperLTL(MSO)**
- Explicit quantification over multiple traces

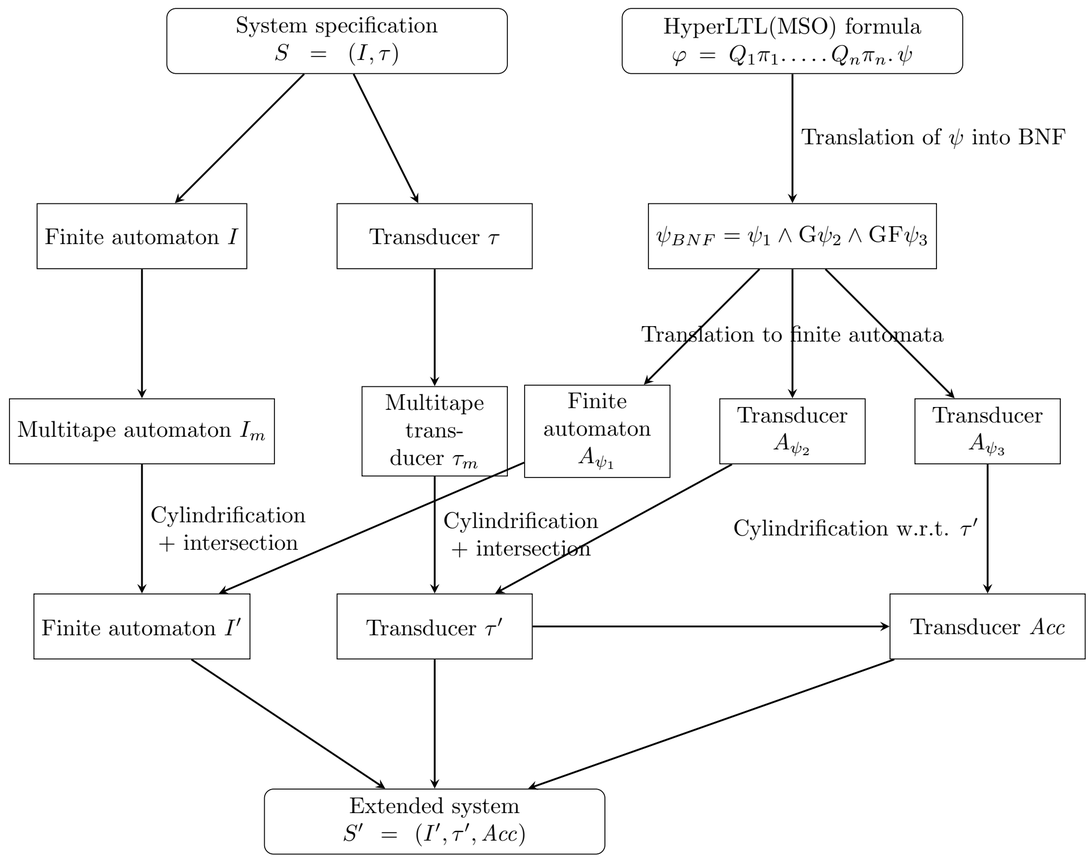
$$\forall \pi_1.\, \exists \pi_2.\, \mathsf{G}(\forall i.\, \exists j.\, (i = j + 1 \rightarrow (T_{\pi_1}[i] \rightarrow T_{\pi_2}[j])))$$

- **Multitape automata** – one tape for every trace



Extending the System with Configuration Variables

- Body of the formula is translated into **Büchi normal form**
- $\psi_{BNF} = \psi_1 \wedge \mathsf{G}\psi_2 \wedge \mathsf{GF}\psi_3$ describes initial configurations, transitions and accepting transitions
- New configuration variables are added to the system
- Extended system describes all accepting runs (without taking into account quantifiers over traces)



Generating Advice Bits

- If a pair $\langle A, \prec \rangle$ of a finite automaton A and a transducer $\prec$ satisfies the following conditions, the property holds in the system

$$\mathcal{L}(\Pi(I')) \subseteq \mathcal{L}(\Pi(A)), \tag{1}$$

$$A \text{ is } \xrightarrow{\tau'}\text{-inductive, i.e., } \forall x.\, \forall y.\, x \in A \wedge (x \xrightarrow{\tau'} y) \Rightarrow y \in A, \tag{2}$$

$$\prec \text{ is a } \textit{strict preorder} \text{ on } A, \tag{3}$$

$$\forall \{\pi_1 \colon x_1, \ldots, \pi_n \colon x_n, c \colon c_1\} \in \mathcal{L}(A).\, \exists \{\pi_1 \colon y_1, \ldots, \pi_n \colon y_n, c \colon c_2\} \in \mathcal{L}(A). \tag{4}$$
$$(\{\pi_1 \colon y_1, \ldots, \pi_n \colon y_n, c \colon c_2\} \xrightarrow{\tau'} \{\pi_1 \colon x_1, \ldots, \pi_n \colon x_n, c \colon c_1\} \wedge$$
$$\{\pi_1 \colon y_1, \ldots, \pi_n \colon y_n, c \colon c_2\} \xrightarrow{\prec} \{\pi_1 \colon x_1, \ldots, \pi_n \colon x_n, c \colon c_1\}) \vee$$
$$\{\pi_1 \colon x_1, \ldots, \pi_n \colon x_n, c \colon c_1\} \in \mathcal{L}(I')$$

$$\forall \{\pi_1 \colon x_1, \ldots, \pi_n \colon x_n\} \in \mathcal{L}(\Pi(A)).\, Q_1\pi_1 \colon y_1 \ldots Q_n\pi_n \colon y_n.\, \exists c \colon c_1.\, \exists c \colon c_2. \tag{5}$$
$$\Big(\bigwedge_{i \in \{j \mid 1 \le j \le n \wedge Q_j = \forall\}} x_i \xrightarrow{\tau} y_i \Big) \Rightarrow$$
$$\{\pi_1 \colon x_1, \ldots, \pi_n \colon x_n, c \colon c_1\} \in \mathcal{L}(A) \wedge \{\pi_1 \colon y_1, \ldots, \pi_n \colon y_n, c \colon c_2\} \in \mathcal{L}(A) \wedge$$
$$\{\pi_1 \colon x_1, \ldots, \pi_n \colon x_n, c \colon c_1\} \xrightarrow{\tau'} \{\pi_1 \colon y_1, \ldots, \pi_n \colon y_n, c \colon c_2\} \wedge$$
$$(\{\pi_1 \colon x_1, \ldots, \pi_n \colon x_n, c \colon c_1\} \xrightarrow{\prec} \{\pi_1 \colon y_1, \ldots, \pi_n \colon y_n, c \colon c_2\} \vee$$
$$\{\pi_1 \colon x_1, \ldots, \pi_n \colon x_n, c \colon c_1\} \xrightarrow{Acc} \{\pi_1 \colon y_1, \ldots, \pi_n \colon y_n, c \colon c_2\}).$$

Implementation

- Algorithm implemented in the tool ParaHyper
- SAT solver is used for generating candidate pairs of advice bits