# The Complexity of Model Checking Multi-Stack Systems

Benedikt Bollig[1] Dietrich Kuske[2] Roy Mennicke[2]

[1]LSV, CNRS and ENS de Cachan, France

[2]Ilmenau University of Technology, Germany

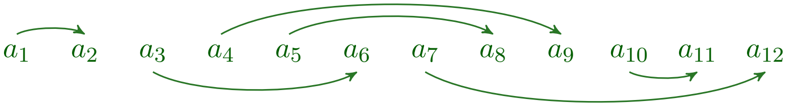
# Multiply Nested Words

Concurrent programs with recursive procedure calls can be modelled by pushdown automata with multiple stacks.

An execution of a multi-stack system can be considered as a word with multiple nesting relations.

Each edge of a nesting relation relates a push (procedure call) with its matching pop (return).

Consider the following 2-nested word:

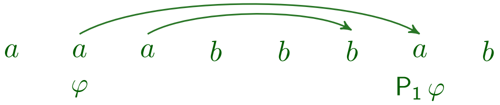



The first (second) nesting relation is represented by the upper (lower) edges.

# Example Modalities

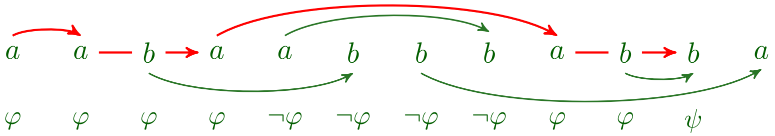
Matching Call: $\mathsf{P}_s\,\varphi$ (where $s$ is a stack)

(expresses that $\varphi$ holds at the matching call position)




Existential Until: $\varphi\,\mathsf{EU}\,\psi$

(there exists a path such that $\varphi$ holds until $\psi$ holds)




$\varphi\,\mathsf{EU}\,\psi$ holds at the minimal position

# MSO-definable Temporal Logics

We consider all temporal logics whose modalities are MSO-definable, based on the atomic formulas:

$$P_a(x) \mid x \lessdot y \mid x \prec_s y \mid x = y \mid x \in Z$$
$$\mid \mathrm{call}_s(x) \mid \mathrm{return}_s(x) \mid \min(x) \mid \max(x)$$

Modality is $\mathsf{M}\Sigma_n$-definable if it is definable by a formula

$$\exists \overline{Z}_1 \forall \overline{Z}_2 \ldots \exists/\forall \overline{Z}_n : \psi$$

where $\psi$ is a first-order formula ($\overline{Z}_i$ are tuples of set variables)

# Example Modalities Continued

Matching Call: $\mathsf{P}_s\,\varphi$

(expresses that $\varphi$ holds at the matching call position)

$$[\![\mathsf{P}_s]\!](Z_1, x) = \exists y\,(y \prec_s x \wedge y \in Z_1)$$

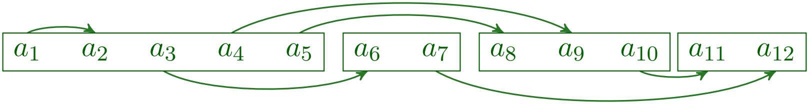
Existential Until: $\varphi\,\mathsf{EU}\,\psi$

(there exists a path such that $\varphi$ holds until $\psi$ holds)

$$[\![\mathrm{EU}]\!](Z_1, Z_2, x) = \exists P \left[ \begin{array}{l} P \cap Z_2 \neq \emptyset \wedge P \subseteq Z_1 \cup Z_2 \wedge x \in P \\ \wedge \forall y \in P\,(x = y \vee \exists z\colon (z \in P \wedge \varphi(z, y))) \end{array} \right]$$

where $\varphi(z, y) = z \lessdot y \vee z \prec_1 y \vee z \prec_2 y \vee \ldots$

# Phase-Boundedness (La Torre et al. '07)




Phase: interval in which all returns refer to the same stack

$\tau$-phase nested word: can be divided into $\tau$ many phases

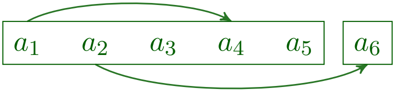
example nested word is a 4-phase NW;
it is no 3-phase nested word since no two of the positions 2, 6, 8, and 10 can belong to the same phase

**Bounded Satisfiability Problem of a temporal logic TL**
Input: temporal formula $F$ from TL and phase bound $\tau \in \mathbb{N}$
Question: Is there a $\tau$-phase nested word satisfying $F$?

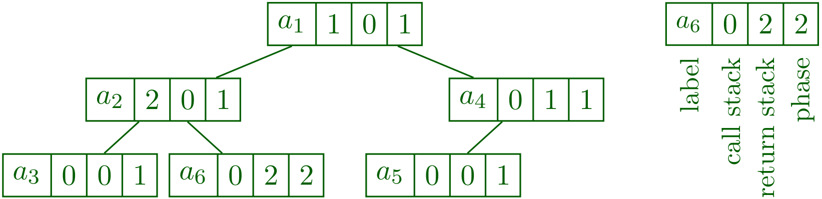
decidable in EXPTIME if phase bound $\tau$ is fixed (Bollig, Cyriac, Gastin, Zeitoun '11)

# Upper Bound: NWs to Trees (La Torre et al.)

2-phase 2-nested word $\nu$:



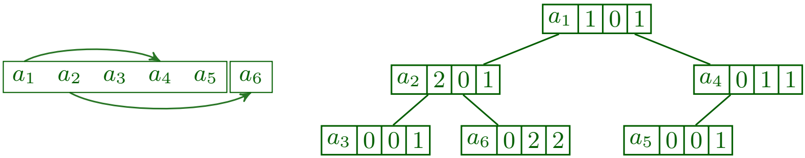

Tree $t_\nu$ representing $\nu$:




**Theorem (La Torre, Madhusudan, and Parlato '07)**

From $\tau \in \mathbb{N}$, one can construct in 2-EXPTIME a tree automaton recognizing the set of all tree representations of $\tau$-phase nested words.

# Upper Bound: From Formulas to Tree Automata




Given: $\varphi(x_1, \ldots, x_k, Z_1, \ldots, Z_\ell)$ MSO-formula and $\tau \in \mathbb{N}$;
Goal: construct a "small" tree automaton $\mathcal{A}$ such that
$\nu, x_1, \ldots, x_k, Z_1, \ldots, Z_\ell \models \varphi \iff (t_\nu, x_1, \ldots, x_k, Z_1, \ldots, Z_\ell) \in L(\mathcal{A})$

For this: construct tree automata for all (negated) atomic formulas in space polynomial in $\tau$

Quite easy for:

- $\mathrm{call}_s(x)$
- $\mathrm{return}_s(x)$
- $x \prec_s y$
- $\min(x)$
- $P_a(x)$
- $x \in Z$
- $x = y$

Also easy: transforming their negations into automata

# Upper Bound: Recovering the Relation $\lessdot$

La Torre, Madhusudan, Parlato '07: $x \leq y$

We have $\neg(x \lessdot y)$ iff

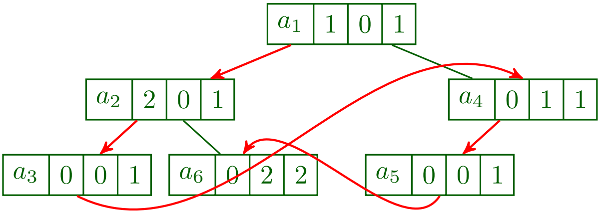
- $y \leq x$ or
- there exists $z$ such that $x < z < y$.

Standard automata techniques: $\neg(x \lessdot y)$

Similarly: $\neg\max(x)$

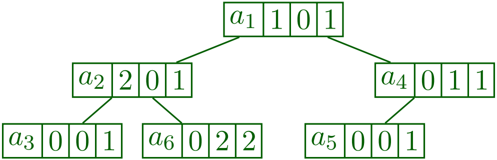
Remaining: $x \lessdot y$ and $\max(x)$

Recovering the direct successor relation is quite difficult.



# Upper Bound: Characterizing $\leq$

New characterisation of the order relation $\leq = (\lessdot)^*$ of a nested word $\nu$ in the tree $t_\nu$:




phase word pw$(v)$ of a node $v$: sequence of the phases on the path from the root of $t_\nu$ to $v$ where repetitions are deleted

strict partial order $\sqsubset$: $s = (s_1, \ldots, s_m) \sqsubset t = (t_1, \ldots, t_n)$ if and only if

- $s_m < t_n$ or
- $s_m = t_n$ and $(s_1, \ldots, s_{m-1}) \sqsupset (t_1, \ldots, t_{n-1})$.

For instance: $(1, 2, 4) \sqsubset (1, 5)$ and therefore $(1, 2, 4, 6) \sqsupset (1, 5, 6)$.

# Upper Bound: Characterizing $\leq$

**Lemma**

Let $\nu$ be a nested word and $x, y$ be positions. Then $x < y$ iff

(1) $\mathrm{pw}(x) \sqsubset \mathrm{pw}(y)$

(2) $\mathrm{pw}(x) = \mathrm{pw}(y)$ and $x$ is a predecessor of $y$ in $t_\nu$

(3) $\mathrm{pw}(x) = \mathrm{pw}(y)$ and there exist positions $z, x', y'$ such that $x' \neq y'$, $x'$ and $y'$ are children of $z$, $x'$ is a predecessor of $x$ and $y'$ one of $y$, and
$x'$ is left child of $z \iff$
$\big(|\mathrm{pw}(x)| - |\mathrm{pw}(z)|$ even iff $x'$ and $y'$ belong to the same phase$\big)$

This allows us to construct tree automata for $x \lessdot y$ and $\max(x)$ in polynomial space.

We save one exponent timewise compared to La Torre, Madhusudan, Parlato.

# Upper Bound

**Theorem**

A temporal formula $F$ can be transformed in polynomial time into an equivalent formula

$$\psi = \exists \overline{Z}(\neg\psi_1(\overline{Z}) \wedge \forall x\, \psi_2(x, \overline{Z}))$$

such that $\psi_i$ is of the form $\exists \overline{Z}_1 \forall \overline{Z}_2 \ldots \exists/\forall \overline{Z}_{n+1} \colon \varphi$ where $\varphi$ is quantifier-free.

Proof uses Hanf's locality principle and exploits the fact that every position has at most one preceding (resp. succeeding, matching return, and matching call) position

**Theorem**

Let TL be some $\mathsf{M}\Sigma_n$-definable temporal logic. The bounded satisfiability problem of TL is in $(n+2)$-EXPTIME.

# Lower Bound: Labeled Grids

**Theorem**

For every $n > 0$, there exists an $\mathsf{M}\Sigma_n$-definable temporal logic $\mathrm{TL}^G$ over labeled grids such that the following problem is $n$-EXPSPACE-hard:

Input: temporal formula $F$ from $\mathrm{TL}^G$ and $m \in \mathbb{N}$

Question: Is there a labeled grid with $m$ columns satisfying $F$?

Proof idea: let $M$ be Turing machine solving an $n$-EXPSPACE-complete problem; reduce the word problem of $M$ to the bounded satisfiability problem of some $\mathsf{M}\Sigma_n$-definable temporal logic over labeled grids

Core of the proof: encoding of large counters using formulas of low monadic quantifier alternation depth (cf. Kuske and Gastin '10, Matz '02)

# Lower Bound: Representing Grids by NWs

Labeled grid $G$ over alphabet $\Gamma$:

| | |
|---|---|
| $a_1$ | $a_2$ |
| $a_3$ | $a_4$ |
| $a_5$ | $a_6$ |

Representation of $G$ as 2-nested word $\nu_G$ over $\Gamma \uplus \{\perp\}$:

$$a_1 \ \perp \ a_3 \ \perp \ a_5 \ \perp \quad \perp \ \perp \ \perp \ \perp \ \perp \ \perp \quad a_2 \ \perp \ a_4 \ \perp \ a_6 \ \perp$$

If $G$ has exactly $m$ columns, $\nu_G$ can be divided into $(2m-2)$ phases.

**Theorem**

For all $n > 0$, there is an $\mathsf{M}\Sigma_n$-definable temporal logic whose bounded satisfiability problem is $n$-EXPSPACE-hard.

# Conclusion

bounded satisfiability problem of every $\mathsf{M}\Sigma_n$-definable temporal logic is solvable in $(n+2)$-EXPTIME

for each level $n$, there exists an $\mathsf{M}\Sigma_n$-definable temporal logic whose bounded satisfiability problem is $n$-EXPSPACE-hard

Future Work:

- close the gap between the lower and upper bounds
- consider other under-approximation concepts for nested words (like bounded split-width recently introduced by Cyriac, Gastin, and Narayan Kumar)
- investigate the complexity of model checking message-passing automata using MSO-definable temporal logics