A Linear-Time Nominal $\mu$-Calculus with Name Allocation

Daniel Hausmann, Stefan Milius and Lutz Schröder

Gothenburg University, Sweden and University Erlangen-Nürnberg, Germany

FM Seminar, Gothenburg

17 August 2021
Linear-time (e.g. LTL) vs. branching-time (CTL, μ-calculus)

Basic linear-time model checking principle:
Transform $\varphi$ to automaton $A(\varphi)$, check inclusion of model in $A(\varphi)$

Inclusion checking for “data automata” (infinite alphabet $\leadsto$ data):

- Register Automata (RA) [Kaminski et al. 1994] undecidable
- Nondeterministic Orbit-finite Automata (NOFA) [Neven et al. 2004, Boyančzyk et al. 2014] undecidable
- Variable Automata [Grumberg et al. 2010] undecidable
Logics with Freeze Quantification

Freeze LTL [Demri, Lazić, 2007]:
- paths: data words \((P_1, d_1), (P_2, d_2), \ldots\)
- operators \(\downarrow_r \varphi: \text{"} r \leftarrow d_i; \varphi \text{"}, \uparrow_r: \text{"} d_i = r? \text{"}\)

Flat Freeze LTL [Bollig et al. 2019]:
- for all subformulae \(\phi_1 U \phi_2\), no freeze operator in \(\phi_1\)

Model Checking for Freeze LTL:
- Freeze LTL over RA [Demri, Lazić, 2007] undecidable
- Flat Freeze LTL over OCA [Bollig et al. 2019] \(\text{NExpTime}\)

One-Counter Automata
[Schröder et al. 2017]: Regular bar expressions and Regular Nondeterministic Nominal Automata (RNNA), using nominal sets

- RNNA inclusion checking is in $\text{ExpSpace}$

**Our progress here:** A linear-time fixpoint logic for RNNA

- Define Logic
- Introduce extended RNNA (ERNNA)
- Translate formulae to equivalent ERNNA
- Generalize RNNA inclusion checking to ERNNA and obtain $2\text{ExpTime}$ model checking and validity checking
Fix countable set $\mathbb{A}$ of names, $G$: group of fin. permutations on $\mathbb{A}$

### Nominal sets

**Action of $G$ on set $X$:**

$\cdot : G \times X \to X$ s.t. for all $x \in X$, $\pi, \pi' \in G$

- $\text{id} \cdot x = x$
- $\pi \cdot (\pi' \cdot x) = (\pi\pi') \cdot x$

Given set $X$, set $S \subseteq \mathbb{A}$ is a **support** of $x \in X$ if $\pi(x) = x$ for all $\pi \in G$ such that $\pi(a) = a$ for all $a \in S$

**Nominal set:** $(X, \cdot)$ s.t. all $x \in X$ have min. finite support $\text{supp}(x)$

**Orbit of $x \in X$:** $\{\pi \cdot x \mid \pi \in G\}$
Bar Strings

Abstraction set: \([\mathbb{A}]X = (\mathbb{A} \times X)/\sim\) where

\[(a, x) \sim (b, y)\] if and only if \((ac) \cdot x = (bc) \cdot y\) for any fresh \(c\)

\(\langle a \rangle x\): \(\sim\)-equivalence class of \((a, x)\)

Bar strings

Set of finite bar strings: \(\overline{\mathbb{A}}^*\) where \(\overline{\mathbb{A}} = \mathbb{A} \cup \{ |a | a \in \mathbb{A} \}\)

\(\equiv_\alpha\) on bar strings: equivalence generated by

\[w|av \equiv_\alpha w|bu\] iff \(\langle a \rangle v = \langle b \rangle u\) in \([\mathbb{A}]\overline{\mathbb{A}}^*\)

E.g. \(|a|bb \equiv_\alpha |a|aa \nmid_\alpha |a|ba\)

Bar languages: subsets of \(\overline{\mathbb{A}}^*/\equiv_\alpha\)
Data languages from bar languages

Put $ub(a) = ub(|a|) = a$, extend $ub$ to bar strings

Given bar language $L$, put

$$N(L) = \{ub(w) \mid [w]_\alpha \in L, w \text{ clean}\}$$

no name bound twice

$$D(L) = \{ub(w) \mid [w]_\alpha \in L\}$$

global freshness semantics

local freshness semantics

E.g.

$$D(|a|ba) = N(|a|ba) = \{aba \mid a, b \in A, a \neq b\},$$

while

$$N(|a|b) = \{ab \mid a, b \in A, a \neq b\} \quad \text{but} \quad D(|a|b) = \{ab \mid a, b \in A\}$$
A Linear-time Logic for Bar Strings

Syntax of Bar-$\mu$TL

\[ \varphi, \psi ::= \epsilon \mid \neg \varphi \mid \varphi \land \psi \mid \Diamond_a \varphi \mid \Diamond_I \varphi \mid X \mid \mu X. \varphi \ (a \in A, X \in V) \]

requiring positivity and guardedness of fixpoint variables

Put \[ \square_\sigma \psi := \neg \Diamond_\sigma \neg \psi \] for \( \sigma \in \overline{A} \)

Define \( \equiv_\alpha \) on formulae, e.g. \[ \Diamond_I a (\Diamond_I a \epsilon \lor \square_b \neg \epsilon) \equiv_\alpha \Diamond_I c (\Diamond_I c \epsilon \lor \square_b \neg \epsilon) \]
Semantics of Bar-$\mu$TL

Interpret in context $S \subseteq A$ over bar strings $w$ s.t. $\text{FN}(w) \subseteq S$:

$S, w \models \epsilon \iff w = \epsilon$

$S, w \models \mu X. \varphi \iff S, w \models \varphi[X/\mu X. \varphi]$

$S, w \models \Diamond a \varphi \iff w = av$ and $S, v \models \varphi$

$S, w \models \Diamond_{\alpha} a \varphi \iff \exists b \in A, v \in \overline{A}^*, \psi. w \equiv_{\alpha} bv,$

$$\langle a \rangle \varphi = \langle b \rangle \psi \text{ and } S \cup \{b\}, v \models \psi$$

Put $[\varphi] = \{ w \in \overline{A}^* | \emptyset, w \models \varphi \} / \equiv_{\alpha}$
Set $S \subseteq X$ is equivariant if $\pi \cdot x \in S$ for all $\pi \in G$, $x \in S$. 
Set $S \subseteq X$ is equivariant if $\pi \cdot x \in S$ for all $\pi \in G$, $x \in S$

**Extended Regular Nondeterministic Nominal Automata (ERNNA)**

$A = (Q, \rightarrow, s, f)$ with
- orbit-finite nominal set $Q$ of states, initial state $s \in Q$
- equivariant transition relation $\rightarrow \subseteq Q \times (\overline{A} \cup \epsilon) \times Q$
- equivariant acceptance function $f : Q \rightarrow \{0, 1, \top\}$

such that $q \xrightarrow{a} q'$ and $\langle a \rangle q' = \langle b \rangle q''$ imply $q \xrightarrow{b} q''$ ($\alpha$-invariance)

and such that $\{(a, q') \mid q \xrightarrow{a} q'\}$ and $\{\langle a \rangle q' \mid q \xrightarrow{a} q'\}$ are finite.

**Degree of $A$:** maximal size of support of some state $q \in Q$
Definition (ERNNA acceptance)

Bar string $w \in \overline{A}^*$ is **accepted** by $A = (Q, \rightarrow, s, f)$ if

- $\exists q \in Q. s \xrightarrow{w} q$ and $f(q) = 1$, or
- $\exists q \in Q$, prefix $u$ of $w. s \xrightarrow{u} q$ and $f(q) = \top$.

Literal acceptance: $L_0(A) = \{w \in \overline{A}^* \mid A \text{ accepts } w\}$

Accepted bar language: $L_\alpha(A) = L_0/\equiv_\alpha$
\( s() \) accepts \( |a|bb \) but not \( |a|aa \)
**ERNNA, Example**

\[ s() \text{ accepts } |a|bb \text{ but not } |a|aa \]

\[ x() \text{ accepts both } |a|bb \text{ and } |a|aa \]
Lemma [following Schröder et al. 2017]

For all ERNNAs $A$ of degree $k$ and with $n$ orbits, there is ERNNA $\text{nd}(A)$ of degree $k + 1$ and with $n \cdot 2^{k+1}$ orbits, s.t.

1. $L_\alpha(A) = L_\alpha(\text{nd}(A))$ and
2. $L_0(\text{nd}(A))$ is closed under $\alpha$-equivalence of bar strings.

Corollary [following Schröder et al. 2017]

Inclusion checking for ERNNAs is in $\text{ExpSpace} / \text{para-PSPACE}$. 
Translating Formulae to ERNNA

### Problem

Let \( \varphi(b) = \mu Y. (\Box_b \bot \land \Box_c Y) \) and \( \psi = \mu X. (\Box_{|a} X \land \Box_{|b} \varphi(b)) \).

To check \(|a_1|a_2 \ldots |a_n|a_i|^+ \models \psi\), have to check \(a_i^+ \models \varphi(a_n)\) for all \(n\).

Solution: use nondeterminism to guess relevant letter \(a_i\), keep just one copy \(\varphi(a_i)\) of \(\varphi(_)\).

Further complication: Elimination of \(\Box\)-formulae.

Given \(\varphi\) of size \(n\) and degree \(k\), define **formula automaton** \(A(\varphi)\).

### Theorem

We have \(L_\alpha(A(\varphi)) = [\varphi]\) and \(A(\varphi)\) has \(2^O(n^2 \cdot 2^k)\) orbits.
Model Checking

Input: RNNA $M$, formula $\varphi$ of size $n$ and degree $k$

- Model checking: check whether

$$L_\alpha(M) \subseteq [\varphi] = L_\alpha(A(\varphi)) = L_\alpha(nd(A(\varphi)))$$

formulae are ERNNA

name dropping construction

- $A(\varphi)$ has at most $2^{O(n^2 \cdot 2^k)}$ orbits,
$nd(A(\varphi))$ has at most $2^{k+1} \cdot 2^{O(n^2 \cdot 2^k)}$ orbits

Theorem

Model checking and validity checking for Bar$\mu$TL are in $2\text{ExpSpace}$ (para-$\text{ExpSpace}$ with parameter $k$). Satisfiability checking is in $\text{ExpSpace}$ (para-$\text{PSPACE}$ with parameter $k$).
Conclusion

Results

- Linear-time logic for finite bar strings
- Extended regular nominal automata (ERNNA)
  - inclusion checking for ERNNA in $\text{ExpSpace}$
- Non-trivial translation of formulae into ERNNA, removing universal branching by nondeterminism
- Model / validity checking over RNNA is elementary!
  (in $2\text{ExpSpace}$ and para-$\text{ExpSpace}$)

Future work:

- Extend this to infinite bar strings (nominal Büchi automata, see [Urbat, H, Milius, Schröder, CONCUR 2021])
- Is translation from formulae to RNNA a nondeterminisation procedure for alternating nominal automata?