Harnessing LTL With Freeze Quantification

Daniel Hausmann, Stefan Milius and Lutz Schröder

University Erlangen-Nürnberg, Germany

Oberseminar

4 August 2020

Model Checking for Data Languages

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

Basic linear-time model checking principle:

Transform φ to automaton $A(\varphi)$, check inclusion of model in $A(\varphi)$

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

- Register Automata (RA) (Kaminski et al. 1994) undecidable
- Nondeterministic Orbit-finite Automata (NOFA) (Neven et al. 2004, Boyańczyk 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 $\downarrow_r \varphi$: "$d_i \to r; \varphi$", $\uparrow_r$: "$d_i = r$?"

Flat Freeze LTL (Bollig et al. 2019):

- for all subformulae $\phi_1 \, \mathsf{U} \, \phi_2$, no freeze operator in ϕ_1

Model Checking for Freeze LTL:

- Freeze LTL over RA (Demri, Lazić, 2007) — undecidable
- Flat Freeze LTL over OCA (Bollig et al. 2019) — NExpTime

OCA: One-Counter Automata

Model Checking for Bar Strings

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

- RNNA inclusion checking is in para-PSPACE

Our aim here: Linear-time fixpoint logic for RNNA

- Introduce alternating nominal automata (ANA)
- Transform formulae to equivalent ANA
- Generalize RNNA inclusion checking to ANA inclusion checking to obtain decidable model checking

Nominal Sets

G-sets

G-set for group G: $(X, \cdot : G \times X \to X)$ such that

$$\pi \cdot (\rho \cdot x) = (\pi\rho) \cdot x \qquad\qquad 1 \cdot x = x$$

For $x \in X$, $Y \subseteq X$, put

$$\mathsf{fix}\, x = \{\pi \in G \mid \pi \cdot x = x\} \qquad\qquad \mathsf{Fix}\, Y = \bigcap_{x \in Y} \mathsf{fix}\, x$$

$x \in X$ has finite support if there is finite set $Y \subseteq X$ such that

$$\mathsf{Fix}(Y) \subseteq \mathsf{fix}(x)$$

Then let $\mathsf{supp}(x)$ denote least supporting set

G-sets

G-set for group G: $(X, \cdot : G \times X \to X)$ such that

$$\pi \cdot (\rho \cdot x) = (\pi\rho) \cdot x \qquad\qquad 1 \cdot x = x$$

For $x \in X$, $Y \subseteq X$, put

$$\mathsf{fix}\, x = \{\pi \in G \mid \pi \cdot x = x\} \qquad\qquad \mathsf{Fix}\, Y = \bigcap_{x \in Y} \mathsf{fix}\, x$$

$x \in X$ has finite support if there is finite set $Y \subseteq X$ such that

$$\mathsf{Fix}(Y) \subseteq \mathsf{fix}(x)$$

Then let $\mathsf{supp}(x)$ denote least supporting set

Names, permutations

Fix countable set $\mathbb{A}$ of names, G: group of fin. permutations on $\mathbb{A}$

Then $(\mathbb{A}, \cdot : G \times \mathbb{A} \to \mathbb{A})$ with $\pi \cdot a = \pi(a)$ is a G-set

Nominal Sets, ctd.

Nominal sets

Nominal set X: G-set $(X, \cdot)$ s.t. all $x \in X$ have finite support

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

$$(a, x) \sim (b, y) \text{ if and only if } (ac) \cdot x = (bc) \cdot y \text{ for any fresh } c$$

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

Nominal Sets, ctd.

Nominal sets

Nominal set X: G-set $(X, \cdot)$ s.t. all $x \in X$ have finite support

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

$$(a, x) \sim (b, y) \text{ if and only if } (ac) \cdot x = (bc) \cdot y \text{ for any fresh } c$$

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

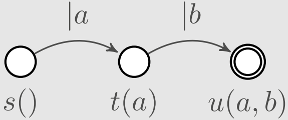
Bar strings

Set of finite bar strings: $\mathbb{B} = \mathsf{B}^*$ where $\mathsf{B} = \mathsf{A} \cup \{|a \mid a \in \mathsf{A}\}$

$\equiv_\alpha$ on bar strings: equivalence generated by

$$w|av \equiv_\alpha w|bu \text{ iff } \langle a \rangle v = \langle b \rangle u \text{ in } [\mathsf{A}]\mathbb{B}$$

RNNA by Example



$s()$ accepts e.g. $|a|b$ and $|b|a$ but does not accept $|a|a$

A Linear-time Logic for RNNA

Syntax

$$\varphi, \psi ::= \top \mid \epsilon \mid \neg\varphi \mid \varphi \wedge \psi \mid \Diamond_a \varphi \mid \Diamond_{|a} \varphi \mid X \mid \mu X.\varphi \qquad (a \in \mathsf{A}, X \in \mathsf{V})$$

requiring positivity of fixpoint variables

Define $\equiv_\alpha$ on formulae, e.g. $\Diamond_{|a}(\Diamond_a \top \vee \Box_b \top) \equiv_\alpha \Diamond_{|c}(\Diamond_c \top \vee \Box_b \top)$

A Linear-time Logic for RNNA

Syntax

$$\varphi, \psi ::= \top \mid \epsilon \mid \neg\varphi \mid \varphi \wedge \psi \mid \Diamond_a \varphi \mid \Diamond_{|a} \varphi \mid X \mid \mu X.\varphi$$

$$(a \in \mathsf{A}, X \in \mathsf{V})$$

requiring positivity of fixpoint variables

Define $\equiv_\alpha$ on formulae, e.g. $\Diamond_{|a}(\Diamond_a \top \vee \Box_b \top) \equiv_\alpha \Diamond_{|c}(\Diamond_c \top \vee \Box_b \top)$

Semantics (attempt)

Interpret formulae over bar strings using $\sigma : \mathsf{V} \rightharpoonup \mathcal{P}(\mathbb{B})$:

$$\begin{aligned} [\![\epsilon]\!]_\sigma &= \{\epsilon\} \\ [\![\Diamond_a \varphi]\!]_\sigma &= \{w \in \mathbb{B} \mid w = av, v \in [\![\varphi]\!]_\sigma\} \\ [\![\Diamond_{|a} \varphi]\!]_\sigma &= \{w \in \mathbb{B} \mid w = |bv, \exists \psi.\, \Diamond_{|a}\varphi \equiv_\alpha \Diamond_{|b}\psi, v \in [\![\psi]\!]_\sigma\} \end{aligned}$$

A Linear-time Logic for RNNA, Example

Recall: $[\![\Diamond_a \varphi]\!]_\sigma = \{w \in \mathbb{B} \mid w = av, v \in [\![\varphi]\!]_\sigma\}$

$[\![\Diamond_{|a} \varphi]\!]_\sigma = \{w \in \mathbb{B} \mid w = |bv, \exists \psi.\, \Diamond_{|a}\varphi \equiv_\alpha \Diamond_{|b}\psi, v \in [\![\psi]\!]_\sigma\}$

A Linear-time Logic for RNNA, Example

Recall: $[\![\Diamond_a \varphi]\!]_\sigma = \{w \in \mathbb{B} \mid w = av, v \in [\![\varphi]\!]_\sigma\}$

$[\![\Diamond_{|a} \varphi]\!]_\sigma = \{w \in \mathbb{B} \mid w = |bv, \exists \psi.\, \Diamond_{|a}\varphi \equiv_\alpha \Diamond_{|b}\psi, v \in [\![\psi]\!]_\sigma\}$

Let $\varphi = \mu X.\, \psi$, $\psi = \Diamond_{|a}(\Diamond_a \top \vee \Box_b X)$ so that $\mathsf{FN}(\varphi) = b$

A Linear-time Logic for RNNA, Example

Recall: $[\![\Diamond_a \varphi]\!]_\sigma = \{w \in \mathbb{B} \mid w = av, v \in [\![\varphi]\!]_\sigma\}$

$[\![\Diamond_{|a} \varphi]\!]_\sigma = \{w \in \mathbb{B} \mid w = |bv, \exists \psi.\, \Diamond_{|a}\varphi \equiv_\alpha \Diamond_{|b}\psi, v \in [\![\psi]\!]_\sigma\}$

Let $\varphi = \mu X.\, \psi$, $\psi = \Diamond_{|a}(\Diamond_a \top \vee \Box_b X)$ so that $\mathsf{FN}(\varphi) = b$

We have e.g.

- $|cc \in [\![\varphi]\!]$ since $\psi \equiv_\alpha \Diamond_{|c}(\Diamond_c \top \vee \Box_b X)$ and $c \in [\![\Diamond_c \top \vee \Box_b X]\!]$
- $|c|dc \notin [\![\varphi]\!]$ since $|dc \notin [\![\Diamond_c \top \vee \Box_b X]\!]$ since $c \notin [\![\phi]\!]$
- $|cb|db|ee \in [\![\varphi]\!]$ since $|ee \in [\![\psi]\!]$ so that $b|ee \in [\![\Box_b \psi]\!]$, $|db|ee \in [\![\Diamond_{|a}(\Diamond_a \top \vee \Box_b \psi)]\!]$, $b|db|ee \in [\![\Box_b(\Diamond_{|a}(\Diamond_a \top \vee \Box_b \psi))]\!]$

Ensuring Monotonicity

Modal operators are not monotone (yet)!

e.g. $|bb \in [\![\Diamond_{|a}(\Diamond_a\top \vee \bot)]\!]$ since $b \in [\![\Diamond_b\top \vee \bot]\!]$ but
$|bb \notin [\![\Diamond_{|a}(\Diamond_a\top \vee \Diamond_b\top)]\!]$ since $\forall\chi.\Diamond_{|a}(\Diamond_a\top \vee \Diamond_b\top) \not\equiv_\alpha \Diamond_{|b}\chi$

Ensuring Monotonicity

Modal operators are not monotone (yet)!

e.g. $|bb \in [\![\Diamond_{|a}(\Diamond_a \top \vee \bot)]\!]$ since $b \in [\![\Diamond_b \top \vee \bot]\!]$ but
$|bb \notin [\![\Diamond_{|a}(\Diamond_a \top \vee \Diamond_b \top)]\!]$ since $\forall \chi. \Diamond_{|a}(\Diamond_a \top \vee \Diamond_b \top) \not\equiv_\alpha \Diamond_{|b}\chi$

Stepping stone: alternative semantics $[\![-]\!]'_\sigma$, closed under $\equiv_\alpha$

$$[\![\Diamond_{|a}\varphi]\!]'_\sigma = \{w \in \mathbb{B} \mid w \equiv_\alpha |bv, \exists \psi. \Diamond_{|a}\varphi \equiv_\alpha \Diamond_{|b}\varphi, v \in [\![\psi]\!]'_\sigma\}$$

Ensuring Monotonicity

Modal operators are not monotone (yet)!

e.g. $|bb \in [\![\Diamond_{|a}(\Diamond_a\top \vee \bot)]\!]$ since $b \in [\![\Diamond_b\top \vee \bot]\!]$ but
$|bb \notin [\![\Diamond_{|a}(\Diamond_a\top \vee \Diamond_b\top)]\!]$ since $\forall\chi.\Diamond_{|a}(\Diamond_a\top \vee \Diamond_b\top) \not\equiv_\alpha \Diamond_{|b}\chi$

Stepping stone: alternative semantics $[\![-]\!]'_\sigma$, closed under $\equiv_\alpha$

$$[\![\Diamond_{|a}\varphi]\!]'_\sigma = \{w \in \mathbb{B} \mid w \equiv_\alpha |bv, \exists\psi.\, \Diamond_{|a}\varphi \equiv_\alpha \Diamond_{|b}\varphi, v \in [\![\psi]\!]'_\sigma\}$$

Then $|bb \in [\![\Diamond_{|a}(\Diamond_a\top \vee \bot)]\!]'$ since $b \in [\![\Diamond_b\top \vee \bot]\!]'$ and
$|bb \in [\![\Diamond_{|a}(\Diamond_a\top \vee \Diamond_b\top)]\!]'$ since $|bb \equiv_\alpha |cc$, $c \in [\![\Diamond_c\top \vee \Diamond_b\top]\!]'$

Ensuring Monotonicity

Modal operators are not monotone (yet)!

e.g. $|bb \in [\![\Diamond_{|a}(\Diamond_a \top \vee \bot)]\!]$ since $b \in [\![\Diamond_b \top \vee \bot]\!]$ but
$|bb \notin [\![\Diamond_{|a}(\Diamond_a \top \vee \Diamond_b \top)]\!]$ since $\forall \chi. \Diamond_{|a}(\Diamond_a \top \vee \Diamond_b \top) \not\equiv_\alpha \Diamond_{|b}\chi$

Stepping stone: alternative semantics $[\![-]\!]'_\sigma$, closed under $\equiv_\alpha$

$$[\![\Diamond_{|a}\varphi]\!]'_\sigma = \{w \in \mathbb{B} \mid w \equiv_\alpha |bv, \exists \psi.\, \Diamond_{|a}\varphi \equiv_\alpha \Diamond_{|b}\varphi, v \in [\![\psi]\!]'_\sigma\}$$

Then $|bb \in [\![\Diamond_{|a}(\Diamond_a \top \vee \bot)]\!]'$ since $b \in [\![\Diamond_b \top \vee \bot]\!]'$ and
$|bb \in [\![\Diamond_{|a}(\Diamond_a \top \vee \Diamond_b \top)]\!]'$ since $|bb \equiv_\alpha |cc$, $c \in [\![\Diamond_c \top \vee \Diamond_b \top]\!]'$

$\rightsquigarrow$ Semantics well-defined but does not match RNNA (yet)!

Name Dropping Formulae

Idea: incorporate explicit name "losing" ("forgetting") in formulae

Abbreviate

$$\mathsf{choice}(S, a, \psi) = \mathsf{nd}(S \setminus \{a\}, \psi) \vee \mathsf{nd}(S \cup \{a\}, \psi)$$

Put $\mathsf{nd}(\varphi) = \bigvee_{S \subseteq \mathsf{FN}(\varphi)} \mathsf{nd}(S, \varphi)$ where $\mathsf{nd}(S, \varphi)$ is defined by

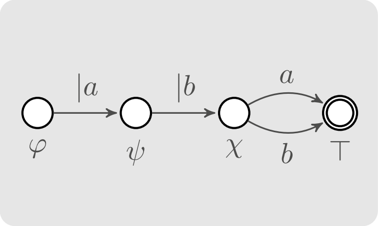
$$\mathsf{nd}(S, \Diamond_a \psi) = \begin{cases} \Diamond_a(\mathsf{choice}(S, a, \psi)) & a \in S \\ \bot & a \notin S \end{cases}$$

$$\mathsf{nd}(S, \Diamond_{|a} \psi) = \Diamond_{|a}(\mathsf{choice}(S, a, \psi))$$

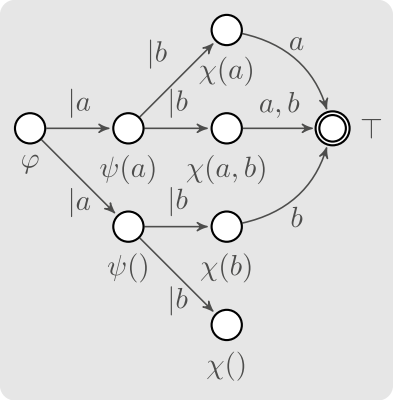
plus commutation with non-modal operators

Name dropping formulae, Example

Let $\varphi = \Diamond_{|a}\psi$, $\psi = \Diamond_{|b}\chi$, $\chi = \Diamond_a\top \vee \Diamond_b\top$



$\rightsquigarrow$



$|a|bb \in [\![\phi]\!]$

$|a|aa \notin [\![\phi]\!]$

$|a|bb \in [\![\mathsf{nd}(\phi)]\!]$

$|a|aa \in [\![\mathsf{nd}(\phi)]\!]$

Name Dropping Formulae, Results

Lemma

For all formulae φ, we have $[\![\varphi]\!]' = [\![\mathsf{nd}(\varphi)]\!]' = [\![\mathsf{nd}(\varphi)]\!]$.

Define degree $\mathsf{deg}(\varphi) = \max\{|\mathsf{FN}(\psi)| \mid \psi \text{ is subformula of } \varphi\}$,
closure $\mathsf{cl}(\varphi)$ (can be seen as syntax graph of φ)

Lemma

For all formulae φ such that $\mathsf{deg}(\varphi) = k$, $|\mathsf{cl}(\varphi)| = n$, we have

$$|\mathsf{cl}(\mathsf{nd}(\varphi))| \leq 2^{k+1} n.$$

For nominal set X, the orbit of $x \in X$ is $\{\pi \cdot x \mid \pi \in G\}$ and $S \subseteq X$ is equivariant if $\pi \cdot x \in S$ for all $\pi \in G$, $x \in S$

Alternating Nominal Automata

For nominal set X, the orbit of $x \in X$ is $\{\pi \cdot x \mid \pi \in G\}$ and $S \subseteq X$ is equivariant if $\pi \cdot x \in S$ for all $\pi \in G$, $x \in S$

Definition (Alternating nominal automaton (ANA))

$A = (Q_\exists, Q_\forall, \to, s, F)$ with

- orbit-finite nominal set $Q = Q_\exists \sqcup Q_\forall$ of states
- equivariant transition relation $\to \subseteq Q \times (\mathsf{B} \cup \epsilon) \times Q$
- equivariant set F of accepting states

such that $q \xrightarrow{|a} q'$ and $\langle a \rangle q' = \langle b \rangle q''$ imply $q \xrightarrow{|b} q''$ (α-invariance) and such that $\{(a, q') \mid q \xrightarrow{a} q'\}$ and $\{\langle a \rangle q' \mid q \xrightarrow{|a} q'\}$ are finite.

Alternating Nominal Automata, acceptance

Runs of ANA $A = (Q_\exists, Q_\forall, \to, s, F)$ are trees labelled with states, not sequences of states

Definition (Accepting run trees)

A run tree for $w \in \mathbb{B}$ is accepting if its branching follows $\to$ along w and adheres $Q_\exists$ and $Q_\forall$, and all its leaves have labels from F.

Definition (Accepted language)

Literal acceptance:

$$L_0(A) = \{w \in \mathbb{B} \mid \text{there is an accepting run tree of } A \text{ for } w\}$$

Accepted bar language:

$$L_\alpha(A) = L_0/{\equiv_\alpha}$$

Formulae as Automata

Given φ, define formula automaton $\mathsf{A}(\varphi) = (Q_\exists, Q_\forall, \to, s, F)$:

- $Q = \{\pi\psi \mid \psi \in \mathsf{cl}(\varphi), \pi \in G\}$ with obvious kind ($Q_\exists$ or $Q_\forall$)
- $s = \varphi$, $F = \{\top, \neg\epsilon\}$

$$\phi \wedge \psi \xrightarrow{\epsilon} \phi \qquad \phi \wedge \psi \xrightarrow{\epsilon} \psi$$

$$\phi \vee \psi \xrightarrow{\epsilon} \phi \qquad \phi \vee \psi \xrightarrow{\epsilon} \psi$$

$$\mu X.\,\phi \xrightarrow{\epsilon} \phi[\mu X.\,\phi/X] \qquad \nu X.\,\phi \xrightarrow{\epsilon} \phi[\nu X.\,\phi/X]$$

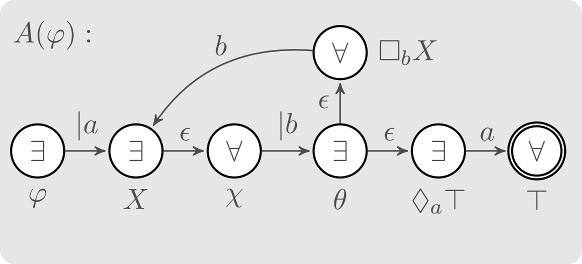
$$\Diamond_a\phi \xrightarrow{a} \phi \qquad \Box_a\phi \xrightarrow{a} \phi$$

$$\Diamond_{|a}\phi \xrightarrow{|b} \chi \quad \langle a\rangle\phi = \langle b\rangle\chi \qquad \Box_{|a}\phi \xrightarrow{|b} \chi \quad \langle a\rangle\phi = \langle b\rangle\chi$$

Lemma

For monotone φ, we have $L_\alpha(A(\varphi)) = [\![\varphi]\!]$.

Let $\varphi = \Diamond_{|a}\psi$, $\psi = \mu X.\, \chi$, $\chi = \Box_{|b}\theta$, $\theta = \Diamond_a \top \vee \Box_b X$



Model Checking

bar NFA: NFA M with alphabet B; $L_\alpha(M) = L_0(M)/{\equiv_\alpha}$

Definition (satisfaction over bar NFA)

For monotone φ,

$$M \models \varphi \text{ if and only if } L_\alpha(M) \subseteq [\![\varphi]\!]'$$

Model checking: Given M and φ, check whether

$$L_\alpha(M) \subseteq [\![\varphi]\!]' = [\![\mathsf{nd}(\varphi)]\!] = L_\alpha(A(\mathsf{nd}(\varphi)))$$

name dropping construction   formulae are ANA

Language Inclusion Checking

Given: bar NFA M, ANA A

Nondeterministic Algorithm (check whether $L_\alpha(M) \not\subseteq L_\alpha(A)$)

1. Initialize $q = q_0$, $\Phi = \{\{q'_0\}\}$.
2. If $\emptyset \in \Phi$, abort. If q is accepting, guess whether word ends now. If it ends, terminate positively if all $\Gamma \in \Phi$ contain non-accepting state.
3. Guess α and q' s.t. $q \xrightarrow{\alpha} q'$ in M. Put $\Phi := \bigcup_{\Gamma \in \Phi}(\mathsf{succ}(\Gamma, \alpha))$,
$\mathsf{succ}(\psi, \alpha) = \begin{cases} \{\{\chi \mid \psi \xrightarrow{\alpha} \chi \text{ in } A\}\} & \psi \in Q_\forall \\ \{\{\chi\} \mid \psi \xrightarrow{\alpha} \chi \text{ in } A\} & \psi \in Q_\exists \end{cases}$. Goto 2.

Language Inclusion Checking

Given: bar NFA M, ANA A

Nondeterministic Algorithm (check whether $L_\alpha(M) \not\subseteq L_\alpha(A)$)

1. Initialize $q = q_0$, $\Phi = \{\{q_0'\}\}$.
2. If $\emptyset \in \Phi$, abort. If q is accepting, guess whether word ends now. If it ends, terminate positively if all $\Gamma \in \Phi$ contain non-accepting state.
3. Guess α and q' s.t. $q \xrightarrow{\alpha} q'$ in M. Put $\Phi := \bigcup_{\Gamma \in \Phi}(\text{succ}(\Gamma, \alpha))$,
   $$\text{succ}(\psi, \alpha) = \begin{cases} \{\{\chi \mid \psi \xrightarrow{\alpha} \chi \text{ in } A\}\} & \psi \in Q_\forall \\ \{\{\chi\} \mid \psi \xrightarrow{\alpha} \chi \text{ in } A\} & \psi \in Q_\exists \end{cases}.$$ Goto 2.

Lemma

The inclusion problem is in ExpSpace.

(complement and nondeterminism do not affect space complexity)

Model Checking and Satisfiability Checking

Ingredients:

- Model checking: Given bar NFA M and φ, check whether

$$L_\alpha(M) \subseteq L_\alpha(A(\mathsf{nd}(\varphi)))$$

- Validity checking: Given universal RNNA $M_\top$ and φ, check

$$L_\alpha(M_\top) \subseteq L_\alpha(A(\mathsf{nd}(\varphi)))$$

- We have $|\mathsf{cl}(\mathsf{nd}(\varphi))| \leq 2^{k+1}n$
- Inclusion problem is in $\textsc{ExpSpace}$ $(\mathcal{O}(|M| + 2^{|\mathsf{cl}(\mathsf{nd}(\varphi))|}))$

Corollary

The model checking and validity problems are in $2\textsc{ExpSpace}$ (and in para-$\textsc{ExpSpace}$ with k as parameter).

Conclusion

Results so far:

- Linear-time logic for finite bar strings
  - name-dropping construction on formulae, blow-up: $2^{k+1}n$
- Alternating nominal automata (ANA), generalizing RNNA
- Name-dropping formulae are ANA
- Model / satisfiability checking over bar NFA is elementary!
  (in $2\textsc{ExpSpace}$ and para-$\textsc{ExpSpace}$)

Future work:

- How about infinite bar strings? (nominal Büchi automata)
- Conjecture: Inclusion checking between ANA is elementary