Friedrich-Alexander-Universität
Erlangen-Nürnberg

Nominal Alternating Parity Automata

Extending Regular Alternating Nominal Automata for Infinitely Bar Strings

Max Ole Elliger   Florian Frank   Stefan Milius   Lutz Schröder

January 13, 2026

Motivation

- Automata with name binding (e.g. RNNA's [Sch+17], RANA's [Fra+25]) have been introduced to accept literal/bar/data languages over infinite alphabets.
- RANA's provide full alternation and have a corresponding lineartime logic Bar-μTL.
- Büchi RNNA's [Urb+21] extend RNNA's for words of infinite length.
- All three automata models come with decidable inclusion and emptiness problems.

Motivation

- Automata with name binding (e.g. RNNA's [Sch+17], RANA's [Fra+25]) have been introduced to accept literal/bar/data languages over infinite alphabets.
- RANA's provide full alternation and have a corresponding lineartime logic Bar-μTL.
- Büchi RNNA's [Urb+21] extend RNNA's for words of infinite length.
- All three automata models come with decidable inclusion and emptiness problems.
- Can we extend RANA's for words of infinite length with similar results?

Nominal Sets [Pit13]

Intuition

- Intuitively, a *nominal set* is a set X whose elements $x \in X$ depend on a finite subset $\mathrm{supp}(x) \subseteq \mathbb{A}$ of names: $\pi \cdot x = x$ if π fixes all $a \in \mathrm{supp}(x)$.
- A nominal set is equipped with a permutation action $\cdot \colon \mathrm{Perm}(\mathbb{A}) \times X \to X$ to allow renamings.

Nominal Sets [Pit13]

Intuition

- Intuitively, a *nominal set* is a set X whose elements $x \in X$ depend on a finite subset $\mathrm{supp}(x) \subseteq \mathbb{A}$ of names: $\pi \cdot x = x$ if π fixes all $a \in \mathrm{supp}(x)$.
- A nominal set is equipped with a permutation action $\cdot \colon \mathrm{Perm}(\mathbb{A}) \times X \to X$ to allow renamings.

Example

- FO-Formulae: $\mathrm{supp}(\forall x.\, P(x, y)) = \{x, y\}$
- FO-Formulae modulo α-equivalence: $\mathrm{supp}(\forall x.\, P(x, y)) = \{y\}$
- Finitely supported functions together with the (pointwise) group action $(\pi \cdot f)(x) := \pi^{-1} \cdot f(\pi \cdot x)$

Nominal Sets [Pit13]

Intuition

- Intuitively, a *nominal set* is a set X whose elements $x \in X$ depend on a finite subset $\operatorname{supp}(x) \subseteq \mathbb{A}$ of names: $\pi \cdot x = x$ if π fixes all $a \in \operatorname{supp}(x)$.
- A nominal set is equipped with a permutation action $\cdot \colon \operatorname{Perm}(\mathbb{A}) \times X \to X$ to allow renamings.

Example

- FO-Formulae: $\operatorname{supp}(\forall x.\, P(x, y)) = \{x, y\}$
- FO-Formulae modulo α-equivalence: $\operatorname{supp}(\forall x.\, P(x, y)) = \{y\}$
- Finitely supported functions together with the (pointwise) group action $(\pi \cdot f)(x) := \pi^{-1} \cdot f(\pi \cdot x)$

- An object $x \in X$ is *equivariant*, if $\operatorname{supp}(x) = \emptyset$.
- Nominal sets form a category together with equivariant functions $f \colon X \to X$.

Nominal Sets [Pit13]
Name Abstraction

- Equivalence Relation $\sim_\alpha \subseteq (\mathbb{A} \times X) \times (\mathbb{A} \times X)$ where $(a, x) \sim_\alpha (b, y) :\Longleftrightarrow \exists c \notin \operatorname{supp}(a, b, x, y).\, (a\ c) \cdot x = (b\ c) \cdot y$.

Example

$$(x, \forall x.\, P(x, y)) \sim_\alpha (z, \forall z.\, P(z, y))$$
$$(x, \forall x.\, P(x, y)) \not\sim_\alpha (y, \forall y.\, P(y, y))$$

- Equivalence Classes $\langle a \rangle x := \{(b, y) \mid (b, y) \sim_\alpha (a, x)\}$
- Abstraction Functor $[\mathbb{A}]X := \{\langle a \rangle x \mid a \in \mathbb{A}, x \in X\}$, defined on equivariant functions via $([\mathbb{A}]f(\langle a \rangle x) := \langle a \rangle f(x)$.

Bar Strings [Sch+17]

- Duplicate name set in order to introduce binders: $\bar{\mathbb{A}} := \mathbb{A} \cup \{|a \mid a \in \mathbb{A}\}$
- Finite bar strings are words over $\bar{\mathbb{A}}$ and form a nominal set $\bar{\mathbb{A}}^*$.

Bar Strings [Sch+17]

- Duplicate name set in order to introduce binders: $\bar{\mathbb{A}} := \mathbb{A} \cup \{|a \mid a \in \mathbb{A}\}$
- Finite bar strings are words over $\bar{\mathbb{A}}$ and form a nominal set $\bar{\mathbb{A}}^*$.
- What about the set of infinite bar strings $w \in \bar{\mathbb{A}}^\omega$?
- Only finitely supported infinite bar strings form a nominal set $\bar{\mathbb{A}}^\omega_{\mathsf{fs}}$.

Bar Strings [Sch+17]

- Duplicate name set in order to introduce binders: $\bar{\mathbb{A}} := \mathbb{A} \cup \{|a \mid a \in \mathbb{A}\}$
- Finite bar strings are words over $\bar{\mathbb{A}}$ and form a nominal set $\bar{\mathbb{A}}^*$.
- What about the set of infinite bar strings $w \in \bar{\mathbb{A}}^\omega$?
- Only finitely supported infinite bar strings form a nominal set $\bar{\mathbb{A}}^\omega_{\mathsf{fs}}$.
- α-equivalence $\equiv_\alpha$ on finite bar strings is generated by $w|au \equiv_\alpha w|bv$ where $\langle a \rangle u = \langle b \rangle v$.
- Two infinite bar strings $w, v \in \bar{\mathbb{A}}^\omega$ are α-equivalent, if all of their finite prefixes are α-equivalent.

Example

$$|a|a|a \ldots \equiv_\alpha |b|b|b \ldots$$

Bar Strings [Sch+17]

- Duplicate name set in order to introduce binders: $\bar{\mathbb{A}} := \mathbb{A} \cup \{|a \mid a \in \mathbb{A}\}$
- Finite bar strings are words over $\bar{\mathbb{A}}$ and form a nominal set $\bar{\mathbb{A}}^*$.
- What about the set of infinite bar strings $w \in \bar{\mathbb{A}}^\omega$?
- Only finitely supported infinite bar strings form a nominal set $\bar{\mathbb{A}}^\omega_{\mathsf{fs}}$.
- α-equivalence $\equiv_\alpha$ on finite bar strings is generated by $w|au \equiv_\alpha w|bv$ where $\langle a\rangle u = \langle b\rangle v$.
- Two infinite bar strings $w, v \in \bar{\mathbb{A}}^\omega$ are α-equivalent, if all of their finite prefixes are α-equivalent.

Example

$$|a|a|a \ldots \equiv_\alpha |b|b|b \ldots$$

- Literal Languages are subsets of $\bar{\mathbb{A}}^*$ (resp. $\bar{\mathbb{A}}^\omega$).
- Bar Languages are subsets of $\bar{\mathbb{A}}^*/\equiv_\alpha$ (resp. $\bar{\mathbb{A}}^\omega/\equiv_\alpha$).
- Data Languages are subsets of $\mathbb{A}^*$ (resp. $\mathbb{A}^\omega$).

RNNA's [Sch+17]
Syntax

Definition

A *regular nondeterministic nominal automaton (RNNA)* is a tuple (Q, δ, q_0, F) consisting of

- an orbit-finite set Q of *states*,
- an equivariant *transition relation* $\delta \subseteq Q \times \bar{\mathbb{A}} \times Q$,
- an *initial state* $q_0 \in Q$,
- an equivariant subset $F \subseteq Q$ of *final states*,

such that

1. δ is α-invariant, meaning $q \xrightarrow{|a} q'$ and $\langle a \rangle q' = \langle b \rangle q''$ implies $q \xrightarrow{|b} q''$,
2. δ is finitely branching up to α-equivalence, meaning that the two sets $\{(a, q') \mid q \xrightarrow{a} q'\}$ and $\{\langle a \rangle q' \mid q \xrightarrow{|a} q'\}$ are finite.

RNNA's [Sch+17]
Semantics

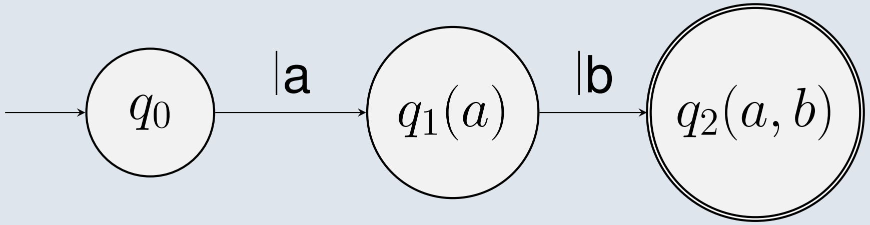
- RNNA $A = (Q, \delta, q_0, F)$
- A *run* for a finite bar string $w \in \bar{\mathbb{A}}^\omega$ from $q \in Q$ is a finite sequence of transitions

$$q \xrightarrow{\beta_0} q_1 \xrightarrow{\beta_1} \cdots \xrightarrow{\beta_{n-1}} q_n.$$

- A run for w from q is *accepting*, if $q_n \in F$.
- A state q *accepts* w, if there exists an accepting run for w from q.
- The RNNA A *accepts* w, if q_0 accepts w.

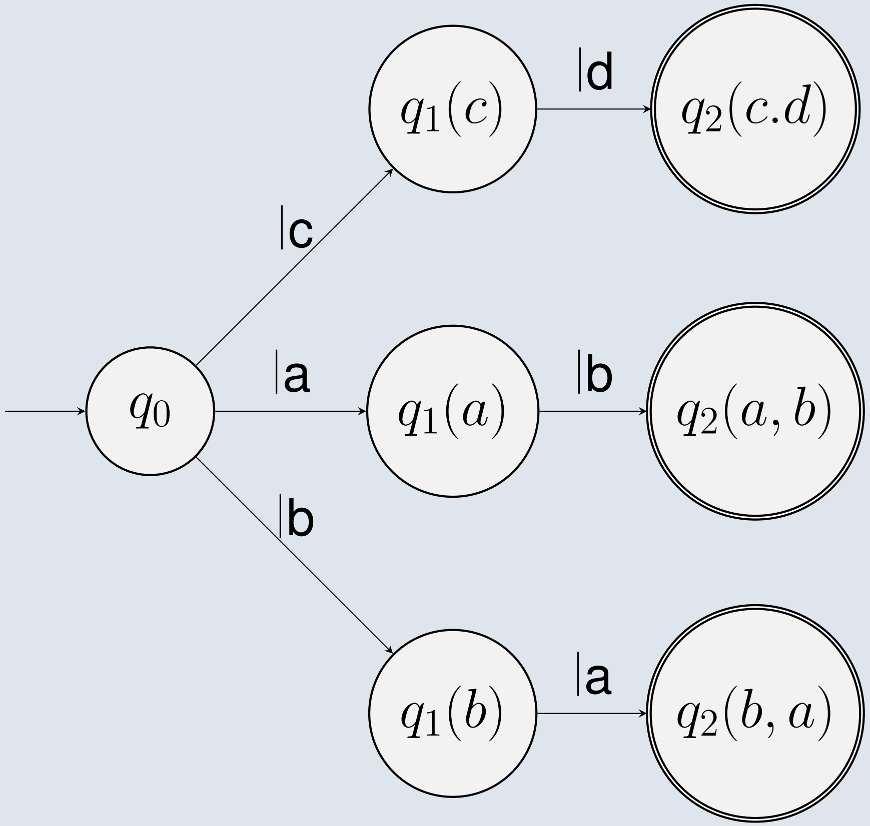
RNNA’s [Sch+17]

Example



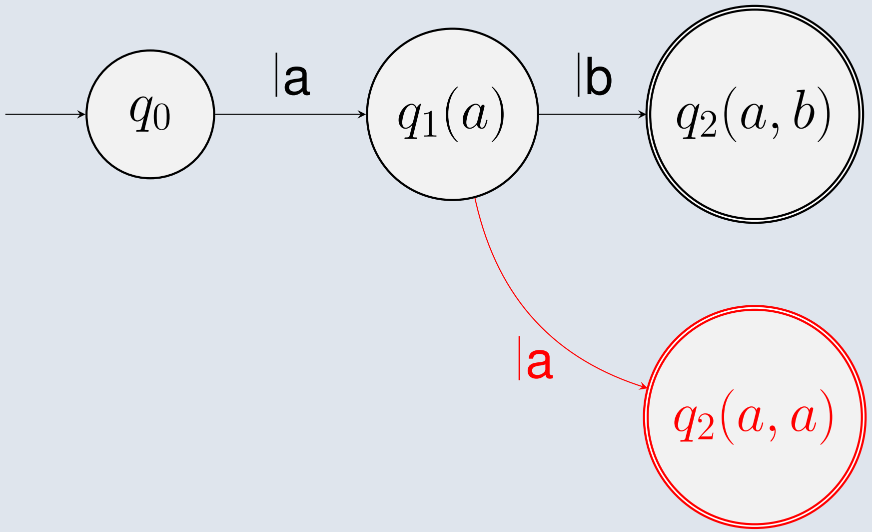
RNNA's [Sch+17]

Example



RNNA's [Sch+17]

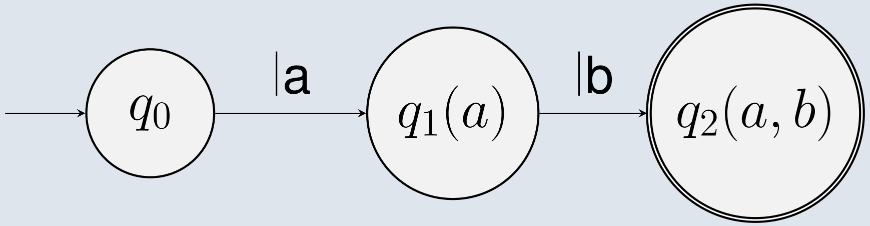
Example



- Clashes with α-invariance, as $\langle b \rangle q_2(a, b) \not\equiv_\alpha \langle a \rangle q_2(a, a)$.

RNNA's [Sch+17]

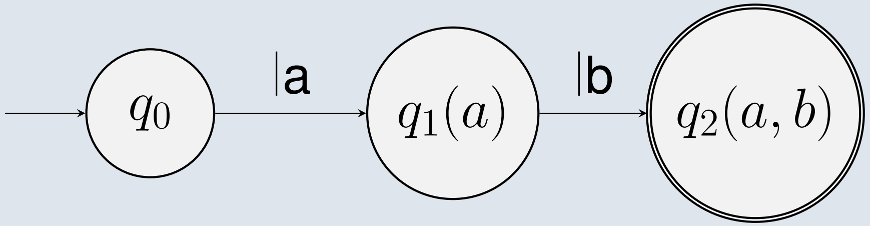
Example



- Literal Language: $\{|a|b \mid a \neq b\}$ where $|a|a$ is not included although α-equivalent to $|a|b$.
- Bar Language: $\{[|a|b]_{\equiv_\alpha}\}$

RNNA's [Sch+17]

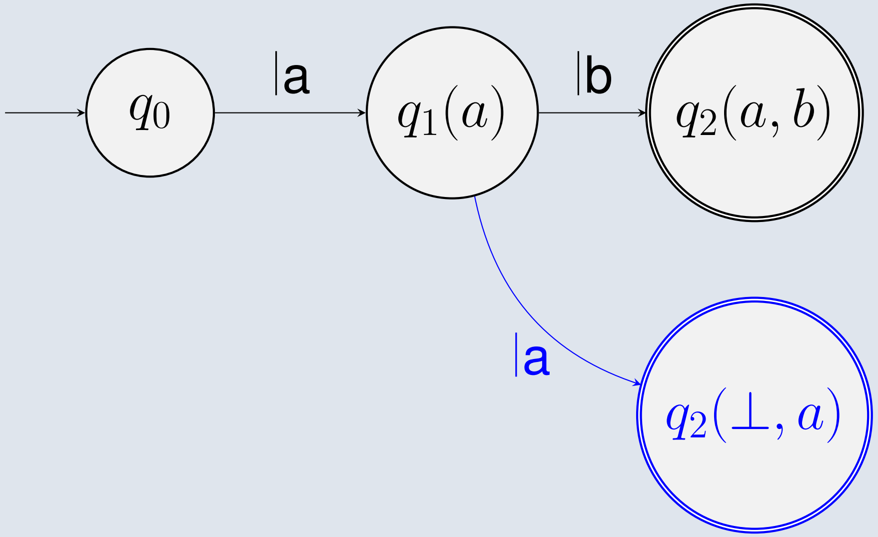
Example



- Literal Language: $\{|a|b \mid a \neq b\}$ where $|a|a$ is not included although α-equivalent to $|a|b$.
- Bar Language: $\{[|a|b]_{\equiv_\alpha}\}$
- Closure of literal language under α-equivalence?

RNNA's [Sch+17]

Example



- Idea: *Name-Dropping Modification*
- Now, $|a|a$ is (literally) accepted.

Büchi RNNA's [Urb+21]

(Same) Syntax

- Extend RNNA's from finite to infinite bar strings.
- No syntactical changes, just other semantics.

Büchi RNNA's [Urb+21]

Semantics

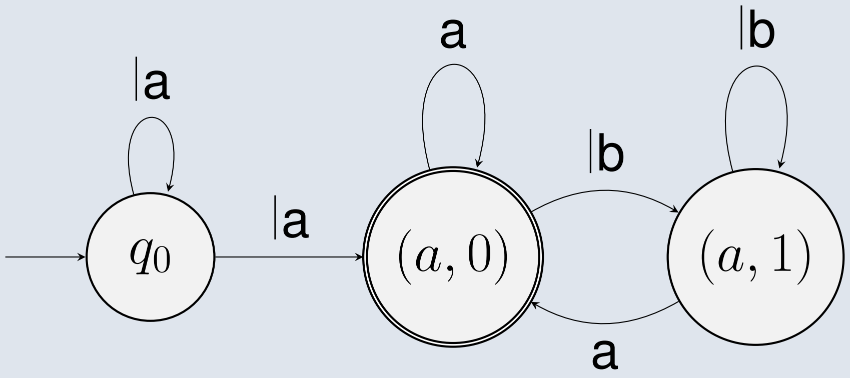
- **Büchi** RNNA $A = (Q, \delta, q_0, F)$
- A *run* for an **in**finite bar string $w \in \bar{\mathbb{A}}^\omega$ from $q \in Q$ is an **in**finite sequence of transitions

$$q \xrightarrow{\beta_0} q_1 \xrightarrow{\beta_1} \cdots$$

- A run for w from q is *accepting*, if $q_i \in F$ for **infinitely many** $i \in \mathbb{N}$.
- A state q *accepts* w, if there exists an accepting run for w from q.
- The **Büchi** RNNA A *accepts* w, if q_0 accepts w.

Büchi RNNA's [Urb+21]

Example



- Data Language consists of all $w \in \mathbb{A}^\omega$ where some letter occurs infinitely often.

(Positive) RANA's [Fra+25]

Syntax

Definition

Positive Boolean Formulae $\mathcal{B}_+(X)$ over X:

$$\phi, \psi ::= \top \mid \bot \mid x \in X \mid \phi \wedge \psi \mid \phi \vee \psi$$

(Positive) RANA's [Fra+25]
Syntax

Definition

Positive Boolean Formulae $\mathcal{B}_+(X)$ over X:

$$\phi, \psi ::= \top \mid \bot \mid x \in X \mid \phi \wedge \psi \mid \phi \vee \psi$$

Definition

A *positive regular alternating nominal automaton (RANA)* is a tuple $A = (Q, \delta, q_0)$ consisting of
- an orbit-finite nominal set Q of *states*,
- an equivariant *transition function* $\delta\colon Q \to \mathcal{B}_+(1 + \mathbb{A} \times Q + [\mathbb{A}]Q)$.
- an equivariant *initial state* $q_0 \in Q$,

(Positive) RANA's [Fra+25]

Syntax

Definition

Positive Boolean Formulae $\mathcal{B}_+(X)$ over X:

$$\phi, \psi ::= \top \mid \bot \mid x \in X \mid \phi \wedge \psi \mid \phi \vee \psi$$

Definition

A *positive regular alternating nominal automaton (RANA)* is a tuple $A = (Q, \delta, q_0)$ consisting of

- an orbit-finite nominal set Q of *states*,
- an equivariant *transition function* $\delta \colon Q \to \mathcal{B}_+(1 + \mathbb{A} \times Q + [\mathbb{A}]Q)$.
- an equivariant *initial state* $q_0 \in Q$,

Define some notation for atomic formulae:

$$\varepsilon := * \in 1$$
$$\Diamond_a q := (a, q) \in \mathbb{A} \times Q$$
$$\Diamond_{|a} q := \langle a \rangle q \in [\mathbb{A}]Q$$

(Positive) RANA's [Fra+25]

Semantics

- RANA $A = (Q, \delta, q_0)$
- For $w \in \bar{\mathbb{A}}^*$ and $\phi \in \mathcal{B}_+(1 + \mathbb{A} \times Q + [\mathbb{A}]Q)$, define satisfaction $w \models \phi$ recursively:

$$
\begin{aligned}
w &\models \varepsilon &&:\Longleftrightarrow w = \epsilon \\
bv &\models \Diamond_a q &&:\Longleftrightarrow b = a \text{ and } v \models \delta(q) \\
|bv &\models \Diamond_{|a} q &&:\Longleftrightarrow |bv \equiv_\alpha |cv' \text{ and } \langle a \rangle q = \langle c \rangle q' \text{ and } v' \models \delta(q') \text{ for some } c \in \mathbb{A}, v' \in \bar{\mathbb{A}}^*, q' \in Q \\
w &\models \phi \wedge \psi &&:\Longleftrightarrow w \models \phi \text{ and } w \models \psi \\
w &\models \phi \vee \psi &&:\Longleftrightarrow w \models \phi \text{ or } w \models \psi
\end{aligned}
$$

- A state $q \in Q$ *accepts* $w \in \bar{\mathbb{A}}^*$, if $w \models \delta(q)$.
- The RANA A *accepts* $w \in \bar{\mathbb{A}}^*$, if q_0 accepts w.

Correspondence between RANA's and Bar-μTL [Fra+25; HMS21]

- Syntax given by grammar $\phi, \psi \in \mathrm{Bar} ::= \varepsilon \mid \neg\varepsilon \mid \phi \wedge \psi \mid \phi \vee \psi \mid \Diamond_\beta \phi \mid X \mid \mu X.\, \phi$ where $\beta \in \bar{\mathbb{A}}$.
- For simplicity, we leave out $\Box_\sigma$ here.
- Satisfaction $w \models \phi$ for finite bar strings w and closed formulae ϕ defined as expected, e.g.

$$
\begin{aligned}
bv \models \Diamond_a \phi \quad & :\iff b = a \text{ and } v \models \phi \\
|bv \models \Diamond_{|a} \phi \quad & :\iff |bv \equiv_\alpha |cv' \text{ and } \langle a \rangle \phi = \langle c \rangle \psi \text{ and } v' \models \psi \text{ for some } c \in \mathbb{A}, v' \in \bar{\mathbb{A}}^*, \psi \in \mathrm{Bar} \\
w \models \mu X.\, \phi \quad & :\iff w \models \phi[X \mapsto \mu X.\, \phi]
\end{aligned}
$$

- As formulae are only evaluated over *finite* bar strings, least and greatest fixpoints coincide, therefore the syntax has only least fixpoints.

Proposition

For every $\phi \in \mathrm{Bar}$, there is a RNNA A that accepts the same closed bar strings w that satisfy ϕ, and vice versa.

NAPA's
Syntax

Definition

A *nominal alternating parity automata (NAPA)* is a tuple $A = (Q, \delta, q_0, c)$ consisting of

- an orbit-finite nominal set Q of *states*,
- an equivariant *transition function* $\delta\colon Q \to \mathcal{B}_+(\mathbb{A} \times Q + [\mathbb{A}]Q)$,
- an equivariant *initial state* $q_0 \in Q$,
- an equivariant *color function* $c\colon Q \to \mathbb{N}$.

NAPA's

Syntax

Definition

A *nominal alternating parity automata (NAPA)* is a tuple $A = (Q, \delta, q_0, c)$ consisting of

- an orbit-finite nominal set Q of *states*,
- an equivariant *transition function* $\delta\colon Q \to \mathcal{B}_+(\mathbb{A} \times Q + [\mathbb{A}]Q)$,
- an equivariant *initial state* $q_0 \in Q$,
- an equivariant *color function* $c\colon Q \to \mathbb{N}$.

Define some notation for atomic formulae:

$$\Diamond_a q := (a, q) \in \mathbb{A} \times Q$$
$$\Diamond_{|a} q := \langle a \rangle q \in [\mathbb{A}]Q$$

Nominal Parity Games

Positions

- Given a NAPA A and $w \in \bar{\mathbb{A}}^\omega$, we define a nominal parity game between ∀belard and ∃loise.
- $\mathrm{Perm}(\mathbb{A})$-set of positions:

$$
\begin{aligned}
\mathrm{Pos} &:= (Q + \mathcal{B}_+(\mathbb{A} \times Q + [\mathbb{A}] \times Q)) \times \bar{\mathbb{A}}^\omega \times \mathbb{N} \\
\mathrm{pos}_\forall &:= \{(\phi \wedge \psi, v, i) \mid \phi, \psi \in \mathcal{B}_+(\mathbb{A} \times Q + [\mathbb{A}]Q), v \in \bar{\mathbb{A}}^\omega, i \in \mathbb{N}\} \\
&\quad \cup \{(\top, v, i) \mid v \in \bar{\mathbb{A}}^\omega, i \in \mathbb{N}\} \\
\mathrm{pos}_\exists &:= \mathrm{Pos} \setminus \mathrm{pos}_\forall
\end{aligned}
$$

Nominal Parity Game

Moves and Plays

- Moves:

$$(q, v, i) \xrightarrow{\exists} (\delta(q), v, i)$$
$$(\phi \wedge \psi, v, i) \xrightarrow{\forall} (\phi, v, i)$$
$$(\phi \wedge \psi, v, i) \xrightarrow{\forall} (\psi, v, i)$$
$$(\phi \vee \psi, v, i) \xrightarrow{\exists} (\phi, v, i)$$
$$(\phi \vee \psi, v, i) \xrightarrow{\exists} (\psi, v, i)$$
$$(\Diamond_a q, \beta v, i) \xrightarrow{\exists} (q, v, i+1) \quad :\Longleftrightarrow \quad \beta = a$$
$$(\Diamond_{|a} q, \beta v, i) \xrightarrow{\exists} (q', v', i+1) \quad :\Longleftrightarrow \quad \exists a', c \in \mathbb{A}.\ \langle a \rangle q = \langle c \rangle q', \beta = |a' \text{ and } |a'v \equiv_\alpha |cv'$$

Nominal Parity Game

Moves and Plays

- Moves:

$$
\begin{aligned}
(q, v, i) &\xrightarrow{\exists} (\delta(q), v, i) \\
(\phi \wedge \psi, v, i) &\xrightarrow{\forall} (\phi, v, i) \\
(\phi \wedge \psi, v, i) &\xrightarrow{\forall} (\psi, v, i) \\
(\phi \vee \psi, v, i) &\xrightarrow{\exists} (\phi, v, i) \\
(\phi \vee \psi, v, i) &\xrightarrow{\exists} (\psi, v, i) \\
(\Diamond_a q, \beta v, i) &\xrightarrow{\exists} (q, v, i+1) && :\iff \beta = a \\
(\Diamond_{|a} q, \beta v, i) &\xrightarrow{\exists} (q', v', i+1) && :\iff \exists a', c \in \mathbb{A}.\, \langle a \rangle q = \langle c \rangle q', \beta = |a' \text{ and } |a'v \equiv_\alpha |cv'
\end{aligned}
$$

- Plays: finite or infinite sequences of moves.

Nominal Parity Game

Winning Conditions and NAPA Semantics

- $\forall$belard wins a play r, if one of the following conditions is fulfilled:
- $\exists$loise wins a play r, if one of the following conditions is fulfilled:

Nominal Parity Game

Winning Conditions and NAPA Semantics

- $\forall$belard wins a play r, if one of the following conditions is fulfilled:
  - $r = p_1 \longrightarrow \cdots \longrightarrow p_n$ is finite and $p_n \in \mathrm{pos}_\exists$.
- $\exists$loise wins a play r, if one of the following conditions is fulfilled:
  - $r = p_1 \longrightarrow \cdots \longrightarrow p_n$ is finite and $p_n \in \mathrm{pos}_\forall$.

Nominal Parity Game

Winning Conditions and NAPA Semantics

- $\forall$belard wins a play r, if one of the following conditions is fulfilled:
  - $r = p_1 \longrightarrow \cdots \longrightarrow p_n$ is finite and $p_n \in \mathrm{pos}_\exists$.
  - r is infinite and the highest infinitely often occuring colour is odd.
- $\exists$loise wins a play r, if one of the following conditions is fulfilled:
  - $r = p_1 \longrightarrow \cdots \longrightarrow p_n$ is finite and $p_n \in \mathrm{pos}_\forall$.
  - r is infinite and the highest infinitely often occuring colour is even.

Nominal Parity Game

Winning Conditions and NAPA Semantics

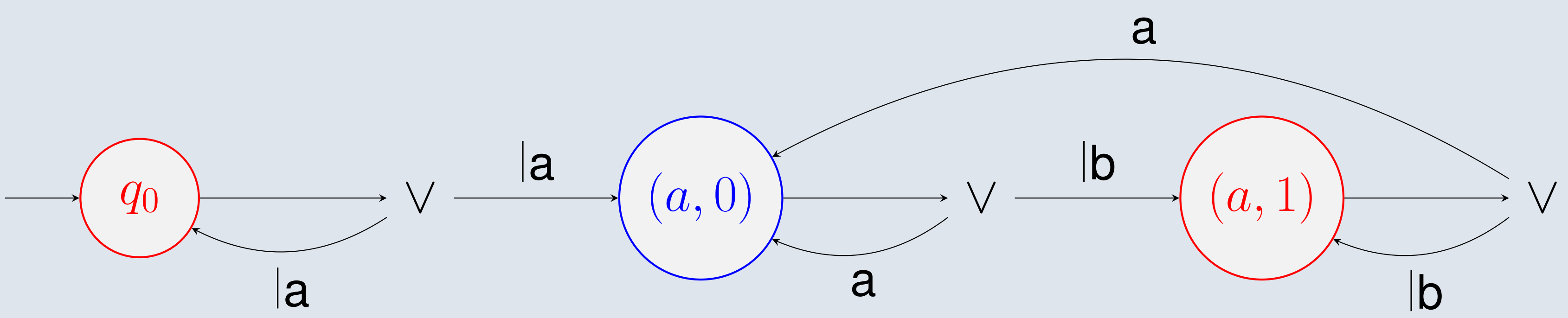
- $\forall$belard wins a play r, if one of the following conditions is fulfilled:
  - $r = p_1 \longrightarrow \cdots \longrightarrow p_n$ is finite and $p_n \in \mathrm{pos}_\exists$.
  - r is infinite and the highest infinitely often occuring colour is odd.
- $\exists$loise wins a play r, if one of the following conditions is fulfilled:
  - $r = p_1 \longrightarrow \cdots \longrightarrow p_n$ is finite and $p_n \in \mathrm{pos}_\forall$.
  - r is infinite and the highest infinitely often occuring colour is even.
- A NAPA A accepts an infinite bar string $w \in \bar{\mathbb{A}}^\omega$, if $\exists$loise has a winning strategy for the corresponding nominal parity game.

NAPA's

Example

Construct an equivalent NAPA $A = (\{q_0\} + \mathbb{A} \times \{0,1\}, \delta, q_0, c)$ for the Büchi RNNA from before.

$$\begin{aligned} \delta(q_0) &:= \Diamond_{|a} q_0 \vee \Diamond_{|a}(a,0) \\ \delta(a,0) &:= \Diamond_{a}(a,0) \vee \Diamond_{|b}(a,1) \\ \delta(a,1) &:= \Diamond_{a}(a,0) \vee \Diamond_{|b}(a,1) \end{aligned} \qquad \begin{aligned} c(q_0) &:= 1 \\ c(a,0) &:= 2 \\ c(a,1) &:= 1 \end{aligned}$$



From Büchi RNNA's to NAPA's

Construction

- Given: Büchi RNNA $A = (Q, \delta, q_0, F)$
- Construct a NAPA $A' := (Q, \delta', q_0, c)$ as follows:

$$\delta'(q) := \bigvee_{(a,q')\in S_q} \Diamond_a q' \vee \bigvee_{\langle a\rangle q' \in S_{|q}} \Diamond_{|a} q'$$

$$c(q) := \begin{cases} 2 & \iff q \in F \\ 1 & \iff q \notin F \end{cases}$$

From Büchi RNNA’s to NAPA’s

FAU

Construction

- Given: Büchi RNNA $A = (Q, \delta, q_0, F)$
- Construct a NAPA $A' := (Q, \delta', q_0, c)$ as follows:

$$\delta'(q) := \bigvee_{(a,q')\in S_q} \Diamond_a q' \vee \bigvee_{\langle a\rangle q' \in S_{|q}} \Diamond_{|a} q'$$

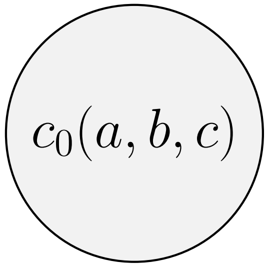
$$c(q) := \begin{cases} 2 & \iff q \in F \\ 1 & \iff q \notin F \end{cases}$$

Proposition

1. *A accepts $w \in \bar{\mathbb{A}}^\omega$ implies A' accepts w.*
2. *A' accepts $w \in \bar{\mathbb{A}}^\omega$ implies A accepts some $v \equiv_\alpha w$.*

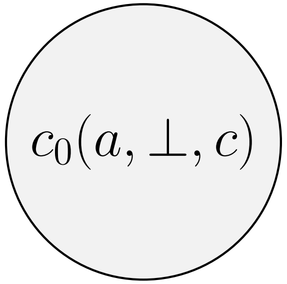
Name-Dropping Modification

- Nominal set $\mathbb{A}^{\#n}$ of total injective maps $\{0, \ldots, n-1\} \to \mathbb{A}$.



Name-Dropping Modification

- Nominal set $\mathbb{A}^{\#n}$ of total injective maps $\{0, \ldots, n-1\} \to \mathbb{A}$.
- Nominal set $\mathbb{A}^{\$n}$ of partial injective maps $\{0, \ldots, n-1\} \to \mathbb{A}$.



Name-Dropping Modification

- Nominal set $\mathbb{A}^{\#n}$ of total injective maps $\{0, \dots, n-1\} \to \mathbb{A}$.
- Nominal set $\mathbb{A}^{\$n}$ of partial injective maps $\{0, \dots, n-1\} \to \mathbb{A}$.

Construction

- Given: NAPA $A = (Q, \delta, q_0, c)$ with strong nominal state set $Q = \coprod_{i=1}^{n} \mathbb{A}^{\#n_i}$
- Construct a NAPA (the *name-dropping modification of* A) $A_{\mathrm{nd}} = (Q_{\mathrm{nd}}, \delta_{\mathrm{nd}}, q_0, c_{\mathrm{nd}})$ as follows:

Name-Dropping Modification

- Nominal set $\mathbb{A}^{\#n}$ of total injective maps $\{0, \dots, n-1\} \to \mathbb{A}$.
- Nominal set $\mathbb{A}^{\$n}$ of partial injective maps $\{0, \dots, n-1\} \to \mathbb{A}$.

Construction

- Given: NAPA $A = (Q, \delta, q_0, c)$ with strong nominal state set $Q = \coprod_{i=1}^{n} \mathbb{A}^{\#n_i}$
- Construct a NAPA (the *name-dropping modification of* A) $A_{\mathrm{nd}} = (Q_{\mathrm{nd}}, \delta_{\mathrm{nd}}, q_0, c_{\mathrm{nd}})$ as follows:
- For $(i, r) \in Q_{\mathrm{nd}}$, choose an extension $\overline{r}$ of r such that $(i, \overline{r}) \in Q$ and set $c_{\mathrm{nd}}(i, r) := c(i, \overline{r})$.

Name-Dropping Modification

- Nominal set $\mathbb{A}^{\#n}$ of total injective maps $\{0, \dots, n-1\} \to \mathbb{A}$.
- Nominal set $\mathbb{A}^{\$n}$ of partial injective maps $\{0, \dots, n-1\} \to \mathbb{A}$.

Construction

- Given: NAPA $A = (Q, \delta, q_0, c)$ with strong nominal state set $Q = \coprod_{i=1}^{n} \mathbb{A}^{\#n_i}$
- Construct a NAPA (the *name-dropping modification of* A) $A_{\mathrm{nd}} = (Q_{\mathrm{nd}}, \delta_{\mathrm{nd}}, q_0, c_{\mathrm{nd}})$ as follows:
- For $(i, r) \in Q_{\mathrm{nd}}$, choose an extension $\overline{r}$ of r such that $(i, \overline{r}) \in Q$ and set $c_{\mathrm{nd}}(i, r) := c(i, \overline{r})$.
- For each $m \in \mathbb{N}$ and $r \in \mathbb{A}^{\$m}$ define a map

$$f_r \colon \mathcal{B}_+(\mathbb{A} \times Q + [\mathbb{A}]Q) \to \mathcal{B}_+(\mathbb{A} \times Q_{\mathrm{nd}} + [\mathbb{A}]Q_{\mathrm{nd}}).$$

  to add name-dropping options for ∃loise.

Name-Dropping Modification

- Nominal set $\mathbb{A}^{\#n}$ of total injective maps $\{0, \dots, n-1\} \to \mathbb{A}$.
- Nominal set $\mathbb{A}^{\$n}$ of partial injective maps $\{0, \dots, n-1\} \to \mathbb{A}$.

Construction

- Given: NAPA $A = (Q, \delta, q_0, c)$ with strong nominal state set $Q = \coprod_{i=1}^{n} \mathbb{A}^{\#n_i}$
- Construct a NAPA (the *name-dropping modification of* A) $A_{\mathrm{nd}} = (Q_{\mathrm{nd}}, \delta_{\mathrm{nd}}, q_0, c_{\mathrm{nd}})$ as follows:
- For $(i, r) \in Q_{\mathrm{nd}}$, choose an extension $\overline{r}$ of r such that $(i, \overline{r}) \in Q$ and set $c_{\mathrm{nd}}(i, r) := c(i, \overline{r})$.
- For each $m \in \mathbb{N}$ and $r \in \mathbb{A}^{\$m}$ define a map

$$f_r \colon \mathcal{B}_+(\mathbb{A} \times Q + [\mathbb{A}]Q) \to \mathcal{B}_+(\mathbb{A} \times Q_{\mathrm{nd}} + [\mathbb{A}]Q_{\mathrm{nd}}).$$

  to add name-dropping options for ∃loise.
- For $(i, r) \in Q_{\mathrm{nd}}$, choose an extension $\overline{r}$ of r such that $(i, \overline{r}) \in Q$ and set $\delta_{\mathrm{nd}}(i, r) = f_r(\delta(i, \overline{r}))$.

Name-Dropping Modification

- Nominal set $\mathbb{A}^{\#n}$ of total injective maps $\{0, \ldots, n-1\} \to \mathbb{A}$.
- Nominal set $\mathbb{A}^{\$n}$ of partial injective maps $\{0, \ldots, n-1\} \to \mathbb{A}$.

Construction

- Given: NAPA $A = (Q, \delta, q_0, c)$ with strong nominal state set $Q = \coprod_{i=1}^{n} \mathbb{A}^{\#n_i}$
- Construct a NAPA (the *name-dropping modification of* A) $A_{\mathrm{nd}} = (Q_{\mathrm{nd}}, \delta_{\mathrm{nd}}, q_0, c_{\mathrm{nd}})$ as follows:
- For $(i, r) \in Q_{\mathrm{nd}}$, choose an extension $\overline{r}$ of r such that $(i, \overline{r}) \in Q$ and set $c_{\mathrm{nd}}(i, r) := c(i, \overline{r})$.
- For each $m \in \mathbb{N}$ and $r \in \mathbb{A}^{\$m}$ define a map

$$f_r \colon \mathcal{B}_+(\mathbb{A} \times Q + [\mathbb{A}]Q) \to \mathcal{B}_+(\mathbb{A} \times Q_{\mathrm{nd}} + [\mathbb{A}]Q_{\mathrm{nd}}).$$

  to add name-dropping options for $\exists$loise:

$$f_r(\Diamond_a(j, \overline{s})) := \bigvee \{\Diamond_a(j, s) \mid s \in \mathbb{A}^{\$n_j}, s \leq \overline{s}, \mathrm{supp}(s) \cup \{a\} \subseteq \mathrm{supp}(r)\}$$

$$f_r(\Diamond_{|a}(j, \overline{s})) := \bigvee \{\Diamond_{|a}(j, s) \mid s \in \mathbb{A}^{\$n_j}, s \leq \overline{s}, \mathrm{supp}(s) \subseteq \mathrm{supp}(r) \cup \{a\}\}$$

- For $(i, r) \in Q_{\mathrm{nd}}$, choose an extension $\overline{r}$ of r such that $(i, \overline{r}) \in Q$ and set $\delta_{\mathrm{nd}}(i, r) = f_r(\delta(i, \overline{r}))$.

Name-Dropping Modification

Lemma

The name-dropping modification A_{nd} as described is indeed a NAPA.

Proof Sketch.

Show that δ_{nd} and c_{nd} are well-defined and equivariant. □

Name-Dropping Modification

Lemma

The name-dropping modification A_{nd} as described is indeed a NAPA.

Proof Sketch.

Show that δ_{nd} and c_{nd} are well-defined and equivariant. $\square$

- Goal of the name-dropping modification: Make α-renamings unnecessary when processing $\Diamond_{|a}$-transitions.

Name-Dropping Modification

Lemma

The name-dropping modification A_{nd} as described is indeed a NAPA.

Proof Sketch.

Show that δ_{nd} and c_{nd} are well-defined and equivariant. □

- Goal of the name-dropping modification: Make α-renamings unnecessary when processing $\Diamond_{|a}$-transitions.

Conjecture

If a NAPA A accepts some $w \in \bar{\mathbb{A}}^\omega$ and $w \equiv_\alpha w'$ then A_{nd} accepts w'.

Name-Dropping Modification

Lemma

The name-dropping modification A_{nd} as described is indeed a NAPA.

Proof Sketch.

Show that δ_{nd} and c_{nd} are well-defined and equivariant. □

- Goal of the name-dropping modification: Make α-renamings unnecessary when processing $\Diamond_{|a}$-transitions.

Conjecture

If a NAPA A accepts some $w \in \bar{\mathbb{A}}^\omega$ and $w \equiv_\alpha w'$ then A_{nd} accepts w'.

- Problem: No inductive principle for $w \in \bar{\mathbb{A}}^\omega$ available.

Name-Dropping Modification

Lemma

The name-dropping modification A_{nd} as described is indeed a NAPA.

Proof Sketch.

Show that δ_{nd} and c_{nd} are well-defined and equivariant. □

- Goal of the name-dropping modification: Make α-renamings unnecessary when processing $\Diamond_{|a}$-transitions.

Conjecture

If a NAPA A accepts some $w \in \bar{\mathbb{A}}^\omega$ and $w \equiv_\alpha w'$ then A_{nd} accepts w'.

- Problem: No inductive principle for $w \in \bar{\mathbb{A}}^\omega$ available.
- Idea: Use a similar approach as in [Urb+21] using König's Lemma [Kö27].

Lemma (König's Lemma (simplified))

Every infinite tree that is finitely branching has some infinite path in it.

Conclusion and Future Work

Conclusion

- We introduced Nominal Alternating Parity Automata which extend RANA's for infinite bar strings.
- We presented a construction from Büchi RNNA's to NAPA's.
- We showed a name-dropping modification, although without a proof of correctness yet.

Conclusion and Future Work

Conclusion

- We introduced Nominal Alternating Parity Automata which extend RANA's for infinite bar strings.
- We presented a construction from Büchi RNNA's to NAPA's.
- We showed a name-dropping modification, although without a proof of correctness yet.

Future Work

- Prove correctness of the name-dropping modification.
- Extend Bar-μTL for infinite bar strings.
- De-Alternation of NAPA's?
- Complexity of emptiness and inclusion problem?

[Fra+25] F. Frank, D. Hausmann, S. Milius, L. Schröder, and H. Urbat. "Alternating Nominal Automata with Name Allocation". In: *2025 40th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 2025 40th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). June 2025, pp. 57–70. DOI: 10.1109/LICS65433.2025.00012.

[HMS21] D. Hausmann, S. Milius, and L. Schröder. "A Linear-Time Nominal \mu-Calculus with Name Allocation". In: *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*. Ed. by F. Bonchi and S. J. Puglisi. Vol. 202. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 58:1–58:18. DOI: 10.4230/LIPIcs.MFCS.2021.58.

[Kö27] D. König. "Über eine Schlussweise aus dem Endlichen ins Unendliche : Punktmengen. — Kartenfärben. — Verwandtschaftsbeziehungen. — Schachspiel". In: *Acta litterarum ac scientiarum Regiae Universitatis Hungaricae Francisco-Josephinae : Sectio scientiarum mathematicarum* 3 (1927), pp. 121–130.

[Pit13] A. M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*. 1st ed. Cambridge University Press, May 30, 2013. DOI: 10.1017/CBO9781139084673.

[Sch+17] L. Schröder, D. Kozen, S. Milius, and T. Wißmann. "Nominal Automata with Name Binding". In: *Foundations of Software Science and Computation Structures*. Ed. by J. Esparza and A. S. Murawski. Berlin, Heidelberg: Springer, 2017, pp. 124–142. DOI: 10.1007/978-3-662-54458-7_8.

[Urb+21] H. Urbat, D. Hausmann, S. Milius, and L. Schröder. “Nominal Büchi Automata with Name Allocation”. In: *LIPIcs, Volume 203, CONCUR 2021* 203 (2021). Ed. by S. Haddad and D. Varacca, 4:1–4:16. DOI: 10.4230/LIPICS.CONCUR.2021.4.