Friedrich-Alexander-Universität
Erlangen-Nürnberg

Choose your (Equivariant?) Strategy: NAPA’s and Nominal Parity Games

Max Ole Elliger   Florian Frank   Daniel Hausmann   Stefan Milius   Lutz Schröder

June 23, 2026

Overview

- Automata with name binding (e.g. RNNAs [Sch+17], Büchi RNNAs [Urb+21], RANAs [Fra+25]) have been introduced to accept words over infinite alphabets with explicit binders.
- All three automata models come with decidable inclusion and emptiness problems.
- Currently, we develop *nominal alternating parity automata (NAPAs)*.

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 (symmetric) 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 (symmetric) 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 $\mathrm{supp}(x) \subseteq \mathbb{A}$ 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 (symmetric) 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 $(\pi \cdot f)(x) := \pi^{-1} \cdot f(\pi \cdot x)$

- An object $x \in X$ is *equivariant*, if $\mathrm{supp}(x) = \emptyset$.
- 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: $\bar{\mathbb{A}} := \mathbb{A} \cup \{|a \mid a \in \mathbb{A}\}$.

Bar Strings [Sch+17]

- Duplicate name set in order to introduce binders: $\bar{\mathbb{A}} := \mathbb{A} \cup \{|a \mid a \in \mathbb{A}\}$.
- Nominal set $\bar{\mathbb{A}}^*$ of finite bar strings
- $\mathrm{Perm}(\mathbb{A})$-set $\bar{\mathbb{A}}^\omega$ of infinite bar strings
- Nominal set $\bar{\mathbb{A}}^\omega_{\mathsf{fs}}$ of finitely supported infinite bar strings

Bar Strings [Sch+17]

- Duplicate name set in order to introduce binders: $\bar{\mathbb{A}} := \mathbb{A} \cup \{|a \mid a \in \mathbb{A}\}$.
- Nominal set $\bar{\mathbb{A}}^*$ of finite bar strings
- $\mathrm{Perm}(\mathbb{A})$-set $\bar{\mathbb{A}}^\omega$ of infinite bar strings
- Nominal set $\bar{\mathbb{A}}^\omega_{\mathsf{fs}}$ of finitely supported infinite bar strings
- 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

$$|a|b|a\ldots \equiv_\alpha |b|a|b\ldots$$
$$|ab|a\ldots \not\equiv_\alpha |ba|b\ldots$$
$$|a|a|a\ldots \equiv_\alpha |a|b|c\ldots$$

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:

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

NAPA's
Positions

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

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

NAPA's
Moves and Plays

- (Equivariant) relation of *moves*:

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

NAPA's
Moves and Plays

- (Equivariant) relation of *moves*:

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

- Plays are finite or infinite sequences of moves.

NAPA's

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:

NAPA's
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 $p_n \in \mathrm{pos}_\forall$.

NAPA's

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.

NAPA's

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 *strategy* for a player x is a partial function of type $\mathrm{pos}_x \to \mathrm{Pos}$ that respects the move relation.
- A strategy is a *winning strategy* for a player x, if all plays consistent with that strategy are won by x.

NAPA's

Winning Conditions and NAPA Semantics

- A strategy is a *winning strategy* for a player x, if all plays consistent with that strategy are won by x.
- A NAPA A *accepts* an infinite bar string $w \in \bar{\mathbb{A}}^{\omega}$, if ∃loise has a winning strategy.

NAPA's

Winning Conditions and NAPA Semantics

- A NAPA A *accepts* an infinite bar string $w \in \bar{\mathbb{A}}^\omega$, if ∃loise has a winning strategy.
- A NAPA A *accepts* an infinite bar string $w \in \bar{\mathbb{A}}^\omega$ *in whole*, if ∃loise has a winning strategy such that at least one play consistent with that strategy is infinite
- A NAPA A *finitely accepts* an infinite bar string $w \in \bar{\mathbb{A}}^\omega$, if ∃loise has a winning strategy such that all plays consistent with that strategy are finite (and end in $\top$).

NAPA's

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 *strategy* for a player x is a partial function of type $\mathrm{pos}_x \to \mathrm{Pos}$ that respects the move relation.
- A strategy is a *winning strategy* for a player x, if all plays consistent with that strategy are won by x.
- A NAPA A *accepts* an infinite bar string $w \in \bar{\mathbb{A}}^\omega$, if $\exists$loise has a winning strategy.
- A NAPA A *accepts* an infinite bar string $w \in \bar{\mathbb{A}}^\omega$ *in whole*, if $\exists$loise has a winning strategy such that at least one play consistent with that strategy is infinite
- A NAPA A *finitely accepts* an infinite bar string $w \in \bar{\mathbb{A}}^\omega$, if $\exists$loise has a winning strategy such that all plays consistent with that strategy are finite (and end in $\top$).

Bounded Finite Support

Proposition

Let $w \equiv_\alpha w'$ be α-equivalent infinite bar strings. A NAPA A accepts w in whole iff A accepts w' in whole.

Bounded Finite Support

Proposition

Let $w \equiv_\alpha w'$ *be* α*-equivalent infinite bar strings. A NAPA* A *accepts* w *in whole iff* A *accepts* w' *in whole.*

Proposition

If $w \in \bar{\mathbb{A}}^\omega$ *is accepted in whole by a NAPA* A*, then there exists some* $w' \equiv_\alpha w$ *whose support is bounded by* $\deg(A) + 1$ *where* $\deg(A) = \max_{q \in Q} |\operatorname{supp}(q)|$.

Bounded Finite Support

Proposition

Let $w \equiv_\alpha w'$ *be* α*-equivalent infinite bar strings. A NAPA* A *accepts* w *in whole iff* A *accepts* w' *in whole.*

Proposition

If $w \in \bar{\mathbb{A}}^\omega$ *is accepted in whole by a NAPA* A*, then there exists some* $w' \equiv_\alpha w$ *whose support is bounded by* $\deg(A) + 1$ *where* $\deg(A) = \max_{q \in Q} |\operatorname{supp}(q)|$.

Corollary

If $w \in \bar{\mathbb{A}}^\omega$ *is accepted in whole by a NAPA* A*, then there exists some* $w' \equiv_\alpha w$ *that is accepted in whole by* A *and whose support is bounded by* $\deg(A) + 1$ *where* $\deg(A) = \max_{q \in Q} |\operatorname{supp}(q)|$.

Bounded Finite Support

Proposition

Let $w \equiv_\alpha w'$ be α-equivalent infinite bar strings. A NAPA A accepts w in whole iff A accepts w' in whole.

Proposition

If $w \in \bar{\mathbb{A}}^\omega$ is accepted in whole by a NAPA A, then there exists some $w' \equiv_\alpha w$ whose support is bounded by $\deg(A) + 1$ where $\deg(A) = \max_{q \in Q} |\operatorname{supp}(q)|$.

Corollary

If $w \in \bar{\mathbb{A}}^\omega$ is accepted in whole by a NAPA A, then there exists some $w' \equiv_\alpha w$ that is accepted in whole by A and whose support is bounded by $\deg(A) + 1$ where $\deg(A) = \max_{q \in Q} |\operatorname{supp}(q)|$.

Remark

If an infinite bar string w is (only) finitely accepted, similar results are proven regarding an accepted pre-word of w.

Bounded Finite Support

Proposition

Let $w \equiv_\alpha w'$ be α-equivalent infinite bar strings. A NAPA A accepts w in whole iff A accepts w' in whole.

Proposition

If $w \in \bar{\mathbb{A}}^\omega$ is accepted in whole by a NAPA A, then there exists some $w' \equiv_\alpha w$ whose support is bounded by $\deg(A) + 1$ where $\deg(A) = \max_{q \in Q} |\operatorname{supp}(q)|$.

Corollary

If $w \in \bar{\mathbb{A}}^\omega$ is accepted in whole by a NAPA A, then there exists some $w' \equiv_\alpha w$ that is accepted in whole by A and whose support is bounded by $\deg(A) + 1$ where $\deg(A) = \max_{q \in Q} |\operatorname{supp}(q)|$.

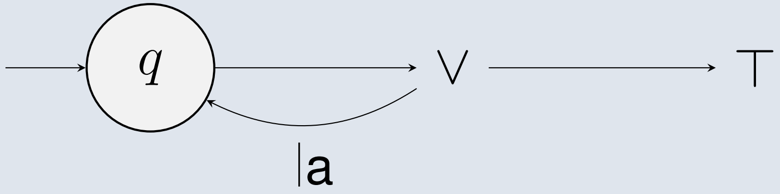
Remark

If an infinite bar string w is (only) finitely accepted, similar results are proven regarding an accepted pre-word of w. This pre-word can be constructed using König's Lemma [Kön27], since all plays consistent with the witnessing winning strategy are finite and can be arranged in a finitely branching tree.

Equivariant Strategies?

Example

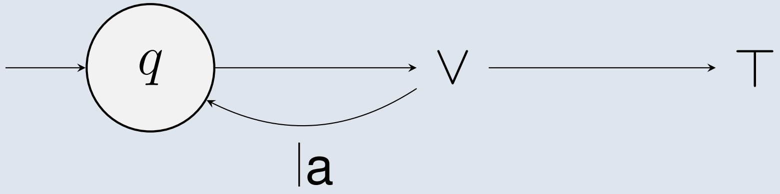
$$\text{NAPA } A := (\{q\}, \delta, q, c)$$
$$\delta(q) := \top \vee \Diamond_{|a} q$$
$$c(q) := 0$$



Equivariant Strategies?

Example

$$\text{NAPA } A := (\{q\}, \delta, q, c)$$
$$\delta(q) := \top \vee \Diamond_{|a} q$$
$$c(q) := 0$$



(At least) three winning strategies for $\exists$ (for $(|a)^\omega$) exist:

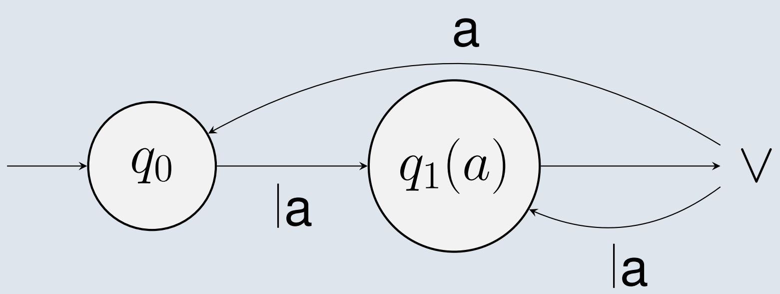
$$s_0(q, (|a)^\omega) := (\delta(q), (|a)^\omega)$$
$$s_0(\delta(q), (|a)^\omega) := (\top, (|a)^\omega)$$

$$s_1(q, (|a)^\omega) := (\delta(q), (|a)^\omega)$$
$$s_1(\delta(q), (|a)^\omega) := (\Diamond_{|a} q, (|a)^\omega)$$
$$s_1(\Diamond_{|a} q, (|a)^\omega) := (q, (|b)^\omega)$$
$$s_1(\delta(q), (|b)^\omega) := (\top, (|b)^\omega)$$

$$s_\infty(q, (|a)^\omega) := (\delta(q), (|a)^\omega)$$
$$s_\infty(\delta(q), w) := (\Diamond_{|a} q, (|a)^\omega)$$
$$s_\infty(\Diamond_{|a} q, (|a)^\omega) := (q, (|a)^\omega)$$

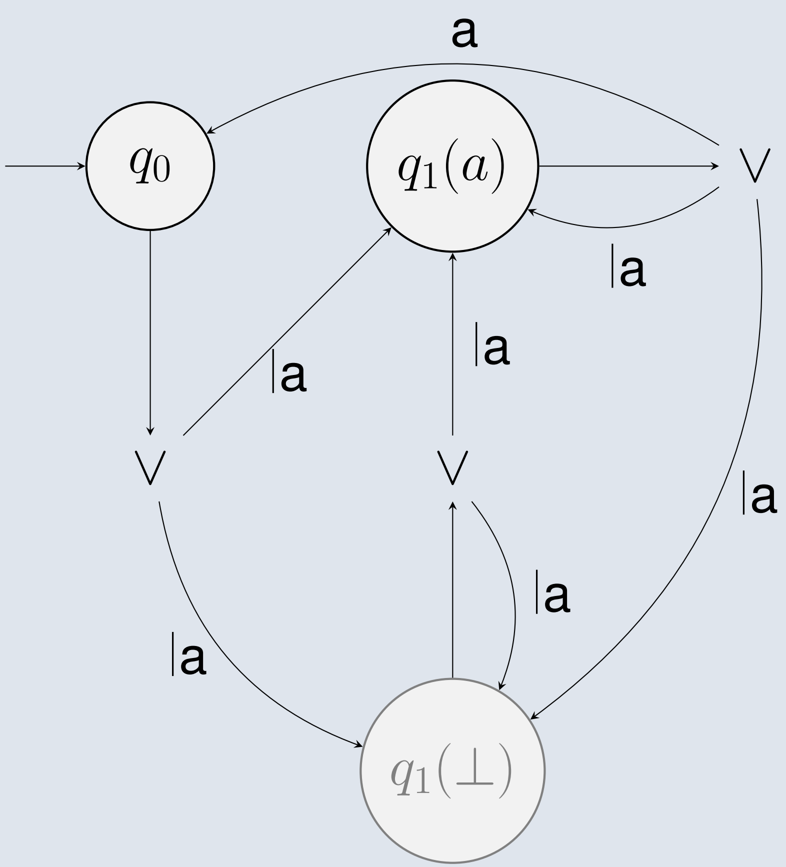
Name-Dropping without equivariant strategies?

Example



Name-Dropping without equivariant strategies?

Example



Infinite Games [Maz02] revisited

Infinite Games [Maz02] revisited

Definition

An *arena* is a triple $\mathcal{A} = (V_0, V_1, E)$ where $V_0 \cap V_1 = \emptyset$, $V := V_0 \cup V_1$ and $E \subseteq V \times V$.

Infinite Games [Maz02] revisited

Definition

An *arena* is a triple $\mathcal{A} = (V_0, V_1, E)$ where $V_0 \cap V_1 = \emptyset$, $V := V_0 \cup V_1$ and $E \subseteq V \times V$.

Definition

A *play* in $\mathcal{A}$ is a finite or infinite path $p \in V^+ \cup V^\omega$ in V that respects E.

Infinite Games [Maz02] revisited

Definition

An *arena* is a triple $\mathcal{A} = (V_0, V_1, E)$ where $V_0 \cap V_1 = \emptyset$, $V := V_0 \cup V_1$ and $E \subseteq V \times V$.

Definition

A *play* in $\mathcal{A}$ is a finite or infinite path $p \in V^+ \cup V^\omega$ in V that respects E.

Definition

A *winning condition* is a subset $\mathrm{Win} \subseteq V^\omega$.

Infinite Games [Maz02] revisited

Definition

An *arena* is a triple $\mathcal{A} = (V_0, V_1, E)$ where $V_0 \cap V_1 = \emptyset$, $V := V_0 \cup V_1$ and $E \subseteq V \times V$.

Definition

A *play* in $\mathcal{A}$ is a finite or infinite path $p \in V^+ \cup V^\omega$ in V that respects E.

Definition

A *winning condition* is a subset $\mathrm{Win} \subseteq V^\omega$.

Definition

A play is *won* by player 0, if it is finite and ends in a dead end for $\overline{\sigma}$ or if it is infinite and $p \in \mathrm{Win}$. A play is *won* by 1, if it is not won by 0.

Infinite Games [Maz02] revisited

Definition

An *arena* is a triple $\mathcal{A} = (V_0, V_1, E)$ where $V_0 \cap V_1 = \emptyset$, $V := V_0 \cup V_1$ and $E \subseteq V \times V$.

Definition

A *play* in $\mathcal{A}$ is a finite or infinite path $p \in V^+ \cup V^\omega$ in V that respects E.

Definition

A *winning condition* is a subset $\mathrm{Win} \subseteq V^\omega$.

Definition

A play is *won* by player 0, if it is finite and ends in a dead end for $\overline{\sigma}$ or if it is infinite and $p \in \mathrm{Win}$. A play is *won* by 1, if it is not won by 0.

Definition

A *game* is a tuple $J = (\mathcal{A}, \mathrm{Win})$ for some arena $\mathcal{A}$ and some winning condition Win.

Infinite Games [Maz02] revisited

Definition

An *arena* is a triple $\mathcal{A} = (V_0, V_1, E)$ where $V_0 \cap V_1 = \emptyset$, $V := V_0 \cup V_1$ and $E \subseteq V \times V$.

Definition

A *play* in $\mathcal{A}$ is a finite or infinite path $p \in V^+ \cup V^\omega$ in V that respects E.

Definition

A *winning condition* is a subset $\mathrm{Win} \subseteq V^\omega$.

Definition

A play is *won* by player 0, if it is finite and ends in a dead end for $\overline{\sigma}$ or if it is infinite and $p \in \mathrm{Win}$. A play is *won* by 1, if it is not won by 0.

Definition

A *game* is a tuple $J = (\mathcal{A}, \mathrm{Win})$ for some arena $\mathcal{A}$ and some winning condition Win. An *initialized game* is a tuple (J, v_0) consisting of a game J and an initial vertex $v_0 \in V$.

Infinite G-Games

When Groups play Games...

Definition

Let G be a group.

- A G*-arena* is an arena $\mathcal{A} = (V_0, V_1, E)$ where V_0 and V_1 are G-sets and E is equivariant.
- A G*-game* is a tuple $J = (\mathcal{A}, \mathrm{Win})$ for some G-arena $\mathcal{A}$ and some equivariant winning condition Win.
- An *initialized* G*-game* is a tuple (J, v) for some G-game J and some vertex $v \in V$.

Infinite G-Games

When Groups play Games...

Definition

Let G be a group.

- A *G-arena* is an arena $\mathcal{A} = (V_0, V_1, E)$ where V_0 and V_1 are G-sets and E is equivariant.
- A *G-game* is a tuple $J = (\mathcal{A}, \mathrm{Win})$ for some G-arena $\mathcal{A}$ and some equivariant winning condition Win.
- An *initialized G-game* is a tuple (J, v) for some G-game J and some vertex $v \in V$.

Example

Let $\chi\colon V \to C$ be an equivariant *colouring function* where $C \subseteq \mathbb{N}$ is finite. The set $\mathrm{Inf}(p)$ of all colours that occur infinitely often in an infinite play $p = v_0 v_1 \ldots \in V^\omega$ is defined as follows:

$$\mathrm{Inf}(p) := \{c \in C \mid \forall i \in \mathbb{N}.\, \exists j \geq i.\, \chi(v_j) = c\}$$

The *max-parity winning condition* is then defined as

$$\{p \in V^\omega \mid \exists k \in \mathbb{N}.\ \max(\mathrm{Inf}(p)) = 2k\}$$

which is clearly equivariant, as χ is equivariant and $\mathrm{supp}(c) = \emptyset$ for all $c \in C$.

Infinite (G-)Games
Strategies

Definition

- A *strategy* for a player $\sigma \in \{0,1\}$ is a partial function $s_\sigma \colon V^*V_\sigma \rightharpoonup V$ that respects E.
- A strategy $s_\sigma \colon V^*V_\sigma \rightharpoonup V$ is *memoryless*, if for all $p, p' \in V^*$ and $v \in V_\sigma$, we have $s_\sigma(pv) = s_\sigma(p'v)$.

Definition

1. A prefix $u = v_0v_1 \dots v_n$ of a play p is *consistent* with s_σ, if for every $0 \leq i < n$ and $v_i \in V_\sigma$ we have $s_\sigma(v_0 \dots v_i) = v_{i+1}$.
2. p is *consistent* with s_σ if every prefix of p is consistent with s_σ.

Definition

A strategy $s_\sigma \colon V^*V_\sigma \rightharpoonup V$ is a *winning strategy* for σ in (J, v_0), if every play p that is consistent with s_σ and that starts in v_0 is won by σ.

Infinite (G-)Games

Strategies

Definition

- A *strategy* for a player $\sigma \in \{0,1\}$ is a partial function $s_\sigma \colon V^*V_\sigma \rightharpoonup V$ that respects E.
- A strategy $s_\sigma \colon V^*V_\sigma \rightharpoonup V$ is *memoryless*, if for all $p, p' \in V^*$ and $v \in V_\sigma$, we have $s_\sigma(pv) = s_\sigma(p'v)$.

Definition

1. A prefix $u = v_0 v_1 \ldots v_n$ of a play p is *consistent* with s_σ, if for every $0 \le i < n$ and $v_i \in V_\sigma$ we have $s_\sigma(v_0 \ldots v_i) = v_{i+1}$.
2. p is *consistent* with s_σ if every prefix of p is consistent with s_σ.

Definition

A strategy $s_\sigma \colon V^*V_\sigma \rightharpoonup V$ is a *winning strategy* for σ in (J, v_0), if every play p that is consistent with s_σ and that starts in v_0 is won by σ.

Remark

Classical parity games enjoy memoryless determinacy [EJ91; Zie98; Küs02].

Orbit Game
Construction

Definition

Let G be a group, X be a G-set and $x \in X$ be an element of X. The *orbit* $\mathrm{orb}(x)$ of x is defined as follows:

$$\mathrm{orb}(x) := \{g \cdot x \mid g \in G\}$$

Construction

Let $J = (\mathcal{A}, \mathrm{Win})$ be a G-game. The *orbit game* $\mathrm{Orb}(J) := (\mathrm{Orb}(\mathcal{A}), \mathrm{Orb}(\mathrm{Win}))$ with $\mathrm{Orb}(\mathcal{A}) := (\mathrm{Orb}(V_0), \mathrm{Orb}(V_1), \mathrm{Orb}(E))$ is defined as follows:

$$\begin{aligned} \mathrm{Orb}(V_0) &:= \{\mathrm{orb}(v) \mid v \in V_0\} \\ \mathrm{Orb}(V_1) &:= \{\mathrm{orb}(v) \mid v \in V_1\} \\ \mathrm{Orb}(V) &:= \mathrm{Orb}(V_0) \cup \mathrm{Orb}(V_1) \\ (\mathrm{orb}(v), \mathrm{orb}(v')) \in \mathrm{Orb}(E) &:\iff (v, v') \in E \\ (\mathrm{orb}(v_0)\,\mathrm{orb}(v_1) \ldots) \in \mathrm{Orb}(\mathrm{Win}) &:\iff v_0 v_1 \ldots \in \mathrm{Win} \end{aligned}$$

Orbit Game
Theorems

Proposition

Let $J = (\mathcal{A}, \mathrm{Win})$ *be a* G*-game. If a player* $\sigma \in \{0,1\}$ *has an equivariant winning strategy* $s_\sigma \colon V_\sigma \rightharpoonup V$ *in* (J, v_0)*, then* σ *has a winning strategy* $s'_\sigma \colon \mathrm{Orb}(V_\sigma) \rightharpoonup \mathrm{Orb}(V)$ *in* $(\mathrm{Orb}(J), \mathrm{orb}(v_0))$*.*

Theorem

Let $J = (\mathcal{A}, \mathrm{Win})$ *be a* G*-game. A player* $\sigma \in \{0,1\}$ *has an equivariant memoryless winning strategy* $s_\sigma \colon V_\sigma \rightharpoonup V$ *in* (J, v_0) *iff* σ *has a memoryless winning strategy* $s'_\sigma \colon \mathrm{Orb}(V_\sigma) \rightharpoonup \mathrm{Orb}(V)$ *in* $(\mathrm{Orb}(J), \mathrm{orb}(v_0))$*.*

[1] In the discussion in the seminar we noticed a mistake in this proof: It is not guaranteed that s_σ is actually a function.

Orbit Game
Theorems

Proposition

Let $J = (\mathcal{A}, \mathrm{Win})$ be a G-game. If a player $\sigma \in \{0,1\}$ has an equivariant winning strategy $s_\sigma \colon V_\sigma \rightharpoonup V$ in (J, v_0), then σ has a winning strategy $s'_\sigma \colon \mathrm{Orb}(V_\sigma) \rightharpoonup \mathrm{Orb}(V)$ in $(\mathrm{Orb}(J), \mathrm{orb}(v_0))$.

Theorem

Let $J = (\mathcal{A}, \mathrm{Win})$ be a G-game. A player $\sigma \in \{0,1\}$ has an equivariant memoryless winning strategy $s_\sigma \colon V_\sigma \rightharpoonup V$ in (J, v_0) iff σ has a memoryless winning strategy $s'_\sigma \colon \mathrm{Orb}(V_\sigma) \rightharpoonup \mathrm{Orb}(V)$ in $(\mathrm{Orb}(J), \mathrm{orb}(v_0))$.

Proof Sketch.

For the forth direction, use that the given strategy is equivariant.

[1] In the discussion in the seminar we noticed a mistake in this proof: It is not guaranteed that s_σ is actually a function.

Orbit Game
Theorems

Proposition

Let $J = (\mathcal{A}, \mathrm{Win})$ be a G-game. If a player $\sigma \in \{0,1\}$ has an equivariant winning strategy $s_\sigma \colon V_\sigma \rightharpoonup V$ in (J, v_0), then σ has a winning strategy $s'_\sigma \colon \mathrm{Orb}(V_\sigma) \rightharpoonup \mathrm{Orb}(V)$ in $(\mathrm{Orb}(J), \mathrm{orb}(v_0))$.

Theorem

Let $J = (\mathcal{A}, \mathrm{Win})$ be a G-game. A player $\sigma \in \{0,1\}$ has an equivariant memoryless winning strategy $s_\sigma \colon V_\sigma \rightharpoonup V$ in (J, v_0) iff σ has a memoryless winning strategy $s'_\sigma \colon \mathrm{Orb}(V_\sigma) \rightharpoonup \mathrm{Orb}(V)$ in $(\mathrm{Orb}(J), \mathrm{orb}(v_0))$.

Proof Sketch.

For the forth direction, use that the given strategy is equivariant. For the back direction, we have a choice function $f \colon \mathrm{Orb}(V) \to V$ and can define $s_\sigma \colon V_\sigma \rightharpoonup V$ [1] as follows:

$$s_\sigma(\pi \cdot f(x)) := \begin{cases} \pi \cdot s'_\sigma(x) \text{ for all } \pi \in G, x \in \mathrm{Orb}(V_\sigma) \text{ such that } s'_\sigma(x) \text{ is defined} \\ \text{undefined otherwise} \end{cases}$$

□

[1] In the discussion in the seminar we noticed a mistake in this proof: It is not guaranteed that s_σ is actually a function.

Orbit Game

Applications

Corollary

Parity G-games enjoy memoryless G-determinacy:

1. *For every $v_0 \in V$ either 0 or 1 has an equivariant winning strategy in (J, v_0).*
2. *If σ has an equivariant winning strategy in (J, v_0), then σ has also an equivariant memoryless winning strategy for (J, v_0).*

Orbit Game
Applications

Corollary

Parity G-games enjoy memoryless G-determinacy:

1. *For every $v_0 \in V$ either 0 or 1 has an equivariant winning strategy in (J, v_0).*
2. *If σ has an equivariant winning strategy in (J, v_0), then σ has also an equivariant memoryless winning strategy for (J, v_0).*

Example

Parity $\mathrm{Perm}(\mathbb{A})$-games enjoy memoryless G-determinacy as well as nominal parity games.

Orbit Game
Applications

Corollary

Parity G-games enjoy memoryless G-determinacy:
1. *For every $v_0 \in V$ either 0 or 1 has an equivariant winning strategy in (J, v_0).*
2. *If σ has an equivariant winning strategy in (J, v_0), then σ has also an equivariant memoryless winning strategy for (J, v_0).*

Example

Parity $\mathrm{Perm}(\mathbb{A})$-games enjoy memoryless G-determinacy as well as nominal parity games.

Corollary

The parity $\mathrm{Perm}(\mathbb{A})$-game induced by a NAPA enjoys memoryless $\mathrm{Perm}(\mathbb{A})$-determinacy.

Conclusion

- We proved that we can reduce the input alphabet for NAPAs to a finite one, bound by the degree of the automaton.
- We discussed the (potential) need for equivariant (winning) strategies for NAPAs.
- We introduced infinite G-games for arbitrary groups.
- We proved memoryless G-determinacy for parity G-games which can be instantiated to $\mathrm{Perm}(\mathbb{A})$-games.

[EJ91] E. Emerson and C. Jutla. “Tree Automata, Mu-Calculus and Determinacy”. In: *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*. Oct. 1991, pages 368–377. DOI: 10.1109/SFCS.1991.185392.

[Fra+25] F. Frank, D. Hausmann, S. Milius, H. Urbat, and L. Schröder. “Alternating Nominal Automata with Name Allocation”. In: *Proceedings of the 40th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2025)*. Edited by L. Birkedal and B. König. IEEE Computer Society, 2025. DOI: 10.1109/LICS65433.2025.00012.

[Kön27] 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), pages 121–130. ISSN: 0324-5462.

[Küs02] R. Küsters. “Memoryless Determinacy of Parity Games”. In: *Automata Logics, and Infinite Games: A Guide to Current Research*. Edited by E. Grädel, W. Thomas, and T. Wilke. Berlin, Heidelberg: Springer, 2002, pages 95–106. ISBN: 978-3-540-36387-3. DOI: 10.1007/3-540-36387-4_6.

[Maz02] R. Mazala. “Infinite Games”. In: *Automata Logics, and Infinite Games: A Guide to Current Research*. Edited by E. Grädel, W. Thomas, and T. Wilke. Berlin, Heidelberg: Springer, 2002, pages 23–38. ISBN: 978-3-540-36387-3. DOI: 10.1007/3-540-36387-4_2.

[Pit13] A. M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*. 1st edition. Cambridge University Press, May 2013. ISBN: 978-1-139-08467-3. 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*. Edited by J. Esparza and A. S. Murawski. Berlin, Heidelberg: Springer, 2017, pages 124–142. ISBN: 978-3-662-54458-7. 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: *32nd International Conference on Concurrency Theory (CONCUR 2021)*. Edited by S. Haddad and D. Varacca. Volume 203. Leibniz International Proceedings in Informatics (Lipics). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 4:1–4:16. ISBN: 978-3-95977-203-7. DOI: 10.4230/LIPIcs.CONCUR.2021.4.

[Zie98] W. Zielonka. “Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees”. In: *Theoretical Computer Science* 200.1 (June 1998), pages 135–183. ISSN: 0304-3975. DOI: 10.1016/S0304-3975(98)00009-7.