First-order logic for permutations

Mathilde Bouvel

talk based on joint work with M. Albert and V. Féray

Universität Zürich[UZH]

Discrete Maths Seminar, Uni. Zürich, May 2018.

What is a permutation (of size n)?

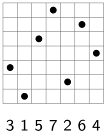
- A bijection from $\{1, 2, \ldots, n\}$ to itself,
- or more generally from X to X, for $|X| = n$.

Ex.: $\sigma = (1, 3, 5, 2)(4, 7)(6)$

What is a permutation (of size n)?

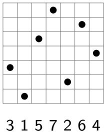
- A bijection from $\{1, 2, \ldots, n\}$ to itself,
- or more generally from X to X, for $|X| = n$.
- A word containing exactly once each letter from $\{1, 2, \ldots, n\}$,
- or more visually a diagram.

Ex.: $\sigma = (1, 3, 5, 2)(4, 7)(6) = 3\ 1\ 5\ 7\ 2\ 6\ 4 =$



What is a permutation (of size n)?

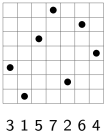
- A bijection from $\{1, 2, \ldots, n\}$ to itself,
- or more generally from X to X, for $|X| = n$.
- A word containing exactly once each letter from $\{1, 2, \ldots, n\}$,
- or more visually a diagram.

Ex.: $\sigma = (1, 3, 5, 2)(4, 7)(6) = 3\ 1\ 5\ 7\ 2\ 6\ 4 =$ 

- The questions addressed are different, depending on the point of view.
- Very few results consider both points of view simultaneously.
- The two points of view are believed to be rather orthogonal.

What is a permutation (of size n)?

- A bijection from $\{1, 2, \ldots, n\}$ to itself,
- or more generally from X to X, for $|X| = n$.
- A word containing exactly once each letter from $\{1, 2, \ldots, n\}$,
- or more visually a diagram.

Ex.: $\sigma = (1,3,5,2)(4,7)(6) = 3\ 1\ 5\ 7\ 2\ 6\ 4 =$ 

- The questions addressed are different, depending on the point of view.
- Very few results consider both points of view simultaneously.
- The two points of view are believed to be rather orthogonal.

Goal: Give a "proof" that the two points of view are hardly reconciled.

How? Logic to the rescue!

Formalize each point of view as a logic for permutations.
More precisely, we consider two first-order (logical) theories.

How? Logic to the rescue!

Formalize each point of view as a logic for permutations.
More precisely, we consider two first-order (logical) theories.

For each theory,

- permutations are models of our theory,
- (logical) formulas express properties of the permutations.

How? Logic to the rescue!

Formalize each point of view as a logic for permutations.
More precisely, we consider two first-order (logical) theories.

For each theory,

- permutations are models of our theory,
- (logical) formulas express properties of the permutations.

To prove that the two points of view are essentially different, we study the expressivity of the theories:

- describe properties expressible in each theory,
- show that the properties expressible in both theories are trivial.

Two logics for permutations

TOOB: models

TOOB: the Theory Of One Bijection *(already appeared in the literature)*

TOOB: models

TOOB: the Theory Of One Bijection *(already appeared in the literature)*

Two components of a logical theory:

- its formulas = what the theory can say about its models *syntax*
- its models = the objects the theory talks about *interpretation*

TOOB: models

TOOB: the Theory Of One Bijection *(already appeared in the literature)*

Two components of a logical theory:

- its formulas = what the theory can say about its models *syntax*
- its models = the objects the theory talks about *interpretation*

(Finite) models of TOOB:
Pairs (X, R_X) where X is a finite set and R_X a binary relation on X.

TOOB: models

TOOB: the Theory Of One Bijection *(already appeared in the literature)*

Two components of a logical theory:

- its formulas = what the theory can say about its models *syntax*
- its models = the objects the theory talks about *interpretation*

(Finite) models of TOOB:
Pairs (X, R_X) where X is a finite set and R_X a binary relation on X.

Axioms of TOOB: ensure that R_X is a bijection from X to X.

TOOB: models

TOOB: the Theory Of One Bijection *(already appeared in the literature)*

Two components of a logical theory:

- its formulas = what the theory can say about its models *syntax*
- its models = the objects the theory talks about *interpretation*

(Finite) models of TOOB:
Pairs (X, R_X) where X is a finite set and R_X a binary relation on X.

Axioms of TOOB: ensure that R_X is a bijection from X to X.

- Surjectivity: $\forall x \, \exists y \; yRx$
- Injectivity: $\neg \exists x, y, z \, (x \neq y \wedge xRz \wedge yRz)$

TOOB: models

TOOB: the Theory Of One Bijection *(already appeared in the literature)*

Two components of a logical theory:

- its formulas = what the theory can say about its models *syntax*
- its models = the objects the theory talks about *interpretation*

(Finite) models of TOOB:
Pairs (X, R_X) where X is a finite set and R_X a binary relation on X.

Axioms of TOOB: ensure that R_X is a bijection from X to X.

Permutations are models, and every model is a permutation.
(Possibly, up to a conjugating by a bijection between X and $\{1, 2, \ldots, n\}$.)

The relation R_σ associated to σ of size n is given by:

$$i \ R_\sigma \ \sigma(i) \text{ for all } i \leq n$$

TOOB: formulas

- Atomic formulas of TOOB are $x = y$ and xRy, for any variables x and y.
  - $\rightsquigarrow$ A variable is intended as representing an element of the permutation.

TOOB: formulas

- Atomic formulas of TOOB are $x = y$ and xRy, for any variables x and y.
  - $\rightsquigarrow$ A variable is intended as representing an element of the permutation.
- Formulas (ϕ, or $\phi(\mathbf{x})$) are obtained inductively from the atomic ones using the connectives and quantifiers.
  - $\rightsquigarrow$ $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$.
  - $\rightsquigarrow$ We restrict ourselves to first-order logic, so we consider only quantification on variables: $\exists x\, \phi$, $\forall x\, \phi$.

TOOB: formulas

- Atomic formulas of TOOB are $x = y$ and xRy, for any variables x and y.
  - $\rightsquigarrow$ A variable is intended as representing an element of the permutation.
- Formulas (ϕ, or $\phi(\mathbf{x})$) are obtained inductively from the atomic ones using the connectives and quantifiers.
  - $\rightsquigarrow$ $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$.
  - $\rightsquigarrow$ We restrict ourselves to first-order logic, so we consider only quantification on variables: $\exists x\, \phi$, $\forall x\, \phi$.
- Sentences (ψ) are formulas where all variables are quantified (no free variable).

Ex.: $\phi(x) := xRx$ and $\psi := \exists x\, xRx$.

TOOB: formulas

- Atomic formulas of TOOB are $x = y$ and xRy, for any variables x and y.
  - $\rightsquigarrow$ A variable is intended as representing an element of the permutation.
- Formulas (ϕ, or $\phi(\mathbf{x})$) are obtained inductively from the atomic ones using the connectives and quantifiers.
  - $\rightsquigarrow$ $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$.
  - $\rightsquigarrow$ We restrict ourselves to first-order logic, so we consider only quantification on variables: $\exists x\, \phi$, $\forall x\, \phi$.
- Sentences (ψ) are formulas where all variables are quantified (no free variable).

Ex.: $\phi(x) := xRx$ and $\psi := \exists x\, xRx$.

A model of a sentence ψ is a model which in addition satisfies ψ.

Ex.: The models of $\exists x\, xRx$ are the permutations having a fixed point.

TOOB: expressivity

A property of permutations is expressible in a theory (here, TOOB) if it can be described by a sentence, *i.e.*, there is a sentence whose models are exactly the permutations for which this property holds.

Ex.: $\psi := \exists x\, xRx$ expresses the property of having a fixed point.

TOOB: expressivity

A property of permutations is expressible in a theory (here, TOOB) if it can be described by a sentence, *i.e.*, there is a sentence whose models are exactly the permutations for which this property holds.

Ex.: $\psi := \exists x \, xRx$ expresses the property of having a fixed point.

Definition-by-example of $\models$: we write $\sigma \models \psi$ when σ has a fixed point.

TOOB: expressivity

A property of permutations is expressible in a theory (here, TOOB) if it can be described by a sentence, *i.e.*, there is a sentence whose models are exactly the permutations for which this property holds.

Ex.: $\psi := \exists x \, xRx$ expresses the property of having a fixed point.

Definition-by-example of $\models$: we write $\sigma \models \psi$ when σ has a fixed point.

In TOOB, only properties about the cycle decomposition of a permutation are expressible.

But not all such! For instance, being a full cycle is not expressible.

TOOB: expressivity

A property of permutations is expressible in a theory (here, TOOB) if it can be described by a sentence, *i.e.*, there is a sentence whose models are exactly the permutations for which this property holds.

Ex.: $\psi := \exists x\, xRx$ expresses the property of having a fixed point.

Definition-by-example of $\models$: we write $\sigma \models \psi$ when σ has a fixed point.

In TOOB, only properties about the cycle decomposition of a permutation are expressible.

But not all such! For instance, being a full cycle is not expressible.

Thm.: If $\sigma \models \psi$, then for any τ in the conjugacy class of σ, $\tau \models \psi$.

In other words, TOOB does not distinguish between conjugate permutations.

TOTO: syntax and models

TOTO: the Theory Of Two Orders *(new as a logic for permutations)*

TOTO: syntax and models

TOTO: the Theory Of Two Orders *(new as a logic for permutations)*

- Symbols available: same logical symbols (including $=$), no relation symbol R, but instead, two binary relation symbols $<_P$ and $<_V$

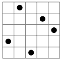
TOTO: syntax and models

TOTO: the Theory Of Two Orders *(new as a logic for permutations)*

- Symbols available: same logical symbols (including $=$), no relation symbol R, but instead, two binary relation symbols $<_P$ and $<_V$
- Axioms: ensure that $<_P$ and $<_V$ represent total orders.

TOTO: syntax and models

TOTO: the Theory Of Two Orders *(new as a logic for permutations)*

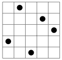
- Symbols available: same logical symbols (including $=$), no relation symbol R, but instead, two binary relation symbols $<_P$ and $<_V$
- Axioms: ensure that $<_P$ and $<_V$ represent total orders.
- Models: permutations as pairs of total orders on a finite set:
  - $<_P$ represents the position order between the elements;
  - $<_V$ represents their value order.
  - Ex.: $\sigma =$  is represented for instance by $(\{a, b, c, d, e\}, \lhd, \blacktriangleleft)$

    25143

    where $a \lhd b \lhd c \lhd d \lhd e$ and $c \blacktriangleleft a \blacktriangleleft e \blacktriangleleft d \blacktriangleleft b$.

TOTO: syntax and models

TOTO: the Theory Of Two Orders *(new as a logic for permutations)*

- Symbols available: same logical symbols (including $=$), no relation symbol R, but instead, two binary relation symbols $<_P$ and $<_V$
- Axioms: ensure that $<_P$ and $<_V$ represent total orders.
- Models: permutations as pairs of total orders on a finite set:
  - $<_P$ represents the position order between the elements;
  - $<_V$ represents their value order.
  - Ex.: $\sigma =$  25143 is represented for instance by $(\{a, b, c, d, e\}, \lhd, \blacktriangleleft)$

    where $a \lhd b \lhd c \lhd d \lhd e$ and $c \blacktriangleleft a \blacktriangleleft e \blacktriangleleft d \blacktriangleleft b$.

Summary of differences:

- TOOB speaks about the cycle structure but the total order on $\{1, 2, \ldots, n\}$ is lost.
- TOTO speaks about the relative order of the elements, but the cycle structure is lost.

TOTO: expressivity

- Unlike TOOB, TOTO does distinguish between any two different permutations.
- In other words, for any permutation σ, there exists a sentence whose only model is σ (up to isomorphism on the ground set).

TOTO: expressivity

- Unlike TOOB, TOTO does distinguish between any two different permutations.
- In other words, for any permutation σ, there exists a sentence whose only model is σ (up to isomorphism on the ground set).

Some concepts expressible in TOTO:

- Containment/avoidance of a classical pattern;

  Ex.: Avoidance of 231 is expressed by the sentence

$$\phi_{Av(231)} := \neg\, \exists x\, \exists y\, \exists z\ (x <_P y <_P z) \ \wedge\ (z <_V x <_V y)$$

TOTO: expressivity

- Unlike TOOB, TOTO does distinguish between any two different permutations.
- In other words, for any permutation σ, there exists a sentence whose only model is σ (up to isomorphism on the ground set).

Some concepts expressible in TOTO:

- Containment/avoidance of a classical pattern;
- Extension to consecutive/vincular/mesh patterns (and further);

TOTO: expressivity

- Unlike TOOB, TOTO does distinguish between any two different permutations.
- In other words, for any permutation σ, there exists a sentence whose only model is σ (up to isomorphism on the ground set).

Some concepts expressible in TOTO:

- Containment/avoidance of a classical pattern;
- Extension to consecutive/vincular/mesh patterns (and further);
- $\oplus$- (resp. $\ominus$-)decomposability;
- Generalization to being an inflation of π for any π;
- Being simple;

TOTO: expressivity

- Unlike TOOB, TOTO does distinguish between any two different permutations.
- In other words, for any permutation σ, there exists a sentence whose only model is σ (up to isomorphism on the ground set).

Some concepts expressible in TOTO:

- Containment/avoidance of a classical pattern;
- Extension to consecutive/vincular/mesh patterns (and further);
- $\oplus$- (resp. $\ominus$-)decomposability;
- Generalization to being an inflation of π for any π;
- Being simple;
- Being West-k-stack sortable, for any k
  (+ construction of the corresponding sentences)

TOTO and stack-sorting

Let **S** be the stack-sorting operator.

- Known description by pattern-avoidance of **S**-, $\mathbf{S}^2$- and $\mathbf{S}^3$-sortable permutations.
- But the notion of patterns are more and more complicated for every additional **S**.

TOTO and stack-sorting

Let $\mathbf{S}$ be the stack-sorting operator.

- Known description by pattern-avoidance of $\mathbf{S}$-, $\mathbf{S}^2$- and $\mathbf{S}^3$-sortable permutations.
- But the notion of patterns are more and more complicated for every additional $\mathbf{S}$.

$\Rightarrow$ Prefer logic to patterns:

Thm.:
For any k, the property of being $\mathbf{S}^k$-sortable is expressible in TOTO.

TOTO and stack-sorting

Let $\mathbf{S}$ be the stack-sorting operator.

- Known description by pattern-avoidance of $\mathbf{S}$-, $\mathbf{S}^2$- and $\mathbf{S}^3$-sortable permutations.
- But the notion of patterns are more and more complicated for every additional $\mathbf{S}$.

$\Rightarrow$ Prefer logic to patterns:

Thm.:
For any k, the property of being $\mathbf{S}^k$-sortable is expressible in TOTO.

A formula ϕ_k expressing it may be derived automatically, starting from $\phi_1 = \phi_{Av(231)}$ and iterating the operation of replacing every $x <_P y$ by $\left(x <_P y \wedge \exists z \left(x <_P z \leq_P y \wedge x <_V z\right)\right) \vee \left(y <_P x \wedge \forall z \left(y <_P z \leq_P x \to z <_V y\right)\right)$ to obtain ϕ_k from ϕ_{k-1}.

Obtained formulas are complicated... Automatic simplification open.

TOTO and stack-sorting

Let **S** be the stack-sorting operator.

- Known description by pattern-avoidance of **S**-, $\mathbf{S}^2$- and $\mathbf{S}^3$-sortable permutations.
- But the notion of patterns are more and more complicated for every additional **S**.

$\Rightarrow$ Prefer logic to patterns:

Thm.:
For any k, the property of being $\mathbf{S}^k$-sortable is expressible in TOTO.

A formula ϕ_k expressing it may be derived automatically, starting from $\phi_1 = \phi_{Av(231)}$ and iterating the operation of replacing every $x <_P y$ by $\left(x <_P y \wedge \exists z \left(x <_P z \leq_P y \wedge x <_V z\right)\right) \vee \left(y <_P x \wedge \forall z \left(y <_P z \leq_P x \rightarrow z <_V y\right)\right)$ to obtain ϕ_k from ϕ_{k-1}.

Obtained formulas are complicated... Automatic simplification open.

Rk.: This extends to other sorting procedures/devices.

Inexpressibility results in TOTO

Inexpressibility of fixed points

Thm.: There is no sentence ψ in TOTO such that
$\sigma \models \psi$ if and only if σ has a fixed point.

Inexpressibility of fixed points

Thm.: There is no sentence ψ in TOTO such that
$\sigma \models \psi$ if and only if σ has a fixed point.

Intermezzo: Expressing properties of elements of permutations.

- A formula $\phi(x)$ with one (or several) free variable(s) expresses properties of one (or several) element(s) of a permutation.
- Ex: xRx expresses the property that a given element is a fixed point: For π a permutation and a an element of π, we write $(\pi, a) \models \phi(x)$ when a is a fixed point of π.

Inexpressibility of fixed points

Thm.: There is no sentence ψ in TOTO such that
$\sigma \models \psi$ if and only if σ has a fixed point.

Intermezzo: Expressing properties of elements of permutations.

- A formula $\phi(x)$ with one (or several) free variable(s) expresses properties of one (or several) element(s) of a permutation.
- Ex: xRx expresses the property that a given element is a fixed point: For π a permutation and a an element of π, we write $(\pi, a) \models \phi(x)$ when a is a fixed point of π.

Cor.: There is no formula with one free variable in TOTO expressing the property that a given element is a fixed point.

Inexpressibility of fixed points

Thm.: There is no sentence ψ in TOTO such that
$\sigma \models \psi$ if and only if σ has a fixed point.

Inexpressibility of fixed points

Thm.: There is no sentence ψ in TOTO such that
$\sigma \models \psi$ if and only if σ has a fixed point.

Proof strategy:

- Assume such a sentence ψ exists.
  Call k its quantifier depth (=max. number of nested quantifiers in ψ).
- Exhibit two permutations σ and σ' such that
  - σ has a fixed point but σ' does not; and
  - $\sigma \models \psi$ if and only if $\sigma' \models \psi$.
    (Actually, σ and σ' satisfy the same sentences of quantifier depth at most k)

Inexpressibility of fixed points

Thm.: There is no sentence ψ in TOTO such that
$\sigma \models \psi$ if and only if σ has a fixed point.

Proof strategy:

- Assume such a sentence ψ exists.
  Call k its quantifier depth (=max. number of nested quantifiers in ψ).
- Exhibit two permutations σ and σ' such that
  - σ has a fixed point but σ' does not; and
  - $\sigma \models \psi$ if and only if $\sigma' \models \psi$.
    (Actually, σ and σ' satisfy the same sentences of quantifier depth at most k)

To show that two permutations satisfy the same sentences, use the Ehrenfeucht-Fraïssé Theorem:

> Two permutations σ and σ' satisfy the same sentences of quantifier depth at most k if and only if Duplicator wins the EF-game with k rounds on σ and σ'.

EF-games (a.k.a. Duplicator-Spoiler games)

The setting:

- Two players: Duplicator (D) and Spoiler (S).
- They play on a pair of permutations σ and σ'.
- Goal of D: show that σ and σ' cannot be distinguish in k rounds.
- Goal of S: show that σ and σ' are different.

EF-games (a.k.a. Duplicator-Spoiler games)

The setting:

- Two players: Duplicator (D) and Spoiler (S).
- They play on a pair of permutations σ and σ'.
- Goal of D: show that σ and σ' cannot be distinguish in k rounds.
- Goal of S: show that σ and σ' are different.

At each round i:

- S picks an element s_i in σ or s'_i in σ';
- D replicates with an element s'_i or s_i in the other permutation.

EF-games (a.k.a. Duplicator-Spoiler games)

The setting:

- Two players: Duplicator (D) and Spoiler (S).
- They play on a pair of permutations σ and σ'.
- Goal of D: show that σ and σ' cannot be distinguish in k rounds.
- Goal of S: show that σ and σ' are different.

At each round i:

- S picks an element s_i in σ or s'_i in σ';
- D replicates with an element s'_i or s_i in the other permutation.

Winner of the EF-game with k rounds:

- D if $\mathbf{s} = (s_1, \ldots, s_k)$ and $\mathbf{s}' = (s'_1, \ldots, s'_k)$ are isomorphic,
  i.e., if the position- and value-orders on $\mathbf{s}$ and $\mathbf{s}'$ are identical;
- S otherwise.

Inexpressibility of fixed points: Proof

Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Inexpressibility of fixed points: Proof

Goal: For each k, exhibit σ and σ' such that

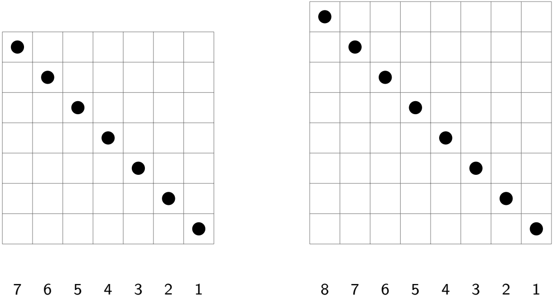
- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.

Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.
For $k = 3$:

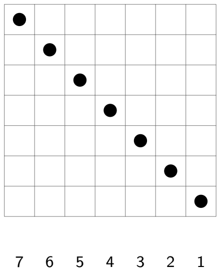


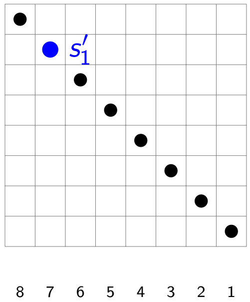
S and D alternate turns.

Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.
For $k = 3$:



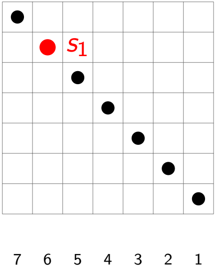


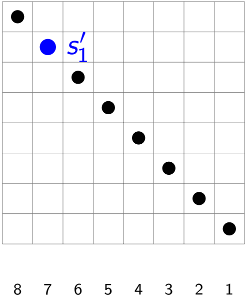
S and D alternate turns.

Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.
For $k = 3$:





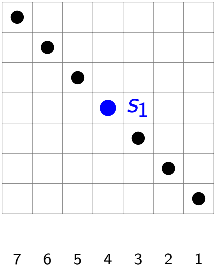
S and D alternate turns.

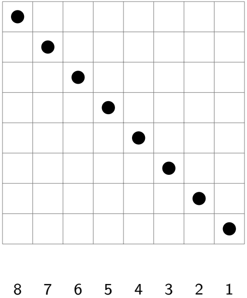
Inexpressibility of fixed points: Proof

Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.
For $k = 3$:





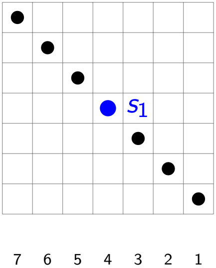
S and D alternate turns.

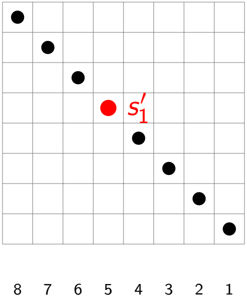
Inexpressibility of fixed points: Proof

Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.
For $k = 3$:





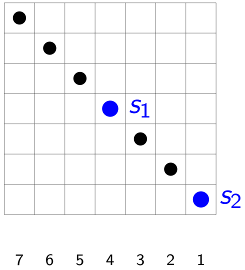
S and D alternate turns.

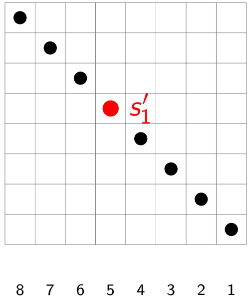
Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.

For $k = 3$:





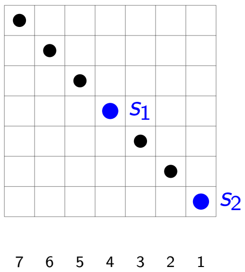
S and D alternate turns.

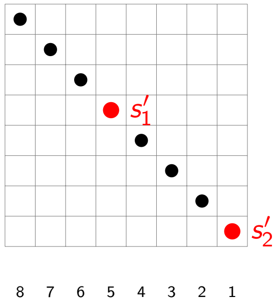
Inexpressibility of fixed points: Proof

Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.
For $k = 3$:



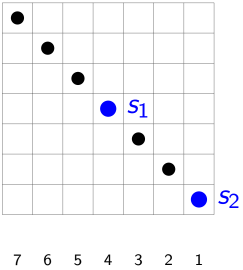


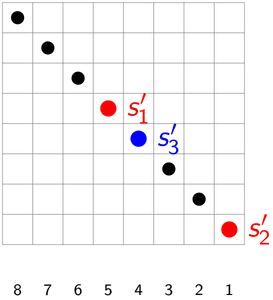
S and D alternate turns.

Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.
For $k = 3$:





S and D alternate turns.

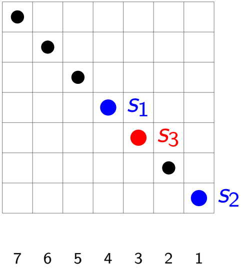
Inexpressibility of fixed points: Proof

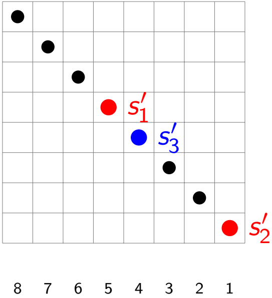
Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.

For $k = 3$:





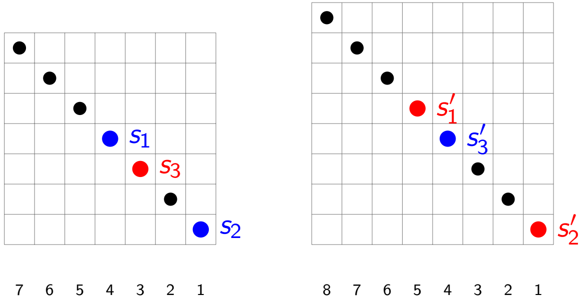
S and *D* alternate turns.

Inexpressibility of fixed points: Proof

Goal: For each k, exhibit σ and σ' such that

- σ has a fixed point but σ' does not;
- D wins the EF-game with k rounds on σ and σ'.

Answer: σ and σ' are decreasing permutations of sizes $2^k - 1$ and 2^k.
For $k = 3$:



S and D alternate turns. After 3 rounds, D wins!

Intersection of TOTO **and** TOOB

Properties expressible in one/both theories

Examples of properties expressible in one of TOOB and TOTO only:

- Having a fixed point: expressible in TOOB but not in TOTO;
- Avoiding a 231-pattern: expressible in TOTO but not in TOOB.
  (TOOB does not distinguish between $231 = (1,2,3)$ and $312 = (1,3,2)$)

Properties expressible in one/both theories

Examples of properties expressible in one of TOOB and TOTO only:

- Having a fixed point: expressible in TOOB but not in TOTO;
- Avoiding a 231-pattern: expressible in TOTO but not in TOOB.
  (TOOB does not distinguish between $231 = (1, 2, 3)$ and $312 = (1, 3, 2)$)

Which properties are expressible in both TOOB and TOTO?

Properties expressible in one/both theories

Examples of properties expressible in one of TOOB and TOTO only:

- Having a fixed point: expressible in TOOB but not in TOTO;
- Avoiding a 231-pattern: expressible in TOTO but not in TOOB.
  (TOOB does not distinguish between $231 = (1,2,3)$ and $312 = (1,3,2)$)

Which properties are expressible in both TOOB and TOTO?

Thm.: Such properties $\mathcal{P}$ are eventually true or eventually false, where eventually means "for all permutations of sufficiently large support".

Dfn.: The support of a permutation is the set of the non-fixed points.

Properties expressible in one/both theories

Examples of properties expressible in one of TOOB and TOTO only:

- Having a fixed point: expressible in TOOB but not in TOTO;
- Avoiding a 231-pattern: expressible in TOTO but not in TOOB.
  (TOOB does not distinguish between $231 = (1, 2, 3)$ and $312 = (1, 3, 2)$)

Which properties are expressible in both TOOB and TOTO?

Thm.: Such properties $\mathcal{P}$ are eventually true or eventually false, where eventually means "for all permutations of sufficiently large support".

i.e., $\mathcal{P}$ is satisfied for *all permutations of sufficiently large support or there is a bound on the size of the support of any permutation* satisfying $\mathcal{P}$.

Dfn.: The support of a permutation is the set of the non-fixed points.

Properties expressible in one/both theories

Examples of properties expressible in one of TOOB and TOTO only:

- Having a fixed point: expressible in TOOB but not in TOTO;
- Avoiding a 231-pattern: expressible in TOTO but not in TOOB.
  (TOOB does not distinguish between $231 = (1, 2, 3)$ and $312 = (1, 3, 2)$)

Which properties are expressible in both TOOB and TOTO?

Thm.: Such properties $\mathcal{P}$ are eventually true or eventually false, where eventually means "for all permutations of sufficiently large support".

i.e., $\mathcal{P}$ is satisfied for *all permutations of sufficiently large support or there is a bound on the size of the support of any permutation* satisfying $\mathcal{P}$.

Dfn.: The support of a permutation is the set of the non-fixed points.

The proof uses EF-games.

Properties expressible in one/both theories

Examples of properties expressible in one of TOOB and TOTO only:

- Having a fixed point: expressible in TOOB but not in TOTO;
- Avoiding a 231-pattern: expressible in TOTO but not in TOOB.
  (TOOB does not distinguish between $231 = (1, 2, 3)$ and $312 = (1, 3, 2)$)

Which properties are expressible in both TOOB and TOTO?

Thm.: Such properties $\mathcal{P}$ are eventually true or eventually false, where eventually means "for all permutations of sufficiently large support".

i.e., $\mathcal{P}$ is satisfied for *all permutations of sufficiently large support or there is a bound on the size of the support of any permutation* satisfying $\mathcal{P}$.

Dfn.: The support of a permutation is the set of the non-fixed points.

The proof uses EF-games.

$\Rightarrow$ The intersection of TOOB and TOTO is trivial, so, as claimed, permutations-as-elts-of-the-symmetric-group $\neq$ permutations-as-words.

Exact description of the intersection of TOOB and TOTO

For any partition λ, define

- $\mathcal{C}_\lambda$ the set of permutations of cycle-type λ;
- $\mathcal{D}_\lambda = \biguplus_{k\geq 0} \mathcal{C}_{\lambda\cup(1^k)}$.

Exact description of the intersection of TOOB and TOTO

For any partition λ, define

- $\mathcal{C}_\lambda$ the set of permutations of cycle-type λ;
- $\mathcal{D}_\lambda = \biguplus_{k\geq 0} \mathcal{C}_{\lambda\cup(1^k)}$.

Thm.: A set E of permutations is defined by a property expressible in both TOOB and TOTO if and only if it belongs to the Boolean algebra generated by all $\mathcal{C}_\lambda$ and $\mathcal{D}_\lambda$ (where λ runs over all partitions).

Exact description of the intersection of TOOB and TOTO

For any partition λ, define

- $\mathcal{C}_\lambda$ the set of permutations of cycle-type λ;
- $\mathcal{D}_\lambda = \biguplus_{k \geq 0} \mathcal{C}_{\lambda \cup (1^k)}$.

Thm.: A set E of permutations is defined by a property expressible in both TOOB and TOTO if and only if it belongs to the Boolean algebra generated by all $\mathcal{C}_\lambda$ and $\mathcal{D}_\lambda$ (where λ runs over all partitions).

Rk: This is more precise than the previous theorem. Indeed:

- in $\mathcal{C}_\lambda$ and $\mathcal{D}_\lambda$ there is a bound on the size of the support.
- the property *either E contains all permutations of sufficiently large support, or there is a bound on the size of the support of permutations in E* is stable by union, intersection and complement.

Exact description of the intersection of TOOB and TOTO

For any partition λ, define

- $\mathcal{C}_\lambda$ the set of permutations of cycle-type λ;
- $\mathcal{D}_\lambda = \biguplus_{k\geq 0} \mathcal{C}_{\lambda\cup(1^k)}$.

Thm.: A set E of permutations is defined by a property expressible in both TOOB and TOTO if and only if it belongs to the Boolean algebra generated by all $\mathcal{C}_\lambda$ and $\mathcal{D}_\lambda$ (where λ runs over all partitions).

Rk: This is more precise than the previous theorem.

Tricks/tools in the proof:

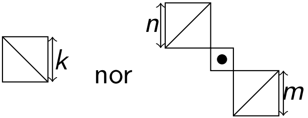
- expressing $\mathcal{D}_\lambda$ in TOTO;
- use previous theorem to write E as a finite union of $\mathcal{C}_\lambda$'s and $\mathcal{D}_\lambda$'s;
- and more EF games!

Some other things we know (or not)

- Characterization of the permutation classes $\mathcal{C}$ such that "having a fixed point" is expressible in the restriction of TOTO to $\mathcal{C}$.

- Characterization of the permutation classes $\mathcal{C}$ such that "having a fixed point" is expressible in the restriction of TOTO to $\mathcal{C}$.

The condition is: there exist k, n, m such that $\mathcal{C}$ does not contain

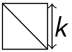


.

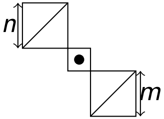
Some other things we know (or not)

- Characterization of the permutation classes $\mathcal{C}$ such that "having a fixed point" is expressible in the restriction of TOTO to $\mathcal{C}$.

The condition is: there exist k, n, m such that $\mathcal{C}$ does not contain



nor



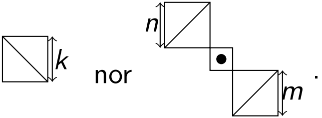
.

- *Formula-variant*: Describe classes TOTO can express (by $\phi(x)$) the property that a given element is a fixed point. The same as above!

Some other things we know (or not)

- Characterization of the permutation classes $\mathcal{C}$ such that "having a fixed point" is expressible in the restriction of TOTO to $\mathcal{C}$.

The condition is: there exist k, n, m such that $\mathcal{C}$ does not contain

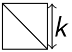


- *Formula-variant*: Describe classes TOTO can express (by $\phi(x)$) the property that a given element is a fixed point. The same as above!
- Extension to description of classes where TOTO can express that two (resp. more) given elements form a transposition (resp. cycle)

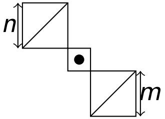
Some other things we know (or not)

- Characterization of the permutation classes $\mathcal{C}$ such that "having a fixed point" is expressible in the restriction of TOTO to $\mathcal{C}$.

The condition is: there exist k, n, m such that $\mathcal{C}$ does not contain



nor



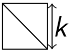
.

- *Formula-variant*: Describe classes TOTO can express (by $\phi(x)$) the property that a given element is a fixed point. The same as above!
- Extension to description of classes where TOTO can express that two (resp. more) given elements form a transposition (resp. cycle)
- But we don't know in which classes the existence of a transposition (resp. cycle of a given size) is expressible in TOTO.

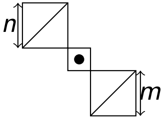
Some other things we know (or not)

- Characterization of the permutation classes $\mathcal{C}$ such that "having a fixed point" is expressible in the restriction of TOTO to $\mathcal{C}$.

The condition is: there exist k, n, m such that $\mathcal{C}$ does not contain



nor



.

- *Formula-variant*: Describe classes TOTO can express (by $\phi(x)$) the property that a given element is a fixed point. The same as above!
- Extension to description of classes where TOTO can express that two (resp. more) given elements form a transposition (resp. cycle)
- But we don't know in which classes the existence of a transposition (resp. cycle of a given size) is expressible in TOTO.
- Further project with M. Noy: Prove convergence laws in permutation classes (for properties expressible in TOTO).