Monadic second-order logic of permutations

Michal Opler
Joint work with Vít Jelínek

Czech Technical University in Prague
Charles University

PP 2023
July 7, 2023

First-order logic

First-order (FO) formulas are formed from equality $x = y$ and two special relation symbols $x <_1 y$, $x <_2 y$ using logical connectives ($\wedge, \vee, \neg, \rightarrow, \leftrightarrow$) and quantifiers ($\exists x, \forall x$)

FO sentence is a first-order logic formula with all variables quantified.

First-order logic

First-order (FO) formulas are formed from equality $x = y$ and two special relation symbols $x <_1 y$, $x <_2 y$ using logical connectives ($\wedge, \vee, \neg, \rightarrow, \leftrightarrow$) and quantifiers ($\exists x, \forall x$)

FO sentence is a first-order logic formula with all variables quantified.

Example

The existence of a classical pattern 132 is expressed by the FO sentence

$$\exists\, x, y, z\ (x <_1 y)\ \wedge\ (y <_1 z)\ \wedge\ (x <_2 z)\ \wedge\ (z <_2 y).$$

First-order logic

First-order (FO) formulas are formed from equality $x = y$ and two special relation symbols $x <_1 y$, $x <_2 y$ using logical connectives ($\wedge, \vee, \neg, \rightarrow, \leftrightarrow$) and quantifiers ($\exists x, \forall x$)

FO sentence is a first-order logic formula with all variables quantified.

Example

The existence of a classical pattern 132 is expressed by the FO sentence

$$\exists\, x, y, z\ (x <_1 y) \ \wedge\ (y <_1 z) \ \wedge\ (x <_2 z) \ \wedge\ (z <_2 y).$$

We say that a permutation π models an FO sentence φ if π satisfies φ. We denote this by $\pi \models \varphi$.

Expressive power of FO logic

Proposition (Albert, Bouvel, Féray 2020)

Each of the following properties of a permutation σ can be defined using an FO sentence:

- σ *contains a fixed classical pattern* π*,*
- σ *contains a fixed mesh/barred/partially-ordered pattern,*
- σ *belongs to the class Grid*$(\mathcal{M})$ *for a fixed gridding matrix* $\mathcal{M}$*,*
- σ *is* $\oplus$*-decomposable (and symmetrically for* $\ominus$*-decomposability),*
- σ *has exactly* k *inversions,*
- σ *is simple,*
- σ *is West-*k*-stack-sortable for a fixed* k*,*

$\vdots$

Expressive power of FO logic

Proposition (Albert, Bouvel, Féray 2020)

Each of the following properties of a permutation σ can be defined using an FO sentence:

- σ *contains a fixed classical pattern* π,
- σ *contains a fixed mesh/barred/partially-ordered pattern,*
- σ *belongs to the class Grid*$(\mathcal{M})$ *for a fixed gridding matrix* $\mathcal{M}$,
- σ *is* $\oplus$*-decomposable (and symmetrically for* $\ominus$*-decomposability),*
- σ *has exactly k inversions,*
- σ *is simple,*
- σ *is West-k-stack-sortable for a fixed* k,

$\vdots$

Proposition (Albert, Bouvel, Féray 2020)

The property of having a fixed point is not expressible by an FO sentence.

Monadic second-order logic

Monadic second-order (MSO) formulas extend FO formulas by introducing set variables (X), quantification over them ($\exists X, \forall X$) and set membership predicates ($x \in X$).

Monadic second-order logic

Monadic second-order (MSO) formulas extend FO formulas by introducing set variables (X), quantification over them ($\exists X, \forall X$) and set membership predicates ($x \in X$).

Example

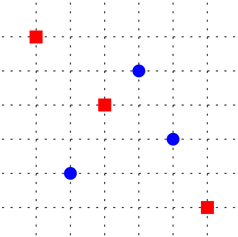
$$\begin{aligned}
\mathit{partition}(X, Y) &= \forall x \, [(x \in X \vee x \in Y) \wedge \neg(x \in X \wedge x \in Y)] \, , \\
\mathit{increasing}(X) &= \forall x, y \, [(x \in X \wedge y \in X) \rightarrow (x <_1 y \leftrightarrow x <_2 y)] \, , \\
\mathit{decreasing}(X) &= \forall x, y \, [(x \in X \wedge y \in X) \rightarrow (x <_1 y \leftrightarrow y <_2 x)] \, .
\end{aligned}$$

Using these predicates, we can describe the skew-merged permutations

$$\exists X \exists Y \, (\mathit{partition}(X, Y) \wedge \mathit{increasing}(X) \wedge \mathit{decreasing}(Y)) \, .$$

Merges

Permutation π is a merge of permutations σ and τ if the elements of π can be colored red and blue, so that the red elements form a copy of σ and the blue ones of τ.



For two permutation classes $\mathcal{C}$ and $\mathcal{D}$, let $\mathcal{C} \odot \mathcal{D}$ be the class of permutations obtained by merging a $\sigma \in \mathcal{C}$ with a $\tau \in \mathcal{D}$.

Expressive power of MSO logic

Proposition

For arbitrary MSO sentences $\varphi_1, \ldots, \varphi_k$, we can construct an MSO sentence ρ such that $\pi \models \rho$ if and only if π can be obtained as a merge of permutations $\pi_1, \ldots, \pi_k$ such that $\pi_i \models \varphi_i$ for every $i \in [k]$.

Proof.

$$\rho = \exists X_1 \exists X_2 \cdots \exists X_k \left(\mathit{partition}(X_1, \ldots, X_k) \wedge \bigwedge_{i=1}^{k} \varphi_i(X_i) \right). \qquad \square$$

Expressive power of MSO logic

Proposition

For arbitrary MSO sentences $\varphi_1, \ldots, \varphi_k$, we can construct an MSO sentence ρ such that $\pi \models \rho$ if and only if π can be obtained as a merge of permutations $\pi_1, \ldots, \pi_k$ such that $\pi_i \models \varphi_i$ for every $i \in [k]$.

Proof.

$$\rho = \exists X_1 \exists X_2 \cdots \exists X_k \left(partition(X_1, \ldots, X_k) \wedge \bigwedge_{i=1}^{k} \varphi_i(X_i) \right). \qquad \square$$

Are there (natural) classes definable in MSO but not in FO?

Proving inexpressibility in FO

Theorem

Let α be a simple permutation of size at least 4. There is no FO sentence φ such that $\sigma \models \varphi$ iff $\sigma \in Av(\alpha) \odot Av(\alpha)$.

Proving inexpressibility in FO

Theorem

Let α be a simple permutation of size at least 4. There is no FO sentence φ such that $\sigma \models \varphi$ iff $\sigma \in Av(\alpha) \odot Av(\alpha)$.

Proof outline

- Assume such a sentence φ exists and let k be its quantifier depth – the maximum number of nested quantifiers in φ.
- Find two permutations σ and τ such that
  - $\sigma \in \mathrm{Av}(\alpha) \odot \mathrm{Av}(\alpha)$ and $\tau \notin \mathrm{Av}(\alpha) \odot \mathrm{Av}(\alpha)$, and
  - σ and τ satisfy the same FO sentences of quantifier depth at most k and thus, $\sigma \models \varphi$ if and only if $\tau \models \varphi$.

Ehrenfeucht–Fraïssé games

- Two players: Spoiler and Duplicator.
- Gameboard: a pair of permutations σ and τ.
- In the i-th turn:
  - Spoiler chooses element s_i in σ or t_i in τ,
  - Duplicator chooses an element in the other permutation.

The winner of the EF-game with k rounds is

- Duplicator if $(s_1, \ldots, s_k)$ and $(t_1, \ldots, t_k)$ are isomorphic,
- Spoiler otherwise.

Ehrenfeucht–Fraïssé games

- Two players: Spoiler and Duplicator.
- Gameboard: a pair of permutations σ and τ.
- In the i-th turn:
  - Spoiler chooses element s_i in σ or t_i in τ,
  - Duplicator chooses an element in the other permutation.

The winner of the EF-game with k rounds is

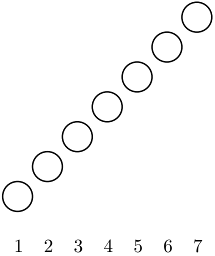
- Duplicator if $(s_1, \ldots, s_k)$ and $(t_1, \ldots, t_k)$ are isomorphic,
- Spoiler otherwise.

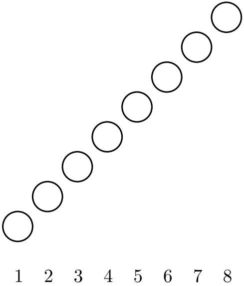
Theorem (Ehrenfeucht, Fraïssé)

Two permutations σ and τ satisfy the same FO sentences of quantifier depth at most k if and only if Duplicator has a winning strategy in the EF-game with k rounds on σ and τ.

EF-game example

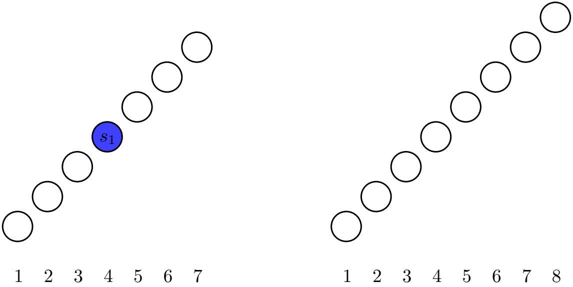
Spoiler and Duplicator play EF game with 3 rounds





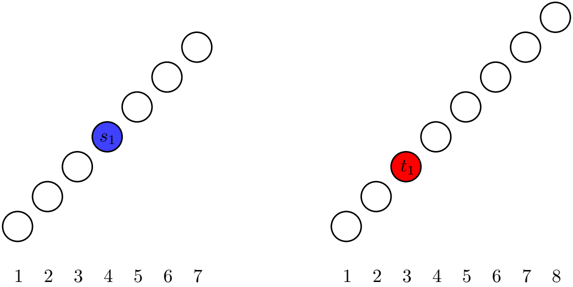
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



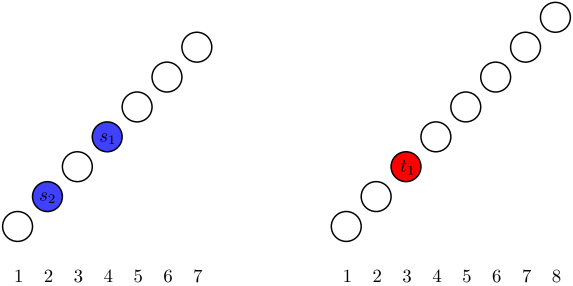
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



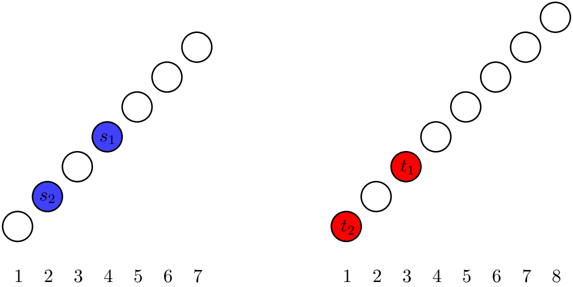
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



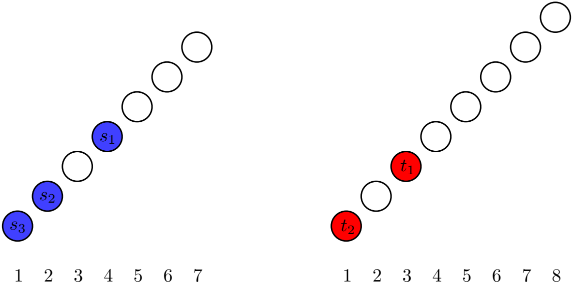
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



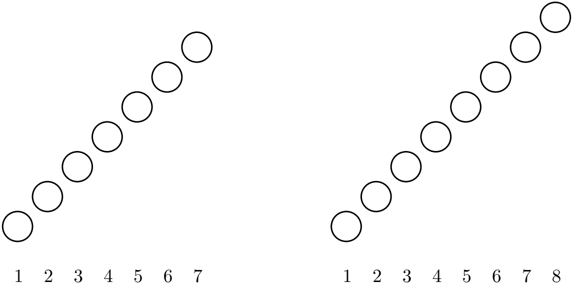
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



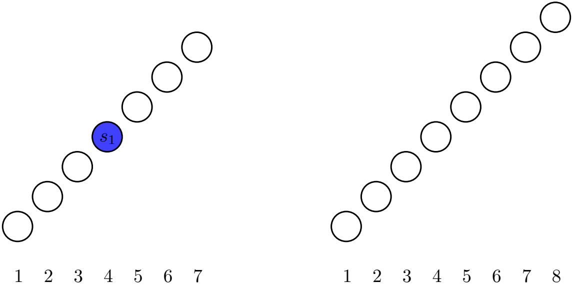
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



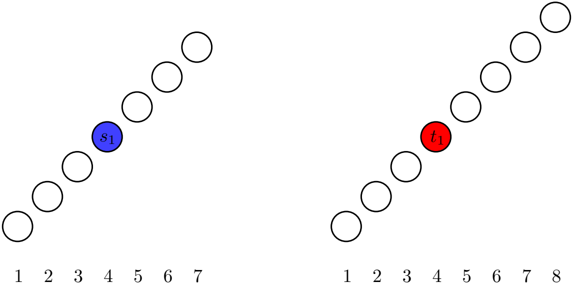
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



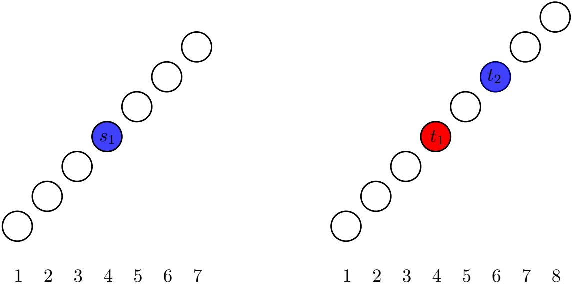
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



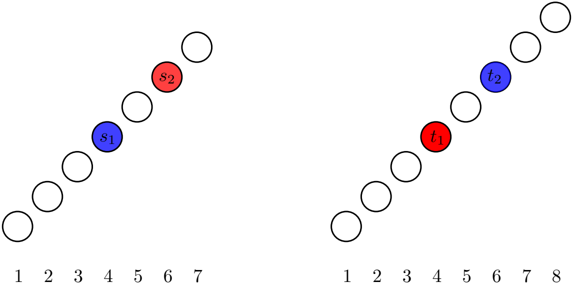
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



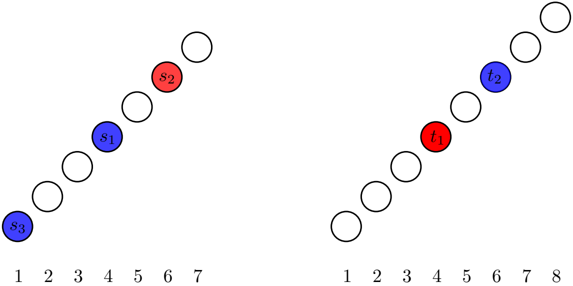
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



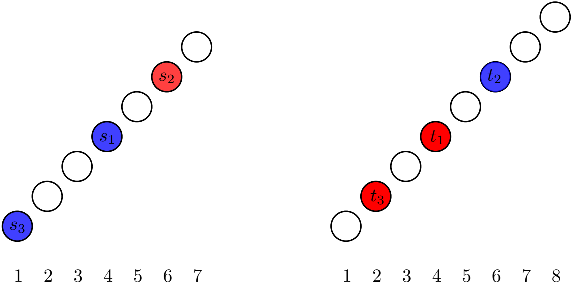
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



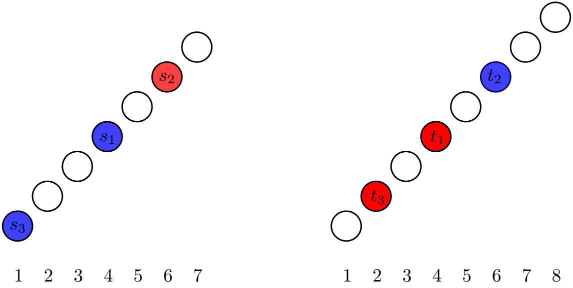
EF-game example

Spoiler and Duplicator play EF game with 3 rounds



EF-game example

Spoiler and Duplicator play EF game with 3 rounds

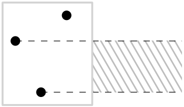


Proposition

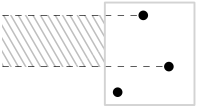
Duplicator wins the EF-game with k rounds on permutations $12\cdots\ell$ and $12\cdots m$ for every $\ell, m \geq 2^k - 1$.

Construction of the gameboard

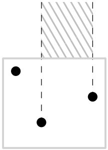
Let $\alpha = 3142$.



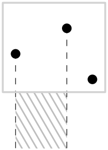
Right arrow $\alpha^{\triangleright}$



Left arrow $\alpha^{\triangleleft}$



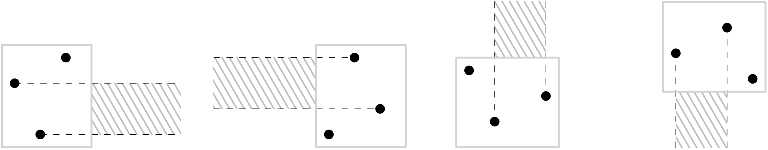
Top arrow $\alpha^{\vartriangle}$



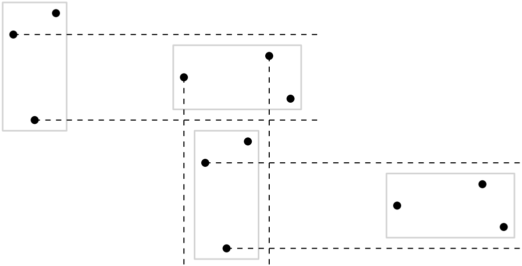
Down arrow $\alpha^{\triangledown}$

Construction of the gameboard

Let $\alpha = 3142$.

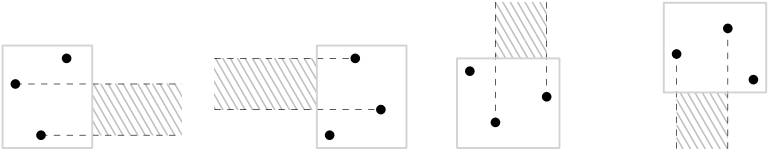


Right arrow $\alpha^{\triangleright}$ Left arrow $\alpha^{\triangleleft}$ Top arrow $\alpha^{\vartriangle}$ Down arrow $\alpha^{\triangledown}$

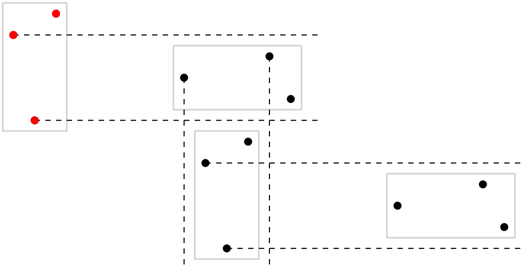


Construction of the gameboard

Let $\alpha = 3142$.

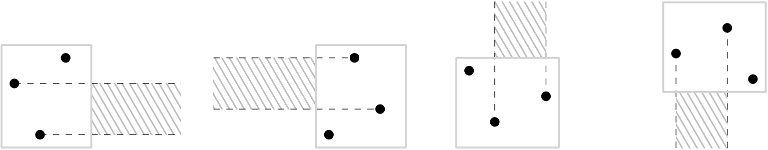


Right arrow $\alpha^{\triangleright}$ Left arrow $\alpha^{\triangleleft}$ Top arrow $\alpha^{\vartriangle}$ Down arrow $\alpha^{\triangledown}$

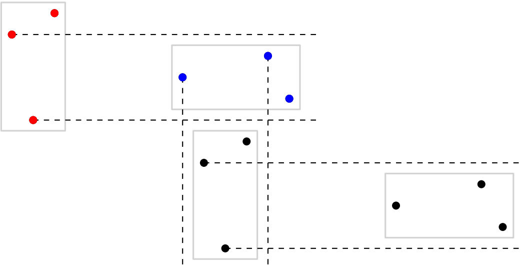


Construction of the gameboard

Let $\alpha = 3142$.

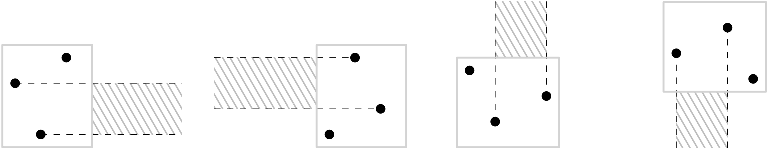


Right arrow $\alpha^{\triangleright}$ Left arrow $\alpha^{\triangleleft}$ Top arrow $\alpha^{\vartriangle}$ Down arrow $\alpha^{\triangledown}$

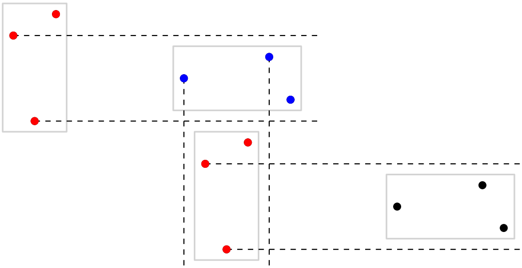


Construction of the gameboard

Let $\alpha = 3142$.

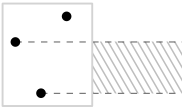


Right arrow $\alpha^{\triangleright}$ Left arrow $\alpha^{\triangleleft}$ Top arrow $\alpha^{\vartriangle}$ Down arrow $\alpha^{\triangledown}$

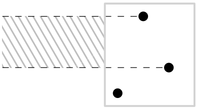


Construction of the gameboard

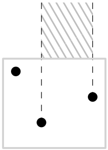
Let $\alpha = 3142$.



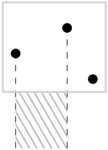
Right arrow $\alpha^{\triangleright}$



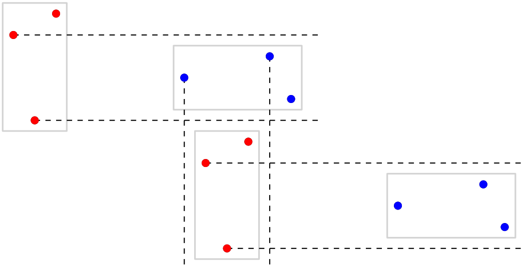
Left arrow $\alpha^{\triangleleft}$

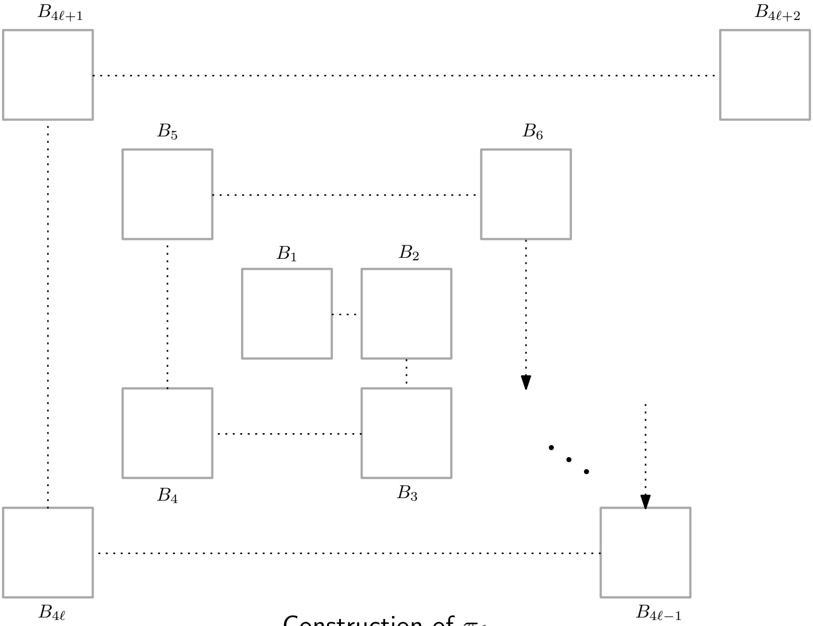


Top arrow $\alpha^{\vartriangle}$

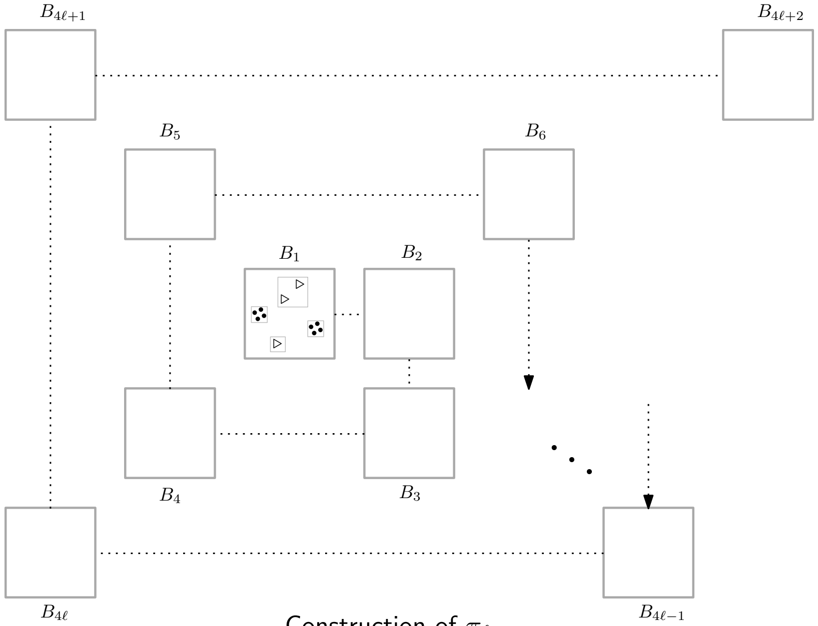


Down arrow $\alpha^{\triangledown}$

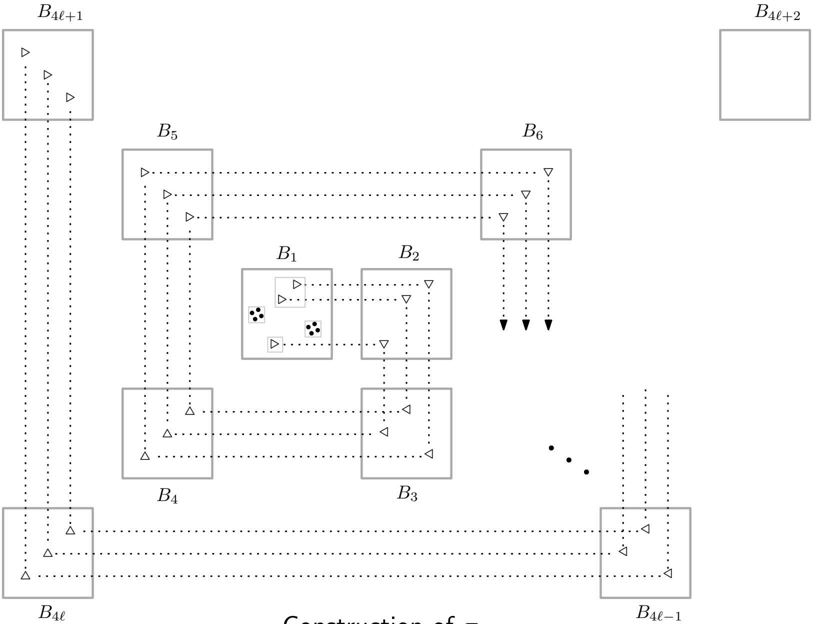




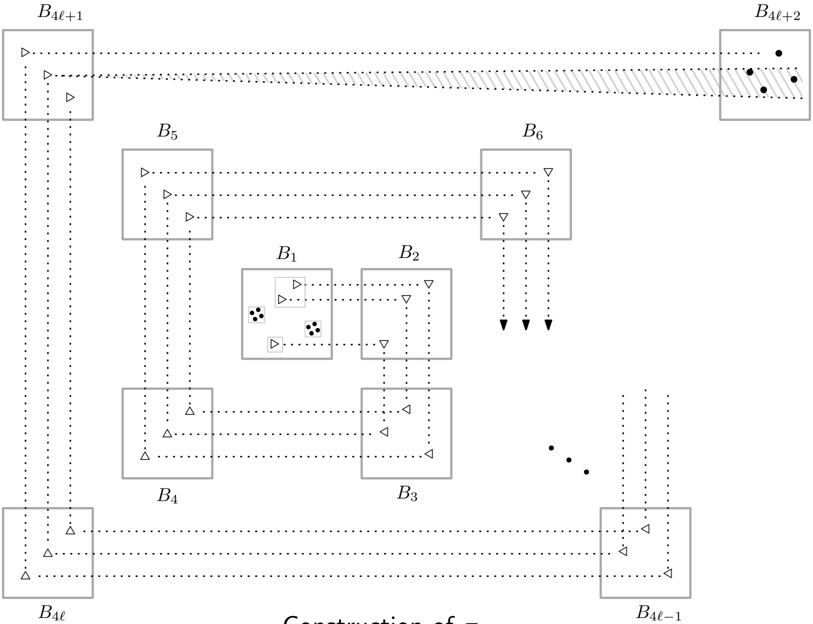
Construction of π_ℓ.



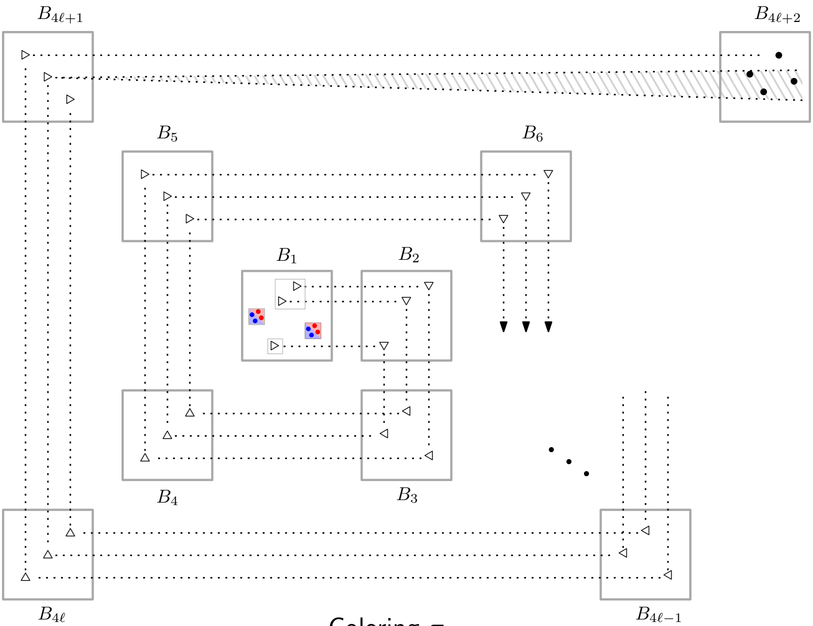
Construction of π_ℓ.



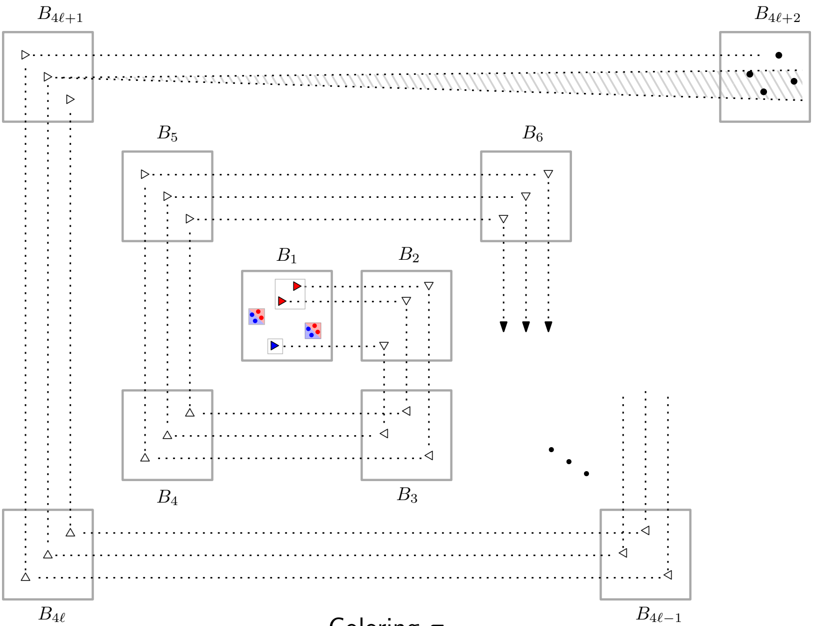
Construction of π_ℓ.



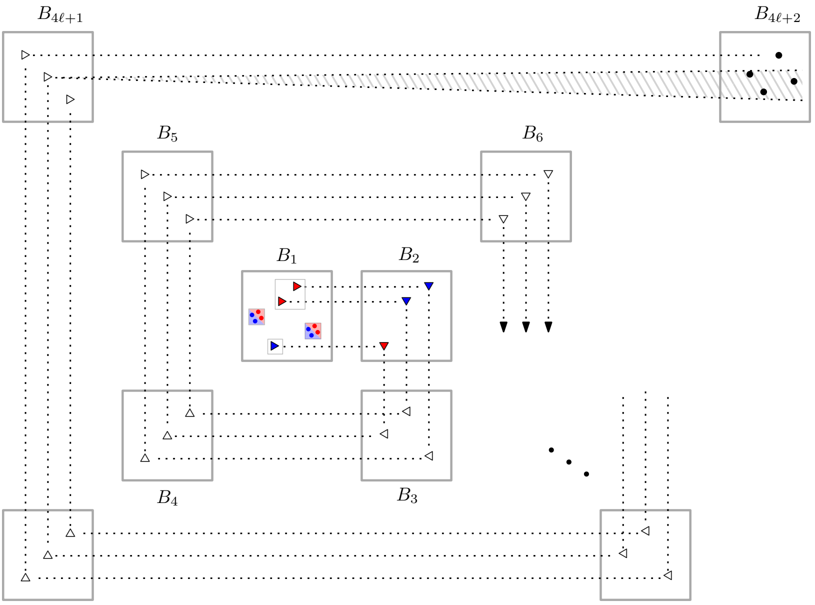
Construction of π_ℓ.



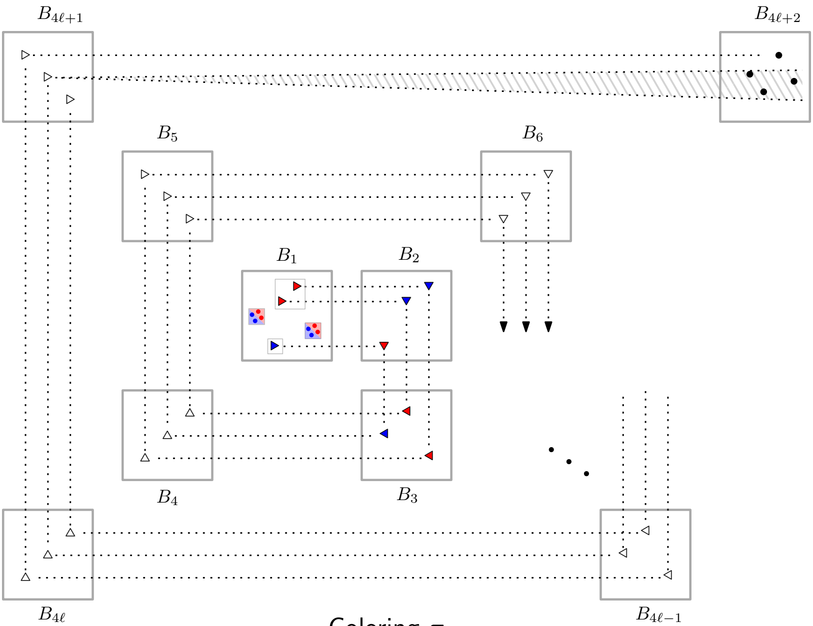
Coloring π_ℓ.



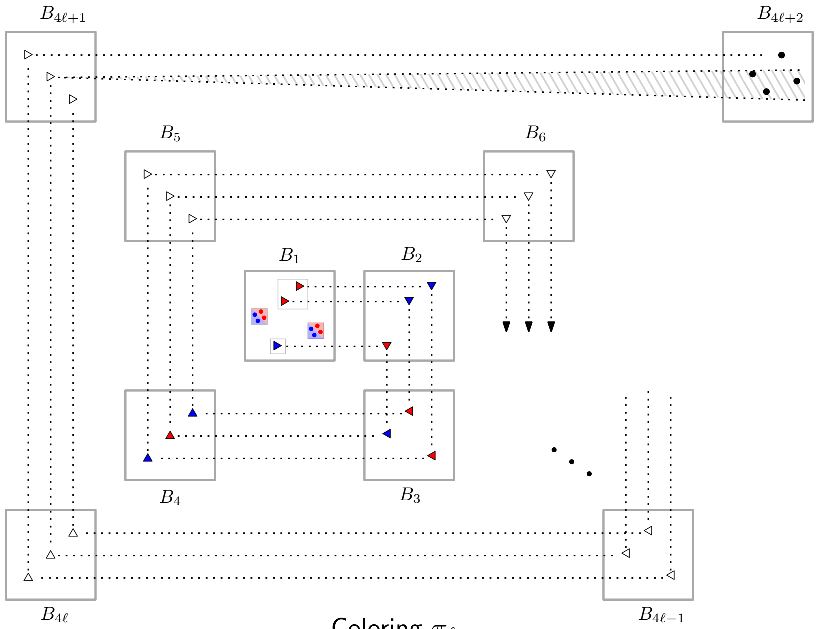
Coloring π_ℓ.



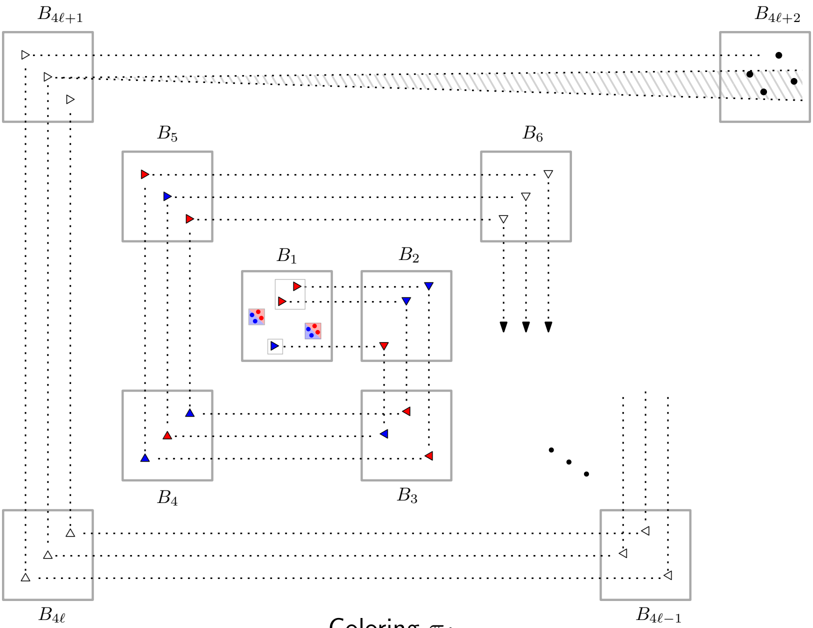
Coloring π_ℓ.



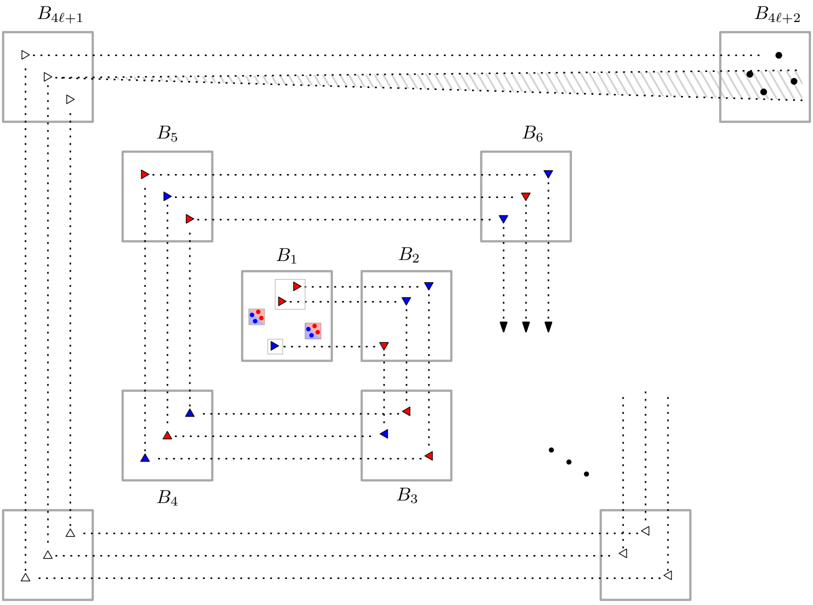
Coloring π_ℓ.



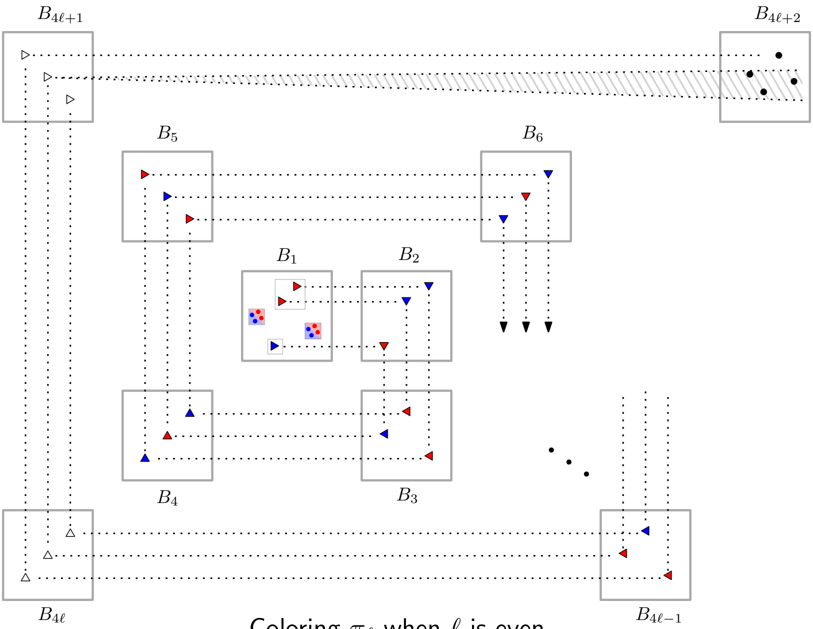
Coloring π_ℓ.



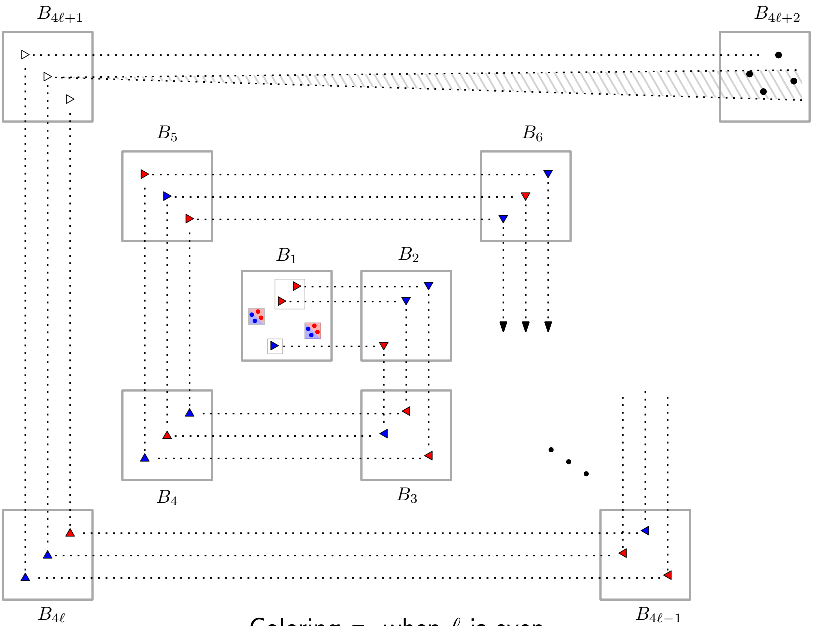
Coloring π_ℓ.



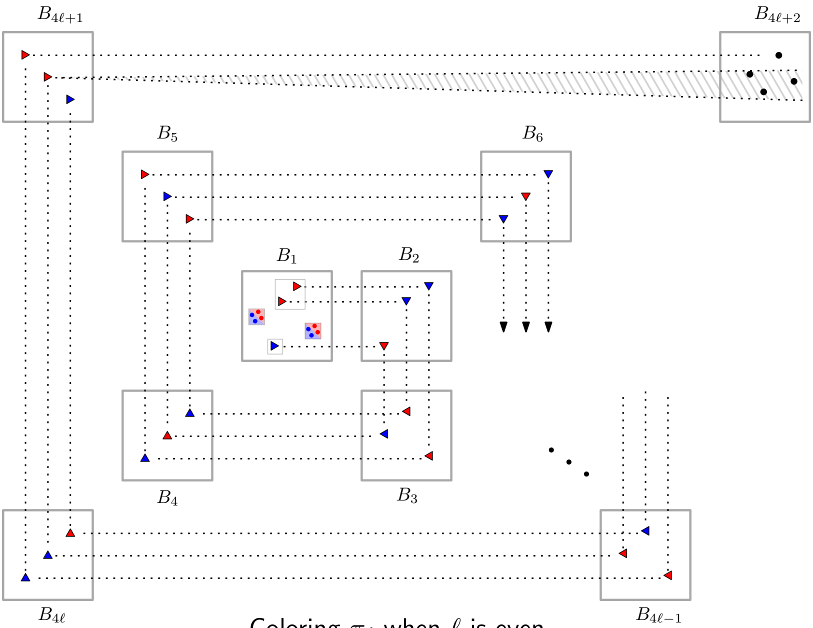
Coloring π_ℓ.



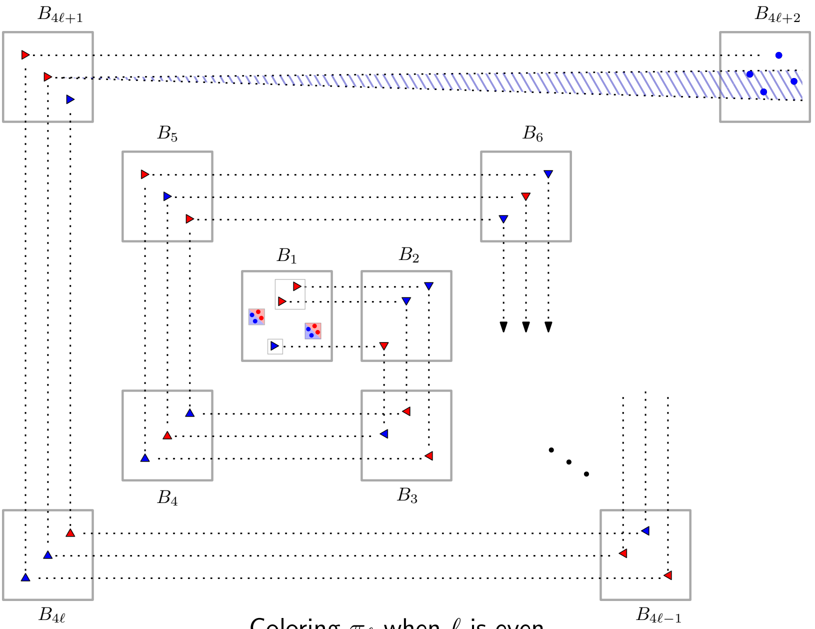
Coloring π_ℓ when ℓ is even.



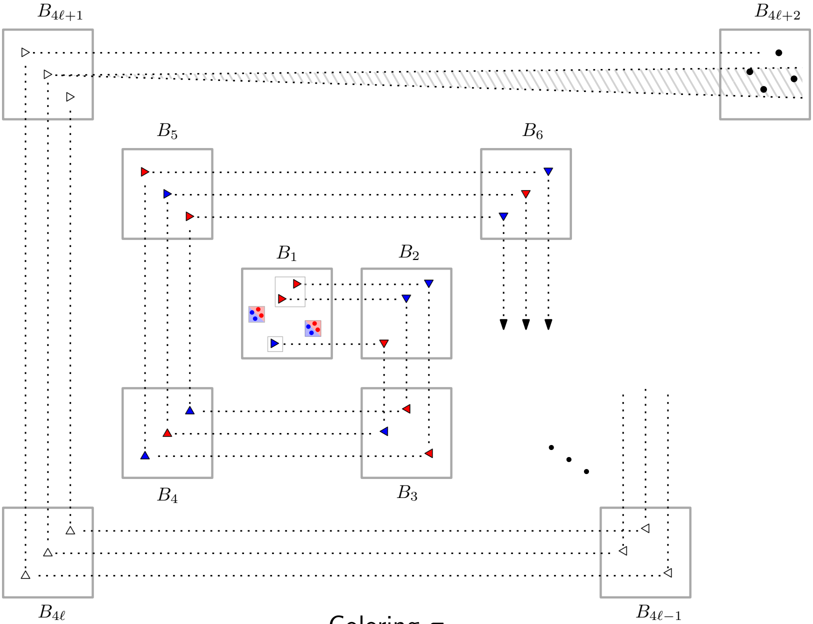
Coloring π_ℓ when ℓ is even.



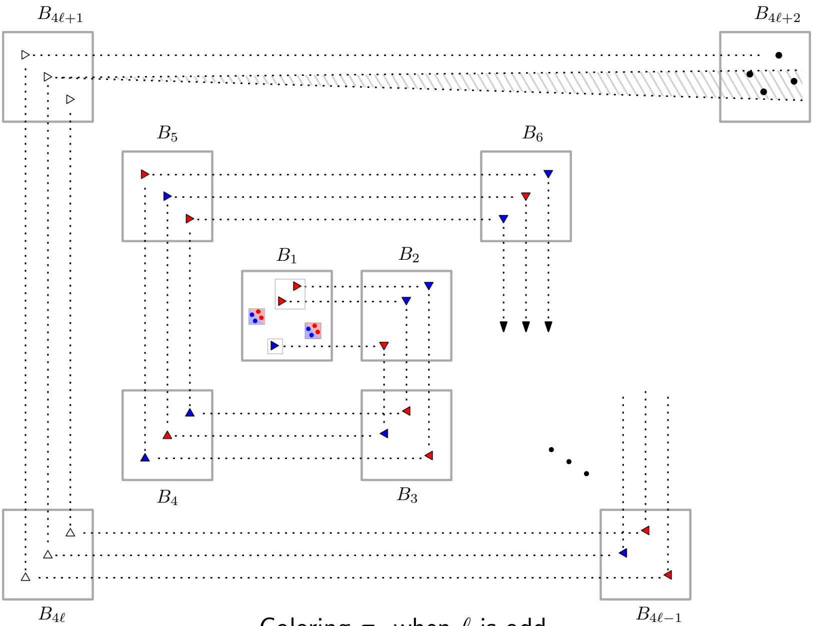
Coloring π_ℓ when ℓ is even.



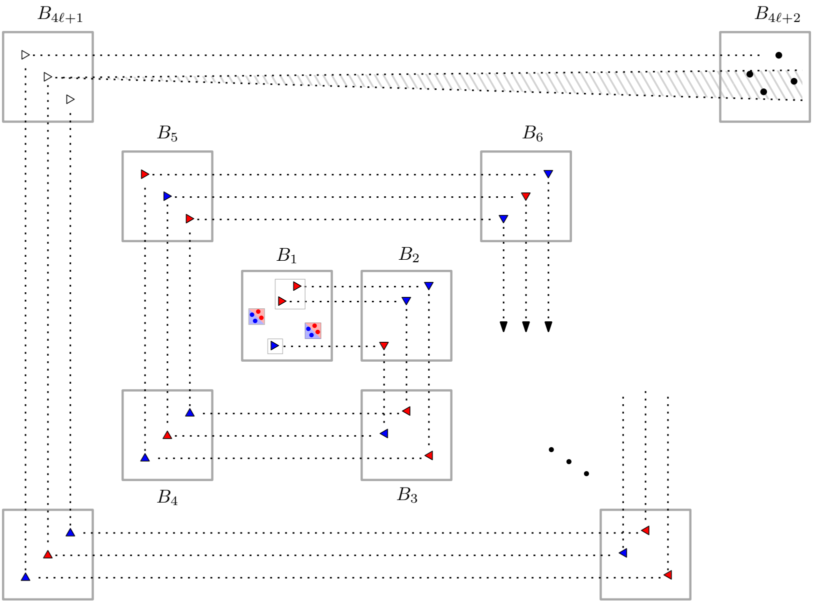
Coloring π_ℓ when ℓ is even.



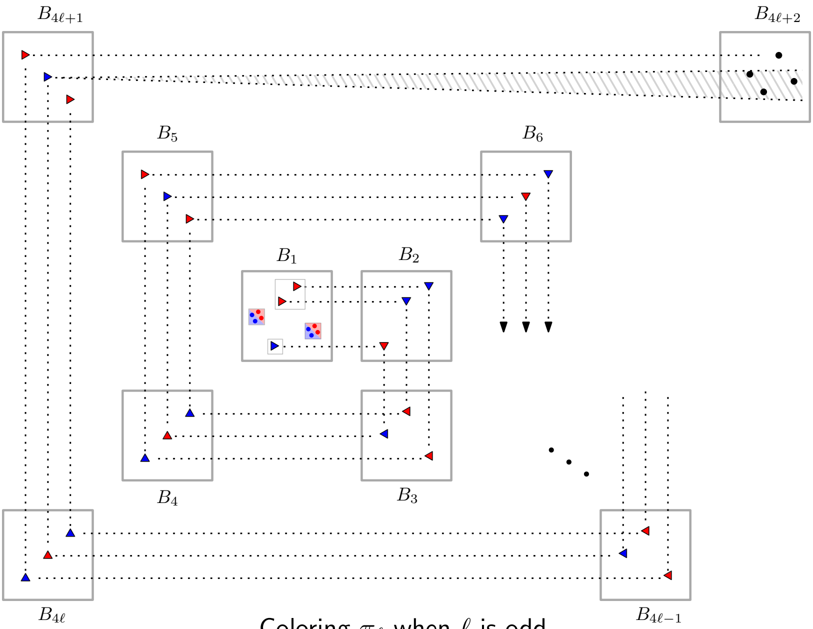
Coloring π_ℓ.



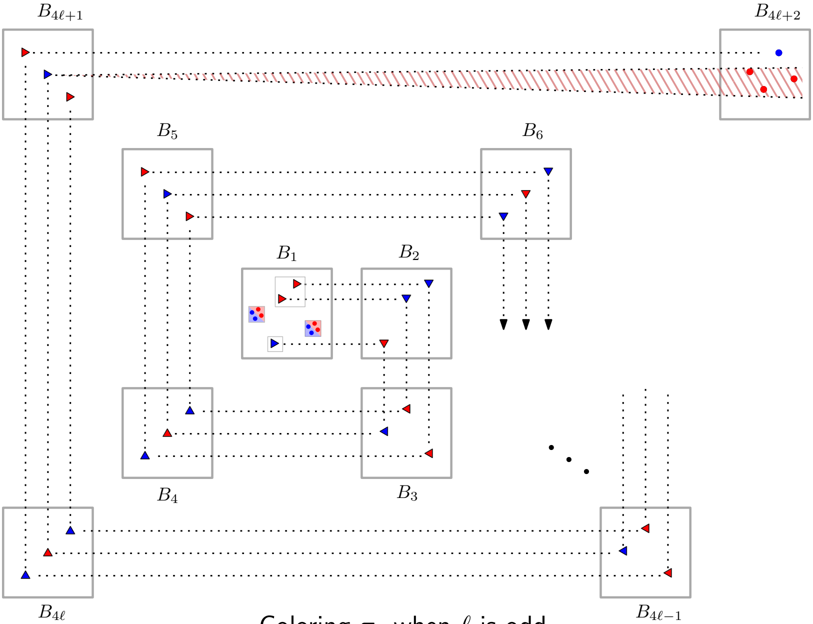
Coloring π_ℓ when ℓ is odd.



Coloring π_ℓ when ℓ is odd.



Coloring π_ℓ when ℓ is odd.



Coloring π_ℓ when ℓ is odd.

Claim

The permutation π_ℓ belongs to $\mathsf{Av}(\alpha) \odot \mathsf{Av}(\alpha)$ if and only if ℓ is odd.

Claim

The permutation π_ℓ belongs to $\mathsf{Av}(\alpha) \odot \mathsf{Av}(\alpha)$ if and only if ℓ is odd.

Claim

For every $n, m \geq 2^{k+1} - 2$, Duplicator wins the EF-game with k rounds on π_n and π_m and thus, π_n and π_m satisfy the same set of FO sentences of quantifier depth at most k.

Claim

The permutation π_ℓ belongs to $\mathsf{Av}(\alpha) \odot \mathsf{Av}(\alpha)$ if and only if ℓ is odd.

Claim

For every $n, m \geq 2^{k+1} - 2$, Duplicator wins the EF-game with k rounds on π_n and π_m and thus, π_n and π_m satisfy the same set of FO sentences of quantifier depth at most k.

Recap of the proof

- Assume there is an FO sentence φ defining $\mathsf{Av}(\alpha) \odot \mathsf{Av}(\alpha)$ and let k be its quantifier depth.
- Pick $n, m \geq 2^{k+1} - 2$ such that n is odd and m is even.
- We have that π_n belongs to $\mathsf{Av}(\alpha) \odot \mathsf{Av}(\alpha)$ while π_m does not, but also $\pi_n \models \varphi$ iff $\pi_m \models \varphi$. $\square$

Expressive power of MSO logic

- We can express any merge of several classes that are themselves MSO-definable.
- Any class $\mathsf{Av}(\alpha) \odot \mathsf{Av}(\alpha)$ where α is a simple permutation of size at least 4 cannot be expressed by an FO sentence.
- The property of having a fixed point cannot be expressed even in MSO.

Model checking

First-Order (Monadic Second-Order) Model Checking

Input: A permutation π of size n and an FO (MSO) sentence φ.

Question: Does $\pi \models \varphi$?

Model checking

First-Order (Monadic Second-Order) Model Checking

Input: A permutation π of size n and an FO (MSO) sentence φ.
Question: Does $\pi \models \varphi$?

Theorem (Bonnet et al. 2021)

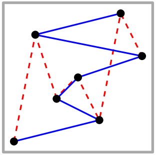
There is an algorithm that decides FO Model Checking *in time* $f(|\varphi|, \mathrm{tww}(\pi)) \cdot n$ *where* f *is some computable function* f *and* $\mathrm{tww}(\pi)$ *is the twin-width of* π*.*

Corollary

Let φ *be an FO sentence that defines a proper permutation class* $\mathcal{C}$*. Then there is an algorithm that decides in* $\mathcal{O}(n)$ *time whether a permutation* π *of size* n *belongs to* $\mathcal{C}$*, i.e., if* $\pi \models \varphi$*.*

MSO model checking

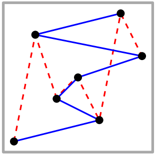
An incidence graph G_π of a permutation π is the graph whose vertices are the elements of π with π_i and π_j connected by an edge if $|i - j| = 1$ or $|\pi_i - \pi_j| = 1$.



The tree-width of a permutation π is defined as the tree-width of G_π.

MSO model checking

An incidence graph G_π of a permutation π is the graph whose vertices are the elements of π with π_i and π_j connected by an edge if $|i-j|=1$ or $|\pi_i-\pi_j|=1$.

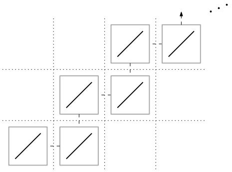


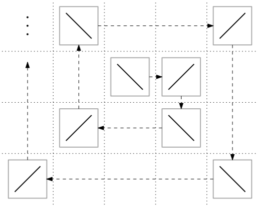
The tree-width of a permutation π is defined as the tree-width of G_π.

Theorem

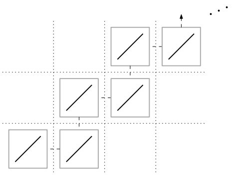
There is an algorithm that decides MSO Model Checking *in time* $f(|\varphi|, \mathrm{tw}(\pi)) \cdot n$ *where* f *is some computable function* f *and* $\mathrm{tw}(\pi)$ *is the tree-width of* π*.*

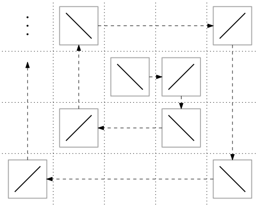
A permutation class $\mathcal{C}$ has the long path property (LPP) if for every k the class $\mathcal{C}$ contains a monotone grid subclass whose cell graph is a path of length k.





A permutation class $\mathcal{C}$ has the long path property (LPP) if for every k the class $\mathcal{C}$ contains a monotone grid subclass whose cell graph is a path of length k.





Theorem (informally)

Let $\mathcal{C}$ be a permutation class with the poly-time computable long path property. Then MSO model checking inside $\mathcal{C}$ is already as hard as MSO model checking on general undirected graphs.

Summary

- Merge classes are natural example of MSO-definable classes.
- Moreover, many of them are not FO-definable.
- Tree-width seems to be the right parameter for MSO model checking in permutations.

Summary

- Merge classes are natural example of MSO-definable classes.
- Moreover, many of them are not FO-definable.
- Tree-width seems to be the right parameter for MSO model checking in permutations.

Question

What other natural classes are definable in MSO but not in FO?

Summary

- Merge classes are natural example of MSO-definable classes.
- Moreover, many of them are not FO-definable.
- Tree-width seems to be the right parameter for MSO model checking in permutations.

Question

What other natural classes are definable in MSO but not in FO?

Conjecture

A permutation class $\mathcal{C}$ has unbounded tree-width if and only if it has the long path property.

Summary

- Merge classes are natural example of MSO-definable classes.
- Moreover, many of them are not FO-definable.
- Tree-width seems to be the right parameter for MSO model checking in permutations.

Question

What other natural classes are definable in MSO but not in FO?

Conjecture

A permutation class $\mathcal{C}$ has unbounded tree-width if and only if it has the long path property.

Thank you!