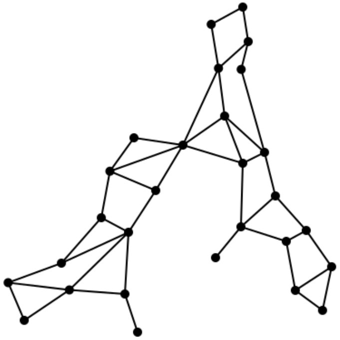
Transducing Linear Decompositions of Tournaments

Colin Geniet   Fatemeh Ghasemi   Mamadou Moustapha Kanté

DIMAG, IBS

ICALP 2026

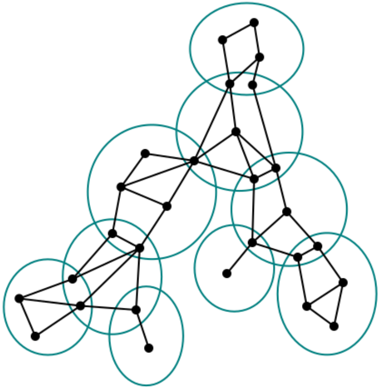
Tree-width



Small tree-width graph G

"Looks like a tree"

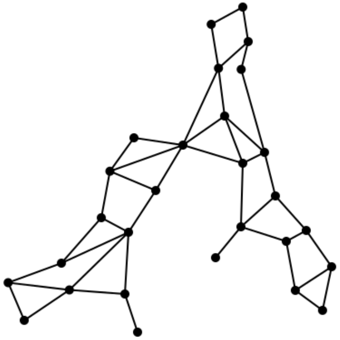
Tree-width



Small tree-width graph G

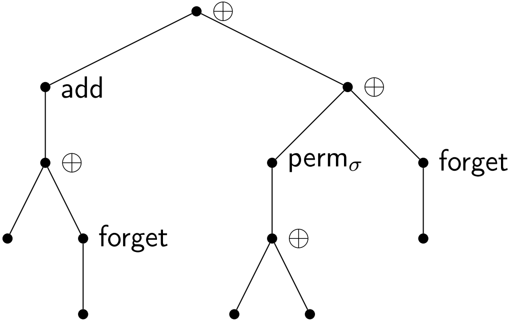
"Looks like a tree"

Tree-width



Small tree-width graph G

"Looks like a tree"



Tree decomposition

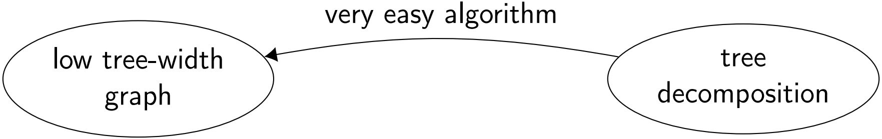
Labelled tree encoding G

Graphs and tree-decompositions

How to go back and forth?

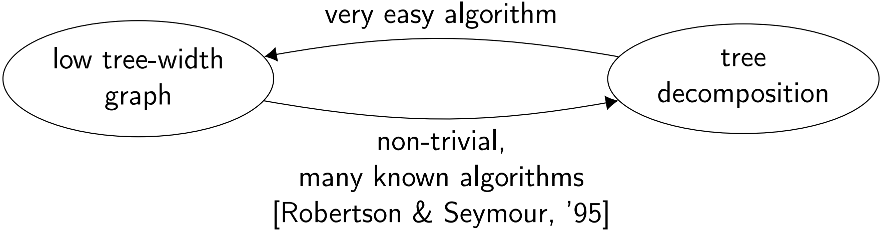
Graphs and tree-decompositions

How to go back and forth? Algorithmically:



Graphs and tree-decompositions

How to go back and forth? Algorithmically:



MSO Interpretations

MSO logic: (in the case of graphs)

- $\forall x, \exists y, \forall X, \exists Y$: quantifiers on elements (vertices) and *sets*
- $\neg, \vee, \wedge$: boolean operators
- $E(x, y)$: edge predicate
- $x = y, x \in X$

MSO Interpretations

MSO logic: (in the case of graphs)

- $\forall x, \exists y, \forall X, \exists Y$: quantifiers on elements (vertices) and *sets*
- $\neg, \vee, \wedge$: boolean operators
- $E(x, y)$: edge predicate
- $x = y, x \in X$

MSO interpretation $\Phi = (\phi_V; \phi_1, \ldots, \phi_k)$

Given input G, define $\Phi(G)$:

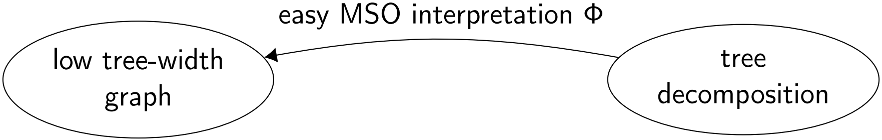
- universe (vertices)

$$\{x \in V(G) : G \models \phi_V(x)\}$$

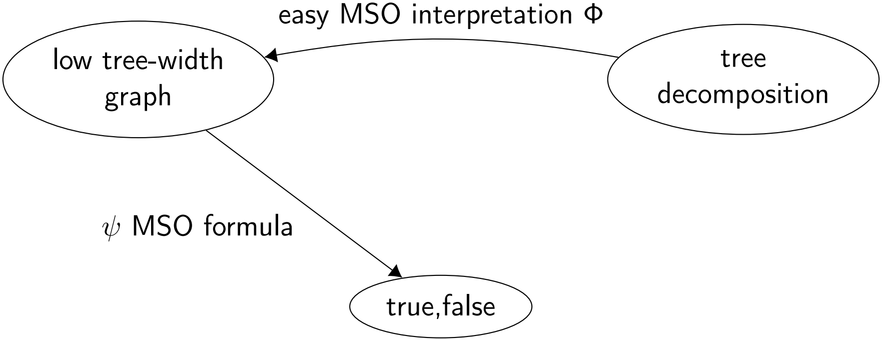
- for $i = 1, \ldots, k$, add the relation

$$\{(v_1, \ldots, v_r) : G \models \phi_i(v_1, \ldots, v_r)\}$$

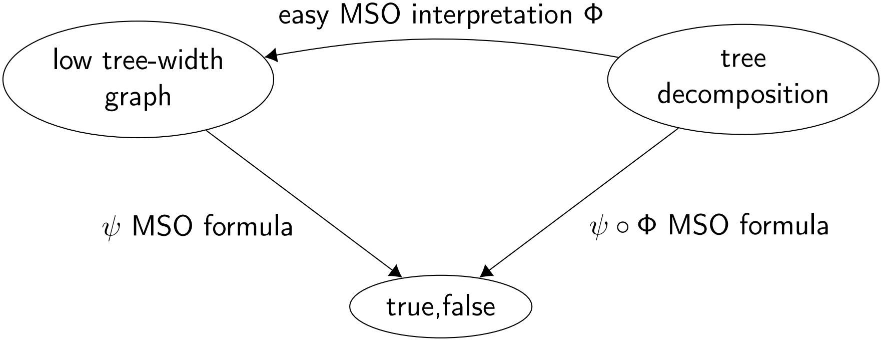
Courcelle's Theorem



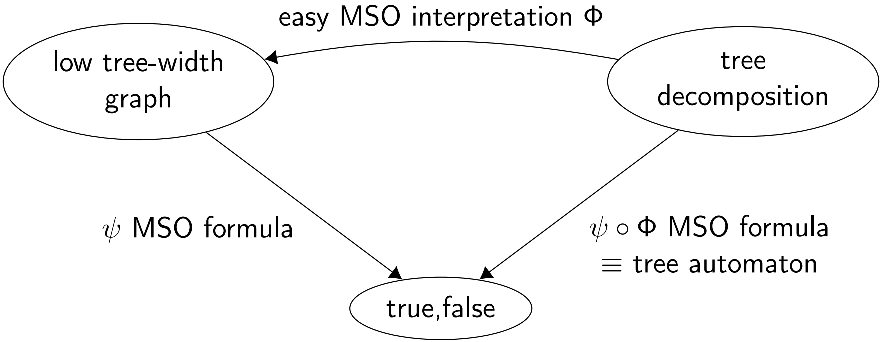
Courcelle's Theorem



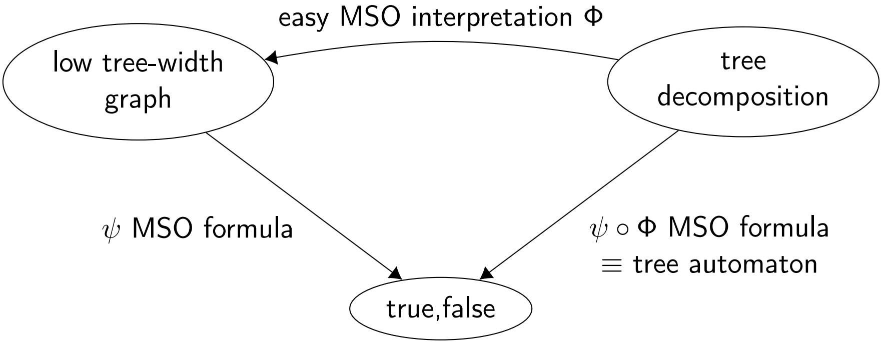
Courcelle's Theorem



Courcelle's Theorem



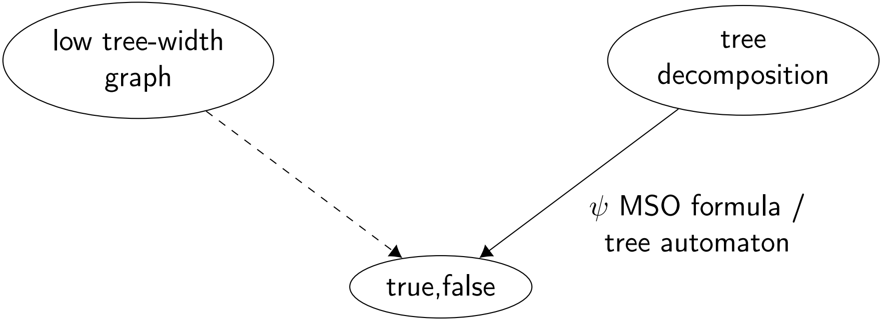
Courcelle's Theorem



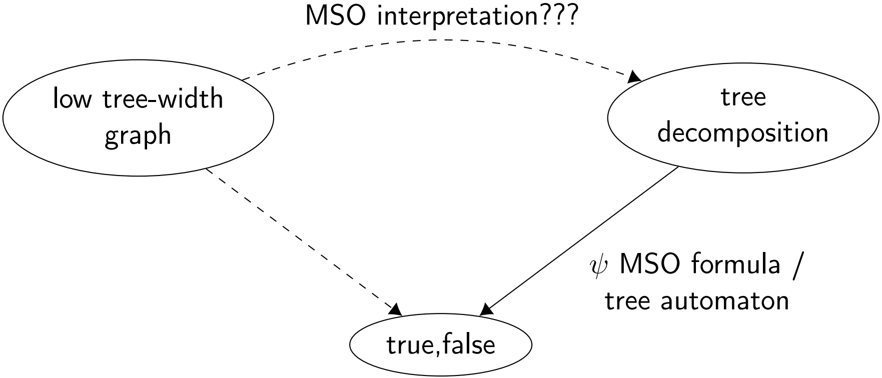
Theorem (Courcelle's theorem)

Any MSO formula on graphs is equivalent to a tree automaton on tree decompositions (for fixed tree-width).

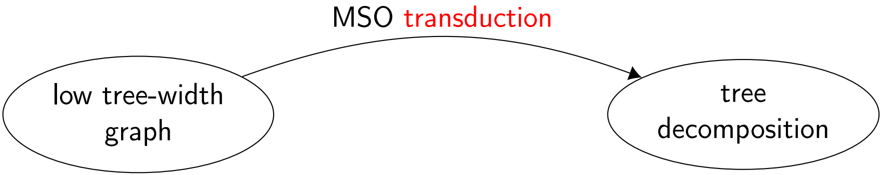
In the other direction



In the other direction



Definable Tree Decompositions



Theorem (Bojańczyk & Pilipczuk '16)

For any k, there is an MSO interpretation Φ such that for any graph G of tree-width k, there is a way to add vertex colours to G so that $\Phi(G)$ is a tree decomposition of G of width $f(k)$.

MSO transduction

MSO transduction Φ: given G

- Consider all possible vertex colourings $\lambda : V(G) \to [k]$.
  This is non-deterministic! $\Phi(G)$ *is a set of structures.*
- Apply an MSO interpretation (where formulas can talk about the colours λ)

And some other operations which are not that important here.

MSO transduction

MSO transduction Φ: given G

- Consider all possible vertex colourings $\lambda : V(G) \to [k]$.
  This is non-deterministic! $\Phi(G)$ *is a set of structures.*
- Apply an MSO interpretation (where formulas can talk about the colours λ)

And some other operations which are not that important here.

Theorem (Bojańczyk & Pilipczuk '16)

For any k, there is an MSO transduction Φ such that for any graph G of tree-width k, given G, Φ produces a tree decomposition of G of width $f(k)$.

Proof Structure

Theorem (Bojańczyk & Pilipczuk '16)

For any k, there is an MSO transduction Φ such that for any graph G of tree-width k, given G, Φ produces a tree decomposition of G of width $f(k)$.

The proof has two parts:

1. Transduce decompositions on graphs of small path-width.
   Uses Simon's Factorisation to reduce to 'very regular' cases.

2. 'Clever' reduction from tree-width to path-width using reasoning on paths.

Proof Structure

Theorem (Bojańczyk & Pilipczuk '16)

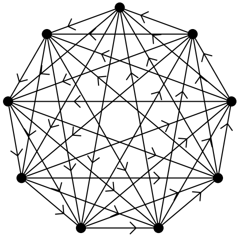
For any k, there is an MSO transduction Φ such that for any graph G of tree-width k, given G, Φ produces a tree decomposition of G of width $f(k)$.

The proof has two parts:

1. Transduce decompositions on graphs of small path-width.
   Uses Simon's Factorisation to reduce to 'very regular' cases.
   Generalisations:
   - linear clique-width for dense graphs [Bojańczyk, Grohe, Pilipczuk '18],
   - linear rank-width for some matroids [Campbell, Guillon, Kanté, E.J. Kim, Oum '25]
2. 'Clever' reduction from tree-width to path-width using reasoning on paths.
   No known generalisation.

Tournaments

Tournament: directed graph where for all vertices $u \neq v$, exactly one of $u \to v$ or $u \leftarrow v$ is an edge.



Linear Clique-width

Object: coloured tournament $T = (V, E, \lambda)$, $\lambda : V \to [k]$.
Operations:

Add For some specified $c \in [k]$, $N \subseteq [k]$,

- create a new vertex v
- define $\lambda(v) := c$
- for any existing vertex u, add $v \to u$ if $\lambda(u) \in N$, and $v \leftarrow u$ otherwise.

Recolour Set $\lambda := \lambda \circ f$ for some $f : [k] \to [k]$.

$\simeq k^k$ operations in total.

Linear Clique-width

Object: coloured tournament $T = (V, E, \lambda)$, $\lambda : V \to [k]$.
Operations:

Add For some specified $c \in [k]$, $N \subseteq [k]$,

- create a new vertex v
- define $\lambda(v) := c$
- for any existing vertex u, add $v \to u$ if $\lambda(u) \in N$, and $v \leftarrow u$ otherwise.

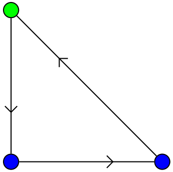
Recolour Set $\lambda := \lambda \circ f$ for some $f : [k] \to [k]$.

$\simeq k^k$ operations in total.

T has *linear clique-width* $\leq k$ if it can be created by these operations.
The sequence of operations constructing T is a *linear decomposition*.

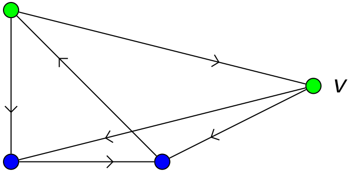
For k fixed, a linear decomposition is a word over some fixed alphabet.

Linear Clique-width – Example

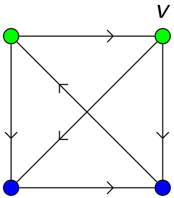


Linear Clique-width – Example

Add a vertex v with colour green
and edges $v \to$ blue and $v \leftarrow$ green.

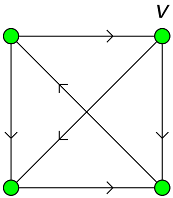


Linear Clique-width – Example



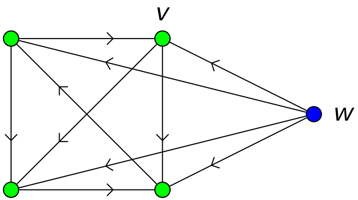
Linear Clique-width – Example

Recolour blue to green



Linear Clique-width – Example

Add a vertex w with colour blue
and edges $w \to$ green.



Linear Clique-width

Object: coloured tournament $T = (V, E, \lambda)$, $\lambda : V \to [k]$.
Operations:

Add For some specified $c \in [k]$, $N \subseteq [k]$,

- create a new vertex v
- define $\lambda(v) := c$
- for any existing vertex u, add $v \to u$ if $\lambda(u) \in N$, and $v \leftarrow u$ otherwise.

Recolour Set $\lambda := \lambda \circ f$ for some $f : [k] \to [k]$.

$\simeq k^k$ operations in total.

T has *linear clique-width* $\leq k$ if it can be created by these operations.
The sequence of operations constructing T is a *linear decomposition*.

For k fixed, a linear decomposition is a word over some fixed alphabet.

FO definable decompositions for tournaments

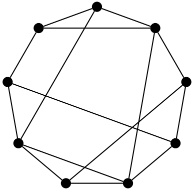
Theorem (G., Ghasemi, Kanté)

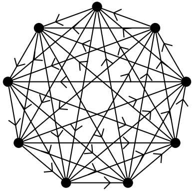
For any k, there is an FO transduction Φ such that for any tournament T of linear clique-width k, given G, Φ produces a linear decomposition of G of width $f(k)$.

- First-order (FO) logic is enough (no set quantifiers)
- The output is always a *linear* decomposition (= encoding of T as a word).
  This is specific to tournaments, and impossible for bounded path-width graphs!
- But it only applies to tournaments with small linear clique-width (= encoded by words), not clique-width in general (= encoded by trees).

Why Tournaments

Graph



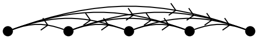


Tournament

Ramsey Theorem

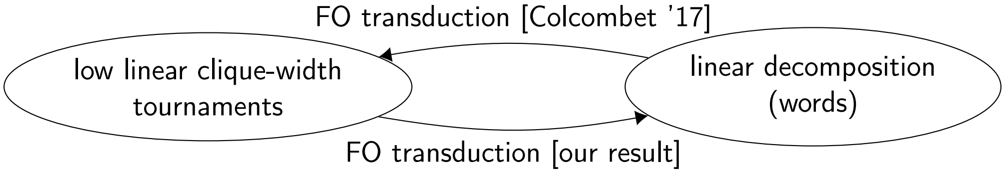
Ramsey Theorem

clique/stable set
= no structure



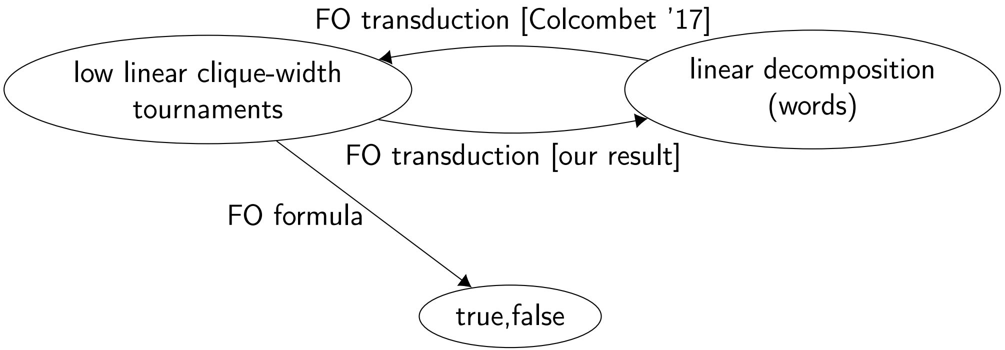
transitive tournament
= total ordering

Existential MSO



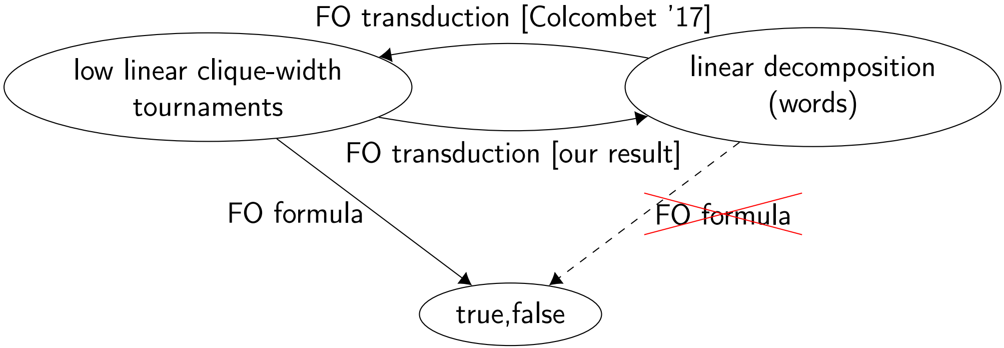
Linear decomposition / words encoded as ordered structures (this matters for FO logic)

Existential MSO



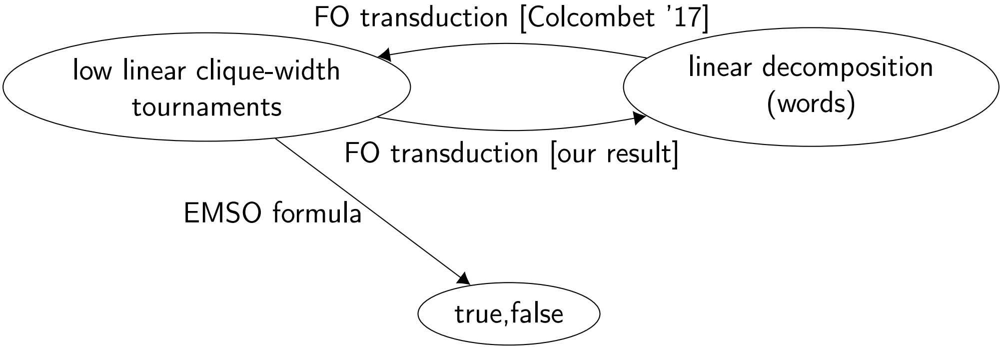
FO transductions and FO formulas cannot be composed.

Existential MSO



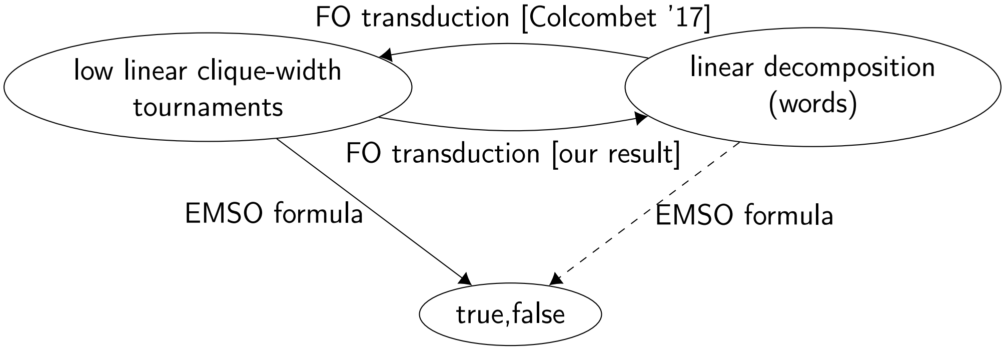
FO transductions and FO formulas cannot be composed.

Existential MSO



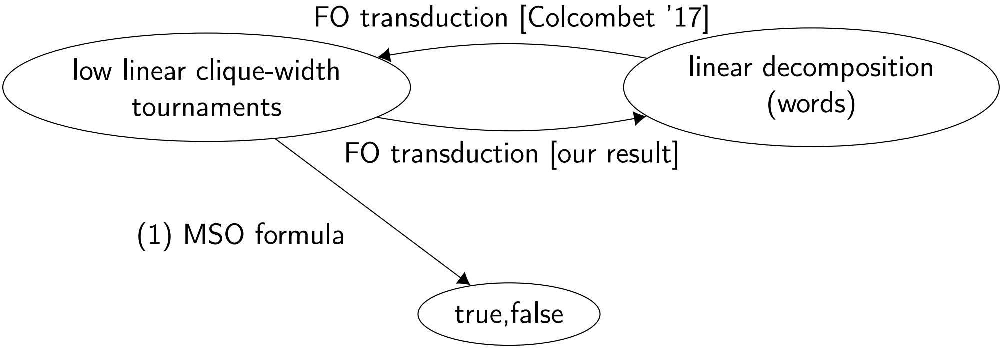
FO transductions and EMSO formulas can be composed!
EMSO: $\exists X_1, \ldots, \exists X_k\ \psi$ (with ψ FO formula)

Existential MSO

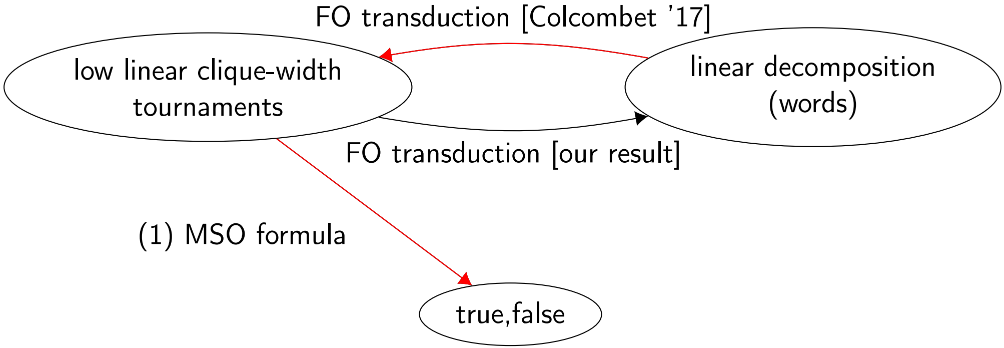


FO transductions and EMSO formulas can be composed!
EMSO: $\exists X_1, \ldots, \exists X_k\ \psi$ (with ψ FO formula)

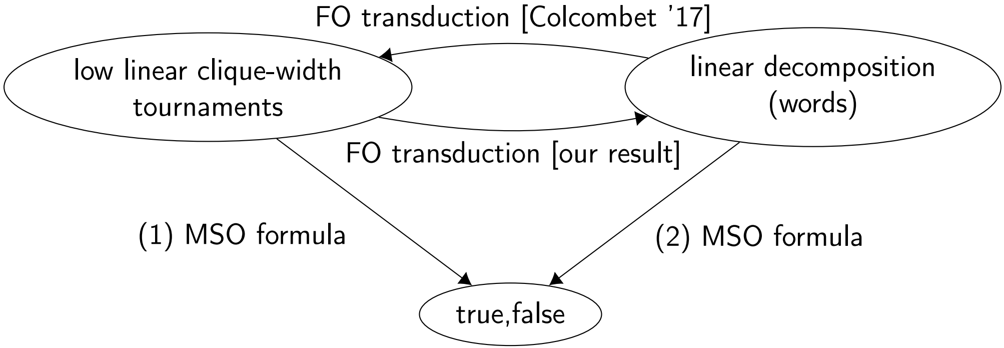
Existential MSO



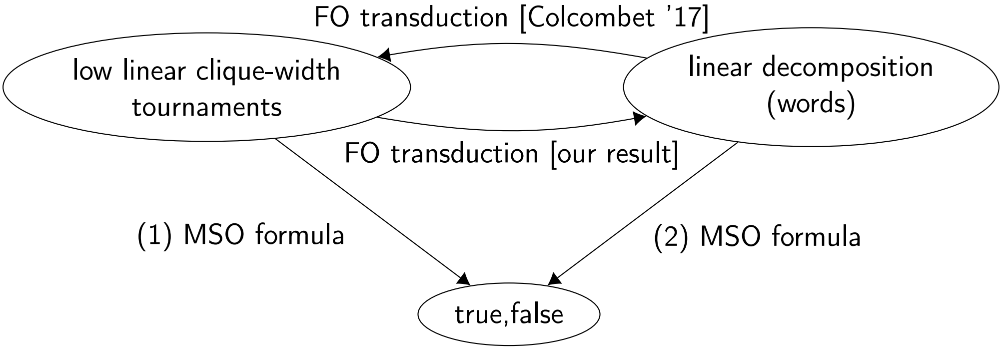
Existential MSO



Existential MSO

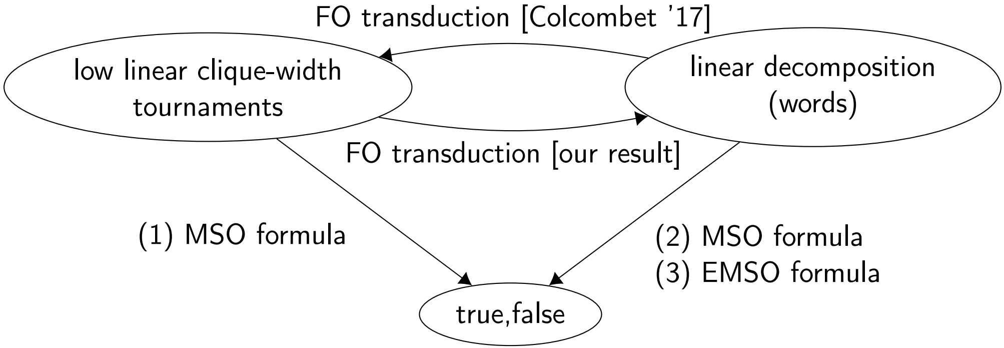


Existential MSO

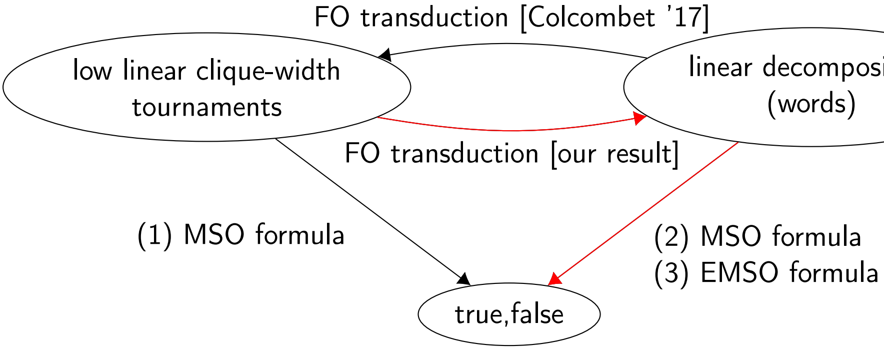


Büchi–Elgot–Trakhtenbrot: MSO and EMSO are equivalent on words

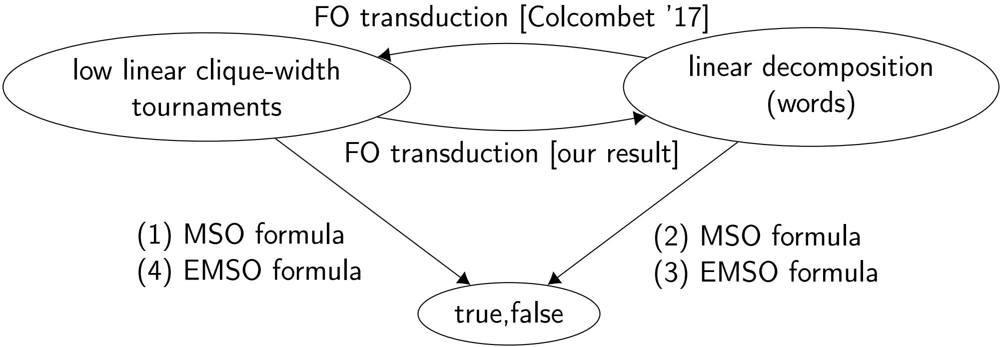
Existential MSO



Existential MSO



Existential MSO



Corollary (G., Ghasemi, Kanté)

For bounded linear clique-width tournaments, MSO and EMSO have the same expressive power.

Proof idea

Theorem (G., Ghasemi, Kanté)

For any k, there is an FO transduction Φ such that for any tournament T of linear clique-width k, given G, Φ produces a linear decomposition of G of width $f(k)$.

Proof idea

Theorem (G., Ghasemi, Kanté)

For any k, there is an FO transduction Φ such that for any tournament T of linear clique-width k, given G, Φ produces a linear decomposition of G of width $f(k)$.

- There is some (hidden) linear decomposition D of G, which we want to describe.

Proof idea

Theorem (G., Ghasemi, Kanté)

For any k, there is an FO transduction Φ such that for any tournament T of linear clique-width k, given G, Φ produces a linear decomposition of G of width $f(k)$.

- There is some (hidden) linear decomposition D of G, which we want to describe.
- Actual goal: describing the order in which vertices of T are added.
  (Adding the rest of D afterwards is easy.)

Proof idea

Theorem (G., Ghasemi, Kanté)

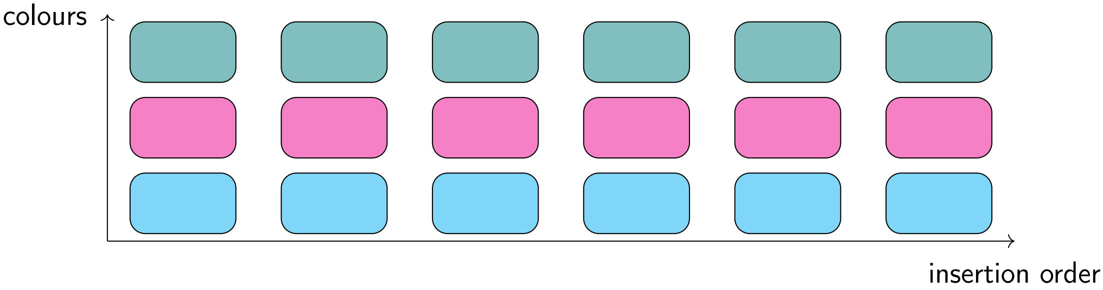
For any k, there is an FO transduction Φ such that for any tournament T of linear clique-width k, given G, Φ produces a linear decomposition of G of width $f(k)$.

- There is some (hidden) linear decomposition D of G, which we want to describe.
- Actual goal: describing the order in which vertices of T are added.
  (Adding the rest of D afterwards is easy.)
- Big hammer: Simon's Factorisation allows to pretend that D is 'very regular'.

Simon's Factorisation Magic (grossly oversimplified)

The linear decomposition is split into intervals:

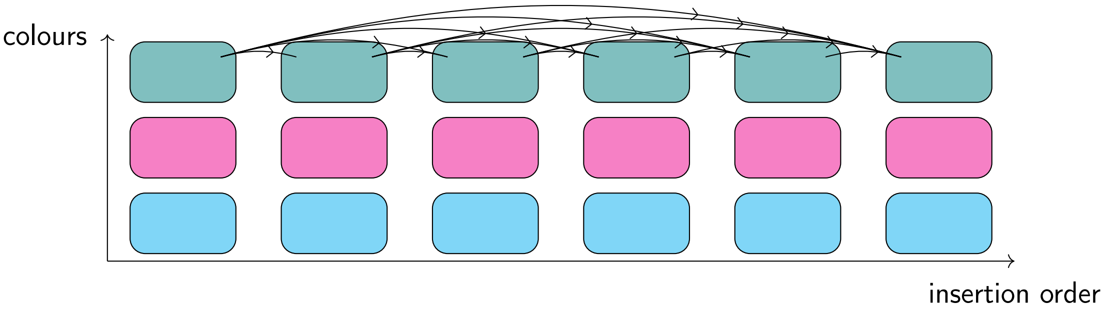
- each interval is 'strictly simpler' (bounded induction)
- vertex colours are consistent at the interval boundaries
- edges between intervals only depend on (1) the relative ordering and (2) the colour.



Simon's Factorisation Magic (grossly oversimplified)

The linear decomposition is split into intervals:

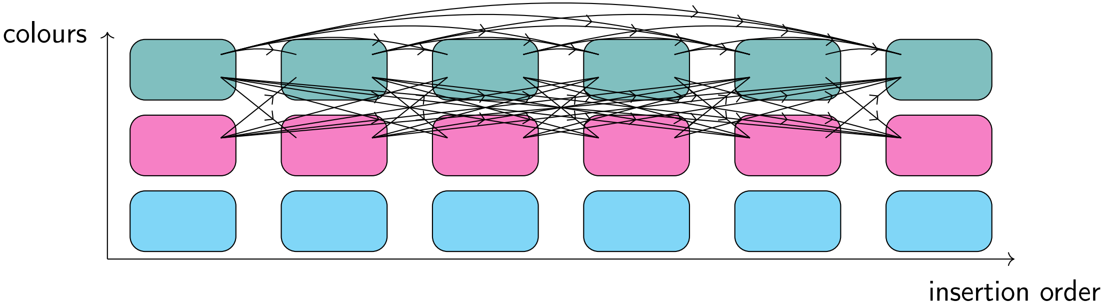
- each interval is 'strictly simpler' (bounded induction)
- vertex colours are consistent at the interval boundaries
- edges between intervals only depend on (1) the relative ordering and (2) the colour.



Simon's Factorisation Magic (grossly oversimplified)

The linear decomposition is split into intervals:

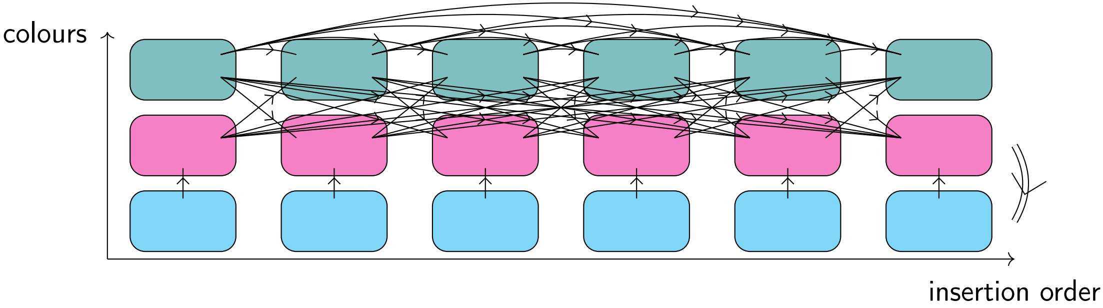
- each interval is 'strictly simpler' (bounded induction)
- vertex colours are consistent at the interval boundaries
- edges between intervals only depend on (1) the relative ordering and (2) the colour.



Simon's Factorisation Magic (grossly oversimplified)

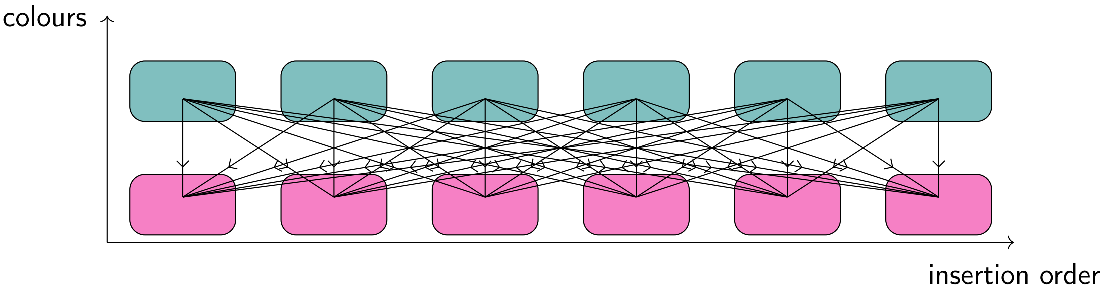
The linear decomposition is split into intervals:

- each interval is 'strictly simpler' (bounded induction)
- vertex colours are consistent at the interval boundaries
- edges between intervals only depend on (1) the relative ordering and (2) the colour.



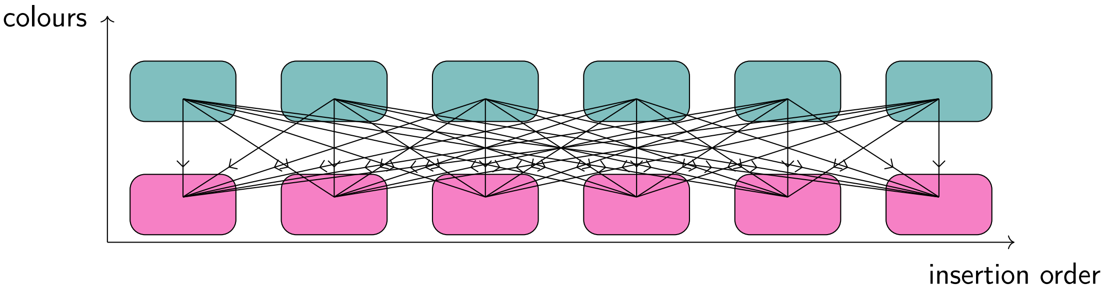
The bad case

All edges green $\rightarrow$ red:

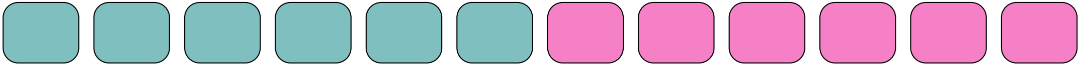


The bad case

All edges green $\to$ red:



Rearrange as:



Open problems

Big question: CMSO-definable decomposition for bounded clique-width graphs.

- Is it easier for tournaments?

Open problems

Big question: CMSO-definable decomposition for bounded clique-width graphs.

- Is it easier for tournaments?
- Is it possible for tournaments using only FO?
  NO: iterated lexicographic product of directed triangles is a counter-example [Bojańczyk]

Open problems

Big question: CMSO-definable decomposition for bounded clique-width graphs.

- Is it easier for tournaments?
- Is it possible for tournaments using only FO?
  NO: iterated lexicographic product of directed triangles is a counter-example [Bojańczyk]

  Maybe with FO + counting modulo 2?

Open problems

Big question: CMSO-definable decomposition for bounded clique-width graphs.

- Is it easier for tournaments?
- Is it possible for tournaments using only FO?
  NO: iterated lexicographic product of directed triangles is a counter-example [Bojańczyk]

  Maybe with FO + counting modulo 2?

- Where does the EMSO / MSO equivalence break down for tournaments?

Open problems

Big question: CMSO-definable decomposition for bounded clique-width graphs.

- Is it easier for tournaments?
- Is it possible for tournaments using only FO?
  NO: iterated lexicographic product of directed triangles is a counter-example [Bojańczyk]

  Maybe with FO + counting modulo 2?

- Where does the EMSO / MSO equivalence break down for tournaments?

Thank you!