Labelled Well Quasi Ordered Classes of Bounded Linear Clique-Width

Aliaume Lopez
University of Warsaw

LaBRI
2024-11-04

https://www.irif.fr/~alopez/

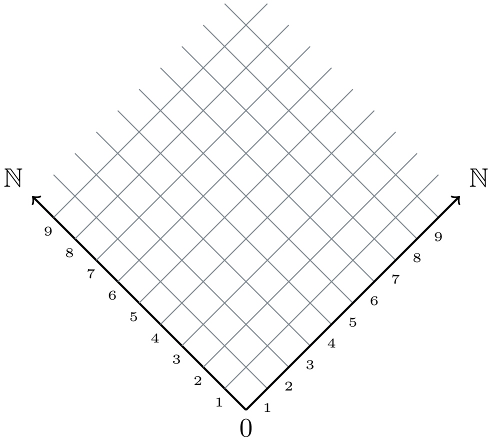
Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.

What is a well-quasi-order (WQO)?

Well Quasi Orderings

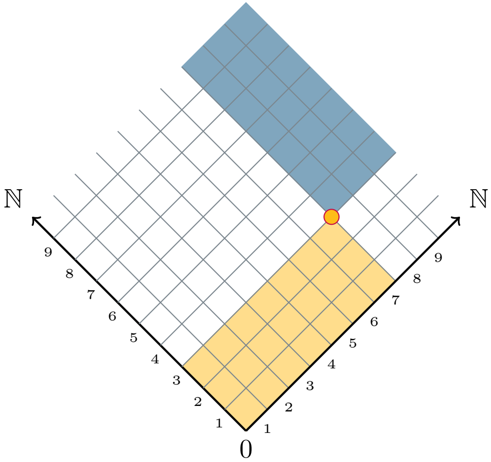
Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Well Quasi Orderings

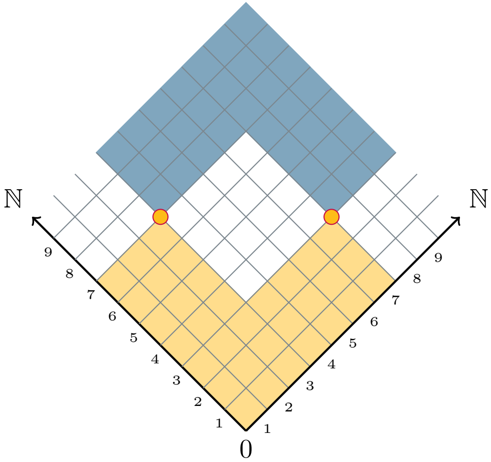
Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Well Quasi Orderings

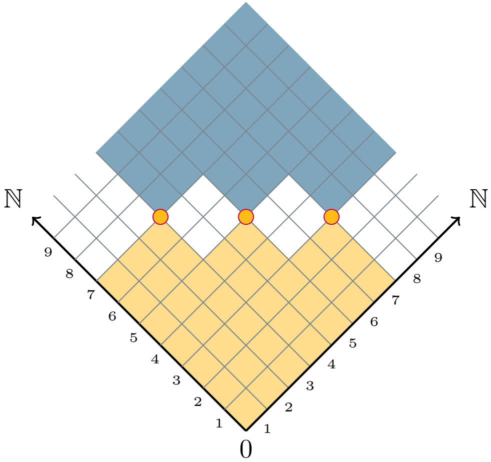
Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Well Quasi Orderings

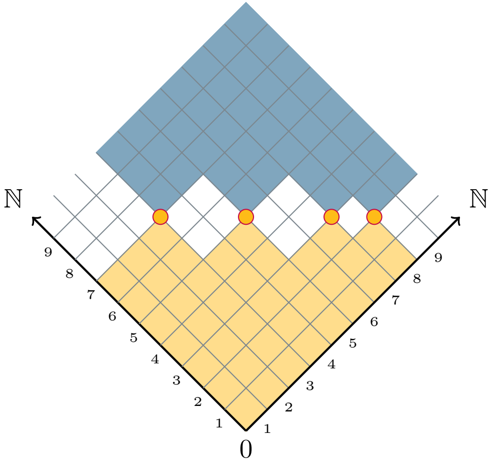
Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Well Quasi Orderings

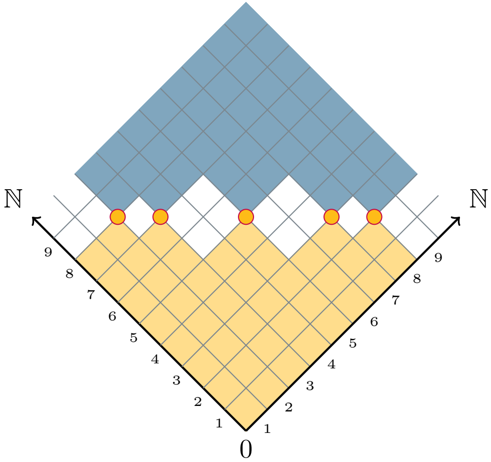
Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Well Quasi Orderings

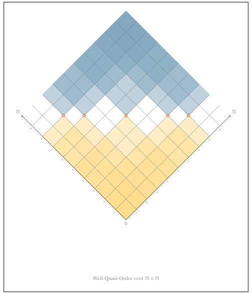
Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

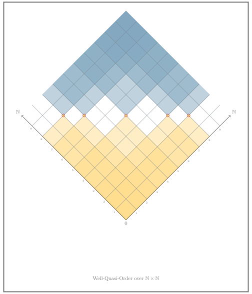
Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



Well Quasi Orderings

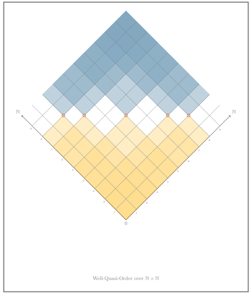
Every infinite sequence of elements contains an increasing pair.



Theory **and** *Practice*

Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



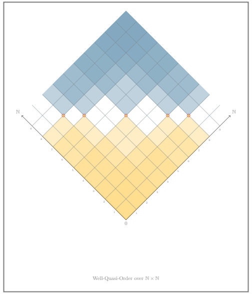
Theory **and** *Practice*

Verification of Well-Structured-Transition-Systems [Abd+96]

Karp-Miller Coverability [KM69]

Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



Theory **and** *Practice*

Verification of Well-Structured-Transition-Systems [Abd+96]

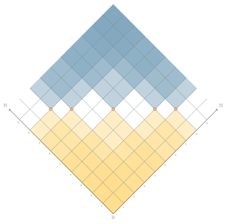
Karp-Miller Coverability [KM69]

Graph Minor Theorem [RS04]

Polynomial time algorithm for graph embeddability into a given surface

Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Theory **and** *Practice*

Verification of Well-Structured-Transition-Systems [Abd+96]

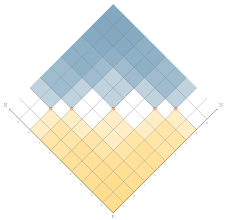
Karp-Miller Coverability [KM69]

Graph Minor Theorem [RS04]

Polynomial time algorithm for graph embeddability into a given surface

Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Theory **and** *Practice*

Verification of Well-Structured-Transition-Systems [Abd+96]

Karp-Miller Coverability [KM69]

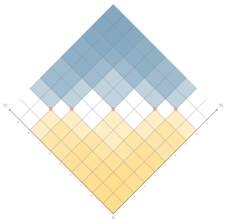
Graph Minor Theorem [RS04]

Polynomial time algorithm for graph embeddability into a given surface

A beautiful *compositional* theory

Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Theory **and** *Practice*

Verification of Well-Structured-Transition-Systems [Abd+96]

Karp-Miller Coverability [KM69]

Graph Minor Theorem [RS04]

Polynomial time algorithm for graph embeddability into a given surface

A beautiful *compositional* theory

$$(X, \leq_X) \times (Y, \leq_Y)$$

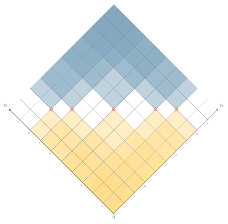
$$(X, \leq_X) + (Y, \leq_Y)$$

$$(X, \leq_X) \subseteq (Y, \leq_Y)$$

Algebraic Operations

Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Theory **and** *Practice*

- Verification of Well-Structured-Transition-Systems [Abd+96]
- Karp-Miller Coverability [KM69]
- Graph Minor Theorem [RS04]
- Polynomial time algorithm for graph embeddability into a given surface

A beautiful *compositional* theory

$$(X, \leq_X) \times (Y, \leq_Y)$$

$$(X, \leq_X) + (Y, \leq_Y)$$

$$(X, \leq_X) \subseteq (Y, \leq_Y)$$

Algebraic Operations

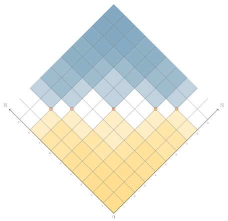
$$(\mathcal{P}_{\text{fin}}(X), \leq_{\text{hoare}})$$

$$(\mathcal{M}^{\diamond}(X), \leq_{\text{multiset}})$$

Set Functors

Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Theory **and** *Practice*

Verification of Well-Structured-Transition-Systems [Abd+96]

Karp-Miller Coverability [KM69]

Graph Minor Theorem [RS04]

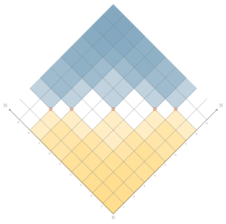
Polynomial time algorithm for graph embeddability into a given surface

A beautiful *compositional* theory

Algebraic Operations	Set Functors	Structural Functors
$(X, \leq_X) \times (Y, \leq_Y)$	$(\mathcal{P}_{\text{fin}}(X), \leq_{\text{hoare}})$	$(X^*, \leq_{\text{subword}})$ [Hig52]
$(X, \leq_X) + (Y, \leq_Y)$	$(\mathcal{M}^\diamond(X), \leq_{\text{multiset}})$	$(\mathsf{Trees}(X), \leq_{\text{subtree}})$ [Kru72]
$(X, \leq_X) \subseteq (Y, \leq_Y)$		$(\mathsf{Graphs}(X), \leq_{\text{minor}})$ [RS04]

Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Theory **and** *Practice*

Verification of Well-Structured-Transition-Systems [Abd+96]

Karp-Miller Coverability [KM69]

Graph Minor Theorem [RS04]

Polynomial time algorithm for graph embeddability into a given surface

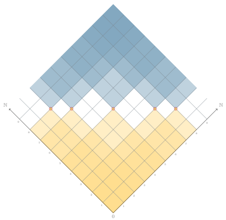
A beautiful *compositional* theory

$(X^*, \leq_{\text{subword}})$	[Hig52]
$(\mathsf{Trees}(X), \leq_{\text{subtree}})$	[Kru72]
$(\mathsf{Graphs}(X), \leq_{\text{minor}})$	[RS04]

Structural Functors

Well Quasi Orderings

Every infinite sequence of elements contains an increasing pair.



Well-Quasi-Order over $\mathbb{N} \times \mathbb{N}$

Theory **and** *Practice*

Verification of Well-Structured-Transition-Systems [Abd+96]

Karp-Miller Coverability [KM69]

Graph Minor Theorem [RS04]

Polynomial time algorithm for graph embeddability into a given surface

A beautiful *compositional* theory

$(X^*, \leq_{\text{subword}})$	[Hig52]
$(\mathsf{Trees}(X), \leq_{\text{subtree}})$	[Kru72]
$(\mathsf{Graphs}(X), \leq_{\text{minor}})$	[RS04]

Structural Functors

Some of these are *inductive* datatypes [Lop23; Fre20], but we want *more*!
Also, can we avoid *minimal bad sequence arguments* [Nas65]?

Classes of Structures...

... as Operators

Structures as Operators

Classes of Structures...

... as Operators

Higman's Subword Embedding

b e a u

Classes of Structures...

... as Operators

Higman's Subword Embedding

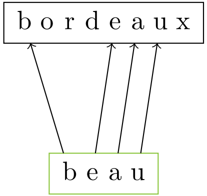
b o r d e a u x

b e a u

Classes of Structures...

... as Operators

Higman's Subword Embedding

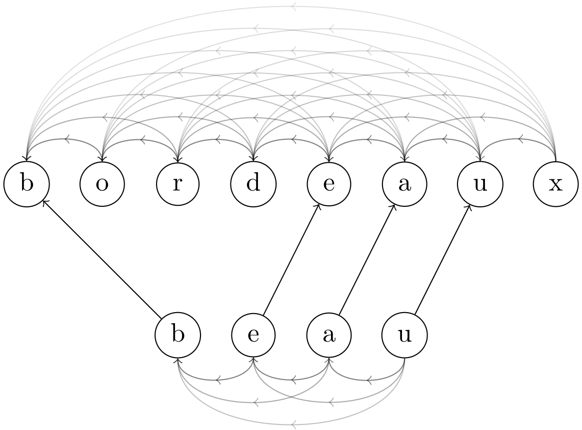


$$\sigma = \{(\leq, 2)\}$$

Classes of Structures...

... as Operators

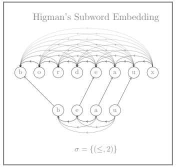
Higman's Subword Embedding



$$\sigma = \{(\leq, 2)\}$$

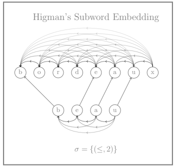
Classes of Structures...

... as Operators

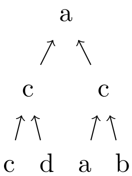


Classes of Structures...

... as Operators

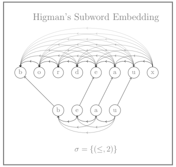


Kruskal's Tree Embedding

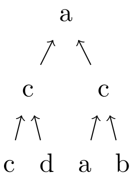


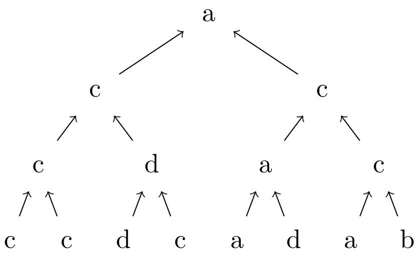
Classes of Structures...

... as Operators



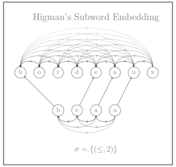
Kruskal's Tree Embedding



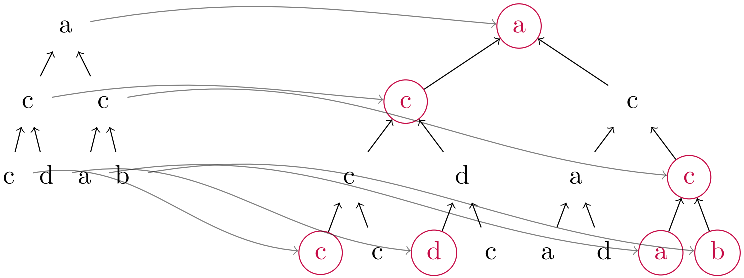


Classes of Structures...

... as Operators



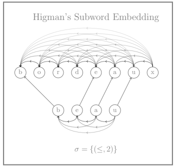
Kruskal's Tree Embedding

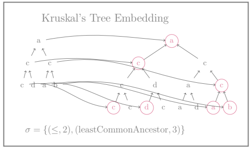


$$\sigma = \{(\leq, 2), (\text{leastCommonAncestor}, 3)\}$$

Classes of Structures...

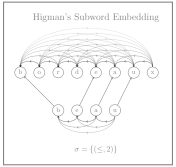
... as Operators

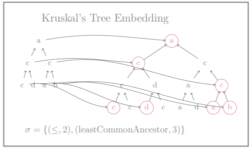




Classes of Structures...

... as Operators





In General

$\mathcal{C}$ a class of finite structures

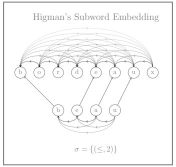
$\mathsf{Label}_X(\mathcal{C})$ the X-labelled structures in $\mathcal{C}$

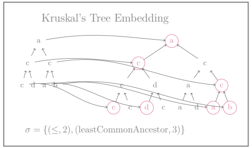
$\mathfrak{A} \leq_X \mathfrak{B}$ if there exists a monotone embedding

$(X, \leq)$ WQO $\implies$ $(\mathsf{Label}_X(\mathcal{C}), \leq_X)$ is a WQO?

Classes of Structures...

... as Operators





In General	This talk
$\mathcal{C}$ a class of finite structures	$\mathcal{C}$ is a class of **finite undirected graphs**
$\mathsf{Label}_X(\mathcal{C})$ the X-labelled structures in $\mathcal{C}$	$\mathsf{Label}_X(\mathcal{C})$ is a class of **X-labelled-graphs**
$\mathfrak{A} \leq_X \mathfrak{B}$ if there exists a monotone embedding	$\leq_X$ is the **induced subgraph** relation
$(X, \leq)$ WQO $\implies$ $(\mathsf{Label}_X(\mathcal{C}), \leq_X)$ is a WQO?	Is the class of graphs a WQO-operator?

Pouzet Conjectures

Graph Theory

Is
$\mathcal{C}$
WQO?

Operators

Is
$\mathcal{C}$
+ WQO
set of labels
WQO?

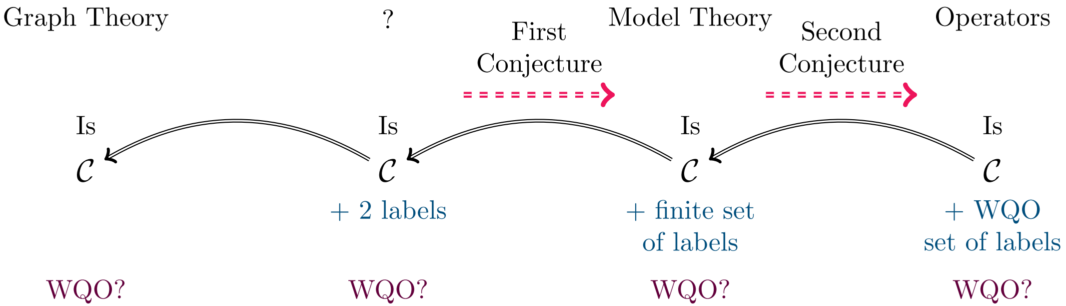
Pouzet Conjectures

Graph Theory	?	Model Theory	Operators
Is $\mathcal{C}$ WQO?	Is $\mathcal{C}$ + 2 labels WQO?	Is $\mathcal{C}$ + finite set of labels WQO?	Is $\mathcal{C}$ + WQO set of labels WQO?

Pouzet Conjectures

Graph Theory	?	Model Theory	Operators
Is $\mathcal{C}$	⇐ Is $\mathcal{C}$ + 2 labels	⇐ Is $\mathcal{C}$ + finite set of labels	⇐ Is $\mathcal{C}$ + WQO set of labels
WQO?	WQO?	WQO?	WQO?

Pouzet Conjectures

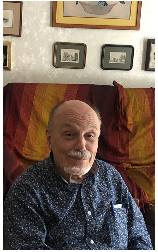


Pouzet Conjectures

Graph Theory	?	Model Theory	Operators
Is $\mathcal{C}$	Is $\mathcal{C}$ + 2 labels	Is $\mathcal{C}$ + finite set of labels	Is $\mathcal{C}$ + WQO set of labels
WQO?	WQO?	WQO?	WQO?

First Conjecture (between "+ 2 labels" and "+ finite set of labels")

Second Conjecture (between "+ finite set of labels" and "+ WQO set of labels")



First Pouzet Conjecture: $2 = \omega$ [Pou72]

Second Pouzet Conjecture: $\omega = \mathsf{WQO}$

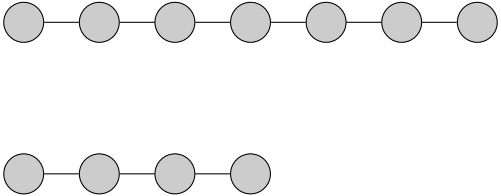
Schmitz's Conjecture: *finite paths* are the only obstruction to 2-WQO

Finite Paths

Finite Paths are WQO but not 2-WQO

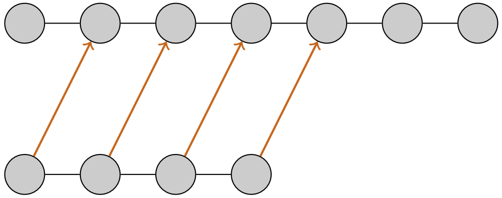
Finite Paths

Finite Paths are WQO but not 2-WQO



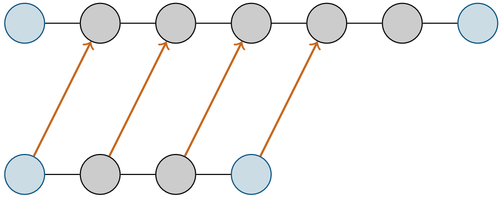
Finite Paths

Finite Paths are WQO but not 2-WQO



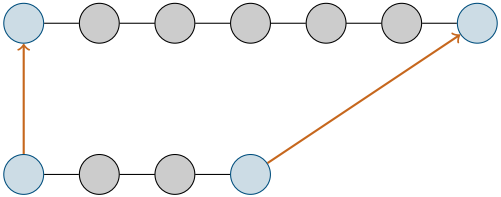
Finite Paths

Finite Paths are WQO but not 2-WQO



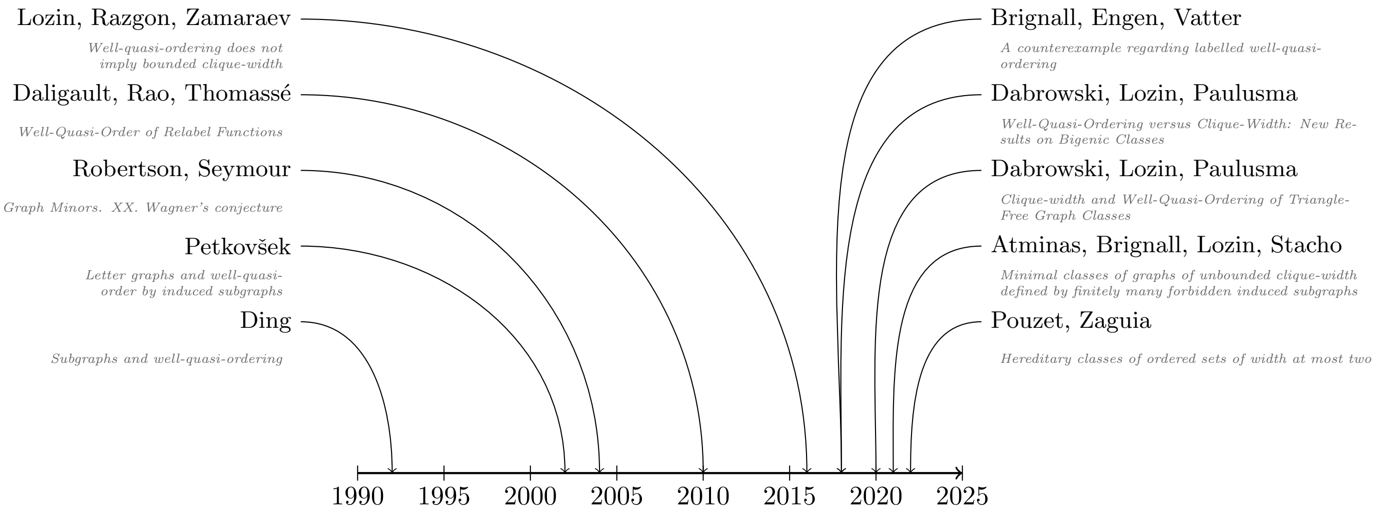
Finite Paths

Finite Paths are WQO but not 2-WQO



State of the Art

Well Quasi Ordering Graph Classes in 2024



Usual Proof Method

From Graphs To Trees

$$(\mathsf{Trees}, \leq) \xrightarrow{\text{monotone, surjective}} (\mathcal{C}, \subseteq_i)$$

Usual Proof Method

From Graphs To Trees

Nested Higman?

$$(\mathsf{Trees}, \leq) \xrightarrow{\text{monotone, surjective}} (\mathcal{C}, \subseteq_i)$$

Usual Proof Method

From Graphs To Trees

Nested Higman?

Kruskal?

$$(\mathsf{Trees}, \leq) \xrightarrow{\text{monotone, surjective}} (\mathcal{C}, \subseteq_i)$$

Usual Proof Method

From Graphs To Trees

Nested Higman?

Kruskal?

$$(\mathsf{Trees}, \leq) \xrightarrow{\text{monotone, surjective}} (\mathcal{C}, \subseteq_i)$$

Typical Example

$$T \in \mathsf{Trees}(\Sigma, \mathbb{P}(\Sigma \times \Sigma)) \longrightarrow G \in \mathcal{C}$$

m-partite cographs

Σ finite with $m = |\Sigma|$

order = Kruskal

$$V(G) = \text{leaves of } T$$

$$(x, y) \in E(G) \iff (c(x), c(y)) \in \mathsf{lca}(x, y)$$

Labelled
Well Quasi Ordered Classes
of
Bounded Linear Clique-Width

Bounded Clique Width

A Logical Approach to Graph Decompositions

Bounded Clique Width

A Logical Approach to Graph Decompositions

$$\mathsf{Trees} \xrightarrow{\mathsf{MSO}\text{ definable, surjective}} \mathcal{C}$$

Bounded Clique Width

A Logical Approach to Graph Decompositions

Introduced by Courcelle [Cou91; Cou94; CER93] with the idea of generalizing **regular languages** to graphs.

$$\text{Trees} \xrightarrow{\text{MSO definable, surjective}} \mathcal{C}$$

Bounded Clique Width

A Logical Approach to Graph Decompositions

Introduced by Courcelle [Cou91; Cou94; CER93] with the idea of generalizing **regular languages** to graphs.

$$\text{Trees} \xrightarrow{\text{MSO definable, surjective}} \mathcal{C}$$

φ_Δ : consider this tree?

$\varphi_{\text{dom}}(x)$: selects vertices of the graph

$\varphi_{\text{edge}}(x, y)$: selects edges of the graph

Bounded Clique Width

A Logical Approach to Graph Decompositions

Introduced by Courcelle [Cou91; Cou94; CER93] with the idea of generalizing **regular languages** to graphs.

$$\text{Trees} \xrightarrow{\text{MSO definable, surjective}} \mathcal{C}$$

φ_Δ : consider this tree?

$\varphi_{\text{dom}}(x)$: selects vertices of the graph

$\varphi_{\text{edge}}(x, y)$: selects edges of the graph

$\varphi_\Delta = \top$, $\varphi_{\text{dom}}(x) = \top$, $\varphi_{\text{edge}}(x, y) = \top$: builds all cliques

Bounded Clique Width

A Logical Approach to Graph Decompositions

Introduced by Courcelle [Cou91; Cou94; CER93] with the idea of generalizing **regular languages** to graphs.

$$\text{Trees} \xrightarrow{\text{MSO definable, surjective}} \mathcal{C}$$

φ_Δ : consider this tree?
$\varphi_{\mathrm{dom}}(x)$: selects vertices of the graph
$\varphi_{\mathrm{edge}}(x, y)$: selects edges of the graph

$\varphi_\Delta = \top$, $\varphi_{\mathrm{dom}}(x) = \top$, $\varphi_{\mathrm{edge}}(x, y) = \top$: builds all cliques

Bounded Clique Width

A Logical Approach to Graph Decompositions

Introduced by Courcelle [Cou91; Cou94; CER93] with the idea of generalizing **regular languages** to graphs.

Trees $\xrightarrow{\text{MSO definable, surjective}}$ $\mathcal{C}$

φ_Δ : consider this tree?

$\varphi_{\text{dom}}(x)$: selects vertices of the graph

$\varphi_{\text{edge}}(x, y)$: selects edges of the graph

$\varphi_\Delta = \top$, $\varphi_{\text{dom}}(x) = \top$, $\varphi_{\text{edge}}(x, y) = \top$: builds all cliques

- Gives a finite representation of graph classes
- Is a well-studied parameter
- Is conjectured to be tightly related to being labelled-WQO

Bounded Clique Width

A Logical Approach to Graph Decompositions

Introduced by Courcelle [Cou91; Cou94; CER93] with the idea of generalizing **regular languages** to graphs.

$$\text{Trees} \xrightarrow{\text{MSO definable, surjective}} \mathcal{C}$$

φ_Δ : consider this tree?

$\varphi_{\text{dom}}(x)$: selects vertices of the graph

$\varphi_{\text{edge}}(x, y)$: selects edges of the graph

$\varphi_\Delta = \top$, $\varphi_{\text{dom}}(x) = \top$, $\varphi_{\text{edge}}(x, y) = \top$: builds all cliques

- Gives a finite representation of graph classes
- Is a well-studied parameter
- Is conjectured to be tightly related to being labelled-WQO

Conjecture [DRT10, Conjecture 5]: $\mathcal{C}$ is hereditary and 2-WQO $\implies$ $\mathcal{C}$ has bounded clique-width

Conjecture (L.): $\mathcal{C}$ is labelled-wqo $\implies$ $\mathcal{C}$ has bounded clique-width

Bounded Clique Width

A Logical Approach to Graph Decompositions

Introduced by Courcelle [Cou91; Cou94; CER93] with the idea of generalizing **regular languages** to graphs.

$$\text{Trees} \xrightarrow{\text{MSO definable, surjective}} \mathcal{C}$$

φ_Δ : consider this tree?

$\varphi_{\mathrm{dom}}(x)$: selects vertices of the graph

$\varphi_{\mathrm{edge}}(x, y)$: selects edges of the graph

$\varphi_\Delta = \top$, $\varphi_{\mathrm{dom}}(x) = \top$, $\varphi_{\mathrm{edge}}(x, y) = \top$: builds all cliques

- Gives a finite representation of graph classes
- Is a well-studied parameter
- Is conjectured to be tightly related to being labelled-WQO

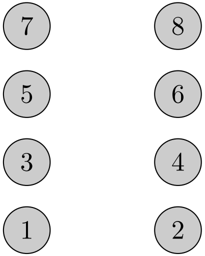
Conjecture [DRT10, Conjecture 5]: $\mathcal{C}$ is hereditary and 2-WQO $\implies$ $\mathcal{C}$ has bounded clique-width

Conjecture (L.): $\mathcal{C}$ is labelled-wqo $\implies$ $\mathcal{C}$ has bounded clique-width

To simplify *everything* consider linear trees, i.e., **finite words**

Bounded **Linear** Clique-Width

b	o	r	d	e	a	u	x
1	2	3	4	5	6	7	8



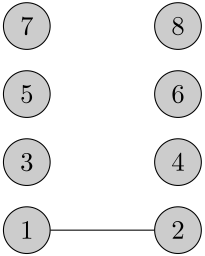
$$\varphi(x, y) = \text{isOdd}(x) \wedge \text{isEven}(y)$$

Bounded **Linear** Clique-Width

b o r d e a u x

1 2 3 4 5 6 7 8

x y



$$\varphi(x, y) = \mathrm{isOdd}(x) \wedge \mathrm{isEven}(y)$$

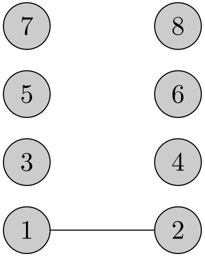
$$\varphi(1, 2) \vee \varphi(2, 1) = \top$$

Bounded **Linear** Clique-Width

b o r d e a u x

1 2 3 4 5 6 7 8

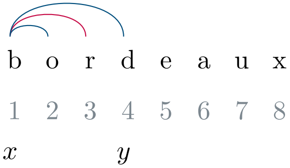
x y

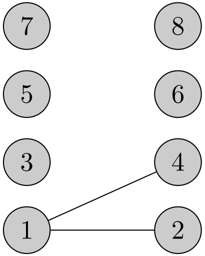


$$\varphi(x, y) = \text{isOdd}(x) \wedge \text{isEven}(y)$$

$$\varphi(1, 3) \vee \varphi(3, 1) = \bot$$

Bounded **Linear** Clique-Width

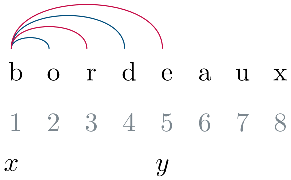


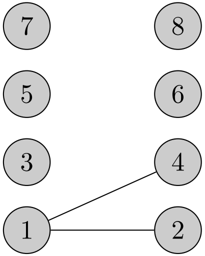


$$\varphi(x, y) = \text{isOdd}(x) \wedge \text{isEven}(y)$$

$$\varphi(1, 4) \vee \varphi(4, 1) = \top$$

Bounded **Linear** Clique-Width

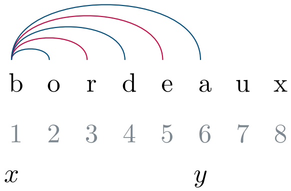


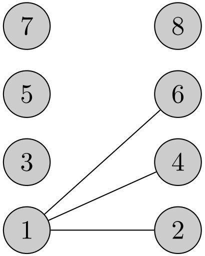


$$\varphi(x, y) = \text{isOdd}(x) \wedge \text{isEven}(y)$$

$$\varphi(1, 5) \vee \varphi(5, 1) = \bot$$

Bounded **Linear** Clique-Width

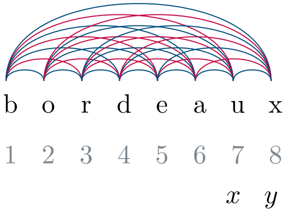


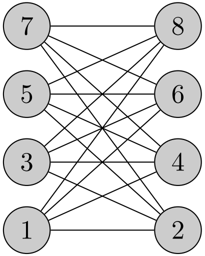


$$\varphi(x, y) = \text{isOdd}(x) \wedge \text{isEven}(y)$$

$$\varphi(1, 6) \vee \varphi(6, 1) = \top$$

Bounded **Linear** Clique-Width





$$\varphi(x, y) = \text{isOdd}(x) \wedge \text{isEven}(y)$$

$$\varphi(7, 8) \vee \varphi(8, 7) = \top$$

The Automata Toolbox

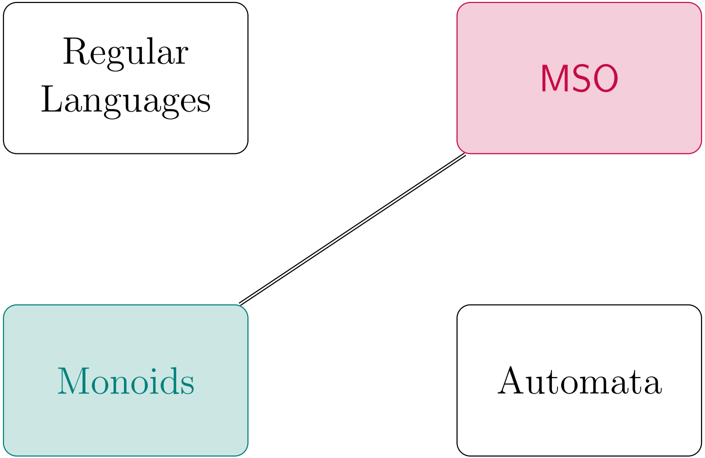
Regular Languages

MSO

Monoids

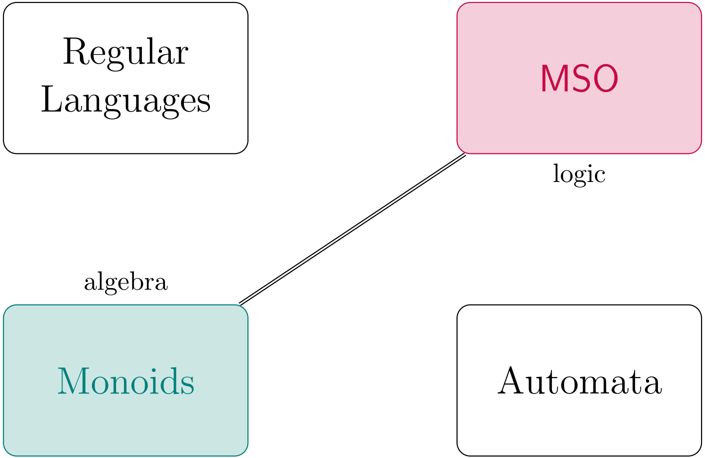
Automata

The Automata Toolbox



The Automata Toolbox

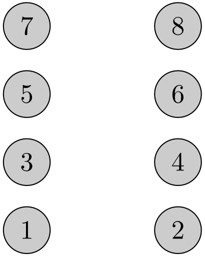
$\mathrm{isOdd}(x) \wedge \mathrm{isEven}(y)$



$M = (\mathbb{Z}/2\mathbb{Z}, +)$

$\mu \colon \Sigma \to M$

Bounded **Linear** Clique-Width



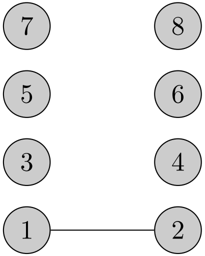
b o r d e a u x

1 2 3 4 5 6 7 8

$M = (\mathbb{Z}/2\mathbb{Z}, +), \mu(x) = 1$

$P \subseteq M^3 = \{(x, 0, y) \mid (x, y) \in \mathbb{Z}/2\mathbb{Z}\}$

Bounded **Linear** Clique-Width



b o r d e a u x

1 2 3 4 5 6 7 8

$\vdash$ —— x - y ———————— $\dashv$

0 0 0

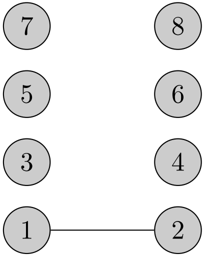
$M = (\mathbb{Z}/2\mathbb{Z}, +), \mu(x) = 1$ $P \subseteq M^3 = \{(x, 0, y) \mid (x, y) \in \mathbb{Z}/2\mathbb{Z}\}$

$$(\mu(w[1:1]), \mu(w[1:2]), \mu(w[2:|w|])$$

$$= (0, 0, 0)$$

$$\in P? \ \top$$

Bounded **Linear** Clique-Width



b o r d e a u x

1 2 3 4 5 6 7 8

$\vdash$ — x — y ———— $\dashv$

0 1 1

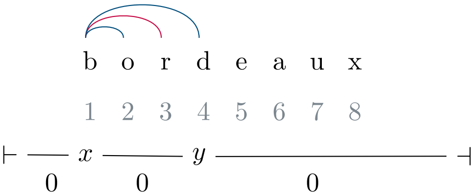
$M = (\mathbb{Z}/2\mathbb{Z}, +), \mu(x) = 1$ $\quad$ $P \subseteq M^3 = \{(x, 0, y) \mid (x, y) \in \mathbb{Z}/2\mathbb{Z}\}$

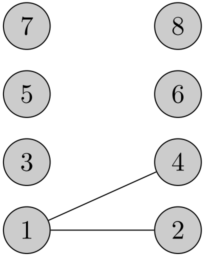
$$(\mu(w[1:1]), \mu(w[1:3]), \mu(w[3:|w|])$$

$$= (0, 1, 1)$$

$$\in P?\ \bot$$

Bounded **Linear** Clique-Width





$M = (\mathbb{Z}/2\mathbb{Z}, +), \mu(x) = 1$

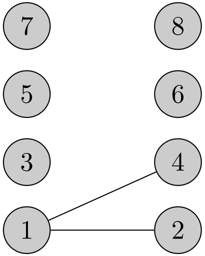
$P \subseteq M^3 = \{(x, 0, y) \mid (x, y) \in \mathbb{Z}/2\mathbb{Z}\}$

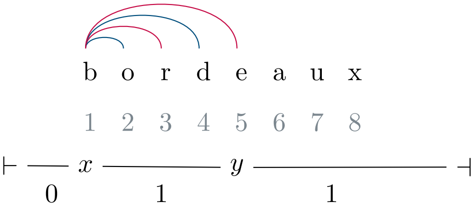
$$(\mu(w[1:1]), \mu(w[1:4]), \mu(w[4:|w|])$$

$$= (0, 0, 0)$$

$$\in P? \ \top$$

Bounded **Linear** Clique-Width





$M = (\mathbb{Z}/2\mathbb{Z}, +), \mu(x) = 1$

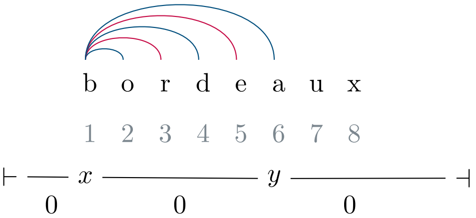
$P \subseteq M^3 = \{(x, 0, y) \mid (x, y) \in \mathbb{Z}/2\mathbb{Z}\}$

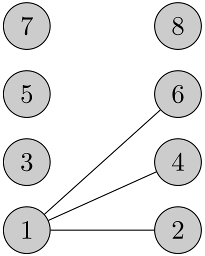
$$(\mu(w[1:1]), \mu(w[1:5]), \mu(w[5:|w|])$$

$$= (0, 1, 1)$$

$$\in P? \ \bot$$

Bounded **Linear** Clique-Width





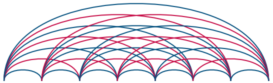
$M = (\mathbb{Z}/2\mathbb{Z}, +), \mu(x) = 1$ $\quad P \subseteq M^3 = \{(x, 0, y) \mid (x, y) \in \mathbb{Z}/2\mathbb{Z}\}$

$$(\mu(w[1:1]), \mu(w[1:6]), \mu(w[6:|w|])$$

$$= (0, 0, 0)$$

$$\in P? \ \top$$

Bounded **Linear** Clique-Width



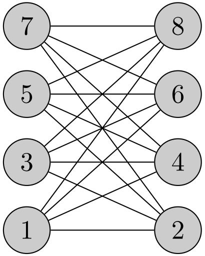
b o r d e a u x

1 2 3 4 5 6 7 8

$\vdash$ ———————— x - y ——— $\dashv$

0 0 0

$M = (\mathbb{Z}/2\mathbb{Z}, +), \mu(x) = 1$



$P \subseteq M^3 = \{(x, 0, y) \mid (x, y) \in \mathbb{Z}/2\mathbb{Z}\}$

$(\mu(w[1:7]), \mu(w[7:8]), \mu(w[8:|w|])$

$= (0, 0, 0)$

$\in P?\ \top$

Simon's Factorisation(s)

A Ramsey-like fundamental theorem of semigroup theory [Sim90].

Simon's Factorisation(s)

A Ramsey-like fundamental theorem of semigroup theory [Sim90].

$M = (\mathbb{Z}/2\mathbb{Z}, +)$

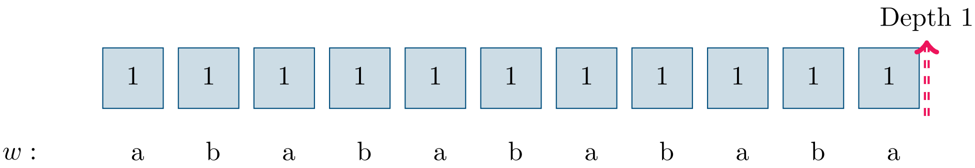
$w :$ a b a b a b a b a b a

$0 + 0 = 0$

Simon's Factorisation(s)

A Ramsey-like fundamental theorem of semigroup theory [Sim90].

$$M = (\mathbb{Z}/2\mathbb{Z}, +)$$

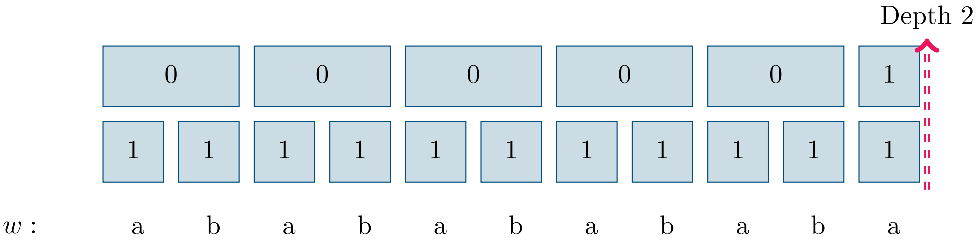


$$0 + 0 = 0$$

Simon's Factorisation(s)

A Ramsey-like fundamental theorem of semigroup theory [Sim90].

$$M = (\mathbb{Z}/2\mathbb{Z}, +)$$

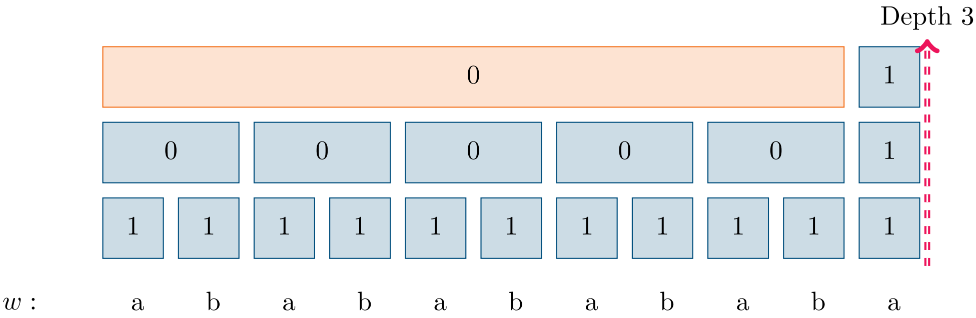


$$0 + 0 = 0$$

Simon's Factorisation(s)

A Ramsey-like fundamental theorem of semigroup theory [Sim90].

$$M = (\mathbb{Z}/2\mathbb{Z}, +)$$

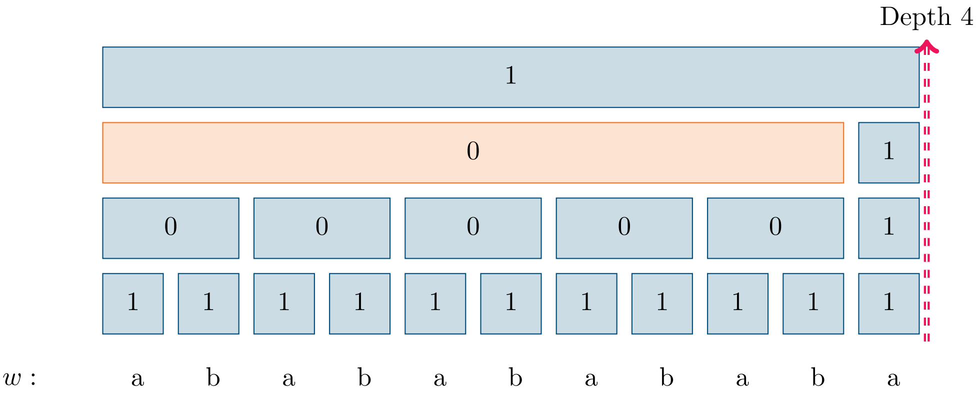


$$0 + 0 = 0$$

Simon's Factorisation(s)

A Ramsey-like fundamental theorem of semigroup theory [Sim90].

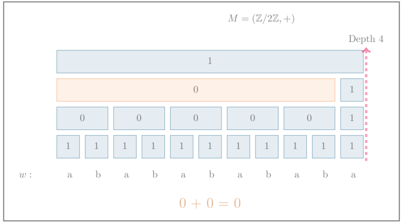
$M = (\mathbb{Z}/2\mathbb{Z}, +)$



$0 + 0 = 0$

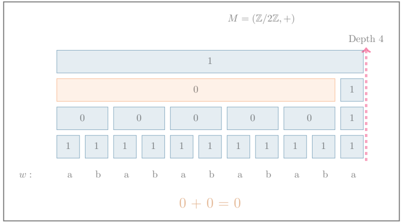
Simon's Factorisation(s)

A Ramsey-like fundamental theorem of semigroup theory [Sim90].



Simon's Factorisation(s)

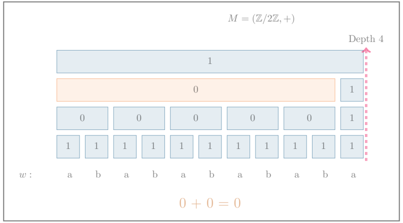
A Ramsey-like fundamental theorem of semigroup theory [Sim90].



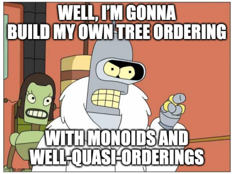
Theorem (Simon [Sim90]): for every word w and monoid M, there exists a factorisation of depth $f(|M|)$ independent of w.

Simon's Factorisation(s)

A Ramsey-like fundamental theorem of semigroup theory [Sim90].



Theorem (Simon [Sim90]): for every word w and monoid M, there exists a factorisation of depth $f(|M|)$ independent of w.



BACK TO THE TREES!

Take Away Food For Thought

If there was only one thing to remember from this talk ...

Good Orders On Factorisations $\simeq$ Well-Quasi-Ordered classes of graphs

Take Away Food For Thought

If there was only one thing to remember from this talk ...

Nested Higman Over Simon's	$\simeq$	Well-Quasi-Ordered classes of graphs

Take Away Food For Thought

If there was only one thing to remember from this talk ...

Gap-Embedding [DT03] $\simeq$ Well-Quasi-Ordered classes of graphs

Take Away Food For Thought

If there was only one thing to remember from this talk ...

Gap-Embedding [DT03]	$\simeq$	Well-Quasi-Ordered classes of graphs

Logic + WQO = ♡

The Main Issue

Inserting idempotents does not affect **far away** positions...

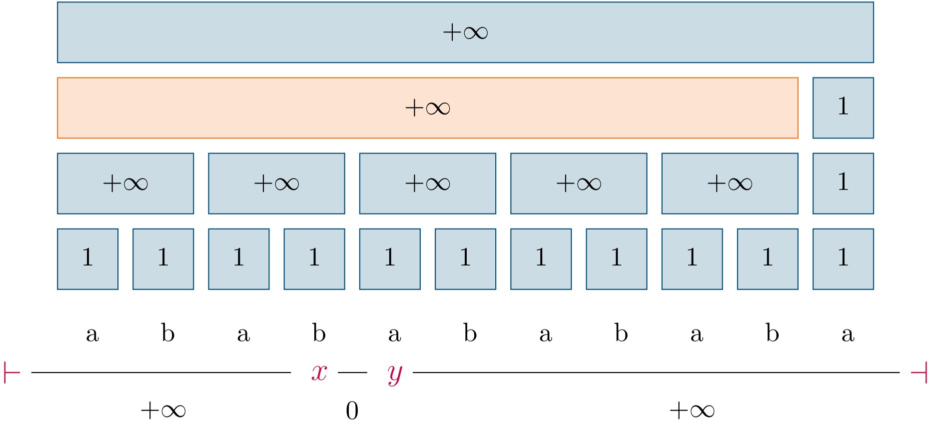
... But does affect **close** positions!

$$M = (\{0, 1, +\infty\}, +),\ \mu(\varepsilon) = 0,\ \mu(a) = \mu(b) = 1$$

The Main Issue

Inserting idempotents does not affect **far away** positions...

... But does affect **close** positions!

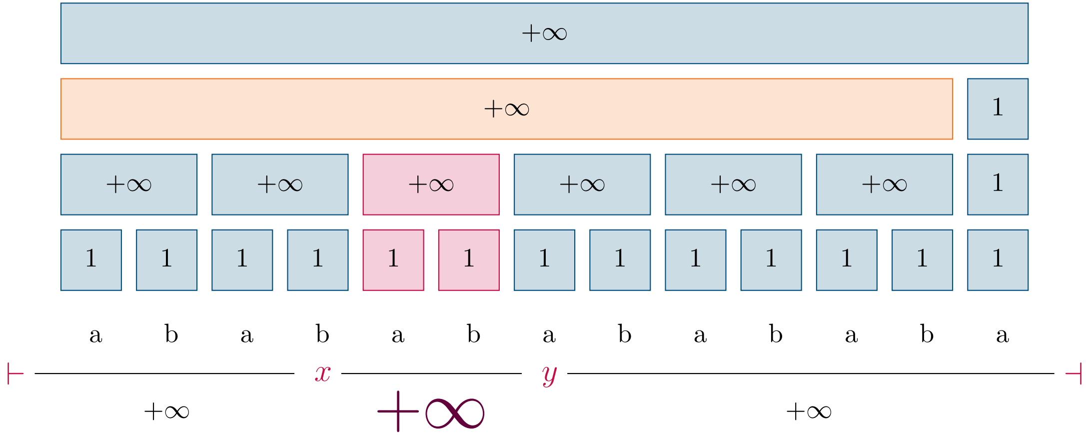


$$M = (\{0, 1, +\infty\}, +),\ \mu(\varepsilon) = 0,\ \mu(a) = \mu(b) = 1$$

The Main Issue

Inserting idempotents does not affect **far away** positions...

... But does affect **close** positions!



$$M = (\{0, 1, +\infty\}, +),\ \mu(\varepsilon) = 0,\ \mu(a) = \mu(b) = 1$$

Conclusion / Ask Me Anything

Theorem 1: Given an MSO transduction, one can **decide** if its *image* is labelled-WQO, and **compute** a k such that k-wqo $\iff$ labelled-wqo.

Theorem 2: For every class $\mathcal{C}$ of bounded *linear* clique-width, labelled-wqo $\iff$ wqo-operator.

Conclusion / Ask Me Anything

Theorem 1: Given an MSO transduction, one can **decide** if its *image* is labelled-WQO, and **compute** a k such that k-wqo $\iff$ labelled-wqo.

Theorem 2: For every class $\mathcal{C}$ of bounded *linear* clique-width, labelled-wqo $\iff$ wqo-operator.

Remark 3: Previous works of [DRT10] can be explained in this framework, using a nice monoid variety [Kun05].

Remark 4: No *minimal bad sequence argument* [Nas65], just the *Gap-Embedding* of [DT03].

Conclusion / Ask Me Anything

Theorem 1: Given an MSO transduction, one can **decide** if its *image* is labelled-WQO, and **compute** a k such that k-wqo $\iff$ labelled-wqo.

Theorem 2: For every class $\mathcal{C}$ of bounded *linear* clique-width, labelled-wqo $\iff$ wqo-operator.

Remark 3: Previous works of [DRT10] can be explained in this framework, using a nice monoid variety [Kun05].

Remark 4: No *minimal bad sequence argument* [Nas65], just the *Gap-Embedding* of [DT03].

New Interests: Successor-free tree representations! Monotone MSO-transductions!

Conclusion / Ask Me Anything

Theorem 1: Given an MSO transduction, one can **decide** if its *image* is labelled-WQO, and **compute** a k such that k-wqo $\iff$ labelled-wqo.

Theorem 2: For every class $\mathcal{C}$ of bounded *linear* clique-width, labelled-wqo $\iff$ wqo-operator.

Remark 3: Previous works of [DRT10] can be explained in this framework, using a nice monoid variety [Kun05].

Remark 4: No *minimal bad sequence argument* [Nas65], just the *Gap-Embedding* of [DT03].

New Interests: Successor-free tree representations! Monotone MSO-transductions!

In the near future:
Clique-Width (almost written up, using [Lop23]),
Schmitz's Conjecture (almost done for clique-width),
labelled-wqo implies ω-categorical (in progress).

References

[Lop23] Aliaume Lopez. “Fixed Points and Noetherian Topologies”. In: *Foundations of Software Science and Computation Structures*. Ed. by Orna Kupferman and Pawel Sobocinski. Vol. 13992. Springer Nature Switzerland, 2023, pp. 456–476. ISBN: 978-3-031-30828-4. DOI: 10.1007/978-3-031-30829-1_22. eprint: 2207.07614. URL: https://link.springer.com/chapter/10.1007/978-3-031-30829-1_22.

[Fre20] Anton Freund. “From Kruskal’s theorem to Friedman’s gap condition”. In: *Mathematical Structures in Computer Science* 30.8 (2020), pp. 952–975. DOI: 10.1017/S0960129520000298.

[DRT10] Jean Daligault, Michael Rao, and Stéphan Thomassé. “Well-Quasi-Order of Relabel Functions”. In: *Order* 27.3 (Sept. 2010), pp. 301–315. ISSN: 1572-9273. DOI: 10.1007/s11083-010-9174-0. URL: http://dx.doi.org/10.1007/s11083-010-9174-0.

[Kun05] Michal Kunc. “Regular solutions of language inequalities and well quasi-orders”. In: *Theoretical Computer Science* 348.2 (2005). Automata, Languages and Programming: Algorithms and Complexity (ICALP-A 2004), pp. 277–293. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2005.09.018. URL: https://www.sciencedirect.com/science/article/pii/S0304397505005384.

[RS04] Neil Robertson and Paul D. Seymour. “Graph Minors. XX. Wagner’s conjecture”. In: *Journal of Combinatorial Theory, Series B* 92.2 (2004). Special Issue Dedicated to Professor W.T. Tutte, pp. 325–357. ISSN: 0095-8956. DOI: 10.1016/j.jctb.2004.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0095895604000784.

[DT03] Nachum Derhowitz and Iddo Tzameret. “Gap Embedding for Well-Quasi-Orderings1”. In: *Electronic Notes in Theoretical Computer Science* 84 (2003). WoLLIC’2003, 10th Workshop on Logic, Language, Information and Computation, pp. 80–90. ISSN: 1571-0661. DOI: 10.1016/S1571-0661(04)80846-6. URL: https://www.sciencedirect.com/science/article/pii/S1571066104808466.

[Abd+96] Parosh Aziz Abdulla, Karlis Čerāns, Bengt Jonsson Tsay, and Yih-Kuen. “General decidability theorems for infinite-state systems”. In: *Proceedings of LICS’96*. IEEE, 1996, pp. 313–321. DOI: 10.1109/LICS.1996.561359.

[Cou94] Bruno Courcelle. “Monadic second-order definable graph transductions: a survey”. In: *Theoretical Computer Science* 126.1 (1994), pp. 53–75. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(94)90268-2. URL: https://www.sciencedirect.com/science/article/pii/0304397594902682.

[CER93] Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. “Handle-rewriting hypergraph grammars”. In: *Journal of Computer and System Sciences* 46.2 (1993), pp. 218–270. ISSN: 0022-0000. DOI: https://doi.org/10.1016/0022-0000(93)90004-G. URL: https://www.sciencedirect.com/science/[illegible].

[Cou91] Bruno Courcelle. “The monadic second-order logic of graphs V: on closing the gap between definability and recognizability”. In: *Theoretical Computer Science* 80.2 (1991), pp. 153–202. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(91)90387-H. URL: https://www.sciencedirect.com/science/article/pii/030439759190387H.

[Sim90] Imre Simon. “Factorization forests of finite height”. In: *Theoretical Computer Science* 72.1 (1990), pp. 65–94. ISSN: 0304-3975. DOI: 10.1016/0304-3975(90)90047-L. URL: https://www.sciencedirect.com/science/article/pii/030439759090047L.

[Kru72] Joseph B. Kruskal. “The theory of well-quasi-ordering: A frequently discovered concept”. In: *Journal of Combinatorial Theory, Series A* 13 (1972), pp. 297–305. DOI: 10.1016/0097-3165(72)90063-5.

[Pou72] Maurice Pouzet. “Un bel ordre d’abritement et ses rapports avec les bornes d’une multirelation”. In: *CR Acad. Sci. Paris Sér. AB* 274 (1972), A1677–A1680.

[KM69] Richard M. Karp and Raymond E. Miller. “Parallel program schemata”. In: *Journal of Computer and System Sciences* 3.2 (1969), pp. 147–195. ISSN: 0022-0000. DOI: https://doi.org/10.1016/S0022-0000(69)80011-5. URL: https://www.sciencedirect.com/science/article/pii/S0022000069800115.

[Nas65] Crispin St. John Alvah Nash-Williams. “On well-quasi-ordering transfinite sequences”. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 61 (1965), pp. 33–39. DOI: 10.1017/S0305004100038603.

[Hig52] Graham Higman. “Ordering by divisibility in abstract algebras”. In: *Proceedings of the London Mathematical Society* 3 (1952), pp. 326–336. DOI: 10.1112/plms/s3-2.1.326.