Big Ramsey degrees of homogeneous structures

Jan Hubička

Department of Applied Mathematics
Charles University
Prague

Joint work with Martin Balko, Natasha Dobrinen, David Chodounský, Matěj Konečný, Jaroslav Nešetřil, Stevo Todorcevic, Lluis Vena, Andy Zucker

XVI International Luminy Workshop in Set Theory

Big Ramsey Degrees

Theorem (Infinite Ramsey Theorem, 1930)

$$\forall_{p,k\geq 1} : \omega \longrightarrow (\omega)^{p}_{k,1}.$$

Big Ramsey Degrees

Theorem (Infinite Ramsey Theorem, 1930)

$$\forall_{p,k\geq 1} : \omega \longrightarrow (\omega)^p_{k,1}.$$

$N \longrightarrow (n)^p_{k,t}$: For every partition of $\binom{\omega}{p}$ into k classes (colours) there exists $X \in \binom{\omega}{\omega}$ such that $\binom{X}{p}$ belongs to at most t parts.

($t = 1$ means that $\binom{X}{p}$ is monochromatic.)

Big Ramsey Degrees

Theorem (Infinite Ramsey Theorem, 1930)

$$\forall_{p,k\geq 1} : \omega \longrightarrow (\omega)^p_{k,1}.$$

Structural formulation:

Theorem (Infinite Ramsey Theorem, 1930)

Let $\mathcal{O}$ be the class of all finite linear orders.

$$\forall_{(O,\leq_O)\in\mathcal{O},k\geq 1} : (\omega,\leq) \longrightarrow (\omega,\leq)^{(O,\leq_O)}_{k,1}.$$

Big Ramsey Degrees

Theorem (Infinite Ramsey Theorem, 1930)

$$\forall_{p,k\geq 1}: \omega \longrightarrow (\omega)^p_{k,1}.$$

Structural formulation:

Theorem (Infinite Ramsey Theorem, 1930)

Let $\mathcal{O}$ be the class of all finite linear orders.

$$\forall_{(O,\leq_O)\in\mathcal{O},k\geq 1}: (\omega,\leq) \longrightarrow (\omega,\leq)^{(O,\leq_O)}_{k,1}.$$

$\binom{\mathbf{B}}{\mathbf{A}}$ is the set of all embeddings of structure **A** to structure **B**.

$\mathbf{C} \longrightarrow (\mathbf{B})^{\mathbf{A}}_{k,t}$: For every k-colouring of $\binom{\mathbf{C}}{\mathbf{A}}$ there exists $f \in \binom{\mathbf{C}}{\mathbf{B}}$ such that $\binom{f(\mathbf{B})}{\mathbf{A}}$ has at most t colours.

Big Ramsey Degrees

Theorem (Infinite Ramsey Theorem, 1930)

$$\forall_{p,k\geq 1} : \omega \longrightarrow (\omega)^p_{k,1}.$$

Structural formulation:

Theorem (Infinite Ramsey Theorem, 1930)

Let $\mathcal{O}$ be the class of all finite linear orders.

$$\forall_{(O,\leq_O)\in\mathcal{O},k\geq 1} : (\omega,\leq) \longrightarrow (\omega,\leq)^{(O,\leq_O)}_{k,1}.$$

Is the same true for $(\mathbb{Q},\leq)$?

$$\forall_{(O,\leq_O)\in\mathcal{O},k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,1}.$$

Big Ramsey Degrees

Theorem (Infinite Ramsey Theorem, 1930)

$$\forall_{p,k\geq 1} : \omega \longrightarrow (\omega)^p_{k,1}.$$

Structural formulation:

Theorem (Infinite Ramsey Theorem, 1930)

Let $\mathcal{O}$ be the class of all finite linear orders.

$$\forall_{(O,\leq_O)\in\mathcal{O},k\geq 1} : (\omega,\leq) \longrightarrow (\omega,\leq)^{(O,\leq_O)}_{k,1}.$$

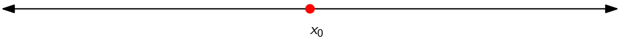
Is the same true for $(\mathbb{Q},\leq)$?

$$\forall_{(O,\leq_O)\in\mathcal{O},k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,1}.$$

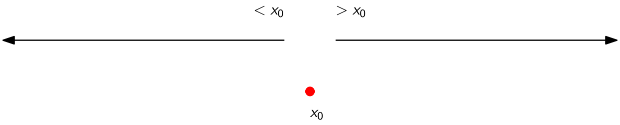
Sierpiński: not true for $|O| = 2$.

Rich colouring of $\mathbb{Q}$

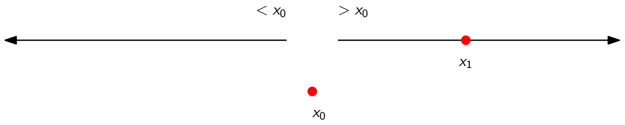
Rich colouring of $\mathbb{Q}$



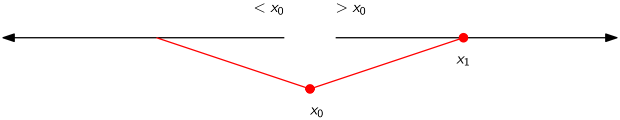
Rich colouring of $\mathbb{Q}$



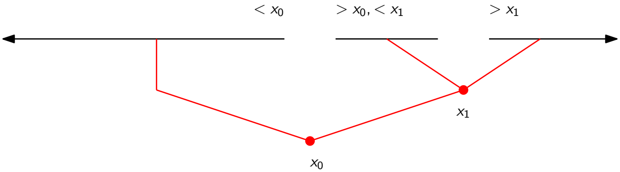
Rich colouring of $\mathbb{Q}$



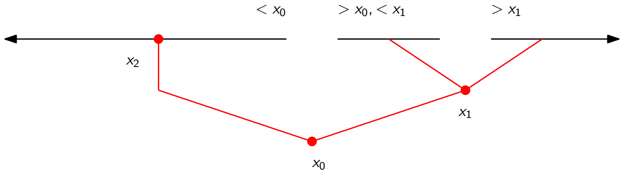
Rich colouring of $\mathbb{Q}$



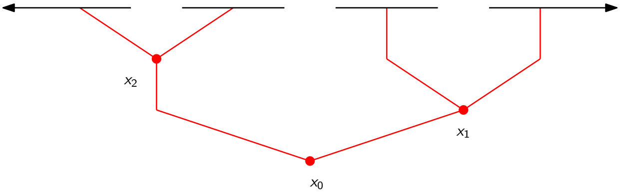
Rich colouring of $\mathbb{Q}$



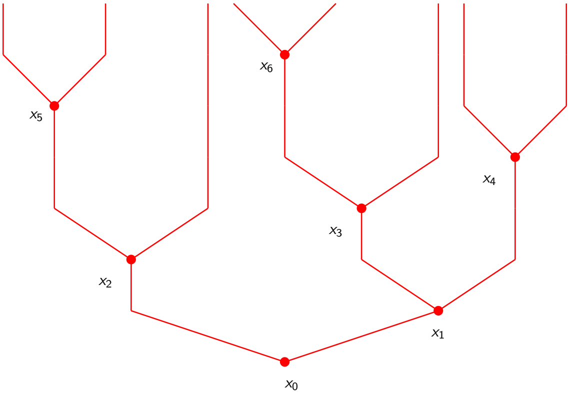
Rich colouring of $\mathbb{Q}$



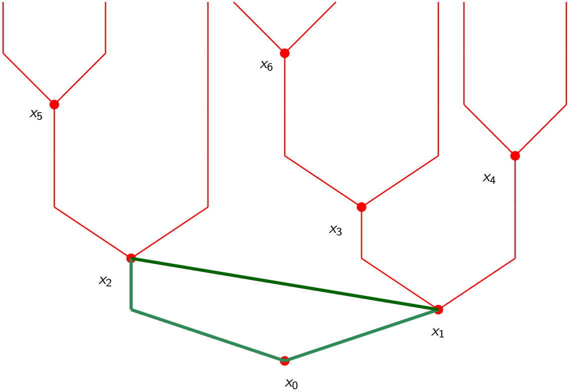
Rich colouring of $\mathbb{Q}$



Rich colouring of $\mathbb{Q}$

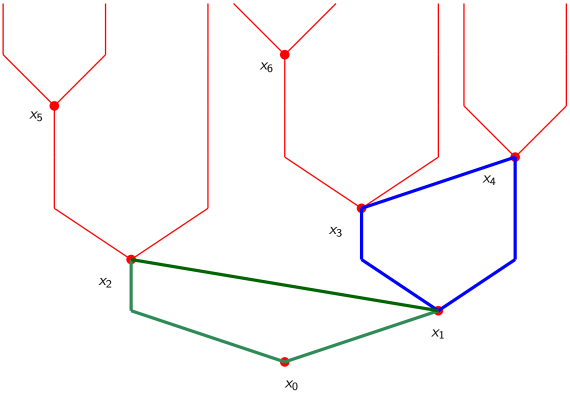


Rich colouring of $\mathbb{Q}$



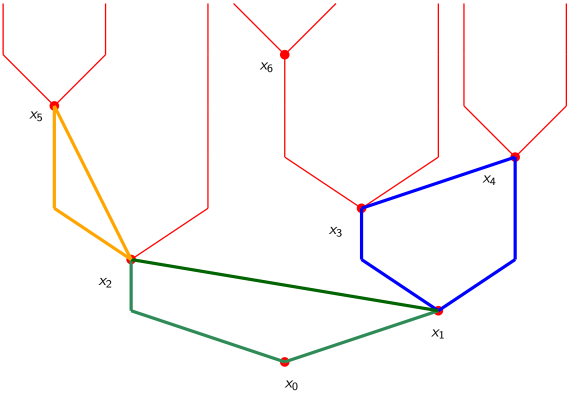
Colour of k-tuple = shape of meet closure in the tree

Rich colouring of $\mathbb{Q}$



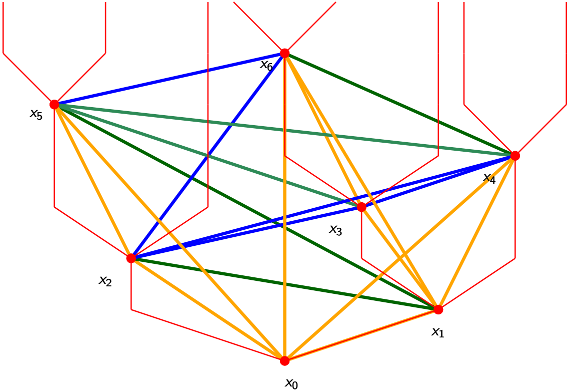
Colour of k-tuple = shape of meet closure in the tree

Rich colouring of $\mathbb{Q}$



Colour of k-tuple = shape of meet closure in the tree

Rich colouring of $\mathbb{Q}$



Colour of k-tuple = shape of meet closure in the tree

Big Ramsey Degrees

In late 1960's Laver developed method of finding copies of $\mathbb{Q}$ in $\mathbb{Q}$ with bounded number of colours using Milliken's tree theorem.

Theorem (Devlin, 1979)

$$\forall_{(O,\leq_O)\in\mathcal{O}}\exists_{T=T(|O|)\in\omega}\forall_{k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,T}.$$

$T(n)$ is the big Ramsey degree of n tuple in $\mathbb{Q}$.

Big Ramsey Degrees

In late 1960's Laver developed method of finding copies of $\mathbb{Q}$ in $\mathbb{Q}$ with bounded number of colours using Milliken's tree theorem.

Theorem (Devlin, 1979)

$$\forall_{(O,\leq_O)\in\mathcal{O}}\exists_{T=T(|O|)\in\omega}\forall_{k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,T}.$$

$T(n)$ is the big Ramsey degree of n tuple in $\mathbb{Q}$.

$$T(n) = \tan^{(2n-1)}(0).$$

$\tan^{(2n-1)}(0)$ is the $(2n-1)^{\text{st}}$ derivative of the tangent evaluated at 0.

Big Ramsey Degrees

In late 1960's Laver developed method of finding copies of $\mathbb{Q}$ in $\mathbb{Q}$ with bounded number of colours using Milliken's tree theorem.

Theorem (Devlin, 1979)

$$\forall_{(O,\leq_O)\in\mathcal{O}}\exists_{T=T(|O|)\in\omega}\forall_{k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,T}.$$

$T(n)$ is the big Ramsey degree of n tuple in $\mathbb{Q}$.

$$T(n) = \tan^{(2n-1)}(0).$$

$\tan^{(2n-1)}(0)$ is the $(2n-1)^{\text{st}}$ derivative of the tangent evaluated at 0.

$$T(1) = 1,\ T(2) = 2,\ T(3) = 16,\ T(4) = 272,$$

$$T(5) = 7936,\ T(6) = 353792,\ T(7) = 22368256$$

Big Ramsey Degrees

In late 1960's Laver developed method of finding copies of $\mathbb{Q}$ in $\mathbb{Q}$ with bounded number of colours using Milliken's tree theorem.

Theorem (Devlin, 1979)

$$\forall_{(O,\leq_O)\in\mathcal{O}}\exists_{T=T(|O|)\in\omega}\forall_{k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,T}.$$

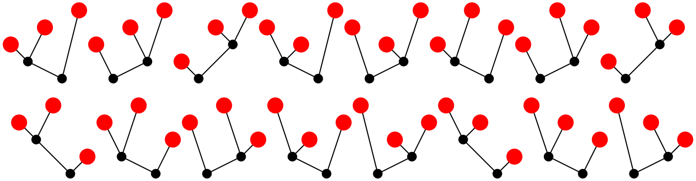
$T(n)$ is the big Ramsey degree of n tuple in $\mathbb{Q}$.

$$T(n) = \tan^{(2n-1)}(0).$$

$\tan^{(2n-1)}(0)$ is the $(2n-1)^{\text{st}}$ derivative of the tangent evaluated at 0.

$$T(1) = 1,\ T(2) = 2,\ T(3) = 16,\ T(4) = 272,$$

$$T(5) = 7936,\ T(6) = 353792,\ T(7) = 22368256$$



Upper bounds

Big picture: proof techniques

Ramsey's Theorem

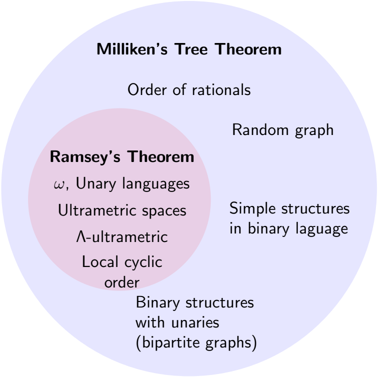
ω, Unary languages

Ultrametric spaces

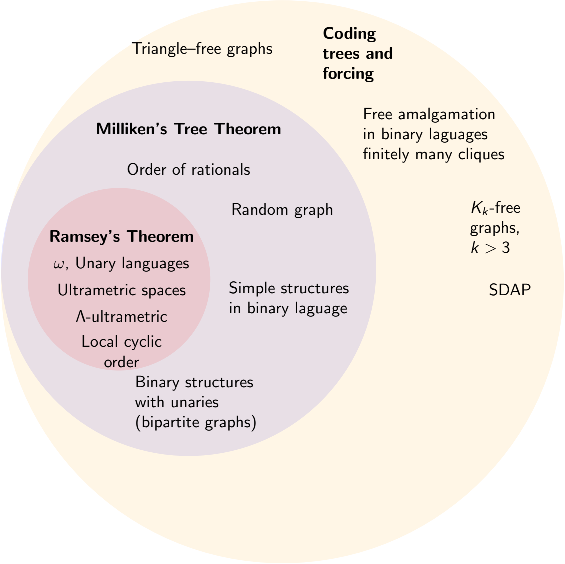
Λ-ultrametric

Local cyclic order

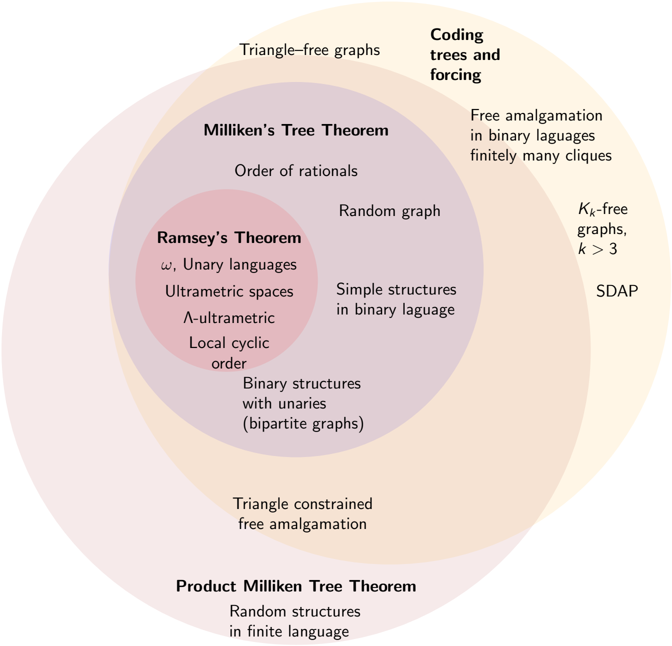
Big picture: proof techniques



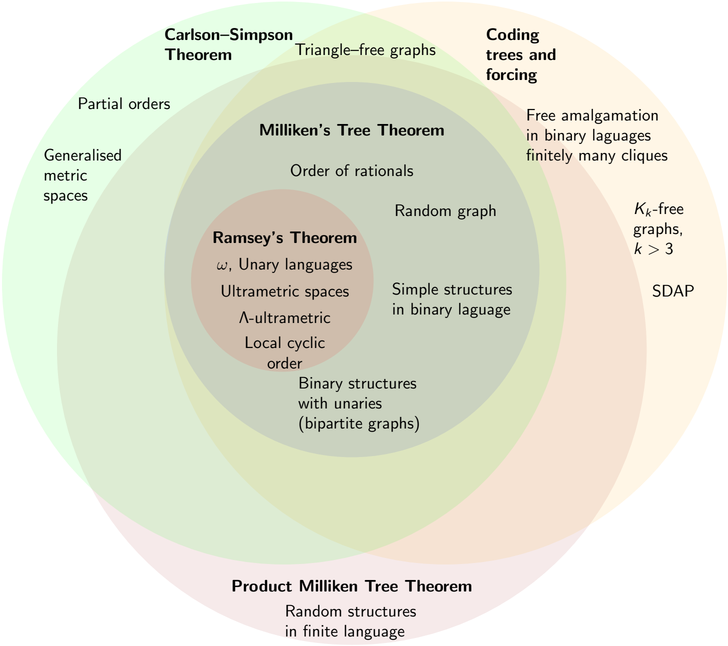
Big picture: proof techniques



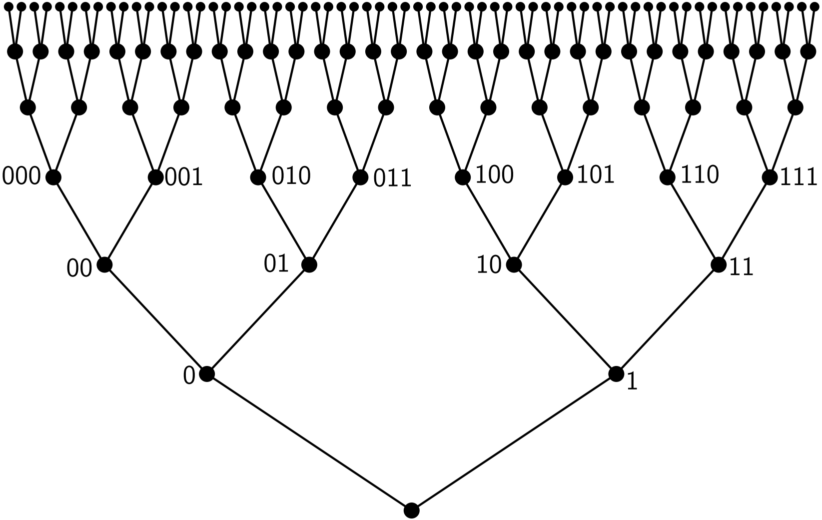
Big picture: proof techniques



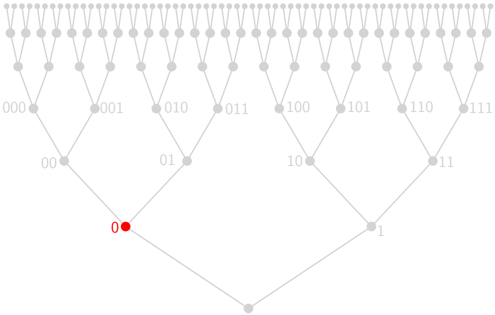
Big picture: proof techniques



Tree



Strong subtree

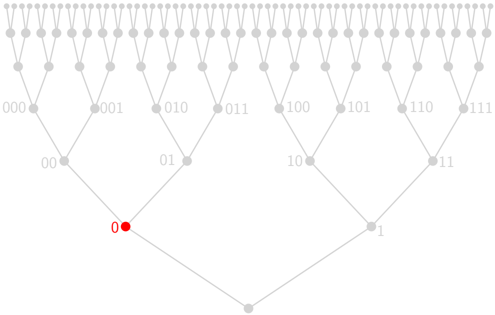


Definition

A subtree $\mathbf{S} \subseteq \mathbf{T}$ is a subset closed for meets. It is strong if:

1. S has a root node.

Strong subtree

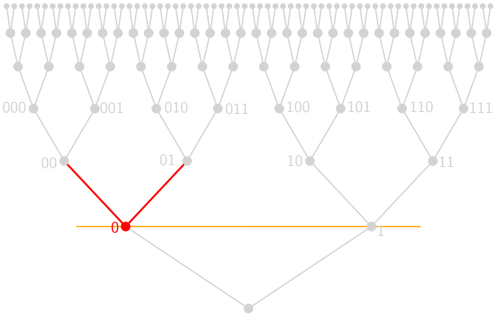


Definition

A subtree $\mathbf{S} \subseteq \mathbf{T}$ is a subset closed for meets. It is strong if:

1. S has a root node.
2. $\forall s \in S$ either s is leaf or $\forall t \in \mathrm{ImmSucc}_{\mathbf{T}}(s)$ the set $\mathrm{ImmSucc}_{\mathbf{S}}(s) \cap \mathrm{Succ}_{\mathbf{T}}(t)$ is a singleton.

Strong subtree

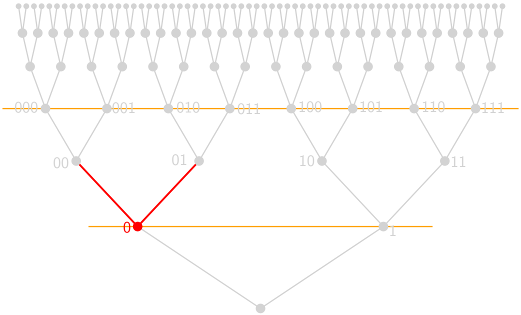


Definition

A subtree $\mathbf{S} \subseteq \mathbf{T}$ is a subset closed for meets. It is strong if:

1. S has a root node.
2. $\forall s \in S$ either s is leaf or $\forall t \in \mathrm{ImmSucc}_{\mathbf{T}}(s)$ the set $\mathrm{ImmSucc}_{\mathbf{S}}(s) \cap \mathrm{Succ}_{\mathbf{T}}(t)$ is a singleton.
3. Every level of S is a subset of some level of T.

Strong subtree

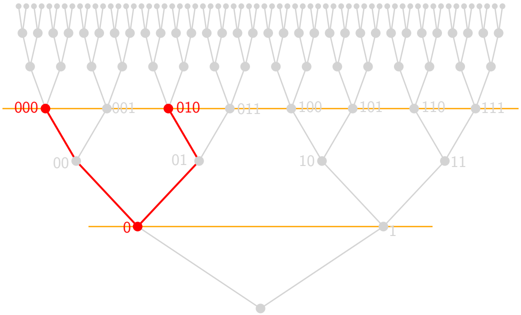


Definition

A subtree $\mathbf{S} \subseteq \mathbf{T}$ is a subset closed for meets. It is strong if:

1. S has a root node.
2. $\forall s \in S$ either s is leaf or $\forall t \in \mathrm{ImmSucc}_{\mathbf{T}}(s)$ the set $\mathrm{ImmSucc}_{\mathbf{S}}(s) \cap \mathrm{Succ}_{\mathbf{T}}(t)$ is a singleton.
3. Every level of S is a subset of some level of T.

Strong subtree

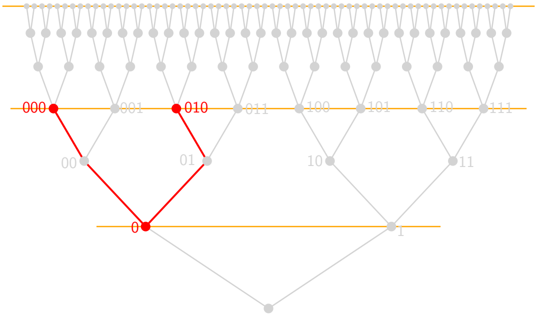


Definition

A subtree $\mathbf{S} \subseteq \mathbf{T}$ is a subset closed for meets. It is strong if:

1. S has a root node.
2. $\forall s \in S$ either s is leaf or $\forall t \in \mathrm{ImmSucc}_{\mathbf{T}}(s)$ the set $\mathrm{ImmSucc}_{\mathbf{S}}(s) \cap \mathrm{Succ}_{\mathbf{T}}(t)$ is a singleton.
3. Every level of S is a subset of some level of T.

Strong subtree

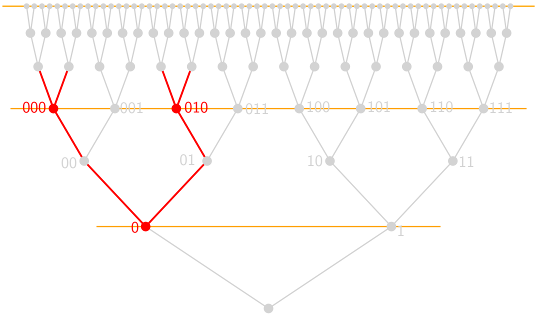


Definition

A subtree $\mathbf{S} \subseteq \mathbf{T}$ is a subset closed for meets. It is strong if:

1. S has a root node.
2. $\forall s \in S$ either s is leaf or $\forall t \in \mathrm{ImmSucc}_{\mathbf{T}}(s)$ the set $\mathrm{ImmSucc}_{\mathbf{S}}(s) \cap \mathrm{Succ}_{\mathbf{T}}(t)$ is a singleton.
3. Every level of S is a subset of some level of T.

Strong subtree

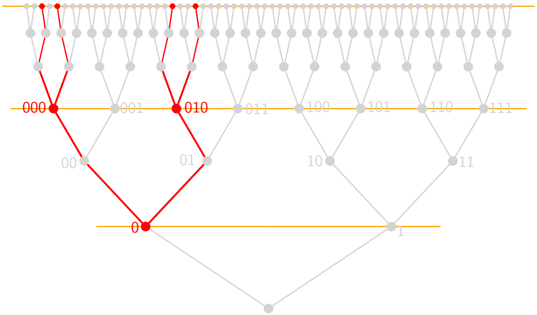


Definition

A subtree $\mathbf{S} \subseteq \mathbf{T}$ is a subset closed for meets. It is strong if:

1. S has a root node.
2. $\forall s \in S$ either s is leaf or $\forall t \in \mathrm{ImmSucc}_{\mathbf{T}}(s)$ the set $\mathrm{ImmSucc}_{\mathbf{S}}(s) \cap \mathrm{Succ}_{\mathbf{T}}(t)$ is a singleton.
3. Every level of S is a subset of some level of T.

Strong subtree

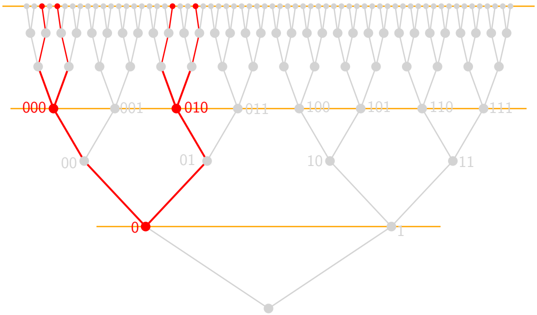


Definition

A subtree **S** $\subseteq$ **T** is a subset closed for meets. It is strong if:

1. S has a root node.
2. $\forall s \in S$ either s is leaf or $\forall t \in \mathrm{ImmSucc}_{\mathbf{T}}(s)$ the set $\mathrm{ImmSucc}_{\mathbf{S}}(s) \cap \mathrm{Succ}_{\mathbf{T}}(t)$ is a singleton.
3. Every level of S is a subset of some level of T.

Strong subtree



Definition

A subtree $\mathbf{S} \subseteq \mathbf{T}$ is a subset closed for meets. It is strong if:

1. S has a root node.
2. $\forall s \in S$ either s is leaf or $\forall t \in \mathrm{ImmSucc}_{\mathbf{T}}(s)$ the set $\mathrm{ImmSucc}_{\mathbf{S}}(s) \cap \mathrm{Succ}_{\mathbf{T}}(t)$ is a singleton.
3. Every level of S is a subset of some level of T.
4. S either has no leaves or all are at the same level.

Strong subtree

Theorem (Milliken tree theorem 1979)

For every infinitely branching tree $\mathbf{T}$ *with no leaves,* $k \geq 1$*, and every finite colouring of strong subtrees of depth* k *there exists infinite strong subtree* $\mathbf{S} \subseteq \mathbf{T}$ *which is monochromatic.*

Strong subtree

Theorem (Milliken tree theorem 1979)

For every infinitely branching tree $\mathbf{T}$ *with no leaves,* $k \geq 1$*, and every finite colouring of strong subtrees of depth* k *there exists infinite strong subtree* $\mathbf{S} \subseteq \mathbf{T}$ *which is monochromatic.*

Theorem (Laver, late 1969, publised in 1984)

$$\forall_{(O,\leq_O)\in\mathcal{O}} \exists_{T=T(|O|)\in\omega} \forall_{k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,T}.$$

Strong subtree

Theorem (Milliken tree theorem 1979)

For every infinitely branching tree $\mathbf{T}$ *with no leaves,* $k \geq 1$*, and every finite colouring of strong subtrees of depth* k *there exists infinite strong subtree* $\mathbf{S} \subseteq \mathbf{T}$ *which is monochromatic.*

Theorem (Laver, late 1969, publised in 1984)

$$\forall_{(O,\leq_O)\in\mathcal{O}} \exists_{T=T(|O|)\in\omega} \forall_{k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,T}.$$

Proof of Lavers's theorem.

1. Consider binary tree $\mathbf{B} = (B, \sqsubseteq)$. Define order $\leq$ lexicographically.

Strong subtree

Theorem (Milliken tree theorem 1979)

For every infinitely branching tree $\mathbf{T}$ *with no leaves,* $k \geq 1$*, and every finite colouring of strong subtrees of depth* k *there exists infinite strong subtree* $\mathbf{S} \subseteq \mathbf{T}$ *which is monochromatic.*

Theorem (Laver, late 1969, publised in 1984)

$$\forall_{(O,\leq_O)\in\mathcal{O}}\exists_{T=T(|O|)\in\omega}\forall_{k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,T}.$$

Proof of Lavers's theorem.

1. Consider binary tree $\mathbf{B} = (B, \sqsubseteq)$. Define order $\leq$ lexicographically.
2. Fix $\mathbf{O}$ and consider finite colouring of $\binom{\mathbf{B}}{\mathbf{O}}$

Strong subtree

Theorem (Milliken tree theorem 1979)

For every infinitely branching tree **T** *with no leaves,* $k \geq 1$*, and every finite colouring of strong subtrees of depth* k *there exists infinite strong subtree* $\mathbf{S} \subseteq \mathbf{T}$ *which is monochromatic.*

Theorem (Laver, late 1969, publised in 1984)

$$\forall_{(O,\leq_O)\in\mathcal{O}}\exists_{T=T(|O|)\in\omega}\forall_{k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,T}.$$

Proof of Lavers's theorem.

1. Consider binary tree $\mathbf{B} = (B, \sqsubseteq)$. Define order $\leq$ lexicographically.
2. Fix **O** and consider finite colouring of $\binom{\mathbf{B}}{\mathbf{O}}$
3. Every copy of **O** is contained in a strong subtree of bounded height (an envelope). This implies colouring of strong subtrees

Strong subtree

Theorem (Milliken tree theorem 1979)

For every infinitely branching tree **T** *with no leaves,* $k \geq 1$*, and every finite colouring of strong subtrees of depth* k *there exists infinite strong subtree* $\mathbf{S} \subseteq \mathbf{T}$ *which is monochromatic.*

Theorem (Laver, late 1969, publised in 1984)

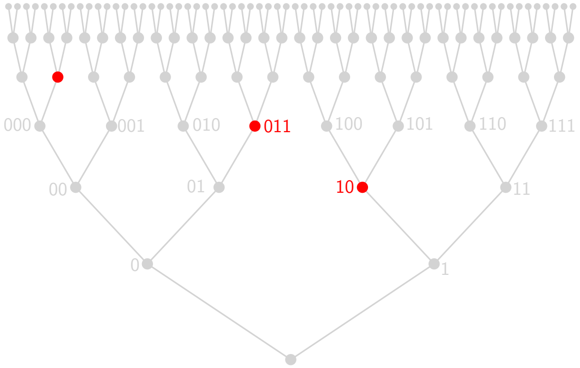
$$\forall_{(O,\leq_O)\in\mathcal{O}}\exists_{T=T(|O|)\in\omega}\forall_{k\geq 1} : (\mathbb{Q},\leq) \longrightarrow (\mathbb{Q},\leq)^{(O,\leq_O)}_{k,T}.$$

Proof of Lavers's theorem.

1. Consider binary tree $\mathbf{B} = (B, \sqsubseteq)$. Define order $\leq$ lexicographically.
2. Fix **O** and consider finite colouring of $\binom{\mathbf{B}}{\mathbf{O}}$
3. Every copy of **O** is contained in a strong subtree of bounded height (an envelope). This implies colouring of strong subtrees
4. By Milliken tree theorem there is infinite monochromatic subtree of **B**. This subtree ordered by $\sqsubseteq$ contains a copy of **Q**. Colour of a copy of **O** depends only on a shape within envelope, the embedding type.

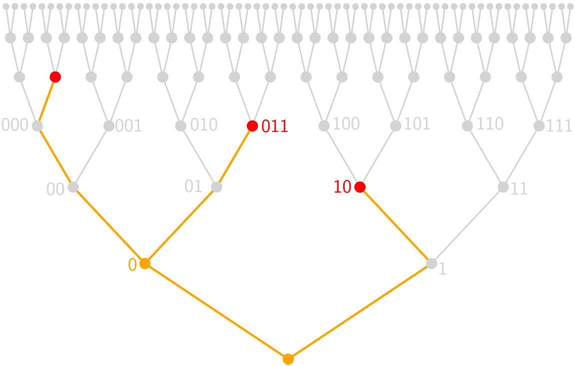
□

Envelopes of subsets



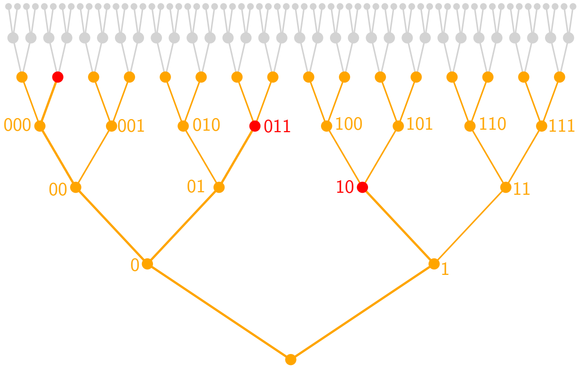
For every subset a strong tree envelope needs to be constructed

Envelopes of subsets



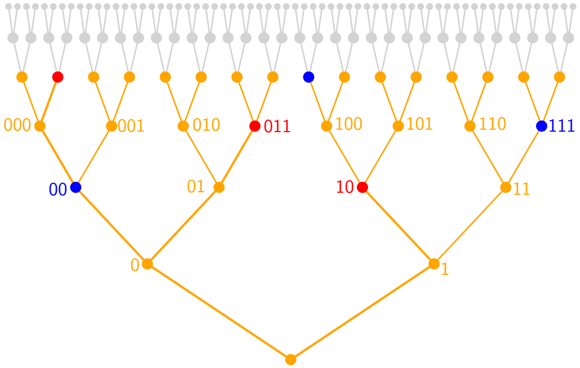
For every subset a strong tree envelope needs to be constructed

Envelopes of subsets



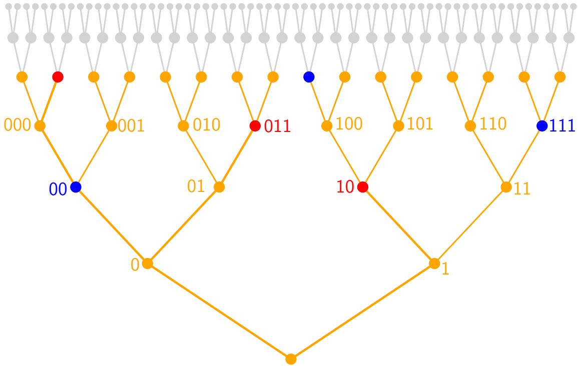
For every subset a strong tree envelope needs to be constructed

Envelopes of subsets



For every subset a strong tree envelope needs to be constructed
Many subsets leads to a given strong subtree.

Envelopes of subsets



For every subset a strong tree envelope needs to be constructed
Many subsets leads to a given strong subtree.

Main observation

Number of sets yielding a given envelope is bounded. Based on finite coloring of subsets of a given size k it is possible to produce finite coloring of envelopes and apply Milliken tree theorem to find a copy with bounded number of colors

Big Ramsey degrees of $(P, \leq)$

Let $\mathcal{P}$ be the class of all finite partial orders.

Theorem (J. H. 2020+)

Every (countable) universal partial order $(P, \leq)$ has finite big Ramsey degrees:

$$\forall_{(O,\leq)\in\mathcal{P}}\exists_{T=T(|O|)\in\omega}\forall_{k\geq 1} : (P, \leq) \longrightarrow (P, \leq)^{(O,\leq)}_{k,T}.$$

Universality: every countable partial order has embedding to $(P, \leq)$.

Big Ramsey degrees of $(P, \leq)$

Let $\mathcal{P}$ be the class of all finite partial orders.

Theorem (J. H. 2020+)

Every (countable) universal partial order $(P, \leq)$ has finite big Ramsey degrees:

$$\forall_{(O,\leq)\in\mathcal{P}} \exists_{T=T(|O|)\in\omega} \forall_{k\geq 1} : (P,\leq) \longrightarrow (P,\leq)^{(O,\leq)}_{k,T}.$$

Universality: every countable partial order has embedding to $(P, \leq)$.

- Proof is based on a new connection between big Ramsey degrees and the Carlson–Simpson theorem.

Big Ramsey degrees of $(P, \leq)$

Let $\mathcal{P}$ be the class of all finite partial orders.

Theorem (J. H. 2020+)

Every (countable) universal partial order $(P, \leq)$ has finite big Ramsey degrees:

$$\forall_{(O,\leq)\in\mathcal{P}} \exists_{T=T(|O|)\in\omega} \forall_{k\geq 1} : (P, \leq) \longrightarrow (P, \leq)^{(O,\leq)}_{k,T}.$$

Universality: every countable partial order has embedding to $(P, \leq)$.

- Proof is based on a new connection between big Ramsey degrees and the Carlson–Simpson theorem.
- We also get easy proof finite big Ramsey degrees of triangle-free graphs giving a new proof Dobrinen's theorem.

Big Ramsey degrees of $(P, \leq)$

Let $\mathcal{P}$ be the class of all finite partial orders.

Theorem (J. H. 2020+)

Every (countable) universal partial order $(P, \leq)$ has finite big Ramsey degrees:

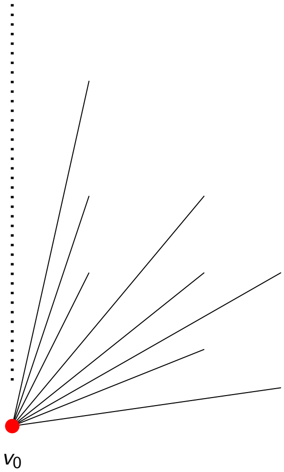
$$\forall_{(O,\leq)\in\mathcal{P}} \exists_{T=T(|O|)\in\omega} \forall_{k\geq 1} : (P, \leq) \longrightarrow (P, \leq)^{(O,\leq)}_{k,T}.$$

Universality: every countable partial order has embedding to $(P, \leq)$.

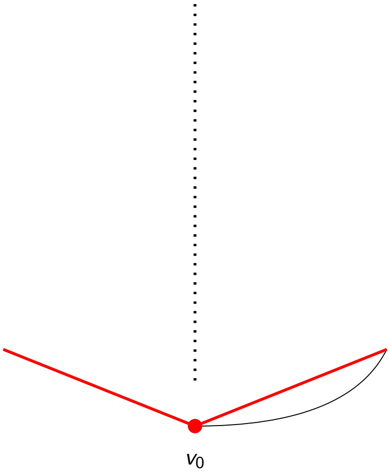
- Proof is based on a new connection between big Ramsey degrees and the Carlson–Simpson theorem.
- We also get easy proof finite big Ramsey degrees of triangle-free graphs giving a new proof Dobrinen's theorem.
- Construction generalizes to large family of classes that are described by forbidden induced cycles in particular to metric spaces with bounded number of distances (joint work with M. Balko, D. Chodounský, N. Dobrinen, M. Konečný, J. Nešetřil, L. Vena, A. Zucker)
- Big Ramsey degrees of posets and triangle-free graphs was fully characterised (joint work with M. Balko, D. Chodounský, N. Dobrinen, M. Konečný, L. Vena, A. Zucker
- Generalizations to bigger forbidden cliques and higher arities is work in progress

Tree of types of a universal triangle-free graph

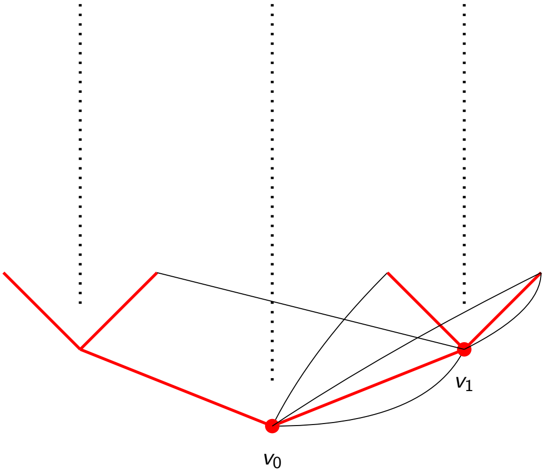
Tree of types of a universal triangle-free graph



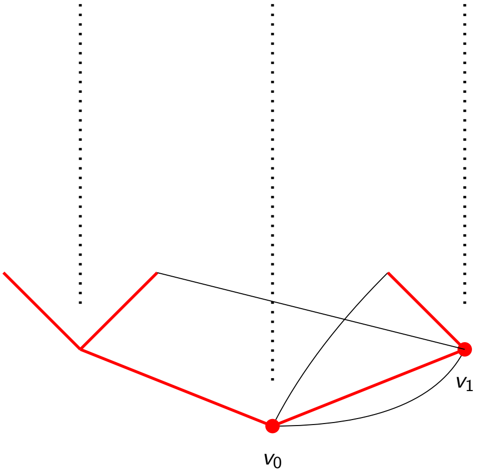
Tree of types of a universal triangle-free graph



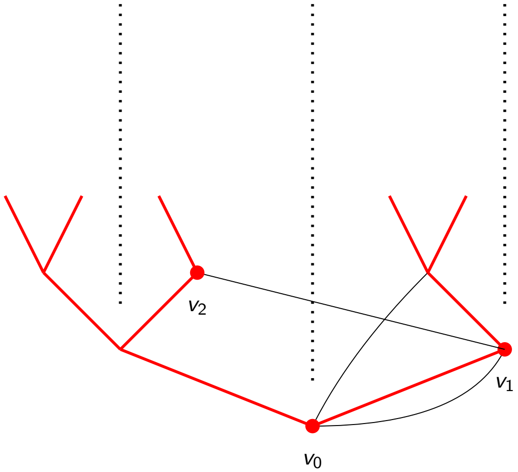
Tree of types of a universal triangle-free graph



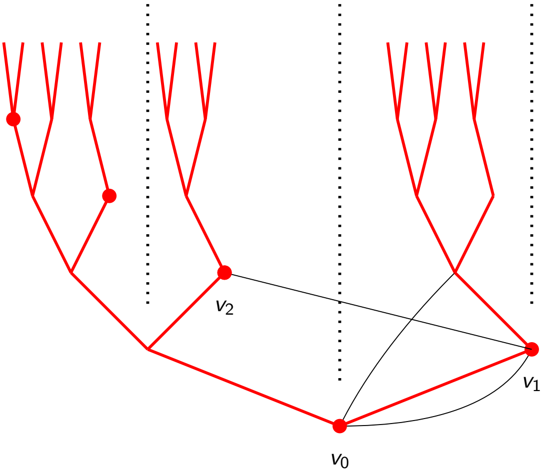
Tree of types of a universal triangle-free graph



Tree of types of a universal triangle-free graph

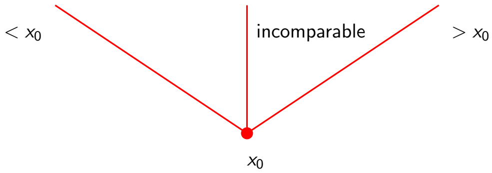


Tree of types of a universal triangle-free graph

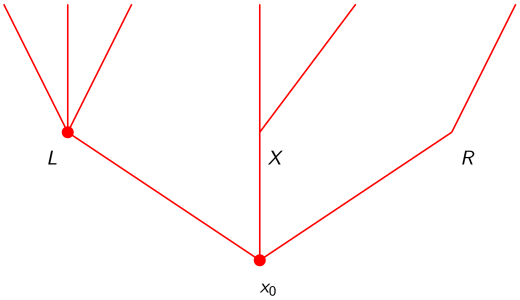


Tree of types of $(P, \leq)$

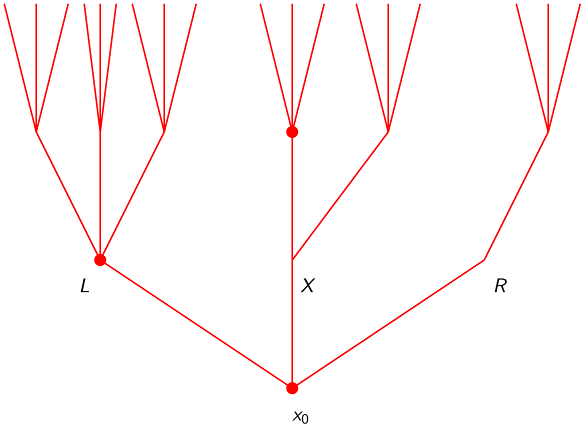
Tree of types of $(P, \leq)$



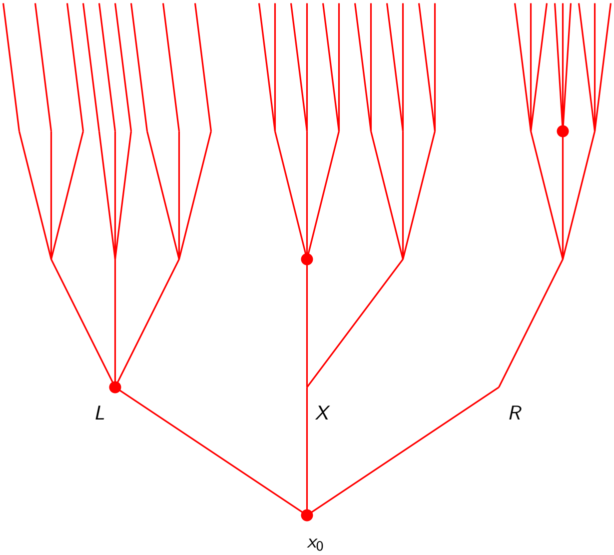
Tree of types of $(P, \leq)$



Tree of types of $(P, \leq)$



Tree of types of $(P, \leq)$



Parameter words

Definition (Parameter word)

Given a finite alphabet Σ and $k \in \omega + 1$, a *k-parameter word* is a (possibly infinite) word W in alphabet $\Sigma \cup \{\lambda_i : 0 \leq i < k\}$ such that $\forall i \in k$ word W contains λ_i and for every $j \in k - 1$, the first occurrence of λ_{j+1} appears after the first occurrence of λ_j.

Example (2-parameter word)

$\Sigma = \{\mathrm{L}, \mathrm{X}, \mathrm{R}\}$.

$$\mathrm{LRL}\lambda_0\lambda_0\mathrm{X}\lambda_1\lambda_0\mathrm{R}$$

Parameter words

Definition (Parameter word)

Given a finite alphabet Σ and $k \in \omega + 1$, a k-parameter word is a (possibly infinite) word W in alphabet $\Sigma \cup \{\lambda_i : 0 \leq i < k\}$ such that $\forall i \in k$ word W contains λ_i and for every $j \in k - 1$, the first occurrence of λ_{j+1} appears after the first occurrence of λ_j.

Example (2-parameter word)

$\Sigma = \{\mathrm{L}, \mathrm{X}, \mathrm{R}\}$.

$$\mathrm{LRL}\lambda_0\lambda_0\mathrm{X}\lambda_1\lambda_0\mathrm{R}$$

Definition (Substitution)

$$\mathrm{LRL}\lambda_0\lambda_0 X\lambda_1\lambda_0\mathrm{R}(\mathrm{LR}) \quad = \quad \mathrm{LRLLLXRLR}$$

Parameter words

Definition (Parameter word)

Given a finite alphabet Σ and $k \in \omega + 1$, a k-parameter word is a (possibly infinite) word W in alphabet $\Sigma \cup \{\lambda_i : 0 \leq i < k\}$ such that $\forall i \in k$ word W contains λ_i and for every $j \in k - 1$, the first occurrence of λ_{j+1} appears after the first occurrence of λ_j.

Example (2-parameter word)

$\Sigma = \{\mathrm{L}, \mathrm{X}, \mathrm{R}\}$.

$$\mathrm{LRL}\lambda_0\lambda_0\mathrm{X}\lambda_1\lambda_0\mathrm{R}$$

Definition (Substitution)

$$\begin{aligned} \mathrm{LRL}\lambda_0\lambda_0 X\lambda_1\lambda_0\mathrm{R}(\mathrm{LR}) &= \mathrm{LRLLLXRLR} \\ \mathrm{LRL}\lambda_0\lambda_0 X\lambda_1\lambda_0\mathrm{R}(\mathrm{X}) &= \mathrm{LRLXX}X \end{aligned}$$

Parameter words

Definition (Parameter word)

Given a finite alphabet Σ and $k \in \omega + 1$, a k-parameter word is a (possibly infinite) word W in alphabet $\Sigma \cup \{\lambda_i : 0 \leq i < k\}$ such that $\forall i \in k$ word W contains λ_i and for every $j \in k - 1$, the first occurrence of λ_{j+1} appears after the first occurrence of λ_j.

Example (2-parameter word)

$\Sigma = \{\mathrm{L}, \mathrm{X}, \mathrm{R}\}$.

$$\mathrm{LRL}\lambda_0\lambda_0\mathrm{X}\lambda_1\lambda_0\mathrm{R}$$

Definition (Substitution)

$$\begin{aligned} \mathrm{LRL}\lambda_0\lambda_0 X\lambda_1\lambda_0\mathrm{R}(\mathrm{LR}) &= \mathrm{LRLLLXRLR} \\ \mathrm{LRL}\lambda_0\lambda_0 X\lambda_1\lambda_0\mathrm{R}(\mathrm{X}) &= \mathrm{LRLXX}X \end{aligned}$$

For set S of parameter words and a parameter word W:

$$W(S) = \{W(U) : U \in S\}.$$

Ramsey theorem for parameter words

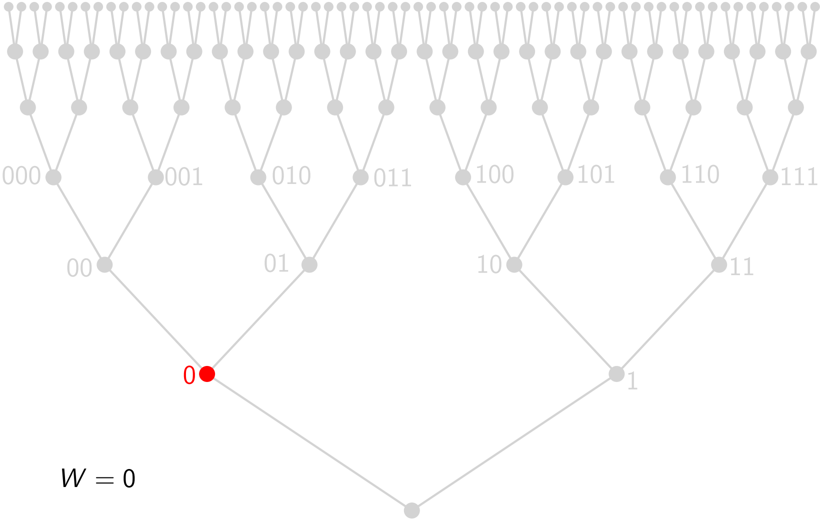
The following infinitary version of Graham–Rothschild Theorem is a direct consequence of the Carlson–Simpson theorem. It was also independently proved by Voight in 1983 (apparently unpublished):

Theorem (Ramsey theorem for parameter words)

Let Σ be a finite alphabet and $k \geq 0$ a finite integer. If the set of all finite k-parameter words in alphabet Σ is coloured by finitely many colours, then there exists a monochromatic infinite-parameter word W.

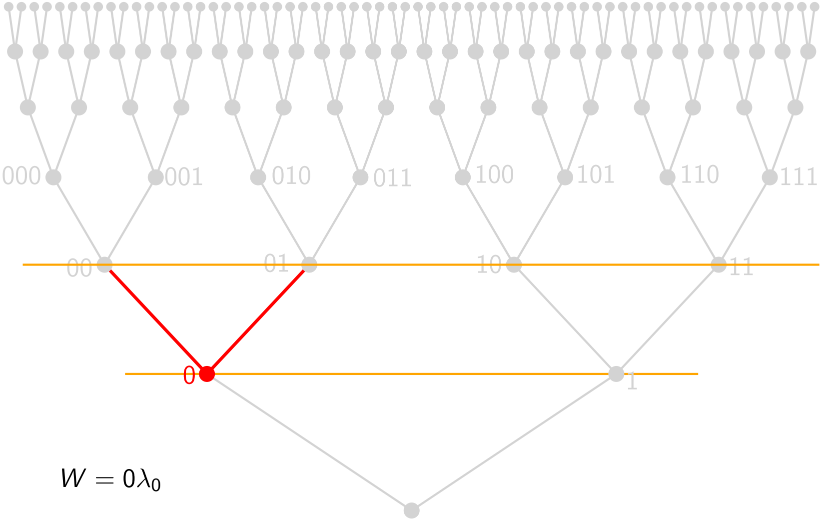
By W being monochromatic we mean that for every pair of k-parameter words U, V the colour of $W(U)$ is the same as colour of $W(V)$.

Parameter words as subtrees



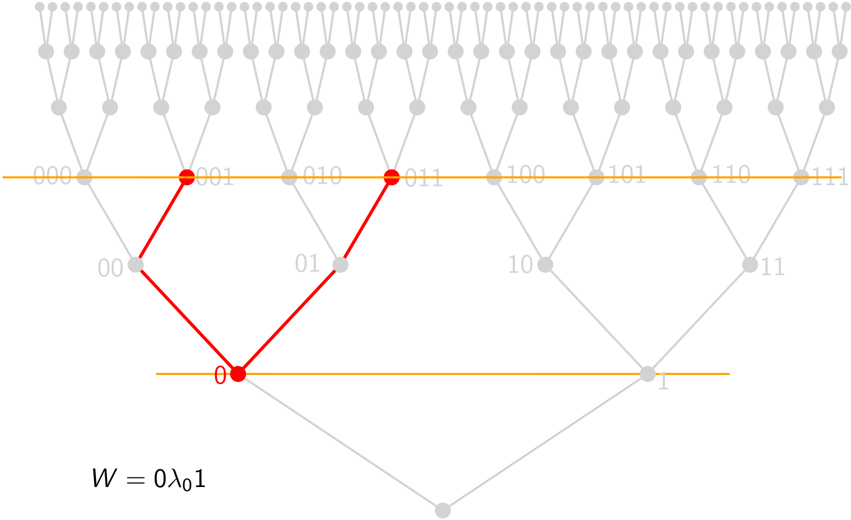
$W = 0$

Parameter words as subtrees



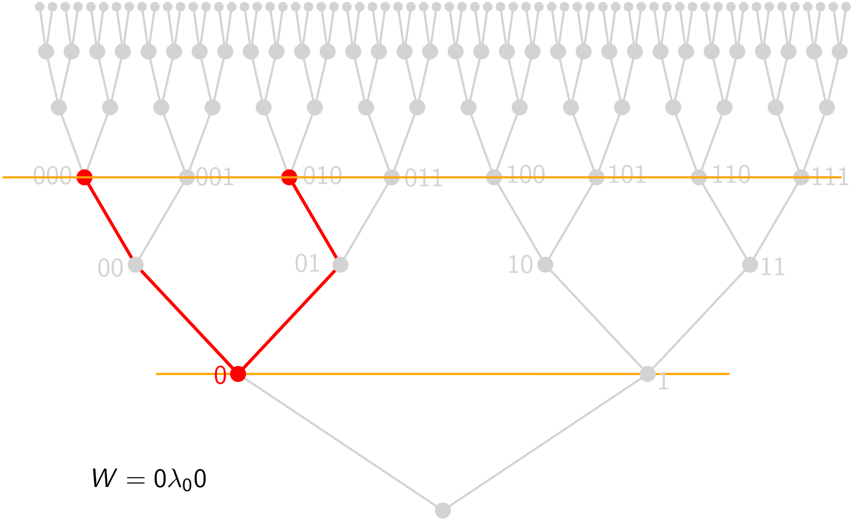
$W = 0\lambda_0$

Parameter words as subtrees



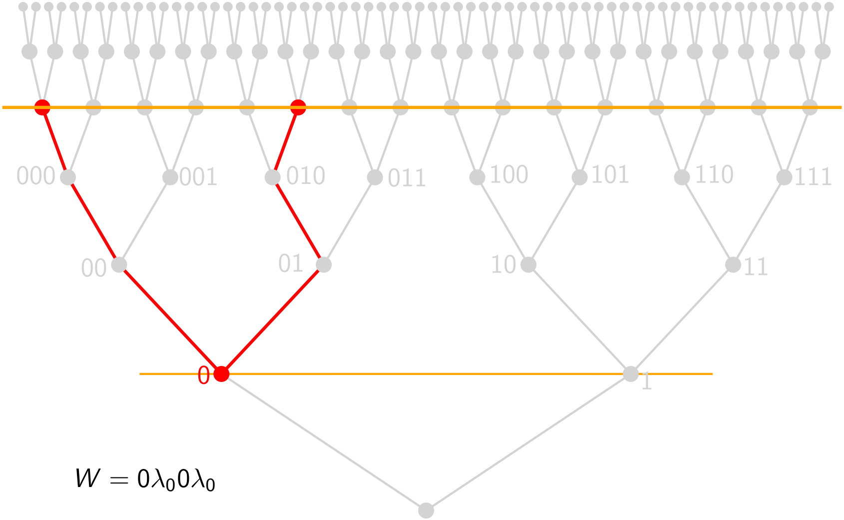
$W = 0\lambda_0 1$

Parameter words as subtrees



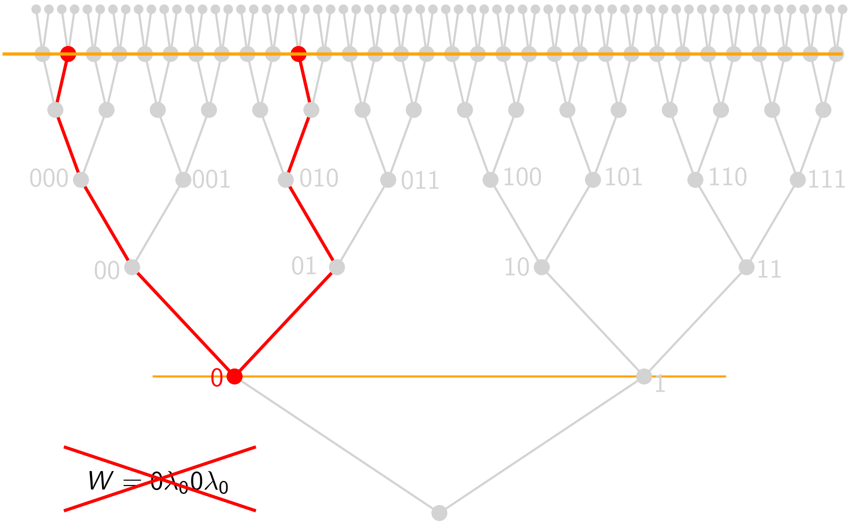
$W = 0\lambda_0 0$

Parameter words as subtrees

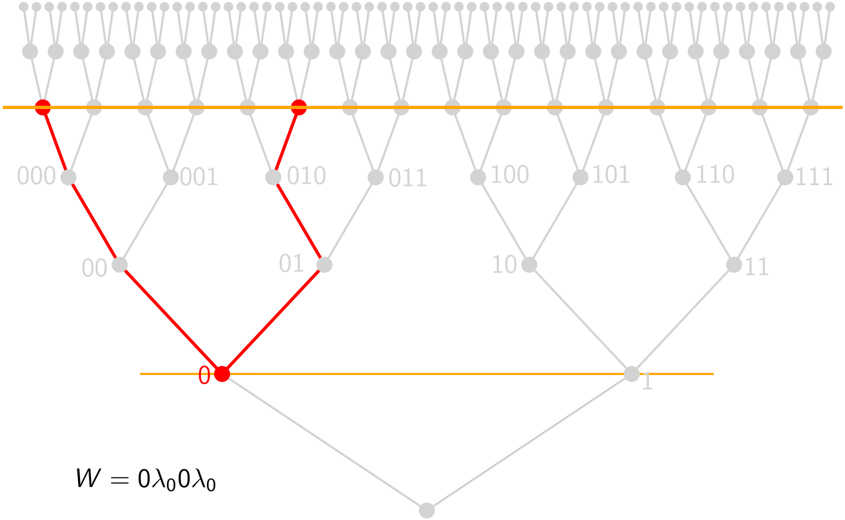


$W = 0\lambda_0 0\lambda_0$

Parameter words as subtrees

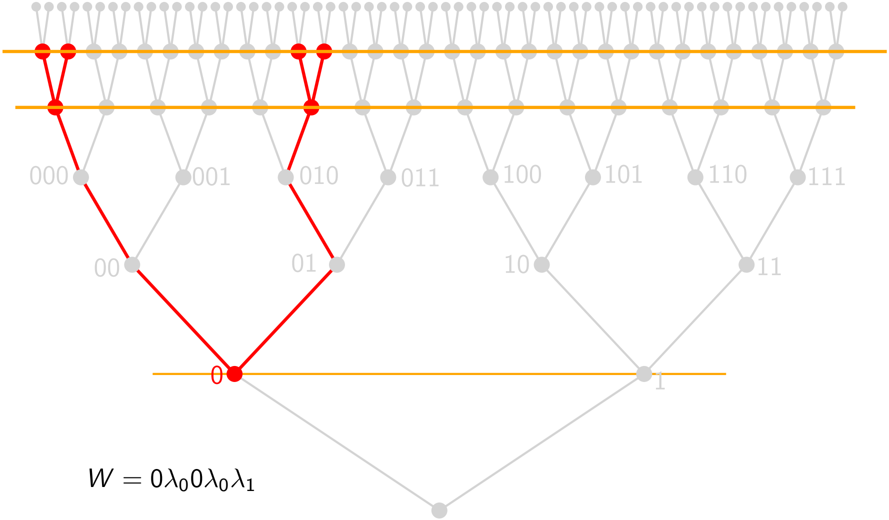


Parameter words as subtrees



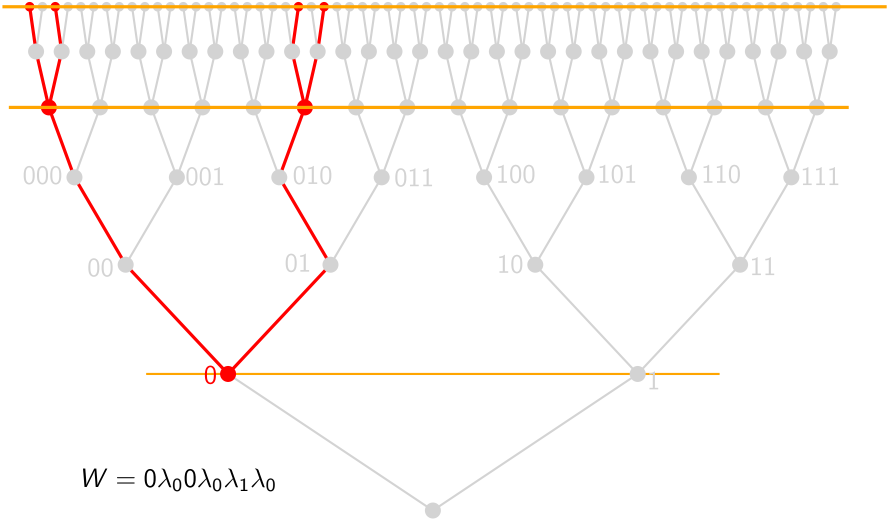
$W = 0\lambda_0 0\lambda_0$

Parameter words as subtrees



$W = 0\lambda_0 0\lambda_0\lambda_1$

Parameter words as subtrees



$W = 0\lambda_0 0\lambda_0\lambda_1\lambda_0$

Definition

Given a finite alphabet Σ, a finite integer $k \geq 0$ and a finite set k-parameter words, an envelope of S is every n-parameter word W (for some $n \geq k$) such that

$$\forall_{w \in S} \exists_u W(u) = w.$$

Definition

Given a finite alphabet Σ, a finite integer $k \geq 0$ and a finite set k-parameter words, an envelope of S is every n-parameter word W (for some $n \geq k$) such that

$$\forall_{w \in S} \exists_u W(u) = w.$$

Example

Envelopes of $\{0, 000\}$ are: $0\lambda_0\lambda_0$, $0\lambda_0 0$, $0\lambda_0 0\lambda_1$, $\ldots$.

Definition

Given a finite alphabet Σ, a finite integer $k \geq 0$ and a finite set k-parameter words, an envelope of S is every n-parameter word W (for some $n \geq k$) such that

$$\forall_{w \in S} \exists_u W(u) = w.$$

Example

Envelopes of $\{0, 000\}$ are: $0\lambda_0\lambda_0$, $0\lambda_0 0$, $0\lambda_0 0\lambda_1$, $\ldots$.

Proposition (Envelopes are bounded)

Let Σ be a finite alphabet, and $k, s \geq 0$ be a finite integers. Then there exists $T(|\Sigma|, s, k)$ such that for every set S of size s of k-parameter words in alphabet Σ there exists an envelope of S with at most $T(|\Sigma|, s, k)$ parameters.

Proof.

$$\begin{array}{lccccccc} U & = & 0 & 1 & 1 & 0 & 1 & \\ V & = & 0 & 0 & 1 & 1 & 0 & 1 \end{array}$$

Envelope:

Definition

Given a finite alphabet Σ, a finite integer $k \geq 0$ and a finite set k-parameter words, an envelope of S is every n-parameter word W (for some $n \geq k$) such that

$$\forall_{w \in S} \exists_u W(u) = w.$$

Example

Envelopes of $\{0, 000\}$ are: $0\lambda_0\lambda_0$, $0\lambda_0 0$, $0\lambda_0 0\lambda_1$, $\ldots$.

Proposition (Envelopes are bounded)

Let Σ be a finite alphabet, and $k, s \geq 0$ be a finite integers. Then there exists $T(|\Sigma|, s, k)$ such that for every set S of size s of k-parameter words in alphabet Σ there exists an envelope of S with at most $T(|\Sigma|, s, k)$ parameters.

Proof.

U	$=$	0	1	1	0	1	
V	$=$	0	0	1	1	0	1
Envelope:		0					

Definition

Given a finite alphabet Σ, a finite integer $k \geq 0$ and a finite set k-parameter words, an envelope of S is every n-parameter word W (for some $n \geq k$) such that

$$\forall_{w \in S} \exists_u W(u) = w.$$

Example

Envelopes of $\{0, 000\}$ are: $0\lambda_0\lambda_0$, $0\lambda_0 0$, $0\lambda_0 0\lambda_1$, $\dots$.

Proposition (Envelopes are bounded)

Let Σ be a finite alphabet, and $k, s \geq 0$ be a finite integers. Then there exists $T(|\Sigma|, s, k)$ such that for every set S of size s of k-parameter words in alphabet Σ there exists an envelope of S with at most $T(|\Sigma|, s, k)$ parameters.

Proof.

$$\begin{array}{rcllllll} U & = & 0 & 1 & 1 & 0 & 1 & \\ V & = & 0 & 0 & 1 & 1 & 0 & 1 \\ \text{Envelope:} & & 0 & \lambda_0 & & & & \end{array}$$

Definition

Given a finite alphabet Σ, a finite integer $k \geq 0$ and a finite set k-parameter words, an envelope of S is every n-parameter word W (for some $n \geq k$) such that

$$\forall_{w \in S} \exists_u W(u) = w.$$

Example

Envelopes of $\{0, 000\}$ are: $0\lambda_0\lambda_0$, $0\lambda_0 0$, $0\lambda_0 0\lambda_1$, $\ldots$.

Proposition (Envelopes are bounded)

Let Σ be a finite alphabet, and $k, s \geq 0$ be a finite integers. Then there exists $T(|\Sigma|, s, k)$ such that for every set S of size s of k-parameter words in alphabet Σ there exists an envelope of S with at most $T(|\Sigma|, s, k)$ parameters.

Proof.

U	$=$	0	1	1	0	1	
V	$=$	0	0	1	1	0	1
Envelope:		0	λ_0	1			

Definition

Given a finite alphabet Σ, a finite integer $k \geq 0$ and a finite set k-parameter words, an envelope of S is every n-parameter word W (for some $n \geq k$) such that

$$\forall_{w \in S} \exists_u W(u) = w.$$

Example

Envelopes of $\{0, 000\}$ are: $0\lambda_0\lambda_0$, $0\lambda_0 0$, $0\lambda_0 0\lambda_1$, $\ldots$.

Proposition (Envelopes are bounded)

Let Σ be a finite alphabet, and $k, s \geq 0$ be a finite integers. Then there exists $T(|\Sigma|, s, k)$ such that for every set S of size s of k-parameter words in alphabet Σ there exists an envelope of S with at most $T(|\Sigma|, s, k)$ parameters.

Proof.

U	$=$	0	1	1	0	1	
V	$=$	0	0	1	1	0	1
Envelope:		0	λ_0	1	λ_1		

Definition

Given a finite alphabet Σ, a finite integer $k \geq 0$ and a finite set k-parameter words, an **envelope** of S is every n-parameter word W (for some $n \geq k$) such that

$$\forall_{w \in S} \exists_u W(u) = w.$$

Example

Envelopes of $\{0, 000\}$ are: $0\lambda_0\lambda_0$, $0\lambda_0 0$, $0\lambda_0 0\lambda_1$, ...

Proposition (Envelopes are bounded)

Let Σ be a finite alphabet, and $k, s \geq 0$ be a finite integers. Then there exists $T(|\Sigma|, s, k)$ such that for every set S of size s of k-parameter words in alphabet Σ there exists an envelope of S with at most $T(|\Sigma|, s, k)$ parameters.

Proof.

U	$=$	0	1	1	0	1	
V	$=$	0	0	1	1	0	1
Envelope:		0	λ_0	1	λ_1	λ_0	

Definition

Given a finite alphabet Σ, a finite integer $k \geq 0$ and a finite set k-parameter words, an envelope of S is every n-parameter word W (for some $n \geq k$) such that

$$\forall_{w \in S} \exists_u W(u) = w.$$

Example

Envelopes of $\{0, 000\}$ are: $0\lambda_0\lambda_0,\quad 0\lambda_0 0,\quad 0\lambda_0 0\lambda_1,\quad \ldots.$

Proposition (Envelopes are bounded)

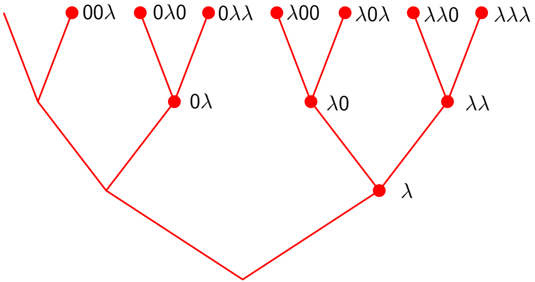
Let Σ be a finite alphabet, and $k, s \geq 0$ be a finite integers. Then there exists $T(|\Sigma|, s, k)$ such that for every set S of size s of k-parameter words in alphabet Σ there exists an envelope of S with at most $T(|\Sigma|, s, k)$ parameters.

Proof.

$$\begin{array}{lcllllll} U & = & 0 & 1 & 1 & 0 & 1 & \\ V & = & 0 & 0 & 1 & 1 & 0 & 1 \\ \text{Envelope:} & & 0 & \lambda_0 & 1 & \lambda_1 & \lambda_0 & \lambda_2 \end{array}$$

□

Triangle-free graph on 1-parameter words

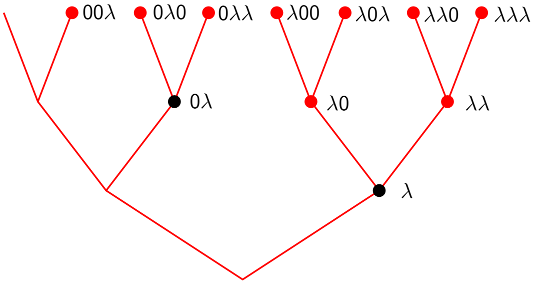


Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.

Triangle-free graph on 1-parameter words

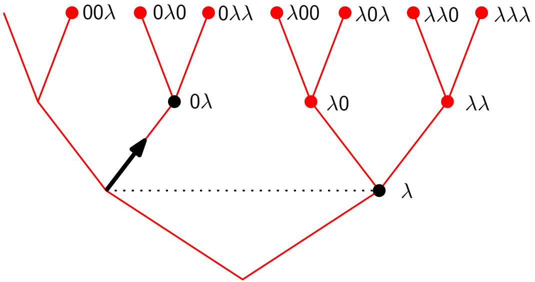


Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $v_{|u|} = \lambda$

Triangle-free graph on 1-parameter words

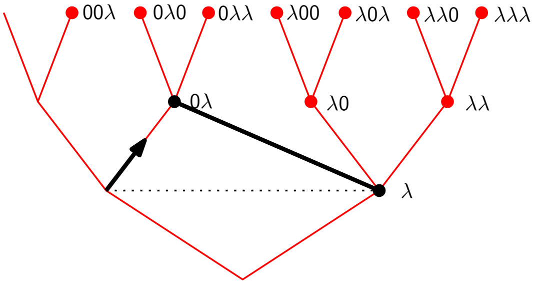


Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $V_{|U|} = \lambda$

Triangle-free graph on 1-parameter words

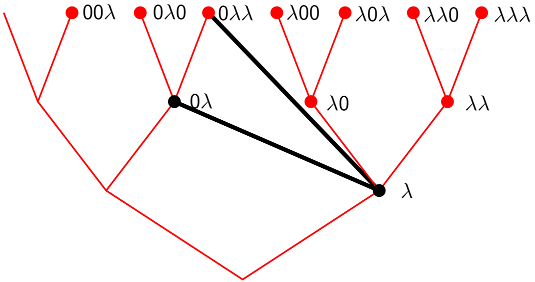


Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $v_{|u|} = \lambda$

Triangle-free graph on 1-parameter words

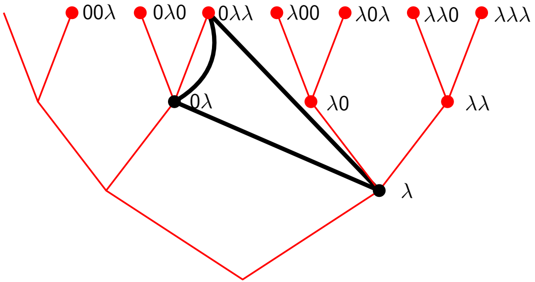


Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $v_{|u|} = \lambda$

Triangle-free graph on 1-parameter words

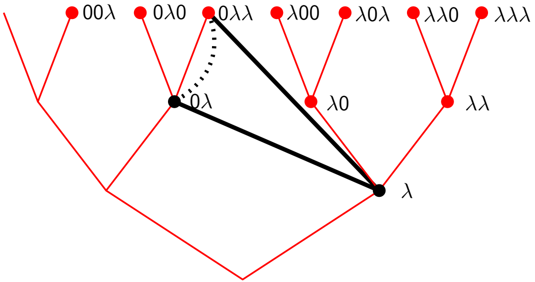


Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $V_{|U|} = \lambda$

Triangle-free graph on 1-parameter words

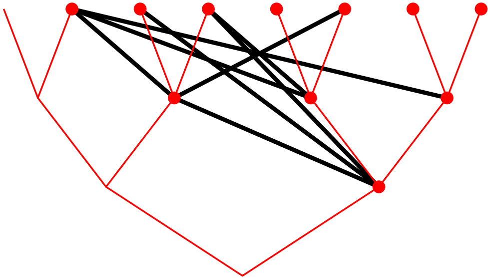


Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $V_{|U|} = \lambda$ and
  2. for no $i \in |u|$ it holds that $U_i = V_i = \lambda$.

Triangle-free graph on 1-parameter words

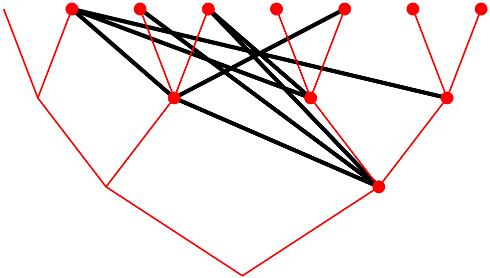


Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $V_{|U|} = \lambda$ and
  2. for no $i \in |u|$ it holds that $U_i = V_i = \lambda$.

Triangle-free graph on 1-parameter words



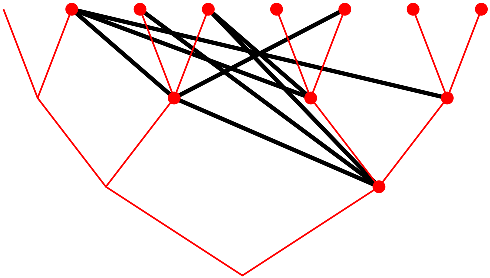
Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $V_{|U|} = \lambda$ and
  2. for no $i \in |u|$ it holds that $U_i = V_i = \lambda$.

Key observation 1: G is universal triangle-free graph.

Triangle-free graph on 1-parameter words



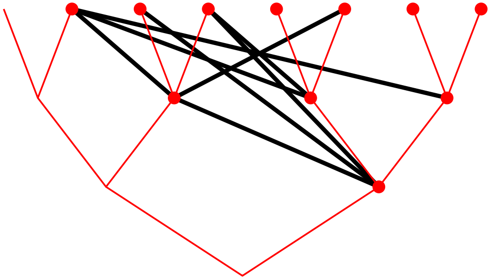
Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $V_{|U|} = \lambda$ and
  2. for no $i \in |u|$ it holds that $U_i = V_i = \lambda$.

Key observation 1: G is universal triangle-free graph.
Given any triangle-free graph H with vertex set ω assign every $i \in \omega$ word w of length i putting $\forall_{j<i}\ w_j = \lambda$ iff $\{i, j\}$ is an edge of H.

Triangle-free graph on 1-parameter words



Put $\Sigma = \{0\}$.

Definition (Triangle-free graph G)

- Vertices of G are all finite 1-parameter words in alphabet Σ.
- For 1-parameter words u, v satisfying $|u| < |v|$ we put $u \sim v$ (u adjacent to v) iff
  1. $V_{|U|} = \lambda$ and
  2. for no $i \in |u|$ it holds that $U_i = V_i = \lambda$.

Key observation 1: G is universal triangle-free graph.

Key observation 2: For every pair of 1-parmeter words U and V and every ω-parameter W

$$U \sim V \iff W(U) \sim W(V).$$

Observation

G is a universal triangle-free graph.

Observation

For every infinite-parameter word W it holds that $u \sim v \iff W(u) \sim W(v)$.
(Substitution is also graph embedding on $G \to G$.)

Theorem (Ramsey theorem for parameter words)

Let Σ be a finite alphabet and $k \geq 0$ a finite integer. If the set of all finite k-parameter words in alphabet Σ is coloured by finitely many colours, then there exists a monochromatic infinite-parameter word W.

Proposition (Envelopes are bounded)

There exists $T(|\Sigma|, s, k)$ such that for every set S of size s of k-parameter words in alphabet Σ there exists an envelope of S with at most $T(|\Sigma|, s, k)$ parameters.

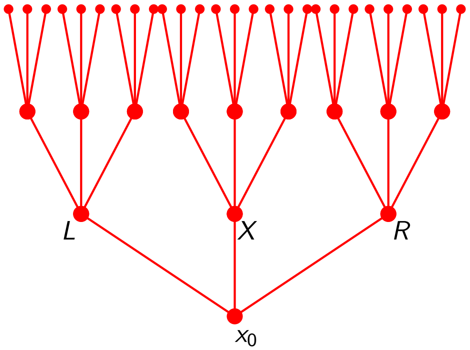
Theorem (Dobrinen 2020)

The big Ramsey degrees of universal triangle-free graph are finite.

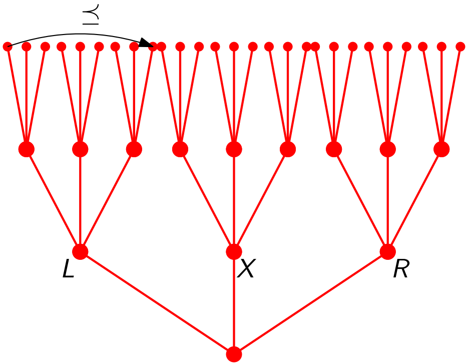
Proof.

Fix graph A and a finite coloring of $\binom{G}{A}$. Because envelopes of copies of A are bounded, apply the theorem above for every embedding type and obtain a copy of G with bounded number of colors. $\square$

Partial order on infinite ternary tree



Partial order on infinite ternary tree



Put $\Sigma = \{\mathrm{L}, \mathrm{X}, \mathrm{R}\}$ and order $\mathrm{L} <_{\mathrm{lex}} \mathrm{X} <_{\mathrm{lex}} \mathrm{R}$.

Definition (Partial order $(\Sigma^*, \preceq)$)

For $w, w' \in \Sigma^*$ we put $w \prec w'$ if and only if there exists $0 \leq i < \min(|w|, |w'|)$ such that

1. $(w_i, w'_i) = (\mathrm{L}, \mathrm{R})$ and
2. for every $0 \leq j < i$ it holds that $w_j \leq_{\mathrm{lex}} w'_j$.

Key observations: $\preceq$ is universal partial order and is stable for substitution.

Lower bounds

Devlin-type

Given n, the big Ramsey degree of linear order of size n is to the number of Devlin-types.

Notation:

- Σ^* is the set of all finite words in alphabet Σ.
- Given $S \subseteq \Sigma^*$ by $\overline{S}$ we denote the set of all initial segments of words in S.
- By $\overline{S}_i$ we denote the set of all initial segments of S of length i.
- By $w^\frown c$ we denote word w extended by character c (concatenation).
- $S^\frown c = \{w^\frown c : w \in S\}$.

Devlin-type

Given n, the big Ramsey degree of linear order of size n is to the number of Devlin-types.

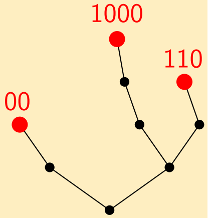
Notation:

- Σ^* is the set of all finite words in alphabet Σ.
- Given $S \subseteq \Sigma^*$ by $\overline{S}$ we denote the set of all initial segments of words in S.
- By $\overline{S}_i$ we denote the set of all initial segments of S of length i.
- By $w^\frown c$ we denote word w extended by character c (concatenation).
- $S^\frown c = \{w^\frown c : w \in S\}$.

Definition (Devlin-type)

A Devlin-type is any subset S of $\{0,1\}^*$ such that for every $\ell \leq \max_{w\in S} |w|$ precisely one of the following happens:

1. **Leaf:** There is $w \in \overline{S}_\ell$ such that $\overline{S}_{\ell+1} = (\overline{S}_\ell \setminus \{w\})^\frown 0$.
2. **Branching**: There is $w \in \overline{S}_\ell$ such that $\overline{S}_{\ell+1} = w^\frown 1 \cup (\overline{S}_\ell)^\frown 0$.



Devlin-type

Given n, the big Ramsey degree of linear order of size n is to the number of Devlin-types.

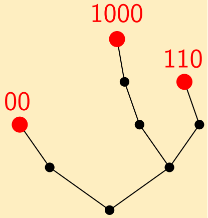
Notation:

- Σ^* is the set of all finite words in alphabet Σ.
- Given $S \subseteq \Sigma^*$ by $\overline{S}$ we denote the set of all initial segments of words in S.
- By $\overline{S}_i$ we denote the set of all initial segments of S of length i.
- By $w^\frown c$ we denote word w extended by character c (concatenation).
- $S^\frown c = \{w^\frown c : w \in S\}$.

Definition (Devlin-type)

A Devlin-type is any subset S of $\{0,1\}^*$ such that for every $\ell \leq \max_{w\in S} |w|$ precisely one of the following happens:

1. **Leaf:** There is $w \in \overline{S}_\ell$ such that $\overline{S}_{\ell+1} = (\overline{S}_\ell \setminus \{w\})^\frown 0$.
2. **Branching**: There is $w \in \overline{S}_\ell$ such that $\overline{S}_{\ell+1} = w^\frown 1 \cup (\overline{S}_\ell)^\frown 0$.



$$t_1 = 1, t_n = \sum_{\ell=1}^{n-1} \binom{2n-2}{2\ell-1} t_\ell \cdot t_{n-\ell}.$$

Level-structures

Definition

Given level ℓ of the tree of types, we can consider its level-structure:

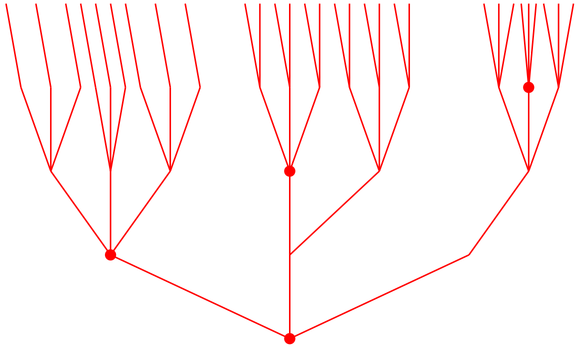
1. Vertices are nodes (types) of level ℓ.
2. We write $a \preceq b$ if it is true that $a' \leq b'$ for every successor a' of a and b' of b.
3. We write $a \trianglelefteq b$ if it is true that $a' \leq b'$ for some successor a' of a and b' of b.
4. We write $a \perp b$ if it is true that $a' \perp b'$ for every successor a' of a and b' of b.

Level-structures

Definition

Given level ℓ of the tree of types, we can consider its level-structure:

1. Vertices are nodes (types) of level ℓ.
2. We write $a \preceq b$ if it is true that $a' \leq b'$ for every successor a' of a and b' of b.
3. We write $a \trianglelefteq b$ if it is true that $a' \leq b'$ for some successor a' of a and b' of b.
4. We write $a \perp b$ if it is true that $a' \perp b'$ for every successor a' of a and b' of b.

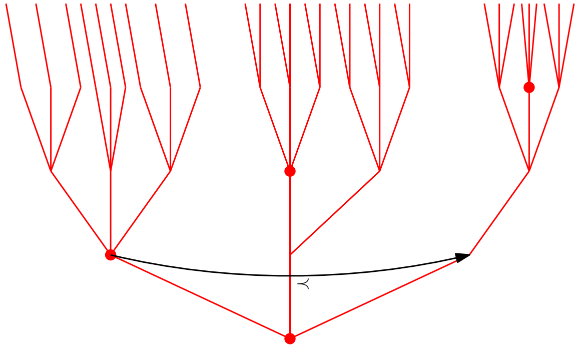


Level-structures

Definition

Given level ℓ of the tree of types, we can consider its level-structure:

1. Vertices are nodes (types) of level ℓ.
2. We write $a \preceq b$ if it is true that $a' \leq b'$ for every successor a' of a and b' of b.
3. We write $a \trianglelefteq b$ if it is true that $a' \leq b'$ for some successor a' of a and b' of b.
4. We write $a \perp b$ if it is true that $a' \perp b'$ for every successor a' of a and b' of b.

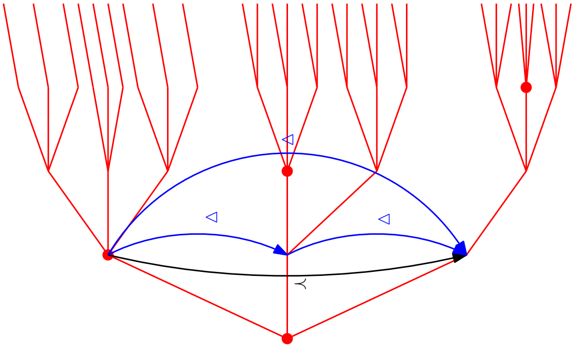


Level-structures

Definition

Given level ℓ of the tree of types, we can consider its level-structure:

1. Vertices are nodes (types) of level ℓ.
2. We write $a \preceq b$ if it is true that $a' \leq b'$ for every successor a' of a and b' of b.
3. We write $a \trianglelefteq b$ if it is true that $a' \leq b'$ for some successor a' of a and b' of b.
4. We write $a \perp b$ if it is true that $a' \perp b'$ for every successor a' of a and b' of b.

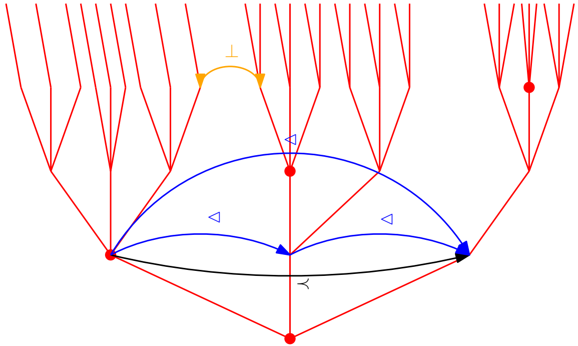


Level-structures

Definition

Given level ℓ of the tree of types, we can consider its level-structure:

1. Vertices are nodes (types) of level ℓ.
2. We write $a \preceq b$ if it is true that $a' \leq b'$ for every successor a' of a and b' of b.
3. We write $a \trianglelefteq b$ if it is true that $a' \leq b'$ for some successor a' of a and b' of b.
4. We write $a \perp b$ if it is true that $a' \perp b'$ for every successor a' of a and b' of b.



Level-structures

Definition

Given level ℓ of the tree of types, we can consider its level-structure:

1. Vertices are nodes (types) of level ℓ.
2. We write $a \preceq b$ if it is true that $a' \leq b'$ for every successor a' of a and b' of b.
3. We write $a \trianglelefteq b$ if it is true that $a' \leq b'$ for some successor a' of a and b' of b.
4. We write $a \perp b$ if it is true that $a' \perp b'$ for every successor a' of a and b' of b.

Fun fact

It turns out that both $\preceq$ and $\trianglelefteq$ are partial orders and whenever $a \preceq b$ also $a \trianglelefteq b$. One can think of the level structure $(A, \preceq, \trianglelefteq)$ as of an finite approximation of the infinite partial order located above the given level.

Definition (Poset-type)

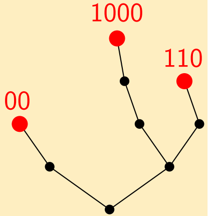
A set $S \subseteq \{\mathrm{L}, \mathrm{X}, \mathrm{R}\}^*$ is called a poset-type if precisely one of the following four conditions is satisfied for every level ℓ with $0 \leq \ell < \max_{w \in S} |w|$:

1. **Leaf:** There is $w \in \overline{S}_\ell$ related to every $u \in \overline{S}_\ell \setminus \{w\}$ and $\overline{S}_{\ell+1} = (\overline{S}_\ell \setminus \{w\})^\frown \mathrm{X}$.
2. **Branching:** There is $w \in \overline{S}_\ell$ such that
$$\overline{S}_{\ell+1} = \{z \in \overline{S}_\ell : z <_{\text{lex}} w\}^\frown \mathrm{X} \cup \{w^\frown \mathrm{X}, w^\frown \mathrm{R}\} \cup \{z \in \overline{S}_\ell : w <_{\text{lex}} z\}^\frown \mathrm{R}.$$
3. **New $\perp$:** There are unrelated words $v <_{\text{lex}} w \in \overline{S}_\ell$ such that
$$\begin{aligned}\overline{S}_{\ell+1} = &\{z \in \overline{S}_\ell : z <_{\text{lex}} v\}^\frown \mathrm{X} \cup \{v^\frown \mathrm{R}\} \cup \{z \in \overline{S}_\ell : v <_{\text{lex}} z <_{\text{lex}} w \text{ and } z \perp v\}^\frown \mathrm{X} \\ &\cup \{z \in \overline{S}_\ell : v <_{\text{lex}} z <_{\text{lex}} w \text{ and } z \not\perp v\}^\frown \mathrm{R} \cup \{w^\frown \mathrm{X}\} \cup \{z \in \overline{S}_\ell : w <_{\text{lex}} z\}^\frown \mathrm{R}.\end{aligned}$$
   Moreover for every $u \in \overline{S}_\ell$, $v <_{\text{lex}} u <_{\text{lex}} w$ implies that at least one of $u \perp v$ or $u \perp w$ holds.
4. **New $\prec$:** There are unrelated words $v <_{\text{lex}} w \in \overline{S}_\ell$ such that
$$\begin{aligned}\overline{S}_{\ell+1} = &\{z \in \overline{S}_\ell : z <_{\text{lex}} v \text{ and } z \perp v\}^\frown \mathrm{X} \cup \{z \in \overline{S}_\ell : z <_{\text{lex}} v \text{ and } z \not\perp v\}^\frown \mathrm{L} \cup \{v^\frown \mathrm{L}\} \\ &\cup \{z \in \overline{S}_\ell : v <_{\text{lex}} z <_{\text{lex}} w\}^\frown \mathrm{X} \cup \{w^\frown \mathrm{R}\} \cup \{z \in \overline{S}_\ell : w <_{\text{lex}} z \text{ and } w \perp z\}^\frown \mathrm{X} \\ &\cup \{z \in \overline{S}_\ell : w <_{\text{lex}} z \text{ and } w \not\perp z\}^\frown \mathrm{R}.\end{aligned}$$
   Moreover for every $u \in \overline{S}_\ell$ such that $u <_{\text{lex}} v$, at least one of $u \preceq w$ or $u \perp v$ holds.
   Symmetrically for every $u \in \overline{S}_\ell$ such that $w <_{\text{lex}} u$, at least one of $v \preceq u$ or $w \perp u$ holds.

Definition (Devlin-type)

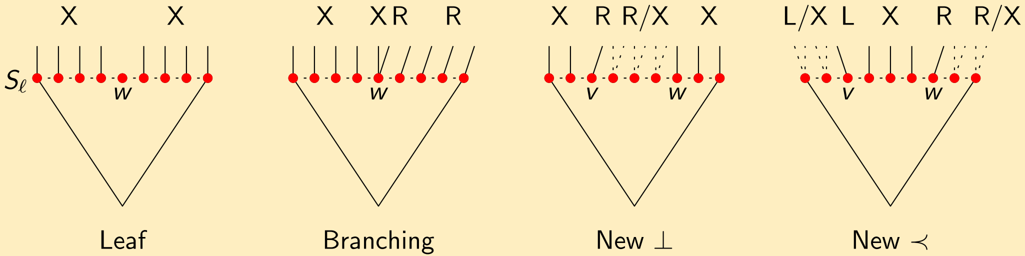
A Devlin-type is any subset S of $\{0,1\}^*$ such that for every $\ell \leq \max_{w \in S} |w|$ precisely one of the following happens:

1. **Leaf:** There is $w \in \overline{S}_\ell$ such that $\overline{S}_{\ell+1} = (\overline{S}_\ell \setminus \{w\})^\frown 0$.
2. **Branching**: There is $w \in \overline{S}_\ell$ such that $\overline{S}_{\ell+1} = w^\frown 1 \cup (\overline{S}_\ell)^\frown 0$.



Definition (Poset-type)

A Poset-type is any subset S of $\{L, X, R\}^*$ such that for every $\ell \leq \max_{w \in S} |w|$ precisely one of the following happens:



Main result

Given a finite partial order $(A, \leq)$, we let $T(A, \leq)$ be the set of all poset-types S such that $(S, \preceq)$ is isomorphic to $(A, \leq)$.

Theorem (M. Balko, D. Chodounský, N. Dobrinen, J. H., M. Konečný, L. Vena, A. Zucker)

For every finite partial order $(O, \leq)$, the big Ramsey degree of $(O, \leq)$ in the universal partial order $(P, \leq)$ equals $|T(O, \leq)| \cdot |\mathrm{Aut}(O, \leq)|$.

Main result

Given a finite partial order $(A, \leq)$, we let $T(A, \leq)$ be the set of all poset-types S such that $(S, \preceq)$ is isomorphic to $(A, \leq)$.

Theorem (M. Balko, D. Chodounský, N. Dobrinen, J. H., M. Konečný, L. Vena, A. Zucker)

For every finite partial order $(O, \leq)$, the big Ramsey degree of $(O, \leq)$ in the universal partial order $(P, \leq)$ equals $|T(O, \leq)| \cdot |\mathrm{Aut}(O, \leq)|$.

Example

Denote by $\mathbf{A}_n$ the anti-chain with with n vertices and by $\mathbf{C}_n$ the chain with n vertices.

$$T(\mathbf{A}_1) = T(\mathbf{C}_1) = \{\emptyset\}$$

$$T(\mathbf{A}_2) = \{\{\mathrm{XR}, \mathrm{RXX}\}, \{\mathrm{XRX}, \mathrm{RX}\}\}$$
$$T(\mathbf{C}_2) = \{\{\mathrm{XL}, \mathrm{RRX}\}, \{\mathrm{XLX}, \mathrm{RR}\}\}$$

Main result

Given a finite partial order $(A, \leq)$, we let $T(A, \leq)$ be the set of all poset-types S such that $(S, \preceq)$ is isomorphic to $(A, \leq)$.

Theorem (M. Balko, D. Chodounský, N. Dobrinen, J. H., M. Konečný, L. Vena, A. Zucker)

For every finite partial order $(O, \leq)$, the big Ramsey degree of $(O, \leq)$ in the universal partial order $(P, \leq)$ equals $|T(O, \leq)| \cdot |\mathrm{Aut}(O, \leq)|$.

Example

Denote by $\mathbf{A}_n$ the anti-chain with with n vertices and by $\mathbf{C}_n$ the chain with n vertices.

$$T(\mathbf{A}_1) = T(\mathbf{C}_1) = \{\emptyset\}$$

$$T(\mathbf{A}_2) = \{\{\mathrm{XR}, \mathrm{RXX}\}, \{\mathrm{XRX}, \mathrm{RX}\}\}$$

$$T(\mathbf{C}_2) = \{\{\mathrm{XL}, \mathrm{RRX}\}, \{\mathrm{XLX}, \mathrm{RR}\}\}$$

$$|T(\mathbf{C}_3)| = 52, |T(\mathbf{C}_4)| = 11000,$$

$$|T(\mathbf{A}_3)| = 84, |T(\mathbf{A}_4)| = 75642$$

Main result

Given a finite partial order $(A, \leq)$, we let $T(A, \leq)$ be the set of all poset-types S such that $(S, \preceq)$ is isomorphic to $(A, \leq)$.

Theorem (M. Balko, D. Chodounský, N. Dobrinen, J. H., M. Konečný, L. Vena, A. Zucker)

For every finite partial order $(O, \leq)$, the big Ramsey degree of $(O, \leq)$ in the universal partial order $(P, \leq)$ equals $|T(O, \leq)| \cdot |\mathrm{Aut}(O, \leq)|$.

Example

Denote by $\mathbf{A}_n$ the anti-chain with with n vertices and by $\mathbf{C}_n$ the chain with n vertices.

$$T(\mathbf{A}_1) = T(\mathbf{C}_1) = \{\emptyset\}$$

$$T(\mathbf{A}_2) = \{\{\mathrm{XR}, \mathrm{RXX}\}, \{\mathrm{XRX}, \mathrm{RX}\}\}$$
$$T(\mathbf{C}_2) = \{\{\mathrm{XL}, \mathrm{RRX}\}, \{\mathrm{XLX}, \mathrm{RR}\}\}$$

$$|T(\mathbf{C}_3)| = 52, |T(\mathbf{C}_4)| = 11000,$$

$$|T(\mathbf{A}_3)| = 84, |T(\mathbf{A}_4)| = 75642$$

Overall there are:

- 1 poset-types a vertex,
- 4 poset-types of posets of size 2,
- 464 poset-types of posets of size 3,
- 1874880 poset-types of posets of size 4.

Approximating posets

Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \not\trianglelefteq b$, $b \not\trianglelefteq a$.)

Approximating posets

Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \not\trianglelefteq b$, $b \not\trianglelefteq a$.)

Definition

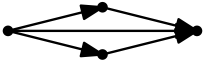
Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

Approximating posets

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation

Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \ntrianglelefteq b$, $b \ntrianglelefteq a$.)

Definition

Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

Approximating posets

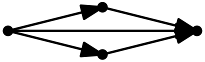
Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting “finished vertices”.
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \ntrianglelefteq b$, $b \ntrianglelefteq a$.)

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation

Definition

Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

Approximating posets

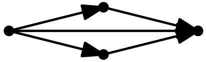
$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Definition (Approximate poset)

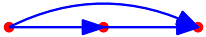
An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \not\trianglelefteq b$, $b \not\trianglelefteq a$.)

Goal



Approximation



Definition

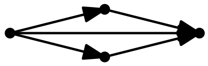
Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

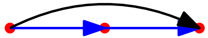
Approximating posets

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \not\trianglelefteq b$, $b \not\trianglelefteq a$.)

Definition

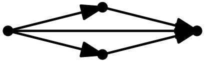
Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

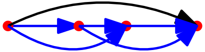
Approximating posets

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \not\trianglelefteq b$, $b \not\trianglelefteq a$.)

Definition

Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

Approximating posets

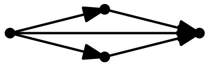
Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

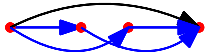
1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \ntrianglelefteq b$, $b \ntrianglelefteq a$.)

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition

Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

Approximating posets

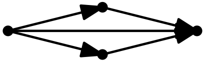
Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

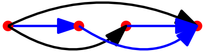
1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting “finished vertices”.
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \ntrianglelefteq b$, $b \ntrianglelefteq a$.)

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition

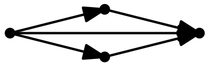
Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

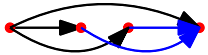
Approximating posets

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting “finished vertices”.
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \not\trianglelefteq b$, $b \not\trianglelefteq a$.)

Definition

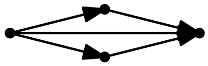
Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

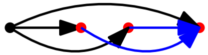
Approximating posets

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \ntrianglelefteq b$, $b \ntrianglelefteq a$.)

Definition

Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

Approximating posets

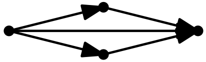
$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Definition (Approximate poset)

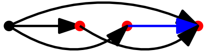
An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \not\trianglelefteq b$, $b \not\trianglelefteq a$.)

Goal



Approximation



Definition

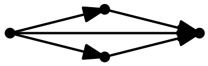
Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

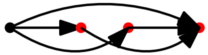
Approximating posets

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting "finished vertices".
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \not\trianglelefteq b$, $b \not\trianglelefteq a$.)

Definition

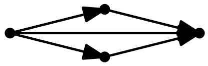
Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

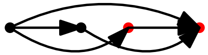
Approximating posets

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting “finished vertices”.
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \not\trianglelefteq b$, $b \not\trianglelefteq a$.)

Definition

Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

Approximating posets

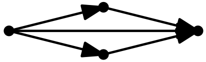
Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

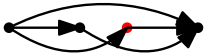
1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting “finished vertices”.
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \ntrianglelefteq b$, $b \ntrianglelefteq a$.)

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition

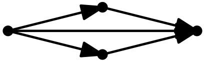
Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

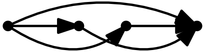
Approximating posets

$\preceq$ is black, $\trianglelefteq$ is blue
P is black, $\neg P$ is red

Goal



Approximation



Definition (Approximate poset)

An approximate poset is a structure $(A, \preceq, \trianglelefteq, P)$ where

1. both $\preceq$ and $\trianglelefteq$ are partial order.
2. P is an unary predicate denoting “finished vertices”.
3. $\preceq \subseteq \trianglelefteq$ (whenever $a \preceq b$ also $a \trianglelefteq b$).
4. $a \preceq b \trianglelefteq c \implies a \preceq c$ and $a \trianglelefteq b \preceq c \implies a \preceq c$.
5. If $a \in p$ then for every $b \in A$, $b \neq a$ it holds one of $a \preceq b$, $b \preceq a$ or $a \perp b$ ($a \perp b$ is a shortcut for $a \ntrianglelefteq b$, $b \ntrianglelefteq a$.)

Definition

Poset-type of a poset $(B, \leq)$ is equivalently sequence of approximations of $(B, \leq)$ where on each step one of the following happens:

1. **Leaf**: New vertex is added to predicate P.
2. **Branching**: A vertex u is split into vertices $u_1 \trianglelefteq u_2$.
3. **New** $\perp$: A pair is removed from relation $\trianglelefteq$.
4. **New** $\prec$: A new pair is added to relation $\preceq$.

Proof outline and work in progress

Proof outline (brief):

- New Ramsey theorem for shape-preserving functions
  (somewhere in between Milliken Tree theorem and the Carlson-Simpson Theorem)
- Lower bounds are obtained by similar techniques as used by
  Laflamme–Sauer–Vuskanovic

Proof outline and work in progress

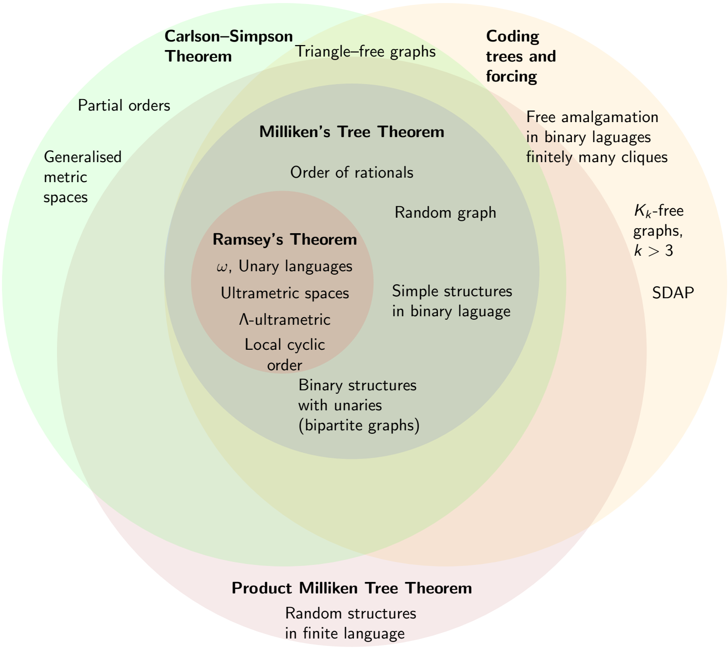
Proof outline (brief):

- New Ramsey theorem for shape-preserving functions
  (somewhere in between Milliken Tree theorem and the Carlson-Simpson Theorem)
- Lower bounds are obtained by similar techniques as used by Laflamme–Sauer–Vuskanovic

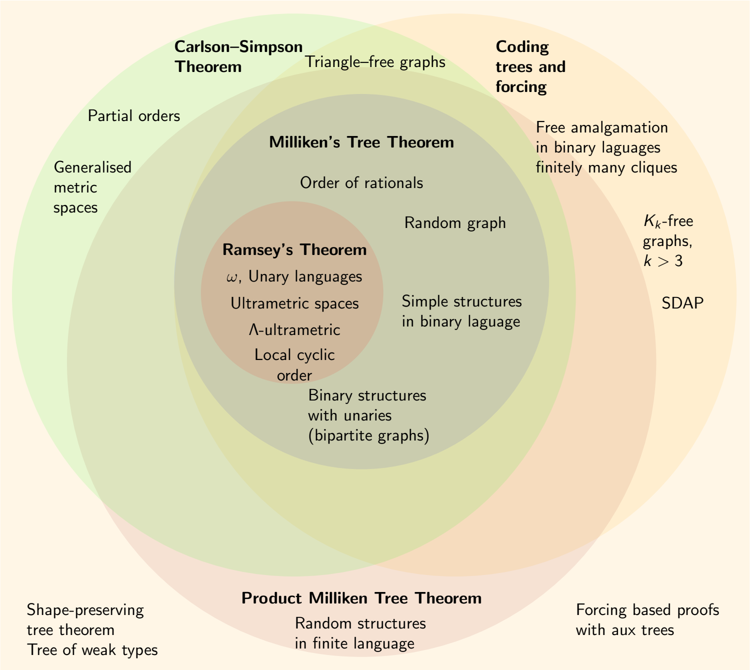
Work in progress

1. Generalizations to metric spaces and other strong amalgamation classes
2. Abstract theorem for trees with successor operations
3. New proof of upper bounds of big Ramsey degrees of free amalgamation classes in binary language
4. Generalizations to higher arity

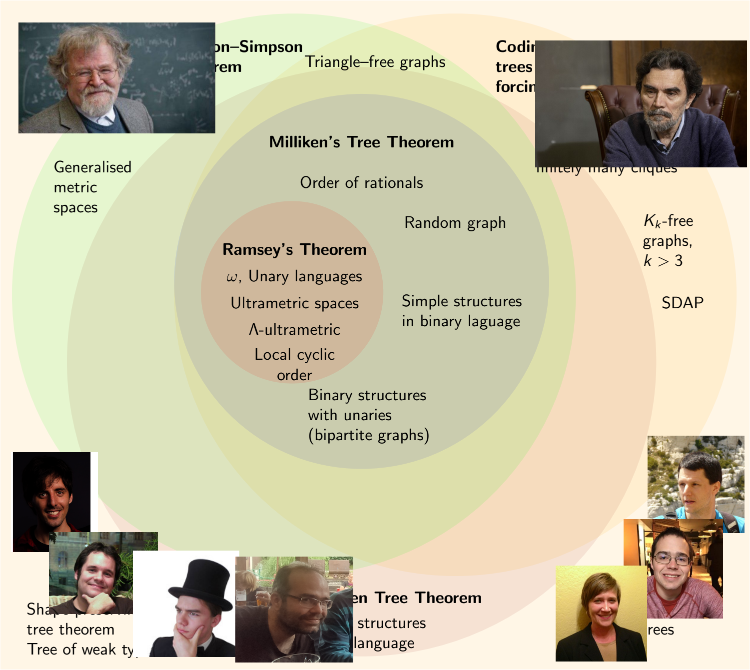
Thank you for the attention



Thank you for the attention



Thank you for the attention



Thank you for the attention

Thank you for the attention

- D. Devlin: Some partition theorems and ultrafilters on ω, PhD thesis, Dartmouth College, 1979.
- N. Sauer: Coloring subgraphs of the Rado graph, Combinatorica 26 (2) (2006), 231–253.
- C. Laflamme, L. Nguyen Van Thé, N. W. Sauer, Partition properties of the dense local order and a colored version of Milliken's theorem, Combinatorica 30(1) (2010), 83–104.
- L. Nguyen Van Thé, Big Ramsey degrees and divisibility in classes of ultrametric spaces, Canadian Mathematical Bulletin, Bulletin Canadien de Mathematiques, 51 (3) (2008),413–423.
- N. Dobrinen, The Ramsey theory of the universal homogeneous triangle-free graph, Journal of Mathematical Logic 2020.
- N. Dobrinen, The Ramsey Theory of Henson graphs, arXiv:1901.06660 (2019).
- A. Zucker, Big Ramsey degrees and topological dynamics, Groups Geom. Dyn., 2018.
- J.H., J. Nešetřil: All those Ramsey classes (Ramsey classes with closures and forbidden homomorphisms), Advances in Mathematics, 2020.
- N. Sauer: Coloring homogeneous structures, arXiv:2008.02375v2.
- N. Dobrinen: The Ramsey theory of the universal homogeneous triangle-free graph Part II: Exact big Ramsey degrees, arXiv:2009.01985.
- R. Coulson, N. Dobrinen, R. Patel: The Substructure Disjoint Amalgamation Property implies big Ramsey structures, arXiv:2010.02034.
- J.H.: Big Ramsey degrees using parameter spaces, arXiv:2009.00967.
- M. Balko, D. Chodounský, J.H., M. Konečný, L. Vena: Big Ramsey degrees of 3-uniform hypergraphs are finite, To appear in Combinatorica.