A combinatorial theorem for trees

Thomas Colcombet
CNRS, Université Paris Diderot
Algorithmic-Logical Theory of Infinite Structures
Dagstuhl, 01.11.2007

MOTIVATIONS

Exhibit the 'regularity' of the behaviour of an automaton over a word, by adding 'unary information' on the word.

Theorem of Ramsey
Every ω-word can be written as $uv_1v_2\ldots$
such that all the v_i are equivalent for the automaton.

Theorem of factorisation forests of Simon

We do not want to guess the factorisation.
Goal: make its computation deterministic, online.
Make the computation of the information deterministicaly computable

Application to logic on trees, to infinite structures, to McNaugton's determinisation theorem, to...

SEMIGROUPS

Def: Semigroup: $(S, .)$, with multiplication associative.

Intuition: Finite semigroups are "equivalent" to finite state automata.
Fix a finite automata (Q, δ):

$$S = \mathcal{P}(Q \times Q)$$
$$a.b = \{(p, q) \ : \ (p, r) \in a, \ (r, q) \in b\}$$

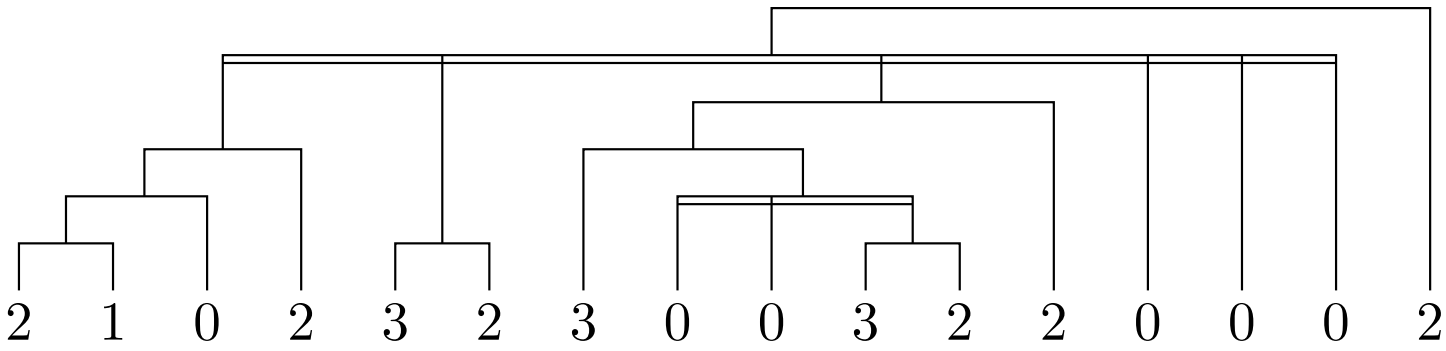
Below we identify S with the alphabet.

FACTORISATION THEOREM BY AN EXAMPLE

Set $S = (\mathbb{Z}/5\mathbb{Z}, +)$ and the word u =210232300322002.

FACTORISATION THEOREM BY AN EXAMPLE

Set $S = (\mathbb{Z}/5\mathbb{Z}, +)$ and the word u =210232300322002.

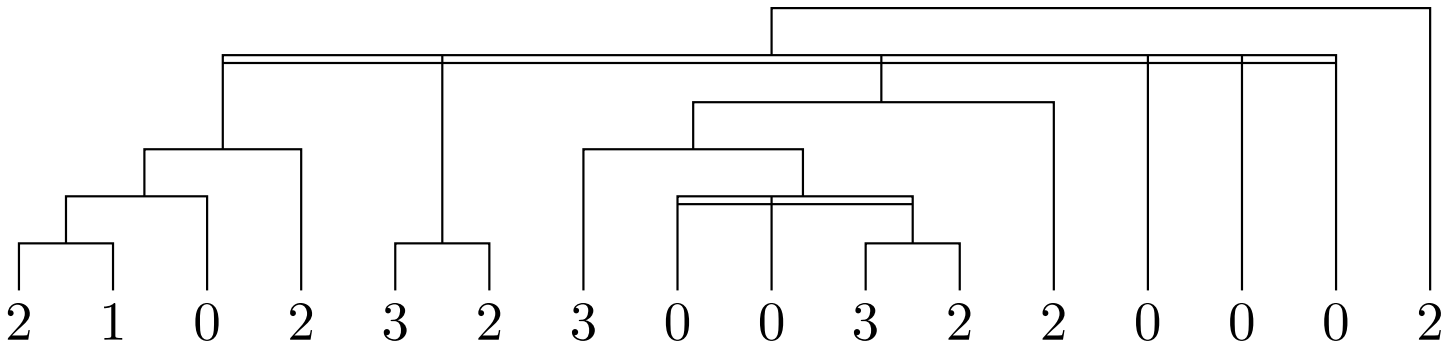


A factorising tree is a tree such that:

- leaves are labeled by letters,
- reading leaves from left to right yields the word.

FACTORISATION THEOREM BY AN EXAMPLE

Set $S = (\mathbb{Z}/5\mathbb{Z}, +)$ and the word u =210232300322002.



A factorising tree is a tree such that:

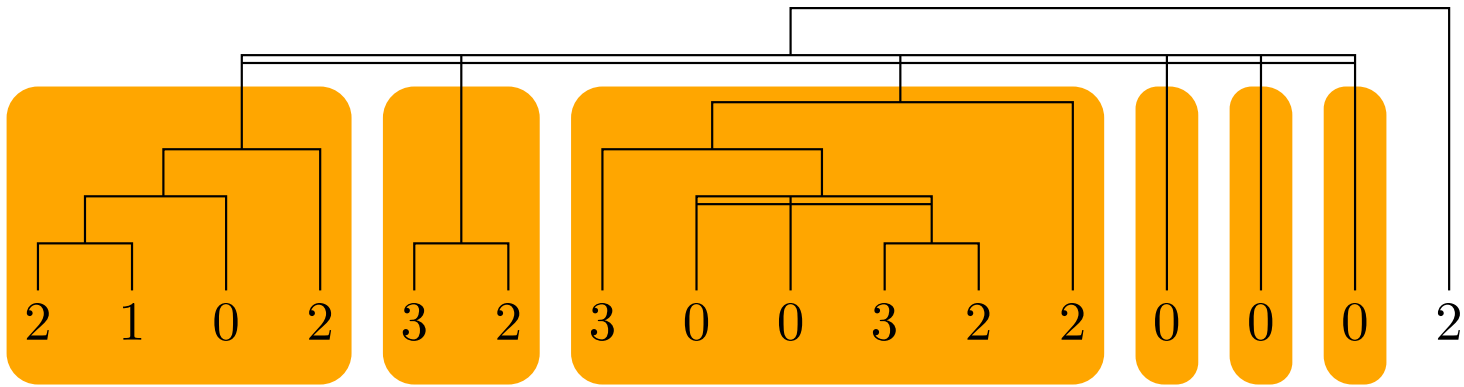
- leaves are labeled by letters,
- reading leaves from left to right yields the word.

A factorising tree is Ramseyan if every node:

- is a leaf, or;
- has two children, or;
- all its children have as value the same idempotent of S.

FACTORISATION THEOREM BY AN EXAMPLE

Set $S = (\mathbb{Z}/5\mathbb{Z}, +)$ and the word u =210232300322002.



A factorising tree is a tree such that:

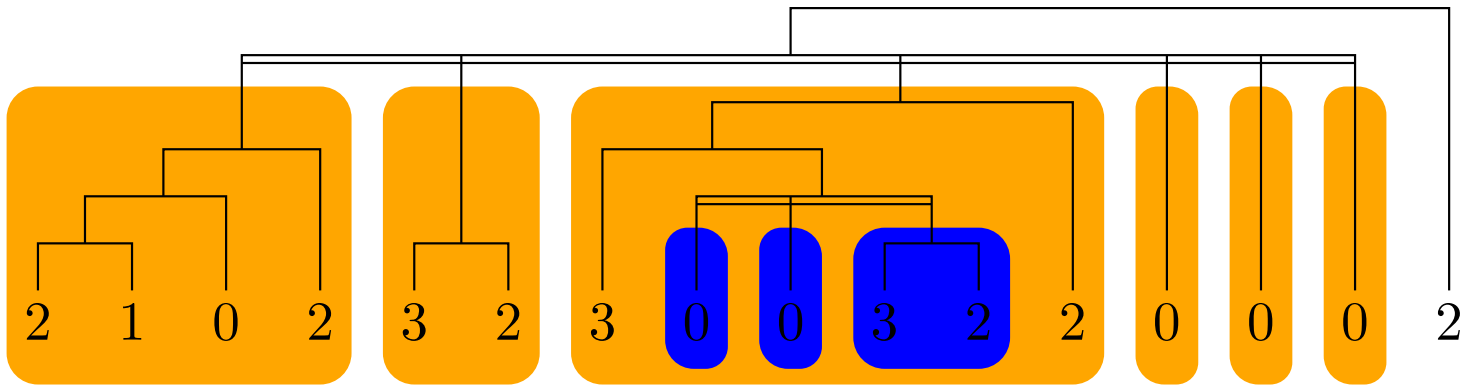
- leaves are labeled by letters,
- reading leaves from left to right yields the word.

A factorising tree is Ramseyan if every node:

- is a leaf, or;
- has two children, or;
- all its children have as value the same idempotent of S.

FACTORISATION THEOREM BY AN EXAMPLE

Set $S = (\mathbb{Z}/5\mathbb{Z}, +)$ and the word u =210232300322002.



A factorising tree is a tree such that:

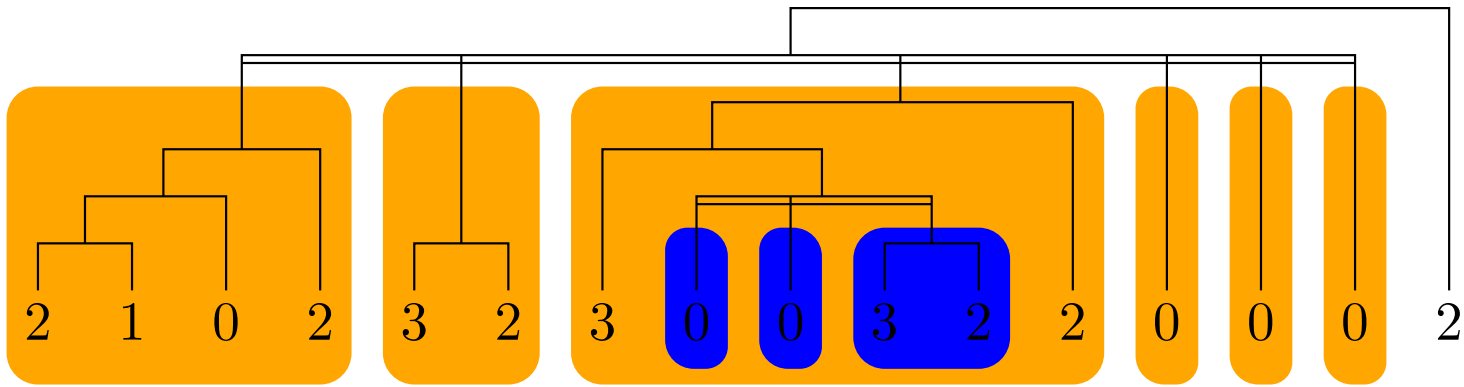
- leaves are labeled by letters,
- reading leaves from left to right yields the word.

A factorising tree is Ramseyan if every node:

- is a leaf, or;
- has two children, or;
- all its children have as value the same idempotent of S.

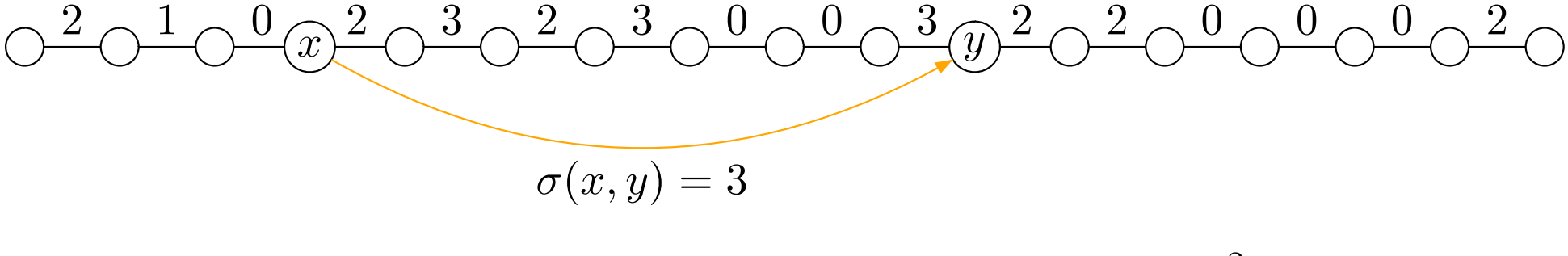
FACTORISATION THEOREM BY AN EXAMPLE

Set $S = (\mathbb{Z}/5\mathbb{Z}, +)$ and the word u =210232300322002.



Th(Simon 90, C07): For every word over a finite semigroup S, it admits a Ramseyan factorisation tree of height at most $3|S|$.

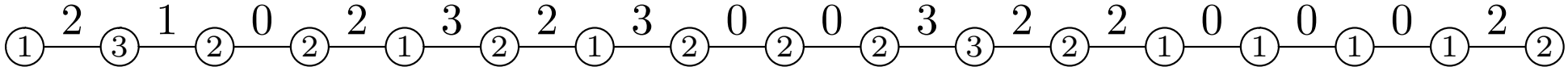
STATEMENT BY SPLITS



Def: Let α be a linear ordering, an additive labeling $\sigma : \alpha^2 \to S$ is such that:

- $\sigma(x, y)$ is defined iff $x < y$, and,
- $\forall x < y < z \in \alpha, \qquad \sigma(x, z) = \sigma(x, y).\sigma(y, z)$

STATEMENT BY SPLITS

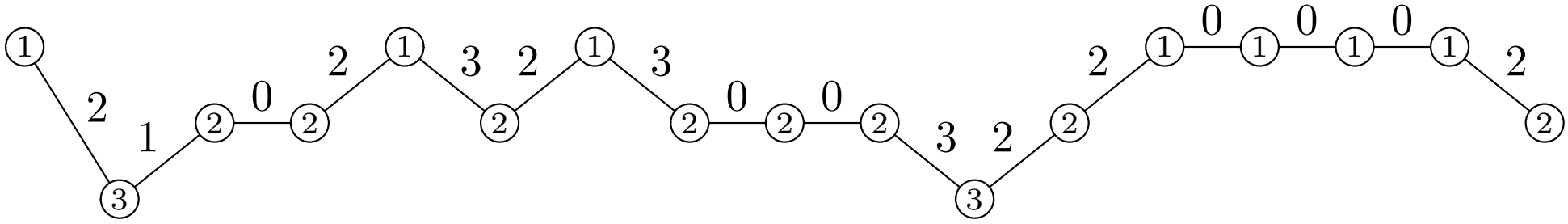


Def: Let α be a linear ordering, an additive labeling $\sigma : \alpha^2 \to S$ is such that:

- $\sigma(x, y)$ is defined iff $x < y$, and,
- $\forall x < y < z \in \alpha, \qquad \sigma(x, z) = \sigma(x, y).\sigma(y, z)$

Def: A split of height N is a mapping $s : \alpha \to [1, N]$.

STATEMENT BY SPLITS

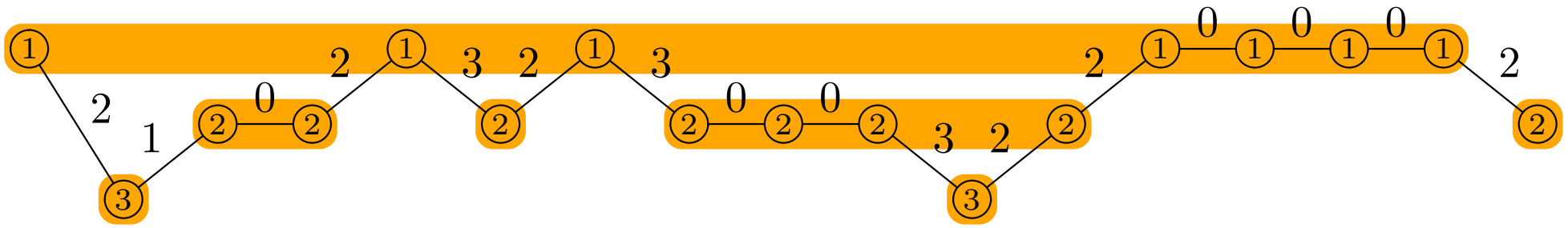


Def: Let α be a linear ordering, an additive labeling $\sigma : \alpha^2 \to S$ is such that:

- $\sigma(x, y)$ is defined iff $x < y$, and,
- $\forall x < y < z \in \alpha, \qquad \sigma(x, z) = \sigma(x, y).\sigma(y, z)$

Def: A split of height N is a mapping $s : \alpha \to [1, N]$.

STATEMENT BY SPLITS



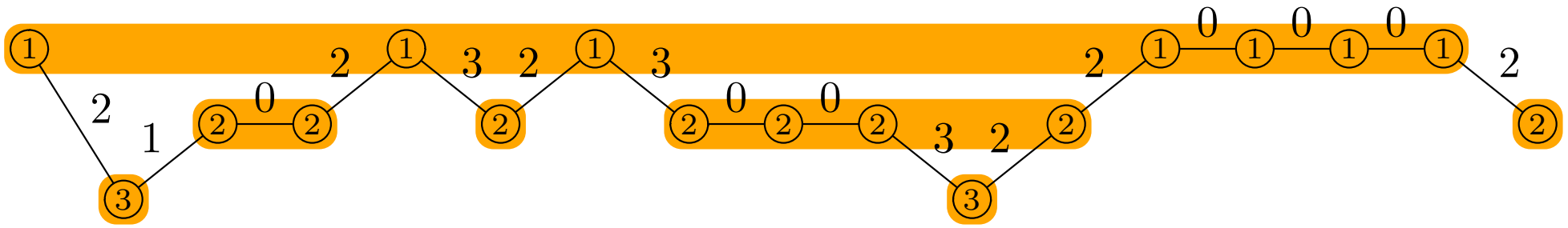
Def: Let α be a linear ordering, an additive labeling $\sigma : \alpha^2 \to S$ is such that:

- $\sigma(x, y)$ is defined iff $x < y$, and,
- $\forall x < y < z \in \alpha, \qquad \sigma(x, z) = \sigma(x, y).\sigma(y, z)$

Def: A split of height N is a mapping $s : \alpha \to [1, N]$.

Def: $x \sim_s y$ si $s(x) = s(y)$ et pour tout $z \in [\min(x, y), \max(x, y)]$, $s(z) \geq s(x)$.

STATEMENT BY SPLITS



Def: Let α be a linear ordering, an additive labeling $\sigma : \alpha^2 \to S$ is such that:

- $\sigma(x, y)$ is defined iff $x < y$, and,
- $\forall x < y < z \in \alpha, \qquad \sigma(x, z) = \sigma(x, y).\sigma(y, z)$

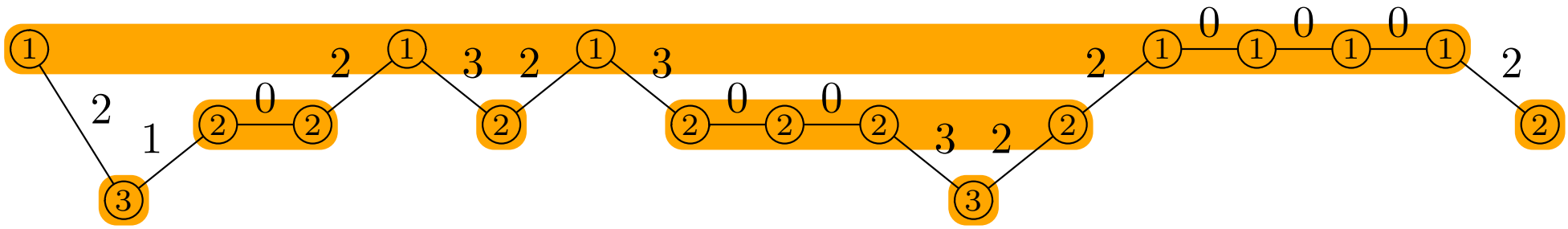
Def: A split of height N is a mapping $s : \alpha \to [1, N]$.

Def: $x \sim_s y$ si $s(x) = s(y)$ et pour tout $z \in [\min(x, y), \max(x, y)]$, $s(z) \geq s(x)$.

Def: A split is Ramseyan for σ if:
$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \to \sigma(x, y) = \sigma(x', y') \qquad (= \sigma(x, y)^2)$

Th(variant of Simon): Every additive labeling of a finite linear ordering by a finite semigroup S admits a Ramseyan split of height at most $|S|$.

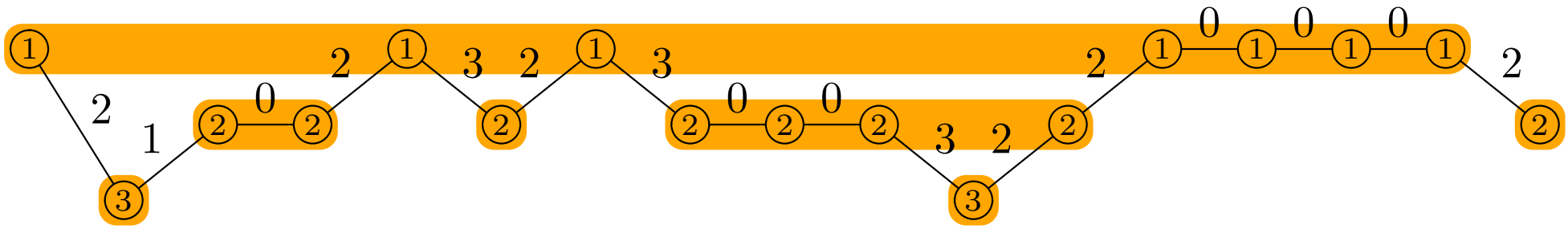
DETERMINISTIC VARIANT



Def: A split is Ramseyan for σ if:

$$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \to \sigma(x, y) = \sigma(x', y') = \sigma(x, y)^2$$

DETERMINISTIC VARIANT



Def: A split is Ramseyan for σ if:

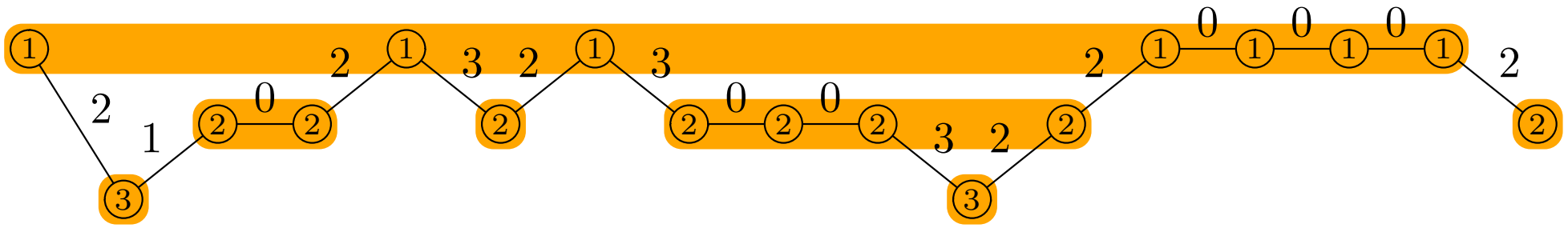
$$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \rightarrow \sigma(x,y) = \sigma(x',y') = \sigma(x,y)^2$$

Weakening of the Ramseyanity

Def: A split is **forward Ramseyan** if:

$$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \rightarrow \sigma(x,y) = \sigma(x,y).\sigma(x',y')$$

DETERMINISTIC VARIANT



Def: A split is Ramseyan for σ if:

$$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \to \sigma(x,y) = \sigma(x',y') = \sigma(x,y)^2$$

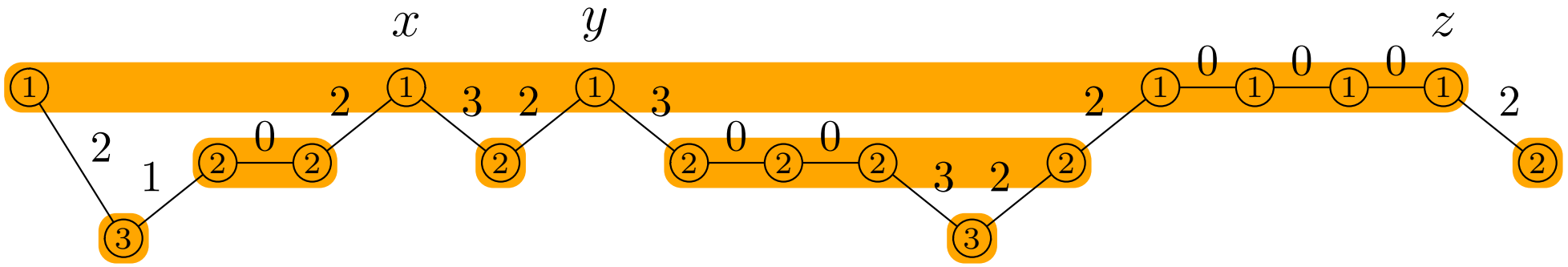
Weakening of the Ramseyanity

Def: A split is **forward Ramseyan** if:

$$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \to \sigma(x,y) = \sigma(x,y).\sigma(x',y')$$

In particular: $\forall x < y < z. \quad (x \sim_s y \sim_s z) \to \sigma(x,y) = \sigma(x,z)$

DETERMINISTIC VARIANT



Def: A split is Ramseyan for σ if:

$$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \to \sigma(x,y) = \sigma(x',y') = \sigma(x,y)^2$$

Weakening of the Ramseyanity

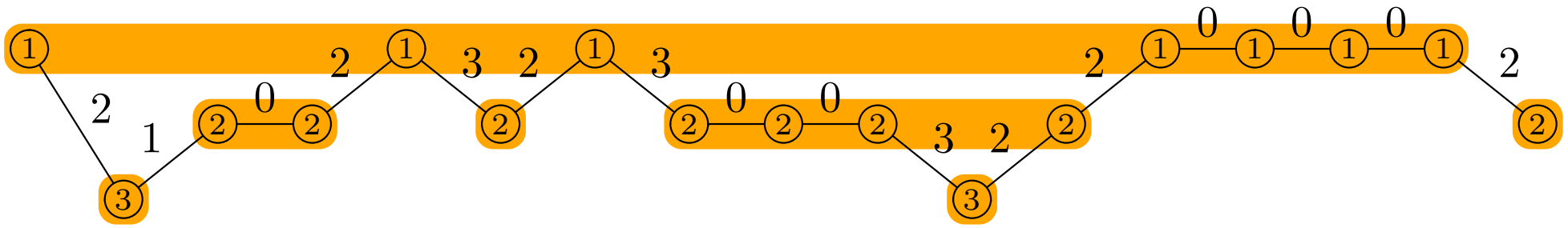
Def: A split is **forward Ramseyan** if:

$$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \to \sigma(x,y) = \sigma(x,y).\sigma(x',y')$$

In particular: $\forall x < y < z. \quad (x \sim_s y \sim_s z) \to \sigma(x,y) = \sigma(x,z)$

Th: Every additive labeling of a finite linear ordering by a finite semigroup S admits a forward Ramseyan split s of height at most $|S|$ which is computable deterministicaly online by a finite automaton.

DETERMINISTIC VARIANT



Def: A split is Ramseyan for σ if:

$$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \to \sigma(x,y) = \sigma(x',y') = \sigma(x,y)^2$$

Weakening of the Ramseyanity

Def: A split is **forward Ramseyan** if:

$$\forall x < y, x' < y'. \quad (x \sim_s y \sim_s x' \sim_s y') \to \sigma(x,y) = \sigma(x,y).\sigma(x',y')$$

In particular: $\forall x < y < z. \quad (x \sim_s y \sim_s z) \to \sigma(x,y) = \sigma(x,z)$

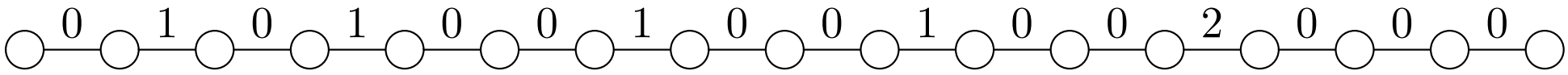
Th: Every additive labeling of a finite linear ordering by a finite semigroup S admits a forward Ramseyan split s of height at most $|S|$ which is computable deterministicaly online by a finite automaton.

There is a partition of S^* in regular langages $L_1, \ldots, L_K$, $K \leq |S|$ such that:

$$s(x) = i \qquad \text{iff} \qquad \langle\sigma\rangle|_{[0,x]} \in L_i \ .$$

SOME ARGUMENTS OF THE PROOF

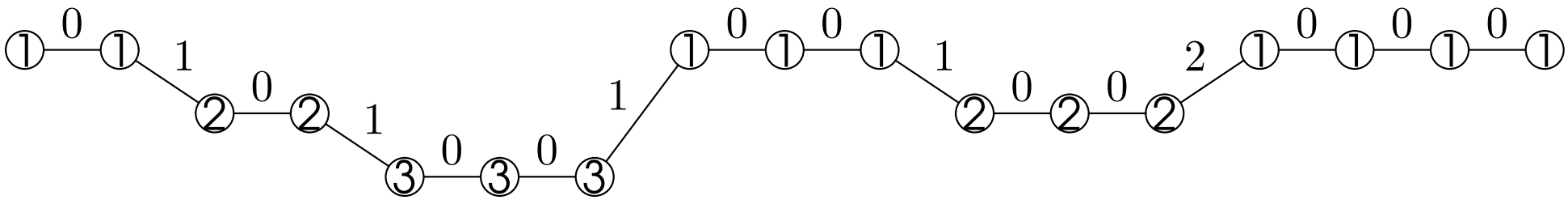
Group case. E.g., $(\mathbb{Z}/3\mathbb{Z}, +)$



Number S by n. Set $s(x) = n(\sigma(0, x))$, $s(0) = n(e)$.

SOME ARGUMENTS OF THE PROOF

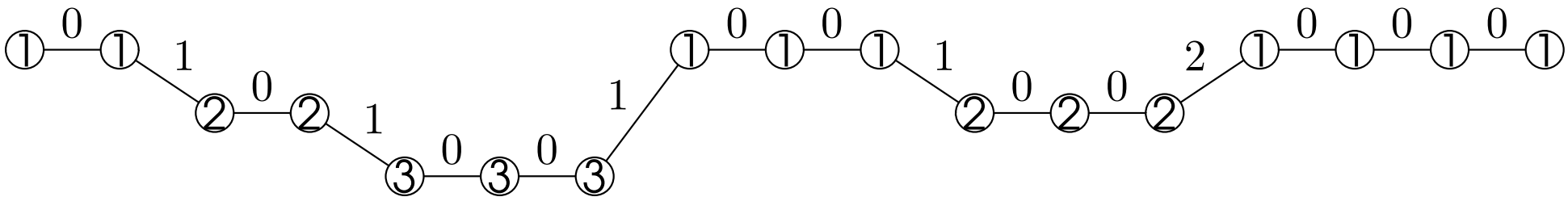
Group case. E.g., $(\mathbb{Z}/3\mathbb{Z}, +)$



Number S by n. Set $s(x) = n(\sigma(0, x))$, $s(0) = n(e)$.

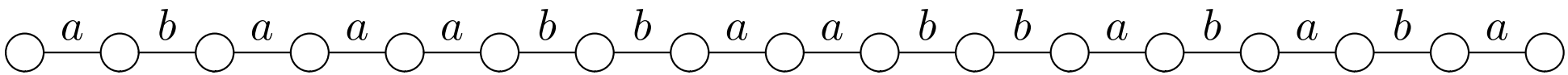
SOME ARGUMENTS OF THE PROOF

Group case. E.g., $(\mathbb{Z}/3\mathbb{Z}, +)$



Number S by n. Set $s(x) = n(\sigma(0, x))$, $s(0) = n(e)$.

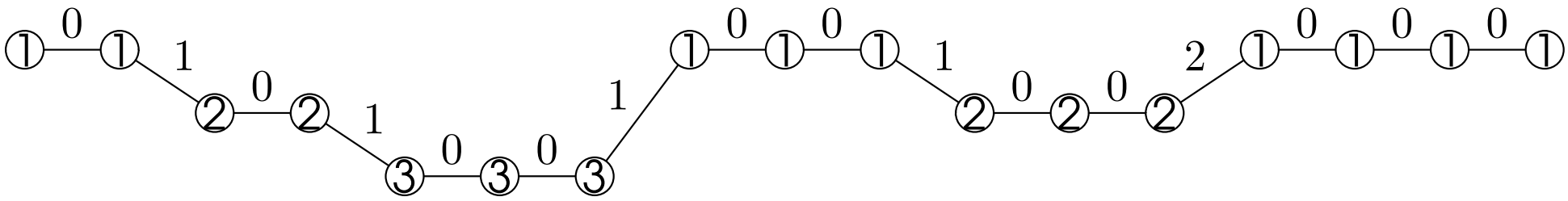
$\mathcal{L}$-equivalent elements. $ab = bb = b$, $aa = ba = a$.



Number S by $n : S \to [1, N]$. Set $s(x) = n(\sigma(x - 1, x))$, $s(0) = 1$.

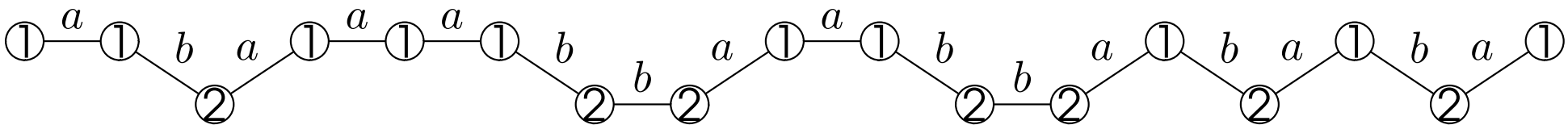
SOME ARGUMENTS OF THE PROOF

Group case. E.g., $(\mathbb{Z}/3\mathbb{Z}, +)$



Number S by n. Set $s(x) = n(\sigma(0, x))$, $s(0) = n(e)$.

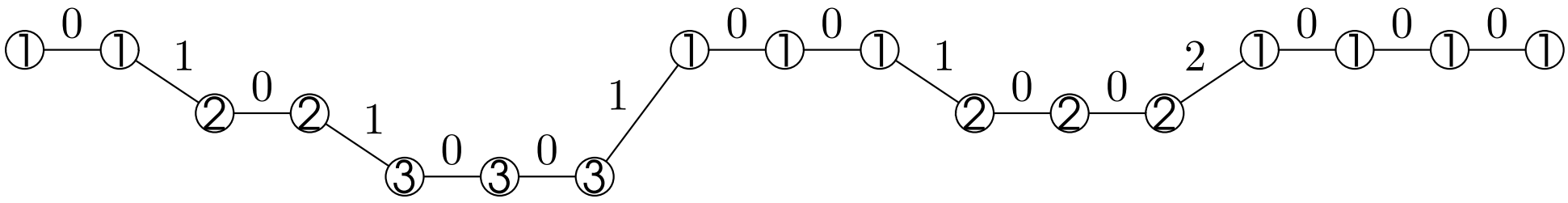
$\mathcal{L}$-equivalent elements. $ab = bb = b$, $aa = ba = a$.



Number S by $n : S \to [1, N]$. Set $s(x) = n(\sigma(x-1, x))$, $s(0) = 1$.

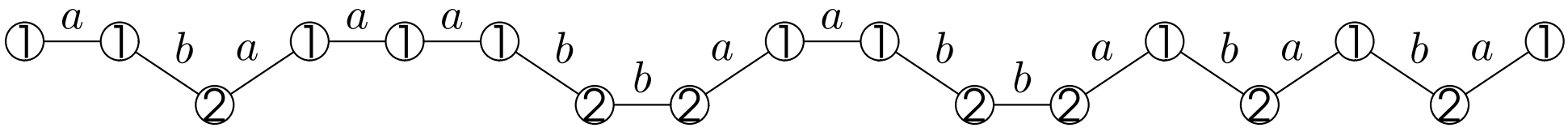
SOME ARGUMENTS OF THE PROOF

Group case. E.g., $(\mathbb{Z}/3\mathbb{Z}, +)$



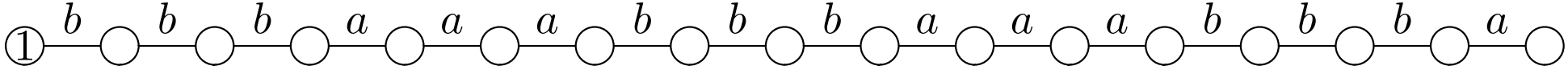
Number S by n. Set $s(x) = n(\sigma(0, x))$, $s(0) = n(e)$.

$\mathcal{L}$-equivalent elements. $ab = bb = b$, $aa = ba = a$.



Number S by $n : S \to [1, N]$. Set $s(x) = n(\sigma(x-1, x))$, $s(0) = 1$.

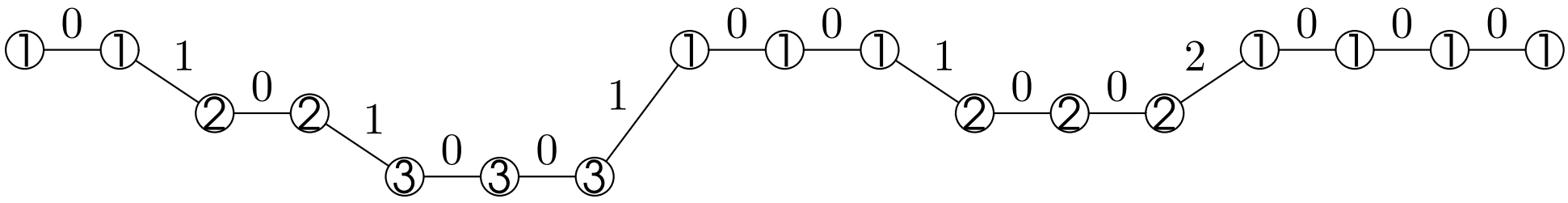
Case of different $\mathcal{J}$-classes. $c = *c = c* = ab \;\; \leq_{\mathcal{J}} \;\; b = bb = ba \;\; \leq_{\mathcal{J}} \;\; a = aa$



By induction on $\leq_{\mathcal{J}}$.

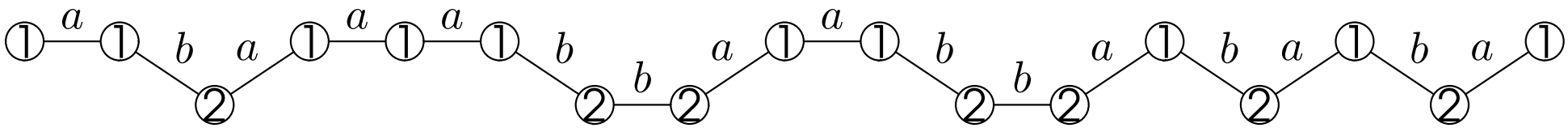
SOME ARGUMENTS OF THE PROOF

Group case. E.g., $(\mathbb{Z}/3\mathbb{Z}, +)$



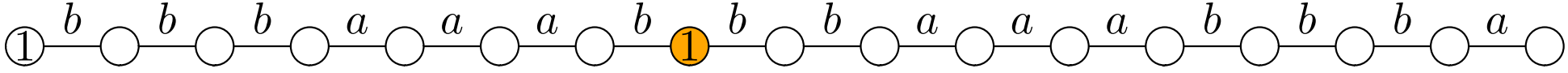
Number S by n. Set $s(x) = n(\sigma(0, x))$, $s(0) = n(e)$.

$\mathcal{L}$-equivalent elements. $ab = bb = b$, $aa = ba = a$.



Number S by $n : S \to [1, N]$. Set $s(x) = n(\sigma(x-1, x))$, $s(0) = 1$.

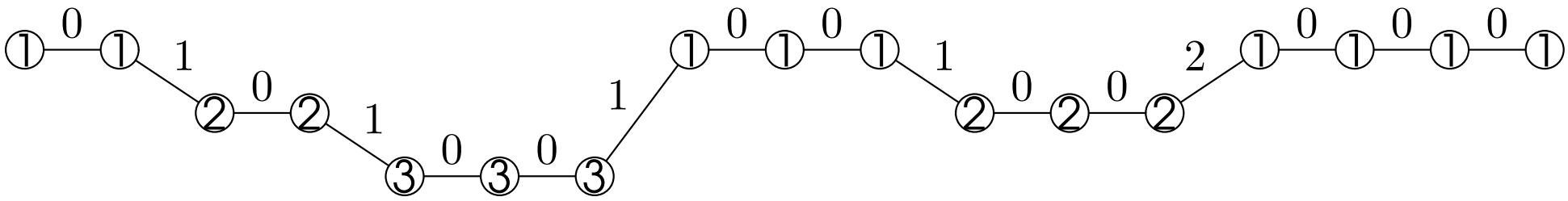
Case of different $\mathcal{J}$-classes. $c = *c = c* = ab \;\; \leq_{\mathcal{J}} \;\; b = bb = ba \;\; \leq_{\mathcal{J}} \;\; a = aa$



By induction on $\leq_{\mathcal{J}}$.

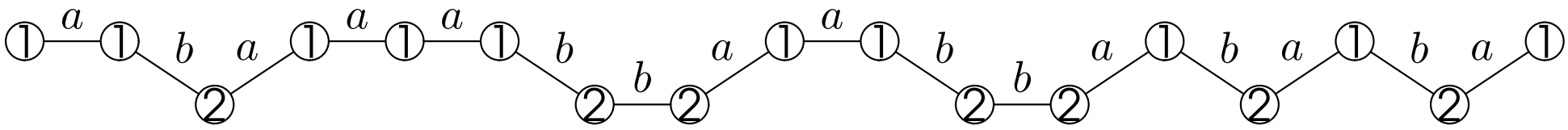
SOME ARGUMENTS OF THE PROOF

Group case. E.g., $(\mathbb{Z}/3\mathbb{Z}, +)$



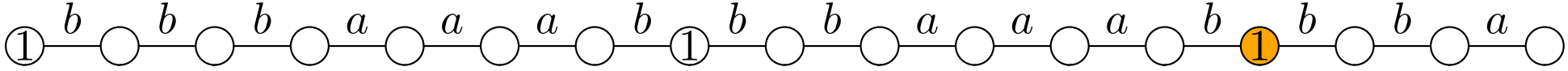
Number S by n. Set $s(x) = n(\sigma(0, x))$, $s(0) = n(e)$.

$\mathcal{L}$-equivalent elements. $ab = bb = b$, $aa = ba = a$.



Number S by $n : S \to [1, N]$. Set $s(x) = n(\sigma(x-1, x))$, $s(0) = 1$.

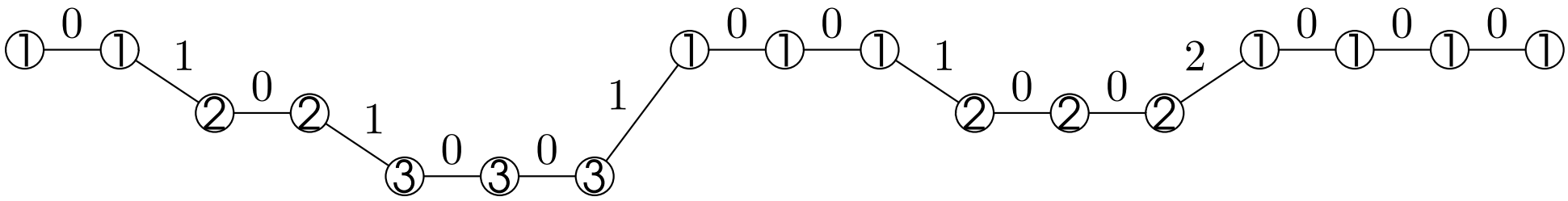
Case of different $\mathcal{J}$-classes. $c = *c = c* = ab \;\; \leq_{\mathcal{J}} \;\; b = bb = ba \;\; \leq_{\mathcal{J}} \;\; a = aa$



By induction on $\leq_{\mathcal{J}}$.

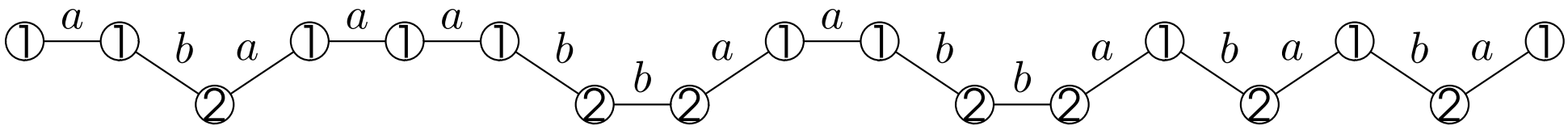
SOME ARGUMENTS OF THE PROOF

Group case. E.g., $(\mathbb{Z}/3\mathbb{Z}, +)$



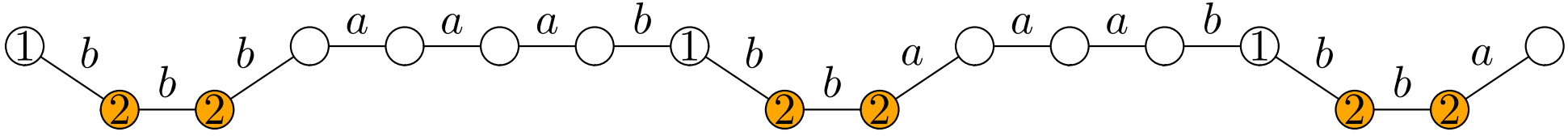
Number S by n. Set $s(x) = n(\sigma(0, x))$, $s(0) = n(e)$.

$\mathcal{L}$-equivalent elements. $ab = bb = b$, $aa = ba = a$.



Number S by $n : S \to [1, N]$. Set $s(x) = n(\sigma(x-1, x))$, $s(0) = 1$.

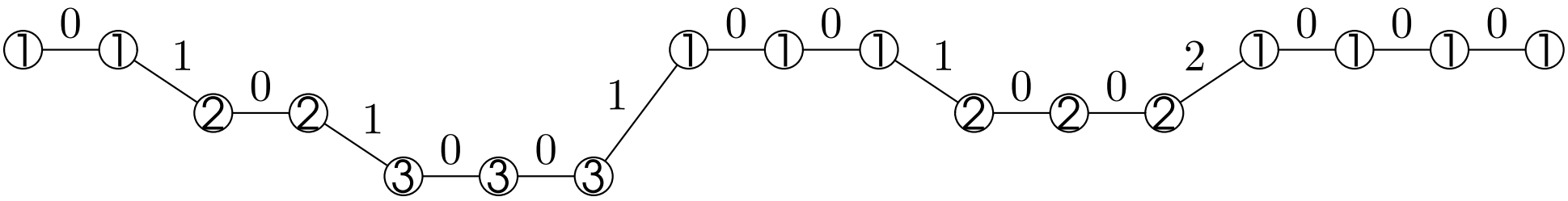
Case of different $\mathcal{J}$-classes. $c = *c = c* = ab \quad \leq_{\mathcal{J}} \quad b = bb = ba \quad \leq_{\mathcal{J}} \quad a = aa$



By induction on $\leq_{\mathcal{J}}$.

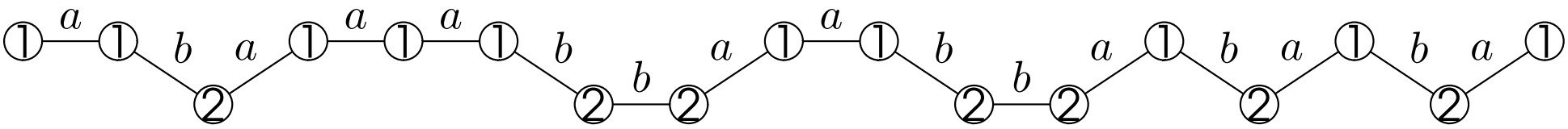
SOME ARGUMENTS OF THE PROOF

Group case. E.g., $(\mathbb{Z}/3\mathbb{Z}, +)$



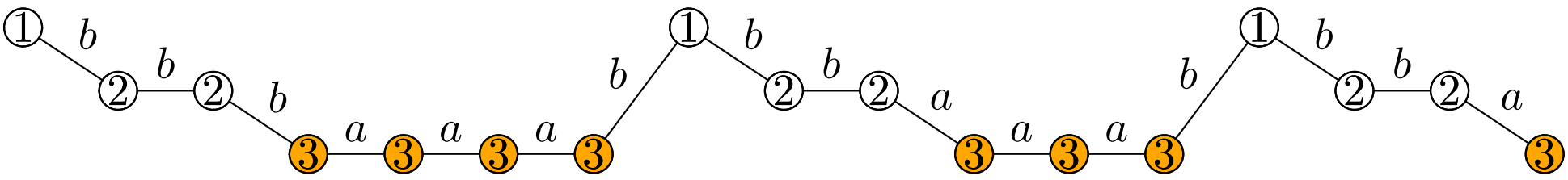
Number S by n. Set $s(x) = n(\sigma(0, x))$, $s(0) = n(e)$.

$\mathcal{L}$-equivalent elements. $ab = bb = b$, $aa = ba = a$.



Number S by $n : S \to [1, N]$. Set $s(x) = n(\sigma(x-1, x))$, $s(0) = 1$.

Case of different $\mathcal{J}$-classes. $c = *c = c* = ab \quad \leq_{\mathcal{J}} \quad b = bb = ba \quad \leq_{\mathcal{J}} \quad a = aa$



By induction on $\leq_{\mathcal{J}}$.

General case: Interleaving of the cases above following Green's relations.

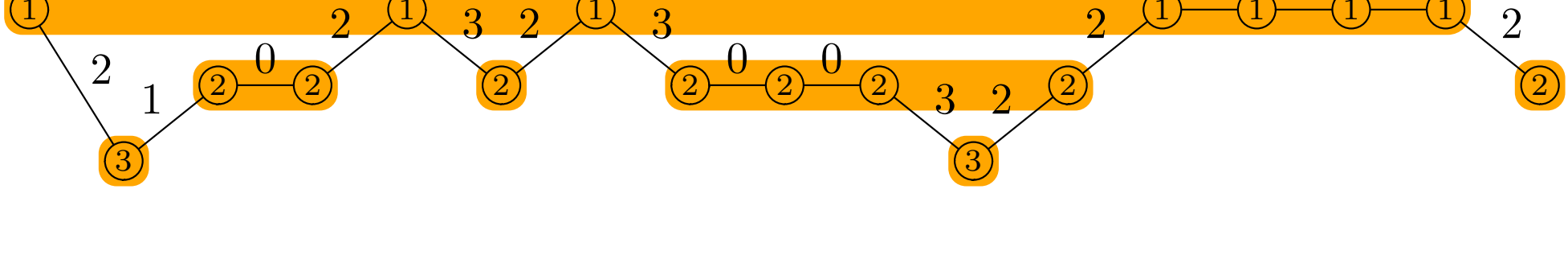
Applications

SPLIT AS AN ACCELERATING STRUCTURE

Problem: Given a regular language L, and a word u, preprocess in linear time the word such that membership of a factor of u in L is constant time.

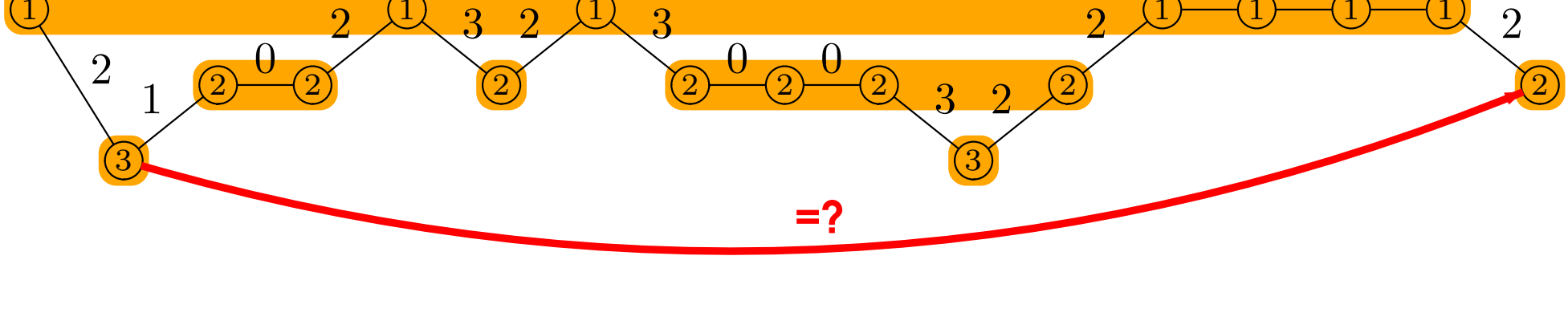
SPLIT AS AN ACCELERATING STRUCTURE

Problem: Given a regular language L, and a word u, preprocess in linear time the word such that membership of a factor of u in L is constant time.



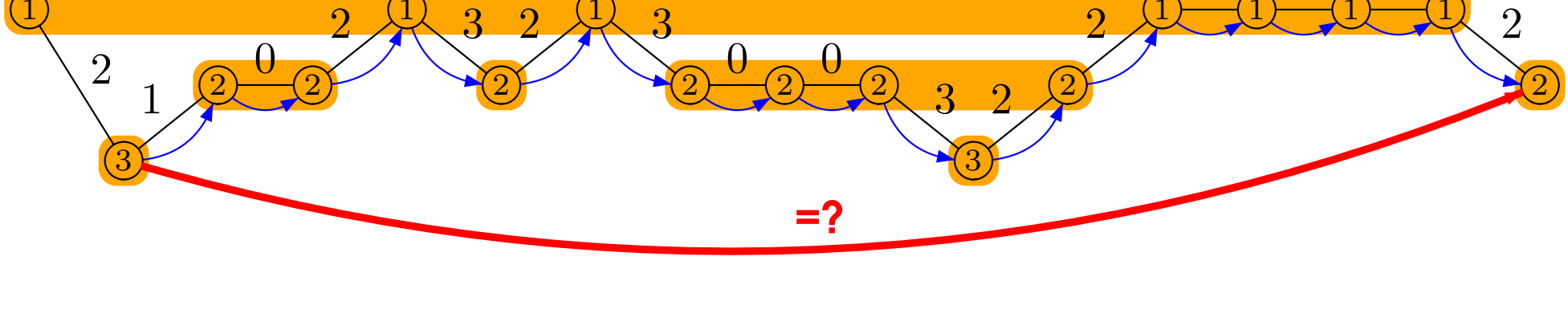
SPLIT AS AN ACCELERATING STRUCTURE

Problem: Given a regular language L, and a word u, preprocess in linear time the word such that membership of a factor of u in L is constant time.



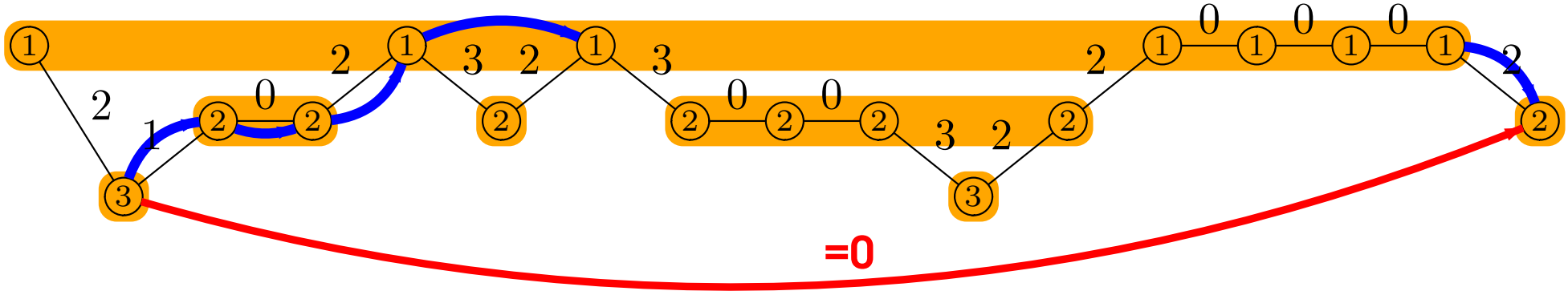
SPLIT AS AN ACCELERATING STRUCTURE

Problem: Given a regular language L, and a word u, preprocess in linear time the word such that membership of a factor of u in L is constant time.



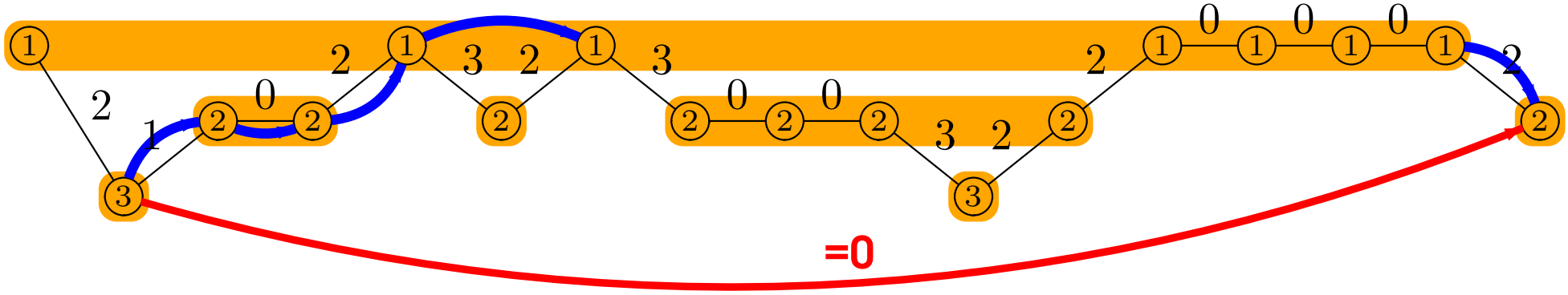
SPLIT AS AN ACCELERATING STRUCTURE

Problem: Given a regular language L, and a word u, preprocess in linear time the word such that membership of a factor of u in L is constant time.



SPLIT AS AN ACCELERATING STRUCTURE

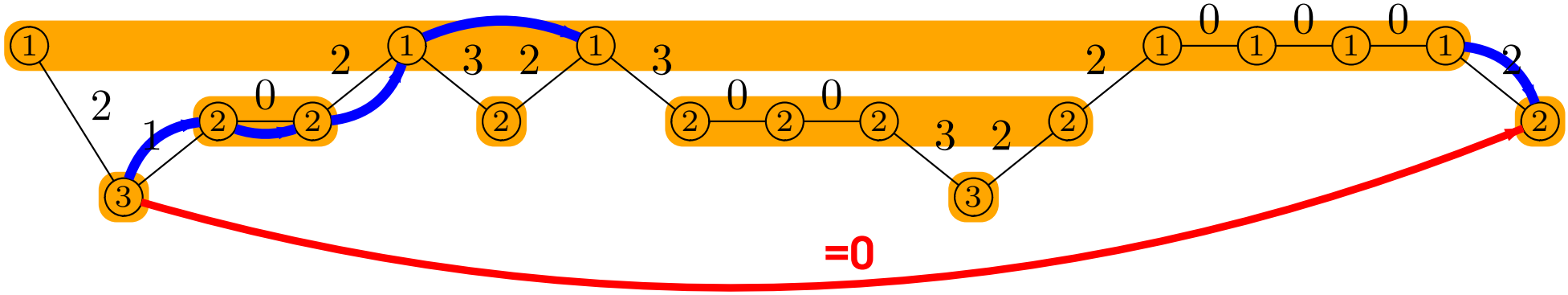
Problem: Given a regular language L, and a word u, preprocess in linear time the word such that membership of a factor of u in L is constant time.



Conclusion: Computing $\sigma(x, y)$ for a given x, y is constant.

SPLIT AS AN ACCELERATING STRUCTURE

Problem: Given a regular language L, and a word u, preprocess in linear time the word such that membership of a factor of u in L is constant time.



Conclusion: Computing $\sigma(x, y)$ for a given x, y is constant.

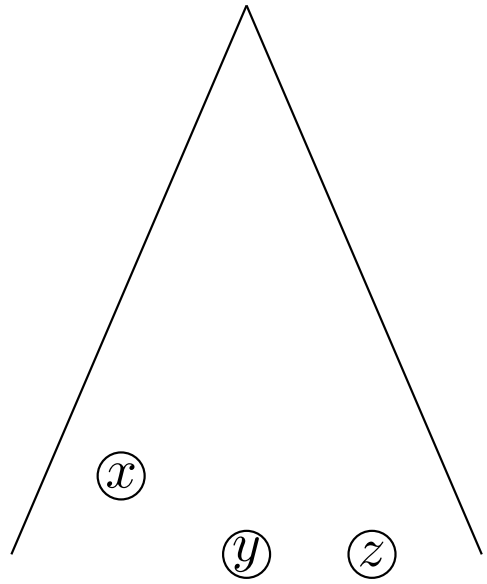
Logic statement: $\sigma(x, y) = a$ can be expressed by a first-order formula $\phi_a(x, y)$ using the order (while monadic second-order is required if no split is given).

MSO OVER TREES

Problem: Given an MSO formula $\Phi(x_1, \ldots, x_n)$, transform it into a FO-formula, equivalent over trees, using $<$ and monadic formulas with a single free variable as predicates.

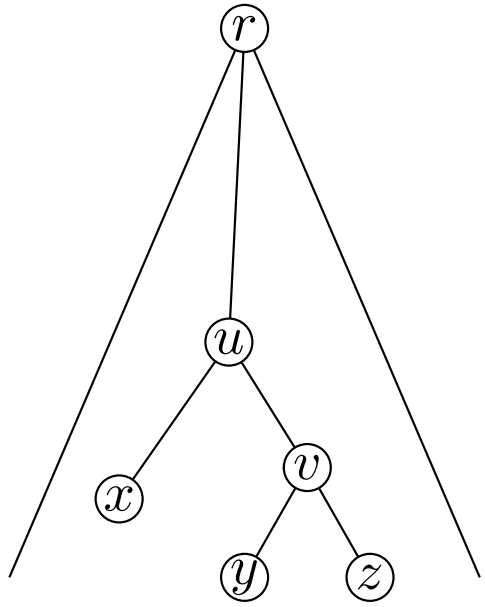
MSO OVER TREES

Problem: Given an MSO formula $\Phi(x_1, \ldots, x_n)$, transform it into a FO-formula, equivalent over trees, using $<$ and monadic formulas with a single free variable as predicates.



MSO OVER TREES

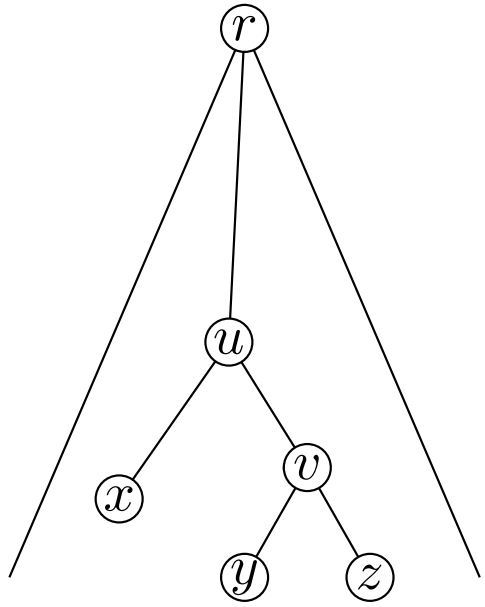
Problem: Given an MSO formula $\Phi(x_1, \ldots, x_n)$, transform it into a FO-formula, equivalent over trees, using $<$ and monadic formulas with a single free variable as predicates.



Step 1: Guess the branching structure of the variables (doable with FO).
$\Longrightarrow$ reduces the problem to formulas of the form $x < y \wedge \Psi(x, y)$

MSO OVER TREES

Problem: Given an MSO formula $\Phi(x_1, \ldots, x_n)$, transform it into a FO-formula, equivalent over trees, using $<$ and monadic formulas with a single free variable as predicates.

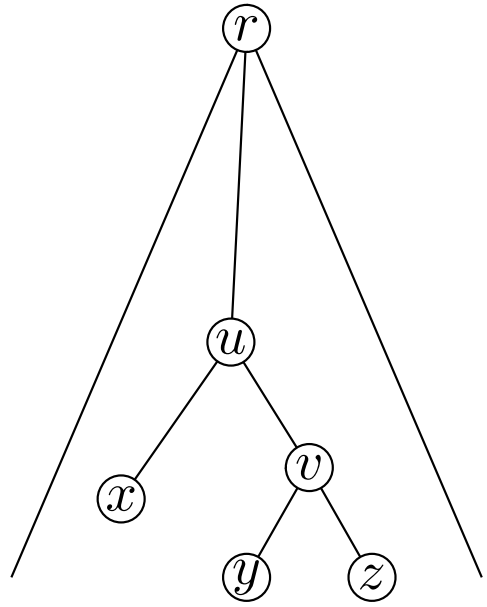


Step 1: Guess the branching structure of the variables (doable with FO).
$\Longrightarrow$ reduces the problem to formulas of the form $x < y \wedge \Psi(x, y)$

Step 2: Apply the acceleration technique for those formulas.
This is possibles because splits are computed deterministically.

MSO OVER TREES

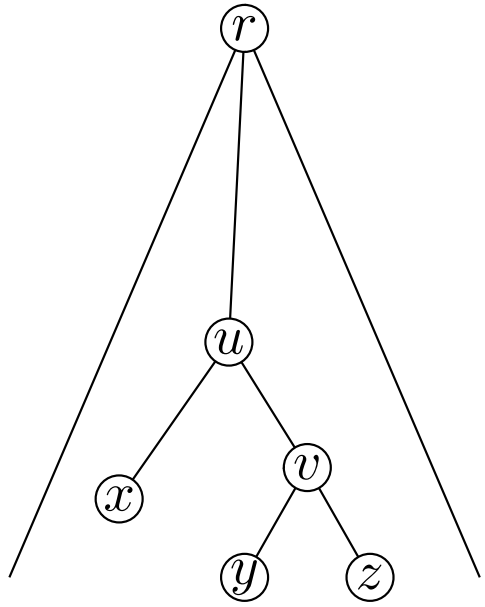
Problem: Given an MSO formula $\Phi(x_1, \ldots, x_n)$, transform it into a FO-formula, equivalent over trees, using $<$ and monadic formulas with a single free variable as predicates.



Theorem: Every MSO-interpretation is equivalent over trees to the composition of an FO-interpretation and an MSO-marking.

MSO OVER TREES

Problem: Given an MSO formula $\Phi(x_1, \ldots, x_n)$, transform it into a FO-formula, equivalent over trees, using $<$ and monadic formulas with a single free variable as predicates.



Theorem: Every MSO-interpretation is equivalent over trees to the composition of an FO-interpretation and an MSO-marking.

$$\mathcal{S} \leq_{MSO} t \qquad \leftrightarrow \qquad \exists t'.\ \mathcal{S} \leq_{FO} t' \leq^1_{MSO} t$$

APPLICATION FOR INFINITE STRUCTURES

Def: A prefix recognizable (relational) structure (Caucal 96) is a structure isomorphic to the MSO-interpretation of the infinite binary tree:

$$\mathcal{S} \leq_{MSO} \Delta$$

APPLICATION FOR INFINITE STRUCTURES

Def: A prefix recognizable (relational) structure (Caucal 96) is a structure isomorphic to the MSO-interpretation of the infinite binary tree:

$$\mathcal{S} \leq_{MSO} \Delta$$

Theorem: A structure is prefix recognizable iff it is isomorphic to an FO-interpretation of the infinite binary tree ($\mathcal{S} \leq_{FO} \Delta$).

APPLICATION FOR INFINITE STRUCTURES

Def: A prefix recognizable (relational) structure (Caucal 96) is a structure isomorphic to the MSO-interpretation of the infinite binary tree:

$$\mathcal{S} \leq_{MSO} \Delta$$

Theorem: A structure is prefix recognizable iff it is isomorphic to an FO-interpretation of the infinite binary tree ($\mathcal{S} \leq_{FO} \Delta$).

Lemma: Every regular binary tree is (up to isomorphism) FO-interpretable in the infinite binary tree.

APPLICATION FOR INFINITE STRUCTURES

Def: A prefix recognizable (relational) structure (Caucal 96) is a structure isomorphic to the MSO-interpretation of the infinite binary tree:

$$\mathcal{S} \leq_{MSO} \Delta$$

Theorem: A structure is prefix recognizable iff it is isomorphic to an FO-interpretation of the infinite binary tree ($\mathcal{S} \leq_{FO} \Delta$).

Lemma: Every regular binary tree is (up to isomorphism) FO-interpretable in the infinite binary tree.

$$\begin{aligned}\mathcal{S} &\leq_{MSO} \Delta \\ &\leq_{FO} t' \leq^1_{MSO} \Delta \\ &\leq_{FO} t' \leq_{FO} \Delta \\ &\leq_{FO} \Delta\end{aligned}$$

AUTOMATA OVER INFINITE OBJECTS

Remark: Splits values act in a way very ressemblant to parities in a parity game.

AUTOMATA OVER INFINITE OBJECTS

Remark: Splits values act in a way very ressemblant to parities in a parity game.

Theorem(McNaughton 66): Every Büchi automaton is equivalent to a deterministic parity automaton.

Proof: Direct application of the deterministic factorisation theorem.

AUTOMATA OVER INFINITE OBJECTS

Remark: Splits values act in a way very ressemblant to parities in a parity game.

Theorem(McNaughton 66): Every Büchi automaton is equivalent to a deterministic parity automaton.

Proof: Direct application of the deterministic factorisation theorem.

Question: What about Rabin's theorem?

SUMMARY

- A theorem of factorizations deterministicaly computable by finite state machines
  (Extension to every ordinal, even uncountable)
- A form of equivalence between MSO and FO queries
- Applications to prefix recognizable structures
  And the pushdown hierarchy

Thanks !