

Factorization forests and applications

Stefano Pessotto

University of Udine

Context: efficient query evaluation

Infix evaluation query

Let $a = (a_1, a_2, \dots, a_n)$ be a sequence of natural numbers.

Given in input two indices $1 \leq i \leq j \leq n$, what is $a_i \cdot \dots \cdot a_j$?

Context: efficient query evaluation

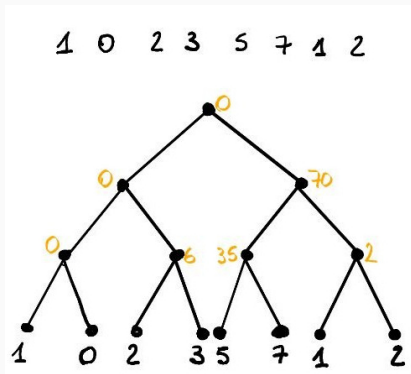
Infix evaluation query

Let $a = (a_1, a_2, \dots, a_n)$ be a sequence of natural numbers.
Given in input two indices $1 \leq i \leq j \leq n$, what is $a_i \cdot \dots \cdot a_j$?

You are allowed to preprocess a

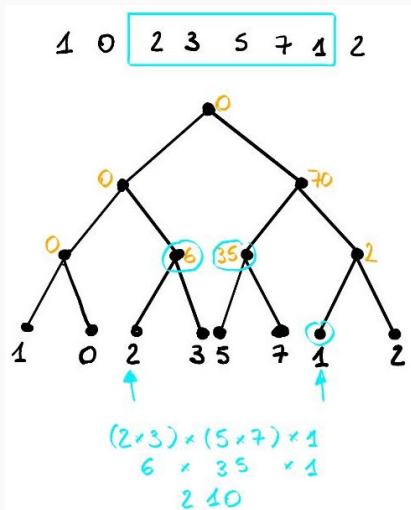
Infix evaluation - Segment trees

We use a binary splitting rule and evaluate \cdot on the subtrees



Infix evaluation - Segment trees

We use a binary splitting rule and evaluate \cdot on the subtrees



Infix evaluation - Segment trees

- Build a (balanced) binary tree of logarithmic height
- Every node represents an interval
- Internal nodes contain precomputed values

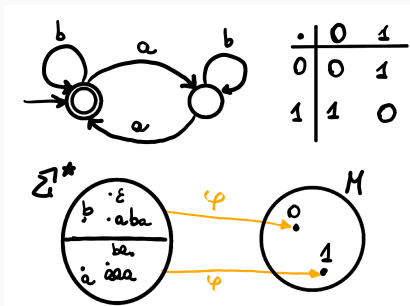
The only requirement is **associativity** of \cdot .

Preprocessing: $\mathcal{O}(n)$

Interval query: $\mathcal{O}(\log_2 n)$

Monoids

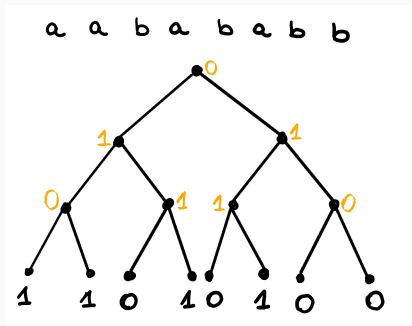
Recall: regular languages are described by finite monoids.



Infix evaluation - Monoids

Let $w \in \mathcal{L}$, (M, \cdot) the corresponding monoid, $\varphi : \Sigma^* \rightarrow M$.

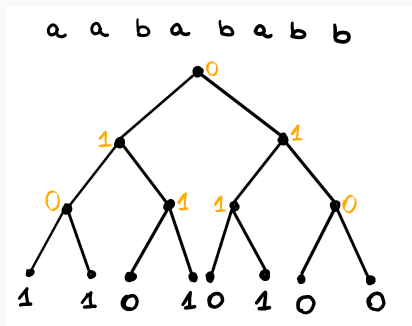
We can build segment trees for the products in M :



Infix evaluation - Monoids

Let $w \in \mathcal{L}$, (M, \cdot) the corresponding monoid, $\varphi : \Sigma^* \rightarrow M$.

We can build segment trees for the products in M :



Does the infix belong to the language?

Preprocessing: $\mathcal{O}(n)$

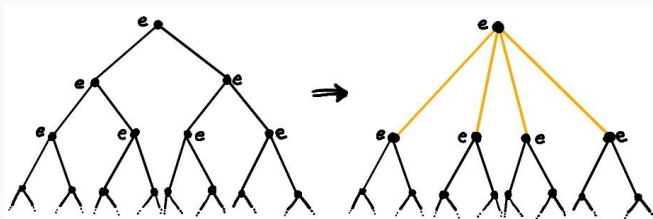
Interval query: $\mathcal{O}(\log_2 n)$

Monoids - idempotents

Can we achieve a better result than a tree of logarithmic height?

We can exploit patterns of idempotent elements:

For every $m \in M$ there exists a power p such that $m^p = (m^p)^2$



Ramsey tree

A tree of w is Ramsey if every internal node has either

- exactly two children (a binary node)
- all children labelled by the same idempotent element of M .

Factorization Forest theorem

Ramsey tree

A tree of w is Ramsey if every internal node has either

- exactly two children (a binary node)
- all children labelled by the same idempotent element of M .

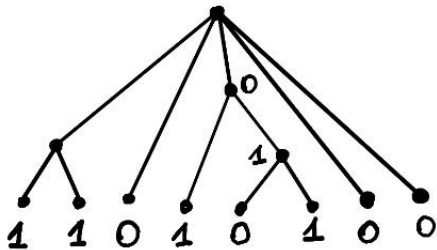
Factorization forest theorem

Let (M, \cdot) be a finite monoid and $\varphi : \Sigma^* \rightarrow M$ a morphism.

For all words $w \in \Sigma^*$ there exists a Ramsey tree with height at most $3|M| - 1$.

Example

a a b a b a b b



.	0	1
0	0	1
1	1	0

Infix evaluation

The tree has constant height wrt w , and assuming M is fixed:

Preprocessing: $\mathcal{O}(n)$

Interval query: $\mathcal{O}(1)$ ★

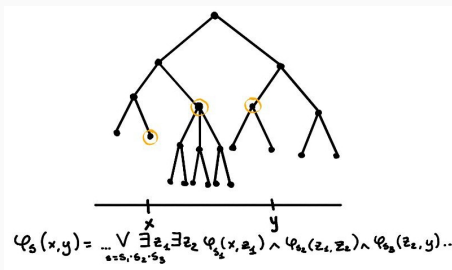
★ Can also be implemented logically in $FO[<]$.

Infix evaluation

The tree has constant height wrt w , and assuming M is fixed:

Preprocessing: $\mathcal{O}(n)$

Interval query: $\mathcal{O}(1)$ ★



★ Can also be implemented logically in $FO[<]$.

$w[x, y] \in \mathcal{L}$ is expressible in $FO[<]\star$

★ First-order logic cannot express counting properties, in general we would need monadic second-order logic!

$w[x, y] \in \mathcal{L}$ is expressible in $FO[<]^\star$

Theorem

Over strings, any MSO formula can be expressed as an initial MSO coloring followed by an FO formula on the annotated structure.

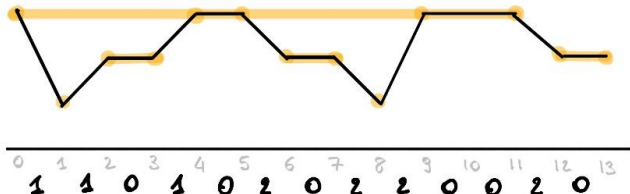
$$\Phi(x, y) = \Psi(x) \circ \varphi(x, y)$$

We should be able to annotate the input word with the tree, how do we do it efficiently?

★ First-order logic cannot express counting properties, in general we would need monadic second-order logic!

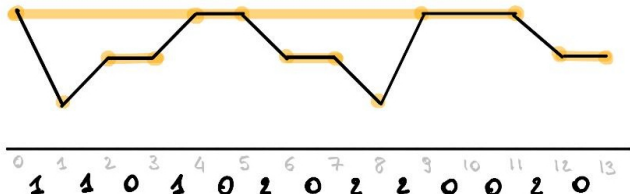
Splits

$(\mathbb{Z}/13\mathbb{Z}, +)$



Splits

$(\mathbb{Z}/13\mathbb{Z}, +)$

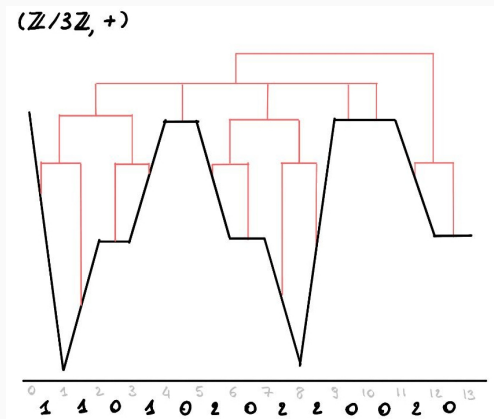


If x and y are at the same height and no elements in between is higher, then $\varphi(x \dots y) = e$.

The height is an annotation on the input.

Splits

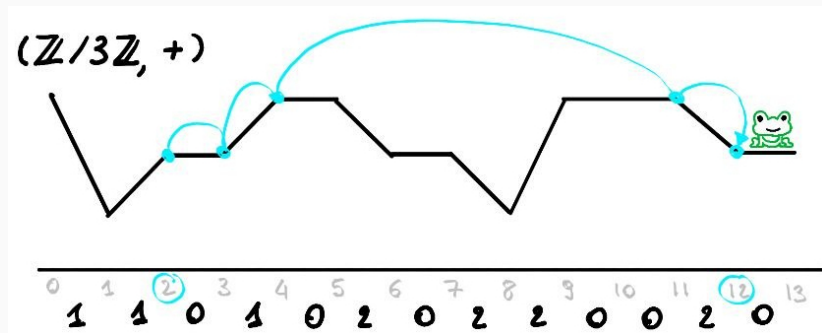
(It hides a factorization tree on the inside)



Splits

As for factorization forests, splits act as an accelerating structure

$$\varphi(2, 12) = \varphi(2, 3) \cdot \varphi(3, 4) \cdot \varphi(11, 12)$$



Theorem

Over finite strings, any MSO formula can be expressed as an initial MSO coloring followed by an FO formula on the annotated structure.

What about finite trees?

Theorem

Over finite strings, any MSO formula can be expressed as an initial MSO coloring followed by an FO formula on the annotated structure.

What about finite trees?

Idea: perform a factorization for every branch.

Issue: common prefix \neq common split (or coloring).

... we need a deterministic variant of FF...

Ramsey split

For all $x < y, x' < y'$ at the same height and with no uphill

$$\varphi(x, y) = \varphi(x', y') = e$$

Ramsey split

For all $x < y, x' < y'$ at the same height and with no uphill

$$\varphi(x, y) = \varphi(x', y') = e$$

Forward Ramsey split

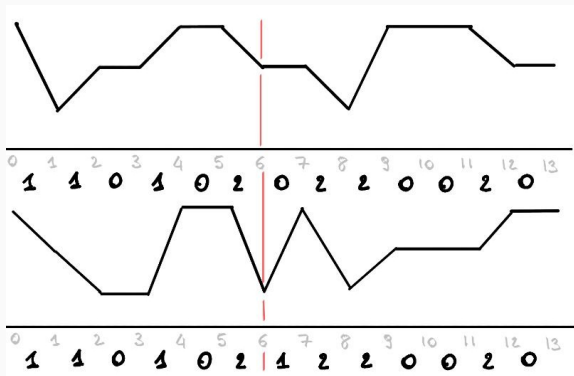
For all $x < y, x' < y'$ at the same height and with no uphill

$$\varphi(x, y) = \varphi(x, y) \cdot \varphi(x', y')$$

(The two are not the same, eg, when $\varphi(x, y) = a^0$ and $\varphi(x', y') = b^0$)

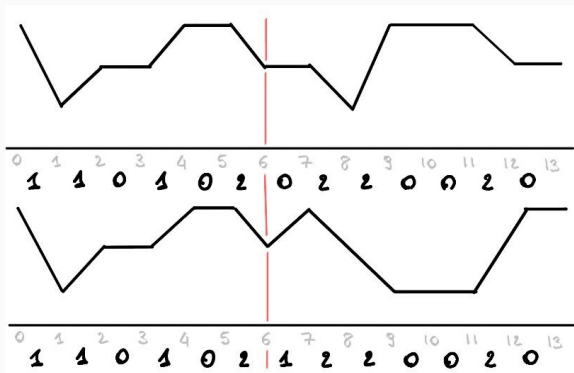
Forward splits

The standard construction is an induction on the structure of the monoid



Forward splits

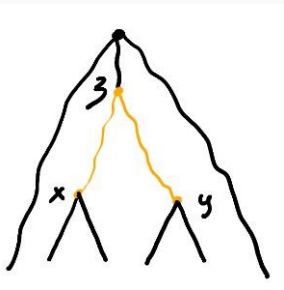
Non-determinism is removed with a left-to-right construction



Theorem

There exists a forward Ramsey split of height at most $|M|$ computable by a DFA.

The rest follows from compositionality of MSO.



Theorem

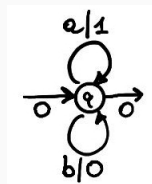
There exists a forward Ramsey split of height at most $|M|$ computable by a DFA.

Theorem

Over finite trees, any MSO formula can be expressed as an initial MSO coloring followed by an FO formula on the annotated structure.

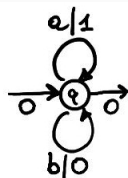
Min-cost automata

Automata where each run has an associated cost
The cost of w is the minimum between its run



Min-cost automata

Automata where each run has an associated cost
The cost of w is the minimum between its run



Limitedness problem

Let \mathcal{A} be a min-cost automaton.

Is there an upper bound $k \in \mathbb{N}$ for all possible costs of \mathcal{A} ?

Related to the star-height problem and generalized to a domination problem

Min-cost automata

Instead of the Min-Plus semiring, we consider the matrix semigroup with products done as in distance matrix product

Problem: The semigroup is not finite \rightarrow No FF theorem

$$(M, \cdot) \quad \uparrow \\ p \rightarrow \left[\square \right] \in (\mathbb{N} \cup \{\infty\})^{n \times n}$$

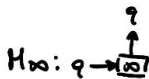
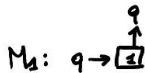
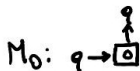
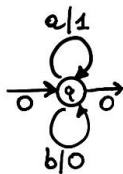
Min-cost automata

Instead of the Min-Plus semiring, we consider the matrix semigroup with products done as in distance matrix product

Solution: Approximation with a finite abstract semigroup

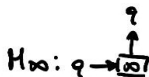
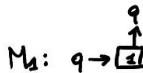
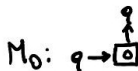
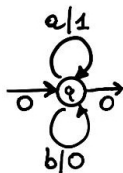
$$\begin{array}{l} (M, \cdot) \quad \dagger \\ \downarrow \rho \rightarrow \left[\square \right] \in (\mathbb{N} \cup \{\infty\})^{n \times n} \\ \widetilde{(M, \cdot)} \quad \dagger \\ \rho \rightarrow \left[\square \right] \in (\{0, 1, \infty\})^{n \times n} \end{array}$$

Min-cost automata



•	M_0	M_1	M_∞
M_0	M_0	M_1	M_∞
M_1	M_1	M_1	M_∞
M_∞	M_∞	M_∞	M_∞

Min-cost automata



\cdot	M_0	M_1	M_∞	#	
M_0	M_0	M_1	M_∞	M_0	M_0
M_1	M_1	M_1	M_∞	M_1	M_∞
M_∞	M_∞	M_∞	M_∞	M_∞	M_∞

$$e^\# = e \cdot e \cdot e \dots$$

Fixpoint algorithm for the limitedness problem

Let $\varphi : \Sigma \rightarrow M$ and $\tilde{\varphi} : \Sigma \rightarrow \tilde{M}$

1. initialize $S = \{\tilde{\varphi}(\sigma) \mid \sigma \in \Sigma\}$
2. fixpoint on $S = S \cup \{s \cdot t \mid s, t \in S\} \cup \{s^\# \mid s \in S, s \cdot s = s\}$
3. verify if $s[i, f] = \infty$ for some $s \in S, i \in I, f \in F$

Min-cost automata

Fixpoint algorithm for the limitedness problem

Let $\varphi : \Sigma \rightarrow M$ and $\tilde{\varphi} : \Sigma \rightarrow \tilde{M}$

1. initialize $S = \{\tilde{\varphi}(\sigma) \mid \sigma \in \Sigma\}$
2. fixpoint on $S = S \cup \{s \cdot t \mid s, t \in S\} \cup \{s^\# \mid s \in S, s \cdot s = s\}$
3. verify if $s[i, f] = \infty$ for some $s \in S, i \in I, f \in F$

Termination: \tilde{M} is finite

Correctness?

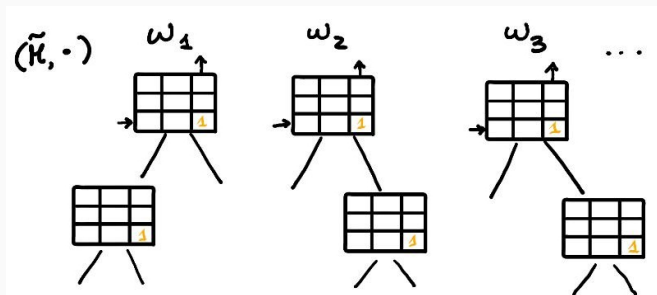
From $s[i, f] = \infty$, we can recover the witness.

What if $s[i, f]$ is always not ∞ ?

By contradiction, let w_1, w_2, \dots be a (strictly) increasing and infinite sequence in M and consider the factorization forest over \tilde{M}
By the pidgeon-hole principle, we can find an infinite sequence that maps to the same abstract element

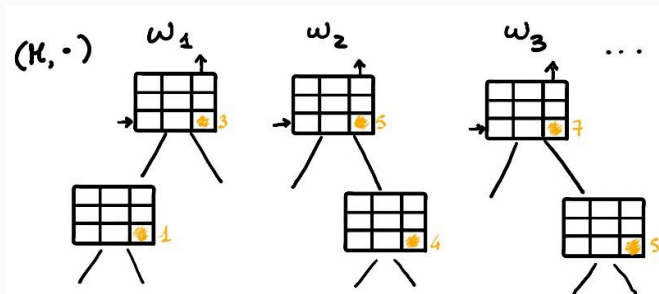
Min-cost automata

By contradiction, let w_1, w_2, \dots be a (strictly) increasing and infinite sequence in M and consider the factorization forest over \tilde{M}



Min-cost automata

In the concrete semigroup, it is actually increasing

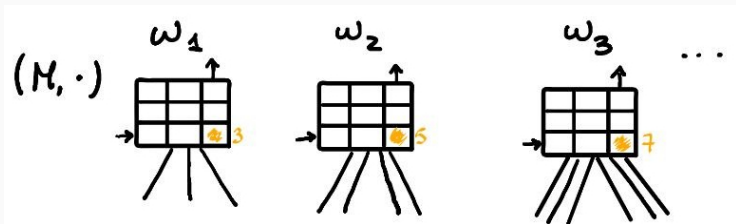


The height of the tree is bounded but there is a sequence of infinite distinct trees

We can find an infinite sequence where the increase is given by the number of branches in an idempotent node

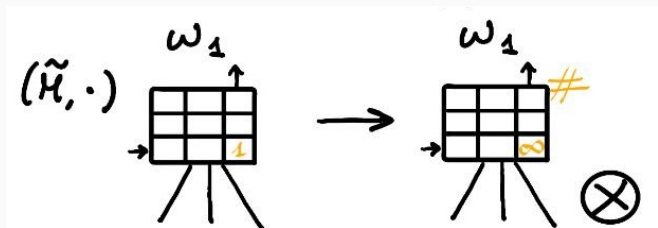
Min-cost automata

The height of the tree is bounded but there is a sequence of infinite distinct trees



Min-cost automata

... but the fixpoint algorithm applies # to idempotents!



1. Colcombet, T. (2021). The factorisation forest theorem. Handbook of Automata Theory.
2. Colcombet, T. (2013). Regular Cost Functions, Part I: Logic and Algebra over Words. Logical Methods in Computer Science, vol 9.
3. Colcombet, T. (2011). Green's Relations and Their Use in Automata Theory. Lecture Notes in Computer Science, vol 6638.
4. Bojańczyk, M. (2009). Factorization Forests. Lecture Notes in Computer Science, vol 5583.