Computing with semigroups defined by presentations

Maria Tsalakou

PiWORKS Seminar

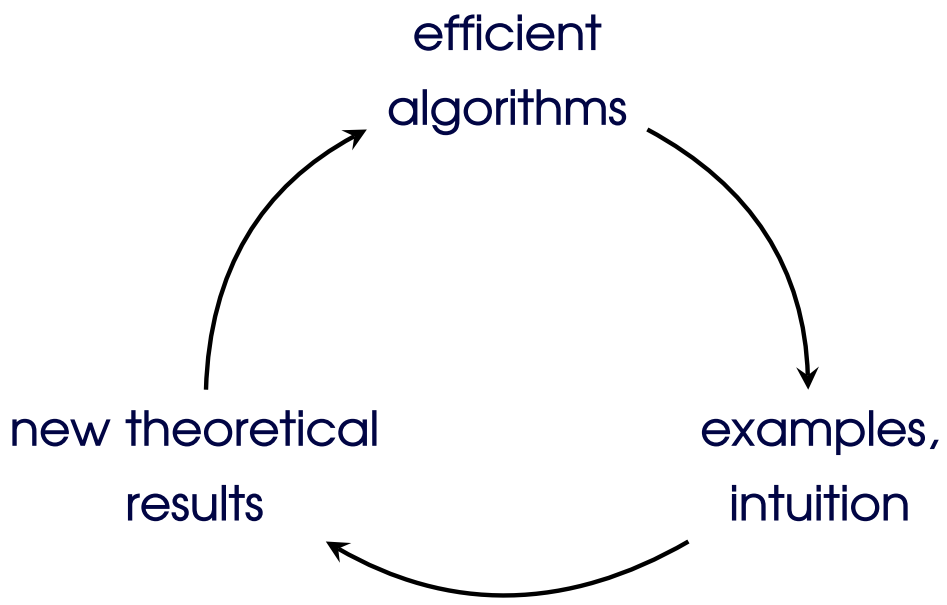
30 January 2024

My background

- BSc Mathematics - University of Cyprus (2013 - 2017)
- MSc Mathematics - University of St Andrews (2017 - 2018)
- Worked at a market research and consulting firm in Cyprus (2018 - 2019)
- PhD Mathematics (research on computational algebra) - University of St Andrews (2019 - 2023)

My research is focused on practical computation with finitely presented semigroups and monoids. I was interested in developing efficient algorithms for problems in semigroup theory.

Practical computation in pure mathematics



Semigroups

A **semigroup** is a set S together with an associative binary operation on S, i.e. an operation $* : S \times S \to S$ such that

$$(x * y) * z = x * (y * z)$$

for all $x, y, z \in S$.

- $(\mathbb{N}, +), (\mathbb{Z}, +)$
- Groups
- The set T_X of all functions from a set X to itself with operation the composition of functions.

Semigroups

A **semigroup** is a set S together with an associative binary operation on S, i.e. an operation $* : S \times S \to S$ such that

$$(x * y) * z = x * (y * z)$$

for all $x, y, z \in S$.

A **monoid** M is a semigroup with an identity element $e \in M$ such that

$$e * m = m * e = m$$

for all $m \in M$.

- $(\mathbb{N}, +), (\mathbb{Z}, +)$
- Groups
- The set T_X of all functions from a set X to itself with operation the composition of functions.
- $(\mathbb{N} \cup \{0\}, +)$

Why we are interested in semigroups

- In many cases mathematicians are interested in the set T_X of all functions from a set X to itself which is a semigroup.
- Applications in other areas of mathematics (PDEs, dynamical systems)
- Applications in biology
- Closely related to theoretical computer science (formal language theory, automata theory)
- They are *fun*.

The free monoid A^*

- A is a non-empty set, an **alphabet**

- $A = \{a, b, c\}$

The free monoid A^*

- A is a non-empty set, an **alphabet**
- elements of A^* are finite sequences of elements of A called **words**

- $A = \{a, b, c\}$
- $aab = a^2b$
  $abcaba$

The free monoid A^*

- A is a non-empty set, an **alphabet**
- elements of A^* are finite sequences of elements of A called **words**
- the operation is the concatenation of words

- $A = \{a, b, c\}$
- $aab = a^2b$
  $abcaba$
- $(bc) * (aba) = bcaba$

The free monoid A^*

- A is a non-empty set, an **alphabet**
- elements of A^* are finite sequences of elements of A called **words**
- the operation is the concatenation of words
- the **empty word** is the identity element

- $A = \{a, b, c\}$
- $aab = a^2b$
  $abcaba$
- $(bc) * (aba) = bcaba$
- $(\varepsilon) * (bc) = bc = (bc) * (\varepsilon)$

Monoids defined by presentations

A presentation for a monoid M is a pair $\langle A \mid R \rangle$, where:

- A is a non-empty set, an **alphabet**

- $A = \{a, b, c\}$

Monoids defined by presentations

A presentation for a monoid M is a pair $\langle A \mid R \rangle$, where:

- A is a non-empty set, an **alphabet**
- elements of M are represented by words in A^*

- $A = \{a, b, c\}$
- $aab = a^2b$
  $abcaba$

Monoids defined by presentations

A presentation for a monoid M is a pair $\langle A \mid R \rangle$, where:

- A is a non-empty set, an **alphabet**
- elements of M are represented by words in A^*
- R is a set of **relations**

- $A = \{a, b, c\}$
- $aab = a^2b$
  $abcaba$
- $R = \{ba = ac,\ cbc = cb\}$

Monoids defined by presentations

A presentation for a monoid M is a pair $\langle A \mid R \rangle$, where:

- A is a non-empty set, an **alphabet**
- elements of M are represented by words in A^*
- R is a set of **relations**
- two words represent the same element of M if we can get from one to the other by applying the relations in R.

- $A = \{a, b, c\}$
- $aab = a^2b$
  $abcaba$
- $R = \{ba = ac,\ cbc = cb\}$
- $acbc = acb = acb = bab$

The word problem for monoids

Suppose that $\langle A \mid R \rangle$ is a presentation for a monoid M. Does there exist an algorithm deciding whether or not two words u and v over the alphabet A represent the same element of M?

This problem is **undecidable** in general...

...but *almost always* decidable.

Small overlap monoids

- A family of monoids defined by presentations that satisfy a simple combinatorial condition, called C(m), $m \in \mathbb{N}$.
- Small overlap monoids that satisfy C(m) with $m \geq 3$ are infinite and have decidable word problem.
- Small overlap monoids are the ``generic'' monoids defined by presentations.

Small overlap conditions

Let M be the monoid defined by $\langle A \mid R \rangle$.

A **piece** is any word in A^* that appears as a factor in two distinct relation words in R or as a factor in two different places (possibly overlapping) in one relation word in R. The empty word ε is a piece by convention.

Example:

$$\mathcal{P} = \langle a, b, c, d, e, f, g \mid a^3ea^2 = abcd, ef = dg \rangle$$

The pieces of $\mathcal{P}$ are:

- ε (by convention)
- a: $aaaeaa$ and $abcd$
- d: $abcd$ and dg
- e: a^3ea^2 and ef
- a^2: $aaaeaa$, $aaaeaa$

The C(n) condition for monoids

A presentation $\mathcal{P} = \langle A \mid R \rangle$ satisfies condition $C(n)$, $n \in \mathbb{N}$ if no relation word in R can be written as a product of strictly less than n pieces.

If $\mathcal{P}$ satisfies $C(n)$ for some $n \in \mathbb{N}$ then it satisfies $C(k)$ for all $k \leq n$.

Example

The set of pieces of $\langle a, b, c, d, e, f, g \mid a^3ea^2 = abcd, ef = dg \rangle$ is $P = \{\varepsilon, a, d, e, a^2\}$.

$$\underbrace{aaaeaa}_{\text{4 pieces}} \qquad abcd \qquad ef \qquad dg$$

The presentation satisfies $C(4)$.

The word problem in $C(3)$ and $C(4)$ monoids

Theorem (Remmers 1971)

The word problem is decidable for monoids defined by presentations satisfying the small overlap condition $C(3)$.

Theorem (Kambites 2009)

For every monoid presentation satisfying $C(4)$, there exists an algorithm which solves the corresponding word problem in time linear in the lengths of the input words.

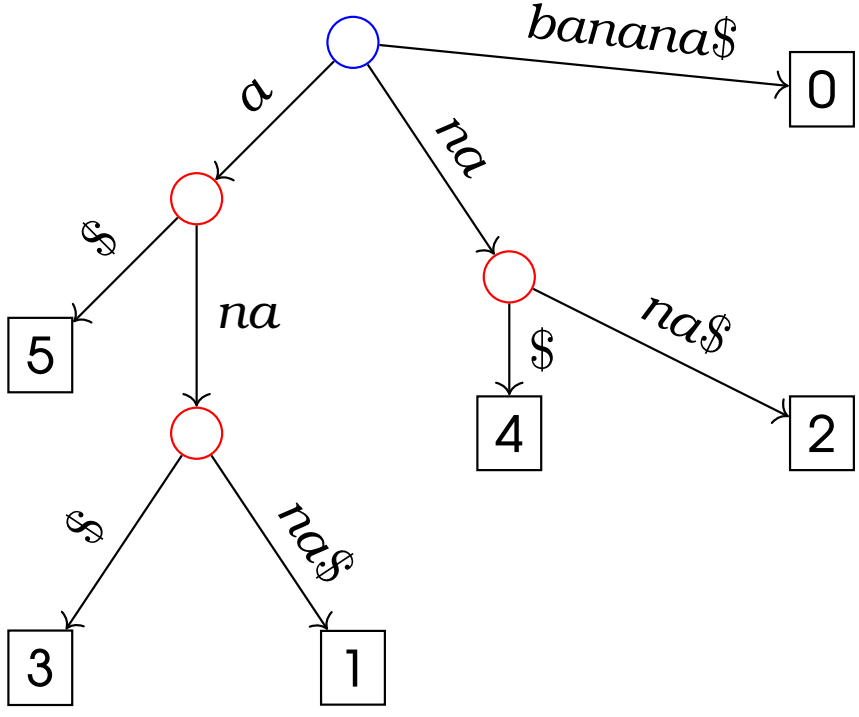
We want to have an efficient algorithm that will determine if a presentation satisfies $C(3)$ or $C(4)$.

Deciding if a presentation satisfies C(m)

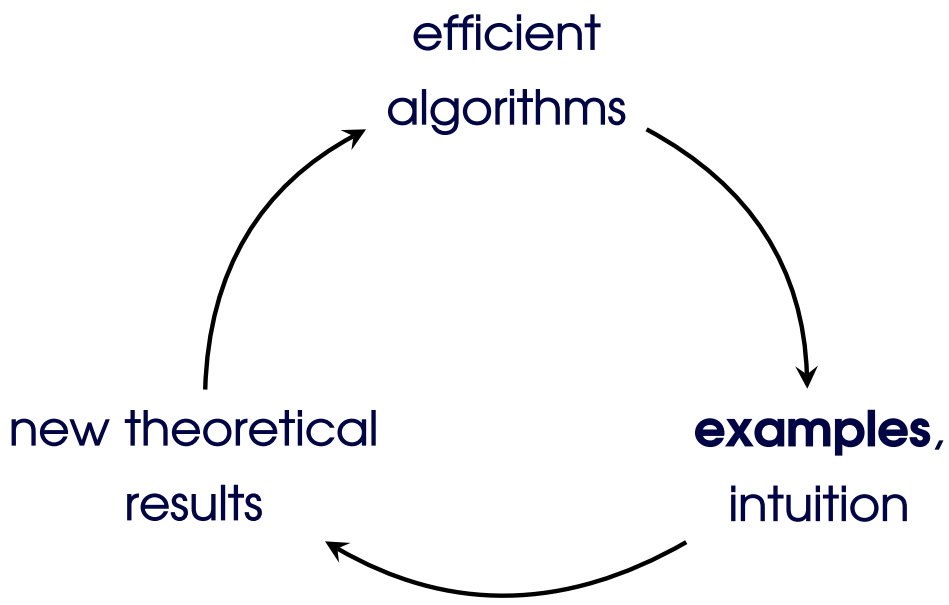
We can use a data structure called **suffix tree**, in order to find the pieces of a presentation.

Suffix tree for $w = banana\$$

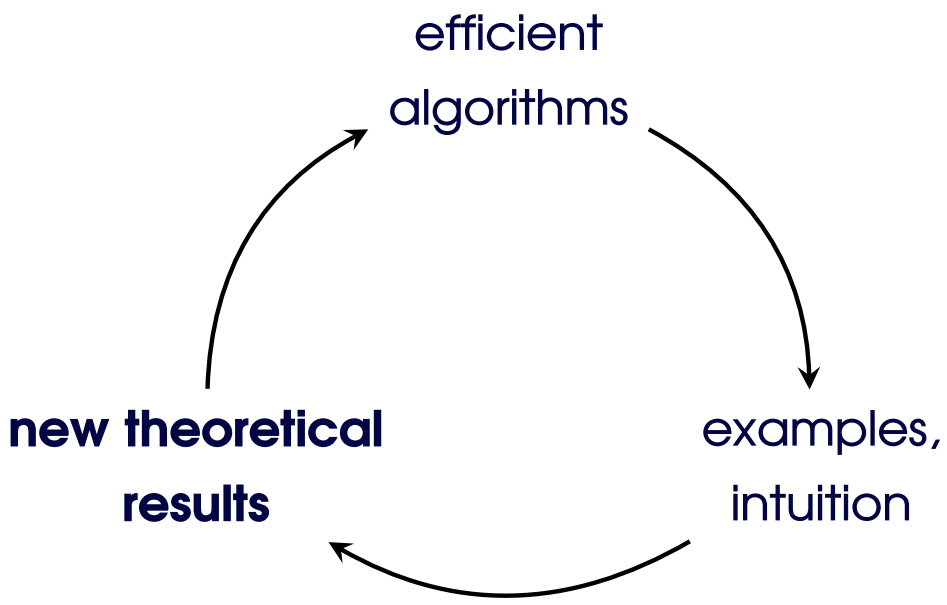
- each leaf node corresponds to a suffix of w
- if v is a prefix of the label of a path from the root to an internal node, then v occurs more than once in w
- it can be constructed in time linear to the length of w



Practical computation in pure mathematics



Practical computation in pure mathematics



PhD experience

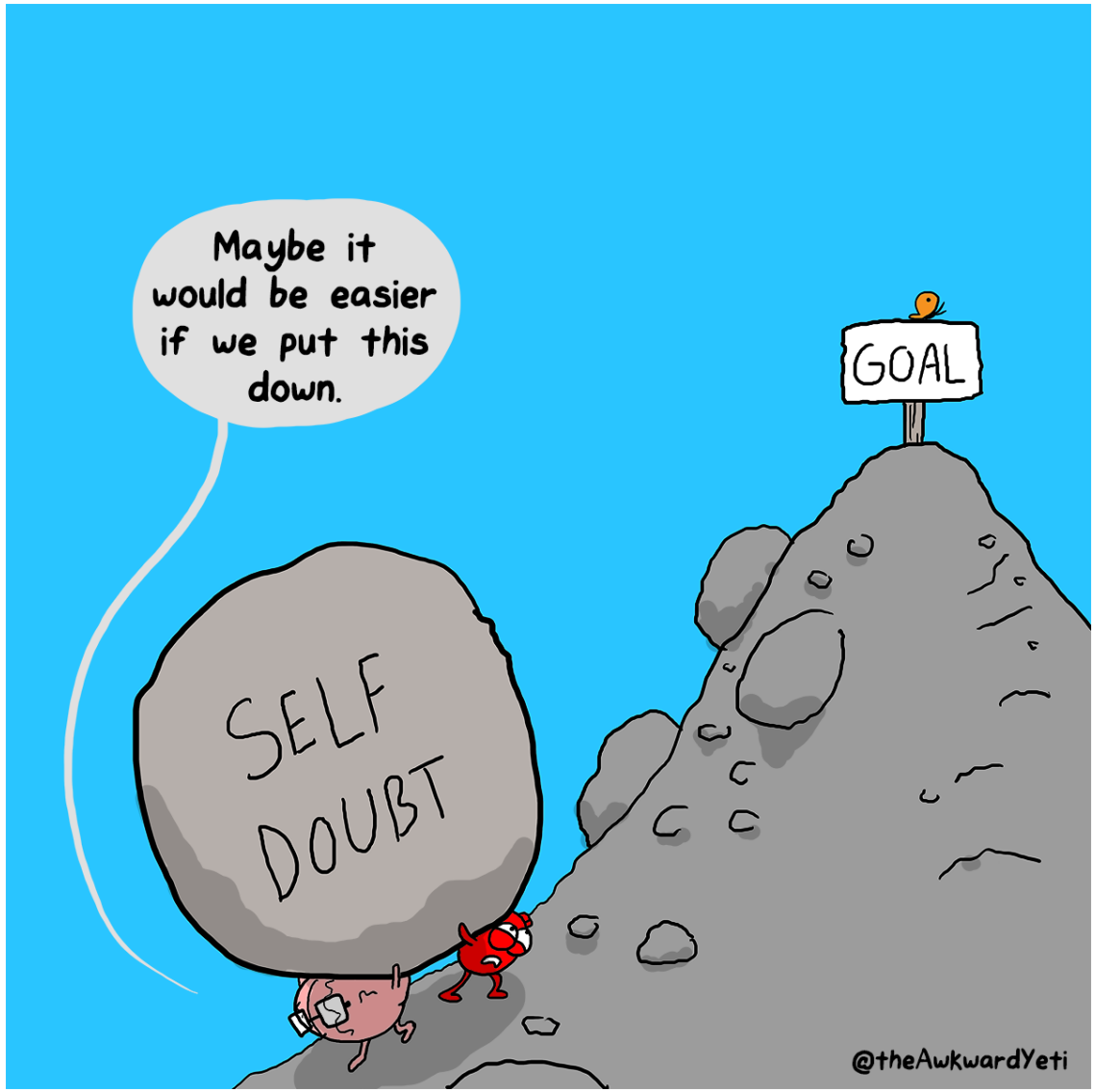


Figure: https://theawkwardyeti.com/comic/burden/.

Dealing with self doubt during your PhD

- If you have been accepted into a PhD program, you have the skills to finish it
- You might feel like your work is less important/easier than everyone else's
  - People talk about their work in seminars and conferences
  - You know your own work very well
- You can ask people you trust for feedback (and you should believe them when they say something good!)
- You don't need to be the best, you need to be interested in bringing something new to the table

Thank you!