On the average complexity of the word problem in subgroups of integral invertible matrices

Frédérique Bassino

LIPN, Université Sorbonne Paris Nord

MAD Days, June 18-20 2025, Rouen

Joint works with Cyril Nicaud (LIGM, Université Gustave Eiffel) & Pascal Weil (CNRS, LIPN & Université Sorbonne Paris Nord)

– The Word Problem –

- The word problem is the problem of **deciding** whether two given expressions are equivalent with respect to a set of rewriting identities (*e.g*, a set of relators).
- This problem is mainly studied in (semi)group theory.

– The Word Problem –

- The word problem is the problem of **deciding** whether two given expressions are equivalent with respect to a set of rewriting identities (*e.g*, a set of relators).
- This problem is mainly studied in (semi)group theory.

The Word Problem in groups (Dehn 1911)

Let Σ be a finite subset of a group G, is it **decidable** whether a finite word w on $\tilde{\Sigma} = \Sigma \cup \Sigma^{-1}$ evaluates to 1 in G?

– The Word Problem –

- The word problem is the problem of **deciding** whether two given expressions are equivalent with respect to a set of rewriting identities (*e.g*, a set of relators).
- This problem is mainly studied in (semi)group theory.

The Word Problem in groups (Dehn 1911)

Let Σ be a finite subset of a group G, is it **decidable** whether a finite word w on $\tilde{\Sigma} = \Sigma \cup \Sigma^{-1}$ evaluates to 1 in G?

- This problem is **not decidable** in general even for finitely presented groups (Novikov 1955, Boone 1959).

– The Word Problem –

- The word problem is the problem of **deciding** whether two given expressions are equivalent with respect to a set of rewriting identities (*e.g*, a set of relators).
- This problem is mainly studied in (semi)group theory.

The Word Problem in groups (Dehn 1911)

Let Σ be a finite subset of a group G, is it **decidable** whether a finite word w on $\tilde{\Sigma} = \Sigma \cup \Sigma^{-1}$ evaluates to 1 in G?

- This problem is **not decidable** in general even for finitely presented groups (Novikov 1955, Boone 1959).
- It is **decidable** for automatic groups inluding finite, free, hyperbolic or braid groups (Epstein *et al.* 1992); 1-relator groups (Magnus, Karass and Solitar 1966)...

– The Word Problem in $\mathsf{GL}_d(\mathbb{Z})$ –

- Let $\mathsf{GL}_d(\mathbb{Z})$ be the set of $d \times d$ invertible matrices with coefficients in $\mathbb{Z}$. The integer d **is fixed**.

– The Word Problem in $\mathsf{GL}_d(\mathbb{Z})$ –

- Let $\mathsf{GL}_d(\mathbb{Z})$ be the set of $d \times d$ invertible matrices with coefficients in $\mathbb{Z}$. The integer d **is fixed**.
- Let Σ be a nonempty **finite subset** of $\mathsf{GL}_d(\mathbb{Z})$.

– The Word Problem in $\mathsf{GL}_d(\mathbb{Z})$ –

- Let $\mathsf{GL}_d(\mathbb{Z})$ be the set of $d \times d$ invertible matrices with coefficients in $\mathbb{Z}$. The integer d **is fixed**.
- Let Σ be a nonempty **finite subset** of $\mathsf{GL}_d(\mathbb{Z})$.

The Word Problem in the subgroup generated by Σ

Given a finite word w on $\tilde{\Sigma} = \Sigma \cup \Sigma^{-1}$, is the (matrix) evaluation $\mathsf{M}(w)$ of w in $\mathsf{GL}_d(\mathbb{Z})$ equal Id ?

– The Word Problem in $\mathsf{GL}_d(\mathbb{Z})$ –

- Let $\mathsf{GL}_d(\mathbb{Z})$ be the set of $d \times d$ invertible matrices with coefficients in $\mathbb{Z}$. The integer d **is fixed**.
- Let Σ be a nonempty **finite subset** of $\mathsf{GL}_d(\mathbb{Z})$.

The Word Problem in the subgroup generated by Σ

Given a finite word w on $\tilde{\Sigma} = \Sigma \cup \Sigma^{-1}$, is the (matrix) evaluation $\mathsf{M}(w)$ of w in $\mathsf{GL}_d(\mathbb{Z})$ equal Id ?

- This problem is of course decidable in $\mathsf{GL}_d(\mathbb{Z})$.

– The Word Problem in $\mathsf{GL}_d(\mathbb{Z})$ –

- Let $\mathsf{GL}_d(\mathbb{Z})$ be the set of $d \times d$ invertible matrices with coefficients in $\mathbb{Z}$. The integer d **is fixed**.
- Let Σ be a nonempty **finite subset** of $\mathsf{GL}_d(\mathbb{Z})$.

The Word Problem in the subgroup generated by Σ

Given a finite word w on $\tilde{\Sigma} = \Sigma \cup \Sigma^{-1}$, is the (matrix) evaluation $\mathsf{M}(w)$ of w in $\mathsf{GL}_d(\mathbb{Z})$ equal Id ?

- This problem is of course decidable in $\mathsf{GL}_d(\mathbb{Z})$.
- But what is its complexity?

– Bit complexity –

Bit complexity

The bit complexity is the number of operations on bits that are needed for running an algorithm.

– Bit complexity –

Bit complexity

The bit complexity is the number of operations on bits that are needed for running an algorithm.

- Integers are identified with their binary expansion.

– Bit complexity –

Bit complexity

The bit complexity is the number of operations on bits that are needed for running an algorithm.

- Integers are identified with their binary expansion.
- The bit size $\ell(m)$ of m is $\lceil \log(|m|+1) \rceil + 1$ (with the sign).

– Bit complexity –

Bit complexity

The bit complexity is the number of operations on bits that are needed for running an algorithm.

- Integers are identified with their binary expansion.
- The bit size $\ell(m)$ of m is $\lceil \log(|m|+1) \rceil + 1$ (with the sign).
- Coefficients of $\mathsf{M}(w)$ grow at most exponentially in $|w| \longrightarrow$ their bit sizes grow at most linearly in $|w|$.

– Bit complexity –

Bit complexity

The bit complexity is the number of operations on bits that are needed for running an algorithm.

- Integers are identified with their binary expansion.
- The bit size $\ell(m)$ of m is $\lceil \log(|m|+1) \rceil + 1$ (with the sign).
- Coefficients of $\mathsf{M}(w)$ grow at most exponentially in $|w| \longrightarrow$ their bit sizes grow at most linearly in $|w|$.

Theorem (Harvey and van der Hoeven 2021)

If $\ell(p), \ell(q) \leq L$, then pq is computed in $\mathcal{O}(L \log L)$.

– Naive algorithm –

Naive algorithm

If $w = a_1 \ldots a_n$ (each $a_i \in \tilde{\Sigma}$)

- Compute the $n-1$ products $w_0 = \mathsf{Id}$, $w_{i+1} = w_i\, a_{i+1}$, …, $w_n = \mathsf{M}(w)$,
- Check whether M(w) is Id.

- In the worst case:
  - The length of the coefficients in $\mathsf{M}(w)$ grows linearly in $n = |w|$.

– Naive algorithm –

Naive algorithm

If $w = a_1 \dots a_n$ (each $a_i \in \tilde{\Sigma}$)

- Compute the $n-1$ products $w_0 = \mathsf{Id}$, $w_{i+1} = w_i\, a_{i+1}, \dots$, $w_n = \mathsf{M}(w)$,
- Check whether M(w) is Id.

- In the worst case:
  - The length of the coefficients in $\mathsf{M}(w)$ grows linearly in $n = |w|$.
  - The cost of each multiplication is $\mathcal{O}(n \log n)$.

Naive algorithm

If $w = a_1 \dots a_n$ (each $a_i \in \tilde{\Sigma}$)

- Compute the $n-1$ products $w_0 = \mathsf{Id}$, $w_{i+1} = w_i\, a_{i+1}, \dots$, $w_n = \mathsf{M}(w)$,
- Check whether M(w) is Id.

- In the worst case:
  - The length of the coefficients in $\mathsf{M}(w)$ grows linearly in $n = |w|$.
  - The cost of each multiplication is $\mathcal{O}(n \log n)$.
  - This algorithm computes $\mathsf{M}(w)$ in $\mathcal{O}(n^2 \log n)$.

– Naive algorithm –

Naive algorithm

If $w = a_1 \ldots a_n$ (each $a_i \in \tilde{\Sigma}$)

- Compute the $n-1$ products $w_0 = \mathsf{Id}$, $w_{i+1} = w_i\, a_{i+1}$, ..., $w_n = \mathsf{M}(w)$,
- Check whether M(w) is Id.

- In the worst case:
  - The length of the coefficients in $\mathsf{M}(w)$ grows linearly in $n = |w|$.
  - The cost of each multiplication is $\mathcal{O}(n \log n)$.
  - This algorithm computes $\mathsf{M}(w)$ in $\mathcal{O}(n^2 \log n)$.
  - Checking whether $\mathsf{M}(w)$ is Id is done in constant time.

– Naive algorithm –

Naive algorithm

If $w = a_1 \dots a_n$ (each $a_i \in \tilde{\Sigma}$)

- Compute the $n - 1$ products $w_0 = \mathsf{Id}$, $w_{i+1} = w_i \, a_{i+1}$, …, $w_n = \mathsf{M}(w)$,
- Check whether M(w) is Id.

- In the worst case:
  - The length of the coefficients in $\mathsf{M}(w)$ grows linearly in $n = |w|$.
  - The cost of each multiplication is $\mathcal{O}(n \log n)$.
  - This algorithm computes $\mathsf{M}(w)$ in $\mathcal{O}(n^2 \log n)$.
  - Checking whether $\mathsf{M}(w)$ is Id is done in constant time.
  - The complexity of this naive algorithm is in $\mathcal{O}(n^2 \log n)$.

– A divide-and-conquer algorithm –

Algorithm 1: Algorithm DC_Σ

Input : a sequence w of n elements of $\tilde{\Sigma}$

Output: $\mathsf{M}(w)$

if $n = 0$ (resp. $n = 1$) **then**

  return Id (resp. $\mathsf{M}(w)$)

$w_1 \leftarrow$ prefix of w of length $\lfloor n/2 \rfloor$

$w_2 \leftarrow$ suffix of w of length $\lceil n/2 \rceil$

return $\mathsf{DC}_\Sigma(w_1) \times \mathsf{DC}_\Sigma(w_2)$

– A divide-and-conquer algorithm –

Algorithm 2: Algorithm DC_Σ

Input : a sequence w of n elements of $\tilde{\Sigma}$

Output: $\mathsf{M}(w)$

```
if n = 0 (resp. n = 1) then
    return Id (resp. M(w))
w_1 ← prefix of w of length ⌊n/2⌋
w_2 ← suffix of w of length ⌈n/2⌉
return DC_Σ(w_1) × DC_Σ(w_2)
```

The **worst-case complexity** $C(n)$ satisfies the functional equation:
$C(n) = C(\lfloor \frac{n}{2} \rfloor) + C(\lceil \frac{n}{2} \rceil) + \text{cost-of-multiplying}(\mathsf{M}(w_1)\mathsf{M}(w_2))$.

– The Master Theorem –

The Master Theorem

Suppose $C(n)$ satisfies $C(n) = C(\lfloor \frac{n}{2} \rfloor) + C(\lceil \frac{n}{2} \rceil) + f(n)$.

- If $f(n) = \mathcal{O}(n^h)$ for some $h < 1$, then $C(n) = \mathcal{O}(n)$.
- If $f(n) = \mathcal{O}(n \log^h n)$ for $h \geq 0$, then $C(n) = \mathcal{O}(n \log^{h+1} n)$.

– The Master Theorem –

The Master Theorem

Suppose $C(n)$ satisfies $C(n) = C(\lfloor \frac{n}{2} \rfloor) + C(\lceil \frac{n}{2} \rceil) + f(n)$.

- If $f(n) = \mathcal{O}(n^h)$ for some $h < 1$, then $C(n) = \mathcal{O}(n)$.
- If $f(n) = \mathcal{O}(n \log^h n)$ for $h \geq 0$, then $C(n) = \mathcal{O}(n \log^{h+1} n)$.

- The length of the coefficients in $\mathsf{M}(w)$ grows linearly in $n = |w|$.

– The Master Theorem –

The Master Theorem

Suppose $C(n)$ satisfies $C(n) = C(\lfloor \frac{n}{2} \rfloor) + C(\lceil \frac{n}{2} \rceil) + f(n)$.

- If $f(n) = \mathcal{O}(n^h)$ for some $h < 1$, then $C(n) = \mathcal{O}(n)$.
- If $f(n) = \mathcal{O}(n \log^h n)$ for $h \geq 0$, then $C(n) = \mathcal{O}(n \log^{h+1} n)$.

- The length of the coefficients in $\mathsf{M}(w)$ grows linearly in $n = |w|$.
- The cost of multiplying $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ is $\mathcal{O}(n \log n)$.

– The Master Theorem –

The Master Theorem

Suppose $C(n)$ satisfies $C(n) = C(\lfloor \frac{n}{2} \rfloor) + C(\lceil \frac{n}{2} \rceil) + f(n)$.

- If $f(n) = \mathcal{O}(n^h)$ for some $h < 1$, then $C(n) = \mathcal{O}(n)$.
- If $f(n) = \mathcal{O}(n \log^h n)$ for $h \geq 0$, then $C(n) = \mathcal{O}(n \log^{h+1} n)$.

- The length of the coefficients in $\mathsf{M}(w)$ grows linearly in $n = |w|$.
- The cost of multiplying $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ is $\mathcal{O}(n \log n)$.

Worst case bit complexity (Olshanskii and Shpilrain 2025)

DC_Σ has worst case bit complexity $\mathcal{O}(n \log^2 n)$.

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,
  - DC_Σ has linear complexity.

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,
  - DC_Σ has linear complexity.
- If Σ contains **only upper-triangular matrices**,

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,
  - DC_Σ has linear complexity.
- If Σ contains **only upper-triangular matrices**,
  - the (i, j)-coefficients grow polynomially in n (in $\mathcal{O}(n^{j-i})$),

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,
  - DC_Σ has linear complexity.
- If Σ contains **only upper-triangular matrices**,
  - the (i, j)-coefficients grow polynomially in n (in $\mathcal{O}(n^{j-i})$),
  - so their lengths grow logarithmically in n,

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,
  - DC_Σ has linear complexity.
- If Σ contains **only upper-triangular matrices**,
  - the (i, j)-coefficients grow polynomially in n (in $\mathcal{O}(n^{j-i})$),
  - so their lengths grow logarithmically in n,
  - DC_Σ has linear complexity.

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,
  - DC_Σ has linear complexity.
- If Σ contains **only upper-triangular matrices**,
  - the (i, j)-coefficients grow polynomially in n (in $\mathcal{O}(n^{j-i})$),
  - so their lengths grow logarithmically in n,
  - DC_Σ has linear complexity.
  - Application to the Word Problem in **finitely generated nilpotent groups** (Olshanskii and Shpilrain 2025).

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,
  - DC_Σ has linear complexity.
- If Σ contains **only upper-triangular matrices**,
  - the (i, j)-coefficients grow polynomially in n (in $\mathcal{O}(n^{j-i})$),
  - so their lengths grow logarithmically in n,
  - DC_Σ has linear complexity.
  - Application to the Word Problem in **finitely generated nilpotent groups** (Olshanskii and Shpilrain 2025).
- The same algorithm in $\mathsf{GL}_d(\mathbb{Z}/m\mathbb{Z})$ computes the **mod** m **projection** of $\mathsf{M}(w)$, written $\mathsf{M}(w)_m$.

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,
  - DC_Σ has linear complexity.
- If Σ contains **only upper-triangular matrices**,
  - the (i, j)-coefficients grow polynomially in n (in $\mathcal{O}(n^{j-i})$),
  - so their lengths grow logarithmically in n,
  - DC_Σ has linear complexity.
  - Application to the Word Problem in **finitely generated nilpotent groups** (Olshanskii and Shpilrain 2025).
- The same algorithm in $\mathsf{GL}_d(\mathbb{Z}/m\mathbb{Z})$ computes the **mod** m **projection** of $\mathsf{M}(w)$, written $\mathsf{M}(w)_m$.
  - If m is fixed, multiplication $\mathsf{M}(w_1)_m \cdot \mathsf{M}(w_2)_m$ costs $\mathcal{O}(1)$,

– Special cases with linear worst-case complexity –

- If $H = \langle \Sigma \rangle$ is **finite**
  - the lengths of the coefficients of the matrices of H are bounded,
  - the multiplication $\mathsf{M}(w_1) \cdot \mathsf{M}(w_2)$ costs $\mathcal{O}(1)$,
  - DC_Σ has linear complexity.
- If Σ contains **only upper-triangular matrices**,
  - the (i, j)-coefficients grow polynomially in n (in $\mathcal{O}(n^{j-i})$),
  - so their lengths grow logarithmically in n,
  - DC_Σ has linear complexity.
  - Application to the Word Problem in **finitely generated nilpotent groups** (Olshanskii and Shpilrain 2025).
- The same algorithm in $\mathsf{GL}_d(\mathbb{Z}/m\mathbb{Z})$ computes the **mod** m **projection** of $\mathsf{M}(w)$, written $\mathsf{M}(w)_m$.
  - If m is fixed, multiplication $\mathsf{M}(w_1)_m \cdot \mathsf{M}(w_2)_m$ costs $\mathcal{O}(1)$,
  - DC_m computes $\mathsf{M}(w)_m$ in linear time.

– Towards an algorithm with $\mathcal{O}(n)$ average-case complexity –

- A natural idea:

– Towards an algorithm with $\mathcal{O}(n)$ average-case complexity –

- A natural idea:
  - Compute $\mathsf{M}(w)$ modulo q.

– Towards an algorithm with $\mathcal{O}(n)$ average-case complexity –

- A natural idea:
  - Compute $\mathsf{M}(w)$ modulo q.
  - If $\mathsf{M}(w)_q \neq \mathsf{Id}$, then $\mathsf{M}(w) \neq \mathsf{Id}$;
    else run Algorithm DC on w.

– Towards an algorithm with $\mathcal{O}(n)$ average-case complexity –

- A natural idea:
  - Compute $\mathsf{M}(w)$ modulo q.
  - If $\mathsf{M}(w)_q \neq \mathsf{Id}$, then $\mathsf{M}(w) \neq \mathsf{Id}$;
    else run Algorithm DC on w.
- For a fixed q, the probability of $\mathsf{M}(w)_q = \mathsf{Id}$ may tend to a constant, so the average-case complexity is still $\mathcal{O}(n \log^2 n)$.

– Towards an algorithm with $\mathcal{O}(n)$ average-case complexity –

- A natural idea:
  - Compute $\mathsf{M}(w)$ modulo q.
  - If $\mathsf{M}(w)_q \neq \mathsf{Id}$, then $\mathsf{M}(w) \neq \mathsf{Id}$;
    else run Algorithm DC on w.
- For a fixed q, the probability of $\mathsf{M}(w)_q = \mathsf{Id}$ may tend to a constant, so the average-case complexity is still $\mathcal{O}(n \log^2 n)$.
- So: take $q = q(n)$, a function of the length of w.

– Towards an algorithm with $\mathcal{O}(n)$ average-case complexity –

- A natural idea:
  - Compute $\mathsf{M}(w)$ modulo q.
  - If $\mathsf{M}(w)_q \neq \mathsf{Id}$, then $\mathsf{M}(w) \neq \mathsf{Id}$;
    else run Algorithm DC on w.
- For a fixed q, the probability of $\mathsf{M}(w)_q = \mathsf{Id}$ may tend to a constant, so the average-case complexity is still $\mathcal{O}(n \log^2 n)$.
- So: take $q = q(n)$, a function of the length of w.
- Arithmetic operations in $\mathbb{Z}/q(n)\mathbb{Z}$ cost $\mathcal{O}(\log q(n)\ \log\log q(n))$.

– Towards an algorithm with $\mathcal{O}(n)$ average-case complexity –

- A natural idea:
  - Compute $\mathsf{M}(w)$ modulo q.
  - If $\mathsf{M}(w)_q \neq \mathsf{Id}$, then $\mathsf{M}(w) \neq \mathsf{Id}$;
    else run Algorithm DC on w.
- For a fixed q, the probability of $\mathsf{M}(w)_q = \mathsf{Id}$ may tend to a constant, so the average-case complexity is still $\mathcal{O}(n \log^2 n)$.
- So: take $q = q(n)$, a function of the length of w.
- Arithmetic operations in $\mathbb{Z}/q(n)\mathbb{Z}$ cost $\mathcal{O}(\log q(n)\ \log\log q(n))$.
- The function $q(n)$ must grow

– Towards an algorithm with $\mathcal{O}(n)$ average-case complexity –

- A natural idea:
  - Compute $\mathsf{M}(w)$ modulo q.
  - If $\mathsf{M}(w)_q \neq \mathsf{Id}$, then $\mathsf{M}(w) \neq \mathsf{Id}$;
    else run Algorithm DC on w.
- For a fixed q, the probability of $\mathsf{M}(w)_q = \mathsf{Id}$ may tend to a constant, so the average-case complexity is still $\mathcal{O}(n \log^2 n)$.
- So: take $q = q(n)$, a function of the length of w.
- Arithmetic operations in $\mathbb{Z}/q(n)\mathbb{Z}$ cost $\mathcal{O}(\log q(n)\ \log\log q(n))$.
- The function $q(n)$ must grow
  - sufficiently slow, so $\mathsf{M}(w)_{q(n)}$ is computed quickly, and multiplication mod $q(n)$ is fast

– Towards an algorithm with $\mathcal{O}(n)$ average-case complexity –

- A natural idea:
  - Compute $\mathsf{M}(w)$ modulo q.
  - If $\mathsf{M}(w)_q \neq \mathsf{Id}$, then $\mathsf{M}(w) \neq \mathsf{Id}$;
    else run Algorithm DC on w.
- For a fixed q, the probability of $\mathsf{M}(w)_q = \mathsf{Id}$ may tend to a constant, so the average-case complexity is still $\mathcal{O}(n \log^2 n)$.
- So: take $q = q(n)$, a function of the length of w.
- Arithmetic operations in $\mathbb{Z}/q(n)\mathbb{Z}$ cost $\mathcal{O}(\log q(n)\ \log\log q(n))$.
- The function $q(n)$ must grow
  - sufficiently slow, so $\mathsf{M}(w)_{q(n)}$ is computed quickly, and multiplication mod $q(n)$ is fast
  - and sufficiently fast, so the probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is low.

– Main result : Algorithm QuickWP –

Algorithm 3: Algorithm QuickWP

Input : a sequence w of n elements of $\tilde{\Sigma}$

Output: True if $\mathsf{M}(w) = \mathsf{Id}$, and False otherwise

```
Compute q(n) = ∏ p where p runs over prime numbers ≤ log^5 n.

if DC_{Σ,q(n)}(w) ≠ Id then
    return False
else
    if DC_Σ(w) ≠ Id then
        return False
    else
        return True
```

– Main result : Algorithm QuickWP –

Algorithm 4: Algorithm QuickWP

Input : a sequence w of n elements of $\tilde{\Sigma}$
Output: True if $\mathsf{M}(w) = \mathsf{Id}$, and False otherwise

```
Compute q(n) = ∏ p where p runs over prime numbers ≤ log^5 n.

if DC_{Σ,q(n)}(w) ≠ Id then
    return False
else
    if DC_Σ(w) ≠ Id then
        return False
    else
        return True
```

Theorem (Bassino, Nicaud and Weil 2025)

For uniform distribution over words of given length over $\tilde{\Sigma}$, QuickWP solves the word problem with **linear** bit complexity **in average**.

– Remarks on the main result –

- Algorithm QuickWP makes no assumption on the algebraic or combinatorial properties of Σ or the subgroup $H = \langle \Sigma \rangle$.

– Remarks on the main result –

- Algorithm QuickWP makes no assumption on the algebraic or combinatorial properties of Σ or the subgroup $H = \langle\Sigma\rangle$.
- The same algorithm is run, with linear average-case complexity, whether Σ consists of triangular matrices or not, and whether H is finite or infinite.

– Remarks on the main result –

- Algorithm QuickWP makes no assumption on the algebraic or combinatorial properties of Σ or the subgroup $H = \langle \Sigma \rangle$.
- The same algorithm is run, with linear average-case complexity, whether Σ consists of triangular matrices or not, and whether H is finite or infinite.
- The latter property is decidable (Jacob 1978) in polynomial time (Babai, Beals and Rockmore 1993).

– Remarks on the main result –

- Algorithm QuickWP makes no assumption on the algebraic or combinatorial properties of Σ or the subgroup $H = \langle\Sigma\rangle$.
- The same algorithm is run, with linear average-case complexity, whether Σ consists of triangular matrices or not, and whether H is finite or infinite.
- The latter property is decidable (Jacob 1978) in polynomial time (Babai, Beals and Rockmore 1993).
- The same algorithm is run, with the same average-case complexity whether H has polynomial or exponential growth, or whether it is nilpotent, polycyclic or virtually solvable.

– Remarks on the main result –

- Algorithm QuickWP makes no assumption on the algebraic or combinatorial properties of Σ or the subgroup $H = \langle\Sigma\rangle$.
- The same algorithm is run, with linear average-case complexity, whether Σ consists of triangular matrices or not, and whether H is finite or infinite.
- The latter property is decidable (Jacob 1978) in polynomial time (Babai, Beals and Rockmore 1993).
- The same algorithm is run, with the same average-case complexity whether H has polynomial or exponential growth, or whether it is nilpotent, polycyclic or virtually solvable.
- In the latter two situations, there is a linear average-case complexity for the Word Problem, using the properties of these subgroups (Olshanskii and Shpilrain 2025).

– Average-case complexity of Algorithm QuickWP –

- Since
$$q(n) = \prod_{\substack{p \;\leq\; \log^5 n \\ p \text{ prime}}} p \;\leq\; \log^{5\log^5 n} n,$$
$\ell(q(n)) = \mathsf{polylog}(n)$, $q(n)$ is computed in $\mathsf{polylog}(n)$ and the computations in $\mathbb{Z}/q(n)\mathbb{Z}$ take $\mathsf{polylog}(n)$ time.

– Average-case complexity of Algorithm QuickWP –

- Since
$$q(n) = \prod_{\substack{p \;\le\; \log^5 n \\ p \text{ prime}}} p \;\le\; \log^{5\log^5 n} n,$$
$\ell(q(n)) = \mathsf{polylog}(n)$, $q(n)$ is computed in $\mathsf{polylog}(n)$ and the computations in $\mathbb{Z}/q(n)\mathbb{Z}$ take $\mathsf{polylog}(n)$ time.
- By the Master Theorem, $\mathsf{DC}_{\Sigma,q(n)}$ runs in $\mathcal{O}(n)$ time.

– Average-case complexity of Algorithm QuickWP –

- Since
$$q(n) = \prod_{\substack{p \ \leq \ \log^5 n \\ p \text{ prime}}} p \ \leq \ \log^{5 \log^5 n} n,$$
$\ell(q(n)) = \mathsf{polylog}(n)$, $q(n)$ is computed in $\mathsf{polylog}(n)$ and the computations in $\mathbb{Z}/q(n)\mathbb{Z}$ take $\mathsf{polylog}(n)$ time.
- By the Master Theorem, $\mathsf{DC}_{\Sigma,q(n)}$ runs in $\mathcal{O}(n)$ time.
- As a result, the average-case complexity of QuickWP is
$$\mathcal{O}\left(n + \mathbb{P}_n \, n \, \log^2 n\right)$$
where $\mathbb{P}_n$ is the probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$.

– Average-case complexity of Algorithm QuickWP –

- Since

$$q(n) = \prod_{\substack{p \ \leq \ \log^5 n \\ p \text{ prime}}} p \ \leq \ \log^{5 \log^5 n} n,$$

  $\ell(q(n)) = \mathsf{polylog}(n)$, $q(n)$ is computed in $\mathsf{polylog}(n)$ and the computations in $\mathbb{Z}/q(n)\mathbb{Z}$ take $\mathsf{polylog}(n)$ time.
- By the Master Theorem, $\mathsf{DC}_{\Sigma,q(n)}$ runs in $\mathcal{O}(n)$ time.
- As a result, the average-case complexity of QuickWP is

$$\mathcal{O}\left(n + \mathbb{P}_n n \log^2 n\right)$$

  where $\mathbb{P}_n$ is the probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$.
- If $H = \langle \Sigma \rangle$ is finite, QuickWP runs in linear time.

– Average-case complexity of Algorithm QuickWP –

- Since
$$q(n) = \prod_{\substack{p \;\leq\; \log^5 n \\ p \text{ prime}}} p \;\leq\; \log^{5\log^5 n} n,$$
$\ell(q(n)) = \mathsf{polylog}(n)$, $q(n)$ is computed in $\mathsf{polylog}(n)$ and the computations in $\mathbb{Z}/q(n)\mathbb{Z}$ take $\mathsf{polylog}(n)$ time.
- By the Master Theorem, $\mathsf{DC}_{\Sigma,q(n)}$ runs in $\mathcal{O}(n)$ time.
- As a result, the average-case complexity of QuickWP is
$$\mathcal{O}\left(n + \mathbb{P}_n n \log^2 n\right)$$
where $\mathbb{P}_n$ is the probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$.
- If $H = \langle \Sigma \rangle$ is finite, QuickWP runs in linear time.
- We need to show that, if H is infinite, then $\mathbb{P}_n = \mathcal{O}(\log^{-2} n)$.

– Random words and Markov chain –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$.

– Random words and Markov chain –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$.
- $|\Sigma| = k$, $H = \langle \Sigma \rangle$, $m \geq 2$, $H_m =$ projection mod m of H.

– Random words and Markov chain –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$.
- $|\Sigma| = k$, $H = \langle\Sigma\rangle$, $m \geq 2$, $H_m =$ projection mod m of H.
- Assumptions: $\Sigma \cap \Sigma^{-1} = \emptyset$, and m sufficiently large: distinct elements of $\tilde{\Sigma}$ are distinct mod m.

– Random words and Markov chain –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$.
- $|\Sigma| = k$, $H = \langle \Sigma \rangle$, $m \geq 2$, $H_m =$ projection mod m of H.
- Assumptions: $\Sigma \cap \Sigma^{-1} = \emptyset$, and m sufficiently large: distinct elements of $\tilde{\Sigma}$ are distinct mod m.
- The matrices $\mathsf{M}(w)_m$ are produced by the length n trajectories in the Markov chain $\mathfrak{U}_m$ such that:

– Random words and Markov chain –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$.
- $|\Sigma| = k$, $H = \langle \Sigma \rangle$, $m \geq 2$, $H_m =$ projection mod m of H.
- Assumptions: $\Sigma \cap \Sigma^{-1} = \emptyset$, and m sufficiently large: distinct elements of $\tilde{\Sigma}$ are distinct mod m.
- The matrices $\mathsf{M}(w)_m$ are produced by the length n trajectories in the Markov chain $\mathfrak{U}_m$ such that:
  - The state set of $\mathfrak{U}_m$ is the subgroup H_m.

– Random words and Markov chain –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$.
- $|\Sigma| = k$, $H = \langle \Sigma \rangle$, $m \geq 2$, $H_m =$ projection mod m of H.
- Assumptions: $\Sigma \cap \Sigma^{-1} = \emptyset$, and m sufficiently large: distinct elements of $\tilde{\Sigma}$ are distinct mod m.
- The matrices $\mathsf{M}(w)_m$ are produced by the length n trajectories in the Markov chain $\mathfrak{U}_m$ such that:
  - The state set of $\mathfrak{U}_m$ is the subgroup H_m.
  - There is an edge $M \xrightarrow{\frac{1}{2k}} M'$ if and only if $\exists A \in \tilde{\Sigma}$ such that $M\,A = M'$.

– Random words and Markov chain –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$.
- $|\Sigma| = k$, $H = \langle \Sigma \rangle$, $m \geq 2$, $H_m =$ projection mod m of H.
- Assumptions: $\Sigma \cap \Sigma^{-1} = \emptyset$, and m sufficiently large: distinct elements of $\tilde{\Sigma}$ are distinct mod m.
- The matrices $\mathsf{M}(w)_m$ are produced by the length n trajectories in the Markov chain $\mathfrak{U}_m$ such that:
  - The state set of $\mathfrak{U}_m$ is the subgroup H_m.
  - There is an edge $M \xrightarrow{\frac{1}{2k}} M'$ if and only if $\exists A \in \tilde{\Sigma}$ such that $M\,A = M'$.
  - The initial vector assigns 1 to Id and 0 to the other states.

– Properties of $\mathfrak{U}_m$ –

Let P_m be the matrix of transition of $\mathfrak{U}_m$.

- Since P_m is symmetric, the uniform distribution is a stationnary distribution of $\mathfrak{U}_m$.

– Properties of $\mathfrak{U}_m$ –

Let P_m be the matrix of transition of $\mathfrak{U}_m$.

- Since P_m is symmetric, the uniform distribution is a stationnary distribution of $\mathfrak{U}_m$.
- Since P_m is irreducible, it is the only stationnary distribution.

– Properties of $\mathfrak{U}_m$ –

Let P_m be the matrix of transition of $\mathfrak{U}_m$.

- Since P_m is symmetric, the uniform distribution is a stationnary distribution of $\mathfrak{U}_m$.
- Since P_m is irreducible, it is the only stationnary distribution.
- But P_m maybe not aperiodic : as there are length 2 circuits in $\mathfrak{U}_m$, the period is 1 or 2.

– A symmetric primitive Markov chain –

- $\mathfrak{U}_m^2$ is symmetric and aperiodic, but maybe not irreducible: let $\tilde{H}_m$ be the set of states accessible from Id in $\mathfrak{U}_m^2$.

– A symmetric primitive Markov chain –

- $\mathfrak{U}_m^2$ is symmetric and aperiodic, but maybe not irreducible: let $\tilde{H}_m$ be the set of states accessible from Id in $\mathfrak{U}_m^2$.
- if $\mathfrak{U}_m$ has period 2, then $\mathfrak{U}_m^2$ splits its state set H_m into two disjoint Markov chains — one on $\tilde{H}_m$ and one on the states at odd distance from Id.

– A symmetric primitive Markov chain –

- $\mathfrak{U}_m^2$ is symmetric and aperiodic, but maybe not irreducible: let $\tilde{H}_m$ be the set of states accessible from Id in $\mathfrak{U}_m^2$.
- if $\mathfrak{U}_m$ has period 2, then $\mathfrak{U}_m^2$ splits its state set H_m into two disjoint Markov chains — one on $\tilde{H}_m$ and one on the states at odd distance from Id.
- $\tilde{H}_m = \langle \tilde{\Sigma}^2 \rangle$ is equal to H_m or to an index 2 subgroup of H_m: $|\tilde{H}_m| \geq \frac{1}{2}|H_m|$.

– A symmetric primitive Markov chain –

- $\mathfrak{U}_m^2$ is symmetric and aperiodic, but maybe not irreducible: let $\tilde{H}_m$ be the set of states accessible from Id in $\mathfrak{U}_m^2$.
- if $\mathfrak{U}_m$ has period 2, then $\mathfrak{U}_m^2$ splits its state set H_m into two disjoint Markov chains — one on $\tilde{H}_m$ and one on the states at odd distance from Id.
- $\tilde{H}_m = \langle \tilde{\Sigma}^2 \rangle$ is equal to H_m or to an index 2 subgroup of H_m: $|\tilde{H}_m| \geq \frac{1}{2}|H_m|$.
- $\tilde{\mathfrak{U}}_m =$ the restriction of $\mathfrak{U}_m$ to $\tilde{H}_m$, with transition matrix $\tilde{P}_m$.

– A symmetric primitive Markov chain –

- $\mathfrak{U}_m^2$ is symmetric and aperiodic, but maybe not irreducible: let $\tilde{H}_m$ be the set of states accessible from Id in $\mathfrak{U}_m^2$.
- if $\mathfrak{U}_m$ has period 2, then $\mathfrak{U}_m^2$ splits its state set H_m into two disjoint Markov chains — one on $\tilde{H}_m$ and one on the states at odd distance from Id.
- $\tilde{H}_m = \langle \tilde{\Sigma}^2 \rangle$ is equal to H_m or to an index 2 subgroup of H_m: $|\tilde{H}_m| \geq \frac{1}{2}|H_m|$.
- $\tilde{\mathfrak{U}}_m =$ the restriction of $\mathfrak{U}_m$ to $\tilde{H}_m$, with transition matrix $\tilde{P}_m$.
- Then $\tilde{\mathfrak{U}}_m$ is primitive and symmetric, and for any distribution μ, $\mu\tilde{P}_m^n$ converges to the uniform distribution $\left(\frac{\mathbf{1}}{|\tilde{H}_m|}\right)$.

– The rate of convergence –

Rate of convergence

The distribution $\tilde{P}_m^n(\mathsf{Id}, \cdot)$, reached after n random steps starting at Id satisfies

$$\left\| \tilde{P}_m^n(\mathsf{Id}, \cdot) - \frac{\mathbf{1}}{|\tilde{H}_m|} \right\|_{\mathrm{Var}} \leq \frac{1}{2} \sqrt{|\tilde{H}_m|} \left(1 - \frac{1}{4k^2 |\tilde{H}_m|^2} \right)^n .$$

– The rate of convergence –

Rate of convergence

The distribution $\tilde{P}_m^n(\mathsf{Id}, \cdot)$, reached after n random steps starting at Id satisfies

$$\left\| \tilde{P}_m^n(\mathsf{Id}, \cdot) - \frac{\mathbf{1}}{|\tilde{H}_m|} \right\|_{\mathrm{Var}} \leq \frac{1}{2} \sqrt{|\tilde{H}_m|} \left(1 - \frac{1}{4k^2 |\tilde{H}_m|^2}\right)^n .$$

- The proof uses results on the second largest and on the least eigenvalues of $\tilde{P}_m$ (Diaconis and Stroock 1991).

Rate of convergence

The distribution $\tilde{P}_m^n(\mathsf{Id}, \cdot)$, reached after n random steps starting at Id satisfies

$$\left\| \tilde{P}_m^n(\mathsf{Id}, \cdot) - \frac{\mathbf{1}}{|\tilde{H}_m|} \right\|_{\mathrm{Var}} \leq \frac{1}{2} \sqrt{|\tilde{H}_m|} \left(1 - \frac{1}{4k^2 |\tilde{H}_m|^2} \right)^n .$$

- The proof uses results on the second largest and on the least eigenvalues of $\tilde{P}_m$ (Diaconis and Stroock 1991).
- Now let's go back to computations mod $m = q(n)$ and evaluate $|\tilde{H}_{q(n)}|$.

Order of an element

The order of an element A of a group is $|\langle A\rangle|$. It is the smallest positive integer ℓ such that $A^\ell = \mathsf{Id}$.

Order of an element

The order of an element A of a group is $|\langle A \rangle|$. It is the smallest positive integer ℓ such that $A^\ell = \mathsf{Id}$.

- An infinite subgroup of $\mathsf{GL}_d(\mathbb{Z})$ always contains a matrix of infinite order (Schur 1911). So $H = \langle \Sigma \rangle$ contains a matrix A with infinite order.

– An element of H of large order –

Order of an element

The order of an element A of a group is $|\langle A\rangle|$. It is the smallest positive integer ℓ such that $A^\ell = \mathsf{Id}$.

- An infinite subgroup of $\mathsf{GL}_d(\mathbb{Z})$ always contains a matrix of infinite order (Schur 1911). So $H = \langle\Sigma\rangle$ contains a matrix A with infinite order.

An element of at least logarithmic order

Let $A \in \mathsf{GL}_d(\mathbb{Z})$ with infinite order. If n is large enough, $q(n)$ has a prime factor p such that A_p has order $> 2\log^2 n$.

– An element of H of large order –

Order of an element

The order of an element A of a group is $|\langle A\rangle|$. It is the smallest positive integer ℓ such that $A^\ell = \mathsf{Id}$.

- An infinite subgroup of $\mathsf{GL}_d(\mathbb{Z})$ always contains a matrix of infinite order (Schur 1911). So $H = \langle\Sigma\rangle$ contains a matrix A with infinite order.

An element of at least logarithmic order

Let $A \in \mathsf{GL}_d(\mathbb{Z})$ with infinite order. If n is large enough, $q(n)$ has a prime factor p such that A_p has order $> 2\log^2 n$.

- Let $A \in \mathsf{GL}_d(\mathbb{Z})$ of infinite order. The number of primes p such that A_p has order $\leq L$ is $\mathcal{O}(L^2)$ (Kurberg, 2003).

– An element of H of large order –

Order of an element

The order of an element A of a group is $|\langle A\rangle|$. It is the smallest positive integer ℓ such that $A^\ell = \mathsf{Id}$.

- An infinite subgroup of $\mathsf{GL}_d(\mathbb{Z})$ always contains a matrix of infinite order (Schur 1911). So $H = \langle \Sigma \rangle$ contains a matrix A with infinite order.

An element of at least logarithmic order

Let $A \in \mathsf{GL}_d(\mathbb{Z})$ with infinite order. If n is large enough, $q(n)$ has a prime factor p such that A_p has order $> 2\log^2 n$.

- Let $A \in \mathsf{GL}_d(\mathbb{Z})$ of infinite order. The number of primes p such that A_p has order $\leq L$ is $\mathcal{O}(L^2)$ (Kurberg, 2003).
- There are $\mathcal{O}(\log^4 n)$ primes p such that A_p has order $\leq 2\log^2 n$, and $q(n)$ is the product of the primes $\leq \log^5 n$ — of which there are, asymptotically, $\sim \frac{\log^5 n}{5\log\log n}$.

– An element of H of large order –

An element of at least logarithmic order

Let $A \in \mathsf{GL}_d(\mathbb{Z})$ with infinite order. If n is large enough, $q(n)$ has a prime factor p such that A_p has order $> 2\log^2 n$.

– An element of H of large order –

An element of at least logarithmic order

Let $A \in \mathsf{GL}_d(\mathbb{Z})$ with infinite order. If n is large enough, $q(n)$ has a prime factor p such that A_p has order $> 2\log^2 n$.

- Let $p_n \leq \log^5 n$ be a prime such that A_{p_n} has order $> 2\log^2 n$.

– An element of H of large order –

An element of at least logarithmic order

Let $A \in \mathsf{GL}_d(\mathbb{Z})$ with infinite order. If n is large enough, $q(n)$ has a prime factor p such that A_p has order $> 2\log^2 n$.

- Let $p_n \leq \log^5 n$ be a prime such that A_{p_n} has order $> 2\log^2 n$.
- **NB:** We are not concerned with the value of the matrix A or the prime p_n, nor with how hard it would be to compute them.

– An element of H of large order –

An element of at least logarithmic order

Let $A \in \mathsf{GL}_d(\mathbb{Z})$ with infinite order. If n is large enough, $q(n)$ has a prime factor p such that A_p has order $> 2\log^2 n$.

- Let $p_n \leq \log^5 n$ be a prime such that A_{p_n} has order $> 2\log^2 n$.
- **NB:** We are not concerned with the value of the matrix A or the prime p_n, nor with how hard it would be to compute them.
- Then $|\tilde{H}_{p_n}| \geq \frac{1}{2}|H_{p_n}| \geq \frac{1}{2}|\langle A_{p_n}\rangle| > \log^2 n$.

– An element of H of large order –

An element of at least logarithmic order

Let $A \in \mathsf{GL}_d(\mathbb{Z})$ with infinite order. If n is large enough, $q(n)$ has a prime factor p such that A_p has order $> 2\log^2 n$.

- Let $p_n \leq \log^5 n$ be a prime such that A_{p_n} has order $> 2\log^2 n$.
- **NB:** We are not concerned with the value of the matrix A or the prime p_n, nor with how hard it would be to compute them.
- Then $|\tilde{H}_{p_n}| \geq \frac{1}{2}|H_{p_n}| \geq \frac{1}{2}|\langle A_{p_n}\rangle| > \log^2 n$.
- Also: $|\tilde{H}_{p_n}| \leq p_n^{d^2} \leq \log^{5d^2} n$.

– The probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is $\mathcal{O}(\log^{-2} n)$ –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$

– The probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is $\mathcal{O}(\log^{-2} n)$ –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$
- $P^n_{q(n)}(\mathsf{Id}, \mathsf{Id}) \leq P^n_{p_n}(\mathsf{Id}, \mathsf{Id})$.

– The probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is $\mathcal{O}(\log^{-2} n)$ –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$
- $P^n_{q(n)}(\mathsf{Id}, \mathsf{Id}) \leq P^n_{p_n}(\mathsf{Id}, \mathsf{Id})$.
- if $n = 2\nu$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \tilde{P}^{\nu}_{p_n}(\mathsf{Id}, \mathsf{Id})$

– The probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is $\mathcal{O}(\log^{-2} n)$ –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$
- $P^n_{q(n)}(\mathsf{Id}, \mathsf{Id}) \leq P^n_{p_n}(\mathsf{Id}, \mathsf{Id})$.
- if $n = 2\nu$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \tilde{P}^{\nu}_{p_n}(\mathsf{Id}, \mathsf{Id})$
  - $\tilde{P}^{\nu}_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{|\tilde{H}_{p_n}|} + \frac{1}{2}\sqrt{|\tilde{H}_{p_n}|}\left(1 - \frac{1}{4k^2|\tilde{H}_{p_n}|^2}\right)^{\nu}$

– The probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is $\mathcal{O}(\log^{-2} n)$ –

- *We want to show that* $\mathbb{P}_n(M(w)_{q(n)} = Id)$ *is* $\mathcal{O}(\log^{-2} n)$
- $P^n_{q(n)}(\mathsf{Id}, \mathsf{Id}) \leq P^n_{p_n}(\mathsf{Id}, \mathsf{Id})$.
- if $n = 2\nu$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id})$
    - $\tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{|\tilde{H}_{p_n}|} + \frac{1}{2}\sqrt{|\tilde{H}_{p_n}|}\left(1 - \frac{1}{4k^2|\tilde{H}_{p_n}|^2}\right)^\nu$
    - $\tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{\log^2 n} + \frac{1}{2}\sqrt{(\log n)^{5d^2}}\,\exp\left(\frac{n}{8k^2(\log n)^{10d^2}}\right)$

– The probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is $\mathcal{O}(\log^{-2} n)$ –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$
- $P^n_{q(n)}(\mathsf{Id}, \mathsf{Id}) \leq P^n_{p_n}(\mathsf{Id}, \mathsf{Id})$.
- if $n = 2\nu$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \tilde{P}^{\nu}_{p_n}(\mathsf{Id}, \mathsf{Id})$
  - $\tilde{P}^{\nu}_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{|\tilde{H}_{p_n}|} + \frac{1}{2}\sqrt{|\tilde{H}_{p_n}|}\left(1 - \frac{1}{4k^2|\tilde{H}_{p_n}|^2}\right)^{\nu}$
  - $\tilde{P}^{\nu}_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{\log^2 n} + \frac{1}{2}\sqrt{(\log n)^{5d^2}}\,\exp\left(\frac{n}{8k^2(\log n)^{10d^2}}\right)$
  - that is $\mathcal{O}(\log^{-2} n)$

– The probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is $\mathcal{O}(\log^{-2} n)$ –

- *We want to show that* $\mathbb{P}_n(M(w)_{q(n)} = Id)$ *is* $\mathcal{O}(\log^{-2} n)$
- $P^n_{q(n)}(\mathsf{Id}, \mathsf{Id}) \leq P^n_{p_n}(\mathsf{Id}, \mathsf{Id})$.
- if $n = 2\nu$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id})$
  - $\tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{|\tilde{H}_{p_n}|} + \frac{1}{2}\sqrt{|\tilde{H}_{p_n}|}\left(1 - \frac{1}{4k^2|\tilde{H}_{p_n}|^2}\right)^\nu$
  - $\tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{\log^2 n} + \frac{1}{2}\sqrt{(\log n)^{5d^2}}\,\exp\left(\frac{n}{8k^2(\log n)^{10d^2}}\right)$
  - that is $\mathcal{O}(\log^{-2} n)$
- If $n = 2\nu + 1$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \sum_{h \in H_{p_n}} \tilde{P}^\nu_{p_n}(\mathsf{Id}, h) P_{p_n}(h, \mathsf{Id})$

– The probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is $\mathcal{O}(\log^{-2} n)$ –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$
- $P^n_{q(n)}(\mathsf{Id}, \mathsf{Id}) \leq P^n_{p_n}(\mathsf{Id}, \mathsf{Id})$.
- if $n = 2\nu$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id})$
  - $\tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{|\tilde{H}_{p_n}|} + \frac{1}{2}\sqrt{|\tilde{H}_{p_n}|}\left(1 - \frac{1}{4k^2|\tilde{H}_{p_n}|^2}\right)^\nu$
  - $\tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{\log^2 n} + \frac{1}{2}\sqrt{(\log n)^{5d^2}}\exp\left(\frac{n}{8k^2(\log n)^{10d^2}}\right)$
  - that is $\mathcal{O}(\log^{-2} n)$
- If $n = 2\nu + 1$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \sum_{h \in H_{p_n}} \tilde{P}^\nu_{p_n}(\mathsf{Id}, h) P_{p_n}(h, \mathsf{Id})$
  - The second factor is non-zero for $h = B_{p_n}$ where $B \in \tilde{\Sigma}$: $2k$ values, all equal to $\frac{1}{2k}$

– The probability that $\mathsf{M}(w)_{q(n)} = \mathsf{Id}$ is $\mathcal{O}(\log^{-2} n)$ –

- *We want to show that* $\mathbb{P}_n(\mathsf{M}(w)_{q(n)} = \mathsf{Id})$ *is* $\mathcal{O}(\log^{-2} n)$
- $P^n_{q(n)}(\mathsf{Id}, \mathsf{Id}) \leq P^n_{p_n}(\mathsf{Id}, \mathsf{Id})$.
- if $n = 2\nu$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id})$
  - $\tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{|\tilde{H}_{p_n}|} + \frac{1}{2}\sqrt{|\tilde{H}_{p_n}|}\left(1 - \frac{1}{4k^2|\tilde{H}_{p_n}|^2}\right)^\nu$
  - $\tilde{P}^\nu_{p_n}(\mathsf{Id}, \mathsf{Id}) \leq \frac{1}{\log^2 n} + \frac{1}{2}\sqrt{(\log n)^{5d^2}}\exp\left(\frac{n}{8k^2(\log n)^{10d^2}}\right)$
  - that is $\mathcal{O}(\log^{-2} n)$
- If $n = 2\nu + 1$, $P^n_{p_n}(\mathsf{Id}, \mathsf{Id}) = \sum_{h \in H_{p_n}} \tilde{P}^\nu_{p_n}(\mathsf{Id}, h) P_{p_n}(h, \mathsf{Id})$
  - The second factor is non-zero for $h = B_{p_n}$ where $B \in \tilde{\Sigma}$: $2k$ values, all equal to $\frac{1}{2k}$
  - so again $\mathcal{O}(\log^{-2} n)$.

– Open problems –

- The algorithm QuickWP solve the Word Problem in the subgroups of $\mathsf{GL}(\mathbb{Z})$ with a linear bit complexity in average for the uniform distribution on words of a given length.

– Open problems –

- The algorithm QuickWP solve the Word Problem in the subgroups of $\mathsf{GL}(\mathbb{Z})$ with a linear bit complexity in average for the uniform distribution on words of a given length.
- A reduced word is a word that contains neither AA^{-1} nor $A^{-1}A$ as factor.

– Open problems –

- The algorithm QuickWP solve the Word Problem in the subgroups of $\mathsf{GL}(\mathbb{Z})$ with a linear bit complexity in average for the uniform distribution on words of a given length.
- A reduced word is a word that contains neither AA^{-1} nor $A^{-1}A$ as factor.
- What is the average-case complexity of the Word Problem for the uniform distribution on reduced words of a given length?

– Open problems –

- The algorithm QuickWP solve the Word Problem in the subgroups of $\mathsf{GL}(\mathbb{Z})$ with a linear bit complexity in average for the uniform distribution on words of a given length.
- A reduced word is a word that contains neither AA^{-1} nor $A^{-1}A$ as factor.
- What is the average-case complexity of the Word Problem for the uniform distribution on reduced words of a given length?
- What is the average-case complexity of the Word Problem in the subgroups of $\mathsf{GL}(\mathbb{Q})$?

Thank you for your attention!