Asymptotic algebra and modern cryptanalysis

Alexei Myasnikov

McGill University (Montreal)

Algebraic Cryptography Center (Stevens Inst.)

2

Search for generically hard problems

Plan:

- Generic complexity of NP-complete problems
- Generic complexity of NP-complete on average problems.
- Generic complexity of the Halting Problem
- Black Holes

Generic complexity of NP-complete problems

From R.Gilman's talk:

3-SAT, Hamiltonian cycle problem, subset sum problem, knapsack problem, ..., are generically linear time.

Question: are all NP-complete problems generically polynomial?

Generic complexity of NP-complete on average problems

Theorem The following NP-complete on average problems are generically polynomial:

1) [Hamkins-Myasnikov] Halting Problem for Turing machines;

2) [Myasnikov-Ushakov] The Word Problem for finitely presented groups.

Average case NP-completeness of these problems is due to Gurevich and Wang.

Question: what is generic complexity of other NP-complete on average problems?

Generic complexity of undecidable problems

The Halting Problem: Decide whether or not a Turing machine M halts if started on an empty tape.

Turing: the Halting Problem for Turing machines is undecidable.

Generic complexity of the Halting problem

M = the set of all Turing machines (programs).

Stratification:

M_n = the set of all programs with n states.

$$M = \cup_n M_n$$

Asymptotic density ρ on M with respect to this stratification.

Theorem [Hamkins, Myasnikov (2004)]

There is a set $P \subseteq M$ of Turing machine programs such that:

1) P is generic (with respect to ρ).

2) P is linear time decidable.

3) The halting problem in P is linear time decidable.

Here we use the standard model of a Turing machine with one-way infinite tape (there is the left-most cell).

Idea of the proof

A Turing machine M has a finite program directing the operation of a head reading and writing 0s and 1s while moving left or right on a one-way infinite tape.

M has n states $Q = \{q_1, \ldots, q_n\}$ and the halting state *halt*.

M is given by a *program*, which is a function

$$p : Q \times \{0, 1\} \to (Q \cup \{halt\}) \times \{0, 1\} \times \{L, R\}.$$

The computation of a program proceeds by iteratively performing the instructions of such transition rules, halting when the *halt* state is reached. If the machine attempts to move left from the left-most cell, then the head falls off the tape and all computation ceases.

Idea of the proof

Let P be the set of programs that on input 0 either halt before repeating a state or fall off the tape before repeating a state.

Then:

2) P is linear time decidable, since we need only run a program p for at most n steps, where n is the number of states in p, to determine whether or not it is in P.

3) the halting problem is linear time decidable for programs p in P, since again we need only simulate p for n steps to know whether it halted or fell off.

What remains is to prove that P is generic.

Key points

- for a fixed k the set B_k of programs not repeating states within the first k steps is generic

- for the first k steps of computation, the programs in B_k behave like a simple random walk on the tape (with uniform probability of going left or right)

- Polya's classical result on random walks on $\mathbb{Z}$ states that with probability one the random walk reaches any given fixed position of the tape.

  Hence the machine falls off the tape.

More on generic properties of the Halting Problem

Question. Does the conclusion of the Theorem hold for Turing machine models with two-way infinite tapes?

A. Rybalov will talk today about:

the size of the generic sets P, where the Halting Problem is decidable (an invariant result!);

generically undecidable sets.

Black Holes

Let S be a set with a measure μ and P an algorithmic problem with inputs in S.

$\mathcal{A}$ - generically "fast" partial algorithm solving P.

P is easy on the halting set $S_{\mathcal{A}}$ of $\mathcal{A}$, so the hard instances of P are in the complement $S - S_{\mathcal{A}}$.

The complement $S - S_{\mathcal{A}} = BH_{\mathcal{A}}$ is called the **Black Hole** of $\mathcal{A}$.

There are natural algorithms with non r.e. black holes - almost any general algorithm in combinatorial group theory.

Black Holes of algorithmic problems

The Black Hole $BH_{\mathcal{A}}$ of the algorithm $\mathcal{A}$ is a **"grey area"** for P.

To make it smaller take another algorithm $\mathcal{A}_1$ and run both in parallel. The grey area becomes

$$BH_{\mathcal{A}} \cap BH_{\mathcal{A}_1}$$

In general, if $\mathcal{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ is a list of "currently known" partial algorithms for P then

$$BH_{\mathcal{A}}(P) = \bigcap BH_{\mathcal{A}_i}$$

is the **Black Hole** for P relative to $\mathcal{A}$.

Black Holes and cryptosystems

Think of P as the underlying algorithmic problem for a cryptosystem $\mathcal{C}$ and of $\mathcal{A}$ as a list of known attacks.

Then the black hole $BH_{\mathcal{A}}(P)$ is the space of "good keys".

Question: can one always find another algorithm $\mathcal{A}'$ that solves the problem P on a subset S' from the Black Hole $BH_{\mathcal{A}}(P)$?

Of course, yes, if S' is finite. But is it always possible for an infinite S'?

Immune Black Holes

Recall from classics:

a subset of $\mathbb{N}$ is **immune** if it is infinite and contains no infinite r.e. subsets.

Theorem [Post] There exists a r.e. set $T \subset \mathbb{N}$ (called **simple**) such that its complement $\bar{T} = \mathbb{N} - T$ is immune.

Corollary: There is an algorithm $\mathcal{A}$ (whose halting set is simple) such that no algorithm can do better then $\mathcal{A}$ on an infinite r.e. subset of $BH_{\mathcal{A}}$.

Making Black Holes really hard

Given P and $\mathcal{A}$ as above.

Question: can one always find another algorithm $\mathcal{A}'$ such that running both $\mathcal{A}$ and $\mathcal{A}'$ in parallel gives a solution to P on an infinite (non r.e.) subset of the black hole $BH_{\mathcal{A}}$?

It seems, there are black holes that resist any essential attacks (private communication).

Algorithmic portraits

It seems, the ideal description of algorithmic complexity of a problem should include:

- description of the worst-case complexity;

- a good generic algorithm $\mathcal{A}$;

- description of the Black Hole $BH_{\mathcal{A}}$ (may not be easy - see research on 3-SAT problem);

- a procedure to generate random elements from $BH_{\mathcal{A}}$.

Note: the emphasis is on the Black Hole and on the generating procedure.

Generic complexity of the Word Problem in finitely presented groups

Main Problem:

What is generic complexity of the Word and Conjugacy problems in groups?

Generic complexity in Groups:

Recent general results:

Kapovich, Myasnikov, Schupp, Shpilrain: generic and average complexity of WP and CP via "big" quotients - "NO" part;

Borovik, Myasnikov, Remeslennikov: generic complexity of WP and CP in free products with amalgamation and HNN extensions;

Myasnikov, Ushakov: generic complexity of WP and CP in the "Yes" part.

Outcome: for many groups the Word and the Conjugacy problems are generically fast.

"No" part

Quotient test: let $\phi : G \to H$ be a homomorphism and $g \in G$ then

$$g^{\phi} \neq 1 \Longrightarrow g \neq 1$$

Application: Choose a homomorphism $\phi : G \to H$ (if it exists) such that

1) WP is "fast" in H;

2) the kernel of ϕ is small.

Then the answer to the question "Is $g = 1$ in G?" is almost always "No" and it is fast.

Theorem [KMSS]. Let G be a finitely generated group. Suppose G has a finite index subgroup that possesses an infinite quotient group H for which the word problem is solvable in the complexity class $\mathcal{C}$. Then the word problem for G has generic-case complexity in the class $\mathcal{C}$. Moreover, if the group H is non-amenable, then the generic-case complexity of the word problem for G is strongly in $\mathcal{C}$.

Applications to cryptography:

NONE:

Recall, in crypto schemes we know in advance that $u^x = v$ has a solution in G!

So we are interested only in the **"Yes"** part of the conjugacy problem.

The main question remains:

what is the generic complexity of the "Yes" parts of the classical decision problems in groups?

$$WP_{Yes} = \{w \mid w = 1 \ in \ G\}$$

$$CP_{Yes} = \{(u, v) \mid u^x = v \ for \ some \ x \in G\}$$

"Search" algorithmic problems

"Yes" part $\Longrightarrow$ Search problems:

Search Word Problem in $G = \langle X \mid R \rangle$:

given $w \in WP_{Yes} \Longrightarrow$ write as $w = \prod r_i^{s_i}$ $(r_i \in R^{\pm 1}, s_i \in F(X))$

Search Conjugacy Problem in $G = \langle X \mid R \rangle$:

given $(u, v) \in CP_{Yes} \Longrightarrow$ find a conjugator y: $u^y = v$.

Complexity: much harder.

"Yes" part in groups

Big problems here:

- what are random *trivial* or *conjugated* elements in G?

- how to define *size* of elements in the "Yes" part? (definitely not the length of a word)

- what is the *measure* on the "Yes" part?

Theorem [MU]. In any finitely presented group the *Search Word and Conjugacy* problem are generically polynomial.

But the generic set mentioned above is **not** necessary strongly generic.

Tools used: random van Kampen diagrams. "Generic hyperbolicity" of diagrams.

Van Kampen diagrams

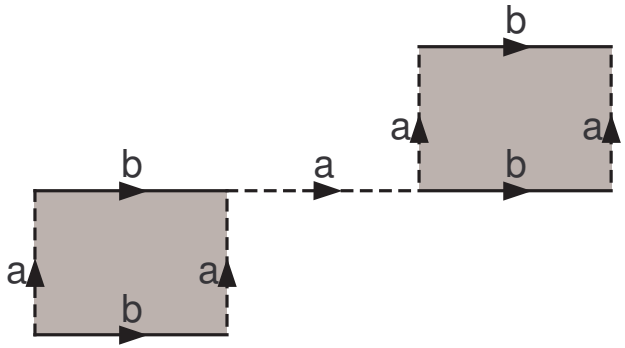
Main idea: *Represent trivial words by Van Kampen diagrams*

Lemma (van Kampen). Let $G = \langle X; R \rangle$ and $w \in (X^{\pm 1})^*$. Then $w =_G 1$ if and only if there exists a diagram D over $\langle X; R \rangle$ with perimeter w.

Now the question is: what are random van Kampen diagrams and what is a measure (asymptotic density) on them?

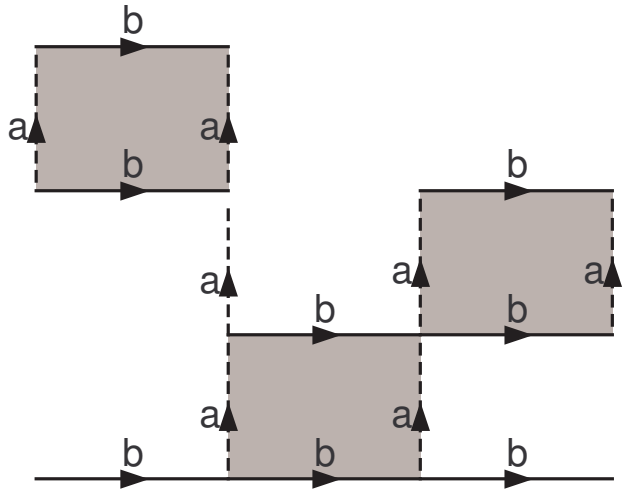
Geometric size of van Kampen diagrams.

Geometric size of a diagram D is the total number of cells and free edges (i.e. edges not on boundaries of cells) in D.



$\chi(D_1) = 3$

in $A_2 = \langle a,b \,;\, a\, b\, a^{-1}\, b^{-1} \rangle$



$\chi(D_2) = 6$

in $A_2 = \langle a,b \,;\, a\, b\, a^{-1}\, b^{-1} \rangle$

Partition of diagrams.

The set Ω of all diagrams over $\langle X; R\rangle$ can be partitioned as

$$\Omega = \cup_{i=0}^{\infty} \Omega_i,$$

where Ω_n is the set of all diagrams of complexity n.

We use the asymptotic density on Ω (relative to the partition above) to measure sets of diagrams.

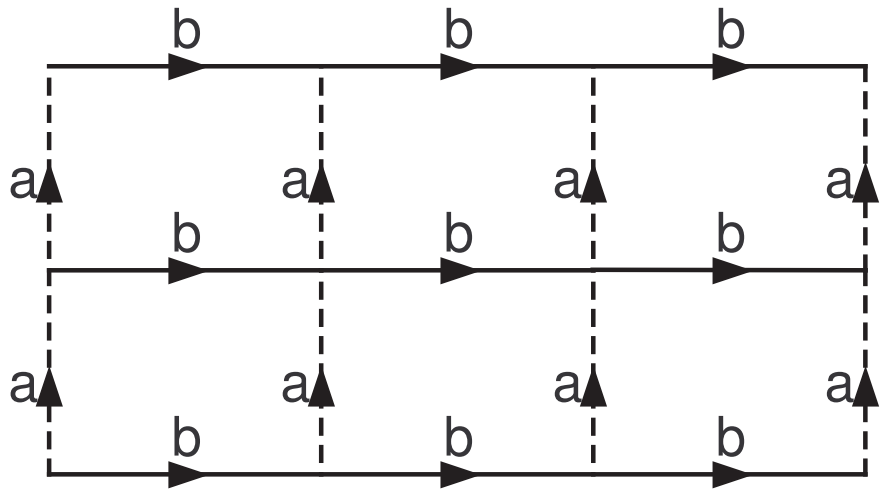
Algorithmic complexity of van Kampen diagrams.

Let D be a diagram over $\langle X; R\rangle$.

The *depth* of D is the longest chain-distance from the outer face to inner faces.

Depth of a word $w =_G 1$ is the minimal depth of van Kampen diagrams representing w.

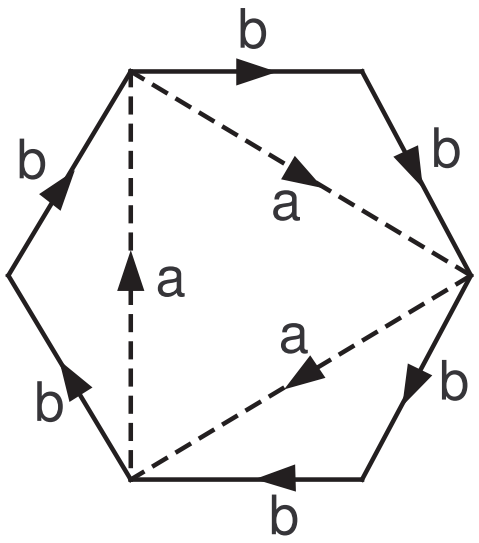
Depth of w is the "algorithmic" complexity of w.



$a^2\, b^3\, a^{-2}\, b^{-3} = 1$

in $A_2 = < a,b \,;\, a\, b\, a^{-1}\, b^{-1} >$

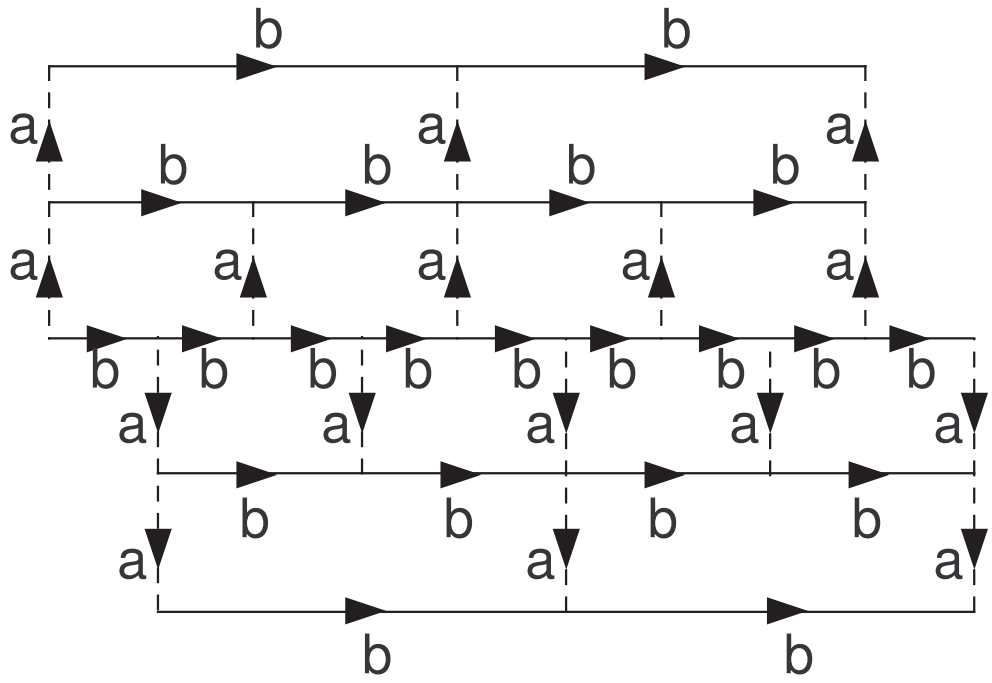
$\delta(D) = 1$



$b^6 = 1$

in $G = < a,b ; a^3 , a\, b^{-2} >$

$\delta(D) = 1$

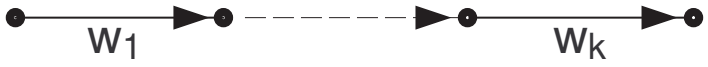


$b^{-1}\, a^2\, b^2\, a^{-2}\, b\, a^2\, b^{-2}\, a^{-2} = 1$

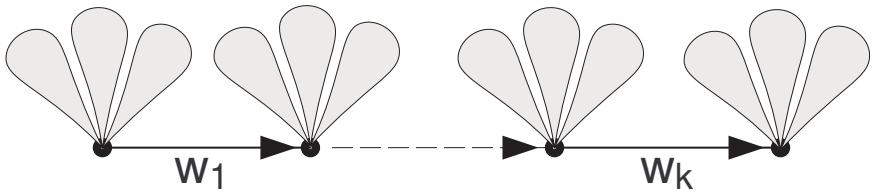
in $BS(1,2) = \langle a,b \,;\, a\, b\, a^{-1}\, b^{-2} \rangle$

$\delta(D) = 2$

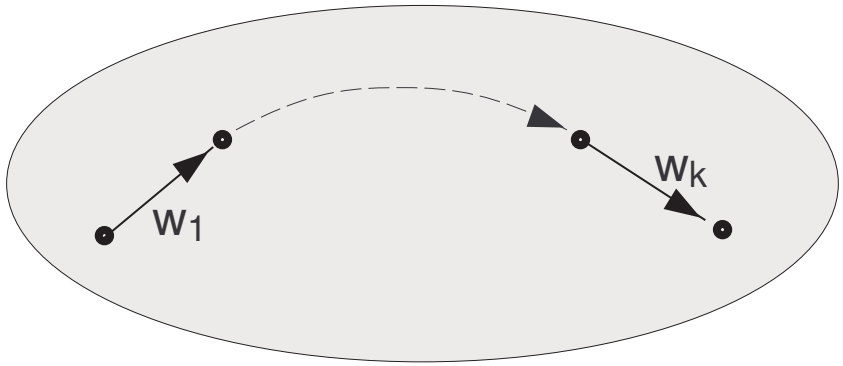
Algorithm $\mathcal{A}_W$ for WP.



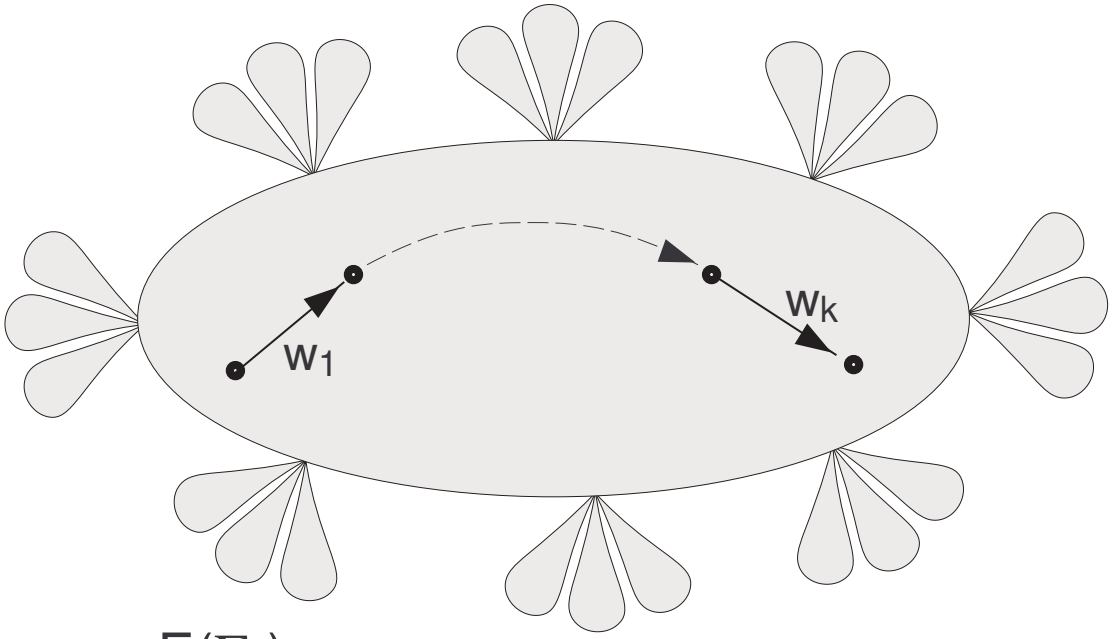
Γ_0



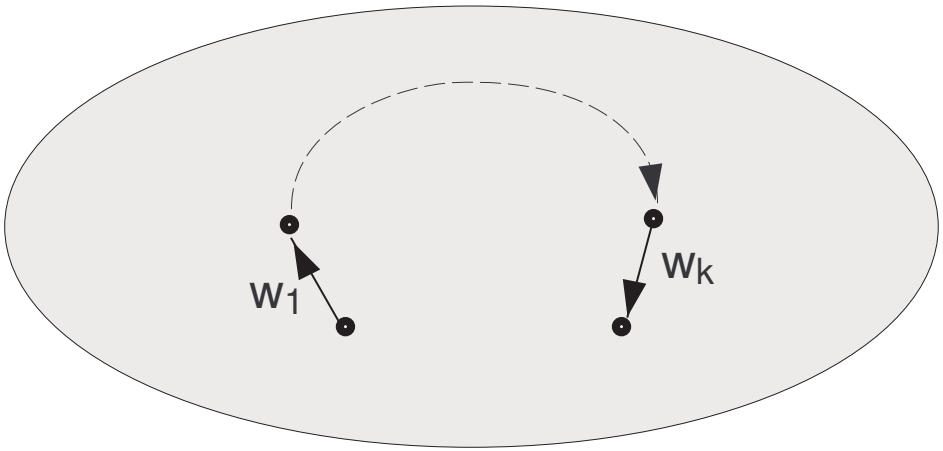
$E(\Gamma_0)$



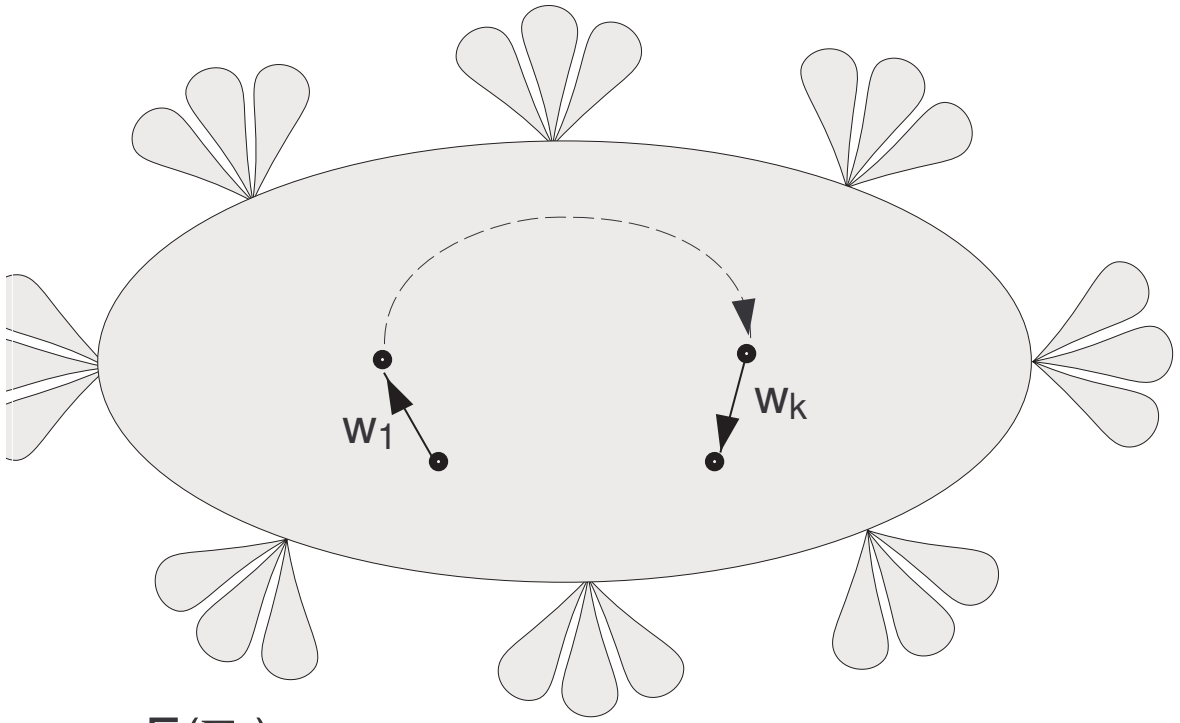
$$\Gamma_1 = S(E(\Gamma_0))$$



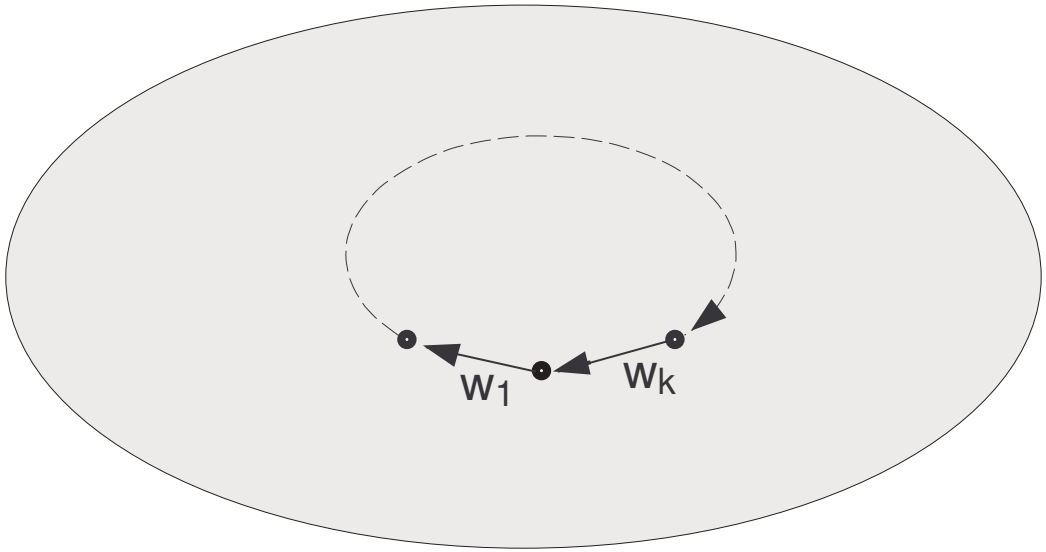
$E(\Gamma_1)$



$\Gamma_2 = S(E(\Gamma_1))$



$E(\Gamma_2)$



$$\Gamma_3 = S(E(\Gamma_2))$$

Iterative procedure: Inductively compute $\Gamma_i = S(E(\Gamma_{i-1}))$ until the original path w becomes a loop.

Time complexity of computing Γ_i is $O(|w|L(R)^m)$.

Standard generator of trivial words.

$$
\begin{array}{ll}
\varepsilon & \\
\downarrow & \\
c_1^{-1} r_1 c_1 & \equiv w_1^{(1)} w_2^{(1)} \\
\downarrow & \\
w_1^{(1)} c_2^{-1} r_2 c_2 w_2^{(1)} & \equiv w_1^{(2)} w_2^{(2)} \\
\downarrow & \\
w_1^{(2)} c_3^{-1} r_3 c_3 w_2^{(2)} & \equiv w_1^{(3)} w_2^{(3)} \\
\downarrow & \\
\ldots & \\
\downarrow & \\
w' & \\
\downarrow & \text{- freely reduce} \\
w &
\end{array}
$$

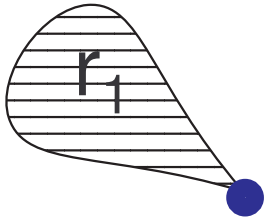
Complexity of the algorithm $\mathcal{A}_W$.

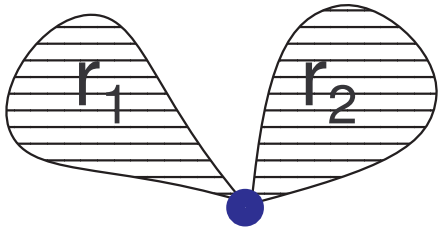
Theorem. The algorithm $\mathcal{A}_W$ solves the search word problem for G in

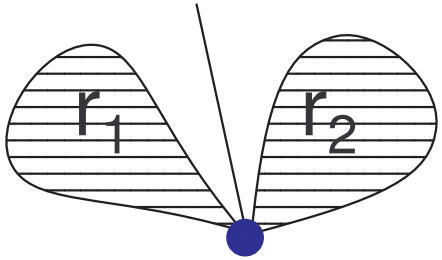
$$O(|w|L(R)^{\delta(w)})$$

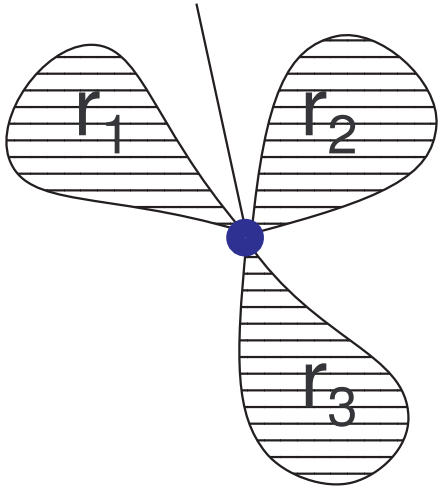
steps on an input $w =_G 1$, where $L(R)$ is the total length of the presentation of G, and $\delta(w)$ is the depth of w.

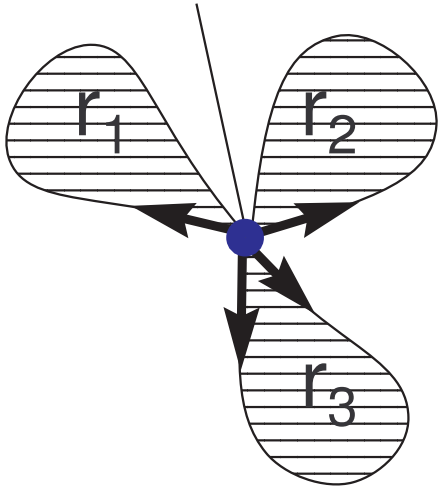
Random (locally stable) generator RG_S of van Kampen diagrams.

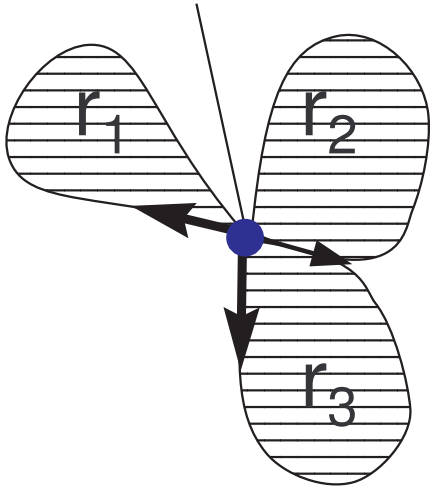

r1

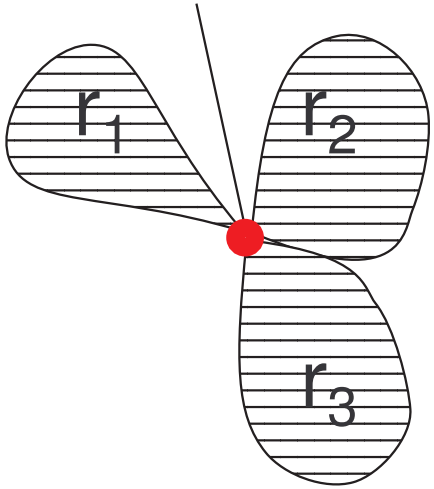

r1
r2

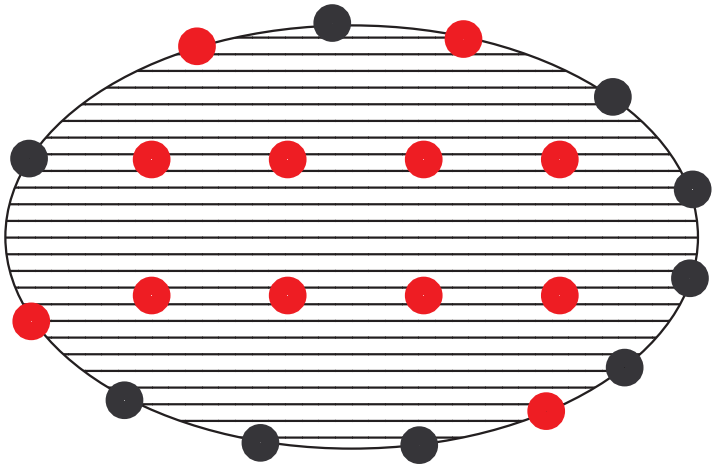

r1
r2

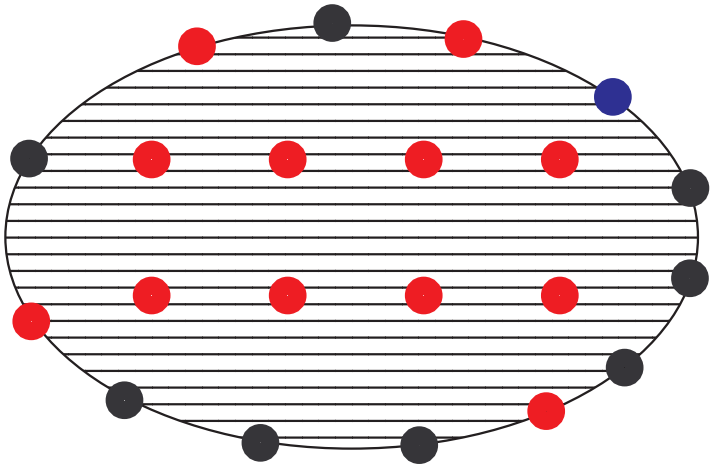

r1
r2
r3

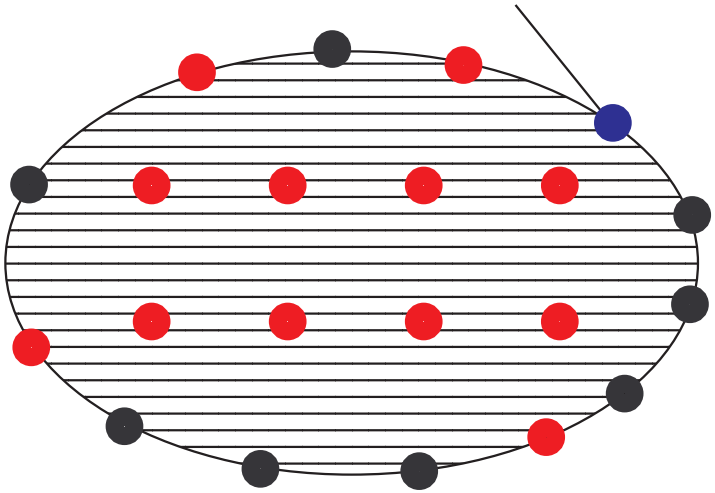

r1
r2
r3

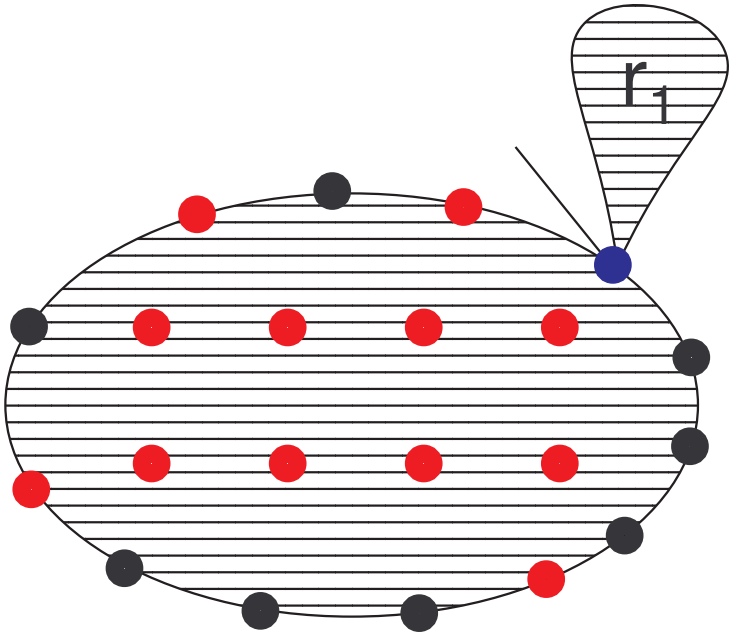

r1
r2
r3

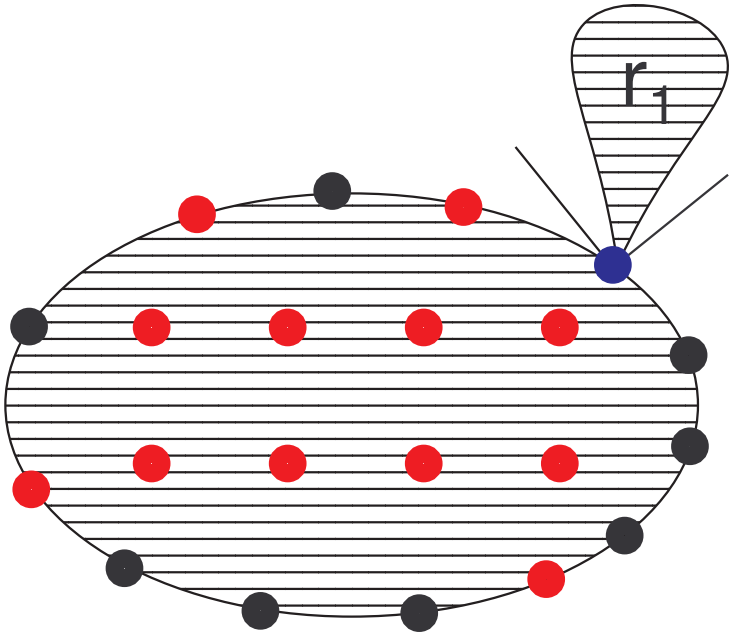

r1
r2
r3

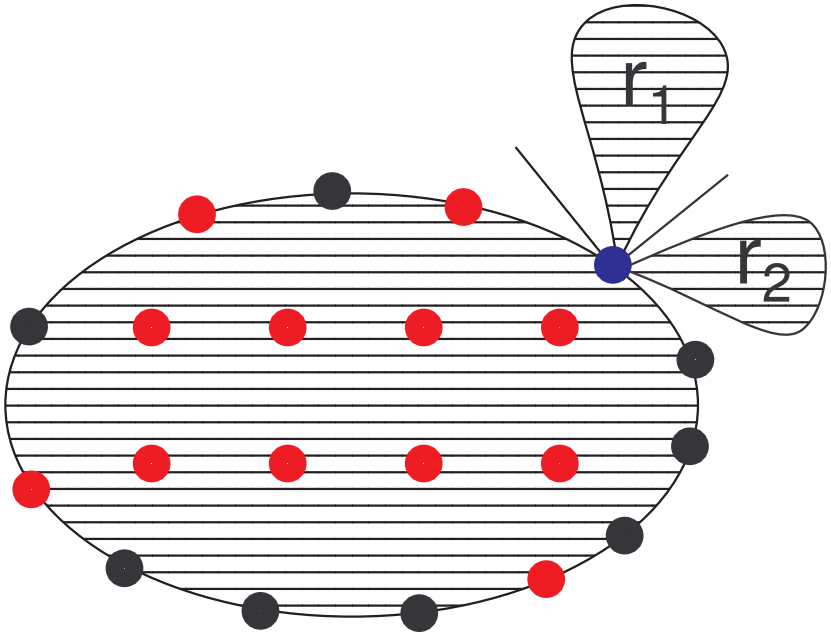


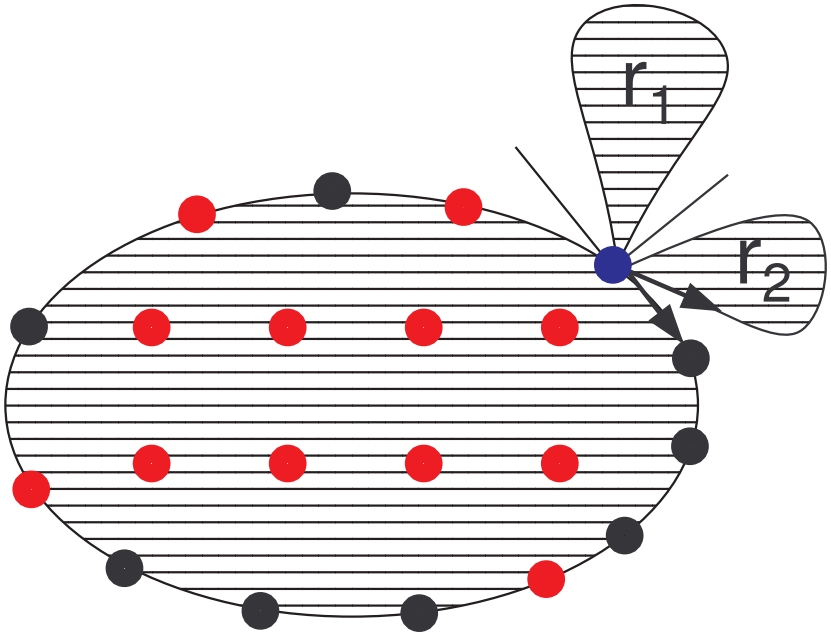


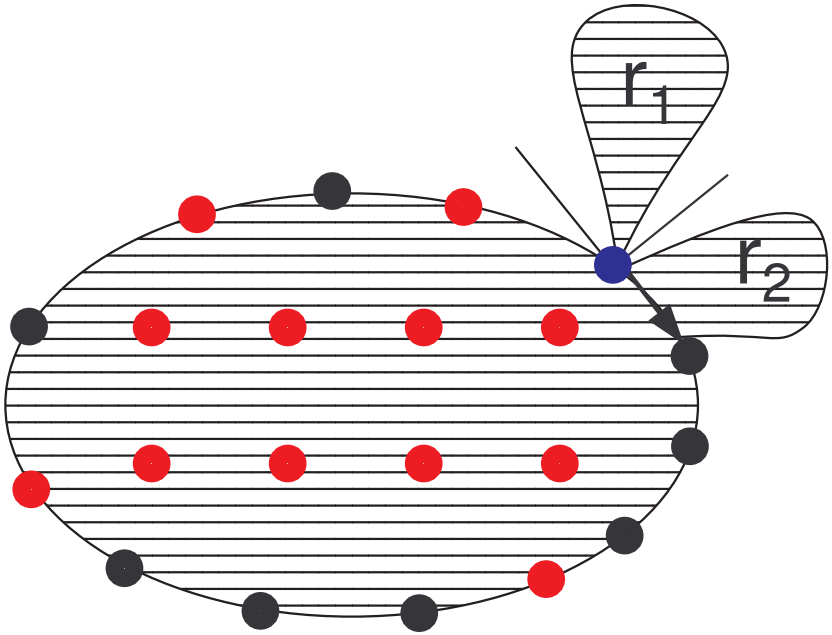


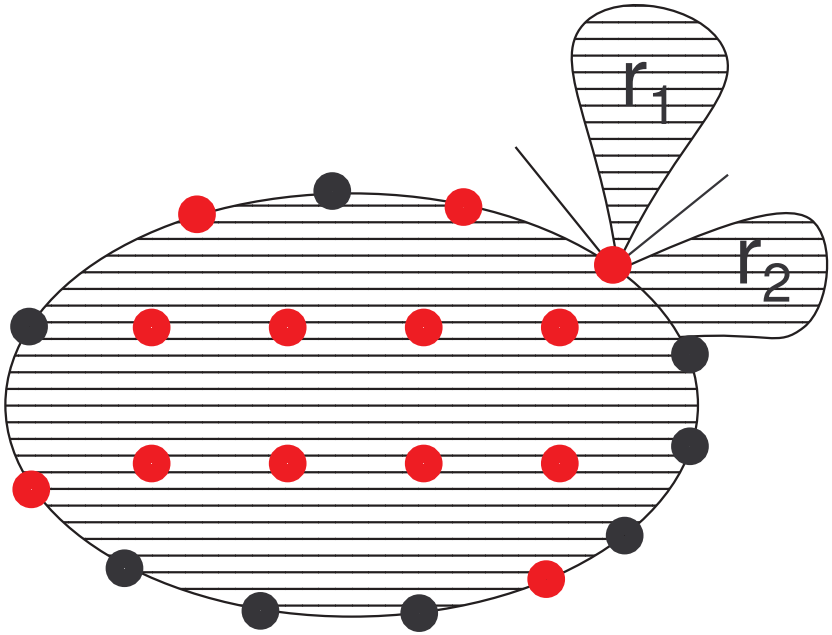

r1


r1


r1
r2


r1
r2


r1
r2


r1
r2

Completeness of RG_S.

Theorem. Algorithm RG_S produces any diagram $D \in \Omega$ with non trivial probability.

Transition graph of RG_S.

Let Φ be the set of all marked diagrams that can be produced by the Random Generator RG_S. We construct a directed graph $T = (V(T), E(T))$ of transitions of RG_S as follows:
1) $V(T) = \Phi$;
2) $E(T) = \{(D, D^*) \mid D \in V(T)\}$;
3) each edge (D, D^*) has an associated number $p(D, D^*)$ which is equal to the probability to obtain D^* from D according to Random Generator RG_S.

Lemma. The Random Generator RG_S induces a homogeneous no-return Markov's process $\mathcal{W}_S$ on Φ which starts at D_0. The locally finite graph T is the transition graph of $\mathcal{W}_S$.

Random walk $\mathcal{W}_S$ defines measure μ on van Kampen diagrams.

Generic properties of van Kampen diagrams.

Let X be a finite set (generators), R a finite set of words in the alphabet $(X \cup X^{-1})^*$ (relators),

$P = \langle X; R \rangle$ a finite presentation (of a group G).

$\mathcal{D}(P)$ - the set of all van Kampen diagrams over P equipped with the measure constructed above.

Our goal: to study generic properties of van Kampen diagrams over P.

A property $\mathcal{C}$ is **generic** if the set $\mathcal{D}_{\mathcal{C}}$ of van Kampen diagrams satisfying $\mathcal{C}$ is generic in $\mathcal{D}(P)$.

Linear isoperimetric inequalities and hyperbolicity

A van Kampen diagram D over P satisfies the **linear isoperimetric inequality** with coefficient c if

$$Area(D) \leq cL(D)$$

where $area(D)$ is the area of D and $L(D)$ is the length of the boundary (perimeter) of D.

Gromov: A group G given by a finite presentation P is hyperbolic if and only if there exists a constant c such that every reduced van Kampen diagram over P satisfies the linear isoperimetric inequality with the coefficient c.

Generic hyperbolicity

Theorem [M.-Ushakov] The set of all van Kampen diagrams satisfying the linear isoperimetric inequality with coefficient 4

$$area(D) \leq 4L(D)$$

is strongly generic in $\mathcal{D}(P)$.

Global and local generic hyperbolicity

Generic Global Hyperbolicity [Gromov]: A generic finitely presented group is hyperbolic.

Generic Local Hyperbolicity [M.–Ushakov]: Every finitely presented group is generically hyperbolic.

Generic depth of van Kampen diagrams.

Theorem [M.-Ushakov] The set of all van Kampen diagrams satisfying the logarithmic depth-area inequality

$$Depth(D) \leq \log Area(D)$$

is generic in $\mathcal{D}(P)$.

Corollary. The set of all van Kampen diagrams satisfying the logarithmic depth-perimeter inequality

$$Depth(D) \leq \log L(D) + \log 4$$

is generic in $\mathcal{D}(P)$.

Generic complexity of the word problem

Theorem [M.-Ushakov] For a given finitely presented group $G = \langle X; R\rangle$ the algorithm $\mathcal{A}_W$ solves the Search Word Problem in G generically in polynomial time.

Indeed, recall the complexity of the algorithm $\mathcal{A}_W$:

$$O(|w|L(R)^{Depth(w)})$$

where $L(R)$ is the total length of the presentation of G, and $Depth(w)$ is the minimal depth of van Kampen diagrams presenting w.

Structure of a random van Kampen diagram.