How to use entropy compression for existential proofs?

Louis Esperet (G-SCOP, Grenoble, France)
Aline Parreau (LIFL, Lille, France)

Liège, February 21st, 2013

Probabilistic method in Combinatorics

Idea

To prove the existence of an object with a property, show that a random object satifies the property with probability > 0.

Probabilistic method in Combinatorics

Idea

To prove the existence of an object with a property, show that a random object satifies the property with probability > 0.

Example: In a graph G with m edges, there is always a cut with $\frac{m}{2}$ edges.

Probabilistic method in Combinatorics

Idea

To prove the existence of an object with a property, show that a random object satifies the property with probability > 0.

Example: In a graph G with m edges, there is always a cut with $\frac{m}{2}$ edges.

Proof:

- Make a random partition $\{X, Y\}$ of the vertices: for each vertex, choose with probability $\frac{1}{2}$ if it is in X or in Y.
- An edge has probability $\frac{1}{2}$ to be in the cut.
- The average number of edges in the cut is $m/2$:

Probabilistic method in Combinatorics

Idea

To prove the existence of an object with a property, show that a random object satifies the property with probability > 0.

Example: In a graph G with m edges, there is always a cut with $\frac{m}{2}$ edges.

Proof:

- Make a random partition $\{X, Y\}$ of the vertices: for each vertex, choose with probability $\frac{1}{2}$ if it is in X or in Y.
- An edge has probability $\frac{1}{2}$ to be in the cut.
- The average number of edges in the cut is $m/2$:

$\rightarrow$ There exists a cut with $m/2$ edges.

Construction ?

Lovász Local Lemma

Local Lemma (symmetric version)

Let $A_1,\ldots,A_k$ be some 'bad' events. If:

- each event occurs with small probability, $Pr(A_i) \leq p$,
- each event is dependent of at most d events,
- $4pd \leq 1$,

$\Rightarrow$ with nonzero probability, no bad event occurs.

Lovász Local Lemma

Local Lemma (symmetric version)

Let $A_1,\ldots,A_k$ be some 'bad' events. If:

- each event occurs with small probability, $Pr(A_i) \leq p$,
- each event is dependent of at most d events,
- $4pd \leq 1$,

$\Rightarrow$ with nonzero probability, no bad event occurs.

Example: Any k-SAT instance where the support of each clause intersects at most 2^{k-2} other clauses is satifsiable.

Lovász Local Lemma

Local Lemma (symmetric version)

Let A_1,...,A_k be some 'bad' events. If:

- each event occurs with small probability, $Pr(A_i) \leq p$,
- each event is dependent of at most d events,
- $4pd \leq 1$,

$\Rightarrow$ with nonzero probability, no bad event occurs.

Example: Any k-SAT instance where the support of each clause intersects at most 2^{k-2} other clauses is satifsiable.

- Choose true or false for each variable with probability $1/2$.
- $A_i =$ clause i is not satisfied, $Pr(A_i) = 1/2^k$
- A_i is dependant with at most 2^{k-2} events.

Lovász Local Lemma

Local Lemma (symmetric version)

Let A_1,...,A_k be some 'bad' events. If:

- each event occurs with small probability, $Pr(A_i) \leq p$,
- each event is dependent of at most d events,
- $4pd \leq 1$,

$\Rightarrow$ with nonzero probability, no bad event occurs.

Example: Any k-SAT instance where the support of each clause intersects at most 2^{k-2} other clauses is satifsiable.

- Choose true or false for each variable with probability $1/2$.
- A_i = clause i is not satisfied, $Pr(A_i) = 1/2^k$
- A_i is dependant with at most 2^{k-2} events.

Construction ?

Constructive proof of Moser

- 2009, Moser at STACS conference, and then, Moser and Tardos:
  $\rightarrow$ algorithmic and constructive proof of LLL.
- Idea: entropy compression

"a random string cannot be compressed"

Constructive proof of Moser

- 2009, Moser at STACS conference, and then, Moser and Tardos:
  $\rightarrow$ algorithmic and constructive proof of LLL.
- Idea: entropy compression

  "a random string cannot be compressed"

- Used more specifically:
  $\rightarrow$ Non repetitive words (Grytczuk, Kozik and Micek, 2011)
  $\rightarrow$ Non repetitive colorings (Dujmović, Joret, Kozik and Wood, 2013)
  $\rightarrow$ Acyclic edge colorings of graphs (Esperet and P., 2013)

Non repetitive words

A sequence is non-repetitive if it does not contain a square uu.

Question:

What is the minimum size of an alphabet on which we can make arbitrary large non-repetitive sequences ?

With two letters ? aba

a

Non repetitive words

A sequence is non-repetitive if it does not contain a square uu.

Question:

What is the minimum size of an alphabet on which we can make arbitrary large non-repetitive sequences ?

With two letters ? *aba*

Theorem Thue, 1906

The morphism $a \to abcab$, $b \to acabcb$, $c \to acbcacb$ is stable on non-repetitive sequences on $\{a, b, c\}$

a $\to$ abcab

Non repetitive words

A sequence is non-repetitive if it does not contain a square uu.

Question:

What is the minimum size of an alphabet on which we can make arbitrary large non-repetitive sequences ?

With two letters ? aba

Theorem Thue, 1906

The morphism $a \to abcab$, $b \to acabcb$, $c \to acbcacb$ is stable on non-repetitive sequences on $\{a, b, c\}$

a $\to$ abcab $\to$ abcabacabcbacbcacbabcabacabcb

Non repetitive words

A sequence is non-repetitive if it does not contain a square uu.

Question:

What is the minimum size of an alphabet on which we can make arbitrary large non-repetitive sequences ?

With two letters ? aba

Theorem Thue, 1906

The morphism $a \to abcab$, $b \to acabcb$, $c \to acbcacb$ is stable on non-repetitive sequences on $\{a, b, c\}$

$\mathrm{a} \to \mathrm{abcab} \to \mathrm{abcabacabcbacbcacbabcabacabcb} \to \cdots$

List version

Question:

What is the smallest k such that, for all n, if L_1, L_2,...,L_n are n lists of k letters, there always exists a non-repetitive sequence $a_1 ... a_n$ with $a_i \in L_i$, for all i ?

Example: $L_1 = \{a, b, c\}$, $L_2 = \{a, b, d\}$, $L_3 = \{a, c, d\}$, $L_4 = \{b, c, d\}$

abcd is non-repetitive !

List version

Question:

What is the smallest k such that, for all n, if L_1, L_2,...,L_n are n lists of k letters, there always exists a non-repetitive sequence $a_1 \ldots a_n$ with $a_i \in L_i$, for all i ?

Example: $L_1 = \{a, b, c\}$, $L_2 = \{a, b, d\}$, $L_3 = \{a, c, d\}$, $L_4 = \{b, c, d\}$

abcd is non-repetitive !

Theorem Grytczuk, Przybylo and Zhu, 2010

It is possible to extract a non-repetitive sequence for any sequence of lists of size 4.

Proof

Algorithm:

- The sequence is constructed from left to right.
- Choose a_i randomly in L_i.
- If a square is created, remove the second part.

Example : with $L_1 = L_2 = L_3 = L_4$

Proof

Algorithm:

- The sequence is constructed from left to right.
- Choose a_i randomly in L_i.
- If a square is created, remove the second part.

Example : with $L_1 = L_2 = L_3 = L_4$

Theorem Grytczuk, Kozik and Micek, 2011

This algorithm stops with probability > 0.

Proof

Theorem Grytczuk, Kozik and Micek, 2011

This algorithm stops with probability > 0.

- Assume it is still running after t steps $\rightarrow$ bad scenario.
- At each step, we record the number of letters we remove.

Proof

Theorem Grytczuk, Kozik and Micek, 2011

This algorithm stops with probability > 0.

- Assume it is still running after t steps $\rightarrow$ bad scenario.
- At each step, we record the number of letters we remove.
- Number of records ? $\rightarrow$ Catalan number.

Proof

Theorem Grytczuk, Kozik and Micek, 2011

This algorithm stops with probability > 0.

- Assume it is still running after t steps $\rightarrow$ bad scenario.
- At each step, we record the number of letters we remove.
- Number of records ? $\rightarrow$ Catalan number.
- The record i enough to recover the whole history:

$$1 \text{ record} + \text{final sequence} = 1 \text{ bad scenario}$$

Proof

Theorem Grytczuk, Kozik and Micek, 2011

This algorithm stops with probability > 0.

- Assume it is still running after t steps $\rightarrow$ bad scenario.
- At each step, we record the number of letters we remove.
- Number of records ? $\rightarrow$ Catalan number.
- The record i enough to recover the whole history:

$$1 \text{ record} + \underset{\underset{n4^n}{\uparrow}}{\text{final sequence}} = 1 \text{ bad scenario}$$

Proof

Theorem Grytczuk, Kozik and Micek, 2011

This algorithm stops with probability > 0.

- Assume it is still running after t steps $\rightarrow$ bad scenario.
- At each step, we record the number of letters we remove.
- Number of records ? $\rightarrow$ Catalan number.
- The record i enough to recover the whole history:

$$\underbrace{1 \text{ record}}_{4^t/\sqrt{\pi}t^{3/2}} + \underbrace{\text{final sequence}}_{n4^n} = 1 \text{ bad scenario}$$

Proof

Theorem Grytczuk, Kozik and Micek, 2011

This algorithm stops with probability > 0.

- Assume it is still running after t steps $\to$ bad scenario.
- At each step, we record the number of letters we remove.
- Number of records ? $\to$ Catalan number.
- The record i enough to recover the whole history:

$$1 \text{ record} + \text{final sequence} = 1 \text{ bad scenario}$$

$$4^t/\sqrt{\pi}t^{3/2} \qquad n4^n \qquad \frac{n4^n}{\sqrt{\pi}t^{3/2}} \cdot 4^t$$

Proof

Theorem Grytczuk, Kozik and Micek, 2011

This algorithm stops with probability > 0.

- Assume it is still running after t steps $\rightarrow$ bad scenario.
- At each step, we record the number of letters we remove.
- Number of records ? $\rightarrow$ Catalan number.
- The record i enough to recover the whole history:

$$1 \text{ record} + \text{final sequence} = 1 \text{ bad scenario}$$

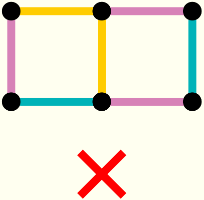
$$4^t/\sqrt{\pi}t^{3/2} \qquad n4^n \qquad \frac{n4^n}{\sqrt{\pi}t^{3/2}} \cdot 4^t$$

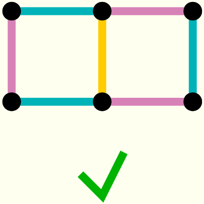
- For large t, number of bad scenarios $\leq 4^t$.

$\Rightarrow$ Some scenarios are good !

Another example : Acyclic edge coloring of graphs

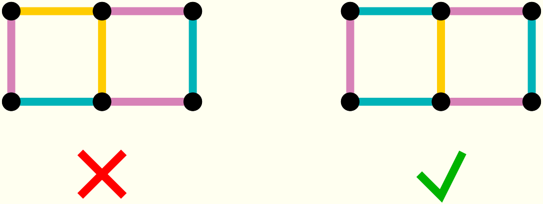
A proper edge coloring of a graph is a coloring of the edges such that two edges sharing a vertex have different colors.





Another example : Acyclic edge coloring of graphs

A proper edge coloring of a graph is a coloring of the edges such that two edges sharing a vertex have different colors.



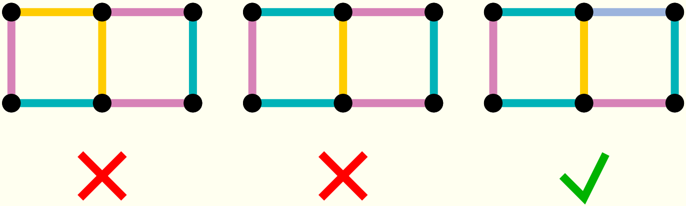
Theorem Vizing, 1964

If G has maximum degree Δ, there is a proper edge coloring in $\Delta + 1$ colors.

Acyclic edge coloring of graphs

An acyclic edge coloring of a graph is:

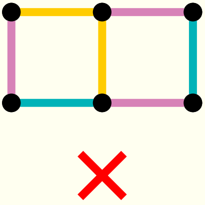
- a proper edge coloring,
- without bicolored cycles.



Acyclic edge coloring of graphs

An acyclic edge coloring of a graph is:

- a proper edge coloring,
- without bicolored cycles.

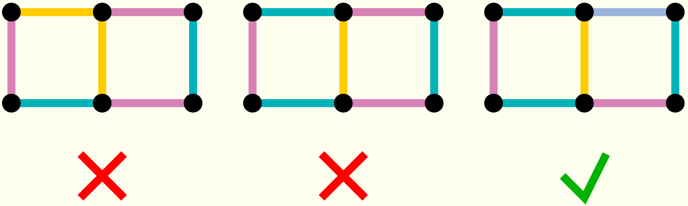


- $a'(G)$: minimum number of colors in an acyclic edge coloring of G.
- If G has maximum degree Δ, $a'(G) \geq \Delta$.

Acyclic edge coloring of graphs

An acyclic edge coloring of a graph is:

- a proper edge coloring,
- without bicolored cycles.



- $a'(G)$: minimum number of colors in an acyclic edge coloring of G.
- If G has maximum degree Δ, $a'(G) \geq \Delta$.

Conjecture Alon, Sudakov and Zaks, 2001

If G has maximum degree Δ, $a'(G) \leq \Delta + 2$.

Results

Using the Lovász Local Lemma and variations:

- $a'(G) \leq 64\Delta$ (Alon, McDiarmid and Reed, 1991)
- $a'(G) \leq 16\Delta$ (Molloy and Reed, 1998)
- $a'(G) \leq 9.62\Delta$ (Ndreca, Procacci and Scoppola, 2012)

Results

Using the Lovász Local Lemma and variations:

- $a'(G) \leq 64\Delta$ (Alon, McDiarmid and Reed, 1991)
- $a'(G) \leq 16\Delta$ (Molloy and Reed, 1998)
- $a'(G) \leq 9.62\Delta$ (Ndreca, Procacci and Scoppola, 2012)

Using entropy compression :

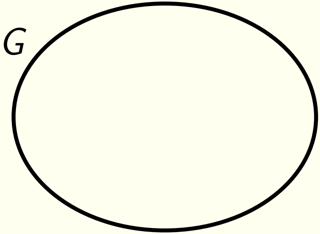
Theorem Esperet and P., 2013

If G has maximum degree Δ, $a'(G) \leq 4\Delta$.

Algorithm

Order the edge set.
While there is an uncolored edge:

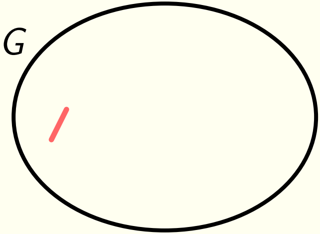
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

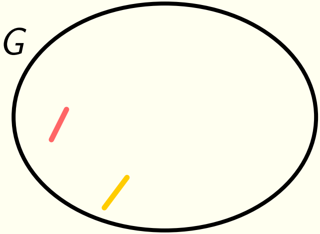
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

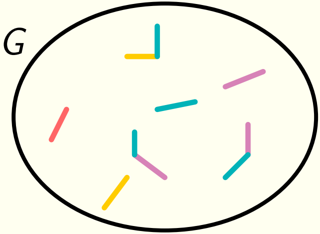
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

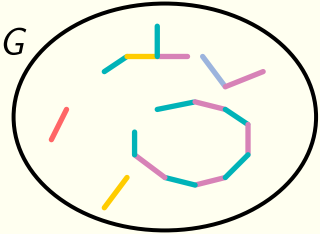
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

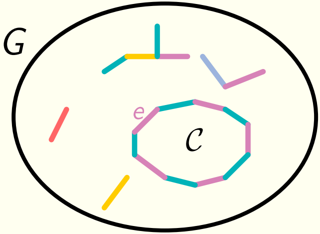
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

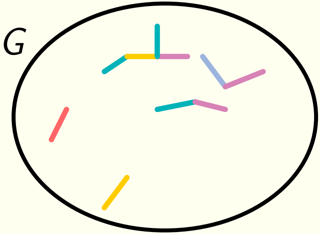
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

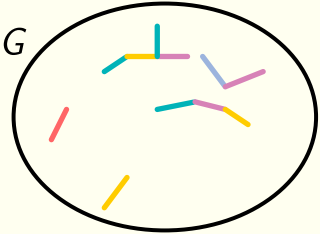
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

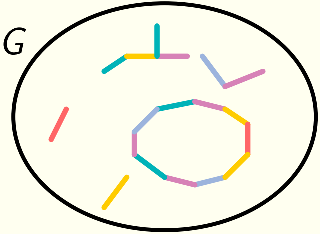
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

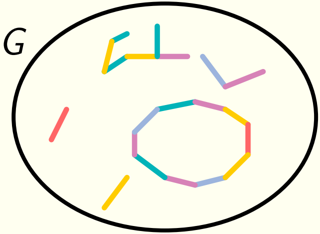
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

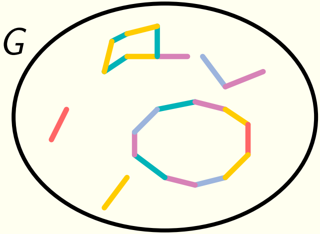
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

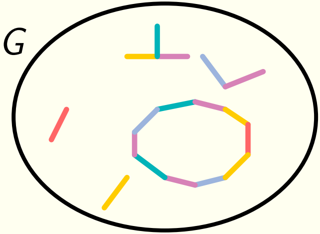
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

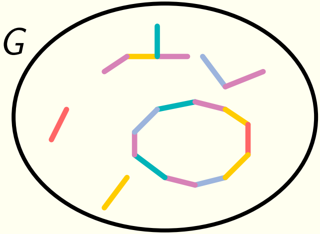
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



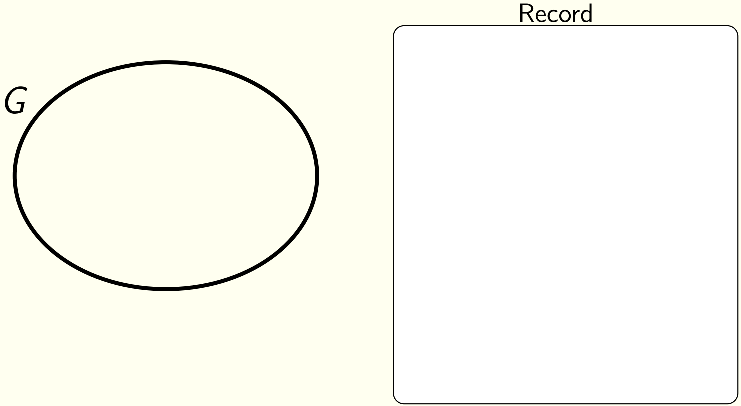
We prove that this algorithm ends with non zero probability.
$\Rightarrow$ Any graph has an acyclic edge coloring with 4Δ colors.

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario

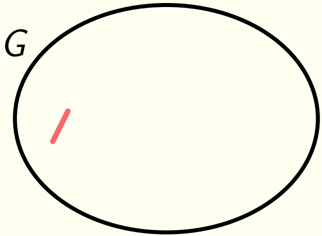
Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.



Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

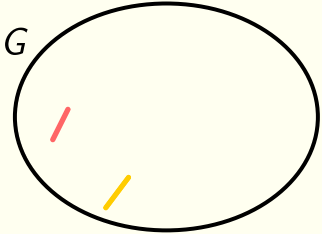


Record

```
1:-
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

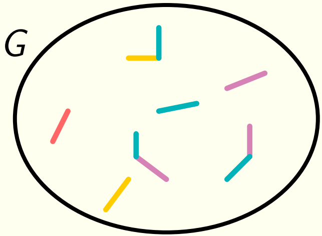


Record

```
1:-
2:-
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

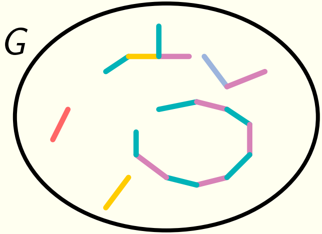


Record

```
1:-
2:-
...
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

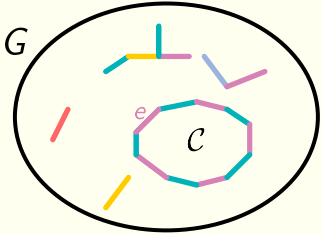


Record

```
1:-
2:-
...
17:-
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

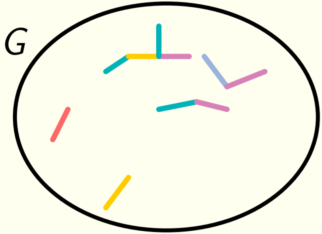


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

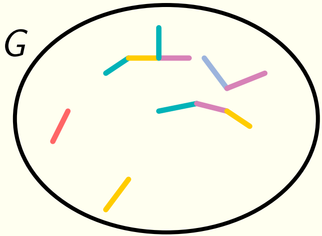


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

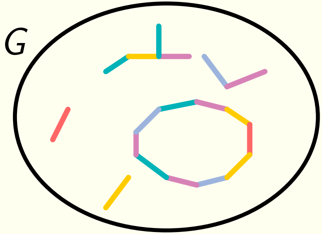


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

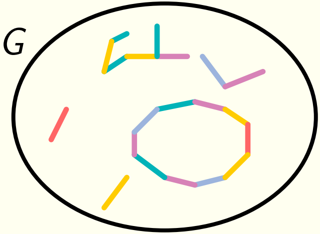


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

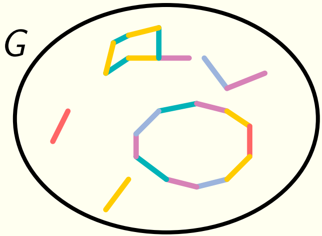


Record

1:-
2:-
...
17:-
18:Cycle $\mathcal{C}$ is uncolored
19:-
...
276:-

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

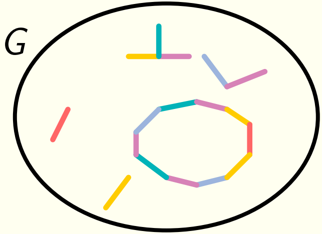


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
276:-
277:Cycle 𝒞′ is uncolored
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

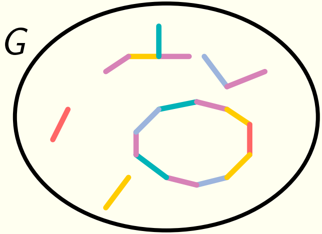


Record

1:-
2:-
...
17:-
18:Cycle $\mathcal{C}$ is uncolored
19:-
...
276:-
277:Cycle $\mathcal{C}'$ is uncolored

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

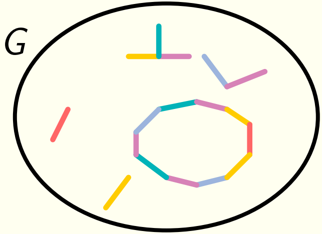


Record

1:-
2:-
...
17:-
18:Cycle $\mathcal{C}$ is uncolored
19:-
...
276:-
277:Cycle $\mathcal{C}'$ is uncolored
278:-
...

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.



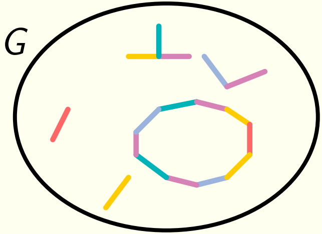
Final partial coloring Φ_t

Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
276:-
277:Cycle 𝒞′ is uncolored
278:-
...
t:-
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.



Final partial coloring Φ_t

Record

1:-
2:-
...
17:-
18:Cycle $\mathcal{C}$ is uncolored
19:-
...
276:-
277:Cycle $\mathcal{C}'$ is uncolored
278:-
...
t:-

1 record + 1 final partial coloring = 1 bad scenario

Rewrite the history I

- X_i: set of uncolored edges after step i
- reading of the record to get X_i:
  - $X_0 = E$

Rewrite the history I

- X_i: set of uncolored edges after step i
- reading of the record to get X_i:
  - $X_0 = E$
  - Step i:

2 cases

`i:-` $\rightarrow$ $X_{i+1} = X_i - \{\text{smallest edge of } X_i\}$

`i:`$\mathcal{C}$ `is uncolored` $\rightarrow$ $X_{i+1} = X_i + \{\mathcal{C} \text{ except two edges}\}$

Rewrite the history I

- X_i: set of uncolored edges after step i
- reading of the record to get X_i:
  - $X_0 = E$
  - Step i:

2 cases

`i:-` $\rightarrow$ $X_{i+1} = X_i - \{\text{smallest edge of } X_i\}$

`i:`$\mathcal{C}$ `is uncolored` $\rightarrow$ $X_{i+1} = X_i + \{\mathcal{C} \text{ except two edges}\}$

With the record, we can find the edge e_i which is colored at step i.

Rewrite the history II: partial colorings

- Φ_i: partial coloring after step i
- Inverse reading of the record to get Φ_i:
  - Φ_t is known

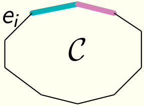
Rewrite the history II: partial colorings

- Φ_i: partial coloring after step i
- Inverse reading of the record to get Φ_i:
  - Φ_t is known
  - Step i:

2 cases

`i:-` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with e_i uncolored

`i:`$\mathcal{C}$ `is uncolored` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with $\mathcal{C}$ recolored and e_i uncolored



Step i

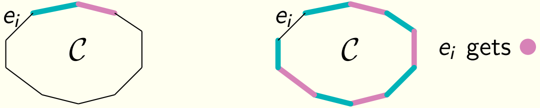
Rewrite the history II: partial colorings

- Φ_i: partial coloring after step i
- Inverse reading of the record to get Φ_i:
  - Φ_t is known
  - Step i:

2 cases

`i:-` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with e_i uncolored

`i:`$\mathcal{C}$` is uncolored` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with $\mathcal{C}$ recolored and e_i uncolored



Step i Step $i-1$

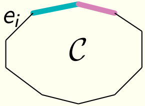
Rewrite the history II: partial colorings

- Φ_i: partial coloring after step i
- Inverse reading of the record to get Φ_i:
  - Φ_t is known
  - Step i:

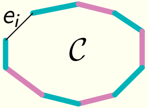
2 cases

`i:-` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with e_i uncolored

`i:`$\mathcal{C}$ `is uncolored` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with $\mathcal{C}$ recolored and e_i uncolored



Step i



Step $i-1$

e_i gets ●

With Φ_t and the record, we can find the partial colorings and the scenario.

Rewrite the history - Summary

1. Top-down reading $\rightarrow$ set of colored edges at each step.

1:-
2:-
...
17:-
18:$\mathcal{C}$ is uncolored
19:-
...
276:-
277:$\mathcal{C}'$ is uncolored
278:-
...
t:-

1
Sets of
colored edges

Rewrite the history - Summary

1. Top-down reading $\rightarrow$ set of colored edges at each step.
2. Buttum-up reading$\rightarrow$ partial coloring at each step and scenario.

1:-
2:-
...
17:-
18:$\mathcal{C}$ is uncolored
19:-
...
276:-
277:$\mathcal{C}'$ is uncolored
278:-
...
t:-

1
Sets of
colored edges

2
partial colorings
and scenario

Rewrite the history - Summary

1. Top-down reading $\rightarrow$ set of colored edges at each step.
2. Buttum-up reading$\rightarrow$ partial coloring at each step and scenario.

```
1:-
2:-
...
17:-
18:𝒞 is uncolored
19:-
...
276:-
277:𝒞′ is uncolored
278:-
...
t:-
```

1 Sets of colored edges

2 partial colorings and scenario

$\Rightarrow$ 1 record + 1 final partial coloring = 1 bad scenario

Summary

1 record $+1$ partial coloring $=$ 1 bad scenario

Summary

$$1 \text{ record } + 1 \text{ partial coloring} = 1 \text{ bad scenario}$$

$$\leq (4\Delta + 1)^m$$

Summary

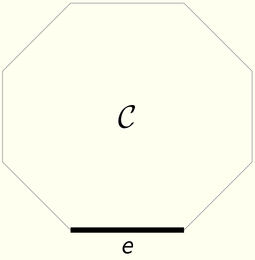
$$1 \text{ record } + 1 \text{ partial coloring} = 1 \text{ bad scenario}$$

? $\qquad \leq (4\Delta + 1)^m \qquad$?

How many possible records ?

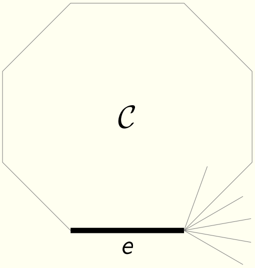
Compact records of cycles

- We know one edge e of $\mathcal{C}$.



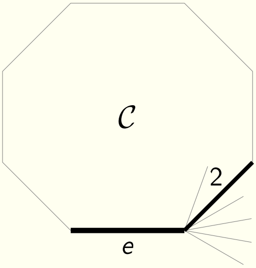
Compact records of cycles

- We know one edge e of $\mathcal{C}$.



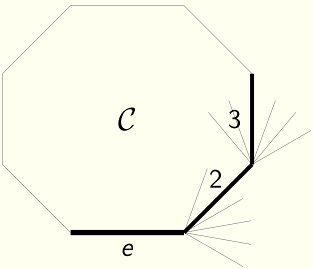
Compact records of cycles

- We know one edge e of $\mathcal{C}$.



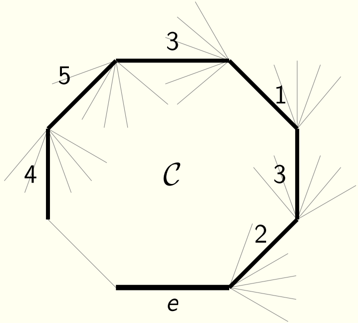
Compact records of cycles

- We know one edge e of $\mathcal{C}$.



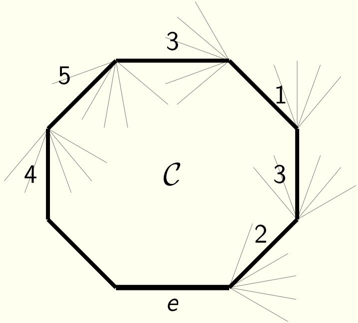
Compact records of cycles

- We know one edge e of $\mathcal{C}$.
- No choice for the last edge



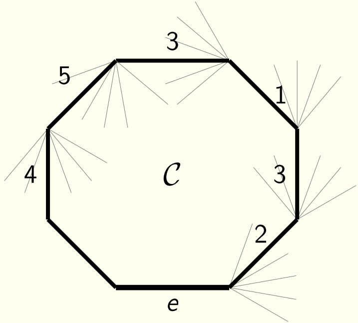
Compact records of cycles

- We know one edge e of $\mathcal{C}$.
- No choice for the last edge



Compact records of cycles

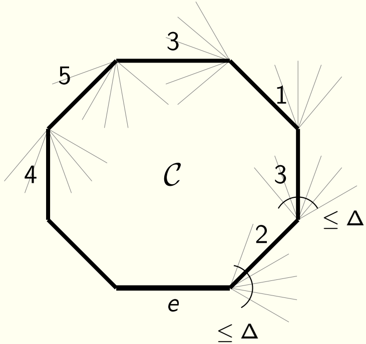
- We know one edge e of $\mathcal{C}$.
- No choice for the last edge



`i:`$\mathcal{C}$` is uncolored` $\Leftrightarrow$ `i:231354`

Compact records of cycles

- We know one edge e of $\mathcal{C}$.
- No choice for the last edge



i:$\mathcal{C}$ is uncolored $\Leftrightarrow$ i:231354

- Cycle coded by a word on $\{1, ..., \Delta\}^{2k-2}$ where $2k$ is the length of $\mathcal{C}$.

Number of records

Record $\quad (- , - , ..., - , 231354 , - , ..., - , 4213 , - , ..., -)$

Number of records

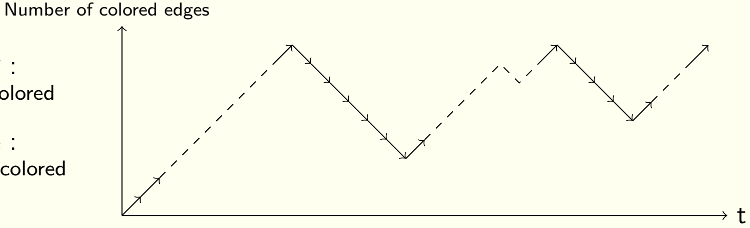
$$\begin{array}{llllllllllll}
\text{Record} & (- , & - , & \ldots, & - , & 231354 , & - , & \ldots, & - , & 4213 , & - , & \ldots, & -) \\
& \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & & \downarrow \\
& 0 & 0 & & 0 & 0111111 & 0 & & 0 & 01111 & 0 & & 0
\end{array}$$

Number of records

Record	(− ,	− ,	...,	− ,	231354 ,	− ,	...,	− ,	4213 ,	− ,	...,	−)
	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$
	0	0		0	0111111	0		0	01111	0		0

$0 \leftrightarrow \nearrow$:
an edge is colored

$1 \leftrightarrow \searrow$:
an edge is uncolored

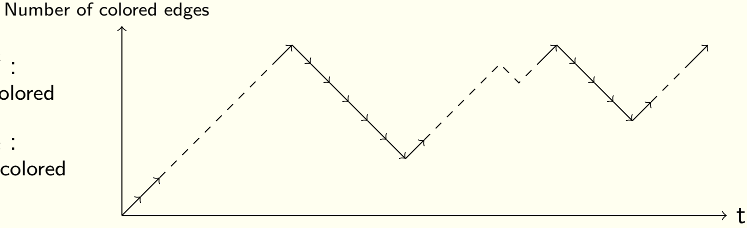


Number of records

$$
\begin{array}{lccccccccccccccc}
\text{Record} & (- , & - , & \ldots, & - , & 231354 , & - , & \ldots, & - , & 4213 , & - , & \ldots, & -) \\
 & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow & \downarrow & & \downarrow \\
 & 0 & 0 & & 0 & 0111111 & 0 & & 0 & 01111 & 0 & & 0
\end{array}
$$

$0 \leftrightarrow \nearrow$:
an edge is colored

$1 \leftrightarrow \searrow$:
an edge is uncolored



Partial Dyck word of length $\leq 2t$ and descents of even size .

Number of records

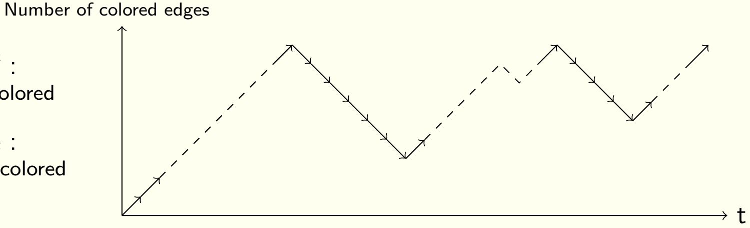
Record $(- , - , \ldots, - , 231354 , - , \ldots, - , 4213 , - , \ldots, -)$

$\downarrow$

$0 \quad 0 \quad 0 \quad 0111111 \quad 0 \quad 0 \quad 01111 \quad 0 \quad 0$

$0 \leftrightarrow \nearrow$:
an edge is colored

$1 \leftrightarrow \searrow$:
an edge is uncolored



Partial Dyck word of length $\leq 2t$ and descents of even size > 2.

Number of records

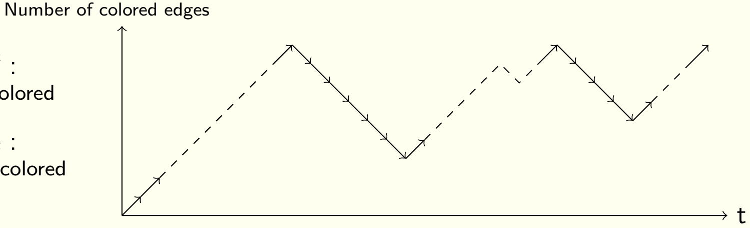
Record $(- , - , ..., - , 231354 , - , ..., - , 4213 , - , ..., -)$

$\downarrow$

$0 \quad 0 \quad 0 \quad 0111111 \quad 0 \quad 0 \quad 01111 \quad 0 \quad 0$

$0 \leftrightarrow \nearrow$:
an edge is colored

$1 \leftrightarrow \searrow$:
an edge is uncolored



Partial Dyck word of length $\leq 2t$ and descents of even size > 2.

$\rightarrow$ Number of such words : $2^t/t^{3/2}$

Number of records

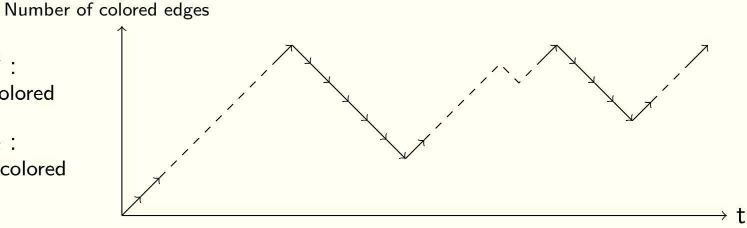
Record $(- , - , \ldots, - , 231354 , - , \ldots, - , 4213 , - , \ldots, -)$

$\downarrow$

$0 \quad 0 \quad 0 \quad 0111111 \quad 0 \quad 0 \quad 01111 \quad 0 \quad 0$

$0 \leftrightarrow \nearrow$:
an edge is colored

$1 \leftrightarrow \searrow$:
an edge is uncolored



Partial Dyck word of length $\leq 2t$ and descents of even size > 2.

$\rightarrow$ Number of such words : $2^t/t^{3/2}$

$\rightarrow$ Number of records : $(2\Delta)^t/t^{3/2}$

End of the proof

1 record $+1$ partial coloring $=$ 1 bad scenario

$(4\Delta + 1)^m$

End of the proof

1 record $+1$ partial coloring $=$ 1 bad scenario

$(2\Delta)^t/t^{3/2}$ $\qquad$ $(4\Delta+1)^m$

End of the proof

$$1 \text{ record } + 1 \text{ partial coloring} = 1 \text{ bad scenario}$$

$(2\Delta)^t/t^{3/2}$ $\qquad$ $(4\Delta+1)^m$ $\qquad$ $\frac{(4\Delta+1)^m(2\Delta)^t}{t^{3/2}}$

End of the proof

$$1 \text{ record } + 1 \text{ partial coloring} = 1 \text{ bad scenario}$$

$(2\Delta)^t / t^{3/2}$ $\qquad$ $(4\Delta + 1)^m$ $\qquad$ $\frac{(4\Delta+1)^m (2\Delta)^t}{t^{3/2}}$

- Number of scenarios: $(2\Delta)^t$
- Number of bad scenarios: $\frac{(4\Delta+1)^m (2\Delta)^t}{t^{3/2}} = o((2\Delta)^t)$

End of the proof

$$1 \text{ record } +1 \text{ partial coloring} = 1 \text{ bad scenario}$$

$(2\Delta)^t/t^{3/2}$ $\qquad$ $(4\Delta+1)^m$ $\qquad$ $\frac{(4\Delta+1)^m(2\Delta)^t}{t^{3/2}}$

- Number of scenarios: $(2\Delta)^t$
- Number of bad scenarios: $\frac{(4\Delta+1)^m(2\Delta)^t}{t^{3/2}} = o((2\Delta)^t)$

$\Rightarrow$ For t large enough, there are good scenarios.

$\Leftrightarrow$ The algorithm stops with nonzero probability !
$\Leftrightarrow$ There is a coloring in 4Δ colors.

Algorithmic aspect

- To have a small propability of a bad event in t steps, we should have:

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4\Delta+1)^m(2\Delta)^t/t^{3/2}}{(2\Delta)^t} < \delta$$

  Equivalently:

$$t^{3/2} > \frac{(4\Delta+1)^m}{\delta}$$

$\rightarrow$ t can be exponential in m.

Algorithmic aspect

- To have a small propability of a bad event in t steps, we should have:

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4\Delta+1)^m (2\Delta)^t / t^{3/2}}{(2\Delta)^t} < \delta$$

  Equivalently:

$$t^{3/2} > \frac{(4\Delta+1)^m}{\delta}$$

$\rightarrow$ t can be exponential in m.

- But if we have $4\Delta + 1$ colors :

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4\Delta+2)^m (2\Delta)^t / t^{3/2}}{((2\Delta+1)^t} < \delta$$

Algorithmic aspect

- To have a small propability of a bad event in t steps, we should have:

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4\Delta + 1)^m (2\Delta)^t / t^{3/2}}{(2\Delta)^t} < \delta$$

Equivalently:

$$t^{3/2} > \frac{(4\Delta + 1)^m}{\delta}$$

$\rightarrow$ t can be exponential in m.

- But if we have $4\Delta + 1$ colors :

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4\Delta + 2)^m (2\Delta)^t / t^{3/2}}{((2\Delta + 1)^t} < \delta$$

$\rightarrow$ t is polynomial in m.

With larger girth

Girth: size of the smallest cycle in G.

With the same method, we get better bounds if the girth is $\geq \ell$

$\Leftrightarrow$ All the uncolored cycles have size at least ℓ

$\Leftrightarrow$ All the descents in the Dyck word have size $2k$ for some $k \geq \ell/2$

With larger girth

Girth: size of the smallest cycle in G.

With the same method, we get better bounds if the girth is $\geq \ell$

$\Leftrightarrow$ All the uncolored cycles have size at least ℓ

$\Leftrightarrow$ All the descents in the Dyck word have size $2k$ for some $k \geq \ell/2$

There are fewer Dyck words !

$\rightarrow$ Analytic combinatorics and generating function to count Dyck Words.

Conclusion

Entropy compression ?

- Input: large random vector
- Output: smaller record

Works well since :

- we can remove a lot of letters/colors
  $\rightarrow$ add entropy;
- while being able to recover the sequence/coloring with a small record
  $\rightarrow$ compression.

Conclusion

Entropy compression ?

- Input: large random vector
- Output: smaller record

Works well since :

- we can remove a lot of letters/colors
  $\rightarrow$ add entropy;
- while being able to recover the sequence/coloring with a small record
  $\rightarrow$ compression.

Thanks !