# Algebraic approach to exact algorithms, Part III: Polynomials over finite fields of characteristic two

Łukasz Kowalik

University of Warsaw

ADFOCS, Saarbrücken, August 2013

# The Schwartz-Zippel Lemma

## Lemma [DeMillo and Lipton 1978, Zippel 1979, Schwartz 1980]

Let $p(x_1, x_2, \ldots, x_n)$ be a non-zero polynomial of degree at most $d$ over a field $F$ and let $S$ be a finite subset of $F$. Sample values $a_1, a_2, \ldots, a_n$ from $S$ uniformly at random. Then,

$$Pr[p(a_1, a_2, \ldots, a_n)] = 0] \leq d/|S|.$$

## A typical application

- We can efficiently **evaluate** a polynomial $p$ of degree $d$.
- We want to test whether $p$ is a non-zero polynomial.
- Then, we pick $S$ so that $|S| \geq 2d$ and we evaluate $p$ on a random vector $\mathbf{x} \in S^n$. We answer YES iff we got $p(\mathbf{x}) \neq 0$.
- If $p$ is the zero polynomial we always get NO, otherwise we get YES with probability at least $\frac{1}{2}$.
- This is called a Monte-Carlo algorithm with one-sided error.

# The Schwartz-Zippel Lemma: Example

## Polynomial equality testing

Input: Two multivariate polynomials $P, Q$ given as an arithmetic circuit.
Question: Does $P \equiv Q$?

**Note:** A polynomial described by an arithmetic circuit of size $s$ can have $2^{\Omega(s)}$ different monomials: $(x_1 + x_2)(x_1 - x_3)(x_2 + x_4) \cdots$.

# The Schwartz-Zippel Lemma: Example

## Polynomial equality testing

Input: Two multivariate polynomials $P, Q$ given as an arithmetic circuit.
Question: Does $P \equiv Q$?

**Note:** A polynomial described by an arithmetic circuit of size $s$ can have $2^{\Omega(s)}$ different monomials: $(x_1 + x_2)(x_1 - x_3)(x_2 + x_4)\cdots$.

## Solution

Test whether the polynomial $P - Q$ is non-zero using the Schwartz-Zippel Lemma.

## Theorem

Polynomial equality testing for two polynomials represented by circuits of size at most $s$ can be solved in $O(s)$ time with a Monte Carlo algorithm with one-sided error probability bounded by $1/2$.

# Question

## Question

What if the bound of $1/2$ for the probability of success is not enough for us?

# Question

## Question

What if the bound of $1/2$ for the probability of success is not enough for us?

## Answer

Repeat the algorithm 1000 times and answer YES if there was **at least one** YES. Then,

$$Pr[error] \leq \frac{1}{2^{1000}}$$

## Note

The probability that an earthquake destroys the computer is probably higher than $\frac{1}{2^{1000}}\ldots$

# Finite fields of characteristic 2

In what follows, we use finite fields of size $2^k$.
We need to know just three things about such fields:

- They exist (for every $k \in \mathbb{N}$),
- We can perform arithmetic operations fast, in $O(k \log k \log \log k)$ time,
- They are of characteristic two, i.e. $1 + 1 = 0$.
- In particular, for any element $a$, we have

$$a + a = a \cdot (1 + 1) = a \cdot 0 = 0$$

# $k$-path problem

## Problem

Input: directed/undirected graph $G$, integer $k$.
Question: Does $G$ contain a path of length $k$?

## A few facts

- NP-complete (why?)

# $k$-path problem

## Problem

Input: directed/undirected graph $G$, integer $k$.
Question: Does $G$ contain a path of length $k$?

## A few facts

- NP-complete (why?)
- even $O(f(k)n^{O(1)})$-time algorithm is non-trivial,

# $k$-path problem

## Problem

Input: directed/undirected graph $G$, integer $k$.
Question: Does $G$ contain a path of length $k$?

## A few facts

- NP-complete (why?)
- even $O(f(k)n^{O(1)})$-time algorithm is non-trivial,
- Monien 1985: $O(k!n^{O(1)})$

# $k$-path problem

## Problem

Input: directed/undirected graph $G$, integer $k$.
Question: Does $G$ contain a path of length $k$?

## A few facts

- NP-complete (why?)
- even $O(f(k)n^{O(1)})$-time algorithm is non-trivial,
- Monien 1985: $O(k!n^{O(1)})$
- Alon, Yuster, Zwick 1994: $O((2e)^k n^{O(1)})$

# $k$-path problem

## Problem

Input: directed/undirected graph $G$, integer $k$.
Question: Does $G$ contain a path of length $k$?

## A few facts

- NP-complete (why?)
- even $O(f(k)n^{O(1)})$-time algorithm is non-trivial,
- Monien 1985: $O(k!n^{O(1)})$
- Alon, Yuster, Zwick 1994: $O((2e)^k n^{O(1)})$
- Kneis, Mölle, Richter, Rossmanith 2006: $O(4^k n^{O(1)})$

# $k$-path problem

## Problem

Input: directed/undirected graph $G$, integer $k$.
Question: Does $G$ contain a path of length $k$?

## A few facts

- NP-complete (why?)
- even $O(f(k)n^{O(1)})$-time algorithm is non-trivial,
- Monien 1985: $O(k!n^{O(1)})$
- Alon, Yuster, Zwick 1994: $O((2e)^k n^{O(1)})$
- Kneis, Mölle, Richter, Rossmanith 2006: $O(4^k n^{O(1)})$
- Koutis 2008: $O(2^{3/2k} n^{O(1)})$

# $k$-path problem

## Problem

Input: directed/undirected graph $G$, integer $k$.
Question: Does $G$ contain a path of length $k$?

## A few facts

- NP-complete (why?)
- even $O(f(k)n^{O(1)})$-time algorithm is non-trivial,
- Monien 1985: $O(k!n^{O(1)})$
- Alon, Yuster, Zwick 1994: $O((2e)^k n^{O(1)})$
- Kneis, Mölle, Richter, Rossmanith 2006: $O(4^k n^{O(1)})$
- Koutis 2008: $O(2^{3/2k} n^{O(1)})$
- Williams 2009: $O(2^k n^{O(1)})$

# $k$-path problem

## Problem

Input: directed/undirected graph $G$, integer $k$.
Question: Does $G$ contain a path of length $k$?

## A few facts

- NP-complete (why?)
- even $O(f(k)n^{O(1)})$-time algorithm is non-trivial,
- Monien 1985: $O(k!n^{O(1)})$
- Alon, Yuster, Zwick 1994: $O((2e)^k n^{O(1)})$
- Kneis, Mölle, Richter, Rossmanith 2006: $O(4^k n^{O(1)})$
- Koutis 2008: $O(2^{3/2k} n^{O(1)})$
- Williams 2009: $O(2^k n^{O(1)})$
- Björklund 2010: $O(1.66^n n^{O(1)})$, undirected Hamiltonian cycle ($k = n$)

# $k$-path problem

## Problem

Input: directed/undirected graph $G$, integer $k$.
Question: Does $G$ contain a path of length $k$?

## A few facts

- NP-complete (why?)
- even $O(f(k)n^{O(1)})$-time algorithm is non-trivial,
- Monien 1985: $O(k!n^{O(1)})$
- Alon, Yuster, Zwick 1994: $O((2e)^k n^{O(1)})$
- Kneis, Mölle, Richter, Rossmanith 2006: $O(4^k n^{O(1)})$
- Koutis 2008: $O(2^{3/2k} n^{O(1)})$
- Williams 2009: $O(2^k n^{O(1)})$
- Björklund 2010: $O(1.66^n n^{O(1)})$, undirected Hamiltonian cycle ($k = n$)
- Björklund, Husfeldt, Kaski, Koivisto 2010: $O(1.66^k n^{O(1)})$, undirected

# $k$-path in $O^*(2^k)$-time

# Notation

$$[k] = \{1, \ldots, k\}$$

# $O^*(2^k)$-time algorithm for $k$-path

## Rough idea

- Want to construct a polynomial $P$, $P \not\equiv 0$ iff $G$ has a $k$-path.

# $O^*(2^k)$-time algorithm for $k$-path

## Rough idea

- Want to construct a polynomial $P$, $P \not\equiv 0$ iff $G$ has a $k$-path.
- First try: $P(\cdots) = \sum_{k\text{-path } R \text{ in } G} \text{monomial}(R).$

  Seems good, but how to evaluate it?

# $O^*(2^k)$-time algorithm for $k$-path

## Rough idea

- Want to construct a polynomial $P$, $P \not\equiv 0$ iff $G$ has a $k$-path.
- First try: $P(\cdots) = \sum_{k\text{-path } R \text{ in } G} \text{monomial}(R)$.

  Seems good, but how to evaluate it?
- Second try: $P(\cdots) = \sum_{k\text{-walk } W \text{ in } G} \text{monomial}(W)$.

  Now we **can** evaluate it but we may get false positives.

# $O^*(2^k)$-time algorithm for $k$-path

## Rough idea

- Want to construct a polynomial $P$, $P \not\equiv 0$ iff $G$ has a $k$-path.
- First try: $P(\cdots) = \sum_{k\text{-path } R \text{ in } G} \text{monomial}(R)$.

  Seems good, but how to evaluate it?
- Second try: $P(\cdots) = \sum_{k\text{-walk } W \text{ in } G} \text{monomial}(W)$.

  Now we **can** evaluate it but we may get false positives.
- Final try: $P(\cdots) = \sum_{k\text{-walk } W \text{ in } G} \sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is bijective}}} \text{monomial}(w, \ell)$.
  - We still can evaluate it,
  - It turns out that every monomial corresponding to a walk which is not a path appears an even number of times so it cancels-out!

# Our Hero

$$P(\mathbf{x}, \mathbf{y}) = \sum_{\text{walk } W = v_1, \ldots, v_k} \sum_{\substack{\ell : [k] \to [k] \\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i, v_{i+1}} \prod_{i=1}^{k} y_{v_i, \ell(i)}}_{\text{mon}_{W,\ell}}$$

Variables:

- a variable $x_e$ for every $e \in E$,
- a variable $y_{v,\ell}$ for every $v \in V$ and $\ell \in [k]$.

# Monomials corresponding to non-path walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a bijection $\ell \in S_k$.

# Monomials corresponding to non-path walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.

# Monomials corresponding to non-path walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : [k] \to [k]$ as follows:

$$\ell'(x) = \begin{cases} \ell(b) & \text{if } x = a, \\ \ell(a) & \text{if } x = b, \\ \ell(x) & \text{otherwise.} \end{cases}$$

# Monomials corresponding to non-path walks cancel-out

- Let $W = v_1, \dots, v_k$ be a walk, and a bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : [k] \to [k]$ as follows:

$$\ell'(x) = \begin{cases} \ell(b) & \text{if } x = a, \\ \ell(a) & \text{if } x = b, \\ \ell(x) & \text{otherwise.} \end{cases}$$

- $(W, \ell) \neq (W, \ell')$ since $\ell$ is injective.

# Monomials corresponding to non-path walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : [k] \to [k]$ as follows:

$$\ell'(x) = \begin{cases} \ell(b) & \text{if } x = a, \\ \ell(a) & \text{if } x = b, \\ \ell(x) & \text{otherwise.} \end{cases}$$

- $(W, \ell) \neq (W, \ell')$ since $\ell$ is injective.
- $\mathrm{mon}_{W,\ell} = \prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{v_i,\ell(i)} =$

$$\prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i\in[k]\setminus\{a,b\}} y_{v_i,\ell(i)} \underbrace{y_{v_a,\ell(a)}}_{y_{v_b \ell'(b)}} \underbrace{y_{v_b,\ell(b)}}_{y_{v_a \ell'(a)}} = \mathrm{mon}_{W,\ell'}$$

# Monomials corresponding to non-path walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : [k] \to [k]$ as follows:

$$\ell'(x) = \begin{cases} \ell(b) & \text{if } x = a, \\ \ell(a) & \text{if } x = b, \\ \ell(x) & \text{otherwise.} \end{cases}$$

- $(W, \ell) \neq (W, \ell')$ since $\ell$ is injective.
- $\text{mon}_{W,\ell} = \text{mon}_{W,\ell'}$

# Monomials corresponding to non-path walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : [k] \to [k]$ as follows:

$$\ell'(x) = \begin{cases} \ell(b) & \text{if } x = a, \\ \ell(a) & \text{if } x = b, \\ \ell(x) & \text{otherwise.} \end{cases}$$

- $(W, \ell) \neq (W, \ell')$ since $\ell$ is injective.
- $\text{mon}_{W,\ell} = \text{mon}_{W,\ell'}$
- If we start from $(W, \ell')$ and follow the same way of assignment we get $(W, \ell)$ back. (This is called <u>a fixed-point free involution</u>)

# Monomials corresponding to non-path walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : [k] \to [k]$ as follows:

$$\ell'(x) = \begin{cases} \ell(b) & \text{if } x = a, \\ \ell(a) & \text{if } x = b, \\ \ell(x) & \text{otherwise.} \end{cases}$$

- $(W, \ell) \neq (W, \ell')$ since $\ell$ is injective.
- $\mathrm{mon}_{W,\ell} = \mathrm{mon}_{W,\ell'}$
- If we start from $(W, \ell')$ and follow the same way of assignment we get $(W, \ell)$ back. (This is called a fixed-point free involution)
- Since the field is of characteristic 2, $\mathrm{mon}_{W,\ell}$ and $\mathrm{mon}_{W,\ell'}$ cancel out!

# Half the way...

**Corollary**

If $P \not\equiv 0$ then there is a $k$-path.

# The second half

Recall:

$$P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W = v_1,\ldots,v_k} \sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{v_i,\ell(i)}}_{\text{mon}_{W,\ell}}$$

## Question

Why do we need exactly $\text{mon}_{W,\ell} = \prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{v_i,\ell(i)}$?
What if, say, $\text{mon}_{W,\ell} = \prod_{i=1}^{k} y_{v_i,\ell(i)}$?

# The second half

Recall:

$$P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W = v_1,\ldots,v_k} \sum_{\substack{\ell:[k]\to[k]\\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{v_i,\ell(i)}}_{\text{mon}_{W,\ell}}$$

## Question

Why do we need exactly $\text{mon}_{W,\ell} = \prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{v_i,\ell(i)}$?
What if, say, $\text{mon}_{W,\ell} = \prod_{i=1}^{k} y_{v_i,\ell(i)}$?

## Answer

Now, every labelled walk which is a path gets a **unique** monomial.

# The second half

Recall:

$$P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W = v_1,\ldots,v_k} \sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{v_i,\ell(i)}}_{\mathrm{mon}_{W,\ell}}$$

## Question

Why do we need exactly $\mathrm{mon}_{W,\ell} = \prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{v_i,\ell(i)}$?
What if, say, $\mathrm{mon}_{W,\ell} = \prod_{i=1}^{k} y_{v_i,\ell(i)}$?

## Answer

Now, every labelled walk which is a path gets a **unique** monomial.

## Corollary

If there is a $k$-path in $G$ then $P \not\equiv 0$.

# Where are we?

**Corollary**

There is a $k$-path in $G$ iff $P \not\equiv 0$.

**The missing element**

How to evaluate $P$ efficiently?
($O^*(2^k)$ is efficiently enough.)

# Weighted inclusion-exclusion

Let $A_1, \ldots, A_n \subseteq U$, where $U$ is a finite set.
Let $w : U \to F$ be a weight function.
For any $X \subseteq U$ denote $w(X) = \sum_{x \in X} w(x)$.
Let us also denote $\bigcap_{i \in \emptyset}(U - A_i) = U$.

Then,

$$w\left(\bigcap_{i \in [n]} A_i\right) = \sum_{X \subseteq [n]} (-1)^{|X|} w\left(\bigcap_{i \in X}(U - A_i)\right).$$

# Weighted inclusion-exclusion

Let $A_1, \ldots, A_n \subseteq U$, where $U$ is a finite set.
Let $w : U \to F$ be a weight function.
For any $X \subseteq U$ denote $w(X) = \sum_{x \in X} w(x)$.
Let us also denote $\bigcap_{i \in \emptyset}(U - A_i) = U$.

Then,

$$w\left(\bigcap_{i \in [n]} A_i\right) = \sum_{X \subseteq [n]} (-1)^{|X|} w\left(\bigcap_{i \in X}(U - A_i)\right).$$

Counting over a field of characteristic 2 we know that $-1 = 1$ so we can remove the $(-1)^{|X|}$:

$$w\left(\bigcap_{i \in [n]} A_i\right) = \sum_{X \subseteq [n]} w\left(\bigcap_{i \in X}(U - A_i)\right).$$

# Evaluating $P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W} \sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is bijective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$

Fix a walk $W$.

- $U = \{\ell : [k] \to [k]\}$ (all functions)
- for $\ell \in U$, define the weight $w(\ell) = \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$.
- for $i = 1,\ldots,k$ let $A_i = \{\ell \in U \ : \ \ell^{-1}(i) \neq \emptyset\}$.
- Then,

$$\sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is bijective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) = \sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is surjective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) = w(\bigcap_{i=1}^{k} A_i).$$

- By weighted I-E,

$$\sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is surjective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) = \sum_{X\subseteq[k]} w\left(\bigcap_{i\in X}(U - A_i)\right) =$$

$$\sum_{X\subseteq[k]} \sum_{\ell:[k]\to[k]\setminus X} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$$

# Evaluating $P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W} \sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is bijective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$

Fix a walk $W$.

- $U = \{\ell : [k] \to [k]\}$ (all functions)
- for $\ell \in U$, define the weight $w(\ell) = \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$.
- for $i = 1,\ldots,k$ let $A_i = \{\ell \in U \;:\; \ell^{-1}(i) \neq \emptyset\}$.
- Then,

$$\sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is bijective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) = \sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is surjective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) = w(\bigcap_{i=1}^{k} A_i).$$

- By weighted I-E,

$$\sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is surjective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) = \sum_{X\subseteq[k]} w\left(\bigcap_{i\in X}(U - A_i)\right) =$$

$$\sum_{X\subseteq[k]} \sum_{\ell:[k]\to X} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$$

# Evaluating $P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W} \sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is bijective}}} \mathrm{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$

We got

$$\sum_{\substack{\ell:[k]\to[k] \\ \ell \text{ is bijective}}} \mathrm{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) = \sum_{X\subseteq[k]} \sum_{\ell:[k]\to X} \mathrm{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$$

Hence,

$$\begin{aligned} P(\mathbf{x},\mathbf{y}) &= \sum_{\text{walk } W} \sum_{X\subseteq[k]} \sum_{\ell:[k]\to X} \mathrm{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) \\ &= \sum_{X\subseteq[k]} \underbrace{\sum_{\text{walk } W} \sum_{\ell:[k]\to X} \mathrm{mon}_{W,\ell}(\mathbf{x},\mathbf{y})}_{P_X(\mathbf{x},\mathbf{y})} \end{aligned}$$

# Evaluating $P_X(\mathbf{x},\mathbf{y}) = \sum_{\substack{\text{walk } W \\ \text{of length } k}} \sum_{\ell:[k]\to X} \mathrm{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$ in $n^{O(1)}$

We use dynamic programming. (How?)

# Evaluating $P_X(\mathbf{x},\mathbf{y}) = \sum_{\substack{\text{walk } W \\ \text{of length } k}} \sum_{\ell:[k]\to X} \mathrm{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$ in $n^{O(1)}$

We use dynamic programming. (How?)
Fill the 2-dimensional table $T$,

$$T[v,d] = \sum_{\substack{\text{walk } W = v_1,\ldots,v_d \\ v_1 = v}} \sum_{\ell:[k]\to X} \prod_{i=1}^{d-1} x_{v_i,v_{i+1}} \prod_{i=1}^{d} y_{v_i,\ell(i)}$$

Then,

$$T[v,d] = \begin{cases} \sum_{l\in X} y_{vl} & \text{when } d = 1, \\ \sum_{l\in X} y_{vl} \sum_{(v,w)\in E} x_{vw} \cdot T[w,d-1] & \text{otherwise.} \end{cases}$$

Hence, $P_X(\mathbf{x},\mathbf{y}) = \sum_{s\in V} T[s,k]$ can be computed in $O(k|E|)$ time.

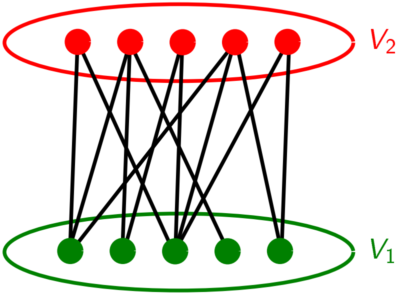
# Conclusion

## Corollary

The $k$-path problem can be solved by a $O^*(2^k)$-time polynomial space one-sided error Monte-Carlo algorithm.

# $k$-path in undirected bipartite graphs in $O^*(2^{k/2})$ time

# $k$-path in undirected bipartite graphs in $O^*(2^{k/2})$ time



# A new hero

**Idea**

Label vertices of $V_1$ only.

$$P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W = v_1,\ldots,v_k} \sum_{\substack{\ell:[k/2]\to[k/2] \\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k/2} y_{v_{2i-1},\ell(i)}}_{\text{mon}_{W,\ell}}$$

Variables:

- a variable $x_e$ for every $e \in E$ $(x_u v = x_v u)$,
- a variable $y_{v,\ell}$ for every $v \in V$ and $\ell \in [k/2]$.

# Checking the hero

$$P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W = v_1,\ldots,v_k} \sum_{\substack{\ell:[k/2]\to[k/2] \\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k/2} y_{v_{2i-1},\ell(i)}}_{\mathrm{mon}_{W,\ell}}$$

## Paths do not cancel-out

If there is a $k$-path with an endpoint in $V_1$ then $P \neq 0$.
(Proof: We can recover $(W,\ell)$ from $\mathrm{mon}_{W,\ell}$ as before.)

# Checking the hero

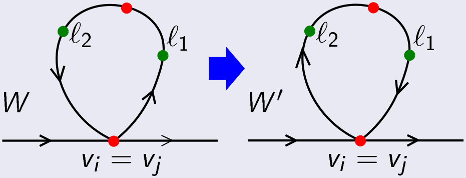
$$P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W = v_1,\ldots,v_k} \sum_{\substack{\ell:[k/2]\to[k/2] \\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k/2} y_{v_{2i-1},\ell(i)}}_{\mathrm{mon}_{W,\ell}}$$

## Do non-path walks cancel-out?

Consider a non-path labelled walk $(W,\ell)$, $W = v_1,\ldots,v_k$.

Case 1 If exist $i,j$, $i<j$ s.t. $v_i = v_j$, $v_i \in V_1$:
pick lexicographically first such pair;
both $v_i$ and $v_j$ have labels so we **swap labels** as before.

Case 2 As in Case 1, but $v_i \in V_2$ and Case 1 does not occur:
**reverse the cycle**:




- $\mathrm{mon}_{W,\ell} = \mathrm{mon}_{W',\ell'}$,
- from $(W',\ell')$ we get $(W,\ell)$,
- Does $(W,\ell) \neq (W',\ell')$ ?

# Checking the hero

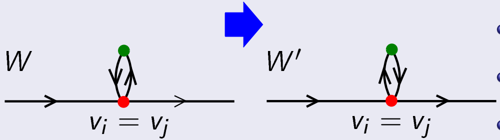
$$P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W = v_1,\ldots,v_k} \; \sum_{\substack{\ell:[k/2]\to[k/2] \\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k/2} y_{v_{2i-1},\ell(i)}}_{\text{mon}_{W,\ell}}$$

## Do non-path walks cancel-out?

Consider a non-path labelled walk $(W,\ell)$, $W = v_1,\ldots,v_k$.

**Case 1** If exist $i,j$, $i<j$ s.t. $v_i = v_j$, $v_i \in V_1$:
pick lexicographically first such pair;
both $v_i$ and $v_j$ have labels so we **swap labels** as before.

**Case 2** As in Case 1, but $v_i \in V_2$ and Case 1 does not occur:
**reverse the cycle**:

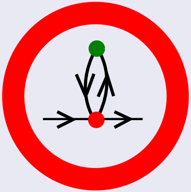



- $\text{mon}_{W,\ell} = \text{mon}_{W',\ell'}$,
- from $(W',\ell')$ we get $(W,\ell)$,
- Does $(W,\ell) \neq (W',\ell')$ ? NO!

# Fixing the hero

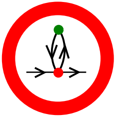
## Admissible walks

Walk $v_1, \ldots, v_k$ is admissible if:
For every $i = 1, \ldots, k-2$, if $v_i \in V_2$ and $v_{i+1} \in V_1$ then $v_{i+2} \neq v_i$.



$$P(\mathbf{x}, \mathbf{y}) = \sum_{\substack{\text{walk } W = v_1, \ldots, v_k \\ W \text{ is admissible}}} \sum_{\substack{\ell : [k/2] \to [k/2] \\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i, v_{i+1}} \prod_{i=1}^{k/2} y_{v_{2i-1}, \ell(i)}}_{\text{mon}_{W,\ell}}$$

# Checking the fixed hero



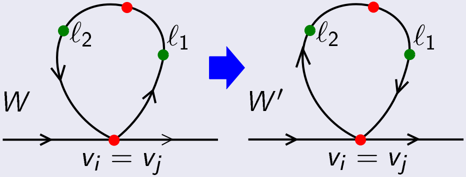
$$P(\mathbf{x},\mathbf{y}) = \sum_{\substack{\text{walk } W = v_1,\ldots,v_k\\ W \text{ is admissible}}} \; \sum_{\substack{\ell:[k/2]\to[k/2]\\ \ell \text{ is bijective}}} \underbrace{\prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k/2} y_{v_{2i-1},\ell(i)}}_{\text{mon}_{W,\ell}}$$

## Do non-path walks cancel-out?

Consider a non-path labelled walk $(W,\ell)$, $W = v_1,\ldots,v_k$.

**Case 1** If exist $i,j$, $i<j$ s.t. $v_i = v_j$, $v_i \in V_1$:
pick lexicographically first such pair;
both $v_i$ and $v_j$ have labels so we **swap labels** as before.

**Case 2** As in Case 1, but $v_i \in V_2$ and Case 1 does not occur:
**reverse the cycle**:




- $\text{mon}_{W,\ell} = \text{mon}_{W',\ell'}$,
- from $(W',\ell')$ we get $(W,\ell)$,
- $(W,\ell) \neq (W',\ell')$ because $W$ admissible,
- $W'$ is admissible.

# Evaluating $P(\mathbf{x},\mathbf{y}) = \sum_{\text{admissible walk } W} \sum_{\substack{\ell:[k/2]\to[k/2] \\ \ell \text{ is bijective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$

As before, from inclusion-exclusion principle we can get

$$\sum_{\substack{\ell:[k/2]\to[k/2] \\ \ell \text{ is bijective}}} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) = \sum_{X\subseteq[k/2]} \sum_{\ell:[k/2]\to X} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y})$$

Hence, as before:

$$\begin{aligned} P(\mathbf{x},\mathbf{y}) &= \sum_{\text{admissible walk } W} \sum_{X\subseteq[k/2]} \sum_{\ell:[k/2]\to X} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y}) \\ &= \sum_{X\subseteq[k/2]} \underbrace{\sum_{\text{admissible walk } W} \sum_{\ell:[k/2]\to X} \text{mon}_{W,\ell}(\mathbf{x},\mathbf{y})}_{P_X(\mathbf{x},\mathbf{y})} \end{aligned}$$

**Note:** Only $2^{k/2}$ polynomials $P_X$ to evaluate.

# Evaluating $P_X(\mathbf{x},\mathbf{y}) = \sum_{\substack{\text{admissible} \\ \text{walk } W \\ \text{of length } k}} \sum_{\ell:[k/2]\to X} \text{mon}_{W,\ell}$ in poly-time

Dynamic programming:

$$T[v,w,d] = \sum_{\substack{\text{admissible walk} \\ W = v_1,\ldots,v_d \\ v_1 = v \\ v_2 = w}} \sum_{\ell:[k/2]\to X} \prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k/2} y_{v_{2i-1},\ell(i)}$$

Then,

$$T[v,w,d] = \begin{cases} x_{vw} \sum_{l\in X} y_{vl} & \text{when } d = 2 \text{ and } v \in V_1, \\ x_{vw} \sum_{l\in X} y_{wl} & \text{when } d = 2 \text{ and } v \in V_2, \\ \sum_{l\in X} y_{vl} \sum_{(w,u)\in E} x_{vw} \cdot T[w,u,d-1] & \text{when } d > 2 \text{ and } v \in V_1, \\ \sum_{\substack{(w,u)\in E \\ u\neq v}} x_{vw} \cdot T[w,u,d-1] & \text{when } d > 2 \text{ and } v \in V_2. \end{cases}$$

# Conclusion

**Theorem (Björklund, Husfeldt, Kaski, Koivisto 2010)**

The $k$-path problem in undirected bipartite graphs can be solved in $O^*(2^{k/2}) = O^*(1.42^k)$ time and polynomial space.

# Conclusion

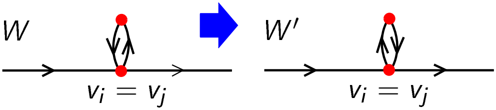
**Theorem (Björklund, Husfeldt, Kaski, Koivisto 2010)**

The $k$-path problem in undirected bipartite graphs can be solved in $O^*(2^{k/2}) = O^*(1.42^k)$ time and polynomial space.

# $k$-path in undirected graphs in $O^*(2^{\frac{3}{4}k})$ time

# $k$-path in undirected graphs in $O^*(2^{\frac{3}{4}k})$ time

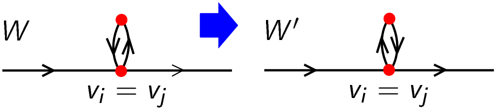
- Choose a **random bipartition** $V = V_1 \cup V_2$, $||V_1| - |V_2|| \leq 1$.
  ($V_1$ and $V_2$ need not be independent now.)
- Where does the bipartite case algorithm fail?



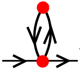

Then $(W, \ell) = (W', \ell')$.

# $k$-path in undirected graphs in $O^*(2^{\frac{3}{4}k})$ time

- Choose a **random bipartition** $V = V_1 \cup V_2$, $||V_1| - |V_2|| \leq 1$. ($V_1$ and $V_2$ need not be independent now.)
- Where does the bipartite case algorithm fail?

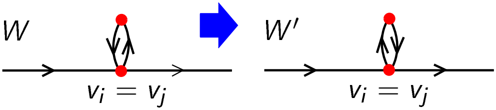



Then $(W, \ell) = (W', \ell')$.

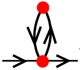
- What if we forbid also ?

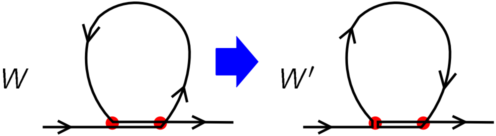
# $k$-path in undirected graphs in $O^*(2^{\frac{3}{4}k})$ time

- Choose a **random bipartition** $V = V_1 \cup V_2$, $||V_1| - |V_2|| \leq 1$. ($V_1$ and $V_2$ need not be independent now.)
- Where does the bipartite case algorithm fail?




Then $(W, \ell) = (W', \ell')$.

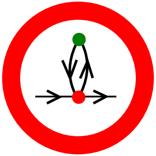
- What if we forbid also  ?
- Then we run into another trouble:



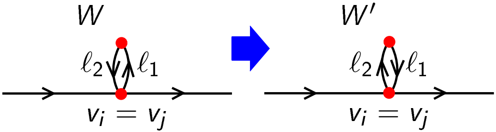

$W'$ contains the forbidden configuration.

# The solution

- Forbidden configuration as before:



- Add more labels:
  label each $V_2 V_2$-edge:




Now $\ell' \neq \ell$.

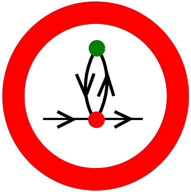
# How many labels do we need now?

- a different label for each $i = 1, \ldots, k$ s.t. $v_i \in V_1$
- a different label for each $i = 1, \ldots, k$ s.t. $v_i v_{i+1} \in V_2$

# $L$-admissible walks

Walk $W = v_1, \ldots, v_k$ is $L$-admissible when

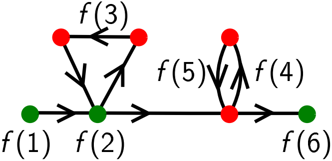
- For every $i = 1, \ldots, k-2$, if $v_i \in V_2$ and $v_{i+1} \in V_1$ then $v_{i+2} \neq v_i$.



- $|\{i \ : \ v_i \in V_1\}| + |\{i \ : \ v_i v_{i+1} \in V_2\}| = L$

$$P_L(\mathbf{x},\mathbf{y}) = \sum_{\substack{\text{walk } W = v_1,\ldots,v_k \\ \color{red}{W \text{ is } L\text{-admissible}}}} \sum_{\substack{\ell:[L]\to[L] \\ \ell \text{ is bijective}}} \prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{L} y_{f(i),\ell(i)},$$

where $f(i) = i$-th labeled object ($V_1$-vertex or $V_2V_2$-edge) in walk $W$.




$$P = \sum_{L=k/2}^{\color{red}{\lceil \frac{3}{4}k \rceil}} P_L$$

# Correctness

- We have checked that:
  $P \not\equiv 0 \Rightarrow$ exists $k$-path
  (i.e. non-path walks cancel-out)

# Correctness

- We have checked that:
  $P \not\equiv 0 \Rightarrow$ exists $k$-path
  (i.e. non-path walks cancel-out)
- The opposite implication not always true! (why?)

# Correctness

- We have checked that:
  $P \not\equiv 0 \Rightarrow$ exists $k$-path
  (i.e. non-path walks cancel-out)
- The opposite implication not always true! (why?)
  it may happen that the only (say) solution $P$ is not $L$-admissible for all $L \leq \lceil \frac{3}{4} k \rceil$.

# Correctness

- We have checked that:
  $P \not\equiv 0 \Rightarrow$ exists $k$-path
  (i.e. non-path walks cancel-out)
- The opposite implication not always true! (why?)
  it may happen that the only (say) solution $P$ is not $L$-admissible for all $L \leq \lceil \frac{3}{4}k \rceil$.
- But...
  - $\mathbb{E}[|\{i \ : \ v_i \in V_1\}| + |\{i \ : \ v_i v_{i+1} \in V_2\}|] = \frac{k}{2} + \frac{k-1}{4} = \frac{3k-1}{4}$

# Correctness

- We have checked that:
  $P \not\equiv 0 \Rightarrow$ exists $k$-path
  (i.e. non-path walks cancel-out)
- The opposite implication not always true! (why?)
  it may happen that the only (say) solution $P$ is not $L$-admissible for all $L \leq \lceil \frac{3}{4}k \rceil$.
- But...
  - $\mathbb{E}[|\{i \ : \ v_i \in V_1\}| + |\{i \ : \ v_i v_{i+1} \in V_2\}|] = \frac{k}{2} + \frac{k-1}{4} = \frac{3k-1}{4}$
- So, by Markov inequality

$$Pr[P \text{ is not } L\text{-admissible for all } L \leq \lceil \tfrac{3}{4}k \rceil] \leq \frac{(3k-1)/4}{\lceil \frac{3}{4}k \rceil + 1} = 1 - 1/O(k)$$

# Correctness

- We have checked that:
  $P \not\equiv 0 \Rightarrow$ exists $k$-path
  (i.e. non-path walks cancel-out)
- The opposite implication not always true! (why?)
  it may happen that the only (say) solution $P$ is not $L$-admissible for all $L \leq \lceil \frac{3}{4}k \rceil$.
- But...
  - $\mathbb{E}[|\{i \;:\; v_i \in V_1\}| + |\{i \;:\; v_i v_{i+1} \in V_2\}|] = \frac{k}{2} + \frac{k-1}{4} = \frac{3k-1}{4}$
- So, by Markov inequality

$$Pr[P \text{ is not } L\text{-admissible for all } L \leq \lceil \tfrac{3}{4}k \rceil] \leq \frac{(3k-1)/4}{\lceil \frac{3}{4}k \rceil + 1} = 1 - 1/O(k)$$

- If we repeat the algorithm $k \log n$ times this probability drops to

$$(1 - 1/O(k))^{k \log n} = (e^{-1/O(k)})^{k \log n} = e^{-O(\log n)} = 1/n^{\Omega(1)}$$

# Conclusion

**Theorem (Björklund, Husfeldt, Kaski, Koivisto 2010)**

The $k$-path problem in undirected graphs can be solved in $O^*(2^{3k/4}) = O^*(1.682^k)$ time and polynomial space.

# Conclusion

**Theorem (Björklund, Husfeldt, Kaski, Koivisto 2010)**

The $k$-path problem in undirected graphs can be solved in $O^*(2^{3k/4}) = O^*(1.682^k)$ time and polynomial space.

Exercises: tune the algorithm to get $O^*(1.66^k)$.

**Corollary (Björklund 2009)**

The Hamiltonian Cycle problem in undirected graphs can be solved in $O^*(1.66^k)$ time and polynomial space.