Probably Optimal Graph Motifs

Andreas Björklund[1], Petteri Kaski[2] and Łukasz Kowalik[3] (speaker)

[1] Lund University (Sweden)
[2] Aalto University (Finland)
[3] University of Warsaw (Poland)

STACS, Kiel, 28.02.2013

GRAPH MOTIF problem

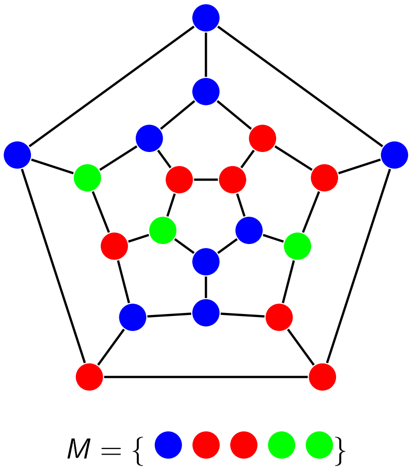
Input:

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M.

Question:

Is there a subset $S \subseteq V$ such that

- $G[S]$ is connected,
- $c(S)$ matches M?



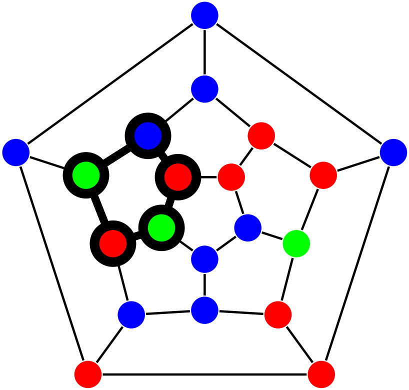
Input:

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M.

Question:

Is there a subset $S \subseteq V$ such that

- $G[S]$ is connected,
- $c(S)$ matches M?



$M = \{$  $\}$

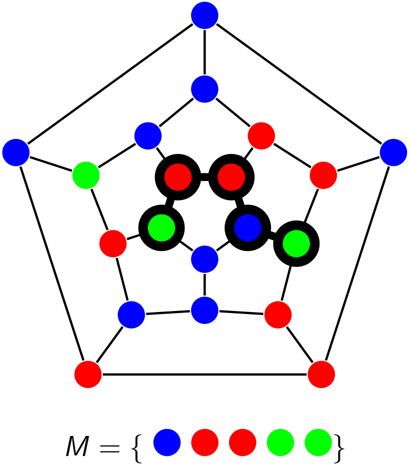
Input:

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M.

Question:

Is there a subset $S \subseteq V$ such that

- $G[S]$ is connected,
- $c(S)$ matches M?



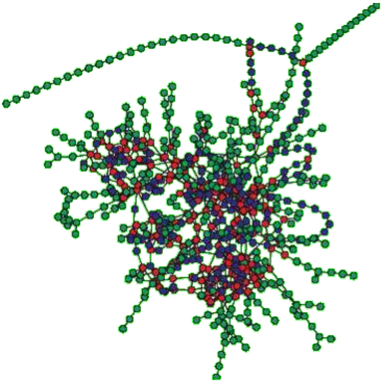
Graph Motif: motivation, complexity

- Introduced in 2006 by Lacroix et al. as a model of functional motif search in metabolic networks

- NP-complete
(even when G is a tree and M is a set)

- In bioinformatics applications
$|V| < 10\,000$, $|M| < 20$.

- So, maybe FPT?



FPT algorithms for Graph Motif

Let k be the size (number of vertices) of the solution (here: $k = |M|$). Denote $O^*(f(k)) = O(f(k)\text{poly}(n))$.

Previous results

- $O^*(87^k)$ [Fellows, Fertin, Hermelin and Vialette 2007]
- $O^*(4.32^k)$ [Betzler, Fellows, Komusiewicz and Niedermeier 2008]
- $O^*(4^k)$ [Guillemot and Sikora 2010]
- $O^*(2.54^k)$ [Koutis 2012]

Our result (to be continued...)

- An $O(2^k mk)$-time algorithm for Graph Motif,
- An $O^*((2-\epsilon)^k)$-time algorithm for Graph Motif gives a $O((2-\epsilon')^n)$-time algorithm for Set Cover

Note: All the algorithms above are randomized Monte-Carlo.

Graph Motif, optimization versions

What if there is no motif in the graph?

Is there something **close** to the motif?

There are three optimization versions
(introduced by Dondi, Fertin, Vialette CPM'09, CPM'11):

- Max Motif,
- Min-Add,
- Min-Substitute

Max Motif problem

Input

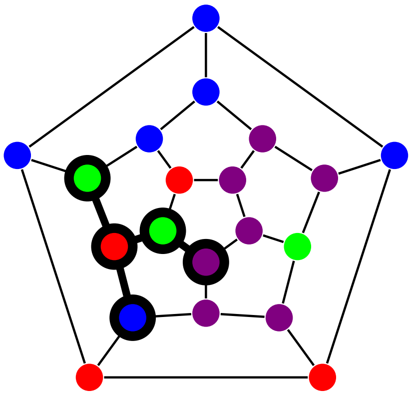
- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M

Optimization Problem

Find the largest subset $S \subseteq V$ s.t.

- $G[S]$ is connected,
- $c(S) \subseteq M$.

(**Remove** as few elements from M as possible to get a YES-instance.)



$M = \{$ ● ● ● ● ● ● $\}$

Max Motif problem

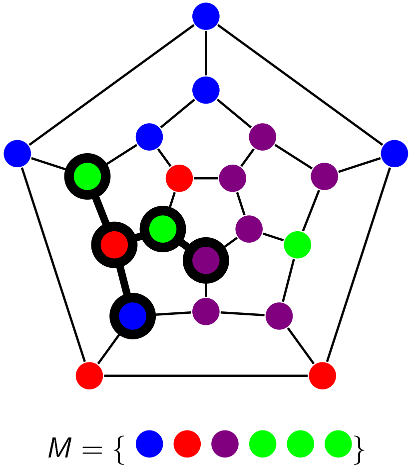
Input

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M
- $k \in \mathbb{N}$.

Decision Version

Is there a subset $S \subseteq V$ s.t.

- $|S| = k$,
- $G[S]$ is connected,
- $c(S) \subseteq M$?



Input

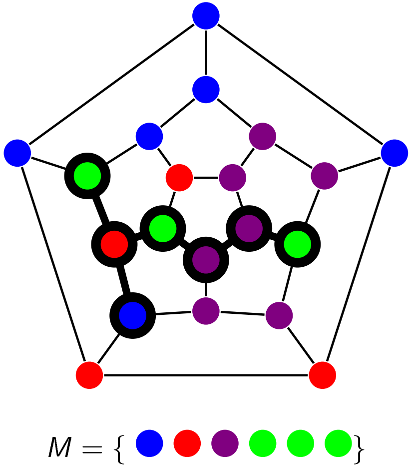
- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M

Optimization Problem

Find the smallest subset $S \subseteq V$ s.t.

- $G[S]$ is connected,
- $c(S) \supseteq M$.

(**Add** as few elements to M as possible to get a YES-instance.)



MIN-ADD problem

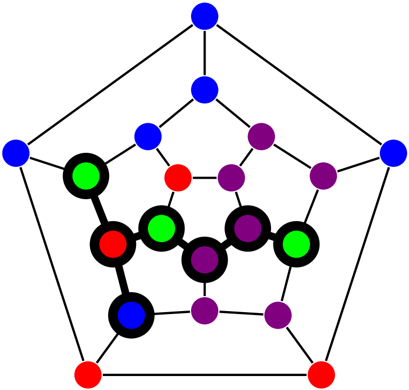
Input

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M
- $k \in \mathbb{N}$.

Decision Version

Is there a subset $S \subseteq V$ s.t.

- $|S| = k$,
- $G[S]$ is connected,
- $c(S) \supseteq M$?



$M = \{$  $\}$

MIN-SUBSTITUTE problem

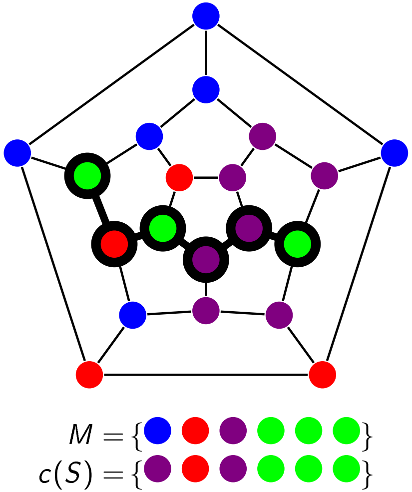
Input

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M

Optimization Problem

Find a subset $S \subseteq V$ s.t.

- $G[S]$ is connected,
- $c(S)$ can be obtained from M by a minimum number of substitutions.



MIN-SUBSTITUTE problem

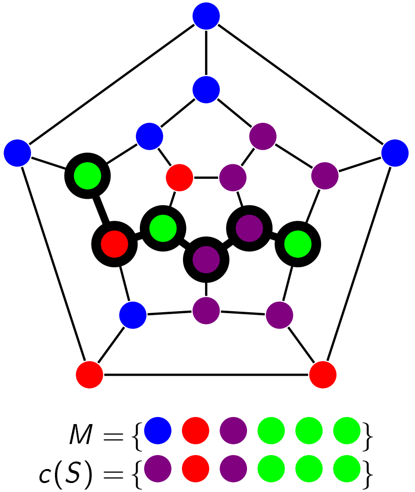
Input

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M
- $d \in \mathbb{N}$.

Decision Version

Is there a subset $S \subseteq V$ s.t.

- $G[S]$ is connected,
- $c(S)$ can be obtained from M by at most d substitutions?



Graph Motif, optimization versions

Previous best results for optimization versions (all by Koutis 2012):

- MAX MOTIF = MIN-DELETE $O^*(2.54^k)$
- MIN-ADD $O^*(2.54^k)$
- MIN-SUBSTITUTE $O^*(5.08^k)$

Closest Motif

1. We introduce a new variant, Closest Motif: minimize the **edit distance** between M and $c(S)$,
2. Closest Motif encompasses all the three optimization versions,
3. We show a $O^*(2^k)$-time algorithm for Closest Motif.

Sketch of our approach. A toy problem

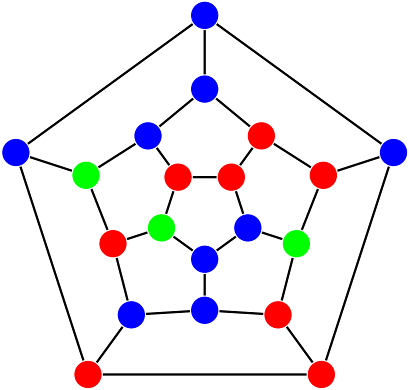
Input:

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M.

Question:

Is there a **path** $P \subseteq G$ such that

- $c(V(P))$ matches M?



$M = \{$ ● ● ● ● ● $\}$

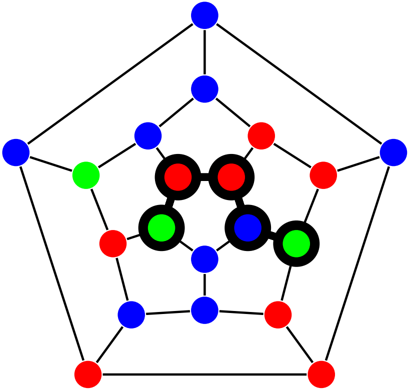
Input:

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M.

Question:

Is there a **path** $P \subseteq G$ such that

- $c(V(P))$ matches M?



$M = \{$  $\}$

Approach: testing whether a polynomial is nonzero

The plan

We construct a multivariate polynomial P over $GF(2^\beta)$ such that:

- $P \not\equiv 0$ iff YES-instance.
- We can evaluate P at a given point (vector) fast.

Schwartz-Zippel Lemma

- Polynomials over fields have few zeroes.
- So, we can test whether a polynomial $P(x_1, \ldots, x_n)$ is nonzero w.h.p. by evaluating it in a random vector $(x_1, \ldots, x_n)$.

The plan continued

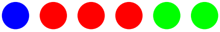
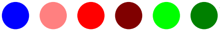
So, we will get a randomized Monte-Carlo one-sided error algorithm running in time of evaluating P.

Shades and consistent labelling

Shades

- Set of colors: $C = c(V)$
- Let $m : C \to \mathbb{N}$ be the multiplicity function of M.
- For $c \in C$ let $D(c) = \{(c, i) \;:\; i = 1, \ldots m(c)\}$ be the set of **shades** of color c
- Let $D = \bigcup_c D(c)$.

Example:

$M = \{$  $\}$, $D = \{$  $\}$

Consistent labellings

- Let $W = v_1, \ldots, v_k$ be a walk.
- Labelling $\ell : \{1, \ldots, k\} \to D$ is **consistent** if for every $i = 1, \ldots, k$ we have $\ell(i) \in D(c(v_i))$.

Our Hero

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

Monomials corresponding to non-simple walks cancel-out

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

Monomials corresponding to non-simple walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a consistent 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-simple walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a consistent bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : \{1, \ldots, k\} \to \{1, \ldots, k\}$ 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-simple walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a consistent bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : \{1, \ldots, k\} \to \{1, \ldots, k\}$ as follows:

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

- ℓ' is bijective and consistent.

Monomials corresponding to non-simple walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a consistent bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : \{1, \ldots, k\} \to \{1, \ldots, k\}$ as follows:

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

- ℓ' is bijective and consistent.
- $(W, \ell) \neq (W, \ell')$ since ℓ is injective.

Monomials corresponding to non-simple walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a consistent bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : \{1, \ldots, k\} \to \{1, \ldots, k\}$ as follows:

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

- ℓ' is bijective and consistent.
- $(W, \ell) \neq (W, \ell')$ since ℓ is injective.
- $\text{mon}(W, \ell) = \prod_{i=1}^{k-1} x_{v_i, v_{i+1}} \prod_{i=1}^{k} y_{v_i, \ell(i)} =$
$\prod_{i=1}^{k-1} x_{v_i, v_{i+1}} \Big(\prod_{i \in \{1,\ldots,k\} \setminus \{a,b\}} \underbrace{y_{v_i, \ell(i)}}_{y_{v_i, \ell'(i)}} \Big) \underbrace{y_{v_a, \ell(a)}}_{y_{v_b \ell'(b)}} \underbrace{y_{v_b, \ell(b)}}_{y_{v_a \ell'(a)}} = \text{mon}(W, \ell')$

Monomials corresponding to non-simple walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a consistent bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : \{1, \ldots, k\} \to \{1, \ldots, k\}$ as follows:

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

- ℓ' is bijective and consistent.
- $(W, \ell) \neq (W, \ell')$ since ℓ is injective.
- $\mathrm{mon}(W, \ell) = \mathrm{mon}(W, \ell')$

Monomials corresponding to non-simple walks cancel-out

- Let $W = v_1, \dots, v_k$ be a walk, and a consistent bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : \{1, \dots, k\} \to \{1, \dots, k\}$ as follows:

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

- ℓ' is bijective and consistent.
- $(W, \ell) \neq (W, \ell')$ since ℓ is injective.
- $\text{mon}(W, \ell) = \text{mon}(W, \ell')$
- If we start from (W, ℓ') and follow the same way of assignment we get (W, ℓ) back.

Monomials corresponding to non-simple walks cancel-out

- Let $W = v_1, \ldots, v_k$ be a walk, and a consistent bijection $\ell \in S_k$.
- Assume $v_a = v_b$ for some $a < b$, if many such pairs take the lexicographically first.
- We define $\ell' : \{1, \ldots, k\} \to \{1, \ldots, k\}$ as follows:

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

- ℓ' is bijective and consistent.
- $(W, \ell) \neq (W, \ell')$ since ℓ is injective.
- $\text{mon}(W, \ell) = \text{mon}(W, \ell')$
- If we start from (W, ℓ') and follow the same way of assignment we get (W, ℓ) back.
- Since the field is of characteristic 2, $\text{mon}(W, \ell)$ and $\text{mon}(W, \ell')$ cancel out!

$P \not\equiv 0$ iff a YES-instance

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

We have proved that

If $P \not\equiv 0$ then we have a YES-instance.

Observation

- Every labelled walk which is a path gets a **unique** monomial.
- So, monomials of simple paths **do not cancel-out**.
- So, if we have a YES-instance then $P \not\equiv 0$.

Corollary

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

Evaluating $P(\mathbf{x},\mathbf{y}) = \sum_{\text{walk } W} \sum_{\substack{\ell:\{1,\dots,k\}\to D \\ \ell \text{ is bijective} \\ \ell \text{ is consistent}}} \prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{v_i,\ell(i)}$

- By the Inclusion-Exclusion Principle one can show that

$$P(\mathbf{x},\mathbf{y}) = \sum_{X\subseteq\{1,\dots,k\}} \underbrace{\sum_{\text{walk } W} \sum_{\substack{\ell:\{1,\dots,k\}\to X \\ \ell \text{ is consistent}}} \prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{v_i,\ell(i)}}_{P_X(\mathbf{x},\mathbf{y})}$$

- By dynamic programming P_X can be evaluated in polynomial time.

Toy problem solved

Corollary

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

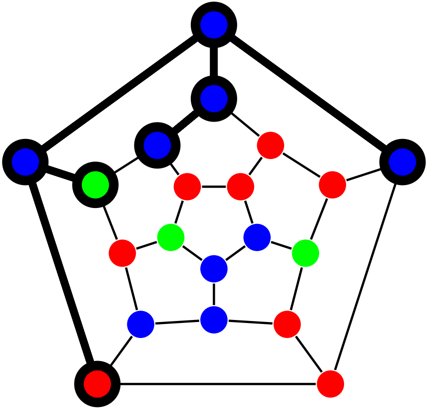
Input:

- Graph $G = (V, E)$,
- (not necesarily proper) coloring $c : V \to \mathbb{N}$,
- a multiset of colors M.

Question:

Is there a **tree** $T \subseteq G$ such that

- $c(V(T))$ matches M?



From Toy Problem (paths) to Graph Motif (trees)

It suffices to replace walks by "tree-like walks". They are called branching walks.

Details skipped.

Maximum Graph Motif Hero (path version)

$$P(\mathbf{x},\mathbf{y}) = \sum_{W = x_1,\ldots,x_k} \sum_{\substack{s:\{1,\ldots,k\}\to D \\ s \text{ is consistent}}} \sum_{\substack{\ell:\{1,\ldots,k\}\to\{1,\ldots,k\} \\ \ell \text{ is bijective}}} \mathrm{mon}(W,s,\ell)$$

$$\mathrm{mon}(W,s,\ell) = \prod_{i=1}^{k-1} x_{v_i,v_{i+1}} \prod_{i=1}^{k} y_{h(x_i),s(i)} z_{s(i),\ell(i)}$$

... and finally ...

the one everybody is waiting for ...

Closest Motif Hero

$$P(\mathbf{x},\mathbf{y}) = \sum_{W=(T,h)} \sum_{f:V(T)\to\{0,1\}} \sum_{\substack{s:V(T)\to D \\ s \text{ is } f\text{-consisent}}} \sum_{\substack{\ell:V(T)\to\{1,\ldots,k\} \\ \ell \text{ is bijective}}} \mathrm{mon}(W,s,\ell,f)\eta^{\kappa(f,s)}$$

$$\mathrm{mon}(W,s,\ell,f) = \prod_{\substack{uv\in E(T) \\ u=\mathrm{parent}(v)}} x_{h(u),h(v)} \prod_{v\in V(T)} y_{h(v),s(v)} z_{s(v),\ell(v)} \prod_{u\in V(T)} w_{h(u)}^{f(u)}.$$

Don't even try to parse it!