Reconstruction of Trees from Jumbled and Weighted Subtrees

Dénes Bartha, Péter Burcsi, Zsuzsanna Lipták

CPM 2016
Tel Aviv, June 27-29, 2016

Reconstruction problems

Given partial information on object x, is it possible to reconstruct x?

Reconstruction problems

Given partial information on object x, is it possible to reconstruct x?
E.g. if x is a string, can it be reconstructed . . .

- from the set of its subsequences (Simon, 1975)

Reconstruction problems

Given partial information on object x, is it possible to reconstruct x?
E.g. if x is a string, can it be reconstructed . . .

- from the set of its subsequences (Simon, 1975)
- from the multiset of its subsequences (Krasikov & Roditty, 1997; Levenshtein, 2001)

Reconstruction problems

Given partial information on object x, is it possible to reconstruct x?
E.g. if x is a string, can it be reconstructed . . .

- from the set of its subsequences (Simon, 1975)
- from the multiset of its subsequences (Krasikov & Roditty, 1997; Levenshtein, 2001)
- from the set of its substrings (de Luca & Carpi, 1999-2001; Fici et al, 2006)

Reconstruction problems

Given partial information on object x, is it possible to reconstruct x?
E.g. if x is a string, can it be reconstructed ...

- from the set of its subsequences (Simon, 1975)
- from the multiset of its subsequences (Krasikov & Roditty, 1997; Levenshtein, 2001)
- from the set of its substrings (de Luca & Carpi, 1999-2001; Fici et al, 2006)
- from the multiset of its substrings (Piña & Uzcágetui, 2008)

Reconstruction problems

Given partial information on object x, is it possible to reconstruct x?
E.g. if x is a string, can it be reconstructed ...

- from the set of its subsequences (Simon, 1975)
- from the multiset of its subsequences (Krasikov & Roditty, 1997; Levenshtein, 2001)
- from the set of its substrings (de Luca & Carpi, 1999-2001; Fici et al, 2006)
- from the multiset of its substrings (Piña & Uzcágetui, 2008)
- from the set of its k-mers (substrings of length k): SBH (Pevzner, 1989; ...)
- from the set of its RC-subsequences (Cicalese et al., 2012)
- from the multiset of Parikh vectors of its substrings (= jumbled substrings) (Acharya et al., 2010, 2014, 2015)

Reconstruction problems

Different types of questions:

- How much information do we need to have a unique solution x?
- When does a solution x exist?
- When does a unique solution x exist?
- If not unique, how many different solutions exist? (equivalence class sizes)
- Find (efficient?) reconstruction algorithms

Reconstruction from multiset of jumbled substrings

Jumbled substrings

Given string t over constant-size ordered alphabet Σ, with $|\Sigma| = \sigma$.
The Parikh vector counts multiplicity of characters in t. Jumbled substring: only P.v. is known.

Reconstruction from multiset of jumbled substrings

Jumbled substrings

Given string t over constant-size ordered alphabet Σ, with $|\Sigma| = \sigma$.
The Parikh vector counts multiplicity of characters in t. Jumbled substring: only P.v. is known.

Ex.: $\Sigma = \{a, b, c\}$, then these 3 substrings have P.v. $(3, 1, 2)$

b b a c a c c a b a b b a b c c a a a c

Weighted substrings

Weighted substrings

Given string t over constant-size alphabet Σ, with $|\Sigma| = \sigma$, and a weight function $\mu : \Sigma \to \mathbb{N}$.

The weight of t is $\mu(t) = \sum_{i=1}^{|t|} \mu(t_i)$. Weighted substring: only weight is known.

Weighted substrings

Given string t over constant-size alphabet Σ, with $|\Sigma| = \sigma$, and a weight function $\mu : \Sigma \to \mathbb{N}$.

The weight of t is $\mu(t) = \sum_{i=1}^{|t|} \mu(t_i)$. Weighted substring: only weight is known.

Ex.: $\Sigma = \{a, b, c\}$, $\mu(a) = 1, \mu(b) = 2, \mu(c) = 5$. Then these 3 substrings have weight 15, and so does *babcc*.

b b a c a c c a b a b b a b c c a a a c

Polynomial method

- Acharya et al. (2010, 2014, 2015): String reconstruction from multiset of jumbled substrings,
- Bansal, Cieliebak, L. (CPM 2004): pattern matching for weighted substrings

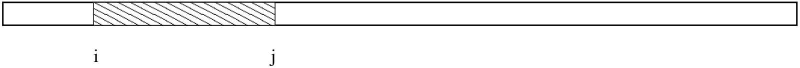
Polynomial method

Polynomial method

- Acharya et al. (2010, 2014, 2015): String reconstruction from multiset of jumbled substrings,
- Bansal, Cieliebak, L. (CPM 2004): pattern matching for weighted substrings

Idea

The weight (or P.v.) of a substring is the difference of that of a prefix and that of another prefix.



Encode these in a polynomial.

Generating polynomials

Let $s = s_1 \dots s_n$, and let $prs_i(s)$, for $i = 0, \dots, n$ be the prefix sums of s. The prefix polynomial of s is

$$p(x) = x^0 + x^{s_1} + x^{s_1+s_2} + \ldots + x^{s_1+\ldots+s_n} = \sum_{i=0}^{n} x^{prs_i(s)}.$$

Then

$$f(x) \quad = \quad p(x)p(\tfrac{1}{x}),$$

is the generating function of the multiset of weighted substrings of s.

Generating polynomials

Example (weighted)

Let $s = aaba$, and $\mu(a) = 1, \mu(b) = 5$.

$$
\begin{aligned}
\text{prefix sums:} \quad & 0, 1, 2, 7, 8 \\
p(x) = \quad & 1 + x + x^2 + x^7 + x^8 \\
f(x) = \quad & (1 + x + x^2 + x^7 + x^8)(1 + x^{-1} + x^{-2} + x^{-7} + x^{-8}) \\
= \quad & x^{-8} + 2x^{-7} + 2x^{-6} + x^{-5} + x^{-2} + 3x^{-1} + \\
& 5 + 3x + x^2 + x^5 + 2x^6 + 2x^7 + x^8.
\end{aligned}
$$

Generating polynomials

Example (weighted)

Let $s = aaba$, and $\mu(a) = 1, \mu(b) = 5$.

$$\begin{aligned}
\text{prefix sums:} \quad & 0, 1, 2, 7, 8 \\
p(x) = \quad & 1 + x + x^2 + x^7 + x^8 \\
f(x) = \quad & (1 + x + x^2 + x^7 + x^8)(1 + x^{-1} + x^{-2} + x^{-7} + x^{-8}) \\
= \quad & x^{-8} + 2x^{-7} + 2x^{-6} + x^{-5} + x^{-2} + 3x^{-1} + \\
& 5 + 3x + x^2 + x^5 + 2x^6 + 2x^7 + x^8.
\end{aligned}$$

positive exponents: substring-weights, coefficients: multiplicities

Generating polynomials

Similar, use multivariate polynomials.

Example (jumbled)

Let $s = aaba$.

$$\begin{aligned}
\text{prefix P.v.s:} \quad & 0, a, 2a, 2a+b, 3a+b \\
p(x) = \; & 1 + x + x^2 + x^2y + x^3y \\
f(x) = \; & (1 + x + x^2 + x^2y + x^3y)(1 + x^{-1} + x^{-2} + x^{-2}y^{-1} + x^{-3}y^{-1}) \\
= \; & x^{-3}y^{-1} + 2x^{-2}y^{-1} + 2x^{-1}y^{-1} + x^{-1} + x^{-2} + 3x^{-1} + \\
& 5 + 3x + x^2 + y + 2xy + 2x^2y + x^3y.
\end{aligned}$$

positive exponents: substring-weights, coefficients: multiplicities

Using generating polynomials

- Multiset of jumbled (or weighted) substrings of s and t are equal iff the generating polynomials are equal
- factorization of polynomials
- pattern matching: p is the P.v. of a substring (m is weight of a substring) iff $c_p \neq 0$ resp. $c_m \neq 0$ (coefficient of x^p resp. x^m)
- c_p resp. c_m gives the multiplicity
- can use FFT for fast multiplication

What we are doing (CPM 2016)

Reconstruction of trees from jumbled or weighted subtrees with some property $\mathcal{A}$. Property $\mathcal{A}$ can be

1. subtree
2. path
3. maximal path (i.e. between leaves)
4. or any other property of subtrees

Note that 1. and 2. are both generalizations of substrings for strings (i.e. the entire tree is a path).

Parikh vectors and jumbled subtrees

- Tree with edge labels from finite ordered alphabet $\Sigma = \{a_1, \ldots, a_\sigma\}$

Parikh vectors and jumbled subtrees

- Tree with edge labels from finite ordered alphabet $\Sigma = \{a_1, \ldots, a_\sigma\}$
- Parikh vector of tree: vector length σ with ith entry = multiplicity of a_i

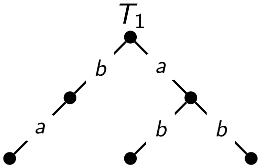
Parikh vectors and jumbled subtrees

- Tree with edge labels from finite ordered alphabet $\Sigma = \{a_1, \ldots, a_\sigma\}$
- Parikh vector of tree: vector length σ with ith entry = multiplicity of a_i
- jumbled subtree: a subtree of which only the Parikh vector is known

Parikh vectors and jumbled subtrees

- Tree with edge labels from finite ordered alphabet $\Sigma = \{a_1, \ldots, a_\sigma\}$
- Parikh vector of tree: vector length σ with ith entry = multiplicity of a_i
- jumbled subtree: a subtree of which only the Parikh vector is known

Example



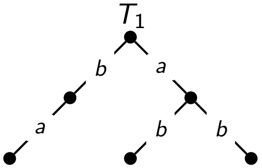
$\Sigma = \{a, b\}$

- Parikh vector of T_1: $(2, 3)$

Parikh vectors and jumbled subtrees

- Tree with edge labels from finite ordered alphabet $\Sigma = \{a_1, \ldots, a_\sigma\}$
- Parikh vector of tree: vector length σ with ith entry = multiplicity of a_i
- jumbled subtree: a subtree of which only the Parikh vector is known

Example



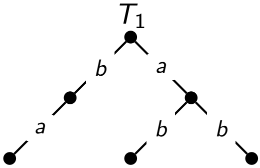
- Parikh vector of T_1: $(2, 3)$
- alternative notation: $a^2 b^3$

$\Sigma = \{a, b\}$

Parikh vectors and jumbled subtrees

- Tree with edge labels from finite ordered alphabet $\Sigma = \{a_1, \ldots, a_\sigma\}$
- Parikh vector of tree: vector length σ with ith entry = multiplicity of a_i
- jumbled subtree: a subtree of which only the Parikh vector is known

Example



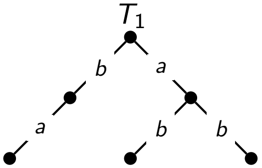
$\Sigma = \{a, b\}$

- Parikh vector of T_1: $(2, 3)$
- alternative notation: $a^2 b^3$
- (size 1) 2 times a, 3 times b: $2a, 3b$

Parikh vectors and jumbled subtrees

- Tree with edge labels from finite ordered alphabet $\Sigma = \{a_1, \ldots, a_\sigma\}$
- Parikh vector of tree: vector length σ with ith entry = multiplicity of a_i
- jumbled subtree: a subtree of which only the Parikh vector is known

Example



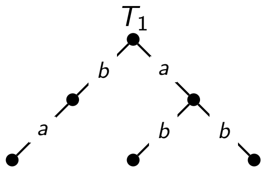
$\Sigma = \{a, b\}$

- Parikh vector of T_1: $(2, 3)$
- alternative notation: $a^2 b^3$
- (size 1) 2 times a, 3 times b: $2a, 3b$
- (size 2) $4ab, 1b^2$

Parikh vectors and jumbled subtrees

- Tree with edge labels from finite ordered alphabet $\Sigma = \{a_1, \ldots, a_\sigma\}$
- Parikh vector of tree: vector length σ with ith entry = multiplicity of a_i
- jumbled subtree: a subtree of which only the Parikh vector is known

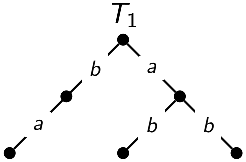
Example



$\Sigma = \{a, b\}$

- Parikh vector of T_1: $(2, 3)$
- alternative notation: a^2b^3
- (size 1) 2 times a, 3 times b: $2a, 3b$
- (size 2) $4ab, 1b^2$
- (size 3) $1a^2b, 3ab^2$
- (size 4) $2a^2b^2, 1ab^3$
- (size 5) $1a^2b^3$

Multi-Parikh-sets of trees



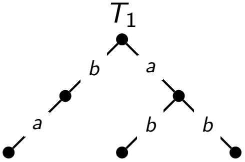
$$MP_{\mathrm{TREE}}(T_1) = \{2a, 3b, 4ab, 1b^2, 1a^2b, 3ab^2, 2a^2b^2, 1ab^3, 1a^2b^3\}$$

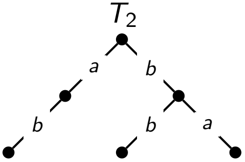
Multi-Parikh-sets of trees

$$MP_{\text{TREE}}(T_1) = \{2a, 3b, 4ab, 1b^2, 1a^2b, 3ab^2, 2a^2b^2, 1ab^3, 1a^2b^3\}$$
$$= MP_{\text{TREE}}(T_2).$$

T_1 and T_2 are MP_{TREE}-equivalent, but non-isomorphic (as edge-labeled trees).

Multi-Parikh-sets of trees

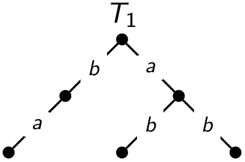


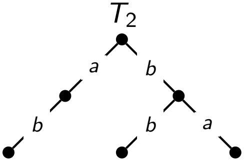


$$MP_{\text{TREE}}(T_1) = \{2a, 3b, 4ab, 1b^2, 1a^2b, 3ab^2, 2a^2b^2, 1ab^3, 1a^2b^3\} = MP_{\text{TREE}}(T_2).$$

T_1 and T_2 are MP_{TREE}-equivalent, but not MP_{PATH}-equivalent:

Multi-Parikh-sets of trees



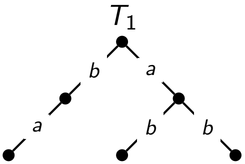


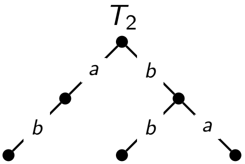
$$MP_{\text{TREE}}(T_1) = \{2a, 3b, 4ab, 1b^2, 1a^2b, 3ab^2, 2a^2b^2, 1ab^3, 1a^2b^3\} \\ = MP_{\text{TREE}}(T_2).$$

T_1 and T_2 are MP_{TREE}-equivalent, but not MP_{PATH}-equivalent:

$$MP_{\text{PATH}}(T_1) = \{2a, 3b, 4ab, 1b^2, 1a^2b, 3ab^2, 2a^2b^2, 1a^2b^3\} \\ \neq \{2a, 3b, 4ab, 1b^2, 1a^2b, 3ab^2, 1a^2b^2, 1ab^3, 1a^2b^3\} = MP_{\text{PATH}}(T_2).$$

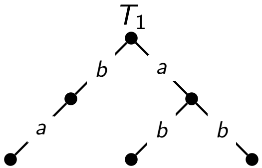
Multi-Parikh-sets of trees





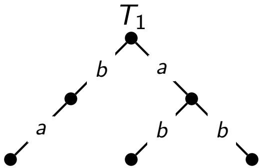
So we can differentiate between T_1 and T_2 via the multi-Parikh-set of paths but not of subtrees.

Multi-Parikh-sets as polynomials



$$MP_{\mathrm{TREE}}(T_1) = \{2a, 3b, 4ab, 1b^2, 1a^2b, 3ab^2, 2a^2b^2, 1ab^3, 1a^2b^3\}$$

Multi-Parikh-sets as polynomials



$$MP_{\text{TREE}}(T_1) = \{2a, 3b, 4ab, 1b^2, 1a^2b, 3ab^2, 2a^2b^2, 1ab^3, 1a^2b^3\}$$

Expressed as a polynomial, where $a \mapsto x$, $b \mapsto y$:

$$f_{\text{TREE}}(T_1) = 6 + 2x + 3y + 4xy + y^2 + x^2y + 3xy^2 + 2x^2y^2 + xy^3 + x^2y^3$$

Polynomials for jumbled subtrees

$$f_{\mathrm{TREE}}(T_1) = 6 + 2x + 3y + 4xy + y^2 + x^2y + 3xy^2 + 2x^2y^2 + xy^3 + x^2y^3$$

In general:

$$f_{\mathrm{TREE}}(T) = \sum_{p=(p_1,\dots,p_\sigma)} c_p \cdot x_1^{p_1} \cdots x_\sigma^{p_\sigma},$$

where c_p is the number of subtrees with Parikh vector $p = (p_1, \dots, p_\sigma)$.

Polynomials for jumbled subtrees

$$f_{\mathrm{TREE}}(T_1) = 6 + 2x + 3y + 4xy + y^2 + x^2y + 3xy^2 + 2x^2y^2 + xy^3 + x^2y^3$$

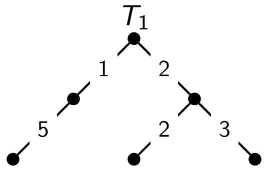
In general:

$$f_{\mathrm{TREE}}(T) = \sum_{p=(p_1,\ldots,p_\sigma)} c_p \cdot x_1^{p_1} \cdots x_\sigma^{p_\sigma},$$

where c_p is the number of subtrees with Parikh vector $p = (p_1, \ldots, p_\sigma)$.

Similar: f_{PATH} ... number of paths ...
f_{MAXPATH} ... number of maximal paths ...

Polynomials for weighted subtrees



$f_{\mathrm{TREE}}(T_1)$
$= 6 + x + 2x^2 + 2x^3 + x^4 + 4x^5 + 2x^6 + x^7 + 2x^8 + x^{10} + x^{11} + x^{13}.$

In general:

$$f_{\mathrm{TREE}}(T) = \sum_m c_m \cdot x^m,$$

where c_m is the number of subtrees with weight m.

Polynomials for jumbled subtrees

Why is using polynomials useful?

- additional algebraic structure (see results for strings)
- efficient decision of equivalence (Schwartz-Zippel lemma)

Efficient decision of equivalence

Schwartz-Zippel lemma (1979,1980)

Let f_1 and f_2 be polynomials in k variables over a field, both of total degree at most d. Let S be a set in the coefficient field. If we evaluate f_1 and f_2 by substituting a uniformly randomly chosen k-tuple $\underline{x} \in S^k$ into them, then the probability of $f_1(\underline{x}) = f_2(\underline{x})$ is at most $d/|S|$.

As a consequence, if we can evaluate f_1 and f_2 in polynomial time, then we have a polynomial time Monte-Carlo algorithm for testing $f_1 = f_2$, by repeatedly substituting random k-tuples and reporting "not equal" if and only at least one substitution fails to evaluate to the same value.

Questions we ask

- **Computation:** How can we compute these polynomials?
- **Reconstruction:** Can T be uniquely reconstructed from the multiset of jumbled subtrees, paths, or maximal paths?

  Two sub-problems:

  1. **Large Unjumble:** Is the unlabeled tree (i.e., its topology) uniquely determined by the multiset of jumbled or weighted subtrees, paths, or maximal paths?
  2. **Small Unjumble:** Given the topology of the tree, is the labeling uniquely determined by the multiset of jumbled or weighted subtrees, paths, or maximal paths?
- **Reconstruction algorithms**

Polynomial for MP_{TREE}

- Root tree T in arbitrary vertex v.
- auxiliary polynomial $r(T, v)$: multiset of all subtrees containing v

Theorem

Let T be a rooted tree with root v. Let $v_1, v_2, \ldots, v_k$ be the children of v. Denote the subtrees rooted at v_i by T_i for $i = 1, \ldots, k$. Denote the index in Σ of the label on the edge connecting v and v_j by l_j. Then

$$r(T, v) = \prod_{j=1}^{k} (1 + x_{l_j} \cdot r(T_j, v_j)) \quad \text{and} \quad f(T) = r(T, v) + \sum_{j=1}^{k} f(T_j)$$

Note that the number of subtrees can be exponential, but evaluation is efficient (for equivalence testing).

Polynomial for MP_{PATH}

- Root tree T in arbitrary vertex v.
- auxiliary polynomial $r(T, v)$: multiset of all paths at least one end is v

Theorem

Let T be a rooted tree with root v. Let $v_1, v_2, \ldots v_k$ be the children of v in T. Denote the subtrees rooted at v_1 (resp. v_2 etc.) by T_1 (resp. T_2 etc.). Denote the index in Σ of the label on the edge connecting v and v_j by l_j. Then

$$r(T, v) = 1 + \sum (x_{l_j} \cdot r(T_j, v_j)),$$

$$f(T) = r(T, v) + \sum_{j=1}^{k} f(T_j) + \sum_{1 \leq i < j \leq k} (x_{l_i} x_{l_j} r(T_i, v_i) r(T_j, v_j))$$

Using generating polynomials for trees

- Recursive computation of polynomials straightforward in all cases
- We do not have all nice properties of string case (not only multiplication of polynomials: also addition)
- Efficient evaluation still possible (using the recursive definition)
- In some cases, can be used for (non-)reconstructibility results

Reconstructibility / Non-reconstructibility

(Case: weighted, MAXPATH, Small Unjumble) n-star: $n-1$ leaves

Theorem

Let T_1 and T_2 be two edge-weighted n stars.

1. If $n-1$ is not a power of 2, then $MW_{\text{MAXPATH}}(T_1) = MW_{\text{MAXPATH}}(T_2)$ implies that T_1 and T_2 are isomorphic as edge weighted trees.
2. If $n-1 = 2^k$ for some $k \leq 0$, then there are non-isomorphic edge weighted n-stars that are MW_{MAXPATH}-equivalent.

Proof

Simple polynomial manipulation, and sets of numbers $\{a_1, \ldots, a_{n-1}\} \neq \{b_1, \ldots, b_{n-1}\}$ s.t. their pairwise sums coincide. Always exist if $n-1$ is a power of 2. (e.g. $\{1, 4\}, \{2, 3\}$; $\{1, 4, 102, 103\}, \{2, 3, 101, 104\}, \ldots$).

Proof: Let $a_1, \ldots, a_{n-1}$ and $b_1, \ldots, b_{n-1}$ distinct labelings s.t. the MW_{MAXPATH} coincide. Then for $f(x) = x^{a_1} + \cdots + x^{a_{n-1}}$ and $g(x) = x^{b_1} + \cdots + x^{b_{n-1}}$

$$f^2(x) - f(x^2) = g^2(x) - g(x^2),$$

since the pairw. sums determine $(f(x))^2 - f(x^2) = \sum_{i,j} x^{a_i + a_j} - \sum_i x^{2a_i}$. So

$$f(x^2) - g(x^2) = f^2(x) - g^2(x) = (f(x) - g(x))(f(x) + g(x))$$

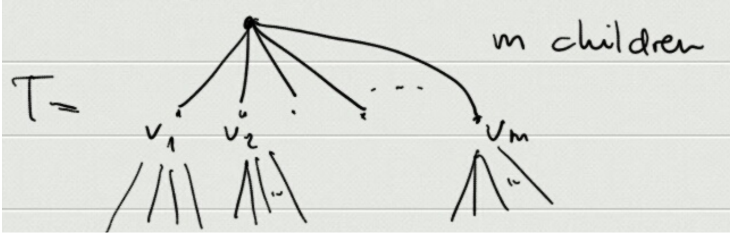
Factor out as many $(x-1)$ factors from $f - g$ as you can: $f(x) - g(x) = (x-1)^k h(x)$. Then

$$\begin{aligned} f(x) + g(x) &= \frac{f^2(x) - g^2(x)}{f(x) - g(x)} = \frac{f(x^2) - g(x^2)}{f(x) - g(x)} = \\ & \frac{(x^2-1)^k h(x^2)}{(x-1)^k h(x)} = (x+1)^k \frac{h(x^2)}{h(x)} \end{aligned}$$

Setting $x = 1$ results in $2(n-1) = 2^k$, so $n-1$ is a power of 2. $\square$

Reconstructibility / Non-reconstructibility

(Case: jumbled/weighted, PATH, Large Unjumble)



v_i has k_i children

Theorem

Let $m \geq 3$. Then there are non-isomorphic trees T_1, T_2 as above s.t. if all edges are labeled with the same character (resp. the same weight), then T_1 and T_2 are MP_{PATH}-equivalent (resp. MW_{PATH}-equivalent).

Proof Idea: The generating polynomial is determined by $\sum_i k_i$ and $\sum_i k_i^2$.
See Prouhet-Tarry-Escott problem:
Distinct sets $\{a_i\}$ and $\{b_i\}$ of card. m s.t. for all $p \leq r : \sum_i a_i^p = \sum_i b_i^p$.

$$f_{\mathrm{PATH}}(T) = r(T, v) + \sum_i f(T_i) + \sum_{i<j} x^2 r(T_i, v_i) r(T_j, v_j)$$

$$r(T, v) = 1 + mx + (\sum_i k_i)x^2 \tag{1}$$

$$f(T_i) = (1 + k_i) + k_i x + \binom{k_i}{2} x^2 \ \text{ and } \ r(T_i, v_i) = (1 + k_i x) \tag{2}$$

$$\sum_{i<j} r(T_i, v_i) r(T_j, v_j) = \binom{m}{2} + x2(m-1)\sum_i k_i + x^2 \sum_{i<j} k_i k_j \tag{3}$$

All coefficients can be computed from $\sum_i k_i$ and $\sum_i k_i^2$:
$\sum_{i<j} k_i k_j = ((\sum_i k_i)^2 - \sum_i k_i^2)$, and $\sum_i \binom{k_i}{2} = \frac{1}{2}(\sum_i k_i^2 - \sum_i k_i)$.

First results

- Generating polynomial method for multisets of P.v.s or subweights can be generalized to trees
- different types of substructures (subtrees, paths etc.)
- not all nice algebraic properties preserved (not only multiplication!)
- polynomial method can be used for uniqueness / non-uniqueness results
- special cases for certain classes of trees
- reconstruction algorithm for some cases

source: hebrewword.org

source: www.israelhebrew.com