An introduction to species theory

Bruno Salvy
@ AriC

1. Introduction

Motivations

Models for data structures

- rational languages;
- context-free grammars;
- beyond.

Enumerative combinatorics

- words;
- trees;
- beyond.

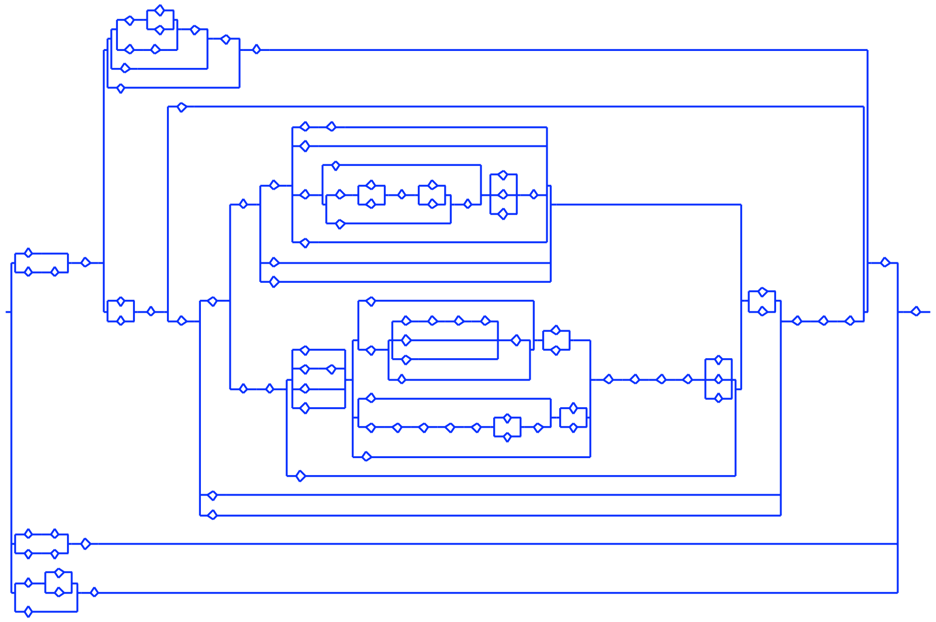
Motivations

Models for data structures

- rational languages;
- context-free grammars;
- beyond.

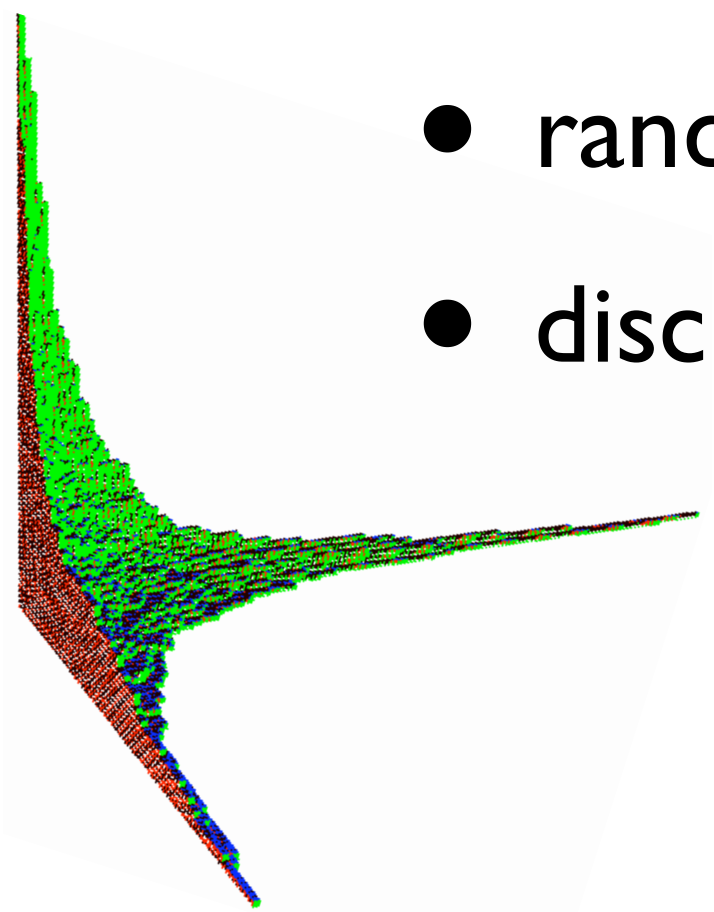
Enumerative combinatorics

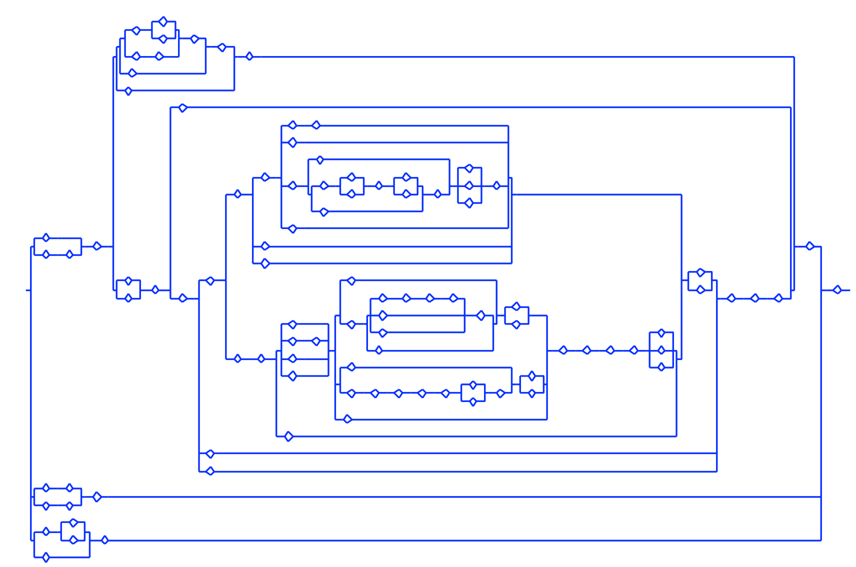
- words;
- trees;
- beyond.



Applications

- Define/handle data-types;
- exhaustive generation of tests;
- random generation of tests;
- discrete simulations.





Recursive Method

Binary trees: $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$

```
DrawBinTree(n) = {
  if n=1 return Z;
  U:=Uniform([0,1]); k:=0; S:=0;
  while (S<U) {k:=k+1;S:=S+bkbn-k-1/bn;}
  return ZxDrawBinTree(k)xDrawBinTree(n-k-1); }
```

b_k: nb binary trees with k nodes (Catalan)

Requires $b_0, b_1, \ldots, b_n$.

[Nijenhuis & Wilf; Flajolet, Zimmermann & Van Cutsem]

Boltzmann Sampling

Principle: Generate each object with a probability depending only on its size.

- The size is not a parameter of the algo.
- Same size $\rightarrow$ same proba. (as before).

[Duchon, Flajolet, Louchard, Schaeffer]

Boltzmann Sampling

Principle: Generate each object with a probability depending only on its size.

- The size is not a parameter of the algo.
- Same size → same proba. (as before).

Proba : $\dfrac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

[Duchon, Flajolet, Louchard, Schaeffer]

Boltzmann Sampling

Principle: Generate each object with a probability depending only on its size.

- The size is not a parameter of the algo.
- Same size → same proba. (as before).

Proba : $\frac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

Def. *Ordinary generating series of* $\mathcal{T}$:

$$T(x) = \sum_{t \in \mathcal{T}} x^{|t|} = \sum_{n \in \mathbb{N}} t_n x^n$$

[Duchon, Flajolet, Louchard, Schaeffer]

Boltzmann: Algorithm

Proba : $\dfrac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

Boltzmann: Algorithm

Proba : $\dfrac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

- Singleton: easy.

Boltzmann: Algorithm

Proba : $\dfrac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

- Singleton: easy.
- Cartesian product:

$\mathcal{H} = \mathcal{F} \times \mathcal{G}$

```
Generate f∈𝓕;
Generate g∈𝓖;
return (f,g);
```

Boltzmann: Algorithm

Proba : $\dfrac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

- Singleton: easy.
- Cartesian product:

$\mathscr{H} = \mathscr{F} \times \mathscr{G}$

```
Generate f∈𝓕;
Generate g∈𝓖;
return (f,g);
```

- Disjoint union

$\mathscr{H} = \mathscr{F} + \mathscr{G}$

Boltzmann: Algorithm

Proba : $\frac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

- Singleton: easy.
- Cartesian product:

$\mathscr{H} = \mathscr{F} \times \mathscr{G}$

```
Generate f∈𝓕;
Generate g∈𝓖;
return (f,g);
```

- Disjoint union

$\mathscr{H} = \mathscr{F} + \mathscr{G}$

```
b=Bernoulli(F(x)/H(x));
if b=1 then Generate f∈𝓕
else Generate g∈𝓖;
```

Boltzmann: Algorithm

Proba : $\dfrac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

- Singleton: easy.
- Cartesian product:

$\mathscr{H} = \mathscr{F} \times \mathscr{G}$

```
Generate f∈𝓕;
Generate g∈𝓖;
return (f,g);
```

- Disjoint union

$\mathscr{H} = \mathscr{F} + \mathscr{G}$

```
b=Bernoulli(F(x)/H(x));
if b=1 then Generate f∈𝓕
else Generate g∈𝓖;
```

Example: binary tree with F(.49)/H(.49)≃.5995 $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$

Coin: 0,1,0,1,0,1,1,0,1,1,1,...

Boltzmann: Algorithm

Proba : $\dfrac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

- Singleton: easy.
- Cartesian product:

$\mathscr{H} = \mathscr{F} \times \mathscr{G}$

```
Generate f∈𝓕;
Generate g∈𝓖;
return (f,g);
```

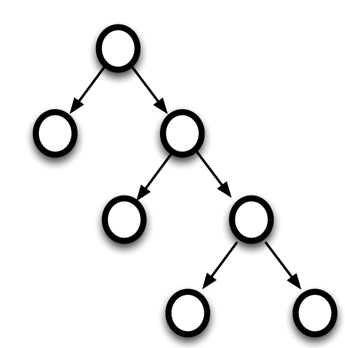
- Disjoint union

$\mathscr{H} = \mathscr{F} + \mathscr{G}$

```
b=Bernoulli(F(x)/H(x));
if b=1 then Generate f∈𝓕
else Generate g∈𝓖;
```

Example: binary tree with F(.49)/H(.49)≃.5995 $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$

Coin: 0,1,0,1,0,1,1,0,1,1,1,...



Boltzmann: Algorithm

Proba : $\frac{x^{|t|}}{T(x)}$ with $T(x) = \sum_{t \in \mathcal{T}} x^{|t|}$

- Singleton: easy.
- Cartesian product:

$\mathscr{H} = \mathscr{F} \times \mathscr{G}$

```
Generate f∈𝓕;
Generate g∈𝓖;
return (f,g);
```

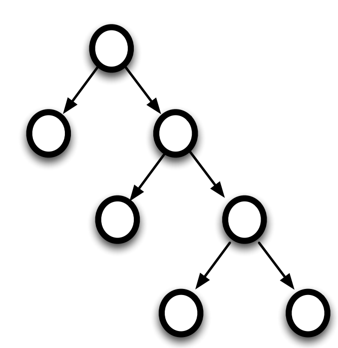
- Disjoint union

$\mathscr{H} = \mathscr{F} + \mathscr{G}$

```
b=Bernoulli(F(x)/H(x));
if b=1 then Generate f∈𝓕
else Generate g∈𝓖;
```

Example: binary tree with F(.49)/H(.49)≃.5995 $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$

Coin: 0,1,0,1,0,1,1,0,1,1,1,...



Complexity linear in |t|.

Generalization

Both methods extend to more general *structured* objects.

They require

- *enumeration sequences* (recursive method);
- *numerical evaluation* of the g.f. (Boltzmann).

Generalization

Both methods extend to more general *structured* objects.

They require

- *enumeration sequences* (recursive method);
- *numerical evaluation* of the g.f. (Boltzmann).

Plan:

2. Species
3. Enumeration
4. Implicit Species
5. Newton Iteration

II. Species

Objects and Arrows

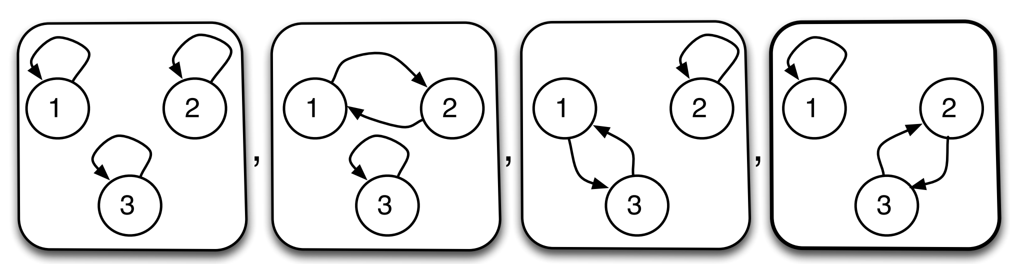
Definition. [Joyal 1981] Une espèce (finitaire) est un endo-foncteur du groupoïde des ensembles finis et bijections.

Differs from species in Bourbaki (Theory of Sets).

Objects and Arrows

Definition. [Joyal 1981] Une espèce (finitaire) est un endo-foncteur du groupoïde des ensembles finis et bijections.

Example: The species of involutions (*s*○*s*=*Id*).

$\mathrm{Inv}[\{1,2,3\}] = \{$ 

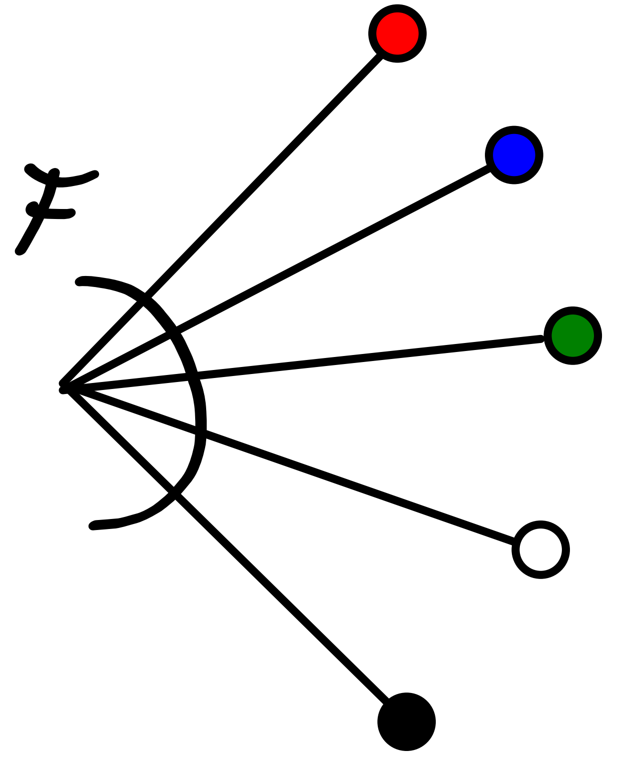
$\}$

$$\mathrm{Inv}[\{a,b,c\}] = \cdots$$

$$\mathrm{Inv}[1 \mapsto a, 2 \mapsto b, 3 \mapsto c] = \cdots\ .$$

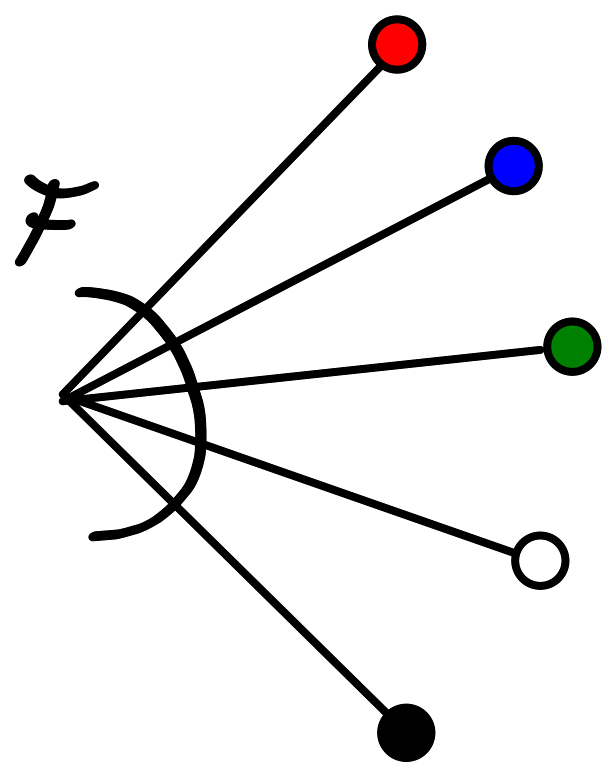
Differs from species in Bourbaki (Theory of Sets).

Examples



Examples

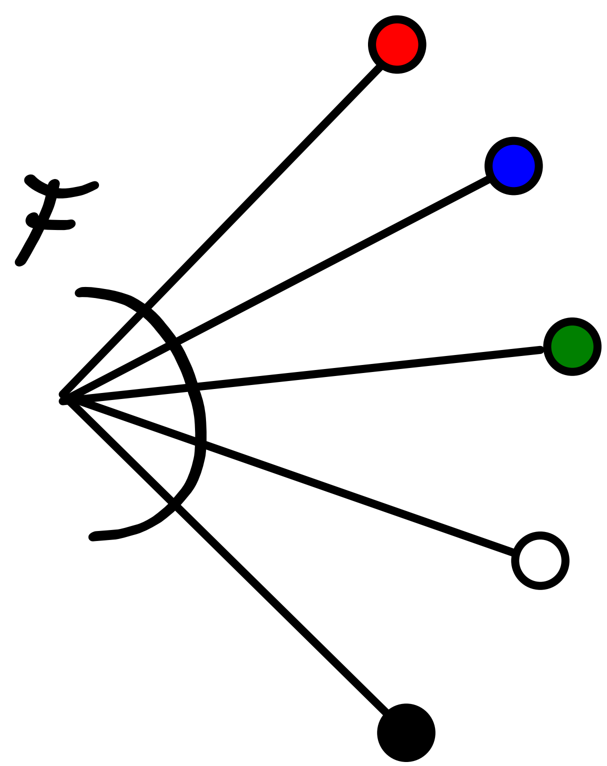
"Constants":



Examples

“Constants”:

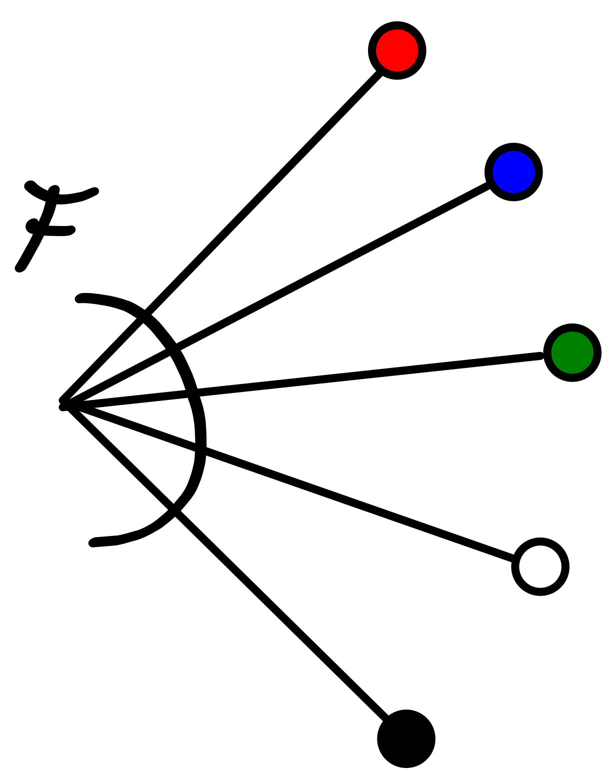
- empty species:
  $0[U]=\varnothing$



Examples

"Constants":

- empty species:
  $0[U]=\varnothing$
- species 1:
  $1[U]=\varnothing$ *if* $U\neq\varnothing$;
  $1[\varnothing]=\{\varnothing\}$

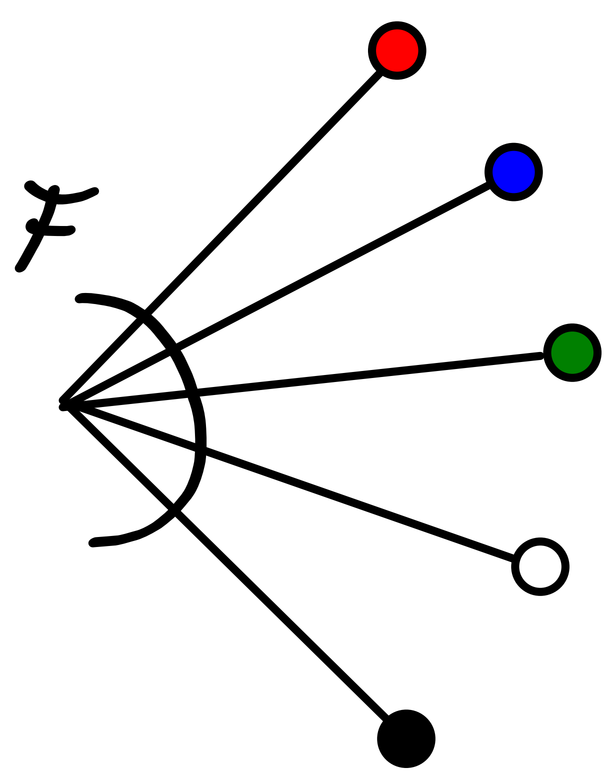


Examples

"Constants":

- empty species:
  $0[U]=\varnothing$

- species 1:
  $1[U]=\varnothing$ *if* $U\neq\varnothing$;
  $1[\varnothing]=\{\varnothing\}$

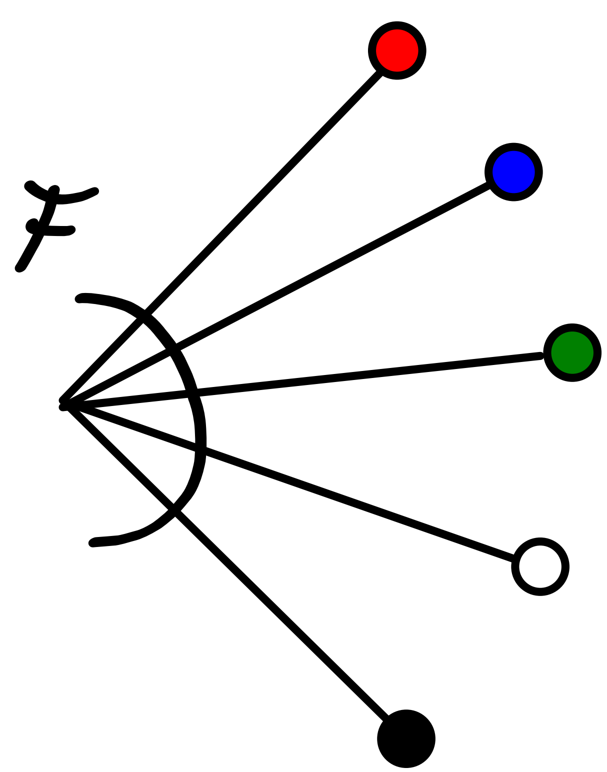


Examples

"Constants":

- empty species:
  $0[U]=\varnothing$
- species 1:
  $1[U]=\varnothing$ *if* $U\neq\varnothing$;
  $1[\varnothing]=\{\varnothing\}$

"Variable" Z:



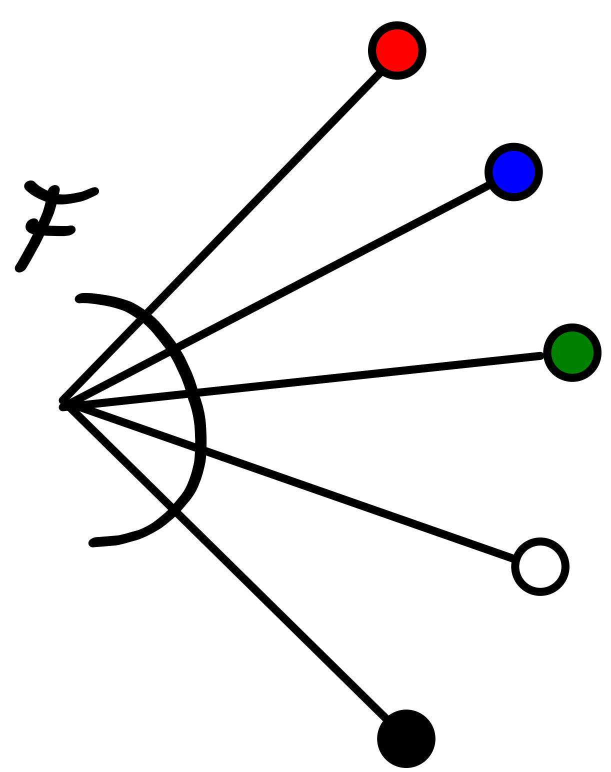
Examples

"Constants":

- empty species:
  $0[U]=\varnothing$
- species 1:
  $1[U]=\varnothing$ *if* $U\neq\varnothing$;
  $1[\varnothing]=\{\varnothing\}$

"Variable" Z:

- $Z[U]=\varnothing$ *if* $|U|\neq 1$;
  $Z[U]=\{U\}$ *otherwise.*



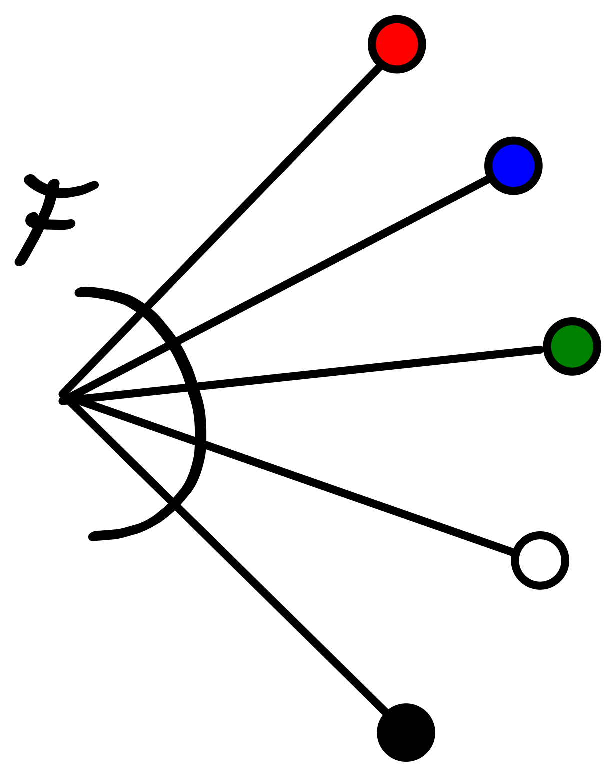
Examples

"Constants":

- empty species:
  $0[U]=\varnothing$
- species 1:
  $1[U]=\varnothing$ *if* $U\neq\varnothing$;
  $1[\varnothing]=\{\varnothing\}$

"Variable" Z:

- $Z[U]=\varnothing$ *if* $|U|\neq 1$;
  $Z[U]=\{U\}$ *otherwise.*



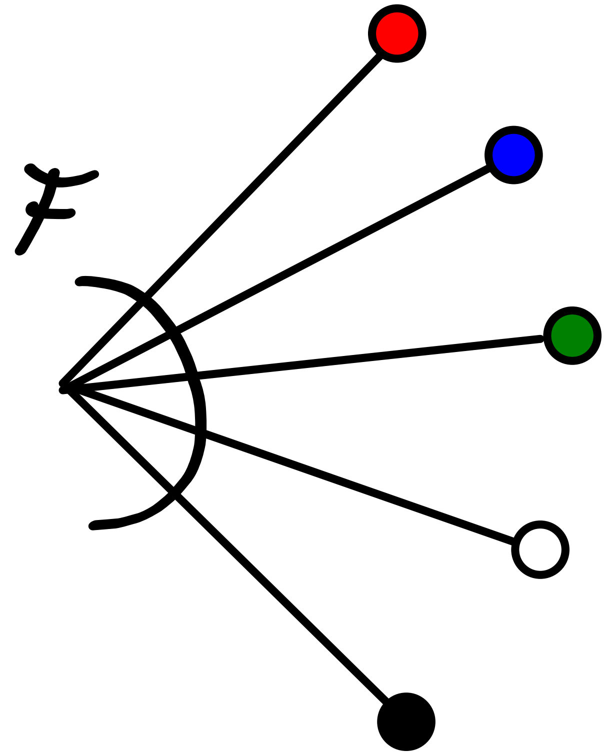
Examples

"Constants":

- empty species:
  $0[U]=\varnothing$
- species 1:
  $1[U]=\varnothing$ *if* $U\neq\varnothing$;
  $1[\varnothing]=\{\varnothing\}$

"Variable" Z:

- $Z[U]=\varnothing$ *if* $|U|\neq 1$;
  $Z[U]=\{U\}$ *otherwise.*



"Functions":

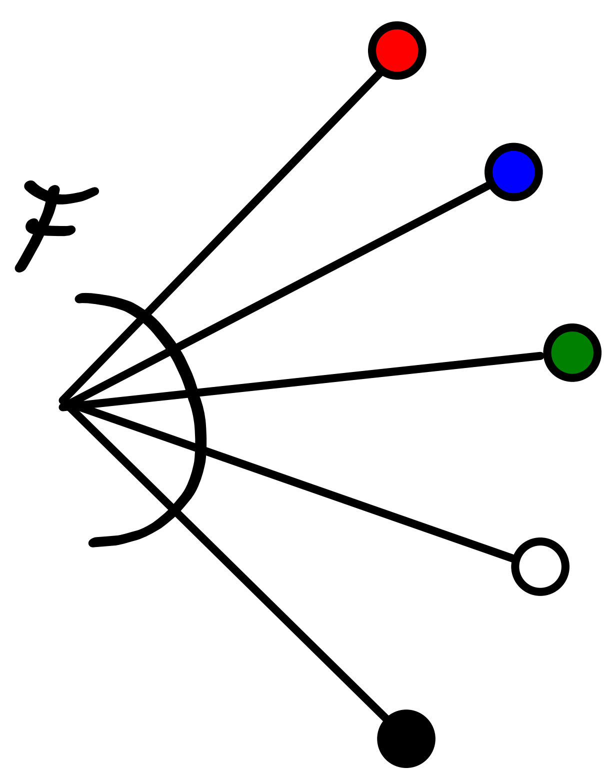
Examples

"Constants":

- empty species: 0[U]=∅
- species 1: 1[U]=∅ *if* U≠∅; 1[∅]={∅}

"Variable" Z:

- Z[U]=∅ *if* |U|≠1; Z[U]={U} *otherwise.*



"Functions":

- SET: SET[U]={U};

Examples

"Constants":

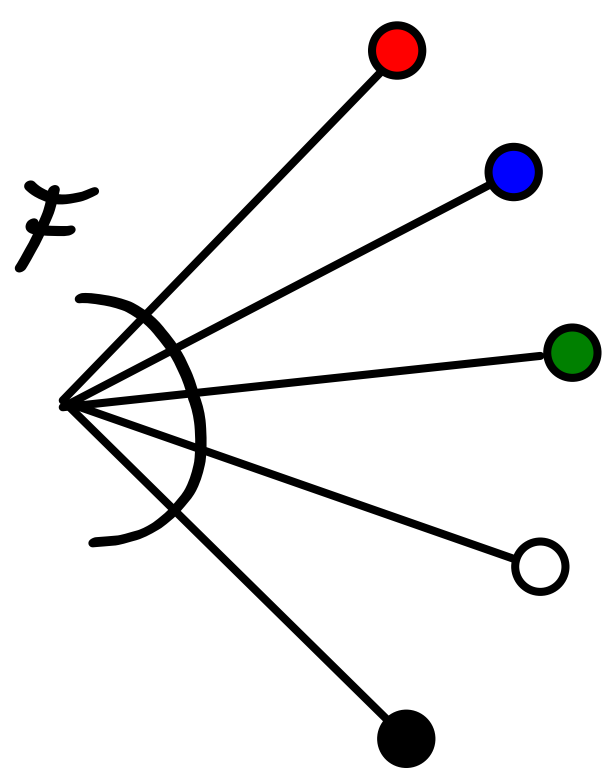
- empty species: $0[U]=\varnothing$
- species 1: $1[U]=\varnothing$ *if* $U\neq\varnothing$; $1[\varnothing]=\{\varnothing\}$

"Variable" Z:

- $Z[U]=\varnothing$ *if* $|U|\neq 1$; $Z[U]=\{U\}$ *otherwise.*

"Functions":

- SET: $\text{SET}[U]=\{U\}$;
- SEQ: $\text{SEQ}[\varnothing]=\{\varnothing\}$; $\text{SEQ}[\{u_1,u_2,\ldots,u_n\}]=\{(u_{\sigma(1)},\ldots,u_{\sigma(n)})\,|\,\sigma\in\mathfrak{S}_n\}$.



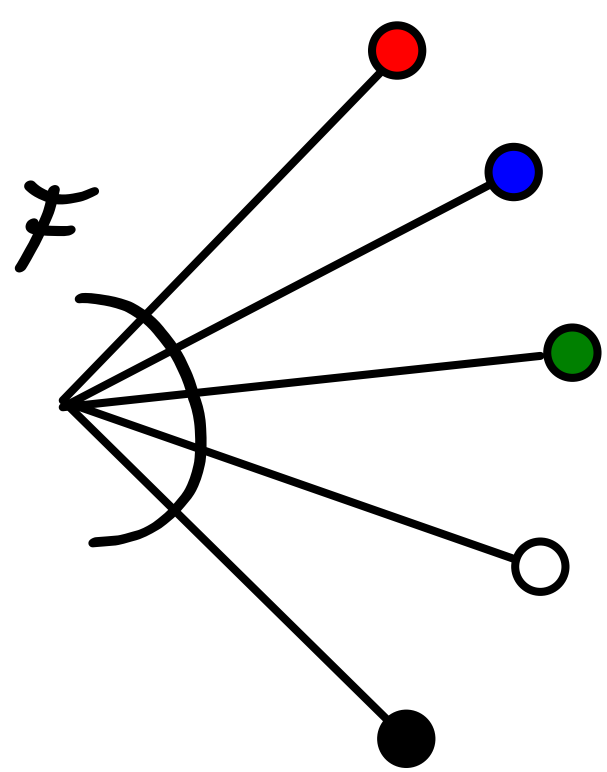
Examples

"Constants":

- empty species: $0[U]=\varnothing$
- species 1: $1[U]=\varnothing$ *if* $U\neq\varnothing$; $1[\varnothing]=\{\varnothing\}$

"Variable" Z:

- $Z[U]=\varnothing$ *if* $|U|\neq 1$; $Z[U]=\{U\}$ *otherwise.*



"Functions":

- SET: $\mathrm{SET}[U]=\{U\}$;
- SEQ: $\mathrm{SEQ}[\varnothing]=\{\varnothing\}$; $\mathrm{SEQ}[\{u_1,u_2,\ldots,u_n\}]=\{(u_{\sigma(1)},\ldots,u_{\sigma(n)})\mid\sigma\in\mathfrak{S}_n\}$.
- CYC: same, with σ ranging over cycles.

Operations

Isomorphism: $\mathcal{F} = \mathcal{G}$ when

$$\mathcal{F}[U] \simeq \mathcal{G}[U] \text{ for all } U,$$

and the expected diagrams commute.

Operations

Isomorphism: $\mathcal{F} = \mathcal{G}$ when

$$\mathcal{F}[U] \simeq \mathcal{G}[U] \text{ for all } U,$$

and the expected diagrams commute.

Sum:

$$(\mathcal{F} + \mathcal{G})[U] := \mathcal{F}[U] + \mathcal{G}[U] \text{ (disjoint union)}$$

ex: $\mathscr{F}+0=0+\mathscr{F}=\mathscr{F}$.

Operations

Isomorphism: $\mathcal{F} = \mathcal{G}$ when

$$\mathcal{F}[U] \simeq \mathcal{G}[U] \text{ for all } U,$$

and the expected diagrams commute.

Sum:

$$(\mathcal{F} + \mathcal{G})[U] := \mathcal{F}[U] + \mathcal{G}[U] \text{ (disjoint union)}$$

ex: $\mathscr{F}+0=0+\mathscr{F}=\mathscr{F}$.

Product:

$$(\mathcal{F} \cdot \mathcal{G})[U] := \sum_{\substack{(U_1,U_2),\\ U=U_1+U_2}} \mathcal{F}[U_1] \times \mathcal{G}[U_2]$$

exs: $\mathscr{F}\cdot 0=0\cdot\mathscr{F}=0$; $\mathscr{F}\cdot 1=1\cdot\mathscr{F}=\mathscr{F}$; $\textsc{Seq}=1+Z\cdot\textsc{Seq}$; $\mathscr{B}=1+Z\cdot\mathscr{B}\cdot\mathscr{B}$.

Operations

Isomorphism: $\mathcal{F} = \mathcal{G}$ when

$$\mathcal{F}[U] \simeq \mathcal{G}[U] \text{ for all } U,$$

and the expected diagrams commute.

Sum:

$$(\mathcal{F} + \mathcal{G})[U] := \mathcal{F}[U] + \mathcal{G}[U] \text{ (disjoint union)}$$

ex: $\mathcal{F}+0=0+\mathcal{F}=\mathcal{F}$.

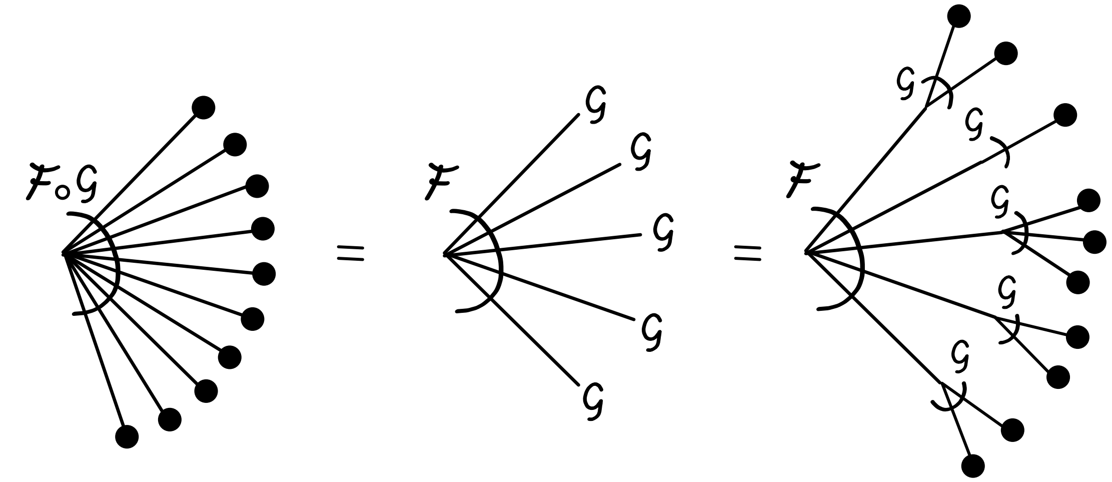
Product:

$$(\mathcal{F} \cdot \mathcal{G})[U] := \sum_{\substack{(U_1,U_2),\\ U=U_1+U_2}} \mathcal{F}[U_1] \times \mathcal{G}[U_2]$$

exs: $\mathcal{F}\cdot 0=0\cdot\mathcal{F}=0$; $\mathcal{F}\cdot 1=1\cdot\mathcal{F}=\mathcal{F}$; $\mathrm{SEQ}=1+Z\cdot\mathrm{SEQ}$; $\mathcal{B}=1+Z\cdot\mathcal{B}\cdot\mathcal{B}$.

All unambiguous context-free languages

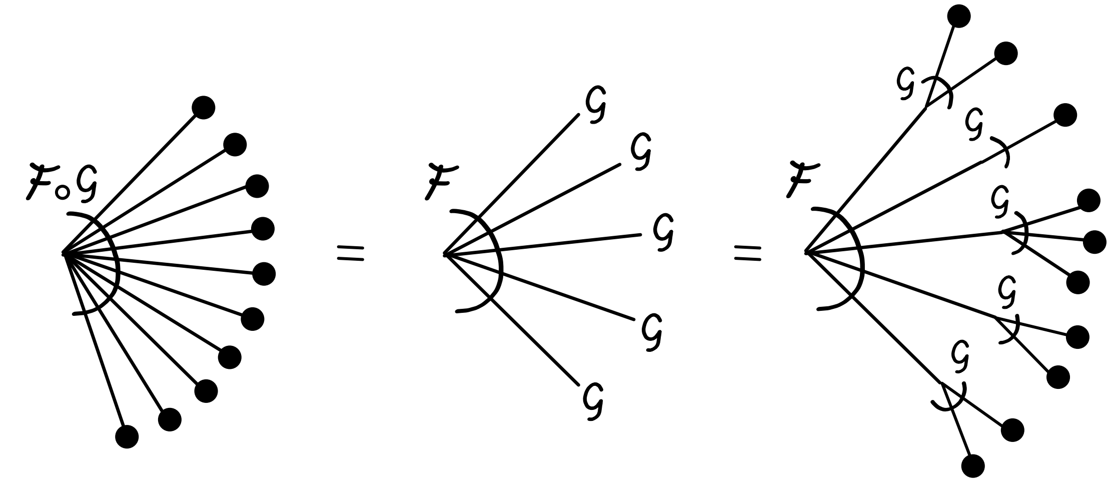
Substitution



Requires $\mathcal{G}[\varnothing]=\varnothing$. Then $\mathcal{F}(\mathcal{G})[\varnothing]=\varnothing$ and

$$(\mathcal{F}(\mathcal{G}))[U] = \sum_{\pi \text{ partition of } U} \mathcal{F}[\pi] \times \prod_{p \in \pi} \mathcal{G}[p], \qquad U \neq \emptyset.$$

Substitution

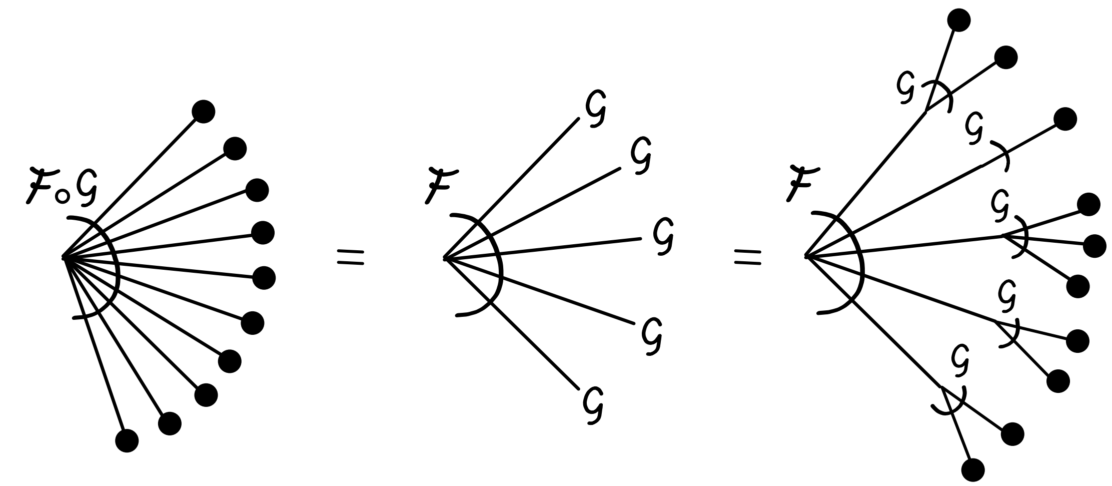


Requires $\mathscr{G}[\varnothing]=\varnothing$. Then $\mathscr{F}(\mathscr{G})[\varnothing]=\varnothing$ and

$$(\mathcal{F}(\mathcal{G}))[U] = \sum_{\pi \text{ partition of } U} \mathcal{F}[\pi] \times \prod_{p\in\pi} \mathcal{G}[p], \qquad U \neq \emptyset.$$

- $\mathscr{F}$(Z)=$\mathscr{F}$;

Substitution

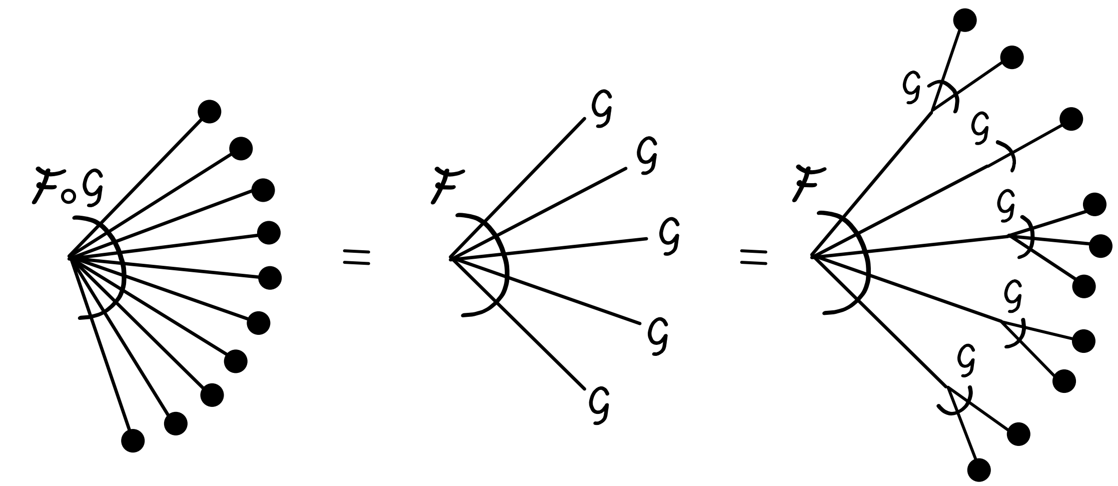


Requires $\mathscr{G}[\varnothing]=\varnothing$. Then $\mathscr{F}(\mathscr{G})[\varnothing]=\varnothing$ and

$$(\mathcal{F}(\mathcal{G}))[U] = \sum_{\pi \text{ partition of } U} \mathcal{F}[\pi] \times \prod_{p\in\pi} \mathcal{G}[p], \qquad U \neq \emptyset.$$

- $\mathscr{F}$(Z)=$\mathscr{F}$;
- Permutations: Perm=SET(CYC);

Substitution

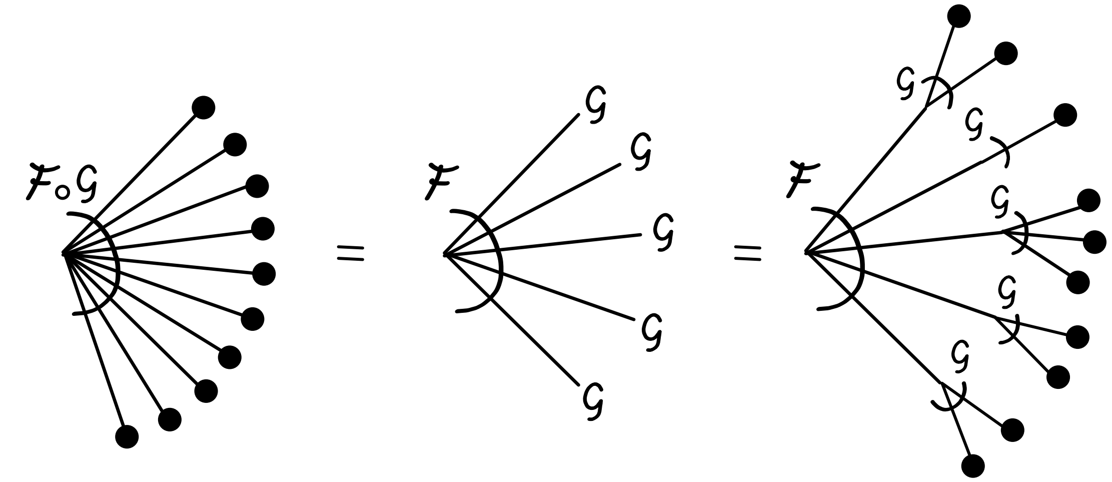


Requires $\mathscr{G}[\varnothing]=\varnothing$. Then $\mathscr{F}(\mathscr{G})[\varnothing]=\varnothing$ and

$$(\mathcal{F}(\mathcal{G}))[U] = \sum_{\pi \text{ partition of } U} \mathcal{F}[\pi] \times \prod_{p\in\pi} \mathcal{G}[p], \qquad U \neq \emptyset.$$

- $\mathscr{F}(Z)=\mathscr{F}$;
- Permutations: Perm=SET(CYC);
- Graphs=SET(Connected_Graphs);

Substitution

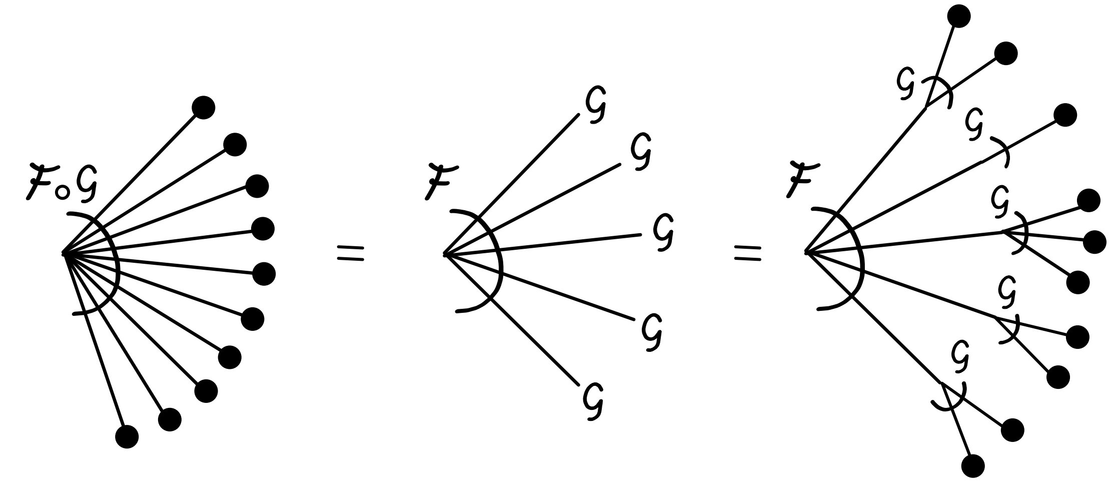


Requires $\mathcal{G}[\varnothing]=\varnothing$. Then $\mathcal{F}(\mathcal{G})[\varnothing]=\varnothing$ and

$$(\mathcal{F}(\mathcal{G}))[U] = \sum_{\pi \text{ partition of } U} \mathcal{F}[\pi] \times \prod_{p\in\pi} \mathcal{G}[p], \qquad U \neq \emptyset.$$

- $\mathcal{F}(Z)=\mathcal{F}$;
- Permutations: Perm=SET(CYC);
- Graphs=SET(Connected_Graphs);
- Trees: $\mathcal{T}$=Z·SET($\mathcal{T}$);

Substitution

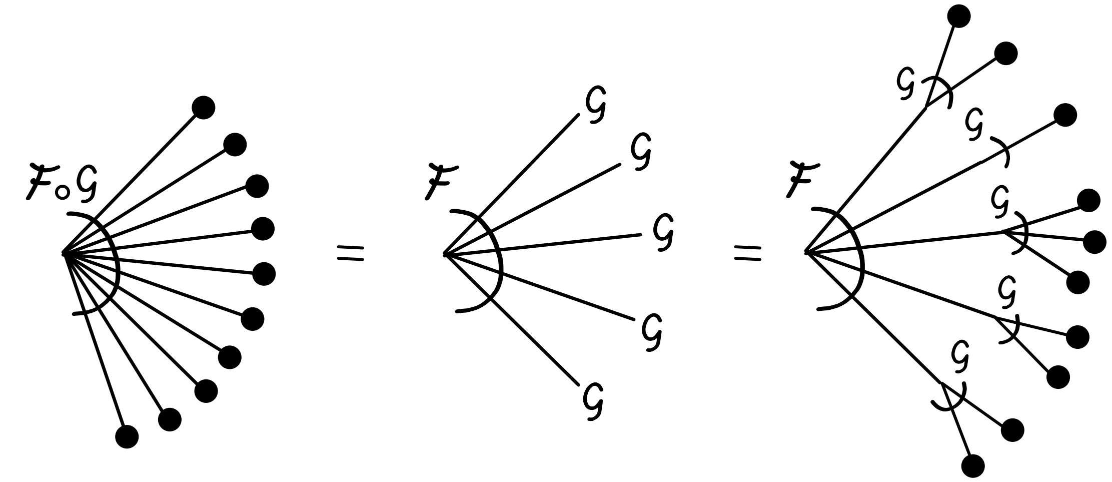


Requires $\mathcal{G}[\varnothing]=\varnothing$. Then $\mathcal{F}(\mathcal{G})[\varnothing]=\varnothing$ and

$$(\mathcal{F}(\mathcal{G}))[U] = \sum_{\pi \text{ partition of } U} \mathcal{F}[\pi] \times \prod_{p\in\pi} \mathcal{G}[p], \qquad U \neq \emptyset.$$

- $\mathcal{F}(Z)=\mathcal{F}$;
- Permutations: Perm=SET(CYC);
- Graphs=SET(Connected_Graphs);
- Trees: $\mathcal{T}$=Z·SET($\mathcal{T}$);
- Functional graphs: $\mathcal{F}$=SET(CYC($\mathcal{T}$))...

Substitution



Requires $\mathcal{G}[\varnothing]=\varnothing$. Then $\mathcal{F}(\mathcal{G})[\varnothing]=\varnothing$ and

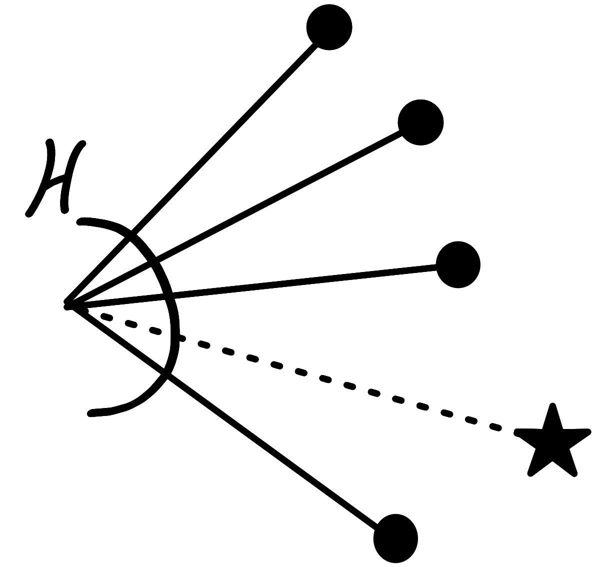
$$(\mathcal{F}(\mathcal{G}))[U] = \sum_{\pi \text{ partition of } U} \mathcal{F}[\pi] \times \prod_{p\in\pi} \mathcal{G}[p], \qquad U \neq \emptyset.$$

- $\mathcal{F}$(Z)=$\mathcal{F}$;
- Permutations: Perm=SET(CYC);
- Graphs=SET(Connected_Graphs);
- Trees: $\mathcal{T}$=Z·SET($\mathcal{T}$);
- Functional graphs: $\mathcal{F}$=SET(CYC($\mathcal{T}$))...

Huet's zipper

Derivative

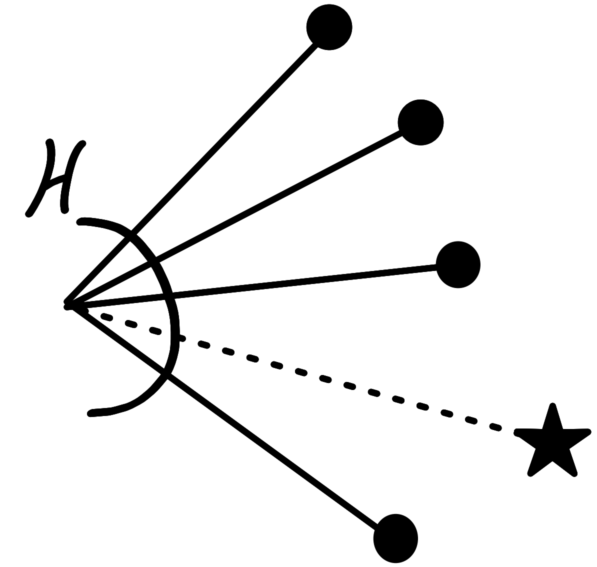
$$\mathcal{H}'[U] = \mathcal{H}[U + \{\star\}]$$



Derivative

Huet's zipper

$$\mathcal{H}'[U] = \mathcal{H}[U + \{\star\}]$$

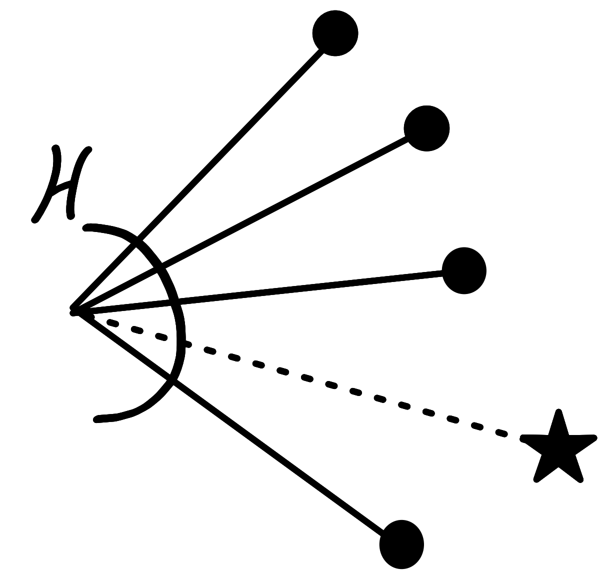


- 0'=1'=0; Z'=1; SET'=SET; CYC'=SEQ; SEQ'=SEQ·SEQ;

Huet's zipper

Derivative

$$\mathcal{H}'[U] = \mathcal{H}[U + \{\star\}]$$

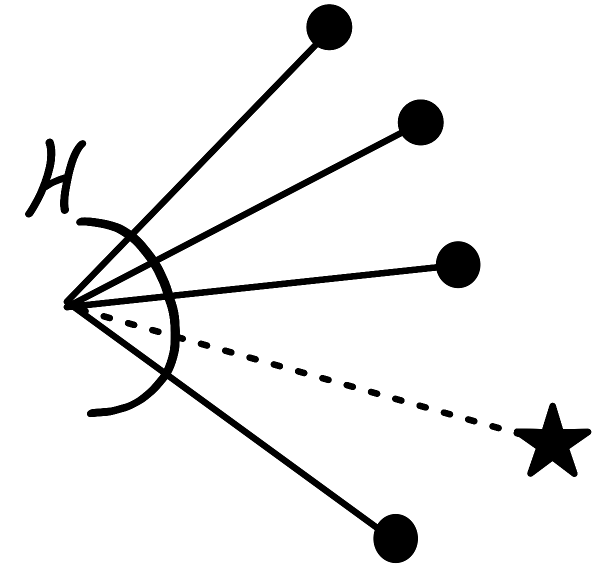


- 0'=1'=0; Z'=1; SET'=SET; CYC'=SEQ; SEQ'=SEQ·SEQ;
- $(\mathscr{F}+\mathscr{G})'=\mathscr{F}'+\mathscr{G}'$;$(\mathscr{F}\cdot\mathscr{G})'=\mathscr{F}'\cdot\mathscr{G}+\mathscr{F}\cdot\mathscr{G}'$;

Derivative

Huet's zipper

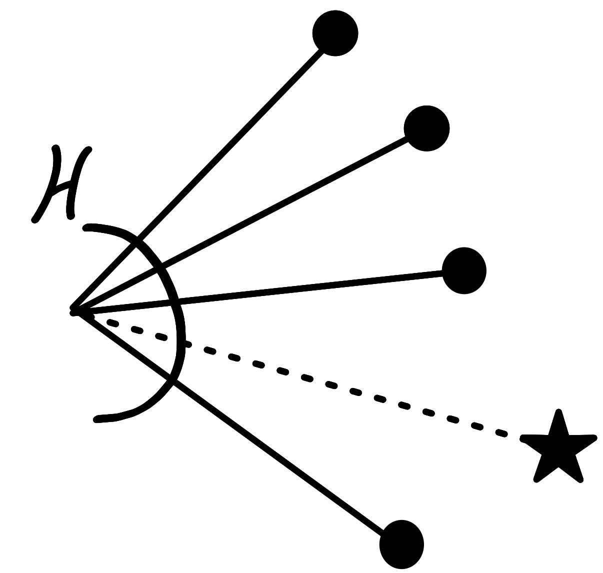
$$\mathcal{H}'[U] = \mathcal{H}[U + \{\star\}]$$



- 0'=1'=0; Z'=1; SET'=SET; CYC'=SEQ; SEQ'=SEQ·SEQ;
- $(\mathcal{F}+\mathcal{G})' = \mathcal{F}'+\mathcal{G}'$; $(\mathcal{F}\cdot\mathcal{G})' = \mathcal{F}'\cdot\mathcal{G}+\mathcal{F}\cdot\mathcal{G}'$;
- $\mathcal{F}(\mathcal{G})' = \mathcal{F}'(\mathcal{G})\cdot\mathcal{G}'$.

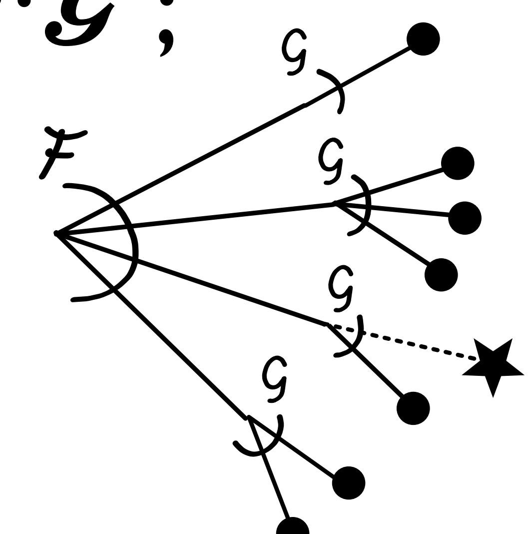
Huet's zipper

Derivative



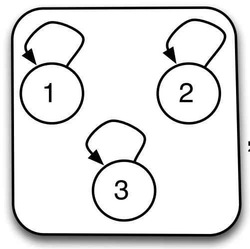
$$\mathcal{H}'[U] = \mathcal{H}[U + \{\star\}]$$

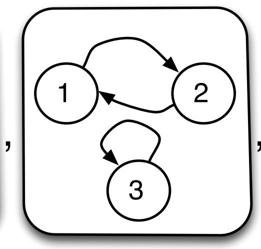
- 0'=1'=0; Z'=1; SET'=SET; CYC'=SEQ; SEQ'=SEQ·SEQ;
- $(\mathcal{F}+\mathcal{G})' = \mathcal{F}' + \mathcal{G}'; (\mathcal{F}\cdot\mathcal{G})' = \mathcal{F}'\cdot\mathcal{G} + \mathcal{F}\cdot\mathcal{G}'$;
- $\mathcal{F}(\mathcal{G})' = \mathcal{F}'(\mathcal{G})\cdot\mathcal{G}'$.

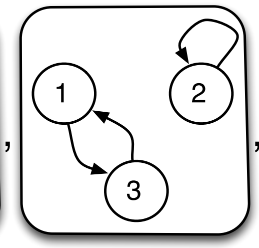


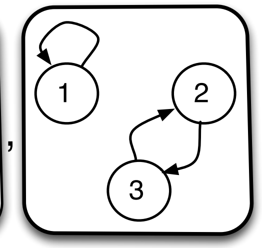
III. Enumeration

Generating Series

$\mathrm{Inv}[\{1,2,3\}] = \{$ 

, 

, 

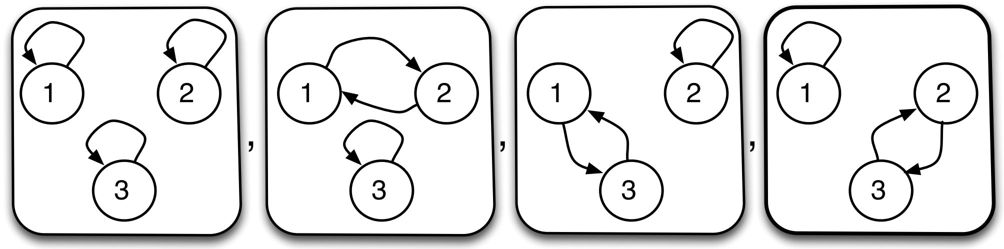
, 

$\}$

4 involutions;

3 of them permuted by $\mathfrak{S}_3$ → 2 *unlabelled structures*.

Generating Series

$\mathrm{Inv}[\{1,2,3\}] = \Big\{$  $\Big\}$

4 involutions;

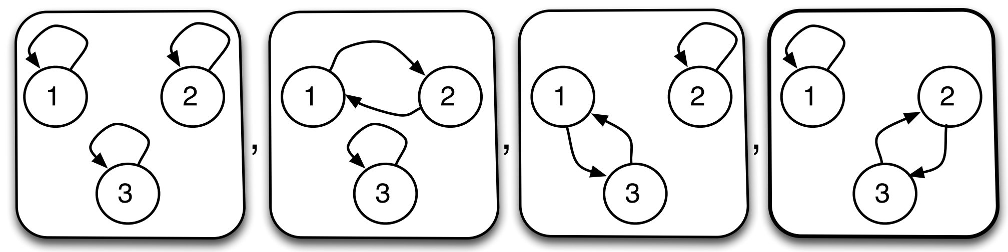
3 of them permuted by $\mathfrak{S}_3 \rightarrow$ 2 *unlabelled structures*.

Exponential generating series:

$$F(z) = \sum_{n=0}^{\infty} |\mathcal{F}[\{1,\dots,n\}]| \frac{z^n}{n!}.$$

$$\mathrm{Inv}_3(z) = \frac{2}{3} z^3$$

Generating Series

$\mathrm{Inv}[\{1,2,3\}] = \Big\{$ 

$\Big\}$

4 involutions;

3 of them permuted by $\mathfrak{S}_3$ → 2 *unlabelled structures*.

Exponential generating series:

$$F(z) = \sum_{n=0}^{\infty} |\mathcal{F}[\{1,\dots,n\}]| \frac{z^n}{n!}.$$

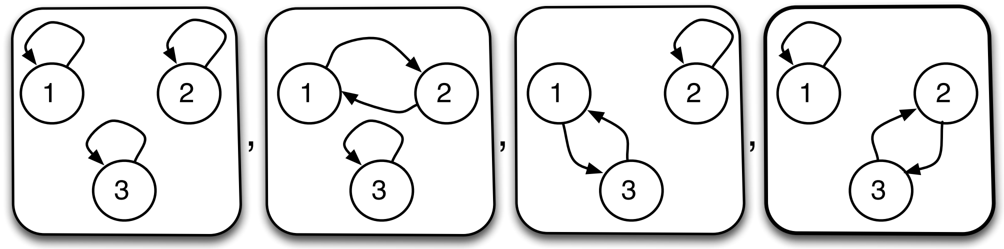
$$\mathrm{Inv}_3(z) = \frac{2}{3} z^3$$

Ordinary generating series:

$$\tilde{F}(z) = \sum_{n=0}^{\infty} \tilde{f}_n z^n, \quad \tilde{f}_n = \text{nb. unlabelled of size } n.$$

$$\widetilde{\mathrm{Inv}}_3(z) = 2z^3$$

Generating Series

$\mathrm{Inv}[\{1,2,3\}] = \Big\{$  $\Big\}$

4 involutions;

3 of them permuted by $\mathfrak{S}_3$ → 2 *unlabelled structures*.

Exponential generating series:

$$F(z) = \sum_{n=0}^{\infty} |\mathcal{F}[\{1,\ldots,n\}]| \frac{z^n}{n!}.$$

$$\mathrm{Inv}_3(z) = \frac{2}{3} z^3$$

Ordinary generating series:

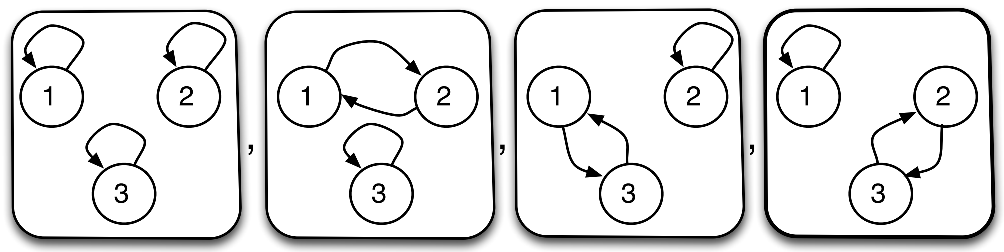
$$\tilde{F}(z) = \sum_{n=0}^{\infty} \tilde{f}_n z^n, \quad \tilde{f}_n = \text{nb. unlabelled of size } n.$$

$$\widetilde{\mathrm{Inv}}_3(z) = 2z^3$$

Cycle index series:

$$Z_{\mathcal{F}}(z_1, z_2, z_3, \ldots) = \sum_{n\geq 0} \frac{1}{n!} \left(\sum_{\sigma\in\mathfrak{S}_n} \mathrm{fix}\,\mathcal{F}[\sigma] z_1^{\sigma_1} z_2^{\sigma_2} \cdots \right).$$

Generating Series

$\mathrm{Inv}[\{1,2,3\}] = \Big\{$  $\Big\}$

4 involutions;

3 of them permuted by $\mathfrak{S}_3$ → 2 *unlabelled structures*.

Exponential generating series:

$$F(z) = \sum_{n=0}^{\infty} |\mathcal{F}[\{1,\dots,n\}]| \frac{z^n}{n!}.$$

$$\mathrm{Inv}_3(z) = \frac{2}{3} z^3$$

Ordinary generating series:

$$\tilde{F}(z) = \sum_{n=0}^{\infty} \tilde{f}_n z^n, \quad \tilde{f}_n = \text{nb. unlabelled of size } n.$$

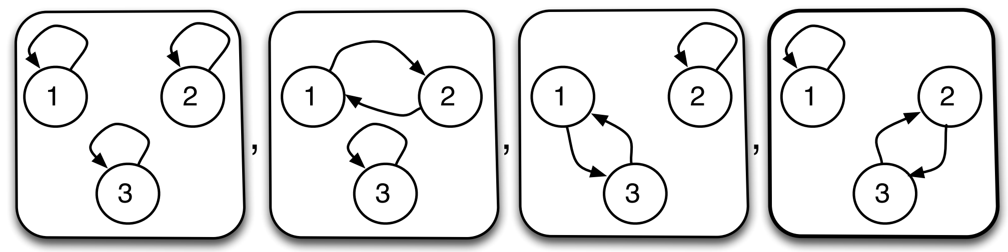
$$\widetilde{\mathrm{Inv}}_3(z) = 2z^3$$

Cycle index series:

$$\mathrm{fix}\,\mathcal{F}[\sigma] = |\{f \in \mathcal{F}[\{1,\dots,n\}], \sigma(f) = f\}|$$

$$Z_{\mathcal{F}}(z_1, z_2, z_3, \dots) = \sum_{n\geq 0} \frac{1}{n!} \left(\sum_{\sigma \in \mathfrak{S}_n} \mathrm{fix}\,\mathcal{F}[\sigma] z_1^{\sigma_1} z_2^{\sigma_2} \cdots \right).$$

Generating Series

$\mathrm{Inv}[\{1,2,3\}] = \{$  $\}$

4 involutions;

3 of them permuted by $\mathfrak{S}_3 \rightarrow$ 2 *unlabelled structures.*

Exponential generating series:

$$F(z) = \sum_{n=0}^{\infty} |\mathcal{F}[\{1,\dots,n\}]| \frac{z^n}{n!}.$$

$$\mathrm{Inv}_3(z) = \frac{2}{3} z^3$$

Ordinary generating series:

$$\tilde{F}(z) = \sum_{n=0}^{\infty} \tilde{f}_n z^n, \quad \tilde{f}_n = \text{nb. unlabelled of size } n.$$

$$\widetilde{\mathrm{Inv}}_3(z) = 2z^3$$

Cycle index series:

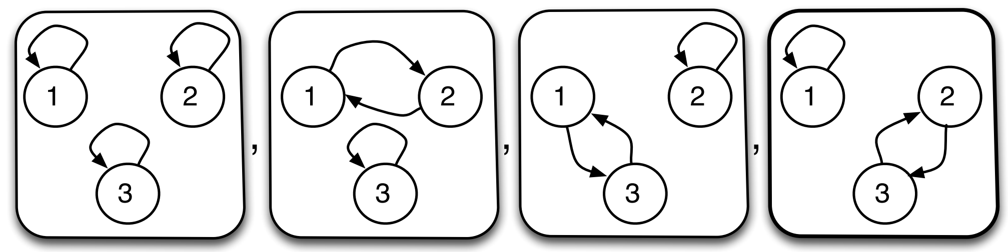
$$\mathrm{fix}\,\mathcal{F}[\sigma] = |\{f \in \mathcal{F}[\{1,\dots,n\}], \sigma(f) = f\}|$$

$$Z_{\mathcal{F}}(z_1, z_2, z_3, \dots) = \sum_{n \geq 0} \frac{1}{n!} \left(\sum_{\sigma \in \mathfrak{S}_n} \mathrm{fix}\,\mathcal{F}[\sigma] z_1^{\sigma_1} z_2^{\sigma_2} \cdots \right).$$

$$Z_{\mathrm{Inv}_3} = \frac{2z_1^3}{3} + z_1 z_2 + \frac{z_3}{3}$$

Generating Series

$\mathrm{Inv}[\{1,2,3\}] = \{$



$\}$

4 involutions;

3 of them permuted by $\mathfrak{S}_3 \rightarrow$ 2 *unlabelled structures*.

Exponential generating series:

$$F(z) = \sum_{n=0}^{\infty} |\mathcal{F}[\{1,\dots,n\}]| \frac{z^n}{n!}.$$

$$\mathrm{Inv}_3(z) = \frac{2}{3} z^3$$

Ordinary generating series:

$$\tilde{F}(z) = \sum_{n=0}^{\infty} \tilde{f}_n z^n, \quad \tilde{f}_n = \text{nb. unlabelled of size } n.$$

$$\widetilde{\mathrm{Inv}}_3(z) = 2z^3$$

Cycle index series:

$$\operatorname{fix} \mathcal{F}[\sigma] = |\{f \in \mathcal{F}[\{1,\dots,n\}], \sigma(f) = f\}|$$

$$Z_{\mathcal{F}}(z_1, z_2, z_3, \dots) = \sum_{n \geq 0} \frac{1}{n!} \left(\sum_{\sigma \in \mathfrak{S}_n} \operatorname{fix} \mathcal{F}[\sigma] z_1^{\sigma_1} z_2^{\sigma_2} \cdots \right).$$

$$Z_{\mathrm{Inv}_3} = \frac{2z_1^3}{3} + z_1 z_2 + \frac{z_3}{3}$$

$$F(z) = Z_{\mathcal{F}}(z, 0, 0, \dots); \quad \tilde{F}(z) = Z_{\mathcal{F}}(z, z^2, z^3, \dots)$$

Computation: a dictionary

Computation: a dictionary

$$Z_0 = 0; Z_1 = 1; Z_{\mathcal{Z}} = z_1;$$

Computation: a dictionary

$$Z_0 = 0; Z_1 = 1; Z_{\mathcal{Z}} = z_1;$$

$$Z_{\mathcal{F}+\mathcal{G}} = Z_{\mathcal{F}} + Z_{\mathcal{G}}; Z_{\mathcal{F}\cdot\mathcal{G}} = Z_{\mathcal{F}} \cdot Z_{\mathcal{G}}; Z_{\mathcal{F}'} = \frac{\partial}{\partial z_1} Z_{\mathcal{F}};$$

Computation: a dictionary

$$Z_0 = 0; Z_1 = 1; Z_{\mathcal{Z}} = z_1;$$

$$Z_{\mathcal{F}+\mathcal{G}} = Z_{\mathcal{F}} + Z_{\mathcal{G}}; Z_{\mathcal{F}\cdot\mathcal{G}} = Z_{\mathcal{F}} \cdot Z_{\mathcal{G}}; Z_{\mathcal{F}'} = \frac{\partial}{\partial z_1} Z_{\mathcal{F}};$$

$$Z_{\textsc{Seq}} = \frac{1}{1 - z_1}; Z_{\textsc{Set}} = \exp(\sum_{k>0} \frac{z_k}{k}); Z_{\textsc{Cyc}} = \sum_{k>0} \frac{\phi(k)}{k} \log \frac{1}{1 - z_k};$$

Computation: a dictionary

$$Z_0 = 0; Z_1 = 1; Z_{\mathcal{Z}} = z_1;$$

$$Z_{\mathcal{F}+\mathcal{G}} = Z_{\mathcal{F}} + Z_{\mathcal{G}}; Z_{\mathcal{F}\cdot\mathcal{G}} = Z_{\mathcal{F}} \cdot Z_{\mathcal{G}}; Z_{\mathcal{F}'} = \frac{\partial}{\partial z_1} Z_{\mathcal{F}};$$

$$Z_{\text{SEQ}} = \frac{1}{1-z_1}; Z_{\text{SET}} = \exp(\sum_{k>0} \frac{z_k}{k}); Z_{\text{CYC}} = \sum_{k>0} \frac{\phi(k)}{k} \log \frac{1}{1-z_k};$$

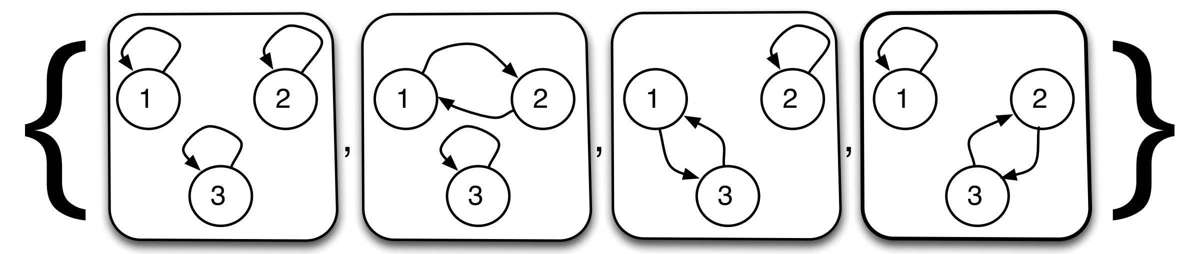
Main Result [Joyal 1981].

$$Z_{\mathcal{F}(\mathcal{G})}(z_1, z_2, \dots) = Z_{\mathcal{F}}(Z_{\mathcal{G}}(z_1, z_2, \dots), Z_{\mathcal{G}}(z_2, z_4, \dots), Z_{\mathcal{G}}(z_3, z_6, \dots), \dots).$$

(plethystic substitution).

Example: involutions

$$\mathrm{Inv} = \textsc{Set}(\textsc{Cyc}_{\leq 2})$$



$$Z_{\textsc{Cyc}_{\leq 2}} = z_1 + \frac{1}{2} z_1^2 + \frac{1}{2} z_2$$

$$Z_{\mathrm{Inv}} = \exp\left(\sum_k \frac{z_k}{k} + \frac{z_k^2}{2k} + \frac{z_{2k}}{2k} \right)$$

$$\mathrm{Inv}(z) = \exp\left(z + \frac{z^2}{2} \right) = 1 + z + z^2 + \frac{2}{3} z^3 + \cdots ,$$

$$\widetilde{\mathrm{Inv}}(z) = \frac{1}{1-z} \frac{1}{1-z^2} = 1 + z + 2z^2 + 2z^3 + \cdots .$$

Short-cuts in practice,
particularly for exponential generating series.

Fast computation in non-recursive cases reduces to exp, log, reciprocal.

More examples

More examples

Permutations: $\mathrm{Perm} = \textsc{Set}(\textsc{Cyc})$

$$\longrightarrow P(z) = \exp(\log \frac{1}{1-z}) = \frac{1}{1-z};$$

$$\longrightarrow \tilde{P}(z) = \prod_k \frac{1}{1-z^k}.$$

More examples

Permutations: $\mathrm{Perm} = \textsc{Set}(\textsc{Cyc})$

$$\longrightarrow P(z) = \exp(\log \frac{1}{1-z}) = \frac{1}{1-z};$$

$$\longrightarrow \tilde{P}(z) = \prod_k \frac{1}{1-z^k}.$$

More examples

Permutations: $\mathrm{Perm} = \textsc{Set}(\textsc{Cyc})$

$$\longrightarrow P(z) = \exp(\log \frac{1}{1-z}) = \frac{1}{1-z};$$

$$\longrightarrow \tilde{P}(z) = \prod_k \frac{1}{1-z^k}.$$

Binary trees: $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$

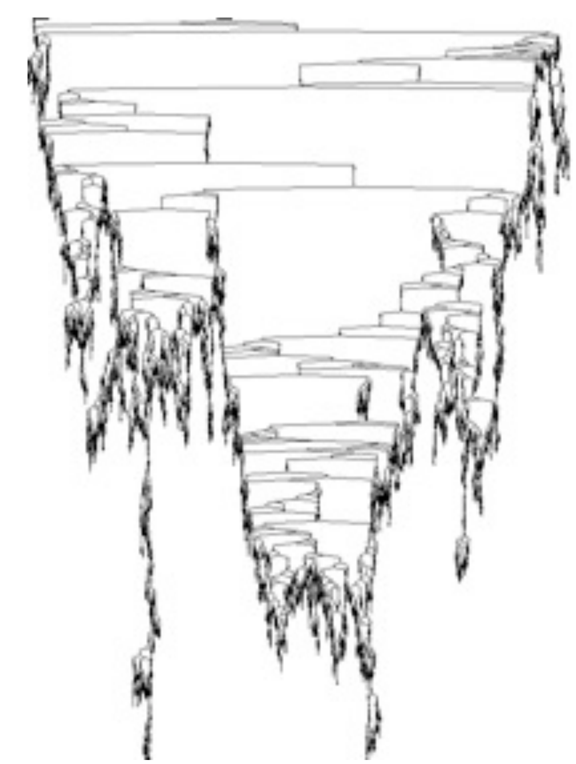
$$\longrightarrow B(z) = z + zB(z)^2 = \tilde{B}(z)$$

More examples

Permutations: $\text{Perm} = \textsc{Set}(\textsc{Cyc})$

$$\longrightarrow P(z) = \exp(\log \frac{1}{1-z}) = \frac{1}{1-z};$$

$$\longrightarrow \tilde{P}(z) = \prod_k \frac{1}{1-z^k}.$$



Binary trees: $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$

$$\longrightarrow B(z) = z + zB(z)^2 = \tilde{B}(z)$$

Cayley trees: $\mathcal{T} = \mathcal{Z} \cdot \textsc{Set}(\mathcal{T})$

$$\longrightarrow T(z) = z\exp(T(z));$$

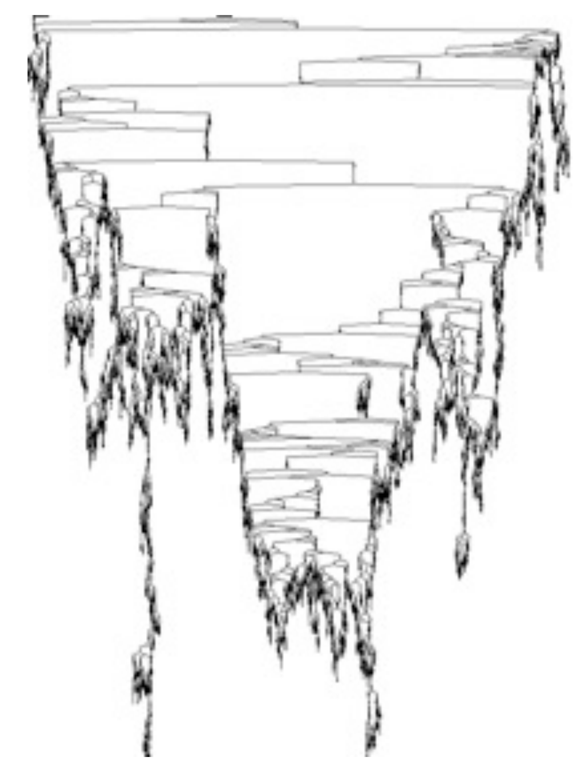
$$\longrightarrow \tilde{T}(z) = z\exp\left(\tilde{T}(z) + \frac{1}{2}\tilde{T}(z^2) + \frac{1}{3}\tilde{T}(z^3) + \cdots\right)$$

More examples

Permutations: $\mathrm{Perm} = \textsc{Set}(\textsc{Cyc})$

$$\longrightarrow P(z) = \exp(\log \frac{1}{1-z}) = \frac{1}{1-z};$$

$$\longrightarrow \tilde{P}(z) = \prod_k \frac{1}{1-z^k}.$$



Binary trees: $\mathcal{B} = \mathcal{Z} + \mathcal{Z} \times \mathcal{B} \times \mathcal{B}$

$$\longrightarrow B(z) = z + zB(z)^2 = \tilde{B}(z)$$

Cayley trees: $\mathcal{T} = \mathcal{Z} \cdot \textsc{Set}(\mathcal{T})$

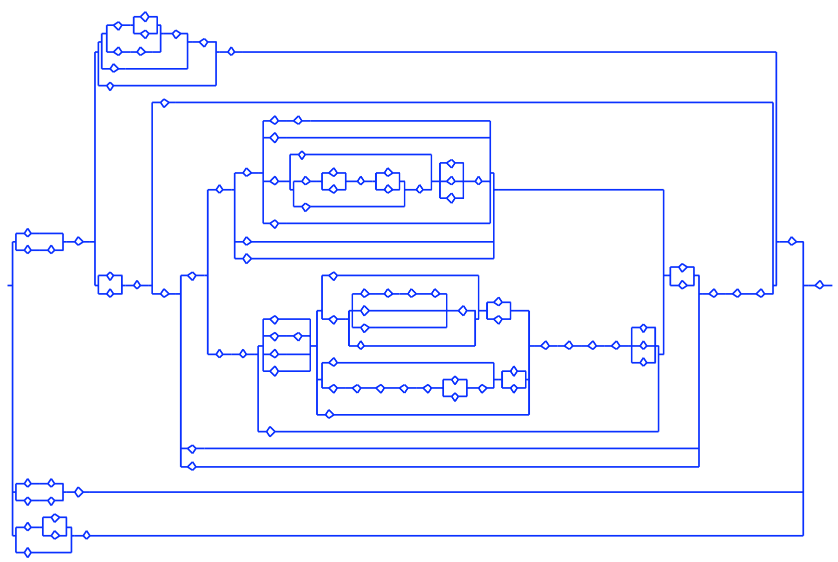
$$\longrightarrow T(z) = z\exp(T(z));$$

$$\longrightarrow \tilde{T}(z) = z\exp\left(\tilde{T}(z) + \frac{1}{2}\tilde{T}(z^2) + \frac{1}{3}\tilde{T}(z^3) + \cdots\right)$$

IV. Implicit Species

Use Equations to Define Species

- Sequences: SEQ=1+Z·SEQ;
- Trees of various kinds:
  $\mathscr{B}$=1+Z·$\mathscr{B}$·$\mathscr{B}$; $\mathscr{T}$=Z·SET($\mathscr{T}$);
- Series-parallel graphs:



$$\mathcal{G} = \mathcal{Z} + \mathcal{S} + \mathcal{P}, \mathcal{S} = \mathrm{S{\scriptstyle EQ}}_{>0}(\mathcal{Z} + \mathcal{P}), \mathcal{P} = \mathrm{S{\scriptstyle ET}}_{>0}(\mathcal{Z} + \mathcal{S})$$

Recursive combinatorial specifications

Multisort Species

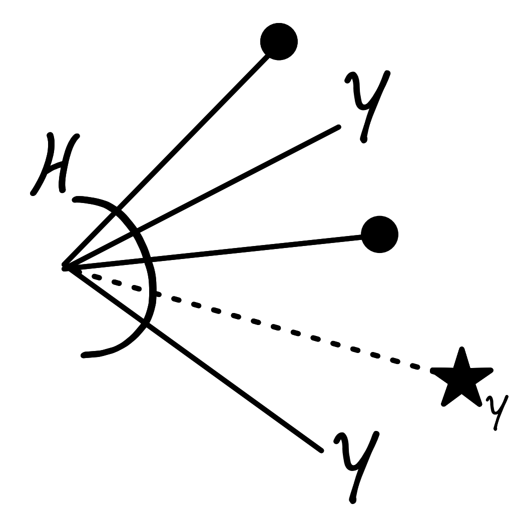
One Z $\longrightarrow$ $Z_1, Z_{2,\ldots,}$ Z_k

defined by $\mathscr{F}[U_1,U_2,\ldots,U_k]$ and $\mathscr{F}[\sigma_1,\ldots,\sigma_k]$.

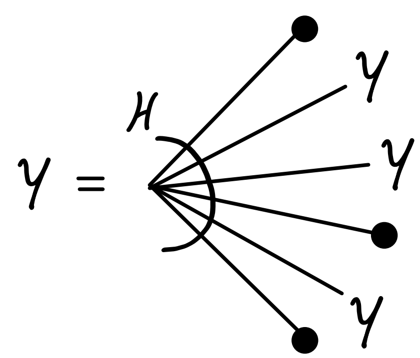
Examples: + and ·

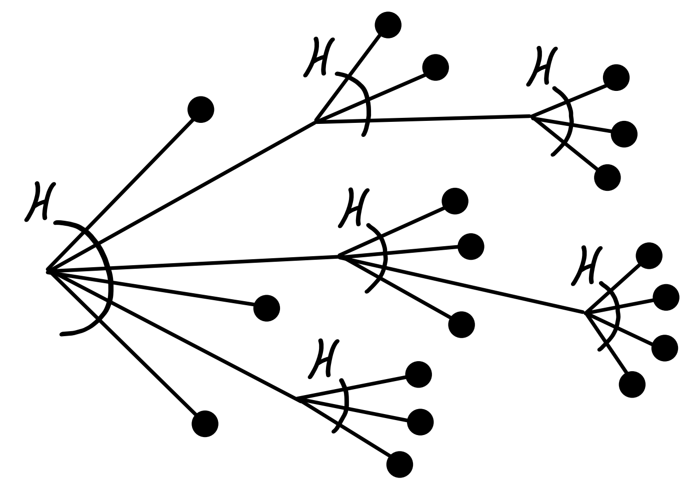
All the definitions extend: substitution, derivative (becomes partial derivative).

$$\frac{\partial}{\partial \mathcal{Y}} \mathcal{H}(\mathcal{Z}, \mathcal{Y}) \quad :$$

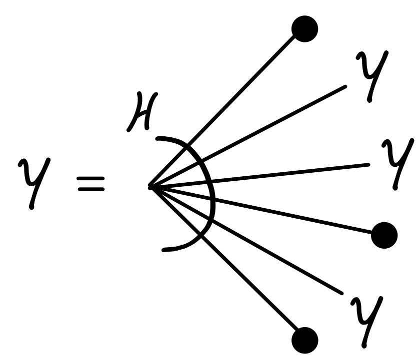


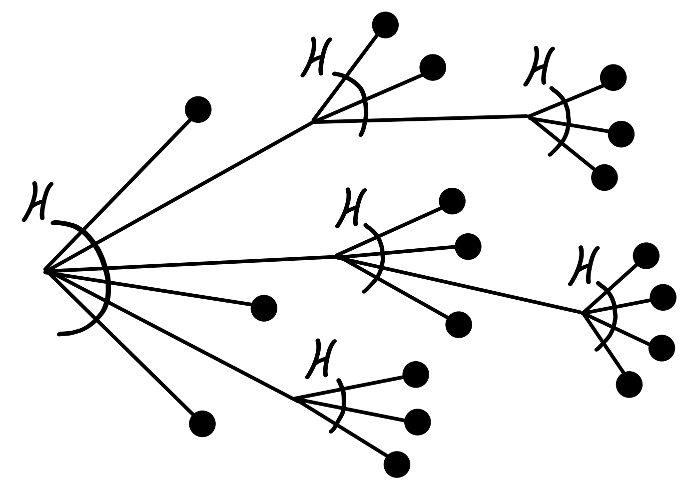
Joyal's Implicit Species Theorem





Joyal's Implicit Species Theorem





Thm. If $\mathscr{H}(0,0)=0$ and $\partial\mathscr{H}/\partial\mathscr{Y}(0,0)$ is nilpotent, then $\mathscr{Y}=\mathscr{H}(\mathscr{Z},\mathscr{Y})$ has a unique solution, limit of $\mathscr{Y}^{[0]}=0,\quad \mathscr{Y}^{[n+1]}=\mathscr{H}(\mathscr{Z},\mathscr{Y}^{[n+1]})\quad (n\geq 0)$.

Joyal’s Implicit Species Theorem

Thm. If $\mathcal{H}(0,0)=0$ and $\partial\mathcal{H}/\partial\mathcal{Y}(0,0)$ is nilpotent, then $\mathcal{Y}=\mathcal{H}(\mathcal{Z},\mathcal{Y})$ has a unique solution, limit of $\mathcal{Y}^{[0]}=0,\quad \mathcal{Y}^{[n+1]}=\mathcal{H}(\mathcal{Z},\mathcal{Y}^{[n+1]})\quad (n\geq 0).$

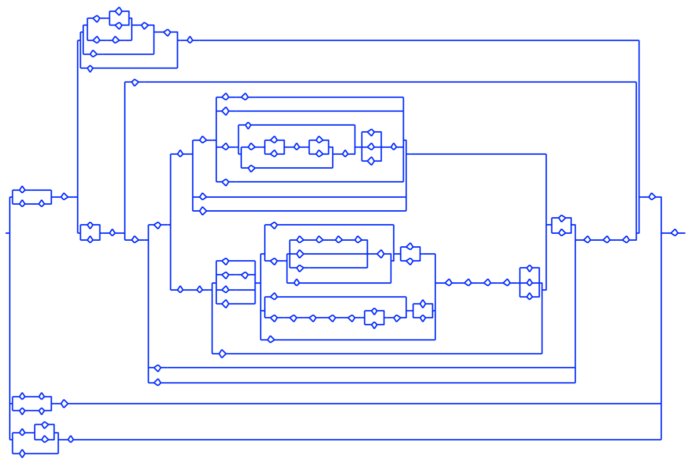
Can be turned into an iff when no 0 coordinate.

Example

$$\mathcal{H}(\mathcal{G}, \mathcal{S}, \mathcal{P}) := (\mathcal{S} + \mathcal{P}, \textsc{Seq}_{>0}(\mathcal{Z} + \mathcal{P}), \textsc{Set}_{>1}(\mathcal{Z} + \mathcal{S}))$$

$$\mathcal{H}(0,0,0) = (0,0,0)$$

$$\frac{\partial \mathcal{H}}{\partial \mathcal{Y}} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & \textsc{Seq}(\mathcal{Z} + \mathcal{P})^2 \\ 0 & \textsc{Set}_{>0}(\mathcal{Z} + \mathcal{S}) & 0 \end{pmatrix}$$



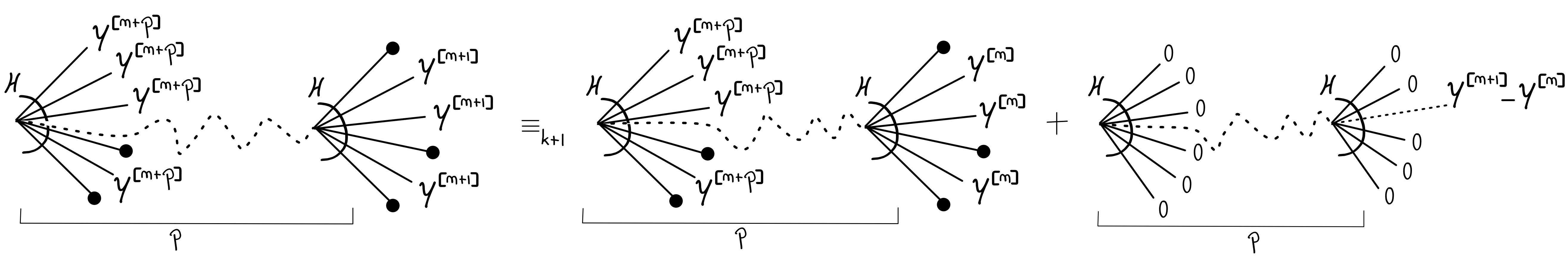
$$\frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(0,0,0) = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$\longrightarrow$ this specification is meaningful.

Proof

Def. $\mathscr{A} =_k \mathscr{B}$ if they coincide up to size k (contact).

Key Lemma.
If $\mathscr{Y}^{[n+1]} =_k \mathscr{Y}^{[n]}$, then $\mathscr{Y}^{[n+p+1]} =_{k+1} \mathscr{Y}^{[n+p]}$ (p=dim).



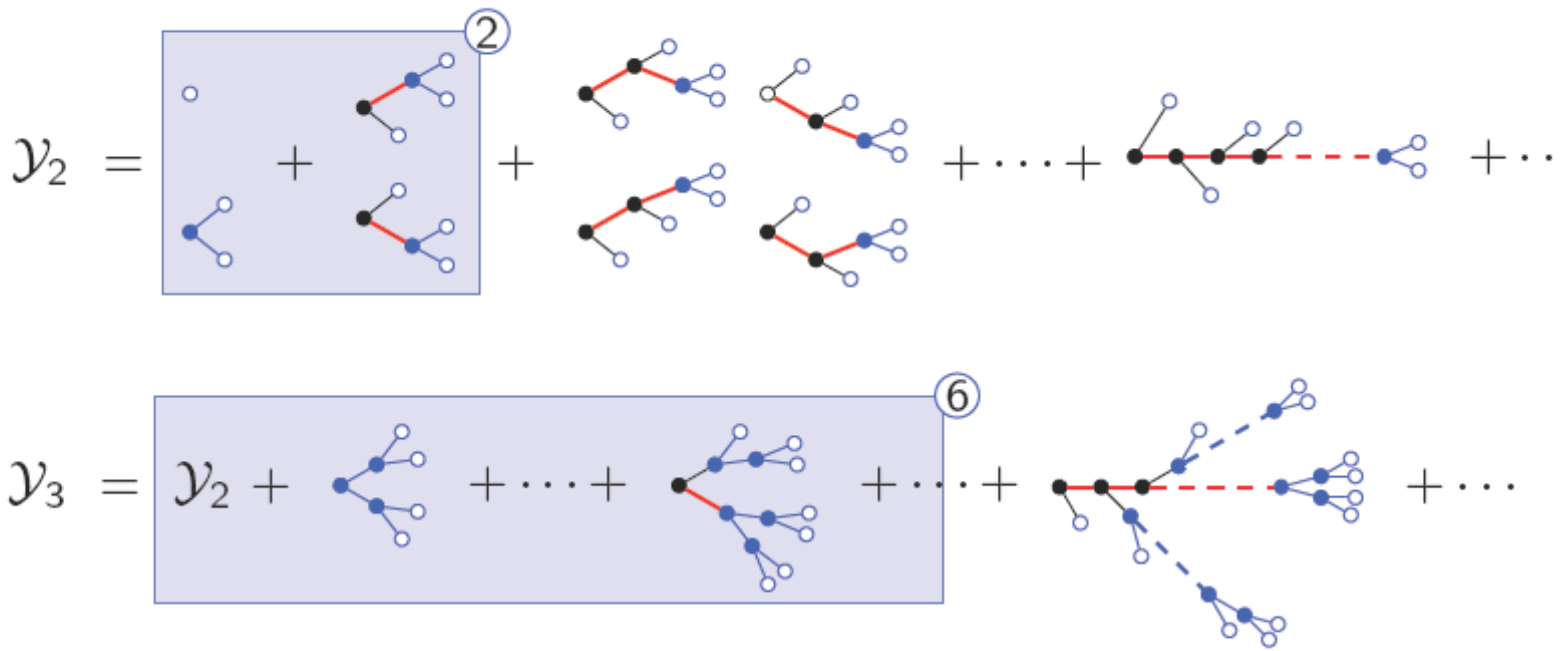
V. Newton Iteration

Newton Iteration for binary trees

$$\mathcal{Y} = 1 + \mathcal{Z} \cdot \mathcal{Y} \cdot \mathcal{Y}$$

$$\mathcal{Y}^{[n+1]} = \mathcal{Y}^{[n]} + \textsc{Seq}(\mathcal{Z} \cdot \mathcal{Y}^{[n]} \cdot \star + \mathcal{Z} \cdot \star \cdot \mathcal{Y}^{[n]}) \cdot ((1 + \mathcal{Z} \cdot \mathcal{Y}^{[n]} \cdot \mathcal{Y}^{[n]}) \setminus \mathcal{Y}^{[n]}).$$

$\mathcal{Y}_0 = \varnothing \qquad \mathcal{Y}_1 = \circ$

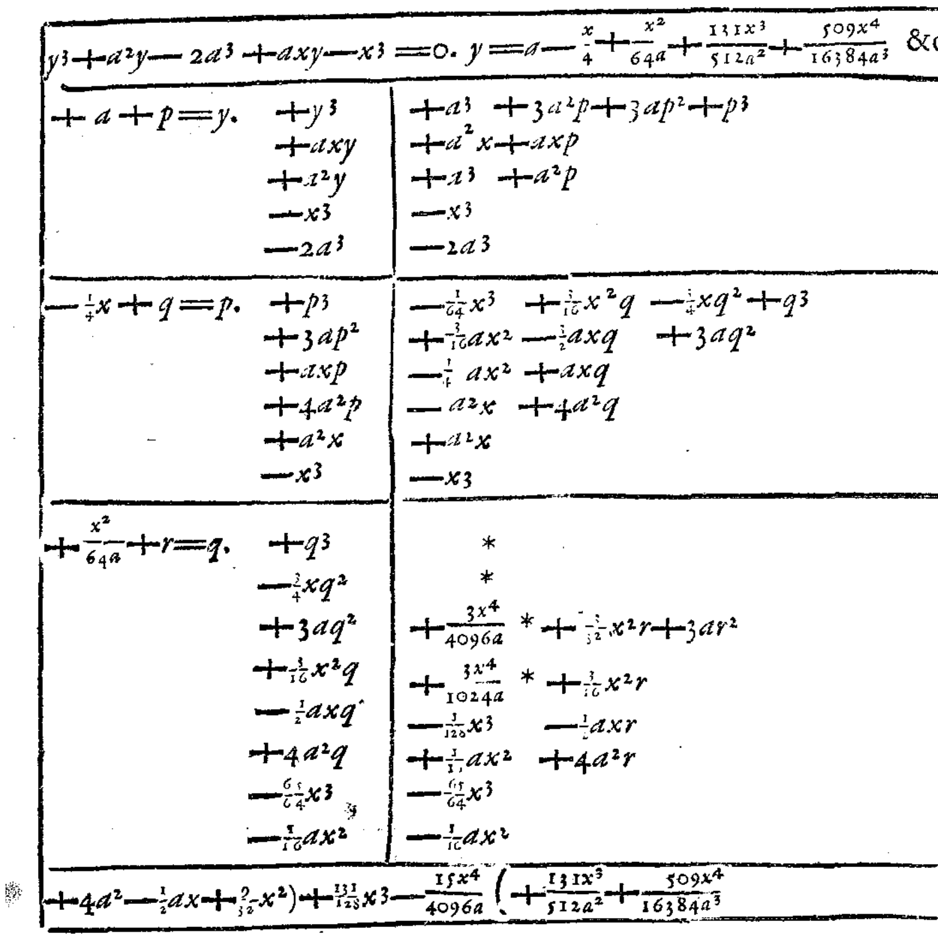


[Décoste, Labelle, Leroux 1982]

Generating Series

$$y^{[n+1]} = y^{[n]} + \frac{1 + z{y^{[n]}}^2 - y^{[n]}}{1 - 2zy^{[n]}}$$

solves $y = 1 + zy^2$


$y^3 + a^2y - 2a^3 + axy - x^3 = 0.\ y = a - \frac{x}{4} + \frac{x^2}{64a} + \frac{131x^3}{512a^2} + \frac{509x^4}{16384a^3}$ &c.

$+a+p=y.$	$+y^3$ $+axy$ $+a^2y$ $-x^3$ $-2a^3$	$+a^3 \quad +3a^2p + 3ap^2 + p^3$ $+a^2x + axp$ $+a^3 \quad +a^2p$ $-x^3$ $-2a^3$
$-\frac{1}{4}x + q = p.$	$+p^3$ $+3ap^2$ $+axp$ $+4a^2p$ $+a^2x$ $-x^3$	$-\frac{1}{64}x^3 \quad +\frac{3}{16}x^2q \quad -\frac{3}{4}xq^2 + q^3$ $+\frac{3}{16}ax^2 \quad -\frac{3}{2}axq \quad +3aq^2$ $-\frac{1}{4}ax^2 \quad +axq$ $-a^2x \quad +4a^2q$ $+a^2x$ $-x^3$
$+\frac{x^2}{64a} + r = q.$	$+q^3$ $-\frac{3}{4}xq^2$ $+3aq^2$ $+\frac{3}{16}x^2q$ $-\frac{1}{2}axq$ $+4a^2q$ $-\frac{65}{64}x^3$ $-\frac{1}{16}ax^2$	* * $+\frac{3x^4}{4096a}$ * $-\frac{3}{32}x^2r + 3ar^2$ $+\frac{3x^4}{1024a}$ * $+\frac{3}{16}x^2r$ $-\frac{3}{128}x^3 \quad -\frac{1}{2}axr$ $+\frac{1}{32}ax^2 \quad +4a^2r$ $-\frac{65}{64}x^3$ $-\frac{1}{16}ax^2$

$+4a^2 - \frac{1}{2}ax + \frac{9}{32}x^2) + \frac{131}{128}x^3 - \frac{15x^4}{4096a}\ (+ \frac{131x^3}{512a^2} + \frac{509x^4}{16384a^3}$

$$y^{[0]} = 0$$

$$y^{[1]} = \boxed{1}$$

$$y^{[2]} = \boxed{1 + z + 2z^2} + 4z^3 + 8z^4 + 16z^5 + 32z^6 + 64z^7 + \dots$$

$$y^{[3]} = \boxed{1 + z + 2z^2 + 5z^3 + 14z^4 + 42z^5 + 132z^6} + 428z^7 + \dots$$

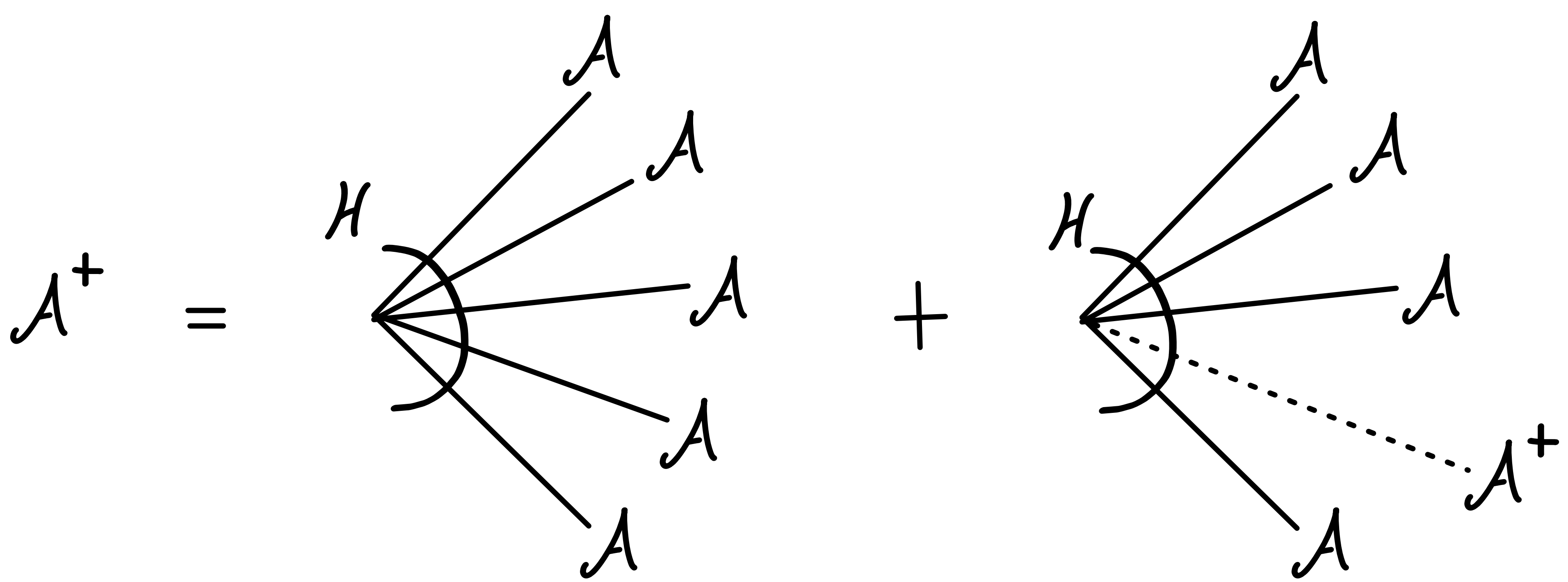
[Newton1671]

Combinatorial Newton Iteration

Thm. [essentially Labelle] For any well-founded system $\mathcal{Y}=\mathcal{H}(\mathcal{Z},\mathcal{Y})$, if $\mathcal{A}$ has contact k with the solution and $\mathcal{A}\subset\mathcal{H}(\mathcal{Z},\mathcal{A})$, then

$$\mathcal{A}+\sum_{i\geq 0}(\partial\mathcal{H}/\partial\mathcal{Y}(\mathcal{Z},\mathcal{A}))^i\cdot(\mathcal{H}(\mathcal{Z},\mathcal{A})\setminus\mathcal{A})$$

has contact $2k+1$ with it.



Example: Unlabelled Rooted Trees

Combinatorial equation: $\mathcal{T} = \mathcal{Z} \cdot \textsc{Set}(\mathcal{T}) =: \mathcal{H}(\mathcal{Z}, \mathcal{T})$

Combinatorial Newton iteration:

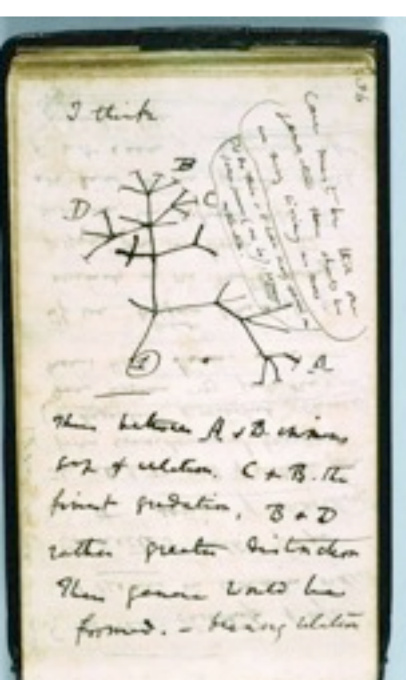
$$\mathcal{T}^{[n+1]} = \mathcal{T}^{[n]} + \textsc{Seq}(\mathcal{H}(\mathcal{T}^{[n]})) \cdot (\mathcal{H}(\mathcal{T}^{[n]}) \setminus \mathcal{T}^{[n]})$$

OGF equation: $\tilde{T}(z) = H(z, \tilde{T}(z))$

$$\tilde{T}(z) = z \exp(\tilde{T}(z) + \frac{1}{2}\tilde{T}(z^2) + \frac{1}{3}\tilde{T}(z^3) + \cdots)$$

Newton for OGF

$$\tilde{T}^{[n+1]} = \tilde{T}^{[n]} + \frac{H(z, \tilde{T}^{[n]}) - \tilde{T}^{[n]}}{1 - H(z, \tilde{T}^{[n]})}$$

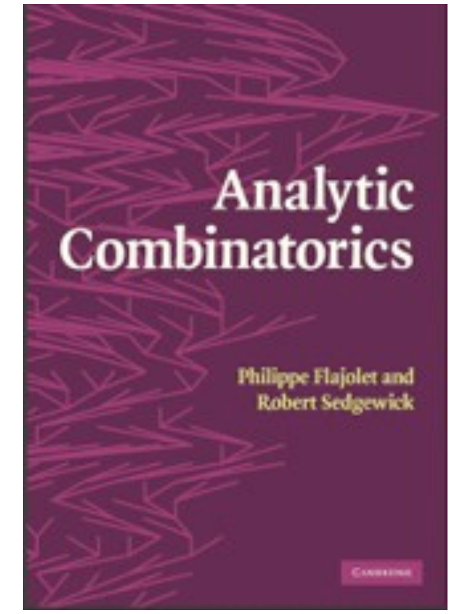


$0,$

$z + z^2 + z^3 + z^4 + \cdots ,$

$z + z^2 + 2z^3 + 4z^4 + 9z^5 + 20z^6 + \cdots$

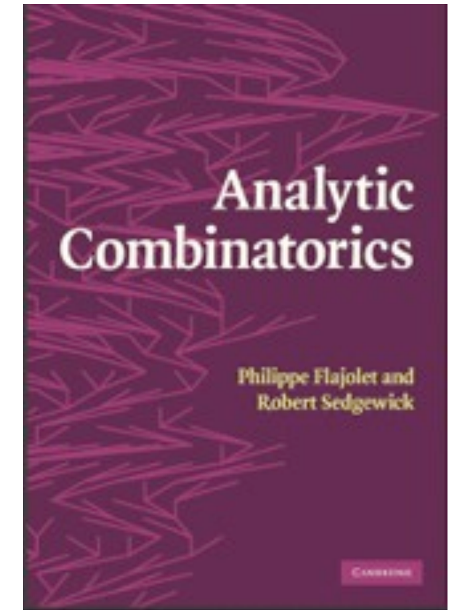
Our Result



Def. Constructible species. Either:

- one of : {1,Z,+,·, SEQ, CYC, SET};
- same with cardinality constraints;
- a substitution of constructible species;
- the solution of a well-founded system $\mathcal{Y}=\mathcal{H}(\mathcal{Z},\mathcal{Y})$ with $\mathcal{H}$ constructible.

Our Result



Def. Constructible species. Either:

- one of : $\{1, Z, +, \cdot, \textsc{Seq}, \textsc{Cyc}, \textsc{Set}\}$;
- same with cardinality constraints;
- a substitution of constructible species;
- the solution of a well-founded system $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$ with $\mathcal{H}$ constructible.

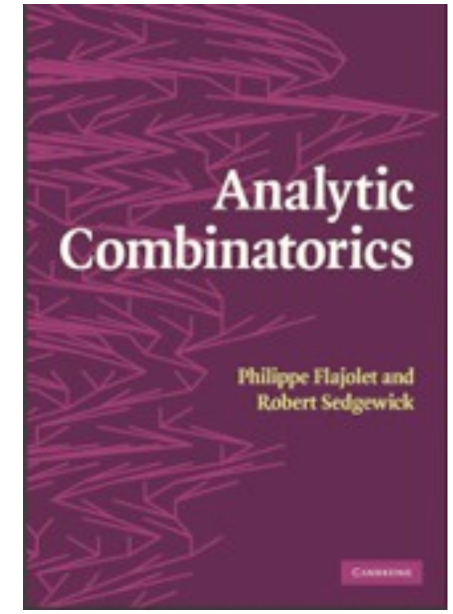
Thm. [Pivoteau-S-Soria 2012] First N coefficients of GFs of constructible species in quasi-optimal complexity

1. arithmetic complexity $O(N \log N)$ (both ogf & egf);

2. bit complexity

- $O(N^2 \log^2 N \log\log N)$ (ogf);
- $O(N^2 \log^3 N \log\log N)$ (egf).

Our Result



Def. Constructible species. Either:

- one of : {1,Z,+,·, Seq, Cyc, Set};
- same with cardinality constraints;
- a substitution of constructible species;
- the solution of a well-founded system $\mathcal{Y}=\mathcal{H}(\mathcal{Z},\mathcal{Y})$ with $\mathcal{H}$ constructible.

Thm. [Pivoteau-S-Soria 2012] First N coefficients of GFs of constructible species in quasi-optimal complexity

1. arithmetic complexity $O(N \log N)$ (both ogf & egf);

2. bit complexity

- $O(N^2 \log^2 N \log\log N)$ (ogf);
- $O(N^2 \log^3 N \log\log N)$ (egf).

Principle:

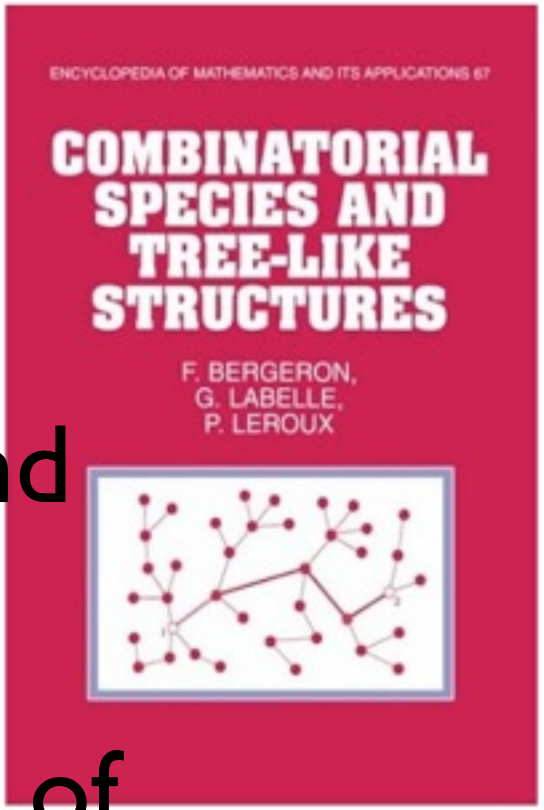
divide-and-conquer (Newton) + fast multiplication

Conclusion

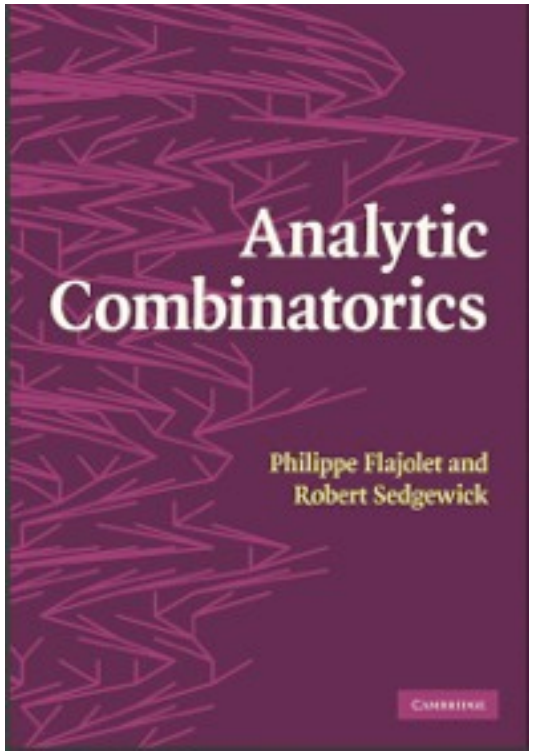
Much more in the litterature

- Linear species;
- Flat species;
- Acyclic species;
- Virtual/weighted species;
- Molecular decomposition;
- Analytic species...

References



- The seminal article by Joyal;
- Bergeron-Labelle-Leroux for a good and complete introduction;
- Flajolet-Sedgewick for a different point of view on the same objects;
- Many articles, notably by Labelle;
- Our own in JCTA 2012 with C. Pivoteau and M. Soria for algorithmic aspects.



References

- The seminal article by Joyal;
- Bergeron-Labelle-Leroux for a good and complete introduction;
- Flajolet-Sedgewick for a different point of view on the same objects;
- Many articles, notably by Labelle;
- Our own in JCTA 2012 with C. Pivoteau and M. Soria for algorithmic aspects.

The end