Between Analytic Combinatorics and Species Theory

ALÉA - March 2013

Carine Pivoteau, joint work with Bruno Salvy and Michèle Soria
I Introduction
Algorithms for combinatorial structures: Well-founded systems and Newton iterations

Carine Pivoteau, Bruno Salvy, Michèle Soria

1 Université Paris-Est, LIGM (CNRS UMR 8049), Marne-la-Vallée, France
2 Inria, France
3 Université Pierre et Marie Curie, LIP6 (CNRS UMR 7606), Paris, France

Keywords: Species theory, Newton iteration, Analytic combinatorics, Generating functions, Complexity

ABSTRACT

We consider systems of recursively defined combinatorial structures. We give algorithms checking that these systems are well-founded, computing generating series and providing numerical values. Our framework is an articulation of the constructible classes of Flajolet and Sedgewick with Joyal's species theory. We extend the implicit species theorem to structures of size zero. A quadratic iterative Newton method is shown to solve well-founded systems combinatorially. From there, truncations of the corresponding generating series are obtained in quasi-optimal complexity. This iteration transfers to a numerical scheme that converges unconditionally to the values of the generating series inside their disk of convergence. These results provide important subroutines in random generation. Finally, the approach is extended to combinatorial differential systems.

Algorithms for analytic combinatorics:
- Enumeration in Quasi-Optimal Complexity
- Exponential and Ordinary Oracle
- Differential Systems

Well-defined input provided by species theory.

Bonus: Unified framework for constructible combinatorial classes.
Definition 1.7. A **specification** for an $r$–tuple $A = (A^{(1)}, \ldots, A^{(r)})$ of classes is a collection of $r$ equations,

$$
\begin{align*}
A^{(1)} &= \Phi_1(A^{(1)}, \ldots, A^{(r)}) \\
A^{(2)} &= \Phi_2(A^{(1)}, \ldots, A^{(r)}) \\
\vdots &= \vdots \\
A^{(r)} &= \Phi_r(A^{(1)}, \ldots, A^{(r)})
\end{align*}
$$

where each $\Phi_i$ denotes a term built from the $A$ using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes $E$ (neutral) and $Z$ (atomic).
Symbolic Method: combinatorial specifications

\[ y = \mathcal{Z} + \mathcal{Z} y \]

**Definition 1.7.** A **specification** for an \( r \)-tuple \( \mathcal{A} = (\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)}) \) of classes is a collection of \( r \) equations,

\[
\begin{align*}
\mathcal{A}^{(1)} &= \Phi_1(\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)}) \\
\mathcal{A}^{(2)} &= \Phi_2(\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)}) \\
&\vdots \\
\mathcal{A}^{(r)} &= \Phi_r(\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)})
\end{align*}
\]

where each \( \Phi_i \) denotes a term built from the \( \mathcal{A} \) using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes \( \mathcal{E} \) (neutral) and \( \mathcal{Z} \) (atomic).
Definition 1.7. A specification for an r–tuple \( A = (A^{(1)}, \ldots, A^{(r)}) \) of classes is a collection of \( r \) equations,

\[
\begin{align*}
A^{(1)} &= \Phi_1(A^{(1)}, \ldots, A^{(r)}) \\
A^{(2)} &= \Phi_2(A^{(1)}, \ldots, A^{(r)}) \\
& \vdots \\
A^{(r)} &= \Phi_r(A^{(1)}, \ldots, A^{(r)})
\end{align*}
\]

where each \( \Phi_i \) denotes a term built from the \( A \) using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes \( E \) (neutral) and \( Z \) (atomic).
Symbolic Method: combinatorial specifications

Definition 1.7. A specification for an r–tuple \( A = (A^{(1)}, \ldots, A^{(r)}) \) of classes is a collection of \( r \) equations,

\[
\begin{align*}
A^{(1)} &= \Phi_1(A^{(1)}, \ldots, A^{(r)}) \\
A^{(2)} &= \Phi_2(A^{(1)}, \ldots, A^{(r)}) \\
\vdots &= \vdots \\
A^{(r)} &= \Phi_r(A^{(1)}, \ldots, A^{(r)})
\end{align*}
\]

where each \( \Phi_i \) denotes a term built from the \( A \) using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes \( E \) (neutral) and \( Z \) (atomic).
Symbolic Method: combinatorial specifications

\[ y = \mathcal{Z} + \mathcal{Z} y \]

\[ y = \mathcal{Z} y \]

\[ y = \mathcal{Z} + y \]

\[ y_1 = \mathcal{Z} + y_2 \]

\[ y_2 = \mathcal{Z} y_1 \text{ SEQ}(y_2) \]

**Definition 1.7.** A specification for an \( r \)-tuple \( \mathcal{A} = (\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)}) \) of classes is a collection of \( r \) equations,

\[
\begin{align*}
\mathcal{A}^{(1)} &= \Phi_1(\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)}) \\
\mathcal{A}^{(2)} &= \Phi_2(\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)}) \\
\vdots &= \vdots \\
\mathcal{A}^{(r)} &= \Phi_r(\mathcal{A}^{(1)}, \ldots, \mathcal{A}^{(r)})
\end{align*}
\]

where each \( \Phi_i \) denotes a term built from the \( \mathcal{A} \) using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes \( \mathcal{E} \) (neutral) and \( \mathcal{Z} \) (atomic).
Symbolic Method: combinatorial specifications

Definition 1.7. A specification for an $r$-tuple $A = (A^{(1)}, \ldots, A^{(r)})$ of classes is a collection of $r$ equations,

$$A^{(1)} \quad \Phi_1(A^{(1)}, \ldots, A^{(r)})$$
$$A^{(2)} \quad \Phi_2(A^{(1)}, \ldots, A^{(r)})$$
$$\vdots$$
$$A^{(r)} \quad \Phi_r(A^{(1)}, \ldots, A^{(r)})$$

where each $\Phi_i$ denotes a term built from the $A$ using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes $E$ (neutral) and $Z$ (atomic).

$$\gamma = \mathcal{Z} + \mathcal{Z} \gamma$$

$$\gamma = \mathcal{Z} \gamma$$

$$\gamma = \mathcal{Z} + \gamma$$

$$\gamma_1 = \mathcal{Z} + \gamma_2$$

$$\gamma_2 = \mathcal{Z} \gamma_1 \text{SEQ}(\gamma_2)$$

$$\gamma_1 = \mathcal{Z} + \gamma_2^2$$

$$\gamma_2 = \gamma_1$$
Definition 1.7. A specification for an r-tuple \( A = (A^{(1)}, \ldots, A^{(r)}) \) of classes is a collection of r equations,

\[
\begin{align*}
A^{(1)} &= \Phi_1(A^{(1)}, \ldots, A^{(r)}) \\
A^{(2)} &= \Phi_2(A^{(1)}, \ldots, A^{(r)}) \\
&\vdots \\
A^{(r)} &= \Phi_r(A^{(1)}, \ldots, A^{(r)})
\end{align*}
\]

(29)

where each \( \Phi_i \) denotes a term built from the \( A \) using the constructions of disjoint union, cartesian product, sequence, power set, multiset, and cycle, as well as the initial classes \( E \) (neutral) and \( Z \) (atomic).
Definition 1.7. A **specification** for an \( r \)-tuple \( A = (A^{(1)}, \ldots, A^{(r)}) \) of classes is a collection of \( r \) equations,

\[
\begin{align*}
A^{(1)} &= \Phi_1(A^{(1)}, \ldots, A^{(r)}) \\
A^{(2)} &= \Phi_2(A^{(1)}, \ldots, A^{(r)}) \\
& \vdots \\
A^{(r)} &= \Phi_r(A^{(1)}, \ldots, A^{(r)})
\end{align*}
\]

where each \( \Phi_i \) denotes a term built from the \( A \) using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes \( E \) (neutral) and \( Z \) (atomic).

\[ Y = Z + Z Y \]

\[ Y = Z Y \]

\[ Y = Z + Y \]

\[ Y_1 = Z + Y_2 \]

\[ Y_2 = Z Y_1 \text{ SEQ}(Y_2) \]

\[ Y_1 = Z + Y_2^2 \]

\[ Y_2 = Y_1 \]

\[ A = B \]

\[ B = Z + C \]

\[ C = ZC \]

\[ Y_1 = Z + Y_2 \]

\[ Y_2 = Z + Y_1 \text{ SEQ}(Y_2) \]
**Symbolic Method: combinatorial specifications**

**Definition 1.7.** A specification for an $r$-tuple $A = (A^{(1)}, \ldots, A^{(r)})$ of classes is a collection of $r$ equations,

\[
\begin{cases}
A^{(1)} = \Phi_1(A^{(1)}_1, \ldots, A^{(r)}_1) \\
A^{(2)} = \Phi_2(A^{(1)}_2, \ldots, A^{(r)}_2) \\
\vdots \\
A^{(r)} = \Phi_r(A^{(1)}_r, \ldots, A^{(r)}_r)
\end{cases}
\]

where each $\Phi_i$ denotes a term built from the $A$ using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes $E$ (neutral) and $Z$ (atomic).
Definition 1.7. A specification for an $r$-tuple $A = (A^{(1)}, \ldots, A^{(r)})$ of classes is a collection of $r$ equations,

\begin{align*}
A^{(1)} &= \Phi_1(A^{(1)}, \ldots, A^{(r)}) \\
A^{(2)} &= \Phi_2(A^{(1)}, \ldots, A^{(r)}) \\
& \vdots \\
A^{(r)} &= \Phi_r(A^{(1)}, \ldots, A^{(r)})
\end{align*} 

where each $\Phi_i$ denotes a term built from the $A$ using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes $E$ (neutral) and $Z$ (atomic).
Symbolic Method: combinatorial specifications

Definition 1.7. A specification for an \( r \)-tuple \( A = (A^{(1)}, \ldots, A^{(r)}) \) of classes is a collection of \( r \) equations,

\[
\begin{align*}
A^{(1)} &= \Phi_1(A^{(1)}, \ldots, A^{(r)}) \\
A^{(2)} &= \Phi_2(A^{(1)}, \ldots, A^{(r)}) \\
& \vdots \\
A^{(r)} &= \Phi_r(A^{(1)}, \ldots, A^{(r)})
\end{align*}
\]

where each \( \Phi_i \) denotes a term built from the \( A \) using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes \( E \) (neutral) and \( Z \) (atomic).

\[
\mathcal{Y} = Z + Z \mathcal{Y}
\]

\[
\mathcal{Y} = Z \mathcal{Y}
\]

\[
\mathcal{Y} = Z + \mathcal{Y}
\]

\[
\mathcal{Y}_1 = Z + \mathcal{Y}_2
\]

\[
\mathcal{Y}_2 = Z \mathcal{Y}_1 \text{ SEQ}(\mathcal{Y}_2)
\]

\[
\mathcal{Y}_1 = Z + \mathcal{Y}_2^2
\]

\[
\mathcal{Y}_2 = \mathcal{Y}_1
\]

\[
\mathcal{Y}_1 = Z + \mathcal{Y}_2
\]

\[
\mathcal{Y}_2 = Z + \mathcal{Y}_1 \text{ SEQ}(\mathcal{Y}_2)
\]
Symbolic Method: combinatorial specifications

**Definition 1.7.** A specification for an \( r \)-tuple \( A = (A^{(1)}, \ldots, A^{(r)}) \) of classes is a collection of \( r \) equations,

\[
\begin{align*}
A^{(1)} &= \Phi_1(A^{(1)}, \ldots, A^{(r)}) \\
A^{(2)} &= \Phi_2(A^{(1)}, \ldots, A^{(r)}) \\
&\vdots \\
A^{(r)} &= \Phi_r(A^{(1)}, \ldots, A^{(r)})
\end{align*}
\]

where each \( \Phi_i \) denotes a term built from the \( A \) using the constructions of disjoint union, cartesian product, sequence, powerset, multiset, and cycle, as well as the initial classes \( \mathcal{E} \) (neutral) and \( \mathcal{Z} \) (atomic).
Symbolic Method: combinatorial specifications

\[ Y = z + zY \]

> N1 = Cycle(Union(N16,Prod(N16,Prod(Prod(Z,Cycle(N16)),N7))))
N2 = Prod(N8,N8)
N3 = Union(Cycle(Prod(Z,Prod(N2,Prod(Set(Z),Union(N14,Z))))),Sequence(Prod(Z,Cycle(N14))))
N4 = Prod(Set(Prod(Z,Sequence(Prod(N8,Z)))),N8)
N5 = Set(Union(Prod(Z,Sequence(Prod(N11,Sequence(Union(Z,Sequence(N11)))))),Prod(N11,Sequence(N11))))
N6 = Union(Cycle(Prod(N11,N4)),Cycle(Union(N11,Union(N11,Z))))
N7 = Cycle(Prod(N15,Prod(Prod(Sequence(N15),Z),N15)))
N8 = Union(N8,Prod(Sequence(Prod(Prod(Z,Z),Union(N8,N8)),Union(N8,N8))),N8)
N9 = Set(Prod(Prod(Union(N5,Cycle(Z)),Z),Z))
N10 = Prod(Prod(Cycle(Union(Z,Z)),N10),Union(Cycle(N13),Cycle(Prod(Prod(N13,Cycle(N13)),N10))))
N11 = Prod(Union(Set(Prod(Sequence(Z),Prod(N12,N5))),N5),Z)
N12 = Prod(N19,Cycle(Prod(N19,Prod(Z,Prod(Prod(Sequence(Z),Z),N1),Prod(N19,N19))))))
N13 = Prod(Sequence(N17),Union(N17,Union(Prod(N17,Sequence(Prod(Prod(N17,N17),N17))),Z)))
N14 = Prod(Prod(Prod(Z,N13),Z),Cycle(Union(Z,Prod(Z,Cycle(N13)))))
N15 = Prod(Prod(N2,Union(Z,Z)),Z)
N16 = Prod(Prod(Prod(Cycle(Z),Sequence(Prod(Set(Z),Union(N15,Z)))),Z),Set(Prod(Z,N10)))
N17 = Prod(Prod(Sequence(Z),Prod(N9,Union(N17,Z))),Z)
N18 = Union(N17,Prod(Cycle(N20),Z)),Prod(Prod(N20,Union(N20,Union(Union(N20,Z),Z))),Set(N20))
N19 = Union(N11,Prod(Sequence(N11),Sequence(Z))),Prod(N4,Prod(Set(Prod(Z,N11),Z)))
N20 = Prod(Union(N19,Prod(N19,Cycle(N19))),Z)
### Symbolic Method: combinatorial specifications

#### Examples

- \( N_1 = \text{Cycle(Union(N16,Prod(N16,Prod(Prod(Z,Cycle(N16)),N7))))} \)
- \( N_2 = \text{Prod(N8,N8)} \)
- \( N_3 = \text{Union(Cycle(Prod(Z,Prod(N2,Prod(Set(Z,Union(N14,Z)))))),Sequence(Prod(Z,Cycle(N14))))} \)
- \( N_4 = \text{Prod(Set(Prod(Z,Sequence(Sequence(N1)))),N5))} \)
- \( N_5 = \text{Set(Union(Prod(Z,Sequence(Sequence(N1)))),Prod(N11,Sequence(N11))))} \)
- \( N_6 = \text{Union(Cycle(Prod(N11,N4)),Cycle(Prod(Prod(Z,Z),Union(N8,N8)))))} \)
- \( N_7 = \text{Cycle(Prod(N15,Prod(Sequence(N11))))} \)
- \( N_8 = \text{Union(N8,Prod(Sequence(Prod(Prod(Z,Z),Union(N8,N8))),Union(N8,N8))))} \)
- \( N_9 = \text{Set(Prod(Prod(Union(N5,Cycle(Z)),Z)),Z))} \)
- \( N_{10} = \text{Prod(Prod(Cycle(Union(Z,Z)),N10),Union(Cycle(N13),Cycle(Prod(Prod(N13,Cycle(N13)),N10))))} \)
- \( N_{11} = \text{Prod(Union(Set(Prod(Sequence(Z),Prod(N12,N5))),N5),Z))} \)
- \( N_{12} = \text{Prod(N19,Cycle(Prod(N19,Prod(Z,Prod(Prod(Sequence(Z),Z),N1),Prod(N11,N19))))))} \)
- \( N_{13} = \text{Prod(Sequence(N2),Union(N17),Prod(Prod(Prod(Sequence(Z),Z),Prod(N17,Z))),Z))} \)
- \( N_{14} = \text{Prod(Prod(Prod(Z,Z),Z),Cycle(Union(Z),Prod(Cycle(N13,Z))))} \)
- \( N_{15} = \text{Prod(N19,Union(Z,Z))} \)
- \( N_{16} = \text{Prod(Prod(Prod(Cycle(Z),Sequence(Prod(Set(Z,Union(N15,Z)))),Z),Z),Set(Prod(Z,N10)))} \)
- \( N_{17} = \text{Prod(Prod(Sequence(Z),Prod(N9,Union(N17,Z))),Z))} \)
- \( N_{18} = \text{Union(Union(Z,Prod(Cycle(N20),Z)),Prod(Prod(N20,Union(N20,Union(Union(N20,Z),Z))),Set(N20)))} \)
- \( N_{19} = \text{Union(Prod(N11,Prod(Sequence(N11),Sequence(Z))),Prod(N4,Prod(Set(Prod(Z,N11)),Z)))} \)
- \( N_{20} = \text{Prod(Union(N19,Prod(N19,Cycle(N19))),Z)} \)

\[ Y = Z + Z Y \]

![Well-founded.png](https://example.com/Well-founded.png)
II Elements of Species Theory
The key point

**Theorem (Implicit Species Theorem - Joyal 81)**

Let $\mathcal{H}$ be a vector of *multisort species*, such that

- $\mathcal{H}(0, 0) = 0$ and
- the Jacobian matrix $\partial \mathcal{H}/\partial \mathcal{Y}(0, 0)$ is nilpotent.

The system of equations

$$\mathcal{Y} = \mathcal{H}( \mathcal{E}, \mathcal{Y})$$

admits a vector $\mathcal{S}$ of *species solution* such that $\mathcal{S}(0) = 0$, which is unique up to isomorphism.

- $\mathcal{H}(0, 0) = 0$ ?
- What about the other condition?
- Is it a well founded system?
Definition (Species $\mathcal{F}$)

A species of structures $\mathcal{F}$ is a rule that,

- $\forall$ finite set $U$ produces a finite set $\mathcal{F}[U]$, and
- $\forall$ bij. $\sigma : U \to V$ produces a bij. $\mathcal{F}[\sigma] : \mathcal{F}[U] \to \mathcal{F}[V]$,

with $\mathcal{F}[\text{Id}_U] = \text{Id}_{\mathcal{F}[U]}$ and $\mathcal{F}[\tau \circ \sigma] = \mathcal{F}[\tau] \circ \mathcal{F}[\sigma]$ for any bijections $\sigma$ and $\tau$.

$\mathcal{F}[U]$-structure : $\leadsto$ Species $\mathcal{F}$:

$U = \{ \circ, \bullet, \bullet, \bullet, \bullet \}$

labeled! labeled and unlabelled…
Exemples

- the **empty species** 0, defined by $0[U] = \emptyset$, $\forall U$.
- the species 1 (the empty set) defined by
  
  $$1[U] = \emptyset \text{ if } U \neq \emptyset \text{ and } 1[\emptyset] = \{\emptyset\}.$$ 
- the species $Z$ of **singletons**, defined by
  
  $$Z[U] = \{U\} \text{ if } |U| = 1 \text{ and } Z[U] = \emptyset \text{ otherwise.}$$
- The species $\text{SET}$ of **sets**, defined by $\text{SET}[U] = \{U\}$.
- The species $\text{SEQ}$ of **sequences** (or linear orders) defined by
  
  $$\text{SEQ}[\emptyset] = \{\emptyset\} \text{ and for } U = \{u_1, \ldots, u_n\} \neq \emptyset,$$
  $$\text{SEQ}[U] = \{(u_{\sigma(1)}, \ldots, u_{\sigma(n)}) | \sigma \in P_n\}.$$  
- The species $\text{CYC}$ of **cycles**, defined by $\text{CYC}[\emptyset] = \emptyset$ and for $U \neq \emptyset$,
  
  $$\text{CYC}[U] = \{\sigma | \sigma \in P[U] \text{ is composed of a unique cycle}\}.$$ 

In all cases, the transport $\mathcal{F}[\sigma]$ is (obvious) carried out in the natural fashion...
Sequences and Permutations

- The species $\text{SEQ}$ of **sequences** is defined by $\text{SEQ}[\emptyset] = \{\emptyset\}$ and for $U = \{u_1, \ldots, u_n\} \neq \emptyset$, $\text{SEQ}[U] = \{(u_{\sigma(1)}, \ldots, u_{\sigma(n)}) \mid \sigma \in \mathcal{P}_n\}$.
- The species $\mathcal{P}$ of **permutations** is defined by
  \[
  \mathcal{P}[U] = \{\psi : U \to U \mid \forall v \in U, \exists! u \in U, \psi(u) = v\}.
  \]

Let $U = \{1, 2, 3\}$, $V = \{1, 2, 3\}$ and $\sigma(1) = 2$, $\sigma(2) = 3$, $\sigma(3) = 1$,

- $\text{SEQ}[U] = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$
- $\mathcal{P}[U] = \begin{cases} 
1 \to 1 & 1 \to 1 & 1 \to 2 & 1 \to 2 & 1 \to 3 & 1 \to 3 \\
2 \to 2 & 2 \to 3 & 2 \to 1 & 2 \to 3 & 2 \to 1 & 2 \to 2 \\
3 \to 3 & 3 \to 2 & 3 \to 3 & 3 \to 1 & 3 \to 2 & 3 \to 1 
\end{cases}$
Sequences and Permutations

- The species $\text{SEQ}$ of sequences is defined by $\text{SEQ}[^\emptyset] = \{\emptyset\}$ and for $U = \{u_1, \ldots, u_n\} \neq \emptyset$, $\text{SEQ}[U] = \{(u_{\sigma(1)}, \ldots, u_{\sigma(n)}) \mid \sigma \in P_n\}$.

- The species $P$ of permutations is defined by

$$P[U] = \{\psi : U \rightarrow U \mid \forall v \in U, \exists! u \in U, \psi(u) = v\}.$$ 

Let $U = \{1, 2, 3\}$, $V = \{1, 2, 3\}$ and $\sigma(1) = 2$, $\sigma(2) = 3$, $\sigma(3) = 1$,

- $\text{SEQ}[U] = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$

- $P[U] = \begin{cases}
1 \rightarrow 1 & 1 \rightarrow 1 & 1 \rightarrow 2 & 1 \rightarrow 2 & 1 \rightarrow 3 & 1 \rightarrow 3 \\
2 \rightarrow 2 & 2 \rightarrow 3 & 2 \rightarrow 1 & 2 \rightarrow 3 & 2 \rightarrow 1 & 2 \rightarrow 2 \\
3 \rightarrow 3 & 3 \rightarrow 2 & 3 \rightarrow 3 & 3 \rightarrow 1 & 3 \rightarrow 2 & 3 \rightarrow 1
\end{cases}$

- $\text{SEQ}[\sigma] = \{(1, 2, 3) \rightarrow (2, 3, 1), (1, 3, 2) \rightarrow (2, 1, 3), \ldots\}$

- $P[\sigma] = \begin{cases}
1 \rightarrow 1 & 2 \rightarrow 2 & 1 \rightarrow 1 & 2 \rightarrow 2 \\
2 \rightarrow 2 & 3 \rightarrow 3 & 2 \rightarrow 3 & 3 \rightarrow 1 & \ldots\ \\
3 \rightarrow 3 & 1 \rightarrow 1 & 3 \rightarrow 2 & 1 \rightarrow 3
\end{cases}$
Operations on Species

- The **Sum** of species:


- The **Product** of two species \(F\) and \(G\):

  \[(F \cdot G)[U] = \sum (U_1, U_2), U = U_1 + U_2 \quad F[U_1] \times G[U_2],\]

- The **Composition** of \(F\) with \(G\) (also denoted by \(F(G)\)):

  \[(F \circ G)[U] = \sum_{\pi \text{ partition of } U} F[\pi] \times \prod_{p \in \pi} G[p].\]

  with \(G[\emptyset] = \emptyset\)
Example: the species $\mathcal{F} = \text{Set} (\text{Cyc}(\mathcal{Z}))$

For $U = \{1, 2, 3\}$, 

\[
\text{SEQ(CYC)}[U] = \left\{ \begin{array}{c}
\begin{array}{c}
1 \rightarrow 2 \rightarrow 3 \\
1 \rightarrow 3 \rightarrow 2 \\
3 \rightarrow 1 \rightarrow 2 \\
3 \rightarrow 2 \rightarrow 1
\end{array}
\end{array} \right\}
\]
Example: the species $\mathcal{F} = \text{Set} (\text{Cyc}(\mathcal{Z}))$

For $U = \{1, 2, 3\}$, $\text{SEQ}(\text{CYC})[U] =$

\[
\left\{ \begin{array}{c}
\begin{array}{c}
1 \quad 2 \\
2 \quad 3 \\
3 \quad 1
\end{array}, \\
\begin{array}{c}
1 \quad 3 \\
3 \quad 2 \\
2 \quad 1
\end{array}, \\
\begin{array}{c}
1 \quad 2 \\
2 \quad 3 \\
3 \quad 1
\end{array}
\end{array} \right\}
\]

What about the species of Permutations?
Example: the species $\mathcal{F} = \text{Set}(\text{Cyc}(\mathcal{Z}))$

For $U = \{1, 2, 3\}$, $\text{SEQ(CYC)}[U] = \{ \}$

- Two species are equal if they produce the same sets and bijections.

- An isomorphism from $\mathcal{F}$ to $\mathcal{G}$ is a family of bijections $\alpha_U : \mathcal{F}[U] \to \mathcal{G}[U]$ such that:

  ![Diagram](image)

  $F[U] \xrightarrow{\alpha_U} G[U]$  $F[\sigma] \downarrow$  $G[\sigma]$

  $F[V] \xrightarrow{\alpha_V} G[V]$

  ![Diagram](image)

  $F[\sigma]$  $\downarrow$  $G[\sigma]$

- $\cong$ denotes a combinatorial equality between isomorphic species.
Combinatorial Systems

\[ Y = \mathcal{H}(Z, Y) : \]

\[
\begin{aligned}
Y_1 & = \mathcal{H}_1(Z, Y_1, Y_2, \ldots, Y_m) \\
Y_2 & = \mathcal{H}_2(Z, Y_1, Y_2, \ldots, Y_m) \\
& \vdots \\
Y_m & = \mathcal{H}_m(Z, Y_1, Y_2, \ldots, Y_m)
\end{aligned}
\]

multisort species: \( \forall \) finite sets \( U_1, \ldots, U_k \), produces a set \( \mathcal{F}[U_1, \ldots, U_k] \)
Combinatorial Systems

\[ \mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}) : \]

\[
\begin{align*}
\mathcal{Y}_1 &= \mathcal{H}_1(\mathcal{Z}, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_m) \\
\mathcal{Y}_2 &= \mathcal{H}_2(\mathcal{Z}, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_m) \\
& \quad \vdots \\
\mathcal{Y}_m &= \mathcal{H}_m(\mathcal{Z}, \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_m)
\end{align*}
\]

Solution :
\[ \mathcal{H} \text{-rooted trees.} \]

multisort species: \( \forall \) finite sets \( U_1, \ldots, U_k \), produces a set \( F[U_1, \ldots, U_k] \)
**Definition**

Exponential generating series of $\mathcal{F}$: 
$$F(z) = \sum_{n=0}^{\infty} f_n \frac{z^n}{n!}$$

Ordinary generating series of $\mathcal{F}$: 
$$\tilde{F}(z) = \sum_{n=0}^{\infty} \tilde{f}_n z^n.$$

**Definition**

Cycle index series of $\mathcal{F}$:

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

where $\sigma_i$ is the number of cycles of length $i$ in the cycle decomposition of the permutation $\sigma$ and $\text{fix} \mathcal{F}[\sigma]$ is the number of $\mathcal{F}$-structures on $\{1, \ldots, n\}$ fixed by $\mathcal{F}[\sigma]$.

$$F(z) = Z_{\mathcal{F}}(z, 0, 0, \ldots)$$ and $$\tilde{F}(z) = Z_{\mathcal{F}}(z, z^2, z^3, \ldots).$$
Definition (Derivative)

The derivative $\mathcal{F}'$ of a species $\mathcal{F}$ is defined by $\mathcal{F}'[U] = \mathcal{F}[U + \{\star\}]$, where $\star$ is an element chosen outside of $U$.

<table>
<thead>
<tr>
<th>species</th>
<th>derivative</th>
<th>species</th>
<th>derivative</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A + B$</td>
<td>$A' + B'$</td>
<td>$\text{SEQ}(B)$</td>
<td>$\text{SEQ}(B) \cdot B' \cdot \text{SEQ}(B)$</td>
</tr>
<tr>
<td>$A \cdot B$</td>
<td>$A' \cdot B + A \cdot B'$</td>
<td>$\text{CYC}(B)$</td>
<td>$\text{SEQ}(B) \cdot B'$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\text{SET}(B)$</td>
<td>$\text{SET}(B) \cdot B'$</td>
</tr>
</tbody>
</table>
Jacobian matrix of $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}) : \partial \mathcal{H}/\partial \mathcal{Y}$

Example:

$\mathcal{H}(\mathcal{G}, \mathcal{S}, \mathcal{P}) := (\mathcal{S} + \mathcal{P}, \text{SEQ}(\mathcal{Z} + \mathcal{P}), \text{SET}(\mathcal{Z} + \mathcal{S}))$.

\[
\frac{\partial \mathcal{H}}{\partial \mathcal{Y}} = \begin{pmatrix}
0 & 1 & 1 \\
0 & 0 & \text{SEQ}(\mathcal{Z} + \mathcal{P}) \cdot 1 \\
0 & \text{SET}(\mathcal{Z} + \mathcal{S}) \cdot 1 & 0
\end{pmatrix}
\]

Not nilpotent at $(0, 0)$: successive powers

\[
\begin{pmatrix}
0 & 1 & 1 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix}, \quad \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix}, \quad \begin{pmatrix}
0 & 1 & 1 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{pmatrix}.
\]

Rmk: \( (\partial \mathcal{H}/\partial \mathcal{Y})^k \bigg|_{(i,j)} = \frac{\partial \mathcal{H}_i}{\partial \mathcal{Y}_1} \frac{\partial \mathcal{H}_{\ell_1}}{\partial \mathcal{Y}_2} \cdots \frac{\partial \mathcal{H}_{\ell_k}}{\partial \mathcal{Y}_j} \)
Jacobian matrix of $Y = \mathcal{H}(Z, Y) : \frac{\partial \mathcal{H}}{\partial Y}$

Example:

$$\mathcal{H}(G, S, P) := (S + P, Z \cdot \text{SEQ}(Z + P), \text{SET}(Z + S)).$$

$$\frac{\partial \mathcal{H}}{\partial Y} = \begin{pmatrix}
0 & 1 & 1 \\
0 & 0 & \text{SEQ}(Z + P) \cdot Z \cdot \text{SEQ}(Z + P) \\
0 & \text{SET}(Z + S) \cdot 1 & 0
\end{pmatrix}$$

Nilpotent at $\mathbf{0}$: successive powers

$$\begin{pmatrix}
0 & 1 & 1 \\
0 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix}, \quad \begin{pmatrix}
0 & 1 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}, \quad \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}.$$

Rmk: $(\frac{\partial \mathcal{H}}{\partial Y})^k|_{(i, j)} = \frac{\partial \mathcal{H}_i}{\partial Y_{\ell_1}} \frac{\partial \mathcal{H}_{\ell_1}}{\partial Y_{\ell_2}} \cdots \frac{\partial \mathcal{H}_{\ell_k}}{\partial Y_j}$
III Well-founded combinatorial systems
Back to Joyal’s Implicit Species Theorem

**Theorem (IST)**

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

$$
\mathcal{Y}[0] = 0, \quad \mathcal{Y}[n+1] = \mathcal{H}(\mathcal{Z}, \mathcal{Y}[n]) \quad (n \geq 0).
$$

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

**Key Lemma**

If $\mathcal{Y}[n+1] =_k \mathcal{Y}[n]$, then $\mathcal{Y}[n+p+1] =_{k+1} \mathcal{Y}[n+p]$, ($p = \text{dimension}$).
Joyal’s Implicit Species Theorem is too restrictive:

- We don’t want the condition $\mathcal{H}(0,0) = 0$.
- To allow equations such as $\mathcal{Y} = 1 + \mathcal{Z}\mathcal{Y}$.
- We want to characterize precisely which are the systems that define combinatorial structures: well-founded systems.

**Bonus:**

A better understanding of the role played by the Jacobian matrix and a better knowledge of the structure of combinatorial systems.
**Theorem (General Implicit Species Theorem)**

Let $\mathcal{H} = (\mathcal{H}_{1:m})$ be any vector of species, such that the system $Y = \mathcal{H}(Z,Y)$ is well-founded. This system admits a solution $S$ such that $S(0) = \mathcal{H}^m(0,0)$, which is unique up to isomorphism.

**Definition (Well-founded combinatorial system)**

$Y = \mathcal{H}(Z,Y)$ is said to be well-founded when the iteration

\[ Y[0] = 0 \quad \text{and} \quad Y[n+1] = \mathcal{H}(Z,Y[n]), \quad n \geq 0 \quad (\Phi) \]

- is well-defined,
- defines a convergent sequence
- and the limit $S$ of this sequence has no zero coordinate.
Examples

Joyal’s conditions:

\[
\begin{align*}
Y &= \text{SEQ}(\mathbb{Z}) \checkmark \\
\mathcal{H}'(0) &= 0 \\
Y &= \text{SEQ}(\mathbb{Z} \text{ SEQ}(\mathbb{Z})) \checkmark \\
\mathcal{H}'(0) &= 0 \\
Y &= \text{SEQ}(\text{SEQ}(\mathbb{Z})) \times \\
\mathcal{H}'(0) &\text{ not defined!} \\
Y &= \mathbb{Z} \checkmark \\
\mathcal{H}'(0, 0) &= 0 \\
Y &= \mathbb{Z} + \mathbb{Z} \checkmark \\
\mathcal{H}'(0, 0) &= 0 \\
Y &= \mathbb{Z} + Y \times \\
\mathcal{H}'(0, 0) &= 1
\end{align*}
\]
Examples

Joyal’s conditions:

\[ \mathcal{Y} = \text{SEQ}(\mathcal{Z}) \]  \(\checkmark\)  \[ \mathcal{Y} = \text{SEQ}(\mathcal{Z} \text{SEQ}(\mathcal{Z})) \]  \(\checkmark\)  \[ \mathcal{Y} = \text{SEQ}(\text{SEQ}(\mathcal{Z})) \]  \(\times\)

\[ \mathcal{H}'(0) = 0 \]  \[ \mathcal{H}'(0) = 0 \]  \[ \mathcal{H}'(0) \text{ not defined!} \]

\[ \mathcal{Y} = \mathcal{Z} \mathcal{Y} \]  \(\checkmark\)  \[ \mathcal{Y} = \mathcal{Z} + \mathcal{Z} \mathcal{Y} \]  \(\checkmark\)  \[ \mathcal{Y} = \mathcal{Z} + \mathcal{Y} \]  \(\times\)

\[ \mathcal{H}'(0, 0) = 0 \]  \[ \mathcal{H}'(0, 0) = 0 \]  \[ \mathcal{H}'(0, 0) = 1 \]

With our conditions:

\[ \mathcal{Y} = \mathcal{Z} \mathcal{Y} \]  \(\times\) because \( \mathcal{Y} = 0 \).

How to detect 0 coordinates:

Look for 0 in \( \mathcal{H}^m(\mathcal{Z}, 0) \).
Examples

Joyal’s conditions:

\[ \mathcal{Y} = \text{SEQ}(\mathcal{Z}) \quad \checkmark \quad \mathcal{Y} = \text{SEQ}(\mathcal{Z} \ \text{SEQ}(\mathcal{Z})) \quad \checkmark \quad \mathcal{Y} = \text{SEQ}(\text{SEQ}(\mathcal{Z})) \quad \times \]

\[ \mathcal{H}'(0) = 0 \quad \mathcal{H}'(0) = 0 \]

\[ \mathcal{Y} = \mathcal{Z} \ \mathcal{Y} \quad \checkmark \quad \mathcal{Y} = \mathcal{Z} + \mathcal{Z} \ \mathcal{Y} \quad \checkmark \]

\[ \mathcal{H}'(0, 0) = 0 \quad \mathcal{H}'(0, 0) = 0 \quad \mathcal{H}'(0, 0) \text{ not defined!} \quad \mathcal{H}'(0, 0) = 1 \]

With our conditions:

\[ \mathcal{Y} = \mathcal{Z} \ \mathcal{Y} \quad \times \quad \text{because } \mathcal{Y} = 0. \]

How to detect 0 coordinates:

Look for 0 in \( \mathcal{H}^m(\mathcal{Z}, 0) \).

Examples:

\[
\begin{align*}
A &= B \\
B &= C \\
C &= \mathcal{Z}
\end{align*}
\quad \begin{align*}
A &= B \\
B &= \mathcal{Z} + C \\
C &= \mathcal{Z} \ C
\end{align*}
\]
Well founded at 0

Algorithm 0-coord: Detection of zero coordinates in the solution of a system

Input: $\mathcal{H} = (\mathcal{H}_1:m)$: a vector of species such that $\mathcal{H}(0,0) = 0$ and the Jacobian matrix $\partial \mathcal{H} / \partial \mathcal{Y}(0,0)$ is nilpotent.

Output: Answer to “Are there 0 coordinates in the solution of the system $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$?”

begin
  Compute $\mathcal{U} := \mathcal{H}^m(\mathcal{Z}, 0)$
  foreach coordinate $C$ of $\mathcal{U}$ do
    if $C = 0$ then return YES
  end
  return NO
end

Algorithm isWellFoundedAt0: Characterization of well-founded systems at 0

Input: $\mathcal{H} = (\mathcal{H}_1:m)$: a vector of species such that $\mathcal{H}(0,0) = 0$.

Output: Answer to “Is the system $\mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y})$ well-founded at 0?”

begin
  Compute $\mathcal{J} := \partial \mathcal{H} / \partial \mathcal{Y}(0,0)$
  if $\mathcal{J}^m = 0$ then return 0-coord($\mathcal{H}$) else return NO
end
### Examples

\[
\begin{align*}
\mathcal{Y}_1 &= \mathcal{Z} \mathcal{Y}_2 \\
\mathcal{Y}_2 &= \mathcal{Z} \mathcal{V}_1 \mathsf{SEQ}(\mathcal{V}_2) \quad \checkmark
\end{align*}
\]

\[
\begin{pmatrix}
0 & 0 \\
\mathcal{Z} \mathsf{SEQ}(\mathcal{V}_2) & \mathcal{Z} \mathcal{V}_1 \mathsf{SEQ}(\mathcal{V}_2)^2
\end{pmatrix}
\bigg|_{0,0} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}
\]

\[
\begin{align*}
\mathcal{Y}_1 &= \mathcal{Z} + \mathcal{Y}_2 \\
\mathcal{Y}_2 &= \mathcal{Z} \mathcal{V}_1 \mathsf{SEQ}(\mathcal{V}_2) \quad \checkmark
\end{align*}
\]

\[
\begin{pmatrix}
0 & 1 \\
0 & 0
\end{pmatrix}
\]

\[
\begin{align*}
\mathcal{Y}_1 &= \mathcal{Z} + \mathcal{Y}_2 \\
\mathcal{Y}_2 &= \mathcal{Z} + \mathcal{V}_1 \mathsf{SEQ}(\mathcal{V}_2) \quad \times
\end{align*}
\]

\[
\begin{pmatrix}
0 & 1 \\
\mathsf{SEQ}(\mathcal{V}_2) & \mathcal{V}_1 \mathsf{SEQ}(\mathcal{V}_2)^2
\end{pmatrix}
\bigg|_{0,0} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}
\]

\[
\begin{align*}
\mathcal{Y}_1 &= \mathcal{Z} + \mathcal{Y}_2^2 \\
\mathcal{Y}_2 &= \mathcal{Y}_1 \quad \checkmark
\end{align*}
\]

\[
\begin{pmatrix}
0 & 0 \\
1 & 0
\end{pmatrix}
\]
Algorithmic Characterization

**Definition (Companion system of \( Y = H(\mathcal{Z}, Y) \))**

The species \( 1 \) is replaced by the sort \( \mathcal{Z}_1 \):

\[
Y = K(\mathcal{Z}_1, \mathcal{Z}, Y), \quad \text{where} \quad K = H(\mathcal{Z}, Y) - H(0, 0) + \mathcal{Z}_1 H(0, 0).
\]

**Theorem (Characterization of well-founded systems)**

The system \( Y = H(\mathcal{Z}, Y) \) is well-founded if and only if

1. the companion system \( Y = K(\mathcal{Z}_1, \mathcal{Z}, Y) \) is well-founded at \( 0 \) (Joyal's conditions without zeroes)
2. the solution \( S_1(\mathcal{Z}_1, \mathcal{Z}) \) of the companion system is polynomial in \( \mathcal{Z}_1 \).

In this case, the limit of the iteration (\( \Phi \)) is \( S_1(1, \mathcal{Z}) \).
**Definition**

\( F(\mathcal{Z}_1, \mathcal{Z}_2) \) is polynomial in the sorts \( \mathcal{Z}_1 \) when, for all \( n \geq 0 \), the species \( F_{=(,n)} = \sum_{k \geq 0} F_{=(k,n)} \) is polynomial.

(The species is polynomial when the size is fixed in the other sorts.)

**Examples:**

- **SEQ(\( \mathcal{Z}_1 + \mathcal{Z}_2 \)):** not polynomial in \( \mathcal{Z}_1 \) or \( \mathcal{Z}_2 \)
- **SEQ(\( \mathcal{Z}_1 \cdot \mathcal{Z}_2 \)):** polynomial in \( \mathcal{Z}_1 \) and \( \mathcal{Z}_2 \) (but not in \( \mathcal{Z} \))
- **\( \mathcal{Z}_1 \)SEQ(\( \mathcal{Z}_2 \)):** polynomial in \( \mathcal{Z}_1 \) and not in \( \mathcal{Z}_2 \).

**Well-founded Systems?**

\[
\begin{aligned}
Y_1 &= Z + Y_1 Y_2 \\
Y_2 &= 1
\end{aligned}
\]

\[
\begin{aligned}
Y_1 &= Z + Y_2 Y_1^2 \\
Y_2 &= 1
\end{aligned}
\]
Definition

\[ F(Z_1, Z_2) \text{ is polynomial in the sorts } Z_1 \text{ when, for all } n \geq 0, \text{ the species } F_{(.,n)} = \sum_{k \geq 0} F_{(k,n)} \text{ is polynomial.} \]

(The species is polynomial when the size is fixed in the other sorts.)

Examples:

- \( \text{SEQ}(Z_1 + Z_2) \): not polynomial in \( Z_1 \) or \( Z_2 \)
- \( \text{SEQ}(Z_1 \cdot Z_2) \): polynomial in \( Z_1 \) and \( Z_2 \) (but not in \( Z \))
- \( Z_1 \text{SEQ}(Z_2) \): polynomial in \( Z_1 \) and not in \( Z_2 \).

Well-founded Systems?

\[
\begin{cases}
Y_1 = Z + Y_1 Y_2 \\
Y_2 = Z_1
\end{cases}
\quad \times \quad 
\begin{cases}
Y_1 = Z + Y_2 Y_1^2 \\
Y_2 = Z_1
\end{cases}
\]

Between Analytic Combinatorics and Species Theory
Information given by the Jacobian Matrix

Role of the Jacobian Matrix (dependency graph of the system):

1. Well-founded systems at 0: nilpotence of $\partial H/\partial Y(0, 0)$

2. Implicit polynomial species: nilpotence of $\partial H/\partial Y(\mathcal{Z}, Y)$
   (detection of cycles in the graph)

3. Implicit partially polynomial species:
   nilpotence of $\partial H/\partial Y(\mathcal{Z}_1, 0, \mathcal{S}(\mathcal{Z}_1, 0))$
   (+ conditions on $\mathcal{H}$ and $\mathcal{S}(\mathcal{Z}_1, 0)$)

\[
\begin{align*}
A &= \mathcal{Z} \times \mathcal{B} + \mathcal{C} \\
B &= \text{SET}(A) \\
C &= \mathcal{Z} + \text{SEQ}(\mathcal{B})
\end{align*}
\]

\[
\begin{pmatrix}
0 & \mathcal{Z} & 1 \\
\text{SET}(A) & 0 & 0 \\
0 & \text{SEQ}(\mathcal{B})^2 & 0
\end{pmatrix}
\begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix}
\]
Role of the Jacobian Matrix (dependency graph of the system):

1. Well-founded systems at 0: nilpotence of $\frac{\partial H}{\partial Y}(0,0)$

2. Implicit polynomial species: nilpotence of $\frac{\partial H}{\partial Y}(Z,Y)$
   (detection of cycles in the graph)

3. Implicit partially polynomial species:
   nilpotence of $\frac{\partial H}{\partial Y}(Z_1,0,S(Z_1,0))$
   (+ conditions on $H$ and $S(Z_1,0)$)

$A = Z \times B + C$

$B = \text{SET}(A)$

$C = Z + B^2$

\[
\begin{pmatrix}
0 & Z & 1 \\
\text{SET}(A) & 0 & 0 \\
0 & 2B & 0
\end{pmatrix}
\]

\[
\begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}
\]
Role of the Jacobian Matrix (dependency graph of the system):

1. Well-founded systems at $0$: nilpotence of $\partial H / \partial Y(0, 0)$
2. Implicit polynomial species: nilpotence of $\partial H / \partial Y(Z, Y)$ (detection of cycles in the graph)
3. Implicit partially polynomial species:
   nilpotence of $\partial H / \partial Y(Z_1, 0, S(Z_1, 0))$  
   (+ conditions on $H$ and $S(Z_1, 0)$)
4. Well-founded systems: both 1 and 3.
5. The key for Newton iteration.

But no information on the 0 coord. (depend on initial conditions).
Analytic Combinatorics:

- **Symbolic method** to describe recursive combinatorial classes
- A **restricted set of combinatorial constructions**, with a **dictionary** for gfs.
- **Powerful tools for enumeration** (singularity analysis,...)
- **Automatic random sampling** (Recursive, Boltzmann)

But no information on well-founded systems...
From Analytic Combinatorics to Species Theory

Analytic Combinatorics:

- Symbolic method to describe recursive combinatorial classes
- A restricted set of combinatorial constructions, with a dictionary for gfs.
- Powerful tools for enumeration (singularity analysis,...)
- Automatic random sampling (Recursive, Boltzmann)

But no information on well-founded systems...

Species Theory:

- A more general framework for combinatorial structures
- Implicit species theorem
  ▶ Well-founded systems
- Combinatorial Derivative Labelle
- Newton iteration (Decoste, Labelle, Leroux)
- Differential systems

But no analytic tools...
IV Newton iteration
Theorem (essentially Labelle)

For any well-founded system \( \mathcal{Y} = \mathcal{H}(\mathcal{Z}, \mathcal{Y}) \), if \( \mathcal{A} \) has contact \( k \) with its solution \( \mathcal{S} \) and \( \mathcal{A} \subset \mathcal{H}(\mathcal{Z}, \mathcal{A}) \), then

\[
\mathcal{A} + \sum_{i \geq 0} \left( \frac{\partial \mathcal{H}}{\partial \mathcal{Y}}(\mathcal{Z}, \mathcal{A}) \right)^i \cdot (\mathcal{H}(\mathcal{Z}, \mathcal{A}) - \mathcal{A})
\]

has contact \( 2k + 1 \) with \( \mathcal{S} \).
**Theorem (essentially Labelle)**

For any well-founded system, the sequence $\mathbf{y}^{[0]} = \mathbf{0}$ and

$$
\mathbf{y}^{[n+1]} = \mathbf{y}^{[n]} + \left( \sum_{k \geq 0} \left( \frac{\partial \mathbf{H}}{\partial \mathbf{y}} (\mathbf{z}, \mathbf{y}^{[n]}) \right)^k \right) \cdot (\mathbf{H} (\mathbf{z}, \mathbf{y}^{[n]}) - \mathbf{y}^{[n]})
$$

is well defined and converges quadratically to the solution $\mathcal{S}$ of the system.

Rmk : Generation by increasing Strahler numbers.
Newton Iteration for Binary Trees

\[ Y_{n+1} = Y_n + \text{SEQ}(Z \times Y_n \times \star + Z \times \star \times Y_n) \times (1 + Z \times Y_n^2 - Y_n). \]
Newton Iteration for Binary Trees

\[ Y_{n+1} = Y_n + \text{SEQ}(Z \times Y_n \times \ast + Z \times \ast \times Y_n) \times (1 + Z \times Y_n^2 - Y_n). \]

\[ Y_0 = 0 \quad Y_1 = \circ \]

\[ Y_2 = \]

\[ Y_3 = Y_2 + \]

[Décosté, Labelle, Leroux 1982]