The Mockingbird lattice

Samuele Giraudo

LaCIM, Université du Québec à Montréal

16th Workshop — Computational Logic and Applications

École Polytechnique
France

January 13, 2023

Outline

1. Combinatory logic

2. Mockingbird lattices

3. Enumerative properties

Outline

1. Combinatory logic

Applicative terms

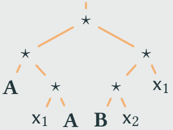
Let $\mathfrak{G}$ be a set, called alphabet.

A $\mathfrak{G}$-term is either

- a variable x_i from the set $\mathbb{X}_n := \{\mathsf{x}_1, \dots, \mathsf{x}_n\}$ for an $n \geqslant 0$;
- a basic combinator $\mathbf{X}$ where $\mathbf{X} \in \mathfrak{G}$;
- a pair $(\mathfrak{t}_1, \mathfrak{t}_2)$ where $\mathfrak{t}_1$ and $\mathfrak{t}_2$ are $\mathfrak{G}$-terms, denoted by $\mathfrak{t}_1 \star \mathfrak{t}_2$.

Let $\mathfrak{T}(\mathfrak{G}) := \bigsqcup_{n \geqslant 0} \mathfrak{T}(\mathfrak{G})(n)$ where $\mathfrak{T}(\mathfrak{G})(n)$ is the set of the $\mathfrak{G}$-terms having all variables in $\mathbb{X}_n$.

– Example –



The tree of the left is the **tree representation** of the $\mathfrak{G}$-term

$$(\mathbf{A} \star (\mathsf{x}_1 \star \mathbf{A})) \star (((\mathbf{B} \star \mathsf{x}_2) \star \mathsf{x}_1))$$

where $\mathfrak{G} := \{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$.
The **short representation** is obtained by considering that $\star$ associates to the left and by removing the superfluous parentheses:

$$\mathbf{A}(\mathsf{x}_1\, \mathbf{A})(\mathbf{B}\, \mathsf{x}_2 \mathsf{x}_1).$$

Combinatory logic systems

A combinatory logic system (CLS) is a pair $(\mathfrak{G}, \to)$ where $\to$ is a binary relation on $\mathfrak{T}(\mathfrak{G})$ such that for each $\mathbf{X} \in \mathfrak{G}$, there is $n \geqslant 1$ and $\mathfrak{t}_{\mathbf{X}} \in \mathfrak{T}(\emptyset)(n)$ such that

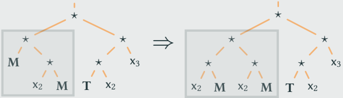
$$\mathbf{X}\, x_1 \dots x_n \to \mathfrak{t}_{\mathbf{X}}.$$

The context closure of $\to$ is the binary relation $\Rightarrow$ on $\mathfrak{T}(\mathfrak{G})$ such that $\mathfrak{t} \Rightarrow \mathfrak{t}'$ if $\mathfrak{t}'$ can be obtained from $\mathfrak{t}$ by replacing a pattern $\mathbf{X}\, x_1 \dots x_n$ by $\mathfrak{t}_{\mathbf{X}}$.

– Example –

Let the CLS $(\mathfrak{G}, \to)$ such that $\mathfrak{G} := \{\mathbf{M}, \mathbf{T}\}$ where $\mathbf{M} x_1 \to x_1 x_1$ and $\mathbf{T}\, x_1 x_2 \to x_2 x_1$. We have

$$\Big(\underline{\mathbf{M}(x_2\mathbf{M})}\Big)(\mathbf{T}\, x_2 x_3) \Rightarrow \Big(\underline{(x_2\mathbf{M})(x_2\mathbf{M})}\Big)(\mathbf{T}\, x_2 x_3)$$



$$(\mathbf{M}(x_2\mathbf{M}))\big(\underline{\mathbf{T}\, x_2 x_3}\big) \Rightarrow (\mathbf{M}(x_2\mathbf{M}))\big(\underline{x_3 x_2}\big)$$

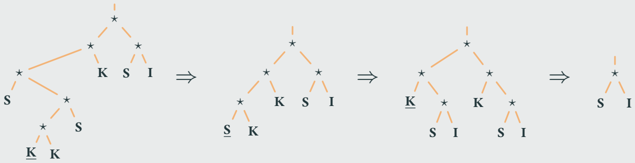
The S, K, I-system

Let the system [**Curry**, 1930] made on the three basic combinators **S**, **K**, and **I**, satisfying

$$\mathbf{S}\,x_1x_2x_3 \to x_1x_3(x_2x_3), \quad \mathbf{K}\,x_1x_2 \to x_1, \quad \mathbf{I}\,x_1 \to x_1.$$

– Example –

Here is a sequence of computation:



This CLS is Turing-complete: there are algorithms to emulate any λ-term by a term of this CLS. These algorithms are called **abstraction algorithms** [**Rosser**, 1955], [**Curry**, **Feys**, 1958].

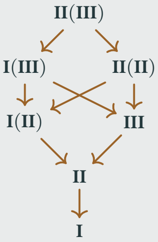
Rewrite graphs

Given a CLS $\mathcal{C} := (\mathfrak{G}, \rightarrow)$, let

- $\preccurlyeq$ be the **reflexive and transitive closure** of $\Rightarrow$;
- $\equiv$ be the **reflexive, symmetric, and transitive closure** of $\Rightarrow$;
- for any $\mathfrak{t} \in \mathfrak{T}(\mathfrak{G})$, let $\mathfrak{t}^* := \{\mathfrak{t}' \in \mathfrak{T}(\mathfrak{G}) : \mathfrak{t} \preccurlyeq \mathfrak{t}'\}$. The graph $(\mathfrak{t}^*, \Rightarrow)$ is the rewrite graph of $\mathfrak{t}$.

– Example –

Let the CLS $(\mathfrak{G}, \rightarrow)$ such that $\mathfrak{G} := \{\mathbf{I}\}$ and $\mathbf{I}\mathfrak{x}_1 \rightarrow \mathfrak{x}_1$.



This is the rewrite graph of $\mathbf{II}(\mathbf{III})$.

We have $\mathbf{I}(\mathbf{III}) \preccurlyeq \mathbf{I}$ and $\mathbf{I}(\mathbf{III}) \not\preccurlyeq \mathbf{II}(\mathbf{II})$.

It is possible to prove that for any $\mathfrak{t}, \mathfrak{t}' \in \mathfrak{T}(\mathfrak{G})$, $\mathfrak{t} \equiv \mathfrak{t}'$.

The Enchanted Forest of combinator birds

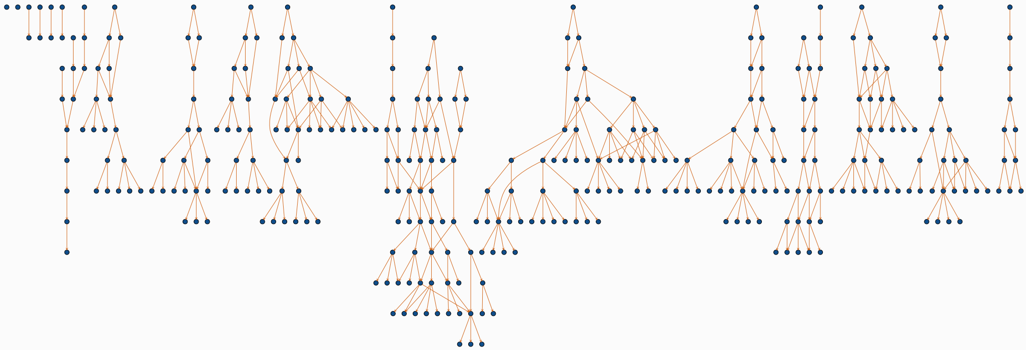
In *To Mock a Mockingbird: and Other Logic Puzzles* [Smullyan, 1985], a great number of basic combinators with their rules are listed, forming the **Enchanted forest of combinator birds**.

Here is a sublist:

- **Identity bird**: $\mathbf{I}\, x_1 \to x_1$
- **Mockingbird**: $\mathbf{M}\, x_1 \to x_1 x_1$
- **Kestrel**: $\mathbf{K}\, x_1 x_2 \to x_1$
- **Thrush**: $\mathbf{T}\, x_1 x_2 \to x_2 x_1$
- **Mockingbird 1**: $\mathbf{M_1}\, x_1 x_2 \to x_1 x_1 x_2$
- **Warbler**: $\mathbf{W}\, x_1 x_2 \to x_1 x_2 x_2$
- **Lark**: $\mathbf{L}\, x_1 x_2 \to x_1 (x_2 x_2)$
- **Owl**: $\mathbf{O}\, x_1 x_2 \to x_2 (x_1 x_2)$
- **Turing bird**: $\mathbf{U}\, x_1 x_2 \to x_2 (x_1 x_1 x_2)$
- **Cardinal**: $\mathbf{C}\, x_1 x_2 x_3 \to x_1 x_3 x_2$
- **Vireo**: $\mathbf{V}\, x_1 x_2 x_3 \to x_3 x_1 x_2$
- **Bluebird**: $\mathbf{B}\, x_1 x_2 x_3 \to x_1 (x_2 x_3)$
- **Starling**: $\mathbf{S}\, x_1 x_2 x_3 \to x_1 x_3 (x_2 x_3)$
- **Jay**: $\mathbf{J}\, x_1 x_2 x_3 x_4 \to x_1 x_2 (x_1 x_4 x_3)$

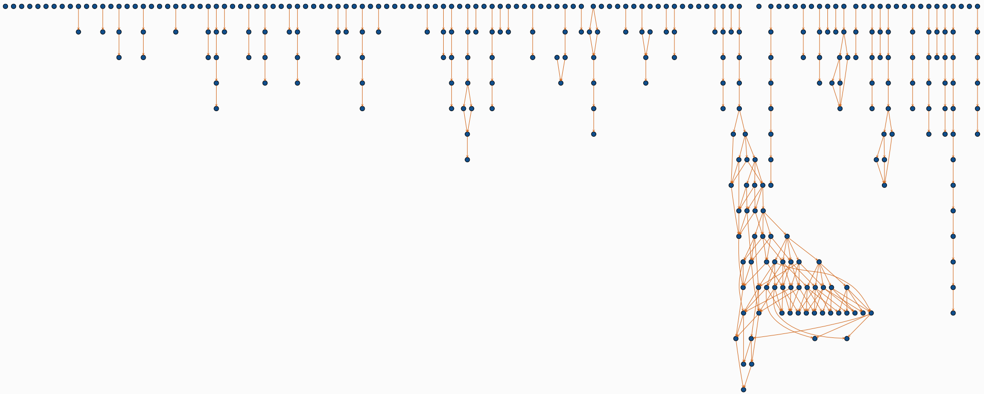
Rewrite graph of L

In the CLS containing only the **Lark L**, the rewrite graphs of closed terms of degrees up to 5 and up to 4 rewrite steps have the shape



Rewrite graph of S

In the CLS containing only the **Starling S**, the rewrite graphs of closed terms of degrees up to 6 and up to 11 rewrite steps have the shape



Usual questions

Let $\mathcal{C}$ be a CLS.

– Word problem –

Is there an algorithm to decide, given two terms t and t' of $\mathcal{C}$, if $t \equiv t'$?
See [**Baader, Nipkow**, 1998], [**Statman**, 2000].

- Yes for the CLS on $\mathbf{L}$ [**Statman**, 1989], [**Sprenger, Wymann-Böni**, 1993].
- Yes for the CLS on $\mathbf{W}$ [**Sprenger, Wymann-Böni**, 1993].
- Yes for the CLS on $\mathbf{M_1}$ [**Sprenger, Wymann-Böni**, 1993].
- Open for the CLS on $\mathbf{S}$ [**RTA Problem #97**, 1975].

– Strong normalization problem –

Is there an algorithm to decide, given a term t of $\mathcal{C}$, if all rewrite sequences from t are finite?

- Yes for the CLS on $\mathbf{S}$ [**Waldmann**, 2000].
- Yes for the CLS on $\mathbf{J}$ [**Probst, Studer**, 2000].

Order theoretical questions

A lattice is a partial order (poset) wherein each pair $\{x, x'\}$ of elements has a greatest lower bound $x \wedge x'$ and a least upper bound $x \vee x'$.

Let $\mathcal{C}$ be a CLS. A $\mathfrak{G}$-term $\mathfrak{t}$ has

1. the poset property if $(\mathfrak{t}^*, \preccurlyeq)$ is a **poset**;
2. the lattice property if $(\mathfrak{t}^*, \preccurlyeq)$, is a **lattice**.

This CLS has the poset (resp. lattice) property if all terms of $\mathcal{C}$ have the poset (resp. lattice) property.

– Poset and lattice properties –

Is there an algorithm to decide, given a term $\mathfrak{t}$ of $\mathcal{C}$, if $\mathfrak{t}$ has the poset (resp. lattice) property?

Given a term $\mathfrak{t}$ of $\mathcal{C}$, perform a **combinatorial study** of $(\mathfrak{t}^*, \preccurlyeq)$ as the enumeration of its elements and intervals.

– A new source of posets –

Use combinatory logic as a source to **build original posets**.

Outline

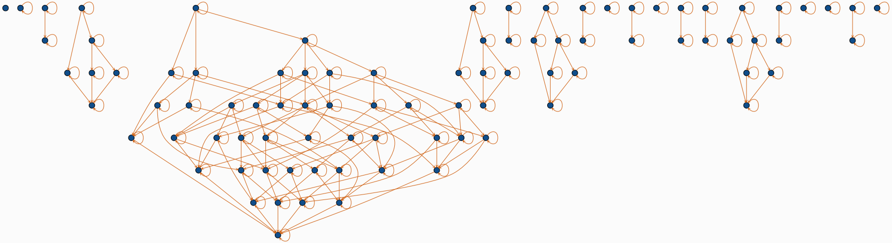
2. Mockingbird lattices

The Mockingbird system

The Mockingbird system is the CLS $\mathcal{C}$ containing only the Mockingbird $\mathbf{M}$.

Recall that $\mathbf{M}$ satisfies $\mathbf{M}\mathsf{x}_1 \to \mathsf{x}_1\mathsf{x}_1$.

The **rewrite graphs** of closed terms of $\mathcal{C}$ of degrees up to 4 have the shape



– Proposition [G., 2022] –

The CLS $\mathcal{C}$ has the **poset property** and each $\equiv$-equivalence class of $\mathcal{C}$ is **finite** and contains a **greatest** and a **least** element.

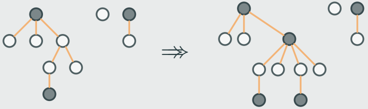
Duplicative forests

A duplicative forest is a **forest of planar rooted trees** where nodes are either black ● or white ○.

Let $\mathcal{D}^{\mathrm{F}}$ be the set of the duplicative forests and $\mathcal{D}^{\mathrm{T}}$ be the set of the duplicative trees.

Let $\Rrightarrow$ be the binary relation on $\mathcal{D}^{\mathrm{F}}$ such that for any $\mathfrak{f}, \mathfrak{f}' \in \mathcal{D}^{\mathrm{F}}$, we have $\mathfrak{f} \Rrightarrow \mathfrak{f}'$ if $\mathfrak{f}'$ is obtained by blackening a white node of $\mathfrak{f}$ and then by **duplicating** its sequence of descendants.

– Example –



The reflexive and transitive closure $\ll$ of $\Rrightarrow$ is a **partial order relation**.

For any $\mathfrak{f} \in \mathcal{D}^{\mathrm{F}}$, let $\mathfrak{f}^* := \left\{\mathfrak{f}' \in \mathcal{D}^{\mathrm{F}} : \mathfrak{f} \ll \mathfrak{f}'\right\}$.

Lattice of duplicative forests

Let $\wedge$ and $\vee$ be the two binary, commutative, and associative partial operations on $\mathcal{D}^{\mathrm{F}}$ defined recursively, for any $\ell \geqslant 0$, $\mathfrak{f}_1, \dots, \mathfrak{f}_\ell \in \mathcal{D}^{\mathrm{T}}$, $\mathfrak{f}'_1, \dots, \mathfrak{f}'_\ell \in \mathcal{D}^{\mathrm{T}}$, and $\mathfrak{f}, \mathfrak{f}', \mathfrak{f}'' \in \mathcal{D}^{\mathrm{F}}$, by

$$\mathfrak{f}_1 \dots \mathfrak{f}_\ell \wedge \mathfrak{f}'_1 \dots \mathfrak{f}'_\ell := (\mathfrak{f}_1 \wedge \mathfrak{f}'_1) \dots (\mathfrak{f}_\ell \wedge \mathfrak{f}'_\ell),$$

$$\circ(\mathfrak{f}) \wedge \circ(\mathfrak{f}') := \circ(\mathfrak{f} \wedge \mathfrak{f}'), \qquad \bullet(\mathfrak{f}) \wedge \bullet(\mathfrak{f}') := \bullet(\mathfrak{f} \wedge \mathfrak{f}'),$$

$$\circ(\mathfrak{f}) \wedge \bullet(\mathfrak{f}'\mathfrak{f}'') := \circ(\mathfrak{f} \wedge \mathfrak{f}' \wedge \mathfrak{f}''),$$

$$\mathfrak{f}_1 \dots \mathfrak{f}_\ell \vee \mathfrak{f}'_1 \dots \mathfrak{f}'_\ell := (\mathfrak{f}_1 \vee \mathfrak{f}'_1) \dots (\mathfrak{f}_\ell \vee \mathfrak{f}'_\ell),$$

$$\circ(\mathfrak{f}) \vee \circ(\mathfrak{f}') := \circ(\mathfrak{f} \vee \mathfrak{f}'), \qquad \bullet(\mathfrak{f}) \vee \bullet(\mathfrak{f}') := \bullet(\mathfrak{f} \vee \mathfrak{f}'),$$

$$\circ(\mathfrak{f}) \vee \bullet(\mathfrak{f}'\mathfrak{f}'') := \bullet((\mathfrak{f} \vee \mathfrak{f}')(\mathfrak{f} \vee \mathfrak{f}'')).$$

– Proposition [G., 2022] **–**

Given a duplicative forest $\mathfrak{f}$, the poset $(\mathfrak{f}^*, \ll)$ is a **lattice** for the operations $\wedge$ and $\vee$.

This can be proved by structural induction on duplicative forests.

From Mockingbird terms to duplicative trees

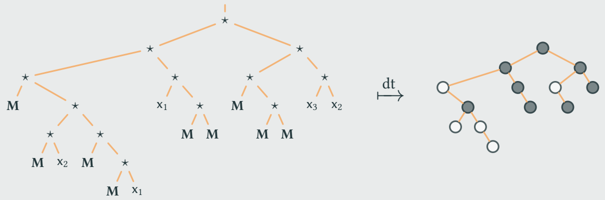
Let $\mathrm{dt} : \mathfrak{T}(\mathfrak{G}) \to \mathcal{D}^{\mathrm{T}}$ be the map defined recursively, for any $x_i \in \mathbb{X}$ and $\mathfrak{t}, \mathfrak{t}' \in \mathfrak{T}(\mathfrak{G})$, by

$$\mathrm{dt}(x_i) := \epsilon,$$

$$\mathrm{dt}(\mathbf{M}) := \epsilon,$$

$$\mathrm{dt}(\mathfrak{t} \star \mathfrak{t}') := \begin{cases} \circ(\mathrm{dt}(\mathfrak{t}')) & \text{if } \mathfrak{t} = \mathbf{M} \text{ and } \mathfrak{t}' \neq \mathbf{M}, \\ \bullet(\mathrm{dt}(\mathfrak{t})\ \mathrm{dt}(\mathfrak{t}')) & \text{otherwise.} \end{cases}$$

– Example –



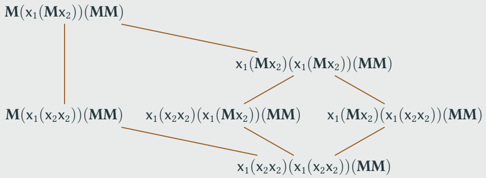
Poset isomorphism

– Proposition [G., 2022] –

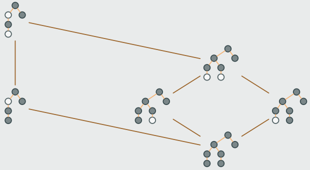
For any $\mathfrak{t} \in \mathfrak{T}(\mathfrak{G})$, the **posets** $(\mathfrak{t}^*, \preccurlyeq)$ and $(\mathrm{dt}(\mathfrak{t})^*, \lll)$ are **isomorphic** and dt is such an isomorphism.

– Example –

Let $\mathfrak{t} := \mathbf{M}(x_1(\mathbf{M}x_2))(\mathbf{MM})$.



The Hasse diagram of the poset $(\mathfrak{t}^*, \preccurlyeq)$.



The Hasse diagram of the poset $(\mathrm{dt}(\mathfrak{t})^*, \lll)$.

– Theorem [G., 2022] –

For any $\mathfrak{t} \in \mathfrak{T}(\mathfrak{G})$, the poset $(\mathfrak{t}^*, \preccurlyeq)$ is a **finite lattice**.

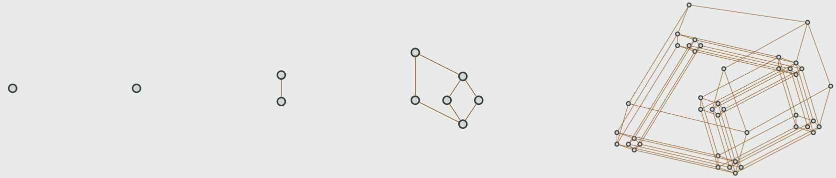
Mockingbird lattices

For any $h \geqslant 0$, the *h*-right comb tree is the $\mathfrak{G}$-term $\mathfrak{r}_h$ satisfying

$$\mathfrak{r}_h = \begin{cases} \mathbf{M} & \text{if } h = 0, \\ \mathbf{M}\,\mathfrak{r}_{h-1} & \text{otherwise.} \end{cases}$$

The Mockingbird lattice of order h is the lattice $\mathcal{M}(h) := (\mathfrak{r}_h^*, \preccurlyeq)$.

– Examples –



$\mathcal{M}(0)$ $\mathcal{M}(1)$ $\mathcal{M}(2)$ $\mathcal{M}(3)$ $\mathcal{M}(4)$

For any $h \geqslant 0$, the *h*-ladder is the duplicative tree $\mathfrak{l}_h$ satisfying

$$\mathfrak{l}_h = \begin{cases} \epsilon & \text{if } h = 0, \\ \circ(\mathfrak{l}_{h-1}) & \text{otherwise.} \end{cases}$$

When $h \geqslant 1$, the lattice $\mathcal{M}(h)$ is isomorphic to $\left(\mathfrak{l}_{h-1}^*, \ll\right)$.

Outline

3. Enumerative properties

Number of elements

Let $\boxtimes$ be the **Hadamard product** of generating series. It satisfies, for two generating series $A_1 = \sum_{h \in \mathbb{N}} a_1(h) z^h$ and $A_2 = \sum_{h \in \mathbb{N}} a_2(n) z^h$,

$$\left(\sum_{h \in \mathbb{N}} a_1(h) z^h \right) \boxtimes \left(\sum_{h \in \mathbb{N}} a_2(h) z^h \right) = \sum_{h \in \mathbb{N}} a_1(h) a_2(h) z^h.$$

– Proposition [G., 2022] –

The generating series $A = \sum_{h \in \mathbb{N}} a(h) z^h$ of the **cardinalities** of $\left(\mathfrak{l}_h^*, \ll\right)$, $h \geqslant 0$, satisfies

$$A = 1 + z A + z (A \boxtimes A).$$

The coefficients $a(h)$, $h \geqslant 0$, satisfy $a(0) = 1$ and for any $h \geqslant 1$,

$$a(h) = a(h-1) + a(h-1)^2.$$

The sequence $(a(h))_{h \geqslant 0}$ starts by 1, 2, 6, 42, 1806, 3263442, 10650056950806 (Sequence **A007018**).

Number of intervals

– Proposition [G., 2022] –

The generating series $A = \sum_{h \in \mathbb{N}} a(h) z^h$ of the **numbers of intervals** of $\left(\mathfrak{l}_h^*, \ll\right)$, $h \geqslant 0$ is the series A_1, where for any $k \geqslant 1$, the series A_k satisfies

$$A_k = 1 + z(A_k \boxtimes A_k) + z \sum_{0 \leqslant i \leqslant k} \binom{k}{i} A_{k+i}.$$

The coefficients $a_k(h)$, $h \geqslant 0$, satisfy $a_k(0) = 1$ and for any $h \geqslant 1$,

$$a_k(h) = a_k(h-1)^2 + \sum_{0 \leqslant i \leqslant k} \binom{k}{i} a_{k+i}(h-1).$$

The sequence $\left(a_1(h)\right)_{h \geqslant 0}$ starts by

$$1, 3, 17, 371, 144513, 20932611523, 438176621806663544657.$$

Number of minimal and maximal elements

A term $\mathfrak{t}$ of $\mathcal{C}$ is minimal (resp. maximal if $\mathfrak{t}' \preccurlyeq \mathfrak{t}$ (resp. $\mathfrak{t} \preccurlyeq \mathfrak{t}'$) implies $\mathfrak{t} = \mathfrak{t}'$.

– Proposition [G., 2022] –

The generating series A of the **minimal elements** of $\mathcal{C}$ enumerated w.r.t. their degrees satisfies

$$A = 1 + z + zA^2 - z\left(A\left[z := z^2\right]\right).$$

The first numbers are 1, 1, 2, 4, 12, 34, 108, 344 (Sequence **A343663** — semi-identity binary trees).

– Proposition [G., 2022] –

The generating series A of the **maximal elements** of $\mathcal{C}$ enumerated w.r.t. their degrees satisfies

$$A = 1 + z - zA + zA^2.$$

The first numbers are 1, 1, 1, 2, 4, 9, 21, 51 (Sequence **A001006** — Motzkin numbers).

Conclusion and perspectives

We have studied a very simple CLS, the Mockingbird system, having nevertheless some **rich combinatorics**:

- its rewrite graphs are Hasse diagrams of posets;
- all intervals of these posets are lattices;
- these lattices are not graded, not self-dual, and not semidistributive;
- enumerative data is accessible but nontrivial.

Some **questions** and **projects**:

1. study, under an order theoretic point of view, some other CLS built from some basic combinators of the Enchanted forest of combinator birds;
2. provide necessary and/or sufficient conditions for a CLS to have the poset or the lattice property;
3. realize some well-known posets (like Tamari lattices, Stanley lattices, or Kreweras lattices) as intervals of posets built from specific CLSs.