Internship Report — Mpri M2
Reduction Graphs of I-terms, how hard can it be?

By Codaline Bourotte, supervised by Giulio Manzonetto, IRIF and Noam Zeilberger, Polytechnique

General Context

The study of programming languages often uses abstract mathematical models to represent programs and their computation. One of the most studied models is the λ-*calculus*, which represents programs as λ-*terms* and the evaluation of a λ-term as β-*reduction*. Understanding the behavior of β-reduction is therefore crucial in many aspects, such as compiler optimization, typing rules, or proving the correctness of a program.

One way to study the β-reduction of a λ-term is to study its *reduction graph*. The *reduction graph* of a λ-term is a directed multigraph whose vertices are all the λ-terms it can be reduced to, and we have an edge from M to N if N is a β-reduct of M, denoted $M \to_\beta N$. Reduction graphs have been studied since the 1980s, starting with the work of Klop and Venturini Zilli [Klo80a, Zil84]. This line of work is also linked with work from influential researchers like Levy [Lév76], and has been a little bit continued in the modern era [EKP16]. Reduction graphs are also very interesting, as they lie at the frontier of two major areas of computer science: graph theory and λ-calculus.

Unfortunately, most conjectures that have been solved turned out to be false, and many questions are still open. For example, Klop's plane conjecture was refuted [SH88]. The study of reduction graphs of λ-terms is very difficult, and not many results have been discovered since the concept was introduced [BM22].

Problem Studied

Since characterizing reduction graphs of λ-terms would be unreasonable even for a PhD thesis, we will here focus on a very small fragment of λ-calculus, the I-terms. The I-*terms* are the terms only built using the identity, denoted I, and its applications. Intuitively, they are all the different ways to parenthesize a series of identities, or equivalently, all the ways to parenthesize a series of "no operations". This fragment is usually not studied because of how easy and trivial it seems: it cannot compute anything other than the identity, it is strongly normalizing since any expression containing n identities always terminates in $n-1$ steps, hence it is trivially optimizable for a compiler. How hard can it be?

Even if this may seem easy, I hope to demonstrate to the reader of this document that even in such a restricted fragment, the problems surrounding reduction graphs are difficult, as I-terms suffer from a lot of the hard problems that the general λ-calculus has with respect to its reduction graphs. For example, two distinct terms might have the same reduction graph, and multiple reductions can lead to the same output term. In this internship we will therefore aim to understand the reduction graphs of I-terms, and will aim to answer those questions:

- Can we relate β-reduction over I-terms to other better-known rewrite systems?
- Can we count the number of ways to reduce an I-term to its normal form?
- Under what conditions do two distinct I-terms have the same reduction graph?

Proposed Contributions

During my internship, I managed to prove the following results:

- An equivalence between reducing I-terms and removing leaves from trees, giving us a combinatorial model for I-term reduction. This crucial step is what allows us to build a better intuition on how β-reduction works over I-terms, and to prove all the following results. (See Theorem 1)
- Counting precisely, with a short and easy-to-compute formula, the number of ways to reduce an I-term to its normal form. (See Theorem 2)
- Defining an inductive relation $\leftrightharpoons$ on trees, called *uniformly dyslexic*, that implies having the same reduction graph, giving us an inductive relation over I-terms due to the first point. (See Theorem 3)
- Proving that if two trees have the same reduction graphs, then they have the same shape (See Theorem 4)
- Study a construction of the reduction graph using a labelled version and a quotienting on labels (See Theorem 5)

I will also discuss a categorical point of view, further research and applications of the theorems.

Arguments Supporting Their Validity[2]

The different arguments supporting their validity are:

- Firstly, proofs will be given for every result.
- Secondly, a computer program has been written to find counter-examples or converses of every theorem, and I will provide examples or counter-examples when possible, or mention when one has not yet been found and how many elements have been checked.
- Lastly, my results have been showcased to a few people for feedback. I therefore want to thank Giulio, Noam, Nino, Antonio, Tito, Florent, Line, Milo, Maya, Zoé, Hélia and Clara for taking the time to listen to me.

Summary and Future Work

My internship was very productive, and removed a blockage on a difficult subject, even if in a very simple case. My work will be made public in a conference paper. My advisor and I have not yet decided which conference we should aim to publish this paper in. We can hope that, with this blockage removed, it will allow us to:

- Find a combinatorial model of a bigger fragment of λ-calculus, allowing us to extrapolate the results to more interesting fragments like linear λ-calculus or combinatory calculus.
- Trying to find a inductive equivalence equivalent to the functoriality lemma, giving us a polynomial-time computable equivalence relation $\sim$.
- Prove which graphs can and cannot be reduction graphs of I-terms and λ-terms using it. This question is also central to many open problems on strongly normalizing systems, especially around the maximal size of a reduction sequence to a normal form. Characterizing the possible graphs would allow us to find the maximal ordinal and therefore, perhaps, have an inductive proof of the strong normalization of certain systems like System F.[1]
- Generalize the differents properties to other rewriting systems, even outside of the lambda calculus.

[1] As mentioned by Wells [Wel99], this is not in contradiction with Girard's proof [Gir71] implying that the strong normalization of System F cannot be proven in PA2.

1 Introduction

In this section, I will first define all the important notions, before formally introducing reduction graphs and providing examples, and I will recall the previously known results.

1.1 General definitions

General. I will denote by $\mathbb{N}$ the set of natural numbers, and by $\mathbb{N}^* := \mathbb{N} \setminus \{0\}$ the set of positive integers. Define $[k] := \{1, ..., k\}$. Denote by $\mathcal{P}(X)$ the set of all subsets of X and by $|X|$ the cardinality of X. For A, B two sets, denote by A^B the set of functions from B to A, by $A \Delta B$ their *symmetric difference* defined as $A \Delta B := A \setminus B \cup B \setminus A$ and by $A \sqcup B$ their *disjoint union*. For $x = (x_1, ..., x_n)$ a sequence, define $|x| := n$ as its length, and $\mathrm{Im}(x) = \{x_1, ..., x_n\}$ as its image. A *multiset* M over X is a function $M : X \longrightarrow \mathbb{N}$ representing, for every element $e \in X$, the *number of times* $M(e)$ that e is present in M, also called its *arity* or *multiplicity*. We denote $M(e) \geq 1$ by $e \in M$ when M is a multiset. For M_1, M_2 two multisets over X, define $M_1 \uplus M_2$ as the *multiset union* defined by $(M_1 \uplus M_2)(x) := M_1(x) + M_2(x)$. For M a multiset over X, define $|M| = \sum_{x \in X} M(x)$. Given a subset S of X, the associated multiset is the characteristic function of the set S over X. We denote by $\{|x_1, ..., x_n|\}$ the multiset with elements $x_1, ..., x_n$ with multiplicity.

Graph Theory. In this internship report we will only consider *directed multigraphs*. A *directed multigraph* is a pair (V, E) with V a finite or countable set of *vertices* and E a multiset of *edges* over $V \times V$.

For $S \subseteq V$, we denote by $G[S]$ the *induced subgraph* on S corresponding to G in which we only keep vertices and edges in S. A *walk* in a graph is a sequence of vertices $(x_1, ..., x_n)$ such that for all $i < n$, we have $(x_i, x_{i+1}) \in E$. A *path* is a walk $(x_1, ..., x_n)$ such that $\forall i \neq j, x_i \neq x_j$. A *circuit* is a walk $(x_1, ..., x_n)$ such that $x_1 = x_n$ and a *cycle* is a circuit such that $\forall i, j < n, i \neq j \Rightarrow x_i \neq x_j$. For $u \in V$ a vertex, we define the multisets of *ingoing* (resp. *outgoing*) *neighbours* by $N^+(u) = \{|\ (x, y) \in E \mid x = u|\}$ (resp. $N^-(u) = \{|\ (x, y) \in E \mid y = u|\}$), and define the multiset of neighbours $N(u) := N^+(u) \uplus N^-(u)$. We then define the *outgoing degree* of a vertex $u \in V$ as $\deg^+(u) = |N^+(u)|$, the *ingoing degree* as $\deg^-(u) = |N^-(u)|$ and the *degree* as $\deg(u) = |N(u)|$. Denote by $\equiv$ the *multigraph isomorphism*, defined by $G_1 \equiv G_2$ with $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ if there exists a $\varphi : V_1 \longleftrightarrow V_2$ bijective such that

$$\forall x, y \in V_1, \qquad E_1(x, y) = E_2(\varphi(x), \varphi(y)). \tag{1}$$

λ-calculus. Let $\mathcal{V}$ be a countably infinite set of variables. We define the set of λ-*terms* Λ as the set of terms generated by the following grammar

$$M, N ::= \lambda x.M \mid MN \mid x \qquad (\forall x \in \mathcal{V}) \tag{2}$$

in which the application MN is left-associative. We identify terms up to α-*renaming*, meaning that bound variables can be renamed: for example, $\lambda x.\lambda y.x$ will be considered the same term as $\lambda y.\lambda z.y$. We define $I := \lambda x.x$ the *identity*. The *substitution of N for x in M* is the term $M[x := N]$ in which every free occurrence of the variable x is replaced by N, making sure that free variables of N are not captured by binders of M.

We define a *context* as an element of the grammar $C ::= \langle\rangle \mid \lambda x.C \mid CM \mid MC$, with $\langle\rangle$ representing the *hole*. For C a context and $M \in \Lambda$, define $C\langle M \rangle$ as the term C in which we replaced the hole with M. A *redex* of a term M is a subterm of M of the shape $(\lambda x.N)N'$, and we define the β-*reduction* between two terms (denoted $\to_\beta$) by

$$C\langle(\lambda x.N)N'\rangle \to_\beta C\langle N[x := N']\rangle \tag{3}$$

Notice that a term M can have multiple redexes, and contracting different redexes can lead to the same output term. For example, $I(II)$ has two reductions leading to the same result : one with $I((\lambda x.x)I) \to_\beta II$ and one with $(\lambda x.x)(II) \to_\beta II$. If two different redexes evaluate to the same term we say that it is a *syntactic accident*. We therefore define the *arity* of $M \to_\beta N$ as the number of different redexes that give the same output: $\to_\beta$ can be seen as a multiset over $\Lambda \times \Lambda$.

We define $\to_\beta^*$ as the transitive reflexive closure of $\to_\beta$: for $M, M' \in \Lambda$ we have $M \to_\beta^* M'$ if and only if there exists $M_1, ..., M_n$ such that $M = M_1$, $M' = M_n$ and $\forall i < n, M_i \to_\beta M_{i+1}$. We say that a term $M \in \Lambda$ is in *normal form* if there is no $N \in \Lambda$ such that $M \to_\beta N$. The Church-Rosser property [CR36] states

that $\to_\beta$ is confluent, meaning that if $M_1 \overset{*}{_\beta\!\leftarrow} M \to_\beta^* M_2$, then there exists a term $N \in \Lambda$ such that $M_1 \to_\beta^* N \overset{*}{_\beta\!\leftarrow} M_2$. This implies that, for $M \in \Lambda$, the normal form reachable from a term, if it exists, is unique.

***I*-calculus[2].** We define the subset $\Lambda_I \subseteq \Lambda$ of *I-terms* by the following grammar (recall that $I = \lambda x.x$)

$$M, N ::= I \mid MN \tag{4}$$

Denote by $|M|_I$ the number of occurrences of I in M. Remark that all redexes are of the form $C\langle IM\rangle \to_\beta C\langle M\rangle$ for $M \in \Lambda_I$, and therefore if $M \in \Lambda_I$ and $M \to_\beta N$, then $N \in \Lambda_I$ (we say that Λ_I is *stable under reduction*). Remark also that if $M \neq I$, then M have at least one redex. Moreover, since each β-reduction removes exactly one occurrence of I, every I-term normalizes in exacly $|M|_I - 1$ steps to the only normal form being I.

For example, $II(II)I = ((\underline{II})(\underline{II}))I \in \Lambda_I$ has two redexes that I underlined. Evaluating the left-most one results in the term $\underline{I(\underline{II})}I$ that possesses itself 2 redexes (underlined) and evaluating the second one results in the term $\underline{II}II$ that has only one redex. While $I(II)I$ has two redexes, they both lead to the same term, namely III : whence the edge $I(II)I \to_\beta III$ has arity two. Intuitively, this gives us a *"reduction graph"*, see Figure 1. We will now define the reduction graph following this logic.

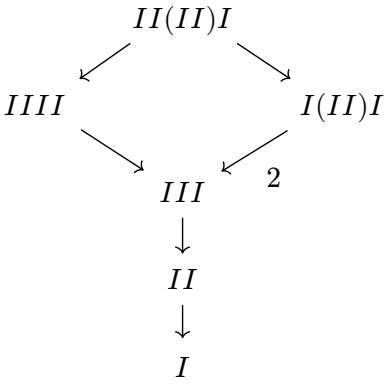

Figure 1: The reduction graph of $II(II)I$.
The number corresponds to the arity of the edge.

1.2 Reduction Graph

I will here introduce reduction graphs as defined by Venturini Zilli [Zil84] and provide some examples.

Definition 1. For a term $M \in \Lambda$, we define its *reduction graph* $\mathrm{RED}(M)$ as the directed multigraph $(V_M, \to)$, where $V_M := \{N \in \Lambda \mid M \to_\beta^* N\}$ are all the terms that are *accessible* from M and where $\to$ is the restriction of $\to_\beta$ over $V_M \times V_M$.

Figure 2: The reduction graph of $II(II)$ (left) and $I(II)I$ (right).
Since they have the same graph, $\mathrm{RED}(II(II)) \equiv \mathrm{RED}(I(II)I)$

Notice that while the definition is given for any $M \in \Lambda$, since Λ_I is stable under reduction, for a $M \in \Lambda_I$, the graph $\mathrm{RED}(M)$ only contains vertices in Λ_I.

Properties. The reduction graph of M always has M as its initial vertex (a vertex is *initial* if it has ingoing degree 0) and has the normal form of M as its final vertex (a vertex is *final* if it has outgoing

[2]This is different from the λI-calculus defined in the literature, as λI-calculus is a fragment without erasure. [BM22]

degree 0). For I-terms, this normal form always exists and is always I. Since a reduction always decreases the number of occurrences of I by 1, the graph $\mathrm{RED}(M)$ for $M \in \Lambda_I$ can be *stratified*. This means that we can partition the graph into *layers* (i.e. the number of occurrences of I) such that every edge (x, y) lies between the $(i+1)$-th layer and the i-th layer. A layer is represented as the height of a vertex in the figure depicting a reduction graph.

Examples 1. Here we will provide simple examples of graphs that are reduction graphs of specific lambda terms. See Figure 3 for a representation of their reduction graphs.

- Consider the term $\Omega := (\lambda x.xx)(\lambda x.xx)$. This term can only be reduced to itself and thus doesn't have a normal form: contracting the only redex leads from Ω to $\Omega \to_\beta \Omega$. This means that its reduction graph $\mathrm{RED}(\Omega)$ is a single vertex with a loop. Note that this is not an I-term.
- For $n \in \mathbb{N}$, consider the term $P_n := \underline{II}\cdots I$ repeated $n+1$ times. Notice that P_n for $n > 0$ only has a single redex (the one underlined), with $P_n \to_\beta P_{n-1}$. Therefore $\mathrm{RED}(P_n)$ is just a line of length n with all edges of multiplicity 1.
- For $n \in \mathbb{N}$, consider the term $E_n := I(I(I\cdots(II)))$ with $n+1$ occurrences of I. Now, for $n > 0$, we have that E_n have n redexes, and every one will reduce $E_n \to_\beta E_{n-1}$. Therefore $\mathrm{RED}(E_n)$ is a line of length n with edges of respective multiplicity $n, n-1, ..., 1$.

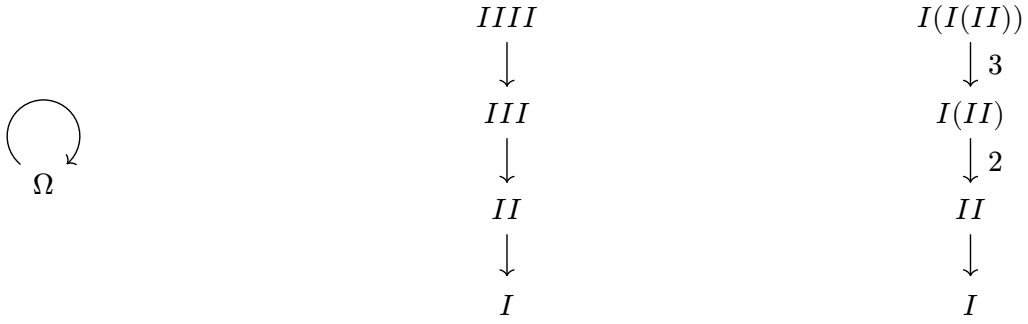

Figure 3: From left to right, the reduction graphs of Ω, P_3 and E_3

Notice that $\Omega \in \Lambda \setminus \Lambda_I$ while $\forall n \in \mathbb{N}^*, P_n, E_n \in \Lambda_I$, with $P_0 = I = E_0$ and $P_1 = II = E_1$.

But despite what those very simple examples show us, the reduction graphs of an identity-term can become very quickly a dense and complicated graph. See Figure 4 for two examples of a reduction graph of two slightly bigger terms, generated by a program I made in Rust [Bou26].

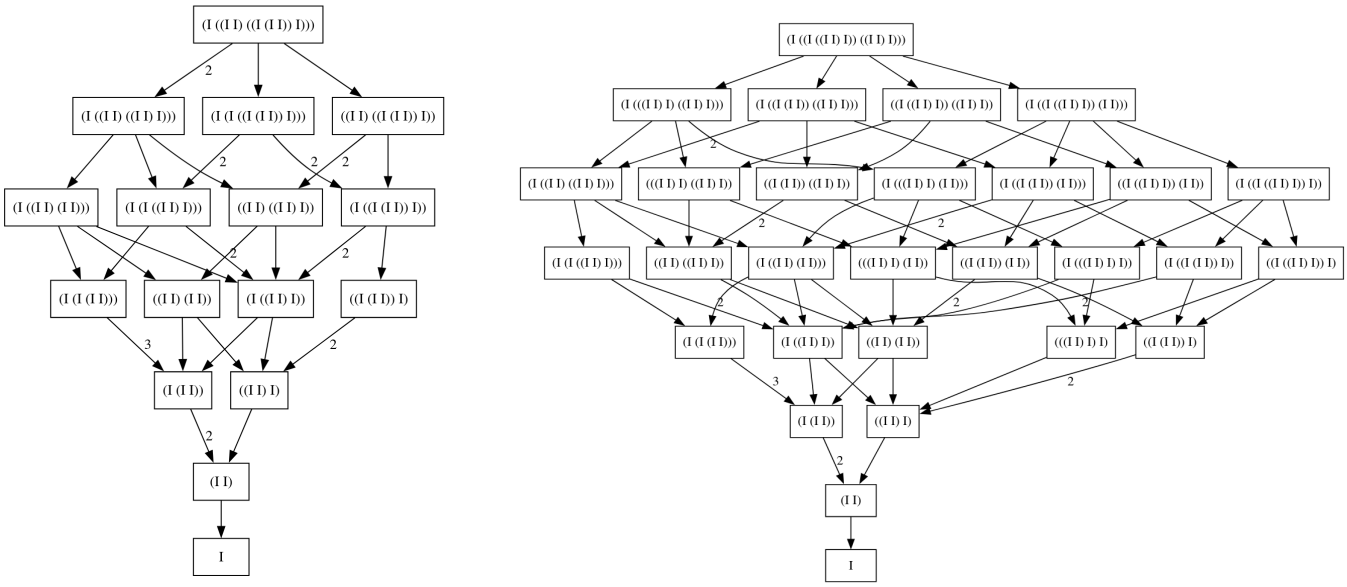

Figure 4: Left, the reduction graph of $I(II(I(II)I))$.
Right, the reduction graph of $I((I(III))(III))$.

1.3 Previous results

Reduction graphs were first studied by Klop in his PhD thesis and by Venturini Zilli [Klo80a, Zil84]. Those early works were about showcasing which graphs had corresponding λ-terms and which did not, for small examples. Venturini Zilli [Zil84] studied *linear graphs* (where every vertex has outgoing degree 1), and also introduced the notion of *cyclic equivalence* (M is cyclically equivalent to N if $M \to_\beta^* N$ and $N \to_\beta^* M$, i.e. they are in the same strongly connected component). While it is difficult to pinpoint a direct origin, cycles of arbitrary length were proven to be reduction graphs of λ-terms: for $n \in \mathbb{N}$, consider $M_n \coloneqq \lambda x_1 x_2 \cdots x_{n-1} y.yyy\cdots y$ (with $n+1$ occurrences of y) and let $C_n \coloneqq M_n M_n \cdots M_n$ (with $n+1$ occurrences of M_n). We have that $\mathrm{RED}(C_n)$ is a cycle of length n with all edges of multiplicity 1, see Figure 5. The existence of a reduction graph of two cycles touching at a single vertex was disproven in 2000, alongside an important conjecture of Venturini Zilli [IL00]. In 2016, a complete characterization of λ-terms with a cycle as a reduction graph was given [EKP16].

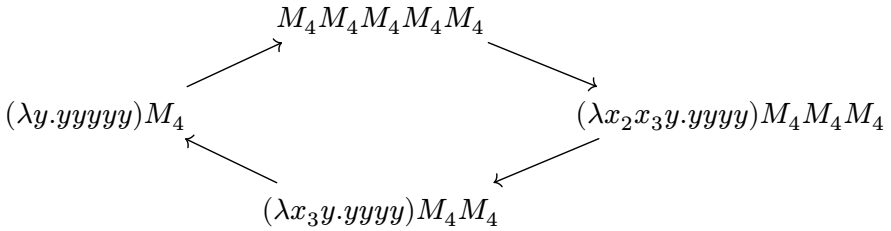

Figure 5: The reduction graph of C_4 is a cycle of length 4.

The notion of cyclic equivalence in the reduction graph was further studied by Klop [Klo80b], where he defined the notion of *plane* (an equivalence class for cyclic equivalence, i.e. a strongly connected component of the reduction graph) and formulated **Klop's conjecture**: if a term of a plane can directly exit the plane, then all terms in the plane can. This somewhat important conjecture was disproven by Shoji Sekimoto and Sachio Hirokawa a few years later [SH88] and independently by Hans Mulder [Mul86] (unpublished). A book was very recently published containing a chapter summarizing what is known about planes and reduction graphs [BM22].

While the topic of reduction graphs is very hard and not many results and conjectures have managed to be proven, it is also important to note the link with the work of Lévy. Lévy introduced a notion of labels on λ-terms to keep track of redexes across reductions [Lév76]. While not directly linked to the reduction graph of λ-terms, the reduction graphs generated by his form of labelled reductions were more regular and gained many mathematical properties. He managed to use those properties to try to capture a notion of "correct" and "optimal" reductions [Lév78, Lév22].

2 ORTrees equivalence

In this section, I will begin by introducing the rewrite system $(\mathbf{ORTrees}, \to_l)$, provide some examples, then prove that it is equivalent to $\big(\Lambda_I, \to_\beta\big)$, before using it to count the number of reductions of a term $M \in \Lambda_I$ to its normal form.

2.1 ORTrees

Definition 2. We denote by **ORTrees** the set of **O**rdered **R**ooted **Trees** whose children are finite and ordered from left to right. Formally, **ORTrees** is the smallest set generated by the following grammar:

$$T ::= l \mid N(T_1, ..., T_n) \qquad \forall n \geq 1 \tag{5}$$

where l is called a *leaf* and $N(T_1, ..., T_n)$ is a *node* having $T_1, ..., T_n$ as children.

Such a tree can also be represented using a **Dyck word**. For $\Sigma = \{(,)\}$ the alphabet containing the two parentheses, the *Dyck words* are well-parenthesized words over Σ, denoted $\mathcal{D}$. We can define a bijection $[\cdot] : \mathbf{ORTrees} \longrightarrow \mathcal{D}$ by induction with $[l] := \varepsilon$ and $[N(T_1, ..., T_n)] := ([T_1])([T_2])\cdots([T_n])$. For example, the tree $N(l)$ would be (), the tree $N(l, l, l)$ would be ()()(), and $N(l, N(l, l))$ would be () (()()). From now on, we will use Dyck words when appropriate to represent **ORTrees**.

We define inductively $T \equiv T'$ as the isomorphism of two **ORTrees** as unordered rooted trees:
- We have $l \equiv l$.
- If $T_1 \equiv T'_1, ..., T_n \equiv T'_n$, then for every bijection $\sigma : [n] \longleftrightarrow [n]$, we have $N(T_1, ..., T_n) \equiv N\Big(T'_{\sigma(1)}, ..., T'_{\sigma(n)}\Big)$.

Equivalently, since **ORTrees** can be seen as graphs in which edges point away from the root, $T_1 \equiv T_2$ corresponds to the already-defined equivalence $\equiv$ on their respective graphs. For $T \in \mathbf{ORTrees}$, we define V_T as its set of vertices when representing T as a graph. If $T_1 \equiv T_2$, then we denote by $\varphi_{T_1}^{T_2} : V_{T_1} \to V_{T_2}$ a bijective morphism transforming T_1 into T_2[3]. If $x \in V_T$, then we will denote by $T(x)$ the subtree rooted at x (we take all descendants starting from x, with x as the root).

RTrees. We define $\mathbf{RTrees} := (\mathbf{ORTrees}/_{\equiv})$. This means that **RTrees** are just **ORTrees** in which we forgot about the order of the children. See Figure 6 for an example. Such a tree can be defined inductively by saying that $l \in \mathbf{RTrees}$ and that if $T_1, ..., T_n \in \mathbf{RTrees}$, then $N(M) \in \mathbf{RTrees}$ with $M := \{|T_1, ..., T_n|\}$ the multiset containing $T_1, ..., T_n$. For $T \in \mathbf{ORTrees}$, let's define $\overline{T}$ as the **RTrees** corresponding to T if we forget its order.

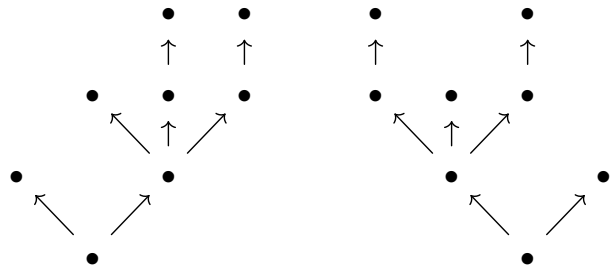

Figure 6: Two distinct **ORTrees** that are the same **RTrees**.
$T_1 =$ () (() (()) (())) on the left and $T_2 =$ ((()) () (())) () on the right.

Parameters. We need to define a few parameters on **ORTrees** and **RTrees**. For $T = N(T_1, ..., T_n) \in \mathbf{ORTrees}$ (or $T \in \mathbf{RTrees}$), denote by $r(T)$ its root, by $h(T)$ its height, and by $|T| := |V_T|$ its number of vertices.

For example, going back to Figure 6, we have $h(T) = 3$ and $|T| = 8$.

[3] it might not be unique.

Reduction. Let $T, T' \in$ **ORTrees**. Let's define the relation $T \to T'$ corresponding to "removing a non-root leaf". We say that $T \to T'$ with multiplicity m if there are m leaves $x_1, ..., x_m$ such that $T - x_i = T'$ (all diferrent from the root as the empty tree is not an **ORTrees**). For example, $N(l, N(l)) \to N(l, l)$ with multiplicity 1 (only removing the second leaf transforms $N(l, N(l))$ to $N(l, l)$), and $N(l, l) \to N(l)$ with multiplicity 2, as removing either the left or right leaf gives the same tree. Similarly to the I-terms, we can define the reduction graph of a tree T with respect to $\to$, see Figure 7 for an example. Removing a leave, when viewing **ORTrees** as Dyck's word, corresponding to removing a factor of the form () (i.e. we have $u()v \to uv$).

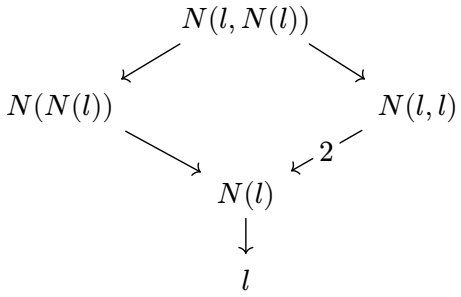

Figure 7: The reduction graph of $N(l, N(l))$

We say that T' is a *reduct* of T if $T \to^* T'$. For $S \subseteq V_T$ such that $r(T) \in S$ and $T[S]$ is connected, if we order $T[S]$ with the same ordering as T, then $T[S]$ is a reduct of T. All reducts can also be written in the form $T[S]$ for that ordering: if $T_1 \to T_2 \to \cdots \to T_k = T'$ is a reduction sequence of T in which we removed $x_2, x_3, ..., x_k$, then $T[V_T \setminus \{x_2, ..., x_k\}]$ with this ordering is T'. For that reason, we define

$$\mathcal{S}(T) \coloneqq \{S \subseteq V_T \mid r(T) \in S \wedge T[S] \text{ connected}\} \tag{6}$$

as the set of possible subsets of vertices corresponding to reducts. All reducts can be written $T[S]$ for some $S \in \mathcal{S}(T)$. Note that such a representation is not necessarily unique, see Figure 8 for an example of two different S, S' such that $T[S] = T[S']$.

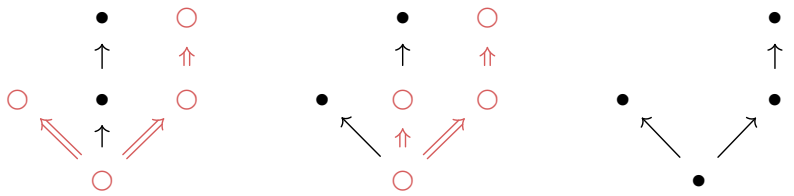

Figure 8: Two distinct $S, S' \in \mathcal{S}(T)$ represented with red empty circles (left and middle) denoting the same tree (right).

2.2 Equivalence

A careful reader may have noticed that Figure 7 looks a lot like the reduction graph of $I((II)I)$. This is not a coincidence, as we will show in this section with the following theorem:

Theorem 1. $(\Lambda_I, \to_\beta) \equiv$ (**ORTrees**, $\to$)

Theorem 1 means that reducing an I-term is exactly the same thing as removing leaves iteratively in a specific tree corresponding to the term. We will provide an easy-to-compute bijective transformation $\Phi : \Lambda_I \leftrightarrow$ **ORTrees** such that $M \to_\beta N$ with multiplicity m iff $\Phi(M) \to \Phi(N)$ with multiplicity m.

The transformation Φ. To every $M \in \Lambda_I$, we associate a tree $\Phi(M)$ by induction:

- If $M = I$, define $\Phi(M) \coloneqq l$.
- Otherwise, write M as its *right-spine decomposition* $M = u_1\ (u_2\ (u_3\ (\cdots (u_d\ I))))$, and let $\Phi(M) \coloneqq N(\Phi(u_1), \Phi(u_2), ..., \Phi(u_d))$.

Examples 2.

- For example, the term $M \coloneqq I((II)I)$ is decomposed as $u_1(u_2 I)$ with $u_1 = I$ and $u_2 = II$. We therefore have $\Phi(u_1) \coloneqq l, \Phi(u_2) \coloneqq N(l)$ and finally, $\Phi(M) = N(l, N(l))$. Looking at Figure 7 and Figure 2, we indeed have that $\Phi(M)$ and M have isomorphic reduction graphs. Figure 2 also gives us another

term with the same graph: $II(II)$. We have $\Phi(II(II)) = N(N(l), l)$, and one can easily verify that $\mathrm{RED}(N(N(l), l)) = \mathrm{RED}(N(l, N(l)))$.

- The terms $P_n = II\cdots I$ defined in Example 1 map to $N(\cdots(N(l))\cdots)$, the path of length n, and there is only a single leaf in $\Phi(P_{n+1})$, with $\Phi(P_{n+1}) \to \Phi(P_n)$. We will write P_n to represent $\Phi(P_n)$ when it is clear from the context that we are talking about a tree.
- The terms $E_n = I(I(\cdots(II)\cdots))$ defined in Example 1 map to $N(l, ..., l)$, and we indeed have $E_{n+1} \to^{n+1} E_n$. Just like for P_n, we will write E_n to represent $\Phi(E_n)$ when it is clear from the context that we are talking about a tree.

This transformation is bijective: we can explicitly define its inverse $\Phi^{-1} : \mathbf{ORTrees} \longrightarrow \Lambda_I$ inductively by

- $\Phi^{-1}(l) := I$, and
- $\Phi^{-1}(N(T_1, T_2, ..., T_d)) := \Phi^{-1}(T_1)(\Phi^{-1}(T_2)(\cdots(\Phi^{-1}(T_d)I)))$.

By a direct induction on M and T, we have $\Phi^{-1}(\Phi(M)) = M$ and $\Phi(\Phi^{-1}(T)) = T$. Therefore Φ is bijective.

Proof of Theorem 1. We prove that $\Phi : \Lambda_I \longleftrightarrow \mathbf{ORTrees}$ is a graph isomorphism. Let $M, N \in \Lambda_I$. We show that the multiplicity of $M \to_\beta N$ is the same as the multiplicity of $\Phi(M) \to \Phi(N)$. To that end, we show that if $M \to_\beta N$ has multiplicity $k > 0$, then $\Phi(M) \to \Phi(N)$ has at least multiplicity $k > 0$, and conversely that if $\Phi(M) \to \Phi(N)$ has multiplicity $k > 0$, then $M \to_\beta N$ has multiplicity at least k, by induction on M. If $M = I$, then $\Phi(M) = l$ and both $\Phi(M) \to \Phi(N)$ and $M \to N$ have multiplicity 0. Otherwise, write $M = u_1(\cdots(u_d I))$ with $\Phi(M) = N(T_1, ..., T_n)$.

($\Longrightarrow$) A given redex can either be strictly inside a u_i, or be of the form $u_i = I$ reducing $(\cdots(u_{i-1}(I(u_{i+1}(\cdots(u_d I)\cdots) \to_\beta (\cdots(u_{i-1}(u_{i+1}(\cdots(u_d I)\cdots)$ (we will call this second case a *rooted redex*). We will show that either all redexes are **all strictly inside one u_i** or they are all rooted redexes **next to each other**, meaning that the set of $\{i : (u_i\ (\cdots(u_d I))\cdots)$ is the redex with $u_i = I\}$ is a set of the form $[\![a; b]\!]$. Suppose they are not all strictly inside one u_i. Then:

- If we have two redexes $u_i \to^{r_1} u_i'$ and $u_j \to^{r_2} u_j'$ in different u_i, u_j with $i \neq j$, then those two redexes map to two different output terms, that is $N(T_1, ..., T_{i-1}, T_i', T_{i+1}, ..., T_d)$ and $N(T_1, ..., T_{j-1}, T_j', T_{j+1}, ..., T_d)$ for $T_i' = \Phi(u_i')$ and $T_j' = \Phi(u_j')$.
- If we have one rooted redex, and one redex strictly inside a u_i, they also have different output terms as one will decrease the number of the children of the root (the rooted one) and not the other one. So they are all rooted redexes.
- Suppose that there is $i < k$ with $u_i = I$ and $u_k = I$ two rooted redexes r_i, r_j and, by contradiction, that there is a $u_j \neq I$ with $i < j < k$. Let j be the smallest integer verifying the property, we show that the output of r_1 and r_2 are distinct: one has the $(j-1)$-th child be a l (if we reduce u_k) and the other one has the $(j-1)$-th child be a $\Phi(u_j)$ (if we reduce u_i). Therefore if u_i, u_k are two rooted redexes giving the same term once evaluated, all $i < j < k$ are also rooted redexes.

Finally,

- If the redexes of $M \to_\beta N$ are in the same u_i, then by induction there are at least k redexes $T_i \to T_i'$ with $\Phi(N) = N(T_1, ..., T_{i-1}, T_i', T_{i+1}, ..., T_d)$.
- Otherwise, it means that all k redexes are next to each other, which means that there are k leaves next to each other in T, and therefore at least k possible suppressions giving the same tree $\Phi(N)$. See Figure 9 for an example.

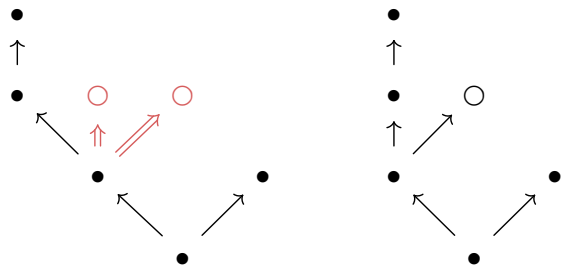

Figure 9: (Left) The tree of the term $((II)(\underline{I}(\underline{II})))(II)$ with the rooted redexes underlined and circled in the tree. They both map to $((II)(\underline{II}))(II)$ (Right)

($\Longleftarrow$) Reciprocally, suppose that $\Phi(M) \to \Phi(N)$ with multiplicity $k > 0$. Let $R = \{l_1, ..., l_k\}$ be the leaves such that removing them transforms $\Phi(M)$ to $\Phi(N)$. By induction, by an argument similar to the previous section, all leaves must be next to each other or in the same subterm. If it is in the same subterm then we can apply the induction hypothesis, so let's focus on the other case. Denote by i, j respectively the first and last indices of the leaves of R, with $j = i + k - 1$. Then $\Phi(M) = N\big(T_1, ..., T_{i-1}, l, ..., l, T_{j+1}, ..., T_d\big)$, and therefore we have

$$M = u_1\Big(\cdots\Big(u_{i-1}\Big(\underline{I\Big(\cdots\Big(I\big(u_{j+1}(\cdots(u_d I))\big)\Big)\Big)}\Big)\Big)\Big). \tag{7}$$

All the underlined subterms are the k redexes, all mapping to the same term N satisfying $\Phi(N) = N\big(T_1, ..., T_{i-1}, l, ..., l, T_{j+1}, ..., T_d\big)$ with one less leaf. □

2.3 Application: counting the number of reduction paths

For $T \in \mathbf{ORTrees}$, let $n = |T|$ and denote by $\#(T)$ the *number of ways* to reduce it to a leaf, so that $\#(T)$ is the number of paths from T to the leaf in $\mathrm{RED}(T)$, counting the multiplicity of the multi-edges. Such a path is called a *reduction sequence* and can be represented by an ordering of the vertices of the tree $\langle x_1, ..., x_n\rangle$, satisfying that for all $i < n$, x_i is a leaf in $T[\{x_i, ..., x_n\}]$: x_i represents the i-th leaf to be removed, and x_n is the final root.

Theorem 2. For all trees T, we have

$$\#(T) = \frac{|T|!}{\prod_{x \in V_T} |T(x)|} \tag{8}$$

Examples 3. Consider the reduction graph of Figure 7 for example. It has 3 reduction sequences:
- $N(l, N(l)) \to N(N(l)) \to N(l) \to l$
- $N(l, N(l)) \to N(l, l) \to^2 N(l) \to l$ (this sequence counts twice because of $N(l, l) \to^2 N(l)$)

If we apply the formula, we have $|T|! = 4! = 24$ and all subtrees are, with multiplicity, $\{|N(l, N(l)), l, N(l), l|\}$. We have $|N(l, N(l))| = |T| = 4$, $|l| = 1$ and $|N(l)| = 2$. Therefore, we have

$$\#(T) = \frac{4 \times 3 \times 2 \times 1}{4 \times 1 \times 2 \times 1} = 3 \tag{9}$$

For another example maximizing $\#(T)$, consider $E_n = N(l, l, ..., l)$ with n leaves, defined in Example 1. We have $|E_n|! = (n + 1)!$ and the multiset of all subtrees is $\{|E_n, l, ..., l|\}$. Therefore, we have $\prod_{x \in V_{E_n}} |E_n(x)| = n + 1$ and $\#(E_n) = n!$. We can check that this is correct: if we number the n leaves of E_n, since they are all independent and the root can only be chosen after all leaves are chosen, there are as many reduction sequences as there are orderings of the n leaves, that is, $n!$.

After writing the proof and obtaining the result, while reviewing the report using LLMs before submitting, they made me realized that this was an application of the *Hook's length formula* [BW89]. The proof, while written by me, follow the same proof as the original paper. It is available in the appendix, and work by induction on $|T|$, see Appendix B.

Corollary of Theorem 2. Define inductively the multiset $\mathrm{RS}(M)$ of *right spines* of $M \in \Lambda_I$ as $\mathrm{RS}(I) = \varnothing$ and, for $M \neq I$ written as $M = u_1\ (u_2\ (u_3\ (\cdots(u_d\ I))))$, define

$$\mathrm{RS}(M) = \{|(u_1, ..., u_d)|\} \uplus \biguplus_{i \leq d} \mathrm{RS}(u_i). \tag{10}$$

For $M \in \Lambda_I$ an identity-term, the number of ways to reduce M to its normal form I is

$$\frac{|M|_I\ !}{\prod_{(u_1,...,u_d) \in \mathrm{RS}(M)} \Big(1 + \sum_{i \leq d} |u_i|_I\Big)}. \tag{11}$$

3 First Characterization

Now that working with I-terms is easier since we can just work on **ORTrees**, we will study under what conditions two $T, T' \in \mathbf{ORTrees}$ verify $\mathrm{RED}(T) \equiv \mathrm{RED}(T')$. We will first introduce the *dyslexic* relation $\leftharpoondown$ [Bow] and the *uniformly dyslexic* relation $\leftrightharpoons$, showing that while the dyslexic relation is not what we want, the uniformly dyslexic relation satisfies the very important "functoriality lemma", before showing that $T \leftrightharpoons T' \Rightarrow \mathrm{RED}(T) \equiv \mathrm{RED}(T')$.

3.1 Dyslexic Trees

Definition 3. A *dyslexic tree* is a tree such that at every node we can read the children from left to right or from right to left [Bow]. Formally, we can define $T \leftharpoondown T'$ by induction:

- We have $l \leftharpoondown l$, and
- If $T_1 \leftharpoondown T'_1, T_2 \leftharpoondown T'_2, ..., T_d \leftharpoondown T'_d$, then we have both

$$\begin{aligned} N(T_1, ..., T_d) &\leftharpoondown N(T'_1, ..., T'_d) \\ N(T_1, ..., T_d) &\leftharpoondown N(T'_d, ..., T'_1) \end{aligned} \tag{12}$$

Examples 4. We have $N(l, N(l)) \leftharpoondown N(N(l), l)$, as we just need to reverse the order at the root. The two trees of Figure 6 are not dyslexic, as $N(l, N(l), N(l))$ is only dyslexic with $N(N(l), N(l), l)$ and not $N(N(l), l, N(l))$. See Figure 10

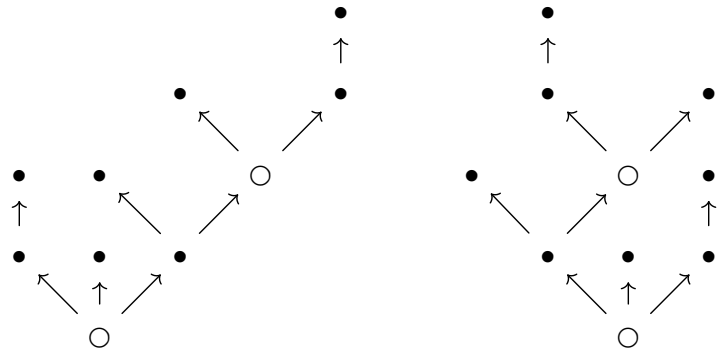

Figure 10: Two dyslexic tree. To get from left to right, we reverse the order of the circled nodes.

Counterexample. Taking the example $N(l, N(l)) \leftharpoondown N(N(l), l)$, which has isomorphic reduction graphs, it was a conjecture of ours that $\forall T, T' \in \mathbf{ORTrees}, T \leftharpoondown T' \Rightarrow \mathrm{RED}(T) \equiv \mathrm{RED}(T')$: intuitively, we thought that if there was a reduction path, we could just "reverse" the role of every child to get some sort of isomorphism from $\mathrm{RED}(T)$ to $\mathrm{RED}(T')$. Unfortunately, here is a counterexample:

- The tree $L_1 := ((((()))())(((()))()))$
- And the tree $L_2 := ((((()))())((((()))())()))$

By denoting $M_1 := N(l, N(l))$, $M_2 := N(N(l), l)$ and $A := N(M_1, l)$, those two terms correspond to $L_1 = N(A, M_1)$ and $L_2 = N(A, M_2)$. $\mathrm{RED}(L_1)$ has 133 nodes and $\mathrm{RED}(L_2)$ has 135. Informally[4], the reason for this is that L_2 can reduce to both M_1 and M_2, while L_1 can only reduce itself to M_1 twice, once in A and once in M_1. The graphs are visible at Figure 11.

How can we fix that? To prevent two terms that are next to each other from differing, we will make the reversing operation of dyslexic trees uniform over the same depth: this leads to *uniformly dyslexic trees*.

[4] "Informally" here means that while the statement is not correct, I believe it to transmit the idea of why this is a valid counter example.

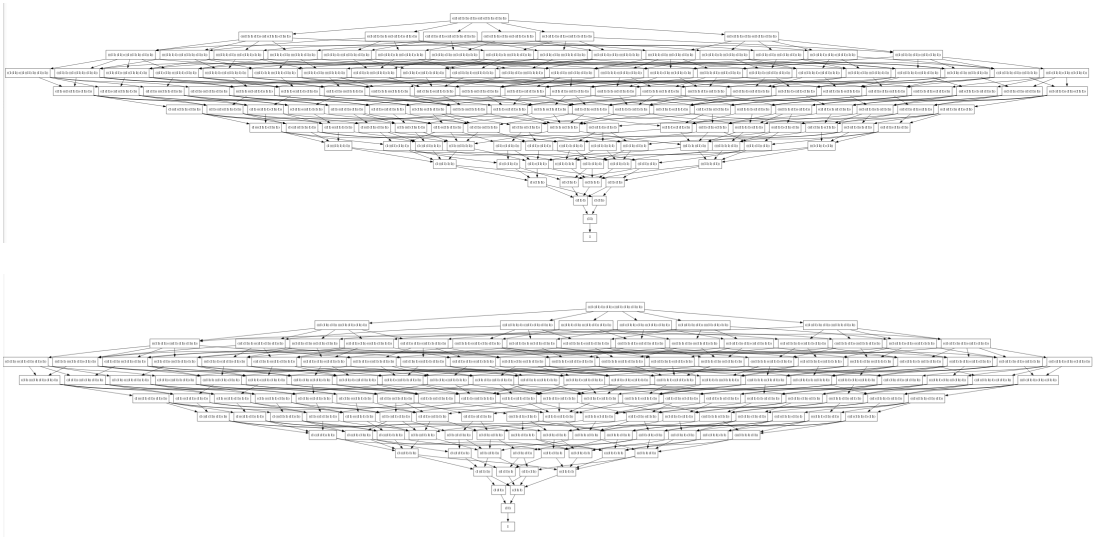

Figure 11: The reduction graphs of L_1 (top) and L_2 (bottom). They are not meant to be read, they are just depicted here to serve as a showcase to the scale of the counter-example.

3.2 Uniformly Dyslexic Trees

The idea of two trees being *uniformly dyslexic* is that, instead of picking the nodes at which we will reverse the order, we now pick the depths at which we reverse the order of all nodes of those depths.

Definition 4. For a fixed $P \subseteq \mathbb{N}$, we define a tree transformation $\rho_P^i : \mathbf{ORTrees} \longrightarrow \mathbf{ORTrees}$, with i the depth of a node, by induction:

$$\begin{aligned} \rho_P^i(l) &= l, \text{ for } \ l \ \text{ a leaf,} \\ \rho_P^i(N(x_0, ..., x_n)) &= \begin{cases} N\big(\rho_P^{i+1}(x_n), ..., \rho_P^{i+1}(x_0)\big), & \text{if } \ i \in P, \\ N\big(\rho_P^{i+1}(x_0), ..., \rho_P^{i+1}(x_n)\big), & \text{if } \ i \notin P. \end{cases} \end{aligned} \tag{13}$$

We write ρ_P for ρ_P^0. We then define the relation $T \leftrightharpoons T' \iff \exists P \subseteq \mathbb{N}, \rho_P(T) = T'$. The idea is that P is the set of depths at which we reverse the left-to-right order into a right-to-left one.

Examples 5.

- We have that $N(l, N(l)) \leftrightharpoons N(N(l), l)$: by picking $P = \{0\}$, we reverse the order only at the root, and $\rho_P^0(N(l, N(l))) = N(N(l), l)$.
- The two dyslexic trees of Figure 10 are also uniformly dyslexic, see Figure 12.
- While all uniformly dyslexic trees are also dyslexic, the converse is not true: if $M_1 = N(l, N(l))$ and $M_2 = N(N(l), l)$, then $N(M_1, M_1) \leftharpoonup N(M_1, M_2)$, but they are not uniformly dyslexic (we would need to reverse only the rightmost M_1). In particular, the counterexample from Figure 11, (L_1, L_2), is not uniformly dyslexic.

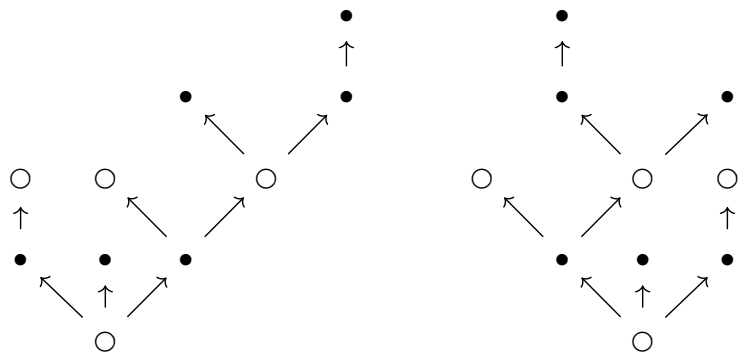

Figure 12: Two uniformly dyslexic tree with $P = \{0, 2\}$. To fo from the left tree to the right one, we reverse the order of the circled nodes, at depths 0 and 2.

Properties of $\leftrightharpoons$. The relation $\leftrightharpoons$ satisfies the following 3 properties:

- For every $T \in \mathbf{ORTrees}$, we have $\rho_\varnothing(T) = T$.
- For $A, B \subseteq \mathbb{N}$, we have $\rho_{A\Delta B} = \rho_A \circ \rho_B$.[5]
- For all $P \subseteq \mathbb{N}$, $\rho_P^i : \mathbf{ORTrees} \longrightarrow \mathbf{ORTrees}$ is an involution.

Proof. The first two are by induction on the tree T, with i always universally quantified. For the base case, we indeed have for all i, $\rho_\varnothing^i(l) = l$ and $\rho_{A\Delta B}^i(l) = l = \rho_A^i(\rho_B^i(l))$. Let $T = N(T_1, ..., T_d)$. Let us prove both inductive cases:

- By induction, we have that $\forall j \in [d], \forall i, \rho_\varnothing^i\big(T_j\big) = T_j$, and so we have
$$\rho_\varnothing^i(T) = N\big(\rho_\varnothing^{i+1}(T_1), ..., \rho_\varnothing^{i+1}(T_d)\big) = N(T_1, ..., T_d) = T. \tag{14}$$
- By induction, we have that $\forall j \in [d], \forall i, \rho_{A\Delta B}^i\big(T_j\big) = \rho_A^i\big(\rho_B^i\big(T_j\big)\big)$, and we have two cases:
  1. Either $i \in A\Delta B$, so $i \in A \setminus B$, or $i \in B \setminus A$, and in both cases we have
$$\begin{aligned}\rho_{A\Delta B}^i(T) &= N\Big(\rho_{A\Delta B}^{i+1}(T_d), ..., \rho_{A\Delta B}^{i+1}(T_1)\Big)\\ &= N\Big(\rho_A^{i+1}\big(\rho_B^{i+1}(T_d)\big), ..., \rho_A^{i+1}\big(\rho_B^{i+1}(T_1)\big)\Big)\\ &= \rho_A^i(\rho_B^i(T)).\end{aligned} \tag{15}$$
  2. Or $i \notin A\Delta B$, so $i \in A \cap B$, or $i \notin A \vee B$, and in both cases we have
$$\begin{aligned}\rho_{A\Delta B}^i(T) &= N\Big(\rho_{A\Delta B}^{i+1}(T_1), ..., \rho_{A\Delta B}^{i+1}(T_d)\Big)\\ &= N\Big(\rho_A^{i+1}\big(\rho_B^{i+1}(T_1)\big), ..., \rho_A^{i+1}\big(\rho_B^{i+1}(T_d)\big)\Big)\\ &= \rho_A^i(\rho_B^i(T)).\end{aligned} \tag{16}$$

The third property is a direct consequence of both: we have $\rho_S^i\big(\rho_S^i(T)\big) = \rho_{S\Delta S}^i(T) = \rho_\varnothing^i(T) = T$, so ρ_S^i is an involution. □

Corollary 2. $\leftrightharpoons$ is an equivalence relation:

- For $T \in \mathbf{ORTrees}$, we have $\rho_\varnothing(T) = T$, so $T \leftrightharpoons T$.
- For $T, T' \in \mathbf{ORTrees}$, if we have $\rho_S(T) = T'$, then $\rho_S(T') = \rho_S(\rho_S(T)) = T$ because ρ_S is an involution.
- For $T, T', T'' \in \mathbf{ORTrees}$, suppose $\rho_S(T) = T'$ and $\rho_{S'}(T') = T''$, then for $S^* \coloneqq S\Delta S'$ we have $\rho_{S\Delta S'}(T) = \rho_{S'}(\rho_S(T)) = T''$.

The morphisms. By induction, it is easy to show that if $T \leftrightharpoons T'$, then $T \equiv T'$ as defined in Section 2.1 (they have the same **RTrees** if we forget the order, i.e. the same parent/children graph). Therefore, recall that we denote by $\varphi_T^{T'} : V_T \longleftrightarrow V_{T'}$ the tree isomorphism. When it is clear from context, we will refer to $\varphi_T^{T'}$ as just φ. It is important to understand that $\rho_P : \mathbf{ORTrees} \longrightarrow \mathbf{ORTrees}$ is an operation on **ORTrees**, while $\varphi_T^{T'}$ is the operation on the vertices of a fixed tree.

3.3 The Functoriality Lemma

To show that $\forall T, T' \in \mathbf{ORTrees}, T \leftrightharpoons T' \Longrightarrow \mathrm{RED}(T) \equiv \mathrm{RED}(T')$, we will prove in this section the very important *Functoriality Lemma*:

Functoriality Lemma[6]. Let $T, T' \in \mathbf{ORTrees}$ such that $T \leftrightharpoons T'$. Then, for all $S, S' \in \mathcal{S}(T)$, we have
$$T[S] = T[S'] \Longleftrightarrow T'\left[\varphi_T^{T'}(S)\right] = T'\left[\varphi_T^{T'}(S')\right] \tag{17}$$

This lemma states that if there are two reducts of T in its reduction graph, one being $T \to^* T[S]$ and the other being $T \to^* T[S']$, then the two corresponding reducts of T' are equal iff the original ones are. To prove the functoriality lemma, we will first prove a preliminary lemma:

Lemma 1. For all $T, T' \in \mathbf{ORTrees}$, if $\rho_P(T) = T'$, then for all $S \in \mathcal{S}(T)$ we have

[5] Recall that $A\Delta B$ is the symmetric difference.

[6] The name of this lemma has not yet been fixed. Since this proof has a categorical approach (see last section), we want to make sure that this lemma indeed consists of φ being a functor between two categories before fixing the name, and this requires a bit more research.

$$\rho_P(T[S]) = \rho_P(T)\left[\varphi_T^{T'}(S)\right] \tag{18}$$

This, intuitively, corresponds to ρ_P "distributing" itself over $T[S]$.

Proof of Lemma 1. We prove Lemma 1 by induction on T, with both i and S universally quantified. If $T = l$ is a leaf, then necessarily $S = V_T$ since it must contain the root, hence $\rho_P^i(T)\left[\varphi_T^{T'}(S)\right] = l = \rho_P^i(T[S])$.

For the inductive case, let $S \in \mathcal{S}(T)$ and $i \in \mathbb{N}$, and write $T = N(T_1, ..., T_d)$. and $T' = N(T'_1, ..., T'_d)$. For every $0 \leq j \leq d$, let $S_j \coloneqq S \cap V_{T_j}$. Since S is connected and contains the root, every S_j is either empty or belongs to $\mathcal{S}(T_j)$. We adopt the convention that $N(T_1, ..., T_j[\varnothing], ..., T_d)$ denotes that the j-th child doesn't exists. We have two cases:

- If $i \notin P$, by induction hypothesis, we have that $\forall j \in [d], \rho_P^{i+1}(T_j)\left[\varphi_{T_j}^{T'_j}(S_j)\right] = \rho_P^{i+1}(T_j[S_j])$ for non-empty S_j (and for the empty ones the quantity doesn't exist in both side). Therefore, we have

$$\begin{aligned}\rho_P^i(T)[\varphi_P^T(S)] &= N\left(\rho_P^{i+1}(T_1)\left[\varphi_{T_1}^{T'_1}(S_1)\right], ..., \rho_P^{i+1}(T_d)\left[\varphi_{T_d}^{T'_d}(S_d)\right]\right) \\ &= N(\rho_P^{i+1}(T_1[S_1]), ..., \rho_P^{i+1}(T_d[S_d])) \\ &= \rho_P^i(T[S])\end{aligned} \tag{19}$$

- If $i \in P$, by induction hypothesis, we have the similar property that $\forall j \in [d], \rho_P^{i+1}(T_j)\left[\varphi_{T_j}^{T'_{d-j+1}}(S_j)\right] = \rho_P^{i+1}(T_j[S_j])$ for non-empty S_j (and the same remark applies for empty S_j). Therefore, we have

$$\begin{aligned}\rho_P^i(T)[\varphi_P^T(S)] &= N\left(\rho_P^{i+1}(T_d)\left[\varphi_{T_d}^{T'_1}(S_d)\right], ..., \rho_P^{i+1}(T_1)\left[\varphi_{T_1}^{T'_d}(S_1)\right]\right) \\ &= N(\rho_P^{i+1}(T_d[S_d]), ..., \rho_P^{i+1}(T_1[S_1])) \\ &= \rho_P^i(T[S])\end{aligned} \tag{20}$$

Finally, we can prove the functoriality lemma:

Corollary 3. Suppose $T[S] = T[S']$. Then $\rho_P(T[S]) = \rho_P(T[S'])$, so we have

$$T'[\varphi(S)] = \rho_P(T)\left[\varphi_T^{T'}(S)\right] \equiv \rho_P(T[S]) \equiv \rho_P(T[S']) \equiv \rho_P(T)\left[\varphi_T^{T'}(S')\right] = T'[\varphi(S')]. \tag{21}$$

The other direction follows by symmetry of $\leftrightharpoons$: since we have $\varphi_T^{T'} : V_T \longrightarrow V_{T'}$ bijective, $\mathcal{S}(T') = \varphi(\mathcal{S}(T))$. □

3.4 First reduction graph characterisation

Now that the functoriality lemma has been proven, the proof of Theorem 3 is actually very short!

Theorem 3. For all $T, T' \in \mathbf{ORTrees}$, $T \leftrightharpoons T' \implies \mathrm{RED}(T) \equiv \mathrm{RED}(T')$.

Proof. Suppose $T \leftrightharpoons T'$. Then define the isomorphism:

$$\begin{aligned}\Psi : \mathrm{RED}(T) &\longleftrightarrow \mathrm{RED}(T') \\ T[S] &\longmapsto\ T'[\varphi(S)]\end{aligned} \tag{22}$$

This is well-defined by the functoriality lemma: if $T[S] = T[S']$ are two distinct representations of the same term, then $T'[\varphi(S)] = T'[\varphi(S')]$. It is also injective because of the functoriality lemma: if $T'[\varphi(S)] = T'[\varphi(S')]$, then $T[S] = T[S']$. Finally, it is bijective as $T'[S'] \mapsto T[\varphi^{-1}(S')]$ is a clear inverse.

We only need to show that Ψ preserves the edges. We show that if $T_1 \to T_2$ is an edge in $\mathrm{RED}(T)$ with multiplicity m, then $\Psi(T_1) \to \Psi(T_2)$ is an edge in $\mathrm{RED}(T')$ with multiplicity $m' \geq m$. By doing the same proof for Ψ^{-1}, we get equality.

Suppose that there is an edge $T_1 \to T_2$ in $\mathrm{RED}(T)$ with multiplicity m. If $m = 0$ there is nothing to do, so suppose $m \geq 1$. Fix $S \in \mathcal{S}(T)$ such that $T[S] = T_1$. Then there are exactly m leaves $l_1, ..., l_m$ of $T[S]$ such that $T[S \setminus \{l_i\}] = T_2$.

Let x be one of those leaves. Because φ is a rooted tree isomorphism, $\varphi(x)$ is a leaf of $T'[\varphi(S)]$. Also, since $\varphi(S \setminus \{x\}) = \varphi(S) \setminus \{\varphi(x)\}$, there is an edge $T'[\varphi(S)] \to T'[\varphi(S \setminus \{x\})]$. Moreover, by definition of Ψ, we have $T'[\varphi(S)] = \Psi(T_1)$ and $T'[\varphi(S \setminus \{x\})] = \Psi(T_2)$.

The map $l_i \mapsto \varphi(l_i)$ is injective over the set $\{l_1, ..., l_m\}$ of the m leaves, since φ is injective. Therefore the m distinct reductions from T_1 to T_2 give m distinct reductions from $\Psi(T_1)$ to $\Psi(T_2)$, so $m' \geq m$. By applying the same argument to Ψ^{-1} we obtain $m \geq m'$, hence $m = m'$.

Therefore $\mathrm{RED}(T) \overset{\Psi}{\equiv} \mathrm{RED}(T')$. □

We do not have a counterexample to the reciprocal. Although it seems unlikely, it could be a full characterisation. I tested all terms T, T' of size $|T| \leq 11$, as well as some other good candidates built from $N(l, N(l))$ and $N(N(l), l)$. While we did not manage to prove or disprove that $T \leftrightharpoons T' \Longleftarrow \mathrm{RED}(T) \equiv \mathrm{RED}(T')$, we can manage to prove some weaker "reconstruction" properties.

4 (Re)construction of Reduction Graphs

In this section, we will focus on trying to recover properties on T, T' from the reduction graph, and understand what fundamental structure is behind it. We will first prove a *reconstruction lemma* giving us that $\mathrm{RED}(T) \equiv \mathrm{RED}(T') \Longrightarrow T \leftharpoonup T'$, before showing a way that the reduction graph of T can be build using the functoriality lemma and T as an **RTrees**.

4.1 RED(T) Reconstruction Lemma

For $T \in$ **ORTrees**, write $T = N(T_1, ..., T_d)$. We define:
- $n(T) \coloneqq d$, the number of children of the root, with $n(l) \coloneqq 0$,
- $l(T) \coloneqq |\{1 \leq i \leq n \mid T_i = l\}|$, the number of children of the root which are leaves, with $l(l) \coloneqq 0$.

For example, the term $M \coloneqq N(l, l, N(l), l, N(l), N(N(l), l), l, l)$ has $n(M) = 8$ and $l(M) = 5$.

In this subsection we will prove the following theorem

Theorem 4. For all $T, T' \in$ **ORTrees**, if $\mathrm{RED}(T) \equiv \mathrm{RED}(T')$, then $T \leftharpoonup T'$.

To proceed, we will have to retrieve some information about T from its graph $\mathrm{RED}(T)$. This is the goal of the following "Reconstruction Lemma":

Reconstruction Lemma. Let $T, T' \in$ **ORTrees** be different from leaves and be such that $\mathrm{RED}(T) \equiv \mathrm{RED}(T')$. Write $T = N(T_1, ..., T_n)$ and $T' = N(T'_1, ..., T'_m)$. Then:
1. $n = m$, $h(T) = h(T')$, $|T| = |T'|$ and $l(T) = l(T')$.
2. There exists a bijection $\theta : [n] \longrightarrow [n]$ such that, for every $i \in [n]$, $\mathrm{RED}(T_i) \equiv \mathrm{RED}\left(T'_{\theta(i)}\right)$.
3. Furthermore, this bijection θ is either the indentity or the reverse function $\theta : i \mapsto n - i + 1$

This is exactly what we are after: it means that given only the structure of a reduction graph $\mathrm{RED}(T)$, we can assert that, up to the left-right order of the children, we know the list of the children. A key part of the proof is first to prove that in the unlabelled version of the graph, we can find the position of special trees: the trees $(E_i)_i$ and $(P_i)_i$ (recall that P_i is the path of length i and E_i is the tree of height 1 with i leaves).

This proof was by far the hardest part of the internship. I will here only present the first part, the second and third being in Addendum of this report.

Proof. Denote by $\Psi : \mathrm{RED}(T) \longleftrightarrow \mathrm{RED}(T')$ the DAG isomorphism. For $u, v \in \mathrm{RED}(T)$, define $u \preceq v$ iff u is below or equal to v in $\mathrm{RED}(T)$, that is, iff $v \to^* u$.

Since we need to prove that $n = m$, $h(T) = h(T')$, $|T| = |T'|$ and $l(T) = l(T')$, let us prove that we can compute all those measurements only using the shape of the graph, i.e. without looking at the labels. For example, $|T|$ can easily be computed: since at every reduction we remove a leaf, $|T| - 1$ is exactly the height of the reduction graph. Since this can be computed only using the graph, i.e. up to isomorphism, we have that if $\mathrm{RED}(T) \equiv \mathrm{RED}(T')$ then $|T| = |T'|$. We will proceed similarly for the other measurements.

Computing $h(T)$. Recall that $P_i \coloneqq N(\cdots(N(l))\cdots)$ is the path of length i, and that P_{i+1} only has a single reduction to P_i. It is the only term of size i with this property, as all other trees have at least 2 leaves, and

therefore two reductions. We can therefore find the position of all the P_i present in the reduction graph: they form the longest chain of out-degree 1 elements that terminates at the final vertex. For example, P_0 is the final vertex, P_1 is the only one above it, P_2 is the only one with an edge of arity 1 above it, and similarly for P_3, etc. See the left graph of Figure 13 for an example. Then we can compute $h(T)$: it is the largest i such that $P_i \in \mathrm{RED}(T)$:

- If $h(T) = k$, then there is a path $\langle x_1, ..., x_k \rangle$ with $r(T) = x_1$ from the root to a leaf of length k, and $T[\{x_1, ..., x_k\}] = P_k$ is a reduct of T (we iteratively remove all nodes not in the path, starting from the leaves).
- If $P_h \in \mathrm{RED}(T)$, then $P_h = T[S]$ and this S is a path of length h in T from the root to a leaf, so $h(T) \geq h$.

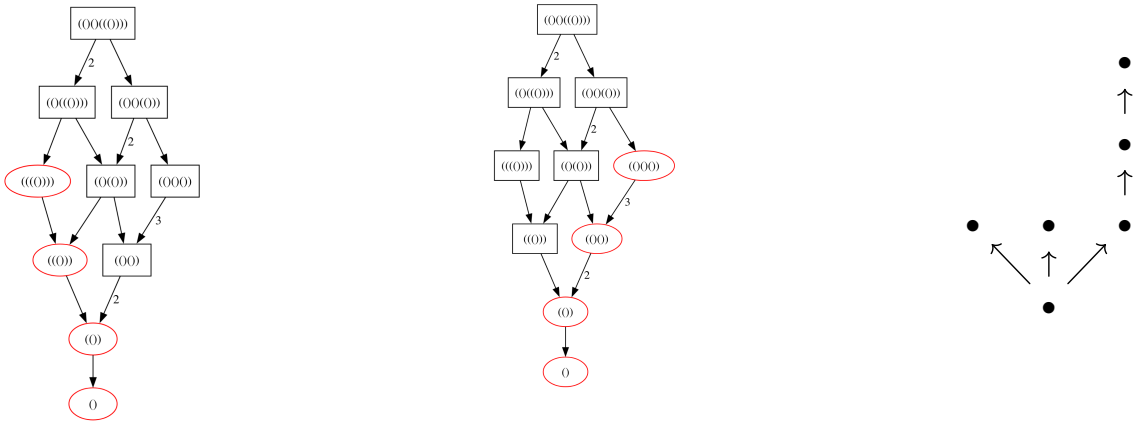

Figure 13: The reduction graph of $N(l, l, N(N(l)))$.
(Left) Circled the nodes of outdegree 1, i.e. the P_i
(Middle) Circled the nodes of increasing outgoing degree, i.e. the E_i
(Right) The tree $T := N(l, l, N(N(l)))$ with $h(T) = 3$ and $n(T) = 3$

Computing $n(T)$. We will proceed in a similar way to compute $n(T)$. Recall that $E_i = N(l, ..., l)$ with i leaves. E_{i+1} only has one outgoing edge, namely $E_{i+1} \to^{i+1} E_i$. It is the only term of size $i + 2$ to have a reduction of arity $i + 1$ to E_i: suppose a term $T \to^{i+1} E_i$. Then we added a leaf to E_i, and it can only be at the root, so $T = E_{i+1}$. Therefore, we can find the position of all the E_i present in the reduction graph, as they form the longest chain of increasing outgoing-degree elements that terminates at the final vertex (see the middle of Figure 13). The proof that $n(T)$ is the largest i such that $E_i \in \mathrm{RED}(T)$ is exactly the same as for P_i and $h(T)$, but instead of having S be a path, we have S be a star with i branches (i.e. S select the root and i of its children).

We can compute $h(T)$ and $n(T)$ for the root, but since $\mathrm{RED}(t) = \mathrm{RED}(T)[\{x \in \mathrm{RED}(T) \mid x \preceq t\}]$, we can also compute $h(t)$ and $n(t)$ for every $t \in \mathrm{RED}(T)$: $h(t)$ is the largest i such that $P_i \preceq t$, and similarly, $n(t)$ is the largest i such that $E_i \preceq t$.

Computing $l(T)$. Write $T = N(T_1, ..., T_d)$. We want to find how many i's satisfy $T_i = l$. First, notice that if $T \to T'$ with $n(T') < n(T)$, it means that we lost a child of the root, and the only possibility to do that in one removal of a leaf is that we removed a leaf which is a direct child of the root. So every edge of the form $T \to^k t$ with $n(t) < n(T)$ corresponds to a leaf at the root. Reciprocally, for every $i \in [d]$ with $T_i = l$, removing the leaf T_i gives $T \to N(T_1, ..., T_{i-1}, T_{i+1}, ..., T_d)$, whose number of children is $n(T) - 1 < n(T)$, so every direct leaf of the root contributes exactly one to the sum. Denoting $\#(T \to t)$ the arity of the edge $T \to t$, this gives us:

$$l(T) = \sum_{t \in \mathrm{RED}(T), n(t) < n(T)} \#(T \to t). \tag{23}$$

Since $l(T)$ is computed only by looking at the outgoing edges of $\mathrm{RED}(T)$ and the values $n(t)$, $l(T)$ is also invariant under isomorphism. So we have proved the first part of the reconstruction lemma.

The second and third part is proven in Addendum 1. □

Corollary 4. Now that we have the reconstruction lemma, we can easily prove Theorem 4 by induction on $|T|$:

If $T = l$, then $\mathrm{RED}(T)$ has only one vertex, so $\mathrm{RED}(T')$ also has only one vertex. Therefore $T' = l$, and indeed $T \leftharpoonup T'$.

Let us prove the inductive case. By the reconstruction lemma, we can write $T = N(T_1, ..., T_d)$ and $T' = N(T'_1, ..., T'_d)$ with a bijection $\theta \in \{\mathrm{Id}_d, \mathrm{Rev}_d\}$[7] such that $\forall i \in [d], \mathrm{RED}(T_i) \equiv \mathrm{RED}\left(T'_{\theta(i)}\right)$. By the induction hypothesis, we have that $\forall i \in [d], T_i \leftharpoonup T'_{\theta(i)}$, and therefore

$$N(T_1, ..., T_d) \equiv N\left(T'_{\theta(1)}, ..., T'_{\theta(d)}\right) \equiv N(T'_1, ..., T'_d). \tag{24}$$

Hence $T \equiv T'$. Furthermore, the isomorphism $\varphi_T^{T'} : V_T \longleftrightarrow V_{T'}$ can be inductively defined using θ, by mapping the root of T to the root of T' and, for every $i \in [d]$, by mapping the vertices of T_i to the vertices of $T'_{\theta(i)}$ using the isomorphism given by the induction hypothesis. □

4.2 A Labelled construction of the Reduction Graph

We shall show here that there is a way to construct the reduction graph of a term T using both T as an **RTrees** and a functoriality-lemma type of property. Recall that for $T \in \mathbf{ORTrees}$, V_T is its set of vertices, $r(T)$ is its root and that

$$\mathcal{S}(T) = \{S \subseteq V_T \mid r(T) \in S \wedge S \text{ is connected}\} \tag{25}$$

The set $\mathcal{S}(T)$ can be ordered using $\subseteq$ and in that case, represent a lattice that looks very similar to $\mathrm{RED}(T)$. Indeed, since every $t \in \mathrm{RED}(T)$ can be written $T[S]$ for some $S \in \mathcal{S}(T)$, and that $t \to^m t'$ iff there exists m leaves $l_1, ..., l_m$ of t such that $t - l_i = t'$, we can construct $\mathrm{RED}(T)$ using $\mathcal{S}(T)$. $\mathcal{S}(T)$ can be seen as a "labelled" version of $\mathrm{RED}(T)$, were every edge is labeled by the unique identifier of the leaf we removed. Furthermore, since $(\mathcal{S}(T), \subseteq)$ have no notion of "left" or "right" (removing different children always yield different trees), we have that $T \equiv T' \Longrightarrow (\mathcal{S}(T), \subseteq) \equiv (\mathcal{S}(T'), \subseteq)$. But, since we want only to consider "one-step" edges, and not have an edge $T \to l$ directly, we arrive at considering the *Hasse graph* of $\mathcal{S}(T)$.

Oriented Hasse graph. Let $T \in \mathbf{RTrees}$, we define $\mathcal{H}(T) = (\mathcal{S}(T), \subset_1)$ the *Hasse graph* of $\mathcal{S}(T)$, formally defined with $S \to S'$ iff $|S - S'| = 1 \wedge S' \subseteq S$. Remark that such a graph is never a multi-graph.

Let $G = (V, E)$ be a (multi)-graph and $\sim$ an equivalence relation on V. Suppose that, if C is an equivalence class of $\sim$ and $x \sim y$, then (denoting $\#(x, y)$ the arity of the edge (x, y)) we have

$$\sum_{c \in C} \#(x, c) = \sum_{c \in C} \#(y, c). \tag{26}$$

Then we can define ${}^{G}\!/_{\sim}$ as the graph $G = ({}^{V}\!/_{\sim}, E_{\sim})$ with $\#_{E_{\sim}}(C_1, C_2) = \sum_{c \in C_2} \#(x, c)$ for any $x \in C_1$.

Morally, ${}^{G}\!/_{\sim}$ is the graph of the equivalence class in which we picked one vertex (any one) as a representative for outgoing edges, and is well defined since the arity of outgoing edges does not depend on the representative we picked.

If we then define for $S, S' \in \mathcal{S}(T)$ the equivalence relation $S \underset{T}{\sim} S'$ by $T[S] = T[S']$ (and will sometimes simply write $S \sim S'$), we have the following theorem :

Theorem 5. For all $T \in \mathbf{ORTrees}$, we have ${}^{\mathcal{H}(T)}\!/_{\sim} \equiv \mathrm{RED}(T)$

This is a very interesting theorem, for multiple reasons:

- The relation $\sim$ is very close to the functoriality lemma, as the functoriality lemma can be staten as

$$\forall S, S' \in \mathcal{S}(T), \quad S \underset{T}{\sim} S' \Longleftrightarrow \varphi(S) \underset{T'}{\sim} \varphi(S') \tag{27}$$

- Since $\mathcal{S}(T)$ only depend on T as an **RTrees**, if $T \equiv T'$ then $\mathcal{H}(T) \equiv \mathcal{H}(T')$

The proof doesn't use some clever techniques: it is therefore completly left in Addendum 2.

[7] Id_d is the identity over $[d]$ and $\mathrm{Rev}_d : [d] \longrightarrow [d]$ is defined by $\mathrm{Rev}_d(i) := d - i + 1$ the "reverse" function.

It seems that such a theorem would allow us to easily prove some kind of equivalence between $\mathrm{RED}(T) \equiv \mathrm{RED}(T')$ and the fact that $T \equiv T'$ and T, T' respecting the functoriality lemma. Unfortunately, the operation ${}^{G}/_{\sim}$ is quite complicated and I did not manage to prove such an equivalence. This shall therefore be the final conjecture of this internship:

Conjecture. For all $T, T' \in \mathbf{ORTrees}$, we have

$$\mathrm{RED}(T) \equiv \mathrm{RED}(T') \iff \exists \varphi : V_T \leftrightarrow V_{T'}, T \overset{\varphi}{\equiv} T' \wedge \forall S, S' \in \mathcal{S}(T), S \sim S' \iff \varphi(S) \sim \varphi(S') \quad (28)$$

Something that further solidify this conjecture is that we have already proven one way and a part of the other way:

- If $T \equiv T'$ and the functoriality lemma holds, then $\Phi : T[S] \mapsto T'\left[\varphi_T^{T'}(S)\right]$ is an isomophism between $\mathrm{RED}(T)$ and $\mathrm{RED}(T')$, and the same proof as the one of Theorem 3 works.
- If $\mathrm{RED}(T) \equiv \mathrm{RED}(T')$, then $T \equiv T'$ and $T \leftharpoonup T'$ (the reconstruction lemma).

But I did not manage to prove that the functoriality lemma holds if $\mathrm{RED}(T) \equiv \mathrm{RED}(T')$, even if we know that $\mathcal{H}(T) \equiv \mathcal{H}(T')$ and that $\mathrm{RED}(T)$ can be built from $\mathcal{H}(T)$ and the functoriality lemma.

5 Conclusion

Overall, I manage to prove that reducing identity-terms are the same as removing leaves of **ORTrees** (or removing () from Dyck words), giving us a combinatorial model of a fragment of identity-terms. I also gave a simple formula to compute the number of way to evaluate the normal form of an identity-term, I manage to prove that for all $T, T' \in \mathbf{ORTrees}$:

$$T \leftrightharpoons T' \implies T \equiv T' \wedge \text{Functoriality Lemma} \implies \mathrm{RED}(T) \equiv \mathrm{RED}(T') \implies T \leftharpoonup T' \implies T \equiv T' \quad (29)$$

And also gave In this conclusion I will motivate why the dyslexic relation is still probably very important part to giving a full equivalence characterization, before talking about other proof techniques we searched for and offering a conclusion to my internship.

5.1 Categorical approach

TODO: (half a page)

For more insight, see Appendix E.

5.2 Conclusion

During this internship, I showed that the reduction of I-terms admits a simple combinatorial representation using ordered rooted trees from which leaves are removed (see Theorem 1). This representation allowed me to count the number of ways to reduce a term to its normal form (see Theorem 2), and also helped me study the reduction graphs of I-terms. I proved that reversing the left-to-right order uniformly at each depth creates terms with the same reduction graph (see Theorem 3), and revealed through the proof a very important "Functoriality Lemma". While trying to find some sort of converse, I proved the "Reconstruction Lemma": if two terms have the same reduction graph, then they correspond to the same unordered tree (see Theorem 4). This led me to try to prove a full equivalence between "having the same reduction graph" and satisfying the "Functoriality Lemma". I managed to show that the reduction graph can be seen as a quotient of the unordered reduction graph by the equivalence relation induced by the "Functoriality Lemma" (see Theorem 5). Unfortunately, the full equivalence is still left as a conjecture (see Conjecture).

To answer the title of this report, I hope I managed to convince the reader that studying reduction graphs of λ-terms is indeed quite difficult, even when restricted to such a seemingly trivial set of terms.

6 Appendixes

Appendix A: Bibliography

[Klo80] J. W. Klop, "Combinatory Reduction Systems," Doctoral dissertation, 1980a. [Online]. Available: https://eprints.illc.uva.nl/id/eprint/1876/1/HDS-33-Jan-Willem-Klop.text.pdf

[Zil84] M. V. Zilli, "Reduction Graphs in the Lambda Calculus," *Theoretical Computer Science*, vol. 29, no. 3, pp. 251–275, 1984, doi: 10.1016/0304-3975(84)90002-1.

[Lév76] J.-J. Lévy, "An Algebraic Interpretation of the Lambda-Beta-K-Calculus; and an Application of a Labelled Lambda-Calculus," *Theoretical Computer Science*, vol. 2, no. 1, pp. 97–114, 1976, doi: 10.1016/0304-3975(76)90009-8.

[EKP16] J. Endrullis, J. W. Klop, and A. Polonsky, "Reduction Cycles in Lambda Calculus and Combinatory Logic," in *Liber Amicorum Alberti: A Tribute to Albert Visser*, vol. 30, J. van Eijck, R. Iemhoff, and J. J. Joosten, Eds., in Tributes, vol. 30., London: College Publications, 2016, pp. 111–124.

[SH88] S. Sekimoto and S. Hirokawa, "One-Step Recurrent Terms in Lambda-Beta-Calculus," *Theoretical Computer Science*, vol. 56, pp. 223–231, 1988, doi: 10.1016/0304-3975(88)90079-5.

[BM22] H. P. Barendregt and G. Manzonetto, *A Lambda Calculus Satellite*. College Publications, 2022. [Online]. Available: https://www.collegepublications.co.uk/logic/mlf/?00035=

[Wel99] J. B. Wells, "Typability and Type Checking in System F Are Equivalent and Undecidable," *Annals of Pure and Applied Logic*, vol. 98, no. 1–3, pp. 111–156, 1999, doi: 10.1016/S0168-0072(98)00047-5.

[Gir71] J.-Y. Girard, "Une extension de l'interprétation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types," in *Proceedings of the Second Scandinavian Logic Symposium*, J. E. Fenstad, Ed., in Studies in Logic and the Foundations of Mathematics, vol. 63. Amsterdam: North-Holland, 1971, pp. 63–92. doi: 10.1016/S0049-237X(08)70843-7.

[CR36] A. Church and J. B. Rosser, "Some Properties of Conversion," *Transactions of the American Mathematical Society*, vol. 39, no. 3, pp. 472–482, 1936, doi: 10.1090/S0002-9947-1936-1501858-0.

[Bou26] C. Bourotte, "Identity Calculus." [Online]. Available: https://github.com/Cypooos/identity-calculus

[IL00] B. Intrigila and A. R. Laurenzi, "Two Problems on Reduction Graphs in Lambda Calculus," *Fundamenta Informaticae*, vol. 44, no. 1–2, pp. 133–144, 2000, doi: 10.3233/FUN-2000-441-206.

[Klo80] J. W. Klop, "Reduction Cycles in Combinatory Logic," in *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, J. P. Seldin and J. R. Hindley, Eds., London: Academic Press, 1980b, pp. 193–214. [Online]. Available: https://archive.org/details/tohbcurryessayso00edit

[Mul86] H. Mulder, "On a conjecture by J. W. Klop," 1986.

[Lév78] J.-J. Lévy, "Réductions correctes et optimales dans le lambda-calcul," Doctoral dissertation, 1978. [Online]. Available: https://pauillac.inria.fr/~levy/pubs/78phd.pdf

[Lév22] J.-J. Lévy, "Tracking Redexes in the Lambda Calculus." [Online]. Available: https://pauillac.inria.fr/~levy/pubs/22trackredex.pdf

[BW89] A. Björner and M. L. Wachs, "q-Hook length formulas for forests," *Journal of Combinatorial Theory, Series A*, vol. 52, no. 2, pp. 165–187, 1989, doi: https://doi.org/10.1016/0097-3165(89)90028-9.

[Bow] C. G. Bower, "Further Transformations of Integer Sequences."

[Moi97] A. de Moivre, "A Method of Raising an Infinite Multinomial to Any Given Power, or Extracting Any Given Root of the Same," *Philosophical Transactions of the Royal Society of London*, vol. 19, no. 230, pp. 619–625, 1697, doi: 10.1098/rstl.1695.0110.

6.1 Appendix B: Theorem 2

Proof of Theorem 2. By induction on $|T|$. If $|T| = 1$, then there is one reduction sequence, the empty one. Let us consider a tree $T \neq l$ with $|T| = n + 1$. Denote by r the root and by $T_1, ..., T_d$ the direct subtrees of the root. Let us first count the number of reduction sequences of T in terms of $(\#(T_i))_i$.

For all $i \leq d$, let $p^{(i)}$ be a reduction sequence of T_i. Remark that, for all $i \in [d]$, we have $|p^{(i)}| = |T_i|$. For every partitioning[8] of the set $[n]$ into $X_1 \sqcup \cdots \sqcup X_d$ such that $|X_i| = |T_i|$, we can create the reduction sequence $p = \langle p_1, ..., p_n \rangle$ defined by induction by letting p_i be the next element not yet chosen of $p^{(j)}$ such that $i \in X_j$ (such a choice of X_j is unique because we have a partition). By appending the root to p, we obtain a reduction sequence of T.

So, for every d-tuple of reduction sequences of the children and each partitioning of $[n]$ respecting the number of vertices of each subtree, we obtain a reduction sequence. This is a bijective mapping: let p be a reduction sequence. It ends with r, so let us ignore this node. We can extract the subsequence $p^{(i)}$ by splitting p according to which nodes occur in which T_i. This gives us both the partition and the subsequences.

For example, if we take for T_1 the reduction sequence $\langle a, b, c \rangle$ and for T_2 the sequence $\langle D, E \rangle$, for the partition $\{1, 3, 4\} \sqcup \{2, 5\}$, we place the reductions of T_1 in positions $\{1, 3, 4\}$ and the elements of T_2 in positions $\{2, 5\}$. This gives us the reduction sequence $\langle a, D, b, c, E \rangle$, to which we add the root at the end.

The *multinomial coefficient* [Moi97], for $x = x_1 + \cdots + x_k$, is defined as:

$$\binom{x}{x_1, ..., x_k} := \frac{x!}{x_1! \times \cdots \times x_k!} \tag{30}$$

It counts the number of partitionings of $[x]$ into $A_1 \sqcup \cdots \sqcup A_k$ such that $\forall i \in [k], |A_i| = x_i$. We therefore obtain the formula:

$$\#(T) = \binom{n}{|T_1|, ..., |T_d|} \prod_{i=1}^{d} \#(T_i). \tag{31}$$

Then, for every $i \leq d$, $|T_i| \leq n$, so we can apply the induction hypothesis to every T_i. Denote $n_i = |T_i|$. We have:

$$\begin{aligned}
\#(T) &= \binom{n}{n_1, ..., n_d} \prod_{i=1}^{d} \#(T_i) \\
&= \frac{n!}{n_1! ... n_d!} \prod_{i=1}^{d} \frac{n_i!}{\prod_{x \in V_{T_i}} |T_i(x)|} \\
&= \frac{n!}{\prod_{i=1}^{d} \prod_{x \in V_{T_i}} |T_i(x)|} \\
&= \frac{n!}{\prod_{x \in V_T \setminus \{r\}} |T(x)|} \\
&= \frac{(n+1)!}{\prod_{x \in V_T} |T(x)|} \\
&= \frac{|T|!}{\prod_{x \in V_T} |T(x)|}.
\end{aligned} \tag{32}$$

[8] A *partition of a set X in k parts* is a disjoint union $X_1 \sqcup ... \sqcup X_k$ such that $\bigcup_{i \in [k]} X_i = X$

6.2 Appendix C: Second and third part of the reconstruction lemma

We keep the same naming convention.

Finding RED(T_i). Now, since we know $n(T) = n(T')$, write $T = N(T_1, ..., T_d)$ and $T' = N(T'_1, ..., T'_d)$. We can map every leaf child of T to a leaf child of T' since $l(T) = l(T')$, so let us define the mapping for every non-leaf child of T. For every $i \in [d]$ such that T_i is not a leaf, define

$$T_{i\leftarrow l} \coloneqq N(T_1, ..., T_{i-1}, l, T_{i+1}, ..., T_d). \tag{33}$$

First, let us show that $\{T_{i\leftarrow l} : i \leq n \wedge T_i \neq l\} = \max_{\preceq}\{t \in \mathrm{RED}(T) \mid l(t) = l(T) + 1 \wedge n(t) = n(T)\}$ where the right-hand side is the set of maximal elements for $\preceq$ in $\{t \in \mathrm{RED}(T) \mid l(t) = l(T) + 1 \wedge n(t) = n(T)\}$:

- $\Longrightarrow$: Suppose that $T_{i\leftarrow l}$ is not maximal in $\{t \in \mathrm{RED}(T) \mid l(t) = l(T) + 1 \wedge n(t) = n(T)\}$. Then there exists $t \in \mathrm{RED}(T)$ with $T_{i\leftarrow l} \preceq t$, $t \neq T_{i\leftarrow l}$, $l(t) = l(T) + 1$ and $n(t) = n(T)$. Since $n(t) = n(T)$, no child of the root has been removed, so we can write $t = N(t_1, ..., t_d)$, with each t_j a reduct of T_j. Since $T_{i\leftarrow l} \preceq t$, for every $j \neq i$, t_j must reduce to T_j. But t_j is already a reduct of T_j, hence $t_j = T_j$. Therefore, the only possible difference is in the i-th component. If $t_i \neq l$, then $l(t) = l(T)$, which contradicts $l(t) = l(T) + 1$. Hence $t_i = l$, so $t = T_{i\leftarrow l}$, a contradiction.
- $\Longleftarrow$: Let $t \in \max_{\preceq}\{t \in \mathrm{RED}(T) \mid l(t) = l(T) + 1 \wedge n(t) = n(T)\}$. Write $t = N(t_1, ..., t_n)$. Since $n(t) = n(T)$, no child of the root has been removed, so each t_i is a reduct of T_i. Since $l(t) = l(T) + 1$, exactly one non-leaf child of T has been reduced to a leaf. Let i be its index. If there exists $j \neq i$ such that $t_j \neq T_j$, then replacing t_j by T_j gives a strictly larger element that still satisfies $l(t) = l(T) + 1$ and $n(t) = n(T)$, contradicting the maximality of t. Therefore, for all $j \neq i$, we have $t_j = T_j$, and hence $t = T_{i\leftarrow l}$.

Because of this characterization, we have that $\{T_{i\leftarrow l} : i \leq n \wedge T_i \neq l\}$ is preserved under isomorphism, as it can be defined only using $\preceq$, $l(t)$ and $n(t)$.

For every $t_1, t_2 \in \mathrm{RED}(T)$, consider the interval from t_1 to t_2 in $\mathrm{RED}(T)$ defined by

$$[t_1, t_2] \coloneqq \mathrm{RED}(T)[\{t' \in \mathrm{RED}(T) \mid t_1 \preceq t' \preceq t_2\}]. \tag{34}$$

We will prove that $\mathrm{RED}(T_i) \equiv [T_{i\leftarrow l}, T]$: morally, this is because the only part that can move is the i-th component, since all the other ones are already maximal. Define the isomorphism

$$\varphi : \mathrm{RED}(T_i) \longleftrightarrow [T_{i\leftarrow l}, T] \tag{35}$$

by

$$\varphi(u) = N(T_1, ..., T_{i-1}, u, T_{i+1}, ..., T_n). \tag{36}$$

The map φ is a bijection by definition of $[T_{i\leftarrow l}, T]$, and $(u \to v)$ has multiplicity m in $\mathrm{RED}(T_i)$ iff $(\varphi(u) \to \varphi(v))$ has multiplicity m in $[T_{i\leftarrow l}, T]$. Therefore, while we cannot know exactly which child T_i with $T_i \neq l$ maps to which child of T', we know that one must map, and that is enough to show the existence of θ.

To define θ, proceed as follows:

- Map every leaf child of T to a leaf child of T' (possible since $l(T) = l(T')$).
- For every non-leaf child, map T_i to the T'_j such that $T'_{j\leftarrow l} = \Psi(T_{i\leftarrow l})$: since $\{T_{i\leftarrow l} : T_i \neq l\}$ is preserved under isomorphism, we have that Ψ is a bijection between $\{T_{i\leftarrow l} : T_i \neq l\}$ and $\{T'_{i\leftarrow l} : T'_i \neq l\}$. Since Ψ preserves the structure between any pair of mapped elements, we finally have

$$\mathrm{RED}(T_i) \equiv [T_{i\leftarrow l}, T] \equiv [\Psi(T_{i\leftarrow l}), T'] \equiv \left[T'_{\theta(i)\leftarrow l}, T'\right] \equiv \mathrm{RED}\left(T'_{\theta(i)}\right). \tag{37}$$

We therefore have proven the second part of the reconstruction lemma.

Third Part. [9] We will show here that θ can be assumed to be either $\theta = \mathrm{Id}_d$ or $\theta = \mathrm{Rev}_d$. We will proceed in an analog way as before. We first generalize a bit the notations of $T_{i\leftarrow l}$.

For every $I \subseteq [d]$ such that $\forall i \in I, T_i \neq l$, define T_I to be the tree $N\left(T_1^{(I)}, ..., T_d^{(I)}\right)$ with $T_i^I = T_i$ if $i \in I$ and $T_i^I = l$ otherwise. I represent the set of children of T we keep. For example, we have $T_{\{1,...,i-1,i+1,...,d\}} = T_{i\leftarrow l}$

[9] This part was done in collaboration with Péa

Let $I_{\text{pos}} = \{i \in [d] \mid T_i \neq l\}$ the set of non-leaves children. We can now prove that for all $k \leq |I_{\text{pos}}|$, we have

$$\left\{T_I : I \subseteq I_{\text{pos}} \wedge |I| = k\right\} = \max_{\preceq}\{t \in \text{RED}(T) \mid n(t) = n(T) \wedge l(t) = n(T) - k\}. \tag{38}$$

Denote $M = \{t \in \text{RED}(T) \mid n(t) = n(T) \wedge l(t) = n(T) - l(T) - k\}$.

- $\Longrightarrow$: We prove that every T_I is maximal for $\preceq$ in M. It is indeed in it since it has the correct number of leaves and children on the root, and any bigger element would be bigger in one component which is impossible.
- $\Longleftarrow$: Given a maximal element $t \in M$, we know that it has $n(T) - k$ leaves so only k elements are not leaves, and since it is maximal they must be completed T_i. So by writing $t = N(t_1, ..., t_d)$, we have that $t = T_I$ for $I = \{i \in [d] \mid t_i \neq l\}$.

We then show that $\theta \in \{\text{Id}_d, \text{Rev}_d\}$ by studying the information given by $\left\{T_{\{i\}} : i \in [d] \mid T_i \neq l\right\}$ and $\left\{T_{\{i,j\}} : 0 < i < j \leq d \mid T_i \neq l \neq T_j\right\}$.

For every $i \in [d]$ with $T_i \neq l$, there are 2 arrows going from T_i to $\{t \in \text{RED}(T) \mid n(t) < n(T)\}$: the one in which we reduce the first part $N\left(\underline{l, ..., l}, T_i, l, ..., l\right)$ and the other in which we reduce the part $N\left(l, ..., l, T_i, \underline{l, ..., l}\right)$. To every $t \in \left\{T_{\{i\}} : i \in [d] \mid T_i \neq l\right\}$, we can therefore compute :

- $\left[E_{n(T)}; t\right]$ the reduction graph of t (for the same reasoning as the previous part, as only the ith component can change)
- And two numbers $\{|i - 1, d - i|\}$, corresponding to 2 possibles place for t: t is either at position i or $d - i$. If the multiset contains only one entry, it means that the other is $0 : t$ is on the border, and we can morally add a 0 to the multiset.

Thus every non-leaf children is individually placed either at its original position (if we choose i from $i - 1$) or at its mirror position (if we choose $d - i$ from $d - i$). We can therefore suppose that $\theta(i) \in \{i, d - i + 1\}$, as $\Psi\left(T_{\{i\}}\right)$ must be map to a $T'_{\{i\}}$ or $T'_{\{d-i+1\}}$ with $\text{RED}\left(T_{\{i\}}\right) \equiv \text{RED}\left(\Psi\left(T_{\{i\}}\right)\right)$. It remains to show that these choices are coherent through all T_i, and for this we will consider the set of the form $T_{\{i,j\}}$.

For every $0 < i < j \leq d$ with both $T_i \neq l \neq T_j$, there are 3 arrows going from $T_{\{i,j\}}$ to $\{t \in \text{RED}(T) \mid n(t) < n(T)\}$: such a $T_{\{i,j\}}$ can be written of the form $N\left(\underline{l, ..., l}, T_i, \underline{l, ..., l}, T_j, \underline{l, ..., l}\right)$, with the 3 reduction underlined, of respective arity $i - 1, j - i - 1, n - j - 1$. To every $t \in \left\{T_{\{i,j\}} : 0 < i < j \leq d \mid T_i \neq l \neq T_j\right\}$, we can therefore compute :

- $[t_1; t]$ the reduction graph of T_i and $[t_2; T]$ the reduction graph of T_j, by letting $\{t_1, t_2\} = \max_{\preceq}\{t' \in \text{RED}(T) \mid l(t') = n(T) - 1 \wedge t' \preceq t\}$ the only two maximal trees with 1 more leaf that t that are below t.
- And the multiset $\{|i - 1, j - i - 1, n - j|\}$ of arity 3. Simmilarelly, if we are missing one or more element from this multiset, we need to add as many 0 as necessary.

Suppose by constradiction that a $\theta : [d] \longleftrightarrow [d]$ maps a $\theta(i) = d - i + 1$ and a $\theta(j) = j$ (i.e. they are not coherent), with neither $i = \frac{d}{2}$ nor $j = \frac{d}{2}$ (because otherwise the mapping of the two would be coherent). We will show that we can detect such a case.

Consider

Extracting more information. Intuitively, this "reconstruction lemma" managed to extract all the possible information out of a part of a reduction graph. We can easily prove that the induced reduction graph of all elements with $n(T)$ children at the root is exactly the graph product[10] of the reduction graphs of the non-leaf children:

$$\text{RED}(T)[\{t \in \text{RED}(T) : n(t) = n(T)\}] \equiv \text{RED}\left(T_{i_1}\right) \times \text{RED}\left(T_{i_2}\right) \times \cdots \times \text{RED}\left(T_{i_k}\right). \tag{39}$$

for $\{i_1, ..., i_k\} = \{i \in [n(T)] \mid T_i \neq l\}$ with $i_1 < \cdots < i_k$. If we want to extract more information from the reduction graph, we have to look at vertices in the induced graph of $\{t \in \text{RED}(T) \mid n(t) < n(T)\}$.

The isomorphism is just the map sending $(t_1, ..., t_k)$ to the tree obtained by putting each t_j back in the i_j -th position and adding back the leaves at the root:

[10]Here, the graph product $G_1 \times G_2$ is the graph with vertices $V = V_{G_1} \times V_{G_2}$ and edges $(x, y) \to^k (x', y') \iff \left(x \to^k x' \wedge y = y'\right) \vee \left(x = x' \wedge y \to^k y'\right)$.

$$\begin{aligned}\Psi : \prod_{1\leq j\leq k} \mathrm{RED}\big(T_{i_j}\big) &\longrightarrow \mathrm{RED}(T)[\{t \in \mathrm{RED}(T) : n(t) = n(T)\}] \\ (t_1, ..., t_k) &\mapsto N\Bigg(\underbrace{l, ..., l}_{i_1-1 \text{ times}}, t_1, \underbrace{l, ..., l}_{i_2-i_1-1 \text{ times}}, t_2, ..., t_k, \underbrace{l, ..., l}_{n(T)-i_k \text{ times}}\Bigg).\end{aligned} \tag{40}$$

It is bijective, and by definition of the graph product, there is an edge $t_i \to^m t'_i$ in one of the graphs iff there is an edge

$$\Psi(t_1, ..., t_i, ..., t_k) \to^m \Psi(t_1, ..., t'_i, ..., t_k). \tag{41}$$

6.3 Appendix D: Construction of $\mathrm{RED}(T)$ as a quotient

Recall that, for $S, S' \in \mathcal{S}(T)$, we defined $S \sim_T S' \iff T[S] = T[S']$. For $S \in \mathcal{S}(T)$, denote by $[S]_\sim$ its equivalence class in $\sim_T$.

Theorem 5. For every $T \in \mathbf{ORTrees}$, we have $\mathcal{H}(T)/\sim_T \equiv \mathrm{RED}(T)$.

Proof. Let's first show that the quotient graph is well-defined. Let $S, S' \in \mathcal{S}(T)$ such that $S \underset{T}{\sim} S'$, and C be an equivalence class of $\underset{T}{\sim}$: we show that S and S' have the same number of outgoing edges toward C. Notice that since the hasse graph $\mathcal{H}(T)$ only has edges of arity 1, it suffice to show that

$$|\{c \in C \mid (S \to c) \in \mathcal{H}(T)\}| = |\{c \in C \mid (S' \to c) \in \mathcal{H}(T)\}| \tag{42}$$

We first show that $|\{c \in C \mid (S \to c) \in \mathcal{H}(T)\}| \leq |\{c \in C \mid (S' \to c) \in \mathcal{H}(T)\}|$. Denote $l_1, ..., l_n$ the n leaves of S such that $\forall i \in [n], S - l_i \in C$. Since $T[S] = T[S']$, there is an **ORTrees**-morphism $\varphi : V_{T[S]} \hookleftarrow V_{T[S']}$. For every $i \in [n]$ consider $l'_i := \varphi(l_i)$. those are also n leaves of S', and $\forall i \in [n], T[S' \setminus l'_i] = T[\varphi(S \setminus l_i)] = T[S \setminus l_i]$, so $S' \setminus l'_i \equiv_T S \setminus l_i$ and $S' \setminus l'_i \in C$. Therefore $\{l'_1, ..., l'_n\}$ denote n edges from S' to C, so $|\{c \in C \mid (S \to c) \in \mathcal{H}(T)\}| \leq |\{c \in C \mid (S' \to c) \in \mathcal{H}(T)\}|$. By the same argument the other way, we have an equality. The quotient is therefore well-defined.

Let's now show that ${}^{\mathcal{H}(T)}\!/_\sim \equiv \mathrm{RED}(T)$. We define

$$\begin{aligned} \Psi : \mathcal{H}(T)\!/_\sim &\leftrightarrow \mathrm{RED}(T), \\ [S]_\sim \quad &\mapsto T[S]. \end{aligned} \tag{43}$$

The map is well-defined as $S \sim_T S' \implies T[S] = T[S']$. It is injective, since $T[S] = T[S']$ implies $S \sim_T S'$, and therefore $[S]_{\underset{T}{\sim}} = [S']_{\underset{T}{\sim}}$. It is also surjective since every vertex of $\mathrm{RED}(T)$ is a reduct of T, and can therefore be written $T[S]$ for some $S \in \mathcal{S}(T)$.

It remains to show that Ψ preserves the multiplicity of the edges. Let $C = [S]$ and $C' = [S']$ be two equivalence classes.

- Suppose that $C \to^k C'$ for $k \geq 1$. There is therefore k distincts leaves $l_1, ..., l_k \in S$ such that $\forall i \in [k], S - \{l_i\} \in C'$, so k distincts leaves going to the same $T[S - \{l_i\}]$, so there is at least k edges between $T[S] \to T[S - \{l_1\}]$.
- Suppose that there is an arity $k \geq 1$ edges between two $u, v \in \mathrm{RED}(T)$. By definition of $\mathrm{RED}(T)$, it means that there is k leaves $l_1, ..., l_k$ such that $\forall i \in [k], u - l_i$ all go to v as an **ORTrees**. Writing $u = T[S]$, we therefore have $\forall i \in [k], S - \{l_i\}$ denote the same **ORTrees**, so that they all go to the same $T[S - \{l_1\}]$ and therefore $\forall i, j \in [k], S - \{l_i\} \sim S - \{l_j\}$, and therefore there is at least k edges between S and $C' = [S - \{l_1\}]_\sim$.

Therefore Ψ is a bijection preserving every edge and its multiplicity, hence

$$\mathcal{H}(T)\Big/_{\underset{T}{\sim}} \overset{\Psi}{\equiv} \mathrm{RED}(T). \tag{44}$$

□.

6.4 Appendix E: Categorical Approach

TODO: as much as you want