Notes on Counting and Classification of Relational Trees

Anav Sood and Patrick Revilla

November 2019

1 Definitions and Notation

1.1 Strings and Relations

Definition 1.1. A **relation** is a sequence $r = (r_1, \dots, r_\ell) \in \{-1, +1\}^\ell$. Let ℓ be the **length** of the relation. For a relation r we let $-r := (-r_1, \dots, -r_\ell)$, $\overline{r} := (-r_1, r_2, \dots, (-1)^{\ell+1} r_\ell)$, and $\tilde{r} := (r_\ell, \dots, r_1)$.

Definition 1.2. A **string** $s \in \{-1, +1\}^*$ is a sequence of ± 1 of arbitrary length. We will use $-s$, $\overline{s}$, and $\tilde{s}$ as in Definition 1.1.

Definition 1.3. Suppose s is a string of length n and s' is a string of length m. Then the **concatenation** of s and s' is $ss' = (s_1, \dots, s_n, s'_1, \dots, s'_m)$. We write the concatenation of a string s with ± 1 as $s^+ := (s_1, \dots, s_n, 1)$ and $s^- := (s_1, \dots, s_n, -1)$.

1.2 Applying Relations

Definition 1.4. Consider relation r of length ℓ and a string s of length n. Then for a fixed index k s.t. $k + \ell - 1 \leq n$, we say that r is **applicable** to s from indices k to $k + \ell - 1$ if for all $i \leq \ell$,

$$r_i s_{k+i-1} = r_1 s_k$$

That is, either r or $-r$ "appears" in the string s starting at k. We will say that relation is "applicable at k", "applicable starting at k", or "applicable ending at $k + \ell - 1$".

Definition 1.5. For any relation r and index k let $\alpha_r^{(k)}$ be a map between strings where r is applicable at k given by $\alpha_r^{(k)} : s \mapsto s'$, where

$$s'_i = \begin{cases} -s_i & k \leq i \leq k + \ell - 1 \\ s_i & \text{otherwise} \end{cases}$$

We refer to the set of indices $\{k, \dots, k + \ell - 1\}$ as the **line of action** of $\alpha_r^{(k)}$. The action of mapping s by $\alpha_r^{(k)}$ is referred to as an **application** of the relation at k (or starting at k, or ending at $k - \ell + 1$) to s. Effectively, applying the relation flips the sign of any element in its the line of action.

Example 1.6. Consider the case where $r = (+1, -1, -1)$ and $s = (+1, +1, -1, -1, +1)$. Then we can apply the relation at $k = 2$,

$$\alpha_r^{(2)}(s) = (+1, \mathbf{-1}, \mathbf{+1}, \mathbf{+1}, +1)$$

where the bold terms have been "flipped" by r.

Definition 1.7. We consider two string s, s' equivalent under r, denoted $s \equiv_r s'$, if there are some indices $k_1, \ldots, k_m$ such that $(\alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)})(s) = s'$ We allow for $m = 0$, so $s \equiv_r s$ always.

Example 1.8. Consider $r = (+1, -1, -1)$ and $s = (+1, -1, +1, -1, -1), s' = (-1, +1, +1, +1, +1)$, then

$$(\alpha_r^{(1)} \circ \alpha_r^{(3)})(s) = s'$$

so $s \equiv_r s'$

Definition 1.9. For a string s and relation r, define the **equivalence class** $[s]_r = \{s' | s \equiv_r s\}$. The size of an equivalence class $|[s]_r|$ will be known as its **frequency**. We may denote this as $\text{freq}_r(s)$.

To convince the reader these are good definitions, we will note that r is indeed an equivalence relation on strings of length n:

- Reflexivity: We have $s \equiv_r s$ from Definition 1.7
- Symmetry: If $(\alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)})(s) = s'$, then $(\alpha_r^{(k_1)} \circ \cdots \circ \alpha_r^{(k_m)})(s')$.
- Transitivity: If $(\alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)})(s) = s'$ and $(\alpha_r^{(k'_m)} \circ \cdots \circ \alpha_r^{(k'_1)})(s') = s''$, then

$$(\alpha_r^{(k'_m)} \circ \cdots \circ \alpha_r^{(k'_1)} \circ \alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)})(s) = s''$$

2 Results on Applying Relations

Definition 2.1. We consider the application of the relation r given by $\alpha_r^{(k)}(s)$ to be **well defined**, if s is in the domain of the map $\alpha_r^{(k)}$ as indicated by Definition 1.5. Similarly, letting $A := \alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)}$ be a sequence of such applications, we consider $A(s)$ to be well defined if each sequential application is well defined. We will refer to A as a composition of applications of the relation or a sequential application of the relation. The number of applications in A is its **size** (in this case m). If A is **empty**, then it has size zero, and is simply the identity map.

Lemma 2.2. Consider a relation r, a string s, and distinct indices i, j . Then both $(\alpha_r^{(i)} \circ \alpha_r^{(j)})(s)$ and $(\alpha_r^{(j)} \circ \alpha_r^{(i)})(s)$ are well defined if and only if $\alpha_r^{(i)}(s)$, $\alpha_r^{(j)}(s)$ are well defined and have disjoint lines of action.

Proof. The reverse direction is obvious. To see the forward direction, first note that both $(\alpha_r^{(i)} \circ \alpha_r^{(j)})(s)$ and $(\alpha_r^{(j)} \circ \alpha_r^{(i)})(s)$ well defined immediately implies that $\alpha_r^{(i)}(s)$, $\alpha_r^{(j)}(s)$ are well defined. We thus show that assuming $\alpha_r^{(i)}(s)$, $\alpha_r^{(j)}(s)$ are well defined, if the lines of action are not disjoint then both $(\alpha_r^{(i)} \circ \alpha_r^{(j)})(s)$ and $(\alpha_r^{(j)} \circ \alpha_r^{(i)})(s)$ are not well defined.

Suppose the lines of action are not disjoint. Take the index k common to both. WLOG take $i < j$. From $i < j$ and k in the line of action of $\alpha_r^{(j)}$ we have

$$(\alpha_r^{(j)}(s))_i = s_i,\ (\alpha_r^{(j)}(s))_k = -s_k$$

Since k is in the line of action of $\alpha_r^{(i)}$, for $\alpha_r^{(i)}(s)$ to be well defined we need that $r_{k'}s_k = r_1 s_i$ for $k' := k - i + 1$. Combining this with the above, we see that

$$r_{k'}(\alpha_r^{(j)}(s))_k = -r_{k'}s_k,\ r_1(\alpha_r^{(j)}(s))_i = r_1 s_i$$

However this exactly shows that letting $s' := \alpha_r^{(j)}(s)$, we cannot have that $\alpha_r^{(i)}(s')$ is well defined, and we are done.

□

In the case that both $(\alpha_r^{(i)} \circ \alpha_r^{(j)})(s)$ and $(\alpha_r^{(j)} \circ \alpha_r^{(i)})(s)$ are well defined, it is clear that $(\alpha_r^{(i)} \circ \alpha_r^{(j)})(s) = (\alpha_r^{(j)} \circ \alpha_r^{(i)})(s)$. In this case, we say that $\alpha_r^{(i)}$ and $\alpha_r^{(j)}$ commute.

Lemma 2.3. For a relation r and strings s, s', $s \equiv_r s'$ if and only if $\tilde{s} \equiv_{\tilde{r}} \tilde{s}'$

Proof. Suppose s and s' are of length n. To see the forward direction, consider the permutation on indices that sends $\sigma : i \mapsto n - i + 1$, it is easy to verify that if

$$(\alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)})(s) = s'$$

then

$$(\alpha_{\tilde{r}}^{(\sigma(k_m))} \circ \cdots \circ \alpha_{\tilde{r}}^{(\sigma(k_1))})(\tilde{s}) = \tilde{s}'$$

The same proof applies in the reverse direction. □

Theorem 2.4. Suppose we have strings s, p, q such that p, q are the same length. Then for some relation r

- $sp \equiv_r sq$ if and only if $p \equiv_r q$
- $ps \equiv_r qs$ if and only if $p \equiv_r q$

Proof. We show $sp \equiv_r sq$ if and only if $p \equiv_r q$. To see that $ps \equiv_r qs$ if and only if $p \equiv_r q$, note that this is an equivalent problem to $\tilde{s}\tilde{p} \equiv_{\tilde{r}} \tilde{s}\tilde{q}$ if and only if $\tilde{p} \equiv_{\tilde{r}} \tilde{q}$ by Lemma 2.3

For $sp \equiv_r sq$ if and only if $p \equiv_r q$, the reverse direction is obvious. To see the forward direction suppose that $sp \equiv_r sq$. Let L be the length of s, l be the length of p and q, and ℓ be the length of r. We know that for some set of indices $k_1, \ldots, k_m$

$$(\alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)})(sp) = sq$$

In the case that all $k_1, \ldots, k_m > L_1$ we have

$$(\alpha_r^{(k_m - L)} \circ \cdots \circ \alpha_r^{(k_1 - L)})(p) = q$$

and we are done. Alternatively, suppose that there is some index $\leq L$. We let $A := \alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)}$. We then claim there is always A^* of size strictly less than m such that $A^*(sp) = sq$. Prior to showing this claim we show that it implies the result. Let $A^{(0)} := A$. Then so long as $A^{(i-1)}$ has some application of the relation at index $\leq L$ we can find $A^{(i)}$ of strictly smaller size such that $A^{(i)}(sp) = sq$. Since the size strictly decreases on each iteration, either we find some non-empty $A^{(i)}$ such that all the applications of the relation are at index $> L$ or we iterate until $A^{(i)}$ is empty, so $sp = sq$. Either way the claim implies the result.

To see the claim, suppose there is some index $\leq L$, and take i_1 to represent the value of the smallest index the relation is applied in A (so $i_1 \leq L$). There must be an even number of applications $\alpha_r^{(i_1)}$, as otherwise we contradict the fact that $(sp)_{i_1} = s_{i_1} = (sq)_{i_1}$. We then see that we can write

$$A = \cdots \underbrace{\alpha_r^{(i_1)} \circ \cdots \circ \alpha_r^{(i_1)} \circ \cdots \circ \alpha_r^{(i_1)} \circ \cdots \circ \alpha_r^{(i_1)}}_{\text{even number of } \alpha_r^{(i_1)}} \cdots$$

Consider the innermost pair of $\alpha_r^{(i_1)}$ and denote the sequence of relations applied between them $M^{(1)}$. Then each application of the relation in $M^{(1)}$ starts at index $> i_1$. In particular, we can write A as

$$A = \cdots \alpha_r^{(i_1)} \circ M^{(1)} \circ \alpha_r^{(i_1)} \cdots$$

Let $i_2 > i_1$ be the smallest index the relation is applied in $M^{(1)}$. There are two cases:

- Case One: Suppose that the line of action of $\alpha_r^{(i_2)}$ does not intersect with that of $\alpha_r^{(i_1)}$. In this case, none of the applications of the relation in $M^{(1)}$ have a line of action that intersects that of $\alpha_r^{(i_1)}$. Thus by Lemma 2, we can commute $M^{(1)}$ out of the innermost pair of $\alpha_r^{(i_1)}$ when computing $A(s)$. From this it follows that

  $$A^* = \cdots M^{(1)} \cdots$$

  is the composition of $m - 2$ applications of the relation such that $A^*(s) = s'$.

- Case Two: Suppose it is not the case that the two lines of action are disjoint. Since the application of $M^{(1)}$ never flips value at index i_1, it must be the case that there are an even number of $\alpha_r^{(i_2)}$ in $M^{(1)}$ as otherwise the sequential application of $\alpha_r^{(i_1)} \circ M^{(1)} \circ \alpha_r^{(i_1)}$ cannot be well defined. Thus we can take the innermost pair, and define $M^{(2)}$ to be the sequence of relations applied between them. Now iteratively repeat the reasoning above with the new knowledge that

  $$A = \cdots \alpha_r^{(i_2)} \circ M^{(2)} \circ \alpha_r^{(i_2)} \cdots$$

  where each application of the relation in $M^{(2)}$ starts at index $> i_2$

On the pth iteration of the algorithm we either find ourselves in Case One, in which case we are done, or Case Two in which case we iterate again. Since $i_1 < i_2 < \ldots i_p$, if we manage to reach the $(L + l) - \ell + 1$st iteration, then it must be the case that $M^{((L+l)-\ell+1)}$ is empty as no application of the relation can happen at an index $> L + l - \ell + 1$. In this situation we are certainly in Case One. Thus at some finite iteration we will be in Case One, and we will find a suitable A^*. □

3 Counting Equivalence Classes

With some groundwork now laid down, we start to wonder, what kind of equivalence classes do these relations create? How many?. For a relation r will define $N_n(r)$ (or N_n when r is unambiguous) to be the number of equivalence classes the relation induces on binary strings of length n. We'll take $N_0 = 1$ to represent the equivalence class of the empty string. We will let $N_n^f(r)$ (or N_n^f when r is unambiguous) represent the number of equivalence classes of binary strings of length n with frequency f. Lemma 2.3 already tells us that r and $\tilde{r}$ induce the same values N_n^f for all n, f. What else can we say? We start with a discussion of N_n that is rather surprising.

Lemma 3.1. For a relation r of length ℓ, the number of equivalence classes induced on binary strings of length n is given by

$$N_n = \begin{cases} 2^n & n < \ell \\ 2N_{n-1} - N_{n-\ell} & n \geq \ell \end{cases}$$

Proof. The relation cannot be applied to strings of length $n < \ell$, so for binary strings of such length we find 2^n equivalence classes each of size 1.

Now suppose $n \geq \ell$. Consider the equivalence classes induced on binary strings of length $n-1$, $[p_1]_r, \ldots, [p_{N_{n-1}}]_r$. All the resulting equivalence classes for level n can be found in the list $[p_1^-]_r, [p_1^+]_r, \ldots, [p_{N_{n-1}}^-]_r, [p_{N_{n-1}}^+]_r$ but we have no guarantee that each element in the list is unique (some may be equal to one another). Theorem 2.4 does garauntee that no more than two of the above can be equal to one another, and if such a pair exists it must be of the form $[p_i^+]_r, [p_j^-]$, $i \neq j$. Thus if we define the set

$$S = \{\{i, j\} : [p_i^+]_r = [p_j^-], i \neq j\}$$

It follows that $N_n = 2N_n - |S|$. We claim that there exists a bijection between S and the equivalence classes induced on binary strings of length $n - \ell$, so $|S| = N_{n-\ell}$.

To see this, consider $\{i, j\} \in S$ so that $[p_i^+]_r = [p_j^-]$. Note that this implies that $p_i^+ \equiv_r p_j^-$, so there is a sequence of application of relations such that

$$(\alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)})(p_i^+) = p_j^-$$

In particular, since p_i^+ ends in $+1$ and p_j^- ends in -1 at least one of these relations must be applied such that it ends at n, otherwise we would have $(p_i^+)_n = (p_j^-)_n$. Thus there must be some string representative t that is equivalent to $[p_i^+]_r = [p_j^-]$ that ends in the relation r, and can be written

$$t = (t_1, t_2, \ldots, t_{n-\ell}, r_1, \ldots, r_\ell)$$

We define the map f which sends $f : \{i, j\} \mapsto [(t_1, t_2, \ldots, t_{n-\ell})]_r$. Theorem 2.4 guarantees that this map is well-defined and injective. To see that the map is surjective consider any equivalence class $[w]_r$ of length $n - \ell$ binary strings. WLOG assume r ends in $+1$ so that we necessarily have

$$(w_1, w_2, \ldots, w_{n-\ell}, r_1, \ldots, r_\ell) \in [p_i^+]_r$$

$$(w_1, w_2, \ldots, w_{n-\ell}, -r_1, \ldots, -r_\ell) \in [p_j^-]_r$$

For some i, j $i \neq j$ (again, Theorem 2.4 garauntees that $i \neq j$). By construction $f : \{i, j\} \mapsto [w]_r$, giving us surjectivity. ☐

Definition 3.2. For any non-negative integer m, the *m-bonacci numbers* are a sequence $\{F_i^m\}_{i \in \mathbb{Z}}$ defined as follows. When $m = 0$, we set $F_1^0 = 1$ and $F_i^0 = 0$, $i \in \mathbb{Z}/\{1\}$. For $m > 0$ we define the sequence by the recursion

$$F_i^m = \begin{cases} 0 & i < 1 \\ 1 & i = 1 \\ \sum_{j=i-m}^{i-1} F_j^m & i > 1. \end{cases}$$

One can easily verify that $F_1^m = 1$, and $F_i^m = 2^{i-1}$ for $1 < i \leq m+1$.

Theorem 3.3. For a relation r of length ℓ, we have $N_n = \sum_{i=0}^{n+1} F_i^{\ell-1}$.

Proof. This follows directly from Lemma 3.1. The statement is obvious when $\ell = 1$ so we only consider $\ell > 1$. First we note that for $0 \leq n < \ell$, $N_n = 2^n$ and also

$$\begin{aligned} \sum_{i=0}^{n+1} F_i^{\ell-1} &= 1 + \sum_{i=0}^{n-1} 2^i \\ &= 2^n \end{aligned}$$

Using this we can prove our desired result through strong induction. Assume as our strong inductive hypothesis that for some $k \geq \ell - 1$, $N_q = \sum_{i=0}^{q+1} F_i^{\ell-1}$ for all $q \leq k$. From Lemma 3.1 we have $N_{k+1} = 2N_k - N_{k-\ell+1}$. Applying the hypothesis we see

$$\begin{aligned} N_{k+1} &= 2\sum_{i=0}^{k+1} F_i^{\ell-1} - \sum_{i=0}^{k-\ell+2} F_i^{\ell-1} \\ &= \sum_{i=0}^{k+1} F_i^{\ell-1} + \sum_{i=k-\ell+3}^{k+1} F_i^{\ell-1} \\ &= (\sum_{i=0}^{k+1} F_i^{\ell}) + F_{k+2}^{\ell-1} \\ &= \sum_{i=0}^{k+2} F_i^{\ell-1} \end{aligned}$$

as desired.

□

4 Relational Trees and Isomorphisms

4.1 Definitions and Examples

Definition 4.1. For any relation r we can define the **relational tree** $T_r = (V, E)$ to be the directed graph

$$V = \{[s]_r | s \in \{-1, +1\}^*\}, E = \{([s]_r, [s^+]_r)\} \cup \{([s]_r, [s^-]_r)\}$$

We will refer to the vertex $[\{\}]_r$ the **root**, as it corresponds to the root in a traditional binary tree.

Definition 4.2. The n**th level** of the tree is given by the set of vertices $T_r^{(n)} := \{[s]_r \in T_r \| s| = n\}$.

Note, the relational trees are inifinitely deep. For notation, let $T_r^{(\leq n)}$ denote the first n levels of a relational tree.

There are several ways to view the tree T_r (or $T_r^{(\leq n)}$) geometrically, which are appealing in different situations. When representing the tree, we use $+$ and $-$ instead of $+1$ and -1, for concision. For certain "nice" relations, T_r physically looks like a binary tree with branches fused together. In this example, we consider $r = (+1, -1, -1)$ and show $T_r^{(\leq 4)}$.

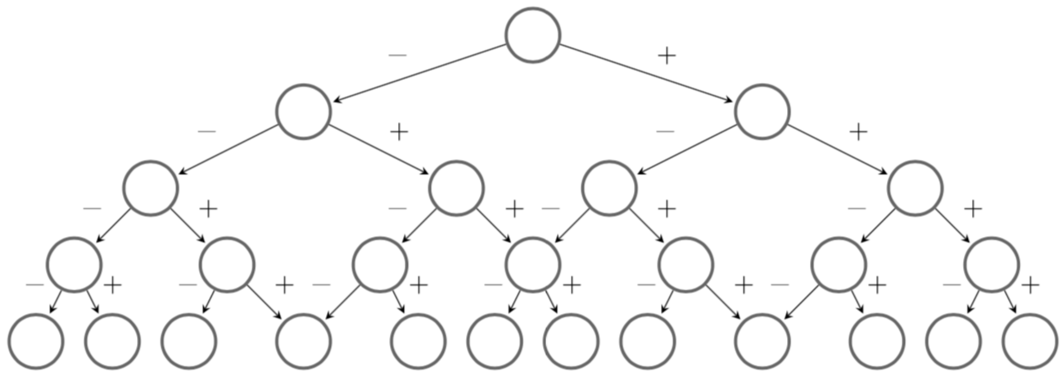

Figure 1: r-relational tree where $r = (+1, -1, -1)$

On the other hand, other relations do not visualize so clearly. However, they can still be described as a fused tree, by simply identifying vertices appropriately. For example, here is a visualization of $T_r^{(\leq 4)}$ for $r = (1, 1, 1)$, where we have color coded vertices that are identified (i.e. the same vertex) under our relation.

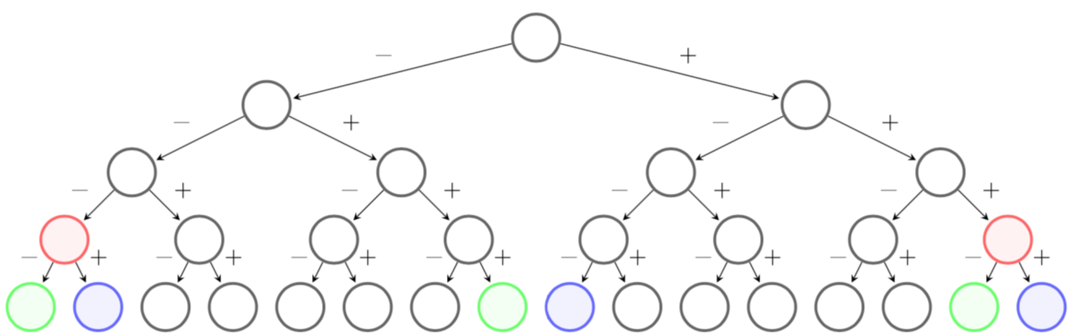

Figure 2: r-relational tree where $r = (+1, +1, +1)$

We saw above that relation of the same length induce the same number of equivalence classes at level n, but do they "connect" them in the same way? In essence, we are wondering which of these trees are isomoprhic.

5 Isomoprhims of Relational Trees

Recall that a graph isomorphism is a bijection between the vertices of the two graphs, that preserves edges (and direction) between them. The main question we want to investigate is: for what r_1, r_2 do we have $T_{r_1} \cong T_{r_2}$? To do so, we define the following function:

Definition 5.1. The **common ancestor** function is defined for a fixed relation r and two equivalance classes of strings, given by

$$CA_r([s_1]_r, [s_2]_r) := \max\{i | \exists x \in [s_1]_r, y \in [s_2]_r \text{ s.t. } x_j = y_j \forall j \leq i\}$$

This definition has an easy visual interpretation. Given two vertices $[s_1]_r$, $[s_2]_r$, the common ancestor function finds the length of the largest possible overlap of paths from the root to each vertex.

Lemma 5.2. If $f : T_{r_1} \to T_{r_2}$ is an isomoprhism of relational trees, then $f : T_{r_1}^{(n)} \to T_{r_2}^{(n)}$ for all $n \geq 0$.

Proof. We will induct on n. Note that the root is the only vertex in either tree with no parents, so f must send root to root. This shows the base case $n = 0$.

Now suppose the result holds for level $n-1$. Consider $[s]_{r_1} \in T_{r_1}^{(n)}$ which necessarily has some parent $[t]_{r_1} \in T_{r_1}^{(n-1)}$. From our inductive assumption we know that $f([t]_{r_1}) \in T_{r_2}^{(n-1)}$. Since f is an isomorphism, it preseves parent-child relationships, so $f([s]_{r_1})$ must be the child of $f([t]_{r_1})$ and thus it must be the case that $f([s]_{r_1}) \in T_{r_2}^{(n)}$ as desired. $\square$

Lemma 5.3. Suppose $[s]_r \in T_r$ is at the nth level where $n > 0$. Then the frequency of $[s]_r$ is the sum of the frequency of its parents.

Proof. Because $[s]_r$ is on the nth level where $n > 0$, we know know it has parents. Call the parents $[p_1]_r, \ldots, [p_m]_{r_1} \in T_{r_1}^{(n-1)}$. In actuality, Theorem 2.4 guarantees that m cannot be bigger than 2 (no node can have ≥ 3 parents) so $m \in \{1, 2\}]$, but we use m for notational simplicity. Now consider two cases:

- Case One: Suppose $m = 1$. Then all the strings $s \in [s]_{r_1}$ are exactly $p^{\pm}$ for $p \in [p_1]_{r_1}$ where the $\pm$ is fixed for all p. This is guaranteed by Theorem 2.4.
- Case Two: Suppose $m = 2$. Then all the strings $s \in [s]_{r_1}$ are exactly $p^{\pm}$ for $p \in [p_1]_{r_1}$ and $p'^{\mp}$ for $p' \in [p_2]_{r_2}$ where the sign is fixed for all string representations from the same parent and opposite for string representations from different parents. This is guaranteed by Theorem 2.4.

In either case we see that

$$|[s]_r| = \sum_{i=1}^{m} |[p_i]_r|$$

$\square$

Lemma 5.4. If $f : T_{r_1} \to T_{r_2}$ is an isomoprhism of relational trees, and $f : [s]_{r_1} \mapsto [t]_{r_2}$, then $|[s]_{r_1}| = |[t]_{r_2}|$

Proof. We again proceed by induction on the level. The root has frequency 1 for both trees, so the result holds for $n = 0$. Assume the result holds for level $n-1$. Consider $[s]_{r_1} \in T_{r_1}^{(n)}$ and suppose it has parents $[p_1]_r, \ldots, [p_m]_{r_1} \in T_{r_1}^{(n-1)}$. Because f is an isomorphism, we know that $[t]_{r_2} = f([s]_{r_1}) \in T_{r_2}^{(n)}$ has parents $f([p_1]_{r_1}), \ldots, f([p_m]_{r_1}) \in T_{r_2}^{(n-1)}$. Using Lemma 5.3 and the inductive hypothesis, we see that

$$|[s]_{r_1}| = \sum_{i=1}^{m} |[p_i]_{r_1}| = \sum_{i=1}^{m} |f([p_i]_{r_1})| = |[t]_{r_2}|$$

as desired. □

Corollary 5.5. If $T_{r_1} \cong T_{r_2}$, then for every frequency f there are an equal number of nodes with frequency f in each level. Namely, for isomorphic trees, N_n and N_n^f are the same for all n, f.

Corollary 5.6. If $T_{r_1} \cong T_{r_2}$ then $|r_1| = |r_2|$

Proof. Suppose for the sake of contradiction that $|r_1| \neq |r_2|$. WLOG assume that $|r_1| < |r_2|$. Then simply note that the node $[r_1]_{r_1}$ on the $|r_1|$th level of T_{r_1} has frequency 2, while no node on the $|r_1|$th level of T_{r_1} can have frequency 2. This contradicts that $T_{r_1} \cong T_{r_2}$. □

Theorem 5.7. Suppose $f : T_{r_1} \to T_{r_2}$ is an isomoprhism. For any $[s]_{r_1}, [s']_{r_1} \in T_{r_1}$, if we let $[t]_{r_1} := f([s]_{r_1})$ and $t' := f([s']_{r_1})$, then

$$CA_{r_1}([s]_{r_1}, [s']_{r_1}) = CA_{r_2}([t]_{r_2}, [t']_{r_2})$$

Proof. Let $k = CA([s]_{r_1}, [s']_{r_1})$. This means there is some $[p]_{r_1} \in T_{r_1}^{(k)}$ that is an ancestory of both $[s]_{r_1}$ and $[s']_{r_1}$ in T_{r_1}. That is, there exist vertices $[p_1]_{r_1}, \ldots, [p_m]_{r_1}$, $[p'_1]_{r_1}, \ldots, [p'_m]_{r_1}$ such that

- $[p_1]_{r_1}, [p'_1]_{r_1}$ are children of $[p]_{r_1}$
- $[p_{i+1}]_{r_1}$ is a child of $[p_i]_{r_1}$ and $[p'_{i+1}]_{r_1}$ is a child of $[p'_i]_{r_1}$
- $[s]_{r_1}$ is a child of $[p_m]_{r_1}$ and $[s']_{r_1}$ is a child of $[p'_m]_{r_1}$

Now, since f preserves parents, we can trace the above path up to see that $f([p]_{r_1})$ is both an ancestor of $[t]_{r_2}$ and $[t']_{r_2}$. By Lemma 5.2 we know that $f([p]_{r_1}) \in T_r^{(k)}$, letting us know that

$$CA_{r_1}([s]_{r_1}, [s']_{r_1}) = k \leq CA_{r_2}([t]_{r_2}, [t']_{r_2})$$

Now, replacing f with f^{-1} in the above proof and working in the opposite direction tells us that

$$CA_{r_2}([t]_{r_1}, [t']_{r_1}) = k' \leq CA_{r_1}([s]_{r_2}, [s']_{r_2})$$

Combining the two gives the desired result that

$$CA_{r_1}([s]_{r_1}, [s']_{r_1}) = CA_{r_2}([t]_{r_2}, [t']_{r_2})$$

□

Lemma 5.8. For every r, $T_r \cong T_{-r} \cong T_{\overline{r}}$

Proof. We define our map on the string level and then show it is well defined for mapping equivalent classes. Then we show that it induces a graph isomoprhism. We show that $T_r \cong T_{\overline{r}}$. To show that $T_r \cong T_{-r}$, one can simply take the identity map between strings and follow the reasoning below.

In attempting to show that $T_r \cong T_{\overline{r}}$, consider the map $f : s \mapsto \overline{s}$. Note that if $s \equiv_r t$ then we have a sequence of indices $k_1, \ldots, k_m$ such that

$$(\alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)})(s) = t$$

For an arbitrary string w, if $\alpha_r^{(k)}(w)$ is well defined, then so is $\alpha_{\overline{r}}^{(k)}(f(w))$ as either $\overline{r}$ or $-\overline{r}$ will appear in w at index k. Then it is easy to verify that

$$\alpha_{\overline{r}}^{(k)}(f(w)) = f(\alpha_r^{(k)}(w))$$

From this we see that

$$(\alpha_{\overline{r}}^{(k_m)} \circ \cdots \circ \alpha_{\overline{r}}^{(k_1)})(f(s)) = f((\alpha_r^{(k_m)} \circ \cdots \circ \alpha_r^{(k_1)})(s)) = f(t)$$

Thus f induces a well-defined map $h : T_r \to T_{\overline{r}}$ given by $h([s]_r) = [f(s)]_{\overline{r}}$. We can also define a map $g : T_{\overline{r}} \to T_r$ given by $g([s]_{\overline{r}}) = [f(s)]_r$ which is well defined by the same reasoning. The map g is the right and left inverse of h, so h is a bijection.

Now consider a child parent/pair $[p]_r$ and $[c]_r$ in T_r, and take p and c to be representatives of the equivalence classes such that $c = p^{\pm}$. Note that $h([c]_r) = [f(c)]_{\overline{r}}$ and $h([p]_r) = [f(p)]_{\overline{r}}$. Since $c = p^{\pm}$ it follows $f(c) = f(p)^{\pm}$, and thus $[f(p)]_{\overline{r}}$ is indeed the parent of $[f(c)]_{\overline{r}}$ in $T_{\overline{r}}$. Thus h is a bijection between vertices that maintains the parent/child structure, so it is indeed and isomorphism. $\square$

The following Lemma is easy to verify and we state it without proof

Lemma 5.9. For a relation r we let the **family** of r be $F(r) := \{r, -r, \overline{r}, -\overline{r}\}$. Then

- Each element of the family is distinct, i.e. $|F(r)| = 4$.
- For some other relation r', $r' \notin F(r)$ if and only if $F(r)$ and $F(r')$ are disjoint.
- There is exactly one $r^* \in F(r)$ such that $(r^*)_1 = (r^*)_2 = 1$. We refer to this element as the **primary** relation of of $F(r)$.

Theorem 5.10. Suppose we have two relations r_1, r_2. Then $T_{r_1} \cong T_{r_2}$ if and only if $r_2 \in F(r)$

Proof. Lemma 5.8 implies the reverse direction. To show the forward direction, we will show that if $T_{r_1} \cong T_{r_2}$ for relations r_1, r_2 such that $(r_1)_1 = (r_2)_1 = +1$, $(r_1)_2 = (r_2)_2 = +1$, it must be true that $r_1 = r_2$. To see why this implies the result, suppose then that $T_{r_1} \cong T_{r_2}$ but $r_2 \notin F(r_1)$. Then it is the case by Lemma 5.9 that the two families $F(r_1)$ and $F(r_2)$ are disjoint. Thus the primary members r_1^* and r_2^* are distinct, but such that

$$T_{r_1^*} \cong T_{r_1} \cong T_{r_2} \cong T_{r_2^*}$$

which contradicts the claim above.

Now we show the claim. Consider r_1, r_2 such that $(r_1)_1 = (r_2)_1 = +1$, $(r_1)_2 = (r_2)_2 = +1$ such that $T_{r_1} \cong T_{r_2}$. By Corollary 5.6, we know that $|r_1| = |r_2| := \ell$. We write r_1 and r_2 as blocks of $+1$'s and -1's. Namely

$$r_1 = (+1)^{a_0}(-1)^{a_1}(+1)^{a_2} \ldots (\pm 1)^{a_m}$$

meaning that r_1 consists of a_0 +1's followed by a_1 -1's etc. Similarly we let

$$r_2 = (+1)^{b_0}(-1)^{b_1}(+1)^{b_2}\dots(\pm 1)^{b_n}$$

Assume for the sake of contradiction that $r_1 \neq r_2$, so there is some minimal index i such that $a_i \neq b_i$. WLOG we will take $a_i < b_i$ (simply interchange r_1 and r_2 if this is not the case). Now, define

$$V := a_0 + a_1 + \cdots + a_i < \ell$$

$$s := \{(r_1)_1, (r_1)_2, \dots, (r_1)_{V-1}, (r_1)_1, \dots, (r_1)_\ell\}$$

In other words, s is a string which is the concatenation of r_1 truncated to index $V-1$ and r_1.

We will consider the value of $CA_{r_1}([r_1]_{r_1}, [s]_{r_1})$. To do so, note that the equivalence class $[r_1]_{r_1}$ has two string elements: r_1, $-r_1$. We will compute $CA_{r_1}([r_1]_{r_1}, [s]_{r_1})$ directly by finding the string representative of $[s]_{r_1}$ that maximizes x such that the first x values of r_1 and the string representative are identical. First note that by construction the first $V-1$ and of s and r_1 are already identical. We consider two cases:

- Case One: In the first case suppose $(r_1)_V = +1$. Then s such that the first V elements of r_1 and s are identical, as $s_V = r_1 = +1$. By construction however $r_{V+1} = -1$ and $s_{V+1} = r_2 = +1$, so exactly only the first V elements of r_1 and s are identical. Theorem 2.4 tells us that if there is some s' with a deeper common ancestry to r_1 that is equivalent to s then we must be able to find s' from s by applying relations starting at indices $\geq V+1$, but since s only has length $V+\ell-1$, this isn't possible. Thus V is the largest overlap we can achieve in this case, and we do indeed achieve it.

- Case Two: In the second case we have $(r_1)_V = -1$. Then by construction $r_{V+1} = +1$. In this case we can use the string $s' = \alpha_{r_1}^{(V)}(s)$. This string will be such that $(s')_V = -(r_1)_1 = -1$ and $(s')_{V+1} = -(r_1)_2 = -1$. The same argument as in first case tells us that V is the largest overlap we can achieve, and we do indeed achieve it.

Applying the analogous argument when attempting to the maximize deepest common ancestry of a representative of $[s]_{r_1}$ and $-r_1$ we get the identical result. We thus see that $CA_{r_1}([r_1]_{r_1}, [s]_{r_1}) = V$.

Now, since the trees are isomoprhic, we have some isomorphism f between them. Consider $f([r_1]_{r_1})$. By Lemma 5.2 and Lemma 5.4 we see that $f([r_1]_{r_1})$ must be $[r_2]_{r_2}$. Now consider $[t]_{r_2} := f([s]_{r_1})$. Since $[s]_{r_1}$ had two parents (it ends in the relation so it must), $[t]_{r_2}$ must also have two parents. From the proof of Lemma 3.1 we know thus that there is some string representative of $[t]_{r_2}$ that ends in the relation r_2. Noting that $[t]_{r_2} \in T_{r_2}^{(V+\ell-1)}$ by Lemma 5.2, there is thus some string representative of $[t]_{r_2}$ that can be written as

$$t' = (t'_1, \dots, t'_{V-1}, \pm(r_2)_1, \pm(r_2)_2, \dots, \pm(r_2)_\ell)$$

where the $\pm$ are all either $+$ or $-$. In particular, since $(r_2)_1 = (r_2)_2$, we know that there is a string representative $[t]_{r_2}$ with $t'_V = t'_{V+1}$. Now, by Theorem 5.7 it must be the case that

$$CA_{r_2}([r_2]_{r_2}, [t]_{r_2}) = V$$

Namely, there exists a string representation that is identical to either r_2 or $-r_2$ for the first V spots. Thus there exists some string representative of $[t]_{r_2}$ that can be written as

$$t'' = (\pm(r_2)_1, \dots, \pm(r_2)_{V-1}, \pm(r_2)_V, t''_{V+1}, \dots, t''_{V+\ell-1})$$

where all the $\pm$ are all either $+$ or $-$. Recall that by construction V is such that $(r_2)_V = (r_2)_{V+1}$. Now consider two cases

- Case One: Suppose t''_{V+1} has the same sign as whatever the sign of $\pm(r_2)_V$ is. Then this assumption combined with the fact that $(r_2)_V = (r_2)_{V+1}$ implies that $CA_{r_2}([r_2]_{r_2}, [t]_{r_2}) > V$, which is a contradiction.

- Case Two: Suppose t''_{V+1} has a the opposite sign of whatever $\pm(r_2)_V$ is. Then, we have a string representation t'' of $[t]_{r_2}$ where $t''_V \neq t''_{V+1}$. Since $V < \ell$ and $[t]_{r_2}$ is on the $V + \ell - 1$st level, there is no well defined application of the relation that has line of action that doesn't include both V and $V + 1$. Thus t'' cannot be equivalent to any string representation where the Vth and $V + 1$st values are equal (the application of any relation must flip both the values). Note that t' above is a string representation equivalent to t'' such that $t'_V = t'_{V+1}$, so we again have a contradiction.

$\square$

Corollary 5.11. Considering relations r of length ℓ, there are $2^{\ell-2}$ relational trees up to isomorphism when $\ell \geq 2$, and 1 when $\ell = 1$.

6 Final Remarks and Future Work

We have accomplished two main tasks

- Counting the number of equivalence classes at each level n
- Classifying relational trees up to isomorphism

We still don't quite understand the exact dynamics of frequency. Computing $N_n^f(r)$ for all n, f seems to be quite difficult. As a first step, we can try and understand when $N_n^f(r) = N_n^f(r')$ for two relations r, r'. Looking at the ℓth level of the tree it is clear that we must have $|r| = |r'|$ for this to be the case. It is not the case however that we require $r' \in F(r)$. We leave the reader with this final motivating example, that has been touched on above.

Example 6.1. Consider a relation r such that $\tilde{r} \notin F(r)$. Such a relation is not difficult to find, for example $r = (+1, -1, +1, -1, -1)$. Then $N_n^f(r) = N_n^f(\tilde{r})$ for all n, f by Lemma 2.3 but $T_r \ncong T_{\tilde{r}}$ by Theorem 5.10