Successor-Invariant First-Order Logic on Classes of Bounded Degree

Julien Grange
ENS Paris, PSL, INRIA, CNRS
jgrange@di.ens.fr

Abstract

We study the expressive power of successor-invariant first-order logic, which is an extension of first-order logic where the usage of an additional successor relation on the structure is allowed, as long as the validity of formulas is independent on the choice of a particular successor.

We show that when the degree is bounded, successor-invariant first-order logic is no more expressive than first-order logic.

CCS Concepts: • **Theory of computation** → **Finite Model Theory**.

Keywords: finite model theory, successor-invariance, first-order logic

ACM Reference Format:
Julien Grange. 2020. Successor-Invariant First-Order Logic on Classes of Bounded Degree. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20), July 8–11, 2020, Saarbrücken, Germany*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3373718.3394767

1 Introduction

First order logic, FO, is the standard formalism to express properties of finite structures. Its expressive power is well known, and very restrained, as it can only express properties that are local, which roughly means that it can only talk about the immediate surroundings of a small number of elements, and it is unable to count.

A number of logics with higher expressivity can be defined with FO as a building block, such as MSO, in which quantification over sets is allowed, and LFP, which adds a fixpoint operator to FO. These additions break the local character of the logic.

Another way to define logics from FO is through the addition, in an invariant way, of arithmetic predicates on the structure that are exterior to the vocabulary. This amounts to arbitrarily identifying the universe of the structure with an initial segment of the integers, and allowing some arithmetic on them. However, we want these extensions to define properties of the structures, and not to depend on a particular ordering on their elements: thus we focus on invariant extensions of FO.

If the only predicate allowed is the order, we get order-invariant first-order logic, $<$-inv FO. Restricting a bit the additional relation, we get successor-invariant first-order, Succ-inv FO. In this formalism, we only grant an access to the successor relation derived from the order, provided that the evaluation of a sentence using this successor relation is independent of the choice of a particular successor.

The problem of determining whether an FO-sentence using an order or a successor relation is invariant wrt. this relation is undecidable, by reduction from Trakhtenbrot's theorem [10]. Hence we use here the term "logic" somewhat liberally, since having a recursive syntax is a usual requirement for a logic.

The study of these two formalisms finds its motivation, among other topics such as descriptive complexity, in database theory. As databases are commonly stored on disk that implicitly order their memory segments, when one wishes to express a query in FO, one has access to an additional order on the elements of the database. However, making use of this order without care could result in queries that evaluate differently on two implementations of the same database, which is clearly an undesirable behavior. We want to use this order only in an invariant way; this way, the result of a query depends only on the database it is run on, and not on the way the data is stored on disk. This amounts exactly to the definition of $<$-inv FO, or Succ-inv FO if we restrict the way this order can be accessed.

It is straightforward that $<$-inv FO is at least as expressive as Succ-inv FO, which in turn can express any FO-definable property. Gurevich constructed a class of finite structures that can be defined by an $<$-inv FO sentence, but which is not FO-definable. Though this construction wasn't published by Gurevich, it can be found e.g. in Section 5.2 of [8]. Rossman extended this result, and proved in [9] that on finite structures, Succ-inv FO is strictly more expressive than FO.

LICS '20, July 8–11, 2020, Saarbrücken, Germany

ACM ISBN 978-1-4503-7104-9/20/07…$15.00
https://doi.org/10.1145/3373718.3394767

Grohe and Schwentick [7] proved that these logics were Gaifman-local, giving an upper bound to their power of expression. Other upper bounds were given by Benedikt and Segoufin [1], who proved that $<$ -inv FO, and hence Succ-inv FO, are included in MSO on classes of bounded treewidth and on classes of bounded degree. Elberfeld, Frickenschmidt and Grohe [3] extended the first inclusion to a broader setting, that of decomposable structures. Whether these logics are included in MSO in general is still an open question.

The classes of structures involved in the separating examples by Gurevich and Rossman are dense, and no other example is known on classes that are sparse. Far from it, $<$-inv FO and *a fortiori* Succ-inv FO are known to collapse to FO on several sparse classes. Benedikt and Segoufin [1] proved the collapse on trees; Eickmeyer, Elberfeld and Harwarth [2] obtained an analogous result on graphs of bounded tree-depth; Grange and Segoufin [6] proved the collapse on hollow trees.

Whether $<$ -inv FO or Succ-inv FO collapse to FO on classes of graphs of bounded treewidth (or even bounded pathwidth) are still open questions. We go in another direction in this paper, and prove that Succ-inv FO collapses to FO on classes of structures of bounded degree. To do this, we show how to construct successors on two FO-similar structures of bounded degree, such that the two structures remain FO-similar when considering the additional successor relation.

Related work: The general method used in [2] to prove that $<$-inv FO collapses to FO when the tree-depth is bounded is the same as ours: starting from two FO-similar structures, they show how to construct orders that maintain the similarity. However, the techniques we use to construct our successors are nothing like the ones used in [2], as the settings are very different.

Instead of directly constructing similar orders on two similar structures, [1] and [6] exhibit a chain of intermediate structures and intermediate orders that are pairwise similar, in order to prove the collapse on trees and hollow trees. Once again, our construction differs wildly from these ones.

2 Preliminaries

The remainder in the division of $n \in \mathbb{N}$ by $m > 0$ is denoted $n[m]$.

A binary relation on a finite set X is a **successor relation on** X if it is the graph of a circular permutation of X, i.e. a bijective function from X to X with a single orbit. This differs from the standard notion of successor in that there is neither minimal nor maximal element. However, this doesn't have any impact on our result, as discussed at the end of the present Section.

We use the standard definition of first-order logic FO(Σ) over a signature Σ composed of relation and constant symbols. We only consider finite Σ-structures, which are denoted by calligraphic upper-case letters, while their universes are denoted by the corresponding standard upper-case letters; for instance, A is the universe of the structure $\mathcal{A}$.

Definition 2.1 (Succ-inv FO)**.** A sentence $\varphi \in \text{FO}(\Sigma \cup \{S\})$, where S is a binary relation symbol, is said to be **successor-invariant** if for every Σ-structure $\mathcal{A}$, and every successor relations S_1 and S_2 on A, $(\mathcal{A}, S_1) \models \varphi$ iff $(\mathcal{A}, S_2) \models \varphi$. We can then omit the interpretation for the symbol S, and if $(\mathcal{A}, S_1) \models \varphi$ for any (every) successor S_1, we write $\mathcal{A} \models \varphi$.

The set of successor-invariant sentences on Σ is denoted Succ-inv FO(Σ).

Given two Σ-structures $\mathcal{A}$ and $\mathcal{B}$, and $\mathcal{L}$ being either FO(Σ) or Succ-inv FO(Σ), we write $\mathcal{A} \equiv_k^{\mathcal{L}} \mathcal{B}$ if $\mathcal{A}$ and $\mathcal{B}$ satisfy the same $\mathcal{L}$-sentences of quantifier rank at most k. For FO(Σ) as well as for Succ-inv FO(Σ), we omit Σ when it is clear from the context.

Definition 2.2 (Gaifman graph)**.** The Gaifman graph $\mathcal{G}_{\mathcal{A}}$ of a Σ-structure $\mathcal{A}$ is defined as (A, V) where $(x, y) \in V$ iff x and y appear in the same tuple of a relation of $\mathcal{A}$. In particular, if a graph is seen as a relational structure on the vocabulary $\{E\}$, its Gaifman graph is the unoriented version of itself. By $\text{dist}_{\mathcal{A}}(x, y)$, we denote the distance between x and y in $\mathcal{G}_{\mathcal{A}}$. The **degree** of $\mathcal{A}$ is the degree of its Gaifman graph, and a class C of Σ-structures is said to be of **bounded degree** if there exists some $d \in \mathbb{N}$ such that the degree of every $\mathcal{A} \in C$ is at most d.

Definition 2.3 (Types)**.** We now give the definition of the type of an element. Beware that this has a different meaning in classical model theory, where it denotes the set of sentences with one free variable that are satisfied by this element. Here, the type of an element describes its surroundings in the structure.

Let c be a constant symbol that doesn't appear in Σ.

For $k \in \mathbb{N}$ and $x \in A$, the k-**neighborhood** $\mathcal{N}_{\mathcal{A}}^k(x)$ of x is the $(\Sigma \cup \{c\})$-structure whose Σ-restriction is the substructure of $\mathcal{A}$ induced by $\{y \in A : \text{dist}_{\mathcal{A}}(x, y) \leq k\}$, and where c is interpreted as x.

The k-**type** $\tau = \text{tp}_{\mathcal{A}}^k(x)$ is the isomorphism class of its k-neighborhood. We say that τ is a type over Σ, and that x is an **occurrence** of τ. $|\mathcal{A}|_\tau$ denotes the number of occurrences of τ in $\mathcal{A}$, and we write $[\![\mathcal{A}]\!]_r =^t [\![\mathcal{B}]\!]_r$ to mean that for every r-type τ, $|\mathcal{A}|_\tau$ and $|\mathcal{B}|_\tau$ are either equal, or both larger than t.

Definition 2.4 (Path and cycles)**.** A **cycle** of length $l \geq 2$ in the $\Sigma \cup \{S\}$-structure $\mathcal{A}$ is a sequence $(x_0, \ldots, x_{l-1})$ of distinct vertices of A such that for every $0 \leq i < l$, x_i and $x_{i+1[l]}$ appear in the same tuple of some relation of $\mathcal{A}$ (in other words, it is a cycle in $\mathcal{G}_{\mathcal{A}}$). If furthermore $(x_i, x_{i+1[l]}) \in S$

for every i, then we say that it is an S-cycle. If for some i, $(x_i, x_{i+1[l]}) \in S$ or $(x_{i+1[l]}, x_i) \in S$, then we say that the cycle goes through an S-edge. A **path** is defined similarly, without the requirement on x_{l-1} and x_0, and its length is $l-1$ instead of l.

From now on, we assume that Σ is purely relational (i.e. contains only relation symbols) and doesn't contain the binary symbol S.

For a class C of Σ-structures, we say that

$$\text{Succ-inv FO} = \text{FO on } C$$

if the properties of C definable in Succ-inv FO and in FO are the same. In other words, if for every $\varphi \in$ Succ-inv FO, there exists some $\bar{\varphi} \in$ FO such that

$$\forall \mathcal{A} \in C, \quad \mathcal{A} \models \varphi \quad \text{iff} \quad \mathcal{A} \models \bar{\varphi} \,.$$

The reverse inclusion, i.e. FO $\subseteq$ Succ-inv FO, always holds and needs no verification.

We are now ready to state our main result:

Theorem 2.5. *For every vocabulary Σ and for every class C of Σ-structures of bounded degree,*

$$\text{Succ-inv } FO = FO \text{ on } C \,.$$

Let's first state a corollary of this result. Gajarský, Hliněný, Obdržálek, Lokshtanov and Ramanujan [5] characterized the graph classes which are FO-interpretable in a class of graphs of bounded degree as the near-uniform graph classes.

Let $\mathcal{D}$ be such a class of graphs, which is FO-interpretable in a class of graphs of bounded degree C'. The construction from the aforementioned paper exhibits a class C of graphs of bounded degree (which may be larger that the degree of C') and an FO-interpretation $\mathcal{I}$ such that every $\mathcal{H} \in \mathcal{D}$ is isomorphic to $\mathcal{I}(\mathcal{G})$ for some $\mathcal{G} \in C$, and such that $\mathcal{I}$ admits a converse which preserves FO-similarity (although the converse is not an FO-interpretation itself); namely, for every $k \in \mathbb{N}$, there exists $k' \in \mathbb{N}$ such that for every $\mathcal{H}, \mathcal{H}' \in \mathcal{D}$ such that

$$\mathcal{H} \equiv^{\text{FO}}_{k'} \mathcal{H}'$$

then there exist $\mathcal{G}, \mathcal{G}' \in C$ such that

$$\mathcal{H} \simeq \mathcal{I}(\mathcal{G}) \quad \wedge \quad \mathcal{H}' \simeq \mathcal{I}(\mathcal{G}') \quad \wedge \quad \mathcal{G} \equiv^{\text{FO}}_{k} \mathcal{G}' \,.$$

Theorem 2.5 can then be lifted to near-uniform graph classes, and we get the following corollary:

Corollary 2.6. *Let $\mathcal{D}$ be a near-uniform class of graphs, i.e. a class of graphs FO-interpretable in a class of graphs of bounded degree. Then*

$$\text{Succ-inv } FO = FO \text{ on } \mathcal{D} \,.$$

The proof of Theorem 2.5 is given in Section 4, and constitutes the core of this paper. We give here a sketch of this proof; this will motivate the definitions given in Section 3.

Proof overview. Our goal is, given two structures $\mathcal{G}_1$ and $\mathcal{G}_2$ of degree at most d that are FO-similar (that is, such that $\mathcal{G}_1 \equiv^{\text{FO}}_n \mathcal{G}_2$ for a large enough n), to construct a successor relation S_1 on $\mathcal{G}_1$ and S_2 on $\mathcal{G}_2$ such that $(\mathcal{G}_1, S_1)$ and $(\mathcal{G}_2, S_2)$ stay FO-similar. This will entail that $\equiv^{\text{FO}}$ refines $\equiv^{\text{Succ-inv FO}}$ when the degree is bounded. From there, a standard finite-model-theoretic argument (namely, that $\equiv^{\text{FO}}_n$ has finite index and that each one of its classes is FO-definable) gives the inclusion Succ-inv FO $\subseteq$ FO on classes of bounded degree.

It thus remains to construct suitable successor relations S_1 and S_2. First, we separate the types occurring in $\mathcal{G}_1$ and $\mathcal{G}_2$ into two categories:

- on the one hand, the rare types, which have few occurrences in $\mathcal{G}_1$ and $\mathcal{G}_2$ (and thus, that have the same number of occurrences in both structures, by FO-similarity)
- on the other hand, the frequent types, which have many occurrences both in $\mathcal{G}_1$ and $\mathcal{G}_2$.

In order to make the proof of FO-similarity of $(\mathcal{G}_1, S_1)$ and $(\mathcal{G}_2, S_2)$ as simple as possible, we want an element of $\mathcal{G}_1$ (and similarly for $\mathcal{G}_2$ and S_2) and its successor by S_1 to have the same type in $\mathcal{G}_1$ as much as possible, and to be far enough in $\mathcal{G}_1$, in order for the types occurring in $(\mathcal{G}_1, S_1)$ to be as "regular" as possible. As long as there are at least two different types, the first constraint obviously cannot be satisfied, but we will construct S_1 as close as possible to satisfying it.

For instance, suppose that $\mathcal{G}_1$ contains three frequent types τ_0, τ_1 and τ_2, and one rare type χ with two occurrences. At the end of the construction, S_1 will (mostly) look like in Figure 1, where the relations of $\mathcal{G}_1$ have been omitted and the arrows represent S_1, which is indeed a circular successor.

Note that all the elements of type τ_1 form a segment wrt. S_1, as well as all the elements of type τ_2. τ_0, the first frequent type, has a special role in that it is used to embed all the elements of rare type (here, χ). Furthermore, and this is not apparent in the figure, two successive elements for S_1 are always distant in $\mathcal{G}_1$.

Keeping this idea in mind, S_1 (and similarly, S_2) is constructed iteratively, by adding S-edges to the initial structures one at a time. For practical reasons, we will start the construction of S_1 around occurrences of rare types: for each element x of rare type, we find two elements of type τ_0 that are far apart in $\mathcal{G}_1$, and far from x. Then we add two S-edges in order for those two elements to become the S_1-predecessor and the S_1-successor of x. We repeat this process for every element of rare type (and actually, for every element that belongs to the neighborhood of a rare element) until each one is protected by a ball of elements of frequent type. This is possible because there are few elements of rare type, and many elements of any frequent type; since the degree is bounded, those elements of frequent type are spread across

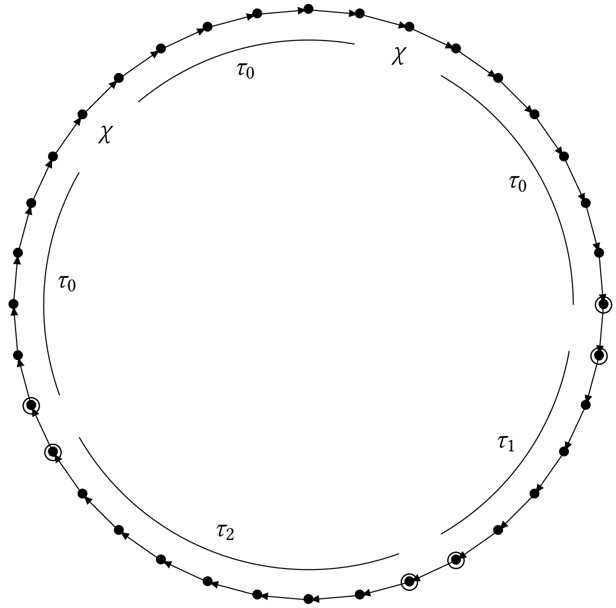

Figure 1. Illustration of S_1 when there are three frequent types (τ_0, τ_1, τ_3) and one rare type (χ) in $\mathcal{G}_1$. The elements of rare type are surrounded by occurrences of the first frequent type, τ_0. Junction elements are circled.

the structure, and can be found far from the current construction.

Once this is done, we apply a similar construction around elements of frequent types that will, in the end, be the S_1-predecessor or S_1-successor of an element of another frequent type - that is, elements that will be at the border of the segments (for S_1) of a given frequent type. Such elements are circled in Figure 1. We must choose only a small number of such elements (two for each frequent type, of which there are few due to the degree boundedness hypothesis), hence we can find enough far-apart elements of frequent type to embed them. Once again, degree boundedness is crucial.

After these two steps, S_1 has been constructed around all the singular points. It only remains to complete S_1 by adding edges between the remaining elements (all of which are occurrences of frequent types), in such a way that elements of a same frequent type end up forming a segment for S_1, and such that S_1 brings together elements that were far apart in the initial structure $\mathcal{G}_1$. Once again, the high number of occurrences of each frequent type allows us to do so.

Applying the same construction to $\mathcal{G}_2$, we end up with two structures $(\mathcal{G}_1, S_1)$ and $(\mathcal{G}_2, S_2)$ that cannot be distinguished by FO-formulas of small (wrt. the initial FO-similarity index between $\mathcal{G}_1$ and $\mathcal{G}_2$) quantifier rank, which concludes the proof.

We have given a global overview of the construction process of S_1; however, there are technical difficulties to take care of, which are dealt with in Section 4. For that, we need the definitions given in Section 3, which formalize the notion of regularity of a type in $(\mathcal{G}_1, S_1)$ and $(\mathcal{G}_2, S_2)$. □

Let's now prove that our decision to consider circular successors instead of the more traditional linear ones (with a minimal and a maximal element) bears no consequence on this result. If we define LinSucc-inv FO in the same way as Succ-inv FO, but where the invariant relation is a linear successor $\bar{S}$, we get:

Lemma 2.7. *For every vocabulary Σ,* LinSucc-inv *FO and* Succ-inv *FO define the same properties of Σ-structures.*

Proof. Given $\varphi \in$ Succ-inv FO, let's prove that there exists a formula $\psi \in$ LinSucc-inv FO such that ψ is equivalent to φ (i.e. for every Σ-structure $\mathcal{A}$, $\mathcal{A} \models \varphi$ iff $\mathcal{A} \models \psi$).

Let ψ be defined as φ in which every atom $S(x,y)$ has been replaced with $\bar{S}(x,y) \vee \neg\exists z(\bar{S}(x,z) \vee \bar{S}(z,y))$.

Let $\mathcal{A}$ be a Σ-structure and $\bar{S}$ be a linear successor on A. Then $(\mathcal{A}, \bar{S}) \models \psi$ iff $(\mathcal{A}, S) \models \varphi$, where S is the circular successor obtained from $\bar{S}$ by adding an edge from the maximal element to the minimal one.

This guarantees that $\psi \in$ LinSucc-inv FO, and that ψ and φ are equivalent.

Conversely, let $\psi \in$ LinSucc-inv FO and let φ be the formula $\exists \min, \mathrm{Cut}(\psi)$, where $\mathrm{Cut}(\psi)$ is obtained by replacing in ψ every $\bar{S}(x,y)$ with $S(x,y) \wedge \neg y = \min$.

Let $\mathcal{A}$ be a Σ-structure, let S be a circular successor on A, and let $\min \in A$. Then $(\mathcal{A}, S, \min) \models \mathrm{Cut}(\psi)$ iff $(\mathcal{A}, \bar{S}) \models \psi$, where $\bar{S}$ is the linear successor obtained from S by removing the edge pointing to min. Hence $(\mathcal{A}, S) \models \varphi$ iff there exists a linear successor $\bar{S}$ obtained from S by an edge removal such that $(\mathcal{A}, \bar{S}) \models \psi$, that is iff $\mathcal{A} \models \psi$.

This ensures that $\varphi \in$ Succ-inv FO and that φ and ψ are equivalent. □

Note that although LinSucc-inv FO and Succ-inv FO have the same expressive power, one may be more concise than the other, in terms of quantifier rank.

3 Fractal types and layering

To prove Theorem 2.5, we will start from two structures $\mathcal{G}_1$ and $\mathcal{G}_2$ that are FO-similar, and construct successor relations S_1 and S_2 on their universes so that the structures remain FO-similar when we take into account these additional successor relations.

We want to construct S_ϵ, for $\epsilon \in \{1,2\}$, in a way that makes $\mathrm{tp}^k_{(\mathcal{G}_\epsilon, S_\epsilon)}(a)$ as regular as possible for every $a \in G_\epsilon$, in order to ease the proof of FO-similarity of $(\mathcal{G}_1, S_1)$ and $(\mathcal{G}_2, S_2)$.

Ideally, the S_ϵ-successors and S_ϵ-predecessors of any element should have the same k-type in $\mathcal{G}_\epsilon$ as this element. On top of that, there should not be any overlap between the k-neighborhoods in $\mathcal{G}_\epsilon$ of elements that are brought closer by

S_ϵ (this "independence" is captured by the layering property, introduced in Definition 3.2).

If we now try to visualize what $\mathrm{tp}^k_{(\mathcal{G}_\epsilon, S_\epsilon)}(a)$ would look like in those perfect conditions, we realize that it reminds of a fractal (although the patterns - that is, the types - are obviously repeated only a finite number of times).

This is why we introduce in Definition 3.1 the **fractal type** $[\tau]_k$.

Aside from a small number of exceptions (namely, for types that don't occur frequently enough, and around the transitions between frequent types), every element of k-type τ in $\mathcal{G}_\epsilon$ will have the fractal type $[\tau]_k$ in $(\mathcal{G}_\epsilon, S_\epsilon)$.

If $\mathcal{N}$ is a representative of a type τ, $c^{\mathcal{N}}$ is called the **center of** $\mathcal{N}$. Recall from Definition 2.3 that c is the constant symbol added to Σ when considering types to pinpoint the central element of a neighborhood.

Definition 3.1 (Fractal types)**.** We define by induction on $k \in \mathbb{N}$, for every k-type τ over Σ, the k-types $[\tau]_k$, $[\tau]^+_k$ and $[\tau]^-_k$ over $\Sigma \cup \{S\}$.

For $k = 0$, $[\tau]_0 = [\tau]^+_0 = [\tau]^-_0 = \tau$ (meaning that S is interpreted as the empty relation in $[\tau]_0$, $[\tau]^+_0$ and $[\tau]^-_0$).

Starting from a representative $\mathcal{N}$ of center a of the isomorphism class τ, we construct $\mathcal{N}'$ as follows.

For every $x \in N$ at distance $d \leq k-1$ from a, let $\mathcal{M}^+_x$ and $\mathcal{M}^-_x$ be structures of respective isomorphism type $[\chi]^+_{k-d-1}$ and $[\chi]^-_{k-d-1}$, where χ is the $(k-d-1)$-type of x in $\mathcal{N}$, and of respective center x^+ and x^-.

$\mathcal{N}'$ is defined as the disjoint union of $\mathcal{N}$ and all the $\mathcal{M}^+_x$ and the $\mathcal{M}^-_x$, for $x \neq a$, together with all the edges $S(x, x^+)$ and $S(x^-, x)$.

From there, $\mathcal{N}^+$ (resp. $\mathcal{N}^-$) is defined as the the disjoint union of $\mathcal{N}'$ and $\mathcal{M}^+_a$ (resp. $\mathcal{M}^-_a$) together with the edge $S(a, a^+)$ (resp. $S(a^-, a)$). Likewise, $\mathcal{N}^{+/-}$ is defined as the disjoint union of $\mathcal{N}'$, $\mathcal{M}^+_a$ and $\mathcal{M}^-_a$ together with the edges $S(a, a^+)$ and $S(a^-, a)$. In each case, a is taken as the center.

Now, $[\tau]_k$, $[\tau]^+_k$ and $[\tau]^-_k$ are defined respectively as the isomorphism type of $\mathcal{N}^{+/-}$, $\mathcal{N}^+$ and $\mathcal{N}^-$.

An illustration of this definition is given in Figure 2.

Definition 3.2 (Layering)**.** We say that an r-neighborhood $\mathcal{N}$ over $\Sigma \cup \{S, c\}$ is **layered** if it doesn't contain any cycle going through an S-edge. Every $[\tau]_r$ is obviously layered by construction.

We say that a structure over $\Sigma \cup \{S\}$ satisfies the property (`Layer[`r`]`) iff all the r-neighborhoods of this structure are layered.

It turns out (`Layer[`r`]`) can be reformulated in a way that doesn't involve the r-neighborhoods of the structure:

Lemma 3.3. *A structure $\mathcal{G}$ over $\Sigma \cup \{S\}$ satisfies* (`Layer[`r`]`) *if and only if it contains no cycle of length at most $2r+1$ going through an S-edge.*

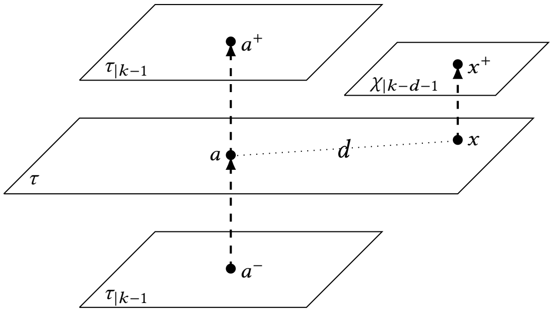

Figure 2. Partial representation of $\mathcal{N}^{+/-}$, of type $[\tau]_k$. Here, χ is the $(k-d)$-type of the element x, at distance d from a in τ. The dashed arrows represent S-edges.

Proof. If $\mathcal{G}$ contains a cycle of length at most $2r+1$ going through an S-edge, then the r-neighborhood of any vertex of this cycle contains the whole cycle, thus (`Layer[`r`]`) doesn't hold in $\mathcal{G}$.

Conversly, suppose that there exists $x \in G$ such that $\mathcal{N}^r_{\mathcal{G}}(x)$ contains a cycle going through an S-edge, and let $S(y, z)$ be such an edge.

For any $u \in \mathcal{N}^r_{\mathcal{G}}(x)$, we define the **cone** C_u **at** u as the set of elements $v \in \mathcal{N}^r_{\mathcal{G}}(x)$ such that every shortest path from x to v in $\mathcal{N}^r_{\mathcal{G}}(x)$ goes through u.

There are two cases, depending on the relative position of y, z and their cones:

- If $z \notin C_y$ and $y \notin C_z$, let $p_{y \to x}$ (resp. $p_{x \to z}$) be a path of minimal length from y to x, not going through z (resp. from x to z, not going through y).
  Let X be the set of nodes appearing both in $p_{y \to x}$ and $p_{x \to z}$. X is not empty, as $x \in X$, and $y, z \notin X$. Let $v \in X$ such that $\mathrm{dist}_{\mathcal{G}}(x, v)$ is maximal among the nodes of X, and let $p_{y \to v}$ (resp. $p_{v \to z}$) be the segment of $p_{y \to x}$ (resp. of $p_{x \to z}$) from y to v (resp. from v to z).
  Then $p_{v \to z} \cdot (z, y) \cdot p_{y \to v}$ is a cycle going through an S-edge, and is of length $\leq 2r+1$. This is illustrated in Figure 3.

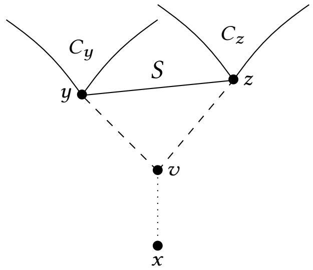

Figure 3. Existence of a short cycle joining y, z and v

- Otherwise, suppose without loss of generality that $z \in C_y$. This entails that $y \notin C_z$ and $\mathrm{dist}_{\mathcal{G}}(x, z) = d+1$ where $d := \mathrm{dist}_{\mathcal{G}}(x, y)$.

Let the initial cycle be $(z, v_1, \cdots, v_{m-1}, y)$, with the notation $v_0 = z$ and $v_m = y$.
Let i be the minimal integer such that $v_i \notin C_z$. Let $p_{x \to v_i}$ be a shortest path from x to v_i: by definition, it doesn't intersect C_z, and has length at most r. Thus, there exists a path $p_{y \to v_i} = p_{y \to x} \cdot p_{x \to v_i}$ from y to v_i of length at most $r + d$ going only through nodes outside of C_z.
Since $v_{i-1} \in C_z$, there exists a path $p_{v_{i-1} \to z}$ from v_{i-1} to z of length at most $r - (d + 1)$ going only through nodes of C_z.
Hence $p_{y \to v_i} \cdot (v_i, v_{i-1}) \cdot p_{v_{i-1} \to z} \cdot (z, y)$ is a cycle going though an S-edge, and its length is at most $2r + 1$. This is depicted in Figure 4.

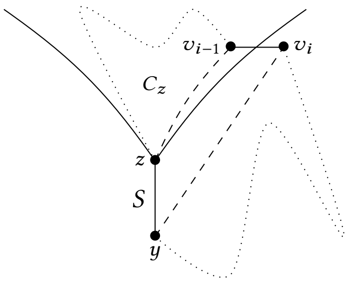

Figure 4. Existence of a short cycle joining y, z, v_{i-1} and v_i

□

This characterization of (Layer[r]) allows us to state the following lemma, which is now straightforward. It gives a method to add an S-edge without breaking the property (Layer[r]).

Lemma 3.4. *Let $r \in \mathbb{N}$, and $(\mathcal{G}, S)$ be a structure satisfying* (Layer[r]).
If $x, y \in G$ are such that $dist_{(\mathcal{G},S)}(x, y) > 2r + 1$, then (Layer[$r$]) *holds in $(\mathcal{G}, S \cup \{(x, y)\})$*

4 Proof of Theorem 2.5

We are now ready to prove Theorem 2.5. Recall the sketch of proof from Section 2. We proceed in several steps:

Section 4.1 details the general framework of the proof. In Section 4.2, we divide the types into rare ones and frequent ones.

We then begin the construction of S_1: Section 4.3 is dedicated to the construction of S_1 around the occurrences in $\mathcal{G}_1$ of rare types. Then, in Section 4.4, we keep constructing S_1 around the occurrences (two for each type) of frequent types that are designed to make, when the construction is complete, the S_1-junction between two frequent types.

At this point, S_1 will be fully built around the singular points of $\mathcal{G}_1$. Section 4.5 deals with the transfer of this partial successor relation S_1 over to $\mathcal{G}_2$: this will result in a partial S_2, built in a similar way around the singular points of $\mathcal{G}_2$.

In Section 4.6, S_1 and S_2 are completed independently, to cover G_1 and G_2. These expansions do not need to be coordinated, since at this point, the elements that are not already covered by S_1 and S_2 are occurrences of frequent types and their resulting types will be regular (i.e. fractal) both in $(\mathcal{G}_1, S_1)$ and $(\mathcal{G}_2, S_2)$.

We then give some simple examples in Section 4.7, before establishing properties of S_1 and S_2 in Section 4.8, and concluding the proof in Section 4.9.

4.1 General method

Let C be a class of Σ-structures of degree at most d. We show the following: for every $\alpha \in \mathbb{N}$, there exists some $f(\alpha) \in \mathbb{N}$ such that, given $\mathcal{G}_1, \mathcal{G}_2 \in C$, if $\mathcal{G}_1 \equiv^{\text{FO}}_{f(\alpha)} \mathcal{G}_2$ then $\mathcal{G}_1 \equiv^{\text{Succ-inv FO}}_{\alpha} \mathcal{G}_2$. For that, we will exhibit successor relations S_1 and S_2 such that $(\mathcal{G}_1, S_1) \equiv^{\text{FO}}_{\alpha} (\mathcal{G}_2, S_2)$.

More precisely, using the notations from Definition 2.3, we will show that $[\![(\mathcal{G}_1, S_1)]\!]_r =^t [\![(\mathcal{G}_2, S_2)]\!]_r$ where r and t depend on α and are large enough to ensure that $(\mathcal{G}_1, S_1) \equiv^{\text{FO}}_{\alpha} (\mathcal{G}_2, S_2)$. The existence of such r and t follows from the well-known Hanf threshold theorem, whose finite version is given in [4].

We will construct S_1 and S_2 iteratively in a way that ensures, at each step, that the property (Layer[r]) holds in $(\mathcal{G}_1, S_1)$ and in $(\mathcal{G}_2, S_2)$. (Layer[r]) is obviously satisfied in $(\mathcal{G}_1, \emptyset)$. Each time we add an S_1-edge or an S_2-edge, we will make sure that we are in the right conditions to call upon Lemma 3.4, so that (Layer[r]) is preserved.

Definition 4.1. Note that the size of any r-neighborhood of degree at most d is bounded by a function N of d and r; namely by $N(d, r) := d \cdot \frac{(d-1)^r - 1}{d-2} + 1$ if $d \neq 2$, and by $N(2, r) := 2r + 1$ if $d = 2$.

4.2 Separation between rare and frequent types

Knowing the values of r and t as defined in Section 4.1, we are now able to divide the r-types into two categories: the **rare types** and the **frequent types**. The intent is that the two structures have the same number of occurrences of every rare type, and that frequent types have many occurrences (wrt. the total number of occurrences of rare types) in both structures. This "many occurrences wrt." is formalized through a function g which is to be specified later on.

More precisely,

Lemma 4.2. *Given $d, r \in \mathbb{N}$ and an increasing function $g : \mathbb{N} \to \mathbb{N}$, there exists $p \in \mathbb{N}$ such that for every Σ-structures $\mathcal{G}_1, \mathcal{G}_2 \in C_d$ satisfying $\mathcal{G}_1 \equiv^{FO}_p \mathcal{G}_2$, we can divide the r-types over Σ of degree at most d into rare types and frequent types, such that*

- *every rare type has the same number of occurrences in $\mathcal{G}_1$ and in $\mathcal{G}_2$*

- *both in $\mathcal{G}_1$ and in $\mathcal{G}_2$, every frequent type has at least $g(\beta)$ occurrences, where β is the number of occurrences of all the rare types in the structure*
- *if there is no frequent type, then $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic.*

Proof. Let $\chi_1, \cdots, \chi_n$ be an enumeration of all the r-types over Σ of degree at most d, ordered in such a way that $\forall i < j, |\mathcal{G}_1|_{\chi_i} \leq |\mathcal{G}_1|_{\chi_j}$. Note that n is a function of d and r.

The classification of types between rare ones and frequent ones is done through Algorithm 1.

Algorithm 1 Separation between rare and frequent types

```
$\beta \leftarrow 0$
$i \leftarrow 1$
while $i \leq n$ and $|\mathcal{G}_1|_{\chi_i} < g(\beta)$ do
    $\beta \leftarrow \beta + |\mathcal{G}_1|_{\chi_i}$
    $i$++
end while
                        ▷ If $i \leq n$, $\chi_i$ is the frequent type with
                          the least occurrences in $\mathcal{G}_1$.
                          If $i = n + 1$, all the types are rare.
```

At the end of Algorithm 1, we call $\chi_1, \cdots, \chi_{i-1}$ the rare types, and $\chi_i, \cdots, \chi_n$ the frequent ones.

Note that β indeed counts the total number of occurrences of rare types in $\mathcal{G}_1$.

We now define the integers $(a_i)_{1 \leq i \leq n}$ as $a_1 := g(0)$ and

$$a_{i+1} := \max\{a_i, g(ia_i)\}\,.$$

As g is monotone, it is easy to show by induction that for each rare type χ_j with $j < i$, $|\mathcal{G}_1|_{\chi_j} < a_j$.

As long as p is chosen large enough so that $\mathcal{G}_1 \equiv_p^{\mathrm{FO}} \mathcal{G}_2$ entails $[\![\mathcal{G}_1]\!]_r \ =^{a_n} \ [\![\mathcal{G}_2]\!]_r$, we have by construction that every rare type has the same number of occurrences (which is smaller that a_n) in $\mathcal{G}_1$ and in $\mathcal{G}_2$. Furthermore, in $\mathcal{G}_1$ as in $\mathcal{G}_2$, if β denotes the total number of occurrences of rare types, every frequent type has at least $g(\beta)$ occurrences.

We just need to make sure that the two structures are isomorphic when all the types are rare. If this is the case, then $|\mathcal{G}_1| = |\mathcal{G}_2| \leq n(a_n - 1)$. Hence, as long as $p \geq n(a_n - 1)$, $\mathcal{G}_1 \equiv_p^{\mathrm{FO}} \mathcal{G}_2$ implies $\mathcal{G}_1 \simeq \mathcal{G}_2$ when all the types are rare. □

Let $\tau_0, \cdots, \tau_{m-1}$ be the frequent types. From now on, we suppose that $m \geq 1$: there is nothing to do if $m = 0$, since $\mathcal{G}_1$ and $\mathcal{G}_2$ are isomorphic. Let β be the total number of occurrences of rare types in $\mathcal{G}_1$.

4.3 Construction of S_1 around elements of rare type

To begin with, let's focus on $\mathcal{G}_1$, and start the construction of S_1 around occurrences of rare types. Algorithm 2 deals with this construction.

For a given occurrence x of some rare type, we choose as its S_1-successor and S_1-predecessor two occurrences of type τ_0 (the first frequent type), far apart from one another and from x. The existence of those elements relies on the bounded degree hypothesis. This is done on lines 8 and 11.

When line 14 is reached, every occurrence of rare type has an S_1-predecessor and an S_1-successor of type τ_0.

It is not enough, however, only to deal with the occurrences of rare types. We need to "protect" them up to distance r in $(\mathcal{G}_1, S_1)$. For that purpose, we construct the subsets R_k of G_1, for $0 \leq k \leq r$. In the following, $R_{\leq k}$ denotes $\bigcup_{0 \leq j \leq k} R_j$.

For each k, the subset R_k is constructed in order to be the set of elements at distance exactly k in $(\mathcal{G}_1, S_1)$ from the set of occurrences of rare types. Until we have reached $k = r$ (that is, distance r from occurrences of rare types), every element of R_k is given an S_1-successor (line 21) and/or an S_1-predecessor (line 26) of its type, if it doesn't already have one. Once again, those elements are required to be far (i.e. at distance greater than $2r + 1$) from what already has been constructed.

Provided that g is large enough, it is always possible to find x^+ and x^- on lines 8, 11, 21 and 26. Indeed, all the types considered are frequent ones, and the size of the $(2r + 1)$-neighborhood of $R_{\leq k+1}$ is bounded by a function of d, r and β (the total number of occurrences of rare types in $\mathcal{G}_1$). More precisely, at any point of the construction, $(\mathcal{G}_1, S_1)$ has degree at most $d + 1$. Hence, the $(2r + 1)$-neighborhood of R_r has size at most

$$\beta N(d + 1, 3r + 1)$$

(recall the definition of N from Definition 4.1), and it is enough to make sure that

$$g(\beta) \geq \beta N(d + 1, 3r + 1) + 1\,.$$

4.4 Construction of S_1 around the junctions between two frequent types

Recall that there is a second kind of singular elements: those which will be at the junction between two successive frequent types. That is, elements of type τ_i that will, in the final structure $(\mathcal{G}_1, S_1)$, have an S_1-successor of type $\tau_{i+1[m]}$, or an S_1-predecessor of type $\tau_{i-1[m]}$.

Those junction elements need to be treated in a similar way as the occurrences of rare types in Section 4.3. This construction is done throughout Algorithm 3.

The idea of Algorithm 3 is very similar to that of Algorithm 2. We start by picking two elements x_i^+ and x_i^- for every frequent type τ_i (for loop line 2), that are far from each other and from the previous construction.

Then we build m S_1-edges between those elements on line 9: these edges are intended to be at the junction between the frequent types in the final structure.

The set P_0 of those $2m$ elements will have the same role as the set R_0 of occurrences of rare types for Algorithm 2: we build S_1-edges at depth r around it. This is done through the subsets P_k of G_1, for $0 \leq k \leq r$, P_k being the set of elements

Algorithm 2 Construction of S_1 around elements of rare type

```
 1: $S_1 \leftarrow \emptyset$
 2: $R_0 \leftarrow \{x \in G_1 : \mathrm{tp}^r_{\mathcal{G}_1}(x)$ is rare$\}$
 3: $R_1, \cdots, R_r \leftarrow \emptyset$
 4: for all $x \in R_0$ do
 5:     for all neighbor $y \notin R_{\leq 1}$ of $x$ in $\mathcal{G}_1$ do
 6:         $R_1 \leftarrow R_1 \cup \{y\}$
 7:     end for
 8:     find $x^+$ such that
            $\mathrm{tp}^r_{\mathcal{G}_1}(x^+) = \tau_0$ and
            $\mathrm{dist}_{(\mathcal{G}_1,S_1)}(x^+, R_{\leq 1}) > 2r+1$
                        ▹ We pick a node at distance greater than $2r+1$ in compliance with Lemma 3.4, so that neighborhoods stay layered. Recall that $\tau_0$ is the first frequent type.
 9:     $R_1 \leftarrow R_1 \cup \{x^+\}$
10:     $S_1 \leftarrow S_1 \cup \{(x, x^+)\}$
11:     find $x^-$ such that
            $\mathrm{tp}^r_{\mathcal{G}_1}(x^-) = \tau_0$ and
            $\mathrm{dist}_{(\mathcal{G}_1,S_1)}(x^-, R_{\leq 1}) > 2r+1$
12:     $R_1 \leftarrow R_1 \cup \{x^-\}$
13:     $S_1 \leftarrow S_1 \cup \{(x^-, x)\}$
14: end for
                        ▹ At this point, every element of rare type has an $S_1$-predecessor and an $S_1$-successor of type $\tau_0$
15: for $k$ from 1 to $r-1$ do
16:     for all $x \in R_k$ do          ▹ $\mathrm{tp}^k_{\mathcal{G}_1}(x)$ is a frequent type
17:         for all neighbor $y \notin R_{\leq k+1}$ of $x$ in $\mathcal{G}_1$ do
18:             $R_{k+1} \leftarrow R_{k+1} \cup \{y\}$
19:         end for
20:         if $x$ doesn't have a successor by $S_1$ then
21:             find $x^+$ such that
                    $\mathrm{tp}^r_{\mathcal{G}_1}(x^+) = \mathrm{tp}^r_{\mathcal{G}_1}(x)$ and
                    $\mathrm{dist}_{(\mathcal{G}_1,S_1)}(x^+, R_{\leq k+1}) > 2r+1$
22:             $R_{k+1} \leftarrow R_{k+1} \cup \{x^+\}$
23:             $S_1 \leftarrow S_1 \cup \{(x, x^+)\}$
24:         end if
25:         if $x$ doesn't have a predecessor by $S_1$ then
26:             find $x^-$ such that
                    $\mathrm{tp}^r_{\mathcal{G}_1}(x^-) = \mathrm{tp}^r_{\mathcal{G}_1}(x)$ and
                    $\mathrm{dist}_{(\mathcal{G}_1,S_1)}(x^-, R_{\leq k+1}) > 2r+1$
27:             $R_{k+1} \leftarrow R_{k+1} \cup \{x^-\}$
28:             $S_1 \leftarrow S_1 \cup \{(x^-, x)\}$
29:         end if
30:     end for
31: end for
```

at distance k from P_0 in $(\mathcal{G}_1, S_1)$. Once again, $P_{\leq k}$ denotes $\bigcup\limits_{0 \leq j \leq k} P_j$.

For the same reason as for Algorithm 2, it is always possible to find elements x^+ and x^- on lines 17 and 22.

Note that if $m = 1$, there is obviously no transition elements: we simply construct an S_1-edge between x_0^+ and x_0^-.

Algorithm 3 Construction of S_1 around the junctions between two frequent types

```
 1: $P_0, \cdots, P_r \leftarrow \emptyset$
 2: for $i$ from 0 to $m-1$ do
 3:     find $x_i^+$ such that
            $\mathrm{tp}^r_{\mathcal{G}_1}(x_i^+) = \tau_i$ and
            $\mathrm{dist}_{(\mathcal{G}_1,S_1)}(x_i^+, R_{\leq r} \cup P_0) > 2r+1$
 4:     $P_0 \leftarrow P_0 \cup \{x_i^+\}$
 5:     find $x_i^-$ such that
            $\mathrm{tp}^r_{\mathcal{G}_1}(x_i^-) = \tau_i$ and
            $\mathrm{dist}_{(\mathcal{G}_1,S_1)}(x_i^-, R_{\leq r} \cup P_0) > 2r+1$
 6:     $P_0 \leftarrow P_0 \cup \{x_i^-\}$
 7: end for
 8: for $i$ from 0 to $m-1$ do
 9:     $S_1 \leftarrow S_1 \cup \{(x_i^-, x_{i+1[m]}^+)\}$
10: end for
11: for $k$ from 0 to $r-1$ do
12:     for all $x \in P_k$ do
13:         for all neighbor $y \notin P_{\leq k+1}$ of $x$ in $\mathcal{G}_1$ do
14:             $P_{k+1} \leftarrow P_{k+1} \cup \{y\}$
15:         end for
16:         if $x$ doesn't have a successor by $S_1$ then
17:             find $x^+$ such that
                    $\mathrm{tp}^r_{\mathcal{G}_1}(x^+) = \mathrm{tp}^r_{\mathcal{G}_1}(x)$ and
                    $\mathrm{dist}_{(\mathcal{G}_1,S_1)}(x^+, R_{\leq r} \cup P_{\leq k+1}) > 2r+1$
18:             $P_{k+1} \leftarrow P_{k+1} \cup \{x^+\}$
19:             $S_1 \leftarrow S_1 \cup \{(x, x^+)\}$
20:         end if
21:         if $x$ doesn't have a predecessor by $S_1$ then
22:             find $x^-$ such that
                    $\mathrm{tp}^r_{\mathcal{G}_1}(x^-) = \mathrm{tp}^r_{\mathcal{G}_1}(x)$ and
                    $\mathrm{dist}_{(\mathcal{G}_1,S_1)}(x^-, R_{\leq r} \cup P_{\leq k+1}) > 2r+1$
23:             $P_{k+1} \leftarrow P_{k+1} \cup \{x^-\}$
24:             $S_1 \leftarrow S_1 \cup \{(x^-, x)\}$
25:         end if
26:     end for
27: end for
```

4.5 Carrying S_1 over to $\mathcal{G}_2$

In Sections 4.3 and 4.4, S_1 has been constructed around the singular points of $\mathcal{G}_1$, i.e. occurrences of rare types and elements that are to make the junction between two S_1-segments of frequent types.

Before we extend S_1 to the remaining elements (all of them being occurrences of frequent types) of $\mathcal{G}_1$, we carry it over to $\mathcal{G}_2$. This transfer is possible under the starting hypothesis that $\mathcal{G}_1$ and $\mathcal{G}_2$ are FO-similar.

Let

$$\begin{cases} A_1 := R_{\leq r} \cup P_{\leq r} \\ B := \{x \in G_1 : \mathrm{dist}_{(\mathcal{G}_1,S_1)}(x, A_1) \leq r\} \end{cases}$$

If we let t_r^d be the number of r-types of degree at most d over Σ, we must have that $m \leq n$ thus $|A_1|$ can be bounded by

$$(\beta + 2t_r^d)N(d+1, r)\,.$$

Similarly, the size of B can be bounded by

$$(\beta + 2t_r^d)N(d+1, 2r)\,,$$

which is a function of β, r and d. Hence as long as $f(\alpha)$ is larger than that number, the Duplicator has a winning strategy in the Ehrenfeucht-Fraïssé game between $\mathcal{G}_1$ and $\mathcal{G}_2$ in which the Spoiler chooses every element of B. Let $h : B \to G_2$ be the function resulting from such a strategy.

h defines an isomorphism from $\mathcal{G}_1|_B$ to $\mathcal{G}_2|_{\mathrm{Im}(h)}$. Let $A_2 := h(A_1)$. By taking $f(\alpha)$ one higher than required, to make sure that $\mathrm{Im}(h)$ covers the r-neighborhood in $\mathcal{G}_2$ of every element of A_2, we have that for every $x \in A_1$, $\mathrm{tp}^r_{\mathcal{G}_2}(h(x)) = \mathrm{tp}^r_{\mathcal{G}_1}(x)$.

We set $S_2 := \{(h(x), h(y)) : (x, y) \in S_1\}$. h now defines an isomorphism from $(\mathcal{G}_1, S_1)|_B$ to $(\mathcal{G}_2, S_2)|_{\mathrm{Im}(h)}$, and for every $x \in A_1$, $\mathrm{tp}^r_{(\mathcal{G}_2,S_2)}(h(x)) = \mathrm{tp}^r_{(\mathcal{G}_1,S_1)}(x)$.

4.6 Completion of S_1 and S_2

Now that S_1 and S_2 are constructed around all the singular points both in $\mathcal{G}_1$ and $\mathcal{G}_2$, it remains to extend their construction to all the other elements of the structures. Recall that all the remaining elements are occurrences of frequent types.

From $(\mathcal{G}_\epsilon, S_\epsilon)$ ($\epsilon \in \{1, 2\}$) at any point in the construction, let's define the partial function $S^*_\epsilon : G_\epsilon \to G_\epsilon$ that maps $x \in G_\epsilon$ to the (unique) y that is S_ϵ-reachable (while taking the orientation into account) from x and that doesn't have an S_ϵ-successor. This function is defined on every element that doesn't belong to an S_ϵ-cycle (and in particular, on every element without an S_ϵ-predecessor).

Likewise, we define S^{-*}_ϵ by reversing the arrows of S_ϵ.

At this point, for every $x \notin A_1$, $S^*_1(x) = S^{-*}_1(x) = x$, and for every $x \notin A_2$, $S^*_2(x) = S^{-*}_2(x) = x$.

We now run Algorithm 4. We first treat $\mathcal{G}_1$, and then apply a similar method to $\mathcal{G}_2$, replacing x_i^+ and x_i^- by $h(x_i^+)$ and $h(x_i^-)$. The idea is, for every frequent type τ_i, to insert all its remaining occurrences between (in the sense of S_1) x_i^+ and x_i^-.

The first approach (the loop at line 2) is greedy: while constructing S_ϵ on nodes of type τ_i, we choose as the successor of the current node any occurrence of τ_i that is at distance greater than $2r + 1$ from the current node s and the closing node of type τ_i, $S^{-*}_\epsilon(x_i^-)$. This, together with Lemma 3.4, ensures that (Layer[r]) holds after every addition. The conditions line 11 also ensure that the final edge addition, line 15, doesn't break (Layer[r]).

Once we cannot apply this greedy approach anymore, we know that only a small number (which can be bounded by $2N(d + 1, 2r + 1)$) of nodes of type τ_i remain without S_1-predecessor. The loop at line 17 considers one such node x at a time. As long as g is large enough, we have constructed S_1 around enough elements of type τ_i in the greedy approach to ensure the existence of some $S_1(y, z)$, with y, z of type τ_i and at distance greater than $2r + 1$ from x; x is inserted between y and z (line 20). For that, it is enough to have constructed at least

$$2N(d + 1, 2r + 1) + 1$$

S_1-edges in the greedy phase. This is the case in particular when there are at least

$$4N(d + 1, 2r + 1) + 1$$

elements of type τ_i without S_1-predecessor at the beginning of Algorithm 4, which can be ensured by having

$$g(\beta) \geq |A_1| + 4N(d + 1, 2r + 1) + 1\,.$$

This holds in particular when

$$g(\beta) \geq (\beta + 2t_r^d)N(d + 1, r) + 4N(d + 1, 2r + 1) + 1\,.$$

We will prove in Lemma 4.6 that all these insertions preserve (Layer[r]).

Algorithm 4 Completion of S_ϵ

```
 1: for $\epsilon$ from 1 to 2 do
 2:     for $i$ from 0 to $m - 1$ do
 3:         if $\epsilon = 1$ then
 4:             $s \leftarrow S^*_1(x_i^+)$
 5:             $t \leftarrow S^{-*}_1(x_i^-)$
 6:         else
 7:             $s \leftarrow S^*_2(h(x_i^+))$
 8:             $t \leftarrow S^{-*}_2(h(x_i^-))$
 9:         end if
10:         while such an $x$ exists do
11:             find $x$ with no $S_\epsilon$-predecessor, such that
                    $\mathrm{tp}^r_{\mathcal{G}_\epsilon}(x) = \tau_i$,
                    $\mathrm{dist}_{(\mathcal{G}_\epsilon,S_\epsilon)}(s, x) > 2r + 1$,
                    $\mathrm{dist}_{(\mathcal{G}_\epsilon,S_\epsilon)}(x, t) > 2r + 1$ and
                    $\mathrm{dist}_{(\mathcal{G}_\epsilon,S_\epsilon)}(S^*_\epsilon(x), t) > 2r + 1$
12:             $S_\epsilon \leftarrow S_\epsilon \cup \{(s, x)\}$
13:             $s \leftarrow S^*_\epsilon(x)$
14:         end while
                ▷ At this point, only a bounded number of elements of type $\tau_i$ are left without an $S_\epsilon$-predecessor
15:         $S_\epsilon \leftarrow S_\epsilon \cup \{(s, t)\}$
16:     end for
17:     for $i$ from 0 to $m - 1$ do
18:         for all $x$ without $S_\epsilon$-predecessor, s.t. $\mathrm{tp}^r_{\mathcal{G}_\epsilon}(x) = \tau_i$ do
19:             find $y, z \notin A_\epsilon$ such that
                    $\mathrm{tp}^r_{\mathcal{G}_\epsilon}(y) = \mathrm{tp}^r_{\mathcal{G}_\epsilon}(z) = \tau_i$,
                    $(y, z) \in S_\epsilon$,
                    $\mathrm{dist}_{(\mathcal{G}_\epsilon,S_\epsilon)}(y, x) > 2r + 1$ and
                    $\mathrm{dist}_{(\mathcal{G}_\epsilon,S_\epsilon)}(S^*_\epsilon(x), z) > 2r + 1$
20:             $S_\epsilon \leftarrow S_\epsilon \setminus \{(y, z)\} \cup \{(y, x), (S^*_\epsilon(x), z)\}$
21:         end for
22:     end for
23: end for
```

4.7 Examples of construction

Before we give the proof of correctness of these algorithms, let us see how they apply in some simple cases

Example 4.3. Suppose that there are no occurrences of rare types, and only one frequent type τ_0, and assume $r = 2$.

In this case, Algorithm 2 is irrelevant, and all Algorithm 3 does is pick x_0^- and x_0^+ far from each other, and start building S_1 around those nodes in order to construct their complete r-neighborhood in $(\mathcal{G}_1, S_1)$. In order to make the figure more readable, let us consider that x_0^- and x_1^+ have only one neighbor. In Figure 5, the plain lines represent edges in $\mathcal{G}_1$, and the dashed arrows represent S_1.

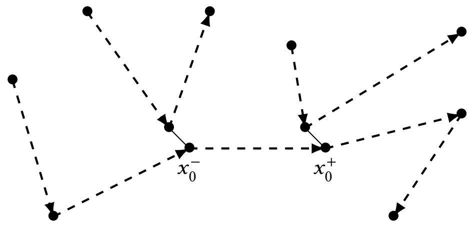

Figure 5. After Algorithm 3, with one frequent type

We now apply Algorithm 4. The first step is to add elements between (in the sense of S_1) $S_1^*(x_0^+)$ and $S_1^{-*}(x_0^-)$ in order to join them, in a greedy fashion. Once this is done, there only remain a few elements that haven't been assigned an S_1-predecessor. This is depicted in Figure 6.

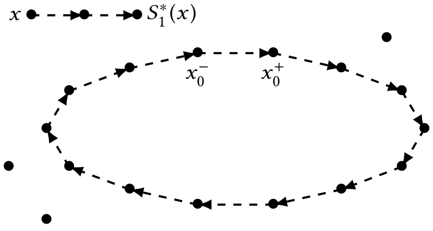

Figure 6. After the greedy part of Algorithm 4, with one frequent type

Now we consider one by one each of the elements that don't have an S_1-predecessor: let's start with x in Figure 6. Our goal is to insert it in the S_1-cycle while still respecting (Layer[r]). For that, we find two successive elements y, z of the cycle that are far from x and $S_1^*(x)$, and we insert x between them, as shown in Figure 7.

We treat all the elements without an S_1-predecessor in the same way, until S_1 is fully built.

Example 4.4. Suppose now that there are two frequent types τ_0 and τ_1, and still no occurrences of rare types.

The procedure is very similar: in Algorithm 3, we build the r-neighborhood in $(\mathcal{G}_1, S_1)$ of the four nodes x_0^-, x_0^+, x_1^- and x_1^+.

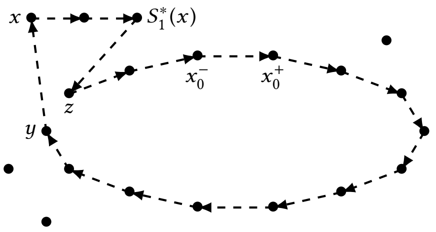

Figure 7. Inserting x in the S_1-cycle, as in the second part of Algorithm 4, with one frequent type

After the greedy part of Algorithm 4, S_1 looks like in Figure 8, where occurrences of τ_0 are represented as • and occurrences of τ_1 as ∘. The remaining of Algorithm 4 is as unchanged.

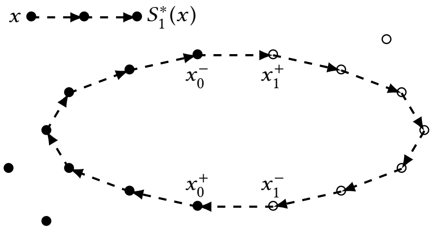

Figure 8. After the greedy part of Algorithm 4, with two frequent types

Note that if there existed some occurrences of rare types, they would be embedded in the τ_0 part of the S_1-cycle.

4.8 Properties of S_1 and S_2

We are now ready to show that, after the successive run of Algorithms 2, 3 and 4:

- S_1 and S_2 are indeed successor relations (Lemma 4.5)
- $(\mathcal{G}_1, S_1)$ and $(\mathcal{G}_2, S_2)$ satisfy (Layer[r]) (Lemma 4.6)
- A singular element (around a rare or a junction element) of $(\mathcal{G}_1, S_1)$ and its corresponding element via h in $(\mathcal{G}_2, S_2)$ have the same r-type (Lemma 4.9), while any other element in both structures has a regular (i.e. fractal) r-type (Lemma 4.8)

These properties will allow us to prove in Section 4.9 that $(\mathcal{G}_1, S_1)$ and $(\mathcal{G}_2, S_2)$ have the same number of occurrences of every r-type, up to a threshold t.

Lemma 4.5. *S_1 (resp. S_2) is a successor relation on G_1 (resp. G_2).*

Proof. This result is rather transparent, but a rigorous proof requires the usage of a somewhat cumbersome invariant.

Let us focus on $\mathcal{G}_1$: the proof is the same for $\mathcal{G}_2$, replacing every x_i^+ and x_i^- with $h(x_i^+)$ and $h(x_i^-)$.

Let $a \in G_1$ be defined as $S_1^{-*}(x_{m-1}^-)$ at the beginning of Algorithm 4. By construction, $\mathrm{tp}^r_{\mathcal{G}}(a) = \tau_{m-1}$ and a has no S_1-predecessor as of now.

We show that at any point before line 15 of the loop iteration $i = m - 1$ of Algorithm 4,

(i). $S_1^{-*}(s) = a$
(ii). $S_1^{-*}(x_{m-1}^-) = a$
(iii). let $y, z \notin A_1$ such that $(y, z) \in S_1$ and

$$\mathrm{tp}^r_{\mathcal{G}_1}(y) = \mathrm{tp}^r_{\mathcal{G}_1}(z) = \tau_j$$

for some j; then $S_1^{-*}(y) = a$
(iv). for every i, $(x_i^-, x_{i+1[m]}^+) \in S_1$
(v). there is no S_1-cycle
(vi). for every $j > i$, $\mathrm{tp}^r_{\mathcal{G}_1}(S_1^{-*}(x_j^-)) = \tau_j$

This is obviously satisfied at the beginning of Algorithm 4: there are not yet such y, z as in (iii), and $s = S_1^*(x_0^+)$ is S_1-reachable from x_{m-1}^- (since $(x_{m+1}^-, x_0^+) \in S_1$) hence (i) holds.

Line 4 preserves the invariant. Indeed, the new value of s is S_1-reachable from its previous value (this is guaranteed by (iv)), which means that they have the same image through S_1^{-*}, namely a.

Let's prove that line 12 preserves the invariant. (i) and (ii) still hold since $x \neq a$: indeed, for $i < m - 1$, x and a don't share the same type, while for $i = m-1$, $a = t$ (because of (ii)) and the distance condition prohibits $x = a$. (iii) still holds, as the only new possibility for such a couple (y, z) is (s, x), which is such that $S_1^{-*}(y) = a$ (because of (i)). (iv) obviously holds, as does (v), since the only way for an S_1-cycle to have been created is if $x = S_1^{-*}(s)$, that is $x = a$. We have seen that this is absurd. (vi) is satisfied, as the only way for it to fail is for x to be some $S_1^{-*}(x_j^-)$, for $j > i$, which is impossible due to type requirements.

Now, let's move to line 13. Only (i) needs verification, and the argument is the same as for line 4.

Finally, let's look at line 15, for $i < m - 1$. $t \neq a$ since their types are different, hence (i), (ii) and (v) still hold. (iii) still holds, as the only new possibility for such a couple (y, z) is (s, t), which is such that $S_1^{-*}(y) = a$ because of (i) (actually, (s, t) doesn't even fit the condition, since $t \in A_1$). (iv) is still satisfied. (vi) holds, as the only way for it to fail is for t to be some $S_1^{-*}(x_j^-)$, for $j > i$, which is impossible due to type requirements.

We now prove that from line 17 until the end of Algorithm 4, there is exactly one S_1-cycle, which contains every $y, z \notin A_1$ such that $(y, z) \in S_1$ and $\mathrm{tp}^r_{\mathcal{G}_1}(y) = \mathrm{tp}^r_{\mathcal{G}_1}(z) = \tau_j$ for some j.

This is true after line 15 of the loop iteration $m - 1$, which creates the first S_1-cycle, as (i) and (ii) ensure $t = a = S_1^{-*}(s)$. (iii) guarantees that this newly created S_1-cycle contains all the couple (y, z) satisfying the condition.

It remains to show that line 20 preserves this property: by hypothesis, y and z belong to the S_1-cycle. After line 20, there is still exactly one S_1-cycle, which corresponds to the previous one where the S_1-edge has been replaced by the S_1-segment $[x, S_1^*(x)]$. The only S_1-edges that have been added belong to the S_1-cycle, hence the second part of the property still holds.

In the end, every element of G_1 has a predecessor by S_1, hence S_1 is a permutation of G_1. We've shown that it has a single orbit. □

Lemma 4.6. (Layer[r]) *holds in* $(\mathcal{G}_\epsilon, S_\epsilon)$, *for* $\epsilon \in \{1, 2\}$

Proof. This property is guaranteed by the distance conditions of the form "$\mathrm{dist}_{(\mathcal{G}_\epsilon, S_\epsilon)}(.,.) > 2r + 1$" imposed throughout Algorithms 2, 3 and 4, and by Lemma 3.4.

One can very easily verify that (Layer[r]) is guaranteed by Lemma 3.4 to hold in $(\mathcal{G}_\epsilon, S_\epsilon)$ prior to the run of Algorithm 4.

We focus on Algorithm 4, and we use Lemma 3.4 to prove that (Layer[r]) remains valid in $(\mathcal{G}_\epsilon, S_\epsilon)$ throughout its run. There are three edge additions we have to prove correct:

- For the edge addition of line 12, this follows directly from Lemma 3.4.
- For the edge addition of line 15, we show that the invariant $\mathrm{dist}_{(\mathcal{G}_\epsilon, S_\epsilon)}(s, t) > 2r + 1$ is satisfied at the beginning and at the end of the while line 10. This invariant, together with Lemma 3.4, will be enough to conclude.
  The invariant holds before the first execution of the while loop (except for $m = 1$, where it only bootstraps after two executions of the loop).
  Working towards a contradiction, suppose that the invariant is broken during an execution of the loop. We use the pre notations. There must exists a path from $S_\epsilon^*(x)$ (which is to become the new value of s at the end of the loop) to t in $(\mathcal{G}_\epsilon, S_\epsilon \cup \{(s, x)\})$ of length at most $2r + 1$; consider a shortest one. As is cannot be valid in $(\mathcal{G}_\epsilon, S_\epsilon)$ by choice of x, it must go through the newly added edge (s, x). This means that in $(\mathcal{G}_\epsilon, S_\epsilon)$, either there exist paths of length at most $2r + 1$ from $S_\epsilon^*(x)$ to s and from x to t, or paths of length at most $2r + 1$ from $S_\epsilon^*(x)$ to x and from s to t. The former is absurd considering the way x was chosen, and the latter contradicts the previous invariant.
- Let's prove that the addition of the two S_ϵ-edges of line 20 doesn't break (Layer[r]). By choice of y, we know that $\mathrm{dist}_{(\mathcal{G}_\epsilon, S_\epsilon)}(y, x) > 2r + 1$. *A fortiori*, we must have $\mathrm{dist}_{(\mathcal{G}_\epsilon, S_\epsilon \setminus \{(y,z)\})}(y, x) > 2r + 1$, and Lemma 3.4 ensures that $(\mathcal{G}_\epsilon, S_\epsilon \setminus \{(y, z)\} \cup \{(y, x)\})$ satisfies the property (Layer[r]).
  Now, to the second addition: let's prove that, at the beginning of line 20, $\mathrm{dist}_{(\mathcal{G}_\epsilon, S_\epsilon \setminus \{(y,z)\} \cup \{(y,x)\})}(S_\epsilon^*(x), z) > 2r + 1$. We then conclude with Lemma 3.4.
  Suppose it's not the case and consider a shortest path from $S_\epsilon^*(x)$ to z, which must be of length at most $2r+1$. This path cannot be valid in $(\mathcal{G}_\epsilon, S_\epsilon)$, thus it has to go through the new edge (y, x). Since there cannot exist a path of length at most $2r$ from $S_\epsilon^*(x)$ to y in $(\mathcal{G}_\epsilon, S_\epsilon)$

(as this would contradict $\mathrm{dist}_{(\mathcal{G}_\epsilon,S_\epsilon)}(S^*_\epsilon(x),z) > 2r+1$), it has to borrow the edge from x to y.
Then in $(\mathcal{G}_\epsilon,S_\epsilon \setminus \{(y,z)\})$, there is a path of length at most $2r$ from y to z, which contradicts `(Layer[r])` in $(\mathcal{G}_\epsilon,S_\epsilon)$.

□

The following Lemma states that the only time S_ϵ joins two nodes that have different r-types in $\mathcal{G}_\epsilon$ is when one of them is an occurrence of a rare type (in which case its S_ϵ-predecessor and S_ϵ-successor are of type τ_0) or when they are the elements which make the transition between two frequent types (that is, one is x_i^- and the other is $x^+_{i+1[m]}$, for some $i < m$):

Lemma 4.7. *$\forall x,y \in G_1$ such that $(x,y) \in S_1$ and ($x \notin R_0$ and $y \notin R_0$) and ($x \notin P_0$ or $y \notin P_0$), then $tp^r_{\mathcal{G}_1}(x) = tp^r_{\mathcal{G}_1}(y)$*

$\forall x,y \in G_2$ such that $(x,y) \in S_2$ and ($x \notin h(R_0)$ and $y \notin h(R_0)$) and ($x \notin h(P_0)$ or $y \notin h(P_0)$), then $tp^r_{\mathcal{G}_2}(x) = tp^r_{\mathcal{G}_2}(y)$

Proof. The property clearly holds at the end of Algorithm 2 and Algorithm 3.

For any i from 0 to $m-1$, the only S_1-edges (resp. S_2-edges) that are added during the i-th loop are between two nodes of type τ_i. □

Recall the discussion at the beginning of Section 3. We now prove that, as long as an element is far from any occurrence of a rare type and from the elements that make the transition between two frequent types, its type in $(\mathcal{G}_\epsilon,S_\epsilon)$ is the fractal of its type in $\mathcal{G}_\epsilon$:

Lemma 4.8. *For $\epsilon \in \{1,2\}$ and for every $0 \leq k \leq r$ and $x \notin R_{\leq k} \cup P_{\leq k}$ (if $\epsilon = 1$) or $x \notin h(R_{\leq k} \cup P_{\leq k})$ (if $\epsilon = 2$),*

$$tp^k_{(\mathcal{G}_\epsilon,S_\epsilon)}(x) = [tp^k_{\mathcal{G}_\epsilon}(x)]_k\,.$$

Proof. We prove the result by induction on k. For $k = 0$, there is nothing to do but note that no edge $S_\epsilon(x,x)$ has been created.

Suppose that we've proven the result for some $k < r$, and let $x \notin R_{\leq k+1} \cup P_{\leq k+1}$, or $x \notin h(R_{\leq k+1} \cup P_{\leq k+1})$.

Let y be such that $\mathrm{dist}_{\mathcal{G}_\epsilon}(x,y) = d$, for some $1 \leq d \leq k+1$. By construction of the R_i and P_i, and of h, we have that $y \notin R_{\leq k+1-d} \cup P_{\leq k+1-d}$, or $y \notin h(R_{\leq k+1-d} \cup P_{\leq k+1-d})$ (this is easily shown by induction on d). Hence, $\mathrm{tp}^{k+1-d}_{(\mathcal{G}_\epsilon,S_\epsilon)}(y) = [\mathrm{tp}^{k+1-d}_{\mathcal{G}_\epsilon}(y)]_{k+1-d}$.

Because Lemma 4.6 ensures that the $(k+1)$-neighborhood of x in $(\mathcal{G}_\epsilon,S_\epsilon)$ is layered, it only remains to show that the S_ϵ-successor x^+ and predecessor x^- of x are such that $\mathrm{tp}^k_{(\mathcal{G}_\epsilon,S_\epsilon)}(x^+) = \mathrm{tp}^k_{(\mathcal{G}_\epsilon,S_\epsilon)}(x^-) = [\mathrm{tp}^k_{\mathcal{G}_\epsilon}(x)]_k$. Let's show this for x^+.

Lemma 4.7 ensures $\mathrm{tp}^r_{\mathcal{G}_\epsilon}(x^+) = \mathrm{tp}^r_{\mathcal{G}_\epsilon}(x)$. It only remains to note that $x^+ \notin R_{\leq k} \cup P_{\leq k}$, or $x^+ \notin h(R_{\leq k} \cup P_{\leq k})$, and the induction hypothesis allows us to conclude. □

When we first defined h, it preserved r-types by construction. The last step before we are able to conclude the proof of Theorem 2.5 is to make sure that h still preserves r-types, taking into account the S_ϵ-edges added during the run of Algorithm 4:

Lemma 4.9. *$\forall x \in A_1, tp^r_{(\mathcal{G}_2,S_2)}(h(x)) = tp^r_{(\mathcal{G}_1,S_1)}(x)$*

Proof. We prove by induction on $0 \leq k \leq r$ that $\forall x \in A_1, \mathrm{tp}^k_{(\mathcal{G}_2,S_2)}(h(x)) = \mathrm{tp}^k_{(\mathcal{G}_1,S_1)}(x)$

There is nothing to prove for $k = 0$.

Moving from k to $k+1$, let $x \in A_1$ and let y be such that $\mathrm{dist}_{\mathcal{G}_1}(x,y) = d$, for some $1 \leq d \leq k+1$. Note that $y \in B$, hence it has an image by h.

If $y \in A_1$, the induction hypothesis allows us to conclude that $\mathrm{tp}^{k+1-d}_{(\mathcal{G}_2,S_2)}(h(y)) = \mathrm{tp}^{k+1-d}_{(\mathcal{G}_1,S_1)}(y)$.

Else, Lemma 4.8 ensures that:

$$\mathrm{tp}^r_{(\mathcal{G}_2,S_2)}(h(y)) = [\mathrm{tp}^r_{\mathcal{G}_2}(h(y))]_r = [\mathrm{tp}^r_{\mathcal{G}_1}(y)]_r = \mathrm{tp}^r_{(\mathcal{G}_1,S_1)}(y)\,.$$

In both cases, $\mathrm{tp}^{k+1-d}_{(\mathcal{G}_2,S_2)}(h(y)) = \mathrm{tp}^{k+1-d}_{(\mathcal{G}_1,S_1)}(y)$.

Because of `(Layer[r])`, it only remains to show that the S_ϵ-successors of x and $h(x)$, as well as their S_ϵ-predecessors, have the same k-type in $(\mathcal{G}_\epsilon,S_\epsilon)$. Let's prove this for the successors, respectively named x^+ and $h(x)^+$.

If $x^+ \in A_1$, then by construction $h(x)^+ = h(x^+)$, and the induction hypothesis allows us to conclude.

Otherwise, neither x^+ nor $h(x)^+$ belongs to A_1. Under this hypothesis, Lemma 4.7 ensures that

$$\mathrm{tp}^r_{\mathcal{G}_2}(h(x)^+) = \mathrm{tp}^r_{\mathcal{G}_2}(h(x)) = \mathrm{tp}^r_{\mathcal{G}_1}(x) = \mathrm{tp}^r_{\mathcal{G}_1}(x^+)\,.$$

Now, Lemma 4.8 ensures that

$$\begin{aligned}\mathrm{tp}^r_{(\mathcal{G}_2,S_2)}(h(x)^+) &= [\mathrm{tp}^r_{\mathcal{G}_2}(h(x)^+)]_r\\ &= [\mathrm{tp}^r_{\mathcal{G}_1}(x^+)]_r\\ &= \mathrm{tp}^r_{(\mathcal{G}_1,S_1)}(x^+)\,.\end{aligned}$$

□

4.9 Conclusion of the proof

We are now able to conclude the proof. Recall that we want to prove that $[\![(\mathcal{G}_1,S_1)]\!]_r =^t [\![(\mathcal{G}_2,S_2)]\!]_r$.

Let τ be an r-type over $\Sigma \cup \{S\}$ which occurs in $(\mathcal{G}_1,S_1)$. There are two cases to consider:

- if τ occurs outside of A_1, then Lemma 4.8 ensures that $\tau = [\chi]_r$ for some frequent r-type χ. We can choose g so that χ is guaranteed to have at least t occurrences in $\mathcal{G}_1$ outside of A_1, and in $\mathcal{G}_2$ outside of A_2. This is ensured as long as

  $$g(\beta) \geq |A_1| + t\,,$$

  and in particular when

  $$g(\beta) \geq (\beta + 2t_r^d)N(d+1,r) + t\,.$$

  Lemma 4.8 then ensures that τ occurs at least t times both in $(\mathcal{G}_1,S_1)$ and in $(\mathcal{G}_2,S_2)$.

- if τ occurs only in A_1, then it cannot occur in $(\mathcal{G}_2, S_2)$ outside of A_2 (for the same reasons as above).
  Lemma 4.9 guarantees that τ has the same number of occurrences in A_1 and in A_2, hence in $(\mathcal{G}_1, S_1)$ and in $(\mathcal{G}_2, S_2)$.

Thus, by Hanf threshold theorem, we have shown that for every $\alpha \in \mathbb{N}$, there exists some $f(\alpha) \in \mathbb{N}$ such that for any Σ-structures $\mathcal{G}_1, \mathcal{G}_2$ of degree at most d,

$$\mathcal{G}_1 \equiv^{\text{FO}}_{f(\alpha)} \mathcal{G}_2 \quad \rightarrow \quad \mathcal{G}_1 \equiv^{\text{Succ-inv FO}}_{\alpha} \mathcal{G}_2 \,.$$

This means that on the class of Σ-structures of degree at most d, any equivalence class C for $\equiv^{\text{Succ-inv FO}}_{\alpha}$ is a finite union of equivalence classes for $\equiv^{\text{FO}}_{f(\alpha)}$, and is consequently definable by an FO-sentence φ_C of quantifier rank $f(\alpha)$.

Let $\mathcal{P}$ be a property of structures of degree at most d definable by a Succ-inv FO-sentence of quantifier rank at most α. It is a finite union $\bigcup_i C_i$ of equivalence classes for $\equiv^{\text{Succ-inv FO}}_{\alpha}$. Hence, the FO$[f(\alpha)]$-sentence $\bigvee_i \varphi_{C_i}$ defines $\mathcal{P}$.

This proves the inclusion Succ-inv FO $\subseteq$ FO on structures of degree at most d.

5 Conclusion

We have shown that Succ-inv FO collapses to FO on any class of bounded degree, as well as on classes of graphs which are FO-interpretable in graph classes of bounded degree, namely near-uniform graph classes as defined in [5]. Our proof gives a constructive translation from Succ-inv FO to FO on classes of bounded degree. The quantifier rank of the translated sentence is triple-exponential in the quantifier rank of the original formula. It is an easy exercise to prove that the blowup is at least exponential, but we do not know if an exponential translation is at all possible.

Similar considerations arise when we take into account the length of the sentences instead of their quantifier rank - in this regard, our construction is even non-elementary, and all we know is that the blowup is at least exponential.

An interesting task would be to improve the succinctness of the translation, or to give tighter lower bounds on such constructions.

Apart from these considerations, there are two main directions in which one could look to extend the present result. One possibility would be to keep looking at classes of bounded degree while climbing up in the ladder of expressivity, and ask whether $<$ -inv FO collapses to FO as well on these classes of structures. New techniques would be needed, as contrary to what was the case with a successor, the addition of an order doesn't preserve the bounded degree property. Furthermore, even if $<$-inv FO = FO in this setting, it is not clear whether such orders can be directly constructed. It may be necessary to construct, as in [1], a chain of intermediate structures and orders.

Alternatively, we could change the setting, and study the expressivity of Succ-inv FO on other sparse classes of structures, e.g. on classes of bounded treewidth. If showing the collapse of Succ-inv FO to FO on these classes proved itself to be out of reach, a possibility would be to aim at proving that Succ-inv FO is Hanf-local (which would be stronger than the known Gaifman-locality). In that case, the starting hypothesis on the structures $\mathcal{G}_1$ and $\mathcal{G}_2$ would be stronger, as the existence of a k-type-preserving bijection between the two structures would be assumed.

These tasks are much harder without any bound on the degree, which was what guaranteed that we could find elements of a given frequent type far from each other.

References

[1] Michael Benedikt and Luc Segoufin. Towards a characterization of order-invariant queries over tame graphs. *J. Symb. Log.*, 2009.

[2] Kord Eickmeyer, Michael Elberfeld, and Frederik Harwath. Expressivity and succinctness of order-invariant logics on depth-bounded structures. In *Mathematical Foundations of Computer Science (MFCS) - 39th International Symposium*, 2014.

[3] Michael Elberfeld, Marlin Frickenschmidt, and Martin Grohe. Order invariance on decomposable structures. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*, 2016.

[4] Ronald Fagin, Larry J. Stockmeyer, and Moshe Y. Vardi. On monadic NP vs. monadic co-np. *Inf. Comput.*, 1995.

[5] Jakub Gajarský, Petr Hliněný, Daniel Lokshtanov, Jan Obdržálek, and M. S. Ramanujan. A new perspective on FO model checking of dense graph classes. *CoRR*, 2018.

[6] Julien Grange and Luc Segoufin. Order-Invariant First-Order Logic over Hollow Trees. In *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*. Schloss Dagstuhl, 2020.

[7] Martin Grohe and Thomas Schwentick. Locality of order-invariant first-order formulas. *ACM Trans. Comput. Log.*, 2000.

[8] Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[9] Benjamin Rossman. Successor-invariant first-order logic on finite structures. *J. Symb. Log.*, 2007.

[10] Boris A Trakhtenbrot. Impossibility of an algorithm for the decision problem in finite classes. *Doklady Akademii Nauk SSSR*, 1950.