Decidability of Graph Neural Networks via Logical Characterizations

Michael Benedikt, Chia-Hsuan Lu, Boris Motik, and Tony Tan

Abstract

We present results concerning the expressiveness and decidability of a popular graph learning formalism, graph neural networks (GNNs), exploiting connections with logic. We use a family of recently-discovered decidable logics involving "Presburger quantifiers". We show how to use these logics to measure the expressiveness of classes of GNNs, in some cases getting exact correspondences between the expressiveness of logics and GNNs. We also employ the logics, and the techniques used to analyze them, to obtain decision procedures for verification problems over GNNs. We complement this with undecidability results for static analysis problems involving the logics, as well as for GNN verification problems.

2012 ACM Subject Classification Logic and Verification

Keywords and phrases Logic, Graph Neural Networks

Digital Object Identifier 10.4230/LIPIcs.Arxiv.2024.

Funding *Michael Benedikt, Chia-Hsuan Lu, Boris Motik, and Tony Tan*: (Optional) author-specific funding acknowledgements

1 Introduction

Graph Neural Networks (GNNs) have become the most common model for learning functions that work on graph data. Like traditional neural networks, GNNs consist of a layered architecture where layer $k+1$ takes as input the output of layer k. Each layer computes a function from graph vertices to a vector of numerical values – the *feature vector*. Computation of the feature vector at layer $k+1$ for a node u is based on aggregating vectors for layer k of nodes v that are related to u in the source graph: for example aggregating vectors associated to nodes adjacent to u in the graph. In an aggregation, the vectors of the previous vectors may be transformed using linear functions. A layer can perform multiple aggregations – corresponding to different linear functions – and then combine them to get the feature vector for the next layer. The use of graph structure ensures that the computation of the network is *invariant*: depending only on the input graph and the node up to isomorphism. There are many variations of GNN. One key design choice is the kind of aggregation used - one can use "local aggregation", over the neighbors of a node, or aggregation over all nodes in the graph. A second design choice is the kind of numerical functions that can be applied to vector components, in particular the kind of *activation functions* that can be applied at each layer: e.g. ReLU, sigmoid, piecewise linear functions.

An important issue in the study of graph learning is the *expressiveness* of a learning model. What kinds of computations can a given type of GNN express? The first results in this line were about the *separating power of a graph learning model*: what pairs of nodes can be distinguished using GNNs within a certain class. For example, it is known that the separating power of standard GNN models is limited by the Weisfeiler-Leman (WL) test [19].

A finer-grained classification would characterize the functions computed by GNNs within a certain class, in terms of some formalism that is easier to analyze. Such characterizations are referred to as *uniform expressiveness results* and there has been much less work in this area. [1] provides a classification of a class of GNNs in terms of *modal logic*. The main result in [1] is a characterization of the classifiers expressible in first-order logic that can be performed with a GNN having only *local aggregation* and *truncated* ReLU *activations* over *undirected graphs*. They also provide a lower bound on the expressiveness of GNNs having in addition a "global aggregator", that sums over all nodes in the graph.

In this work we continue the line of work on uniform expressiveness. Our work improves on the state of the art in a number of directions:

- *From first order expressiveness to general expressiveness* In contrast to [1], we provide logical characterizations of *all* the functions that can be computed by certain GNN formalisms, not just the intersection with first-order logic. To do this we utilize logics that go beyond first order, but which are still amenable to analysis.
- *From expressiveness to verification* While we deal with GNNs that go beyond first-order logic, we can still obtain characterizations in a logic where the basic satisfiability problems are decidable. This provides us with decidability of a number of natural verification problems related to GNNs. In doing this, we show a surprising link between GNNs and recently-devised decidable logics going beyond first-order logic, so-called *Presburger logics*.
- *From undirected graphs to directed graphs* While prior work focused on undirected graphs, we explore how the expressiveness characterizations vary with GNNs that can recognize directionality of graph edges. The aim is to show that the logical characterizations and decidability are often independent of the restriction to undirected graphs.
- *From bounded to unbounded activations* We explore the impact of the activation functions. We begin with the case of *bounded activation functions*, like the truncated ReLU of [1], and establish characterizations and decidability results for GNNs using this function. We show both some contrasts and some similarity to the case of *unbounded activation functions*, including the standard ReLU. Here some, but not all, of the corresponding decidability results fail.

Related work. Logics have been used to characterized the separating power of GNN languages ("non-uniform expressiveness") for a number of years: see [9] for an overview. The recent [10] provides logical characterizations of GNNs with piecewise linear activations. The logic is not decidable; indeed our undecidability results imply that one cannot capture such GNNs with a decidable logic.

We employ logical characterizations to gain insight on two basic verification problems – whether a given classification can be achieved on some nodes or on all nodes. There is prior work on verification of GNNs, but it focuses on more complex (but arguably more realistic) problems, adversarial robustness. The closest paper to ours is the recent [17], which formalizes a broad set of problems related to verifying that the output is in a certain region in Euclidean space. [17] provides both decidability and undecidability theorems, but they are incomparable to ours both in the results and in the techniques. For example Theorem 1 of [17] shows undecidability of a satisfiability problem where we verify that certain nodes output a particular value, over GNNs which always distinguish a node from its neighbor. Theorem 2 of [17] shows a decidability result with a different kind of specification, where the degree of input graphs is bounded.

Recently, logics that combine uninterpreted relations with Presburger arithmetic have been applied to the analysis of transformers – transducers that process strings [4, 2]. Since this is outside of the context of general graphs, the details of the logics that are employed are a bit different than those we consider, and the focus is not on the decidability border.

Organization. We formalize our GNN model and the basic logics we study in Section 2. We present results on logical characterizations of GNNs with "bounded activation functions" – like the truncated ReLU of [1]. We apply these characterizations to get decidability results. Section 4 turns to the case of unbounded activation functions, which includes the traditional

ReLU function. Here we provide lower bounds for expressiveness, and then turn to the implications for decidability. Section 5 gives conclusions and discusses several open issues. For expository purposes, *many details of the constructions are deferred to the appendix.*

2 Preliminaries

Let $\mathbb{N}$, $\mathbb{N}^+$, $\mathbb{Z}$, and $\mathbb{Q}$ be the set of natural numbers, positive natural numbers, integers, and rational numbers, respectively. For $p, q \in \mathbb{Z}$ and $p \leq q$, $[p\,, q]$ is the set of integers between p and q, including p and q. For $r \in \mathbb{Q}$, $\lceil r \rceil$ is the smallest integer greater than or equal to r.

For a function f mapping from $\mathbb{Q}$ to $\mathbb{Q}$ and a vector $b \in \mathbb{Q}^m$, $f(b)$ denotes that f is applied to each entry of b.

▶ **Definition 1.** *An n-*graph *is a tuple $\left\langle V, E, \{U_c\}_{1\leq c\leq n}\right\rangle$, where $n \in \mathbb{N}$ is the number of vertex colors; V is a nonempty finite set of vertices; $E \subseteq V \times V$ is a set of edges; each $U_c \subseteq V$ is the set of c-colored vertices.*

Note that we allow self-loops in graphs, and a graph is by default a *directed graph*. For a graph $\mathcal{G}$, we say that $\mathcal{G}$ is a *undirected graph* if for all $v, u \in V$, $(v, u) \in E$ if and only if $(u, v) \in E$. For a vertex $v \in V$, we let $\mathcal{N}_{\text{out},\mathcal{G}}(v) := \{u|\ (v, u) \in E\}$ and refer to this as the set of *out-neighbors of* v. The set of *in-neighbors of* v, denoted $\mathcal{N}_{\text{in},\mathcal{G}}(v)$ are defined analogously.
Graph Neural Networks. We use a standard notion of "aggregate-combine" graph neural networks with rational coefficients. The only distinction from the usual presentation is that we allow GNNs to work over directed graphs, with separate aggregations over incoming and outgoing edges, while traditional GNNs work on undirected graphs.

▶ **Definition 2.** *An n-*graph neural network (GNN) *is a tuple*

$$\left\langle \{d_\ell\}_{0\leq\ell\leq L}, \{f^\ell\}_{1\leq\ell\leq L}, \{C^\ell\}_{1\leq\ell\leq L}, \{A^\ell_x\}_{\substack{1\leq\ell\leq L\\ x\in\{out,in\}}}, \{R^\ell\}_{1\leq\ell\leq L}, \{b^\ell\}_{1\leq\ell\leq L} \right\rangle,$$

where $L \in \mathbb{N}^+$ is the number of layers; each $d_\ell \in \mathbb{N}^+$, called the dimension *of the ℓ^{th} layer, requiring $d_0 := n$, the number of colors; each $f^\ell : \mathbb{Q} \to \mathbb{Q}$, the* activation function *of the ℓ^{th} layer; each $C^\ell, A^\ell_x, R^\ell \in \mathbb{Q}^{d_\ell \times d_{\ell-1}}$, the* coefficient matrices *of the ℓ^{th} layer; and $b^\ell \in \mathbb{Q}^{d_\ell}$, the* bias vector *of the ℓ^{th} layer.*

All the coefficients are rational. In order to have an effective representation of a GNN, we will also *assume that the activation functions are computable.*

▶ **Definition 3.** *For an n-GNN $\mathcal{A}$ and an n-graph $\mathcal{G}$, the* computation of $\mathcal{A}$ on $\mathcal{G}$ *is a sequence of* derived feature functions $\{\xi^\ell_{\mathcal{G}} : V \to \mathbb{Q}^{d_\ell}\}_{0\leq\ell\leq L}$ *defined inductively: for $\ell = 0$, if $v \in U_c$, then the c^{th} entry of $\xi^0_{\mathcal{G}}(v)$ is 1; otherwise, the entry is 0. For $1 \leq \ell \leq L$,*

$$\xi^\ell_{\mathcal{G}}(v) \;:=\; f^\ell\left(C^\ell\xi^{\ell-1}_{\mathcal{G}}(v) + \sum_{x\in\{out,in\}}\left(A^\ell_x \sum_{u\in\mathcal{N}_{x,\mathcal{G}}(v)} \xi^{\ell-1}_{\mathcal{G}}(u)\right) + R^\ell \sum_{u\in V} \xi^{\ell-1}_{\mathcal{G}}(u) + b^\ell\right).$$

For $v \in V$, $\xi^\ell_{\mathcal{G}}(v)$ is called the ℓ-feature vector of v, *and $\xi^\ell_{\mathcal{G},i}(v)$ is the i^{th} entry of $\xi^\ell_{\mathcal{G}}(v)$.*

That is, we compute the feature values of a node v at layer $\ell + 1$ by adding several components. One component aggregates over the ℓ-layer feature vector from the outgoing neighbors of v, and applies a linear transformation. Another component does the same for the incoming neighbors of v, a third does this for every node in the graph, while another applies a transformation to the ℓ-layer feature vector of v itself. The linear transformation can be different for each component, and in particular can be a zero matrix that just drops that component. The final component of the sum is the bias vector.

When the graph $\mathcal{G}$ is clear from the context, we omit it and simply write $\xi^\ell(v)$ and $\xi_i^\ell(v)$, and similarly when the graph $\mathcal{G}$ is clear from the context, write $\mathcal{N}_{\text{out}}(v)$ and $\mathcal{N}_{\text{in}}(v)$ for the in-neighbors and out-neighbors.

Note that in most presentations of GNNs, one deals with only undirected edges. The above definition degenerates in that setting to two aggregations per layer, with the aggregation over all nodes often referred to in the literature as the *global readout*.

In some presentations of GNNs, a *classification function*, which associates a final Boolean decision to a node, is included in the definition. In our case, we have separated out the classification function as an independent component in defining the expressiveness: see the last part of the preliminaries.

Classes of activation functions. Following prior work on analysis of GNNs, some of our results will deal with activation functions that are bounded in value:

▶ **Definition 4.** *We say that the function $f : \mathbb{Q} \to \mathbb{Q}$ is* eventually constant*, if there exists $t_{\mathsf{left}}, t_{\mathsf{right}} \in \mathbb{Q}$ satisfying $t_{\mathsf{left}} < t_{\mathsf{right}}$, called the* left and right thresholds *of f, such that for every $x \leq t_{\mathsf{left}}$, $f(x) = f(t_{\mathsf{left}})$; for every $x \geq t_{\mathsf{right}}$, $f(x) = f(t_{\mathsf{right}})$.*

A standard eventually constant function is the *truncated* ReLU *function*, which is 0 for negative reals, 1 for x greater than 1, and x otherwise [1]. There are other eventually constant functions that are used in practice: for example, the linear approximation of standard bounded functions used in graph learning, like the Sigmoid activation function. We will be interested in functions that are defined on the reals, but which preserve the rationals. The definition of eventually constant extends to such a function in the obvious way.

For a GNN with eventually constant activation functions, we use $\left\{t^\ell_{\mathsf{left}}\right\}_{1\leq\ell\leq L}$ and $\left\{t^\ell_{\mathsf{right}}\right\}_{1\leq\ell\leq L}$ to denote the left and right thresholds of the GNN's activation functions.

We also consider *unbounded activation functions*, such as the *standard* ReLU *function*, which is x for non-negative reals and 0 for negative reals.

Flavors of GNN. For a GNN $\mathcal{A}$, we say that $\mathcal{A}$ is *outgoing-only*, denoted by $\mathcal{O}$, if for every $1 \leq \ell \leq L$, A^ℓ_{in} is a zero matrix. $\mathcal{A}$ is *bidirectional*, denoted by $\mathcal{B}$, if there is no restriction on A^ℓ_{in}. $\mathcal{A}$ is *local*, denoted by $\mathcal{L}$, if for every $1 \leq \ell \leq L$, R^ℓ is a zero matrix. In the usual GNN terminology, this would mean that *there is no global readout*. $\mathcal{A}$ is *global*, denoted by $\mathcal{G}$, if global readout is allowed. $\mathcal{A}$ is *eventually constant*, denoted by $\mathcal{C}$, if for $1 \leq \ell \leq L$, f^ℓ is an eventually constant function. Our results outside of eventually constant will deal with either *piecewise linear* activations, denoted $\mathcal{PW}$, truncated ReLU activations, denoted TrReLU, or standard ReLU activations. We use the following naming, $(\mathcal{O}|\mathcal{B})(\mathcal{L}|\mathcal{G})(\mathcal{C}|\mathcal{PW}|$TrReLU$|$ReLU$)$-GNN, for the set of GNNs satisfying constraints given by the prefix. For example, $\mathcal{OLC}$-GNN is the set of outgoing-only, local, and eventually constant GNNs; $\mathcal{BGPW}$-GNN is the set of GNNs allowing both incoming and outgoing, global readout, and piecewise linear activations.

Classifiers and Boolean semantics. Our GNNs define vector-valued classification functions on nodes. But for comparing with expressiveness and in defining verification problems, we will often use a derived function from nodes to Booleans. We do this by thresholding at the end – below we use .5 for convenience, but other choices do not impact the results.

▶ **Definition 5.** *For a L-layer n-GNN $\mathcal{A}$, an n-graph $\mathcal{G}$, and a vertex $v \in V$, we say that $\mathcal{A}$* accepts *the tuple $\langle \mathcal{G}, v\rangle$, if $\xi^L_{\mathcal{G},1}(v) \geq 0.5$.*

Note that the global readout component can interact with the activation functions f^ℓ, which can behave very differently on translated values due to non-linearity – think of a typical f^ℓ as a piece-wise linear function. Global readout can also interact with the classification threshold, pushing some values above the threshold while leaving others below.

Two-variable Modal Logic with Presburger Quantifiers. We review logic with Presburger quantifiers. The basic idea is to combine a decidable logic on uninterpreted structures, like two-variable logic or guarded logic, with the ability to perform some arithmetic on the number of elements. There are several formalisms in the literature that combine Presburger arithmetic with a decidable uninterpreted logic, some originating many years ago [12]. We will rely on a recent logic from [3], but we will need several variations of the underlying idea here.

▶ **Definition 6.** *A* Presburger quantifier *is of the form:*

$$\mathcal{P}(x) \;:=\; \sum_{i=1}^{k} \lambda_i \cdot \#_y[\varphi_i(x,y)] \;\circledast\; \delta,$$

where $\delta \in \mathbb{Z}$; each $\lambda_i \in \mathbb{Z}$; each $\varphi_i(x,y)$ is a formula with free variables x and y; $\circledast$ is one of $=, \neq, \leq, \geq, <,$ or $>$. Note that $\mathcal{P}(x)$ has one free variable x.

We give the semantics of these quantifiers inductively, assuming a semantics for $\varphi_i(x,y)$. Given a graph $\mathcal{G}$ and a vertex $v \in V$, we say that $\mathcal{P}(x)$ holds in $\mathcal{G}, x/v$, denoted by $\mathcal{G} \models \mathcal{P}(v)$, if the following (in)equality holds in $\mathbb{Z}$.

$$\sum_{i=1}^{k} \lambda_i \cdot |\{u \in V |\ \mathcal{G} \models \varphi_i(v,u)\}| \;\circledast\; \delta$$

▶ Remark 7. Note that each Presburger quantifier can be rewritten as a Boolean combination of expressions which *only use the inequality symbol $\geq$ as $\circledast$*. For example, $(\#_y[\varphi(x,y)] = \delta)$ and $(\#_y[\varphi(x,y)] \geq \delta) \wedge \neg(\#_y[\varphi(x,y)] \geq \delta + 1)$ are semantically equivalent. Therefore it is sufficient to consider Presburger quantifiers which only use the inequality symbol $\geq$.

▶ Remark 8. We will make use of Presburger quantifiers that allow for rational coefficients of the form:

$$\widetilde{\mathcal{P}}(x) \;:=\; \kappa_0 + \sum_{i=1}^{k} \kappa_i \cdot \#_y[\varphi_i(x,y)] \;\circledast\; \lambda_0 + \sum_{i=1}^{\ell} \lambda_i \cdot \#_y[\psi_i(x,y)],$$

where each $\kappa_i, \lambda_i \in \mathbb{Q}$. This is a shorthand for the Presburger quantifier:

$$\mathcal{P}(x) \;:=\; \sum_{i=1}^{k} (D\kappa_i) \cdot \#_y[\varphi_i(x,y)] + \sum_{i=1}^{\ell} (-D\lambda_i) \cdot \#_y[\psi_i(x,y)] \;\circledast\; D(\lambda_0 - \kappa_0),$$

where D is the least common multiplier of the denominators of the coefficients in $\widetilde{\mathcal{P}}(x)$.

▶ **Definition 9.** *We give the syntax of* two-variable modal logic with Presburger quantifiers (MP^2) *over vocabulary τ. Formulas will have exactly one free variable, denoted x below:*

- *$\top$ is an MP^2 formula.*
- *for a unary predicate $U \in \tau$, $U(x)$ is an MP^2 formula.*
- *if $\varphi(x)$ is an MP^2 formula, then so is $\neg\varphi(x)$.*
- *if $\varphi_1(x)$ and $\varphi_2(x)$ are MP^2 formulas, then so is $\varphi_1(x) \wedge \varphi_2(x)$.*

- *if* $\{\varphi_i(x)\}_{1\leq i\leq k}$ *is a set of* MP^2 *formulas and* $\{\epsilon_i(x,y)\}_{1\leq i\leq k}$ *is a set of* guard atoms, *of form* $E(x,y)$, $E(y,x)$, *or* $\top$, *then* $\left(\sum_{i=1}^{k} \lambda_i \cdot \#_y[\epsilon_i(x,y) \wedge \varphi_i(y)] \circledast \delta\right)$ *is also an* MP^2 *formula.* $\{\epsilon_i(x,y)\}_{1\leq i\leq k}$ *are the* guards *of the formula. Consistent with the restriction we announced on the logic, we consider the result as a formula with free variable* x*: if all* ϵ_i *are* $\top$ *it returns either every node or no node.*

The semantics of the Boolean connectives is as usual, while the semantics of the Presburger quantifiers is given by Definition 6.

An MP^2 formula $\varphi(x)$ is an n-formula if its vocabulary consists of n unary predicates. We use abbreviations $\vee$ and $\rightarrow$ as usual. Note that the guarded universal quantifier $\forall y\ E(x,y) \rightarrow \varphi(y)$ can be expressed as $(1 \cdot \#_y[E(x,y) \wedge \neg\varphi(y)] = 0)$, and the guarded existential quantifier $\exists y\ E(x,y) \wedge \varphi(y)$ can be expressed as $(1 \cdot \#_y[E(x,y) \wedge \varphi(y)] \geq 1)$.

The logic MP^2 combines Presburger arithmetic and quantification over the model. Thus one might worry that it has an undecidable satisfiability problem. And indeed, we will show this: see Theorem 29. An idea to gain decidability is to impose that the quantification is *guarded* – again, the underlying idea is from [3]. The logic $\mathcal{L}$-MP^2 (or "*local* MP^2") is obtained by excluding $\top$ as a guard. Analogously to what we did with GNNs, we use $\mathcal{L}$ to indicate that quantification is "local".

The logic $\mathcal{L}$-MP^2 is contained in the following logic, defined in [3]:

▶ **Definition 10.** *The syntax of the* guarded fragment of two-variable logic with Presburger quantifiers (GP^2) *over colored graph vocabulary* τ *starts with arbitrary atoms over the vocabulary, with the usual connective closure and the following rules for quantifiers:*

- *if* $\varphi(x)$ *is a* GP^2 *formula, then so are* $\forall x\ \epsilon(x) \rightarrow \varphi(x)$ *and* $\exists x\ \epsilon(x) \wedge \varphi(x)$*, where* ϵ *is either* $U(x)$ *or* $x = x$ *for some unary predicate* $U \in \tau$*.*
- *if* $\varphi(x,y)$ *is a* GP^2 *formula, then so are* $\forall x\ \epsilon(x,y) \rightarrow \varphi(x,y)$ *and* $\exists x\ \epsilon(x,y) \wedge \varphi(x,y)$*, where* $\epsilon(x,y)$ *is one of* $E(x,y)$ *or* $E(y,x)$*.*
- *if* $\{\varphi_i(x,y)\}_{1\leq i\leq k}$ *is a set of* GP^2 *formulas and* $\{\epsilon_i(x,y)\}_{1\leq i\leq k}$ *is a set of formulas, each of form* $E(x,y)$ *or* $E(y,x)$*, then* $\left(\sum_{i=1}^{k} \lambda_i \cdot \#_y[\epsilon_i(x,y) \wedge \varphi_i(x,y)] \circledast \delta\right)$ *is also a* GP^2 *formula.*

The main difference between the logic $\mathcal{L}$-MP^2 and the logic above is that the former is "modal", restricting to one-variable formulas, and allowing two variables only in the guards. While in the logic above we can build up more interesting two variable formulas, for example conjoining two guards.

We will make use of the following prior decidability result:

▶ **Theorem 11** ([3], Theorem 10)**.** *The finite satisfiability problem of* GP^2 *is decidable.*

From this we easily derive the decidability of $\mathcal{L}$-MP^2:

▶ **Corollary 12.** *The finite satisfiability problem of* $\mathcal{L}$-MP^2 *is decidable.*

Notions of expressiveness for GNNs and MP^2 Formulas. Recalling that we have a node-to-Boolean semantics available for both logical formulas and GNNs (via thresholding), we use the term *n-specification* for either a n-GNN or a n-MP^2 formula.

▶ **Definition 13.** *If* S_1, S_2 *are* n*-GNNs, they are said to be* equivalent *if they accept the same nodes within* n*-graphs. If* S_1 *is a GNN and* S_2 *a node formula in some logic, we say* S_1 *and* S_2 *are equivalent if for every* n*-graph* $\mathcal{G}$ *and vertex* $v \in V$*,* S_1 *accepts* $\langle \mathcal{G}, v\rangle$ *if and only if* $\mathcal{G}, v$ *satisfies* S_2*.*

The notions of two languages of specifications being equally expressive, or equally expressive over undirected graphs, is defined in the obvious way:

Verification Problems for GNNs. We focus on two verification problems. The first is the most obvious analog of satisfiability for GNNs, whether it accepts some node of some graph:

▶ **Definition 14.** *For an n-GNN $\mathcal{A}$, we say that $\mathcal{A}$ is* satisfiable*, if there exist an n-graph $\mathcal{G}$ and a vertex $v \in V$, such that $\mathcal{A}$ accepts $\langle \mathcal{G}, v \rangle$.*

We will also consider a variation of the problem which asks whether a GNN accepts every node of some graph:

▶ **Definition 15.** *For an n-GNN $\mathcal{A}$, we say that $\mathcal{A}$ is* universally satisfiable*, if there exist an n-graph $\mathcal{G}$, such that for every vertex $v \in V$, $\mathcal{A}$ accepts $\langle \mathcal{G}, v \rangle$.*

Two GNNs $\mathcal{A}$ and $\mathcal{B}$ are equivalent if they accept the same tuples. Note that, like satisfiability and unlike universal satisfiability, this does not require a quantifier alternation. For brevity we will not state results for equivalence, but *it can easily be seen that both our positive and negative results on satisfiability also apply to equivalence.*

3 Characterization and decidability of GNNs with eventually constant activation functions

In this section, we only consider GNNs with eventually constant activations. In Section 3.1, we establish a key tool to analyzing these GNNs: we show that the set of possible activation values is finite, and one can compute an overapproximation of this set. We use this for two purposes. First we give a decidability result for GNNs with eventually constant activations and only local aggregation, and then we show that even with global aggregation we get an equivalence of the GNNs in expressiveness with a logic.

In Section 3.2, we show that the finite satisfiability of MP^2 is undecidable. Using the expressiveness characterization, this will imply that satisfiability problems for global GNNs are undecidable. These results were presented for GNNs and logics on directed graphs. In Section 3.3 we use the logical characterizations to show that they also apply to the standard setting for GNNs of undirected graphs.

3.1 Decidability of satisfiability problems for GNNs with eventually constant functions, via logic

We now come to one of the crucial definitions in the paper, the spectrum of a GNN.

▶ **Definition 16.** *For a $\mathcal{BGC}$-GNN $\mathcal{A}$ and $0 \leq \ell \leq L$, the ℓ-*spectrum *of $\mathcal{A}$, denoted by $\mathcal{S}^\ell$, is the set $\{\xi^\ell(v) |$ for every n-graph $\mathcal{G}$ and vertex $v \in V\}$.*

That is, the ℓ-spectrum is the range of the feature vectors computed at layer ℓ, as we range over all input graphs and nodes. We show that the spectrum is actually finite, and a finite superset is computable:

▶ **Theorem 17.** *For every $\mathcal{BGC}$-GNN $\mathcal{A}$ and $0 \leq \ell \leq L$, the ℓ-spectrum of $\mathcal{A}$ is finite. We can compute a finite superset of the ℓ-spectrum from the specification of $\mathcal{A}$.*

We give some intuition for the proof. Our effective overapproximation of the spectrum will simulate the computation of the GNN, and will be defined inductively on the layers. Recall that a $\mathcal{BGC}$-GNN is given by dimensions $\{d_\ell\}_{0 \leq \ell \leq L}$, activation functions $\{f^\ell\}_{1 \leq \ell \leq L}$, coefficient matrices for transforming the prior node value $\left\{C^\ell\right\}_{1\leq\ell\leq L}$, coefficient matrices for local aggregation $\left\{A_x^\ell\right\}_{\substack{1\leq\ell\leq L\\ x\in\{\text{out},\text{in}\}}}$, coefficient matrices for global readout $\left\{R^\ell\right\}_{1\leq\ell\leq L}$, and bias vectors $\left\{b^\ell\right\}_{1\leq\ell\leq L}$.

▶ **Definition 18.** *For a $\mathcal{BGC}$-GNN $\mathcal{A}$ and $0\leq\ell\leq L$, the set $\uparrow\mathcal{S}^\ell$ is defined as follows:*

$$\begin{aligned}\uparrow\mathcal{S}^0 &:= \{0,1\}^{d_0}\\ \uparrow\mathcal{S}^\ell_s &:= \left\{f^\ell\left(C^\ell s+\sum_{x\in\{out,in\}}A^\ell_x\sum_{s'\in\uparrow\mathcal{S}^{\ell-1}}s'n_x^{A,s'}+R^\ell\sum_{s'\in\uparrow\mathcal{S}^{\ell-1}}s'n^{R,s'}+b^\ell\right)\middle|\ n_x^{A,s'},n^{R,s'}\in\mathbb{N}\right\}\\ \uparrow\mathcal{S}^\ell &:= \bigcup_{s\in\uparrow\mathcal{S}^{\ell-1}}\uparrow\mathcal{S}^\ell_s\end{aligned}$$

We show that the set $\uparrow\mathcal{S}^\ell$ overapproximates the ℓ-spectrum:

▶ **Lemma 19.** *For every n-$\mathcal{BGC}$-GNN $\mathcal{A}$ and $0\leq\ell\leq L$, for every n-graph $\mathcal{G}$ and vertex $v\in V$, there exists $s\in\uparrow\mathcal{S}^\ell$, such that $\xi^\ell(v)=s$.*

It is quite straightforward to see that every element of the spectrum is captured. It is an overapproximation because different integers that we sum in an inductive step may not be realized in the same graph.

We can show by induction on the number of the layers that the set is finite – regardless of computability of the activation functions!

▶ **Lemma 20.** *For every n-$\mathcal{BGC}$-GNN $\mathcal{A}$ and $0\leq\ell\leq L$, $\uparrow\mathcal{S}^\ell$ has finite size and can be computed.*

In the inductive step, we have a finite set of rationals, thus some fixed precision. We take some integer linear combinations and we will obtain an infinite set of values, but only finitely many between the left and right thresholds of the eventually constant activations. Thus when we apply the activation functions to these values, we will get a finite set of rational values – since the activation functions map rationals to rationals.

▶ Remark 21. The restriction to rational coefficients is crucial in the argument. Consider the following 1-layer 1-$\mathcal{BL}$TrReLU-GNN. The dimensions are $d_0=d_1=1$; the coefficient matrix C^1 is a zero matrix; $\left(A^1_{\text{out}}\right)_{1,1}=\sqrt{2}$; $\left(A^1_{\text{in}}\right)_{1,1}=-1$; the bias vector b^1 is a zero vector. It is not difficult to see that its 1-spectrum is $\left\{\mathsf{TrReLU}\left(\sqrt{2}k_1-k_2\right)\middle|\ k_1,k_2\in\mathbb{N}\right\}$, whose size is infinite since $\sqrt{2}$ is irrational.

▶ Remark 22. Even simple GNNs may have exponential size spectra. For example, let $\mathcal{A}_k$ be a 1-layer 1-$\mathcal{BL}$TrReLU-GNN defined as follows: the dimensions are $d_0=d_1=1$; the coefficient matrices C^1 and A^1_{in} are zero matrices; $\left(A^1_{\text{out}}\right)_{1,1}=k^{-1}$; the bias vector b^1 is a zero vector. By definition, its 1-spectrum is $\left\{ik^{-1}\middle|\ i\in[0\,,k]\right\}$, whose size is $k+1$. But the description of $\mathcal{A}_k$ is only linear in $\log k$.

For GNNs with truncated ReLU and only local aggregation, there is a matching upper bound, as discussed after Theorem 25.

We now give several applications of the spectrum result. First we can use the finiteness of the spectrum to get a characterization of the expressiveness of $\mathcal{BGC}$-GNN and logic:

▶ **Theorem 23.** *For every n-$\mathcal{BGC}$-GNN $\mathcal{A}$, there exists an n-MP^2 formula $\Psi_\mathcal{A}(x)$, effectively computable from the description of $\mathcal{A}$, such that $\mathcal{A}$ and $\Psi_\mathcal{A}(x)$ are equivalent. In the case we start with an n-$\mathcal{BLC}$-GNN, the formula we obtain is in n-$\mathcal{L}$-MP^2.*

This expressiveness equivalence will be useful in getting further decidability results, as well as separations in expressiveness, for GNNs.

The idea of the proof of the theorem is that we have only finitely many elements in the overapproximation set to worry about, so we can fix each in turn and write a formula for each.

Recall that finite satisfiability of $\mathcal{L}$-MP^2 is decidable by Corollary 12. Combining this with Theorem 23 we get decidability of satisfiability for $\mathcal{BLC}$-GNN:

▶ **Theorem 24.** *The satisfiability problem for $\mathcal{BLC}$-GNNs is decidable.*

A more realistic analysis of complexity requires stronger assumptions on the activation functions. For now we note only one special case, where everything is a truncated ReLU:

▶ **Theorem 25.** *For $\mathcal{BLC}$-GNNs with truncated* ReLU *activations, the satisfiability problem is* PSPACE*-complete. It is* NP*-complete when the number of layers is fixed.*

We briefly discuss the PSPACE upper bound argument. We can show that for an arbitrary input graph, there are only exponentially many activation values, each representable with a polynomial number of bits. We also show, via an "unravelling construction", a common technique use in analysis of modal and guarded logics [13, 8], that a satisfying model can be taken to be a tree of polynomial depth and branching. These two facts immediately give an elementary bound, since we could guess the tree and the activation values. We can improve to PSPACE by exploring a satisfying tree-like model on-the-fly: again, this is in line with the PSPACE algorithm for modal logic [13].

The PSPACE lower bound is established by embedding the description logic $\mathcal{ALC}$ into $\mathcal{L}$-MP^2. PSPACE-hardness will follow from this, since concept satisfiability problem of $\mathcal{ALC}$ with one role is PSPACE-hard [18]. The NP upper bound will use the same on-the-fly algorithm as in the PSPACE case, just observing that for fixed depth it can be implemented in NP. A direct encoding of SAT gives the lower bound.

The following converse to Theorem 23 shows that the logic is equally expressive as the GNN model:

▶ **Theorem 26.** *For every n-MP^2 formula $\Psi(x)$, there exists an n-$\mathcal{BG}$TrReLU-GNN $\mathcal{A}_\Psi$, such that $\Psi(x)$ and $\mathcal{A}_\Psi$ are equivalent. If we start with an n-$\mathcal{L}$-MP^2 formula, we obtain an n-$\mathcal{BL}$TrReLU-GNN.*

The idea of the proof is induction on the formula structure. For each subformula there will be an entry of a feature vector for the GNN which represents the subformula, in the sense that – for the final layer – its value is 1 if the subformula holds, or 0 otherwise. We will have an entry for each subformula at every iteration, but as we progress to later layers of the GNN, more of these entries will be correct with respect to the corresponding subformula. In an inductive case for a Presburger quantifier that uses some coefficients λ_i, the corresponding matrix will be multiplying certain quantifies by λ_i. Note that this translation is polynomial time, thus the size of the corresponding GNN is polynomial in the formula.

Putting together the two translation results, we have:

▶ **Corollary 27.** *The logic MP^2 and $\mathcal{BGC}$-GNNs are expressively equivalent, as are $\mathcal{L}$-MP^2 and $\mathcal{BLC}$-GNNs.*

The translations also tell us that *the expressiveness of GNNs with truncated* ReLU *is the same as that of GNNs with arbitrary eventually constant activations* – provided we use the Boolean semantics based on thresholds.

Recall from Corollary 12 that finite satisfiability for the richer logic GP^2, allowing unguarded unary quantification and containing $\mathcal{L}$-MP^2, is decidable. Using this and the expressiveness characterization gives decidability of universal satisfiability for these GNNs:

▶ **Theorem 28.** *The universal satisfiability problem of $\mathcal{BLC}$-GNNs is decidable.*

3.2 Undecidability of MP^2, and of GNNs with truncated ReLU and global readout

Note that we claimed that the spectrum is finite for GNNs with eventually constant activations, even when they have global readout. And we could compute a finite overapproximation of the spectrum. But in our decidability argument for $\mathcal{BLC}$-GNN, we required further the ability to decide membership in the spectrum for any fixed rational, and for this we utilized decidability of the logic. So what happens to decidability of the GNNs – or the corresponding logic – when global readout is allowed?

We show undecidability of finite satisfiability for the logic MP^2, and of the corresponding GNN satisfiability problem. First for the logic:

▶ **Theorem 29.** *The finite satisfiability problem of MP^2 is undecidable.*

For the proof we apply an approach based on ideas in [3], using a reduction from Hilbert's tenth problem.

▶ **Definition 30.** *A* simple equation system *ε (with n variables and m equations) is a set of m equations of one of the forms $v_{i_1} = 1$, $v_{i_1} = v_{i_2} + v_{i_3}$, or $v_{i_1} = v_{i_2} \cdot v_{i_3}$, where $1 \leq i_1, i_2, i_3 \leq n$. We say the system ε is solvable if it has a solution in $\mathbb{N}$.*

▶ **Lemma 31.** *For every simple equation system ε with n variables and m equations, there exists an $(n+m)$- MP^2 formula $\Psi_\varepsilon(x)$ such that ε has a solution in $\mathbb{N}$ if and only if $\Psi_\varepsilon(x)$ is finitely satisfiable.*

Since the solvability (over $\mathbb{N}$) of simple equation systems is undecidable, Theorem 29 follows. From the theorem and Corollary 27 we obtain undecidability of static analysis for GNNs with global readout:

▶ **Theorem 32.** *The satisfiability problem of $\mathcal{BG}$TrReLU-GNNs is undecidable.*

Using a similar reduction, we obtain undecidability for universal satisfiability:

▶ **Theorem 33.** *The universal satisfiability problem of $\mathcal{BG}$TrReLU-GNNs is undecidable.*

3.3 Variations for the undirected case

Thus far we have been dealing with both logics and GNNs that work over directed graphs. We now show that all of the prior results apply to undirected graphs, the standard setting for GNNs.

We can enforce undirectedness within the larger decidable logic GP^2 to obtain decidability:

▶ **Corollary 34.** *The finite satisfiability problem of $\mathcal{L}$-MP^2 over undirected graphs is decidable.*

By reducing to decidability in the logic $\mathcal{L}$-MP^2, we can show that the satisfiability problem for GNNs on undirected graphs – that is, the standard notion of GNN – is decidable.

▶ **Theorem 35.** *The satisfiability problem of $\mathcal{BLC}$-GNNs over undirected graphs is decidable.*

▶ **Theorem 36.** *The universal satisfiability problem of* $\mathcal{BLC}$-GNN*s over undirected graphs is decidable.*

We can also revise our undecidability results for global GNNs to the undirected case, thus giving undecidability for the usual notion of GNN with global readout. This is done with the same reduction from solvability of simple equation systems to the finite satisfiability of MP^2, which we can show works over undirected graphs.

▶ **Theorem 37.** *The finite satisfiability problem of* MP^2 *over undirected graphs is undecidable.*

▶ **Theorem 38.** *The satisfiability problem of* $\mathcal{BG}$TrReLU-GNN*s over undirected graphs is undecidable.*

▶ **Theorem 39.** *The universal satisfiability problem of* $\mathcal{BG}$TrReLU-GNN*s over undirected graphs is undecidable.*

4 GNNs with unbounded activation functions

In this section, we consider GNNs with unbounded activations, such as the standard ReLU. Since we have already shown that global aggregation leads to undecidability even in the bounded case, in this section *we will only deal with GNNs having only local aggregation.* In Section 4.1 we show that the universal satisfiability problem of $\mathcal{BL}$ReLU-GNN is undecidable, a contrast to the case with eventually constant activation functions. In the process, we introduce a logic that also helps with understanding expressiveness of this class of GNNs.

In Section 4.2, we turn to the satisfiability problem, and give a partial positive result about decidability. Here we will not use the logic directly, but rather use components from decidability proofs for Presburger logics [3]. We will use the idea of representing the possible values of activations which was also used in the case of decidability for eventually constant activations. But in this case we will be representing an infinite set of values, using Presburger formulas.

4.1 (Un)decidability of GNNs with unbounded activation functions

We prove the undecidability of the universal satisfiability problem of $\mathcal{BL}$ReLU-GNN. Here we will use logic again. We will not obtain an expressiveness characterization, but merely a logic that embeds in $\mathcal{BL}$ReLU-GNNs: local two-variable modal logic with two-hop Presburger quantifiers ($\mathcal{L}$-$\mathsf{M2P}^2$), which is the extension of MP^2 where the guards are conjunctions of at most two binary predicates.

▶ **Definition 40.** *The syntax of* local two-variable modal logic with two-hop Presburger quantifiers ($\mathcal{L}$-$\mathsf{M2P}^2$) *over vocabulary* τ *is defined inductively:*

- $\top$ *is a* $\mathcal{L}$-$\mathsf{M2P}^2$ *formula.*
- *for a unary predicate* $U \in \tau$, $U(x)$ *is a* $\mathcal{L}$-$\mathsf{M2P}^2$ *formula.*
- *if* $\varphi(x)$ *is a* $\mathcal{L}$-$\mathsf{M2P}^2$ *formula, then so is* $\neg\varphi(x)$.
- *if* $\varphi_1(x)$ *and* $\varphi_2(x)$ *are* $\mathcal{L}$-$\mathsf{M2P}^2$ *formulas, then so is* $\varphi_1(x) \wedge \varphi_2(x)$.
- *if* $\{\varphi_i(x)\}_{1\le i\le k} \cup \{\varphi'_i(x)\}_{1\le i\le k'}$ *is a set of* $\mathcal{L}$-$\mathsf{M2P}^2$ *formulas,* $\{\epsilon_i(x,z,y)\}_{1\le i\le k}$ *is a set of guard formulas, each of form* $E(x,z)\wedge E(z,y)$, $E(x,z)\wedge E(y,z)$, $E(z,x)\wedge E(z,y)$, *or* $E(z,x)\wedge E(y,z)$, *and* $\{\epsilon'_i(x,y)\}_{1\le i\le k'}$ *is another set of guard formulas, each of form* $E(x,y)$ *or* $E(y,x)$, *then*
  $\left(\sum_{i=1}^{k} \lambda_i \cdot \#_{z,y}[\epsilon_i(x,z,y)\wedge\varphi_i(y)] + \sum_{i=1}^{k'} \lambda'_i \cdot \#_y[\epsilon'_i(x,y)\wedge\varphi'_i(y)] \ \circledast \ \delta\right)$
  is also a $\mathcal{L}$-$\mathsf{M2P}^2$ *formula. The numbers* δ, λ_i, λ'_i, *and the comparison* $\circledast$ *are as in the standard Presburger quantifier definition.*

The idea is that we can still count a linear combination of cardinalities of the number of nodes satisfying a given lower-level formula that are one-hop away from the current node – as in $\mathcal{L}$-MP^2. Optionally, we can add on a linear combination of the number of two-hop paths that lead to a node satisfying other lower-level formulas.

The semantics of the formulas is given inductively, with the only step that is different from the usual cases being for the quantification, which is the obvious one. We call these *two-hop Presburger quantifiers.* We can apply a similar proof technique as in Theorem 26 to show that the $\mathcal{L}$-M2P^2 are expressible using $\mathcal{BL}$ReLU-GNNs.

▶ **Theorem 41.** *For every n-$\mathcal{L}$-M2P^2 formula $\Psi(x)$, there exists an n-$\mathcal{BL}$ReLU-GNN $\mathcal{A}_\Psi$, such that $\Psi(x)$ and $\mathcal{A}_\Psi$ are equivalent.*

Note that we do not claim an expressive equivalence here. Nevertheless this containment of the logic in the GNN class is useful, since we can show undecidability of the logic by reduction from the halting problem of two-counter machines, which is known to be undecidable [15].

▶ **Definition 42.** *A* two-counter machine *$\mathcal{M}$ is a finite list $d_1 \dots d_n$ of instructions having one of the forms* `INC` (c_i), `IF` $(c_i = 0)$ `GOTO` (j), *or* `HALT`, *where $i \in \{0, 1\}$ and $1 \le j \le n$.*

A configuration *is a tuple $\langle q, c_0, c_1\rangle$, where $1 \le q \le n$ and $c_0, c_1 \in \mathbb{N}$. We say $\langle q', c'_0, c'_1\rangle$ is the* successor configuration *of $\langle q, c_0, c_1\rangle$ if, letting d_q be the q^{th} instruction of the machine:*

- *If d_q is* `INC` (c_i), *then $q' = q + 1$, $c'_i = c_i + 1$, and $c'_{1-i} = c_{1-i}$.*
- *If d_q is* `IF` $(c_i = 0)$ `GOTO` (j), *if $c_i = 0$, then $q' = j$, $c'_0 = c_0$, and $c'_1 = c_1$; otherwise, $q' = q + 1$, $c'_i = c_i - 1$, and $c'_{1-i} = c_{1-i}$.*

Note that if d_q is `HALT`, *there is no successor. This configuration is called a halting configuration.*

The computation *of the machine is a (possibly infinite) sequence of configurations where the first is $\langle 1, 0, 0\rangle$, consecutive pairs are in the succcessor relationship above, the last configuration is a halting configuration. The machine* halts *if its computation is a finite sequence.*

The reduction is by encoding the computation of a two-counter machine into the graph directly. We have illustrated it in Figure 2. Each configuration is encoded as a height 1 tree, which is denoted by a dashed box. Its line number is represented by the unary predicate Q_i realized by the root vertex, and the values of the counters are represented by the number of "labeled leaves" – those with predicate C_0 or C_1 being true. There are edges connected to the roots of each configuration, which encode the computation sequence. Then it is possible to assert the (in)equality between the number of leaves of some root and the root of the successor tree, which encodes the condition of a valid transition.

▶ **Lemma 43.** *For every two-counter machine $\mathcal{M}$ with n instructions, there exists an $(n + 5)$-$\mathcal{L}$-M2P^2 formula $\Psi_\mathcal{M}(x)$ such that $\mathcal{M}$ halts if and only if $\forall x\ \Psi_\mathcal{M}(x)$ is finitely satisfiable.*

Since the halting problem of two-counter machines is undecidable, and $\mathcal{L}$-M2P^2 formulas can be translted to $\mathcal{BL}$ReLU-GNNs, we obtain the undecidability of the universal satisfibility problem of $\mathcal{BL}$ReLU-GNN, by reduction from $\mathcal{L}$-M2P^2.

▶ **Theorem 44.** *The universal satisfiability problem of $\mathcal{BL}$ReLU-GNNs is undecidable.*

We will later show that this holds also for undirected graphs: see Theorem 50 below.

We can also use the logic to get an expressiveness separation for GNNs: By Theorem 41 to show that $\mathcal{BL}$ReLU-GNNs can do more than $\mathcal{BLC}$-GNNs, it is sufficient to show that there is a $\mathcal{L}$-M2P^2 formula that is not given by a $\mathcal{BLC}$-GNN:

▶ **Lemma 45.** *$\mathcal{L}$-M2P^2 is strictly more expressive than $\mathcal{BLC}$-GNN.*

The following results are direct consequences of the lemma above and the logical characterization in the prior section:

▶ **Corollary 46.** *$\mathcal{L}$-M2P^2 is strictly more expressive than $\mathcal{L}$-MP^2.*

▶ **Corollary 47.** *$\mathcal{BL}$ReLU-GNN is strictly more expressive than $\mathcal{BLC}$-GNN.*

We comment on the proof of Lemma 45. We claim that the property "the number of two-hop paths from the vertex v to the green vertices is the same as the number of two-hop paths from v to the blue vertices" gives the separation. It is easy to express in the two-hop logic. To show that no $\mathcal{BLC}$-GNN can express it, we construct a sequence of pairs of graphs, each with a special node, such that the property holds in the special node of the first graph and fails in the special node of the second, while for every $\mathcal{BLC}$-GNN $\mathcal{A}$, for any sufficiently large pairs of graphs in this sequence, the special nodes are indistinguishable by $\mathcal{A}$.

Thus far the results in this section are stated for directed graphs. We explain briefly why the undecidability and expressiveness separation results on GNNs with unbounded activation functions apply also to undirected graphs. For the expressiveness results, note that the graphs that we constructed in the proof of Lemma 45 are undirected. Hence the expressiveness gap between $\mathcal{BL}$ReLU-GNN and $\mathcal{BLC}$-GNN still exists for the undirected case.

▶ **Theorem 48.** *$\mathcal{BL}$ReLU-GNN is strictly more expressive over undirected graphs than $\mathcal{BLC}$-GNN.*

To obtain the undecidability of the universal satisfiability problem over undirected graphs of $\mathcal{BL}$ReLU-GNN, we again reduce from two-counter machines, but now with a modification to guarantee the direction of the transition.

▶ **Lemma 49.** *For every two-counter machine $\mathcal{M}$ with n instructions, there exists an $(n+8)$-$\mathcal{L}$-M2P^2 formula $\Psi_{\mathcal{M}}(x)$ such that $\mathcal{M}$ halts if and only if $\forall x\ \Psi_{\mathcal{M}}(x)$ is finitely satisfiable over undirected graphs.*

▶ **Theorem 50.** *The universal satisfiability problem of $\mathcal{BL}$ReLU-GNNs over undirected graphs is undecidable.*

4.2 Decidability of satisfiability for "modal" GNNs with unbounded activation functions

Thus the situation for universal satisfiability contrasts with the eventually constant case. What about the *satisfiability problem*? We do not know whether it is decidable for GNNs with piecewise linear activations, or even with just ReLU. We can see that even simple unbounded activation functions produce unbounded spectra, so the proof technique in the truncated case certainly will not work. For example, consider the following 1-layer 1-$\mathcal{BL}$ReLU-GNN. The dimensions are $d_0 = d_1 = 1$; the coefficient matrices C^1 and A^1_{in} are zero matrices; $\left(A^1_{\text{out}}\right)_{1,1} = 1$; the bias vector b^1 is a zero vector. It is not difficult to see that the value of $\xi^1_1(v)$ is the number of out-neighbors of v. Hence, the 1-spectrum of this GNN is the set of natural numbers.

We present a decidability result for the "modal version": aggregation over nodes connected by outgoing edges only, within a directed graph:

▶ **Theorem 51.** *The satisfiability problem of $\mathcal{OLPW}$-GNNs is decidable.*

Analogously to what we did in the eventually constant case, we describe all the possible values of a given activation function. Unlike in the eventually constant case, this will not be a finite set, but it will be *semi-linear*: that is, describable using a formula of Presburger arithmetic. We will first review the notion of semi-linear set that we use, where we modify the standard notion to deal with rational numbers. We then show that the set of all possible values output by a GNN is a semi-linear set.

For $a_0 \in \mathbb{Q}^k$ and $A = \{a_1, a_2, \dots, a_m\}$ a finite subset of $\mathbb{Q}^k$, we define:

$$\mathbb{N}\text{-}\mathsf{Span}(a_0, A) \;:=\; \left\{ a_0 + \sum_{1\leq i\leq m} n_i a_i \,\middle|\, n_i \in \mathbb{N} \right\}.$$

A set $S \subseteq \mathbb{Q}^k$ is a *linear set*, if there is $a_0 \in \mathbb{Q}^k$ and a finite set $A \subseteq \mathbb{Q}^k$, such that S is $\mathbb{N}\text{-}\mathsf{Span}(a_0, A)$. The pair (a_0, A) is called *the basis of* S. A *semi-linear set* is a finite union of linear sets. A basis of a semi-linear set $\bigcup_{1\leq i\leq k} \mathbb{N}\text{-}\mathsf{Span}\left(a_0^k, A^k\right)$ is the set $\left\{\left(a_0^1, A^1\right), \left(a_0^2, A^2\right), \dots, \left(a_0^k, A^k\right)\right\}$.

For semi-linear sets $S_1, S_2, S \subseteq \mathbb{Q}^k$, we use the following operators:

$$T(S) \;:=\; \{T(a) |\; a \in S\} \qquad \text{where } T : \mathbb{Q}^k \to \mathbb{Q}^m \text{ is an affine transformation}$$

$$\mathsf{KleeneStar}(S) \;:=\; \left\{ \sum_{s\in S'} s \,\middle|\, \text{For every finite multi-subset } S' \text{ of } S \right\}$$

We recall that in the context of integers, both operators are known to preserve semi-linearity and the basis of the resulting semi-linear set can be computed. See, e.g., [5, 11, 7]. The arguments adapt easily to our rational setting, thus *we assume below that we have an algorithm for pushing semi-linear representations through these operators.*

We consider piecewise linear functions, defined by a sequence $((I_1, f_1), \dots, (I_p, f_p))$ where $I_1 \cup \dots \cup I_p$ is a partition of $\mathbb{Q}$ into p intervals and each $f_i : \mathbb{Q} \to \mathbb{Q}$ is an affine function. The sequence $((I_1, f_1), \dots, (I_p, f_p))$ defines a function where x is mapped to $f_i(x)$ if x is in the interval I_i. We apply a piecewise linear function on some fixed components of a vector, which is captured by the following notation. For a piecewise linear function $f : \mathbb{Q} \to \mathbb{Q}$, a rational vector $a \in \mathbb{Q}^k$ and $K \subseteq [1, k]$, we write $f_K(a)$ to denote the vector $b \in \mathbb{Q}^k$ where $b_i = f(a_i)$ for every $i \in K$ and $b_i = a_i$ for every $i \notin K$. In other words, $f_K(a)$ only applies the function f on the components in K and the identity function on the components outside K. Similar to affine transformation and Kleene star, *piecewise linear functions also preserve semi-linearity and the basis of the resulting semi-linear sets can be computed.* Again, we can easily adapt the argument in [7] to our rational setting.

To prove Theorem 51, we need two more definitions. Let $\mathcal{A}$ be a L-layer $\mathcal{OLPW}$-GNN. Let $d_0, d_1, \dots, d_L$ be the dimension of the layers. We denote by $\mathbb{Q}^{[d_0, d_1, \dots, d_{\ell-1}]}$ the Cartesian product $\mathbb{Q}^{d_0} \times \mathbb{Q}^{d_1} \dots \times \mathbb{Q}^{d_{\ell-1}}$. Given an element m of this product, the i^{th} component of m, denoted by $m[i]$, is the projection of m to $\mathbb{Q}^{d_i}$.

For a graph $\mathcal{G}$, vertex $v \in V$, and $0 \leq \ell \leq L$, the *ℓ-history of v in $\mathcal{G}$* (w.r.t. $\mathcal{A}$), denoted by $\mathsf{hist}^{\ell}_{\mathcal{G}}(v) \in \mathbb{Q}^{[d_0, d_1, \dots, d_\ell]}$, is the tuple that collects the first $(\ell + 1)$ feature vectors of v. Formally, for $0 \leq i \leq \ell$, $\left(\mathsf{hist}^{\ell}_{\mathcal{G}}(v)\right)[i] = \xi^i_{\mathcal{G}}(v)$. When the graph $\mathcal{G}$ is clear from the context, we omit it and simply write $\mathsf{hist}^{\ell}(v)$. The *ℓ-history-space* of $\mathcal{A}$ is the set of all possible histories.

We now state our representation theorem, which immediately implies Theorem 51:

▶ **Theorem 52.** *For every $\mathcal{OLPW}$-GNN $\mathcal{A}$ and $0 \leq \ell \leq L$, the ℓ-history-space is semi-linear, and its basis can be effectively computed.*

We contrast the theorem with Theorem 17. There we could only overapproximate the spectrum, because we could not determine which numbers from previously layers were simultaneously realizable. By inductively maintaining the entire history at each node, we have enough information to resolve these questions of consistency, and compute an *exact* representation of the semantic object, not just an overapproximation.

The rest of this section is devoted to the proof of Theorem 52. We will first explain the intuition behind it. Let $\mathcal{A}$ be a L-layer $\mathcal{OLPW}$-GNN, as in Definition 2. Let $\mathcal{G}$ be a graph and v be a vertex. Recall that for every $1 \leq \ell \leq L$, the ℓ-feature vector of v is:

$$\xi^\ell(v) \;:=\; f^\ell\left(C^\ell \xi^{\ell-1}(v) + A^\ell_{\text{out}} \sum_{u\in\mathcal{N}_{\text{out}}(v)} \xi^{\ell-1}(u) + b^\ell\right).$$

We can rewrite it in terms of history:

$$\mathsf{hist}^\ell(v)[0] \;= \xi^0(v), \tag{1}$$

and for each $1 \leq i \leq \ell$:

$$\mathsf{hist}^\ell(v)[i] \;= f^i\left(C^i \cdot \mathsf{hist}^\ell(v)[i-1] \;+\; A^i_{\text{out}} \cdot \left(\sum_{u\in\mathcal{N}_{\text{out}}(v)} \mathsf{hist}^{\ell-1}(u)\right)[i-1] \;+\; b^i\right). \tag{2}$$

Thus, the ℓ-history of v can be computed by applications of sum, affine transformations, and piecewise linear functions on the sum of the history of its out-neighbors.

We formalise this intuition in the following paragraphs. For each $0 \leq \ell \leq L$, we define the set $\mathcal{H}^\ell$:

$$\mathcal{H}^0 \;:=\; \{0,1\}^{d_0}$$
$$\mathcal{H}^\ell \;:=\; \bigcup_{e\in\{0,1\}^{d_0}} \mathsf{proj}_\ell \circ T_\ell \circ T_{\ell-1} \circ \ldots \circ T_{0,e} \circ \mathsf{KleeneStar}\left(\mathcal{H}^{\ell-1}\right),$$

where the definition and intuition of each $\mathsf{proj}_\ell, T_\ell, \ldots, T_1, T_{0,e}$ is as follows.

- Intuitively $\mathsf{KleeneStar}\left(\mathcal{H}^{\ell-1}\right)$ captures the term $\sum_{u\in\mathcal{N}_{\text{out}}(v)} \mathsf{hist}^{\ell-1}(u)$ in Equation (2).
- $T_{0,e} : \mathbb{Q}^{[d_0,\ldots,d_{\ell-1}]} \to \mathbb{Q}^{[d_0,\ldots,d_{\ell-1},d_0]}$ is an affine transformation that maps a to (a,e), i.e., it simply "pads" e into a.
- For each $1 \leq i \leq \ell$, the transformation $T_i : \mathbb{Q}^{[d_0,\ldots,d_{\ell-1},d_0,\ldots,d_{i-1}]} \to \mathbb{Q}^{[d_0,\ldots,d_{\ell-1},d_0,\ldots,d_{i-1},d_i]}$ computes the vector $\mathsf{hist}^\ell(v)[i]$ defined in Equation (2) and pads it at the end. Formally, T_i maps a to (a,c) where $c = \; f^i\left(C^i a[\ell+i-1] + A^i_{\text{out}} a[i-1] + b^i\right)$.
- Finally, $\mathsf{proj}_\ell : \mathbb{Q}^{[d_0,\ldots,d_{\ell-1},d_0,\ldots,d_\ell]} \to \mathbb{Q}^{[d_0,\ldots,d_\ell]}$ is a projection that projects out the first ℓ components.

We can show that $\mathcal{H}^\ell$ is a semi-linear set, and this captures the ℓ-history-space, as stated formally in Lemma 53. Note that Theorem 51 follows easily from the lemma and the computability of the basis of $\mathcal{H}^\ell$.

▶ **Lemma 53.** *For every $\mathcal{OLPW}$-GNN $\mathcal{A}$ and $0 \leq \ell \leq L$,*

1. *$\mathcal{H}^\ell$ is a semi-linear set.*
2. *For every $h \in \mathbb{Q}^{[d_0,d_1,\ldots,d_\ell]}$, the following are equivalent.*
   - *$h \in \mathcal{H}^\ell$*
   - *There exists a graph $\mathcal{G}$ and vertex $v \in V$ such that $\mathsf{hist}^\ell(v) = h$.*

Proof. The first item follows immediately from the fact that $\mathcal{H}^0$ is semi-linear and the operators Kleene star, affine transformations and piecewise linear functions all preserve semi-linearity.

We now prove the second item by induction on ℓ. The base case $\ell = 0$ is trivial.

For the induction hypothesis, we assume that the lemma holds for $\ell - 1$. We show that $h \in \mathcal{H}^\ell$ if and only if there is a graph $\mathcal{G}$ and a vertex v such that $\mathsf{hist}^\ell(v) = h$.

We start with the "only if" direction. Suppose $h \in \mathcal{H}^\ell$. By definition, there is $e \in \{0,1\}^{d_0}$ and a finite multi-subset $\{\{h_1, h_2, \dots, h_k\}\}$ of $\mathcal{H}^{\ell-1}$ such that:

$$h = \mathsf{proj}_\ell \circ T_\ell \circ T_{\ell-1} \circ \dots \circ T_{0,e} \left(h_1 + h_2 + \dots + h_k\right)$$

By the induction hypothesis, there exist graphs $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_k$ and vertices $v_1, v_2, \dots, v_k$ such that $\mathsf{hist}^{\ell-1}_{\mathcal{G}_i}(v_i) = h_i$ for every $1 \le i \le k$.

Let $\mathcal{G}$ be the graph obtained by taking the disjoint union of $\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_k$ and adding a fresh vertex v. Recalling that $\xi^0_{\mathcal{G}}(v)$ can achieve an arbitrary combination of $\{0.1\}$ vectors, based on the colors of v, we set the colors so that $\xi^0_{\mathcal{G}}(v) = e$. We have an outgoing edge from v to v_i for each $1 \le i \le k$. It is routine to verify that the ℓ-history of v is precisely h. Note that *because $\mathcal{A}$ is outgoing-only, the edge from v to v_i has no effect on the $(\ell - 1)$-history of v_i*. Thus $\mathsf{hist}^{\ell-1}_{\mathcal{G}}(v_i) = \mathsf{hist}^{\ell-1}_{\mathcal{G}_i}(v_i)$

For the "if" direction, let $\mathcal{G}$ be a graph and v be a vertex. Let $v_1, \dots, v_k$ be the out-neighbors of v. By definition, for each $1 \le i \le \ell$:

$$\mathsf{hist}^\ell_{\mathcal{G}}(v)[i] \;=\; f^i \left(C^i \cdot \mathsf{hist}^\ell_{\mathcal{G}}(v)[i-1] \;+\; A^i_{\text{out}} \cdot \left(\sum_{u \in \mathcal{N}_{\text{out}}(v)} \mathsf{hist}^{\ell-1}_{\mathcal{G}}(u) \right)[i-1] \;+\; b^i \right).$$

It is routine to verify that:

$$\mathsf{hist}^\ell_{\mathcal{G}}(v) \;=\; \mathsf{proj}_\ell \circ T_\ell \circ T_{\ell-1} \circ \dots \circ T_{0,e} \left(\mathsf{hist}^{\ell-1}_{\mathcal{G}}(v_1) + \mathsf{hist}^{\ell-1}_{\mathcal{G}}(v_2) + \dots + \mathsf{hist}^{\ell-1}_{\mathcal{G}}(v_k) \right),$$

where $e = \xi^0_{\mathcal{G}}(v)$. Therefore, $\mathsf{hist}^\ell_{\mathcal{G}}(v) \in \mathcal{H}^\ell$. ◀

5 Discussion

This work extends the exploration of the relationship between aggregate-combine GNNs and logic, with exact characterizations of expressiveness for GNNs with eventually constant activation functions, and embedding a logic into the GNNs with standard ReLU activations. We also obtain both decidability and undecidability results, some using the logical characterizations and some by porting the techniques used for decidability of the logics to apply directly on the GNNs. Perhaps the main take-away, echoing the theme of [1], is that Presburger logics and the techniques for analyzing them can be relevant to GNNs.

We have left open one major technical problem: the decidability of satisfiability for standard GNNs using the ReLU activation function. Here we have proven decidability only for the "outgoing-only" variant. We also do not know whether the undecidability results we have proven – e.g. for standard GNNs with global readout – still hold for the variants with outgoing-only aggregation. Thus, for all we know, the most crucial dividing line for decidability could revolve around outgoing-only vs bidirectional aggregation, rather than (e.g.) local vs global aggregation or truncation vs non-truncation in the activation function.

Looking at broader open issues, we focused here on some very basic verification problems on GNNs: can a certain classification be achieved? But it is clear that our techniques apply to many other logic-based verification problems; for example, it can be applied to determine whether a GNN can achieve a certain classification on a graph satisfying a certain sentence – provided that the sentence is also in one of our decidable logics.

We have not focused on complexity in this paper. Of course, for the broad class of GNNs with eventually constant activation functions, it is difficult to talk about complexity bounds. For GNNs based on truncated ReLU and local aggregation, we have shown satisfiability is PSPACE-complete, and is NP-complete for a fixed number of layers. The finer-grained complexity analysis for other decidability results is left for future work.

Our work provides motivation for exploring the properties of Presburger logics over relational structures and their connections with GNNs beyond the setting here, which considers only graphs with discrete feature values from a fixed set. In our ongoing work we are adapting our techniques to deal with GNNs whose feature values are *unbounded integers*, specified by an initial semi-linear set.

References

1 Pablo Barceló, Egor V. Kostylev, Mikaël Monet, Jorge Pérez, Juan L. Reutter, and Juan Pablo Silva. The Logical Expressiveness of Graph Neural Networks. In *ICLR*, 2020.

2 Pablo Barceló, Alexander Kozachinskiy, Anthony Wijdada Lin, and Vladamir Podolskii. Logical languages accepted by transformer encoders with hard attention, 2023. https://arxiv.org/pdf/2310.03817.pdf.

3 Bartosz Bednarczyk, Maja Orlowska, Anna Pacanowska, and Tony Tan. On classical decidable logics extended with percentage quantifiers and arithmetics. In *FSTTCS*, 2021.

4 David Chiang, Peter Cholak, and Anand Pillay. Tighter bounds on the expressivity of transformer encoders. In *ICML*, 2023.

5 Dmitry Chistikov and Christoph Haase. The taming of the semi-linear set. In *ICALP*, 2016.

6 F. Eisenbrand and G. Shmonin. Carathéodory bounds for integer cones. *Oper. Res. Lett.*, 34(5):564–568, 2006.

7 Seymour Ginsburg and Edwin H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific J. Math.*, 16(2):285–296, 1966.

8 Valentin Goranko and Martin Otto. Model theory of modal logic. In Patrick Blackburn, J. F. A. K. van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*. North-Holland, 2007.

9 Martin Grohe. The logic of graph neural networks. In *LICS*, 2021.

10 Martin Grohe. The descriptive complexity of graph neural networks. In *LICS*, 2023.

11 C. Haase and Georg Zetzsche. Presburger arithmetic with stars, rational subsets of graph groups, and nested zero tests. *LICS*, 2019.

12 Viktor Kuncak, Huu Hai Nguyen, and Martin Rinard. An Algorithm for Deciding BAPA: Boolean Algebra with Presburger Arithmetic. In *CADE*, 2005.

13 Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput.*, 6(3):467–480, sep 1977.

14 Chia-Hsuan Lu and Tony Tan. On two-variable guarded fragment logic with expressive local Presburger constraints. *CoRR*, abs/2206.13731, 2022.

15 Marvin L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., USA, 1967.

16 C. Papadimitriou. On the complexity of integer programming. *J. ACM*, 28(4):765–768, 1981.

17 Marco Sälzer and Martin Lange. Fundamental limits in formal verification of message-passing neural networks. In *ICLR*, 2023.

18 Manfred Schmidt-Schaubß and Gert Smolka. Attributive concept descriptions with complements. *Artif. Intell.*, 48(1):1–26, feb 1991.

19 Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks. In *ICLR*, 2019.

A Proofs from Section 2: Corollary 12

A.1 Proof of Corollary 12

Let us recall the corollary (of Theorem 11).

▶ **Corollary 12.** *The finite satisfiability problem of* $\mathcal{L}$-MP^2 *is decidable.*

Proof. Let $\varphi(x)$ be an n-$\mathcal{L}$-MP^2 formula and U_{n+1} be a fresh unary predicate. We claim that $\varphi(x)$ is finitely satisfiable if and only if the GP^2 sentence $\psi := \exists x\, U_{n+1}(x) \wedge \varphi(x)$ is also finitely satisfiable. Then the corollary follows from the decidability of the finite satisfiability problem of GP^2 by Theorem 11.

If $\varphi(x)$ is finitely satisfiable by the n-graph $\mathcal{G}$ and vertex $v \in G$, let $\mathcal{G}'$ be the $(n+1)$-graph that extended $\mathcal{G}$ with $U_{n+1} := \{v\}$. Then $\mathcal{G}' \models U_{n+1}(v)$, which implies that $\mathcal{G}' \models \psi$. Hence ψ is finitely satisfiable by $\mathcal{G}'$.

If ψ is finitely satisfiable by the $(n+1)$-graph $\mathcal{G}$, let $\mathcal{G}'$ be the n-graph that restricted $\mathcal{G}$ by removing U_{n+1}. By definition, there exists at least one vertex $v \in V$ such that $\mathcal{G} \models U_{n+1}(v) \wedge \varphi(v)$, which implies that $\mathcal{G} \models \varphi(v)$. Since there is no U_{n+1} in $\varphi(x)$, $\mathcal{G}' \models \varphi(v)$. Hence $\varphi(x)$ is finitely satisfiable by $\mathcal{G}'$. ◀

B Proofs from Subsection 3.1: results about the spectrum, translation from GNNs with eventually constant activation functions to logic, and decidability results for eventually constant local GNNs

B.1 Proof of Lemma 19: syntactic overapproximation of the spectrum

We recall first the definition of our overapproximation of the spectrum

$$\begin{aligned}
\uparrow\mathcal{S}^0 &:= \{0,1\}^{d_0} \\
\uparrow\mathcal{S}^\ell_s &:= \left\{ f^\ell\left(C^\ell s + \sum_{x\in\{\text{out},\text{in}\}} A^\ell_x \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} s' n_x^{A,s'} + R^\ell \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} s' n^{R,s'} + b^\ell \right) \middle| \, n_x^{A,s'}, n^{R,s'} \in \mathbb{N} \right\} \\
\uparrow\mathcal{S}^\ell &:= \bigcup_{s\in\uparrow\mathcal{S}^{\ell-1}} \uparrow\mathcal{S}^\ell_s
\end{aligned}$$

With this in mind, recall Lemma 19:

▶ **Lemma 19.** *For every* n-$\mathcal{BGC}$-GNN $\mathcal{A}$ *and* $0 \leq \ell \leq L$, *for every* n*-graph* $\mathcal{G}$ *and vertex* $v \in V$, *there exists* $s \in \uparrow\mathcal{S}^\ell$, *such that* $\xi^\ell(v) = s$.

Proof. The proof is by induction on layers. The base case $\ell = 0$ is straightforward. For the inductive step $1 \leq \ell \leq L$, for every vertex $u \in V$, by the induction hypothesis, there exists $s_u \in \uparrow\mathcal{S}^{\ell-1}$ such that $\xi^{\ell-1}(u) = s_u$.

For every $s' \in \uparrow\mathcal{S}^{\ell-1}$, let $V_{s'} := \left\{u \in V \middle| \, \xi^{\ell-1}(u) = s'\right\}$. We can rewrite the following summations.

$$\begin{aligned}
\sum_{u\in\mathcal{N}_x(v)} \xi^{\ell-1}(u) &= \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} \sum_{u\in\mathcal{N}_x(v)\cap V_{s'}} s' = \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} n_{x,v}^{A,s'} s' \\
\sum_{u\in V} \xi^{\ell-1}(u) &= \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} \sum_{u\in V\cap V_{s'}} s' = \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} n_v^{R,s'} s',
\end{aligned}$$

where $n_{x,v}^{A,s'} := |\mathcal{N}_x(v) \cap V_{s'}| \in \mathbb{N}$ and $n_v^{R,s'} := |V \cap V_{s'}| \in \mathbb{N}$. Hence $\xi^\ell(v)$ can be rewritten:

$$\xi^\ell(v) = f^\ell\left(C^\ell s_v + \sum_{x\in\{\text{out},\text{in}\}}\left(A_x^\ell \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} s' n_{x,v}^{A,s'}\right) + R^\ell \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} s' n_v^{R,s'} + b^\ell\right)$$

By the definition of $\uparrow\mathcal{S}^\ell$, $\xi^\ell(v) \in \uparrow\mathcal{S}^\ell$. ◀

B.2 Proof of Lemma 20: overapproximation is finite and computable

We recall Lemma 20.

▶ **Lemma 20.** *For every n-$\mathcal{BGC}$-GNN $\mathcal{A}$ and $0 \le \ell \le L$, $\uparrow\mathcal{S}^\ell$ has finite size and can be computed.*

To prove the lemma, we need more tools. The following "canonical eventually-constant function" will play a key role:

▶ **Definition 54.** *For $t_{\mathsf{left}}, t_{\mathsf{right}} \in \mathbb{Q}$ and $t_{\mathsf{left}} \le t_{\mathsf{right}}$, the* clamp between t_{left} and t_{right}, *denoted by $\mathsf{clp}_{t_{\mathsf{left}}}^{t_{\mathsf{right}}}$, is a function mapping $\mathbb{Q}$ to $\mathbb{Q}$. For every $x \in \mathbb{Q}$, if $x \le t_{\mathsf{left}}$, $\mathsf{clp}_{t_{\mathsf{left}}}^{t_{\mathsf{right}}}(x) = t_{\mathsf{left}}$; if $x \ge t_{\mathsf{right}}$, $\mathsf{clp}_{t_{\mathsf{left}}}^{t_{\mathsf{right}}}(x) = t_{\mathsf{right}}$; otherwise, $\mathsf{clp}_{t_{\mathsf{left}}}^{t_{\mathsf{right}}}(x) = x$.*

We also need the following set $\uparrow\tilde{\mathcal{S}}^\ell$. Recall that t_{left}^ℓ and t_{right}^ℓ are the left and right thresholds of the ℓ^{th} eventually constant activation function in the GNN.

▶ **Definition 55.** *For every $\mathcal{BGC}$-GNN $\mathcal{A}$, for $1 \le \ell \le L$, the set $\uparrow\tilde{\mathcal{S}}^\ell$ is defined as follows:*

$$\uparrow\tilde{\mathcal{S}}_s^\ell := \left\{\mathsf{clp}_{t_{\mathsf{left}}^\ell}^{t_{\mathsf{right}}^\ell}\left(C^\ell s + \sum_{x\in\{out,in\}} A_x^\ell \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} s' n_x^{A,s'} + R^\ell \sum_{s'\in\uparrow\mathcal{S}^{\ell-1}} s' n^{R,s'} + b^\ell\right) \middle| \; n_x^{A,s'}, n^{R,s'} \in \mathbb{N}\right\}$$

$$\uparrow\tilde{\mathcal{S}}^\ell := \bigcup_{s\in\uparrow\mathcal{S}^{\ell-1}} \uparrow\tilde{\mathcal{S}}_s^\ell$$

The intuition for $\uparrow\tilde{\mathcal{S}}^\ell$ is that it represents the inverse image of $\uparrow\mathcal{S}^\ell$ under the activation function f^ℓ, modified by the clamp function. We will show that this modification preserves the value after applying f^ℓ. Note that the value of each element in $\uparrow\tilde{\mathcal{S}}^\ell$ is bounded, by the definition of the clamp function.

We will first show that the clamp function would be absorbed by any eventually constant function.

▶ **Lemma 56.** *For every eventually constant function f with threshold t_{left} and t_{right}, for every $x \in \mathbb{Q}$, $f\left(\mathsf{clp}_{t_{\mathsf{left}}}^{t_{\mathsf{right}}}(x)\right) = f(x)$.*

Proof. If $x \le t_{\mathsf{left}}$, $f\left(\mathsf{clp}_{t_{\mathsf{left}}}^{t_{\mathsf{right}}}(x)\right) = f(t_{\mathsf{left}}) = f(x)$; if $x \ge t_{\mathsf{right}}$, $f\left(\mathsf{clp}_{t_{\mathsf{left}}}^{t_{\mathsf{right}}}(x)\right) = f(t_{\mathsf{right}}) = f(x)$; otherwise, $f\left(\mathsf{clp}_{t_{\mathsf{left}}}^{t_{\mathsf{right}}}(x)\right) = f(x)$. ◀

Next, we prove that subsets of $\uparrow\tilde{\mathcal{S}}^\ell$ are finite and can be computed from the GNN descriptions.

▶ **Lemma 57.** *For $d \in \mathbb{N}^+$, a finite set $\mathcal{Q} \subseteq \mathbb{Q}^d$, $c \in \mathbb{Q}^d$, $p, q \in \mathbb{Q}$, and $p \le q$, let $\mathcal{X}_{\mathcal{Q},c,[p,q]} := \left\{\mathsf{clp}_p^q\left(\sum_{r\in Q} r n_r + c\right) \middle| \; n_r \in \mathbb{N}\right\}$. The set $\mathcal{X}_{\mathcal{Q},c,[p,q]}$ is finite and can be computed.*

Proof. Let $D \in \mathbb{N}^+$ be the least common multiple of the numerators of c, p, q, and elements in $\mathcal{Q}$. There exist $a_r, a_c \in \mathbb{Z}^d$, $a_p, a_q \in \mathbb{Z}$ satisfying that $c = \frac{a_c}{D}$, $p = \frac{a_p}{D}$, $q = \frac{a_q}{D}$, and $r = \frac{a_r}{D}$ for each $r \in \mathcal{Q}$. Thus we can rewrite $\mathcal{X}_{\mathcal{Q},c,[p,q]}$ as follows:

$$\mathcal{X}_{\mathcal{Q},c,[p,q]} \;=\; \left\{ \frac{1}{D}\mathsf{clp}_{a_p}^{a_q}\left(\sum_{r \in Q} a_r n_r + a_c\right) \middle|\; n_r \in \mathbb{N} \right\} \;\subseteq\; \left\{ \frac{e}{D} \middle|\; e \in [a_p\,, a_q]^d \right\}.$$

The size of the the right hand side set is $(a_q - a_p + 1)^d$. Hence $\left|\mathcal{X}_{\mathcal{Q},c,[p,q]}\right| \le (a_q - a_p + 1)^d < \infty$.

Note that it is straightforward to enumerate the right hand side set. We can compute the set $\mathcal{X}_{\mathcal{Q},c,[p,q]}$ with the following procedure. For every element s in the right hand side set, we check if the equation $\sum_{r \in Q} a_r x_r + a_c = Ds$ has solution over $\mathbb{N}$. It is not difficult to show that $s \in \mathcal{X}_{\mathcal{Q},c,[p,q]}$ if and only if the equation has a solution over $\mathbb{N}$. ◀

We can now prove Lemma 20.

Proof. We prove the lemma by induction on layers. The base case $\ell = 0$ is straightforward. For the induction step $1 \le \ell \le L$, we first show that $\uparrow\tilde{\mathcal{S}}^{\ell-1}$ is finite. Note that $\uparrow\tilde{\mathcal{S}}^{\ell}_s = \mathcal{X}_{\mathcal{Q}^\ell, c^\ell_s, [t^\ell_{\mathsf{left}}, t^\ell_{\mathsf{right}}]}$, where

$$\begin{aligned} \mathcal{Q}^\ell \;&:=\; \left\{ A^\ell_{\text{out}} s' \middle|\; s' \in \uparrow\mathcal{S}^{\ell-1} \right\} \cup \left\{ A^\ell_{\text{in}} s' \middle|\; s' \in \uparrow\mathcal{S}^{\ell-1} \right\} \cup \left\{ R^\ell s' \middle|\; s' \in \uparrow\mathcal{S}^{\ell-1} \right\} \\ c^\ell_s \;&:=\; C^\ell s + b^\ell. \end{aligned}$$

By the induction hypothesis, $\mathcal{Q}^\ell$ is finite. Hence the size of $\uparrow\tilde{\mathcal{S}}^{\ell}$ can be upper bounded by Lemma 57:

$$\left|\uparrow\tilde{\mathcal{S}}^{\ell}\right| \;\le\; \sum_{s \in \uparrow\mathcal{S}^{\ell-1}} \left|\uparrow\tilde{\mathcal{S}}^{\ell}_s\right| \;=\; \sum_{s \in \uparrow\mathcal{S}^{\ell-1}} \left|\mathcal{X}_{\mathcal{Q}^\ell, c^\ell_s, [t^\ell_{\mathsf{left}}, t^\ell_{\mathsf{right}}]}\right| \;<\; \infty$$

Next, by Lemma 56, for every $s \in \uparrow\mathcal{S}^{\ell-1}$ and $n^{A,s'}_x, n^{R,s'} \in \mathbb{N}$,

$$\begin{aligned} & f^\ell\left(C^\ell s + \sum_{x \in \{\text{out},\text{in}\}} \left(A^\ell_x \sum_{s' \in \uparrow\mathcal{S}^{\ell-1}} s' n^{A,s'}_x \right) + R^\ell \sum_{s' \in \uparrow\mathcal{S}^{\ell-1}} s' n^{R,s'} + b^\ell \right) \\ =\; & f^\ell\left(\mathsf{clp}_{t^\ell_{\mathsf{left}}}^{t^\ell_{\mathsf{right}}} \left(C^\ell s + \sum_{x \in \{\text{out},\text{in}\}} \left(A^\ell_x \sum_{s' \in \uparrow\mathcal{S}^{\ell-1}} s' n^{A,s'}_x \right) + R^\ell \sum_{s' \in \uparrow\mathcal{S}^{\ell-1}} s' n^{R,s'} + b^\ell \right)\right) \end{aligned}$$

Hence we have the following relationship between $\uparrow\mathcal{S}^\ell_s$ and $\uparrow\tilde{\mathcal{S}}^\ell_s$:

$$\uparrow\mathcal{S}^\ell_s \;=\; \left\{ f^\ell(s) \middle|\; s \in \uparrow\tilde{\mathcal{S}}^{\ell} \right\}.$$

Therefore $\left|\uparrow\mathcal{S}^\ell\right| \le \left|\uparrow\tilde{\mathcal{S}}^\ell\right| < \infty$.

We can compute the set $\uparrow\mathcal{S}^\ell$ with the following recursive procedure. The base case $\ell = 0$ is trivial. For $1 \le \ell \le L$, we first compute the set $\uparrow\tilde{\mathcal{S}}^\ell$. For every $s \in \uparrow\mathcal{S}^{\ell-1}$, by Lemma 57, the set $\mathcal{X}_{\mathcal{Q}^\ell, c^\ell_s, [t^\ell_{\mathsf{left}}, t^\ell_{\mathsf{right}}]}$ can be computed. Therefore $\uparrow\tilde{\mathcal{S}}^\ell = \bigcup_{s \in \uparrow\mathcal{S}^{\ell-1}} \mathcal{X}_{\mathcal{Q}^\ell, c^\ell_s, [t^\ell_{\mathsf{left}}, t^\ell_{\mathsf{right}}]}$ can also be computed. Finally, we can obtain $\uparrow\mathcal{S}^\ell$ by applying f^ℓ on each element in $\uparrow\tilde{\mathcal{S}}^\ell$. ◀

B.3 Proof of Theorem 23: from GNNs with eventually constant activations to logic

We recall the theorem:

▶ **Theorem 23.** *For every n-$\mathcal{BGC}$-GNN $\mathcal{A}$, there exists an n-MP² formula $\Psi_{\mathcal{A}}(x)$, effectively computable from the description of $\mathcal{A}$, such that $\mathcal{A}$ and $\Psi_{\mathcal{A}}(x)$ are equivalent. In the case we start with an n-$\mathcal{BLC}$-GNN, the formula we obtain is in n-$\mathcal{L}$-MP².*

We prove a more general version:

▶ **Lemma 58.** *For every n-$\mathcal{BGC}$-GNN $\mathcal{A}$, $0 \leq \ell \leq L$, and $s \in \uparrow\mathcal{S}^\ell$, letting ξ^ℓ be the ℓ^{th} derived feature function of $\mathcal{A}$, there exists an n-MP² formula $\varphi^\ell_s(x)$, such that for every n-graph $\mathcal{G}$ and vertex $v \in V$, $\mathcal{G} \models \varphi^\ell_s(v)$ if and only if $\xi^\ell(v) = s$. In the case we start with an n-$\mathcal{BLC}$-GNN, the formula we obtain is in n-$\mathcal{L}$-MP².*

Proof. The intuition is that our formula hard codes all possible values of the spectrum, with formulas verifying that the computed value is that specific value. Since for arbitrary eventually constant functions we cannot represent the spectrum exactly, we use the overapproximations defined in the prior argument.

We define an MP² formula $\varphi^\ell_s(x)$ inductively on layers. For the base case $\ell = 0$, for $1 \leq i \leq d_0$,

$$\psi_{i,c}(x) \;:=\; \begin{cases} U_i(x), & \text{if } c = 1 \\ \neg U_i(x), & \text{if } c = 0 \end{cases}.$$

For $s \in \uparrow\mathcal{S}^0$,

$$\varphi^0_s(x) \;:=\; \bigwedge_{1 \leq i \leq d_0} \psi_{i,s_i}(x).$$

For the inductive case $1 \leq \ell \leq L$, for $\theta \in \uparrow\tilde{\mathcal{S}}^\ell$, $s' \in \uparrow\mathcal{S}^{\ell-1}$, and $1 \leq i \leq d_\ell$, we define the formula:

$$\tilde{\phi}^\ell_{\theta,s',i}(x) \;:=\; \Bigg(\Bigg(\left(C^\ell s' + b^\ell\right)_i + \sum_{\substack{x \in \{\text{out},\text{in}\} \\ s'' \in \uparrow\mathcal{S}^{\ell-1}}} \left(A^\ell_x s''\right)_i \cdot \#_y[\epsilon_x(x,y) \wedge \psi^{\ell-1}_{s''}(y)] + \sum_{s'' \in \uparrow\mathcal{S}^{\ell-1}} \left(R^\ell s''\right)_i \cdot \#_y[\psi^{\ell-1}_{s''}(y)]\Bigg) \circledast_{\theta_i} \theta_i\Bigg),$$

where $\epsilon_{\text{out}}(x,y) \;:=\; E(x,y)$ and $\epsilon_{\text{in}}(x,y) \;:=\; E(y,x)$. If $\theta_i = t^\ell_{\mathsf{left}}$, then $\circledast_{\theta_i}$ is $\leq$; if $\theta_i = t^\ell_{\mathsf{right}}$, then $\circledast_{\theta_i}$ is $\geq$; otherwise, $\circledast_{\theta_i}$ is $=$. For $s \in \uparrow\mathcal{S}^\ell$, $s' \in \uparrow\mathcal{S}^{\ell-1}$, and $1 \leq i \leq d_\ell$, we define:

$$\phi^\ell_{s,s'}(x) \;:=\; \bigvee_{\theta \text{ s.t. } \theta \in \uparrow\tilde{\mathcal{S}}^\ell \text{ and } f^\ell(\theta) = s} \left(\bigwedge_{1 \leq i \leq d_\ell} \tilde{\phi}^\ell_{\theta,s',i}(x)\right).$$

Finally, for $s \in \uparrow\mathcal{S}^\ell$, we define:

$$\varphi^\ell_s(x) \;:=\; \bigvee_{s' \in \uparrow\mathcal{S}^{\ell-1}} \left(\varphi^{\ell-1}_{s'}(x) \wedge \phi^\ell_{s,s'}(x)\right).$$

Note that by Lemma 20, both $\uparrow\mathcal{S}^\ell$ and $\uparrow\tilde{\mathcal{S}}^\ell$ are finite. Thus the disjunction in the construction is over a finite set.

We prove the correctness of the construction by induction on the layers.

For the base case $\ell = 0$, for every n-graph $\mathcal{G}$, vertex $v \in V$, and $s \in \uparrow\mathcal{S}^0$, it is straightforward to check that $\mathcal{G} \models \varphi^0_s(v)$ if and only if $\xi^0(v) = s$.

For the induction step $1 \le \ell \le L$, for every n-graph $\mathcal{G}$ and $s' \in \uparrow\mathcal{S}^{\ell-1}$, we define $w^\ell_{s'} : V \to \mathbb{Q}$ as follows:

$$w^\ell_{s'}(v) \;:=\; C^\ell s' + b^\ell + \sum_{\substack{x\in\{\text{out},\text{in}\}\\ s''\in\uparrow\mathcal{S}^{\ell-1}}} A^\ell_x s'' \cdot \left|\left\{u \in V \middle|\, \mathcal{G} \models \epsilon_x(v,u) \wedge \psi^{\ell-1}_{s''}(u)\right\}\right| + \sum_{s''\in\uparrow\mathcal{S}^{\ell-1}} R^\ell s'' \cdot \left|\left\{u \in V \middle|\, \mathcal{G} \models \psi^{\ell-1}_{s''}(u)\right\}\right|.$$

By the semantics of Presburger quantifiers, for $1 \le i \le d_\ell$, $\mathcal{G} \models \tilde{\phi}^\ell_{\theta,s',i}(v)$ if and only if $\left(w^\ell_{s'}(v)\right)_i \circledast_{\theta_i} \theta_i$. Let $V^\ell_{s''} := \left\{u \in V \middle|\, \mathcal{G} \models \psi^{\ell-1}_{s''}(u)\right\}$. By the induction hypothesis, $\mathcal{G} \models \psi^{\ell-1}_{s''}(u)$ if and only if $\xi^{\ell-1}(u) = s''$. Hence $V^\ell_{s''} = \left\{u \in V \middle|\, \xi^{\ell-1}(u) = s''\right\}$. We can rewrite $w^\ell_{s'}(v)$ as follows.

$$\begin{aligned} w^\ell_{s'}(v) &= C^\ell s' + b^\ell + \sum_{x\in\{\text{out},\text{in}\}} \left(A^\ell_x \sum_{s''\in\uparrow\mathcal{S}^{\ell-1}} \sum_{u\in\mathcal{N}_x(v)\cap V^\ell_{s''}} s'' \right) + R^\ell \sum_{s''\in\uparrow\mathcal{S}^{\ell-1}} \sum_{u\in V\cap V^\ell_{s''}} s'' \\ &= C^\ell s' + b^\ell + \sum_{x\in\{\text{out},\text{in}\}} \left(A^\ell_x \sum_{u\in\mathcal{N}_x(v)} \xi^{\ell-1}(u) \right) + R^\ell \sum_{u\in V} \xi^{\ell-1}(u) \end{aligned}$$

By the definition of feature vectors, we obtain that $\xi^\ell(v) = f^\ell\left(w^\ell_{\xi^{\ell-1}(v)}(v)\right)$.

If $\mathcal{G} \models \varphi^\ell_s(v)$, then there exists $s' \in \uparrow\mathcal{S}^{\ell-1}$ such that $\mathcal{G} \models \psi^{\ell-1}_{s'}(v)$. By the induction hypothesis, $\xi^{\ell-1}(v) = s'$. Because $\mathcal{G} \models \phi^\ell_{s,s'}(v)$, there exists $\theta \in \uparrow\tilde{\mathcal{S}}^\ell$ such that $f^\ell(\theta) = s$ and for $1 \le i \le d_\ell$, $\mathcal{G} \models \tilde{\phi}^\ell_{\theta,s',i}(v)$. By the semantics of Presburger quantifiers, $\mathcal{G} \models \tilde{\phi}^\ell_{\theta,s',i}(v)$ implies that $\left(w^\ell_{s'}(v)\right)_i \circledast_{\theta_i} \theta_i$.

- If $\theta_i = t^\ell_{\mathsf{left}}$, then $\left(w^\ell_{s'}(v)\right)_i \le t^\ell_{\mathsf{left}}$. Since f^ℓ is eventually constant with the left threshold t^ℓ_{left}, $\xi^\ell_i(v) = f^\ell\left(\left(w^\ell_{s'}(v)\right)_i\right) = f^\ell\left(t^\ell_{\mathsf{left}}\right) = f^\ell(\theta_i) = s_i$.
- If $\theta_i = t^\ell_{\mathsf{right}}$, then $\left(w^\ell_{s'}(v)\right)_i \ge t^\ell_{\mathsf{right}}$. Since f^ℓ is eventually constant with the right threshold t^ℓ_{right}, $\xi^\ell_i(v) = f^\ell\left(\left(w^\ell_{s'}(v)\right)_i\right) = f^\ell\left(t^\ell_{\mathsf{right}}\right) = f^\ell(\theta_i) = s_i$.
- If $t^\ell_{\mathsf{left}} < \theta_i < t^\ell_{\mathsf{right}}$, then $\left(w^\ell_{s'}(v)\right)_i = \theta_i$, which implies that $\xi^\ell_i(v) = f^\ell\left(\left(w^\ell_{s'}(v)\right)_i\right) = f^\ell(\theta_i) = s_i$.

Therefore $\xi^\ell(v) = s$.

Conversely, supppose $\xi^\ell(v) = s$. By the definition, $\xi^\ell(v) \in \uparrow\mathcal{S}^\ell$ and $\xi^{\ell-1}(v) \in \uparrow\mathcal{S}^{\ell-1}$. By the induction hypothesis, $\mathcal{G} \models \psi^{\ell-1}_{\xi^{\ell-1}(v)}(v)$. Let $\theta := \mathsf{clp}^{t^\ell_{\mathsf{right}}}_{t^\ell_{\mathsf{left}}}\left(w^\ell_{\xi^{\ell-1}(v)}(v)\right)$. By the definition, $\theta \in \uparrow\tilde{\mathcal{S}}^\ell$. For $1 \le i \le d_\ell$,

- if $\left(w^\ell_{\xi^{\ell-1}(v)}(v)\right)_i \le t^\ell_{\mathsf{left}}$, then $\left(w^\ell_{\xi^{\ell-1}(v)}(v)\right)_i \le \theta_i = t^\ell_{\mathsf{left}}$.
- if $\left(w^\ell_{\xi^{\ell-1}(v)}(v)\right)_i \ge t^\ell_{\mathsf{right}}$, then $\left(w^\ell_{\xi^{\ell-1}(v)}(v)\right)_i \ge \theta_i = t^\ell_{\mathsf{right}}$,
- if $t^\ell_{\mathsf{left}} < \left(w^\ell_{\xi^{\ell-1}(v)}(v)\right)_i < t^\ell_{\mathsf{right}}$, then $\left(w^\ell_{\xi^{\ell-1}(v)}(v)\right)_i = \theta_i$.

Hence by the semantics of Presburger quantifiers $\mathcal{G} \models \tilde{\phi}^{\ell}_{\theta,\xi^{\ell-1}(v),i}(v)$. Furthermore, by Lemma 56, $f^{\ell}(\theta) = f^{\ell}\left(\mathsf{clp}^{t^{\ell}_{\text{right}}}_{t^{\ell}_{\text{left}}}\left(w^{\ell}_{\xi^{\ell-1}(v)}(v)\right)\right) = f^{\ell}\left(w^{\ell}_{\xi^{\ell-1}(v)}(v)\right) = \xi^{\ell}(v)$, which implies that $\mathcal{G} \models \phi^{\ell}_{\xi^{\ell}(v),\xi^{\ell-1}(v)}(v)$. Therefore $\mathcal{G} \models \varphi^{\ell}_{\xi^{\ell}(v)}(v)$.

If $\mathcal{A}$ is an n-$\mathcal{BLC}$-GNN, then for $1 \leq \ell \leq L$, R^{ℓ} is a zero matrix. Then, there is no $\top$ guarded term in $\tilde{\phi}^{\ell}_{\theta,s',i}(x)$. Hence the formula we obtained is in $\mathcal{L}$-MP^2. ◀

We can now prove Theorem 23.

Proof. Let $\uparrow\mathcal{S}^{L}_{\geq 0.5} := \left\{s \in \uparrow\mathcal{S}^{L} \,\middle|\, s_1 \geq 0.5\right\}$ and $\Psi_{\mathcal{A}}(x) := \bigvee_{s \in \uparrow\mathcal{S}^{L}_{\geq 0.5}} \varphi^{L}_{s}(x)$, where $\varphi^{L}_{s}(x)$ is the formulas defined in Lemma 58.

For every n-graph $\mathcal{G}$ and vertex $v \in V$, if $\mathcal{G} \models \Psi_{\mathcal{A}}(v)$, then there exists $s \in \uparrow\mathcal{S}^{L}_{\geq 0.5}$, such that $\mathcal{G} \models \varphi^{L}_{s}(v)$. By Lemma 58, $\xi^{L}(v) = s$. Hence $\xi^{L}_{1}(v) = s_1 \geq 0.5$, which implies that $\mathcal{A}$ accepts $\langle \mathcal{G}, v\rangle$.

On the other hand, if $\mathcal{A}$ accepts $\langle \mathcal{G}, v\rangle$, as shown above, $\mathcal{G} \models \varphi^{L}_{\xi^{L}(v)}(v)$. By definition of acceptance, $\xi^{L}_{1}(v) \geq 0.5$, which implies that $\xi^{L}(v) \in \uparrow\mathcal{S}^{L}_{\geq 0.5}$. Therefore $\mathcal{G} \models \Psi_{\mathcal{A}}(v)$. This completes the proof of the theorem.

If $\mathcal{A}$ is an n-$\mathcal{BLC}$-GNN, the formulas defined in the paragraphs above are in n-$\mathcal{L}$-MP^2. Hence $\Psi_{\mathcal{A}}(x)$ is also in $\mathcal{L}$-MP^2. ◀

B.4 Proof of Theorem 26: from logic to GNNs with eventually constant activations

We recall the theorem, which is about going from logic to GNNs with truncated ReLU activations:

▶ **Theorem 26.** *For every n-MP^2 formula $\Psi(x)$, there exists an n-$\mathcal{BG}$TrReLU-GNN $\mathcal{A}_{\Psi}$, such that $\Psi(x)$ and $\mathcal{A}_{\Psi}$ are equivalent. If we start with an n-$\mathcal{L}$-MP^2 formula, we obtain an n-$\mathcal{BL}$TrReLU-GNN.*

For every n-MP^2 formula $\Psi(x)$, let L be the number of subformulas of $\Psi(x)$ and $\{\varphi_i(x)\}_{1 \leq i \leq L}$ be an enumeration of subformulas of $\Psi(x)$ that $\varphi_L(x)$ is $\Psi(x)$, and for each $\varphi_i(x)$ and $\varphi_j(x)$, if $\varphi_i(x)$ is a strict subformula of $\varphi_j(x)$, then $i < j$.

We define the $(L+1)$-layer n-$\mathcal{BG}$TrReLU-GNN $\mathcal{A}_{\Psi}$ as follows. The input dimension d_0 is n. For $1 \leq \ell \leq L$, the dimension d_{ℓ} is L, and $d_{L+1} = 1$. The numbers in the coefficient matrices and bias vectors are defined by the following rules. For $1 \leq i \leq L$,

- if $\varphi_i(x) = \top$, then $b^{i}_{i} = 1$.
- if $\varphi_i(x) = U_j(x)$ for some unary predicate U_j, then $C^{1}_{i,j} = 1$ and for $2 \leq \ell \leq i$, $C^{\ell}_{i,i} = 1$.
- if $\varphi_i(x) = \neg\varphi_j(x)$, then $C^{i}_{i,j} = -1$, $b^{i}_{i} = 1$.
- if $\varphi_i(x) = \varphi_{j_1}(x) \wedge \varphi_{j_2}(x)$, then $C^{i}_{i,j_1} = C^{i}_{i,j_2} = 1$, $b^{i}_{i} = -1$.
- if $\varphi_i(x) = \left(\sum_{t=1}^{k} \lambda_t \cdot \#_y[\epsilon_t(x,y) \wedge \varphi_{j_t}(y)] \geq \delta\right)$, then $b^{i}_{i} = 1 - \delta$. For $1 \leq t \leq k$,
  - if $\epsilon_t(x,y) = E(x,y)$, then $\left(A^{i}_{\text{out}}\right)_{i,j_t} = \lambda_t$.
  - if $\epsilon_t(x,y) = E(y,x)$, then $\left(A^{i}_{\text{in}}\right)_{i,j_t} = \lambda_t$.
  - if $\epsilon_t(x,y) = \top$, then $R^{i}_{i,j_t} = \lambda_t$.

For $1 \leq i \leq L$ and $i+1 \leq \ell \leq L$, $C^{\ell}_{i,i} = 1$. $C^{L+1}_{1,L} = 1$. All other numbers in the coefficient matrices are 0.

The theorem will follow once we have shown the following property of $\mathcal{A}_{\Psi}$:

▶ **Lemma 59.** *Let ξ^ℓ be the derived feature functions of $\mathcal{A}_\Psi$. For every n-graph $\mathcal{G}$ and vertex $v \in V$, for $1 \le i \le L$ and $i \le \ell \le L$, if $\mathcal{G} \models \varphi_i(v)$, then $\xi_i^\ell(v) = 1$. Otherwise, if $\mathcal{G} \not\models \varphi_i(x)$, then $\xi_i^\ell(v) = 0$.*

Proof. For $i+1 \le \ell \le L$, $\xi_i^\ell(v) = \mathsf{TrReLU}\left(\xi_i^{\ell-1}(v)\right) = \xi_i^i(v)$. Thus it is sufficient to show the property holds for $\ell = i$. We prove the property by induction on subformulas.

- If $\varphi_i(x) = \top$, then $\xi_i^i(v) = \mathsf{TrReLU}\,(1) = 1$.
- If $\varphi_i(x) = U_j(x)$ for some unary predicate U_j, $\xi_i^\ell(v) = \mathsf{TrReLU}\left(\xi_i^{\ell-1}(v)\right) = \xi_j^0(v)$. If $\mathcal{G} \models \varphi_i(v)$, then $\mathcal{G} \models U_j(v)$, by definition, $\xi_j^0(v) = 1$. Hence $\xi_i^i(v) = 1$. Otherwise, if $\mathcal{G} \not\models \varphi_i(v)$, then $\mathcal{G} \not\models U_j(v)$, by definition, $\xi_j^0(v) = 0$. Hence $\xi_i^i(v) = 0$.
- If $\varphi_i(x) = \neg\varphi_j(x)$, then $\xi_i^i(v) = \mathsf{TrReLU}\left(1 - \xi_j^{i-1}(v)\right)$. Because $\varphi_j(x)$ is a strict subformula of $\varphi_i(x)$, $i > j$. If $\mathcal{G} \models \neg\varphi_j(v)$, then $\mathcal{G} \not\models \varphi_j(v)$. By the induction hypothesis, $\xi_j^{i-1}(v) = 0$. Hence $\xi_i^i(v) = 1$. Otherwise, if $\mathcal{G} \not\models \neg\varphi_j(v)$, then $\mathcal{G} \models \varphi_j(v)$. By the induction hypothesis, $\xi_j^{i-1}(v) = 1$. Hence $\xi_i^i(v) = 0$.
- If $\varphi_i(x) = \varphi_{j_1}(x) \wedge \varphi_{j_2}(x)$, then $\xi_i^i(v) = \mathsf{TrReLU}\left(\xi_{j_1}^{i-1}(v) + \xi_{j_2}^{i-1}(v) - 1\right)$. Because $\varphi_{j_1}(x)$ and $\varphi_{j_2}(x)$ are strict subformulas of $\varphi_i(x)$, $i > j_1$ and $i > j_2$. If $\mathcal{G} \models \varphi_{j_1}(v) \wedge \varphi_{j_2}(v)$, then $\mathcal{G} \models \varphi_{j_1}(v)$ and $\mathcal{G} \models \varphi_{j_2}(v)$. By the induction hypothesis, $\xi_{j_1}^{i-1}(v) = \xi_{j_2}^{i-1}(v) = 1$. Hence $\xi_i^i(v) = 1$. Otherwise, if $\mathcal{G} \not\models \varphi_{j_1}(v) \wedge \varphi_{j_2}(v)$, then $\mathcal{G} \not\models \varphi_{j_1}(v)$ or $\mathcal{G} \not\models \varphi_{j_2}(v)$. By the induction hypothesis, $\xi_{j_1}^{i-1}(v) + \xi_{j_2}^{i-1}(v) \le 1$. Hence $\xi_i^i(v) = 0$.
- If $\varphi_i(x) = \left(\sum_{t=1}^k \lambda_j \cdot \#_y[\epsilon_t(x,y) \wedge \varphi_{j_t}(y)] \ge \delta\right)$, for $1 \le t \le k$, because $\varphi_{j_t}(x)$ is a strict subformula of $\varphi_i(x)$, $i > j_t$. By the induction hypothesis, for every $u \in V$, if $\mathcal{G} \models \varphi_{j_t}(u)$, then $\xi_{j_t}^{i-1}(u) = 1$. Otherwise, if $\mathcal{G} \not\models \varphi_{j_t}(u)$, $\xi_{j_t}^{i-1}(u) = 0$.
  - If $\epsilon_t(x,y) = E(x,y)$, then
  $$|\{u \in V|\ \mathcal{G} \models \epsilon_t(v,u) \wedge \varphi_{j_t}(u)\}| = |\{u \in \mathcal{N}_{\text{out}}(v)|\ \mathcal{G} \models \varphi_{j_t}(u)\}| = \sum_{u \in \mathcal{N}_{\text{out}}(v)} \xi_{j_t}^{i-1}(u).$$
  We can treat the other two cases analogously:
  - If $\epsilon_t(x,y) = E(y,x)$, then
  $$|\{u \in V|\ \mathcal{G} \models \epsilon_t(v,u) \wedge \varphi_{j_t}(u)\}| = \sum_{u \in \mathcal{N}_{\text{in}}(v)} \xi_{j_t}^{i-1}(u).$$
  - If $\epsilon_t(x,y) = \top$, then
  $$|\{u \in V|\ \mathcal{G} \models \epsilon_t(v,u) \wedge \varphi_{j_t}(u)\}| = \sum_{u \in V} \xi_{j_t}^{i-1}(u).$$

  Let w be the value defined as follows.
  $$w := \sum_{t=1}^k \lambda_t \cdot |\{u \in V|\ \mathcal{G} \models \epsilon_t(v,u) \wedge \varphi_{j_t}(v)\}| = \left(\sum_{x \in \{\text{out},\text{in}\}} \left(A_x^i \sum_{u \in \mathcal{N}_x(v)} \xi^{i-1}(u)\right) + R^i \sum_{u \in V} \xi^{i-1}(u)\right)_i$$
  Because for $1 \le j \le L$, $C_{i,j}^i = 0$, $\left(C^i \xi^{i-1}(v)\right)_i = 0$. Thus
  $$\xi_i^i(v) = \mathsf{TrReLU}\left(\left(C^i\xi^{i-1}(v)\right)_i + w + b_i^i\right) = \mathsf{TrReLU}(w + 1 - \delta).$$
  If $\mathcal{G} \models \varphi_i(v)$, by the semantic of Presburger quantifiers, $w \ge \delta$. Hence $\xi_i^i(v) = 1$. On the other hand, if $\mathcal{G} \not\models \varphi_i(v)$, by the semantic of Presburger quantifiers, $w < \delta$. Hence $\xi_i^i(v) = 0$.

◀

We can now prove Theorem 26.

Proof. For every n-MP^2 formula $\Psi(x)$, let $\mathcal{A}_\Psi$ be the n-$\mathcal{BG}$TrReLU-GNN defined above. Note that $\xi_1^{L+1}(v) = \mathsf{TrReLU}\left(\xi_L^L(v)\right)$.

For every n-graph $\mathcal{G}$ and vertex $v \in V$, if $\mathcal{G} \models \Psi(v)$, since $\varphi_L(x)$ is $\Psi(x)$, we have $\mathcal{G} \models \varphi_L(v)$. By Lemma 59, $\xi_L^L(v) = 1$. Thus $\xi_1^{L+1}(v) = 1$ and $\mathcal{A}_\Psi$ accepts $\langle \mathcal{G}, v \rangle$. On the other hand, if $\mathcal{G} \not\models \Psi(v)$, we have $\mathcal{G} \not\models \varphi_L(v)$. By Lemma 59 again, $\xi_L^L(v) = 0$. Thus $\xi_1^{L+1}(v) = 0$ and $\mathcal{A}_\Psi$ does not accept $\langle \mathcal{G}, v \rangle$.

If $\Psi(x)$ is an n-$\mathcal{L}$-MP^2 formula, by construction, for $1 \le \ell \le L$, R^ℓ are zero matrices. Therefore $\mathcal{A}_\Psi$ is an n-$\mathcal{BL}$TrReLU-GNN. ◀

B.5 Proof of Theorem 28: decidability of universal satisfiability for GNNs with eventually constant activations and only local aggregation

We recall the theorem:

▶ **Theorem 28.** *The universal satisfiability problem of $\mathcal{BLC}$-GNNs is decidable.*

Proof. For every $\mathcal{BLC}$-GNN $\mathcal{A}$, by Theorem 23, there exists a $\mathcal{L}$-MP^2 formula $\varphi_\mathcal{A}(x)$ such that $\mathcal{A}$ and $\varphi_\mathcal{A}(x)$ are equivalent. We claim that $\mathcal{A}$ is universally satisfiable if and only if the GP^2 sentence $\psi := \forall x\ (x = x) \to \varphi_\mathcal{A}(x)$ is finitely satisfiable.

If ψ is finitely satisfiable by the graph $\mathcal{G}$, which implies that for every $v \in V$, $\mathcal{G} \models \varphi_\mathcal{A}(v)$. Since $\mathcal{A}$ and $\varphi_\mathcal{A}(x)$ are equivalent, $\mathcal{A}$ accepts $\langle \mathcal{G}, v \rangle$. Hence $\mathcal{A}$ is universally satisfiable, with witness graph $\mathcal{G}$. On the other hand, if $\mathcal{A}$ is universally satisfiable, with witness the finite graph $\mathcal{G}$, by definition, for every $v \in V$, $\mathcal{A}$ accepts $\langle \mathcal{G}, v \rangle$. Since $\mathcal{A}$ and $\varphi_\mathcal{A}(x)$ are equivalent, $\varphi_\mathcal{A}(x)$ also accepts $\langle \mathcal{G}, v \rangle$. Hence ψ is satisfiable by $\mathcal{G}$. Note that the size of $\mathcal{G}$ is finite, which implies that ψ is finitely satisfiable. ◀

C PSPACE-completeness of satisfiability for GNNs with local aggegation and truncated ReLU activations

Recall that in the body we proved decidability of satisfiability for GNNs with local aggregation, where the activation functions are computable, map rationals to rationals, and are eventually constant. We will deal here with a subclass: $\mathcal{BL}$TrReLU-GNNs, where the activations are truncated ReLU.

Our goal is to prove Theorem 25, which we now recall.

▶ **Theorem 25.** *For $\mathcal{BLC}$-GNNs with truncated ReLU activations, the satisfiability problem is PSPACE-complete. It is NP-complete when the number of layers is fixed.*

C.1 Exponential history-space property

The first step is to establish a bound on the size of the numbers that can be computed by GNNs with ReLU-based activation functions. Remember that we begin with a graph where the feature values are only binary, and in each layer we do one aggregation and truncate the result. Thus it is intuitive that we cannot build up large values in any intermediate result at any node, regardless of the size of the graphs. This property actually holds even with global aggregation, but we prove it here only for the local case, since this is the only one relevant to this proof.

▶ **Definition 60.** *For every* $\mathcal{BL}$TrReLU-GNN $\mathcal{A}$ *and* $1 \le \ell \le L$, *let* $\tilde{c}^\ell$ *be the product of the denominators of all entries of* C^ℓ, A^ℓ_{in}, A^ℓ_{out}, *and* b^ℓ. *For* $0 \le \ell \le L$, *the* ℓ-capacity *of* $\mathcal{A}$, *denoted by* c^ℓ, *is defined inductively:*

$$\begin{aligned} c^0 &:= 1 \\ c^\ell &:= \tilde{c}^\ell c^{\ell-1} \end{aligned}$$

It is clear that $\tilde{c}^\ell, c^\ell \in \mathbb{N}^+$. We now formalize the intuition that the values of $\tilde{c}^\ell$ and c^ℓ are only exponential in the description of $\mathcal{A}$.

▶ **Lemma 61.** *For every* $\mathcal{BL}$TrReLU-GNN $\mathcal{A}$ *and* $1 \le \ell \le L$, *for every graph* $\mathcal{G}$ *and vertex* $v \in V$, $c^\ell \xi^\ell(v) \in \left[0, c^\ell\right]^{d_\ell}$.

Proof. We prove the lemma by induction on layers. The base case $\ell = 0$ is straightforward.

For the induction step $1 \le \ell \le L$, for every graph $\mathcal{G}$ and vertex $v \in V$, we define w as follows:

$$w \ := \ C^\ell \xi^{\ell-1}(v) + \sum_{x \in \{\text{out},\text{in}\}} \left(A^\ell_x \sum_{u \in \mathcal{N}_x(v)} \xi^{\ell-1}(u) \right) + b^\ell.$$

It is clear that $\xi^\ell(v) = \mathsf{TrReLU}(w)$. For $1 \le i \le d_\ell$, there are three cases. The first is when $w_i \ge 1$ in which case $\xi^\ell_i(v) = 1$, and thus, $\xi^\ell_i(v) \in \left[0, c^\ell\right]$. The second is when $w_i \le 0$ in which case $\xi^\ell_i(v) = 0$, and thus, $\xi^\ell_i(v) \in \left[0, c^\ell\right]$. The third is when $0 < w_i < 1$ in which case $\xi^\ell(v) = w_i$. Note that

$$c^\ell w_i \ = \ \left(\left(\tilde{c}^\ell C^\ell\right) \left(c^{\ell-1}\xi^{\ell-1}(v)\right) + \sum_{x \in \{\text{out},\text{in}\}} \left(\left(\tilde{c}^\ell A^\ell_x\right) \sum_{u \in \mathcal{N}_x(v)} \left(c^{\ell-1}\xi^{\ell-1}(u)\right) \right) + c^{\ell-1}\left(\tilde{c}^\ell b^\ell\right) \right)_i$$

By the definition of $\tilde{c}$, $\left(\tilde{c}^\ell C^\ell\right)$, $\left(\tilde{c}^\ell A^\ell_c\right)$, and $\left(\tilde{c}^\ell b^\ell\right)$ are matrices over integers. By the induction hypothesis, $\left(c^\ell \xi^{\ell-1}(u)\right)$ are vectors of integers. Hence $c^\ell w_i$ is an integer. Since $0 \le \xi^\ell(v) \le 1$, $0 < c^\ell \xi^\ell(v) < c^\ell$. Thus $c^\ell \xi^\ell(v) \in \left[0, c^\ell\right]^{d_\ell}$. ◀

▶ **Lemma 62.** *For every* $\mathcal{BL}$TrReLU-GNN $\mathcal{A}$ *and* $0 \le \ell \le L$, *every element in* $\mathcal{H}^\ell$ *can be rewritten with a number of bits which is only polynomial in the description of* $\mathcal{A}$.

Proof. Let $\uparrow\mathcal{H}^\ell$ be the following set.

$$\uparrow\mathcal{H}^\ell \ := \ \left\{ \left(c^0\right)^{-1} i \,\middle|\, i \in \left[0, c^0\right]^{d_0} \right\} \times \left\{ \left(c^1\right)^{-1} i \,\middle|\, i \in \left[0, c^1\right]^{d_1} \right\} \times \cdots \times \left\{ \left(c^\ell\right)^{-1} i \,\middle|\, i \in \left[0, c^\ell\right]^{d_\ell} \right\}$$

By Lemma 61, for every $h \in \mathcal{H}^\ell$ and $0 \le i \le \ell$, $c^i h[i] \in \left[0, c^i\right]$. Thus $h \in \uparrow\mathcal{H}^\ell$. Therefore $\mathcal{H}^\ell \subseteq \uparrow\mathcal{H}^\ell$.

Note that the number of bits of c^i is only polynomial in the description of $\mathcal{A}$. Thus, every element in $\uparrow\mathcal{H}^\ell$ requires only polynomially many bits in the description of $\mathcal{A}$. Since $\mathcal{H}^\ell \subseteq \uparrow\mathcal{H}^\ell$, the lemma follows. ◀

C.2 Exponential tree model property

The previous subsection bounded the size of individual values, independent of the input graph. We now show that whenever a GNN is satisfiable, there is *some* satisfying graph that is both reasonably small and nicely-structured. Recall that we are dealing here with directed graphs with some number of node colors. An *n-tree* is an n-graph such that when we remove the direction we have a tree. For a vertex $v \in V$ in a tree model, we say that u is an out-child of v, if u is a child of v in the tree and there exists an edge from v to u.

We can now state the *exponential tree model property*.

▶ **Theorem 63.** *There are constants c_1, c_2 so that, for every n, for every n-$\mathcal{BL}$TrReLU-GNN $\mathcal{A}$, if $\mathcal{A}$ is satisfiable, then there is a n-tree $\mathcal{G}$ with root v_r and height at most L such that*

1. *$\mathcal{A}$ accepts $\langle \mathcal{G}, v_r \rangle$.*
2. *For every $v \in V$, let ℓ be the height of v, if $\ell < L$, then for $x \in \{out, in\}$, the number of different $(L - \ell - 1)$-histories realized by x-children is at most $\alpha_{\mathcal{A}}$, and each $(L - \ell - 1)$-history is realized by at most $\beta_{\mathcal{A}}$ children, where*

$$\alpha_{\mathcal{A}} := c_1 t_{\mathcal{A}} \log (c_2 t_{\mathcal{A}} M_{\mathcal{A}})$$
$$\beta_{\mathcal{A}} := c_1 t_1 (t_{\mathcal{A}} M_{\mathcal{A}})^{c_2 t_{\mathcal{A}}},$$

$t_{\mathcal{A}} := \sum_{1 \le i \le L} d_i$, $M_{\mathcal{A}} := c^L M$, c^L is the L-capacity of $\mathcal{A}$, and M is the maximum numerator in the coefficient matrices and bias vectors of $\mathcal{A}$.

We prove the theorem in two steps. First, we show that if a $\mathcal{BL}$TrReLU-GNN is satisfiable, then it has a tree model. We construct such a tree model by an *unravelling procedure*, a common tool for modal and description logics.

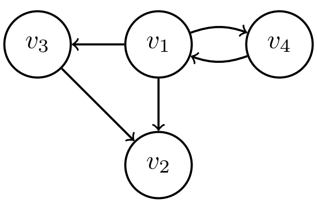
▶ **Definition 64.** *For a n-graph $\mathcal{G}$ and vertices $v_r, v_n \in V$, we will define:*

- *the L-unravelling of $\mathcal{G}$ on v_r, denoted $\mathcal{G}^L_{v_r}$.*
- *the L-unravelling of $\mathcal{G}$ on v_r without outgoing edge to v_n, denoted $\mathcal{G}^L_{v_r \not\to v_n}$.*
- *the L-unravelling of $\mathcal{G}$ on v_r without incoming edge from v_n, denoted $\mathcal{G}^L_{v_r \not\leftarrow v_n}$.*

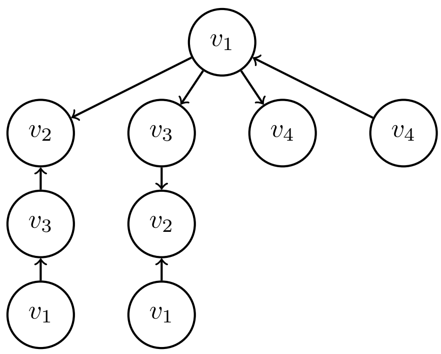
These are the n-graphs defined inductively as follows.

- *For $L = 0$, $\mathcal{G}^0_{v_r}$, $\mathcal{G}^0_{v_r \not\to v_n}$, and $\mathcal{G}^0_{v_r \not\leftarrow v_n}$, are the n-graphs with only one vertex $\tilde{v}_r$. The colors assigned to $\tilde{v}_r$ are the same as those assigned to v_r.*
- *For $L > 0$, $V^L_{v_r}$ is the disjoint union of $\tilde{v}_r$, $V^{L-1}_{v_{out} \not\leftarrow v_r}$, and $V^{L-1}_{v_{in} \not\to v_r}$, where $v_{out} \in \mathcal{N}_{out}(v_r)$ and $v_{in} \in \mathcal{N}_{in}(v_r)$. The colors holding of $\tilde{v}_r$ are those assigned to v_r. For every $v_{out} \in \mathcal{N}_{out}(v_r)$, there is a edge from $\tilde{v}_r$ to the root of $\mathcal{G}^{L-1}_{v_{out} \not\leftarrow v_r}$. For every $v_{in} \in \mathcal{N}_{in}(v_r)$, there is a edge from the root of $\mathcal{G}^{L-1}_{v_{in} \not\to v_r}$ to $\tilde{v}_r$.*
  The n-graph $\mathcal{G}^L_{v_r \not\to v_n}$ is defined analogously except that $\mathcal{N}_{out}(v_r)$ is replaced by $\mathcal{N}_{out}(v_r) \setminus \{v_n\}$.
  The n-graph $\mathcal{G}^L_{v_r \not\leftarrow v_n}$ is defined analogously except that $\mathcal{N}_{in}(v_r)$ is replaced by $\mathcal{N}_{in}(v_r) \setminus \{v_n\}$.

It is obvious that the depth of an L-unravelling tree is at most L.



(a) $\mathcal{G}$



(b) $\mathcal{G}^4_{v_1}$

Figure 1 Example of a 4-unravelling of $\mathcal{G}$ on v_1.

▶ **Lemma 65.** *For every* $\mathcal{BL}$TrReLU-GNN $\mathcal{A}$*, for every* n*-graph* $\mathcal{G}$ *and vertex* $v_r \in V$*,* $\mathsf{hist}^L\left(\tilde{v}_r, \mathcal{G}^L_{v_r}\right) = \mathsf{hist}^L(v_r, \mathcal{G})$*.*

Proof. We will prove the following stronger properties:

There exists a mapping $\sigma : V^L_{v_r} \to V$ such that

1. $\sigma(\tilde{v}_r) = v_r$.
2. For every $\tilde{v} \in V^L_{v_r}$, the colors assigned to $\tilde{v}$ in $\mathcal{G}^L_{v_r}$ are the same as the colors holding of $\sigma(\tilde{v})$ in $\mathcal{G}$.
3. For every $\tilde{v} \in V^L_{v_r}$, if the height of $\tilde{v}$ is less than L, then the restriction of σ to $\mathcal{N}_{\mathrm{out},\mathcal{G}^L_{v_r}}(\tilde{v})$ is a bijection between $\mathcal{N}_{\mathrm{out},\mathcal{G}^L_{v_r}}(\tilde{v})$ and $\mathcal{N}_{\mathrm{out},\mathcal{G}}(\sigma(\tilde{v}))$. The restriction of σ to $\mathcal{N}_{\mathrm{in},\mathcal{G}^L_{v_r}}(\tilde{v})$ is a bijection between $\mathcal{N}_{\mathrm{in},\mathcal{G}^L_{v_r}}(\tilde{v})$ and $\mathcal{N}_{\mathrm{in},\mathcal{G}}(\sigma(\tilde{v}))$.
4. For $0 \le \ell \le L$, for every $\tilde{v} \in V^L_{v_r}$, if the depth of $\tilde{v}$ is less than $L-\ell$, then $\mathsf{hist}^\ell\left(\tilde{v}, \mathcal{G}^L_{v_r}\right) = \mathsf{hist}^\ell(\sigma(\tilde{v}), \mathcal{G})$.

Note that the depth of the root vertex $\tilde{v}_r$ is 0, the lemma follows the first and fourth properties directly.

We define the mapping σ as follows. For every $\tilde{v} \in V^L_{v_r}$, $\tilde{v}$ is the root of a subtree with one of the following forms: $\mathcal{G}^\ell_v$, $\mathcal{G}^\ell_{v\not\leftarrow v'}$, or $\mathcal{G}^\ell_{v\not\rightarrow v'}$. Then $\sigma(\tilde{v}) := v$.

First of all, because $\tilde{v}_r$ is the root of $\mathcal{G}^\ell_{v_r}$, by the definition of σ, $\sigma(\tilde{v}_r) = v_r$. For every $\tilde{v} \in V^L_{v_r}$, $\tilde{v}$ is the root of a subtree with one of the following forms: $\mathcal{G}^\ell_v$, $\mathcal{G}^\ell_{v\not\leftarrow v'}$, or $\mathcal{G}^\ell_{v\not\rightarrow v'}$. By the definition of σ, $\sigma(\tilde{v}) = v$. Moreover, by the construction of the subtree, the colors holding of $\tilde{v}$ in $\mathcal{G}^L_{v_r}$ are the same as the colors assigned to v in $\mathcal{G}$. Therefore the second property holds.

Next, we prove the third property by considering the construction of subtrees.

- If $\tilde{v} = \tilde{v}_r$, then it is the root of the subtree $\mathcal{G}^\ell_{v'}$. $\mathcal{N}_{\mathrm{out}}\left(\tilde{v}, \mathcal{G}^\ell_{v'}\right)$ is the set of roots $\tilde{v}_{\mathrm{out}}$ of the subtree $\mathcal{G}^{\ell-1}_{v_{\mathrm{out}}\not\leftarrow v'}$, where $v_{\mathrm{out}} \in \mathcal{N}_{\mathrm{out}}(v', \mathcal{G})$. Since $\sigma(\tilde{v}_{\mathrm{out}}) = v_{\mathrm{out}}$, the property holds. We can treat in-neighbors analogously.
- If $\tilde{v}$ is the root of the subtree $\mathcal{G}^\ell_{v\not\rightarrow v_p}$, then there exists parent vertex $\tilde{v}_p$ such that $\sigma(\tilde{v}_p) = v_p$, there exists a edge from v to v_p, and there exists a edge from $\tilde{v}$ to $\tilde{v}_p$. Note that the neighbors of $\tilde{v}$ in $\mathcal{G}^\ell_{v'}$ are its parent vertex $\tilde{v}_p$ along with roots $\tilde{v}_{\mathrm{out}}$ of the subtree $\mathcal{G}^{\ell-1}_{v_{\mathrm{out}}\not\leftarrow v'}$, where $v_{\mathrm{out}} \in \mathcal{N}_{\mathrm{out}}(v', \mathcal{G}) \setminus \{v_p\}$. Since $\sigma(\tilde{v}_{\mathrm{out}}) = v_{\mathrm{out}}$, the property holds. We can treat in-neighbors analogously.
- If $\tilde{v}$ is the root of the subtree $\mathcal{G}^\ell_{v\not\leftarrow v_p}$, We can treat it analogously as previous case.

Finally, we prove the fourth property by induction on ℓ. For the base case $\ell = 0$, since the 0-history only depends on the colors assigned to the vertex. The base case follows the second property. For the inductive step $1 \le \ell \le L$, for every vertex $\tilde{v} \in V^L_{v_r}$, if the height of $\tilde{v}$ is less than $L-\ell$, because an out-neighbor of $\tilde{v}$ is either the parent of $\tilde{v}$ or a child of $\tilde{v}$, its height is less than $L-(\ell-1)$. By the induction hypothesis, for every $\tilde{u} \in \mathcal{N}_{\mathrm{out}}\left(\tilde{v}, \mathcal{G}^L_{v_r}\right)$, $\mathsf{hist}^{\ell-1}\left(\tilde{u}, \mathcal{G}^L_{v_r}\right) = \mathsf{hist}^{\ell-1}(\sigma(\tilde{u}), \mathcal{G})$. by the third property, there exist a bijection from the out-neighbors of $\tilde{v}$ in $\mathcal{G}^L_{v_r}$ to the out-neighbors of $\sigma(\tilde{v})$ in $\mathcal{G}$. Therefore

$$\begin{aligned}\sum_{\tilde{u}\in\mathcal{N}_{\mathrm{out}}\left(\tilde{v},\mathcal{G}^L_{v_r}\right)} \mathsf{hist}^{\ell-1}\left(\tilde{u}, \mathcal{G}^L_{v_r}\right) &= \sum_{\tilde{u}\in\mathcal{N}_{\mathrm{out}}\left(\tilde{v},\mathcal{G}^L_{v_r}\right)} \mathsf{hist}^{\ell-1}\left(\sigma(\tilde{u}), \mathcal{G}\right) \\ &= \sum_{u\in\mathcal{N}_{\mathrm{out}}(\sigma(\tilde{v}),\mathcal{G})} \mathsf{hist}^{\ell-1}(u, \mathcal{G})\end{aligned}$$

We can treat in-neighbors analogously and obtain:

$$\sum_{\tilde{u}\in\mathcal{N}_{\mathrm{in}}\left(\tilde{v},\mathcal{G}^L_{v_r}\right)} \mathsf{hist}^{\ell-1}\left(\tilde{u}, \mathcal{G}^L_{v_r}\right) = \sum_{u\in\mathcal{N}_{\mathrm{in}}(\sigma(\tilde{v}),\mathcal{G})} \mathsf{hist}^{\ell-1}(u, \mathcal{G})$$

By Lemma 52, the value of ℓ-history of $\tilde{v}$ only depends on the colors of $\tilde{v}$, the summation of $(\ell-1)$-history of out-neighbors, and the summation of $(\ell-1)$-history of in-neighbors. Hence $\mathsf{hist}^{\ell}\left(\tilde{v}, \mathcal{G}_{v_r}^{L}\right) = \mathsf{hist}^{\ell}\left(\sigma\left(\tilde{v}\right), \mathcal{G}\right)$. ◀

Next, we show that for every tree model, we can apply some surgery and obtain another tree model whose size is bounded.

We will need some more terminology. For every tree $\mathcal{G}$ and vertex $v \in V$ with depth $\ell < L$, we let $\mathcal{NC}_{\mathrm{out},\mathcal{G}}(v)$ be the set of out-children of v and $\mathcal{HC}_{\mathrm{out},\mathcal{G}}(v) := \left\{\mathsf{hist}^{L-\ell-1}(u) \middle| \; u \in \mathcal{NC}_{\mathrm{out},\mathcal{G}}(v)\right\}$ be the set of $(L-\ell-1)$-histories that are realized by the out-children of v. We define in-child, $\mathcal{NC}_{\mathrm{in},\mathcal{G}}(v)$, $\mathcal{HC}_{\mathrm{in},\mathcal{G}}(v)$ analogously. When the graph $\mathcal{G}$ is clear from the context, we omit it and simply write $\mathcal{NC}_{\mathrm{out}}(v)$, $\mathcal{HC}_{\mathrm{out}}(v)$, $\mathcal{NC}_{\mathrm{in}}(v)$, and $\mathcal{HC}_{\mathrm{in}}(v)$.

For every tree $\mathcal{G}$ and vertex $v \in V$ with depth $\ell < L$, the *characteristic equation system of* v, denoted by $\mathcal{Q}_v$, is a linear equation system with variables $\{z_{x,h}\}_{\substack{x\in\{\mathrm{out},\mathrm{in}\}\\ h\in\mathcal{HC}_x}}$ defined as follows.

- Suppose v has a parent v_p and there exists an edge from v to v_p. For $1 \le i \le L-\ell$, let

  $$w^i(z_{x,h}) \;:=\; C^i\xi^{i-1}(v) + A^i_{\mathrm{out}}\xi^{i-1}(v_p) + \sum_{x\in\{\mathrm{out},\mathrm{in}\}} A^i_x \sum_{h\in\mathcal{HC}_x(v)} z_{x,h}h[i-1] + b^i.$$

  For $1 \le j \le d_i$,

  $$c^i\left(w^i(z_{x,h})\right)_j \;\circledast_{i,j}\; c^i\xi^i_j(v)$$

  is an equation in $\mathcal{Q}_v$, where c^i is the i-capacity of $\mathcal{A}$ and $\circledast_{i,j}$ is $\ge$ if $\xi^i_j(v) = 1$; $\le$ if $\xi^i_j(v) = 0$; $=$ if $0 < \xi^i_j(v) < 1$.

  Note that the coefficients in $\mathcal{Q}_v$ are all integers since we multiply c^i on both sides. The maximum coefficient in $\mathcal{Q}_v$ is $4c^{L-\ell}M \le 4c^L M = M_{\mathcal{A}}$, where M is the maximum of the numerators for all numbers in the coefficient matrices and bias vectors of $\mathcal{A}$. There are $\sum_{1\le i\le L-\ell} d_i \le \sum_{1\le i\le L} d_i = t_{\mathcal{A}}$ equations in $\mathcal{Q}_v$.

  We can define $\mathcal{Q}_v$ for the other two cases analogously:
- Suppose v has a parent v_p and there exists an edge from v_p to v. We define $\mathcal{Q}_v$ by replacing $A^i_{\mathrm{out}}\xi^{i-1}(v_p)$ with $A^i_{\mathrm{in}}\xi^{i-1}(v_p)$.
- Suppose v has no parent. We definite $\mathcal{Q}_v$ by removing $A^i_{\mathrm{out}}\xi^{i-1}(v_p)$.

▶ **Lemma 66.** *For every tree $\mathcal{G}$ and vertex $v \in V$ with depth $\ell < L$, $\mathcal{Q}_v$ is solvable in $\mathbb{N}$.*

Proof. For $x \in \{\mathrm{out}, \mathrm{in}\}$, let $n_{x,h}$ be the number of v's x-children whose $(L-\ell-1)$-history is h. We claim that $\{z_{x,h} \leftarrow n_{x,h}\}_{\substack{x\in\{\mathrm{out},\mathrm{in}\}\\ h\in\mathcal{HC}_x}}$ is a solution of $\mathcal{Q}_v$.

Suppose v has a parent v_p and there exists an edge from v to v_p. We can treat the other two cases analogously. Note that for $1 \le i \le \ell$,

$$\begin{aligned}
\sum_{u\in\mathcal{N}_{\mathrm{out}}(v)} \xi^{i-1}(u) &= \xi^{i-1}(v_p) + \sum_{u\in\mathcal{NC}_{\mathrm{out}}(v)} \xi^{i-1}(u) \\
&= \xi^{i-1}(v_p) + \sum_{u\in\mathcal{NC}_{\mathrm{out}}(v)} \left(\mathsf{hist}^{L-\ell-1}(u)\right)[i-1] \\
&= \xi^{i-1}(v_p) + \sum_{h\in\mathcal{HC}_{\mathrm{out}}(v)} n_{\mathrm{out},h}h[i-1] \\
\sum_{u\in\mathcal{N}_{\mathrm{in}}(v)} \xi^{i-1}(u) &= \sum_{h\in\mathcal{HC}_{\mathrm{in}}(v)} n_{\mathrm{in},h}h[i-1].
\end{aligned}$$

Then we can rewrite $w^i(n_{x,h})$ as follows:

$$w^i(n_{x,h}) \;=\; C^i\xi^{i-1}(v) + \sum_{u\in\mathcal{N}_{\text{out}}(v)} \xi^{i-1}(u) + \sum_{u\in\mathcal{N}_{\text{in}}(v)} \xi^{i-1}(u) + b^i$$

By the definition of feature vectors, $\xi^i_j(v) = \mathsf{TrReLU}\left(\left(w^i(n_{x,h})\right)_j\right)$. By the definition of truncated ReLU, we can verify that $\{z_{x,h} \leftarrow n_{x,h}\}_{\substack{x\in\{\text{out},\text{in}\}\\ h\in\mathcal{HC}_x}}$ is a solution of $\mathcal{Q}_v$. ◀

For every tree $\mathcal{G}$ and vertex $v \in V$ with depth $\ell < L$, a *simple operation* of $\mathcal{G}$ on v is one of the following operations.

- Delete a subtree induced by u, where u is a child of v.
- Copy a subtree induced by u and add an edge from v to the root of the copy, where u is a out-child of v.
- Copy a subtree induced by u and add an edge from the root of the copy to v, where u is a in-child of v.

A *simple transformation* of $\mathcal{G}$ on v is an operation of $\mathcal{G}$ by applying finitely many simple operations.

▶ **Lemma 67.** *For every tree $\mathcal{G}$ and vertex $v \in V$ with depth $\ell < L$, let $\mathcal{G}'$ be the tree obtained by applying a simple transformation on v. For $x \in \{out, in\}$, let $n'_{x,h}$ be the number of v's x-children in $\mathcal{G}'$ whose $(L-\ell-1)$-history is h. If $\left\{z_{x,h} \leftarrow n'_{x,h}\right\}_{\substack{x\in\{out,in\}\\ h\in\mathcal{HC}_x}}$ is a solution of $\mathcal{Q}_v$, then for every $u \in V$ with depth $\ell_u \le \ell$, $h^{L-\ell_u}_{\mathcal{G}'}(u) = h^{L-\ell_u}_{\mathcal{G}}(u)$.*

Proof. It is sufficient to show that $h^{L-\ell}_{\mathcal{G}'}(v) = h^{L-\ell}_{\mathcal{G}}(v)$.

Suppose v has a parent v_p and there exists an edge from v to v_p. We can treat the other two cases analogously. Note that $\mathcal{HC}_{x,\mathcal{G}}(v) \subseteq \mathcal{HC}_{x,\mathcal{G}}(v)$. For $1 \le i \le \ell$,

$$\begin{aligned}
\sum_{u\in\mathcal{N}_{\text{out},\mathcal{G}'}(v)} \xi^{i-1}_{\mathcal{G}'}(u) &= \xi^{i-1}(v_p) + \sum_{u\in\mathcal{NC}_{\text{out},\mathcal{G}'}(v)} \xi^{i-1}_{\mathcal{G}'}(u) \\
&= \xi^{i-1}_{\mathcal{G}}(v_p) + \sum_{u\in\mathcal{NC}_{\text{out},\mathcal{G}'}(v)} \left(\mathsf{hist}^{L-\ell-1}(u)\right)[i-1] \\
&= \xi^{i-1}_{\mathcal{G}'}(v_p) + \sum_{h\in\mathcal{HC}_{\text{out},\mathcal{G}}(v)} n'_{\text{out},h}h[i-1] \\
\sum_{u\in\mathcal{N}_{\text{in},\mathcal{G}'}(v)} \xi^{i-1}(u) &= \sum_{h\in\mathcal{HC}_{\text{in},\mathcal{G}}(v)} n'_{\text{in},h}h[i-1].
\end{aligned}$$

Then we can rewrite $w^i(n'_{x,h})$ as follows:

$$w^i(n'_{x,h}) \;=\; C^i\xi^{i-1}_{\mathcal{G}'}(v) + \sum_{u\in\mathcal{N}_{\text{out}}(v)} \xi^{i-1}(u) + \sum_{u\in\mathcal{N}_{\text{in}}(v)} \xi^{i-1}(u) + b^i$$

By the definition of feature vectors, $\xi^i_{\mathcal{G}',j}(v) = \mathsf{TrReLU}\left(\left(w^i(n'_{x,h})\right)_j\right)$. On the other hand, since $\{z_{x,h} \leftarrow n_{x,h}\}_{\substack{x\in\{\text{out},\text{in}\}\\ h\in\mathcal{HC}_x}}$ is a solution of $\mathcal{Q}_v$, $\left(w^i(n'_{x,h}\right)_j \circledast \xi^i_{\mathcal{G},j}(v)$.

- If $\xi^i_{\mathcal{G},j}(v) = 1$, then $\left(w^i(n'_{x,h})\right)_j > 1$. Thus $\xi^i_{\mathcal{G}',j}(v) = \mathsf{TrReLU}(1) = 1$.
- If $\xi^i_{\mathcal{G},j}(v) = 0$, then $\left(w^i(n'_{x,h})\right)_j < 0$. Thus $\xi^i_{\mathcal{G}',j}(v) = \mathsf{TrReLU}(0) = 0$.

- If $0 < \xi^i_{\mathcal{G},j}(v) < 1$, then $\left(w^i(n'_{x,h})\right)_j = \xi^i_{\mathcal{G},j}(v)$. Thus $\xi^i_{\mathcal{G}',j}(v) = \mathsf{TrReLU}\left(\xi^i_{\mathcal{G},j}(v)\right) = \xi^i_{\mathcal{G},j}(v)$.

Thus $h^{L-\ell}_{\mathcal{G}'}(v) = h^{L-\ell}_{\mathcal{G}}(v)$. ◀

To reduce the number of children in the tree, we will use the following lemma, which stems from [6, 16], see also [14, Corollary 2.2].

▶ **Lemma 68.** *There are constants $c_1, c_2 \in \mathbb{N}$ such that for every system Q of linear constraints, if $\mathcal{Q}$ admits a solution in $\mathbb{N}$, then it admits a solution in $\mathbb{N}$ in which the number of variables assigned with non-zero values is at most $c_1 t \log(c_2 t M)$ and every variable is assigned with a value at most $c_1 t (tM)^{c_2 t}$, where $t = |\mathcal{Q}|$ and M is the maximal constant in $\mathcal{Q}$.*

Finally, we are ready to prove Theorem 63.

Proof. For every satisfiable n-$\mathcal{BL}$TrReLU-GNN $\mathcal{A}$, by Lemma 65, $\mathcal{A}$ has a n-tree model $\mathcal{G}$ with root v_r and height at most L such that $\mathcal{A}$ accepts $\langle \mathcal{G}, v_r \rangle$.

For every vertex $u \in V$, we say that u is a *bed vertex* if u does not satisfy the second property of the theorem. If there is no bad vertex, then $\mathcal{G}$ is a desired tree of the theorem. Otherwise, let v be the bad vertex with maximum depth. Note that for every u in the subtree induced by v, u satisfies the second property of the theorem.

By Lemma 66, $\mathcal{Q}_v$ has a solution in $\mathbb{N}$. By Lemma 68, $\mathcal{Q}_v$ has a *small* solution $\left\{ z_{x,h} \leftarrow n'_{x,h} \right\}_{\substack{x \in \{\text{out},\text{in}\} \\ h \in \mathcal{HC}_x}}$ satisfying that the number of variables assigned with non-zero values is at most $c_1 t_{\mathcal{A}} \log(c_2 t_{\mathcal{A}} M_{\mathcal{A}})$, and every variable is assigned to a value bounded by $c_1 t (t_{\mathcal{A}} M_{\mathcal{A}})^{c_2 t_{\mathcal{A}}}$.

Let $n_{x,h}$ be the number of x-children of v in $\mathcal{G}$ whose $(L - \ell - 1)$-history is h. Let $\mathcal{G}'$ be the n-tree obtained by applying the following simple transformation of $\mathcal{G}$ on v. For every $x \in \{\text{in}, \text{out}\}$ and $h \in \mathcal{HC}_x$, if $n'_{x,h} < n_{x,h}$, then we choose $(n_{x,h} - n'_{x,h})$ x-children u of v whose $(L - \ell - 1)$-history is h, and remove the subtree induced by each u. If $n'_{x,h} > n_{x,h}$, we choose an x-child u of v whose $(L - \ell - 1)$-history is h, and duplicate $(n'_{x,h} - n_{x,h})$ copies of the subtree induced by u. If x is out, then there exists an edge from v to the root of the duplicated subtrees. If x is in, then there exists an edge from the root of the duplicated subtrees to v.

It is clear that the number of x-children of v in $\mathcal{G}'$ whose $(L - \ell - 1)$-history is h is $n'_{x,h}$. Thus v satisfies the second property in $\mathcal{G}'$, which implies that v is not a bad vertex in $\mathcal{G}'$. By Lemma 67, $\xi^L_{\mathcal{G}'}(v_r) = \xi^L_{\mathcal{G}}(v_r)$. Since $\mathcal{A}$ accepts $\langle \mathcal{G}, v_r \rangle$, $\mathcal{A}$ also accepts $\langle \mathcal{G}', v_r \rangle$.

Finally, note that v is the bad vertex with maximum depth in $\mathcal{G}$, there is no bad vertex in the subtree induced by v in $\mathcal{G}'$. The number of bad vertices *decrease by 1* after the above simple transformation. Since $\mathcal{G}$ is a finite graph, the number of bad vertices in $\mathcal{G}$ is finite. We can repeatly apply the procedure until there are no more bad vertices and obtain the desired tree. ◀

C.3 PSPACE-completeness for the satisfiability problem of $\mathcal{BL}$TrReLU-GNNs

Now we are ready to prove Theorem 25. Intuitively, in Theorem 63, we showed that every satisfiable $\mathcal{BL}$TrReLU-GNN has an exponential size tree model. The theorem did not say anything about the size of the numbers in features. But Lemma 62 tells us that for every graph the size of computed features is not very large.

However, since the size of the model may be exponential, the naïve algorithm, which guesses the whole model and checks it, takes nondeterministic exponential time. However, we can reduce to PSPACE using the same approach as in the PSPACE bound for modal logic [13]: in order to check the validity of the tree model, it is sufficient to check the validity of each vertex locally. That is, we only need to compute the history from its parent and children and check this value. Thus the following Algorithm 1 decides the satisfiability problem of $\mathcal{BL}$TrReLU-GNNs by guessing children of a vertex and checking at one time. Though the number of children may be exponential, most of them share the same history. The algorithm guesses children by guessing polynomial many different histories and polynomial many exponential numbers, the sizes of which are only polynomial. Thus Algorithm 1 is a nondeterministic polynomial space algorithm. The correctness follows Theorem 63 and Lemma 62.

The lower bound is established by embedding the description logic $\mathcal{ALC}$ into $\mathcal{L}$-MP^2. Since the concept satisfiability problem of $\mathcal{ALC}$ with one role is PSPACE-hard [18], it will follow from the embedding that the finite satisfiability problem of $\mathcal{L}$-MP^2 is also PSPACE-hard. Since the reduction from $\mathcal{L}$-MP^2 to $\mathcal{BL}$TrReLU-GNN mentioned in Theorem 23 is polynomial, the satisfiability problem of $\mathcal{BL}$TrReLU-GNN is also PSPACE-hard.

▶ **Lemma 69.** *There exists a polynomial time translation π_x from $\mathcal{ALC}$ concepts with one role R to $\mathcal{L}$-MP^2 formulas such that the $\mathcal{ALC}$ concept C is satisfiable if and only if $\pi_x(C)$ is finitely satisfiable.*

Proof. We will define π_x and π_y, which are the standard translation from $\mathcal{ALC}$ concepts to first-order logic formulas, except with some slight modification on quantifiers to fit our logic. It is routine to check that the $\mathcal{ALC}$ concept C is satisfiable if and only if $\pi_x(C)$ is finitely satisfiable.

$$\begin{aligned}
\pi_x(A) &= A(x) & \pi_y(A) &= A(y)\\
\pi_x(\neg C) &= \neg\pi_x(C) & \pi_y(\neg C) &= \neg\pi_y(C)\\
\pi_x(C\sqcap D) &= \pi_x(C)\wedge\pi_x(D) & \pi_y(C\sqcap D) &= \pi_y(C)\wedge\pi_y(D)\\
\pi_x(C\sqcup D) &= \pi_x(C)\vee\pi_x(D) & \pi_y(C\sqcup D) &= \pi_y(C)\vee\pi_y(D)\\
\pi_x(\exists R.C) &= \#_y[E(x,y)\wedge\pi_y(C)]\geq 1 & \pi_y(\exists R.C) &= \#_x[E(y,x)\wedge\pi_x(C)]\geq 1\\
\pi_x(\forall R.C) &= \#_y[E(x,y)\wedge\neg\pi_y(C)]=0 & \pi_y(\forall R.C) &= \#_y[E(y,x)\wedge\neg\pi_x(C)]=0
\end{aligned}$$

◀

C.4 NP-completeness for the satisfiability problem of fixed layer $\mathcal{BL}$TrReLU-GNNs

Let us consider the run time of Algorithm 1. The check procedure calls itself $2\alpha_{\mathcal{A}}$ times. The depth of recursion is L. It takes only polynomial time to compute the history from the parent and children. Hence the runtime of the algorithm is proportional to

$$1+2\alpha_{\mathcal{A}}+(2\alpha_{\mathcal{A}})^2+\cdots+(2\alpha_{\mathcal{A}})^L \;=\; \frac{\alpha_{\mathcal{A}}^{L+1}-1}{\alpha_{\mathcal{A}}-1}.$$

Note that $\alpha_{\mathcal{A}}$ is polynomial in the length of the description of $\mathcal{A}$. Thus for fixed layer $\mathcal{BL}$TrReLU-GNN $\mathcal{A}$, Algorithm 1 only takes nondeterministic polynomial time.

We show that when the number of layers is fixed, the satisfiability problem of $\mathcal{BL}$TrReLU-GNN is NP-hard by reducing 3-SAT to it. Since 3-SAT is NP-hard, so is the satisfiability problem of fixed layer $\mathcal{BL}$TrReLU-GNN.

Algorithm 1 Algorithm for the satisfiability problem of $\mathcal{BL}$TrReLU-GNNs

procedure SAT($\mathcal{A}$) ▷ $\mathcal{BL}$TrReLU-GNN $\mathcal{A}$
    Guess a over-approximated L-history h_r
    if Check($\mathcal{A}$, L, out, 0, h_r) Reject **then**
        Reject
    end if
    if $(h_r[L])_1 \geq 0.5$ **then**
        Accept
    else
        Reject
    end if
end procedure
procedure CHECK($\mathcal{A}$, ℓ, x, h_p, h) ▷ $\mathcal{BL}$TrReLU-GNN $\mathcal{A}$, height $0 \leq \ell \leq L$,
▷ direction $x \in \{\text{out}, \text{in}\}$, parent's $(\ell+1)$-history h_p, ℓ-history h
    if $\ell = 0$ **then**
        Accept
    else
        Guess e from $\{0,1\}^{d_0}$
        Guess $\alpha_{\mathcal{A}}$ over-approximated $(\ell-1)$-histories $h_{\text{out},i}$ and $h_{\text{in},i}$ from $\uparrow\mathcal{H}^{\ell-1}$
        Guess $\alpha_{\mathcal{A}}$ numbers $n_{\text{out},i}$ and $n_{\text{in},i}$ from $[0\,,\beta_{\mathcal{A}}]$
        for $1 \leq i \leq \alpha_{\mathcal{A}}$ **do**
            if Check($\mathcal{A}$, $\ell-1$, out, h, $h_{\text{out},i}$) Reject **then**
                Reject
            end if
            if Check($\mathcal{A}$, $\ell-1$, in, h, $h_{\text{in},i}$) Reject **then**
                Reject
            end if
        end for
        Compute h' from e, x, h_p, $h_{\text{out},i}$, $h_{\text{in},i}$, $n_{\text{out},i}$, $n_{\text{in},i}$
        if $h = h'$ **then**
            Accept
        else
            Reject
        end if
    end if
end procedure

▶ **Lemma 70.** *There exists a polynomial time reduction from 3-CNF formulas φ to 2-layer $\mathcal{BL}$TrReLU-GNN $\mathcal{A}_\varphi$ such that φ is satisfiable if and only if $\mathcal{A}_\varphi$ is satisfiable.*

Proof. Let φ be a 3-CNF formula with n variables and m clauses.

$$\varphi \;:=\; (\ell_{11} \vee \ell_{12} \vee \ell_{13}) \wedge (\ell_{21} \vee \ell_{22} \vee \ell_{23}) \wedge \cdots \wedge (\ell_{m1} \vee \ell_{m2} \vee \ell_{m3})$$

We define the 2-layer $\mathcal{BL}$TrReLU-GNN $\mathcal{A}_\varphi$ as follows. The input dimensions are $d_0 := n$, $d_1 := m$, and $d_2 := 1$. The coefficient matrix A^1_{out}, A^1_{in} A^2_{out}, and A^2_{in} are zero matrices.

For $1 \leq i \leq m$, for $1 \leq j \leq 3$, if $\ell_{ij} = x_s$, then $C^1_{i,s} = 1$; otherwise, if $\ell_{ij} = \neg x_s$, then $C^1_{i,s} = -1$. All other numbers in C^1 are 0. b^1_i is the number of negative literals in the i^{th} clause. For $1 \leq i \leq m$, $C^2_{1,i} = 1$. $b^2_1 = 1 - m$.

For every literal ℓ,

$$g(\ell) \;:=\; \begin{cases} \xi^0_s(v), & \text{if } \ell = x_s \\ 1 - \xi^0_s(v), & \text{if } \ell = \neg x_s \end{cases}$$

It is not difficult to check that for $1 \leq i \leq m$,

$$\xi^1_i(v) = \mathsf{TrReLU}\left(f(\ell_{i1}) + f(\ell_{i2}) + f(\ell_{i3})\right)$$

Thus $\xi^1_i(v) = 1$ if $\left\{x_s \leftarrow \xi^0_s(v)\right\}_{1 \leq s \leq n}$ is a valid assignment for the m^{th} clause.

Finally, since

$$\xi^2_1(v) = \mathsf{TrReLU}\left(\sum_{1 \leq i \leq m} \xi^1_i(v) + (1 - m)\right),$$

$\xi^2_1(v) = 1$ if and only if $\xi^1_i(v)$ are all 1. Otherwise, $\xi^2_1(v) = 0$. Therefore, $\xi^2_1(v) = 1$ if and only if $\xi^1_i(v) = 1$ if $\left\{x_s \leftarrow \xi^0_s(v)\right\}_{1 \leq s \leq n}$ is a valid assignment for all clauses, that is a valid assignment of φ. Thus φ is satisfiable if and only if $\mathcal{A}_\varphi$ is satisfiable. ◀

D Proofs from Subsection 3.2: Undecidability of MP^2, and of GNNs with truncated ReLU and global readout

Subsection 3.2 dealt with GNNs that have eventually constant activations but allow global aggregation. The main results are undecidability theorems, contrasting with the case of local aggregation.

D.1 Proof of Lemma 31: encoding simple equation systems with MP^2 formula

We recall the lemma:

▶ **Lemma 31.** *For every simple equation system ε with n variables and m equations, there exists an $(n+m)$- MP^2 formula $\Psi_\varepsilon(x)$ such that ε has a solution in $\mathbb{N}$ if and only if $\Psi_\varepsilon(x)$ is finitely satisfiable.*

Proof. We construct the formula $\Psi_\varepsilon(x)$ as follows. The vocabulary of $\Psi_\varepsilon(x)$ consists of unary predicates P_i and U_j, where $1 \leq i \leq m$ and $1 \leq j \leq n$. For $1 \leq i \leq m$, we define $\varphi_i(x)$ depending on the i^{th} equation in ε.

- If the equation is $v_j = 1$, then $\varphi_i(x) := (\#_y[P_i(y) \wedge U_j(y)] = 1)$.

- If the equation is $v_{j_1} = v_{j_2} + v_{j_3}$, then $\varphi_i(x) := (\#_y[\psi_i(y)] - \#_y[\top] = 0)$ where

$$\begin{aligned}\psi_i(y) \;:=\; & (P_i(y) \wedge (U_{j_2}(y) \vee U_{j_3}(y)) \rightarrow (\#_x[E(y,x) \wedge P_i(x) \wedge U_{j_1}(x)] = 1)) \;\wedge \\ & (P_i(y) \wedge U_{j_1}(y) \rightarrow (\#_x[E(x,y) \wedge P_i(x) \wedge (U_{j_2}(x) \vee U_{j_3}(x))] = 1)).\end{aligned}$$

- If the equation is $v_{j_1} = v_{j_2} \cdot v_{j_3}$, then $\varphi_i(x) := (\#_y[\psi_i(y)] - \#_y[\top] = 0)$ where

$$\begin{aligned}\psi_i(y) \;:=\; & (P_i(y) \wedge U_{j_2}(y) \rightarrow (\#_x[E(y,x) \wedge P_i(x) \wedge U_{j_1}(x)] - \#_x[P_i(x) \wedge U_{j_3}(x)] = 0)) \;\wedge \\ & (P_i(y) \wedge U_{j_1}(y) \rightarrow (\#_x[E(x,y) \wedge P_i(x) \wedge U_{j_2}(x)] = 1)).\end{aligned}$$

We now define $\Psi_\varepsilon(x)$:

$$\begin{aligned}\psi^{disj}(x) \;&:=\; \bigwedge_{1 \le i_1 < i_2 \le m} (\#_y[P_{i_1}(y) \wedge P_{i_2}(y)] = 0) \;\wedge \bigwedge_{1 \le j_1 < j_2 \le n} (\#_y[U_{j_1}(y) \wedge U_{j_1}(y)] = 0) \\ \psi^{eq}(x) \;&:=\; \bigwedge_{\substack{1 \le i_1 < i_2 \le m \\ 1 \le j \le n}} (\#_y[P_{i_1}(y) \wedge U_j(y)] - \#_y[P_{i_2}(y) \wedge U_j(y)] = 0) \\ \Psi_\varepsilon(x) \;&:=\; \psi^{disj}(x) \wedge \psi^{eq}(x) \wedge \bigwedge_{1 \le i \le m} \varphi_i(x).\end{aligned}$$

Suppose $\Psi_\varepsilon(x)$ is finitely satisfiable with a finite model $\mathcal{G}$ and substitution x/v. First of all, let $V_{i,j} := \{u \in V |\ \mathcal{G} \models P_i(u) \wedge U_j(u)\}$. Because $\mathcal{G} \models \psi^{disj}(v)$, each vertex in $\mathcal{G}$ realizes at most one P_i and one U_j. Therefore $V_{i,j}$ are disjoint. Next, because $\mathcal{G} \models \psi^{eq}(v)$, for every $1 \le i_1 < i_2 \le m$ and $1 \le j \le n$, $|V_{i_1,j}| = |V_{i_2,j}|$. Finally, we claim that $\{v_j \leftarrow |V_{1,j}|\}_{1 \le j \le n}$ is a solution of ε. We show that $\mathcal{G} \models \varphi_i(v)$ implies that $\{v_j \leftarrow |V_{1,j}|\}_{1 \le j \le n}$ is a solution of the i^{th} equation in ε.

- If the equation is $v_j = 1$, then $\mathcal{G} \models (\#_y[P_i(y) \wedge U_j(y)] = 1)$, which implies that $|V_{i,j}| = 1$. Hence $|V_{1,j}| = |V_{i,j}| = 1$.
- If the equation is $v_{j_1} = v_{j_2} + v_{j_3}$, then $\mathcal{G} \models (\#_y[\psi_i(y)] - \#_y[\top] = 0)$, which implies that $|\{u \in V |\ \mathcal{G} \models \psi_i(u)\}| - |V| = 0$. Hence for each $u \in V$, $\mathcal{G} \models \psi_i(u)$.
  We will argue that the edges between $V_{i,j_2} \cup V_{i,j_3}$ and V_{i,j_1} give us a bijection between $V_{i,j_2} \cup V_{i,j_3}$ and V_{i,j_1}, which will establish the satisfaction of the equation. For each $u \in V_{i,j_2} \cup V_{i,j_3}$, since $\mathcal{G} \models (\#_x[E(u,x) \wedge P_i(x) \wedge U_{j_1}(x)] = 1)$, which implies that there exists only one outgoing edge from u to V_{i,j_1}. On the other hand, for each $u \in V_{i,j_1}$, $\mathcal{G} \models (\#_x[E(x,u) \wedge P(x) \wedge (U_{j_2}(x) \vee U_{j_3}(x))] = 1)$, which implies that there exists only one incoming edge from $V_{i,j_2} \cup V_{i,j_3}$ to u. Therefore we have $1 \cdot |V_{i,j_1}| = 1 \cdot |V_{i,j_2} \cup V_{i,j_3}|$. Thus $|V_{i,j_1}| = |V_{i,j_2}| + |V_{i,j_3}|$
- If the equation is $v_{j_1} = v_{j_2} \cdot v_{j_3}$, $\mathcal{G} \models (\#_y[\psi_i(y)] - \#_y[\top] = 0)$. Then for each $u \in V$, $\mathcal{G} \models \psi_i(u)$.
  We will establish the equality by using the edges between V_{i,j_2} and V_{i,j_1} to show that there is a $|V_{i,j_3}|$-to-one relationship between V_{i,j_1} and V_{i,j_2}. For each $u \in V_{i,j_2}$,

  $$\mathcal{G} \models (\#_x[E(u,x) \wedge P_i(x) \wedge U_{j_1}(x)] - \#_x[P_i(x) \wedge U_{j_3}(x)] = 0),$$

  which implies that there exists $|V_{i,j_3}|$ outgoing edges from u to V_{i,j_1}. On the other hand, for each $u \in V_{i,j_1}$, $\mathcal{G} \models (\#_x[E(x,u) \wedge P_i(x) \wedge U_{j_2}(x)] = 1)$, which implies that there exists only one incoming edge from V_{i,j_2} to u. Therefore we have $1 \cdot |V_{i,j_1}| = |V_{i,j_3}| \cdot |V_{i,j_2}|$.

If ε has a solution $\{v_j \leftarrow a_j\}_{1 \le j \le n}$. Let $\mathcal{G}$ be the $(n+m)$-graph defined as follows. For $1 \le i \le m$, $1 \le j \le n$, and $1 \le k \le a_j$, let $v_{i,j,k}$ be a fresh vertex and V be the set of all such vertices. For $1 \le i \le m$, $P_i := \{v_{i,j,k} |\ 1 \le j \le n; 1 \le k \le a_i\}$. For $1 \le j \le n$, $U_j := \{v_{i,j,k} |\ 1 \le i \le m; 1 \le k \le a_i\}$. For $1 \le i \le m$, we define E_i depending on the i^{th} equation in ε:

- If the equation is $v_j = 1$, then $E_i := \emptyset$.
- If the equation is $v_{j_1} = v_{j_2} + v_{j_3}$, then

$$E_i := \{(v_{i,j_2,k}, v_{i,j_1,k}) |\ 1 \le k \le a_{j_2}\} \cup \left\{(v_{i,j_3,k}, v_{i,j_1,a_{j_2}+k}) \middle|\ 1 \le k \le a_{j_3}\right\}.$$

- If the equation is $v_{j_1} = v_{j_2} \cdot v_{j_3}$, then

$$E_i := \left\{(v_{i,j_2,k}, v_{i,j_1,(k-1)\cdot a_{j_3}+\ell}) \middle|\ 1 \le k \le a_{j_2}; 1 \le \ell \le a_{j_3}\right\}.$$

We set $E := \bigcup_{1\le i\le m} E_i$.

It is straightforward to verify that $\mathcal{G}$ defined over the vertices V with colors given by P_i and U_j, and edges given by E as above is a finite model of $\Psi_\varepsilon(x)$. ◀

D.2 Proof of Theorem 32: undecidability of satisfiability for GNNs with global readout and truncated ReLU

We now recall the theorem, which is one of our main undecidability results:

▶ **Theorem 32.** *The satisfiability problem of* $\mathcal{BG}$TrReLU-GNN*s is undecidable.*

Proof. For every MP^2 formula $\varphi(x)$, by Theorem 26 there exists an equivalent $\mathcal{BG}$TrReLU-GNN $\mathcal{A}_\varphi$. It is easy to see that $\varphi(x)$ is finitely satisfiable if and only if $\mathcal{A}_\varphi$ is satisfiable. Assuming the claim, the undecidibility of the satisfibility problem for $\mathcal{BG}$TrReLU-GNNs follows from Theorem 29. ◀

D.3 Proof of Theorem 33: undecidability of universal satisfiability for GNNs with global readout and truncated ReLU

We recall the theorem:

▶ **Theorem 33.** *The universal satisfiability problem of* $\mathcal{BG}$TrReLU-GNN*s is undecidable.*

We first claim the following strong version of Lemma 31. For every simple equation system ε, there exists a MP^2 formula $\Psi_\varepsilon(x)$, such that the following are equivalent,

1. ε has a solution in $\mathbb{N}$;
2. there exists a graph $\mathcal{G}$ such that for every vertex $v \in V$, $\mathcal{G} \models \Psi_\varepsilon(v)$.

Proof. Let $\Psi_\varepsilon(x)$ be the MP^2 formula defined in Lemma 31 satisfying that ε has a solution in $\mathbb{N}$ if and only if $\Psi_\varepsilon(x)$ is finitely satisfiable.

We note that x is a dummy variable in $\Psi_\varepsilon(x)$. Therefore for every graph $\mathcal{G}$ and vertex $v, u \in V$, $\Psi_\varepsilon(v)$ and $\Psi_\varepsilon(u)$ are the same, which implies that $\mathcal{G} \models \Psi_\varepsilon(v)$ if and only if $\mathcal{G} \models \Psi_\varepsilon(u)$.

If ε has a solution in $\mathbb{N}$, then there exists a graph $\mathcal{G}$ and vertex $v \in V$, such that $\mathcal{G} \models \Psi_\varepsilon(v)$. By the observation above, for every vertex $u \in V$, $\mathcal{G} \models \Psi_\varepsilon(u)$.

The other direction is obvious. ◀

Theorem 33 follows easily from the claim:

Proof. For every simple equation system ε, let $\Psi_\varepsilon(x)$ be the MP^2 formula from the claim. By Theorem 26, there exists a $\mathcal{BG}$TrReLU-GNN $\mathcal{A}_{\Psi_\varepsilon}$, such that $\Psi_\varepsilon(x)$ and $\mathcal{A}_{\Psi_\varepsilon}$ are equivalent. We claim that ε has a solution in $\mathbb{N}$ if and only if $\mathcal{A}_{\Psi_\varepsilon}$ is universally satisfiable. Since the solvability of simple equation systems is undecidable, so is the universal satisfibility problem of $\mathcal{BG}$TrReLU-GNNs.

If ε has a solution in $\mathbb{N}$, by the claim, there exists a graph $\mathcal{G}$, such that for every vertex $v \in V$, $\mathcal{G} \models \Psi_\varepsilon(v)$. Since $\Psi_\varepsilon(x)$ and $\mathcal{A}_{\Psi_\varepsilon}$ are equivalent, $\langle \mathcal{G}, v \rangle$ also satisfies $\mathcal{A}_{\Psi_\varepsilon}$. Therefore $\mathcal{A}_{\Psi_\varepsilon}$ is universally satisfiable with $\mathcal{G}$ as the witness.

On the other hand, if $\mathcal{A}_{\Psi_\varepsilon}$ is universally satisfiable, with witness graph $\mathcal{G}$, then for every vertex $v \in V$, $\langle \mathcal{G}, v \rangle \models \mathcal{A}_{\Psi_\varepsilon}$. Because $\Psi_\varepsilon(x)$ and $\mathcal{A}_{\Psi_\varepsilon}$ are equivalent, $\mathcal{G} \models \Psi_\varepsilon(v)$. Therefore for every vertex $v \in V$, $\mathcal{G} \models \Psi_\varepsilon(v)$. By the claim, ε has a solution in $\mathbb{N}$. ◀

E Proofs from Subsection 3.3: decidability and undecidability for the undirected case

Recall that Section 3.3 refines the decidability and undecidability results for eventually constant activations to focus on undirected graphs, the usual setting for GNNs.

E.1 Proof of Corollary 34: decidability of satisfiability for local MP^2 formulas over undirected graphs

▶ **Corollary 34.** *The finite satisfiability problem of $\mathcal{L}$-MP^2 over undirected graphs is decidable.*

Proof. Let $\varphi(x)$ be a $\mathcal{L}$-MP^2 formula and ψ be the GP^2 sentence mentioned in Corollary 12, which satisfy that $\varphi(x)$ is finitely satisfiable if and only if ψ is also finitely satisfiable. We claim that $\varphi(x)$ is finitely satisfiable over undirected graphs if and only if the GP^2 sentence

$$\psi' \;:=\; \psi \wedge (\forall x\ \forall y\ E(x,y) \to E(y,x))$$

is finitely satisfiable. Since the finite satisfiability problem of GP^2 is decidable, so is the finite satisfiability problem of $\mathcal{L}$-MP^2 over undirected graphs.

The claim can by proven using an argument similar to that used in proving Corollary 12. The only difference is the following observation: for every graph $\mathcal{G}$, $\mathcal{G}$ is undirected if and only $\mathcal{G} \models \forall x\ \forall y\ E(x,y) \to E(y,x)$. ◀

E.2 Proof of Theorem 35: decidability of satisfiability for local, eventually constant GNN over undirected graphs

▶ **Theorem 35.** *The satisfiability problem of $\mathcal{BLC}$-GNNs over undirected graphs is decidable.*

Proof. The proof is similar to Theorem 24. For every $\mathcal{BLC}$-GNN $\mathcal{A}$, by Theorem 23, there exists a $\mathcal{L}$-MP^2 formula $\Psi_{\mathcal{A}}(x)$ such that $\mathcal{A}$ and $\Psi_{\mathcal{A}}(x)$ are equivalent. We claim that $\mathcal{A}$ is satisfiable over undirected graphs if and only if $\Psi_{\mathcal{A}}(x)$ is finitely satisfiable over undirected graphs. Since the finite satisfiability problem over undirected graphs of $\mathcal{L}$-MP^2 is decidable by Corollary 34, we conclude that satisfiability over undirected graphs of $\mathcal{BLC}$-GNNs is decidable. ◀

E.3 Proof of Theorem 36

▶ **Theorem 36.** *The universal satisfiability problem of $\mathcal{BLC}$-GNNs over undirected graphs is decidable.*

Proof. The proof is similar to Theorem 28. For every $\mathcal{BLC}$-GNN $\mathcal{A}$, by Theorem 23, there exists a $\mathcal{L}$-MP^2 formula $\varphi_{\mathcal{A}}(x)$ such that $\mathcal{A}$ and $\varphi_{\mathcal{A}}(x)$ are equivalent. We claim that $\mathcal{A}$ is universally satisfiable over undirected graphs if and only if the GP^2 sentence

$$\psi \;:=\; (\forall x\ (x = x) \to \varphi_{\mathcal{A}}(x)) \wedge (\forall x\ \forall y\ E(x,y) \to E(y,x))$$

is finitely satisfiable. Since the sentence ψ enforces the edges to be undirected and the finite satisfiability of GP^2 is decidable by Corollary 11, so is universal satisfiability of $\mathcal{BLC}$-GNNs over undirected graphs. ◀

E.4 Proofs of Theorems 37 – 39: undecidability results for global GNNs over undirected graphs

In the body of the paper we had Theorem 37, Theorem 38, and Theorem 39, which are variations of prior results for the case of undirected graphs. The proofs are similar to Theorem 29, Theorem 32, and Theorem 33. The only difference is that they are based on the following stronger version of Lemma 31.

▶ **Lemma 71.** *For every simple equation system ε with n variables and m equations, there exists an $(n+m)$-MP^2 formula $\varphi_\varepsilon(x)$ such that ε has a solution in $\mathbb{N}$ if and only if $\Psi_\varepsilon(x)$ is finitely satisfiable* over undirected graphs*.*

Proof. For every $(n+m)$-graph $\mathcal{G}$, let $\mathcal{G}'$ be the $(n+m)$-graph by modifying the edges: $E' := E \cup \{(u,v) | \ (v,u) \in E\}$. It is clear that $\mathcal{G}'$ is an undirected graph. For every vertex $v \in V$, it is routine to check that $\mathcal{G} \models \Psi_\varepsilon(v)$ if and only if $\mathcal{G}' \models \Psi_\varepsilon(v)$. The lemma follows from the above observation and Lemma 31. ◀

F Proofs from Subsection 4.1: undecidability and expressiveness results for unbounded activation functions

F.1 Proof of Theorem 41: from $\mathcal{L}$-$\mathsf{M2P}^2$ to $\mathcal{BL}$ReLU-GNN

We recall the theorem:

▶ **Theorem 41.** *For every n-$\mathcal{L}$-$\mathsf{M2P}^2$ formula $\Psi(x)$, there exists an n-$\mathcal{BL}$ReLU-GNN $\mathcal{A}_\Psi$, such that $\Psi(x)$ and $\mathcal{A}_\Psi$ are equivalent.*

Recall that $\mathcal{L}$-$\mathsf{M2P}^2$ is a logic with "two-hop Presburger quantifiers", while $\mathcal{BL}$ReLU-GNN refers to bidirectional GNNs with ReLU activated functions, but only local aggregation. We will apply a proof technique similar to the one used in Theorem 26 to show that the $\mathcal{L}$-$\mathsf{M2P}^2$ formulas are captured by $\mathcal{BL}$ReLU-GNNs.

For every n-$\mathcal{L}$-$\mathsf{M2P}^2$ formula $\Psi(x)$, let L be the number of subformulas of $\Psi(x)$ and $\{\varphi_i(x)\}_{1 \le i \le L}$ be an enumeration of subformulas of $\Psi(x)$ that satisfy $\varphi_L(x)$ is $\Psi(x)$, and for each $\varphi_i(x)$ and $\varphi_j(x)$, if $\varphi_i(x)$ is a strict subformula of $\varphi_j(x)$, then $i < j$.

We define the $(3L+1)$-layer n-$\mathcal{BL}$ReLU-GNN $\mathcal{A}_\Psi$ as follows. The input dimension d_0 is n. For $1 \le \ell \le 3L$, the dimension d_ℓ is $3L$, and $d_{3L+1} = 1$. The numbers in the coefficient matrices and bias vectors are defined by the following rules. For $1 \le i \le L$,

- if $\varphi_i(x) = \top$, then $b_i^{3i} = 1$.
- if $\varphi_i(x) = U_j(x)$ for some unary predicate U_j, then $C_{i,j}^1 = 1$ and for $2 \le \ell \le 3i$, $C_{i,i}^\ell = 1$.
- if $\varphi_i(x) = \neg\varphi_j(x)$, then $C_{i,j}^{3i} = -1$, $b_i^{3i} = 1$.
- if $\varphi_i(x) = \varphi_{j_1}(x) \wedge \varphi_{j_2}(x)$, then $C_{i,j_1}^{3i} = C_{i,j_2}^{3i} = 1$, $b_i^{3i} = -1$.
- if

$$\varphi_i(x) = \left(\sum_{t=1}^{k} \lambda_t \cdot \#_{z,y}[\epsilon_t(x,z,y) \wedge \varphi_{j_t}(y)] + \sum_{t=1}^{k'} \lambda'_t \cdot \#_y[\epsilon'_t(x,y) \wedge \varphi_{j'_t}(y)] \ge \delta \right),$$

then $b_i^{3i-1} = \delta$, $b_i^{3i} = 1$, and $C_{i,i}^{3i} = -1$. For $1 \le t \le k$,

- if $\epsilon_t(x,z,y) = E(x,z) \wedge E(z,y)$, then $\left(A^{3i-1}_{\text{out}}\right)_{i,L+j_t} = -\lambda_t$.
- if $\epsilon_t(x,z,y) = E(x,z) \wedge E(y,z)$, then $\left(A^{3i-1}_{\text{out}}\right)_{i,2L+j_t} = -\lambda_t$.
- if $\epsilon_t(x,z,y) = E(z,x) \wedge E(z,y)$, then $\left(A^{3i-1}_{\text{in}}\right)_{i,L+j_t} = -\lambda_t$.
- if $\epsilon_t(x,z,y) = E(z,x) \wedge E(y,z)$, then $\left(A^{3i-1}_{\text{in}}\right)_{i,2L+j_t} = -\lambda_t$.

For $1 \leq t \leq k'$,

- if $\epsilon'_t(x,y) = E(x,y)$, then $\left(A^{3i-1}_{\text{out}}\right)_{i,j'_t} = -\lambda'_t$.
- if $\epsilon'_t(x,y) = E(y,x)$, then $\left(A^{3i-1}_{\text{in}}\right)_{i,j'_t} = -\lambda'_t$.

For $1 \leq i \leq L$ and $3i+1 \leq \ell \leq 3L$, $C^{\ell}_{i,i} = 1$. For $1 \leq i \leq L$ and $3i+2 \leq \ell \leq 3L$, $C^{\ell}_{L+i,L+i} = C^{\ell}_{2L+i,2L+i} = 1$. All other numbers in the coefficients matrices and bias vectors are 0.

The theorem relies on the following inductive invariant:

▶ **Lemma 72.** *For every n-graph $\mathcal{G}$ and vertex $v \in V$, for $1 \leq i \leq L$,*

1. *for $3i \leq \ell \leq 3L$, if $\mathcal{G} \models \varphi_i(v)$, then $\xi^{\ell}_i(v) = 1$. Otherwise, if $\mathcal{G} \not\models \varphi_i(v)$, then $\xi^{\ell}_i(v) = 0$.*
2. *for $3i+1 \leq \ell \leq 3L$, $\xi^{\ell}_{L+i}(v) = |\{u \in V|\ \mathcal{G} \models E(v,u) \wedge \varphi_i(u)\}|$.*
3. *for $3i+1 \leq \ell \leq 3L$, $\xi^{\ell}_{2L+i}(v) = |\{u \in V|\ \mathcal{G} \models E(u,v) \wedge \varphi_i(u)\}|$.*

Proof. For $3i+1 \leq \ell \leq L$, $\xi^{\ell}_i(v) = \mathsf{ReLU}\left(\xi^{\ell-1}_i(v)\right) = \xi^{3i}_i(v)$. For $3i+2 \leq \ell \leq L$, $\xi^{\ell}_{L+i}(v) = \mathsf{ReLU}\left(\xi^{\ell-1}_{L+i}(v)\right) = \xi^{3i}_{L+i}(v)$ and $\xi^{\ell}_{2L+i}(v) = \mathsf{ReLU}\left(\xi^{\ell-1}_{2L+i}(v)\right) = \xi^{3i}_{2L+i}(v)$. It is sufficient to show the first property holds for $\ell = 3i$ and the last two properties hold for $\ell = 3i+1$.

We prove the properties by induction on subformulas. For the first property, the proof for the cases of $\top$, $U_j(x)$, $\neg\varphi_j(x)$ and $\varphi_{j_1}(x) \wedge \varphi_{j_2}(x)$ is the same as in Lemma 59. Here we only consider the case of two-hop Presburger quantifiers.

- Suppose

$$\varphi_i(x) = \left(\sum_{t=1}^{k} \lambda_t \cdot \#_{z,y}[\epsilon_t(x,z,y) \wedge \varphi_{j_t}(y)] + \sum_{t=1}^{k'} \lambda'_t \cdot \#_y[\epsilon'_t(x,y) \wedge \varphi_{j'_t}(y)] \geq \delta\right).$$

Note that for $1 \leq t \leq k$, because $\varphi_{j_t}(x)$ is a strict subformula of $\varphi_i(x)$, we have $3i-2 \geq 3j_t+1$. By the induction hypothesis, for every $u \in V$,

$$\begin{aligned}\xi^{3i-2}_{L+j_t}(u) &= |\{u' \in V|\ \mathcal{G} \models E(u,u') \wedge \varphi_{j_t}(u')\}| \\ \xi^{3i-2}_{2L+j_t}(u) &= |\{u' \in V|\ \mathcal{G} \models E(u',u) \wedge \varphi_{j_t}(u')\}|\,.\end{aligned}$$

- If $\epsilon_t(x,z,y) = E(x,z) \wedge E(z,y)$, then

$$\begin{aligned}&\left|\left\{(u_z,u_y) \in V^2\right|\ \mathcal{G} \models \epsilon_t(v,u_z,u_y) \wedge \varphi_{j_t}(u_y)\right\}\Big| \\ =& \sum_{u_z \in \mathcal{N}_{\text{out}}(v)} |\{u_y \in V|\ \mathcal{G} \models E(u_z,u_y) \wedge \varphi_{j_t}(u_y)\}| \\ =& \sum_{u \in \mathcal{N}_{\text{out}}(v)} \xi^{3i-2}_{L+j_t}(u)\end{aligned}$$

  We can treat the other three cases analogously:
- If $\epsilon_t(x,z,y) = E(x,z) \wedge E(y,z)$, then

$$\left|\left\{(u_z,u_y) \in V^2\right|\ \mathcal{G} \models \epsilon_t(v,u_z,u_y) \wedge \varphi_{j_t}(u_y)\right\}\Big| = \sum_{u \in \mathcal{N}_{\text{out}}(v)} \xi^{3i-2}_{2L+j_t}(u).$$

- If $\epsilon_t(x, z, y) = E(z, x) \wedge E(z, y)$, then

$$\left|\left\{(u_z, u_y) \in V^2 \middle| \mathcal{G} \models \epsilon_t(v, u_z, u_y) \wedge \varphi_{j_t}(u_y)\right\}\right| = \sum_{u \in \mathcal{N}_{\text{in}}(v)} \xi^{3i-2}_{L+j_t c}(u).$$

- If $\epsilon_t(x, z, y) = E(z, x) \wedge E(y, z)$, then

$$\left|\left\{(u_z, u_y) \in V^2 \middle| \mathcal{G} \models \epsilon_t(v, u_z, u_y) \wedge \varphi_{j_t}(u_y)\right\}\right| = \sum_{u \in \mathcal{N}_{\text{in}}(v)} \xi^{3i-2}_{2L+j_t}(u).$$

For $1 \leq t \leq k'$, because $\varphi_{j'_t}(x)$ is a strict subformula of $\varphi_i(x)$, we have $3i - 2 \geq 3j'_t$. By the induction hypothesis, for every $u \in V$, if $\mathcal{G} \models \varphi_{j'_t}(u)$, then $\xi^{3i-2}_{j'_t}(u) = 1$. Otherwise, if $\mathcal{G} \not\models \varphi_{j'_t}(u)$, then $\xi^{3i-2}_{j'_t}(u) = 0$.

- If $\epsilon_{j'_t}(x, y) = E(x, y)$, then

$$\left|\left\{u \in V \middle| \mathcal{G} \models \epsilon'_t(v, u) \wedge \varphi_{j'_t}(u)\right\}\right| = \sum_{u \in \mathcal{N}_{\text{out}}(v)} \xi^{3i-2}_{j'_t}(u)$$

- If $\epsilon'_t(x, y) = E(y, x)$, then

$$\left|\left\{u \in V \middle| \mathcal{G} \models \epsilon'_t(v, u) \wedge \varphi_{j'_t}(u)\right\}\right| = \sum_{u \in \mathcal{N}_{\text{in}}(v)} \xi^{3i-2}_{j'_t}(u)$$

Let w be the value defined:

$$\begin{aligned} w \ := \ & -\sum_{t=1}^{k} \lambda_j \cdot \left|\left\{(u_z, u_y) \in V^2 \middle| \mathcal{G} \models \epsilon'_t(v, u_z, u_y) \wedge \varphi_{j_t}(u_y)\right\}\right| \\ & -\sum_{t=1}^{k'} \lambda'_j \cdot \left|\left\{u \in V \middle| \mathcal{G} \models \epsilon'_t(v, u) \wedge \varphi_{j'_t}(u)\right\}\right| \\ = \ & \left(\sum_{x \in \{\text{out},\text{in}\}} A^{3i-1}_x \sum_{u \in \mathcal{N}_x(v)} \xi^{3i-2}(u)\right)_i \end{aligned}$$

Since for $1 \leq j \leq 3L$, $C^{3i-1}_{i,j} = 0$, $\left(C^{3i-1}\xi^{3i-2}(v)\right)_i = 0$. Thus

$$\begin{aligned} \xi^{3i-1}_i(v) &= \mathsf{ReLU}\left(\left(C^{3i-1}\xi^{3i-2}(v)\right)_i + w + b^{3i-1}\right) \ = \ \xi^{3i-1}(v) \ = \ \mathsf{ReLU}(w + \delta) \\ \xi^{3i}_i(v) &= \mathsf{ReLU}\left(1 - \xi^{3i-1}_i(v)\right) \ = \ \mathsf{ReLU}\left(1 - \mathsf{ReLU}(w + \delta)\right) \end{aligned}$$

If $\mathcal{G} \models \varphi_i(v)$, by the semantics of two-hop Presburger quantifiers, $-w \geq \delta$. Hence $\xi^{3i}_i(v) = 1$. On the other hand, if $\mathcal{G} \not\models \varphi_i(v)$, by the semantic of two-hop Presburger quantifiers, $-w < \delta$. Hence $\xi^{3i}_i(v) = 0$.

For the last two properties, By the induction hypothesis of the first property, for every $u \in V$, if $\mathcal{G} \models \varphi_i(u)$, then $\xi^{3i}_i(u) = 1$. Otherwise, if $\mathcal{G} \not\models \varphi_i(u)$, then $\xi^{3i}_i(u) = 0$. Therefore

$$\begin{aligned} \xi^{3i+1}_{L+i}(v) &= \mathsf{ReLU}\left(\sum_{u \in \mathcal{N}_{\text{out}}(v)} \xi^{3i}_i(u)\right) = |\{u \in \mathcal{N}_{\text{out}}(v) |\ \mathcal{G} \models \varphi_i(u)\}| \\ &= |\{u \in V |\ \mathcal{G} \models E(v, u) \wedge \varphi_i(u)\}| \\ \xi^{3i+1}_{2L+i}(v) &= \mathsf{ReLU}\left(\sum_{u \in \mathcal{N}_{\text{in}}(v)} \xi^{3i}_i(u)\right) = |\{u \in \mathcal{N}_{\text{in}}(v) |\ \mathcal{G} \models \varphi_i(u)\}| \\ &= |\{u \in V |\ \mathcal{G} \models E(u, v) \wedge \varphi_i(u)\}| \end{aligned}$$

This completes the proof of Lemma 72. ◀

Theorem 41 follows easily from the first property in Lemma 72.

F.2 Proof of Lemma 43: Reduction from a two-counter machine to an M2P^2 formula.

We recall the lemma.

▶ **Lemma 43.** *For every two-counter machine $\mathcal{M}$ with n instructions, there exists an $(n+5)$-$\mathcal{L}$-M2P^2 formula $\Psi_{\mathcal{M}}(x)$ such that $\mathcal{M}$ halts if and only if $\forall x\ \Psi_{\mathcal{M}}(x)$ is finitely satisfiable.*

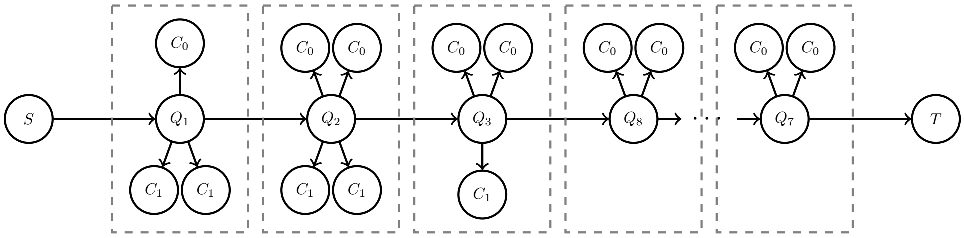


Figure 2 An example of the encoding of the computation of the two-counter machines to directed graphs.

Recall the intuition of the reduction from the body. We have illustrated it in Figure 2. Each configuration is encoded as a height 1 tree, which is denoted by a dashed box. Its line number is represented by the unary predicate Q_i realized by the root vertex, and the values of the counters are represented by the number of "labeled leaves" – those with predicate C_0 or C_1 being true. There are edges connected to the roots of each configuration, which encode the computation sequence. Then it is possible to assert the (in)equality between the number of leaves of some root and the root of the successor tree, which encodes the condition of a valid transition.

Proof. Let τ be the vocabulary consisting of unary predicates $\{S, T, Q, C_0, C_1\} \cup \{Q_i\}_{1 \le i \le n}$ and a binary predicate E. Informally, S and T will be indicators for the beginning and the end of the sequence. Q will represent the central "state holding" vertices in the figure, with each such vertex satisfying exactly one Q_i, which will represent the particular state of the computation.

We first define some useful gadgets.

$$\begin{aligned}
\psi^{diff}_{i,\delta}(x) &:= (1 \cdot \#_{z,y}[E(x,z) \wedge E(z,y) \wedge C_i(y)] - 1 \cdot \#_y[E(x,y) \wedge C_i(y)] = \delta) \\
\psi^{zero}_i(x) &:= (1 \cdot \#_y[E(x,y) \wedge C_i(y)] = 0) \\
\psi^{succ}_j(x) &:= (1 \cdot \#_y[E(x,y) \wedge Q_j(y)] = 1) \\
\psi^{\#out}_i(x) &:= (1 \cdot \#_y[E(x,y) \wedge \top] = i) \\
\psi^{\#in}_i(x) &:= (1 \cdot \#_y[E(y,x) \wedge \top] = i)
\end{aligned}$$

Next, for $1 \le q \le n$, letting d_q be the q^{th} instruction of $\mathcal{M}$:

- If d_q is `INC` (c_i), then $\psi_q(x) := \psi^{succ}_{q+1}(x) \wedge \psi^{diff}_{i,1}(x) \wedge \psi^{diff}_{1-i,0}(x)$.
- If d_q is `IF` $(c_i = 0)$ `GOTO` (j), then

$$\begin{aligned}
\psi_q(x) &:= \left(\psi^{zero}_i(x) \to \psi^{if}_q(x)\right) \wedge \left(\neg\psi^{zero}_i(x) \to \psi^{else}_q(x)\right) \\
\psi^{if}_q(x) &:= \psi^{succ}_j(x) \wedge \psi^{diff}_{0,0}(x) \wedge \psi^{diff}_{1,0}(x) \\
\psi^{else}_q(x) &:= \psi^{succ}_{q+1}(x) \wedge \psi^{diff}_{i,-1}(x) \wedge \psi^{diff}_{1-i,0}(x).
\end{aligned}$$

- If d_q is HALT, then $\psi_q(x) := (1 \cdot \#_y[E(x,y) \wedge T(y)] = 1)$.

Finally, we define $\Psi_{\mathcal{M}}(x)$ as follows.

$$
\begin{aligned}
\varphi_1(x) &:= \left(\bigvee_{U \in \{S,T,Q,C_0,C_1\}} U(x) \right) \wedge \left(\bigwedge_{\substack{U_1,U_2 \in \{S,T,Q,C_0,C_1\} \\ U_1 \neq U_2}} \neg U_1(x) \vee \neg U_2(x) \right) \\
\varphi_2(x) &:= \varphi_1^{\#in}(x) \wedge (1 \cdot \#_y[E(y,x) \wedge Q(y)] = 1) \wedge \varphi_1^{\#out}(x) \wedge (1 \cdot \#_y[E(x,y) \wedge S(y)] = 1) \\
\varphi_3(x) &:= \varphi_1^{\#in}(x) \wedge (1 \cdot \#_y[E(y,x) \wedge (Q(y) \vee S(y)] = 1) \wedge \\
&\quad (1 \cdot \#_y[E(x,y) \wedge (Q(y) \vee T(y))] = 1) \wedge (1 \cdot \#_y[E(x,y) \wedge S(y)] = 1) \\
\varphi_4(x) &:= \left(\bigvee_{1 \leq i \leq n} Q_i(x) \right) \wedge \left(\bigwedge_{1 \leq i < j \leq n} \neg Q_i(x) \vee \neg Q_j(x) \right) \\
\varphi_5(x) &:= \varphi_1^{\#out}(x) \wedge (1 \cdot \#_y[E(x,y) \wedge Q(y)] = 1) \\
\varphi_6(x) &:= (\#_y[E(x,y) \wedge Q(y) \wedge Q_1(y) \wedge \psi_0^{zero}(y) \wedge \psi_1^{zero}(y)] = 1) \\
\Psi_{\mathcal{M}}(x) &:= \varphi_1(x) \wedge \bigwedge_{1 \leq q \leq n} (Q_q(x) \rightarrow \psi_q(x)) \wedge \\
&\quad ((C_1(x) \vee C_2(x) \vee T(x)) \rightarrow \varphi_2(x)) \wedge \\
&\quad (Q(x) \rightarrow \varphi_3(x) \wedge \varphi_4(x)) \wedge \\
&\quad (S(x) \rightarrow \varphi_5(x) \wedge \varphi_6(x))
\end{aligned}
$$

We first suppose that $\forall x\ \Psi_{\mathcal{M}}(x)$ is finitely satisfiable with witness $\mathcal{G}$. We will argue that the machine halts. For every vertex $v \in V$, if $\mathcal{G} \models \varphi_i^{\#out}(v)$, then the number of outgoing edges from v is i. if $\mathcal{G} \models \varphi_i^{\#in}(v)$, then the number of incoming edges from v is i. Because $\mathcal{G} \models \varphi_1(v)$, v is in exactly one of S, T, Q, C_0, or C_1. We say a vertex is a Q vertex if it is in Q. We use a similar naming convention in the other cases. We first claim that every model of $\forall x\ \Psi_{\mathcal{M}}(x)$ has a structure as depicted in Figure 2 except that for every non S vertex, there exists an edge from it to a S vertex.

- For every T vertex or C_i vertex v, where $i \in \{0,1\}$, because $\mathcal{G} \models \varphi_2(v)$, there exists exactly one edge incoming from a Q vertex to v and there exists exactly one edge from v to a S vertex.
- For every Q vertex v, because $\mathcal{G} \models \varphi_3(v)$, there exists a unique Q vertex or S vertex u such that there exists an edge from u to v. We refer to this vertex as the *predecessor* of v. There exists a unique Q vertex or T vertex u such that there exists an edge from v to u. If this vertex is a Q vertex, then we refer to it as the *successor* of v.
  Since $\mathcal{G} \models \varphi_4(v)$, v is in exactly one Q_q, where $1 \leq q \leq n$. We refer to this q as the *instruction number* of v. The configuration of v, denoted by $\mathsf{conf}(v)$ is defined as $\langle q, c_0, c_1 \rangle$, where q is the instruction number of v; $c_0 := |\{u \in \mathcal{N}_{\text{out}}(v) | \ \mathcal{G} \models C_0(u)\}|$; $c_1 := |\{u \in \mathcal{N}_{\text{out}}(v) | \ \mathcal{G} \models C_1(u)\}|$.
- For every S vertex v, because $\mathcal{G} \models \varphi_5(v)$, there exists a unique Q vertex u such that there exists an edge from v to u.
  Moreover, because $\mathcal{G} \models \varphi_6(v)$, the configuration of the u is $\langle 1, 0, 0 \rangle$.

Next, we claim that for every Q vertex v and its successor u, $\mathsf{conf}(u)$ is the successor configuration of $\mathsf{conf}(v)$. Let $\mathsf{conf}(v) = \langle q, c_0, c_1 \rangle$ and $\mathsf{conf}(u) = \langle q', c_0', c_1' \rangle$. We first note that if $\mathcal{G} \models \psi_{i,\delta}^{diff}(v)$, then $c_i' - c_i = \delta$; if $\mathcal{G} \models \psi_j^{succ}(v)$, then $q' = j$ Let d_q be the q^{th} instruction of $\mathcal{M}$.

- If d_q is $\texttt{INC}$ (c_i), then $\mathcal{G} \models \psi^{succ}_{q+1}(v)$, $\mathcal{G} \models \psi^{diff}_{i,1}(v)$, and $\mathcal{G} \models \psi^{diff}_{1-i,0}(v)$. Thus $q' = q+1$, $c'_i = c_i + 1$, and $c'_{1-i} = c_{1-i}$. Hence $\mathsf{conf}(u)$ is the successor configuration of $\mathsf{conf}(v)$.
- If d_q is $\texttt{IF}$ $(c_i = 0)$ $\texttt{GOTO}$ (j), if $c_i = 0$, then $\mathcal{G} \models \psi^{succ}_{j}(v)$, $\mathcal{G} \models \psi^{diff}_{0,0}(v)$, and $\mathcal{G} \models \psi^{diff}_{1,0}(v)$. Thus $q' = j$, $c'_0 = c_0$, and $c'_1 = c_1$. Otherwise, if $c_i > 0$, then $\mathcal{G} \models \psi^{succ}_{q+1}(v)$, $\mathcal{G} \models \psi^{diff}_{i,-1}(v)$, and $\mathcal{G} \models \psi^{diff}_{1-i,0}(v)$. Thus $q' = q+1$, $c'_i = c_i - 1$, and $c'_{1-i} = c_{1-i}$. Hence $\mathsf{conf}(u)$ is the successor configuration of $\mathsf{conf}(v)$.
- If d_q is $\texttt{HALT}$, then $\mathcal{G} \models (1 \cdot \#_y[E(v,y) \wedge T(y)] = 1)$, v has no successor.

Finally, we claim that $\mathcal{M}$ halts. Our definition of graph requires the vertex set V to be nonempty. Let v be a vertex in V. If v is not a S vertex, then there exists an edge from v to a S vertex v_s. For an S vertex s, there exists a Q vertex v_1, such that v_1 has no predecessor and $\mathsf{conf}(v_1) = \langle 1,0,0\rangle$.

We consider the sequence $v_1, v_2, \ldots, v_\ell$, where v_{i+1} is the successor of v_i and v_ℓ has no successor. We claim that there exists such a vertex v_ℓ. If there is no such v_ℓ, since $\mathcal{G}$ is a finite graph, there exists v' and v'' in the sequence such that v' and v'' are the same. Let $1 \leq i < j$ be the smallest pair satisfying $v_i = v_j$. If $i = 1$, then v_{j-1} is a predecessor of v_1, but v_1 has no predecessor. Hence we have a contradiction. If $i > 1$, then v_{i-1} and v_{j-1} are both predecessor of v_i, but v_i has at most one predecessor. We again have a contradiction. Thus we can always find such a v_ℓ. By the claim above, $\mathsf{conf}(v_{i+1})$ is the successor configuration of $\mathsf{conf}(v_i)$. In addition, $\mathsf{conf}(v_1) = \langle 1,0,0\rangle$ and $\mathsf{conf}(v_\ell)$ is a halt configuration, since $\mathsf{conf}(v_\ell)$ has no successor. Therefore $\mathsf{conf}(v_1), \mathsf{conf}(v_2), \ldots, \mathsf{conf}(v_\ell)$ is a computation of $\mathcal{M}$. Because its length is finite, $\mathcal{M}$ halts.

If $\mathcal{M}$ halts, then it is straightforward to encode its computation into a finite graph as in Figure 2 and to check that the graph is a model of $\forall x\ \Psi_{\mathcal{M}}(x)$. ◀

F.3 Proof of Theorem 44: undecidability for universal satifiability of GNNs with ReLU

We now recall the theorem:

▶ **Theorem 44.** *The universal satisfiability problem of $\mathcal{BL}$ReLU-GNNs is undecidable.*

Proof. For every two-counter machine $\mathcal{M}$, by Lemma 43, there exists a $\mathcal{L}$-M2P^2 formula $\Psi_{\mathcal{M}}(x)$ such that the machine halts if and only if $\forall x\ \Psi_{\mathcal{M}}(x)$ is finitely satisfiable. By Theorem 41, there exists a $\mathcal{BL}$ReLU-GNN $\mathcal{A}_{\Psi_{\mathcal{M}}}$ such that $\Psi_{\mathcal{M}}(x)$ and $\mathcal{A}_{\Psi_{\mathcal{M}}}$ are equivalent. We claim that $\mathcal{M}$ halts if and only if $\mathcal{A}_{\Psi_{\mathcal{M}}}$ is universally satisfiable. From this it would immediately follow that the universal satisfiability problem of $\mathcal{BL}$ReLU-GNNs is undecidable.

It is sufficient to show that $\forall x\ \Psi_{\mathcal{M}}(x)$ is finitely satisfiable if and only if $\mathcal{A}_{\Psi_{\mathcal{M}}}$ is universally satisfiable. If $\forall x\ \Psi_{\mathcal{M}}(x)$ is finitely satisfiable, then there exists a graph $\mathcal{G}$ such that for every vertex $v \in V$, $\langle \mathcal{G}, v\rangle \models \Psi_{\mathcal{M}}(x)$. By the equivalence between $\Psi_{\mathcal{M}}(x)$ and $\mathcal{A}_{\Psi_{\mathcal{M}}}$, each $\langle \mathcal{G}, v\rangle$ satisfies $\mathcal{A}_{\Psi_{\mathcal{M}}}$. Then, by definition, $\mathcal{A}_{\Psi_{\mathcal{M}}}$ is universally satisfiable by $\mathcal{G}$. If $\mathcal{A}_{\Psi_{\mathcal{M}}}$ is universally satisfiable, then there exists a graph $\mathcal{G}$ such that for every vertex $v \in V$, $\langle \mathcal{G}, v\rangle \models \mathcal{A}_{\Psi_{\mathcal{M}}}$. By the equivalence, $\langle \mathcal{G}, v\rangle \models \Psi_{\mathcal{M}}(x)$. Hence by the definition, $\mathcal{G} \models \forall x\ \Psi_{\mathcal{M}}(x)$, which implies that $\forall x\ \Psi_{\mathcal{M}}(x)$ is finitely satisfiable. ◀

F.4 Proof of Lemma 49: reduction from a 2-counter machine to a $\mathcal{L}$-M2P^2 formula over undirected graphs

We recall the lemma.

▶ **Lemma 49.** *For every two-counter machine $\mathcal{M}$ with n instructions, there exists an $(n+8)$-$\mathcal{L}$-M2P^2 formula $\Psi_{\mathcal{M}}(x)$ such that $\mathcal{M}$ halts if and only if $\forall x\ \Psi_{\mathcal{M}}(x)$ is finitely satisfiable over undirected graphs.*

We cannot apply the exact encoding from Figure 2 and Lemma Lemma 43 here, because in that encoding we distinguished the predecessor and successor configurations by the direction of edges. Here, we sketch the trick that overcomes the lack of direction in the edges. We will utilize the predicates from the proof of Lemma 43: in particular we will have a predicate Q and an associated notion of Q vertex as in that proof.

We introduce three fresh unary predicates I_0, I_1, and I_2 to label the configuration's index modulo 3. We add an extra clause to the formula to guarantee that each element has exactly one of these three index labels. The elements in each 1-level tree will have the same index, in the sense of satisfying the same index predicates. Finally, for each Q vertex v with index i, there exists at most one Q vertex v' with index $(i+1 \bmod 3)$, such that v and v' are connected; there exists at most one Q vertex v'' with index $(i-1 \bmod 3)$, such that v and v' are connected. Therefore we can modify the formula which identifies the successor and predecessor based on the index, rather than the direction of the edges, and show that the two-counter machine halts if and only if the modified formula is finitely satisfiable over undirected graphs.

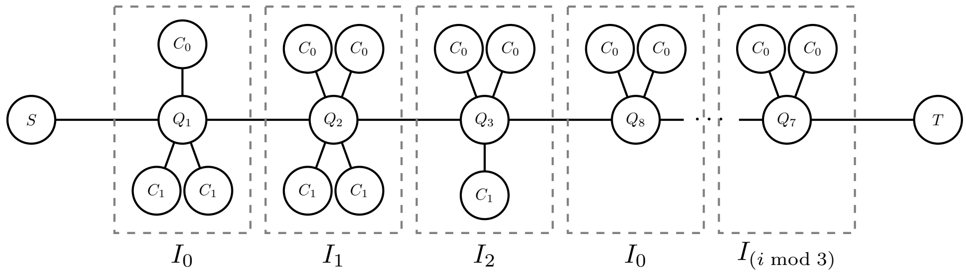


Figure 3 An example of the encoding of the computation of a two-counter machines in undirected graphs.

Formally the reduction is as follows. Given a two-counter machine $\mathcal{M}$, we construct the formula $\Psi_{\mathcal{M}}(x)$ as in the proof of Lemma 43. The only difference are the formulas:

$$\psi_j^{succ}(x) := \bigwedge_{0\le i\le 2} I_i(x) \to (1 \cdot \#_y[E(x,y) \wedge Q_j(y) \wedge I_{i+1 \bmod 3}(y)] = 1)$$

$$\varphi_4(x) := \left(\bigvee_{1\le i\le n} Q_i(x)\right) \wedge \left(\bigwedge_{1\le i<j\le n} \neg Q_i(x) \vee \neg Q_j(x)\right) \wedge \left(\bigvee_{0\le i\le 2} I_i(x)\right) \wedge \left(\bigwedge_{0\le i<j\le 2} \neg I_i(x) \vee \neg I_j(x)\right)$$

The proof that $\mathcal{M}$ halts if and only if $\forall x \Psi_{\mathcal{M}}(x)$ is finitely satisfiable over undirected graphs is similar to the one for Lemma 43, thus, omitted.

F.5 Proof of Theorem 50: undecidability of the universal satisfiability problem for $\mathcal{BL}$ReLU-GNN over undirected graphs

We recall the theorem.

▶ **Theorem 50.** *The universal satisfiability problem of* $\mathcal{BL}$ReLU-GNN*s over undirected graphs is undecidable.*

The theorem follows from Lemma 49, using the undecidability result for two-counter machines, as in Theorem 44.

F.6 Proof of Lemma 45: strict inclusion of $\mathcal{BLC}$-GNN in $\mathcal{L}$-M2P^2

We first recall the lemma:

▶ **Lemma 45.** $\mathcal{L}$-M2P^2 *is strictly more expressive than* $\mathcal{BLC}$-GNN.

The proof is by constructing a sequence of pairs of graphs which can be distinguished by a $\mathcal{L}$-M2P^2 formula, but not any $\mathcal{BLC}$-GNN.

▶ **Definition 73.** *For* $n_1, n_2 \in \mathbb{N}$*, the* (n_1, n_2)-bipolar graph $\langle V, E, U_1, U_2\rangle$ *is an undirected* 2*-graph defined as follows.*

$$\begin{aligned} U_1 &:= \{v_{1,i} \mid 1 \le i \le n_1\} \\ U_2 &:= \{v_{2,i} \mid 1 \le i \le n_2\} \\ V &:= U_1 \cup U_2 \cup \{v_0, v_1, v_2\} \\ \widetilde{E} &:= \{(v_0, v_1), (v_0, v_2)\} \cup \{(v_1, v_{1,i}) \mid 1 \le i \le n_1\} \cup \{(v_2, v_{2,i}) \mid 1 \le i \le n_2\} \\ E &:= \widetilde{E} \cup \left\{(u, v) \,\middle|\, (v, u) \in \widetilde{E}\right\} \end{aligned}$$

See Figure 4.

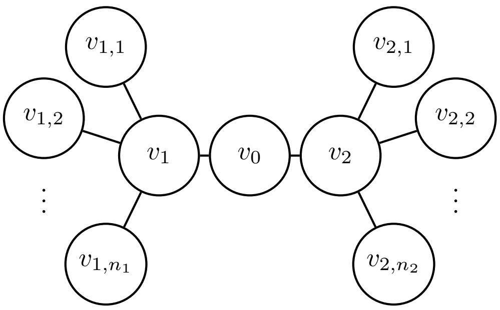


Figure 4 (n_ℓ, n_r)-bipolar graph.

We first show that $\mathcal{BLC}$-GNNs cannot distinguish sufficiently large pairs of bipolar graphs.

▶ **Lemma 74.** *For each* 2-$\mathcal{BLC}$-GNN $\mathcal{A}$*, there exist a threshold* $n_{\mathcal{A}} \in \mathbb{N}$*, such that for every* $n_1, n_2 \ge n_{\mathcal{A}}$*, the following properties hold. Let* $\mathcal{G}$ *be the* $(n_{\mathcal{A}}, n_{\mathcal{A}})$*-bipolar graph and* $\mathcal{G}'$ *be the* (n_1, n_2)*-bipolar graph. For every* $0 \le \ell \le L$*,*

- $\xi^\ell_{\mathcal{G}}(v_0) = \xi^\ell_{\mathcal{G}'}(v'_0)$, $\xi^\ell_{\mathcal{G}}(v_1) = \xi^\ell_{\mathcal{G}'}(v'_1)$, *and* $\xi^\ell_{\mathcal{G}}(v_2) = \xi^\ell_{\mathcal{G}'}(v'_2)$.
- *for* $1 \le i \le n_{\mathcal{A}}$ *and* $1 \le j \le n_1$, $\xi^\ell_{\mathcal{G}}(v_{1,1}) = \xi^\ell_{\mathcal{G}}(v_{1,i}) = \xi^\ell_{\mathcal{G}'}(v'_{1,j})$.
- *for* $1 \le i \le n_{\mathcal{A}}$ *and* $1 \le j \le n_2$, $\xi^\ell_{\mathcal{G}}(v_{2,1}) = \xi^\ell_{\mathcal{G}}(v_{2,i}) = \xi^\ell_{\mathcal{G}'}(v'_{2,j})$.

Above $\xi^\ell_{\mathcal{G}}$ refers to the ℓ^{th} derived feature function of the GNN $\mathcal{A}$ over the graph $\mathcal{G}$.

Proof. Recall from Theorem 17 that the spectrum of a $\mathcal{BLC}$-GNN at any layer ℓ, denoted $\mathcal{S}^\ell$, is finite. We will show that we can compute the required threshold using the thresholds for the eventually constant activations along with the maximum rational number in the spectrum.

Let $A^\ell := A^\ell_{\text{in}} + A^\ell_{\text{out}}$. For $1 \leq \ell \leq L$, for $s_1, s_2, s_3 \in \mathcal{S}^{\ell-1}$, let $p^\ell_{s_1} := A^\ell s_1$ and $q^\ell_{s_2,s_3} := C^\ell s_2 + A^\ell s_3 + b^\ell$. For $1 \leq i \leq d_\ell$, we define $n^\ell_{s_1,s_2,s_3,i} \in \mathbb{N}$ as follows.

- If $\left(q^\ell_{s_1}\right)_i = 0$, then $n^\ell_{s_1,s_2,s_3,i} := 0$.
- If $\left(q^\ell_{s_1}\right)_i > 0$ and $\left(p^\ell_{s_2,s_3}\right)_i \geq t^\ell_{\mathsf{right}}$, then $n^\ell_{s_1,s_2,s_3,i} := 0$.
- If $\left(q^\ell_{s_1}\right)_i > 0$ and $\left(p^\ell_{s_2,s_3}\right)_i < t^\ell_{\mathsf{right}}$, then $n^\ell_{s_1,s_2,s_3,i} := \left\lceil \frac{t^\ell_{\mathsf{right}} - \left(p^\ell_{s_2,s_3}\right)_i}{\left(q^\ell_{s_1}\right)_i} \right\rceil$.
- If $\left(q^\ell_{s_1}\right)_i < 0$ and $\left(p^\ell_{s_2,s_3}\right)_i \leq t^\ell_{\mathsf{left}}$, then $n^\ell_{s_1,s_2,s_3,i} := 0$.
- If $\left(q^\ell_{s_1}\right)_i < 0$ and $\left(p^\ell_{s_2,s_3}\right)_i > t^\ell_{\mathsf{left}}$, then $n^\ell_{s_1,s_2,s_3,i} := \left\lceil \frac{\left(p^\ell_{s_2,s_3}\right)_i - t^\ell_{\mathsf{left}}}{-\left(q^\ell_{s_1}\right)_i} \right\rceil$.

Let $n_\mathcal{A}$ be the maximum of $n^\ell_{s_1,s_2,s_3,i}$. Since $\mathcal{S}^\ell$ has finite size, a maximum value exists.

We prove the lemma by induction on the layers of $\mathcal{A}$. For the base case $\ell = 0$, the properties hold by the definition of bipolar graphs. For the induction step $1 \leq \ell \leq L$, we first compute the feature functions for v_0 and v'_0. Note that by the induction hypothesis, $\xi^{\ell-1}_\mathcal{G}(v_0) = \xi^{\ell-1}_{\mathcal{G}'}(v'_0)$, $\xi^{\ell-1}_\mathcal{G}(v_1) = \xi^{\ell-1}_{\mathcal{G}'}(v'_1)$, and $\xi^{\ell-1}_\mathcal{G}(v_2) = \xi^{\ell-1}_{\mathcal{G}'}(v'_2)$. Therefore

$$\begin{aligned}\xi^\ell_\mathcal{G}(v_0) &= f^\ell\left(C^\ell \xi^{\ell-1}_\mathcal{G}(v_0) + A^\ell(\xi^{\ell-1}_\mathcal{G}(v_1) + \xi^{\ell-1}_\mathcal{G}(v_2)) + b^\ell\right)\\ &= f^\ell\left(C^\ell \xi^{\ell-1}_{\mathcal{G}'}(v'_0) + A^\ell(\xi^{\ell-1}_{\mathcal{G}'}(v'_1) + \xi^{\ell-1}_{\mathcal{G}'}(v'_2)) + b^\ell\right) \;=\; \xi^\ell_{\mathcal{G}'}(v'_0).\end{aligned}$$

We can check $\xi^\ell_\mathcal{G}(v_{1,i})$, $\xi^\ell_{\mathcal{G}'}(v'_{1,i'})$, $\xi^\ell_\mathcal{G}(v_{2,i})$, and $\xi^\ell_{\mathcal{G}'}(v'_{2,i'})$ similarly.

For $\xi^\ell_\mathcal{G}(v_1)$ and $\xi^\ell_{\mathcal{G}'}(v'_1)$, by the induction hypothesis, $\xi^{\ell-1}_\mathcal{G}(v_0) = \xi^{\ell-1}_{\mathcal{G}'}(v'_0)$; $\xi^{\ell-1}_\mathcal{G}(v_1) = \xi^{\ell-1}_{\mathcal{G}'}(v'_1)$; for $1 \leq i \leq n_\mathcal{A}$ and $1 \leq j \leq n_1$, $\xi^{\ell-1}_\mathcal{G}(v_{1,1}) = \xi^{\ell-1}_\mathcal{G}(v_{1,i}) = \xi^{\ell-1}_{\mathcal{G}'}(v'_{1,j})$. We can rewrite $\xi^\ell_\mathcal{G}(v_1)$ and $\xi^\ell_{\mathcal{G}'}(v'_1)$ as follows:

$$\begin{aligned}\xi^\ell_\mathcal{G}(v_1) &= f^\ell\left(C^\ell \xi^{\ell-1}_\mathcal{G}(v_1) + A^\ell\left(\xi^{\ell-1}_\mathcal{G}(v_0) + \sum_{1 \leq i \leq n_\mathcal{A}} \xi^{\ell-1}_\mathcal{G}(v_{1,i})\right) + b^\ell\right)\\ &= f^\ell\left(\left(C^\ell \xi^{\ell-1}_\mathcal{G}(v_1) + A^\ell \xi^{\ell-1}_\mathcal{G}(v_0) + b^\ell\right) + n_\mathcal{A}\left(A^\ell \xi^{\ell-1}_\mathcal{G}(v_{1,1})\right)\right)\\ &= f^\ell\left(p^\ell_{\xi^{\ell-1}_\mathcal{G}(v_1), \xi^{\ell-1}_\mathcal{G}(v_0)} + n_\mathcal{A} q^\ell_{\xi^{\ell-1}_\mathcal{G}(v_{1,1})}\right)\\ \xi^\ell_{\mathcal{G}'}(v'_1) &= f^\ell\left(p^\ell_{\xi^{\ell-1}_\mathcal{G}(v_1), \xi^{\ell-1}_\mathcal{G}(v_0)} + n_1 q^\ell_{\xi^{\ell-1}_\mathcal{G}(v_{1,1})}\right).\end{aligned}$$

Note that $\xi^{\ell-1}_\mathcal{G}(v_{1,1}), \xi^{\ell-1}_\mathcal{G}(v_1), \xi^{\ell-1}_\mathcal{G}(v_0) \in \mathcal{S}^{\ell-1}$. By the definition of $n_\mathcal{A}$,

$$n_\mathcal{A} \geq n^\ell_{\xi^{\ell-1}_\mathcal{G}(v_{1,1}), \xi^{\ell-1}_\mathcal{G}(v_1), \xi^{\ell-1}_\mathcal{G}(v_0)}.$$

Let $q = q^\ell_{\xi^{\ell-1}_\mathcal{G}(v_{1,1})}$ and $p = p^\ell_{\xi^{\ell-1}_\mathcal{G}(v_1), \xi^{\ell-1}_\mathcal{G}(v_0)}$. For $1 \leq i \leq d_\ell$, we consider the following cases.

- If $q_i = 0$, then $\xi^\ell_{\mathcal{G},i}(v_1) = f^\ell(p_i) = \xi^\ell_{\mathcal{G}',i}(v'_1)$.
- If $q_i > 0$ and $p_i \geq t^\ell_{\mathsf{right}}$, then $p_i + n_\mathcal{A} q_i \geq p_i \geq t^\ell_{\mathsf{right}}$ and $p_i + n_1 q_i \geq p_i \geq t^\ell_{\mathsf{right}}$. Hence $\xi^\ell_{\mathcal{G},i}(v_1) = f^\ell(t^\ell_{\mathsf{right}}) = \xi^\ell_{\mathcal{G}',i}(v'_1)$.
- If $q_i > 0$ and $p_i < t^\ell_{\mathsf{right}}$, by the definition of $n_\mathcal{A}$, $n_1 \geq n_\mathcal{A} \geq \left\lceil \frac{t^\ell_{\mathsf{right}} - p_i}{q_i} \right\rceil$, which implies that $p_i + n_\mathcal{A} q_i \geq t^\ell_{\mathsf{right}}$ and $p_i + n_1 q_i \geq t^\ell_{\mathsf{right}}$. Hence $\xi^\ell_{\mathcal{G},i}(v_1) = f^\ell(t^\ell_{\mathsf{right}}) = \xi^\ell_{\mathcal{G}',i}(v'_1)$.
- If $q_i < 0$ and $p_i \leq t^\ell_{\mathsf{left}}$, then $p_i + n_\mathcal{A} q_i \leq p_i \leq t^\ell_{\mathsf{left}}$ and $p_i + n_1 q_i \leq p_i \leq t^\ell_{\mathsf{left}}$. Hence $\xi^\ell_{\mathcal{G},i}(v_1) = f^\ell(t^\ell_{\mathsf{left}}) = \xi^\ell_{\mathcal{G}',i}(v'_1)$.
- If $q_i < 0$ and $p_i > t^\ell_{\mathsf{left}}$, by the definition of $n_\mathcal{A}$, $n_1 \geq n_\mathcal{A} \geq \left\lceil \frac{p_i - t^\ell_{\mathsf{left}}}{-q_i} \right\rceil$, which implies that $p_i + n_\mathcal{A} q_i \leq t^\ell_{\mathsf{left}}$ and $p_i + n_1 q_i \leq t^\ell_{\mathsf{left}}$. Hence $\xi^\ell_{\mathcal{G},i}(v_1) = f^\ell(t^\ell_{\mathsf{left}}) = \xi^\ell_{\mathcal{G}',i}(v'_1)$.

Thus $\xi^{\ell}_{\mathcal{G}}(v_1) = \xi^{\ell}_{\mathcal{G}'}(v'_1)$.

Finally, $\xi^{\ell}_{\mathcal{G}}(v_2)$ and $\xi^{\ell}_{\mathcal{G}'}(v'_2)$ can be treated analogously. ◀

We are now ready to show that there is a $\mathcal{L}$-M2P^2 formula that is not captured by a $\mathcal{BLC}$-GNN, which is the main claim of Lemma 45:

Proof. Consider the following 2-$\mathcal{L}$-M2P^2 formula:

$$\Psi(x) \ := \ (1 \cdot \#_{z,y}[E(x,z) \wedge E(z,y) \wedge U_1(y)] + (-1) \cdot \#_{z,y}[E(x,z) \wedge E(z,y) \wedge U_2(y)] = 0)$$

Let $\mathcal{G}$ be the (n_1, n_2)-bipolar graph. It is routine to check that $\mathcal{G} \models \Psi(v_0)$ if and only if $n_1 = n_2$.

For every L-layer 2-$\mathcal{BLC}$-GNN $\mathcal{A}$, let $n_{\mathcal{A}}$ be the constant defined in Lemma 74. Let $\mathcal{G}$ be the $(n_{\mathcal{A}}, n_{\mathcal{A}})$-bipolar graph and $\mathcal{G}'$ be the $(n_{\mathcal{A}}, n_{\mathcal{A}}+1)$-bipolar graph. By Lemma 74, $\xi^{L}_{\mathcal{G},1}(v_0) = \xi^{L}_{\mathcal{G}',1}(v'_0)$, which implies that $\langle \mathcal{G}, v_0 \rangle$ and $\langle \mathcal{G}', v'_0 \rangle$ are indistinguishable by $\mathcal{A}$. On the other hand, $\mathcal{G} \models \Psi(v_0)$ but $\mathcal{G}' \not\models \Psi(v'_0)$, they are distinguishable by $\Psi(x)$. Therefore $\mathcal{A}$ and $\Psi(x)$ are not equivalent. ◀

We can now prove the inclusion in Lemma 45. We know that $\mathcal{BLC}$-GNNs are equivalent in expressiveness to $\mathcal{L}$-MP^2. So it suffices to show that $\mathcal{L}$-MP^2 is subsumed in expressiveness by $\mathcal{L}$-M2P^2. But note that each Presburger quantifier is also a two-hop Presburger quantifier with no two-hop terms. Then it is obvious that $\mathcal{L}$-MP^2 is subsumed in expressiveness by $\mathcal{L}$-M2P^2.

F.7 Proof of Theorem 48: unbounded GNNs can express strictly more than eventually constant GNNs over undirected graphs

We recall the theorem.

▶ **Theorem 48.** $\mathcal{BL}$ReLU-GNN *is strictly more expressive over undirected graphs than* $\mathcal{BLC}$-GNN.

Recall that the bipolar graphs are undirected graphs. Hence by the same argument as in Lemma 45 we obtain the same reduction for undirected graphs.

▶ **Lemma 75.** *There exists a* $\mathcal{L}$-M2P^2 *formula* $\Psi(x)$ *such that for every* $\mathcal{BLC}$-GNN $\mathcal{A}$, $\Psi(x)$ *and* $\mathcal{A}$ *are not equivalent* over undirected graphs.

Theorem 48 is a direct consequence of the lemma above.