Logical Characterizations of Recurrent Graph Neural Networks with Reals and Floats

NEURAL INFORMATION PROCESSING SYSTEMS

Veeti Ahvonen[1] Damian Heiman[1] Antti Kuusisto[1] Carsten Lutz[2,3]

[1]Tampere University, [2]Leipzig University, [3]ScaDS.AI, Dresden/Leipzig

arXiv paper

https://arxiv.org/abs/2405.14606

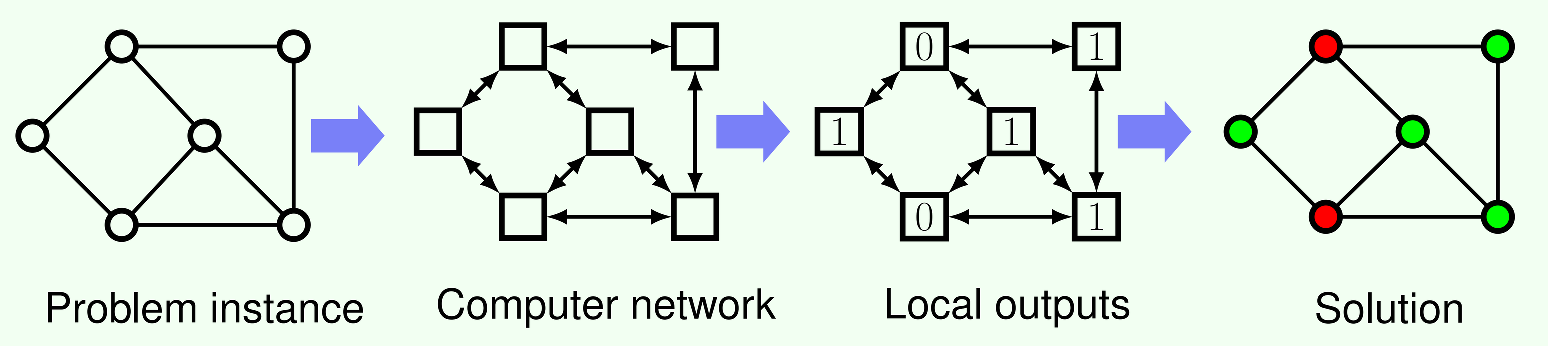
Overview

We give the first logical characterization of recurrent graph neural networks (GNNs) without restriction to a background logic. We cover scenarios with both real numbers and floating-point numbers, and analyze where they coincide.

Related work: Pioneering characterizations of constant-iteration graph neural networks were given in [1] and [2], and a characterization of recurrent graph neural networks was given in [3] w.r.t. a background logic.

Recurrent GNNs with reals and floats

A **recurrent graph neural network** ($\mathrm{GNN}[\mathbb{R}]$) receives a (directed, labeled) graph as input and gives a Boolean classification for each node in the graph.



A $\mathrm{GNN}[\mathbb{R}]$ is a tuple $(\boldsymbol{\pi}, \mathbf{AGG}, \mathbf{COM}, \boldsymbol{F})$, where

- an initialization function $\boldsymbol{\pi}$ maps each set of node label symbols to $\mathbb{R}^d$,
- $\mathbf{AGG}$: multisets$(\mathbb{R}^d) \to \mathbb{R}^d$ is an aggregation function,
- $\mathbf{COM}$: $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ is a combination function,
- $\boldsymbol{F} \subseteq \mathbb{R}^d$ is a set of accepting vectors.

The **feature vector** $\boldsymbol{x}_v^t \in \mathbb{R}^d$ of node v in round $t \in \mathbb{N}$ is computed as follows:

$\boldsymbol{x}_v^0 = \boldsymbol{\pi}(L)$, where L is the set of node labels of v.

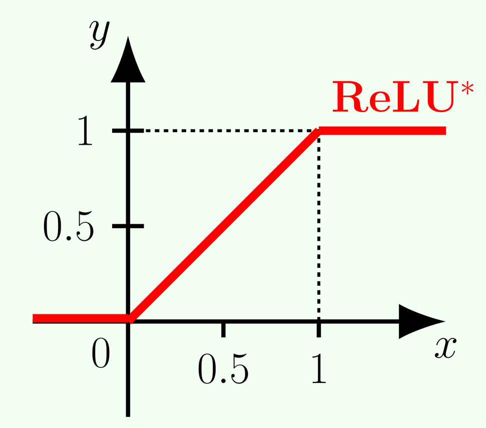
$$\boldsymbol{x}_v^{t+1} = \mathbf{COM}(\boldsymbol{x}_v^t, \mathbf{AGG}(\{\{\, \boldsymbol{x}_u^t \mid u \text{ is a neighbour of } v \,\}\})).$$

The classification for a node is "yes" if during the computation a feature vector of the node is in $\boldsymbol{F}$ and otherwise "no".

A **floating-point GNN** ($\mathrm{GNN}[\mathrm{F}]$) is the same as a $\mathrm{GNN}[\mathbb{R}]$, but it uses floating-point numbers instead of real numbers.

★ A floating-point GNN cannot distinguish arbitrarily large multisets because its calculations are finite!

R-simple GNNs



- Matrices $\boldsymbol{A}, \boldsymbol{C} \in \mathbb{R}^{d \times d}$
- A bias vector $\boldsymbol{b} \in \mathbb{R}^d$
- Truncated ReLU: $\mathbf{ReLU}^*(x) = \min(\max(0, x), 1)$

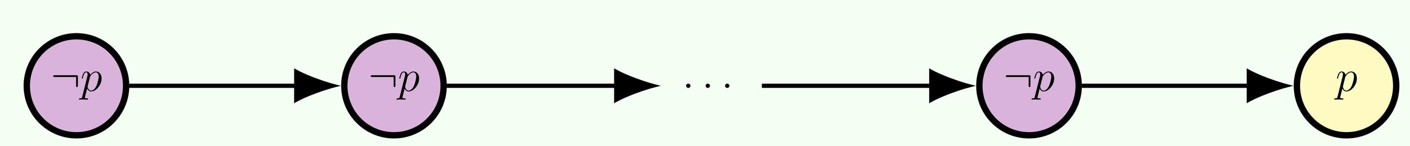
A new feature vector is computed as follows:

$$\boldsymbol{x}_v^{t+1} = \mathbf{ReLU}^*(\boldsymbol{x}_v^t \cdot \boldsymbol{C} + \left(\sum \boldsymbol{x}_u^t\right) \cdot \boldsymbol{A} + \boldsymbol{b}).$$

Example: reachability

A node label symbol p is **reachable** from a node v if there is a path from v to a node where p appears.

★ Reachability can be expressed even with an R-simple recurrent GNN but not with any constant-iteration GNN!



Logics GMSC and ω-GML

Graded modal substitution calculus (GMSC) is a logic consisting of programs. Each program consists of two lists of the below form:

$$\begin{array}{ll} X_1(0) :- p \land q & X_1 :- p \land X_3 \land \Diamond_{\geq 8} X_1 \\ X_2(0) :- q & X_2 :- \neg q \land \neg \Diamond_{\geq 2} X_1 \\ X_3(0) :- p & X_3 :- X_2 \land \Diamond_{\geq 3} \neg X_3 \end{array}$$

Base rules: $\varphi ::= p \mid \neg\varphi \mid \varphi \land \varphi \mid \Diamond_{\geq k}\varphi$

Induction rules: $\varphi ::= p \mid X \mid \neg\varphi \mid \varphi \land \varphi \mid \Diamond_{\geq k}\varphi$

A formula $\Diamond_{\geq k}\varphi$ states that φ is true in at least k neighbours.

We define iteration formulae X^t of variables X as follows.

- X^0 is the base rule of X,
- X^{t+1} is the induction rule of X where each Y is replaced with Y^t.

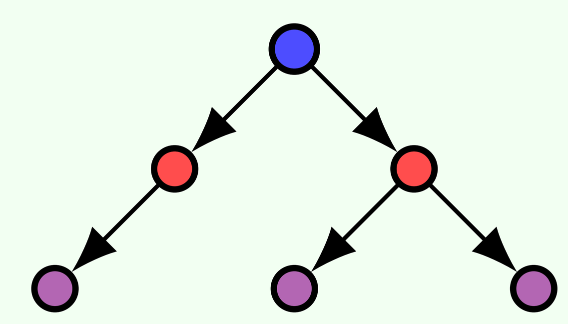
Each program also includes a set $\mathcal{A}$ of accepting variables. A graph node v is accepted if for any $X \in \mathcal{A}$ the iteration formula X^t is true in v for some $t \in \mathbb{N}$.

The **logic** ω-$\mathbf{GML}$ consists of countable disjunctions $\bigvee_{\varphi \in S} \varphi$ where each $\varphi \in S$ is a finite formula using the same grammar as the base rules above.

Example: centre-point property

A node v has the **centre-point property**, if every outgoing path starting from v leads to a dead-end in the *same number of steps*.

★ Expressible in GMSC but not in monadic second-order logic!



Program:

$X(0) :- \Box\bot \quad X :- \Box X \land \Diamond X$

Run of the program:

- $X^0 := \Box\bot$
- $X^1 := \Box X^0 \land \Diamond X^0$
- $X^2 := \Box X^1 \land \Diamond X^1$ (accepting)

...

Logical characterizations

Theorem 1

Same expressive power:

$$\mathrm{GNN}[\mathrm{F}]\text{s} \equiv \text{R-simple } \mathrm{GNN}[\mathrm{F}]\text{s} \equiv \mathrm{GMSC}$$

Theorem 2

Same expressive power:

$$\mathrm{GNN}[\mathbb{R}]\text{s} \equiv \omega\text{-}\mathrm{GML}$$

Theorem 3

In restriction to monadic second-order logic:

$$\mathrm{GMSC} \equiv \mathrm{GNN}[\mathrm{F}]\text{s} \equiv \text{R-simple } \mathrm{GNN}[\mathrm{F}]\text{s} \equiv \mathrm{GNN}[\mathbb{R}]\text{s} \equiv \omega\text{-}\mathrm{GML}$$

★ We also obtain characterizations for models of distributed computing.

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[2] Martin Grohe. The logic of graph neural networks. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–17. IEEE, 2021.

[3] Maximilian Pfluger, David Tena Cucala, and Egor V. Kostylev. Recurrent graph neural networks and their connections to bisimulation and logic. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):14608–14616, 2024.

Acknowledgements Veeti Ahvonen was supported by the Vilho, Yrjö and Kalle Väisälä Foundation of the Finnish Academy of Science and Letters. Damian Heiman was supported by the Magnus Ehrnrooth Foundation. Antti Kuusisto was supported by the Research Council of Finland consortium project Explaining AI via Logic (XAILOG), grant number 345612, and the Research Council of Finland project Theory of computational logics, grant numbers 352419, 352420, 353027, 324435, 328987; Damian Heiman was also supported by the same project, grant number 353027. Carsten Lutz was supported by the Federal Ministry of Education and Research of Germany (BMBF) and by Sächsisches Staatsministerium für Wissenschaft, Kultur und Tourismus in the program Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI. Lutz is also supported by BMBF in DAAD project 57616814 (SECAI, School of Embedded Composite AI) as part of the program Konrad Zuse Schools of Excellence in Artificial Intelligence.