Finite model theory and logics for AI

Grégoire Fournier

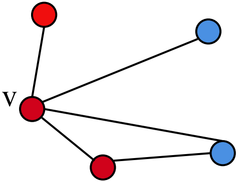
GSCL

March 3, 2025

Graph neural networks

Graph neural networks (GNNs) are deep learning architectures for machine learning problems on graphs.

$$\eta(v) := comb(\zeta(v), \sum_{w \in N_G(v)} \zeta(w))$$



$$\eta(v) := comb(\binom{1}{0}, \binom{2}{2})$$

The combination function *comb* is learned.

Bounded Variable Logic

Let $\mathcal{C}$ be the extension of first-order logic (FO) by counting quantifiers $\exists^{\geq p}$.

$\mathcal{C}_q$ consists of all formulas of $\mathcal{C}$ with at most q variables and $\mathcal{C}[k]$ formulas of quantifier rank at most k; $\mathcal{C}_q[k] = \mathcal{C}_q \cap \mathcal{C}[k]$.

We interpret formulas over labelled graphs; variables range over the vertices.

Example

For every k, a $FO_3[k]$ formula stating that the diameter of a graph is at most 2^k is:

$$\delta_{2^{k+1}}(x, y) = \exists z\ (\delta_{2^k}(x, z) \wedge \delta_{2^k}(z, y))$$

Guarded Counting Logic

The guarded fragment $\mathcal{GC}$ restricts quantifiers to range over the neighbours of the current nodes.

Example

The following $\mathcal{GC}_2$-formula $\Phi(x)$ says that vertex x has at most 1 neighbour that has more than 3 neighbours with label P_1:

$$\Phi(x) := \neg \, \exists^{\geq 2} y \, (E(x,y) \wedge \exists^{\geq 4} x \, (E(y,x) \wedge P_1(x)))$$

GNNs and the logic $\mathcal{GC}_2$

Theorem

Let Q be a unary formula expressible in graded modal logic $\mathcal{GC}_2$. Then there is a GNN that expresses Q.

Theorem (Barceló et al. (2020))

Let Q be a unary formula expressible by a GNN and also expressible in first-order logic. Then Q is expressible in $\mathcal{GC}_2$.

Open problem Grohe (2021): How to characterize the complexity of formulas produced by GNNs?

Invariants, colouring and the WL algorithm

Definition (Colouring algorithm (or 1-WL))

For every graph G, we define a sequence of vertex colourings $cr^{(t)}(G)$ as follows:

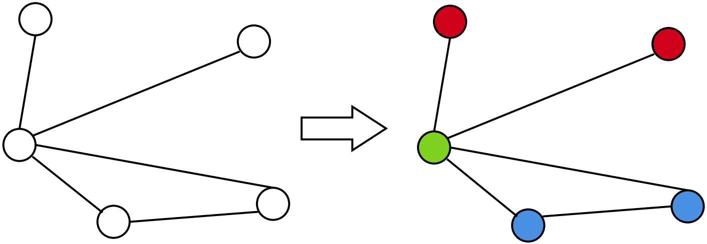
- For every $v \in V(G)$, let $cr^{(0)}(G, v) := col(G, v)$.

  $$cr^{(t+1)}(G, v) := (cr^{(t)}(G, v), \{\{cr^{(t)}(G, w) | w \in N(v)\}\})$$

- The stable colouring is denoted $cr^{(\infty)}(G)$.
- $cr^{(t)}$ is a vertex invariant.

In the same way we can define the k-dimensional WL algorithm ($wl_k^{(\infty)}$).

Invariants, colouring and the WL algorithm



$$cr^{(t+1)}(G, v) := (cr^{(t)}(G, v), \{\{cr^{(t)}(G, w) | w \in N(v)\}\})$$

Invariants, colouring and the WL algorithm

Theorem (Cai et al. (1989))

For all graphs G, G' tfae:

- $wl_k^{(\infty)}$ *does not distinguish G and G'.*
- *G and G' satisfy the same $\mathcal{C}_{k+1}$-sentences.*

Theorem

Let $t \geq 0$. Then for all graphs G, G' and vertices $v \in V(G), v' \in V(G')$ tfae:

- $cr^{(t)}(G, v) = cr^{(t)}(G', v')$;
- *for all formulas $\phi(x) \in \mathcal{GC}_2[t]$, $G \Vdash \phi(v)$ iff $G' \Vdash \phi(v')$.*

EF games

Definition (EF games)

The games is played on two relational structures $\mathfrak{A}$ and $\mathfrak{B}$, and it has two player, a spoiler and a duplicator. It goes as follows:

- For n rounds:
  - ⋆ The spoiler makes a move by picking an element of $\mathfrak{A}$ or $\mathfrak{B}$.
  - ⋆ The duplicator responds by picking an element in the other structure.
- The n-rounds game ends in the position $\vec{a} = (a_1, \ldots, a_n)$, $\vec{b} = (b_1, \ldots, b_n)$. Duplicator wins if:

  $((\vec{a}, \vec{c}^{\mathfrak{A}}), (\vec{b}, \vec{c}^{\mathfrak{B}}))$ is a partial isomorphism btw. $\mathfrak{A}$ and $\mathfrak{B}$.

EF games

Theorem

The following are equivalent:

- $\mathfrak{A}$ *and* $\mathfrak{B}$ *agree on* $FO[k]$.
- $\mathfrak{A} \equiv_k \mathfrak{B}$ *(duplicator has a winning strategy in* k*-round game).*

An application of EF games

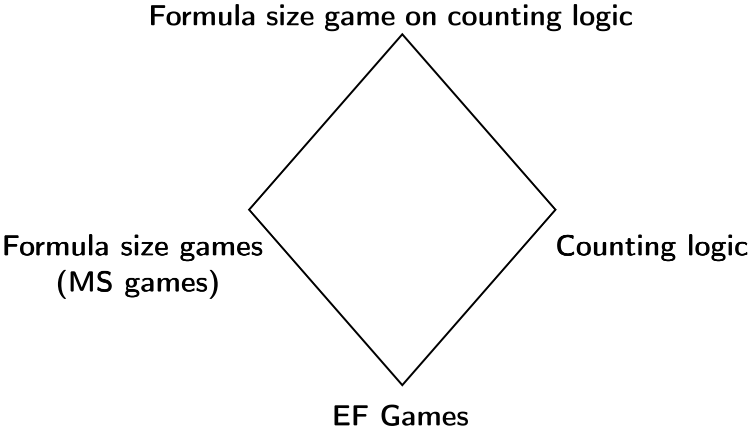
Theorem
Let $k > 0$, L_1, L_2 be linear orders of length at least 2^k, then $L_1 \equiv_k L_2$ and $L_1 \equiv_{FO[k]} L_2$.

Theorem
***(F)**:Let $q, k > 0$, and let L_1, L_2 be linear orders of length at least $(q+1)^k$, then $L_1 \equiv_{C^q[k]} L_2$.*

EF game theory and applications

- Pebble games to describe the quantifier rank of $\mathcal{L}_k$ and $\mathcal{C}_k$ Cai et al. (1989).
- EF game for formula size, in logics such as $\mathrm{FO}_2(\mathrm{TC})$ Adler and Immerman (2003), predicate FO Hella and Väänänen (2012), application to linear order of the FO game Grohe and Schweikardt (2005).
- ***(F)***: EF game characterisation of formula size, on $\mathcal{C}$ with bounded variables.
- ***(F)***: Application to linear orders of the EF game characterisation of formula size on $\mathcal{C}$.

EF game for formula size on counting logic



A Variant of EF games on $\mathcal{C}$ for formula size

We define three levels of sets of structures for the game:

- The tuples $(\mathcal{A}, \alpha)$ of structure and assignment are interpretations **(EF)**.
- We call families the sets of interpretations (written A) **(Counting)**.
- Finally tribes are sets of families, we have two, one on each side of the game (written A^+) **(MS games)**.

A Variant of EF games on $\mathcal{C}$ for formula size

We define two main operations:

$$\text{Big Update}: A^+(F^k_{A^+}/j) := \{A(F^k/j) : A \in A^+, F^k = F^k_{A^+}(A)\}$$

$$\text{Multiply}: A^+(*^k/j) := \{A(F^k, j) : A \in A^+, F^k \in \mathcal{F}^k_A\}$$

Where $\mathcal{F}^k_A \ni F^k$ is the set of all k-choice functions on A.

A Variant of EF games on $\mathcal{C}$ for formula size

Suppose the position after m moves the position is (w, L, A^+, B^+) where $dom(A^+) = dom(B^+)$.

The left k-supplementing move $(\exists^k)$ goes as follows:

Player I chooses two natural numbers $j \in [n], k \in \mathbb{N}^*$ and a k-choice function F^k for A^+, then the game continues from the position:

$$(w - 1, L, A^+(F^k/j), B^+(*^k/j))$$

.

Succinctness on linear orders

Theorem (Grohe and Schweikardt (2005))

Let $\mathcal{A}_m$ and $\mathcal{A}_n$ be two linear orders of size $n > m > 0$.

$\mathcal{A}_m$ and $\mathcal{A}_n$ cannot be distinguished by an $FO_3 - (<, succ, min, max)$ sentence of size less than $\frac{\sqrt{m}}{2}$.

Theorem (***F***)

$\mathcal{A}_m$ and $\mathcal{A}_n$ cannot be distinguished by an $\mathcal{C}_3^{(k)} - (<, succ, min, max)$ sentence of size less than $\frac{\sqrt{m}}{k+1}$.

Back to the open problem

Open problem Grohe (2021): How to characterize the complexity of formulas produced by GNNs?

- ***(F)***: EF game characterisation of formula size, on $\mathcal{C}$ with bounded variables, and application to linear orders.
- ***(F)***: Application to linear orders of the EF game characterisation of formula size on $\mathcal{C}$.
- *Next step*: Application to graphs.

References

Adler, M. and Immerman, N. (2003). An n! lower bound on formula size.

Barceló, P., Kostylev, E. V., Monet, M., Pérez, J., Reutter, J., and Silva, J. P. (2020). The logical expressiveness of graph neural networks.

Cai, J.-Y., Furer, M., and Immerman, N. (1989). An optimal lower bound on the number of variables for graph identification.

Grohe, M. (2021). The logic of graph neural networks.

Grohe, M. and Schweikardt, N. (2005). The succinctness of first-order logic on linear orders.

Hella, L. and Väänänen, J. (2012). The size of a formula as a measure of complexity.

Model theory of finite structures

Theorem (Completeness)

ψ is a consequence of T iff ψ is formally provable from T.

For which we have the direct consequence:

Theorem

The set of logically valid sentences of first-order logic is recursively enumerable. (true in all structures, under all assignments)

But if we consider only finite models, this fails:

Theorem (Trakhtenbrot)

The set of sentences of first-order logic valid in all finite structures is not recursively enumerable.

Examples

Example

- There is a theory T such that T has no finite models, and every finite subset of T has a finite model.

Proof.

Consider $\{\lambda_n := \exists x_1 \dots x_n \bigwedge_{i \neq j} \neg(x_i = x_j) : n \in \mathbb{N}\}$. □

- An inexpressibility proof: Assume that $\sigma = \emptyset$, then *EVEN* is not FO definable.

Proof.

Suppose it is by Φ.
$T_1 = \{\Phi\} \cup \{\lambda_k | k > 0\}$
$T_2 = \{\neg\Phi\} \cup \{\lambda_k | k > 0\}$
By LS theorem, T_1 and T_2 have a countable model, $\mathfrak{A}_1$ and $\mathfrak{A}_2$ that are isomorphic and satisfy $\mathfrak{A}_1 \Vdash \Phi$ and $\mathfrak{A}_2 \Vdash \neg\Phi$. □

The idea behind EF games

A rank-k m-type of $\mathbf{a}$ over $\mathfrak{A}$ is $tp_k(\mathfrak{A}, \mathbf{a}) = \{\phi \in FO[k] \mid \mathfrak{A}| \Vdash \phi(\mathbf{a})\}$.

Theorem

- *For a finite relational vocabulary σ, the number of different rank-k m-types is finite.*
- *Let $T_1, \dots T_n$ enumerate all the rank-k m-types. There exist $FO[k]$ formulas $(\alpha_i(\boldsymbol{x}))_{i \leq n}$ such that:*
  - *For every $\mathfrak{A}$ and $\boldsymbol{a} \in A^m$, it is the case that $\mathfrak{A} \Vdash \alpha_i(\boldsymbol{x})$ iff $tp_k(\mathfrak{A}, \boldsymbol{a}) = T_i$ and,*
  - *Every $FO[k]$ formula $\phi(\boldsymbol{x})$ in m free variables is equivalent to a disjunction of some α_i's.*

In particular, two finite structures that agree on all FO sentences are isomorphic.

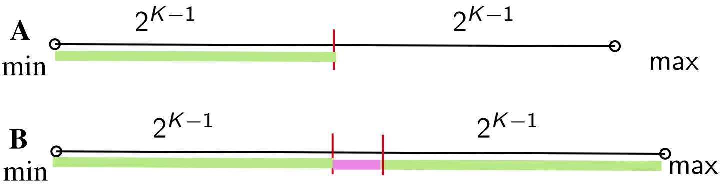
An application of EF games

Theorem
Let $k > 0$, L_1, L_2 be linear orders of length at least 2^k, then $L_1 \equiv_k L_2$ and $L_1 \equiv_{FO[k]} L_2$.

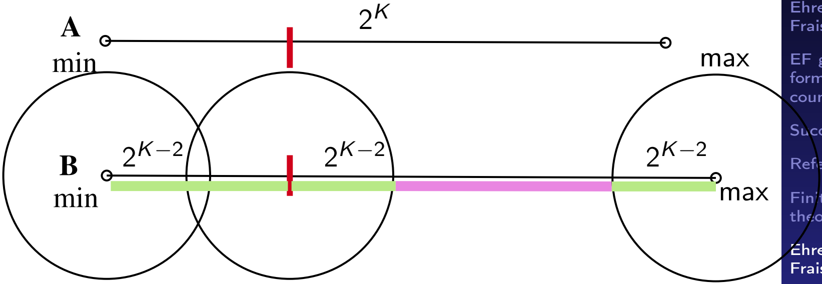
Recall $C_\ell[k]$ is the counting logic where the quantifier rank is k and the counting rank is ℓ.

Theorem
Let $c, k > 0$, and let L_1, L_2 be linear orders of length at least $(\ell + 1)^k$, then $L_1 \equiv_{C_\ell[k]} L_2$.

An application of EF games



An application of EF games



An application of EF games

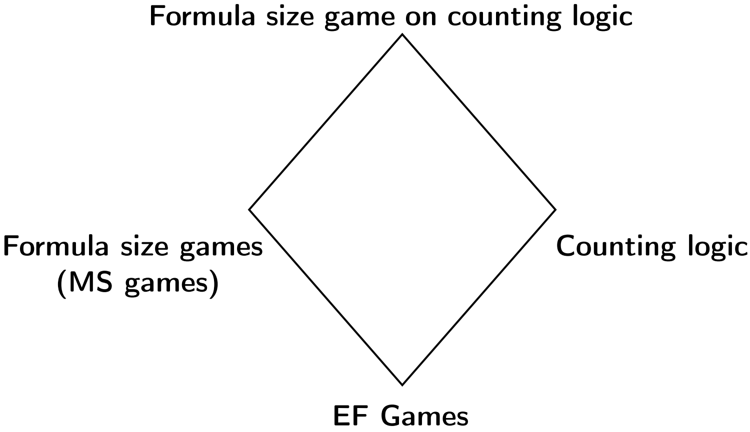
Proof.

We prove that $L_1 \equiv_k L_2$ in the expanded vocabulary with the constants *min* and *max*. Assume $L_1 = [n]$ and $L_2 = [m]$m with $n, m \geq 2^k + 1$. After round i, the moves are denoted $\vec{a} = (a_{-1}, a_0, a_1, \ldots, a_i)$ such that $(a_{-1}, a_0) = (min^{L_1}, max_1^{L})$, similarly for $\vec{b}$.
We claim that duplicator can maintain after move i, for $-1 \leq j, l \leq i$:

- If $d(a_j, a_l) < 2^{k-i}$, then $d(b_j, b_l) = d(a_j, a_l)$.
- If $d(a_j, a_l) \geq 2^{k-i}$, then $d(b_j, b_l) \geq 2^{k-i}$.
- $a_j \leq a_l$ iff $b_j \leq b_l$.

□

EF game for formula size on counting logic



EF games for counting logic

The games is played on two relational structures $\mathfrak{A}$ and $\mathfrak{B}$, and it has two player, a spoiler and a duplicator. It goes as follows:

- For n rounds:
  - Spoiler makes a move by picking a set A of $\mathfrak{A}$ or $\mathfrak{B}$. Duplicator selects a set B in the other structure.
  - Spoiler picks and element in B, duplicator selects an element in A.
- The n-rounds game ends in the position $\vec{a} = (a_1, \ldots, a_n)$, $\vec{b} = (b_1, \ldots, b_n)$. Duplicator wins if:

  $((\vec{a}, \vec{c}^{\mathfrak{A}}), (\vec{b}, \vec{c}^{\mathfrak{B}}))$ is a partial isomorphism btw. $\mathfrak{A}$ and $\mathfrak{B}$.

EF games for formula size

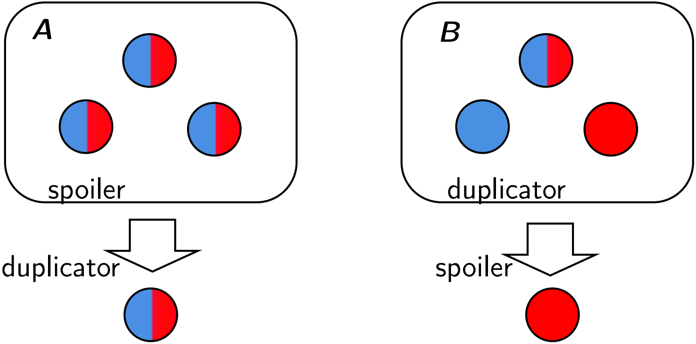
The game $\mathrm{EF}_w(A, B)$ has two players, where A and B are sets of structures. From a position (w, A, B).
There are four possibilities for the continuation of the game:

- ⋆ $\vee$ move: Player I first chooses $1 \leq u, v < w$ st. $u + v = w$.
  Then Player I represents A as $C \cup D$.
  The game continues from (u, C, B) or from (v, D, B) according to player II.
- ⋆ $\wedge$ move: similar but splits B.
- ⋆ $\exists^k$ move: Player I chooses a choice function F for A.
  Then the game continues from $(w - 1, A(F/j), B(*/j))$.
- ⋆ $\forall^k$ move: similar but chooses on A.

Player II wins the game if they reach a position $(1, A, B)$ and there is no atomic formula distinguishing A and B.

EF games for formula size and counting

No natural way to combine the two games, as seen in the following counter example:



The formula $\exists^{\geq 3} B(x) \wedge R(x)$ distinguished $\mathfrak{A}$ and $\mathfrak{B}$, but not $\exists^{\geq 3} B(x)$!