Finite Model Theory
Lecture 14: Weisfeiler-Leman and Logic

Spring 2025

Announcement

- HW4 (the last one!!) to be posted this week.
- Topic for today: WL, Logics, connection to GNNs. Lecture based on [Morris et al., 2023, Grohe, 2021, Morris et al., 2019, Grohe, 2020]
- Next week: descriptive complexity.

This and the previous lecture are based on [Morris et al., 2023, Grohe, 2021, Morris et al., 2019, Grohe, 2020]

Review

Review: Color Refinement

Iterative process that assigns a "color" to each node.

- Initially all nodes have the same color:

$$\boxed{\mathtt{cr}^0(v) \stackrel{\text{def}}{=} ()}$$

- For $t \geq 0$, assign colors based on the neighbors's colors:[1]

$$\boxed{\mathtt{cr}^{t+1}(v) \stackrel{\text{def}}{=} (\mathtt{cr}^t(v), \{\!\{\mathtt{cr}^t(w) \mid w \in N(v)\}\!\})}$$

- The stable coloring is $\mathtt{cr}^\infty \stackrel{\text{def}}{=} \mathtt{cr}^t$ where t is s.t. $\mathtt{cr}^{t+1} = \mathtt{cr}^t$.

[1] $\{\!\{\dots\}\!\}$ is a bag, as in $\{\!\{a, a, b, c, c, c\}\!\}$.

Color Refinement for Isomorphism Test

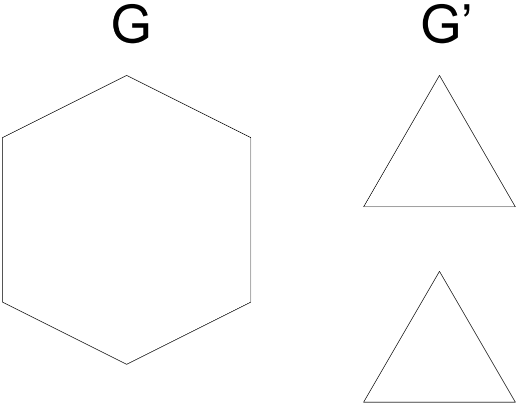
Input: G, G'

Take their disjoint union $G \cup G'$ and compute its stable color $\mathtt{cr}^\infty$.

If for any color, the two graphs a have different number of nodes of that color, then they are not isomorphic.

Other, we need to run an expensive isomorphism test.

Review: Color Refinement is Insufficient to Identify a Graph



All nodes have the same color, yet they are not isomorphic.

Color refinement fails to differentiate any two nodes in a regular graph.

Review: k-WL

Initially, each vector $\boldsymbol{v} \in V(G)^k$ is colored with $\mathtt{atp}_k(\boldsymbol{v})$.

At step $t+1$, the tuple $\boldsymbol{v}$ is colored with:

$$\boxed{\mathtt{wl}_k^{t+1}(\boldsymbol{v}) \stackrel{\text{def}}{=} (\mathtt{wl}_k^t(\boldsymbol{v}), M(\boldsymbol{v}))}$$

where M is:

$$\boxed{M(\boldsymbol{v}) \stackrel{\text{def}}{=} \{\!\{(\mathtt{atp}_{k+1}(\boldsymbol{v}w), \mathtt{wl}_k^t(\boldsymbol{v}[w/v_1]), \ldots, \mathtt{wl}_k^t(\boldsymbol{v}[w/v_k])) \mid w \in V(G)\}\!\}}$$

~~note that $\mathtt{atp}_{k+1}(\boldsymbol{v}w)$ considers all colors of $\mathtt{wl}_k^t$~~
(no need)

Review: Color Refinement is the Same as 1-WL

Fact

Color refinement and 1-WL coincide: $\forall t \geq 0,\ cr^t(v) = wl_1^t(v)$.

Review: Color Refinement is the Same as 1-WL

Fact

Color refinement and 1-WL coincide: $\forall\, t \geq 0,\ cr^t(v) = wl_1^t(v)$.

$\mathtt{cr}^{t+1}(v) =$
$(\mathtt{cr}^t(v), \{\!\{\mathtt{cr}^t(w) \mid w \in N(v)\}\!\})$

Review: Color Refinement is the Same as 1-WL

Fact

Color refinement and 1-WL coincide: $\forall t \geq 0,\ cr^t(v) = wl_1^t(v)$.

$\mathtt{cr}^{t+1}(v) = (\mathtt{cr}^t(v), \{\!\{\mathtt{cr}^t(w) \mid w \in N(v)\}\!\})$

$\mathtt{wl}_1^{t+1}(v) \stackrel{\text{def}}{=} (\mathtt{wl}_1^t(v), M(v))$, where

$M(v) \stackrel{\text{def}}{=} \{\!\{(\mathtt{atp}_2(vw), \mathtt{wl}_1^t(w)) \mid w \in V(G)\}\!\}$

Review: Color Refinement is the Same as 1-WL

Fact

Color refinement and 1-WL coincide: $\forall t \geq 0$, $cr^t(v) = wl_1^t(v)$.

$\mathtt{cr}^{t+1}(v) =$
$(\mathtt{cr}^t(v), \{\{\mathtt{cr}^t(w) \mid w \in N(v)\}\})$

$\mathtt{wl}_1^{t+1}(v) \stackrel{\text{def}}{=} (\mathtt{wl}_1^t(v), M(v))$, where
$M(v) \stackrel{\text{def}}{=} \{\{(\mathtt{atp}_2(vw), \mathtt{wl}_1^t(w)) \mid w \in V(G)\}\}$

M consists of three bags:

$v = w$: then $M_0(v) = \{\{(\mathtt{wl}_1^t(v))\}\}$

$vw \in E(G)$: then $M_1(v) = \{\{\mathtt{wl}_1^t(w) \mid w \in N(v)\}\}$

$vw \notin E(G)$: then $M_2(v) = \{\{\mathtt{wl}_1^t(w) \mid w \notin N(v)\}\}$

Review: Color Refinement is the Same as 1-WL

Fact

Color refinement and 1-WL coincide: $\forall t \geq 0,\ cr^t(v) = wl_1^t(v)$.

$\mathtt{cr}^{t+1}(v) =$
$(\mathtt{cr}^t(v), \{\!\{\mathtt{cr}^t(w) \mid w \in N(v)\}\!\})$

$\mathtt{wl}_1^{t+1}(v) \stackrel{\text{def}}{=} (\mathtt{wl}_1^t(v), M(v))$, where
$M(v) \stackrel{\text{def}}{=} \{\!\{(\mathtt{atp}_2(vw), \mathtt{wl}_1^t(w)) \mid w \in V(G)\}\!\}$

M consists of three bags:

$v = w$: then $M_0(v) = \{\!\{(\mathtt{wl}_1^t(v))\}\!\}$

$vw \in E(G)$: then $M_1(v) = \{\!\{\mathtt{wl}_1^t(w) \mid w \in N(v)\}\!\}$

$vw \notin E(G)$: then $M_2(v) = \{\!\{\mathtt{wl}_1^t(w) \mid w \notin N(v)\}\!\}$

Color refinement counts the colors of the neighbors.

1-WL counts both neighbors and non-neighbors. Same thing!

Discussion

- Color refinement (or 1-WL) provides a canonical labeling of the nodes of a random graph, with high probability. But fails completely on regular graphs.

- 2-WL differentiates two random regular graph with high probability. But fails completely on strongly regular graphs.

- Cai, Führer, Immerman [Cai et al., 1992] described a sequence of pairs of graphs G_k, H_k such that, for all k:
  - G_k, H_k are not isomorphic,
  - G_k, H_k are not distinguished by k-WL,
  - G_k, H_k are distinguished by $k+1$-WL.

- They also described the strong connection to logics. Next.

Counting Logic

Finite Variable Logic

Recall: FO^k is FO restricted to k variables $x_1, \ldots, x_k$

The crux is that we can reuse variables, e.g. check if there exists a path of length 100 in FO^2.

Finite Variable Logic

Recall: FO^k is FO restricted to k variables $x_1, \dots, x_k$

The crux is that we can reuse variables, e.g. check if there exists a path of length 100 in FO^2.

FO^k related to k-WL, but there is a catch:

- With 1-WL we can distinguish two nodes that have 100 and 101 neighbors.
- But in FO we need 101 variables.

Finite Variable Logic

Recall: FO^k is FO restricted to k variables $x_1, \ldots, x_k$

The crux is that we can reuse variables, e.g. check if there exists a path of length 100 in FO^2.

FO^k related to k-WL, but there is a catch:

- With 1-WL we can distinguish two nodes that have 100 and 101 neighbors.
- But in FO we need 101 variables.

Better: extend FO with counting quantifiers

Finite Variable Counting Logic

The logic C extends FO with a counting quantifier:

$$\boxed{\exists^{\geq n} x(\varphi(x))}$$

It means "there are at least n values x s.t. $\varphi(x)$ is true"

The logic C^k is C restricted to k variables.

Discussion

Example: $\exists^{\geq 3} x(\neg \exists^{\geq 2} y(E(x, y)))$

"There are at least 3 nodes with at most 1 neighbor"

Discussion

Example: $\exists^{\geq 3} x(\neg \exists^{\geq 2} y(E(x, y)))$
"There are at least 3 nodes with at most 1 neighbor"

C as expressive as FO, but C^k is more expressive than FO^k:

Example: $\exists^{\geq 3} x(\varphi(x))$ becomes
$$\exists x_1 \exists x_2 \exists x_3 (x_1 \neq x_2) \wedge (x_1 \neq x_3) \wedge (x_2 \neq x_3) \wedge \varphi(x_1) \wedge \varphi(x_2) \wedge \varphi(x_3)$$

Discussion

Example: $\exists^{\geq 3} x(\neg \exists^{\geq 2} y(E(x,y)))$

"There are at least 3 nodes with at most 1 neighbor"

C as expressive as FO, but C^k is more expressive than FO^k:

Example: $\exists^{\geq 3} x(\varphi(x))$ becomes

$$\exists x_1 \exists x_2 \exists x_3 (x_1 \neq x_2) \wedge (x_1 \neq x_3) \wedge (x_2 \neq x_3) \wedge \varphi(x_1) \wedge \varphi(x_2) \wedge \varphi(x_3)$$

Notice that, if $\varphi \in \mathrm{C}^k$, then so are $\exists^{=n}(\varphi)$ and $\exists^{\leq n}(\varphi)$.

$$\exists^{=n}(\varphi) = \exists^{\geq n}(\varphi) \wedge \neg \exists^{\geq n+1}(\varphi)$$

Examples

Which of the following C^2 sentences are true?

$$\forall x(\exists y(E(x,y) \land \exists x(\neg E(y,x))))$$

$$\forall x(\exists^{\geq 2} y(E(x,y) \land \exists x(\neg E(y,x))))$$

$$\forall x(\exists y(E(x,y) \land \exists^{\geq 2} x(\neg E(y,x))))$$

$$\forall x(\exists^{\geq 2} y(E(x,y) \land \exists^{\geq 2} x(\neg E(y,x))))$$

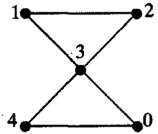


Fig. 1. An Undirected Graph

From [Cai et al., 1992]

Examples

Which of the following C^2 sentences are true?

$$\forall x(\exists y(E(x,y) \wedge \exists x(\neg E(y,x))))$$

TRUE

$$\forall x(\exists^{\geq 2} y(E(x,y) \wedge \exists x(\neg E(y,x))))$$

$$\forall x(\exists y(E(x,y) \wedge \exists^{\geq 2} x(\neg E(y,x))))$$

$$\forall x(\exists^{\geq 2} y(E(x,y) \wedge \exists^{\geq 2} x(\neg E(y,x))))$$

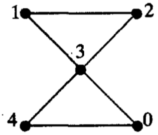


Fig. 1. An Undirected Graph

From [Cai et al., 1992]

Examples

Which of the following C^2 sentences are true?

$\forall x(\exists y(E(x,y) \wedge \exists x(\neg E(y,x))))$

TRUE

$\forall x(\exists^{\geq 2} y(E(x,y) \wedge \exists x(\neg E(y,x))))$

FALSE

$\forall x(\exists y(E(x,y) \wedge \exists^{\geq 2} x(\neg E(y,x))))$

$\forall x(\exists^{\geq 2} y(E(x,y) \wedge \exists^{\geq 2} x(\neg E(y,x))))$

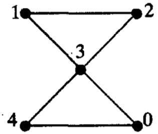


Fig. 1. An Undirected Graph

From [Cai et al., 1992]

Examples

Which of the following C^2 sentences are true?

$$\forall x(\exists y(E(x,y) \wedge \exists x(\neg E(y,x))))$$

TRUE

$$\forall x(\exists^{\geq 2} y(E(x,y) \wedge \exists x(\neg E(y,x))))$$

FALSE

$$\forall x(\exists y(E(x,y) \wedge \exists^{\geq 2} x(\neg E(y,x))))$$

TRUE

$$\forall x(\exists^{\geq 2} y(E(x,y) \wedge \exists^{\geq 2} x(\neg E(y,x))))$$

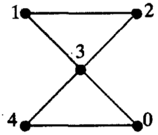


Fig. 1. An Undirected Graph

From [Cai et al., 1992]

Examples

Which of the following C^2 sentences are true?

$$\forall x(\exists y(E(x,y) \wedge \exists x(\neg E(y,x))))$$

TRUE

$$\forall x(\exists^{\geq 2} y(E(x,y) \wedge \exists x(\neg E(y,x))))$$

FALSE

$$\forall x(\exists y(E(x,y) \wedge \exists^{\geq 2} x(\neg E(y,x))))$$

TRUE

$$\forall x(\exists^{\geq 2} y(E(x,y) \wedge \exists^{\geq 2} x(\neg E(y,x))))$$

FALSE

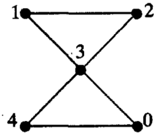


Fig. 1. An Undirected Graph

From [Cai et al., 1992]

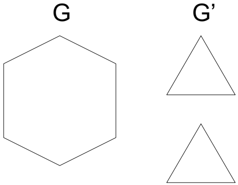
Connection Between WL and Counting Logic

Theorem ([Cai et al., 1992])

Two graphs are C^{k+1}-equivalent iff they cannot be distinguished by k-WL.

Color refinement cannot distinguish two graphs that are C^2-equivalent.

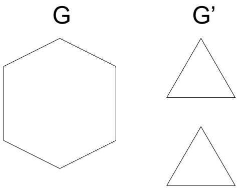
Example



They cannot be distinguished by C^2.

But they can be distinguished by C^3.
What φ distinguishes G, G'???

Example



They cannot be distinguished by C^2.

But they can be distinguished by C^3.
What φ distinguishes G, G'???

$\exists x \exists y \exists z (E(xy) \wedge E(yz) \wedge \neg E(xz))$

Discussion

- As stated, the equivalence between k-WL and C^{k+1} is only about properties of graphs.
- By looking closer at the details we can characterize their equivalence in terms of what properties of nodes they express.
- But first let's talk about applications to ML.

Graph Kernels and Graph Neural Networks

Graph Kernels

From [Grohe, 2020]
We are processing a large collection of graphs, denoted χ, and need to compare the similarity/distance of two graphs in χ.

One possibility is to compute an embedding of each graph in $\mathbb{R}^d$, then use the similarity/distance in $\mathbb{R}^d$.

Some machine learning algorithm do not need to ever compute the embedding, but only compute similarity: this is motivation behind a kernel

Graph Kernels

Fix a set of objects χ, e.g. the set of all graphs.

A kernel function is $K : \chi \times \chi \to \mathbb{R}$ that is:

- Symmetric: $K(x, y) = K(y, x)$
- Positive semidefinite: for any finite set $x_1, \dots, x_n$, the matrix $M_{ij} = K(x_i, x_j)$ is positive semidefinite.
  Recall, this means: $\boldsymbol{z}^T M \boldsymbol{z} \geq 0$ for all $\boldsymbol{z} \in \mathbb{R}^n$.

Graph Kernels

Fix a set of objects χ, e.g. the set of all graphs.

A kernel function is $K : \chi \times \chi \to \mathbb{R}$ that is:

- Symmetric: $K(x, y) = K(y, x)$
- Positive semidefinite: for any finite set $x_1, \ldots, x_n$, the matrix $M_{ij} = K(x_i, x_j)$ is positive semidefinite.

  Recall, this means: $\boldsymbol{z}^T M \boldsymbol{z} \geq 0$ for all $\boldsymbol{z} \in \mathbb{R}^n$.

Canonical example: given an embedding $f : \chi \to \mathbb{R}^d$, then the function $K(x, y) \stackrel{\text{def}}{=} < f(x), f(y) >$ is a kernel function.

A form of converse also holds (replace $\mathbb{R}^d$ with a Hilbert space).

Review: Colors as Tree Unfoldings

One can view the set of colors at round t as the set of all trees of depths t.
Example from [Grohe, 2020]:

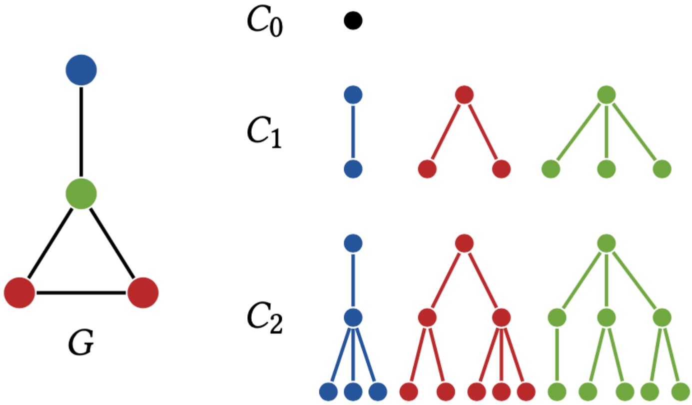


Figure 5: Viewing colours of WL as trees

Weisfeiler-Leman Graph Kernels

Let C_t be the set of trees of depth t. Infinite!

For any "color" $c \in C_t$: $\mathtt{wl}(c, G) \stackrel{\text{def}}{=}$ number of $v \in V(G)$ colored with c

Weisfeiler-Leman Graph Kernels

Let C_t be the set of trees of depth t. Infinite!

For any "color" $c \in C_t$: $\texttt{wl}(c, G) \stackrel{\text{def}}{=}$ number of $v \in V(G)$ colored with c

The *t*-round WL-kernel is:

$$K_{WL}^t(G, H) = \sum_{i=0,t} \sum_{c \in C_t} \texttt{wl}(c, G) \cdot \texttt{wl}(c, H)$$

Weisfeiler-Leman Graph Kernels

Let C_t be the set of trees of depth t. Infinite!

For any “color” $c \in C_t$: $\mathtt{wl}(c, G) \stackrel{\text{def}}{=}$ number of $v \in V(G)$ colored with c

The t-round WL-kernel is:

$$K^t_{WL}(G, H) = \sum_{i=0,t} \sum_{c \in C_t} \mathtt{wl}(c, G) \cdot \mathtt{wl}(c, H)$$

Although C_t is infinite, there are only finitely many non-zero terms.

A variant is: $\sum_{i=0,t} \frac{1}{2^i} \sum_{c \in C_t} \mathtt{wl}(c, G) \cdot \mathtt{wl}(c, H)$

Graph Neural Networks

From [Grohe, 2020]

Given a fixed graph, the goal is to compute a node embedding that maps each node v to $\boldsymbol{x}_v \in \mathbb{R}^d$ such that the distance/similarity between two nodes u, v is approximated by the distance/similarity between $\boldsymbol{x}_u, \boldsymbol{x}_v$.

Graph Neural Networks

From [Grohe, 2020]

Given a fixed graph, the goal is to compute a node embedding that maps each node v to $\boldsymbol{x}_v \in \mathbb{R}^d$ such that the distance/similarity between two nodes u, v is approximated by the distance/similarity between $\boldsymbol{x}_u, \boldsymbol{x}_v$.

GNNs are a method that uses deep learning techniques, while being independent of the graph size, and isomorphism invariant.

We will discuss only a very simple form of GNN

Graph Neural Networks

Each node $v \in V(G)$ has a state $\boldsymbol{x}_v \in \mathbb{R}^d$.

A GNN is defined by two matrices $W_{\text{AGG}}, W_{\text{UP}}$ and a function σ.
The computation of the GNN is:

$$\text{AGGREGATE:} \qquad \boldsymbol{a}_v^{t+1} \leftarrow \sum_{w \in N(v)} W_{\text{AGG}} \cdot \boldsymbol{x}_w^t$$

$$\text{UPDATE:} \qquad \boldsymbol{x}_v^{t+1} \leftarrow \sigma\left(W_{\text{UP}} \cdot \begin{pmatrix} \boldsymbol{x}_v^t \\ \boldsymbol{a}_v^{t+1} \end{pmatrix}\right)$$

Assume the initial configuration $(\boldsymbol{x}_v^0)_{v \in V(G)}$ is the same at all nodes.

We stop at some fixed iteration t.

GNNs and Color Refinement

Theorem

If $cr^\infty(u) = cr^\infty(v)$ then for any GNN parameters, $\mathbf{x}_u^t = \mathbf{x}_v^t$.

Thus, a GNN cannot distinguish more than color refinement.

Some converse statement is possible.

GNNs as Classifiers

Add classification function, $C : \mathbb{R}^d \to \{0, 1\}$; $\quad GNN(G, v) \stackrel{\text{def}}{=} C(\boldsymbol{x}_v^t)$.

The GNN distinguishes (G, v) and (G', v') if $GNN(G, v) \neq GNN(G', v')$.

GNNs as Classifiers

Add classification function, $C : \mathbb{R}^d \to \{0,1\}$; $\quad GNN(G,v) \stackrel{\text{def}}{=} C(\boldsymbol{x}_v^t)$.

The GNN distinguishes (G,v) and (G',v') if $GNN(G,v) \neq GNN(G',v')$.

Fact

If (G,v), (G',v') are C^2 equivalent, they are indistinguishable by GNN.

The converse fails, e.g. $\varphi(x) = \exists^{\geq 2} y (y = y)$. A GNN cannot distinguish between a graph with 1 node and a graph with 2 nodes and no edges.

GNNs as Classifiers

The guarded fragment of C^k restricts all quantifiers to range over neighbors:

$$\exists^{\geq n} y(E(x,y) \wedge \psi)$$

Theorem

GNNs and the guarded fragment of C^2 the same expressive power.

Discussion

- Connection between WL, GNN, and logics has been extensively researched in the last few years.
- GNNs gain more power by using a random initialization function, instead of a constant function.
- Many other variations of GNNs exists, and their connection to logics has also been explored.

A Detailed Look

Overview

Recall: two graphs are C^{k+1}-equivalent iff they cannot be distinguished by k-WL.

The quantifier depth of C^{k+1} does not correspond 1-to-1 to the step of the k-WL

Instead, the quantifier depth corresponds 1-to-1 to the step of a slight variant, called the oblivious WL.

Folklore v.s. Oblivious k-WL

What we discussed until now is called Folklore WL:

$$\boxed{\begin{aligned}&\mathtt{fwl}_k^{t+1}(\boldsymbol{v}) \stackrel{\text{def}}{=} (\mathtt{fwl}_k^t(\boldsymbol{v}), M(\boldsymbol{v}))\\ &M(\boldsymbol{v}) \stackrel{\text{def}}{=} \{\!\{(\mathtt{atp}_{k+1}(\boldsymbol{v}w), \mathtt{fwl}_k^t(\boldsymbol{v}[w/v_1]), \ldots, \mathtt{fwl}_k^t(\boldsymbol{v}[w/v_k])) \mid w \in V(G)\}\!\}\end{aligned}}$$

Folklore v.s. Oblivious k-WL

What we discussed until now is called Folklore WL:

$$\begin{aligned}&\mathtt{fwl}_k^{t+1}(\boldsymbol{v}) \stackrel{\text{def}}{=} (\mathtt{fwl}_k^t(\boldsymbol{v}), M(\boldsymbol{v}))\\&M(\boldsymbol{v}) \stackrel{\text{def}}{=} \{\{(\mathtt{atp}_{k+1}(\boldsymbol{v}w), \mathtt{fwl}_k^t(\boldsymbol{v}[w/v_1]), \ldots, \mathtt{fwl}_k^t(\boldsymbol{v}[w/v_k])) \mid w \in V(G)\}\}\end{aligned}$$

The following variant is called Oblivious WL:

$$\begin{aligned}&\mathtt{owl}_k^{t+1}(\boldsymbol{v}) \stackrel{\text{def}}{=} (\mathtt{owl}_k^t(\boldsymbol{v}), M(\boldsymbol{v}))\\&M(\boldsymbol{v}) \stackrel{\text{def}}{=} (\{\{\mathtt{owl}_k^t(\boldsymbol{v}[w/v_1]) \mid w \in V(G)\}\}, \ldots, \{\{\mathtt{owl}_k^t(\boldsymbol{v}[w/v_k]) \mid w \in V(G)\}\})\end{aligned}$$

Folklore v.s. Oblivious k-WL

What we discussed until now is called Folklore WL:

$$\texttt{fwl}_k^{t+1}(\boldsymbol{v}) \stackrel{\text{def}}{=} (\texttt{fwl}_k^t(\boldsymbol{v}), M(\boldsymbol{v}))$$
$$M(\boldsymbol{v}) \stackrel{\text{def}}{=} \{\!\{(\texttt{atp}_{k+1}(\boldsymbol{v}w), \texttt{fwl}_k^t(\boldsymbol{v}[w/v_1]), \ldots, \texttt{fwl}_k^t(\boldsymbol{v}[w/v_k])) \mid w \in V(G)\}\!\}$$

The following variant is called Oblivious WL:

$$\texttt{owl}_k^{t+1}(\boldsymbol{v}) \stackrel{\text{def}}{=} (\texttt{owl}_k^t(\boldsymbol{v}), M(\boldsymbol{v}))$$
$$M(\boldsymbol{v}) \stackrel{\text{def}}{=} (\{\!\{\texttt{owl}_k^t(\boldsymbol{v}[w/v_1]) \mid w \in V(G)\}\!\}, \ldots, \{\!\{\texttt{owl}_k^t(\boldsymbol{v}[w/v_k]) \mid w \in V(G)\}\!\})$$

Note: $\texttt{owl}_k$ makes sense only for $k \geq 2$:

For $k = 1$ it doesn't check whether the edge $v_1 w$ exists.

Connection Between Folklore and Oblivious WL

Theorem

(1) If G, G' are distinguished by $\mathtt{fwl}_k^t$ then they are distinguished by $\mathtt{owl}_{k+1}^t$
(2) If G, G' are distinguished by $\mathtt{owl}_{k+1}^t$ then they are distinguished by $\mathtt{fwl}_k^{t+1}$

Intuitively: the stable coloring $\mathtt{fwl}_k^\infty$ is the same as $\mathtt{owl}_{k+1}^\infty$.

Connection Between Folklore and Oblivious WL

Theorem

(1) If G, G' are distinguished by $\mathtt{fwl}_k^t$ then they are distinguished by $\mathtt{owl}_{k+1}^t$
(2) If G, G' are distinguished by $\mathtt{owl}_{k+1}^t$ then they are distinguished by $\mathtt{fwl}_k^{t+1}$

Intuitively: the stable coloring $\mathtt{fwl}_k^\infty$ is the same as $\mathtt{owl}_{k+1}^\infty$.

The proof in [Grohe, 2021] establishes the following equivalence:

- $\mathtt{owl}_{k+1}^t(G, \boldsymbol{v}) = \mathtt{owl}_{k+1}^t(G', \boldsymbol{v}')$
- $\mathrm{atp}_{k+1}(G, \boldsymbol{v}) = \mathrm{atp}_{k+1}(G', \boldsymbol{v}')$ and for all $i = 1, k+1$:
  $\mathtt{fwl}_k^t(G, v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{k+1}) = \mathtt{fwl}_k^t(G', v_1', \ldots, v_{i-1}', v_{i+1}', \ldots, v_{k+1}')$

Connection Between WL and Counting Logic

$\mathrm{C}^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \left(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \textit{ iff } G' \vDash \varphi[\boldsymbol{v}']\right)$$

Connection Between WL and Counting Logic

$\mathsf{C}^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v'} \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v'}) \iff \big(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \textit{ iff } G' \vDash \varphi[\boldsymbol{v'}]\big)$$

Proof Induction on q: When $q = 0$ both sides hold iff $\mathtt{atp}_k(G, \boldsymbol{v}) = \mathtt{atp}_k(G', \boldsymbol{v'})$.

Connection Between WL and Counting Logic

$C^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \left(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \textit{ iff } G' \vDash \varphi[\boldsymbol{v}']\right)$$

Assume it holds for q, we prove for $q + 1$.

Connection Between WL and Counting Logic

$\mathrm{C}^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \left(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \textit{ iff } G' \vDash \varphi[\boldsymbol{v}']\right)$$

Assume it holds for q, we prove for $q+1$.

The $\Rightarrow$ direction. Let $\varphi \in \mathrm{C}^k[q+1]$; assume w.l.o.g. $\varphi = \exists x_i \psi(x_1, \ldots, x_k)$.

Connection Between WL and Counting Logic

$\mathsf{C}^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \quad \Longleftrightarrow \quad \left(\forall \varphi \in C^k[q] : G \models \varphi[\boldsymbol{v}] \text{ iff } G' \models \varphi[\boldsymbol{v}']\right)$$

Assume it holds for q, we prove for $q+1$.

The $\Rightarrow$ direction. Let $\varphi \in \mathsf{C}^k[q+1]$; assume w.l.o.g. $\varphi = \exists x_i \psi(x_1, \ldots, x_k)$.

If $G \models \varphi[\boldsymbol{v}]$ then $\exists w \in V(G)$ such that $\boxed{G \models \psi[\boldsymbol{v}[v_i/w]]}$

Connection Between WL and Counting Logic

$\mathrm{C}^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \quad \Longleftrightarrow \quad \left(\forall \varphi \in C^k[q] : G \models \varphi[\boldsymbol{v}] \text{ iff } G' \models \varphi[\boldsymbol{v}']\right)$$

Assume it holds for q, we prove for $q+1$.

The $\Rightarrow$ direction. Let $\varphi \in \mathrm{C}^k[q+1]$; assume w.l.o.g. $\varphi = \exists x_i \psi(x_1, \ldots, x_k)$.

If $G \models \varphi[\boldsymbol{v}]$ then $\exists w \in V(G)$ such that $\boxed{G \models \psi[\boldsymbol{v}[v_i/w]]}$

$\mathtt{owl}_k^{q+1}(\boldsymbol{v}) = \mathtt{owl}_k^{q+1}(\boldsymbol{v}')$ implies $\exists w' \in V(G'), \mathtt{owl}_k^q(\boldsymbol{v}[w/v_i]) = \mathtt{owl}_k^q(\boldsymbol{v}'[w'/v_i'])$

Connection Between WL and Counting Logic

$C^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \big(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \text{ iff } G' \vDash \varphi[\boldsymbol{v}']\big)$$

Assume it holds for q, we prove for $q+1$.

The $\Rightarrow$ direction. Let $\varphi \in C^k[q+1]$; assume w.l.o.g. $\varphi = \exists x_i \psi(x_1, \ldots, x_k)$.

If $G \vDash \varphi[\boldsymbol{v}]$ then $\exists w \in V(G)$ such that $\boxed{G \vDash \psi[\boldsymbol{v}[v_i/w]]}$

$\mathtt{owl}_k^{q+1}(\boldsymbol{v}) = \mathtt{owl}_k^{q+1}(\boldsymbol{v}')$ implies $\exists w' \in V(G'), \mathtt{owl}_k^q(\boldsymbol{v}[w/v_i]) = \mathtt{owl}_k^q(\boldsymbol{v}'[w'/v_i'])$

By induction on q: $\boxed{G' \vDash \psi[\boldsymbol{v}'[v_i'/w']]}$ which implies $G' \vDash \exists x_i(\psi[\boldsymbol{v}'])$

Connection Between WL and Counting Logic

$\mathrm{C}^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \left(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \text{ iff } G' \vDash \varphi[\boldsymbol{v}']\right)$$

The $\Leftarrow$ direction.

Connection Between WL and Counting Logic

$C^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \quad \Longleftrightarrow \quad \left(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \text{ iff } G' \vDash \varphi[\boldsymbol{v}']\right)$$

The $\Leftarrow$ direction. To prove $\mathtt{owl}_k^{q+1}(\boldsymbol{v}) = \mathtt{owl}_k^{q+1}(\boldsymbol{v}')$ we need to check:

- $\mathtt{owl}_k^q(\boldsymbol{v}) = \mathtt{owl}_k^q(\boldsymbol{v}')$.
- $\{\!\{\mathtt{owl}_k^q(\boldsymbol{v}[w/v_i]) \mid w \in V(G)\}\!\} = \{\!\{\mathtt{owl}_k^q(\boldsymbol{v}'[w'/v_i']) \mid w' \in V(G')\}\!\}$, $i = 1, k$.

Connection Between WL and Counting Logic

$C^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \big(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \text{ iff } G' \vDash \varphi[\boldsymbol{v}']\big)$$

The $\Leftarrow$ direction. To prove $\mathtt{owl}_k^{q+1}(\boldsymbol{v}) = \mathtt{owl}_k^{q+1}(\boldsymbol{v}')$ we need to check:

- $\mathtt{owl}_k^q(\boldsymbol{v}) = \mathtt{owl}_k^q(\boldsymbol{v}')$. Holds by induction on q.
- $\{\{\mathtt{owl}_k^q(\boldsymbol{v}[w/v_i]) \mid w \in V(G)\}\} = \{\{\mathtt{owl}_k^q(\boldsymbol{v}'[w'/v_i']) \mid w' \in V(G')\}\}$, $i = 1, k$.

Connection Between WL and Counting Logic

$C^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \left(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \textit{ iff } G' \vDash \varphi[\boldsymbol{v}']\right)$$

The $\Leftarrow$ direction. To prove $\texttt{owl}_k^{q+1}(\boldsymbol{v}) = \texttt{owl}_k^{q+1}(\boldsymbol{v}')$ we need to check:

- $\texttt{owl}_k^q(\boldsymbol{v}) = \texttt{owl}_k^q(\boldsymbol{v}')$. Holds by induction on q.
- $\{\!\{\texttt{owl}_k^q(\boldsymbol{v}[w/v_i]) \mid w \in V(G)\}\!\} = \{\!\{\texttt{owl}_k^q(\boldsymbol{v}'[w'/v_i']) \mid w' \in V(G')\}\!\}$, $i = 1, k$.

  Fix a color c on the left, and let n_c be the number of its occurrences.

Connection Between WL and Counting Logic

$C^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \left(\forall \varphi \in C^k[q] : G \models \varphi[\boldsymbol{v}] \text{ iff } G' \models \varphi[\boldsymbol{v}']\right)$$

The $\Leftarrow$ direction. To prove $\mathtt{owl}_k^{q+1}(\boldsymbol{v}) = \mathtt{owl}_k^{q+1}(\boldsymbol{v}')$ we need to check:

- $\mathtt{owl}_k^q(\boldsymbol{v}) = \mathtt{owl}_k^q(\boldsymbol{v}')$. Holds by induction on q.
- $\{\!\{\mathtt{owl}_k^q(\boldsymbol{v}[w/v_i]) \mid w \in V(G)\}\!\} = \{\!\{\mathtt{owl}_k^q(\boldsymbol{v}'[w'/v_i']) \mid w' \in V(G')\}\!\}$, $i = 1, k$.

  Fix a color c on the left, and let n_c be the number of its occurrences.

  Let $\psi_c(x_1, \ldots, x_k) \stackrel{\text{def}}{=}$ the q, k-type (in C) of tuples $\boldsymbol{v}[w/v_i]$ colored c.

Connection Between WL and Counting Logic

$C^k[q]$ = quantifier depth $\le q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \left(\forall \varphi \in C^k[q] : G \models \varphi[\boldsymbol{v}] \text{ iff } G' \models \varphi[\boldsymbol{v}']\right)$$

The $\Leftarrow$ direction. To prove $\mathtt{owl}_k^{q+1}(\boldsymbol{v}) = \mathtt{owl}_k^{q+1}(\boldsymbol{v}')$ we need to check:

- $\mathtt{owl}_k^q(\boldsymbol{v}) = \mathtt{owl}_k^q(\boldsymbol{v}')$. Holds by induction on q.
- $\{\!\{\mathtt{owl}_k^q(\boldsymbol{v}[w/v_i]) \mid w \in V(G)\}\!\} = \{\!\{\mathtt{owl}_k^q(\boldsymbol{v}'[w'/v_i']) \mid w' \in V(G')\}\!\}$, $i = 1, k$.

  Fix a color c on the left, and let n_c be the number of its occurrences.

  Let $\psi_c(x_1, \ldots, x_k) \stackrel{\text{def}}{=}$ the q, k-type (in C) of tuples $\boldsymbol{v}[w/v_i]$ colored c.

  Then $G \models (\exists^{=n_c} x_i \psi_c(x_1, \ldots, x_k))[\boldsymbol{v}]$, thus $G' \models (\exists^{=n_c} x_i \psi_c(x_1, \ldots, x_k))[\boldsymbol{v}']$.

Connection Between WL and Counting Logic

$C^k[q]$ = quantifier depth $\leq q$. Fix vectors $\boldsymbol{v} \in V(G)^k, \boldsymbol{v}' \in (V(G'))^k$.

Theorem

$$owl_k^q(G, \boldsymbol{v}) = owl_k^q(G', \boldsymbol{v}') \iff \left(\forall \varphi \in C^k[q] : G \vDash \varphi[\boldsymbol{v}] \text{ iff } G' \vDash \varphi[\boldsymbol{v}']\right)$$

The $\Leftarrow$ direction. To prove $\mathtt{owl}_k^{q+1}(\boldsymbol{v}) = \mathtt{owl}_k^{q+1}(\boldsymbol{v}')$ we need to check:

- $\mathtt{owl}_k^q(\boldsymbol{v}) = \mathtt{owl}_k^q(\boldsymbol{v}')$. Holds by induction on q.
- $\{\!\{\mathtt{owl}_k^q(\boldsymbol{v}[w/v_i]) \mid w \in V(G)\}\!\} = \{\!\{\mathtt{owl}_k^q(\boldsymbol{v}'[w'/v_i']) \mid w' \in V(G')\}\!\}$, $i = 1, k$.

  Fix a color c on the left, and let n_c be the number of its occurrences.

  Let $\psi_c(x_1, \ldots, x_k) \stackrel{\text{def}}{=}$ the q, k-type (in C) of tuples $\boldsymbol{v}[w/v_i]$ colored c.

  Then $G \vDash (\exists^{=n_c} x_i \psi_c(x_1, \ldots, x_k))[\boldsymbol{v}]$, thus $G' \vDash (\exists^{=n_c} x_i \psi_c(x_1, \ldots, x_k))[\boldsymbol{v}']$.

  The color c also occurs n_c times on the right.

Discussion

- Color refinement was discovered multiple times in the past.
- Weisfeiler and Leman introduced 2-WL.
- Cai, Führer, Immerman established connection between k-WL, C^k, and certain EF-games. They also described the graphs G_k, H_k.
- There is a connection between GNNs, WL, C^k, however the exact/rigorous statement requires a careful examination of the details.

Cai, J., Fürer, M., and Immerman, N. (1992).
An optimal lower bound on the number of variables for graph identification.
Comb., 12(4):389–410.

Grohe, M. (2020).
word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data.
In Suciu, D., Tao, Y., and Wei, Z., editors, *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2020, Portland, OR, USA, June 14-19, 2020*, pages 1–16. ACM.

Grohe, M. (2021).
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.

Morris, C., Lipman, Y., Maron, H., Rieck, B., Kriege, N. M., Grohe, M., Fey, M., and Borgwardt, K. M. (2023).
Weisfeiler and leman go machine learning: The story so far.
J. Mach. Learn. Res., 24:333:1–333:59.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. (2019).
Weisfeiler and leman go neural: Higher-order graph neural networks.
In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4602–4609. AAAI Press.