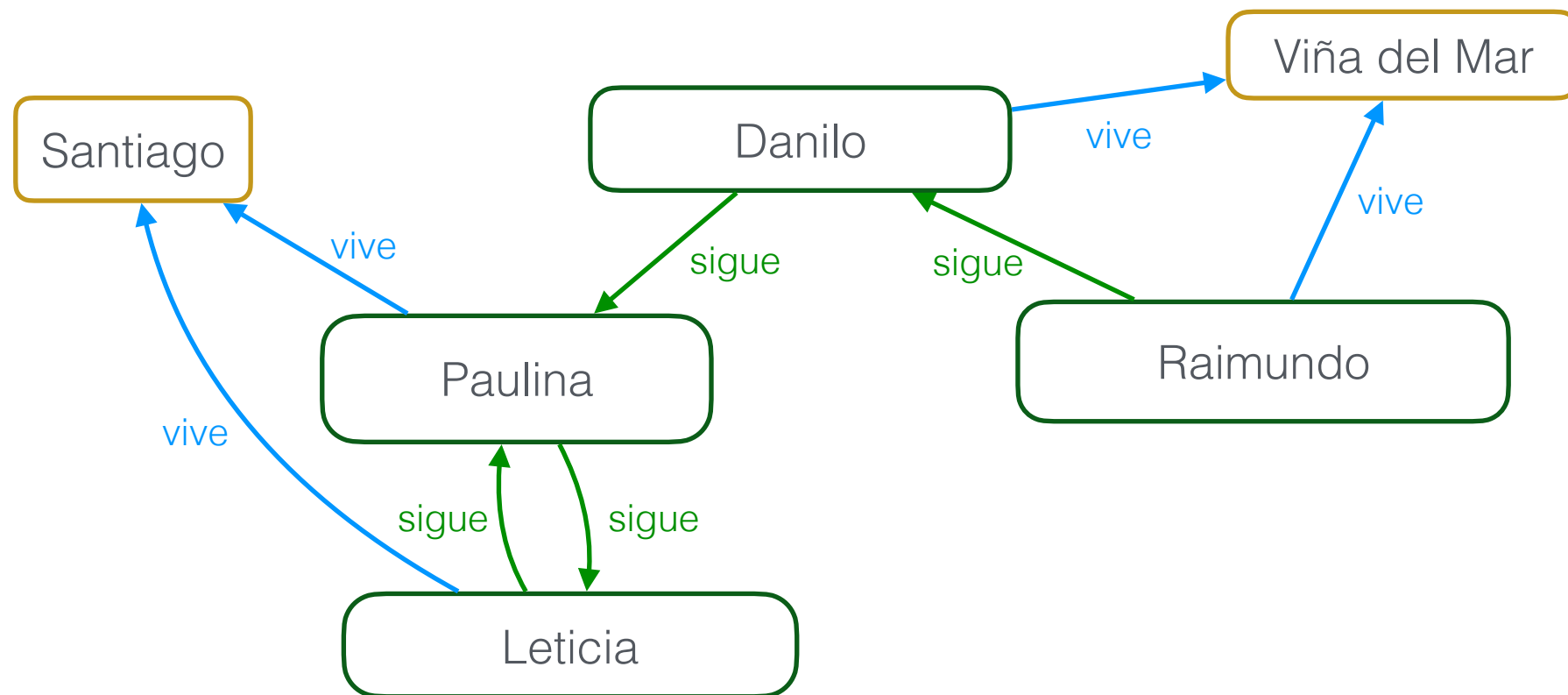


# Hasta el momento

- Nos hemos enfocado en GNNs y sus variantes para grafos no dirigidos
- Existen otros modelos de grafos relevantes en la práctica:
  - Grafos dirigidos
  - Grafos etiquetados o Knowledge graphs
- Veremos algunas versiones de GNNs para estos modelos

# Knowledge graphs

- Grafos con distintos tipos de relaciones
- Formalmente: un **knowledge graph** es un triple  $G = (V, E, R)$  donde
  - $V$  es un conjunto de nodos
  - $E$  es un conjunto de arcos o triples de la forma  $(u, r, v) \in V \times R \times V$
  - $R$  es un conjunto de tipos de relaciones



# R-GCN

- Introducido el 2018:  
Modeling Relational Data with Graph Convolutional Networks.  
M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling.
- Aplicado a clasificación de nodos y predicción de links
- Computación de una R-GCN: similar a la de una GNN clásica
  - Ahora la entrada es un knowledge graph + vectores iniciales en los nodos
  - La R-GCN tiene la capacidad de distinguir relaciones a la hora de agregar la información de los vecinos

# R-GCN: definición

Tenemos un knowledge graph  $G = (V, E, R)$

- Denotamos por  $R^\pm$  al conjunto

$$R^\pm = R \cup \{r^- \mid r \in R\}$$

(conjunto de las relaciones y sus inversos)

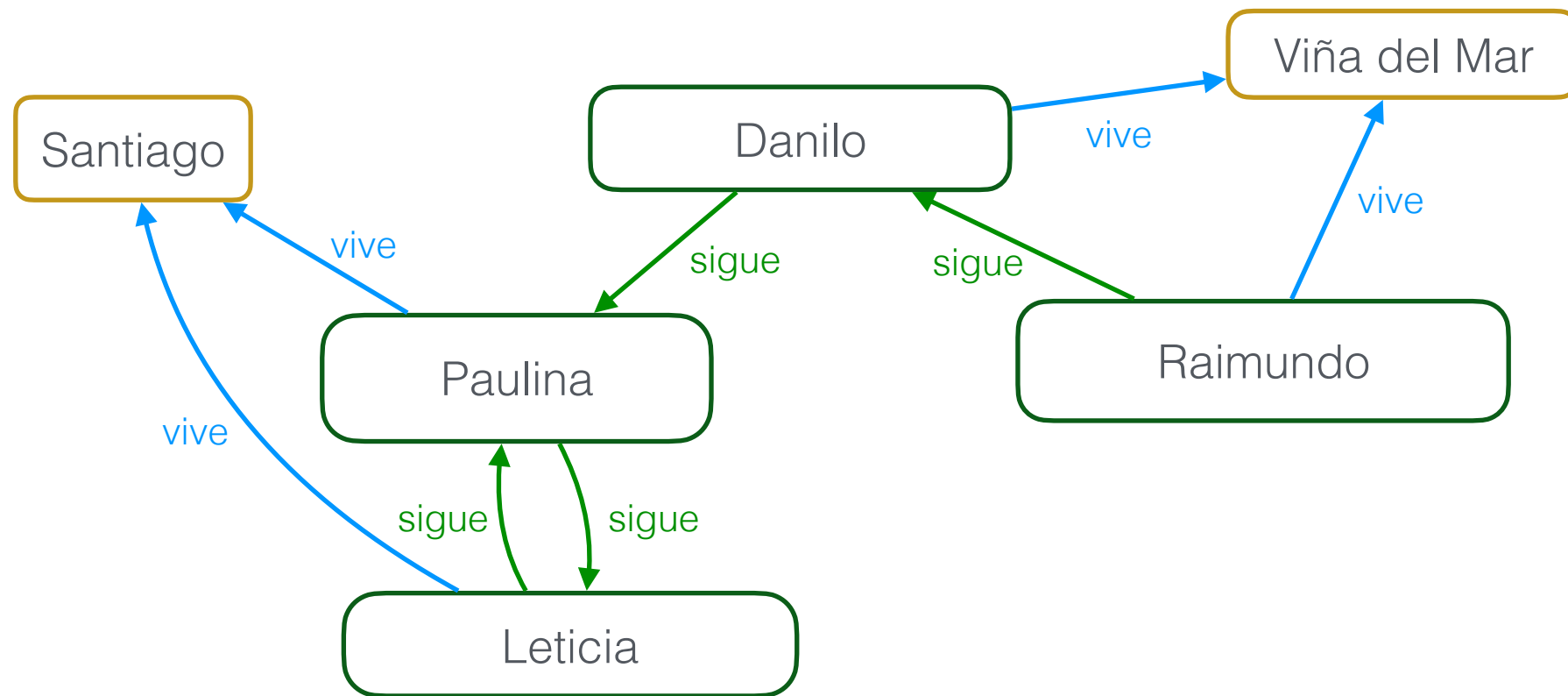
- Para  $r \in R$  decimos que

$$(u, r^-, v) \in E \iff (v, r, u) \in E$$

- Para  $p \in R^\pm$  y  $v \in V$  definimos la vecindad con respecto a  $r$  como

$$N_p(v) = \{u \in V \mid (u, p, v) \in E\}$$

# R-GCN: definición



$$N_{\text{sigue}}(\text{Danilo}) = \{\text{Raimundo}\}$$

$$N_{\text{sigue}^-}(\text{Danilo}) = \{\text{Paulina}\}$$

$$N_{\text{vive}}(\text{Santiago}) = \{\text{Paulina}, \text{Leticia}\}$$

# R-GCN: definición

- Versión sin inversos:

Actualización para nodo  $v$  en la capa  $t \in \{1, \dots, T\}$ :

$$h_v^{(t)} = \sigma\left(W_0^{(t)}h_v^{(t-1)} + \sum_{r \in R} \sum_{w \in N_r(v)} W_r^{(t)}h_w^{(t-1)} + b^{(t)}\right)$$

- Versión con inversos:

Actualización para nodo  $v$  en la capa  $t \in \{1, \dots, T\}$ :

$$h_v^{(t)} = \sigma\left(W_0^{(t)}h_v^{(t-1)} + \sum_{p \in R^\pm} \sum_{w \in N_p(v)} W_p^{(t)}h_w^{(t-1)} + b^{(t)}\right)$$

Por simplicidad asumimos la versión sin inversos

# Poder expresivo de R-GCNs

- En general, podemos usar las herramientas que ya vimos (WL y lógica) para estudiar el poder expresivo de las R-GCNs
- Hay algunas pequeñas diferencias

Weisfeiler and Leman go Relational  
P. Barceló, M. Galkin, C. Morris, M.R.

A Theory of Link Prediction via Relational Weisfeiler-Leman on Knowledge Graphs  
X. Huang, M. R., I. Ceylan, P. Barceló

# El test Relational Weisfeiler-Leman (1-RWL)

- **1-RWL:** Dado un knowledge graph  $G = (V, E, R)$ 
  - Comenzamos con un coloreo para los nodos  $(c_v^{(0)})_{v \in V}$
  - En la iteración  $t \geq 1$ , actualizamos el color de cada nodo según los colores de sus vecinos:

$$c_v^{(t)} = \text{Relabel}\left((c_v^{(t-1)}, \{(c_w^{(t-1)}, r) \mid r \in R, w \in N_r(v)\})\right)$$

- Podemos aplicar el test una cantidad fija de iteraciones o hasta que el coloreo se estabilice  
(Es decir, hasta que la partición obtenida no cambie)



# 1-RWL y R-GCNs

- Fuerte conexión entre R-GCNs y 1-RWL:
  - El poder expresivo de las R-GCNs para distinguir nodos **coincide** con el de 1-RWL

Cota superior: Las R-GCNs no son más poderosas que 1-RWL

Teorema (Barceló et al. 2022):

Sea  $G = (V, E, R)$  un knowledge graph y  $T \geq 1$  un entero positivo.

Sean  $(c_v^{(0)})_{v \in V}$  y  $(h_v^{(0)})_{v \in V}$  coloreos y features iniciales equivalentes (generan la misma partición).

Entonces, toda R-GCN con  $T$  capas cumple:

$$c_u^{(t)} = c_v^{(t)} \implies h_u^{(t)} = h_v^{(t)}$$

para todo  $0 \leq t \leq T$ , y para todo par de nodos  $u, v \in V$

# 1-RWL y R-GCNs

- Fuerte conexión entre R-GCNs y 1-RWL:
  - El poder expresivo de las R-GCNs para distinguir nodos **coincide** con el de 1-RWL

Cota inferior: Las R-GCNs pueden ser igual de poderosas que 1-RWL

Teorema (Barceló et al. 2022):

Sea  $G = (V, E, R)$  un knowledge graph y  $T \geq 1$  un entero positivo.

Sea  $(c_v^{(0)})_{v \in V}$  un coloreo inicial para  $G$ .

Entonces, existen features iniciales  $(h_v^{(0)})_{v \in V}$  equivalentes a  $(c_v^{(0)})_{v \in V}$  y una R-GCN con  $T$  capas tal que:

$$c_u^{(t)} = c_v^{(t)} \iff h_u^{(t)} = h_v^{(t)}$$

para todo  $0 \leq t \leq T$ , y para todo par de nodos  $u, v \in V$

# Lógica y R-GCNs

- Para grafos no dirigidos: [Conexión GNNs vs Graded Modal Logic \(GML\)](#)

The Logical Expressiveness of Graph Neural Networks  
P. Barceló, E. Kostylev, M. Monet, J. Pérez, J. Reutter, J. Silva

Teorema (Barceló et al. 2020):

Sea  $\varphi$  un clasificador lógico de nodos. Entonces

$\varphi$  se puede expresar con una GNN  $\iff \varphi$  se puede expresar en GML

# GML

- Las fórmulas en GML son unarias, es decir, tienen una variable libre
- Tenemos dados un conjunto de colores de nodos  $\mathcal{C}$
- Las fórmulas en GML se definen como:
  - Si  $a \in \mathcal{C}$  es un color, entonces  $\varphi(x) = a(x)$  es una fórmula
  - Si  $\varphi(x)$  y  $\psi(x)$  están en GML, y  $N$  es un natural, entonces

$$\neg\varphi(x) \quad \varphi(x) \wedge \psi(x) \quad \exists^{\geq N} y (\varphi(y) \wedge E(y, x))$$

también son fórmulas en GML

# R-GML

- Las fórmulas en R-GML son unarias, es decir, tienen una variable libre y se evalúan sobre knowledge graphs (con colores en los nodos)
- Tenemos dados un conjunto de colores de nodos  $\mathcal{C}$  y tipos de relaciones  $R$
- Las fórmulas en R-GML se definen como:
  - Si  $a \in \mathcal{C}$  es un color, entonces  $\varphi(x) = a(x)$  es una fórmula
  - Si  $\varphi(x)$  y  $\psi(x)$  están en R-GML,  $r \in R$  y  $N$  es un natural, entonces

$$\neg\varphi(x) \quad \varphi(x) \wedge \psi(x) \quad \exists^{\geq N}y (\varphi(y) \wedge r(y, x))$$

también son fórmulas en R-GML

# R-GML y R-GCNs

Teorema (X. Huang et al. 2023):

Sea  $\varphi$  un clasificador lógico de nodos para knowledge graphs. Entonces

$\varphi$  se puede expresar con una R-GCN  $\iff \varphi$  se puede expresar en R-GML

# GNNs con readout

Teorema (Barceló et al. 2020):

Todo clasificador lógico de nodos en  $\text{FOC}_2$  se puede expresar con una GNN

- No existe un teorema análogo para R-GCNs

# CompGCNs

- Introducido el 2020:  
Composition-based multi-relational graph convolutional networks  
S. Vashishth, S. Sanyal, V. Nitin, and P. P. Talukdar.
- Aplicado a clasificación de nodos y predicción de links
- **Motivación:** Disminuir la cantidad de parámetros

Actualización para nodo  $v$  en la capa  $t \in \{1, \dots, T\}$ :

$$h_v^{(t)} = \sigma\left(W_0^{(t)}h_v^{(t-1)} + \sum_{r \in R} \sum_{w \in N_r(v)} W_1^{(t)}\phi(h_w^{(t-1)}, z_r^{(t)}) + b^{(t)}\right)$$

Ahora los parámetros son  $W_0^{(t)}, W_1^{(t)}, z_r^{(t)}, b^{(t)}$



# CompGCNs

Actualización para nodo  $v$  en la capa  $t \in \{1, \dots, T\}$ :

$$h_v^{(t)} = \sigma\left(W_0^{(t)}h_v^{(t-1)} + \sum_{r \in R} \sum_{w \in N_r(v)} W_1^{(t)}\phi(h_w^{(t-1)}, z_r^{(t)}) + b^{(t)}\right)$$

- Opciones comunes para  $\phi$ :
  - $\phi(h, z) = h + z$  (suma)
  - $\phi(h, z) = (h, z)$  (concatenación)
  - $\phi(h, z) = h * z$  (multiplicación coordenada a coordenada)

# R-GCN y CompGCNs

- El poder expresivo de las CompGCNs dependen de la elección de  $\phi$
- Si  $\phi$  puede expresar escalamiento de vectores, entonces alcanza el mismo poder expresivo que las R-GCNs
  - E.g. si  $\phi$  es la multiplicación coordenada a coordenada

Teorema (Barceló et al. 2022) (informal):

El poder expresivo de 1-RWL coincide con el de las CompGCNs, siempre y cuando la función  $\phi$  pueda expresar escalamiento de vectores.

# R-GCN y CompGCNs

- ¿Qué pasa para suma y concatenación?
- Se puede demostrar que en estos casos las CompGCNs son más débiles que las R-GCNs
- Esto es consecuencia del siguiente teorema:

Teorema (Barceló et al. 2022) (informal):

El poder expresivo de **weak 1-RWL** coincide con el de las CompGCNs, cuando la función  $\phi$  es suma o concatenación.

# Weak 1-RWL

- **Weak 1-RWL:** Dado un knowledge graph  $G = (V, E, R)$ , donde  $R = \{1, \dots, q\}$ 
  - Comenzamos con un coloreo para los nodos  $(c_v^{(0)})_{v \in V}$
  - En la iteración  $t \geq 1$ , actualizamos el color de cada nodo según los colores de sus vecinos:

$$c_v^{(t)} = \text{Relabel}\left((c_v^{(t-1)}, \{c_w^{(t-1)} \mid r \in R, w \in N_r(v)\}, |N_1(v)|, \dots, |N_q(v)|)\right)$$

- Podemos aplicar el test una cantidad fija de iteraciones o hasta que el coloreo se estabilice  
(Es decir, hasta que la partición obtenida no cambie)

# Mas allá de knowledge graphs

- Existen modelos más generales que knowledge graphs (e.g. property graphs)

¿Cómo extendemos las GNNs a estos modelos?

- Podemos ir más allá de grafos:
  - Hipergrafos
  - Estructuras relacionales

¿Cómo extendemos las GNNs a estos modelos?

Link Prediction with Relational Hypergraphs  
X. Huang, M. R., I. Ceylan, P. Barceló