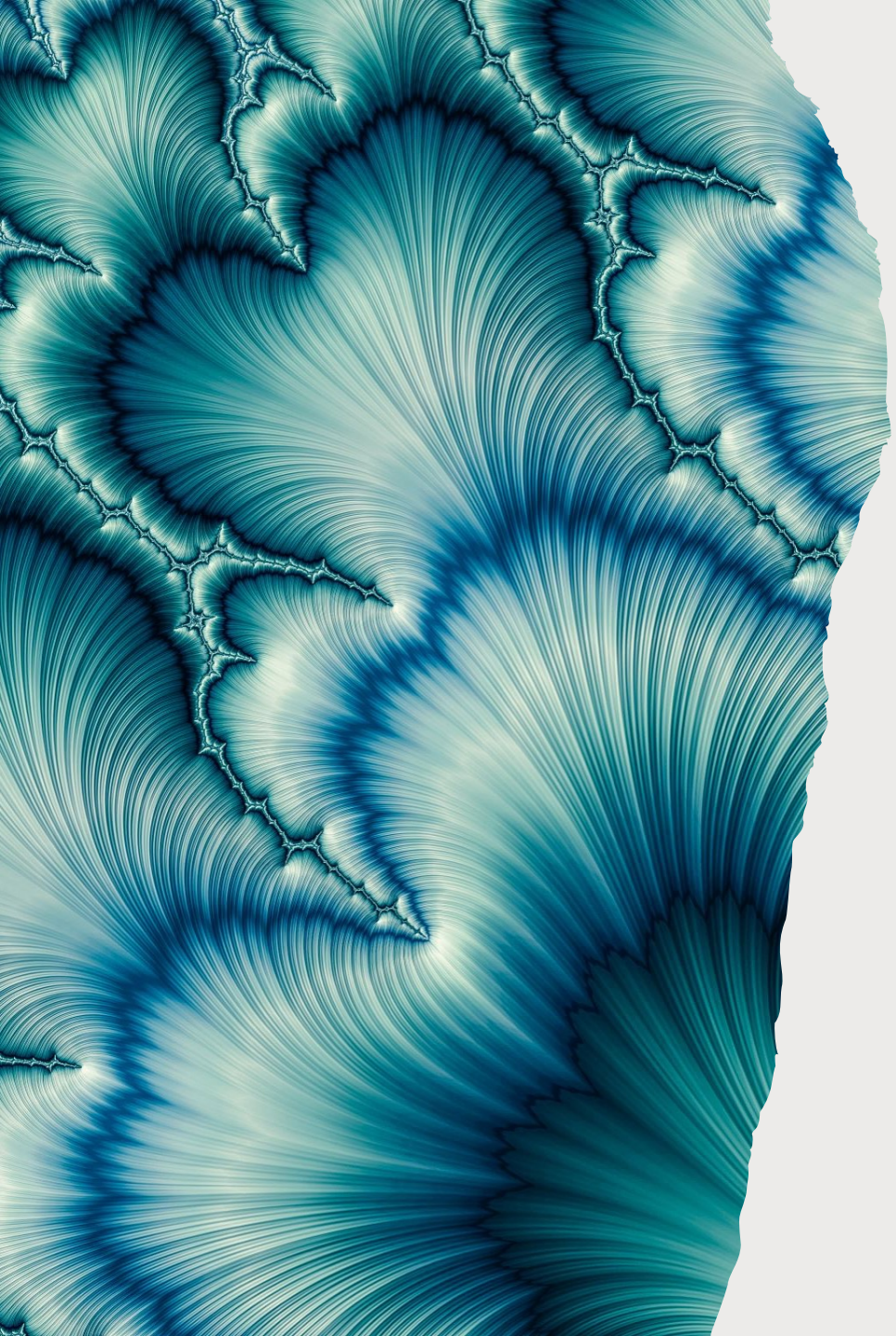


Ontology Design Metapattern for RelationType Role Composition

Utkarshani Jaimini[1], Ruwan Wickramarachchi[1], Cory Henson[2], Amit Sheth[1]

[1]*Artificial Intelligence Institute, University of South Carolina, Columbia, SC, USA*

[2]*Bosch Center for Artificial Intelligence, Pittsburgh, PA, USA*

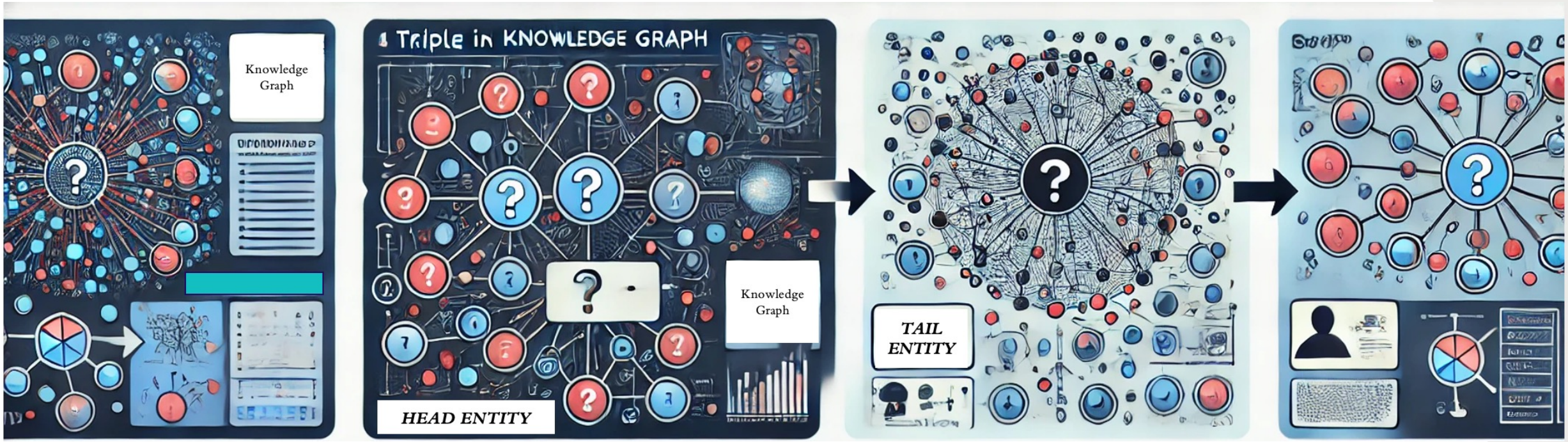
AI Institute
UNIVERSITY OF SOUTH CAROLINA

BOSCH

MOTIVATION

Knowledge graph link prediction

KNOWLEDGE GRAPH LINK PREDICTION



- Traditional link prediction approach on the above predicts entity instance
- Many real-world use-cases require only predicting a type of a node rather than an instance node.

  Such as in medical knowledge graph: ⟨fever_instance symptomOf ? ⟩
- Predicting tail instance leads to ranks ordering possible 1000s of nodes solutions leading to poor performance

*Images created using generative AI

WHY DO WE NEED METAPATTERN AND ROLE COMPOSITION?

METAPATTERN

METAPATTERN

Metapattern is a pattern for patterns, setting a flexible framework that can be filled in with specific details as needed.

It is a high-level, generalizable, abstract pattern that serves as a template or foundational structure for creating more specific patterns.

Help to create consistent, reusable, and adaptable ontologies by offering a foundational structure for diverse types of knowledge representation.

In contrast, a ***pattern*** is more specific (like a data pattern for recording patient information), on the other hand a ***metapattern*** provides an overarching structure that can be customized and applied in multiple scenarios.

CHARACTERISTICS OF METAPATTERNS IN ONTOLOGY

1

Abstract and High-Level: Metapatterns are not domain-specific. Instead, they are abstract structures that can be customized for specific needs.

2

Reusable: Designed to be applicable in multiple contexts and across various domains, making them flexible and adaptable.

3

Organizational: Guide the overall organization and architecture of an ontology, rather than just individual elements.

ROLE COMPOSITION

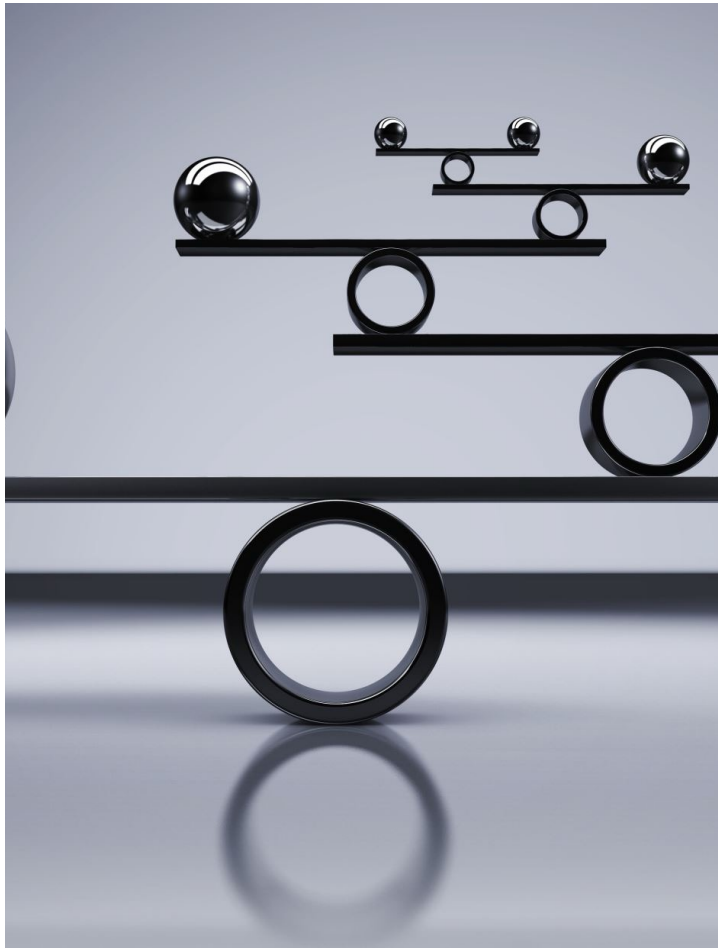
ROLE COMPOSITION

- ***Role composition*** refers to combining two or more roles (or relationships) into a single, derived role.
- When two roles are composed, they form a new role that represents the indirect connection between the entities linked by each role individually.
- Useful for representing chains of relationships.
- In description logics, role composition is usually represented using the symbol ∘ (circle). If we have roles R and S, their composition R∘S represents a new role such that:

$$x\ (R \circ S)\ z, \text{ holds if there exists an entity } y \text{ such that } x\ R\ y \text{ and } y\ S\ z$$

- hasParent ∘ hasSibling would mean that for an individual x and their relative z, there exists an intermediary y (the parent of x) such that x hasParent y, and y hasSibling z.

APPLICATIONS OF ROLE COMPOSITION



Inferring Indirect Relationships: Allows for the definition of new, indirect relationships based on existing ones.

Querying and Reasoning: Composed roles enable reasoning engines to infer new knowledge by chaining relationships, enhancing the expressiveness and querying capabilities of an ontology.

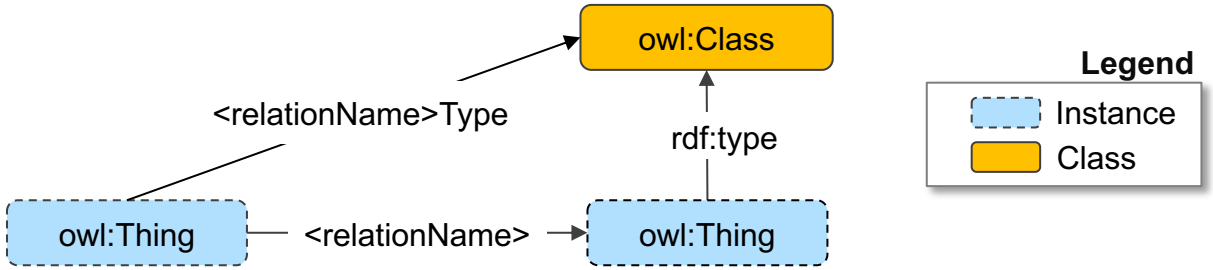
Modeling Complex Domains: Domains with intricate relationships, such as biology, genealogy, and organizational hierarchies, benefit from role composition. Complex domains require propagation of one property to another which can be introduced as path reification to be used for inference at a later stage.

For example, in a corporate ontology, a "colleague" relationship can be defined as the composition of "worksIn" (department) and "hasColleague".

RELATIONTYPE METAPATTERN

RELATION TYPE META-PATTERN

RelationType is a metapattern that uses a role composition for a common path in a knowledge graph linking the head of a triple with the type of the tail.



$$< \text{head, relationName, tail} > \quad (relation\ link)$$

$$< \text{tail, rdf:type, tailClass} > \quad (type\ link)$$

$$< \text{head, relationNameType, tailClass} > \quad (relationNameType\ link)$$

$$\text{<relationName>} \circ \text{rdf:type} \sqsubseteq \text{<relationName>Type}$$

- Reified path from the head-entity instance to the ***rdfs:type*** of the tail-entity for causal link prediction
- RelationType provides a pattern to represent knowledge graph reified relation into the ontology design

RELATION TYPE META-PATTERN

<relationName> ∘ rdf:type ⊑ <relationName>Type

relationName

The head and tail of a relationName link can be an instance of any class.

$$\exists <relationName>.\top \sqsubseteq \top \qquad Scoped\ Domain$$

$$\top \sqsubseteq \forall <relationName>.\top \qquad Scoped\ Range$$

relationNameType

The head of a relationNameType link can be an instance of any class, the tail can be any class.

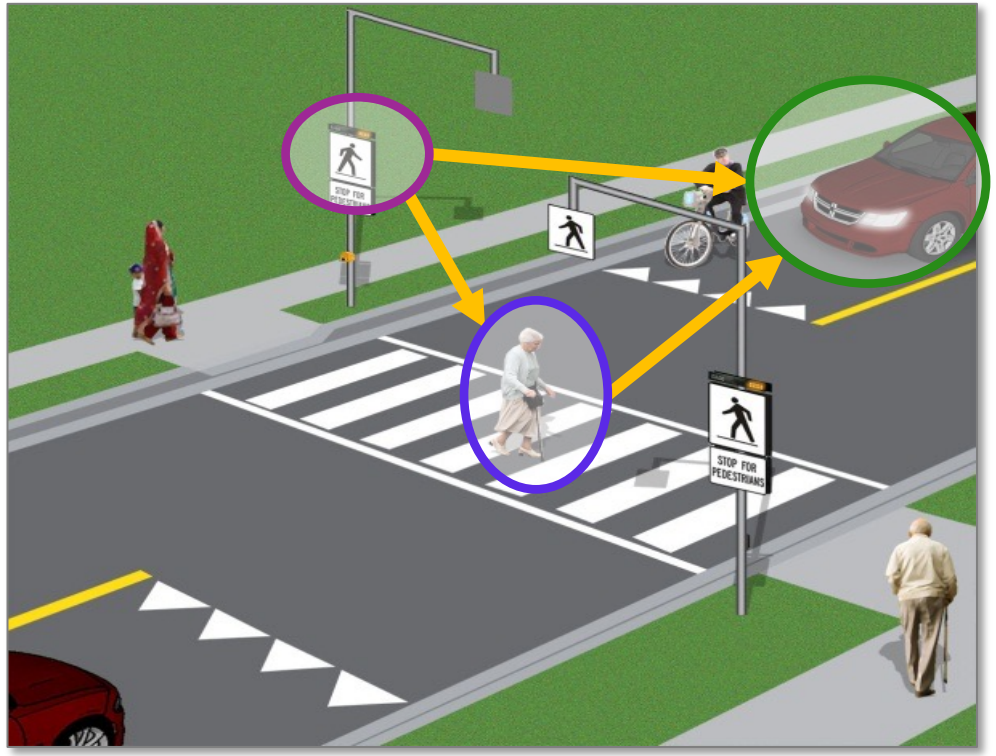
$$\exists <relationName>Type.owl:Class \sqsubseteq \top \qquad Scoped\ Domain$$

$$\top \sqsubseteq \forall <relationName>Type.owl:Class \qquad Scoped\ Range$$

relationName is a place holder for more concrete property names.

RELATIONTYPE IN AUTONOMOUS DRIVING SCENE

DRIVING SCENE

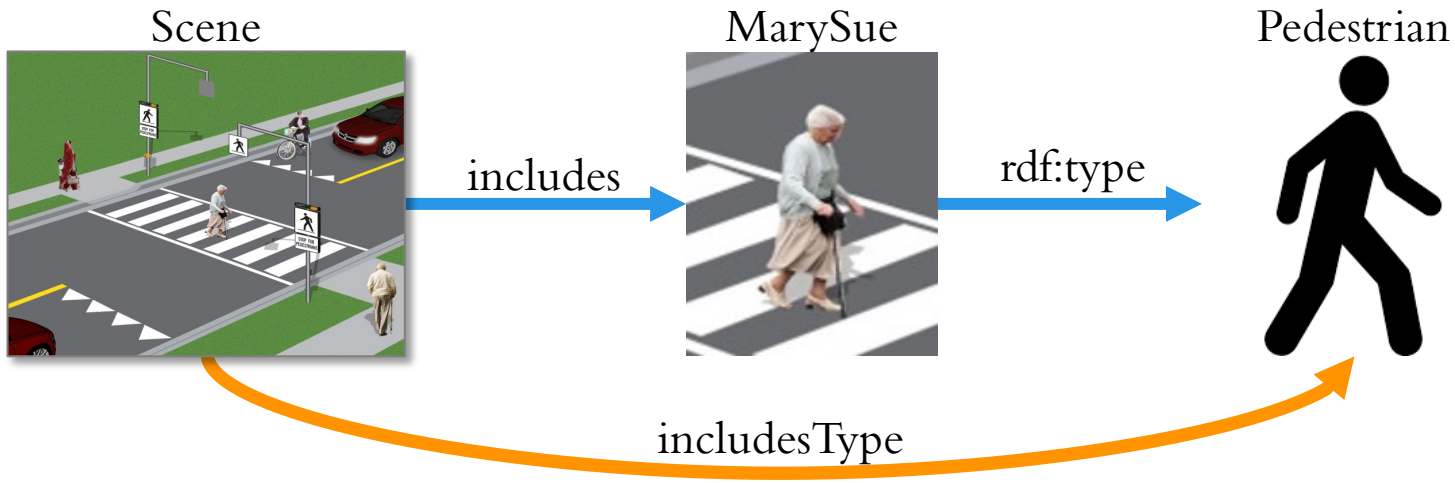


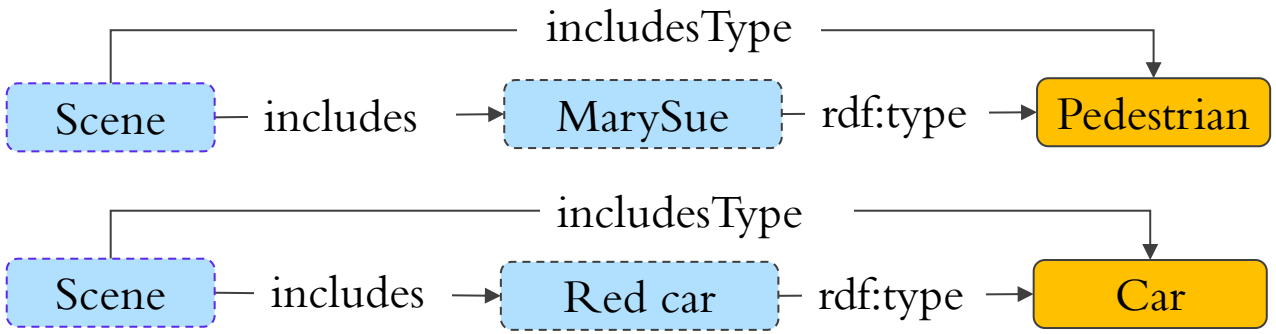
Stop line marking driving scene. Causal relations between entities in a stop line marking driving scene

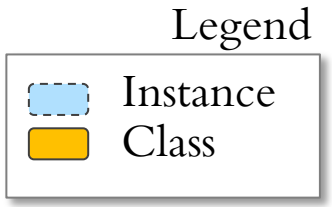
Competency questions

- What unique types of entities are included in a given scene?
- What additional types of entities can be predicted that are not currently included in the scene (i.e., the knowledge-based entity prediction problem)?
- What is the pedestrian activity caused by a stop line marking in a scene?
- What is the pedestrian activity caused by the red car stopped in a scene?
- What is the vehicle activity caused by a stop line marking in a scene?

RELATIONTYPE IN DRIVING SCENE



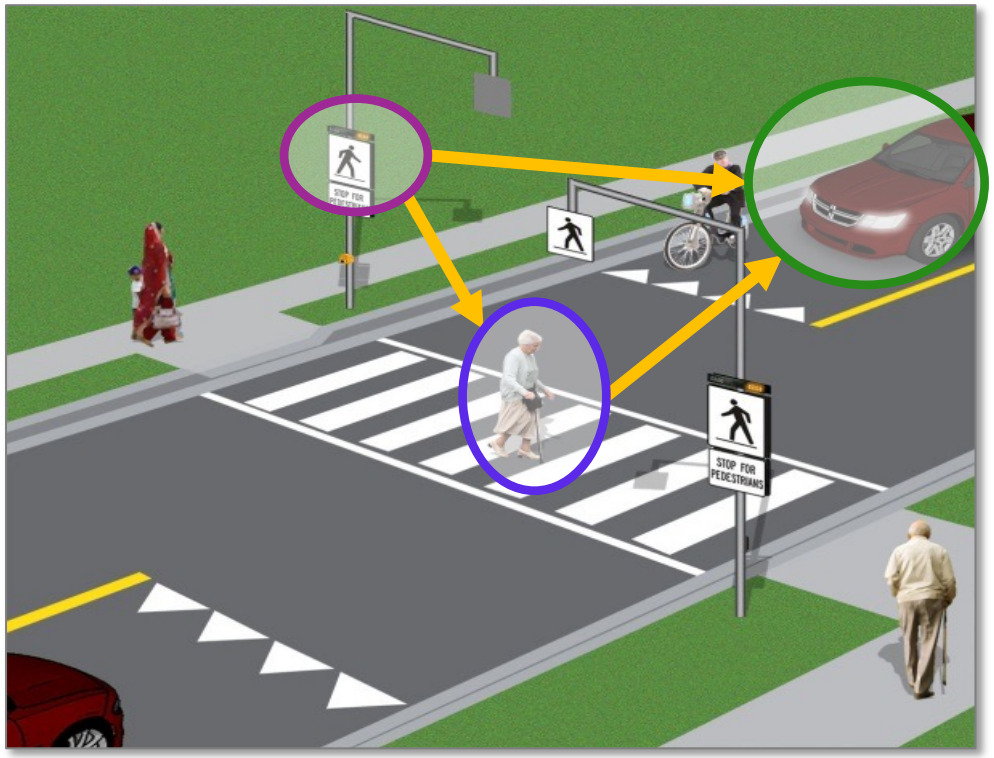




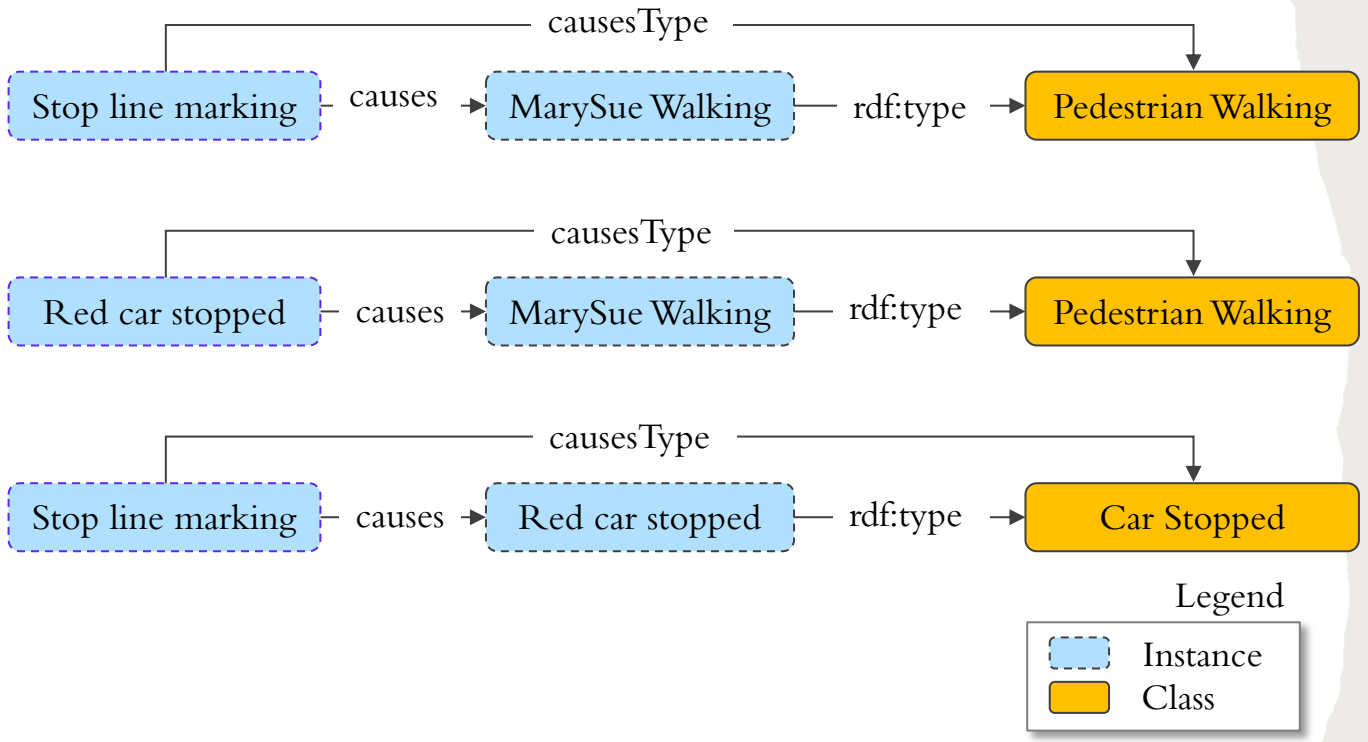
$$\text{includes} \circ \text{rdf:type} \sqsubseteq \text{includesType}$$

R.Wickramarachchi,C.Henson,A.Sheth,Knowledge-infusedlearningforentityprediction in driving scenes, Frontiers in big Data 4 (2021) 759110.

RELATIONTYPE IN DRIVING SCENE



Causal relations between entities in a stop line marking driving scene



$$\text{causes} \circ \text{rdf:type} \sqsubseteq \text{causesType}$$

Jaimini, U., Henson, C., and Sheth, A., 2025. CausalLP: Learning causal relations with weighted knowledge graph link prediction. Link: https://arxiv.org/abs/2405.02327

DISCUSSION AND CONCLUSION

- The RelationType metapattern represents reified path in knowledge graph using role composition axioms in ontology design
- Can have different instantiations for different relation types depending on the use case.
- Solves the knowledge graph link prediction problem by constraining the label space for prediction task

THANK YOU