Representing Graph-based Structures with Logic: Motivation and Goals

Despoina Magka

Oxford University Computing Laboratory

December, 2010

Main Objective

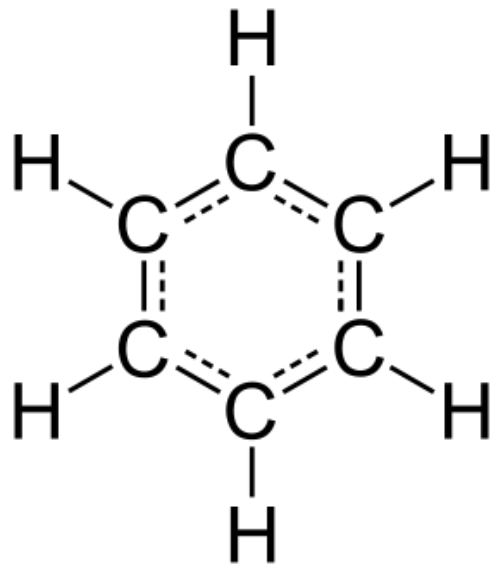
- Design a decidable Knowledge Representation and Reasoning formalism with the following requirements:

Main Objective

- Design a decidable Knowledge Representation and Reasoning formalism with the following requirements:
  - Modeling and reasoning about objects with cyclic structure (graphs)
  - Schema-level reasoning
  - Concept recognition
  - Graph composition

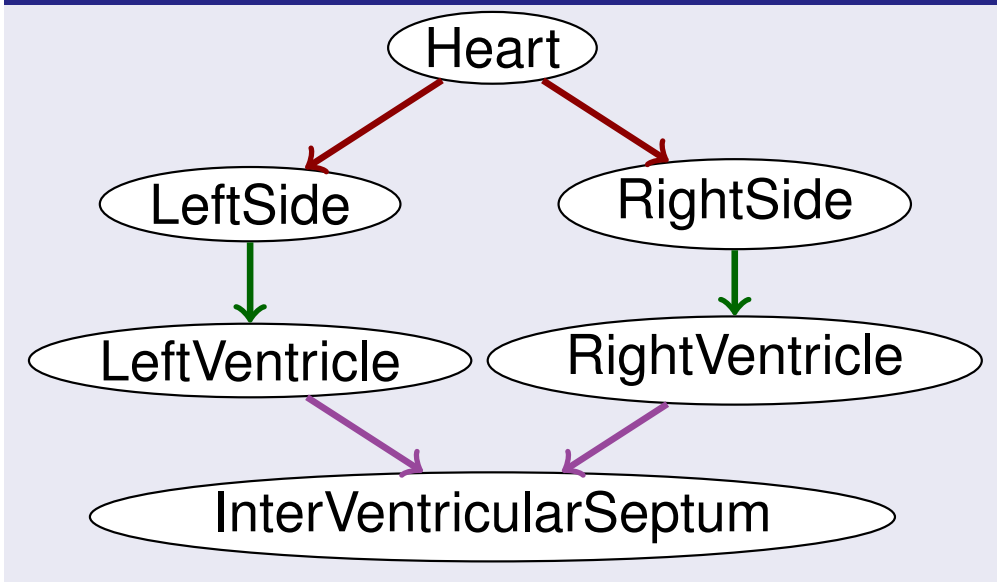
Main Objective

- Design a decidable Knowledge Representation and Reasoning formalism with the following requirements:
  - Modeling and reasoning about objects with cyclic structure (graphs)
  - Schema-level reasoning
  - Concept recognition
  - Graph composition
- Automatically classify the manually curated ChEBI ontology (Chemical Entities of Biological Interest)



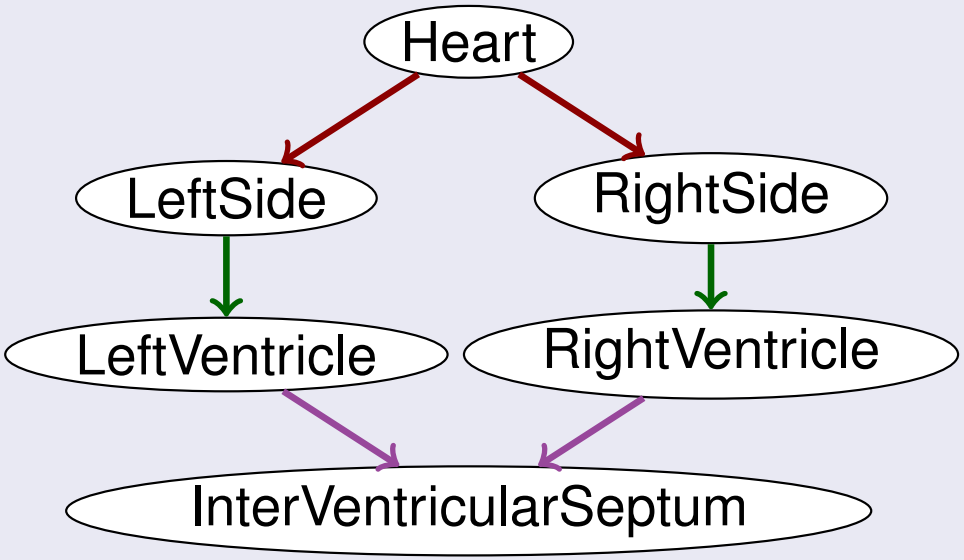
Modeling and Reasoning about Cycles

Heart Structure



Modeling and Reasoning about Cycles

Heart Structure



DL Ontology $\mathcal{O}$

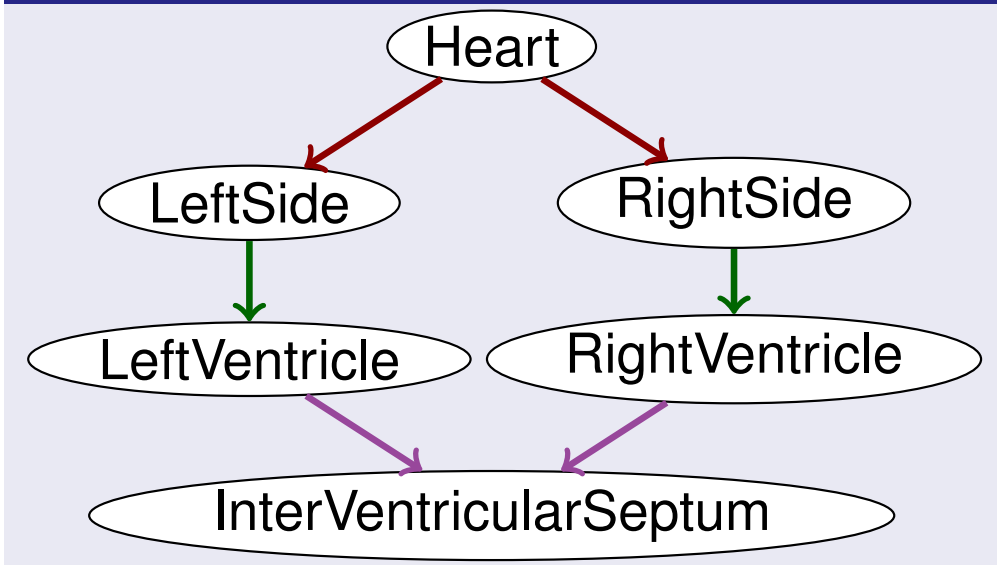
$$\begin{aligned} \text{Heart} &\sqsubseteq \exists \text{hasComponent}.\text{LS} \\ \text{Heart} &\sqsubseteq \exists \text{hasComponent}.\text{RS} \\ \text{LS} &\sqsubseteq \exists \text{hasDivision}.\text{LV} \\ \text{RS} &\sqsubseteq \exists \text{hasDivision}.\text{RV} \\ \text{LV} &\sqsubseteq \exists \text{hasLayer}.\text{IVSeptum} \\ \text{RV} &\sqsubseteq \exists \text{hasLayer}.\text{IVSeptum} \end{aligned}$$

Modeling and Reasoning about Cycles

(A1) : $\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated}$

(A2) : $\exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated} \sqsubseteq \mathsf{HasVSD}$

Heart Structure



DL Ontology $\mathcal{O}$

$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{LS}$

$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{RS}$

$\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{LV}$

$\mathsf{RS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{RV}$

$\mathsf{LV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$

$\mathsf{RV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$

Modeling and Reasoning about Cycles

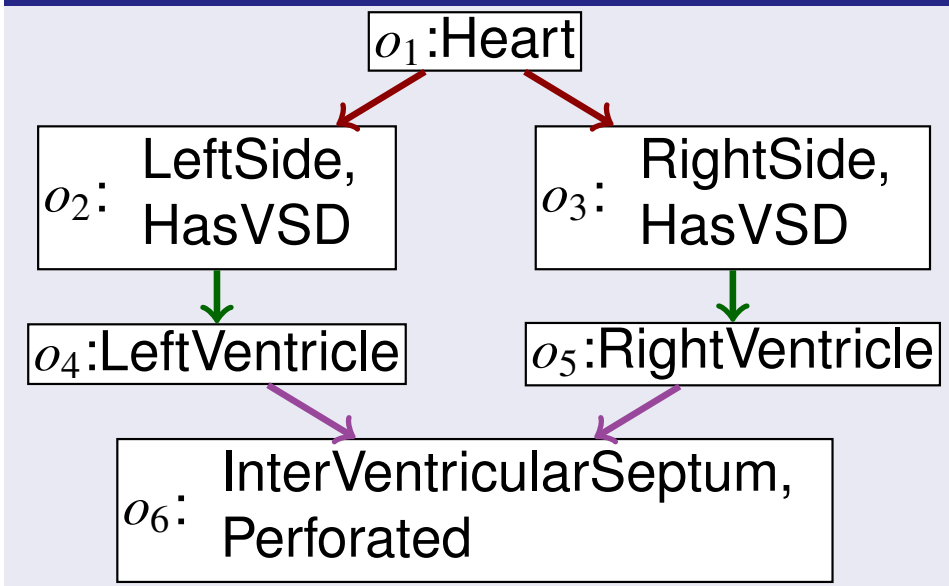
(A1) : $\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated}$

(A2) : $\exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated} \sqsubseteq \mathsf{HasVSD}$

$$\mathcal{I}_1 \models \mathsf{LS} \sqsubseteq \mathsf{HasVSD}$$

$$\mathcal{I}_1 \models \mathsf{RS} \sqsubseteq \mathsf{HasVSD}$$

Cyclic Model $\mathcal{I}_1$



DL Ontology $\mathcal{O}$

$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{LS}$

$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{RS}$

$\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{LV}$

$\mathsf{RS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{RV}$

$\mathsf{LV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$

$\mathsf{RV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$

Modeling and Reasoning about Cycles

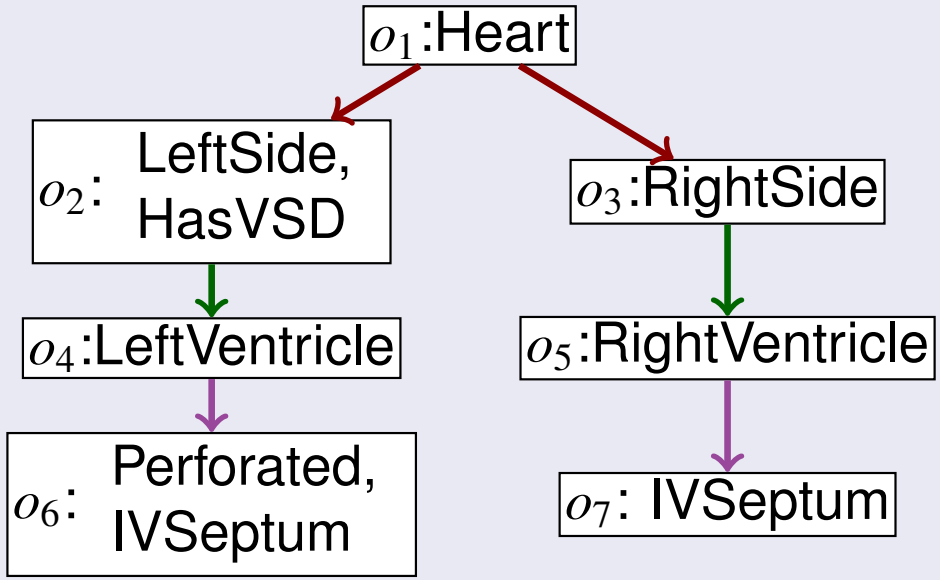
(A1) : $\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated}$

(A2) : $\exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated} \sqsubseteq \mathsf{HasVSD}$

$$\mathcal{I}_2 \models \mathsf{LS} \sqsubseteq \mathsf{HasVSD}$$

$$\mathcal{I}_2 \not\models \mathsf{RS} \sqsubseteq \mathsf{HasVSD}$$

Tree Model $\mathcal{I}_2$



DL Ontology $\mathcal{O}$

$$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{LS}$$
$$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{RS}$$
$$\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{LV}$$
$$\mathsf{RS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{RV}$$
$$\mathsf{LV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$$
$$\mathsf{RV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$$

Modeling and Reasoning about Cycles

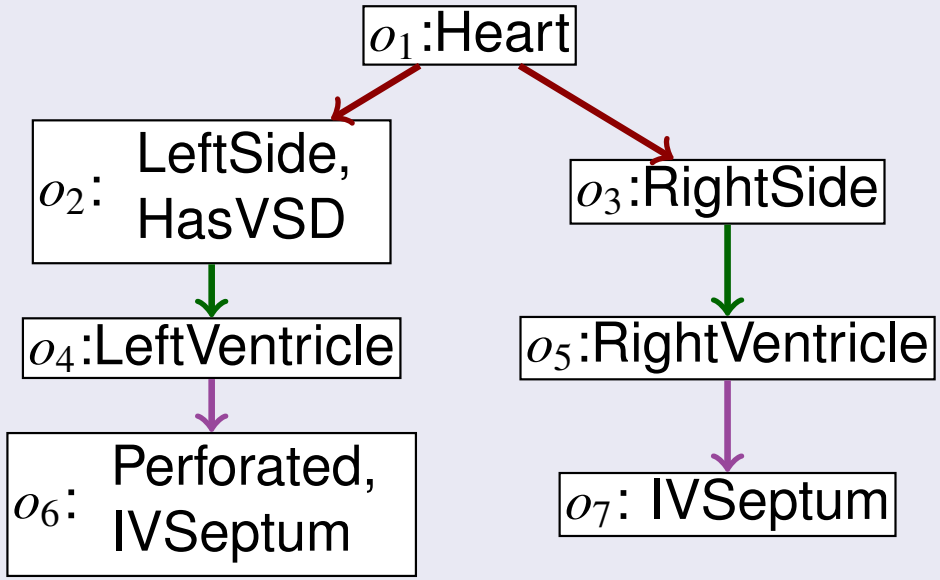
(A1) : $\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated}$

(A2) : $\exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated} \sqsubseteq \mathsf{HasVSD}$

$$\mathcal{O} \cup \{A_1, A_2\} \models \mathsf{LS} \sqsubseteq \mathsf{HasVSD}$$

$$\mathcal{O} \cup \{A_1, A_2\} \not\models \mathsf{RS} \sqsubseteq \mathsf{HasVSD}$$

Tree Model $\mathcal{I}_2$



DL Ontology $\mathcal{O}$

$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{LS}$

$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{RS}$

$\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{LV}$

$\mathsf{RS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{RV}$

$\mathsf{LV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$

$\mathsf{RV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$

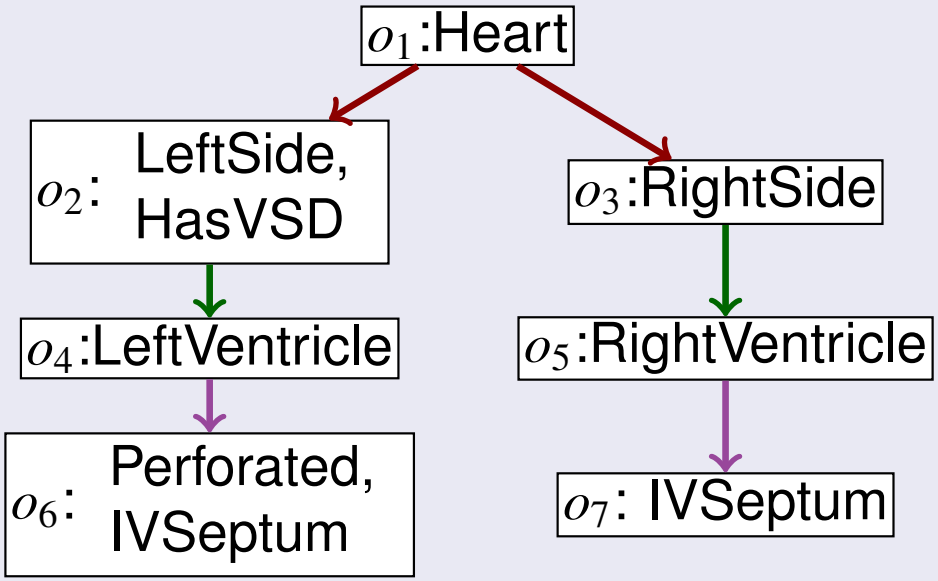
Modeling and Reasoning about Cycles

(A1) : $\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated}$

(A2) : $\exists \mathsf{hasDivision}.\exists \mathsf{hasLayer}.\mathsf{Perforated} \sqsubseteq \mathsf{HasVSD}$

$\rightsquigarrow$ We do not want to lose these inferences

Tree Model $\mathcal{I}_2$



DL Ontology $\mathcal{O}$

$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{LS}$

$\mathsf{Heart} \sqsubseteq \exists \mathsf{hasComponent}.\mathsf{RS}$

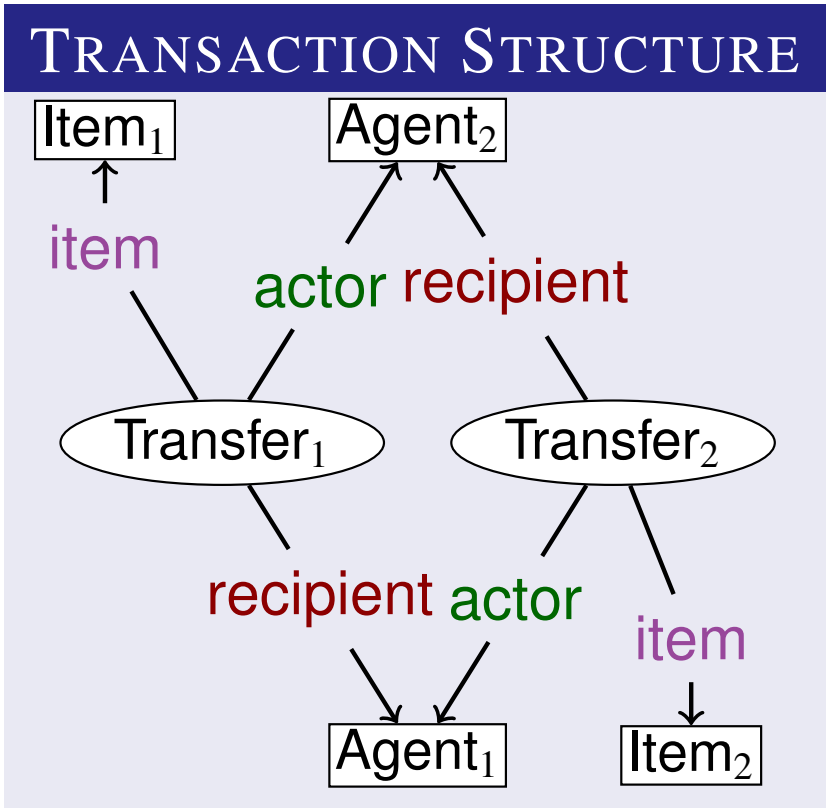
$\mathsf{LS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{LV}$

$\mathsf{RS} \sqsubseteq \exists \mathsf{hasDivision}.\mathsf{RV}$

$\mathsf{LV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$

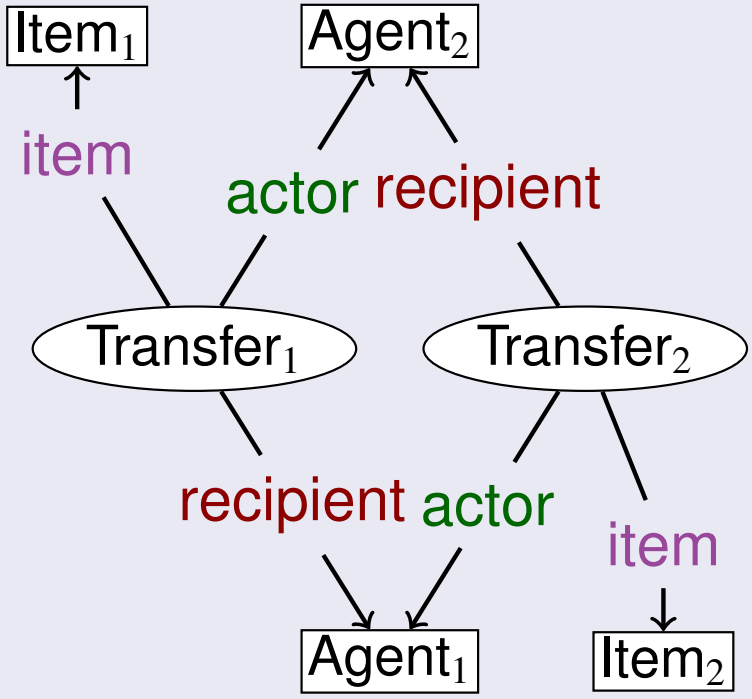
$\mathsf{RV} \sqsubseteq \exists \mathsf{hasLayer}.\mathsf{IVSeptum}$

Instance vs Schema-level Reasoning



Instance vs Schema-level Reasoning

Transaction Structure

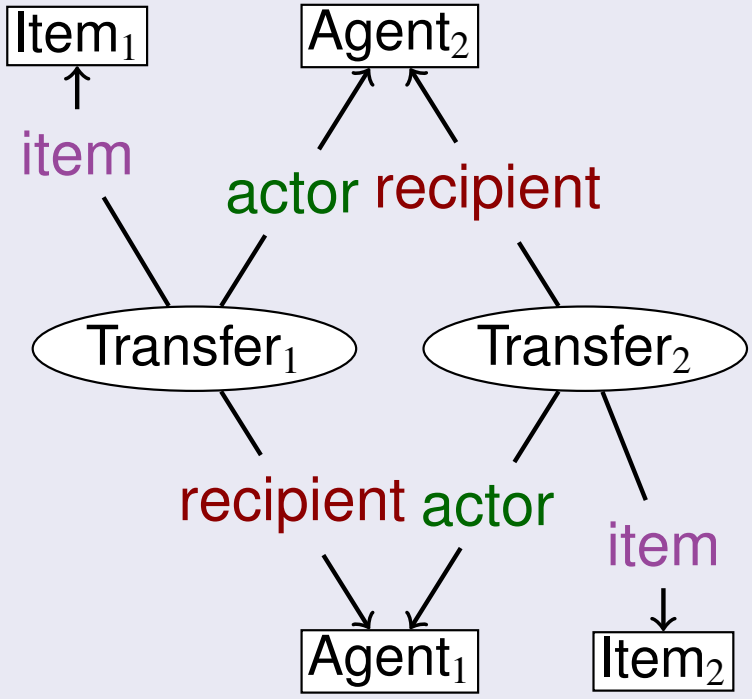


DL-Safe Rule

$$\text{transfer}(x, y) \wedge \text{transfer}(x, z) \wedge \text{actor}(y, u_2) \wedge \text{recipient}(y, u_1) \wedge \text{actor}(z, u_1) \wedge \text{recipient}(z, u_2) \wedge \text{item}(y, v_1) \wedge \text{item}(z, v_2) \wedge O(x) \wedge O(y) \wedge O(z) \wedge O(u_1) \wedge O(u_2) \wedge O(v_1) \wedge O(v_2) \rightarrow \text{Transaction}(x)$$

Instance vs Schema-level Reasoning

Transaction Structure



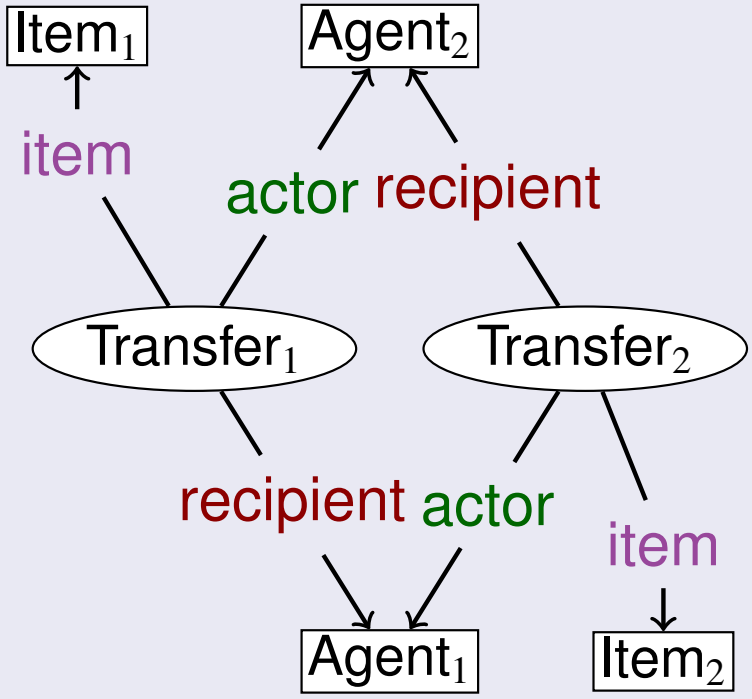
DL-Safe Rule

$$\begin{aligned}&\text{transfer}(x,y) \wedge \text{transfer}(x,z) \wedge \\ &\text{actor}(y,u_2) \wedge \text{recipient}(y,u_1) \wedge \\ &\text{actor}(z,u_1) \wedge \text{recipient}(z,u_2) \wedge \\ &\text{item}(y,v_1) \wedge \text{item}(z,v_2) \wedge \\ &O(x) \wedge O(y) \wedge O(z) \wedge O(u_1) \wedge O(u_2) \\ &\wedge O(v_1) \wedge O(v_2) \rightarrow \text{Transaction}(x)\end{aligned}$$

$$\begin{aligned}&\{\text{transfer}(t,t_1), \text{transfer}(t,t_2), \text{actor}(t_1,a_2), \text{recipient}(t_1,a_1), \\ &\text{actor}(t_2,a_1), \text{recipient}(t_2,a_2), \text{item}(t_1,i_1), \text{item}(t_2,i_2), \\ &O(t), O(t_1), O(t_2), O(a_1), O(a_2), O(i_1), O(i_2)\}\end{aligned}$$

Instance vs Schema-level Reasoning

Transaction Structure



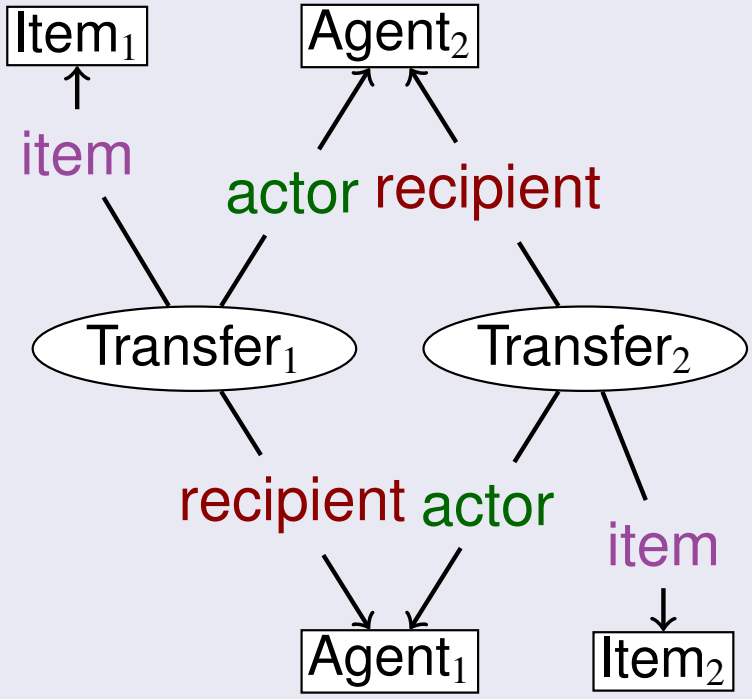
DL-Safe Rule

$$\begin{aligned}&\text{transfer}(x,y) \wedge \text{transfer}(x,z) \wedge \\ &\text{actor}(y,u_2) \wedge \text{recipient}(y,u_1) \wedge \\ &\text{actor}(z,u_1) \wedge \text{recipient}(z,u_2) \wedge \\ &\text{item}(y,v_1) \wedge \text{item}(z,v_2) \wedge \\ &O(x) \wedge O(y) \wedge O(z) \wedge O(u_1) \wedge O(u_2) \\ &\wedge O(v_1) \wedge O(v_2) \rightarrow \text{Transaction}(x)\end{aligned}$$

$$\begin{aligned}&\{\text{transfer}(t,t_1), \text{transfer}(t,t_2), \text{actor}(t_1,a_2), \text{recipient}(t_1,a_1), \\ &\text{actor}(t_2,a_1), \text{recipient}(t_2,a_2), \text{item}(t_1,i_1), \text{item}(t_2,i_2), \\ &O(t), O(t_1), O(t_2), O(a_1), O(a_2), O(i_1), O(i_2)\} \\ &\rightarrow \text{Transaction}(t)\end{aligned}$$

Instance vs Schema-level Reasoning

Transaction Structure



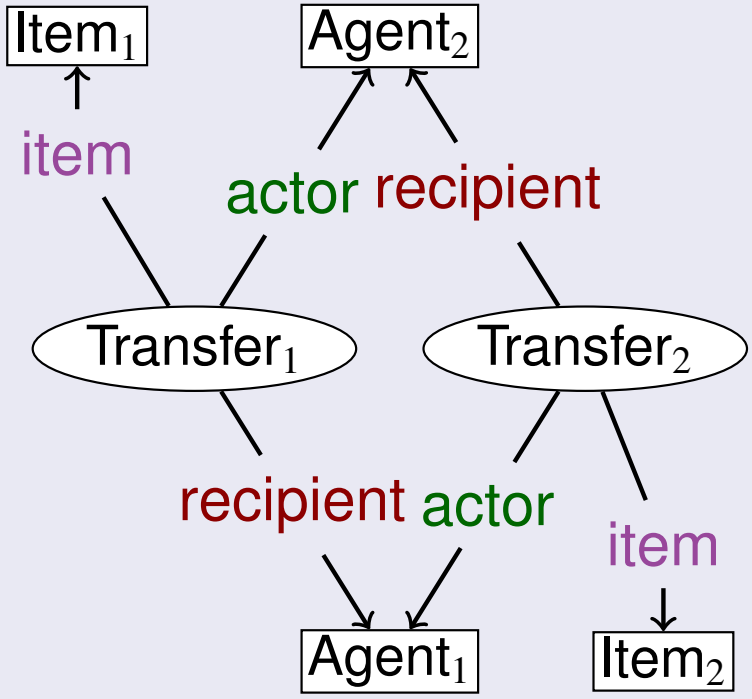
DL-Safe Rule

$$\text{transfer}(x, y) \wedge \text{transfer}(x, z) \wedge \text{actor}(y, u_2) \wedge \text{recipient}(y, u_1) \wedge \text{actor}(z, u_1) \wedge \text{recipient}(z, u_2) \wedge \text{item}(y, v_1) \wedge \text{item}(z, v_2) \wedge O(x) \wedge O(y) \wedge O(z) \wedge O(u_1) \wedge O(u_2) \wedge O(v_1) \wedge O(v_2) \rightarrow \text{Transaction}(x)$$

$$\{\text{transfer}(t, t_1), \text{transfer}(t, t_2), \text{actor}(t_1, a_2), \text{recipient}(t_1, a_1), \text{actor}(t_2, a_1), \text{recipient}(t_2, a_2), \text{item}(t_1, i_1), \text{item}(t_2, i_2), O(t), O(t_1), O(t_2), O(a_1), O(a_2), O(i_1), O(i_2)\}$$

Instance vs Schema-level Reasoning

Transaction Structure



DL-Safe Rule

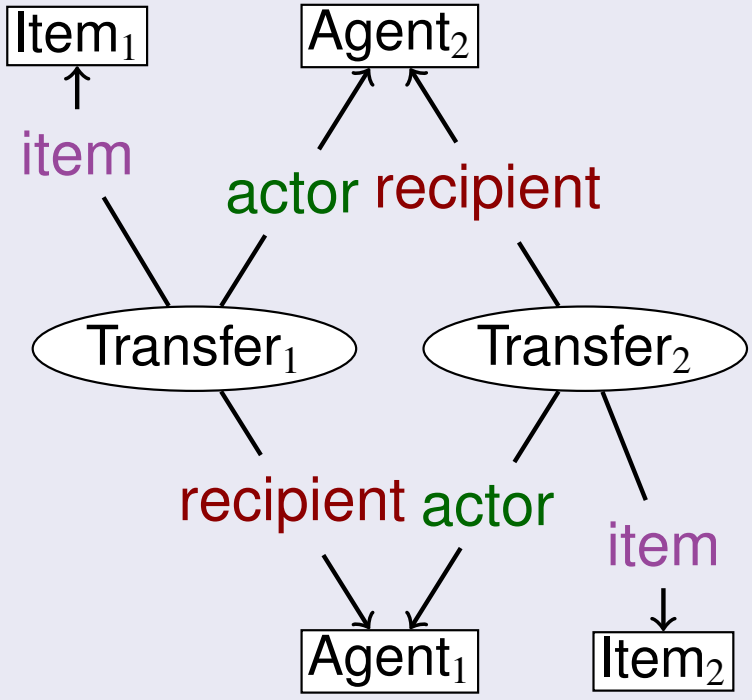
$$\text{transfer}(x,y) \wedge \text{transfer}(x,z) \wedge \text{actor}(y,u_2) \wedge \text{recipient}(y,u_1) \wedge \text{actor}(z,u_1) \wedge \text{recipient}(z,u_2) \wedge \text{item}(y,v_1) \wedge \text{item}(z,v_2) \wedge O(x) \wedge O(y) \wedge O(z) \wedge O(u_1) \wedge O(u_2) \wedge O(v_1) \wedge O(v_2) \rightarrow \text{Transaction}(x)$$

$\{\text{transfer}(t,t_1), \text{transfer}(t,t_2), \text{actor}(t_1,a_2), \text{recipient}(t_1,a_1), \text{actor}(t_2,a_1), \text{recipient}(t_2,a_2), \text{item}(t_1,i_1), \text{WithItem}(t_2), O(t), O(t_1), O(t_2), O(a_1), O(a_2), O(i_1)\}$

DL Axiom: $\text{WithItem} \sqsubseteq \exists \text{item}.\top$

Instance vs Schema-level Reasoning

Transaction Structure



DL-Safe Rule

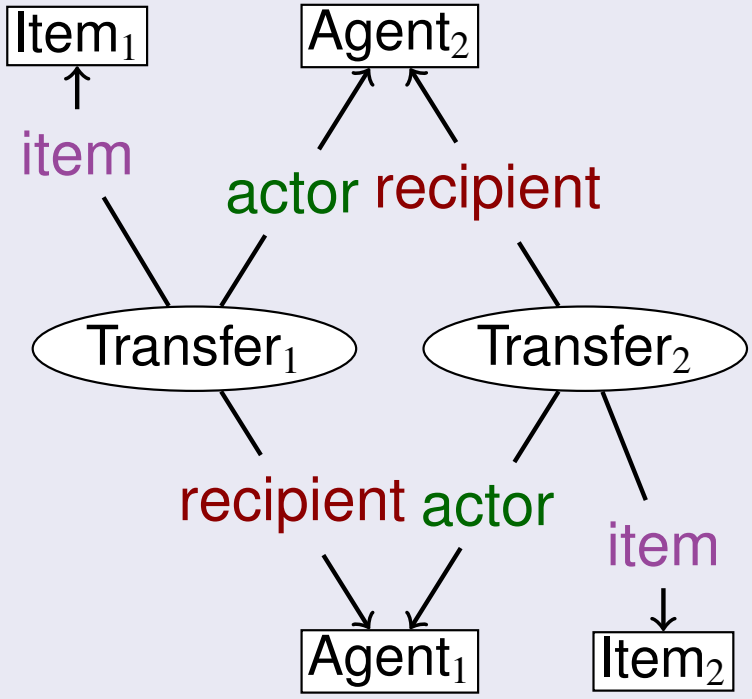
$$\begin{aligned}&\text{transfer}(x,y) \wedge \text{transfer}(x,z) \wedge \\ &\text{actor}(y,u_2) \wedge \text{recipient}(y,u_1) \wedge \\ &\text{actor}(z,u_1) \wedge \text{recipient}(z,u_2) \wedge \\ &\text{item}(y,v_1) \wedge \text{item}(z,v_2) \wedge \\ &O(x) \wedge O(y) \wedge O(z) \wedge O(u_1) \wedge O(u_2) \\ &\wedge O(v_1) \wedge O(v_2) \rightarrow \text{Transaction}(x)\end{aligned}$$

$$\{\text{transfer}(t,t_1), \text{transfer}(t,t_2), \text{actor}(t_1,a_2), \text{recipient}(t_1,a_1), \text{actor}(t_2,a_1), \text{recipient}(t_2,a_2), \text{item}(t_1,i_1), \text{WithItem}(t_2), O(t), O(t_1), O(t_2), O(a_1), O(a_2), O(i_1)\}$$

DL Axiom: $\text{WithItem} \sqsubseteq \exists \text{item}.\top$ $\not\rightarrow \text{Transaction}(t)$

Instance vs Schema-level Reasoning

Transaction Structure



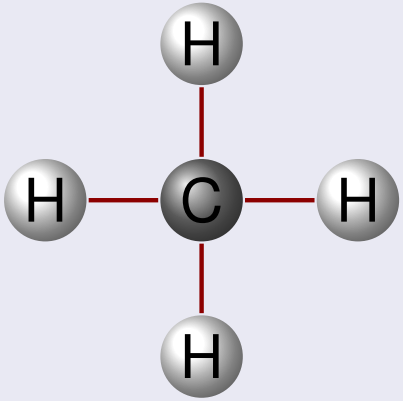
DL-Safe Rule

$$\text{transfer}(x, y) \wedge \text{transfer}(x, z) \wedge \text{actor}(y, u_2) \wedge \text{recipient}(y, u_1) \wedge \text{actor}(z, u_1) \wedge \text{recipient}(z, u_2) \wedge \text{item}(y, v_1) \wedge \text{item}(z, v_2) \wedge O(x) \wedge O(y) \wedge O(z) \wedge O(u_1) \wedge O(u_2) \wedge O(v_1) \wedge O(v_2) \rightarrow \text{Transaction}(x)$$

$\rightsquigarrow$ We want to reason about the structure (schema) of the objects

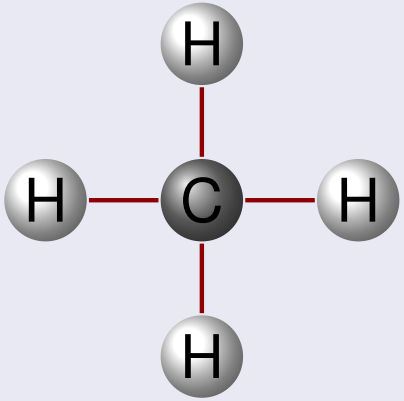
Concept Recognition

Methane Molecule

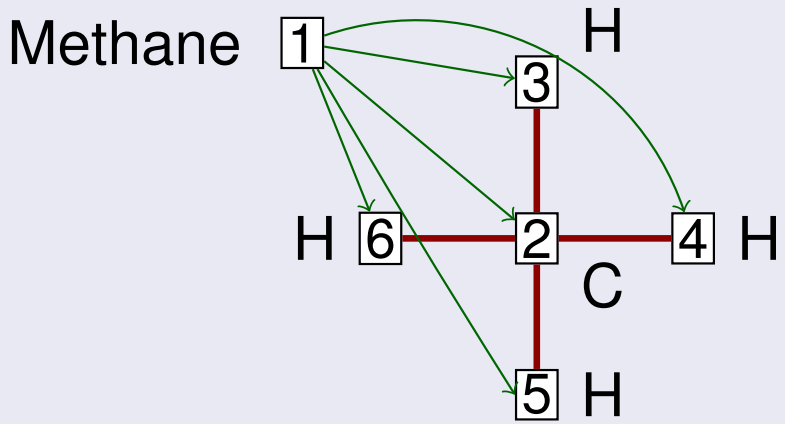


Concept Recognition

Methane Molecule

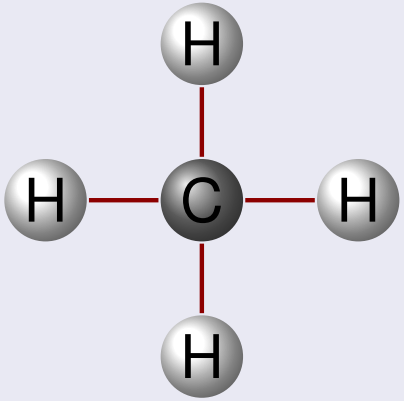


Description Graphs

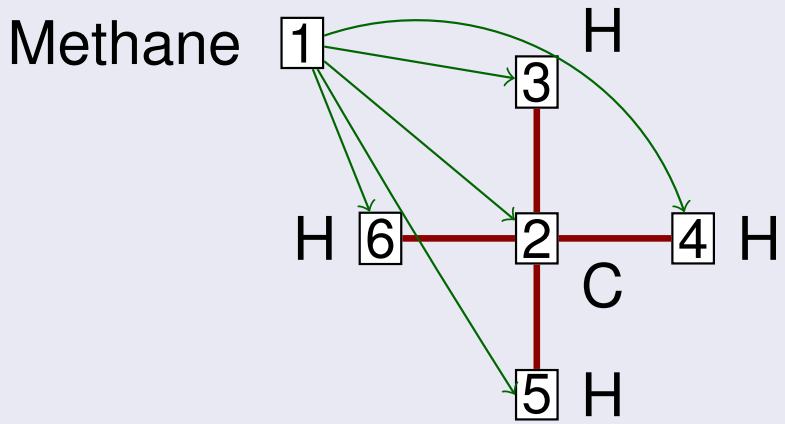


Concept Recognition

Methane Molecule



Description Graphs

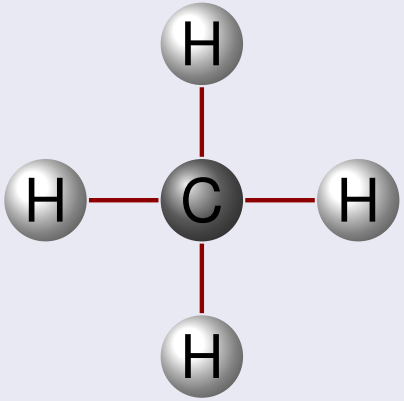


Connected Graph Rule

$$\text{hasAtom}(x, y) \wedge \text{Carbon}(y) \wedge \bigwedge_{i=1}^{4} (\text{hasAtom}(x, z_i) \wedge \text{bond}(y, z_i) \wedge \text{Hydrogen}(z_i)) \rightarrow \text{Methane}(x)$$

Concept Recognition

Methane Molecule



ABox

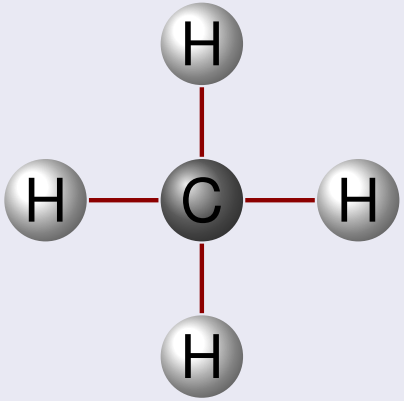
$\{\text{hasAtom}(m, h_1), \text{bond}(c, h_1), \text{Hydrogen}(h_1),$
$\text{hasAtom}(m, h_2), \text{bond}(c, h_2), \text{Hydrogen}(h_2),$
$\text{hasAtom}(m, h_3), \text{bond}(c, h_3), \text{Hydrogen}(h_3),$
$\text{hasAtom}(m, h_4), \text{bond}(c, h_4), \text{Hydrogen}(h_4),$
$\text{hasAtom}(m, c), \text{Carbon}(c)\}$

Connected Graph Rule

$$\text{hasAtom}(x, y) \wedge \text{Carbon}(y) \wedge$$
$$\bigwedge_{i=1}^{4} (\text{hasAtom}(x, z_i) \wedge \text{bond}(y, z_i) \wedge \text{Hydrogen}(z_i))$$
$$\rightarrow \text{Methane}(x)$$

Concept Recognition

Methane Molecule



ABox

$\{\text{hasAtom}(m, h_1), \text{bond}(c, h_1), \text{Hydrogen}(h_1),$
$\text{hasAtom}(m, h_2), \text{bond}(c, h_2), \text{Hydrogen}(h_2),$
$\text{hasAtom}(m, h_3), \text{bond}(c, h_3), \text{Hydrogen}(h_3),$
$\text{hasAtom}(m, h_4), \text{bond}(c, h_4), \text{Hydrogen}(h_4),$
$\text{hasAtom}(m, c), \text{Carbon}(c)\}$

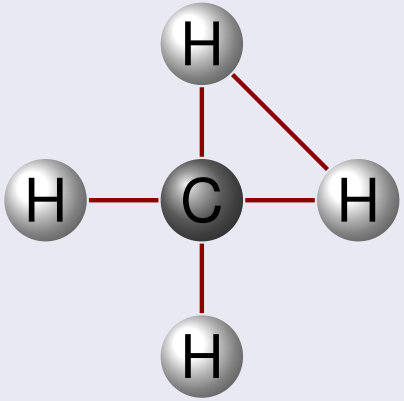
$\rightarrow \text{Methane}(m)$

Connected Graph Rule

$\text{hasAtom}(x, y) \wedge \text{Carbon}(y) \wedge$
$\bigwedge_{i=1}^{4} (\text{hasAtom}(x, z_i) \wedge \text{bond}(y, z_i) \wedge \text{Hydrogen}(z_i))$
$\rightarrow \text{Methane}(x)$

Concept Recognition

Methane Molecule



ABox

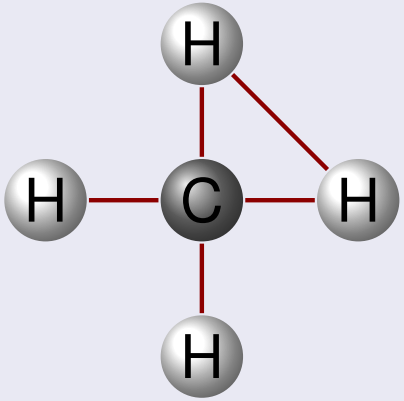
$\{\text{hasAtom}(m, h_1), \text{bond}(c, h_1), \text{Hydrogen}(h_1),$
$\text{hasAtom}(m, h_2), \text{bond}(c, h_2), \text{Hydrogen}(h_2),$
$\text{hasAtom}(m, h_3), \text{bond}(c, h_3), \text{Hydrogen}(h_3),$
$\text{hasAtom}(m, h_4), \text{bond}(c, h_4), \text{Hydrogen}(h_4),$
$\text{hasAtom}(m, c), \text{Carbon}(c), \underline{\text{bond}(h_3, h_4)}\ \}$

Connected Graph Rule

$$\text{hasAtom}(x, y) \wedge \text{Carbon}(y) \wedge$$
$$\bigwedge_{i=1}^{4} (\text{hasAtom}(x, z_i) \wedge \text{bond}(y, z_i) \wedge \text{Hydrogen}(z_i))$$
$$\rightarrow \text{Methane}(x)$$

Concept Recognition

Methane Molecule



ABox

$\{\text{hasAtom}(m, h_1), \text{bond}(c, h_1), \text{Hydrogen}(h_1),$
$\text{hasAtom}(m, h_2), \text{bond}(c, h_2), \text{Hydrogen}(h_2),$
$\text{hasAtom}(m, h_3), \text{bond}(c, h_3), \text{Hydrogen}(h_3),$
$\text{hasAtom}(m, h_4), \text{bond}(c, h_4), \text{Hydrogen}(h_4),$
$\text{hasAtom}(m, c), \text{Carbon}(c), \underline{\text{bond}(h_3, h_4)}\ \}$

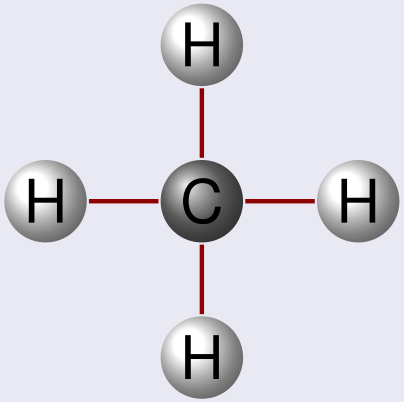
$\rightarrow \text{Methane}(m)$

Connected Graph Rule

$\text{hasAtom}(x, y) \wedge \text{Carbon}(y) \wedge$
$\bigwedge_{i=1}^{4} (\text{hasAtom}(x, z_i) \wedge \text{bond}(y, z_i) \wedge \text{Hydrogen}(z_i))$
$\rightarrow \text{Methane}(x)$

Concept Recognition

Methane Molecule



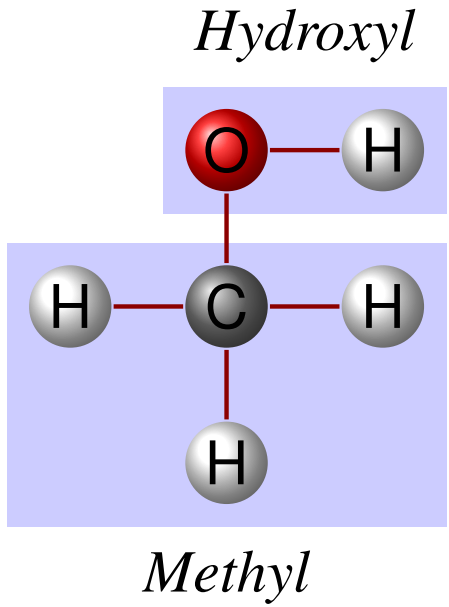
ABox

$\{\text{hasAtom}(m, h_1), \text{bond}(c, h_1), \text{Hydrogen}(h_1),$
$\text{hasAtom}(m, h_2), \text{bond}(c, h_2), \text{Hydrogen}(h_2),$
$\text{hasAtom}(m, h_3), \text{bond}(c, h_3), \text{Hydrogen}(h_3),$
$\text{hasAtom}(m, h_4), \text{bond}(c, h_4), \text{Hydrogen}(h_4),$
$\text{hasAtom}(m, c), \text{Carbon}(c)\ \}$

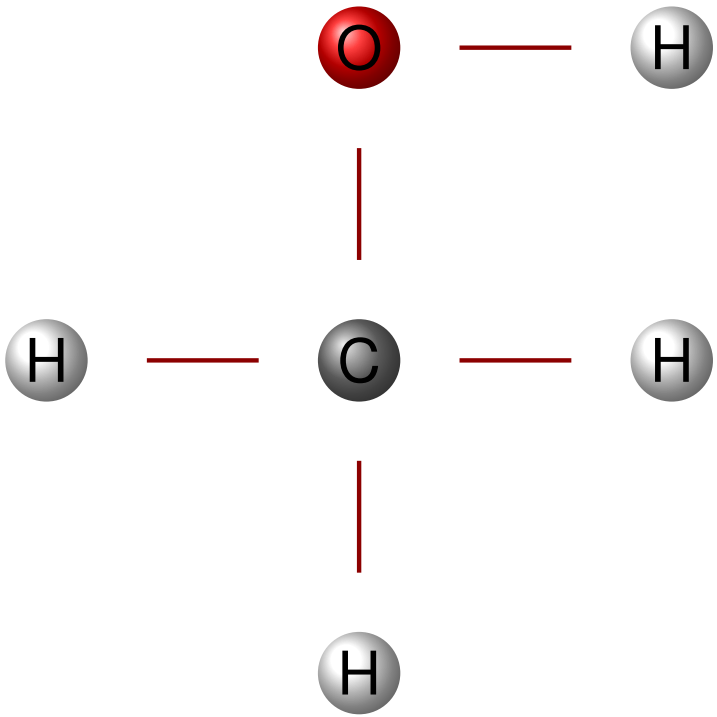
$\rightsquigarrow$ We want concept recognition for objects that have the same structure with the graph—not additional structure

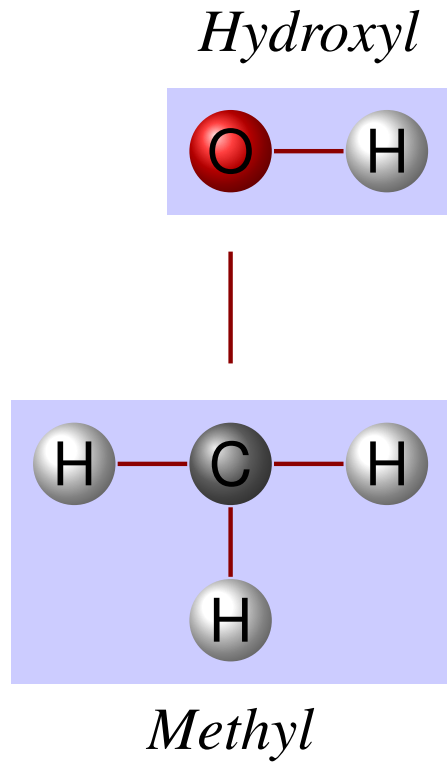
Graph Composition

$\rightsquigarrow$ We want to compose graphs using already defined graphs and additional nodes/edges, e.g. methanole



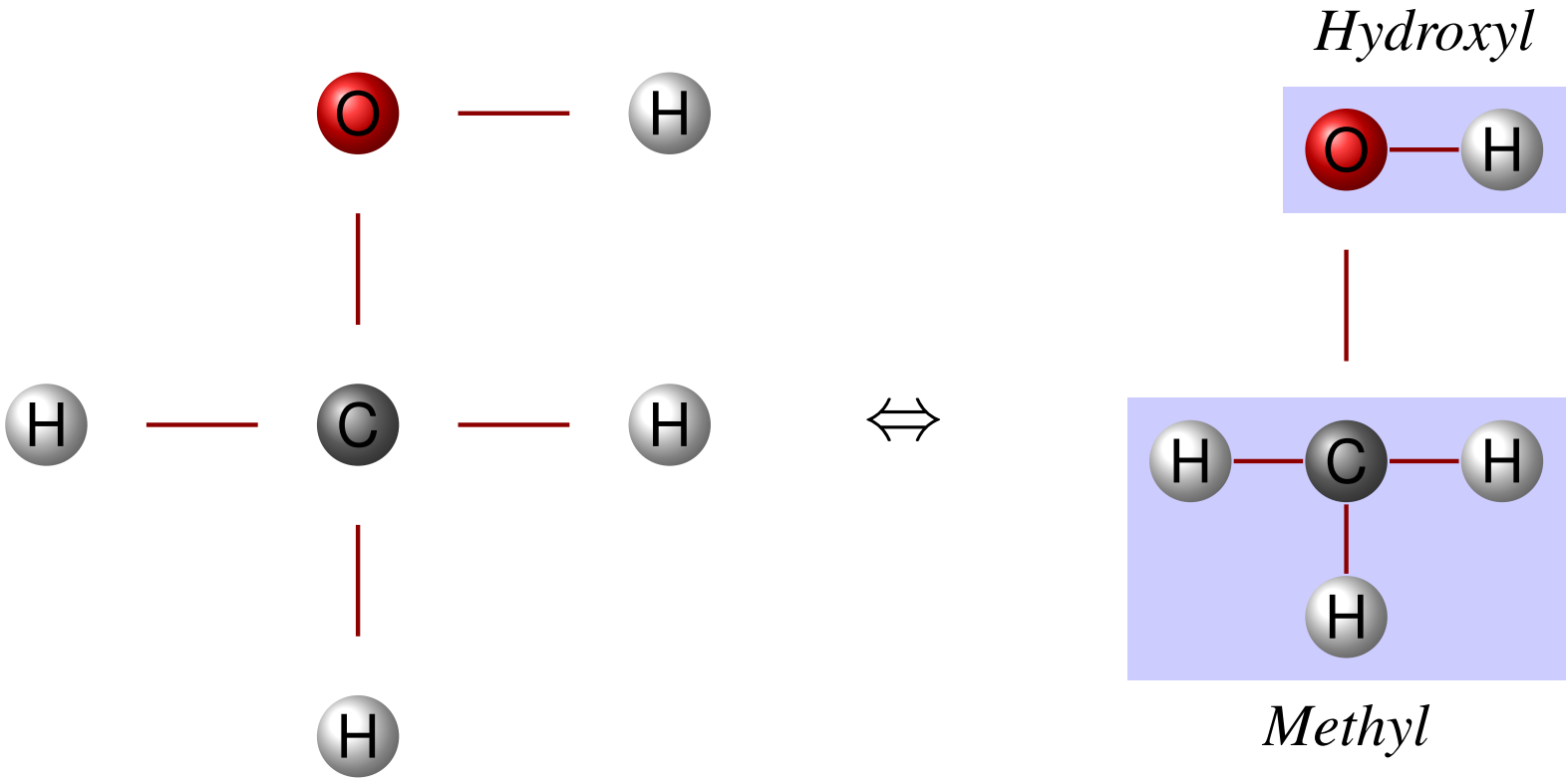
Graph Composition





Graph Composition

⇝ We want the two formulations to be logically equivalent



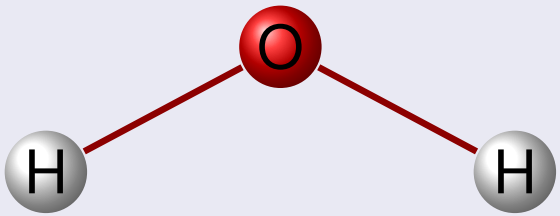
Preliminary Research

- Graph Notation: graph-theoretic language easier for representation

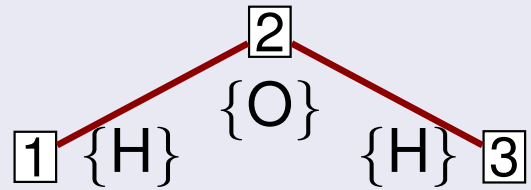
Preliminary Research

- Graph Notation: graph-theoretic language easier for representation

Water Molecule



Graph Notation Example



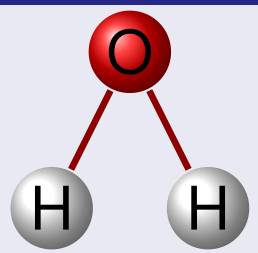
Preliminary Research

- Graph Notation: graph-theoretic language easier for representation
- Graph Logic: logic-based language suitable for reasoning

Preliminary Research

- Graph Notation: graph-theoretic language easier for representation
- Graph Logic: logic-based language suitable for reasoning

Water Molecule



Graph Logic Example

$$\forall x : Water(x) \leftrightarrow \exists o, h_1, h_2 : o \not\approx h_1 \wedge h_1 \not\approx h_2 \wedge h_2 \not\approx o \wedge$$
$$\forall w : GC(x,w) \leftrightarrow ((w = Oxygen(o)) \vee$$
$$(w = Hydrogen(h_1)) \vee (w = Hydrogen(h_2)) \vee$$
$$(w = Bond(o,h_1)) \vee (w = Bond(o,h_2)))$$

Preliminary Research

- Graph Notation: graph-theoretic language easier for representation
- Graph Logic: logic-based language suitable for reasoning

$\rightsquigarrow$ Concept recognition and graph composition requirements

Preliminary Research

- Graph Notation: graph-theoretic language easier for representation
- Graph Logic: logic-based language suitable for reasoning

  $\rightsquigarrow$ Concept recognition and graph composition requirements

- Link between Graph Notation and Graph Logic: ongoing work

Research Plan

- Finalise syntax and semantics
- Sound, complete and terminating reasoning algorith
- Implementation and evaluation using practical test cases:
  - Molecule classification depending on functional group (e.g. manually annotated ChEBI Ontology)
  - Legal assessment using knowledge with recurring diamond-shaped patterns (LKIF Core ontology)

Research Plan

- Finalise syntax and semantics
- Sound, complete and terminating reasoning algorith
- Implementation and evaluation using practical test cases:
  - Molecule classification depending on functional group (e.g. manually annotated ChEBI Ontology)
  - Legal assessment using knowledge with recurring diamond-shaped patterns (LKIF Core ontology)
- Thank you!