The Implementation of the Conditions for the Existence of the Most Specific Generalizations w.r.t. General $\mathcal{EL}$-TBoxes

Adrian Nuradiansyah

Technische Universität Dresden

Supervised by: Anni-Yasmin Turhan

February 12, 2016

Overview

Motivation behind the Implementation

Most Specific Generalization

- Least common subsumer (lcs) and Most Specific Concept (msc).
- The lcs yields a concept that captures all commonalities of pair of concepts (*subsumption*).
- The msc generalizes an individual into a single concept (*instance checking*).

Motivation behind the Implementation

Most Specific Generalization

- Least common subsumer (lcs) and Most Specific Concept (msc).
- The lcs yields a concept that captures all commonalities of pair of concepts (*subsumption*).
- The msc generalizes an individual into a single concept (*instance checking*).
- Support building and maintaining the knowledge base (KB) from bottom up approach.
- Processed, investigated, and added into KB $\Rightarrow$ new knowledge!
- Neither the lcs nor the msc need to exist in general $\mathcal{EL}$-TBox.

Motivation behind the Implementation

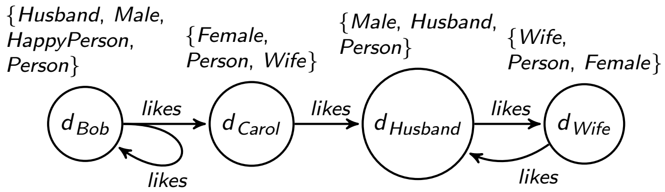
Knowledge Base "Family" and its Canonical Model

$\mathcal{T}_{\textbf{family}_1}$:
{Wife $\sqsubseteq$ Female $\sqcap$ Person $\sqcap$ $\exists$likes.Husband;
HappyPerson $\sqsubseteq$ Person $\sqcap$ $\exists$likes.HappyPerson;
Husband $\sqsubseteq$ Male $\sqcap$ Person $\sqcap$ $\exists$likes.Wife}
$\mathcal{A}_{\textbf{family}_1}$: {*likes*(*bob*,*carol*); *likes*(*bob*,*bob*); Wife(*carol*); HappyPerson(*bob*)}

Motivation behind the Implementation

Knowledge Base "Family" and its Canonical Model

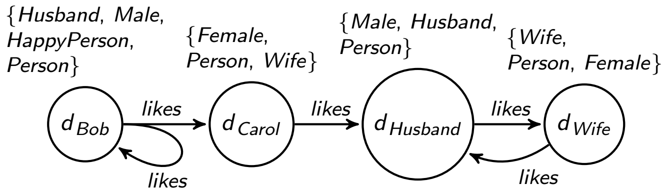
$\mathcal{T}_{\mathbf{family_1}}$:
$\{\text{Wife} \sqsubseteq \text{Female} \sqcap \text{Person} \sqcap \exists \text{likes.Husband};$
$\text{HappyPerson} \sqsubseteq \text{Person} \sqcap \exists \text{likes.HappyPerson};$
$\text{Husband} \sqsubseteq \text{Male} \sqcap \text{Person} \sqcap \exists \text{likes.Wife}\}$
$\mathcal{A}_{\mathbf{family_1}}$: $\{likes(bob, carol);\ likes(bob, bob);\ \text{Wife}(carol);\ \text{HappyPerson}(bob)\}$



Motivation behind the Implementation

Knowledge Base "Family" and its Canonical Model

$\mathcal{T}_{\mathbf{family_1}}$:
$\{\text{Wife} \sqsubseteq \text{Female} \sqcap \text{Person} \sqcap \exists \text{likes}.\text{Husband};$
$\text{HappyPerson} \sqsubseteq \text{Person} \sqcap \exists \text{likes}.\text{HappyPerson};$
$\text{Husband} \sqsubseteq \text{Male} \sqcap \text{Person} \sqcap \exists \text{likes}.\text{Wife}\}$
$\mathcal{A}_{\mathbf{family_1}}$: $\{likes(bob, carol);\ likes(bob, bob);\ \text{Wife}(carol);\ \text{HappyPerson}(bob)\}$

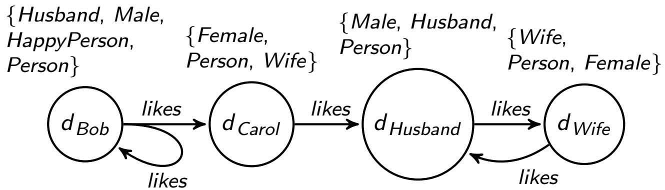


- lcs(`Male`, `Person`)=`⊤`, but there is no lcs for `Husband` and `HappyPerson`
  - `Husband` and `HappyPerson` are cyclic concepts.
- msc(*carol*)=`Wife`, but there is no msc for *bob*
  - `Wife`(*carol*) and `HappyPerson`(*bob*).
  - `Wife` and `HappyPerson` are cyclic concepts.
  - Different results for the msc in a cyclic ontology!

Motivation behind the Implementation

Knowledge Base "Family" and its Canonical Model

$\mathcal{T}_{\text{family}_1}$:
Wife $\sqsubseteq$ Female $\sqcap$ Person $\sqcap$ $\exists$likes.Husband;
Husband $\sqsubseteq$ Male $\sqcap$ Person $\sqcap$ $\exists$likes.Wife
HappyPerson $\sqsubseteq$ Person $\sqcap$ $\exists$likes.HappyPerson;
$\mathcal{A}_{\text{family}_1}$: $likes(Bob, Carol)$; $likes(Bob, Bob)$; Wife$(Carol)$; HappyPerson(Bob)



- How to compute and decide the existence of the most specific generalization w.r.t. general $\mathcal{EL}$ TBox?
- For the sake of simplicity, we only consider the notions related to the least common subsumer in further sections.
- Most specific concept can be defined analogously.

Least Common Subsumer

- A concept E is the **least common subsumer**(lcs) of C and D w.r.t. $\mathcal{T}$ (lcs$_\mathcal{T}$(C, D)) iff:
  - $C \sqsubseteq_\mathcal{T} E$ and $D \sqsubseteq_\mathcal{T} E$
  - For each concept F such that $C \sqsubseteq_\mathcal{T} F$ and $D \sqsubseteq_\mathcal{T} F$, then $E \sqsubseteq_\mathcal{T} F$.

Least Common Subsumer

- A concept E is the **least common subsumer**(lcs) of C and D w.r.t. $\mathcal{T}$ ($\text{lcs}_{\mathcal{T}}(C, D)$) iff:
  - $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$
  - For each concept F such that $C \sqsubseteq_{\mathcal{T}} F$ and $D \sqsubseteq_{\mathcal{T}} F$, then $E \sqsubseteq_{\mathcal{T}} F$.
- We deal with a **general $\mathcal{EL}$ TBox**.
- The computed lcs can be captured in an infinite size.

Least Common Subsumer

- A concept E is the **least common subsumer**(lcs) of C and D w.r.t. $\mathcal{T}$ (lcs$_\mathcal{T}$(C, D)) iff:
  - $C \sqsubseteq_\mathcal{T} E$ and $D \sqsubseteq_\mathcal{T} E$
  - For each concept F such that $C \sqsubseteq_\mathcal{T} F$ and $D \sqsubseteq_\mathcal{T} F$, then $E \sqsubseteq_\mathcal{T} F$.
- We deal with a **general $\mathcal{EL}$ TBox**.
- The computed lcs can be captured in an infinite size.
- Can we obtain a role-depth bounded lcs with a depth k?

Least Common Subsumer

- A concept E is the **least common subsumer**(lcs) of C and D w.r.t. $\mathcal{T}$ (lcs$_\mathcal{T}$(C, D)) iff:
  - $C \sqsubseteq_\mathcal{T} E$ and $D \sqsubseteq_\mathcal{T} E$
  - For each concept F such that $C \sqsubseteq_\mathcal{T} F$ and $D \sqsubseteq_\mathcal{T} F$, then $E \sqsubseteq_\mathcal{T} F$.
- We deal with a **general $\mathcal{EL}$ TBox**.
- The computed lcs can be captured in an infinite size.
- Can we obtain a role-depth bounded lcs with a depth k?
- The **role-depth** ($rd(C)$): the maximal nesting of $\exists$-quantifiers in C.
- Let $k \in \mathbb{N}$ and E, F are the **role-depth bounded concepts** with the role-depth up to k, then E is the **role-depth bounded lcs** (k-lcs$_\mathcal{T}$(C, D)) of C and D w.r.t. $\mathcal{T}$.

Least Common Subsumer

- A concept E is the **least common subsumer**(lcs) of C and D w.r.t. $\mathcal{T}$ (lcs$_\mathcal{T}$(C, D)) iff:
  - $C \sqsubseteq_\mathcal{T} E$ and $D \sqsubseteq_\mathcal{T} E$
  - For each concept F such that $C \sqsubseteq_\mathcal{T} F$ and $D \sqsubseteq_\mathcal{T} F$, then $E \sqsubseteq_\mathcal{T} F$.
- We deal with a **general $\mathcal{EL}$ TBox**.
- The computed lcs can be captured in an infinite size.
- Can we obtain a role-depth bounded lcs with a depth k?
- The **role-depth** ($rd(C)$): the maximal nesting of $\exists$-quantifiers in C.
- Let $k \in \mathbb{N}$ and E, F are the **role-depth bounded concepts** with the role-depth up to k, then E is the **role-depth bounded lcs** (k-lcs$_\mathcal{T}$(C, D)) of C and D w.r.t. $\mathcal{T}$.
- How to obtain this number k?
- How do we know that our k-lcs is our lcs, such that we can check whether the lcs exists or not?

1. Given **two concepts C, D** and a **TBox** $\mathcal{T}$ as the inputs;

Description Logic $\mathcal{EL}$ and TBox

- $\mathcal{EL}$ **concepts** are built by using the following structures:

$$C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C$$

Description Logic $\mathcal{EL}$ and TBox

- $\mathcal{EL}$ **concepts** are built by using the following structures:

$$C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C$$

- An **interpretation** $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of:
  - $\Delta^{\mathcal{I}}$: a non-empty domain.
  - $\cdot^{\mathcal{I}}$ with $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
- The mapping $\cdot^{\mathcal{I}}$ can be extended to $\mathcal{EL}$-concepts

Name	Syntax	Semantic
Top	$\top$	$\Delta^{\mathcal{I}}$
Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Existential Restriction	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e \in \Delta^{\mathcal{I}} : (d,e) \in r^{\mathcal{I}}$ and $e \in C^{\mathcal{I}}\}$

Description Logic $\mathcal{EL}$ and TBox

- $\mathcal{EL}$ **concepts** are built by using the following structures:

$$C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C$$

- An **interpretation** $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of:
  - $\Delta^{\mathcal{I}}$: a non-empty domain.
  - $\cdot^{\mathcal{I}}$ with $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
- The mapping $\cdot^{\mathcal{I}}$ can be extended to $\mathcal{EL}$-concepts

Name	Syntax	Semantic
Top	$\top$	$\Delta^{\mathcal{I}}$
Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Existential Restriction	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e \in \Delta^{\mathcal{I}} : (d,e) \in r^{\mathcal{I}} \text{ and } e \in C^{\mathcal{I}}\}$

- A **(general)** $\mathcal{EL}$ **TBox** $\mathcal{T}$ is a finite set of General Concept Inclusion (GCI) of the form of $C \sqsubseteq D$.
- An interpretation $\mathcal{I}$ **satisfies** a GCI $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
- $\mathcal{I}$ is a **model of** $\mathcal{T}$ iff it satisfies all GCIs in $\mathcal{T}$.
- C is **subsumed** by D w.r.t. $\mathcal{T}$ (denoted by $C \sqsubseteq_{\mathcal{T}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. This reasoning task is called **subsumption**.

Deciding the Existence of the Least Common Subsumer

1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;

Deciding the Existence of the Least Common Subsumer

1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;
2. Compute the **canonical models $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ and $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$** of C and D w.r.t. $\mathcal{T}$;

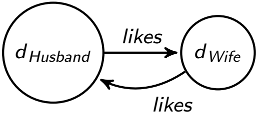
Canonical Model

Canonical Model of Concept w.r.t. TBox

It is denoted by $\mathcal{I}_{C,\mathcal{T}}$. Recall this example:

$\mathcal{T}_{\mathsf{family}_2}$:
Wife $\sqsubseteq$ Female $\sqcap$ Person $\sqcap$ $\exists$likes.Husband;
Husband $\sqsubseteq$ Male $\sqcap$ Person $\sqcap$ $\exists$likes.Wife

$\mathcal{I}_{\mathit{Husband},\mathcal{T}_{\mathsf{family}_2}}$ $\{\mathit{Male}, \mathit{Person}, \mathit{Husband}\}$



$\{\mathit{Wife}, \mathit{Person}, \mathit{Female}\}$

$\mathcal{I}^d$: an interpretation with $d \in \Delta^{\mathcal{I}^d}$ as an initial element such that all $e \in \Delta^{\mathcal{I}^d}$ are **reachable** from d.

Deciding the Existence of the Least Common Subsumer

1. Given **two concepts C, D** and a **TBox** $\mathcal{T}$ as the inputs;
2. Compute the **canonical models** $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ **and** $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$ of C and D w.r.t. $\mathcal{T}$;

Deciding the Existence of the Least Common Subsumer

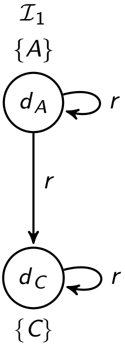
1. Given **two concepts C, D** and a **TBox** $\mathcal{T}$ as the inputs;
2. Compute the **canonical models** $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ **and** $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$ of C and D w.r.t. $\mathcal{T}$;
3. Compute the **product** $\mathcal{I}^{\mathbf{f}}_{\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T}}$ of $\mathcal{I}^{d}_{C,\mathcal{T}}$ and $\mathcal{I}^{e}_{D,\mathcal{T}}$;

Product Interpretation

- Product Interpretation is denoted by $\mathcal{I}_{1\times 2}$

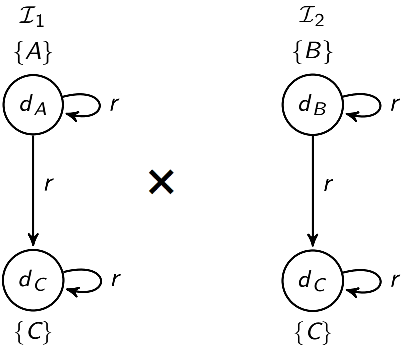
Product Interpretation

- Product Interpretation is denoted by $\mathcal{I}_{1\times 2}$
- **Example:**



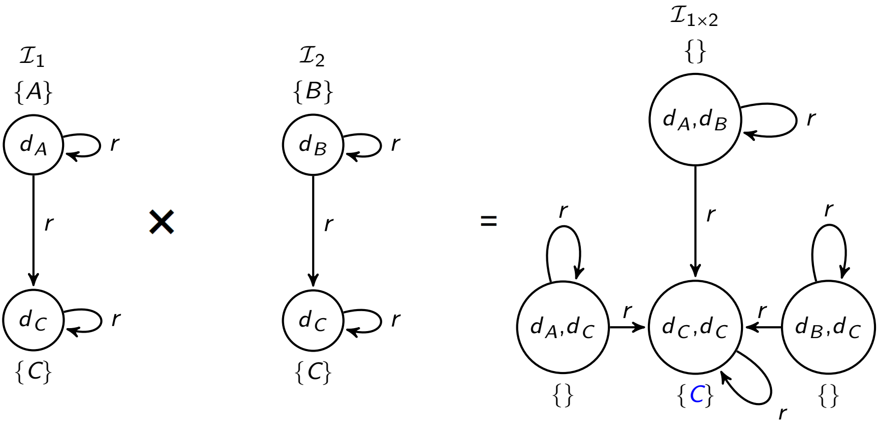
Product Interpretation

- Product Interpretation is denoted by $\mathcal{I}_{1\times 2}$
- **Example:**



Product Interpretation

- Product Interpretation is denoted by $\mathcal{I}_{1\times2}$
- **Example:**



- How to get the product of canonical models in the smaller size?

Deciding the Existence of the Least Common Subsumer

1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;
2. Compute the **canonical models $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ and $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$** of C and D w.r.t. $\mathcal{T}$;
3. Compute the **product $\mathcal{I}^{\mathbf{f}}_{\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T}}$** of $\mathcal{I}^{d}_{C,\mathcal{T}}$ and $\mathcal{I}^{e}_{D,\mathcal{T}}$;

Deciding the Existence of the Least Common Subsumer

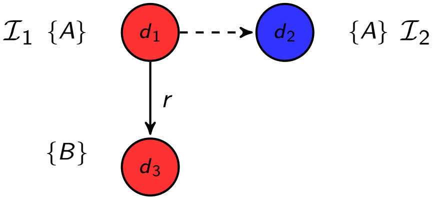
1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;
2. Compute the **canonical models $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ and $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$** of C and D w.r.t. $\mathcal{T}$;
3. Compute the **product $\mathcal{I}^{\mathbf{f}}_{\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T}}$** of $\mathcal{I}^{d}_{C,\mathcal{T}}$ and $\mathcal{I}^{e}_{D,\mathcal{T}}$;
4. Compute the **maximal simulation $\mathcal{S}_{\mathbf{max_1}}$** from $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ to $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ and generate the **set $\mathcal{V}$ of $\simeq$-classes** w.r.t. $\mathcal{S}_{max_1}$;

Simulation Relation

- Subsumption can be characterized by a simulation relation.
- Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be interpretations
  $\mathcal{S} \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ is defined as a **simulation** from $\mathcal{I}_1$ to $\mathcal{I}_2$.
- A simulation $\mathcal{S}_{max}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$ is said to be **maximal** if for all $\mathcal{S}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$, then it holds that $\mathcal{S} \subseteq \mathcal{S}_{max}$.

Simulation Relation

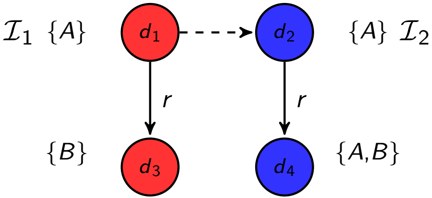
- Subsumption can be characterized by a simulation relation.
- Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be interpretations
  $\mathcal{S} \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ is defined as a **simulation** from $\mathcal{I}_1$ to $\mathcal{I}_2$.
- A simulation $\mathcal{S}_{max}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$ is said to be **maximal** if for all $\mathcal{S}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$, then it holds that $\mathcal{S} \subseteq \mathcal{S}_{max}$.
- **Example:**



$((\mathcal{I}_1, d_1)$ is simulated $(\lesssim)$ by $(\mathcal{I}_2, d_2))$

Simulation Relation

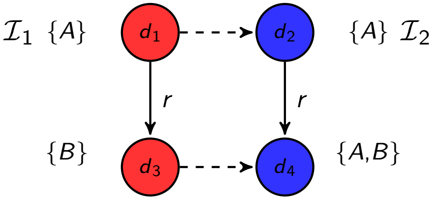
- Subsumption can be characterized by a simulation relation.
- Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be interpretations
  $\mathcal{S} \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ is defined as a **simulation** from $\mathcal{I}_1$ to $\mathcal{I}_2$.
- A simulation $\mathcal{S}_{max}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$ is said to be **maximal** if for all $\mathcal{S}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$, then it holds that $\mathcal{S} \subseteq \mathcal{S}_{max}$.
- **Example:**



$((\mathcal{I}_1, d_1)$ is simulated $(\lesssim)$ by $(\mathcal{I}_2, d_2))$

Simulation Relation

- Subsumption can be characterized by a simulation relation.
- Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be interpretations
  $\mathcal{S} \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ is defined as a **simulation** from $\mathcal{I}_1$ to $\mathcal{I}_2$.
- A simulation $\mathcal{S}_{max}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$ is said to be **maximal** if for all $\mathcal{S}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$, then it holds that $\mathcal{S} \subseteq \mathcal{S}_{max}$.
- **Example:**

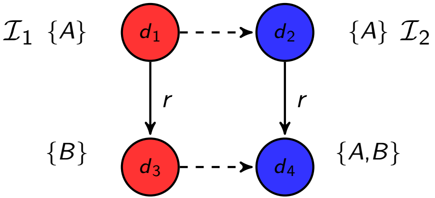


$((\mathcal{I}_1, d_1)$ is simulated $(\lesssim)$ by $(\mathcal{I}_2, d_2))$

$((\mathcal{I}_1, d_3)$ is simulated $(\lesssim)$ by $(\mathcal{I}_2, d_4))$

Simulation Relation

- Subsumption can be characterized by a simulation relation.
- Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be interpretations
  $\mathcal{S} \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ is defined as a **simulation** from $\mathcal{I}_1$ to $\mathcal{I}_2$.
- A simulation $\mathcal{S}_{max}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$ is said to be **maximal** if for all $\mathcal{S}$ from $\mathcal{I}_1$ to $\mathcal{I}_2$, then it holds that $\mathcal{S} \subseteq \mathcal{S}_{max}$.
- **Example:**



$((\mathcal{I}_1, d_1)$ is simulated $(\lesssim)$ by $(\mathcal{I}_2, d_2))$

$((\mathcal{I}_1, d_3)$ is simulated $(\lesssim)$ by $(\mathcal{I}_2, d_4))$

- $(\mathcal{I}_1,d)$ is **equisimilar** to $(\mathcal{I}_2,e)$ (denoted by $(\mathcal{I}_1,d) \simeq (\mathcal{I}_2,e)$) if $(\mathcal{I}_1,d) \lesssim (\mathcal{I}_2,e)$ and $(\mathcal{I}_2,e) \lesssim (\mathcal{I}_1,d)$.
- Let $[d]_{\simeq} := \{e \in \Delta^{\mathcal{I}} \mid (\mathcal{I}, d) \simeq (\mathcal{I}, e)\}$.
- $\mathcal{V}$ as the **set of $\simeq$-classes** w.r.t. a simulation $\mathcal{S}$

Deciding the Existence of the Least Common Subsumer

1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;
2. Compute the **canonical models $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ and $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$** of C and D w.r.t. $\mathcal{T}$;
3. Compute the **product $\mathcal{I}^{\mathbf{f}}_{\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T}}$** of $\mathcal{I}^{d}_{C,\mathcal{T}}$ and $\mathcal{I}^{e}_{D,\mathcal{T}}$;
4. Compute the **maximal simulation $\mathcal{S}_{\mathbf{max_1}}$** from $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ to $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ and generate the **set $\mathcal{V}$ of $\simeq$-classes** w.r.t. $\mathcal{S}_{max_1}$;

Deciding the Existence of the Least Common Subsumer

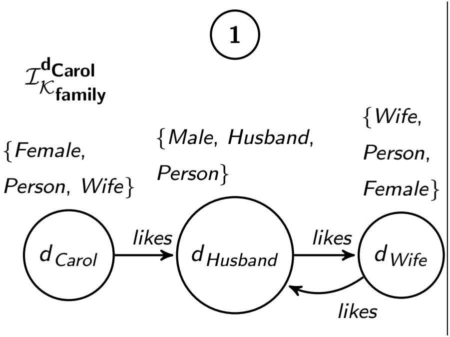
1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;
2. Compute the **canonical models $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ and $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$** of C and D w.r.t. $\mathcal{T}$;
3. Compute the **product $\mathcal{I}^{\mathbf{f}}_{\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T}}$** of $\mathcal{I}^{d}_{C,\mathcal{T}}$ and $\mathcal{I}^{e}_{D,\mathcal{T}}$;
4. Compute the **maximal simulation $\mathcal{S}_{\mathbf{max_1}}$** from $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ to $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ and generate the **set $\mathcal{V}$ of $\simeq$-classes** w.r.t. $\mathcal{S}_{max_1}$;
5. Compute the **equisimulation quotient $\mathcal{I}^{[\mathbf{f}]}_{(\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T})/\simeq}$** of $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ with $\Delta^{\mathcal{I}^{[f]}_{(C,\mathcal{T}\times D,\mathcal{T})/\simeq}} := \mathcal{V}$;

Equisimulation Quotient

- $\mathcal{I}_{/\simeq}$ is an **equisimulation quotient** of $\mathcal{I}$.
- It is computed to:
  - Reduce the number of redundant role-successor nodes
  - Get the smaller number of roles to be traversed during computing the k-characteristic concept.

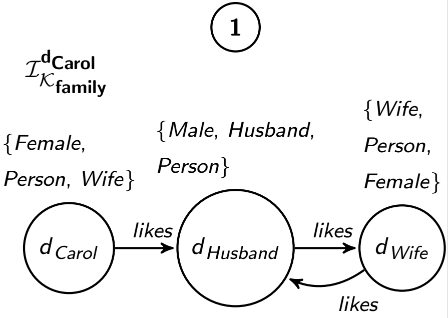
Equisimulation Quotient

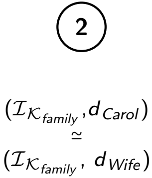
- $\mathcal{I}_{/\simeq}$ is an **equisimulation quotient** of $\mathcal{I}$.
- It is computed to:
  - Reduce the number of redundant role-successor nodes
  - Get the smaller number of roles to be traversed during computing the *k*-characteristic concept.



Equisimulation Quotient

- $\mathcal{I}_{/\simeq}$ is an **equisimulation quotient** of $\mathcal{I}$.
- It is computed to:
  - Reduce the number of redundant role-successor nodes
  - Get the smaller number of roles to be traversed during computing the k-characteristic concept.



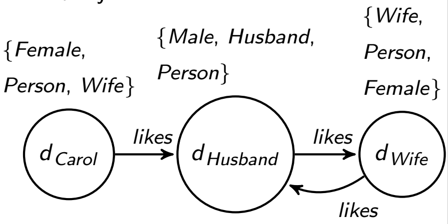


Equisimulation Quotient

- $\mathcal{I}_{/\simeq}$ is an **equisimulation quotient** of $\mathcal{I}$.
- It is computed to:
  - Reduce the number of redundant role-successor nodes
  - Get the smaller number of roles to be traversed during computing the k-characteristic concept.

1

$\mathcal{I}_{\mathcal{K}_{\mathbf{family}}}^{\mathbf{d_{Carol}}}$



2

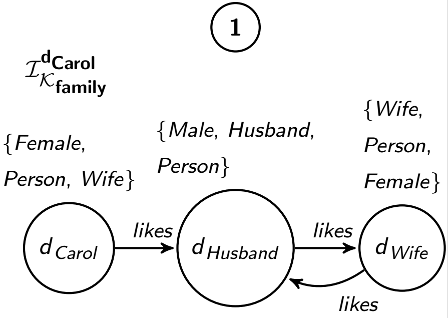
$(\mathcal{I}_{\mathcal{K}_{family}}, d_{Carol}) \simeq (\mathcal{I}_{\mathcal{K}_{family}}, d_{Wife})$

$[d_1]_{\simeq} := \{d_{Husband}\}$

$[d_2]_{\simeq} := \{d_{Carol}, d_{Wife}\}$

Equisimulation Quotient

- $\mathcal{I}_{/\simeq}$ is an **equisimulation quotient** of $\mathcal{I}$.
- It is computed to:
  - Reduce the number of redundant role-successor nodes
  - Get the smaller number of roles to be traversed during computing the k-characteristic concept.

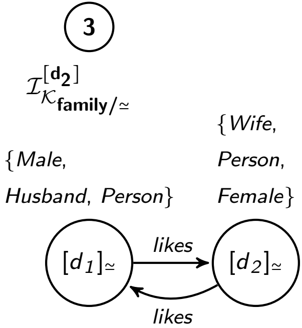


2

$(\mathcal{I}_{\mathcal{K}_{family}}, d_{Carol}) \simeq (\mathcal{I}_{\mathcal{K}_{family}}, d_{Wife})$

$[d_1]_{\simeq} := \{d_{Husband}\}$

$[d_2]_{\simeq} := \{d_{Carol}, d_{Wife}\}$



Deciding the Existence of the Least Common Subsumer

1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;
2. Compute the **canonical models $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ and $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$** of C and D w.r.t. $\mathcal{T}$;
3. Compute the **product $\mathcal{I}^{\mathbf{f}}_{\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T}}$** of $\mathcal{I}^{d}_{C,\mathcal{T}}$ and $\mathcal{I}^{e}_{D,\mathcal{T}}$;
4. Compute the **maximal simulation $\mathcal{S}_{\mathbf{max_1}}$** from $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ to $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ and generate the **set $\mathcal{V}$ of $\simeq$-classes** w.r.t. $\mathcal{S}_{max_1}$;
5. Compute the **equisimulation quotient $\mathcal{I}^{[\mathbf{f}]}_{(\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T})/\simeq}$** of $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ with $\Delta^{\mathcal{I}^{[f]}_{(C,\mathcal{T}\times D,\mathcal{T})/\simeq}} := \mathcal{V}$;

Deciding the Existence of the Least Common Subsumer

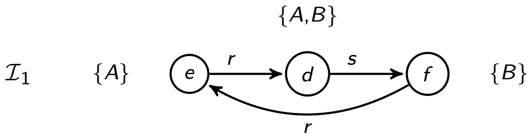
1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;
2. Compute the **canonical models $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ and $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$** of C and D w.r.t. $\mathcal{T}$;
3. Compute the **product $\mathcal{I}^{\mathbf{f}}_{\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T}}$** of $\mathcal{I}^{d}_{C,\mathcal{T}}$ and $\mathcal{I}^{e}_{D,\mathcal{T}}$;
4. Compute the **maximal simulation $\mathcal{S}_{\mathbf{max_1}}$** from $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ to $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ and generate the **set $\mathcal{V}$ of $\simeq$-classes** w.r.t. $\mathcal{S}_{max_1}$;
5. Compute the **equisimulation quotient $\mathcal{I}^{[\mathbf{f}]}_{(\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T})/\simeq}$** of $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ with $\Delta^{\mathcal{I}^{[f]}_{(C,\mathcal{T}\times D,\mathcal{T})/\simeq}} := \mathcal{V}$;
6. Obtain the **number k** as a role-depth for our lcs candidate by computing $\mathbf{k = n^2 + m + 1}$, where:
   - $n = \Delta^{\mathcal{I}^{[f]}_{C,\mathcal{T}\times D,\mathcal{T}/\simeq}}$;
   - $m = max(\{rd(F) \mid F \in sub(\mathcal{T}) \cup \{C,D\}\})$
7. Compute the **k-characteristic concept K** by traversing $\mathcal{I}^{[f]}_{(C,\mathcal{T}\times D,\mathcal{T})/\simeq}$;

k-Characteristic Concept

- Role-Depth bounded concept K with the depth k can be obtained by traversing a canonical model $\mathcal{I}$.
- It is computed recursively by means of **k-characteristic concept** $X^k(\mathcal{I},d)$ with $d \in \Delta^{\mathcal{I}}$.

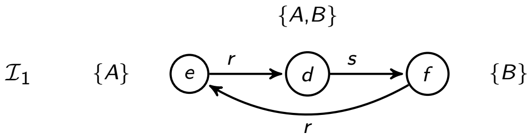
k-Characteristic Concept

- Role-Depth bounded concept K with the depth k can be obtained by traversing a canonical model $\mathcal{I}$.
- It is computed recursively by means of **k-characteristic concept** $X^k(\mathcal{I}, d)$ with $d \in \Delta^{\mathcal{I}}$.
- **Example:**



k-Characteristic Concept

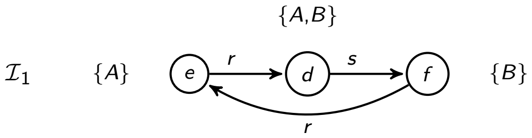
- Role-Depth bounded concept K with the depth k can be obtained by traversing a canonical model $\mathcal{I}$.
- It is computed recursively by means of **k-characteristic concept** $X^k(\mathcal{I},d)$ with $d \in \Delta^{\mathcal{I}}$.
- **Example:**



$\mathbf{k = 0}$; $X^0(\mathcal{I},e) := A$
$\mathbf{k = 1}$; $X^1(\mathcal{I},e) := A \sqcap \exists r.(A \sqcap B)$

k-Characteristic Concept

- Role-Depth bounded concept K with the depth k can be obtained by traversing a canonical model $\mathcal{I}$.
- It is computed recursively by means of **k-characteristic concept** $X^k(\mathcal{I},d)$ with $d \in \Delta^{\mathcal{I}}$.
- **Example:**



k = 0; $X^0(\mathcal{I},e) := A$
k = 1; $X^1(\mathcal{I},e) := A \sqcap \exists r.(A \sqcap B)$
k = 2; $X^2(\mathcal{I},e) := A \sqcap \exists r.(A \sqcap B \sqcap \exists s.B\ )$
k = 3; $X^3(\mathcal{I},e) := A \sqcap \exists r.(A \sqcap B \sqcap \exists s.(B \sqcap \exists r.A))$

Deciding the Existence of the Least Common Subsumer

1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;
2. Compute the **canonical models $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ and $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$** of C and D w.r.t. $\mathcal{T}$;
3. Compute the **product $\mathcal{I}^{\mathbf{f}}_{\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T}}$** of $\mathcal{I}^{d}_{C,\mathcal{T}}$ and $\mathcal{I}^{e}_{D,\mathcal{T}}$;
4. Compute the **maximal simulation $\mathcal{S}_{\mathbf{max_1}}$** from $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ to $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ and generate the **set $\mathcal{V}$ of $\simeq$-classes** w.r.t. $\mathcal{S}_{max_1}$;
5. Compute the **equisimulation quotient $\mathcal{I}^{[\mathbf{f}]}_{(\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T})/\simeq}$** of $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ with $\Delta^{\mathcal{I}^{[f]}_{(C,\mathcal{T}\times D,\mathcal{T})/\simeq}} := \mathcal{V}$;
6. Obtain the **number k** as a role-depth for our lcs candidate by computing $\mathbf{k = n^2 + m + 1}$, where:
   - $n = \Delta^{\mathcal{I}^{[f]}_{C,\mathcal{T}\times D,\mathcal{T}/\simeq}}$;
   - $m = max(\{rd(F) \mid F \in sub(\mathcal{T}) \cup \{C,D\}\})$
7. Compute the **k-characteristic concept K** by traversing $\mathcal{I}^{[f]}_{(C,\mathcal{T}\times D,\mathcal{T})/\simeq}$;

Deciding the Existence of the Least Common Subsumer

1. Given **two concepts C, D** and a **TBox $\mathcal{T}$** as the inputs;
2. Compute the **canonical models $\mathcal{I}^{\mathbf{d}}_{\mathbf{C},\mathcal{T}}$ and $\mathcal{I}^{\mathbf{e}}_{\mathbf{D},\mathcal{T}}$** of C and D w.r.t. $\mathcal{T}$;
3. Compute the **product $\mathcal{I}^{\mathbf{f}}_{\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T}}$** of $\mathcal{I}^{d}_{C,\mathcal{T}}$ and $\mathcal{I}^{e}_{D,\mathcal{T}}$;
4. Compute the **maximal simulation $\mathcal{S}_{\mathbf{max_1}}$** from $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ to $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ and generate the **set $\mathcal{V}$ of $\simeq$-classes** w.r.t. $\mathcal{S}_{max_1}$;
5. Compute the **equisimulation quotient $\mathcal{I}^{[\mathbf{f}]}_{(\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T})/\simeq}$** of $\mathcal{I}^{f}_{C,\mathcal{T}\times D,\mathcal{T}}$ with $\Delta^{\mathcal{I}^{[f]}_{(C,\mathcal{T}\times D,\mathcal{T})/\simeq}} := \mathcal{V}$;
6. Obtain the **number k** as a role-depth for our lcs candidate by computing $\mathbf{k = n^2 + m + 1}$, where:
   - $\text{n} = \Delta^{\mathcal{I}^{[f]}_{C,\mathcal{T}\times D,\mathcal{T}/\simeq}}$;
   - $\text{m} = max(\{rd(F) \mid F \in sub(\mathcal{T}) \cup \{C,D\}\})$
7. Compute the **k-characteristic concept K** by traversing $\mathcal{I}^{[f]}_{(C,\mathcal{T}\times D,\mathcal{T})/\simeq}$;
8. Compute the **canonical model $\mathcal{I}_{\mathbf{K}}$** of K;
9. Check whether **$(\mathcal{I}^{[\mathbf{f}]}_{(\mathbf{C},\mathcal{T}\times\mathbf{D},\mathcal{T})/\simeq}, [\mathbf{f}]_{\simeq})$ is simulated by $(\mathcal{I}^{\mathbf{d_K}}_{\mathbf{K},\mathcal{T}}, \mathbf{d_K})$**. If it is simulated, then K is the $\text{lcs}_{\mathcal{T}}(C, D)$. Otherwise, C and D do not have lcs w.r.t. $\mathcal{T}$.

Implementation of the Algorithm

- Desktop-based application
- Executed in console command-line.
- It is implemented in Java programming language.

Implementation of the Algorithm

- Desktop-based application
- Executed in console command-line.
- It is implemented in Java programming language.

input:

ontology file,
two $\mathcal{EL}$ concepts or
single individual
in OWL format

Implementation of the Algorithm

- Desktop-based application
- Executed in console command-line.
- It is implemented in Java programming language.

input:

ontology file,
two $\mathcal{EL}$ concepts or
single individual
in OWL format

→

Process:

OWL API (OWL 2.0)
+
ELASTIQ library
(computing
canonical model)

Implementation of the Algorithm

- Desktop-based application
- Executed in console command-line.
- It is implemented in Java programming language.

input:

ontology file,
two $\mathcal{EL}$ concepts or
single individual
in OWL format

$\rightarrow$

Process:

OWL API (OWL 2.0)
$+$
ELASTIQ library
(computing
canonical model)

$\rightarrow$

output:

If the lcs/the msc exists,
the returned concept
is represented
of the form of
Manchester OWL syntax

Implementation of the Algorithm

- Desktop-based application
- Executed in console command-line.
- It is implemented in Java programming language.

input:	→	**Process:**	→	**output**:
ontology file, two $\mathcal{EL}$ concepts or single individual in OWL format		OWL API (OWL 2.0) + ELASTIQ library (computing canonical model)		If the lcs/the msc exists, the returned concept is represented of the form of Manchester OWL syntax

Notes:

- The bigger the number of role depth k needed, the bigger the size of computed concepts.
- The presentation of the output of the form of complex concept is quite redundant

Evaluation: Purposes and Test Ontologies

- **Purposes**:
  - To decide the existence of the most specific generalization in cyclic ontologies.
  - To measure the time of computation and analyze the size of computed lcs and msc concepts.

Evaluation: Purposes and Test Ontologies

- **Purposes**:
  - To decide the existence of the most specific generalization in cyclic ontologies.
  - To measure the time of computation and analyze the size of computed lcs and msc concepts.
- **Test Ontologies**
  - Cyclic $\mathcal{EL}$ ontologies that are applied in the real and practical area of knowledge base.
  - Using 10 versions of GeneOntology.

Evaluation: in Cyclic Ontologies

1. **Least Common Subsumer**
   - **Test the cyclicity in all test ontologies**
     - *nnotations1*, *nnotations2*, and *nnotations8* have cyclic concepts.
     - There are 5 cyclic concepts from *nnotations1* and *nnotations2*, respectively.
     - For *nnotations8*, there are only 2 cyclic concepts.

Evaluation: in Cyclic Ontologies

1. **Least Common Subsumer**
   - **Test the cyclicity in all test ontologies**
     - *nnotations1*, *nnotations2*, and *nnotations8* have cyclic concepts.
     - There are 5 cyclic concepts from *nnotations1* and *nnotations2*, respectively.
     - For *nnotations8*, there are only 2 cyclic concepts.
   - **Compute the existence of the LCS of each pair of cyclic concepts w.r.t. their ontologies.**
     - 2 out of 10 pairs of cyclic concepts in both of *nnotations1* *nnotations2* do not have the lcs.
     - One pair of cyclic concepts in *nnotations8* does not have lcs.

Evaluation: in Cyclic Ontologies

1. **Least Common Subsumer**
   - **Test the cyclicity in all test ontologies**
     - *nnotations1*, *nnotations2*, and *nnotations8* have cyclic concepts.
     - There are 5 cyclic concepts from *nnotations1* and *nnotations2*, respectively.
     - For *nnotations8*, there are only 2 cyclic concepts.
   - **Compute the existence of the LCS of each pair of cyclic concepts w.r.t. their ontologies.**
     - 2 out of 10 pairs of cyclic concepts in both of *nnotations1 nnotations2* do not have the lcs.
     - One pair of cyclic concepts in *nnotations8* does not have lcs.

Concept Name 1	Concept Name 2	Ontology	k (role depth)	Result
PomBase_SPBC1685.15c	*PomBase_SPCC18B5.03*	*nnotations1*	148	Yes, the lcs exists
PomBase_SPCC4B3.15	*PomBase_SPBC2F12.13*	*nnotations1*	260	Yes, the lcs exists
PomBase_SPBC1685.15c	*PomBase_SPBC2F12.13*	*nnotations1*	2708	No, the lcs does not exist
PomBase_SPCC18B5.03	*PomBase_SPCC4B3.15*	*nnotations2*	12548	No, the lcs does not exist
PomBase_SPBC1685.15c	*PomBase_SPCC4B3.15*	*nnotations2*	293	Yes, the lcs exists
UniProtKB_D9PTP5	*UniProtKB_Q9GYJ9*	*nnotations8*	260	No, the lcs does not exist

Table: Evaluation for the Existence of the LCS (1)

Evaluation: in Cyclic Ontologies

Concept Name 1	Concept Name 2	Ontology	Size of the LCS	Time of Computation
PomBase_SPBC1685.15c	*PomBase_SPCC18B5.03*	*nnotations1*	48	19,882 s
PomBase_SPCC4B3.15	*PomBase_SPBC2F12.13*	*nnotations1*	75	24,525 s
PomBase_SPBC1685.15c	*PomBase_SPBC2F12.13*	*nnotations1*		52,963 s
PomBase_SPCC18B5.03	*PomBase_SPCC4B3.15*	*nnotations2*		686,037 s
PomBase_SPBC1685.15c	*PomBase_SPCC4B3.15*	*nnotations2*	78	14,936 s
UniProtKB_D9PTP5	*UniProtKB_Q9GYJ9*	*nnotations8*		27,18 s

Table: Evaluation for the Existence of the LCS (2)

Evaluation: in Cyclic Ontologies

Concept Name 1	Concept Name 2	Ontology	Size of the LCS	Time of Computation
PomBase_SPBC1685.15c	*PomBase_SPCC18B5.03*	*nnotations1*	48	19,882 s
PomBase_SPCC4B3.15	*PomBase_SPBC2F12.13*	*nnotations1*	75	24,525 s
PomBase_SPBC1685.15c	*PomBase_SPBC2F12.13*	*nnotations1*		52,963 s
PomBase_SPCC18B5.03	*PomBase_SPCC4B3.15*	*nnotations2*		686,037 s
PomBase_SPBC1685.15c	*PomBase_SPCC4B3.15*	*nnotations2*	78	14,936 s
UniProtKB_D9PTP5	*UniProtKB_Q9GYJ9*	*nnotations8*		27,18 s

Table: Evaluation for the Existence of the LCS (2)

2. Most Specific Concept

- There is no cyclic individual in all test ontologies.
- MSC always exist in this evaluation.

Conclusion and Future Work

- **Conclusions**
  - **Implementing the algorithm** to decide the existence of the lcs and the msc by means of canonical model and simulation relation.
  - Involving the computation of **building the product of canonical model** in the **smaller size**.
    - **Canonical model** with an **initial element**.
    - **Equisimulation quotient** of product of canonical model.
  - Deciding the existence of the lcs and the msc w.r.t. some samples of **GeneOntology version** (**Cyclic ontology**).
    - 3 out of 10 samples of GeneOntology version are cyclic ontologies;
    - Some pairs of cyclic concepts w.r.t. those cyclic ontologies do not have lcs.

Conclusion and Future Work

- **Conclusions**
  - **Implementing the algorithm** to decide the existence of the lcs and the msc by means of canonical model and simulation relation.
  - Involving the computation of **building the product of canonical model** in the **smaller size**.
    - **Canonical model** with an **initial element**.
    - **Equisimulation quotient** of product of canonical model.
  - Deciding the existence of the lcs and the msc w.r.t. some samples of **GeneOntology version** (**Cyclic ontology**).
    - 3 out of 10 samples of GeneOntology version are cyclic ontologies;
    - Some pairs of cyclic concepts w.r.t. those cyclic ontologies do not have lcs.
- **Future Works**
  - Optimizing the simulation algorithm.
  - Simplifying the size of returned concept.
  - **Extended to the other small DL language:** $\mathcal{FL}_0$

The LCS of $\mathcal{FL}_0$ Input Concepts w.r.t. General TBox

Ideas:

- Using a decision procedure similar to EL's case.
- Not using canonical model anymore. Instead, **least functional model** $\mathcal{J}_{C,\mathcal{T}}$ of $\mathcal{FL}_0$ concept C w.r.t. General $\mathcal{FL}_0$ TBox.
- Both of them have similar structure in terms of to label the domain elements and the role-edges.
- But, for the case of least functional model, each role name only connects one element to its single successor element. Due to different types of $\exists$ and $\forall$ semantics.
- $C \sqsubseteq_{\mathcal{T}} D \iff \mathcal{J}_{D,\mathcal{T}} \subseteq \mathcal{J}_{C,\mathcal{T}}$

The LCS of $\mathcal{FL}_0$ Input Concepts w.r.t. General TBox

Research Questions:

- How to characterize subsumption w.r.t. General $\mathcal{FL}_0$ TBox by means of simulation relation?
- How to prove that the canonical model of k-characteristic concept is also a model of TBox?
- How to prove that there exists a k s.t. the canonical model of k-characteristic concept w.r.t. T simulates the product of the canonical models of input concepts?
- Can we also use the same formula, which is $k = n^2 + m + 1$?
  Most probably, it will be different, but the idea will be quite similar to EL's case $\Longrightarrow$ using asynchronous and synchronous elements.

Thank You