Interpolation in Description Logic: A Survey

Frank Wolter

University of Liverpool

Plan

- Introduction to Ontologies/Description Logic
- Interpolation for Rewritings/Beth Definability
- Parallel Interpolation for Decomposition
- Uniform Interpolation

Ontologies

In Computer Science, ontologies $\mathcal{O} = (\boldsymbol{T}, \mathbf{Sig})$ consist of a

a finite axiomatization $\boldsymbol{T}$ *of a logical theory over a signature* Sig.

Sig is the *vocabulary* used to describe a domain of interest and $\boldsymbol{T}$ specifies the *meaning* of the symbols in $\mathbf{Sig}$.

Ontologies

In Computer Science, ontologies $\mathcal{O} = (\boldsymbol{T}, \mathbf{Sig})$ consist of a

a finite axiomatization $\boldsymbol{T}$ *of a logical theory over a signature* Sig.

Sig is the *vocabulary* used to describe a domain of interest and $\boldsymbol{T}$ specifies the *meaning* of the symbols in $\mathbf{Sig}$.

- Ontologies are typically given in description logics (DLs) which underpin the W3C standard OWL.

Ontologies

In Computer Science, ontologies $\mathcal{O} = (\boldsymbol{T}, \textbf{Sig})$ consist of a

a finite axiomatization $\boldsymbol{T}$ *of a logical theory over a signature* Sig.

Sig is the *vocabulary* used to describe a domain of interest and $\boldsymbol{T}$ specifies the *meaning* of the symbols in **Sig**.

- Ontologies are typically given in description logics (DLs) which underpin the W3C standard OWL.
- DLs: well-behaved fragments of first-order logic with convenient syntax.

Ontologies

In Computer Science, ontologies $\mathcal{O} = (\mathcal{T}, \mathbf{Sig})$ consist of a

a finite axiomatization $\mathcal{T}$ of a logical theory over a signature Sig.

Sig is the *vocabulary* used to describe a domain of interest and $\mathcal{T}$ specifies the *meaning* of the symbols in $\mathbf{Sig}$.

- Ontologies are typically given in description logics (DLs) which underpin the W3C standard OWL.
- DLs: well-behaved fragments of first-order logic with convenient syntax.
- Data are not part of the ontology.

Ontologies are often large!

Examples of ontologies in the life sciences and healthcare.

Ontologies are often large!

Examples of ontologies in the life sciences and healthcare.

- SNOMED CT: medical and healthcare ontology used in many countries; $300,000$ terms.

Ontologies are often large!

Examples of ontologies in the life sciences and healthcare.

- SNOMED CT: medical and healthcare ontology used in many countries; $300,000$ terms.
- NCI: National Cancer Institute Thesaurus; 60, 000 terms;

Ontologies are often large!

Examples of ontologies in the life sciences and healthcare.

- SNOMED CT: medical and healthcare ontology used in many countries; $300,000$ terms.
- NCI: National Cancer Institute Thesaurus; 60, 000 terms;
- GO: Gene ontology; more than 50, 000 terms;

Ontologies are often large!

Examples of ontologies in the life sciences and healthcare.

- SNOMED CT: medical and healthcare ontology used in many countries; $300,000$ terms.
- NCI: National Cancer Institute Thesaurus; 60,000 terms;
- GO: Gene ontology; more than 50,000 terms;
- GALEN: medical ontology; lot's of different versions.

Example

$$
\begin{aligned}
\text{Cystic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \exists \text{located\_In.Pancreas} \sqcap \exists \text{has\_Origin.Genetic\_Origin} \\
\text{Genetic\_Fibrosis} &\equiv \text{Fibrosis} \sqcap \exists \text{has\_Origin.Genetic\_Origin} \\
\text{Genetic\_Fibrosis} &\sqsupseteq \text{Fibrosis} \sqcap \exists \text{located\_In.Pancreas} \\
\text{Genetic\_Fibrosis} &\sqsubseteq \text{Genetic\_Disorder} \\
\text{DEFBI\_Gene} &\sqsubseteq \text{Immuno\_Protein\_Gene} \sqcap \exists \text{associated\_With.Cystic\_Fibrosis}
\end{aligned}
$$

Example

$$
\begin{aligned}
\textbf{Cystic\_Fibrosis} &\equiv \textbf{Fibrosis} \sqcap \exists \textbf{located\_In.Pancreas} \sqcap \exists \textbf{has\_Origin.Genetic\_Origin} \\
\textbf{Genetic\_Fibrosis} &\equiv \textbf{Fibrosis} \sqcap \exists \textbf{has\_Origin.Genetic\_Origin} \\
\textbf{Genetic\_Fibrosis} &\sqsupseteq \textbf{Fibrosis} \sqcap \exists \textbf{located\_In.Pancreas} \\
\textbf{Genetic\_Fibrosis} &\sqsubseteq \textbf{Genetic\_Disorder} \\
\textbf{DEFBI\_Gene} &\sqsubseteq \textbf{Immuno\_Protein\_Gene} \sqcap \exists \textbf{associated\_With.Cystic\_Fibrosis}
\end{aligned}
$$

Translation of first axiom into FO:

$$\forall x.(\textbf{Cystic\_Fibrosis}(x) \leftrightarrow C(x))$$

where

$$C(x) = \textbf{Fibrosis}(x) \sqcap \exists y.(\textbf{located\_In}(x, y) \wedge \textbf{Pancreas}(y)) \wedge \exists y.(\textbf{has\_Origin}(x, y) \wedge \textbf{Genetic\_Origin}(y))$$

Description Logics: $\mathcal{EL}$ and $\mathcal{ALC}$

$\mathcal{EL}$*-concepts* are constructed from concept names (unary predicates) $A_1, A_2, \ldots$ and binary relations $r_1, \ldots$

$$C := \top \mid A_i \mid C \sqcap C \mid \exists r_i.C.$$

Description Logics: $\mathcal{EL}$ and $\mathcal{ALC}$

$\mathcal{EL}$-*concepts* are constructed from concept names (unary predicates) $A_1, A_2, \ldots$ and binary relations $r_1, \ldots$

$$C := \quad \top \mid A_i \mid C \sqcap C \mid \exists r_i.C.$$

$\mathcal{ALC}$-*concepts*:

$$C := \quad A_i \mid C \sqcap C \mid \neg C \mid \exists r_i.C \mid \forall r_i.C.$$

Description Logics: $\mathcal{EL}$ and $\mathcal{ALC}$

$\mathcal{EL}$*-concepts* are constructed from concept names (unary predicates) $A_1, A_2, \ldots$ and binary relations $r_1, \ldots$

$$C := \top \mid A_i \mid C \sqcap C \mid \exists r_i.C.$$

$\mathcal{ALC}$*-concepts*:

$$C := A_i \mid C \sqcap C \mid \neg C \mid \exists r_i.C \mid \forall r_i.C.$$

In a *model* $\mathcal{I} = (\Delta^{\mathcal{I}}, A_1^{\mathcal{I}}, \ldots, r_1^{\mathcal{I}}, \ldots)$ the interpretation $C^{\mathcal{I}} \subseteq \Delta$ of a concept $C^{\mathcal{I}}$ is defined inductively:

$$\begin{aligned}
(C_1 \sqcap C_2)^{\mathcal{I}} &= C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\
(\exists r.C)^{\mathcal{I}} &= \{w \in \Delta \mid \exists v\ (w,v) \in r^{\mathcal{I}} \wedge v \in C^{\mathcal{I}}\} \\
(\forall r.C)^{\mathcal{I}} &= \{w \in \Delta \mid \forall v\ (w,v) \in r^{\mathcal{I}} \Rightarrow v \in C^{\mathcal{I}}\}
\end{aligned}$$

Ontologies in Description Logic

A *sentence* is an implication $\boldsymbol{C_1} \sqsubseteq \boldsymbol{C_2}$ between concepts.

Ontologies in Description Logic

A *sentence* is an implication $C_1 \sqsubseteq C_2$ between concepts.

$\mathcal{I} \models C_1 \sqsubseteq C_2$ iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$.

Ontologies in Description Logic

A *sentence* is an implication $C_1 \sqsubseteq C_2$ between concepts.

$\mathcal{I} \models C_1 \sqsubseteq C_2$ iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$.

An *ontology* $\mathcal{O}$ is a finite set of sentences $C_1 \sqsubseteq C_2$. We use $C_1 \equiv C_2$ as an abbreviation for $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$.

Ontologies in Description Logic

A *sentence* is an implication $C_1 \sqsubseteq C_2$ between concepts.

$\mathcal{I} \models C_1 \sqsubseteq C_2$ iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$.

An *ontology* $\mathcal{O}$ is a finite set of sentences $C_1 \sqsubseteq C_2$. We use $C_1 \equiv C_2$ as an abbreviation for $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$.

Deciding whether $\mathcal{O} \models C \sqsubseteq D$ is

- ExpTime-complete for $\mathcal{ALC}$;
- PTime-complete for $\mathcal{EL}$.

Explicit Definitions in Description Logic

Provide definitions of new terms using already defined terms.

$$\textbf{Cystic\_Fibrosis} \equiv \textbf{Fibrosis} \sqcap \exists \textbf{located\_In.Pancreas} \sqcap \exists \textbf{has\_Origin.Genetic\_Origin}$$

Explicit Definitions in Description Logic

Provide definitions of new terms using already defined terms.

$$\textbf{Cystic\_Fibrosis} \equiv \textbf{Fibrosis} \sqcap \exists \textbf{located\_In.Pancreas} \sqcap \exists \textbf{has\_Origin.Genetic\_Origin}$$

Problem: Given an arbitrary ontology $\mathcal{O}$, a signature Σ, and a concept C, does the ontology provide an explicit definition of the C using symbols from Σ only?

Explicit Definitions in Description Logic

Provide definitions of new terms using already defined terms.

$$\textbf{Cystic\_Fibrosis} \equiv \textbf{Fibrosis} \sqcap \exists \textbf{located\_In.Pancreas} \sqcap \exists \textbf{has\_Origin.Genetic\_Origin}$$

Problem: Given an arbitrary ontology $\mathcal{O}$, a signature Σ, and a concept C, does the ontology provide an explicit definition of the C using symbols from Σ only?

Possible aim: rewrite a given ontology into one that (mainly) consists of definitions of the form

$$A \equiv C$$

where A is a concept name. If no cyclic definitions occur, such ontologies are called acyclic TBoxes.

Concrete Application: Ontologies for Querying data

Assume a database schema is given by the signature

$$\Sigma = \{\textbf{diagnosis}, \textbf{heartdisease}\}$$

and a user wants to query $\textbf{heartpatient}(x)$ which is not in the schema.

Concrete Application: Ontologies for Querying data

Assume a database schema is given by the signature

$$\Sigma = \{\textbf{diagnosis}, \textbf{heartdisease}\}$$

and a user wants to query $\textbf{heartpatient}(x)$ which is not in the schema.

Assume the following ontology is given

$$\mathcal{O} = \{\textbf{heartpatient} \equiv \exists \textbf{diagnosis}.\textbf{Heartdisease}\}$$

Then one can equivalently rewrite the query $\textbf{heartpatient}(x)$ into the query

$$\exists \textbf{diagnosis}.\textbf{Heartdisease}$$

which is in the database schema and can be evaluated.

Concrete Application: Ontologies for Querying data

Assume a database schema is given by the signature

$$\Sigma = \{\textbf{diagnosis}, \textbf{heartdisease}\}$$

and a user wants to query $\textbf{heartpatient}(x)$ which is not in the schema.

Assume the following ontology is given

$$\mathcal{O} = \{\textbf{heartpatient} \equiv \exists\textbf{diagnosis}.\textbf{Heartdisease}\}$$

Then one can equivalently rewrite the query $\textbf{heartpatient}(x)$ into the query

$$\exists\textbf{diagnosis}.\textbf{Heartdisease}$$

which is in the database schema and can be evaluated.

Problem: Given an ontology $\mathcal{O}$, a schema Σ, and a query q, can q be equivalently rewritten into a Σ-query?

Explicit Definitions

Let C be a concept, $\mathcal{O}$ an ontology, and Σ a signature. C is explicitly definable using Σ in $\mathcal{O}$ iff there exists a concept D over Σ such that

$$\mathcal{O} \models C \equiv D.$$

Explicit Definitions

Let C be a concept, $\mathcal{O}$ an ontology, and Σ a signature. C is explicitly definable using Σ in $\mathcal{O}$ iff there exists a concept D over Σ such that

$$\mathcal{O} \models C \equiv D.$$

$$\begin{array}{rcl} \textsf{Parent} & \equiv & \exists \textsf{hasChild}.\top \\ \textsf{Parent} & \equiv & \textsf{Father} \sqcup \textsf{Mother} \\ \textsf{Father} & \sqsubseteq & \textsf{Man} \\ \textsf{Mother} & \sqsubseteq & \textsf{Woman} \\ \textsf{Man} & \sqsubseteq & \neg\textsf{Woman} \end{array}$$

Then **Mother** and **Father** are explicitly definable from $\Sigma = \{\textsf{hasChild}, \textsf{Woman}\}$ in $\mathcal{O}$ by

$$\textsf{Mother} \equiv \textsf{Woman} \sqcap \exists \textsf{hasChild}.\top, \quad \textsf{Father} \equiv \textsf{Man} \sqcap \exists \textsf{hasChild}.\top$$

How to test existence and compute explicit definitions?

How to test existence and compute explicit definitions?

C is implicitly definable from Σ in $\mathcal{O}$ iff for any two models $\mathcal{I}$ and $\mathcal{J}$ with the same domain and the same interpretation of Σ-symbols,

$$C^{\mathcal{I}} = C^{\mathcal{J}}.$$

How to test existence and compute explicit definitions?

C is implicitly definable from Σ in $\mathcal{O}$ iff for any two models $\mathcal{I}$ and $\mathcal{J}$ with the same domain and the same interpretation of Σ-symbols,

$$C^{\mathcal{I}} = C^{\mathcal{J}}.$$

A Logic has the Beth Definability Property (projective) if every C that is implicitly definable, is is explicitly definable as well.

How to test existence and compute explicit definitions?

C is implicitly definable from Σ in $\mathcal{O}$ iff for any two models $\mathcal{I}$ and $\mathcal{J}$ with the same domain and the same interpretation of Σ-symbols,

$$C^{\mathcal{I}} = C^{\mathcal{J}}.$$

A Logic has the Beth Definability Property (projective) if every C that is implicitly definable, is is explicitly definable as well.

C is implicitly definable using Σ in $\mathcal{O}$ iff

$$\mathcal{O} \cup \mathcal{O}' \models C \equiv C'$$

where $'$ is the result of replacing non-Σ-symbols by fresh symbols.

Interpolants as explicit definitions

Assume $\mathcal{O} \cup \mathcal{O}' \models C \sqsubseteq C'$. Then there exists an interpolant I with

- $\textbf{sig}(I) \subseteq \textbf{sig}(C, \mathcal{O}) \cap \textbf{sig}(C', \mathcal{O}')$.
- $\mathcal{O} \cup \mathcal{O}' \models C \sqsubseteq I$.
- $\mathcal{O} \cup \mathcal{O}' \models I \sqsubseteq C'$.

Tableau-based algorithms for computing I for various DLs (including $\mathcal{ALC}$) developed in recent JAIR paper.

Issues: Size and Existence

In $\mathcal{ALC}$ minimal interpolants can be of double exponential size.

Issues: Size and Existence

In $\mathcal{ALC}$ minimal interpolants can be of double exponential size.

The OWL standard (and many ontologies) contains inclusions $R \sqsubseteq S$ between binary predicates.

Issues: Size and Existence

In $\mathcal{ALC}$ minimal interpolants can be of double exponential size.

The OWL standard (and many ontologies) contains inclusions $R \sqsubseteq S$ between binary predicates.

Let S, R_1, R_2 be binary relations. and consider ontology $\mathcal{O}$.

$$
\begin{aligned}
S &\sqsubseteq R_1 \\
S &\sqsubseteq R_2 \\
\exists R_1.A \sqcap \forall S.\bot &\sqsubseteq \forall R_2.\neg A \\
\exists R_1.\neg A \sqcap \forall S.\bot &\sqsubseteq \forall R_2.A
\end{aligned}
$$

Issues: Size and Existence

In $\mathcal{ALC}$ minimal interpolants can be of double exponential size.

The OWL standard (and many ontologies) contains inclusions $R \sqsubseteq S$ between binary predicates.

Let S, R_1, R_2 be binary relations. and consider ontology $\mathcal{O}$.

$$
\begin{aligned}
S &\sqsubseteq R_1 \\
S &\sqsubseteq R_2 \\
\exists R_1.A \sqcap \forall S.\bot &\sqsubseteq \forall R_2.\neg A \\
\exists R_1.\neg A \sqcap \forall S.\bot &\sqsubseteq \forall R_2.A
\end{aligned}
$$

$\exists S.\top$ is explicitly defined using $\{R_1, R_2\}$ by

$$\exists S.\top \equiv \exists (R_1 \cap R_2).\top.$$

This is, however, not in the OWL standard.

Decompositions of Ontologies

Assume $\mathcal{O}$ is an ontology.

A partition $\Sigma_1, \ldots, \Sigma_n$ of $\textbf{sig}(\mathcal{O})$ is a decomposition of $\mathcal{O}$ if there are $\mathcal{O}_1, \ldots, \mathcal{O}_n$ such that

- $\textbf{sig}(\mathcal{O}_i) \subseteq \Sigma_i$;
- $\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_n \equiv \mathcal{O}$.

Decompositions of Ontologies

Assume $\mathcal{O}$ is an ontology and $\Delta \subseteq \mathbf{sig}(\mathcal{O})$ a signature.

A partition $\Sigma_1, \ldots, \Sigma_n$ of $\mathbf{sig}(\mathcal{O}) \setminus \Delta$ is a Δ-decomposition of $\mathcal{O}$ if there are $\mathcal{O}_1, \ldots, \mathcal{O}_n$ such that

- $\mathbf{sig}(\mathcal{O}_i) \subseteq \Sigma_i \cup \Delta$;
- $\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_n \equiv \mathcal{O}$.

Decompositions of Ontologies

Assume $\mathcal{O}$ is an ontology and $\Delta \subseteq \mathbf{sig}(\mathcal{O})$ a signature.

A partition $\Sigma_1, \ldots, \Sigma_n$ of $\mathbf{sig}(\mathcal{O}) \setminus \Delta$ is a Δ-decomposition of $\mathcal{O}$ if there are $\mathcal{O}_1, \ldots, \mathcal{O}_n$ such that

- $\mathbf{sig}(\mathcal{O}_i) \subseteq \Sigma_i \cup \Delta$;
- $\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_n \equiv \mathcal{O}$.

Problems:

- Is there a unique finest Δ-decomposition?

Decompositions of Ontologies

Assume $\mathcal{O}$ is an ontology and $\Delta \subseteq \mathbf{sig}(\mathcal{O})$ a signature.

A partition $\Sigma_1, \ldots, \Sigma_n$ of $\mathbf{sig}(\mathcal{O}) \setminus \Delta$ is a Δ-decomposition of $\mathcal{O}$ if there are $\mathcal{O}_1, \ldots, \mathcal{O}_n$ such that

- $\mathbf{sig}(\mathcal{O}_i) \subseteq \Sigma_i \cup \Delta$;
- $\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_n \equiv \mathcal{O}$.

Problems:

- Is there a unique finest Δ-decomposition?
- Do decompositions in a given DL coincide with decompositions in SO?

Decompositions of Ontologies

Assume $\mathcal{O}$ is an ontology and $\Delta \subseteq \textbf{sig}(\mathcal{O})$ a signature.

A partition $\Sigma_1, \ldots, \Sigma_n$ of $\textbf{sig}(\mathcal{O}) \setminus \Delta$ is a Δ-decomposition of $\mathcal{O}$ if there are $\mathcal{O}_1, \ldots, \mathcal{O}_n$ such that

- $\textbf{sig}(\mathcal{O}_i) \subseteq \Sigma_i \cup \Delta$;
- $\mathcal{O}_1 \cup \cdots \cup \mathcal{O}_n \equiv \mathcal{O}$.

Problems:

- Is there a unique finest Δ-decomposition?
- Do decompositions in a given DL coincide with decompositions in SO?
- Compute (unique finest) decomposition.

Parallel Interpolation (without Δ)

Assume $\mathcal{O}_1, \mathcal{O}_2$ and α with $\mathcal{O}_1 \cup \mathcal{O}_2 \models \alpha$ are given.

Parallel Interpolation (without Δ)

Assume $\mathcal{O}_1, \mathcal{O}_2$ and α with $\mathcal{O}_1 \cup \mathcal{O}_2 \models \alpha$ are given.

A pair

$$\mathcal{O}_1', \mathcal{O}_2'$$

is a parallel interpolant of $\mathcal{O}_1, \mathcal{O}_2$ and α if

- $\mathcal{O}_1' \cup \mathcal{O}_2' \models \alpha$;
- $\mathcal{O}_i \models \mathcal{O}_i'$;
- $\mathsf{sig}(\mathcal{O}_i') \subseteq \mathsf{sig}(\mathcal{O}_i) \cap \mathsf{sig}(\alpha)$;

Parallel Interpolation (without Δ)

Assume $\mathcal{O}_1, \mathcal{O}_2$ and α with $\mathcal{O}_1 \cup \mathcal{O}_2 \models \alpha$ are given.

A pair

$$\mathcal{O}'_1, \mathcal{O}'_2$$

is a parallel interpolant of $\mathcal{O}_1, \mathcal{O}_2$ and α if

- $\mathcal{O}'_1 \cup \mathcal{O}'_2 \models \alpha$;
- $\mathcal{O}_i \models \mathcal{O}'_i$;
- $\mathbf{sig}(\mathcal{O}'_i) \subseteq \mathbf{sig}(\mathcal{O}_i) \cap \mathbf{sig}(\alpha)$;

Parallel interpolation: parallel interpolant exists if $\mathcal{O}_1 \cup \mathcal{O}_1 \models \alpha$, $\mathbf{sig}(\mathcal{O}_1) \cap \mathbf{sig}(\mathcal{O}_2) = \emptyset$, and $\mathcal{O}_1, \mathcal{O}_2$ have the same consequences over empty signature.

Parallel Interpolation

Non-trivial to prove because not closed under Boolean operators: $\mathcal{ALC}$ and $\mathcal{EL}$ have parallel interpolation.

Parallel Interpolation

Non-trivial to prove because not closed under Boolean operators: $\mathcal{ALC}$ and $\mathcal{EL}$ have parallel interpolation.

Parallel interpolation implies:

- There is a unique finest Δ-decomposition.
- Decompositions in DL coincide with decompositions in SO.
- Interpolants are axiomatizations of components.

Uniform interpolation

Standard interpolation: if $\mathcal{O} \models \alpha$, then there exists $\mathcal{O}'$ with

- $\textbf{sig}(\mathcal{O}') \subseteq \textbf{sig}(\mathcal{O}) \cap \textbf{sig}(\alpha)$;
- $\mathcal{O} \models \mathcal{O}'$;
- $\mathcal{O}' \models \alpha$.

Uniform interpolation

Standard interpolation: if $\mathcal{O} \models \alpha$, then there exists $\mathcal{O}'$ with

- $\mathbf{sig}(\mathcal{O}') \subseteq \mathbf{sig}(\mathcal{O}) \cap \mathbf{sig}(\alpha)$;
- $\mathcal{O} \models \mathcal{O}'$;
- $\mathcal{O}' \models \alpha$.

A uniform interpolant is an interpolant for all α with $\mathbf{sig}(\alpha) \cap \mathbf{sig}(\mathcal{O}) \subseteq \Sigma$ for a fixed Σ.

I understand that Craig really wanted uniform interpolants.

Uniform interpolation

Standard interpolation: if $\mathcal{O} \models \alpha$, then there exists $\mathcal{O}'$ with

- $\mathbf{sig}(\mathcal{O}') \subseteq \mathbf{sig}(\mathcal{O}) \cap \mathbf{sig}(\alpha)$;
- $\mathcal{O} \models \mathcal{O}'$;
- $\mathcal{O}' \models \alpha$.

A uniform interpolant is an interpolant for all α with $\mathbf{sig}(\alpha) \cap \mathbf{sig}(\mathcal{O}) \subseteq \Sigma$ for a fixed Σ.

I understand that Craig really wanted uniform interpolants.

Definition: A uniform Σ-interpolant $\mathcal{O}'$ of $\mathcal{O}$ has the following properties:

- $\mathcal{O} \models \mathcal{O}'$;
- $\mathbf{sig}(\mathcal{O}') \subseteq \Sigma$;
- if $\mathcal{O} \models \alpha$ and $\mathbf{sig}(\alpha) \cap \mathbf{sig}(\mathcal{O}) \subseteq \Sigma$, then $\mathcal{O}' \models \alpha$.

In FO (and DLs) uniform interpolants do not always exist

Let

$$\mathcal{O} = \{A \sqsubseteq B, B \sqsubseteq \exists r.B\}$$

and $\Sigma = \{A, r\}$.

In FO (and DLs) uniform interpolants do not always exist

Let

$$\mathcal{O} = \{A \sqsubseteq B, B \sqsubseteq \exists r.B\}$$

and $\Sigma = \{A, r\}$.

Infinite "uniform Σ-interpolant" given by

$$\mathcal{O}' = \{A \sqsubseteq \text{infinite } r\text{-chain}\}$$

This cannot be axiomatized in FO or standard DLs.

In FO (and DLs) uniform interpolants do not always exist

Let

$$\mathcal{O} = \{A \sqsubseteq B, B \sqsubseteq \exists r.B\}$$

and $\Sigma = \{A, r\}$.

Infinite "uniform Σ-interpolant" given by

$$\mathcal{O}' = \{A \sqsubseteq \text{infinite } r\text{-chain}\}$$

This cannot be axiomatized in FO or standard DLs.

$\mathcal{ALC}\mu$ (modal μ-calculus) is an extension of $\mathcal{ALC}$ with uniform interpolation.

Why uniform interpolants of ontologies?

- Re-use: from an ontology of size 300 000 one typically needs only a small fraction of its terms for an application. Work with the corresponding uniform Σ-interpolant.

Why uniform interpolants of ontologies?

- Re-use: from an ontology of size 300 000 one typically needs only a small fraction of its terms for an application. Work with the corresponding uniform Σ-interpolant.
- Ontology summary: a uniform interpolant summarises what an ontology says about Σ.

Why uniform interpolants of ontologies?

- Re-use: from an ontology of size 300 000 one typically needs only a small fraction of its terms for an application. Work with the corresponding uniform Σ-interpolant.
- Ontology summary: a uniform interpolant summarises what an ontology says about Σ.
- Predicate-Hiding: if one does not want to publish what the ontologies says about non-Σ-symbols.

Uniform interpolants for acyclic $\mathcal{EL}$-TBoxes

For acyclic $\mathcal{EL}$-TBoxes, uniform interpolants always exist.

Uniform interpolants for acyclic $\mathcal{EL}$-TBoxes

For acyclic $\mathcal{EL}$-TBoxes, uniform interpolants always exist.

In the worst case, exponentially many axioms of exponential size required.

Uniform interpolants for acyclic $\mathcal{EL}$-TBoxes

For acyclic $\mathcal{EL}$-TBoxes, uniform interpolants always exist.

In the worst case, exponentially many axioms of exponential size required.

Proof that exponentially many axioms are required: Let

$$\mathcal{O} = \{A \equiv B_1 \sqcap \cdots \sqcap B_n\} \cup \{A_{ij} \sqsubseteq B_i \mid 1 \leq i, j \leq n\}.$$

and

$$\Sigma = \{A\} \cup \{A_{ij} \mid 1 \leq i, j \leq n\}.$$

Uniform interpolants for acyclic $\mathcal{EL}$-TBoxes

For acyclic $\mathcal{EL}$-TBoxes, uniform interpolants always exist.

In the worst case, exponentially many axioms of exponential size required.

Proof that exponentially many axioms are required: Let

$$\mathcal{O} = \{A \equiv B_1 \sqcap \cdots \sqcap B_n\} \cup \{A_{ij} \sqsubseteq B_i \mid 1 \leq i, j \leq n\}.$$

and

$$\Sigma = \{A\} \cup \{A_{ij} \mid 1 \leq i, j \leq n\}.$$

Then

$$\mathcal{O}' = \{A_{1j_1} \sqcap \cdots \sqcap A_{nj_n} \sqsubseteq A \mid 1 \leq j_1, \ldots, j_n \leq n\}$$

is a smallest uniform Σ-interpolant.

Exponential size axioms in uniform interpolants

Let

$$\mathcal{O} = \{A_i \sqsubseteq \exists r.A_{i+1} \sqcap \exists s.A_{i+1} \mid i \leq n\}$$

and $\Sigma = \{A_0, r, s\}$.

Exponential size axioms in uniform interpolants

Let

$$\mathcal{O} = \{A_i \sqsubseteq \exists r.A_{i+1} \sqcap \exists s.A_{i+1} \mid i \leq n\}$$

and $\Sigma = \{A_0, r, s\}$.

Then

$$\mathcal{O}' = \{A_0 \sqsubseteq \text{ binary tree of depth } n\}$$

is smallest uniform Σ-interpolant.

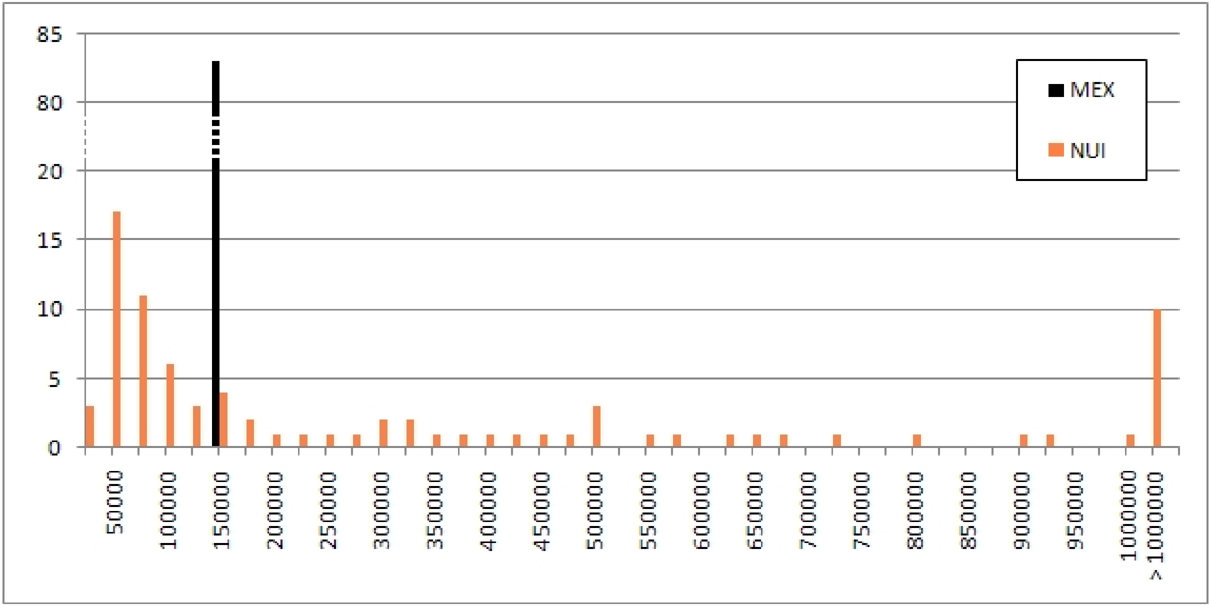
Computing uniform interpolants for SNOMED CT and NCI

100 randomly generated signatures.

$\lvert\Sigma\rvert$	SNOMED CT	$\lvert\Sigma\rvert$	NCI
2 000	**100.0%**	**5 000**	**97.0%**
3 000	**92.2%**	**10 000**	**81.1%**
4 000	**67.0%**	**15 000**	**72.0%**
5 000	**60.0%**	**20 000**	**59.2%**

Comparing the size of Σ-modules and Σ-interpolants for SNOMED CT

- Signatures containing $3\,000$ concept names and 20 role names



Σ-module

Let $\mathcal{O}$ be an ontology and Σ a signature.

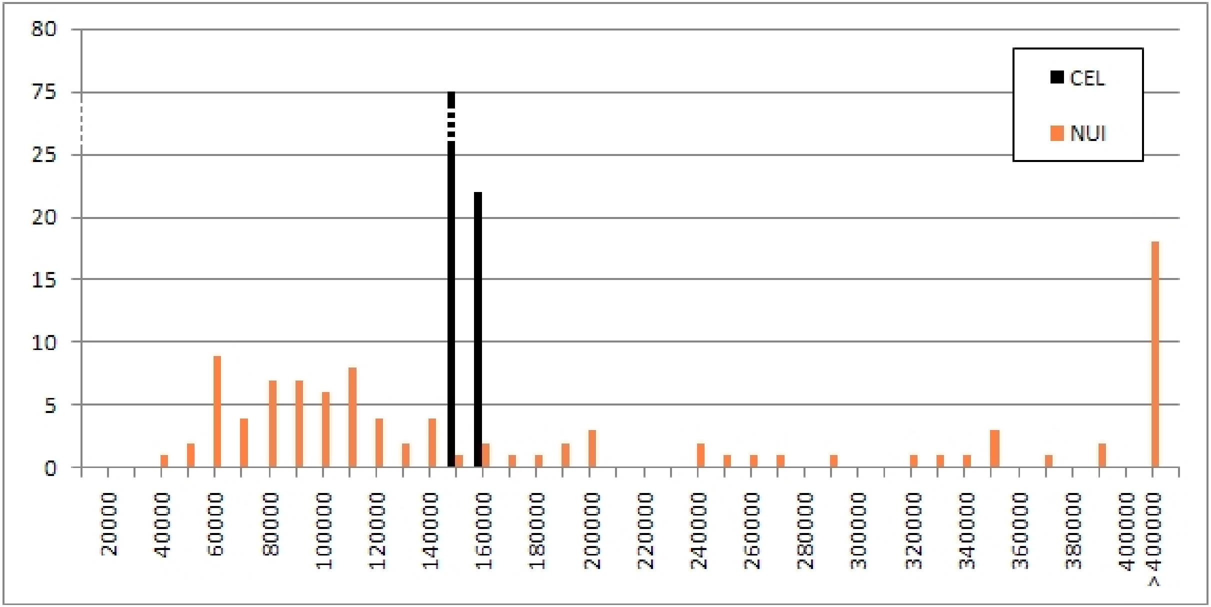
A Σ-module $\mathcal{M} \subseteq \mathcal{O}$ has the following property:

$$\mathcal{M} \models \alpha \quad \Leftrightarrow \quad \mathcal{O} \models \alpha$$

for all α over Σ.

Comparing the size of Σ-modules and Σ-interpolants for NCI

- Σ contains $7\,000$ concept names and 20 role names



Complexity and Size of Uniform Interpolants

In $\mathcal{EL}$ the existence of uniform interpolation for a given $\mathcal{O}$ and Σ is ExpTime complete.

Complexity and Size of Uniform Interpolants

In $\mathcal{EL}$ the existence of uniform interpolation for a given $\mathcal{O}$ and Σ is ExpTime complete.

Uniform interpolants can be of triple exponential size in the worst case.

Complexity and Size of Uniform Interpolants

In $\mathcal{EL}$ the existence of uniform interpolation for a given $\mathcal{O}$ and Σ is ExpTime complete.

Uniform interpolants can be of triple exponential size in the worst case.

In $\mathcal{ALC}$ the existence of uniform interpolation for a given $\mathcal{O}$ and Σ is 2ExpTime complete.

Complexity and Size of Uniform Interpolants

In $\mathcal{EL}$ the existence of uniform interpolation for a given $\mathcal{O}$ and Σ is ExpTime complete.

Uniform interpolants can be of triple exponential size in the worst case.

In $\mathcal{ALC}$ the existence of uniform interpolation for a given $\mathcal{O}$ and Σ is 2ExpTime complete.

Uniform interpolants can be of triple exponential size in the worst case.

Complexity and Size of Uniform Interpolants

In $\mathcal{EL}$ the existence of uniform interpolation for a given $\mathcal{O}$ and Σ is ExpTime complete.

Uniform interpolants can be of triple exponential size in the worst case.

In $\mathcal{ALC}$ the existence of uniform interpolation for a given $\mathcal{O}$ and Σ is 2ExpTime complete.

Uniform interpolants can be of triple exponential size in the worst case.

Work on computing uniform interpolants at this workshop, IJCAR 2014, and KR 2014.

Where do the α come from?

Let

$$\mathcal{O} = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\}$$

and $\Sigma = \{A, r\}$.

Where do the α come from?

Let

$$\mathcal{O} = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\}$$

and $\Sigma = \{A, r\}$.

Then

$$\mathcal{O}' = \{A \sqsubseteq \exists r.\top\}$$

is a uniform Σ-interpolant of $\mathcal{O}$ for $\mathcal{ALC}$ concept inclusions.

Where do the α come from?

Let

$$\mathcal{O} = \{A \sqsubseteq \exists r.B \sqcap \exists r.\neg B\}$$

and $\Sigma = \{A, r\}$.

Then

$$\mathcal{O}' = \{A \sqsubseteq \exists r.\top\}$$

is a uniform Σ-interpolant of $\mathcal{O}$ for $\mathcal{ALC}$ concept inclusions.

This is not a uniform Σ interpolant for FO (or certain DLs).

$\mathcal{EL}$ uniform interpolants are not always $\mathcal{ALC}$ uniform interpolants

$$\mathcal{O} = \{A \sqsubseteq \exists r.B, A_0 \sqsubseteq \exists r.(A_1 \sqcap B), E \equiv A_1 \sqcap B \sqcap \exists r.(A_2 \sqcap B)\}$$

is an acyclic $\mathcal{EL}$-TBox. So uniform interpolants for $\mathcal{EL}$ consequences always exist.

$\mathcal{EL}$ uniform interpolants are not always $\mathcal{ALC}$ uniform interpolants

$$\mathcal{O} = \{A \sqsubseteq \exists r.B, A_0 \sqsubseteq \exists r.(A_1 \sqcap B), E \equiv A_1 \sqcap B \sqcap \exists r.(A_2 \sqcap B)\}$$

is an acyclic $\mathcal{EL}$-TBox. So uniform interpolants for $\mathcal{EL}$ consequences always exist.

However, for $\Sigma = \{A, r, A_0, A_1, E\}$, there is no uniform Σ-interpolant for $\mathcal{ALC}$ consequences.

Other α

Important for ontology-based data access, where one uses queries $\boldsymbol{q}$ (e.g., conjunctive queries) to query data sets $\mathcal{D}$ taking into account ontology $\mathcal{O}$:

$$\mathcal{O} \cup \mathcal{D} \models \boldsymbol{q}$$

Other α

Important for ontology-based data access, where one uses queries q (e.g., conjunctive queries) to query data sets $\mathcal{D}$ taking into account ontology $\mathcal{O}$:

$$\mathcal{O} \cup \mathcal{D} \models q$$

Investigate existence and computation of $\mathcal{O}'$ such that

- $\mathcal{O} \models \mathcal{O}'$;
- $\mathbf{sig}(\mathcal{O}') \subseteq \Sigma$;
- if $\mathcal{O} \cup \mathcal{D} \models q$ and $\mathbf{sig}(D, q) \cap \mathbf{sig}(\mathcal{O}) \subseteq \Sigma$, then $\mathcal{O}' \cup \mathcal{D} \models q$.

For $\mathcal{EL}$ very similar to concept inclusions; for $\mathcal{ALC}$ no results yet.

Conclusion

- Many potential applications of interpolation in Description Logic.
- Many theoretical results: existence of interpolants, size of interpolants, complexity of computing interpolants.
- Implemented algorithms and evaluation needed.

Literature

Beth Definability and Interpolation:

- Balder ten Cate, Enrico Franconi, Inan Seylan: Beth Definability in Expressive Description Logics. J. Artif. Intell. Res. (JAIR) 48: 347-414 (2013)
- Balder ten Cate, Willem Conradie, Maarten Marx, Yde Venema: Definitorially Complete Description Logics. KR 2006

Parallel Interpolation:

- Boris Konev, Carsten Lutz, Denis Ponomaryov, Frank Wolter: Decomposing Description Logic Ontologies. KR 2010

Uniform Interpolation:

- Boris Konev, Dirk Walther, Frank Wolter: Forgetting and Uniform Interpolation in Large-Scale Description Logic Terminologies. IJCAI 2009
- Carsten Lutz, Frank Wolter: Foundations for Uniform Interpolation and Forgetting in Expressive Description Logics. IJCAI 2011

- Carsten Lutz, Inan Seylan, Frank Wolter: An Automata-Theoretic Approach to Uniform Interpolation and Approximation in the Description Logic EL. KR 2012
- Nadeschda Nikitina, Sebastian Rudolph: ExpExpExplosion: Uniform Interpolation in General EL Terminologies. ECAI 2012
- Patrick Koopmann, Renate A. Schmidt: Count and Forget: Uniform Interpolation of $\mathcal{SHQ}$ -Ontologies. IJCAR 2014
- M. Ludwig and B. Konev. Practical uniform interpolation and forgetting for ALC TBoxes with applications to logical difference. In Principles of Knowledge Representation and Reasoning. Proceedings, 2014.
- Kewen Wang, Zhe Wang, Rodney W. Topor, Jeff Z. Pan, Grigoris Antoniou: Eliminating Concepts and Roles from Ontologies in Expressive Descriptive Logics. Computational Intelligence 30(2): 205-232 (2014)