MANCHESTER 1824
The University of Manchester

Formal Methods Group

Resolution-Based Uniform Interpolation for Expressive Description Logics

Patrick Koopmann and Renate A. Schmidt

MANCHESTER 1824
The University of Manchester

Formal Methods Group

Uniform Interpolation

- Restrict ontology $\mathcal{O}$ to a set of given symbols $\mathcal{S}$
- Preserve entailments in $\mathcal{S}$
- *Forget* symbols outside $\mathcal{S}$

Input Ontology

$\mathbf{Margherita} \sqsubseteq \forall \mathbf{topping}.(\mathit{Tomato} \sqcup \mathit{Mozarella})$

$\mathbf{American} \sqsubseteq \exists \mathbf{topping}.\mathit{Tomato}$

$\mathbf{American} \sqsubseteq \exists \mathbf{topping}.\mathit{Mozarella}$

$\mathbf{American} \sqsubseteq \exists \mathbf{topping}.\mathit{Pepperoni}$

$\mathit{Tomato} \sqcup \mathit{Mozarella} \sqsubseteq \mathbf{VegTopping}$

$\mathit{Pepperoni} \sqsubseteq \mathbf{MeatTopping}$

Uniform Interpolant

$\mathbf{Margherita} \sqsubseteq \forall \mathbf{topping}.\mathbf{VegTopping}$

$\mathbf{American} \sqsubseteq \exists \mathbf{topping}.\mathbf{MeatTopping}$

Applications

- Exhibit hidden concept relations
- Compare ontology versions (logical difference)
- Information Hiding
- Obfuscation
- ...

Expressive Description Logics

Concepts $\mathcal{ALC}$

$$\bot \mid \top \mid A \mid \neg C \mid C \sqcup D \mid C \sqcap D \mid \exists r.C \mid \forall r.C$$

TBox Axioms $\mathcal{ALC}$

$$C \sqsubseteq D \mid C \equiv D$$

ABox Axioms $\mathcal{ALC}$

$$C(a) \mid r(a,b)$$

$\mathcal{ALCH}$:	Role Hierarchies	$r \sqsubseteq s$
$\mathcal{SH}$:	Transitive Roles	$trans(r)$
$\mathcal{SHQ}$:	Number Restrictions	$\geq nr.C, \leq nr.C$
$\mathcal{SHI}$:	Inverse Roles	r^{-1}
$\mathcal{L}\mu$:	Fixpoint Operators	$\mu X.C[X], \nu X.C[X]$

Uniform Interpolants

Definition

$\mathcal{O}^{\mathcal{S}}$ is an $\mathcal{L}$ *uniform interpolant* (*UI*) of $\mathcal{O}$ for $\mathcal{S}$ iff

1. $sig(\mathcal{O}^{\mathcal{S}}) \subseteq \mathcal{S}$
2. $\mathcal{O}^{\mathcal{S}} \models \alpha$ iff $\mathcal{O} \models \alpha$ where
   - $sig(\alpha) \subseteq \mathcal{S}$
   - α is expressible in $\mathcal{L}$

Existing Work

Description Logic	Publication
DL-Lite	Wang, Wang et. al 2008
$\mathcal{EL}$	Konev, Walther, et. al 2009 Nikitina 2011 Lutz, Seylan 2012
$\mathcal{ALC}$	Lutz, Wolter 2011 (Foundations) Wang, Wang et. al 2012 (Tableau-based) Ludwig, Konev 2013 (Resolution-based) Koopmann, Schmidt 2013 (Res.+Fixpoints)
$\mathcal{ALCH}$	Koopmann, Schmidt 2013 (Role Forgetting)
$\mathcal{SHQ}$	Koopmann, Schmidt 2014 (Res.+Fixpoints)

Challenges: Finiteness

UIs in input language not always finite

- Example $\mathcal{ALC}$:

  $A \sqsubseteq B, \quad B \sqsubseteq \exists r.B$

  $\mathcal{S} = \{A, r\}$

- Uniform Interpolant in $\mathcal{ALC}$:
  - $A \sqsubseteq \exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\ldots$

Challenges: Finiteness

UIs in input language not always finite

- Example $\mathcal{ALC}$:

  $A \sqsubseteq B, \quad B \sqsubseteq \exists r.B$

  $\mathcal{S} = \{A, r\}$

- Uniform Interpolant in $\mathcal{ALC}$:
  - $A \sqsubseteq \exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\ldots$

- Solutions:

Fixpoints:	$A \sqsubseteq \nu X.(\exists r.X)$
Approximate:	$A \sqsubseteq \exists r.\exists r.\exists r.\top$
Helper concepts:	$A \sqsubseteq \exists r.D, \quad D \sqsubseteq \exists r.D$

Challenges: Complexity

Known complexities $\mathcal{ALC}$:

- If finite, worst size $O(2^{2^{2^n}})$
  $\Rightarrow$ With fixpoints: $O(2^{2^n})$
- Deciding finiteness: $O(2^{2^n})$

Challenges: Reasoning

Using reasoning techniques for uniform interpolation

- Might derive not enough
  - Often optimised for specific problem
  - Represent as $\mathcal{L}$ ontology

- Might derive too much
  - Only entailments in $\mathcal{L}$ needed
  - Termination
  - Goal-oriented inferences necessary

$\Rightarrow$ Need for new calculi

Solutions

Representation:

- Reason on $\mathcal{L}$-statements
  - Only infer entailments in $\mathcal{L}$
  - $\Rightarrow$ Clauses = DL axioms
  - $\Rightarrow$ Cheap conversion of result

- Use finitely bounded representation
  - Ensure termination
  - Preserve all entailments
  - $\Rightarrow$ Structural transformation does the job

Normal form, $\mathcal{ALC}$

$\mathcal{ALC}$-Clause

$$\top \sqsubseteq L_1 \sqcup \ldots \sqcup L_n$$

L_i: $\mathcal{ALC}$-literal

$\mathcal{ALC}$-Literal

$$A \mid \neg A \mid \exists r.D \mid \forall r.D$$

A: any concept symbol, D: definer symbol

- Definer symbols: Special concept symbols, not part of signature
- Invariant: max 1 neg. definer symbol per clause
  $\Rightarrow$ $\neg D_1 \sqcup \exists r.D_2 \sqcup \neg B$, ~~$\neg D_1 \sqcup \neg D_2 \sqcup A$~~

Definer symbols

Invariant: max 1 neg. definer symbol per clause

- Allows easy translation to clausal form and back:

$$C_1 \sqcup \mathsf{Q}r.C_2 \iff C_1 \sqcup \mathsf{Q}r.D_1, \neg D_1 \sqcup C_2$$
$$C_1 \sqcup \nu X.C_2[X] \iff C_1 \sqcup \mathsf{Q}r.D_1, \neg D_1 \sqcup C_2[D]$$

$\Rightarrow$ Any set of clauses can be converted into an $\mathcal{ALC}\mu$-ontology ($\mathcal{ALC}$ with fixpoints)

- New definer symbols introduced by calculus
  - Number finitely bounded

Calculus

Resolution + *Combination rules*

- Resolution rule:
  - Direct inference on concept symbol to forget
  - Resolvent has to fulfil invariant

$$\frac{C_1 \sqcup A \quad C_2 \sqcup \neg A}{C_1 \sqcup C_2}$$

Calculus

Resolution + *Combination rules*

- Resolution rule:
  - Direct inference on concept symbol to forget
  - Resolvent has to fulfil invariant

$$\frac{C_1 \sqcup A \quad C_2 \sqcup \neg A}{C_1 \sqcup C_2}$$

- Combination rules:
  - Combine contexts of nested definer symbols
  - Introduce new definer symbols
    - Representing conjunctions of definers
    - Max. 2^n new definer symbols
  - Make further inferences possible

Combination Rules

$\neg D_1 \sqcup A$

$C_1 \sqcup \exists r.D_1$

$\neg D_2 \sqcup B \sqcup \neg A$

$C_2 \sqcup \forall r.D_2$

$\top \sqsubseteq C_1 \sqcup \exists r.A$ $\quad\quad$ $\top \sqsubseteq C_2 \sqcup \forall r.(B \sqcup \neg A)$

Combination Rules

Cannot resolve due invariant

$\neg D_1 \sqcup A$

$C_1 \sqcup \exists r.D_1$

$\neg D_2 \sqcup B \sqcup \neg A$

$C_2 \sqcup \forall r.D_2$

Combination Rules

Cannot resolve due invariant

$\neg D_1 \sqcup A$
$C_1 \sqcup \exists r.D_1$

$\neg D_2 \sqcup B \sqcup \neg A$
$C_2 \sqcup \forall r.D_2$

combine

$C_1 \sqcup C_2 \sqcup \exists r.D_{12}$
$\neg D_{12} \sqcup A$
$\neg D_{12} \sqcup B \sqcup \neg A$

Combination Rules

Cannot resolve due invariant

$\neg D_1 \sqcup A$
$C_1 \sqcup \exists r.D_1$

$\neg D_2 \sqcup B \sqcup \neg A$
$C_2 \sqcup \forall r.D_2$

combine

$C_1 \sqcup C_2 \sqcup \exists r.D_{12}$
$\neg D_{12} \sqcup A$
$\neg D_{12} \sqcup B \sqcup \neg A$

Resolves to $\neg D_{12} \sqcup B$

Combination Rules $\mathcal{ALC}$

∀∃-Combination

$$\frac{C_1 \sqcup \forall r.D_1 \qquad C_2 \sqcup \exists r.D_2}{C_1 \sqcup C_2 \sqcup \exists r.D_{12}}$$

∀∀-Combination

$$\frac{C_1 \sqcup \forall r.D_1 \qquad C_2 \sqcup \forall r.D_2}{C_1 \sqcup C_2 \sqcup \forall r.D_{12}}$$

MANCHESTER 1824
The University of Manchester

Formal Methods Group

Combination Rules $\mathcal{SHQ}$

$\leq\leq$-Combination:

$$\frac{C_1 \sqcup \leq n_1 r_1.\neg D_1 \quad C_2 \sqcup \leq n_2 r_2.\neg D_2 \quad r \sqsubseteq r_1 \quad r \sqsubseteq r_2}{C_1 \sqcup C_2 \sqcup \leq (n_1+n_2) r.\neg D_{12}}$$

$\geq\leq$-Combination:

$$\frac{C_1 \sqcup \geq n_1 r_1.(D_1 \sqcup \ldots \sqcup D_m) \quad C_2 \sqcup \leq n_2 r_2.\neg D_a \quad r_1 \sqsubseteq_{\mathcal{R}} r_2}{C_1 \sqcup C_2 \sqcup \geq (n_1-n_2) r_1.(D_{1a} \sqcup \ldots \sqcup D_{ma})}$$

$\leq\geq$-Combination:

$$\frac{C_1 \sqcup \leq n_1 r_1.\neg D_1 \quad C_2 \sqcup \geq n_2 r_2.D_2 \quad r_2 \sqsubseteq_{\mathcal{R}} r_1 \quad n_1 \geq n_2}{\begin{array}{c} C_1 \sqcup C_2 \sqcup \leq (n_1-n_2) r_1.\neg(D_1 \sqcup D_2) \sqcup \geq 1 r_1.D_{12} \\ \vdots \\ C_1 \sqcup C_2 \sqcup \leq (n_1-1) r_1.\neg(D_1 \sqcup D_2) \sqcup \geq n_2 r_1.D_{12} \end{array}}$$

$\geq\geq$-Combination:

$$\frac{C_1 \sqcup \geq n_1 r_1.\mathcal{D}_1 \quad C_2 \sqcup \geq n_2 r_2.\mathcal{D}_2 \quad r_1 \sqsubseteq_{\mathcal{R}} r \quad r_2 \sqsubseteq_{\mathcal{R}} r}{\begin{array}{c} C_1 \sqcup C_2 \sqcup \geq (n_1+n_2) r.(\mathcal{D}_1 \sqcup \mathcal{D}_2) \sqcup \geq 1 r.\mathcal{D}_{12} \\ \vdots \\ C_1 \sqcup C_2 \sqcup \geq (n_1+1) r.(\mathcal{D}_1 \sqcup \mathcal{D}_2) \sqcup \geq n_2 r.\mathcal{D}_{12} \end{array}}$$

Transitivity:

$$\frac{C \sqcup \leq 0 r_1.\neg D \quad \mathrm{trans}(r_2) \in \mathcal{R} \quad r_2 \sqsubseteq_{\mathcal{R}} r_1}{C \sqcup \leq 0 r_2.\neg D' \quad \neg D' \sqcup D \quad \neg D' \sqcup \leq 0 r_2.\neg D'}$$

Algorithm

INPUT: Ontology $\mathcal{O}$, signature $\mathcal{S}$
OUTPUT: Uniform interpolant of $\mathcal{O}$ for $\mathcal{S}$

1. Transform $\mathcal{O}$ to normal form **N**
2. For each $x \in sig(\mathcal{O}) \setminus \mathcal{S}$:
   2.1 Derive all inferences on x
   2.2 Remove clauses containing x
3. Transform **N** to ontology $\mathcal{O}^{\mathcal{S}}$
   (eliminate definer symbols)

Empirical Results $\mathcal{ALCH}$

$\lvert\mathcal{S}\rvert$	Timeouts	Fixpoints	Interpolant Size	Axiom Size	Average Duration
50	15.12%	6.99%	22.50%	799.37%	24.2 sec.
100	18.38%	11.57%	45.21%	646.32%	21.0 sec.
150	22.25%	13.58%	76.55%	837.66%	23.7 sec.
All	18.38%	10.44%	45.74%	757.69%	23.0 sec.

- 115 ontologies from NCBO BioPortal
  $\Rightarrow$ $\mathcal{ALCH}$-fragments
- Timeout: 1,000 seconds

Summary + Outlook

- Method implemented for:
  - $\mathcal{ALC}$-ontologies with ABoxes
  - $\mathcal{ALCH}$-TBoxes
  - $\mathcal{SHQ}$-TBoxes

- Future work:
  - ABox support for all approaches
  - $\mathcal{SHI}$ (inverse roles)
  - $\mathcal{SHIQ}$