Uniform Interpolation of $\mathcal{ALC}$-Ontologies Using Fixpoints

Patrick Koopmann and Renate A. Schmidt

MANCHESTER 1824
The University of Manchester

Uniform Interpolation

- Restrict TBox $\mathcal{T}$ to signature Σ
- Preserve logical entailments in Σ
- Dual notion: Forgetting

Input Ontology $\mathcal{T}$

$A \sqsubseteq \exists r.B$

$B \sqsubseteq C$

$\Rightarrow$

Uniform Interpolant over $\Sigma = \{A, C, r\}$

$A \sqsubseteq \exists r.C$

Applications

- Ontology Reuse
- Hide Confidential Concepts
- Obfuscate Ontologies
- Exhibit Hidden Relations
- Compute Logical Difference of Ontologies

The University of Manchester

Known Challenges

- Uniform Interpolants in $\mathcal{ALC}$
  - Not always finitely representable in $\mathcal{ALC}$
    - $\mathcal{T} = \{A \sqsubseteq B, B \sqsubseteq \exists r.B\}$, $\Sigma = \{A, r\}$
    - $\mathcal{T}^{\Sigma} = \{A \sqsubseteq \exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\ldots\}$
  - Worst-case size of result triple-exponential w.r.t input

The University of Manchester

Known Challenges

- Uniform Interpolants in $\mathcal{ALC}$
  - Not always finitely representable in $\mathcal{ALC}$
    - $\mathcal{T} = \{A \sqsubseteq B, B \sqsubseteq \exists r.B\}$, $\Sigma = \{A, r\}$
    - $\mathcal{T}^{\Sigma} = \{A \sqsubseteq \exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\exists r.\ldots\}$
  - Worst-case size of result triple-exponential w.r.t input

$\Rightarrow$ New method to meet these challenges

The University of Manchester

Our Approach

- Use fixpoint operators
  - Ensures finite representations
    - $\mathcal{T} = \{A \sqsubseteq B, B \sqsubseteq \exists r.B\}$, $\Sigma = \{A, r\}$
    - $\mathcal{T}^{\Sigma} = \{A \sqsubseteq \nu X.\exists r.X\}$
- Resolution-based approach
  - Allows for focused elimination of symbols
- First method using fixpoints
- Experiments show feasibility on a lot of real-life ontologies

The University of Manchester

Syntax

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

$$A \mid \neg C \mid C \sqcup D \mid C \sqcap D \mid \exists r.C \mid \forall r.C \mid$$
$$\mu X.C[X] \mid \nu X.C[X]$$

$\mathcal{ALC}\mu$-TBox statements

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

MANCHESTER 1824
The University of Manchester

Semantics

- $\mathcal{ALC}$-connectives and TBox statements are interpreted as usual.
- We only make use of greatest fixpoints ($\nu X.C$)

Fixpoint semantics

$$(\nu X.C)^{\mathcal{I},\mathcal{V}} := \bigcup\{W \subseteq \Delta^{\mathcal{I}} \mid W \subseteq C^{\mathcal{I},\mathcal{V}[X\mapsto W]}\}$$
$$(\mu X.C)^{\mathcal{I},\mathcal{V}} := \bigcap\{W \subseteq \Delta^{\mathcal{I}} \mid C^{\mathcal{I},\mathcal{V}[X\mapsto W]} \subseteq W\}$$

MANCHESTER 1824
The University of Manchester

Uniform Interpolation

Uniform Interpolants

Given TBox $\mathcal{T}$ and signature Σ, we have

1. $sig(\mathcal{T}^{\Sigma}) \subseteq \Sigma$
2. $\mathcal{T}^{\Sigma} \models C \sqsubseteq D$ iff $\mathcal{T} \models C \sqsubseteq D$, for $sig(C \sqsubseteq D) \subseteq \Sigma$

Input Ontology $\mathcal{T}$

$A \sqsubseteq \exists r.B$
$B \sqsubseteq C$

→

T^{Σ}, $\Sigma = \{A, C, r\}$

$A \sqsubseteq \exists r.C$

The University of Manchester

Uniform Interpolation

Uniform Interpolants

Given TBox $\mathcal{T}$ and signature Σ, we have

1. $sig(\mathcal{T}^{\Sigma}) \subseteq \Sigma$
2. $\mathcal{T}^{\Sigma} \models C \sqsubseteq D$ iff $\mathcal{T} \models C \sqsubseteq D$, for $sig(C \sqsubseteq D) \subseteq \Sigma$

- This work concentrates on eliminating *concept symbols*

Overview of the Method

1. Clausify input
2. For every $B \in sig(\mathcal{T}) \setminus \Sigma$:
   - Eliminate B using resolution based approach
     $\Rightarrow$ *Introduces new concept symbols*
3. For every introduced concept symbol D:
   - Eliminate D by applying Ackermann's Lemma
     $\Rightarrow$ *May introduce fixpoint operators*

MANCHESTER 1824
The University of Manchester

Overview of the Method

1. **Clausify input**
2. For every $B \in sig(\mathcal{T}) \setminus \Sigma$:
   - Eliminate B using resolution based approach
     $\Rightarrow$ *Introduces new concept symbols*
3. For every introduced concept symbol D:
   - Eliminate D by applying Ackermann's Lemma
     $\Rightarrow$ *May introduce fixpoint operators*

Clausification

$\mathcal{ALC}$-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

- Transformation using structural transformation
  - $C_1 \sqcup \exists r.C_2 \Longrightarrow C_1 \sqcup \exists r.D, \neg D \sqcup C_2$ $(D \sqsubseteq C_2)$
- $\neg D$ marks *context* of clause in role structure.
- Clauses are represented as sets

The University of Manchester

Overview of the Method

1. Clausify input
2. **For every** $B \in \mathbf{sig}(\mathcal{T}) \setminus \Sigma$**:**
   - **Eliminate** B **using resolution based approach**
     $\Rightarrow$ **Introduces new concept symbols**
3. For every introduced concept symbol D:
   - Eliminate D by applying Ackermann's Lemma
     $\Rightarrow$ *May introduce fixpoint operators*

The University of Manchester

Eliminating Concept Symbol

- Based on new calculus deciding $\mathcal{ALC}$-satisfiability
- Restrict rules to compute inferences on selected symbol B
  - $\Rightarrow$ Clauses containing B can safely be removed

Central Rules of the Calculus

Resolution

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

Role Propagation

$$\frac{C_1 \sqcup \forall r.D_1 \qquad C_2 \sqcup Qr.D_2}{C_1 \sqcup C_2 \sqcup Qr.D_3}$$

- $Q \in \{\forall, \exists\}$
- D_3 is a possibly new definer representing $D_1 \sqcap D_2$
- Side condition: $C_1 \sqcup C_2$ does not contain more than one negative definer literal
  - Ensure back-translatability
  - Function of role propagation: combine contexts to make resolution possible

Introduction of Definers

- New definer $D_3 \sqsubseteq D_1 \sqcap D_2$:
  - Check whether such definer already exists
  - Add $\neg D_3 \sqcup D_1$, $\neg D_3 \sqcup D_2$ otherwise
- Number of introduced definers can be limited by $O(2^n)$
- Limits number of derived clauses to $O(2^{2^n})$

The University of Manchester

Central Rules of the Calculus

Resolution

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

Role Propagation

$$\frac{C_1 \sqcup \forall r.D_1 \qquad C_2 \sqcup Qr.D_2}{C_1 \sqcup C_2 \sqcup Qr.D_3}$$

- $Q \in \{\forall, \exists\}$
- D_3 is a possibly new definer representing $D_1 \sqcap D_2$
- Side condition: $C_1 \sqcup C_2$ does not contain more than one negative definer literal
  - Ensure back-translatability
  - Function of role propagation: combine contexts to make resolution possible

Example

Example TBox $\mathcal{T}$

$$A \sqsubseteq \forall r.B \qquad C \sqsubseteq \exists r.(A \sqcup \neg B)$$

***clauses*($\mathcal{T}$)**

1. $\neg A \sqcup \forall r.D_2$
2. $\neg D_2 \sqcup B$
3. $\neg C \sqcup \exists r.D_3$
4. $\neg D_3 \sqcup A \sqcup \neg B$

MANCHESTER 1824
The University of Manchester

Example

$clauses(\mathcal{T})$

1. $\neg A \sqcup \forall \boldsymbol{r}.\boldsymbol{D_2}$	2. $\neg D_2 \sqcup B$
3. $\neg C \sqcup \exists \boldsymbol{r}.\boldsymbol{D_3}$	4. $\neg D_3 \sqcup A \sqcup \neg B$

5. $\neg A \sqcup \neg C \sqcup \exists \boldsymbol{r}.\boldsymbol{D_4}$	role propagation on 1, 3
6. $\boldsymbol{\neg D_4} \sqcup D_2$	$D_4 \sqsubseteq D_2$
7. $\boldsymbol{\neg D_4} \sqcup D_3$	$D_4 \sqsubseteq D_3$

MANCHESTER 1824
The University of Manchester

Example

$clauses(\mathcal{T})$

1. $\neg A \sqcup \forall r.D_2$	2. $\neg \mathbf{D_2} \sqcup B$
3. $\neg C \sqcup \exists r.D_3$	4. $\neg \mathbf{D_3} \sqcup A \sqcup \neg B$

5. $\neg A \sqcup \neg C \sqcup \exists r.D_4$	role propagation on 1, 3
6. $\neg D_4 \sqcup \mathbf{D_2}$	$D_4 \sqsubseteq D_2$
7. $\neg D_4 \sqcup \mathbf{D_3}$	$D_4 \sqsubseteq D_3$
8. $\neg D_4 \sqcup B$	resolution on 6, 2
9. $\neg D_4 \sqcup A \sqcup \neg B$	resolution on 7, 4

MANCHESTER 1824
The University of Manchester

Example

$clauses(\mathcal{T})$

1. $\neg A \sqcup \forall r.D_2$	2. $\neg D_2 \sqcup B$
3. $\neg C \sqcup \exists r.D_3$	4. $\neg D_3 \sqcup A \sqcup \neg B$

5. $\neg A \sqcup \neg C \sqcup \exists r.D_4$	role propagation on 1, 3
6. $\neg D_4 \sqcup D_2$	$D_4 \sqsubseteq D_2$
7. $\neg D_4 \sqcup D_3$	$D_4 \sqsubseteq D_3$
8. $\neg D_4 \sqcup \boldsymbol{B}$	resolution on 6, 2
9. $\neg D_4 \sqcup A \sqcup \boldsymbol{\neg B}$	resolution on 7, 4
10. $\boldsymbol{\neg D_4 \sqcup A}$	resoluton on 8, 9

MANCHESTER 1824
The University of Manchester

Rules of Calculus

Resolution

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

Role Propagation

$$\frac{C_1 \sqcup \forall r.D_1 \qquad C_2 \sqcup Qr.D_2}{C_1 \sqcup C_2 \sqcup Qr.D_3}$$

Existential Role Restriction Elimination

$$\frac{C \sqcup \exists R.D \qquad \neg D}{C}$$

Theorem: Rules form refutational sound and complete calculus deciding $\mathcal{ALC}$-TBox satisfiability

The Calculus

- Method for eliminating B:
  - Only resolve on definer symbols and B
  - Saturate
  - Remove clauses containing B
  - Remove clauses of form $\neg D_i \sqcup D_j$
- Resulting clause set preserves all consequences not using B
- Maximally $O(2^{2^n})$ clauses are derived

Overview of the Method

1. Clausify input
2. For every $B \in sig(\mathcal{T}) \setminus \Sigma$:
   - Eliminate B using resolution based approach
     $\Rightarrow$ *Introduces new concept symbols*
3. **For every introduced concept symbol D:**
   - **Eliminate D by applying Ackermann's Lemma**
     $\Rightarrow$ **May introduce fixpoint operators**

The University of Manchester

Elimination of Definer Symbols

Non-cyclic definer elimination

$$\frac{\mathcal{T} \cup \{D \sqsubseteq C\}}{\mathcal{T}^{[D \mapsto C]}} \quad \text{provided } D \notin sig(C)$$

Cyclic definer elimination

$$\frac{\mathcal{T} \cup \{D \sqsubseteq C[D]\}}{\mathcal{T}^{[D \mapsto \nu X.C[X]]}} \quad \text{provided } D \in sig(C)$$

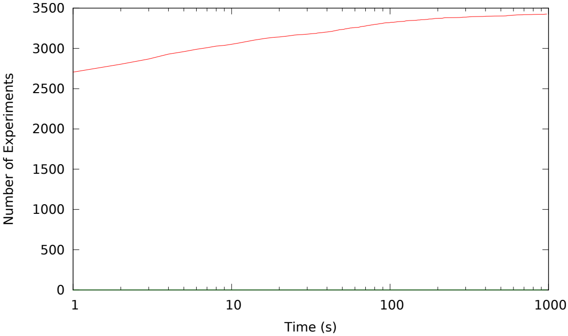
- Replace remaining definers by $\top$
- Apply simplifications

How Practical is The Method?

- Implemented using further optimisations
- Evaluated on $\mathcal{ALC}$-fragments of around 200 ontologies from the BioPortal repository
- Computed uniform interpolants over small signatures (5 – 150 symbols)
  - up to 187,514 thousand concept symbols
  - average: around 5,728

  $\Rightarrow$ Eliminate most concept symbols
- Results suggest that in most cases, computing uniform interpolants is feasible

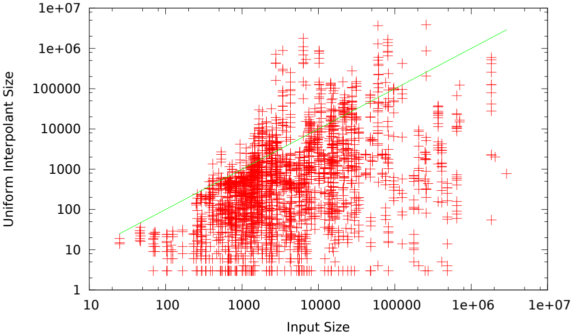
Experimental Results: Duration



- 3,739 runs were performed
- 8% of runs took longer than 1,000 second timeout

The University of Manchester

Experimental Results: Size



- 90.1% smaller than input
- Fixpoints in 20.1% of cases

MANCHESTER 1824
The University of Manchester

Conclusion

- Method to compute uniform interpolants of $\mathcal{ALC}$ TBoxes
- Use fixpoints to represent uniform interpolants finitely
- Combines resolution-based approach with rules based on Ackermann's Lemma
- Experiments suggest practicality in a lot of cases
- Future work
  - Minimal use of fixpoint operators
  - More expressive description logics

Further Information

For more details about the experiments and for the implementation check

http://www.cs.man.ac.uk/~koopmanp/womo_experiments