Lethe: Saturation-Based Reasoning for Non-Standard Reasoning Tasks

Patrick Koopmann, Renate A. Schmidt
LETHE

- “River of Forgetfulness”
- Usage from command line, as Java library, or via GUI
- Non-standard reasoning services relative to signatures
  - Forgetting / Uniform Interpolation
  - TBox Abduction
  - Logical Difference
- Support for expressive description logics (up to $SHQ$)
- Problems reduced to forgetting, uses saturation-based reasoning
Uniform Interpolation/Forgetting

- Core Functionality of LETHE
- Restrict vocabulary in set of axioms
- Preserve entailments over that signature

Input Ontology

Margherita ⊑ ∀topping.(Tomato □ Mozarella)
American ⊑ ∃topping.Tomato
American ⊑ ∃topping.Mozarella
American ⊑ ∃topping.Pepperoni
Tomato □ Mozarella ⊑ VegTopping
Pepperoni ⊑ MeatTopping

Uniform Interpolant

Margherita ⊑ ∀topping.VegTopping
American ⊑ ∃topping.MeatTopping
Applications of Forgetting

• Exhibit hidden concept relations
• Information hiding
• Ontology reuse
• Ontology summary
• Obfuscation
• …
TBox Abduction

- Given TBox $\mathcal{T}$, axioms $O$, find axioms $H$ with $\mathcal{T} \cup H \models O$
- “Complete” ontology such that given set of axioms is entailed
- Abducibles $\Sigma$ specify concepts and roles allowed in solution
- Reducible to uniform interpolation:
  - $\mathcal{T} \cup \neg O \models \neg H$
  - Express $\neg (C \sqsubseteq D)$ as $\exists r^* . (C \cap D)$
  - Interpolate to set of abducibles
- Optimisations for large TBoxes and small inputs
Logical Difference

- “Semantical Diff”
- Analyse ontology changes, compare ontologies
- Look for differing entailments in specified signature $\Sigma$
- Compute new entailments in $O_2$:
  - $LD(O_1, O_2, \Sigma) = \{ \alpha \mid \alpha \in O_2^{\Sigma}, O_1 \not\models \alpha \}$
  - $O_1^{\Sigma}$: Uniform interpolant of $O_1$ for $\Sigma$
- Optimised for two use cases:
  1. Bigger changes, computation in minutes acceptable
  2. Small changes, computation in seconds required
Challenges Uniform Interpolation

1. Need for new reasoning methods
2. Cyclic TBoxes
   \[ A \sqsubseteq B, \quad B \sqsubseteq \exists r. B \]
   \[ S = \{A, r\} \]
   - Uniform Interpolant in \( \mathcal{ALC} \):
     \[ A \sqsubseteq \exists r. \exists r. \exists r. \exists r. \exists r. \exists r. \exists r. \exists r. \exists r. \exists r. \exists r. \ldots \]
   - Solutions:
     Fixpoints: \[ A \sqsubseteq \nu X. (\exists r. X) \]
     Approximate: \[ A \sqsubseteq \exists r. \exists r. \exists r. \top \]
     Helper concepts: \[ A \sqsubseteq \exists r. D, \quad D \sqsubseteq \exists r. D \]
3. High Complexity
   - \( \mathcal{ALC} \) with fixpoints: \( 2^n \), where \( n \) is size of input
   - Goal-oriented approach necessary
Normal form, $\text{ALC}$

$\text{ALC}$-Clause

$\top \subseteq L_1 \cup \ldots \cup L_n$

$L_i$: $\text{ALC}$-literal

$\text{ALC}$-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

$\Rightarrow \neg D_1 \cup \exists r.D_2 \cup \neg B, \quad \neg D_1 \cup \neg D_2 \sqcup A$
Definer symbols

Invariant: max 1 neg. definer symbol per clause

- Allows easy translation to clausal form and back:

\[ C_1 \sqcup Qr.C_2 \iff C_1 \sqcup Qr.D_1, \neg D_1 \sqcup C_2 \]
\[ C_1 \sqcup \nu X.C_2[X] \iff C_1 \sqcup Qr.D_1, \neg D_1 \sqcup C_2[D] \]

\[ \Rightarrow \] 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 obey invariant

\[
C_1 \sqcap A \quad C_2 \sqcap \neg A \\
\hline
C_1 \sqcap C_2
\]

- **Combination rules:**
  - Combine context of nested definer symbols
  - Introduce new definer symbols
    - Representing conjunctions of definers
    - Max. \(2^n\) new definer symbols
  - Make further inferences possible
Combination Rules

\[ \neg D_1 \sqcup A \]
\[ C_1 \sqcup \exists r. D_1 \]

\[ \neg D_2 \sqcup B \sqcup \neg A \]
\[ C_2 \sqcup \forall r. D_2 \]
Combination Rules

Cannot resolve due invariant

\[ \lnot D_1 \sqcup A \]
\[ C_1 \sqcup \exists r. D_1 \]

\[ \lnot D_2 \sqcup B \sqcup \lnot A \]
\[ C_2 \sqcup \forall r. D_2 \]
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

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

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

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

$C_1 \sqcup C_2 \sqcup \exists r.D_{12}$
$\neg D_{12} \sqcup A$
$\neg D_{12} \sqcup B \sqcup \neg A$
Combination Rules $\mathcal{ALC}$

$\forall\exists$-Combination

\[
\frac{C_1 \sqcup \forall r. D_1 \quad C_2 \sqcup \exists r. D_2}{C_1 \sqcup C_2 \sqcup \exists r. D_{12}}
\]

$\forall\forall$-Combination

\[
\frac{C_1 \sqcup \forall r. D_1 \quad C_2 \sqcup \forall r. D_2}{C_1 \sqcup C_2 \sqcup \forall r. D_{12}}
\]
Combination Rules $SHQ$

$\le\le$-Combination:

\[
C_1 \le n_1 r_1, \neg D_1 \\
C_2 \le n_2 r_2, \neg D_2 \\
r \subseteq r_1 \\
C_1 \cup C_2 \le (n_1 + n_2) r \neg D_{12}
\]

$\ge\ge$-Combination:

\[
C_1 \ge n_1 r_1, (D_1 \cup \ldots \cup D_m) \\
C_2 \le n_2 r_2, \neg D_a \\
r_1 \subseteq r_2 \\
C_1 \cup C_2 \ge (n_1 - n_2) r_1 (D_{1a} \cup \ldots \cup D_{ma})
\]

$\le\ge$-Combination:

\[
C_1 \le n_1 r_1, \neg D_1 \\
C_2 \ge n_2 r_2, D_2 \\
r_2 \subseteq r_1 \\
C_1 \cup C_2 \le (n_1 - n_2) r_1 \neg D_{12} \neg D
\]

$\ge\le$-Combination:

\[
C_1 \ge n_1 r_1, D_1 \\
C_2 \le n_2 r_2, D_2 \\
r_1 \subseteq r_2 \\
C_1 \cup C_2 \ge (n_1 + n_2) r_1 (D_1 \cup D_2) \cup (D_{1a} \cup \ldots \cup D_{ma})
\]

Transitivity:

\[
C \le 0 r_1, \neg D \\
\text{trans}(r_2) \in R \\
r_2 \subseteq r_1 \\
C \le 0 r_2, \neg D' \\
\neg D' \cup D \\
\neg D' \cup \le 0 r_2, \neg D'
\]
Algorithm

- Compute all inferences on symbol to forget
- Use resolvents breaking invariant to choose combination rules
- Filter out all occurrences of symbol to forget
- Eliminate introduced symbols
Evaluation of Uniform Interpolation

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Respective fragments of 306 ontologies from BioPortal having at most 100,000 axioms.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeout</td>
<td>30 minutes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mathcal{ALCH}$, forget 50 symbols</th>
<th>$\mathcal{ALCH}$, forget 100 symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success Rate:</td>
<td>Success Rate:</td>
</tr>
<tr>
<td>Without Fixpoints:</td>
<td>Without Fixpoints:</td>
</tr>
<tr>
<td>Duration Mean:</td>
<td>Duration Mean:</td>
</tr>
<tr>
<td>Duration Median:</td>
<td>Duration Median:</td>
</tr>
<tr>
<td>Duration 90th percentile:</td>
<td>Duration 90th percentile:</td>
</tr>
<tr>
<td>91.10%</td>
<td>95.29%</td>
</tr>
<tr>
<td>95.10%</td>
<td>93.27%</td>
</tr>
<tr>
<td>7.68 sec.</td>
<td>18.03 sec.</td>
</tr>
<tr>
<td>2.74 sec.</td>
<td>3.81 sec.</td>
</tr>
<tr>
<td>12.45 sec.</td>
<td>21.17 sec.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mathcal{ALC}$ w. ABoxes, forget 50 symbols</th>
<th>$\mathcal{ALC}$ w. ABoxes, forget 100 symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success Rate:</td>
<td>Success Rate:</td>
</tr>
<tr>
<td>Without Fixpoints:</td>
<td>Without Fixpoints:</td>
</tr>
<tr>
<td>Duration Mean:</td>
<td>Duration Mean:</td>
</tr>
<tr>
<td>Duration Median:</td>
<td>Duration Median:</td>
</tr>
<tr>
<td>Duration 90th percentile:</td>
<td>Duration 90th percentile:</td>
</tr>
<tr>
<td>94.79%</td>
<td>92.91%</td>
</tr>
<tr>
<td>92.91%</td>
<td>92.48%</td>
</tr>
<tr>
<td>23.94 sec.</td>
<td>57.87 sec.</td>
</tr>
<tr>
<td>3.01 sec.</td>
<td>6.43 sec.</td>
</tr>
<tr>
<td>29.00 sec.</td>
<td>99.26 sec.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\mathcal{SHQ}$, forget 50 concept symbols</th>
<th>$\mathcal{SHQ}$, forget 100 concept symbols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success Rate:</td>
<td>Success Rate:</td>
</tr>
<tr>
<td>Without Fixpoints:</td>
<td>Without Fixpoints:</td>
</tr>
<tr>
<td>Duration Mean:</td>
<td>Duration Mean:</td>
</tr>
<tr>
<td>Duration Median:</td>
<td>Duration Median:</td>
</tr>
<tr>
<td>Duration 90th percentile:</td>
<td>Duration 90th percentile:</td>
</tr>
<tr>
<td>95.83%</td>
<td>93.40%</td>
</tr>
<tr>
<td>93.40%</td>
<td>91.99%</td>
</tr>
<tr>
<td>7.62 sec.</td>
<td>13.51 sec.</td>
</tr>
<tr>
<td>1.04 sec.</td>
<td>1.60 sec.</td>
</tr>
<tr>
<td>4.89 sec.</td>
<td>11.65 sec.</td>
</tr>
</tbody>
</table>

Timeout 30 minutes
Conclusion

- **LETHE** supports different non-classical reasoning methods via reduction to forgetting
- Usage as library, command line tool or via simple front end
- Available at [http://cs.man.ac.uk/~koopmanp/lethe](http://cs.man.ac.uk/~koopmanp/lethe)

- Future work
  - Better evaluation on abduction and logical difference
  - Use saturation-based approach for other non-classical reasoning problems such as approximation and ABox abduction
  - Investigate more expressive description logics