An Introduction to Description Logic VIII

Inherent intractability of terminological reasoning

Marco Cerami

Palacký University in Olomouc
Department of Computer Science
Olomouc, Czech Republic

Olomouc, December 4^{th} 2014

european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
INVESTMENTS IN EDUCATION DEVELOPMENT

Introduction

Introduction

- **Structural subsumption algorithms** can be used to reason with **acyclic TBoxes**.
- In the 80's **it was supposed** that it could keep the good computational performance as with respect to empty TBoxes.
- In 1990, Nebel indeed proves that reasoning with acyclic TBoxes in a language even simpler than $\mathcal{FL}^-$ is **co-**NP**-complete**.
- We are mainly following the 1990 paper **Terminological Reasoning is Inherently Intractable**, by B. Nebel.
- The hardness proof is obtained by means of a reduction of a problem from **automata theory**.

The language $\mathcal{TL}$

The language $\mathcal{TL}$

- The name $\mathcal{TL}$ stands for **terminological language**.
- It is a **fragment of** $\mathcal{FL}^{-}$ obtained by omitting existential quantifications.
- Below we define the language $\mathcal{TL}$:

$$
\begin{array}{lcll}
C, D & \longrightarrow & A & \text{atomic concept} \\
 & & C \sqcap D & \text{conjunction} \\
 & & \forall R.C & \text{value restriction}
\end{array}
$$

Reasoning in $\mathcal{TL}$

- In $\mathcal{TL}$ concepts and axioms are **trivially satisfiable**, exactly like in $\mathcal{FL}^-$.
- **Concept subsumption w.r.t. the empty KB** is clearly polynomial, exactly like in $\mathcal{FL}^-$.
- Concept equivalence and subsumtion can be **linearly reduced** to each other in the following way:

$$C \equiv D \quad \text{iff} \quad C \sqsubseteq D \text{ and } D \sqsubseteq C$$

$$C \sqsubseteq D \quad \text{iff} \quad C \sqcap D \equiv C$$

- We will consider the problem of subsumption with respect to **acyclic KBs**.

Acyclic TBoxes

An Acyclic TBox is a **definitional** TBox **without cycles**, that is:

- a TBox is said to be **definitional** when there appear at most one inclusion axiom of the form:

  $$A \equiv C$$

  for each atomic concept A;

- a TBox is said to be **cyclic** or a set of **General Concept Inclusions** (GCIs), when there is a sequence of inclusion axioms $C_1 \sqsubseteq D_1, \ldots, C_n \sqsubseteq D_n$ and a set of concepts $A_1, \ldots, A_{n-1}$, such that, for every $1 < m < n$, A_m appears both in D_{m-1} and in C_m and A_1 appears both in D_n and in C_1.

Reasoning with non-empty TBoxes

- If we consider the **empty TBox**, concept

  $$\forall \texttt{hasParent}.\texttt{Human}$$

  is **not subsumed** by concept

  $$\forall \texttt{hasParent}.\texttt{Mammal}.$$

- If we consider the knowledge base $\mathcal{K} = (\mathcal{T})$, where:

  $\mathcal{T} = \{ \quad \texttt{Human} \sqsubseteq \texttt{Mammal} \quad \}$

  we have that concept $\forall \texttt{hasParent}.\texttt{Human}$ **is subsumed** by concept $\forall \texttt{hasParent}.\texttt{Mammal}$ **in every model of** $\mathcal{K}$.

Expanded terminologies

Primitive and defined concepts

Let $\mathcal{T}$ be an **acyclic TBox**, then:

- An atomic concept A is said to be **defined** if axiom

  $$A \sqsubseteq C$$

  appears in $\mathcal{T}$, for some C.

- An atomic concept is said to be **primitive** if it is not defined.

- If concept name A is defined by axiom $A \sqsubseteq C \in \mathcal{T}$, hence

  $$T(A) = C$$

  is said to be the **definition** of A.

Expanded concepts and terminologies

- The definition function can be extended to complex concepts, in order to obtain **expanded concepts**:

$$\begin{aligned} \hat{T}(A) &= T(A) \\ \hat{T}(C \sqcap D) &= \hat{T}(C) \sqcap \hat{T}(D) \\ \hat{T}(\forall R.C) &= \forall R.\hat{T}(C) \end{aligned}$$

- If the TBox is **acyclic**, then the expansion process is **finite**, that is, there is a natural n such that:

$$\hat{T}^n = \hat{T}^{n+1}$$

- In this case we call n the **depth** of $\mathcal{T}$ and $\hat{T}^n = \tilde{T}$, the **completely expanded terminology**.

Example

Consider the knowledge base $\mathcal{K} = (\mathcal{T})$, where:

$$\begin{aligned}\mathcal{T} = \{ \; & \texttt{Human} \equiv \texttt{Mammal} \sqcap \texttt{Biped}, \\ & \texttt{Man} \equiv \texttt{Human} \sqcap \texttt{Male}, \\ & \texttt{Son} \equiv \texttt{Human} \sqcap \texttt{Male} \sqcap \forall \texttt{hasParents}.\texttt{Human} \; \}\end{aligned}$$

Then:

$$\begin{aligned}\tilde{\mathcal{T}} = \{ \; & \texttt{Human} \equiv \texttt{Mammal} \sqcap \texttt{Biped}, \\ & \texttt{Man} \equiv \texttt{Mammal} \sqcap \texttt{Biped} \sqcap \texttt{Male}, \\ & \texttt{Son} \equiv \texttt{Mammal} \sqcap \texttt{Biped} \sqcap \texttt{Male} \sqcap \\ & \qquad \forall \texttt{hasParents}.(\texttt{Mammal} \sqcap \texttt{Biped}) \; \}\end{aligned}$$

Reducing acyclic to empty TBoxes

- Is is easy to show by **induction on the depth** of $\mathcal{T}$, that, for every concept C and every model $\mathcal{I}$ of $\mathcal{T}$ it holds:

$$C^{\mathcal{I}} = (\tilde{T}(C))^{\mathcal{I}}.$$

- Moreover, the following statements are **equivalent**:
  - concept C is subsumed by concept D **w.r.t. terminology** $\mathcal{T}$,
  - concept $\tilde{T}(C)$ is subsumed by concept $\tilde{T}(D)$ **w.r.t. the empty terminology**.

- Hence, we can define algorithm $TSUBS?[D, C, \mathcal{T}]$ from algorithm $SUBS?[D, C]$:

$$TSUBS?[D, C, \mathcal{T}] \quad = \quad SUBS?[\tilde{T}(D), \tilde{T}(C)]$$

The expansion is not polynomial

Let $n \in \mathbb{N}$ and consider the following terminology $\mathcal{T}_n$:

$$
\begin{array}{rcl}
T_n(C_0) & = & C_0 \\
T_n(C_1) & = & \forall R.C_0 \sqcap \forall P.C_0 \\
T_n(C_2) & = & \forall R.C_1 \sqcap \forall P.C_1 \\
 & \vdots & \\
T_n(C_n) & = & \forall R.C_{n-1} \sqcap \forall P.C_{n-1}
\end{array}
$$

Hence, even though the size of $\mathcal{T}_n$ **grows linearly** in n, the size of $\tilde{T}_n(C_n)$ **grows exponentially** in n.

co-NP-hardness

Unfolded terminologies

- We define the **unfolding function** U which returns sets of concepts from concepts:

$$
\begin{array}{rcl}
U(A) & = & \{A\} \\
U(C \sqcap D) & = & U(C) \cup U(D) \\
U(\forall R.C) & = & \{\forall R.D \mid D \in U(C)\}
\end{array}
$$

- The **completely unfolded form** $U_{\mathcal{T}}$ of a terminology $\mathcal{T}$ is defined by:

$$U_{\mathcal{T}} = U \circ \tilde{\mathcal{T}}$$

Example

Consider the knowledge base $\mathcal{K} = (\mathcal{T})$, where:

$$\mathcal{T} = \{ \; \texttt{Human} \equiv \texttt{Mammal} \sqcap \texttt{Biped}, \\ \texttt{Man} \equiv \texttt{Human} \sqcap \texttt{Male}, \\ \texttt{Son} \equiv \texttt{Human} \sqcap \texttt{Male} \sqcap \forall \texttt{hasParents.Human} \; \}$$

Then:

$$U_{\mathcal{T}} = \{ \; \{\texttt{Human}\} = \{\texttt{Mammal,Biped}\}, \\ \{\texttt{Man}\} = \{\texttt{Mammal,Biped,Male}\}, \\ \{\texttt{Son}\} = \{\texttt{Mammal,Biped,Male}, \forall \texttt{hasParents.Mammal}, \\ \forall \texttt{hasParents.Biped}\} \; \}$$

Properties of the unfolded form

- Is is easy to show that, for every concept C and every model $\mathcal{I}$ of $\mathcal{T}$ it holds:

$$C^{\mathcal{I}} = (U_{\mathcal{T}}(C))^{\mathcal{I}}.$$

- Every concept $D \in U_{\mathcal{T}}(C)$ is a **linear description**, that is a concept of the form:

$$\forall R_1.\forall R_2 \ldots \forall R_n.A$$

  where A is **primitive** in $\mathcal{T}$.

- Moreover, the following statements are **equivalent**:
  - concept C is equivalent to concept D **w.r.t. terminology** $\mathcal{T}$,
  - $U_{\mathcal{T}}(C) = U_{\mathcal{T}}(D)$.

Nondeterministic finite state automata

A **nondeterministic finite state automaton** (NDFA) is a tuple:

$$\mathcal{A} = (\Sigma, \mathcal{Q}, \delta, q_0, \mathcal{F})$$

where:

- Σ is a set of symbols or **alphabet**,
- $\mathcal{Q}$ is a set of **states**,
- $\delta \colon \Sigma \times \mathcal{Q} \longrightarrow 2^{\mathcal{Q}}$ is a **transition function**,
- $q_0 \in \mathcal{Q}$ is the **initial state**,
- $\mathcal{F} \subseteq \mathcal{Q}$ is a set of **accepting states**.

Properties of NDFAs

- A state $q' \in \mathcal{Q}$ is **reachable** from a state q by word $w = s_1 \dots s_n$ iff there exists a sequence of states $q_1, \dots, q_{n+1}$ with:
  - $q = q_1$,
  - $q' = q_{n+1}$,
  - $q_{i+1} \in \delta(q_i, s_i)$, for $1 \leqslant i \leqslant n$.
- The set $\mathcal{L}(\mathcal{A})$ of words w such that some final state is reachable from q_0 is called the **language accepted by** $\mathcal{A}$.
- Two automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are **equivalent** iff $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$.
- A state $q \in \mathcal{Q}$ is **redundant** either if it can not be reached from the initial state or if it can not reach any final state.

Acyclic nonredundant NDFAs

- An automaton is called **nonredundant** if ti does not contain any redundant state.
- A NDFA is called **acyclic** if no state is reachable from itself.
- Let $\mathcal{L}(\mathcal{A})$ be the language of a nonredundant NDFA $\mathcal{A}$. Then the following are equivalent:
  - $\mathcal{A}$ is **acyclic**,
  - $\mathcal{L}(\mathcal{A})$ is **finite**.
- If $\mathcal{L}(\mathcal{A}_1)$ and $\mathcal{L}(\mathcal{A}_2)$ are finite, then **determining equivalence** between $\mathcal{A}_1$ and $\mathcal{A}_2$ is co-NP-complete.
- In what follows we will consider **acyclic nonredundant NDFAs** (ANDFAs).

Reduction of ANDFAs

Let $\mathcal{A}_1 = (\Sigma, \mathcal{Q}_1, \delta_1, q_{0_1}, \mathcal{F}_1)$ and $\mathcal{A}_2 = (\Sigma, \mathcal{Q}_2, \delta_2, q_{0_2}, \mathcal{F}_2)$ be two **acyclic nonredundant NDFAs** with $\mathcal{Q}_1 \cap \mathcal{Q}_2 = \emptyset$. Then we can construct a **terminology** $\mathcal{T}_\mathcal{A}$ in the following way:

- $N_R = \Sigma$,
- $N_C = \mathcal{Q}_1 \cup \mathcal{Q}_2 \cup F$,
- $\mathcal{T}_\mathcal{A}(F) = F$ is the only primitive concept,
- $\mathcal{T}_\mathcal{A}(q) = \begin{cases} F, & \text{if } q \in \mathcal{F}_1 \cup \mathcal{F}_2 \\ \bigsqcap\{\forall s.q' : q' \in \delta_i(q,s), i = 1,2\} & \text{otherwise} \end{cases}$

Since $\mathcal{A}_1$ and $\mathcal{A}_2$ are **nonredundant**, then the concepts are **well-defined**. Since $\mathcal{A}_1$ and $\mathcal{A}_2$ are **acyclic**, then so in $\mathcal{T}_\mathcal{A}$ and it is possible to define $U_{\mathcal{T}_\mathcal{A}}$.

Example

Let $\Sigma = \{s, r, t\}$ and $q \in \mathcal{Q}$ be such that:

- $\delta(q, s) = \{q_1, q_2\}$,
- $\delta(q, r) = \{q_2, q_3, q_4\}$,
- $\delta(q, t) = \{q_5, q_6\}$,

Then

$$T_{\mathcal{A}}(q) = \forall s.q_1 \sqcap \forall s.q_2 \sqcap \forall r.q_2 \sqcap \forall r.q_3 \sqcap \forall r.q_4 \sqcap \forall t.q_5 \sqcap \forall t.q_6$$

Where the states q_1, q_2, q_3, q_4, q_5 and q_6,, if are not final states, **can be further expanded** according to the function δ, but **in the same way** as for $\hat{T}$.

Completeness of the reduction

Let $w = s_1 \dots s_n \in \Sigma^*$ and $i = 1, 2$, then:

$$w \in \mathcal{L}(\mathcal{A}_i) \qquad \text{iff} \qquad \forall s_1 \dots \forall s_n.F \in U_{T_{\mathcal{A}_i}}(q_{0_i})$$

($\Rightarrow$) Assume that w is a word **accepted** by $\mathcal{A}_i$. Then there is a sequence of states $q_1, \dots, q_{n+1}$ such that

- $q_1 = q_{0_i}$,
- $q_n \in \mathcal{F}_i$,
- $q_{j+1} \in \delta(q_j, s_j)$, for $1 \leqslant j \leqslant n$.

By the way in which $T_{\mathcal{A}}$ is constructed and the definition of $U_{T_{\mathcal{A}_i}}$, it is possible to prove by induction on the length of w, that

$$\forall s_1 \dots \forall s_n.F \in U_{T_{\mathcal{A}_i}}(q_{0_i}).$$

($\Leftarrow$) Conversely, if $\forall s_1 \dots \forall s_n.F \in U_{T_{\mathcal{A}_i}}(q_{0_i})$, then by the way $T_{\mathcal{A}}$ is constructed, we have that a state $q \in \mathcal{F}_i$ is reachable from q_{0_i} by w in $\mathcal{A}_i$. That is, $w \in \mathcal{L}(\mathcal{A}_i)$.

Conclusions

- Hence

$$\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2) \qquad \text{iff} \qquad q_{0_1} \equiv_{\mathcal{T}_\mathcal{A}} q_{0_2}.$$

- Therefore, **concept equivalence** and **subsumption** in $\mathcal{TL}$ are co-NP-hard problems.

- The fact that these problem are also **in co-**NP is easy to prove: think on an algorithm guessing a linear description of concept C and checks whether it belongs to the completely unfolded form of concept D.