Approximate Unification in Description Logics

Pavlos Marantidis

Technische Universität Dresden

pavlos.marantidis@tu-dresden.de

March 30, 2016
Overview

1. Description Logics
2. Solving Linear Equations
3. Approximate Solutions
4. Banach’s Fixed Point Theorem
5. Linear Programming
6. Synopsis
Description Logics are a family of knowledge representation languages with a formal, logic-based semantics.

**Description** comes from concept description, i.e. a formal expression that determines a set of individuals with common properties.

**Logics** comes from the fact that the semantics of concept descriptions can be defined using logic.
Pavlos gives a boring talk
Pavlos gives a boring talk
Let’s try again:

Pavlos gives Talk

Boring
Given sets $\mathcal{C} = \{A, B, \ldots\}$ (concept names) and $\mathcal{R} = \{r, s, \ldots\}$ (role names), define complex concepts by using constructors:

- $C := A$ | concept names
- $\top, \bot$ | top, bottom concepts
- $C \sqcap D$ | conjunction
- $C \sqcup D$ | disjunction
- $\exists r.C$ | existential restriction
- $\forall r.C$ | value restriction

Plenty other constructors:
number restrictions, role inversion and composition, nominal concepts...
Using model theory!

Interpretation

Consider a non-empty set $\Delta$ and a function $\mathcal{I}$ such that:

- Concepts are mapped to sets: $A^\mathcal{I} \subseteq \Delta$
- Roles are mapped to binary relations: $r^\mathcal{I} \subseteq \Delta \times \Delta$

Furthermore, the interpretation maps:

- $\top$ to the domain: $\top^\mathcal{I} = \Delta$
- $\bot$ to the empty set: $\bot^\mathcal{I} = \emptyset$
- Conjunction to intersection: $(C \cap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$
- Disjunction to union: $(C \cup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I}$
- $(\exists r. C)^\mathcal{I} = \{d \in \Delta \mid (d, c) \in r^\mathcal{I} \land c \in C^\mathcal{I} \text{ for some } c \in \Delta\}$
- $(\forall r. C)^\mathcal{I} = \{d \in \Delta \mid (d, c) \in r^\mathcal{I} \rightarrow c \in C^\mathcal{I}\}$
Subsumption and Equivalence

Subsumption and equivalence

- $C \subseteq D$ iff $C^I \subseteq D^I$ for all interpretations
- $C \equiv D$ iff $C \subseteq D$ and $D \subseteq C$

Examples

$C = \text{Human} \sqcap \forall \text{hasChild}.\text{Smart} \sqcap \forall \text{hasChild}.\text{Strong}$

$D = \text{Human} \sqcap \forall \text{hasChild}.(\text{Smart} \sqcap \text{Strong})$

It can be deduced that

- $C \subseteq \text{Human}$
- $C \equiv D$
Idea

some concepts may be defined differently by different users or developers of a knowledge base

\[
\text{Woman} \sqcap \forall \text{hasChild}. \text{Woman} \\
\text{Person} \sqcap \text{Female} \sqcap \forall \text{hasChild}.(\text{Person} \sqcap \text{Female}) \\
\text{Human} \sqcap \text{Male} \sqcap \exists \text{loves}. \text{ExtremeSports} \\
\text{Man} \sqcap \exists \text{loves}.(\text{Sport} \sqcap \text{Dangerous})
\]
Consider a partition $\mathcal{C} = \mathcal{C}_c \cup \mathcal{C}_v$. A substitution $\sigma$ is a mapping from concept variables $\mathcal{C}_v$ to concept terms containing only constant concept names.

**Unification Problem**

Given concepts $C, D$, find a unifier $\sigma$, i.e. a substitution $\sigma$ such that

$$\sigma(C) \equiv \sigma(D)$$
Given sets $C = C_C \cup C_Y$ and $R$
the set of $\mathcal{FL}_0$ concept terms only uses the constructors:

$C := A$ \hspace{1cm} \text{concept names}

$\top$ \hspace{1cm} \text{top concept}

$C \sqcap D$ \hspace{1cm} \text{conjunction}

$\forall r. C$ \hspace{1cm} \text{value restriction}

$Woman \sqcap \forall \text{hasChild}. Woman$

$Speaker \sqcap \forall \text{gives}. (Talk \sqcap \text{Boring})$
Reduction to linear equations

**Notation**

\[
\forall r. (C \cap D) \rightarrow \forall r. C \cap \forall r. D \\
\forall r_1 \ldots \forall r_m. A \rightarrow \forall r_1 \ldots r_m. A = \forall w. A \\
\forall w_1. A \cap \ldots \cap \forall w_n. A \rightarrow \forall L. A \\
\forall \varepsilon. A = A \quad \forall \emptyset. A = T
\]

**Normal Form**

Given \( C_C = \{A_1, \ldots, A_k\}, C_V = \{X_1, \ldots, X_n\}, R \), if \( C \) is an \( \mathcal{FL}_0 \) concept term, it can be rewritten as

\[
C \equiv \forall S_1. A_1 \cap \ldots \cap \forall S_k. A_k \cap \forall T_1. X_1 \cap \ldots \cap \forall T_n. X_n
\]

for finite sets of words \( S_1, \ldots, S_k, T_1, \ldots, T_n \).

Example!
Lemma (Baader, Narendran, 2001)

Let $C = \{A_1, \ldots, A_k\}$ and consider the $\mathcal{FL}_0$ concept terms $C, D$ in normal form:

$$C \equiv \forall U_1 . A_1 \cap \cdots \cap \forall U_k . A_k$$
$$D \equiv \forall V_1 . A_1 \cap \cdots \cap \forall V_k . A_k$$

Then $C \equiv D$ iff $U_i = V_i$ for all $i$.

Theorem (Baader, Narendran, 2001)

Let $C, D$ be $\mathcal{FL}_0$ concept terms:

$$C \equiv \forall S_{0,1} . A_1 \cap \cdots \cap \forall S_{0,k} . A_k \cap \forall S_{1} . X_1 \cap \cdots \cap \forall S_{n} . X_n$$
$$D \equiv \forall T_{0,1} . A_1 \cap \cdots \cap \forall T_{0,k} . A_k \cap \forall T_{1} . X_1 \cap \cdots \cap \forall T_{n} . X_n$$

Then, $C, D$ are unifiable iff

$$S_{0,i} \cup S_{1}X_{1,i} \cup \cdots \cup S_{n}X_{n,i} = T_{0,i} \cup T_{1}X_{1,i} \cup \cdots \cup T_{n}X_{n,i}$$

has a solution for every $i$. 
Proof.

\[ C \equiv \forall S_{0,1}.A_1 \cap \ldots \cap \forall S_{0,k}.A_k \cap \forall S_1.X_1 \cap \ldots \cap \forall S_n.X_n \]

Suppose that \( \sigma(X_i) = \forall L_{i,1}.A_1 \cap \ldots \cap \forall L_{i,k}.A_k \). Then

\[
\sigma(C) = \forall S_{0,1}.A_1 \cap \ldots \cap \forall S_{0,k}.A_k \cap \\
\forall S_1.(\forall L_{1,1}.A_1 \cap \ldots \cap \forall L_{1,k}.A_k) \cap \ldots \cap \\
\forall S_n.(\forall L_{n,1}.A_1 \cap \ldots \cap \forall L_{n,k}.A_k) \\
= \forall S_{0,1}.A_1 \cap \ldots \cap \forall S_{0,k}.A_k \cap \\
\forall S_1 L_{1,1}.A_1 \cap \ldots \cap \forall S_1 L_{1,k}.A_k \cap \ldots \cap \\
\forall S_n L_{n,1}.A_1 \cap \ldots \cap \forall S_n L_{n,k}.A_k \\
= \forall (S_{0,1} \cup S_1 L_{1,1} \cup \ldots \cup S_n L_{n,1}).A_1 \cap \ldots \cap \\
\forall (S_{0,k} \cup S_1 L_{1,k} \cup \ldots \cup S_n L_{n,k}).A_k
\]
Thus, Unification Problem reduces to Solving Language Equations.

Example
see blackboard!
Language equations

Alphabet $\Sigma$, variables $X_1, \ldots, X_n$

\[ K_0^{(1)} \cup K_1^{(1)} X_1 \cup \ldots \cup K_n^{(1)} X_n = L_0^{(1)} \cup L_1^{(1)} X_1 \cup \ldots \cup L_n^{(1)} X_n \]

\[ \vdots \]

\[ K_0^{(m)} \cup K_1^{(m)} X_1 \cup \ldots \cup K_n^{(m)} X_n = L_0^{(m)} \cup L_1^{(m)} X_1 \cup \ldots \cup L_n^{(m)} X_n \]

where $K_i^{(\ell)}$, $L_i^{(\ell)}$ are finite languages over $\Sigma$. 
## Language equations

Alphabet $\Sigma$, variables $X_1, \ldots, X_n$

\[
\phi_1(X_1, \ldots, X_n) = \psi_1(X_1, \ldots, X_n) \\
\vdots \\
\phi_m(X_1, \ldots, X_n) = \psi_m(X_1, \ldots, X_n)
\]

for expressions $\psi_i, \xi_i$ over $X_1, \ldots X_n$.

- every variable $X_i$ is an expression
- every regular language $L$ is an expression
- if $\phi$ is an expression, $L\phi$ is also an expression
- if $\phi, \psi$ are expressions, $\phi \cap \psi, \phi \cup \psi, \sim \phi$ are also expressions
Many to one

Normal Form

Transform all equations into a single one of the form

$$\phi(Z_1, \ldots, Z_k) = \emptyset$$

where constant regular languages occurring in $\phi$ are singletons from $\Sigma \cup \{\varepsilon\}$

- regular expressions $\rightarrow \Sigma \cup \{\varepsilon\}$
- $\phi = \psi \rightarrow (\phi \cap \sim \psi) \cup (\psi \cap \sim \phi) = \emptyset$
- $\phi = \emptyset, \psi = \emptyset \rightarrow \phi \cup \psi = \emptyset$

Every transformation leaves the set of solutions unchanged!

Example!
A Nondeterministic Finite Automaton (NFA) is a tuple

\[ A = (Q, \Sigma, I, \delta, F) \]

where
- \( Q \) a set of states
- \( \Sigma \) alphabet
- \( I \subseteq Q \) set of initial states
- \( F \subseteq Q \) set of final states
- \( \delta : Q \times \Sigma \rightarrow 2^Q \) transition relation

Example!
Given an alphabet $\Sigma = \{a_1, \ldots, a_k\}$, consider the (k-ary, unlabeled) infinite tree $T$, representing $\Sigma^*$, where every branch corresponds to a different letter of $\Sigma$. 

\[
\begin{align*}
\varepsilon &\quad \varepsilon \\
 &\quad a \\
 &\quad b \\
 &\qquad a \quad a \\
 &\qquad b \quad b \\
 &\qquad a \quad a \\
 &\qquad a \quad a \\
 &\qquad b \quad b \\
 &\qquad b \quad b \\
 &\quad \vdots
\end{align*}
\]
Definition of ILTA

**Definition**

A looping tree automaton with independent transitions (ILTA) induced by an NFA $A = (Q, \Sigma, I, \delta, F)$ working on a ($k$-ary, unlabeled) tree is a tuple

$$A = (Q, I, \Delta)$$

where $\Delta: Q \rightarrow 2^{Q^k}$ is the transition relation, defined as

$$\Delta(q): = \{(q_1, \ldots, q_k) \mid q_i \in \delta(q, a_i)\}$$

A run of $A$ on the tree $T$

$$r: \Sigma^* \rightarrow Q$$

is a labeling of every node with a state.

A run is called successful, if $r(\varepsilon) \in I$.

**Example!**
Using ILTA to solve Equations

Remove “bad” states.
Check whether there is a successful run.

Complexity results

Solving equations of the above form is ExpTime-complete.

In ExpTime, because of the construction.
ExpTime-hard, by reduction from the intersection emptiness problem for deterministic top-down automata.
Language equations

Alphabet $\Sigma$, variables $X_1, \ldots, X_n$

\[
\begin{align*}
\phi_1(X_1, \ldots, X_n) &= \psi_1(X_1, \ldots, X_n) \\
\vdots \\
\phi_m(X_1, \ldots, X_n) &= \psi_m(X_1, \ldots, X_n)
\end{align*}
\]

for expressions $\psi_i, \xi_i$ over $X_1, \ldots, X_n$. 
Approximations

Suppose there is no solution. What is the “best” we can do?

- Define best
  An approximate solution that is least “bad” wrt some measure

- Define what is an approximate solution
  An approximate solution is an assignment $\sigma: \{X_1, \ldots, X_n\} \rightarrow 2^{\Sigma^*}$ that maps a constant language $L_i$ to every variable $X_i$.

- Define a measure

- Find an algorithm that finds the best solution
Defining a measure

**Idea**

Assign a weight to every word of $\Sigma^*$ and sum the weights of all violating words.

Given a language $L \subseteq \Sigma^*$ define

$$\mu(L) = \frac{1}{2} \sum_{w \in L} (2|\Sigma|)^{-|w|}$$

Has some good properties

- $0 \leq \mu(L) \leq 1$
- $\mu(\emptyset) = 0$
- $\mu(\Sigma^*) = 1$
- $\mu(\bigcup_{i \in I} A_i) = \sum_{i \in I} \mu(A)$
**Lemma**

Given a language $L \subseteq \Sigma^*$, it holds that

$$
\mu(L) = \frac{1}{2} \chi_L(\epsilon) + \frac{1}{2|\Sigma|} \sum_{a \in \Sigma} \mu(a^{-1}L)
$$

It induces automata!

**Example**

$L = \{ \epsilon, a, aba, bbb \}$  
$L_1 = a^{-1}L = \{ \epsilon, ba \}$, $L_2 = b^{-1}L = \{ bb \}$

$$
\mu(L) = \frac{1}{2} \cdot 1 + \frac{1}{2 \cdot 2} (\mu(a^{-1}L) + \mu(b^{-1}L))
$$

$$
= \frac{1}{2} + \frac{1}{4} (\mu(L_1) + \mu(L_2)) = \cdots = \frac{41}{64}
$$
Consider the NFA

\[ A = (Q, \Sigma, I, \delta, F) \]

defined as before, where \( F = \{ q \in Q \mid \phi \in q \} \).

For the best run(s) we have the system of equations

\[
\mu(q) = \frac{1}{2} \chi_F(q) + \frac{1}{2|\Sigma|} \sum_{a \in \Sigma} \min_{p \in \delta(q,a)} \mu(p)
\]
Theorem (Banach’s Fixed Point Theorem)

Let \((X, d)\) be a complete metric space and a function \(f : X \rightarrow X\) be a contraction. Then \(f\) has a unique fixed point i.e. there is a unique \(p \in X\) such that \(f(p) = p\).

Furthermore, the sequence \(x_0, f(x_0), f(f(x_0)), \ldots\) converges to the fixed point, for every \(x_0 \in X\).
Definitions

Definition

A metric space \((X, d)\) is called complete, if every Cauchy sequence converges to a point in \(X\).

Definition

A sequence \((a_n)\) is called a Cauchy sequence, if for every \(\epsilon > 0\), there exists an \(n_0 \in \mathbb{N}\), s.t. for every \(m, n \geq n_0\) it holds that \(d(a_n, a_m) < \epsilon\).

Definition

A function \(f : (X, d) \to (Y, d')\) is called a contraction if there is a \(\lambda \in (0, 1)\) such that \(d'(f(x), f(y)) \leq \lambda d(x, y)\) for any \(x, y \in X\).
Applications

- Picard’s Theorem
- Solving Systems of Linear Equations
- Google’s PageRank Algorithm
- and many more...
Our construction

Define $f_i: [0, 1]^n \to [0, 1]$ for every $i$

$$f_i(x_1, \ldots, x_n) = \frac{1}{2} \chi_F(q_i) + \frac{1}{2|\Sigma|} \sum_{a \in \Sigma} \min_{q_j \in \delta(q_i, a)} x_j$$

Then $f: [0, 1]^n \to [0, 1]^n$

$$f(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n))$$

is a contraction and $[0, 1]^n$ complete.

By BFPT, there exists a unique fixed point, i.e. a tuple $(x_1, \ldots, x_n)$ such that

$$x_i = \frac{1}{2} \chi_F(q_i) + \frac{1}{2|\Sigma|} \sum_{a \in \Sigma} \min_{q_j \in \delta(q_i, a)} x_j$$
Linear Programming Problem (LPP)

objective: \( \min/\max \ z = c_1 x_1 + \ldots + c_n x_n \)

restrictions: \( a_{1,1} x_1 + \ldots + a_{1,n} x_n \leq b_1 \)

\[ \vdots \]

\( a_{m,1} x_1 + \ldots + a_{m,n} x_n \geq b_m \)
Define a new variable $x_{q,a}$, for every $q \in Q$, $a \in \Sigma$.

Intuitively, $x_{q,a} = \min_{p \in \delta(q,a)} \mu(p)$.

Then define:

$$\mu(q) = \frac{1}{2} \chi_F(q) + \frac{1}{2|\Sigma|} \sum_{a \in \Sigma} x_{q,a}$$

and $x_{q,a} \leq \mu(p)$ for all $a \in \Sigma$, for all $p \in \delta(q,a)$.

$$z = \max \sum_{q \in Q} \sum_{a \in \Sigma} x_{q,a}$$

Making use of BFPT, the feasible region is not empty, and the optimal solution corresponds to the fixed point of $f$. 
Finding the optimal assignment

Transform the ILTA, to always make the most efficient choices.

- Set as initial state $q_0$ the state in $I$ with minimum measure.
- For each $q \in Q$ and $a \in \Sigma$

$$\delta(q, a) := \{q'\}$$

where $q'$ has the minimum measure from all states in $\delta(q, a)$. 
Synopsis

- Description Logics
- Unification
- Reduction to Language Equations
- Reduction to Automata
- Approximate case
- Define a measure
- Reduction to automata
- Reduction to system of equations
- BFPT + LP

Mathematics is not that useless!
Thank you!