$\mathcal{ALC}^u_{\mathbb{P}}$: An Integration of Description Logic and General Rules

Jing Mei, Zuoquan Lin, Harold Boley

The First International Conference on Web Reasoning and Rule Systems
Innsbruck, Austria
June 7, 2007

Contents

1. Introduction
   - Hybrid Knowledge Bases
   - Homogeneous Knowledge Bases
   - Related Work
2. $\mathcal{ALC}^u_\mathbb{P}$
   - Syntax
   - Semantics
   - Algorithm
3. Conclusion

Contents

1. Introduction
    - Hybrid Knowledge Bases
    - Homogeneous Knowledge Bases
    - Related Work
2. $\mathcal{ALC}_{\mathbb{P}}^{u}$
    - Syntax
    - Semantics
    - Algorithm
3. Conclusion

Terminology

- Terms: Function-free
  - Constants $\mathcal{C}$
    - Named Individuals: $\mathbf{I} \subseteq \mathcal{C}$
    - Unnamed Individuals
  - Variables $\mathcal{V}$
- Predicates: $\mathcal{N} = \mathcal{N}_{\mathcal{T}} \cup \mathcal{N}_P$
  - $\mathcal{N}_{\mathcal{T}}$: Unary DL concepts and binary DL roles
  - $\mathcal{N}_P$: Arbitrary arity relationships
  - Hybrid: $\mathcal{N}_{\mathcal{T}} \cap \mathcal{N}_P = \emptyset$
  - Homogeneous: $\mathcal{N}_{\mathcal{T}} \subseteq \mathcal{N}_P$
- Atoms: α
  - Unary DL atom: $C(t)$ where $C \in \mathcal{N}_{\mathcal{T}}$ and $t \in \mathbf{I} \cup \mathcal{V}$
  - Binary DL atom: $R(t_1, t_2)$ where $R \in \mathcal{N}_{\mathcal{T}}$ and $t_1, t_2 \in \mathbf{I} \cup \mathcal{V}$
  - n-ary atom: $p(t_1, \cdots, t_n)$ where $p \in \mathcal{N}_P$ and $t_1, \cdots, t_n \in \mathbf{I} \cup \mathcal{V}$
- Literals: α or $\neg\alpha$ where α is an atom

Hybrid KBs

Definition

A hybrid knowledge base is $\mathcal{K} = (\Sigma, \Pi)$, where

- Σ is a DL knowledge base with predicates in $\mathcal{N}_\mathcal{T}$
- Π is a set of rules having the form of

$$h(\vec{u}) \leftarrow b_1(\vec{v}_1), \cdots, b_l(\vec{v}_l) \;\&\; q_1(\vec{w}_1), \cdots, q_n(\vec{w}_n)$$
$$\texttt{not}\ b_{l+1}(\vec{v}_{l+1}), \cdots, \texttt{not}\ b_m(\vec{v}_m)$$

  - each $b_i(\vec{v}_i)$ is a Datalog$^\neg$ atom with $b_i \in \mathcal{N}_P$ and $\vec{v}_i$ is a term sequence of arbitrary arity, for $1 \leqslant i \leqslant l \leqslant m$
  - each $q_j(\vec{w}_j)$ is a DL atom with $q_j \in \mathcal{N}_\mathcal{T}$ and $\vec{w}_j$ is a unary/binary term sequence, for $1 \leqslant j \leqslant n$
  - Uni-directional Information Flow: $h \in \mathcal{N}_P$
  - Bi-directional Information Flow: $h \in \mathcal{N}_P \cup \mathcal{N}_\mathcal{T}$

Homogeneous KBs

Predicates: $\mathbf{C} \cup \mathbf{R} = \mathcal{N}_{\mathcal{T}} \subseteq \mathcal{N}_P$

- **C**: Unary predicates for DL concepts
- **R**: Binary predicates for DL roles

Definition

A homogeneous knowledge base is $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$, where

- TBox $\mathcal{T}$: $C_1 \sqsubseteq C_2$ with $C_1, C_2 \in \mathbf{C}$
- ABox $\mathcal{A}$: $C(a)$ or $R(a, b)$ with $C \in \mathbf{C}$, $R \in \mathbf{R}$, and $a, b \in \mathbf{I}$
- PBox $\mathbb{P}$: $p(\vec{u}) \leftarrow q_1(\vec{v_1}), \cdots, q_m(\vec{v_m}),$
  $\texttt{not}\ q_{m+1}(\vec{v_{m+1}}), \cdots \texttt{not}\ q_n(\vec{v_n})$

  with $p, q_i \in \mathcal{N}_P$, and $\vec{u}, \vec{v_i}$ are vectors of terms in $\mathbf{I} \cup \mathcal{V}$, for each $1 \leqslant i \leqslant m \leqslant n$

Related Work

- Hybrid Approaches
  - AL-log [1]: A combination of $\mathcal{ALC}$ DL and Datalog
  - CARIN [2]: A combination of $\mathcal{ALCNR}$ DL and Datalog
  - DL+log [3]: A combination of OWL DL and Datalog$^{\vee,\neg}$
- Homogeneous Approaches
  - DLP [4]: An intersection of DL and LP
  - SWRL [5]: A undecidable integration of OWL DL and Datalog
  - DL-safe rules [6]: Reducing OWL DL KBs to Datalog$^{\vee}$ programs, followed by DL-safe Datalog$^{\vee}$ rules

Contents

1 Introduction
- Hybrid Knowledge Bases
- Homogeneous Knowledge Bases
- Related Work

2 $\mathcal{ALC}^u_{\mathbb{P}}$
- Syntax
- Semantics
- Algorithm

3 Conclusion

Preliminaries: $\mathcal{ALC}^u_{\mathbb{P}}$

- $\mathcal{ALC}$: The basic and simple DL language
  - The set **R** of $\mathcal{ALC}$-roles is the set of role names
  - The set **C** of $\mathcal{ALC}$-concepts is the smallest set such that
    - The top $\top$ and the bottom $\bot$ are $\mathcal{ALC}$-concepts
    - Every concept name is an $\mathcal{ALC}$-concept
    - If C, C_1, C_2 are $\mathcal{ALC}$-concepts and R is an $\mathcal{ALC}$-role, then $\neg C, C_1 \sqcap C_2, C_1 \sqcup C_2, \exists R.C, \forall R.C$ are also $\mathcal{ALC}$-concepts
- u: Unique Names Assumption
- $\mathbb{P}$: Extending DL TBox $\mathcal{T}$ and ABox $\mathcal{A}$ with PBox $\mathbb{P}$

Knowledge Bases

Definition

An $\mathcal{ALC}^u_{\mathbb{P}}$ KB has the form $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$, where

- TBox $\mathcal{T}$: $C_1 \sqsubseteq C_2$ with $C_1, C_2 \in \mathbf{C}$
- ABox $\mathcal{A}$: $C(a)$ or $R(a, b)$ with $C \in \mathbf{C}$, $R \in \mathbf{R}$, and $a, b \in \mathbf{I}$
- PBox $\mathbb{P}$: $p(\vec{u}) \leftarrow q_1(\vec{v_1}), \cdots, q_m(\vec{v_m}),$
  $\texttt{not}\ q_{m+1}(\vec{v_{m+1}}), \cdots \texttt{not}\ q_n(\vec{v_n})$
  with $p, q_i \in \mathbf{C} \cup \mathbf{R}$, and $\vec{u}, \vec{v_i}$ are vectors of terms in $\mathbf{I} \cup \mathcal{V}$, for each $1 \leqslant i \leqslant m \leqslant n$

DL Structure

Given a function-free first-order language $\mathcal{L}$, an $\mathcal{L}$-structure is $\mathcal{I} = \langle U, I \rangle$

- The universe $U = (\mathcal{D}, \sigma)$
  - A non-empty domain $\mathcal{D}$
  - A function $\sigma : \mathbf{I} \cup \mathcal{D}' \to \mathcal{D}$
    - $\mathbf{I} \cap \mathcal{D}' = \emptyset$
    - $\sigma(d) = d$ for all $d \in \mathcal{D}'$, viz. unnamed individuals
- The interpretation I over $\mathcal{D}$ which assigns a relation $p^I \subseteq \mathcal{D}^n$ to each n-ary predicate symbol p (here $n \geq 1$)

Standard names assumption (SNA)

- Parameter names assumption (PNA): $\mathcal{D}' = \emptyset$
- Unique names assumption (UNA): σ is injective

Being a fragment of function-free first-order logic, DL structures interpret concepts (here $n = 1$) and roles (here $n = 2$)

(Extended) Herbrand Structure

A Herbrand structure is a pair $\mathcal{H} = \langle H, I \rangle$

- The Herbrand universe $H = (\mathbf{I}, id)$
  - A fixed domain $\mathbf{I}$
  - A function $id : \mathbf{I} \to \mathbf{I}$
    - $id(d) = d$ for all $d \in \mathbf{I}$
- The interpretation I over $\mathbf{I}$
- Both PNA ($\mathcal{D}' = \emptyset$) and UNA (id is an identical mapping in $\mathbf{I}$)

An extended Herbrand structure is a pair $e\mathcal{H} = \langle eH, I \rangle$

- The extended Herbrand universe $eH = (\mathcal{D}, id)$
  - A non-empty domain $\mathcal{D}$
  - A function $id : \mathbf{I} \cup \mathcal{D}' \to \mathcal{D}$
    - $\mathbf{I} \cap \mathcal{D}' = \emptyset$
    - $id(d) = d$ for all $d \in \mathbf{I} \cup \mathcal{D}'$
- The interpretation I over $\mathcal{D}$
- Not PNA ($\mathcal{D}' \neq \emptyset$) but UNA ($id$ is an identical mapping in $\mathcal{D}$)

$\mathcal{ALC}^u_{\mathbb{P}}$ Structure

Definition

An extended Herbrand structure $\mathcal{I} = \langle(\mathcal{D}, id), I\rangle$ is defined for a set of named individuals $\mathbf{I}$, a set of concepts $\mathbf{C}$ and a set of roles $\mathbf{R}$, where

- $id : \mathbf{I} \cup \mathcal{D}' \rightarrow \mathcal{D}$ and $id(d) = d$ for all $d \in \mathbf{I} \cup \mathcal{D}'$, given $\mathbf{I} \cap \mathcal{D}' = \emptyset$
- $I : \mathbf{C} \rightarrow 2^{\mathcal{D}}$ for concepts and $I : \mathbf{R} \rightarrow 2^{\mathcal{D}\times\mathcal{D}}$ for roles

  such that for $C, C_1, C_2 \in \mathbf{C}$ and $R \in \mathbf{R}$, the following are satisfied:

  $\top^I = \mathcal{D} \quad \bot^I = \emptyset$

  $(\neg C)^I = \mathcal{D}\backslash C^I \quad (C_1 \sqcap C_2)^I = C_1^I \cap C_2^I \quad (C_1 \sqcup C_2)^I = C_1^I \cup C_2^I$

  $(\exists R.C)^I = \{e_1 \in \mathcal{D} | \exists e_2.(e_1, e_2) \in R^I \text{ and } e_2 \in C^I\}$

  $(\forall R.C)^I = \{e_1 \in \mathcal{D} | \forall e_2.(e_1, e_2) \in R^I \text{ implies } e_2 \in C^I\}$

An associated valuation v_I of an interpretation I over $\mathcal{D}$ is a mapping s.t.
$v_I(C(d)) = true$, if $d \in C^I$, where $C \in \mathbf{C}$ and $d \in \mathcal{D}$
$v_I(R(d_1, d_2)) = true$, if $(d_1, d_2) \in R^I$, where $R \in \mathbf{R}$ and $d_1, d_2 \in \mathcal{D}$

$\mathcal{ALC}^u_{\mathbb{P}}$ Model

- An extended Herbrand structure $\mathcal{I}$ satisfies an $\mathcal{ALC}^u_{\mathbb{P}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$ if $\mathcal{I}$ is a model of $\mathcal{T}, \mathcal{A}$ and $\mathbb{P}$
  Such a structure $\mathcal{I}$ is called a model of $\mathcal{K}$, written as $\mathcal{I} \models \mathcal{K}$
- An extended Herbrand structure $\mathcal{I}$ satisfies a TBox $\mathcal{T}$ if
  - $C_1^I \subseteq C_2^I$ for all $C_1 \sqsubseteq C_2$ in $\mathcal{T}$, where $C_1, C_2 \in \mathbf{C}$

  Such a structure $\mathcal{I}$ is called a model of $\mathcal{T}$, written as $\mathcal{I} \models \mathcal{T}$
- An extended Herbrand structure $\mathcal{I}$ satisfies an ABox $\mathcal{A}$ if
  - $id(a) = a \in C^I$ for all $C(a)$ in $\mathcal{A}$, where $C \in \mathbf{C}$ and $a \in \mathbf{I}$
  - $(id(a_1), id(a_2)) = (a_1, a_2) \in R^I$ for all $R(a_1, a_2)$ in $\mathcal{A}$, where $R \in \mathbf{R}$ and $a_1, a_2 \in \mathbf{I}$

  Such a structure $\mathcal{I}$ is called a model of $\mathcal{A}$, written as $\mathcal{I} \models \mathcal{A}$
- An extended Herbrand structure $\mathcal{I}$ satisfies a PBox $\mathbb{P}$ if
  - $S = \{C(d) | v_I(C(d)) = true\} \cup \{R(d_1, d_2) | v_I(R(d_1, d_2)) = true\}$ is an open answer set of $\mathbb{P}_g$

  Such a structure $\mathcal{I}$ is called a model of $\mathbb{P}$, written as $\mathcal{I} \models \mathbb{P}$

Open Answer Set

- The grounding $\mathbb{P}_g$ of $\mathbb{P}$ w.r.t. an extended Herbrand universe $eH = (\mathcal{D}, id)$ is the set of all rules obtained as follows
  - Keep each named individual $a \in \mathbf{I}$ unchanged as $id(a) = a \in \mathcal{D}$
  - Replace each variable $v \in \mathcal{V}$ appearing in $r \in \mathbb{P}$ with $d \in \mathcal{D}$
  - Replace each variable $v \in \mathcal{V}$ appearing in the head of $r \in \mathbb{P}$ with a certain $d \in \mathbf{I}$ (Semantic Weak Safeness)
- The grounded PBox $\mathbb{P}_g$ of rules without `not`
  - The extended Herbrand model of $\mathbb{P}_g$ is a set S such that
    - For any rule $r : p(\vec{u}) \leftarrow q_1(\vec{v_1}), \cdots, q_m(\vec{v_m})$ in $\mathbb{P}_g$
    - If $q_i(\vec{v_i}) \in S$ for all $1 \leqslant i \leqslant m$, then $p(\vec{u}) \in S$
  - $\lambda(\mathbb{P}_g)$: The least extended Herbrand model of $\mathbb{P}_g$
- The grounded PBox $\mathbb{P}_g$ of general rules with `not`
  - Given a set S, via the Gelfond-Lifschitz transformation, $\Gamma(\mathbb{P}_g, S)$ is the set of rules obtained from $\mathbb{P}_g$ by deleting
    1. Each rule that has a "`not` $q(\vec{v})$" in the body with $q(\vec{v}) \in S$
    2. All "`not` $q(\vec{v})$" occurrences in bodies of the remaining rules
  - $S = \lambda(\Gamma(\mathbb{P}_g, S))$: The open answer set of $\mathbb{P}_g$

Algorithm

In the following, if not stated otherwise, for an $\mathcal{ALC}^u_{\mathbb{P}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$, we denote that

- Σ_C is the closure of concepts occurring in $\mathcal{T}$, $\mathcal{A}$ and $\mathbb{P}$
- Σ_R is the set of roles occurring in $\mathcal{T}$, $\mathcal{A}$ and $\mathbb{P}$
- Σ_I is the set of named individuals occurring in $\mathcal{A}$ and $\mathbb{P}$

The algorithm starts from preprocessing, followed by building completion graphs, and finally, we conclude that

- A decision procedure for the KB satisfiability problem
- A decision procedure for the query entailment problem

Preprocessing

- From $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$ to $\mathcal{K}' = (\emptyset, \mathcal{A}, \mathbb{P}^{\mathcal{T}})$
  - Rewriting concept subsumptions in the TBox $\mathcal{T}$
  - $\mathbb{P}^{\mathcal{T}} = \mathbb{P} \cup \{C_2(x) \leftarrow C_1(x) | C_1 \sqsubseteq C_2 \in \mathcal{T}\}$
- From $\mathcal{K}' = (\emptyset, \mathcal{A}, \mathbb{P}^{\mathcal{T}})$ to $\mathcal{K}'' = (\emptyset, \mathcal{A}, \mathbb{P}^{\mathcal{T},E})$
  - Computing concept expressions in Σ_C

$$C_1 \sqcap C_2(x) \leftarrow C_1(x), C_2(x). \qquad \exists R.C(x) \leftarrow R(x,y), C(y).$$
$$C_1 \sqcup C_2(x) \leftarrow C_1(x). \qquad C \sqcup \neg C(x) \leftarrow \top(x).$$
$$C_1 \sqcup C_2(x) \leftarrow C_2(x). \qquad \forall R.C \sqcup \exists R.\neg C(x) \leftarrow \top(x).$$

- Proposition: The $\mathcal{ALC}^u_{\mathbb{P}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$ and its updated version $\mathcal{K}' = (\emptyset, \mathcal{A}, \mathbb{P}^{\mathcal{T}})$ also $\mathcal{K}'' = (\emptyset, \mathcal{A}, \mathbb{P}^{\mathcal{T},E})$ are equivalent

Completion Graphs

A completion graph is a (directed) graph G where

- Each node u is labeled with a set $\mathcal{L}(u) \subseteq \Sigma_C$
- Each edge $\langle u, v \rangle$ is labeled with a set $\mathcal{L}(\langle u, v \rangle) \subseteq \Sigma_R$
- v is the successor of u, if there is an edge $\langle u, v \rangle$ in G
- u is the predecessor of v, if there is an edge $\langle u, v \rangle$ in G
- Ancestor: The transitive closure of predecessor
- Descendant: The transitive closure of successor
- Clash: $\mathcal{L}(u)$ is said to contain a clash, if $\{C, \neg C\} \subseteq \mathcal{L}(u)$, for some concept $C \in \Sigma_C$

Initialization

Initially, our algorithm constructs a graph $G_{\mathcal{A}}$ for an ABox $\mathcal{A}$

- Creating a node u_a, for each named individual $a \in \Sigma_I$
- Creating an edge $\langle u_a, u_b \rangle$, if $R(a, b) \in \mathcal{A}$ for some role $R \in \Sigma_R$ and $a, b \in \Sigma_I$
- The labels of these nodes and edges are $\mathcal{L}(u_a) = \{C | C(a) \in \mathcal{A}\}$ and $\mathcal{L}(\langle u_a, u_b \rangle) = \{R | R(a, b) \in \mathcal{A}\}$

CompGraph

Then, our algorithm invokes CompGraph($G_{\mathcal{A}}$)

Procedure CompGraph
Input: A graph G_{in}
Output: A set of graphs G_{out}
begin $G_{out} := \emptyset$
  for each g in ExpGraph(G_{in}) **do**
    for each g' in SmsGraph(g) **do**
      if g' is *complete* **then** $G_{out} := G_{out} \cup \{g'\}$
        else $G_{out} := G_{out} \cup$ CompGraph(g')
  return G_{out}
end

A completion graph G is called *complete* when

- For some node u in G, $\mathcal{L}(u)$ contains a clash, or
- None of the expansion principles is applicable

ExpGraph

Expansion Principles for ExpGraph

$\sqcap$:	if	$C_1 \sqcap C_2 \in \mathcal{L}(u)$ and $\{C_1, C_2\} \not\subseteq \mathcal{L}(u)$
	then	$\mathcal{L}(u) := \mathcal{L}(u) \cup \{C_1, C_2\}$
$\sqcup$:	if	$C_1 \sqcup C_2 \in \mathcal{L}(u)$ and $\{C_1, C_2\} \cap \mathcal{L}(u) = \emptyset$
	then	$\mathcal{L}(u) := \mathcal{L}(u) \cup \{C_1\}$ or $\mathcal{L}(u) := \mathcal{L}(u) \cup \{C_2\}$
$\forall$:	if	$\forall R.C \in \mathcal{L}(u)$ and $R \in \mathcal{L}(\langle u, v \rangle)$ but $C \notin \mathcal{L}(v)$
	then	$\mathcal{L}(v) := \mathcal{L}(v) \cup \{C\}$
$\exists$:	if	$\exists R.C \in \mathcal{L}(u)$ where u is an a-node or a b-node not being l-blocked, there does not exist any node v such that $R \in \mathcal{L}(\langle u, v \rangle)$ and $C \in \mathcal{L}(v)$
	then	create a b-node v with $\mathcal{L}(v) := \{C\}$ and an edge $\langle u, v \rangle$ with $\mathcal{L}(\langle u, v \rangle) := \{R\}$

- *a*-nodes: Nodes having been located in $G_{\mathcal{A}}$ (Named Individuals)
- *b*-nodes: The others rather than *a*-nodes (Unnamed Individuals)

SmsGraph

With an input graph g to SmsGraph, we build

- A "bottom" set $B_g = \{C(u) | C \in \mathcal{L}(u)\} \cup \{R(u,v) | R \in \mathcal{L}(\langle u,v \rangle)\}$
- A "top" set $T_g = \{C(u) | C \in \Sigma_C$ and u appears in $g\} \cup \{R(u,v) | R \in \Sigma_R$ and u, v appear in $g\}$

By the Gelfond-Lifschitz transformation, a *stable set* S_g is defined such that

- $B_g \subseteq S_g \subseteq T_g$
- For a rule $r : p(\vec{u}) \leftarrow q_1(\vec{v_1}), \cdots, q_m(\vec{v_m}), \texttt{not}\ q_{m+1}(\vec{v_{m+1}}), \cdots \texttt{not}\ q_n(\vec{v_n})$ satisfying all $q_j(\sigma(\vec{v_j})) \notin S_g$ and $m+1 \leqslant j \leqslant n$, in $\mathbb{P}^{\mathcal{I},E}$, where σ is a term assignment w.r.t. g and r, if $q_i(\sigma(\vec{v_i})) \in S_g$ for each $1 \leqslant i \leqslant m$ then $p(\sigma(\vec{u})) \in S_g$

We replace the input graph g with an output set of graphs, SmsGraph(g), each of which is constructed by a stable set S_g s.t.

- Nodes are created the same as g
- An edge $\langle u, v \rangle$ is created, if $R(u,v) \in S_g$ for some R
- The labels of these nodes and edges are $\mathcal{L}(u) = \{C | C(u) \in S_g\}$ and $\mathcal{L}(\langle u,v \rangle) = \{R | R(u,v) \in S_g\}$

l-Blocking

The n-tree of a node u is the tree that includes the node u and its descendants, whose distance from u is at most n successor edges
We denote the set of nodes in the n-tree of u by $V_n(u)$

Definition (n-Tree Equivalence)

Two nodes u, v in a graph G are said to be n-tree equivalent if there is an isomorphism $\psi : V_n(v) \rightarrow V_n(u)$ s.t. (1) $\psi(v) = u$; (2) $\mathcal{L}(s) = \mathcal{L}(\psi(s))$, for every $s \in V_n(v)$; (3) $\mathcal{L}(\langle s, t \rangle) = \mathcal{L}(\langle \psi(s), \psi(t) \rangle)$, for every $s, t \in V_n(v)$.

Definition (n-Witness)

A node u is an n-witness of a node v in a graph G if (1) u is an ancestor of v, (2) u is n-tree equivalent to v, and (3) v is not in the n-tree of u.

Definition (n-Blocked)

A node w is n-blocked in a graph G if (1) one of its ancestors is n-blocked, or (2) w is in an n-tree of which root has an n-witness. Suppose u be an n-witness of v and $\psi : V_n(v) \rightarrow V_n(u)$ the isomorphism. For any node w in the n-tree of v, w is n-blocked by $\psi(w)$.

Parameter l

The parameter of l will take the value of $l_{\mathcal{K}}$ for the KB satisfiability problem and of $l_{\mathcal{K},Q}$ for the query entailment problem, given that

- $l_{\mathcal{K}}$: The maximal of l_r for all rules r in $\mathbb{P}^{\mathcal{T},E}$
  - l_r: The number of variables in a rule r
- l_Q: The length of a query Q
  - A conjunctive query (CQ) $q : \{p_1(\vec{w_1}), \cdots, p_n(\vec{w_n})\}$, and the length of q is denoted as $l_q = n$
  - A union of conjunctive queries (UCQ) $q' : q_1 \vee \cdots \vee q_m$, and the length of q' is denoted as $l_{q'} = \mathtt{max}\{l_{q_j} | 1 \leqslant j \leqslant m\}$
- $l_{\mathcal{K},Q}$: The maximal of $l_{\mathcal{K}}$ and l_Q

The KB Satisfiability Problem

Lemma (Termination)

For an $\mathcal{ALC}^u_{\mathbb{P}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$, the algorithm terminates.

Lemma (Soundness and Completeness)

Suppose $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$ be an $\mathcal{ALC}^u_{\mathbb{P}}$ KB. The algorithm yields a complete and clash-free $I_{\mathcal{K}}$-completion graph, if and only if, $\mathcal{K}$ is satisfiable.

Theorem

The algorithm is a decision procedure for the satisfiability of an $\mathcal{ALC}^u_{\mathbb{P}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$, and decides the KB satisfiability problem in 3ExpTime w.r.t. the size of $\mathcal{K}$.

The Query Entailment Problem

Lemma (Termination)

For an $\mathcal{ALC}^u_{\mathbb{P}}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$ and a query Q, the algorithm terminates.

Lemma (Soundness and Completeness)

Suppose $\mathcal{K} = (\mathcal{T}, \mathcal{A}, \mathbb{P})$ be an $\mathcal{ALC}^u_{\mathbb{P}}$ KB and Q be a query. $\mathcal{K} \vdash Q$ if and only if $\mathcal{K} \models Q$.

Theorem

The algorithm is a decision procedure for the entailment of an $\mathcal{ALC}^u_{\mathbb{P}}$ KB $\mathcal{K}$ to a query Q, and decides the query entailment problem in 3ExpTime w.r.t. the size of $\mathcal{K}$.

Contents

1. Introduction
   - Hybrid Knowledge Bases
   - Homogeneous Knowledge Bases
   - Related Work
2. $\mathcal{ALC}^u_{\mathbb{P}}$
   - Syntax
   - Semantics
   - Algorithm
3. Conclusion

Conclusion

- Syntax
  - An $\mathcal{ALC}_{\mathbb{P}}^{u}$ KB consists of a TBox $\mathcal{T}$ of subsumptions, an ABox $\mathcal{A}$ of assertions, and a novel PBox $\mathbb{P}$ of general rules that share predicates with DL concepts and DL roles
- Semantics
  - Extended Herbrand structures are used for interpreting DL concepts & roles, while open answer sets hold for general rules
- Algorithm
  - DL tableaux-based algorithms are developed for decision procedures of the KB satisfiability & query entailment problems
- Ongoing work
  - The unary/binary $\mathcal{ALC}_{\mathbb{P}}^{u}$ logic extended for *n*-ary relations
  - The extensions of $\mathcal{ALC}_{\mathbb{P}}^{u}$ towards higher OWL layers, e.g., $\mathcal{SHIF}(\mathbf{D})$ and $\mathcal{SHOIN}(\mathbf{D})$, w.r.t. corresponding DL tableaux-based algorithms that integrate general rules

References

Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf.
AL-log: Integrating Datalog and Description Logics.
Journal of Intelligent Information Systems (JIIS), 10(3):227–252, 1998.

Alon Y. Levy and Marie-Christine Rousset.
Combining Horn Rules and Description Logics in CARIN.
Artifician Intelligence, 104(1-2):165–209, 1998.

Riccardo Rosati.
DL+log: Tight Integration of Description Logics and Disjunctive Datalog.
In *Proceedings of the 10th KR*, pages 68–78. AAAI Press, 2006.

Benjamin N. Grosof, Ian Horrocks, Raphael Volz, and Stefan Decker.
Description Logic Programs: Combining Logic Programs with Description Logic.
In *Proceedings of the 12th International World Wide Web Conference*, pages 48–57. ACM, 2003.

Ian Horrocks, Peter F. Patel-Schneider, Sean Bechhofer, and Dmitry Tsarkov.
OWL Rules: A Proposal and Prototype Implementation.
Journal of Web Semantics, 3(1):23–40, 2005.

Boris Motik, Ulrike Sattler, and Rudi Studer.
Query Answering for OWL-DL with rules.
Journal of Web Semantics, 3(1):41–60, 2005.

Q & A

Thank You!