On Decidability and Complexity of Description Logics with Uniqueness Constraints

Vitaliy L. Khizder, David Toman, and Grant Weddell

Department of Computer Science

University of Waterloo

List of Slides

Description Logics

Description Logic *DLClass*

1. path expressions: $\mathsf{Pf} ::= f.\,\mathsf{Pf} \mid Id$
2. concept descriptions:

$$\begin{array}{rcl} D & ::= & C \\ & \mid & (\text{all } f\ D) \\ & \mid & (\text{fd } C : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}), k > 0 \\ & \mid & (\text{and } D\ D) \end{array}$$

3. subsumption constraints:

   $\Rightarrow C < D$

   $\Rightarrow$ implication question $\Sigma \models C < D$

Semantics for *DLClass*

Interpretation $(\Delta, .^I)$

$\Rightarrow$ fixes interpretation of primitive concepts and attributes

$\Rightarrow$ extended to:

1. path expressions: $Id^I = \lambda x.x$ and $f.\,\mathsf{Pf}^I = \mathsf{Pf}^I \circ f^I$

2. concept descriptions:

$$
\begin{aligned}
(\text{all } f \;D)^I &= \{o \in \Delta : f^I(o) \in D^I\} \\
(\text{fd } C : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \rightarrow \mathsf{Pf})^I &= \{o \in \Delta : \forall o' \in C^I. \\
&\qquad \textstyle\bigwedge_{i=1}^{k} \mathsf{Pf}_i^I(o) = \mathsf{Pf}_i^I(o') \Rightarrow \mathsf{Pf}^I(o) = \mathsf{Pf}^I(o')\} \\
(\text{and } D_1 \;D_2)^I &= D_1^I \cap D_2^I
\end{aligned}
$$

3. subsumption constraints: $C < D$ if $C^I \subseteq D^I$

Applications

- DEMO (Schema description)
- Reasonning about duplicates
- . . .

Example

$$\text{HOSPITAL} < (\text{fd HOSPITAL} : Name \rightarrow Id)$$

$$\text{PATIENT} < (\text{fd PATIENT} : Hospital, Number \rightarrow Id)$$

$$\begin{aligned}\text{ADMISSION} < (\text{all } Patient \ (\text{and } & (\text{all } SSN \text{ VALID}) \\ & (\text{fd PERSON} : SSN \rightarrow Id))\end{aligned}$$

Related Work

Datalog$_{nS}$

Finite set of Horn clauses (a logic program):

$$p_0(t^0, s_1^0, \ldots, s_k^0) \leftarrow p_1(t^1, s_1^1 \ldots, s_{l_1}^1), \ldots, p_k(t^k, s_1^k, \ldots, s_{l_k}^k)$$

is a *Datalog$_{nS}$ program* if

1. t^i is a *functional term*: a variable, a distinguished constant 0, or a term of the form $f(t, s_1, \ldots s_l)$ where f a function symbol, t is a functional term, and $s_1, \ldots, s_l$ are data terms,
2. s_j^i are *data terms*: variables or constants different from 0,
3. no variable appears both in a functional and a data term.

Datalog$_{nS}$ (cont.)

Properties of Datalog$_{nS}$:

- minimal model semantics
- programs can be transformed to *normal form*
- $P \models q$ for a ground atom q is DEXPTIME-complete
  $\Rightarrow$ follows from complexity for $\forall\exists\ldots\exists$ class [Führer]

Dialect *DLFD*

- simplified dialect of DLClass (only FDs):

$$\text{THING} < (\text{fd THING} : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}).$$

- can be simulated in DLClass:

$$\text{THING} < (\text{all } f \text{ THING}) \text{ for all primitive } f$$

Simulation of Datalog$_{nS}$ in *DLFD*

Path descriptions to Datalog$_{nS}$ functional terms:

$$\mathsf{Pf} = f_1.\cdots.f_k.\,Id \mapsto \begin{cases} \overline{\mathsf{Pf}}(0) = f_k(\cdots f_1(0)\cdots) \\ \overline{\mathsf{Pf}}(X) = f_k(\cdots f_1(X)\cdots) \end{cases}$$

Datalog$_{nS}$ query to DLFD implication question:

$$\Sigma_P = \{\mathsf{Pf}'_1\,.p'_1, \ldots, \mathsf{Pf}'_k\,.p'_k \to \mathsf{Pf}'\,.p' : \\ p'(\overline{\mathsf{Pf}}'(X)) \leftarrow p'_1(\overline{\mathsf{Pf}}'_1(X)), \ldots, p'_k(\overline{\mathsf{Pf}}'_k(X)) \in P\},$$

$$\varphi_{P,G} = \mathsf{Pf}_1\,.p_1, \ldots, \mathsf{Pf}_k\,.p_k \to \mathsf{Pf}\,.p$$

where $p_1(\overline{\mathsf{Pf}}_1(0)), \ldots, p_k(\overline{\mathsf{Pf}}_k(0))$ are facts in P.

Theorem 1 $P \models G \iff \Sigma_P \models \varphi_{P,G}$.

Simplified *DLClass*

- terminology simplification (C_1 not in $\Sigma \cup \{C' < D'\}$):

$$\Sigma \cup \{C < (\text{and } D_1\ D_2)\} \models C' < D'$$
$$\iff \Sigma \cup \{C < D_1, C < D_2\} \models C' < D',$$
$$\Sigma \cup \{C < (\text{all } f\ D)\} \models C' < D'$$
$$\iff \Sigma \cup \{C < (\text{all } f\ C_1), C_1 < D\} \models C' < D'.$$

- "question" simplification (Σ simple, C_1 not in $\Sigma \cup \{C < D, C' < D'\}$): $\Sigma \models C < (\text{and } D_1\ D_2)$

$$\iff \Sigma \models C < D_1 \text{ and } \Sigma \models C < D_2,$$
$$\Sigma \models C < (\text{all } f\ D)$$
$$\iff \Sigma \cup \{C < (\text{all } f\ C_1)\} \cup \{C_1 < C_2 : \Sigma \models C < (\text{all } f\ C_2)\} \models C_1 < D.$$

Simulation of ***DLClass*** in Datalog$_{nS}$

$$
\begin{array}{lll}
P_\Sigma = \{ & \mathsf{cl}(X, c_j, Y) \leftarrow \mathsf{cl}(X, c_i, Y) & \text{for } C_i < C_j \in \Sigma \\
& \mathsf{cl}(f(X), c_j, Y) \leftarrow \mathsf{cl}(X, c_i, Y) & \text{for } C_i < (\text{all } f\ C_j) \in \Sigma \\
& \multicolumn{2}{l}{\mathsf{eq}(\overline{\mathsf{Pf}}(X)) \leftarrow \mathsf{cl}(X, c_i, 1), \mathsf{cl}(X, c_j, 2), \mathsf{eq}(\overline{\mathsf{Pf}}_1(X)), \ldots, \mathsf{eq}(\overline{\mathsf{Pf}}_k(X))} \\
& \multicolumn{2}{l}{\mathsf{eq}(\overline{\mathsf{Pf}}(X)) \leftarrow \mathsf{cl}(X, c_i, 2), \mathsf{cl}(X, c_j, 1), \mathsf{eq}(\overline{\mathsf{Pf}}_1(X)), \ldots, \mathsf{eq}(\overline{\mathsf{Pf}}_k(X))} \\
& & \text{for } C_i < (\text{fd } C_j : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}) \in \Sigma\ \} \\
& & \\
& \mathsf{cl}(X, Y, 1) \leftarrow \mathsf{eq}(X), \mathsf{cl}(X, Y, 2) & \\
& \mathsf{cl}(X, Y, 2) \leftarrow \mathsf{eq}(X), \mathsf{cl}(X, Y, 1) & \\
& \mathsf{eq}(f(X)) \leftarrow \mathsf{eq}(X) & \text{for all primitive attributes } f
\end{array}
$$

DLClass in Datalog$_{nS}$ (cont.)

$$P_\varphi = \begin{cases} \{\mathsf{cl}(0, c_i, 1)\} & \text{for } C_i < C_j \\ \{\mathsf{cl}(0, c_i, 1)\} & \text{for } C_i < (\text{all } f\ C_j) \\ \{\mathsf{cl}(0, c_i, 1), \mathsf{cl}(0, c_j, 2), \mathsf{eq}(\overline{\mathsf{Pf}}_1(0)), \ldots, \mathsf{eq}(\overline{\mathsf{Pf}}_k(0))\} & \\ & \text{for } C_i < (\text{fd } C_j : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}) \end{cases}$$

$$G_\varphi = \begin{cases} \mathsf{cl}(0, c_j, 1) & \text{for } C_i < C_j \\ \mathsf{cl}(f(0), c_j, 1) & \text{for } C_i < (\text{all } f\ C_j) \\ \mathsf{eq}(\overline{\mathsf{Pf}}(0)) & \text{for } C_i < (\text{fd } C_j : \mathsf{Pf}_1, \ldots, \mathsf{Pf}_k \to \mathsf{Pf}) \end{cases}$$

Theorem 2 $\Sigma \models \varphi \iff P_\Sigma \cup P_\varphi \models G_\varphi.$

Polynomial Cases

Results

(normal) Datalog$_{nS}$ $\longrightarrow$ DLFD $\longrightarrow$ DLClass $\longrightarrow$ Datalog$_{nS}$

Corollary 3 *The logical implication problem for DLClass is DEXPTIME-hard.*

Extensions and Future Work

We know how to add:

- Rôles
- Equational constraints
- Or (disjunction)

We would like, but . . .

- Limited negation
- Limited inverses ($Id = f.g$)

  $\Rightarrow Id = f.g.h$: undecidable (with PFDs)

Conclusion