On Attributes, Roles, and Dependencies in Description Logics and their relation to the Ackermann Case of the Decision Problem

David Toman and Grant Weddell

Department of Computer Science

University of Waterloo

List of Slides

Description Logics

Variable-free logical formalisms

– Atomic constructs:

  - concepts: sets of objects

    $\Rightarrow$ unary predicates

  - roles: relationships between concepts

    $\Rightarrow$ binary predicates

    $\Rightarrow$ we investigate the power of **attributes**

– Constructors for complex concept/role descriptions

$\Rightarrow$ a fragment of FO

. . . often with preferable computational properties

Dialects $\mathcal{DLF}$, $\mathcal{DLFR}$, and $\mathcal{DLFD}$

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

$$
\begin{array}{llll}
D := C & & \checkmark\checkmark\checkmark & (C)^{\mathcal{I}} \subset \Delta \\
\quad \mid & D_1 \sqcap D_2 & \checkmark\checkmark\checkmark & (D_1)^{\mathcal{I}} \cap (D_2)^{\mathcal{I}} \\
\quad \mid & \forall f.D & \checkmark\checkmark\checkmark & \{x : (f)^{\mathcal{I}}(x) \in (D)^{\mathcal{I}}\} \\
\quad \mid & \neg D & \checkmark\checkmark\checkmark & \Delta \setminus (D)^{\mathcal{I}} \\
\quad \mid & \forall R.D & \checkmark & \{x : \forall y \in \Delta.(x,y) \in (R)^{\mathcal{I}} \Rightarrow y \in (D)^{\mathcal{I}}\} \\
\quad \mid & \exists R.D & \checkmark & \{x : \exists y \in \Delta.(x,y) \in (R)^{\mathcal{I}} \wedge y \in (D)^{\mathcal{I}}\} \\
E := D & & \checkmark\checkmark\checkmark & \\
\quad \mid & E_1 \sqcap E_2 & \checkmark & (E_1)^{\mathcal{I}} \cap (E_2)^{\mathcal{I}} \\
\quad \mid & \forall f.E & \checkmark & \{x : (f)^{\mathcal{I}}(x) \in (E)^{\mathcal{I}}\} \\
\quad \mid & D\{\,\mathsf{Pf}_1^{\sim_1}, ..., \mathsf{Pf}_k^{\sim_k}\,\} \rightarrow \mathsf{Pf}^{\sim} & \checkmark & \{x : \forall\, y \in (D)^{\mathcal{I}}.\ \bigwedge_{i=1}^{k} (\mathsf{Pf}_i)^{\mathcal{I}}(x) \sim_i (\mathsf{Pf}_i)^{\mathcal{I}}(y) \\
 & & & \qquad \Rightarrow (\mathsf{Pf})^{\mathcal{I}}(x) \sim (\mathsf{Pf})^{\mathcal{I}}(y)\}
\end{array}
$$

3. subsumption constraints $D \sqsubseteq E$
   implication question $\Sigma \models D \sqsubseteq E$

University of Waterloo

Applications

- Database schema and physical layout description

  $\Rightarrow$ ordered collections of records

  $\Rightarrow$ automatic plan generation (for C)

- Duplicates, Uniqueness, and Ordering in SQL/OQL

  $\Rightarrow$ "perfect" characterization of **`distinct`** removal

  ... as a $\mathcal{DLF}$ subsumption constraint [KTW00]

- Ordered Plans for Joins

  $\Rightarrow$ removal of **`sort`** operations from plans

- Reasoning about information integration

  $\Rightarrow$ network management

- ...

Results

- Lower bound and decision procedure for $\mathcal{DLF}$

  $\Rightarrow$ DEXPTIME-complete

- Simulation of Role constructors via attributes

  $\Rightarrow$ $\mathcal{DLFR}$ can be simulated by $\mathcal{DLF}$
  $\Rightarrow$ still DEXPTIME-complete

- Path order dependencies

  $\Rightarrow$ inference of ordering of objects
  $\Rightarrow$ DEXPTIME-complete
  (under syntactic restriction on dependencies)

Ackermann Formulæand Datalog$^{\vee}_{nS}$

Definition 1 (Monadic Ackermann Formulæ)
Let P_i be monadic predicate symbols and x, y_i, z_i variables. *A* monadic first-order formula in the Ackermann class *is a formula of the form $\exists z_1 \ldots \exists z_k \forall x \exists y_1 \ldots \exists y_l . \varphi$ where φ is a quantifier-free formula over the symbols P_i.*

Definition 2 (Datalog$^{\vee}_{nS}$) [Chomicki&Imelinski]
A Datalog$^{\vee}_{nS}$ program Π is a finite set of clauses constructed from monadic predicates P_i and their negations, unary function symbols f_i, constants, and a single variable x.

A satisfiability problem *for a Datalog$^{\vee}_{nS}$ program Π is the question if Π has a model.*

Datalog$^{\vee}_{nS}$ (cont.)

Properties of Datalog$^{\vee}_{nS}$:

- minimal model semantics
- programs can be transformed to *normal form*
  - $\Rightarrow$ unary predicates
  - $\Rightarrow$ single variable or ground terms
- satisfiability is DEXPTIME-complete
  - $\Rightarrow$ hardness by reduction to $\exists\forall\exists\ldots\exists$ class [Führer]

Simulating Datalog$_{nS}$ in $\mathcal{DLF}$

- terminology from clauses in Π:

$$\mathcal{T}_\Pi = \{ \quad \forall\,\mathsf{Pf}_1\,.Q_1 \sqcap \ldots \sqcap \forall\,\mathsf{Pf}_k\,.Q_k \sqsubseteq \forall\,\mathsf{Pf}\,.P : \\ P(\overline{\mathsf{Pf}}(x)) \leftarrow Q_1(\overline{\mathsf{Pf}}_1(x)), \ldots, Q_k(\overline{\mathsf{Pf}}_k(x)) \in \Pi \quad \}$$

- posed question from ground facts $Q_i(\overline{\mathsf{Pf}}_i(0)) \in \Pi$ and the goal $G = P(\overline{\mathsf{Pf}}(0))$:

$$\mathcal{C}_{\Pi,G} = \forall\,\mathsf{Pf}_1\,.Q_1 \sqcap \ldots \sqcap \forall\,\mathsf{Pf}_k\,.Q_k \sqsubseteq \forall\,\mathsf{Pf}\,.P$$

Theorem 3 *Let Π be a Datalog$_{nS}$ program and G a goal. Then $\Pi \models G \iff \mathcal{T}_\Pi \models \mathcal{C}_{\Pi,G}$.*

Corollary 4 *The implication problem for $\mathcal{DLF}$ is DEXPTIME hard; this remains true when negation is removed from $\mathcal{DLF}$.*

Simulating $\mathcal{DLF}$ in Datalog$^{\vee}_{nS}$

- Monadic predicates $P_D(x)$ represent concepts D
- Structural rules:

$$\Pi_{\mathcal{DLF}} = \left\{ \begin{array}{l} \forall x.(P_D(x) \vee P_{\neg D}(x)), \forall x.\neg(P_D(x) \wedge P_{\neg D}(x)) \\ \forall x.P_{D_1 \sqcap D_2}(x) \leftrightarrow (P_{D_1}(x) \wedge P_{D_2}(x)) \\ \forall x.P_{\forall f.D}(x) \leftrightarrow P_D(f(x)) \end{array} \right\}$$

- Question mapping:

$\Rightarrow\ \Pi_{\mathcal{T}} = \{\forall x.P_D(x) \rightarrow P_E(x) : D \sqsubseteq E \in \mathcal{T}\}$

$\Rightarrow\ \Pi_{\mathcal{C}} = \{P_D(0), P_{\neg E}(0)\}$.

Theorem 5 *Let $\mathcal{T}$ and $\mathcal{C}$ be a terminology and inclusion dependency in $\mathcal{DLF}$, respectively. Then*

$$\mathcal{T} \models \mathcal{C} \iff \Pi_{\mathcal{DLF}} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}} \text{ is not satisfiable .}$$

Roles vs. Attributes

Definition 6 ($\exists$-rank) *Let $\mathcal{T} \models \mathcal{C}$ be a $\mathcal{DLFR}$ implication problem. The number of distinct occurrences of $\exists R.D$ in $\mathcal{T} \cup \{\mathcal{C}\}$ is denoted $\mathrm{Rank}(\mathcal{T}, \mathcal{C})$.*

IDEA: pick ρ and $\delta_0, \ldots, \delta_{\mathrm{Rank}(\mathcal{T},\mathcal{C})}$, to be function symbols neither in $\mathcal{T}$ nor in $\mathcal{C}$. Then

1) role R is modeled by a monadic predicate $P_R(x)$ and

2) $(o, o') \in (R)^{\mathcal{I}} \iff P_R(\delta_i(o)) \wedge o' = \rho\delta_i(o)$

Roles vs. Attributes (cont.)

Additional “Structural” rules:

$$\Pi_{\mathcal{DLFR}} = \Pi_{\mathcal{DLF}} \cup \left\{ \begin{array}{l} \forall x . P_{\exists R.D}(x) \leftrightarrow \bigvee_{j=0}^{l} \left[P_R(\delta_j(x)) \land P_D(\rho(\delta_j(x))) \right] \\ \forall x . P_{\forall R.D}(x) \leftrightarrow \bigwedge_{j=0}^{l} \left[P_R(\delta_j(x)) \to P_D(\rho(\delta_j(x))) \right] \end{array} \right\}$$

Theorem 7 *Let $\mathcal{T}$ and $\mathcal{C}$ be a $\mathcal{DLFR}$ terminology and inclusion dependency. Then* $\mathcal{T} \models \mathcal{C} \iff \Pi_{\mathcal{DLFR}} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$ is not satisfiable. *The later can be decided in DEXPTIME.*

Role Inverse, etc.

IDEA: similar to Roles, except we need to account for the possibility of parent in the R^-1 relation in a tree model.

Structural rules:

$$
\begin{array}{lll}
\forall x.P_{\exists R.D}(x) & \leftrightarrow \bigvee_{j=0}^{l} [P_R(\delta_j(x)) \wedge P_D(\rho(\delta_j(x)))] & \text{for } x \neq \rho(\delta_i(y)) \\
\forall x.P_{\exists R.D}(\rho(\delta_i(x))) & \leftrightarrow \bigvee_{j=0}^{l} [P_R(\delta_j(\rho(\delta_i(x)))) \wedge P_D(\rho(\delta_j(\rho(\delta_i(x)))))] & \\
 & \qquad \vee [P_{R^{-1}}(\delta_i(x)) \wedge P_D(x)] & \text{otherwise} \\
\forall x.P_{\forall R.D}(x) & \leftrightarrow \bigwedge_{j=0}^{l} [P_R(\delta_j(x)) \rightarrow P_D(\rho(\delta_j(x)))] & \text{for } x \neq \rho(\delta_i(y)) \\
\forall x.P_{\forall R.D}(\rho(\delta_i(x))) & \leftrightarrow \bigwedge_{j=0}^{l} [P_R(\delta_j(\rho(\delta_i(x)))) \rightarrow P_D(\rho(\delta_j(\rho(\delta_i(x)))))] & \\
 & \qquad \wedge [P_{R^{-1}}(\delta_i(x)) \rightarrow P_D(x)] & \text{otherwise}
\end{array}
$$

Similarly: numerical restrictions, role constructors,
role hierarchies, roles with arity > 2, . . .

Order Dependencies

Syntactic guarantee of satisfiability:

Definition 8 (Guarded Order Dependency)
An order dependency $D_1 \sqsubseteq D_2\{\,\mathsf{Pf}_1^{\sim_1},\ldots,\mathsf{Pf}_k^{\sim_k}\,\} \to \mathsf{Pf}^{\sim}$ *is* guarded *if it satisfies the following conditions (conditions 2 and 3 are needed only for linear order):*

1. *if* $= \subseteq \sim_i$ *for all* $0 < i \leq k$ *then also* $= \subseteq \sim$*,*
2. *if* $< \subseteq \sim_i$ *for all* $0 < i \leq k$ *then also* $< \subseteq \sim$ *and*
3. *if* $> \subseteq \sim_i$ *for all* $0 < i \leq k$ *then also* $> \subseteq \sim$*,*

where $\subseteq$ *denotes set inclusion of relations.*

$\Rightarrow$ term model ordered by “depth-first” numbering
$\Rightarrow$ non dependency-based questions reduce to $\mathcal{DLF}$

OD's ($D_1 \sqsubseteq D_2\{\,\mathsf{Pf}_1^{\sim 1}, \ldots, \mathsf{Pf}_k^{\sim k}\,\} \to \mathsf{Pf}^{\sim}$)

IDEA: Construct *two* tree models to *test* the posed question.

$P_D^L(x)$ and $P_D^R(x)$: class membership in the two models
$O^{\sim}(x)$: a constraint $\sim$ between matching nodes in the models

$$\Pi_{\mathcal{DLFD}} = \Pi^L_{\mathcal{DLF}} \cup \Pi^R_{\mathcal{DLF}}$$
$$\cup \left\{ \begin{array}{l} \forall x.\neg(O^{<}(x) \wedge O^{=}(x)) \\ \forall x.\neg(O^{=}(x) \wedge O^{>}(x)) \\ \forall x.\neg(O^{<}(x) \wedge O^{>}(x)) \\ \forall x.O^{\leq}(x) \leftrightarrow O^{<}(x) \vee O^{=}(x) \\ \forall x.O^{\geq}(x) \leftrightarrow O^{>}(x) \vee O^{=}(x) \end{array} \right\} \cup \left\{ \begin{array}{l} \forall x.O^{=}(x) \to O^{=}(f(x)) \\ \forall x.O^{=}(x) \to (P_D^L(x) \leftrightarrow P_D^R(x)) \end{array} \right\}$$

for linear order: $\ldots \cup \{\forall x.O^{<}(x) \vee O^{=}(x) \vee O^{>}(x)\}$.

OD's ($D_1 \sqsubseteq D_2\{\,\mathsf{Pf}_1^{\sim 1},\ldots,\mathsf{Pf}_k^{\sim k}\,\} \rightarrow \mathsf{Pf}^{\sim}$)

Definition 9 *Let $\mathcal{T} \models \mathcal{C}$ be a simple $\mathcal{DLFD}$ problem $\mathcal{C}$ has the form $D_1 \sqsubseteq D_2\{\,\mathsf{Pf}_1^{\sim 1},\ldots,\mathsf{Pf}_k^{\sim k}\,\} \rightarrow \mathsf{Pf}^{\sim}$ $\mathcal{T}'$ be all pure dependencies in $\mathcal{T}$ and $\mathcal{T}'' = \mathcal{T} \setminus \mathcal{T}'$. We define*

$$\Pi_{\mathcal{T}} = \Pi^L_{\mathcal{T}'} \cup \Pi^R_{\mathcal{T}'}$$

$$\cup \left\{ \begin{array}{r} \forall x.P^L_{D_i}(x) \wedge P^R_{D_j}(x) \wedge O^{\sim 1}(\overline{\mathsf{Pf}}_1(x)) \wedge \ldots \wedge O^{\sim k}(\overline{\mathsf{Pf}}_k(x)) \rightarrow O^{\sim}(\overline{\mathsf{Pf}}(x)) \\ \forall x.P^R_{D_i}(x) \wedge P^L_{D_j}(x) \wedge O^{\sim 1}(\overline{\mathsf{Pf}}_1(x)) \wedge \ldots \wedge O^{\sim k}(\overline{\mathsf{Pf}}_k(x)) \rightarrow O^{\sim}(\overline{\mathsf{Pf}}(x)) \\ \textit{for } D_i \sqsubseteq D_2\{\,\mathsf{Pf}_1^{\sim 1},\ldots,\mathsf{Pf}_k^{\sim k}\,\} \rightarrow \mathsf{Pf}^{\sim} \in \mathcal{T}'' \end{array} \right\}$$

$$\Pi_{\mathcal{C}} = \{P^L_{D_1}(0), P^R_{D_2}(0), O^{\sim 1}(\overline{\mathsf{Pf}}_1(0)), \ldots, O^{\sim k}(\overline{\mathsf{Pf}}_k(0)), \neg O^{\sim}(\overline{\mathsf{Pf}}(0))\}.$$

Theorem 10 *Let $\mathcal{T} \models \mathcal{C}$ be a simple $\mathcal{DLFD}$ implication problem in which $\mathcal{C}$ is an order dependency. Then $\mathcal{T} \models \mathcal{C}$ if and only if $\Pi_{\mathcal{DLFD}} \cup \Pi_{\mathcal{T}} \cup \Pi_{\mathcal{C}}$ is not satisfiable.*

University of Waterloo

Conclusion and Open Problems

Main Contributions:

1. Complete picture of expressiveness and complexity for attribute-based dialects of DL's
2. A decision procedure for a large family of path order dependencies (within the original complexity bound)

Open Issues:

- Roles that are inverses of attributes
- Interaction of dependencies and roles
  (undecidable in general)
- Lifting of the Guarded requirement for OD's
- Tractable subclasses