First Steps Towards Taming Description Logics with Strings

Stéphane Demri

Joint work with Karin Quaas
(Leipzig University, Germany)

GT FM & AI, January 2024
Reasoning on Ontologies with Description Logics

- Ontology: formal specification of some domain with concepts, objects, relationships between concepts, objects, etc.

- Backbone of ontologies includes:
  - taxonomy (classification of objects),
  - axioms (to constrain the models of the defined terms).

- Description logics are well-known logical formalisms dedicated to ontologies.
  [Baader et al, Book 2017]

- BioPortal (http://bioportal.bioontology.org/): huge amount of ontologies such as Cell Ontology and Plant Ontology to facilitate scientific activities.

- Computational properties.
  - Acceptable trade-off between expressivity and complexity.
  - Decidability and often tractability.
  - Implementation in tools of the main reasoning tasks.
Description Logics with Concrete Domains

• Need to express concrete properties about data in ontologies. (e.g. age, duration, name, size, etc.)

• Concrete domain $\mathcal{D} = (\mathbb{D}, R_1, R_2, \ldots)$: fixed non-empty domain with a family of relations.

• $(\mathbb{N}, <, +1)$, $(\mathbb{Q}, <, =)$, $(\mathbb{N}, <, =)$, $(\{0, 1\}^*, <_{\text{pre}}, <_{\text{suf}})$.

• Concrete domain RCC8 with space regions in $\mathbb{R}^2$ contains topological relations between spatial regions.
  See e.g. [Wolter & Zakharyaschev, KR’00]

• General scheme for integrating concrete domains in DLs.
  [Baader & Hanschke, IJCAI’91]
  – declarative semantics close to the usual semantics for DLs,
  – generic extensions of DLs with various concrete domains,
  – tableaux-based algorithms combined with theory reasoning.
Methods for Handling Concrete Domains

- Tableaux-based decision procedures for $\omega$-admissible concrete domains. [Lutz & Miličić, JAR 2007]
  - $\mathcal{R} = (\mathbb{R}, <, =, (\equiv)_{r \in \mathbb{R}})$ is $\omega$-admissible.
  - $\mathcal{N} = (\mathbb{N}, <, =, (\equiv)_{n \in \mathbb{N}})$ is not $\omega$-admissible.

- Translation into a decidable extension of MSO with bounding quantifier B. [Carapelle & Turhan, ECAI’16]
  - EHD approach developed with $\text{Bool}(\text{MSO}, \text{WMSO}+\text{B})$ over infinite trees of finite branching degree.
    [Carapelle & Kartzow & Lorhey, JCSS 2016]
  - Decidability of concept satisfiability problem w.r.t. general TBoxes for $\mathcal{ALC}(\mathcal{N})$.

- Translation into Rabin tree automata over finite alphabets using approximations for satisfiable symbolic interpretations.
  - Concept satisfiability problem w.r.t. general TBoxes for $\mathcal{ALC}(\mathcal{N})$ in $\text{ExpTime}$. [Labai & Ortiz & Šimkus, KR’20]
Finite Strings with the Prefix Relation

- $\mathcal{D}_\Sigma = (\Sigma^*, \prec_{\text{pre}}, =, (\equiv_w)_{w \in \Sigma^*})$.

- $(\mathbb{N}, <, =, (\equiv_n)_{n \in \mathbb{N}})$ corresponds to $\mathcal{D}_\Sigma$ with singleton $\Sigma$.

\[
\exists r_0 r_1 r_2 \cdot (\text{name} \prec_{\text{pre}} \text{SSS name}) \land \forall r_0 \cdot (\text{name} \prec_{\text{pre}} S \text{name})
\]

(‘$S$’ similar to next in temporal logics)

- Concept satisfiability problem w.r.t. general TBoxes for $\mathcal{ALC}(\mathcal{W})$ with $\mathcal{W} = (\Sigma^*, \cdot, =, (\equiv_w)_{w \in \Sigma^*})$ is undecidable.

  [Lutz, PhD 2002]

- No known decidability results for description logics with $\mathcal{D}_\Sigma$. 

**ALC in a Nutshell**

- Complex concepts.

\[ C ::= \top | \bot | A | \neg C | C \cap C | C \cup C | \exists r.C | \forall r.C, \]

with concept names \( A \) and role names \( r \).

- Interpretation \( \mathcal{I} \overset{\text{def}}{=} (\Delta^\mathcal{I}, \cdot^\mathcal{I}) \)
  
  \(-\Delta^\mathcal{I}: \) non-empty set (the domain).
  
  \(-\cdot^\mathcal{I}: \) *interpretation function* such that

\[ A^\mathcal{I} \subseteq \Delta^\mathcal{I} \quad r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I} \]

Concept name \( A \)/role name \( r \)

\( \approx \)

unary predicate/binary predicate
Set-Theoretical Semantics for Complex Concepts

\[ \top^\mathcal{I} \overset{\text{def}}{=} \Delta^\mathcal{I} \]

\[ \bot^\mathcal{I} \overset{\text{def}}{=} \emptyset \]

\[ (\neg C)^\mathcal{I} \overset{\text{def}}{=} \Delta^\mathcal{I} \setminus C^\mathcal{I} \]

\[ (C_1 \sqcup C_2)^\mathcal{I} \overset{\text{def}}{=} C_1^\mathcal{I} \cup C_2^\mathcal{I} \]

\[ (C_1 \sqcap C_2)^\mathcal{I} \overset{\text{def}}{=} C_1^\mathcal{I} \cap C_2^\mathcal{I} \]

\[ (\exists r. C)^\mathcal{I} \overset{\text{def}}{=} \{ a \in \Delta^\mathcal{I} \mid r^\mathcal{I}(a) \cap C^\mathcal{I} \neq \emptyset \} \]

\[ (\forall r. C)^\mathcal{I} \overset{\text{def}}{=} \{ a \in \Delta^\mathcal{I} \mid r^\mathcal{I}(a) \subseteq C^\mathcal{I} \} \]

\[ R(a) \overset{\text{def}}{=} \{ b \mid (a, b) \in R \} \]
Inclusion and Decision Problem \( \text{TSAT}(\mathcal{ALC}) \)

- General concept inclusions \( C \sqsubseteq D \) (GCIs).
  
  \[ \text{E.g., } \text{Employee} \sqsubseteq \exists \text{WorksFor}. \top \]

\[ \mathcal{I} \models C \sqsubseteq D \iff C^\mathcal{I} \subseteq D^\mathcal{I} \]

- Terminological Box (TBox) \( \mathcal{T} \): finite collection of GCIs.

- Interpretation \( \mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I}) \), TBox \( \mathcal{T} \).

\[ \mathcal{I} \models \mathcal{T} \iff \text{for all } C \sqsubseteq D \in \mathcal{T}, \mathcal{I} \models C \sqsubseteq D \]

- \textit{Concept satisfiability problem w.r.t. general TBoxes (TSAT(\mathcal{ALC}))}:

  \textbf{Input:} A concept \( C_0 \) and a TBox \( \mathcal{T} \).

  \textbf{Question:} Is there an interpretation \( \mathcal{I} \) such that \( \mathcal{I} \models \mathcal{T} \) and \( C_0^\mathcal{I} \neq \emptyset \)?

- \( \text{TSAT}(\mathcal{ALC}) \) is \( \text{ExpTime} \)-complete.
Description Logic $\mathcal{ALC}^{\mathcal{P}}(\mathcal{D}_\Sigma)$

$$\exists r_0 r_1 r_2 \cdot (\text{name} <_{\text{pre}} \text{SSS name})$$

- New (atomic) concepts of the form $\exists P.[\Theta]$ and $\forall P.[\Theta]$:  
  - non-empty sequence $P$ of role names (role path),
  - Boolean constraint $\Theta$ built over terms of the form $S^j x$ with $j \leq |P|$ and atomic constraints of the form

\[
\begin{align*}
t <_{\text{pre}} t' & \quad \text{and} \quad t = t' = v(t) \text{ (also written } t = w) 
\end{align*}
\]

- Interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I}, v)$ with $v : \Delta^\mathcal{I} \times \text{VAR} \rightarrow \Sigma^*$. 

- $(r_1 r_2 \ldots r_n)^\mathcal{I} \overset{\text{def}}{=} \text{set of tuples } (a_0, \ldots, a_n) \text{ in } (\Delta^\mathcal{I})^{n+1} \text{ such that } (a_{i-1}, a_i) \in r_i^\mathcal{I} \text{ for all } i \in [1, n]$. 

- Satisfaction relation $\mathcal{I}, \pi = (a_0, a_1, \ldots, a_n) \models \Theta$:

  \[
  \begin{array}{cccccc}
  a_0 & \xrightarrow{r_1} & a_1 & \xrightarrow{r_2} & a_2 & \cdots & \cdots & \xrightarrow{r_n} & a_n \\
  x & \quad & Sx & \quad & S^2x & \quad & \cdots & \quad & S^n x \\
  y & \quad & Sy & \quad & S^2y & \quad & \cdots & \quad & S^n y \\
  \end{array}
  \]

- $\mathcal{I}, \pi \models S^i x <_{\text{pre}} S^j y \overset{\text{def}}{=} v(a_i, x) <_{\text{pre}} v(a_j, y)$, (similar for $=$)

- $\mathcal{I}, \pi \models S^i x = w \overset{\text{def}}{=} v(a_i, x) = w$, 

9
Clauses for Interpreting the Concepts Involving $\mathcal{D}_\Sigma$

$$(\exists P. \llbracket \Theta \rrbracket)^\mathcal{I} \overset{\text{def}}{=}$$

$$\{ a_0 \in \Delta^\mathcal{I} \mid \exists a_1, \ldots, a_n \in \Delta^\mathcal{I} \text{ s.t. } \pi = (a_0, a_1, \ldots, a_n) \in P^\mathcal{I} \text{ and } \mathcal{I}, \pi \models \Theta \}$$

$$(\forall P. \llbracket \Theta \rrbracket)^\mathcal{I} \overset{\text{def}}{=}$$

$$\{ a_0 \in \Delta^\mathcal{I} \mid \forall a_1, \ldots, a_n \in \Delta^\mathcal{I}, \pi = (a_0, a_1, \ldots, a_n) \in P^\mathcal{I} \text{ implies } \mathcal{I}, \pi \models \Theta \}$$

- $\mathcal{ALC} \mathcal{F}^P (\mathcal{D}_\Sigma)$ has also functional role names, omitted today.

- Integrating concrete domains may come with some extra costs: we need to solve a (potentially infinite) constraint satisfaction problem.

- No finite interpretation property, e.g. with

$$\mathcal{I} = \{ \top \sqsubseteq \exists r. [x <_{\text{pre}} Sx] \}$$
Main Steps For Getting ExpTime Upper Bound

- Automata-based approach with tree constraint automata (TCA) accepting infinite data trees with domain $\Sigma^*$.

- Step 0: to transform the instance so that every concept is in *simple form*, proper form to perform Step 1.

- Step 1: $C_0, T \rightarrow A$ s.t. $C_0, T$ is a positive instance of TSAT iff $L(A) \not= \emptyset$.

- Step 2: $A$ (for $\mathcal{D}_{\Sigma}$) $\rightarrow A'$ (for $\mathcal{N}$) such that
  \[
  L(A) \not= \emptyset \text{ iff } L(A') \not= \emptyset
  \]

- Step 3: complexity analysis to get $\text{ExpTime}$-completeness of TSAT. (based on [Demri & Quaas, CONCUR’23]).
**Concepts in Simple Form**

- A concept is in simple form if it is in NNF and all its role paths are of length at most one.

- \( \exists rr'.[S^2y <_{pre} x] \) not in simple form and concepts below in simple form:

  \[ \exists r.\exists r'.\exists \varepsilon.[y <_{pre} x^{++}] \quad \top \subseteq \forall r.[x = Sx^+] \quad \top \subseteq \forall r'.[x^+ = Sx^{++}] \]

- Given \( C_0, \mathcal{T} \), one can construct in polynomial-time \( C'_0, \mathcal{T}' \) in simple form s.t. \( C_0, \mathcal{T} \) positive instance of TSAT iff \( C'_0, \mathcal{T}' \) positive instance of TSAT.
Tree Interpretation Property Needed!
Tree Automata on Strings
(or how to recognize infinite data trees)

- A accepts infinite trees $t : [0, d - 1]^* \rightarrow (\Sigma \times (\Sigma^*)^\beta)$.
  (yes, two alphabets involved!)

- Transitions $(q, a, (\Theta_1, q_1), \ldots, (\Theta_d, q_d))$ put constraints on values of current node and children nodes.

- B"uchi and Rabin acceptance conditions.

- Can be adapted to many concrete domains and extends similar definitions for the linear case with $d = 1$.
  E.g. [Segoufin & Toruńczyk, STACS’11; Kartzow & Weider, arXiv 2015]
From TSAT to Nonemptiness
(or how to apply the standard automata-based approach)

• Technically involved construction following a standard pattern.
  – $C_0, \mathcal{T}$ in simple form positive instance of TSAT iff $L(\mathcal{A}) \neq \emptyset$.
  – Locations are propositionally $\mathcal{T}$-consistent set of subconcepts.
  – Each role name has dedicated directions in $[1, d]$.
  – Constraints at the level of concepts translated at the level of transitions.

• Postponing the actual problem to the nonemptiness problem.

• Advantages of translating into TCA:
  – Reveals the size of automaton (depending on parameters like number of variables, maximal size for constants, etc.) – important for complexity!
  – Same construction for other concrete domains.
From String Constraints to Integer Constraints

- Intuition: encoding of prefix of strings \( w \) and \( w' \) by length of common prefix (which is a nonnegative integer).

- \( \text{clen}(w, w') \): length of longest common prefix btw. \( w \) and \( w' \). E.g. \( \text{clen}(aba, abbbab) = 2 \).

- Properties (I)–(III) are “complete” to recover string values in a greedy way \((k = \text{card}(\Sigma))\). [Demri & Deters, JLC 2015]

(I) For \( w, w' \in \Sigma^* \), \(|w| = \text{clen}(w, w) \geq \text{clen}(w, w')\).

(II) For all \( w_0, w_1, \ldots, w_k \in \Sigma^* \) such that
- \( \text{clen}(w_0, w_1) = \cdots = \text{clen}(w_0, w_k) \) and,
- for all \( i \in [0, k] \), \( \text{clen}(w_0, w_1) < |w_i| \),
there are \( i \neq j \in [1, k] \) such that \( \text{clen}(w_0, w_1) < \text{clen}(w_i, w_j) \).

(III) For all \( w_0, w_1, w_2 \in \Sigma^* \),
\( \text{clen}(w_0, w_1) < \text{clen}(w_1, w_2) \) implies \( \text{clen}(w_0, w_1) = \text{clen}(w_0, w_2) \).
Lifting at the Level of Automata

- **TCA** $\mathbb{A} = (Q, \Sigma, d, \beta, Q_{\text{in}}, \delta, F)$ on $\mathcal{D}_\Sigma$ translated into $\mathbb{A}' = (Q, \Sigma, d, \beta', Q_{\text{in}}, \delta', F)$ on $\mathcal{N}$.

- $L(\mathbb{A}) \neq \emptyset$ iff $L(\mathbb{A}') \neq \emptyset$.

- $L(\mathbb{A}') \neq \emptyset$ checked in time

$$R_1 \left( \text{card}(Q) \times \text{card}(\delta') \times \text{MCS}(\mathbb{A}') \times \text{card}(\Sigma) \times R_2(\beta') \right)^{O(R_2(\beta') \times R_3(d))}$$

[Demri & Quaas, CONCUR'23]

- the $R_i$'s are polynomials,

- $\text{MCS}(\mathbb{A}')$: maximal size of a constraint in $\mathbb{A}'$,

- $\text{MCS}(\mathbb{A}')$ in $(\beta + \text{MCS}(\mathbb{A}) \times \text{card}(\delta) \times d)^{O(\text{card}(\Sigma)+3)}$,

- $\beta'$ polynomial in $\beta$ and in the number of constant strings in $\mathbb{A}$. 
Final Complexity Analysis

$C_0, \mathcal{T}$ is a positive instance iff $L(\mathcal{A}) \neq \emptyset$ and $\mathcal{A}$ satisfies the following quantitative properties:

- Degree $d$ bounded by $\text{size}(C_0, \mathcal{T})$.

- Number of locations in $2^\Theta(\text{size}(C_0, \mathcal{T}))$.

- Number of transitions in $2^\Theta(R(\text{size}(C_0, \mathcal{T})))$ for some polynomial $R(\cdot)$.

- Number of variables $\beta$ bounded by $\text{size}(C_0, \mathcal{T})$.

- Cardinality of finite alphabet $\Sigma$ bounded by $2^{\text{size}(C_0, \mathcal{T})}$.

- $\text{MCS}(\mathcal{A})$ quadratic in $\text{size}(C_0, \mathcal{T})$. 
Open Problem Related to XPath on Data Trees

\[(\exists r_0 r_1 \cdot \text{name} <_{\text{pre}} \exists r_0 r_2 r_1 \cdot \text{name})\]

- How to handle this extension and characterise its complexity? \(\text{ExpTime}\)-membership?

- Can we adapt results about XPath on data trees? See e.g. [Figueira, ToCL 2012]
Concluding Remarks

\[
\text{TSAT}(\mathcal{ALC} P (\mathcal{D}_\Sigma)) \rightarrow \text{TSAT}(\mathcal{ALC} P (\mathcal{D}_\Sigma)) \rightarrow \text{NE(TCA}(\mathcal{D}_\Sigma)) \rightarrow \text{NE(TCA}(\mathbb{N}))
\]

in simple form

- First steps towards taming description logics over strings.
- Automata-based approach with tree constraint automata.
- Reuse or adaptations of several results from literature with new insights to combine them.
- How to extend the results with suffix relation \( \lesssim_{\text{suf}} \) or regularity constraints?