Decidability & Undecidability

$L_2$ is the fragment of first-order predicate logic using only two different variable names. \textit{(note: variable names can be reused!).}

$L_2^\equiv$: $L_2$ plus equality.

**Theorem**

$L_2^\equiv$ is decidable.

**Corollary**

Subsumption and satisfiability of concept descriptions is decidable in description logics using only the following concept and role forming operators: $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall r.C$, $\exists r.C$, $r \sqsubseteq s$, $r \sqcap s$, $r \sqcup s$, $\neg r$, $r^{-1}$.

**Potential problems:** Role composition and cardinality restrictions for role fillers. Cardinality restrictions, however, are not a real problem.
Undecidability

- \( r \circ s, r \cap s, \neg r, 1 \) [Schild 88]
  
  ... already shown by Tarski (for relation algebras)

- \( r \circ s, r \leq s, C \cap D, \forall r.C \) [Schmidt-Schauß 89]
  
  ... This is, in fact, a fragment of the early description logic KL-ONE, where people had hoped to come up with a complete subsumption algorithm

Decidable, polynomial-time cases

- \( FL^- \) has obviously a polynomial subsumption problem (in the empty TBox) – the SUB algorithm needs only quadratic time.

- Donini et al. [IJCAI 91] have shown that in the following languages subsumption can be decided using only polynomial time:

  \[ C := A | \neg A | T | \bot | C \cap C' | \forall r.C | (\geq nr) | (\leq nr) \]

  \[ r := t | r^{-1} \]

  and

  \[ C := A | C \cap C' | \forall r.C | \exists r \]

  \[ r := t | r^{-1} | r \cap r' | r \circ r' \]
How hard is $\text{ALC}$ subsumption?

**Proposition**

$\text{ALC}$ subsumption and unsatisfiability are co-NP-hard.

**Proof.**

Unsatisfiability and subsumption are reducible to each other. We give a reduction from UNSAT. A propositional formula $\varphi$ over the atoms $a_i$ is mapped to $\pi(\varphi)$:

- $a_i \mapsto a_i$
- $\psi \land \psi' \mapsto \pi(\psi) \land \pi(\psi')$
- $\psi' \lor \psi \mapsto \pi(\psi) \lor \pi(\psi')$
- $\neg \varphi \mapsto \neg \pi(\varphi)$

Obviously, $\varphi$ is satisfiable iff $\pi(\varphi)$ is satisfiable (use structural induction). If $\varphi$ has a model, construct a model for $\pi(\varphi)$ with just one element $t$ standing for the truth of the atoms and the formula. Conversely, if $\pi(\varphi)$ satisfiable, pick one element $d \in \pi(\varphi)$ and set the truth value of atom $a_i$ according to the fact that $d \in \pi(a_i)^2$.

January 30, 2013 Nebel, Wölfl, Hué – KRR 12 / 31

How hard does it get?

- Is $\text{ALC}$ unsatisfiability and subsumption also complete for co-NP?
- Likely – since models of a single concept description can already become exponentially large!
- We will show PSPACE-completeness, whereby hardness is proved using a complexity result for (un)satisfiability in the modal logic $\mathcal{K}$.
- Satisfiability and unsatisfiability in $\mathcal{K}$ is PSPACE-complete.

January 30, 2013 Nebel, Wölfl, Hué – KRR 13 / 31

Reduction from $\mathcal{K}$-satisfiability

**Lemma (Lower bound for $\text{ALC}$)**

$\text{ALC}$ subsumption, unsatisfiability and satisfiability are all PSPACE-hard.

**Proof.**

Extend the reduction given in the last proof by the following two rules – assuming that $b$ is a fixed role name:

- $\Box \varphi \mapsto \forall b. \pi(\varphi)$
- $\Diamond \varphi \mapsto \exists b. \pi(\varphi)$

Again, obviously, $\varphi$ is satisfiable iff $\pi(\varphi)$ is satisfiable (again using structural induction). If $\varphi$ has a Kripke model, interpret each world $w$ as an object in the universe of discourse, that is, $w$ is an instance of the primitive concept $\pi(a_i)$ iff $a_i$ is true in $w$. For the converse direction use the interpretation the other way around.

January 30, 2013 Nebel, Wölfl, Hué – KRR 14 / 31

Computational complexity of $\text{ALC}$ subsumption

**Lemma (Upper Bound for $\text{ALC}$)**

$\text{ALC}$ subsumption, unsatisfiability and satisfiability are all in PSPACE.

**Proof.**

This follows from the tableau algorithm for $\text{ALC}$. Although there may be exponentially many closed constraint systems, we can visit them step by step generating only one at a time. When closing a system, we have to consider only one role at a time – resulting in an only polynomial space requirement, i.e., satisfiability can be decided in PSPACE.

**Theorem (Complexity of $\text{ALC}$)**

$\text{ALC}$ subsumption, unsatisfiability and satisfiability are all PSPACE-complete.
Further consequences of the reducibility of $K$ to $\mathcal{ALC}$

- In the reduction we used only one role symbol. Are there modal logics that would require more than one such role symbol?
  - The multi-modal logic $K_n$ has $n$ different Box operators $\Box_i$ (for $n$ different agents).
  - $\mathcal{ALC}$ (wrt. TBox reasoning) is a notational variant of $K_n$.
  - [Schild, IJCAI-91]

- Are there other modal logics that correspond to other descriptions logics?
  - propositional dynamic logic (PDL), e.g., transitive closure, composition, role inverse, ...

- DL can be thought as fragments of first-order predicate logic. However, they are much more similar to modal logics.

- Algorithms and complexity results can be borrowed. Works also the other way around.

Expressive power vs. complexity

- Of course, one wants to have a description logic with high expressive power. However, high expressive power implies usually that the computational complexity of the reasoning problems might also be high, e.g., $\mathcal{FL}^-$ vs. $\mathcal{ALC}$.

- Does it make sense to use languages such as $\mathcal{ALC}$ or even extensions (corresponding to PDL) with higher complexity?

- There are three approaches to this problem:
  - Use only small description logics with complete inference algorithms.
  - Use expressive description logics, but employ incomplete inference algorithms.
  - Use expressive description logics with complete inference algorithms.

- For a long time, only options 1 and 2 were studied. Meanwhile, most researcher concentrate on option 3!
Is subsumption in the empty TBox enough?

- We have shown that we can reduce concept subsumption in a given TBox to concept subsumption in the empty TBox.
- However, it is not obvious that this can be done in polynomial time . . .
- In the following example unfolding leads to an exponential blowup:

\[ C_1 = \forall r.C_0 \sqcap \forall s.C_0 \]
\[ C_2 = \forall r.C_1 \sqcap \forall s.C_1 \]
\[ \vdots \]
\[ C_n = \forall r.C_{n-1} \sqcap \forall s.C_{n-1} \]

- Unfolding \( C_n \) leads to a concept description with a size \( \Omega(2^n) \).
- Is it possible to avoid this blowup? Can we avoid exponential preprocessing?

Complexity of TBox subsumption

**Theorem (Complexity of TBox subsumption)**

TBox subsumption for \( \mathcal{FL}_0 \) is NP-hard.

**Proof sketch.**

We use the NDFA-equivalence problem, which is NP-complete for cycle-free automata and PSPACE-complete for general NDAs. We transform a cycle-free NDFA to a \( \mathcal{FL}_0 \)-terminology with the mapping \( \pi \) as follows:

- automaton \( A \) \( \mapsto \) terminology \( T_A \)
- state \( q \) \( \mapsto \) concept name \( q \)
- terminal state \( q_f \) \( \mapsto \) concept name \( q_f \)
- input symbol \( r \) \( \mapsto \) role name \( r \)
- \( r \)-transition from \( q \) to \( q' \) \( \mapsto \) \( q \sqsubseteq \ldots \sqcap \forall r : q' \sqcap \ldots \)

```
q_1 = \forall a.q_3 \sqcap \forall a.q_2
q_2 = \forall a.q_3 \sqcap \forall a.q_5
q_3 = \forall b.q_4
q_4 = \forall b.q_f \sqcap \forall c.q_f
q_5 = \forall b.q_6
q_6 = \forall b.q_f
q_1 \sqsubseteq \forall abc.q_f \sqcap \forall abb.q_f \sqcap
\forall abc.q_f \sqcap \forall abb.q_f
q_2 \sqsubseteq \forall abb.q_f \sqcap \forall abc.q_f
q_1 \sqsubseteq \forall r q_2 \quad \text{and} \quad \mathcal{L}(q_2) \subseteq \mathcal{L}(q_1)
```

In general, we have: \( \mathcal{L}(q) \subseteq \mathcal{L}(q') \) iff \( q' \sqsubseteq q \), from which the correctness of the reduction and the complexity result follows.
What does this complexity result mean?

- Note that for expressive languages such as $\mathcal{ALC}$, we do not notice any difference!
- The TBox subsumption complexity result for less expressive languages does not play a large role in practice
- Pathological situations do not happen very often.
- In fact, if the definition depth is logarithmic in the size of the TBox, the whole problem vanishes.
- However, in order to protect oneself against such problems, one often uses lazy unfolding . . .
- Similarly, also for $\mathcal{ALC}$ concept descriptions, one notices that they are usually very well behaved.

Outlook

- Description logics have a long history (Tarski's relation algebras and Brachman's KL-ONE).
- Early on, either small languages with provably easy reasoning problems (e.g., the system CLASSIC) or large languages with incomplete inference algorithms (e.g., the system Loom) were used.
- Meanwhile, one uses complete algorithms on very large descriptions logics (e.g., SHIQ), e.g., in the systems FaCT++ and RACER.
- Nowadays tools can handle KBs with up to 160,000 concepts (example from unified medical language system) in reasonable time.
- Description logics are used as the semantic backbone for OWL (a Web-language extending RDF).

Literature I

- Bernhard Nebel and Gert Smolka.
  Attributive description formalisms . . . and the rest of the world.

- Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt.
  Tractable concept languages.
Klaus Schild.
A correspondence theory for terminological logics: Preliminary report.

Reasoning with Individuals for the Description Logic SHIQ.

B. Nebel.
Terminological Reasoning is Inherently Intractable,