Description Logics:
a Nice Family of Logics

Day 4: More Complexity & Undecidability

ESSLLI 2016

Uli Sattler and Thomas Schneider

Next

Some

- complexity results: TBox internalisation makes a DL ExpTime-hard
- undecidability results: closing grid makes a DL undecidable
- complexity results: when 3 constructors interact badly and lead to NExpTime-hardness

Is concept satisfiability always easier than TBox reasoning?

Earlier, we have claimed/seen that, for $\mathcal{ALC}$,

- concept satisfiability is in PSpace, but
- concept satisfiability w.r.t. a TBox is in ExpTime

Next, we will see that, for $\mathcal{ALC}^u$, the extension of $\mathcal{ALC}$ with

- **universal role** u with $u^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I}$

$\Rightarrow$ concept satisfiability is as hard as reasoning w.r.t. a TBox, namely **ExpTime-hard**

- this is typical phenomenon where certain constructors enable us to **internalise a TBox**

Internalising a TBox

Definition: for $\mathcal{T} = \{C_1 \sqsubseteq D_1, \ldots, C_n \sqsubseteq D_n\}$, we use

$$C_{\mathcal{T}} = (\neg C_1 \sqcup D_1) \sqcap \ldots \sqcap (\neg C_n \sqcup D_n)$$

for the **universal $\mathcal{T}$-concept** (see Thomas' part earlier)

Reduction: for C a concept and $\mathcal{T}$ a TBox, define

$$\pi(C, \mathcal{T}) = C \sqcap \forall u.C_{\mathcal{T}}$$

Lemma:
1. C is satisfiable w.r.t. $\mathcal{T}$ iff the concept $\pi(C, \mathcal{T})$ is satisfiable
2. the size of $\pi(C, \mathcal{T})$ is linear in that of C plus $\mathcal{T}$

Corollary: satisfiability of $\mathcal{ALC}^u$ concepts is as hard as satisfiability of $\mathcal{ALC}$ concepts w.r.t. TBoxes is, namely ExpTime-hard

Let's do that again!

Is concept satisfiability always easier than TBox reasoning? (II)

Earlier, we have claimed that, for $\mathcal{ALC}$,

- concept satisfiability is in PSpace, but
- concept satisfiability w.r.t. a TBox is in ExpTime

Next, we will see that, for $\mathcal{ALCIO}$, the extension of $\mathcal{ALC}$ with

- **inverse roles** r^- with $(r^-)^\mathcal{I} = \{(y, x) \mid (x, y) \in r^\mathcal{I}\}$ and
- **nominals**, i.e., individual names used as concepts with $(\{a\})^\mathcal{I} = \{a^\mathcal{I}\}$

$\Rightarrow$ concept satisfiability is as hard as reasoning w.r.t. a TBox, namely **ExpTime-hard**

- this is typical phenomenon where the **combination** of certain constructors enables us to **internalise** a TBox

Internalising a TBox II

Remember: for $\mathcal{T} = \{C_1 \sqsubseteq D_1, \ldots, C_n \sqsubseteq D_n\}$, we use

$$C_{\mathcal{T}} = (\neg C_1 \sqcup D_1) \sqcap \ldots \sqcap (\neg C_n \sqcup D_n)$$

for the **universal $\mathcal{T}$ concept** that has to hold everywhere.[a]

[a] This is the spy-point technique developed in ML by Areces, Blackburn, Marx.

Reduction: for C a concept and $\mathcal{T}$ a TBox, define

$$\pi(C, \mathcal{T}) = C \sqcap \exists p.(\{o\} \sqcap \forall p^-.(C_{\mathcal{T}} \sqcap (\bigsqcap_r \forall r.\exists p.\{o\}))$$

Lemma: 1. C is satisfiable w.r.t. $\mathcal{T}$ iff the concept $\pi(C, \mathcal{T})$ is satisfiable

2. the size of $\pi(C, \mathcal{T})$ is linear in that of C plus $\mathcal{T}$

Corollary: satisfiability of $\mathcal{ALCIO}$ concepts is as hard as satisfiability of $\mathcal{ALCIO}$ concepts w.r.t. TBoxes, namely ExpTime-hard

Are all DLs decidable? When do they get undecidable?

Are all DLs decidable?

So far, we have extended $\mathcal{ALC}$ with

- inverse role and
- number restrictions
- ...which resulted in logics whose reasoning problems are **decidable**
- ...although we didn't discuss **decision procedures** for these extensions

Next, we will discuss some undecidable extension

- $\mathcal{ALC}$ with role chain inclusions
- $\mathcal{ALC}$ with number restrictions on complex roles

An undecidable DL: $\mathcal{ALC}$ with role chain inclusions

OWL 2 supports axioms of the form

- $r \sqsubseteq s$: a model of $\mathcal{O}$ with $r \sqsubseteq s \in \mathcal{O}$ must satisfy $r^\mathcal{I} \subseteq s^\mathcal{I}$
- $r \circ s \sqsubseteq t$: a model of $\mathcal{O}$ with $r \circ s \sqsubseteq t \in \mathcal{O}$ must satisfy $r^\mathcal{I} \circ s^\mathcal{I} \subseteq t^\mathcal{I}$
- ...

subject to some complex restrictions

...**why do we need restrictions?**

...because axioms of this form lead to **loss of tree model property** and **undecidability**

How to prove undecidability of a DL

We prove undecidability (or hardness) of a DL as follows:

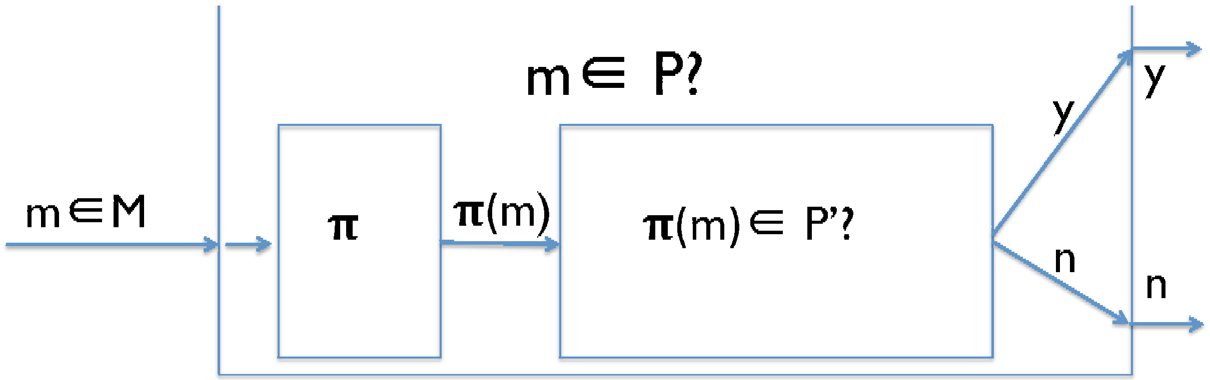
1. **fix reasoning problem**, e.g., satisfiability of a concept w.r.t. a TBox
   - remember Theorem 2?
   - if concept satisfiability w.r.t. TBox is undecidable,
   - then so is consistency of ontology
   - then so is subsumption w.r.t. an ontology
   - ...
2. **pick a decision problem** known to be undecidable, e.g., **the domino problem**
3. reduce latter to former, i.e., **provide a (computable) mapping** $\pi(\cdot)$ that
   - takes an instance D of the domino problem and
   - turns it into a concept A_D and a TBox $\mathcal{T}_D$ such that
   - D has a tiling **if and only if** A_D is satisfiable w.r.t. $\mathcal{T}_D$

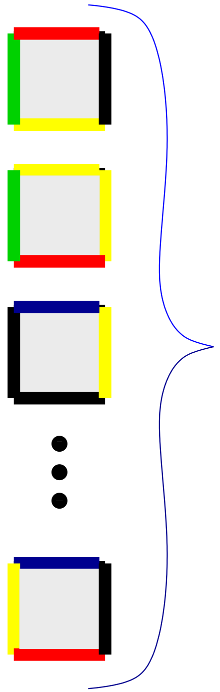
   i.e., a decision procedure of concept satisfiability w.r.t. TBoxes could be used as a decision procedure for the domino problem

How to prove undecidability of a DL

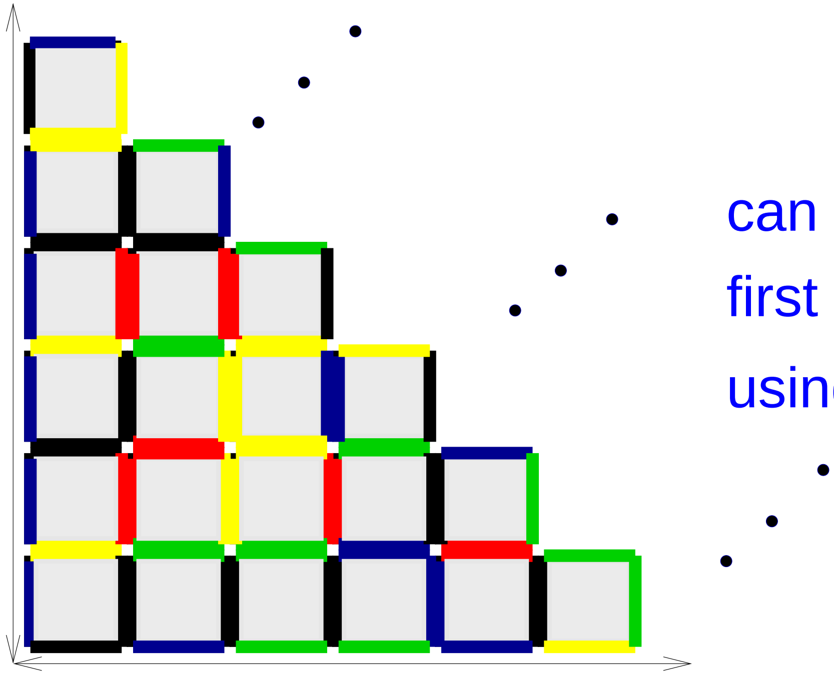
We prove undecidability (or hardness) of a DL as follows:

1. **fix reasoning problem P'**, e.g., satisfiability of a concept w.r.t. a TBox
2. **pick a decision problem P** known to be undecidable, e.g., **the domino problem**
3. reduce latter to former:





***D*,**
a fixed
set
of
dominoe
types



can we tile the
first quadrant
using ***D*?**

Definition: A **domino system** $\mathcal{D} = (D, H, V)$

- set of domino **types** $D = \{D_1, \ldots, D_d\}$, and
- horizontal and vertical **matching conditions** $H \subseteq D \times D$ and $V \subseteq D \times D$

A **tiling** for $\mathcal{D}$ is a (total) function:

$$
\begin{aligned}
t : \mathbb{N} \times \mathbb{N} \rightarrow\ & D \text{ such that} \\
& \langle t(m,n), t(m+1,n) \rangle \in H \text{ and} \\
& \langle t(m,n), t(m,n+1) \rangle \in V
\end{aligned}
$$

Domino problem: given $\mathcal{D}$, has $\mathcal{D}$ a tiling?

It is well-known that this problem is undecidable [Berger66]

Almost Encoding the Classical Domino Problem in $\mathcal{ALC}$

For our reduction, we express various **obligations** of the domino problem in $\mathcal{ALC}$ TBox axioms:

① each element carries **exactly one domino type** D_i

$\rightsquigarrow$ use unary predicate symbol D_i for each domino type and

$$
\begin{array}{rcll}
\top & \sqsubseteq & D_1 \sqcup \ldots \sqcup D_d & \text{\% each element carries a domino type} \\
D_1 & \sqsubseteq & \neg D_2 \sqcap \ldots \sqcap \neg D_d & \text{\% but not more than one} \\
D_2 & \sqsubseteq & \neg D_3 \sqcap \ldots \sqcap \neg D_d & \text{\% ...} \\
\vdots & & \vdots & \\
D_{d-1} & \sqsubseteq & \neg D_d &
\end{array}
$$

Almost Encoding the Classical Domino Problem in $\mathcal{ALC}$

② every element has a horizontal (X-) successor and a vertical (Y-) successor

$$\top \sqsubseteq \exists X.\top \sqcap \exists Y.\top$$

③ every element satisfies D's **horizontal/vertical matching conditions**:

$$\begin{aligned}
D_1 &\sqsubseteq \bigsqcup_{(D_1,D)\in H} \forall X.D \sqcap \bigsqcup_{(D_1,D)\in V} \forall Y.D \\
D_2 &\sqsubseteq \bigsqcup_{(D_2,D)\in H} \forall X.D \sqcap \bigsqcup_{(D_2,D)\in V} \forall Y.D \\
&\vdots \\
D_d &\sqsubseteq \bigsqcup_{(D_d,D)\in H} \forall X.D \sqcap \bigsqcup_{(D_d,D)\in V} \forall Y.D
\end{aligned}$$

Does this suffice?

No: if yes, $\mathcal{ALC}$ would be undecidable!

Encoding the Classical Domino Problem in $\mathcal{ALC}$ with role chain inclusions

④ for each element, its horizontal-vertical-successors **coincide** with their vertical-horizontal-successors & vice versa

$$X \circ Y \sqsubseteq Y \circ X \text{ and } Y \circ X \sqsubseteq X \circ Y$$

Lemma: Let $\mathcal{T}_D$ be the set of axioms ① to ④.
Then $\top$ is satisfiable w.r.t. $\mathcal{T}_D$ iff $\mathcal{D}$ has a tiling.

- since the domino problem is undecidable, this implies undecidability of concept satisfiability w.r.t. TBoxes of $\mathcal{ALC}$ with role chain inclusions
- due to Theorem 2, all other standard reasoning problems are undecidable, too
- Proof: 1. show that, from a tiling for D, you can construct a model of $\mathcal{T}_D$
  2. show that, from a model $\mathcal{I}$ of $\mathcal{T}_D$, you can construct a tiling for D (tricky because elements in $\mathcal{I}$ can have several X- or Y-successors but we can simply take **the right ones**...)