ExpTime Hardness of $\mathcal{ALC}$
Declarative Knowledge Processing

Magdalena Ortiz

Knowledge Base Systems Group
Institute of Information Systems
ortiz@kr.tuwien.ac.at

WS 2012/2013

Complexity of $\mathcal{ALC}$

Theorem:
Concept subsumption w.r.t. an $\mathcal{ALC}$ TBox (and KB satisfiability) is ExpTime-hard

We look at a proof based on encoding the two player corridor tiling problem

Two player corridor tiling game

- Tiling system $\mathbf{T}$: finite set of tile types with horizontal and vertical adjacency conditions
- A corridor tiling is a tiling of a corridor of width $n$ with tiles of $\mathbf{T}$ respecting adjacency conditions

$\forall$belard and $\exists$loise alternatively place a tile, row by row, from left to right, respecting adjacency conditions

$\exists$loise wins if
- she can place a special “winning tile” in the second position of a row, or
- she can play in such a way that $\forall$belard can no longer place a tile (i.e., $\exists$loise loses if she cannot place a tile, or if the game goes on forever)
Two player corridor tiling problem

Instance:
- a tiling system expressed as $T = (k, H, V)$, where
  - $0, 1, \ldots, k$ are the tile types, with $k$ being the winning tile
  - $H \subseteq [0..k] \times [0..k]$ is the horizontal adjacency relation
  - $V \subseteq [0..k] \times [0..k]$ is the vertical adjacency relation
- an initial row of tiles $t_1t_2\cdots t_n$ of length $n$

Question: Does $\exists$loise have a winning strategy?
i.e., for every move $\forall$belard makes, is there a move $\exists$loise can counter with in such a way that she wins?

Theorem:
- Two player corridor tiling is ExpTime-complete

Encoding of two player corridor tiling in $\mathcal{ALC}$

The intention is to represent each placed tile by an object; the object carries the information about the last $n$ moves made.

- We use an atomic role $Next$ to connect individuals representing successive tiles.
- We use the following atomic concepts:
  - $R^t_i$, for each $i \in [1..n]$ and each $t \in [0..k]$, denoting that the last tile placed in column $i$ has been tile $t$
  - $Q_i$, for $i \in [1..n]$, denoting that the next tile will be placed in column $i$
  - $A$, denoting that it is $\forall$belard’s turn to place the next tile
  - $W$, denoting that $\exists$loise wins

We now show how to build an $\mathcal{ALC}$ TBox $\mathcal{T}_T$ using these concepts and roles.

Encoding of two player corridor tiling in $\mathcal{ALC}$ (2)

We introduce in $\mathcal{T}_T$ the following GCIs to ensure that tilings are correctly represented:

- To encode that each column has exactly one tile last placed into it:
  $\top \sqsubseteq R^t_1 \sqcup \cdots \sqcup R^t_k$ for $i \in [1..n]$\n  $R^t_i \sqsubseteq \neg R^{t'}_i$ for $i \in [1..n]$, $t, t' \in [0..k]$, $t \neq t'$

- To encode that each move occurs in exactly one column in the corridor:
  $\top \sqsubseteq Q_1 \sqcup \cdots \sqcup Q_n$\n  $Q_i \sqsubseteq \neg Q_j$ for $i, j \in [1..n]$, $i \neq j$

- To encode that the tiles are placed in the correct left-to-right order:
  $Q_i \sqsubseteq \forall Next. Q_{i+1}$ for $i \in [1..n-1]$\n  $Q_n \sqsubseteq \forall Next. Q_1$

Encoding of two player corridor tiling in $\mathcal{ALC}$ (3)

We introduce in $\mathcal{T}_T$ the following GCIs to encode the adjacency conditions:

- To encode the vertical adjacency relation $V$:
  $Q_i \cap R^t_i \sqsubseteq \forall Next. (\bigcup_{t' \in [t', t] \in V} R^{t'}_i)$ for $i \in [1..n]$, $t \in [0..k]$

- To encode the horizontal adjacency relation $H$:
  $Q_i \cap R^t_{i-1} \sqsubseteq \forall Next. (\bigcup_{t' \in \{t', t\} \in H} R^{t'}_{i-1})$ for $i \in [2..n]$, $t \in [0..k]$

- To encode that in columns where no move is made nothing changes:
  $\neg Q_i \cap R^t_i \sqsubseteq \forall Next. R^t_i$ for $i \in [1..n]$, $t \in [0..k]$
  $\neg Q_i \cap R^t_i \sqsubseteq \forall Next. \neg R^t_i$ for $i \in [1..n]$, $t \in [0..k]$
Encoding of two player corridor tiling in \( ALC \) (4)

We introduce in \( T_T \) the following GCIs to encode the game:

- To encode the existence of all possible moves in the game tree, provided \( \exists \)loise hasn’t already won:
  \[
  \neg R^k_i \cap Q_i \cap R_i^{t_i-1} \cap R_i^{t'_i} \subseteq \exists \text{Next}.R_i^{t''_i}
  \quad \forall (t',t'') \in H \land (t'',t') \in V
  \text{ for } i \in [2..n], \quad t,t' \in [0..k]
  \]

- To encode the alternation of moves:
  \[
  A \subseteq \forall \text{Next}.\neg A
  \]
  \[
  \neg A \subseteq \forall \text{Next}.A
  \]

- To encode the winning of \( \exists \)loise:
  \[
  W \equiv (A \cap R^k_2) \cup (A \cap \forall \text{Next}.W) \cup (\neg A \cap \exists \text{Next}.W)
  \]

Complexity of \( ALC \)

- Since the size of \( T_T \) is polynomial in \( T \) and \( n \), this shows that concept subsumption w.r.t. to an \( ALC \) TBox is ExpTime-complete

- We can easily adapt the reduction to KB satisfiability: simply add an ABox
  \[
  \{ A \cap Q_1 \cap R_1^{t_1} \cap \cdots \cap R_n^{t_n} \cap \neg W(a) \}
  \]

- With some modifications, we can show even stronger results: the fragment of \( ALC \) that has no qualified existentials and no disjunctions is also ExpTime-hard

Tiling systems and TMs

Tiling problems are a very closely related to Turing Machine.

This particular form of a tiling is related to Alternating Turing Machines

An ATM has the form \( M = (\Sigma, Q, Q_\exists, q_0, \delta, q_f) \)

- As usual, \( \Sigma \) is the alphabet, \( q_0 \) is the initial state, \( q_f \) the final one, and \( \delta : \Sigma \times Q_\exists \rightarrow Q \times \Sigma \times \{ \text{right, left} \} \) is the transition function

- \( Q_\forall \) are universal states, and \( Q_\exists \) existential ones

- From an existential state, the machine moves to some successive configuration

- From a universal state, the machine moves to all successive configurations
Tiling systems and TMs (2)

Tiling problems are a very closely related to Turing Machines.

In fact, this form of tiling is a simple ‘disguise’ for an Alternating Turing Machine that runs in polynomial space, and we want to decide acceptance of a word

- Think of each sequence of last $n$-tiles (i.e., each row) as the tape contents
- Configurations evolving in time appear along the corridor
- The two players correspond to existential and universal states
- The winning tile is (the only) final state
- The initial configuration is the word initially written on the tape

Hardness proofs using tilings

Tiling problems are a very useful tool for showing complexity results in Modal Logics and fragments of FOL

In DLs, they are very frequently used to:

- Show NExpTime-hardness (e.g., for extensions of $\mathcal{ALCIOF}$):

  Deciding the existence of a tiling for a $2^n \times 2^n$ torus is NExpTime-complete

- Show undecidability (e.g., for DLs with transitive roles in the number restrictions, role value maps, etc.)

  Deciding the existence of a tiling for an unbounded grid is undecidable