TD 4

14/10/2009

Exercise 1. Consider the problem: *Input* A tree automaton A and a term t
Output "yes" iff at least one ground instance of t is accepted by A
Prove that the problem is *EXPTIME*-complete.

Proof. (Sketch)

1. first we prove that the problem is *EXPTIME*-easy.

   Construct A', a reduced (i.e. all states are accesible) DFTA which accepts the same language as A. The cost of this step and the size of the resulting automaton A' is at most exponential in the size of A. Furthermore, as A' is deterministic, for any ground term s such that $s \rightarrow_{A'} q_1$ and $s \rightarrow_{A'} q_2$ we must have $q_1 = q_2$.

   Let $x_1, \ldots, x_k$ be the variables appearing in t. For all substitutions $\sigma = \{x_1 \mapsto q_1, \ldots, x_k \mapsto q_k\}$, where $q_1, \ldots, q_k$ are states of A', verify if $t\sigma \rightarrow_{A'} q_f$ where q_f is a final state of A'. If this verification works for at least one substitution σ, answer 'yes'. Otherwise 'no'.

   There are at most exponentially many substitutions σ (at most n^k, where n, the number of states of A', is at most exponential in the size of the input and where k, the number of variables in t, is at most the size of the input). Each verification $t\sigma \rightarrow_{A'} q_f$ can be done in polynomial time with a suitable representation of A'.

   Therefore the entire algorithm is exponential and we can conclude that the problem is *EXPTIME*-easy.

2. we now prove that the problem is *EXPTIME*-hard.

   We reduce the problem of the emptyness of the intersection of n tree automata to our problem. Consider n state disjoint tree automata $A_1, \ldots, A_n$. Testing if $L(A_1) \cap \ldots \cap L(A_n) = \emptyset$ is *EXPTIME*-hard.

   Let s be a complete binary tree with $\lceil log_2(n) \rceil + 1$ levels (which implies there are $2^{\lceil log_2(n) \rceil}$ leaves), where all nodes which are not leaves are labelled with a binary symbol f, the first n leaves from left to right are labelled with the variables $x_1, \ldots, x_n$ and the last $2^{\lceil log_2(n) \rceil} - n$ nodes are labelled with a constant symbol a.

   W.l.o.g. we assume that every automaton $A_i = (\mathcal{F}, Q_i, \{q_i\}, \Delta_i)$ has exactly one final state q_i (otherwise add a new state which will be the only final state and add ϵ-transitions from all old final states to the new final state).

   We construct an automaton $A = (\mathcal{F} \cup \{f(,), a\}, \bigcup_i Q_i, \{q_f\}, \Delta)$, where

   $$\Delta = \bigcup_i \Delta_i \cup \{s\{x_i \mapsto q_i\} \rightarrow q_f\}$$

   and where we took some permission to write a transition rule which is not a tree automata transition rule, but which can be transformed into a set of tree automata rules which are equivalent to it by adding some intermediate states.

   We let $t = s\{x_i \mapsto x\}$ for some new variable x. Then A accepts a ground instance of t iff $L(A_1) \cap \ldots \cap L(A_n) \neq \emptyset$

□