Complexité avancée
TD 5

Cristina Sirangelo - LSV, ENS-Cachan

October 22, 2014

Exercise 1. Alternating Turing machines with negations Let us define an alternating Turing machine with negations as a Turing machine where the set of non-halting states is partitioned into the set of *existential states*, the set of *universal states* and the set of *negation states*. Moreover there is the restriction that each configuration on a negation state has exactly one successor configuration. Remark that we do not require that the machine always halts.

For such a machine $\mathcal{M}$ we define the set of *eventually accepting* configurations, and the set of *eventually rejecting* configurations as the minimal sets of configurations satisfying the following conditions :

- if C is an accepting configuration, then C is *eventually accepting* ;
- if C is an existential configuration and there exists a successor configuration C' of C (i.e, $C \rightarrow_{\mathcal{M}} C'$) which is *eventually accepting*, then C is *eventually accepting* ;
- if C is a universal configuration, and all successor configurations C' of C are *eventually accepting*, then C is *eventually accepting* ;
- if C is a negation configuration and the (unique) successor configuration C' of C is *eventually rejecting*, then C is *eventually accepting* ;

- if C is a rejecting configuration, then C is *eventually rejecting* ;
- if C is an existential configuration and all successor configuration C' of C are *eventually rejecting*, then C *is eventually rejecting* ;
- If C is universal configuration, and there exists a successor configuration C' of C which is *eventually rejecting*, then C is *eventually rejecting* ;
- if C is a negation configuration and the (unique) successor configuration C' of C is *eventually accepting*, then C is *eventually rejecting*.

The machine accepts an input x iff the initial configuration on input x is *eventually accepting*. The language accepted by an alternating Turing machine with negations $\mathcal{M}$ is the set of all x accepted by $\mathcal{M}$.

Prove that any alternating Turing machine $\mathcal{M}$ with negations can be simulated by an alternating Turing machine $\mathcal{M}^*$ without negations, with no extra cost in time or space. More precisely prove that there exists a configuration reachable in n steps and using m working tape units in $\mathcal{M}$ iff there exists a configuration reachable in n steps and using m working tape units in $\mathcal{M}^*$. Do not assume any space or time bound on $\mathcal{M}$.

Exercise 2. Alternating logarithmic time vs logarithmic space
Show that **ATIME**($log\ n$) does not coincide with **L**.

Exercise 3. Minimal Formula A boolean formula is minimal if it has no equivalent shorter formula – where the length of the formula is the number of symbols it contains. Let MIN-FORMULA be the problem of deciding whether a boolean formula is minimal. Is MIN-FORMULA in **AP**, in **NP**, in **coNP** ?

Exercise 4. Tautology and coNP

- Describe a polynomial time alternating Turing machine which decides whether a boolean formula is a tautology.
- Show that **coNP** $\subseteq$ **AP**, by exhibiting an alternating polynomial time Turing machine for each problem in **coNP**.

Exercise 5. Linearly and logarithmically bounded alternations Let **AP**($O(n)$) (resp. **AP**($O(log\ n)$)) be the class of problems which can be decided by an alternating polynomial time Turing machine whose computations have a linear (resp. logarithmic) number of alternations (in the size of the input).

- Is QBF in **AP**($O(n)$) ? in **AP**($O(log\ n)$) ?
- Can we conclude **PSPACE** = **AP**($O(n)$) ? **PSPACE** = **AP**($O(log\ n)$) ?