Summary of Homework Assignments for Part I

The requirement of Part I (for grading) is to submit a report for two problems from the following homework assignments. But you cannot choose two from the same category; e.g., Q1.1 and Q1.2.

Due date: November 30, 2015.

Method for submission:

(1) Post your report into my mail box located in the mail room of W8, which is located left right after the entrance hall of W8 building.

(2) Or submit a pdf file to `watanabe@is.titech.ac.jp`

Q1.1. Prove that a language L is regular if and only if Σ^* is divided into a finite number of equivalence classes by $\approx_L$. Then explain with some example why the number of equivalence classes cannot be finite for non-regular languages; estimate the number of equivalence classes in terms of the string length. (You may use any one of the first three definitions of regular language; but maybe the definition by DFA would be much easier.)

Q1.2. Explain some example of using regular languges (and their expressions) to capture/formulate some computational task. (Something unrelated to string processing such as string matching.)

Q2.1. Knowing how to compare algorithms by using complexity measures would be important even though you are not designing algorithms by yourselves. At least you had better understand related notions and notation sufficient enough to answer the following questions.

(1) Is it always appropriate to define a given computational task as a task of computing some function? Suppose you want to design a Chess-software, a software that plays a chess with human. For designing its core algorithm, would it be possible to specify the required task by some function (or a set of functions)?

(2) Prof. W uses his algorithm A to analyze his experimental data, which is crucial for his research project. But his assistant K analyzed this algorithm and showed that it needs $\mathrm{time}_A(\ell) = 4 \cdot 2^{0.3\ell} \cdot 10^{-2}$ seconds for analyzing ℓMbyte data. This is not efficient. So he spent some months to develop a new algorithm B whose running time on ℓMbyte data is $\mathrm{time}_B(\ell) = 4000\ell^2 \cdot 10^{-2}$ seconds. Prof. W is unhappy that his assistant spends so long time only for algorithm design; in particular, there is no so much difference for the size of data (approximately 74Mbytes) that his group needs to analyze currently. Give at least two technical explanations for supporting the effort of assistant K. We may assume that for both programs, their running time depend only on input data size. (*Hint.* The amount of data his group needs to analyze is growing. In fact, Prof. W is planning to buy a new computer (10 times faster than the current one) for preparing this data increase.)

Q2.2. Show that NP $\subseteq$ EXP holds. Give some example of NP problem that does not seem to belong $\mathrm{Time}\big(2^{\ell^2}\big)$.

Q3.1. It is not so difficult to prove Theorem 1, so why don't you prove it (without reading the referenced paper)! You can go back to the definition and consider the probability that one fixed hypothesis h is not an ϵ-approximation of a given target f_* even though h is consistent with f_* on m examples of S. (What is the randomness here for discussing the probability?) Then we can use the union bound to estimate the probability that this situation occurs on some hypothesis of $\mathcal{H}_{n,m}$.

Q3.2. Implement AdaBoost and obtain a good hypothesis for the mushroom data. The data and a sample program can be obtained from

`http://www.is.titech.ac.jp/~watanabe/class/boost/`

Use only `mushroomB5000.txt` for creating your hypothesis and check its performance with `mushroomB3000.txt`.

Q3.3. Study the proof of Theorem 1 (of Lect #5). From this proof, it is not so difficult to see the reason why β_t is defined as (1). Explain this reason.