SCIENTIFIC THEORY IN INFORMATICS (IT731A, VP707A)
SEMINAR 2: AUTOMATA THEORY AND COMPUTABILITY THEORY

Group assignment - October 9, 2019

Work on the following two exercises. Each group will submit a document with commented solutions at Canvas>Assignments>Seminar assignment 2. Clearly display in the front page the group number or team name, and the complete list of members. Include the participation matrix, showing which of the group members contributed to each task solution, together with the signatures of all the group members indicating that they agree with the allocation.
Strictly submit a document in .pdf format within the due date: **October 21, 2019**.

EXERCISE A: BAYES' THEOREM

Most medical tests occasionally produce incorrect results, *called false positives* and *false negatives*. When a test is designed to determine whether a patient has a certain disease, a false positive result indicates that a patient has the disease when the patient does not actually have it. A false negative result indicates that a patient does not have the disease when the patient actually does have it. Consider a medical test that screens for a disease found in 5 people in 1,000. Suppose that the false positive rate is 3% and the false negative rate is 1%. Furthermore, 99% of the time a person who has the condition tests positive for it, and 97% of the time a person who does not have the condition tests negative for it.

I. What is the probability that a randomly chosen person who tests positive for the disease actually has the disease? [max marks 15%]

II. What is the probability that a randomly chosen person who tests negative for the disease does not indeed have the disease? [max marks 15%]

Note: You might be surprised by these numbers, but they are fairly typical of the situation. When large-scale health screenings are performed for diseases with relatively low incidence, those who develop the screening procedures have to balance several considerations: the per-person cost of the screening, follow-up costs for further testing of false positives, and the possibility that people who have the disease will develop unwarranted confidence in the state of their health. The screening test is significantly less expensive than a more accurate test for the same disease yet produces positive results for nearly all people with the disease. Using the screening test limits the expense of unnecessarily using the more costly test to a relatively small percentage of the population being screened (i.e. to the people resulting positive and false positive), while only rarely indicating that a person who has the disease is free of it (of course, these mistaken situations also have a social cost).

III. How does the posterior probability change when the disease is found in 5 people in 100, rather than 5 in 1000 as before? Describe in simple words the general effect of variations in the prior probability on the posterior probability. [max marks 10%]

IV. In a dynamic scenario, where priors and likelihoods might evolve over time, describe in simple words what is the crucial task for an autonomous machine (i.e. a machine that operates independently on human control) whose functioning critically depends on the calculation of posterior probabilities? [max marks 10%]

EXERCISE B: TURING MACHINES

Consider the following description of a Turing machine:

- Alphabet: { *, | }
- Set of states: $\{q_0, q_1, q_2, q_3, q_4\}$
- Start state: q_0
- Transition function:

1. $q_0, | \rightarrow q_1, *, R$
2. $q_1, | \rightarrow q_2, *, R$
3. $q_2, | \rightarrow q_2, |, R$
4. $q_2, * \rightarrow q_3, |, L$
5. $q_3, | \rightarrow q_3, |, L$
6. $q_3, * \rightarrow q_4, *, R$
7. $q_1, * \rightarrow q_4, *, R$

A) Draw its state diagram. [max marks 35%]

B) Consider a simple representation where **|** stands for 0, **||** for 1, **|||** for 2, etc. Can you determine the general task of the outlined Turing machine? - For example, test your machine on the input strings **…*| | | |*| | | | |*…** (i.e. *3*4*) and **…*|*| | |*…** (*0*2*), with initial position of the R/W-head set on the leftmost **|**. [max marks 15%]

Conclusion on Turning machines:

"Once one has gained some familiarity with constructing simple Turing machines, it becomes easy to satisfy oneself that the various basic arithmetical operations, such as adding two numbers together, or multiplying them, or raising one number to the power of another, can indeed all be effected by specific Turing machines. It would not be too cumbersome to give such machines explicitly, but I shall not bother to do this here. Operations where the result is a pair of natural numbers, such as division with a remainder, can also be provided or where the result is an arbitrarily large finite set of numbers. <u>Moreover Turing machines can be constructed for which it is not specified ahead of time which arithmetical operation it is that needs to be performed, but the instructions for this are fed in on the tape</u>**.** (…) Once it is appreciated that one can make Turing machines which perform arithmetic or simple logical operations, it becomes easier to imagine how they can be made to perform more complicated tasks of an algorithmic nature."

(Penrose, *The Emperor's new mind*, 1989)

When the task of a Turing machine is not hardwired, but is defined in part of the input string (while the reminder of the input is the string to operate on), it can be used to simulate simpler automata. If we recall the fact that some Turing machines can *decide* their input strings, then we see how we can use them to prove that the languages A_{DFA}, A_{NFA} and A_{CFG} are Turing decidable. When we try the same for A_{TM}, we discover that the latter is *Turing recognizable*, but not *Turing decidable*. Therefore, we prove that some general problems, even problems of practical interest, are *not computable*.