Information-Theoretic Modeling, Fall 2009

Exercises I, due Friday 18 September.

(Note that the exercise sessions are on Fridays (not Tuesdays) in room C222, the same place as the lectures.)

1. Explain briefly your background and what made you interested in this course. What is your major subject (computer science, math, ...), and at what stage of your studies you are? How much mathematics have you studied (name some courses you have taken)? How about programming — what language(s) you use, and how would you describe your programming skills? If you have studied something else relevant, like statistics, signal processing, etc., you can also mention that. Have you studied any information theory before? Please send your answers to this question to the teaching assistant by e-mail.

2. Find out what compression algorithms are used in the gadgets you use — DVD player, digital TV, CD, iPod, cell-phone, laptop (WinZip), etc, etc. You can simply list the names of the algorithms, you don't have to study how these algorithms work.

3. Recall the Morse code exercise in the first lecture. (If you didn't attend, listen to the lecture on-line at the course web page.) Take a piece of text, at least a couple of hundred words long, and design your own "Morse" code: encode each letter in the alphabet (the size of the alphabet depends on the language) by a unique sequence of dots and dashes. Then count the length of the encoded text by the following rule: each dot costs 1 unit, each dash costs 2 units, and a space between words costs 2 units. Compare this to the actual Morse code — the Morse code-words for the English alphabet can be found in the lecture slides, other symbols you need to google. (You probably want to write a simple program to do the counting.)

4. Find the shortest program (or as short as you can) in a programming language of your choice to print the sequence 141592653589793...375678 (the first 10000 digits of π after the decimal point). Feel free to google to find an algorithm.

5. Try to compress the digits of π you have printed out in the previous exercise using your favourite compression algorithm. (For instance gzip.) Try to come up with other "simple" sequences, i.e., sequences that can be printed out by a simple program, that cannot be compressed (easily).

Bonus exercise. (This one is perhaps best understood after Friday's lecture.) What is the probability that an error occurs in a (binary) k-fold repetition code, i.e., a code where each bit is repeated k times. In other words, what is the probability that $k/2$ or more bits are flipped when each bit is flipped independently with probability $0 < p < 1$. Give an expression with the error rate p and the number of repetitions k as parameters. Can you come up with an approximation formula that is easier to evaluate? Evaluate the exact expression and the approximation for $p = 0.1$. Can you find out what number of repetitions is enough to guarantee that the error probability is less than 0.00001 %?