HOMEWORK 5

Due Tuesday, November 21

- The same rules as for previous assignments apply.
- Working through the assignment on your own will help you to learn the material and identify those areas which you need to study more.
- If you have questions, make sure to clear them up during *office hours* or by asking on the myCourses *discussion board*.

Readings:

- Read the text on Turing Machines by Hamilton (myCourses) and the textbook Chapter 13 carefully.

Problems:

1. *Quiz 3 preparation questions*

   Post two interesting questions and answers on the myCourses discussion board as study practice for the quiz on Tuesday, November 21. (The earlier you post your questions, the higher the likelihood that they will be considered for being on the quiz!)

2. *Turing Machines*

   As presented in class, a Turing Machine is defined in the text by Hamilton (p. 163) by a finite set of **quadruples** (q_i, S, Op, q_j), where the action or operation Op, can be 'move left', 'move right', or the name of a symbol that is written on the tape.

   Alternatively, it is also possible to define TMs in terms of a finite set of **quintuples** (q_1, S, N, M, q_j), where N is the name of the new symbol being written on the tape and M indicates the movement of the machine ('L' for left, 'R' for right, and '$*$' for no move).

   One can easily translate TMs presented as quadruples into TMs presented as quintuples, and vice versa. For example, the TM of Example 7.13 (Hamilton, p. 164) can be represented by quintuples in the following way:

   $$\{\ (q_0, 1, 1, R, q_1),\ (q_1, 1, 1, R, q_0),\ (q_1, B, B, R, q_2),\ (q_2, B, 1, *, q_2)\ \}.$$

   In the representation used in

   http://morphett.info/turing/turing.html

this amounts to:

```
0 1 1 R 1
1 1 1 R 0
1 _ _ R 2
2 _ 1 * 2
```

Note that here '_' is used for a blank cell.

Enter this program in the Turing Machine Simulator on the webpage, click on 'Reset' to load it, and then on 'Run' to run the TM. Play around by changing the input tape, e. g., use one with an even number of 1s and one with an odd number of 1s to observe the different behavior of the TM.

(a) Using the TM Simulator (link above), define a TM (with alphabet '_' and '1') that erases a sequence of 1s on the tape and then halts.
In addition to defining the Turing Machine as a set of quintuples, do give comments on the specific role that is played by parts of the instructions. E. g., "The TM is in state 3 if it has read an odd number of 1s."
Comments can be written in the TM Simulator after a semi-colon ';'.

(b) Using the TM Simulator (link above), define a Turing Machine (with alphabet '_' and '1') that takes as input a sequence of 1s on an otherwise blank tape and gives an output of '1' (on an otherwise blank tape) if the number of 1s in the sequence is a multiple of 3 and an output of '11' otherwise.

- Hand in printouts of the acutal programs you've written.

(c) Read Handout 5. What does the TM presented there as a state-transition diagram do?
In other words, describe its behavior depending on the input.

3. *More on Turing Machines*

(a) On p. 171, Hamilton mentions of a coding (Gödel numbering) of Turing Machines. Find a way of assigning Gödel numbers to Turing Machines (defined as quadruples, with the alphabet 1, 0, and B).
Illustrate your coding scheme with an example.

(b) Explain what a Universal Turing Machine is. Include in your explanation what its inputs and output are, and in what sense is it 'universal'.

4. *Computability*

(a) The set K (Hamilton, p. 177) is also called the *Halting Set*. Explain which numbers are in this set.

(b) Look at examples (b) and (c) of 7.32 (Hamilton, p. 178). Explain why one set is recursively enumerable, but the other isn't.

5. *BlooP programs*

   Read Handout 7 and have a look at the procedures that are defined in the TisLoopy editor:

   http://www.cs.mcgill.ca/~cs230/TisLoopy

   Write BlooP programs for the functions:

   (a) `POWER [M,N]`: Returns the value of M^N.
   (b) `EVEN [N]`: Returns `1` if the number `N` is even, and `0` otherwise (here, zero is considered an even number).
   (c) `REMAINDER [M,N]`
   (d) `PRIME-BEYOND [N]`

   See GEB, pp. 415–416 for a brief description of the functions (c) and (d).

   You may use the functions that are already defined in TisLoopy.

   • Make sure to test your programs with TisLoopy before you hand them in. All programs that you hand in must compile without errors.

6. *FlooP program*

   Explain in your own words the difference between a BlooP and a FlooP program in general, and the difference between a BlooP and a 'terminator' (total) FlooP program. Use small examples for illustration.

7. *Bluediag*

   Explain what the function *Bluediag* (GEB, p. 418–421) does and why it is not a primitive recursive function.

⋆ *Bonus question.* Read Turing's 1936 paper "On computable numbers" (myCourses). In which section does he show that some problems are not computable? Briefly summarize his argument.