COMP 2804 — Assignment 2

Due: Thursday February 14, before 11:55pm.

Assignment Policy:

- Your assignment must be submitted as one single PDF file through cuLearn.
- **Late assignments will not be accepted. I will not reply to emails of the type "my internet connection broke down at 11:53pm" or "my scanner stopped working at 11:54pm".**
- You are encouraged to collaborate on assignments, but at the level of discussion only. When writing your solutions, you must do so in your own words.
- Past experience has shown conclusively that those who do not put adequate effort into the assignments do not learn the material and have a probability near 1 of doing poorly on the exams.
- When writing your solutions, you must follow the guidelines below.
  - You must justify your answers.
  - The answers should be concise, clear and neat.
  - When presenting proofs, every step should be justified.

Question 1:

- Write your name and student number.

Question 2: The function $f : \{1, 2, 3, \ldots\} \to \mathbb{R}$ is defined by

$$\begin{aligned} f(1) &= 2, \\ f(n) &= \tfrac{1}{2}\left(f(n-1) + \tfrac{1}{f(n-1)}\right) \quad \text{if } n \geq 2. \end{aligned}$$

- Prove that for every integer $n \geq 1$,

$$f(n) = \frac{3^{2^{n-1}} + 1}{3^{2^{n-1}} - 1}.$$

  Note that $3^{2^{n-1}}$ is 3 to the power of 2^{n-1}.

Question 3: The function $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ is defined by

$$\begin{array}{rcll} f(m,0) & = & 0, & \text{if } m \geq 0, \\ f(m,n) & = & m + f(m,n-1) & \text{if } m \geq 0 \text{ and } n \geq 1. \end{array}$$

- Solve this recurrence, i.e., express $f(m,n)$ in terms of m and n only. As always, prove that your answer is correct.

Question 4: In class, we have seen that for any integer $m \geq 1$, the number of 00-free bitstrings of length m is equal to f_{m+2}, which is the $(m+2)$-th Fibonacci number.

Let $n \geq 2$ be an integer. For each of the following, justify your answer.

- How many 00-free bitstrings of length n do not contain any 0?
- How many 00-free bitstrings of length n have the following property: The rightmost 0 is at position 1.
- How many 00-free bitstrings of length n have the following property: The rightmost 0 is at position 2.
- Let k be an integer with $3 \leq k \leq n$. How many 00-free bitstrings of length n have the following property: The rightmost 0 is at position k.
- Use the previous results to prove that

$$f_{n+2} = 1 + \sum_{k=1}^{n} f_k.$$

Question 5: Let $n \geq 1$ be an integer and consider a set S consisting of n numbers. A function $f : S \rightarrow S$ is called *cool*, if for all elements x of S,

$$f(f(f(x))) = x.$$

Let A_n be the number of cool functions $f : S \rightarrow S$.

- Let $f : S \rightarrow S$ be a cool function, and let x be an element of S. Prove that the set

$$\{x, f(x), f(f(x))\}$$

  has size 1 or 3.

- Let $f : S \rightarrow S$ be a cool function, and let x and y be two distinct elements of S. Assume that $f(y) = y$. Prove that $f(x) \neq y$.
- Prove that for any integer $n \geq 4$,

$$A_n = A_{n-1} + (n-1)(n-2) \cdot A_{n-3}.$$

  Hint: Let y be the largest element in S. Some cool functions f have the property that $f(y) = y$, whereas some other cool functions f have the property that $f(y) \neq y$.

Question 6: In this exercise, we will denote Boolean variables by lowercase letters, such as p and q. A *proposition* is any Boolean formula that can be obtained by applying the following recursive rules:

1. For every Boolean variable p, p is a proposition.
2. If f is a proposition, then $\neg f$ is also a proposition.
3. If f and g are propositions, then $(f \vee g)$ is also a proposition.
4. If f and g are propositions, then $(f \wedge g)$ is also a proposition.

- Let p and q be Boolean variables. Prove that

$$\neg\left((p \wedge \neg q) \vee (\neg p \vee q)\right)$$

  is a proposition.

- Let $\uparrow$ denote the *not-and* operator. In other words, if f and g are Boolean formulas, then $(f \uparrow g)$ is the Boolean formula that has the following truth table (0 stands for *false*, and 1 stands for *true*):

f	g	$(f \uparrow g)$
0	0	1
0	1	1
1	0	1
1	1	0

  - Let p be a Boolean variable. Use a truth table to prove that the Boolean formulas $(p \uparrow p)$ and $\neg p$ are equivalent.
  - Let p and q be Boolean variables. Use a truth table to prove that the Boolean formulas $((p \uparrow p) \uparrow (q \uparrow q))$ and $p \vee q$ are equivalent.
  - Let p and q be Boolean variables. Express the Boolean formula $(p \wedge q)$ as an equivalent Boolean formula that only uses the $\uparrow$-operator. Use a truth table to justify your answer.

- Prove that any proposition can be written as an equivalent Boolean formula that only uses the $\uparrow$-operator.

Question 7: In this exercise, we consider strings of characters, where each character is an element of $\{a, b, c\}$. Such a string is called *awesome*, if it does not contain the substring ab and does not contain the substring ba. For any integer $n \geq 1$, let

1. S_n denote the number of awesome strings of length n,
2. A_n denote the number of awesome strings of length n that start with a,
3. B_n denote the number of awesome strings of length n that start with b,
4. C_n denote the number of awesome strings of length n that start with c.

- Determine S_1 and S_2.
- Let $n \geq 1$ be an integer. Express S_n in terms of A_n, B_n, and C_n.
- Let $n \geq 2$ be an integer. Express C_n in terms of S_{n-1}.
- Let $n \geq 2$ be an integer. Prove that

$$S_n = (S_{n-1} - B_{n-1}) + (S_{n-1} - A_{n-1}) + S_{n-1}.$$

- Let $n \geq 3$ be an integer. Prove that

$$S_n = 2 \cdot S_{n-1} + S_{n-2}.$$

- Prove that for every integer $n \geq 1$,

$$S_n = \frac{1}{2}\left(1+\sqrt{2}\right)^{n+1} + \frac{1}{2}\left(1-\sqrt{2}\right)^{n+1}.$$

  Hint: What are the solutions of the equation $x^2 = 2x + 1$? Using these solutions will simplify the proof.

Question 8: Consider the following recursive algorithm, which takes as input a sequence $(a_1, a_2, \ldots, a_n)$ of n numbers, where n is a power of two, i.e., $n = 2^k$ for some integer $k \geq 0$:

```
Algorithm MYSTERY(a_1, a_2, ..., a_n):

    if n = 1
    then return a_1
    else for i = 1 to n/2
           do b_i = min(a_{2i-1}, a_{2i})        (*)
           endfor;
           MYSTERY(b_1, b_2, ..., b_{n/2})
    endif
```

- Determine the output of algorithm MYSTERY$(a_1, a_2, \ldots, a_n)$. As always, justify your answer.

- For any integer $n \geq 1$ that is a power of two, let $T(n)$ be the number of times that line $(*)$ is executed when running algorithm MYSTERY$(a_1, a_2, \ldots, a_n)$. Derive a recurrence for $T(n)$ and use it to prove that for any integer $n \geq 1$ that is a power of two,

$$T(n) = n - 1.$$