CSC165, Summer 2014
Assignment 6
Weight: 8%
Solutions

1. State whether the following claim is true, and then prove or disprove it. Give a detailed structured proof, justifying every step.

$$\forall n \in \mathbb{N}, [(\exists k \in \mathbb{N}, n = 4k) \vee (\exists k \in \mathbb{N}, n = 4k + 1)]$$

Solution:

The statement is **false**. We prove its negation. The strategy is to realize that the statement is false since, for example, $2 mod 4 = 2$ and not 0 or 1, and to go back to the definition of $\mathbb{N}$.

Proof:

Let $n = 2$ Then $n \in \mathbb{N}$ # $2 \in \mathbb{N}$
Then $n \notin \{0, 4, 8, 12, ...\}$ # by inspection, 2 is not in that sorted list
Then $n \notin \{4 * 0, 4 * 1, 4 * 2, ...\}$ # algebra
Then $\neg[\exists k \in \mathbb{N}, n = 4k]$
Then also $n \notin \{1, 5, 9, 13, ...\}$ # by inspection, 2 is not in that sorted list
Then $n \notin \{4 * 0 + 1, 4 * 1 + 1, 4 * 2 + 1, ...\}$ # algebra
Then $\neg[\exists k \in \mathbb{N}, n = 4k + 1]$
Then $\neg[\exists k \in \mathbb{N}, n = 4k] \wedge \neg[\exists k \in \mathbb{N}, n = 4k + 1]$ # conjunction of two true statements
Then $\neg[[\exists k \in \mathbb{N}, n = 4k] \vee [\exists k \in \mathbb{N}, n = 4k + 1]]$ # De Morgan
Then $\exists n \in \mathbb{N}, \neg[[\exists k \in \mathbb{N}, n' = 4k] \vee [\exists k \in \mathbb{N}, n = 4k + 1]]$ # introduce existential, $n = 2$ is such an n
Then $\neg[\forall n \in \mathbb{N}, [[\exists k \in \mathbb{N}, n' = 4k] \vee [\exists k \in \mathbb{N}, n = 4k + 1]]]$ # quantifier negation

■

2. Let $\mathcal{F}$ be the set of all function from $\mathbb{N}$ to $\mathbb{R}^+$. Let $\lceil f \rceil$ be a function such that

$$\forall n \in \mathbb{N}, [(\lceil f \rceil)(n) = \lceil f(n) \rceil].$$

State whether the following claim is true, and then prove or disprove it. Give a detailed structured proof, justifying every step.

$$\forall f \in \mathcal{F}, \forall g \in \mathcal{F}, [f \in \mathcal{O}(g) \Rightarrow \lceil f \rceil \in \mathcal{O}(g)]$$

Solution:

The statement is **false**. We prove its negation. The strategy is to prove that for $f(n) = 1/(n+1)$ and $g(n) = 1/(n+1)$, $f \in \mathcal{O}(g)$ but $\lceil f \rceil = 1 \notin \mathcal{O}(g)$. (The reason we use $1/(n+1)$ is that $1/n$ is not defined for $n = 0$.

Proof:

Let $f(n) = 1/(n+1)$
Let $g(n) = 1/(n+1)$
Let $B = 1$, $c = 1$
Then $B \in \mathbb{N}, c \in \mathbb{R}$ # $1 \in \mathbb{R}^+, 1 \in \mathbb{N}$
Then $\forall n \in \mathbb{N}, [n \geqslant B] \Rightarrow [f(n) \leqslant cf(n)]$ # $f(n) \leqslant f(n) = 1 * f(n)$ always, so the consequent is always true
Then $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, [n \geqslant B] \Rightarrow [f(n) \leqslant cf(n)]$ # introduce existential, $n = 1$ and $B = 1$ are such c and B
Then $f \in \mathcal{O}(g)$ # definition of big-Oh, $g = f$
Also $\lceil f \rceil = 1$ # $\forall n \in \mathbb{N}, 0 < 1/(n+1) < 1$

Assume $c \in \mathbb{R}^+, B \in \mathbb{N}$
  Let $n = max(B+1, (\lceil c \rceil + 1)$
  Then $(B+1) \in \mathbb{N}$ # integers are closed under addition
  Also $(\lceil c \rceil + 1) \in \mathbb{N}$ # $\forall x \in \mathbb{R}^+ \lceil x \rceil \in \mathbb{N}$
  Then $n = max(B+1, \lceil c \rceil + 1) \in \mathbb{N}$ # both possibilities for the value of max are integers
  Then $n > B$ # $n = max(x, y) \Rightarrow n \geqslant x$
  Then $(n > c)$ # $(\lceil c \rceil + 1) \geqslant c + 1 > c$
  Then $1 > n/(n+1) > c/(n+1) = c * (1/(n+1_)$ # $c > n > 0$
  Then $1 > c * (1/(n+1))$ # transitivity
  Then $\neg[[n > B] \Rightarrow [1 \leqslant c * (1/(n+1))]]$ # the antecedent is true and the consequent is false
  Then $\exists n \in \mathbb{N}, \neg[[n > B] \Rightarrow [1 \leqslant c * (1/(n+1))]]$ # $n = max(B+1, \lceil c \rceil + 1) \in \mathbb{N}$ works
  Then $\neg[\forall n \in \mathbb{N}, [n > B] \Rightarrow [1 \leqslant c * (1/(n+1))]]$ # quantifier negation
Then $\forall c \in \mathbb{R}^+, \forall B \in \mathbb{N}, \neg[\forall n \in \mathbb{N}, [n > B] \Rightarrow [1 \leqslant c * (1/(n+1))]]$ # introduce universal
Then $\neg[\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, [\forall n \in \mathbb{N}, [n > B] \Rightarrow [1 \leqslant c * (1/(n+1))]]]$ # quantifier negation twice
Then $\neg[1 \in \mathcal{O}(1/(n+1)]$ # definition of big-Oh
Then $\neg[\lceil f \rceil \in \mathcal{O}(g)]$ # substitution
Then $[f \in \mathcal{O}(g)] \wedge [\neg[\lceil f \rceil \in \mathcal{O}(g)]]$ # conjunction
Then $\neg[[f \in \mathcal{O}(g)] \Rightarrow [\lceil f \rceil \in \mathcal{O}(g)]]$ # implication negation
Then $\exists f \in \mathcal{F}, \exists g \in \mathcal{F}, \neg[[f \in \mathcal{O}(g)] \Rightarrow [\lceil f \rceil \in \mathcal{O}(g)]]$ # $f(n) = 1/(n+1)$ and $g(n) = 1/(n+1)$ are such f and g
Then $\neg[\forall f \in \mathcal{F}, \forall g \in \mathcal{F}, [[f \in \mathcal{O}(g)] \Rightarrow [\lceil f \rceil \in \mathcal{O}(g)]]]$ # quantifier negation twice

■

3. The Fibonacci numbers $fib(n)$ are defined as follows:
$fib(0) = 1, fib(1) = 1$, and $fib(n) = fib(n-1) + fib(n-2)$ for $n \in \{2, 3, 4, 5, ...\}$.
Prove that
$$\forall n \in \mathbb{N}, fib(n) \leq 2^n.$$
It will be helpful to prove, using induction, that $[\forall n \in \mathbb{N}, P(n)]$ where
$$P(n) : \forall k \in \mathbb{N}, [k \leqslant n] \Rightarrow fib(k) \leq 2^k.$$

Solution:

Define
$$P(n) : \forall k \in \mathbb{N}, [k \leqslant n] \Rightarrow fib(k) \leq 2^k.$$

We would like to prove that $[\forall n \in \mathbb{N}, P(n)]$.

We first prove $P(0)$ and $P(1)$.
$1 \leqslant 1 = 2^0$ # algebra
Then $fib(0) \leqslant 2^0$ # substitution
$1 \leqslant 2 = 2^1$ # algebra
Then $fib(1) \leqslant 2^1$ # substitution
Then $\forall k \in \mathbb{N}, [k \leqslant 1] \Rightarrow fib(k) \leq 2^k$ # the consequent is true when the antecedent is true
Then $P(1)$ # substitution
Also, $P(0)$ # the consequent is true in $P(0)$ whenever it's true in $P(1)$

Base case:
$P(1)$ # proved above

Induction step:
Assume $n \in \{1, 2, 3, 4, 5, ...\}$
Assume P(n)
Then $\forall k \in \mathbb{N}, [k \leqslant n] \Rightarrow fib(k) \leq 2^k$ # substitution
Then $fib(n-1) \leq 2^{n-1}, fib(n) \leq 2^n$ # $n \leqslant n, (n-1) \leqslant n$
Then $fib(n-1) + fib(n) \leqslant 2^{n-1} + 2^n \leqslant 2^n + 2^n = 2^{n+1}$ # algebra
Then $fib(n+1) \leqslant 2^{n+1}$ # definition of fib() for $n \geqslant 1$
Then $\forall k \in \mathbb{N}, [k \leqslant (n+1)] \Rightarrow fib(k) \leq 2^k$ # proved for $k \leqslant n$ and for $k = n+1$
Then $P(n+1)$ # substitution
$P(n) \Rightarrow P(n+1)$ # introduce implication
$\forall n \in \{1, 2, 3, 4, 5, ...\}, P(n) \Rightarrow P(n+1)$ # introduce universal

We can now conclude:
$P(1)$ # proved above
$\forall n \in \{1, 2, 3, 4, 5, ...\}, P(n) \Rightarrow P(n+1)$ # proved above
Then $\forall n \in \{1, 2, 3, 4, 5, ...\}, P(n)$ # by the principle of simple induction
Also, $P(0)$ # proved above
Then $\forall n \in \mathbb{N}, P(n)$ # true for all $\mathbb{N} = \{0, 1, 2, 3, 4, ...\}$

But $P(n)$ is not exactly what we need to prove. We now conclude:

$[\forall k \in \mathbb{N}, [k \leqslant n] \Rightarrow fib(k) \leq 2^k] \Rightarrow [fib(n) \leq 2^n]$ # $n \leq n$
Then $P(n) \Rightarrow [fib(n) \leq 2^n]$ # substitution
Then $\forall n \in \mathbb{N}, [P(n) \wedge [P(n) \Rightarrow [fib(n) \leq 2^n]]$ # conjunction with a true statement Then $\forall n \in \mathbb{N}, fib(n) \leq 2^n$ # implication elimination

■

(The last part is more formal that it has to be.)

4. (a) Write a function `def lowest_terms(n,m)` that takes two integers as inputs, and returns True iff n/m is a fraction that is reduced to lowest terms. For example, `lowest_terms(2,3)` is True but `lowest_terms(4,6)` is False. In the comments to your code, explain how the code works, and argue informally (i.e., no formal proof is required) that it produces the desired output. In the comments, provide the output for 8 test cases.

   (b) In the comments in `a6.py`, give a tight upper bound on the total number of comparison operators ((==, <, >, <=, >=) and arithmetic operators (+, −, mod, /, ∗, ...) performed when running `def lowest_terms(n,m)`. Your answer should be one expression for the tight upper bound on sum of the number of comparison operators and the number of arithmetic operators. Justify your answer. (A formal proof is not required). Note: your answer will be an expression that may depend on both `m` and `n`.

   Solution:

   See `a6.py`.

5. **Bonus question, worth half the weight of the other questions:** Claims 5.3 and 5.4 in Section 5.4 in the notes present a method to list all the rational numbers. Note, however, that, if that method is used, every rational number is actually listed an infinite number of times (for example, the number 1 is listed as $1/1, 2/2, 3/3, 4/4, 5/5...$). It is possible to modify this method that produces a list of all the rational numbers such that every rational number appears in the list exactly once. Write a Python function `def r(n)` that takes an integer `n` as input and prints the n-th (starting from 0) rational number in such a list. For example, if the list begins with $\{0, 1, -1, 1/2, -1/2, 2, -2, ...\}$, `r(4)` should print "-1/2" (the output should not contain quotes). In the comments to your code, explain how the code works, and argue informally (i.e., no formal proof is required) that it produces the desired output. Also in the comments, include the output for r(0), r(1), r(2), ..., r(20). *Hints: (1) I suggest implementing code that prints a list using the method in Claim 5.3, then modifying that code to print a list using the method in Claim 5.4, and then thinking how to implement `def r(n)` by modifying the code from there. You only need to submit `def r(n)` and not anything else.* **Clearly documented** attempts at a solution that **run** and make progress towards the solution will get part marks. Submit your code and comments in `bonus6.py`.

   Solution:

   See `a6b.py`.