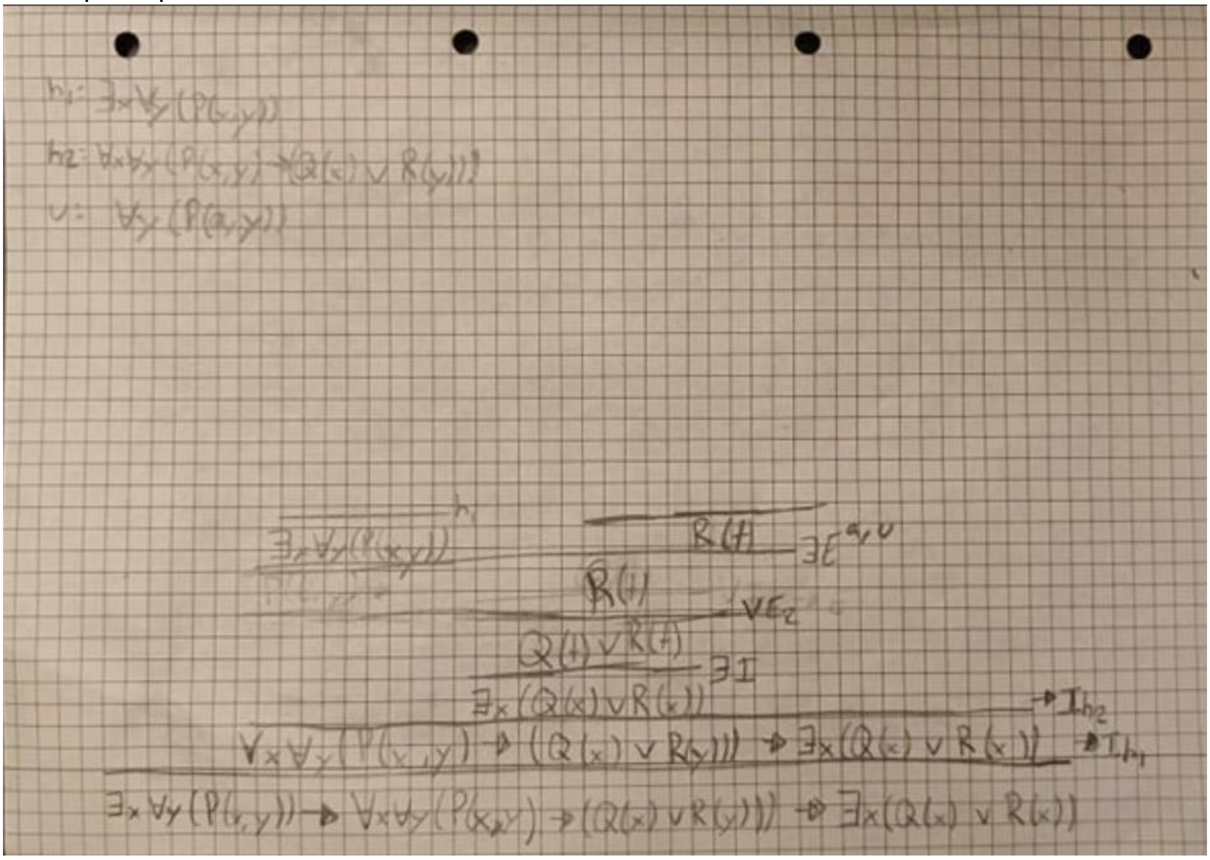
Q1

Incomplete prooftree



Q2

Prove by contradiction the following statement:

Let $a, b, c \in \mathbb{Z}^+. If\ a \mid b\ and\ b \mid c\ then\ a \mid c$.

Assume $a \nmid c$, that is, assume there is no $k \in \mathbb{Z}$ such that $a * k = c$.

From the fact that $a \mid b\ and\ b \mid c$ we have $a * k_1 = b\ and\ b * k_2 = c$ for some $k_1, k_2 \in \mathbb{Z}$.

We also have that $b = \frac{c}{k_2}$ and $a = \frac{b}{k_1}$ and thus $a = \left(\frac{\frac{c}{k_2}}{k_1}\right) = \frac{c}{k_1 k_2}$.

From this we can derive $c = ak_1k_2$ and since $k_1, k_2 \in \mathbb{Z}$, then $k_1k_2 = k \in \mathbb{Z}$.

Thus we have: $a * k = c$ which is a contradiction to our assumption and thus the statement $a \nmid c$ does not hold. Instead we can conclude that $a \mid c$.

QED.

Q3

Let $n \in \mathbb{N}$. Prove by induction that $\sum_{i=0}^{n} 6i^2 = n(n+1)(2n+1)$.

Inductive hypothesis:

$$P(k) \coloneqq \sum_{i=0}^{k} 6i^2 = k(k+1)(2k+1), for\ all\ k \in \mathbb{N}$$

Base case:

$$P(0) = \sum_{i=0}^{0} 6i^2 = 6 * 0^2 = 0(0+1)(20+1) = 0$$

Thus the base case holds.
Inductive case:
Assume $P(k-1)$ holds to prove $P(k)$ holds.

$$P(k-1) = \sum_{i=0}^{k-1} 6i^2 = (k-1)\big((k-1)+1\big)(2(k-1)+1)$$

From the definition of summation we know that $P(k) = P(k-1) + 6k^2$

$$P(k) = P(k-1) + 6k^2 = (k-1)(k)(2*(k-1)+1) + 6k^2$$

We multiply the two leftmost parentheses and the inner parenthesis in the furthest right side of the equality sign.

$$P(k) = (k^2 - k)(2k - 2 + 1) + 6k^2$$

We reduce multiply the parentheses.

$$P(k) = 2k^3 - k^2 - 2k^2 + k + 6k^2$$

We reduce.

$$P(k) = 2k^3 + 3k^2 + k$$

We factorize k.

$$P(k) = k(2k^2 + 3k + 1)$$

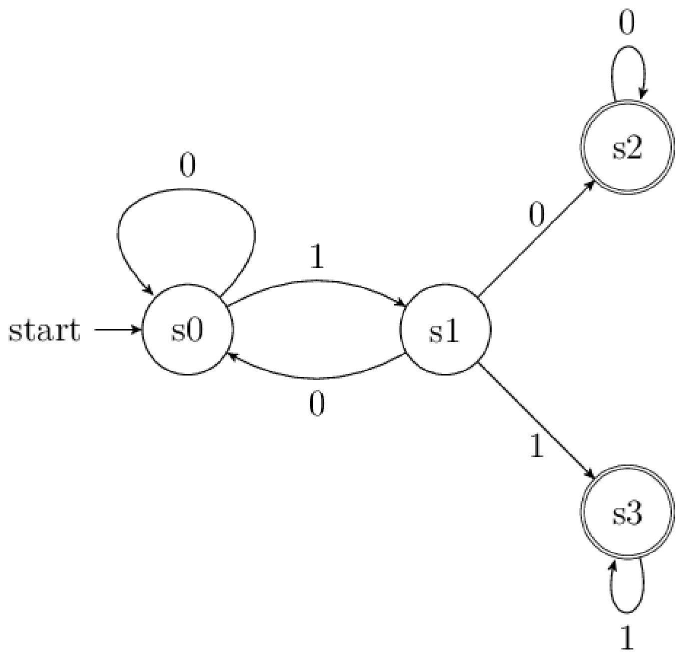
Note that $(2k^2 + 3k + 1) = (k+1)(2k+1)$.

$$P(k) = k(k+1)(2k+1)$$

QED.

Q4

Consider the following finite state automaton:



Present a grammar that recognizes exactly the same language.

Let $G = (V, T, S, P)$ with $V = \{0,1, A, B, C, S\}, T = \{0,1\}$ and the set of productions:

$$P = \{S \to 0S, S \to 1A, A \to 0S, A \to 0B, B \to 0B, B \to \lambda, A \to 1C, C \to 1C, C \to \lambda\}$$

This way, S can produce any number of 0s or go to s_1 by writing 1 and then having the non-terminal A. A can either go back to S with a 0 or it can go to 1C or 0B. When it is in B or C it can terminate (by going to the empty string lambda), or it can repeat B->0B (an infinite number of 0s) or repeat C->1C (an infinite number of 1s).