Problem Set 3.

Xuanqi Wei

1009353209

Q4.

proof: Let $n, m \in \mathbb{N}$. and $n < 2^m$.

WTS: $\forall m, n \in \mathbb{N}, n < 2^m$, calling $r(m, n)$ terminates and $Post(m, n)$.

I'm proving using complete induction on m.

Base Case: $m = 0$.

Since $n \in \mathbb{N}$ and $n < 2^m = 2^0 = 1$, gives $n = 0$.

In this case, from the question, calling $r(0,0)$ terminates and return $()$, gives $()$ is a tuple of zero bits, which satisfies $Post(0,0)$. as. $n = 0 = \sum_{i=0}^{-1} b[i] \cdot 2^i$

I've proved the base case is true.

Inductive Step: Let $m \in \mathbb{N}$, $m > 0$

Inductive Hypothesis: $\forall k \in \mathbb{N}$. $0 \le k < m$, calling $r(k, n)$ with $n < 2^k$ terminates and $Post(k, n)$

WTS: calling $r(m, n)$ with $n < 2^m$ terminates and $Post(m, n)$.

Let $n \in \mathbb{N}$. $n < 2^m$.

According to the Python function, calling $r(m, n)$ result in a recursive call of $r(m-1, n//2)$, which $r(m-1, \lfloor \frac{n}{2} \rfloor)$, and a finite number of steps, as $m \neq 0$.

Since $m \in \mathbb{N}$, $m > 0$, gives $m-1 \in \mathbb{N}$. also, from property of natural number, $m-1 < m$.

Since $n < 2^m$, divides 2 on both sides. give $\frac{n}{2} < 2^{m-1}$

Since $\lfloor \frac{n}{2} \rfloor \le \frac{n}{2}$, gives $\lfloor \frac{n}{2} \rfloor < 2^{m-1}$. ($n \% 2 = \lfloor \frac{n}{2} \rfloor$).

From I.H., gives the recursive call terminates, returning a tuple of $m-1$ bits b with $\lfloor \frac{n}{2} \rfloor = \sum_{i=0}^{m-2} b[i] \cdot 2^i$ —— fact 1.

According to quotient-remainder theorem, since $n \in \mathbb{N}$, $n \in \mathbb{R}$, and $2 \in \mathbb{R}^+$.

gives $\exists x, y \in \mathbb{R}$. s.t. $n = 2x + y$, where $0 \le y < 2$. as n can only congruent to 0 or 1 modular 2.

Since $n \equiv y \pmod 2$, gives $\lfloor \frac{n}{2} \rfloor = \frac{n-y}{2}$.

From fact 1, gives, $\frac{n-y}{2} = \lfloor \frac{n}{2} \rfloor = \sum_{i=0}^{m-2} b[i] \cdot 2^i$; which.

$$n-y = 2 \cdot \sum_{i=0}^{m-2} b[i] \cdot 2^i = \sum_{i=0}^{m-2} b[i] \cdot 2^{i+1} = \sum_{i=1}^{m-1} b[i-1] \cdot 2^i$$

Thus, $r(m-1, n//2)$ returns a tuple of $m-1$ bits b (from b_0 to b_{m-2}), with. $n-y = \sum_{i=1}^{m-1} b[i-1] \cdot 2^i$.

Since we know from the code that $r(m, n)$ will give a tuple $(n\%2) + r(m-1, n//2)$, since $y = n\%2$, gives. $(y,) + r(m-1, n//2)$, which when adding the tuples together give the index in $r(m-1, n//2)$ will increase by 1.

Also the 'y' in tuple will have the index 0, which is b_0.

Since $(y,)$ has 1 bits b and $r(m-1, n//2)$ has $m-1$ bits b, the added tuple has $1+m-1 = m$ bits b, where $b_0 = y$ and $r(m-1, n//2)$ provides $m-1$ bits b from b_1 to b_{m-1}, which to summarize,

$$n = n-y+y = \sum_{i=1}^{m-1} b[i] \cdot 2^i + y = \sum_{i=1}^{m-1} b[i] \cdot 2^i + b_0 \cdot 2^0 = \sum_{i=0}^{m-1} b[i] \cdot 2^i$$, which

the tuple is returned by $r(m, n)$.

I've proved the induction step.

Therefore, I've shown $\forall m, n \in \mathbb{N}$, $n < 2^m$, calling $r(m, n)$ terminates and Post(m, n).