황세현(2016004011)

ASSIGNMENT(PROB1)

Mathematical Induction-Problem1(compute square of a natural number)

PROGRAM

```
unsigned long long  numberSquare( unsigned long long   n){
          unsigned long long result = 0;
          unsigned long long count = 0;
          while(count<n){
                    count+=1;
                    result +=(2*count-1);
          }
           return result;
}
```

PROOF GOAL1:

n 이 입력되면 while 루프문은 n 번 반복한다

PROOF GOAL2:

입력 값이 자연수 n 이고 f(n)=0+1+3+5+…+2n-1=n*n 인 함수가 있고 프로그램이 종료될때, 결과인 리턴값은 n 의 제곱(n*n=n^2)이다

PROOF GOAL1: n 이 입력되면 while 루프문은 n 번 반복한다

Stronger Proof goal1: 어떤 자연수 m을 고려할때, m=n-count크기이고 while조건문에 닿는다, 이때 루프문은 m번 반복한다

PROOF: 반복횟수 N-COUNT 값 증명

Basis 반복조건문을 검사할 때 n-count 가 0 이라 하자 , 이런경우 n=count 가 되고 조건문의 결과는 거짓이 된다

조건문의 결과가 거짓이 되면 while 문의 내부명령은 실행되지 않기 때문에, n-count 가 0 이 되면 while 문을 0 번 반복한다

Induction: **(귀납적 가정)** n-count 가 m 일때 , while 루프문은 m 번 반복한다

n-count 가 m+1 번일때 m+1 번 반복한다는 것을 증명해보자

n-count = m+1 이게 되면 반복 조건문은 참이 된다(왜냐하면 count<m 이 되는데

n-count 가 양의 자연수이기 때문이다)

while 반복문의 반복이 1 회 종료되어 다시 while 조건문에 도달했을때 n-count=m 이다

귀납적 가정에 의해서,우리는 루프문이 m 번 더 반복했다는 것을 알 수 있다

그래서,반복문의 실행횟수는 총 m+1 번이다

Conclsion: n 이 입력으로서 주어질때, 반복문을 n 번 실행한다

PROOF GOAL2: 입력 값이 자연수 n 이고 f(n)=1+3+5+…+2n-1=n*n 인 함수가 있고 프로그램이 종료될때, 결과인 리턴값은 n 의 제곱(n*n=n^2)이다

Stronger Proof goal2: 어떤 자연 반복문이 m번 실행된 후에, 변수인 result의 값과 count는 아래와 같다

Loop invariant: (result= (0+1+3+5+7+ … +(2m-1)) = (m*m)) ^ count=m

PROOF: 귀납에 의한 증명

Basis N 이 0 일때 while 문은 0 번 반복하고 이 경우 result=0 이고 count=0 이다

f(n) = 1+3+5+…+2n-1 은 홀수의 합정리로 인해서 f(n) = n*n = n^2 이 된다

Induction: **(귀납적 가정)** 루프문이 m 번 실행된 뒤에, 불변값은 유지된다

루프문을 m+1 회 반복한후 불변값은 여전히 유지된다는 것을 증명해보자

루프문이 m 번 실행된 후의 result 값과 루프문이 m+1 번 실행된 후의 result 값을 생각해보자

count 는 비슷하기때문에 가설을 따르면 위에서 언급된 공식 또한 유지된다

(계속적인 귀납) result = (0+1 + 3 + 5 + 7 + … + (2*count-1))

루프문이 m+1 번 실행된 후에, 루프문을 1 번 실행당 count 값은 1 씩 증가하고 result 값은 count3 씩 증가한다(단, count3 값은 result 값이 증가하기전에 (count 가 1 이 증가된값의 2 배)+1 이다,이때 count2 = count+1 이라하면 count3=(2*(count+1)-1))

다시 한번 쓰면

count2=count+1

count3=2*count2-1,

sum` = sum+count3

= sum + (2*count2-1)

= sum + (2*(count+1)-1)

= sum + (2*count +1)

= 0 + 1 + 3 + 5 + 7 + … + (2* count-1) + (2*count+1)

= 0 + 1 + 3 + 5 + 7 + … + (2* m-1) + (2*m+1)

=0+ m*m + 2m+1

Conclsion: 반복문의 조건문이 거짓이 되었을때, result = 0 + 1 + … + 2m-1 이게 되면 count = n 임을 알 수 있었고 이것은 result 가 결과값으로서 나올 수 있게 끔 해주었다