HW04 알고리즘 증명 보고서

학번: 201202287
이름: 제갈수민
2016년 10월 13일

INVARIANT를 이용한 증명

증명법

프로그램이 알고리즘을 수행하면서 변하지 않는 조건을 invariant 라고 한다. Loop 를 돌 때, 시작점, 반복 중, 반복 종료 후에 이 Invariant가 변하지 않으면 해당 알고리즘은 correct 하다고 할 수 있다.

Postcondition

Find such e that

$\{e-1<=\log_2 N < e\}$

증명

```
public void binary(){
    int beforeE;
    int e = 1;
    BigInteger k = BigInteger.valueOf(2);
    BigInteger[] checkValue = new BigInteger[65];
    checkValue[e] = k;
    do{
        beforeE = e;
        e = e * 2;
        k = k.multiply(k);
        checkValue[e] = k;
    }while(k.compareTo(BigInteger.valueOf(input)) <= 0);
    while((e-beforeE)>1){
        int gap = e - beforeE;
        gap = gap/2;
        checkValue[beforeE+gap] = checkValue[e].divide(checkValue[gap]);
        if(checkValue[beforeE+gap].compareTo(BigInteger.valueOf(input))==0){
            break;
        }
        else if(checkValue[beforeE+gap].compareTo(BigInteger.valueOf(input))>0){
            e = beforeE + gap;
        }
        else{
            beforeE = beforeE + gap;
        }
    }
    System.out.println("(binary)Floor = " + beforeE);
}
```

두개의 Loop가 있으므로, 각각의 Invariant를 사용하여 증명한다.

1. do-while Loop

Invariant = $\{k=2^e \wedge 2^{e/2}\}$

1) Initialization (Loop에 진입하는 시점)

시작점에서, k = 2, e = 1이다.
따라서 $2 = 2^1 \wedge 2^0 <= n$으로 Invariant가 유지된다.

2) Maintenance (Loop반복 직전 참이라면 다음 반복 작업 수행 후도 참)
1회 반복 후, k = 4, e = 2이다.
따라서 $4 = 2^2 \wedge 2^1 <= n$으로 Invariant가 유지된다.

3) Termination (Loop 종료 되었을 때)
가령, n이 10000000000000000000이라면,
$k = 2^{64}$, e = 64
따라서 $2^{64} = 2^{64} \wedge 2^{32} <= n$으로 Invariant가 유지된다.

2. while Loop
Invariant = $\{n<2^e \wedge n>= 2^{bE}\}$
(n이 10000000000000000000이라고 가정한다면,)
1) Initialization
bE = 32, e = 64이므로 성립한다.

2) Maintenance
bE = 48, e = 64이므로 성립한다.

3) Termination
bE = 59, e = 60 이므로 성립한다.

∴따라서 해당 알고리즘은 성립한다.