
Database Laboratory
A Regression Test
Selection and Prioritization Technique
2012-11-02
이 범 용
bylee@db.konkuk.ac.kr
건국대학교 데이타베이스 연구실

차 례

BACKGROUND

- 이전 regression test의 선택 및 우선순위 기법을 확장
- 우선순위 기준
  - 테스트 케이스에 수정/삭제한 라인이 많이 포함되어 있다면 우선순위가 높음
- 테스트 케이스 선택 기준
  - 소스 코드에 수정/삭제한 라인이 있는 모든 테스트 케이스를 선택

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

- 각 테스트 케이스에서 수정된 라인이 있는 모든 테스트 케이스를 선택
- 선택된 테스트 케이스 중 수정된 라인이 많이 포함될수록 테스트 케이스의 우선순위가 높아짐
- 그 다음으로 높은 우선순위를 갖는 테스트 케이스들을 다시 비교하여 우선순위를 줌
- 수정된 라인에 대해 커버리지가 100%이면 중단
- 그리고 테스트 케이스를 최소화하기 위해 수정 알고리즘과 삭제 알고리즘을 사용

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘 (Modification algorithm)

◆ 소스 코드의 수정된 라인을 기준으로 테스트 케이스를 최소화하고 우선순위를 정하는데 사용

S.No	변수 이름	설명
1	T1	2차원 배열이고 각 테스트 케이스에 의해 포함된 소스 코드 라인의 라인 넘버을 저장하는데 사용
2	Modloc	수정된 소스 코드 라인의 총 넘버를 저장하는데 사용
3	Mod_locode	1차원 배열이고 수정된 소스 코드 라인의 넘버을 저장하는데 사용
4	Nfound	1차원 배열이고 각 테스트 케이스에 수정된 라인과 일치되는 소스 코드의 라인을 하나의 넘버에 저장하는데 사용
5	Pos	1차원 배열이고 Nfound가 정렬될 때 각 테스트 케이스의 위치를 설정하는데 사용
6	Candidate	1차원 배열이고 삭제될 수 있는 테스트 케이스의 위치가 일치하는 곳을 비트 1로 설정
7	priority	1차원 배열이고 선택된 테스트 케이스의 우선순위를 설정하는데 사용

Table 1. Variables used by “modification” algorithm

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘

◆ 단계 1 : 변수 초기화

First portion of the "modification" algorithm

1. Repeat for i=1 to number of test cases
    a. Repeat for j=1 to number of test cases
        i. Initialize array T1[i][j] to zero
2. Repeat for i=1 to number of test cases
    a. Repeat for j=1 to number of test cases
        i. Store line numbers of line of source code covered by each test case.
3. Repeat for i=1 to number of modified lines of source code
    a. Store line numbers of modified lines of source code in array mod_locode.

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘 예제

◆ 라인 번호 5, 8, 10, 20, 23, 28, 35가 수정되었다고 가정

Test case Id	A	B	C	Expected output	Execution history
T1	30	20	40	Obtuse angled triangle	8, 9, 10, 11, 12, 13
T2	30	20	40	Obtuse angled triangle	8, 9, 10, 11, 12, 13, 14, 15, 16 , 20, 21, 22
T3	30	20	40	Obtuse angled triangle	10, 11, 12, 13
T4	30	20	40	Obtuse angled triangle	10, 11, 12, 13, 14, 15, 16, 20, 21, 22
T5	30	20	40	Obtuse angled triangle	12, 13, 14, 15, 16, 20, 21, 22
T6	30	40	50	Right angled triangle	22, 23, 24, 25, 28
T7	30	20	40	Obtuse angled triangle	5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 21
T8	-	-	-	-	15, 16, 20, 21, 35
T9	30	10	15	Invalid triangle	5, 6, 7, 8, 9, 10, 11, 12, 14, 17, 18, 19, 20, 21
T10	30	10	15	Invalid triangle	18, 19, 20, 21, 35
T11	30	20	40	Obtuse angled triangle	24, 25
T12	30	20	40	Obtuse angled triangle	15, 16, 20, 21

Table 2. Test cases with execution history

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘

◆ 단계 2 : 수정 라인을 셈

Second portion of the "modification" algorithm

2. Repeat for all true cases
   a. Repeat for i=0 to number of test cases
      i. Initialize array nfound[i] to zeroes
      ii. Set pos[i] =i
   b. Repeat for i=0 to number of test cases
      i. Initialize l to zero
      ii. If candidate[i] ≠ 1 then
      Repeat for k=0 to modified lines of source code
      If t1[i][j]=mod_locode[k] then
      Increment nfound[i] by one
      Increment l by one

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘 예제

◆ 각 테스트 케이스에 포함된 소스 코드의 수정 라인을 셈

Test Cases	Line Nos. of lines matched	No. of Matches (nfound)
T1	8, 10	2
T2	8, 10, 15, 20	4
T3	10	1
T4	10, 15, 20	3
T5	15, 20	2
T6	23, 28	2
T7	5, 8, 10, 15, 20	5
T8	15, 20, 35	3
T9	5, 8, 10, 20	4
T10	20, 35	2
T11	-	0
T12	15, 20	2

Table 3. Test cases with Number of Matches Found

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘

◆ 단계 3 : 우선순위 정렬

Third portion of the "modification" algorithm

```
d.  Initialize 1 to zero
e.  Repeat for i=0 to number of test cases
        i.  Repeat for j=0 to number of test cases
              If nfound[i]>nfound[j] then
                    t=nfound[i]
                 nfound[i]=nfound[j]
                 nfound[j]=t
                 t=pos[i]
                 pos[i]=pos[j]
                 pos[j]=t
f.  Repeat for i=0 to number of test cases
        i.  If nfound[i]=1 then
              Increment count
g.  If count = 0 then
        i.  Goto end of the algorithm
h.  Initialize candidate[pos[0]] = 1
i.  Initialize priority[pos[0]]= m+1
```

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘 예제

◆ Nfound가 가장 높은 값의 기준으로 우선순위를 정렬

Test Cases	Line Nos. of lines matched	No. of Matches (nfound)	Candidate	Priority
T7	5, 8, 10, 15, 20	5	1	1
T2	8, 10, 15, 20	4	0	0
T9	5, 8, 10, 20	4	0	0
T4	10, 15, 20	3	0	0
T8	15, 20, 35	3	0	0
T1	8, 10	2	0	0
T5	15, 20	2	0	0
T6	23, 28	2	0	0
T10	20, 35	2	0	0
T12	15, 20	2	0	0
T3	10	1	0	0
T11	-	0	0	0

Table 4. Test cases in decreasing order of number of modified lines covered

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘

◆ 단계 4 : 남은 수정 라인 확인

Fourth portion of the "modification" algorithm

i. Repeat for i=0 to length of selected test cases
   i. Repeat for j=0 to modified lines of source code
      If t1[pos[0]][i] = mod[j] then
      mod[j] = 0

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘 예제

◆ Mod_locode=[5,8,10,15,20,23,28,35]-[5,8,10,15,20]=[23,28,25]

Test Cases	No. of matches (nfound)	Matches found	Candidate	Priority
T6	2	23, 28	1	2
T8	1	35	0	0
T10	1	35	0	0
T1	0	-	0	0
T2	0	-	0	0
T3	0	-	0	0
T4	0	-	0	0
T5	0	-	0	0
T9	0	-	0	0
T11	0	-	0	0
T12	0	-	0	0

Table 5. Test cases in descending order of number of matches found (iteration 2)

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘 예제

◆ Mod_locode=[23,28,25]-[23,28]=[35]

Test Cases	No. of matches (nfound)	Matches found	Candidate	Priority
T8	1	35	1	3
T10	1	35	0	0
T1	0	-	0	0
T2	0	-	0	0
T3	0	-	0	0
T4	0	-	0	0
T5	0	-	0	0
T9	0	-	0	0
T11	0	-	0	0
T12	0	-	0	0

Table 6. Test cases in descending order of number of matches found (iteration 3)

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 수정 알고리즘 예제

◆ Mod_locode=[35]-[35]=[Nil]

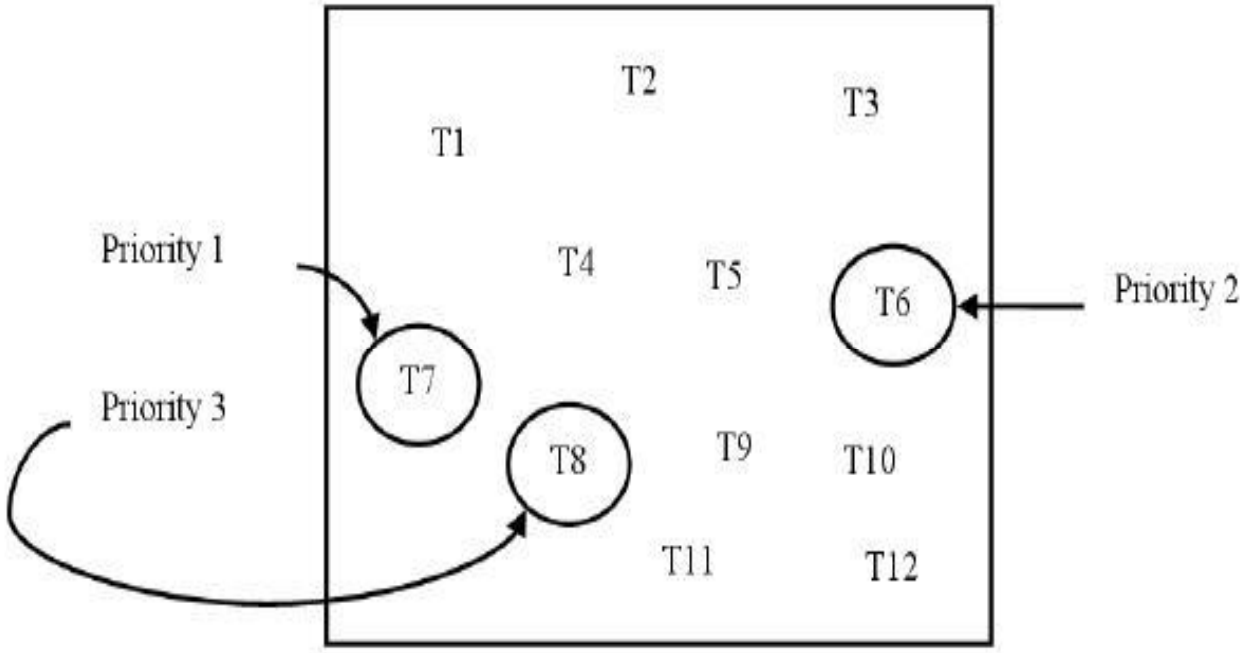

Fig 1. Test case selection and prioritization

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘(Deletion algorithm)

◆ 소스 코드에 삭제 라인을 제거하고 테스트 케이스의 실행 기록을 업데이트하는데 사용

S.No	변수 이름	설명
1	T1	2차원 배열이고 각 테스트 케이스 i에 포함된 소스 코드 라인의 넘버를 유지하는데 사용
2	Deloc	삭제된 소스 코드 라인의 총 넘버를 저장하는데 사용
3	del_locode	1차원 배열이고 삭제된 소스 코드 라인의 넘버을 저장하는데 사용
4	Count	1차원 배열이고 각 테스트 케이스에 소스 코드의 모든 라인이 일치하는 위치를 1로 설정하는데 사용
5	Match	1차원 배열이고 각 테스트 케이스에 대해 1의 총 개수를 배열에 저장하는데 사용
6	deleted	1차원 배열이고 중복 테스트 케이스 기록을 유지하고 만약 테스트 케이스 i에 일치하는 값이 있다면 삭제 배열에서 1로 설정하고 테스트 케이스에서 제거

Table 7. Variables used by “deletion” algorithm

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘

◆ 단계 1 : 변수 초기화

First portion of the "deletion" algorithm

```
1.  Repeat for i=1 to number of test cases
    a.  Repeat for j=1 to length of test case i
            i.  Repeat for l to number of deleted lines of source code
                If T1[i][j]=del_locode then
                    Repeat for k=j to length of test case i
                        T1[i][k]=T1[i][k+1]
                    Initialize T1[i][k] to zero
                    Decrement c[i] by one
```

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘 예제

◆ 라인 번호 6, 28, 36, 44, 50, 61 수정되고 12, 27, 55 삭제된다고 가정

Test case Id	Month	Day	Year	Expected out-put	Execution history
T1	6	15	1900	Friday	6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
T2	1	15	1900	Monday	46, 47, 48, 53, 54, 55, 56, 57, 61, 91
T3	1	15	2009	Thursday	50, 51, 52, 53, 54, 55, 56, 57, 61, 91
T4	1	15	2009	Thursday	56, 57, 61, 91
T5	2	15	2000	Tuesday	67, 68, 69, 91
T6	4	15	2009	Wednesday	74, 75, 91
T7	7	15	2009	Wednesday	89, 90, 91
T8	6	15	1900	Friday	3, 4, 5, 6, 7, 8, 9, 10, 11, 44
T9	1	15	1900	Monday	15, 16, 17, 18, 26, 37, 38, 39, 43, 44, 45, 46, 47, 48, 53, 54, 55
T10	2	15	2000	Tuesday	28, 29, 36, 43, 44
T11	2	30	2009	Invalid Date	34, 35, 36, 43, 44
T12	2	15	1900	Thursday	13, 14, 15, 16, 17, 18, 26, 27

Table 8. Test cases with execution history

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘

◆ 단계 2 : 라인 삭제

Second portion of the "deletion" algorithm

2. Repeat for i=1 to number of test cases
   a. Repeat for j=1 to number of test cases
      i. Initialize array t1[i][j] to zero
      ii. Initialize array count[i][j] to zero
3. Repeat for i=1 to number of test cases
   a. Initialize deleted[i] and match [i] to zero
4. Repeat for i=1 to number of test cases
   a. Initialize c[i] to number of line numbers in each test case i
   b. Repeat for j=1 to c[i]
   c. Initialize t1[i][j] to line numbers of line of source code covered by each test case

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘 예제

◆ 실행 기록에서 라인 12, 27, 55가 삭제된 테스트 케이스는 T2, T4, T12

Test case Id	Execution history
T1	6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19
T2	46, 47, 48, 53, 54, 56, 57, 61, 91
T3	50, 51, 52, 53, 54, 56, 57, 61, 91
T4	56, 57, 61, 91
T5	67, 68, 69, 91
T6	74, 75, 91
T7	89, 90, 91
T8	3, 4, 5, 6, 7, 8, 9, 10, 11, 44
T9	15, 16, 17, 18, 26, 37, 38, 39, 43, 44, 45, 46, 47, 48, 53, 54
T10	28, 29, 36, 43, 44
T11	34, 35, 36, 43, 44
T12	13, 14, 15, 16, 17, 18, 26

Table 9. Modified execution history after deleting line numbers 12, 27, 55

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘

◆ 단계 3 : 중복 테스트 케이스 식별

Third portion of the "deletion" algorithm

```
5.  Repeat for i=1 to number of test cases
    a.  Repeat for j=1 to number of test cases
            i.  If i≠j and deleted[j]≠1 then
                    Repeat for k=1 to until t1[i][k]≠0
                        Repeat for l=1 until t1[j][l]≠0
                            If t1[i][k]=t1[j][l] then
                                Initialize count [i][k]=1
    b.  Repeat for m=1 to c[i]
            i.  If count[i][m]=1 then
                    Increment match[i] with 1
    c.  If match[i]=c[i] then
            i.  Initialize deleted[i] to 1
6.  Repeat for i=1 to number of test cases
    a.  If deleted[i] =1 then
            i.  Remove test case i (as it is a redundant test case)
```

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘 예제

◆ T2, T4, T12의 실행 기록을 다른 테스트 케이스 라인과 비교

Test Case	Line Number of LOC	Found In Test Case	Redundant Y/N
T2	46	T9	Y
	47	T9	Y
	48	T9	Y
	53	T3	Y
	54	T3	Y
	56	T3	Y
	57	T3	Y
	61	T3	Y
	91	T3	Y
T4	56	T3	Y
	57	T3	Y
	61	T3	Y
	91	T3	Y
T12	13	T1	Y
	14	T1	Y
	15	T1	Y
	16	T1	Y
	17	T1	Y
	18	T1	Y
	26	T9	Y

Table 10. Redundant test cases

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘 예제

◆ T1, T3, T5, T6, T7, T8, T9, T10, T11가 남음

Test case Id	Execution history
T1	6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19
T2	46, 47, 48, 53, 54, 56, 57, 61, 91
T3	50, 51, 52, 53, 54, 56, 57, 61, 91
T5	67, 68, 69, 91
T6	74, 75, 91
T7	89, 90, 91
T8	3, 4, 5, 6, 7, 8, 9, 10, 11, 44
T9	15, 16, 17, 18, 26, 37, 38, 39, 43, 44, 45, 46, 47, 48, 53, 54
T10	28, 29, 36, 43, 44
T11	34, 35, 36, 43, 44

Table 11. Modified table after removing T2, T4, and T12

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘 예제

◆ 수정 알고리즘을 사용하여 테스트 케이스를 최소화

Test Cases	Line Nos. of lines matched (found)	No. of matches (nfound)
T10	3	28, 36, 44
T3	2	50, 61
T11	2	36, 44
T1	1	6
T8	1	44
T9	1	44
T5	0	-
T6	0	-
T7	0	-

Table 12. Test cases with modified lines

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘 예제

◆ 수정 알고리즘을 사용하여 테스트 케이스를 최소화

Test Cases	Line Nos. of lines matched (found)	No. of matches (nfound)	Candidate	Priority
T10	3	28, 36, 44	1	1
T3	2	50, 61	0	0
T11	2	36, 44	0	0
T1	1	6	0	0
T8	1	44	0	0
T9	1	44	0	0
T5	0	-	0	0
T6	0	-	0	0
T7	0	-	0	0

Table 13. Test cases in descending order of number of modified lines covered

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘 예제

◆ Mod_locode=[6,8,36,44,50,61]-[28,36,44]=[6,50,61]

Test Cases	Line Nos. of lines matched (found)	No. of matches (nfound)	Candidate	Priority
T3	2	50, 61	1	2
T11	2	36, 44	0	0
T1	1	6	0	0
T8	1	44	0	0
T9	1	44	0	0
T5	0	-	0	0
T6	0	-	0	0
T7	0	-	0	0

Table 14. Test cases in descending order of number of modified lines covered (iteration 2)

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘 예제

◆ Mod_locode=[6,50,61]-[50,61]=[6]

Test Cases	Line Nos. of lines matched (found)	No. of matches (nfound)	Candidate	Priority
T11	1	6	1	3
T1	0	-	0	0
T8	0	-	0	0
T9	0	-	0	0
T5	0	-	0	0
T6	0	-	0	0
T7	0	-	0	0

Table 15. Test cases in descending order of number of modified lines covered (iteration 3)

REGRESSION TEST SELECTION AND PRIORITIZATION TECHNIQUE

❖ 삭제 알고리즘 예제

◆ T2, T4, T12의 중복 테스트 케이스는 T10, T3, T1에 포함

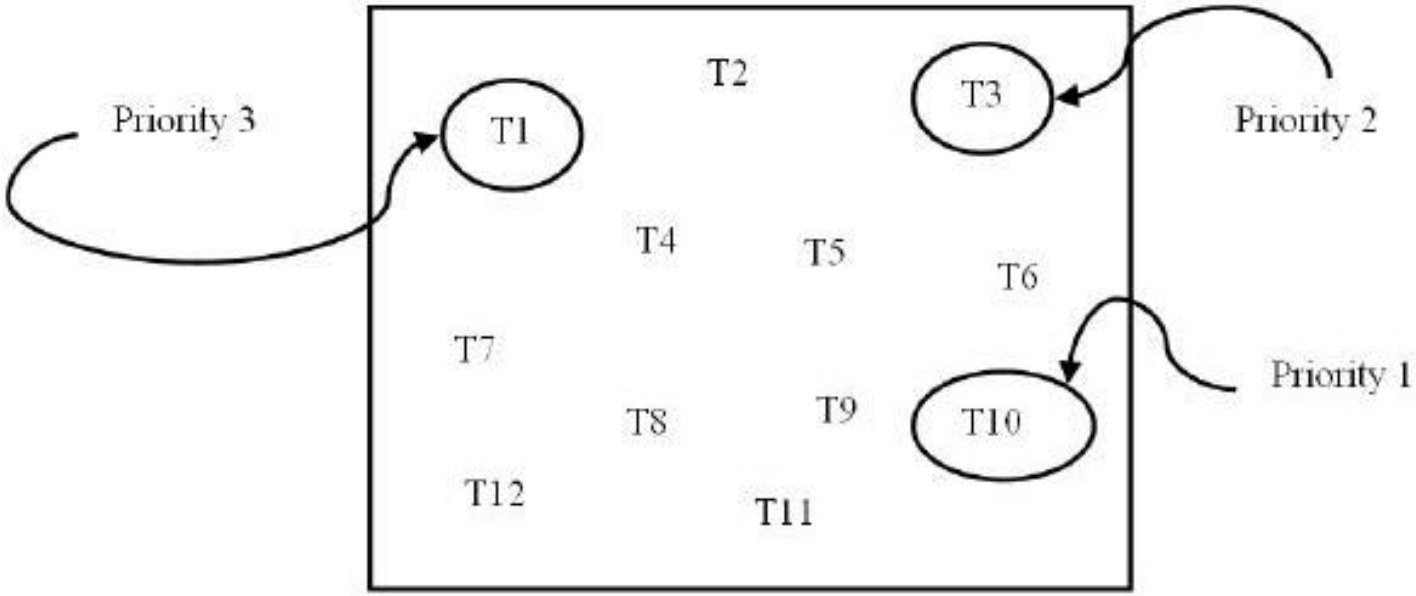

Fig 2. Test case selection and prioritization

CONCLUSIONS

- 본 연구에서 제안한 회귀 테스트 선택 및 우선순위 기법을 통해 테스트 케이스를 상당히 감소시켰음
- 이에 따라 회귀 테스트의 비용을 줄일 수 있음

참고문헌

[1]R. Malhotra, A. Kaur, and Y. Singh, “A Regression Test Selection and Prioritization Technique”, Journal of Information Processing System, Vol.6, No.2, June 2010.

감사합니다