2009년도 대한전기학회 하계학술대회 논문집 2009. 7. 14 - 17

소프트웨어 화이트박스 테스트의 교호 강도 수 자동결정 방법 연구

최형섭*, 박홍성**
강원대*, 강원대**

Automated interaction selection method for software whitebox testing

Hyeong-Seob Choi*, Hong-Seong Park**
kangwon National University*, Kangwon National University**

Abstract - S/W 화이트박스 단위 테스트 시에 교호강도의 수를 결정해야 하는데 이를 소스에서 함수의 인자가 어떤 식으로 사용되는지를 분석하여 자동으로 결정할 수 있는 방법이 있다. 소스 상에서 인자 사용의 패턴을 분석하여 특정 패턴이 되면 강도 수를 늘리고 최종적으로 교호 강도 수를 결정할 수 있게 된다. 본 논문에서는 이를 위해 조합 강도 결정 테이블을 작성하여 이를 이용한 테스트 교호 강도 수 결정 방법을 제시한다.

1. 서 론

최근의 소프트웨어 개발에 있어 소프트웨어의 규모가 점점 늘어남에 따라 테스트의 비용과 시간이 차지하는 부분이 점점 늘어나고 있다. 테스트 기법에는 여러 방법이 존재 하지만 소스코드에 대한 접근 방법에 의해 크게 두 가지 접근 방법이 있다. 하나는 테스트 작업이 원시 코드로부터 시작되는 화이트 박스 테스트와 코드 내부의 구조는 고려하지 않고 소프트웨어의 명세에 의해 결정되는 입력과 출력만을 고려하여 테스트를 하는 블랙박스 테스트가 있다.

테스트 케이스를 줄이는 노력은 많은 분야에서 적용되고 연구되어 왔다. 그 중 눈에 띄는 분야는 pair wise 를 시용한 테스트 방법으로 테스트 케이스의 파라미터 조합들을 가지고 테스트를 진행하는 것이다. 이는 각각의 테스트 케이스에서 모든 파라미터 쌍들이 전체 테스트 케이스에서 최소한도 로만 나타나도록 하는 것이 목적이다. 이를 위해 Orthogonal Array를 생성하여 테스트 케이스를 생성하는 방법들이 연구되어 왔다[2][3]. 일반적으로 모든 파라미터들의 조합을 이용하여 테스트 케이스를 생성하게 되엇을 때 보다 기하급수적으로 줄어든 테스트 케이스의 수를 얻을 수 있다. Kuhn, Wallace, Gallo[1] 은 기존의 개발된 소프트웨어의 오류를 분석한 후, 많은 오류가 2개 이상의 파라미터 간 조합에 의해 발생한다는 것을 밝혀 내었고 D.M Gohen 등이 AETG 시스템 테스트를 n-way 파라미터 조합을 통해 실시함으로써 테스트케이스를 줄이면서 좋은 커버리지를 가지는 실험[2]으로 교호작용을 이용한 테스트 케이스의 생성의 타당성을 증명하였다. pair wise를 이용하여 테스트를 진행할 수 있는 적합한 분야는 api테스트, GUI 테스트, Configuration test(S/W, H/W 호환성 테스트) 등이 있다.

그러나 기존의 OA나 n-way 파라미터 조합을 이용한 테스트 케이스 생성에 대한 연구들에 있어서 교호 작용이 몇 가지 파라미터에 의해 생기는 지에 대한 연구는 없었다. 테스트 대상의 조건에 따라 몇 개의 파라미터의 조합을 대상으로 테스트 케이스를 생성할 것인지는 알 수 없다. 이를 해결하기 위해 본 논문에서는 화이트 박스 테스트의 성격을 빌려 코드 내부에서 파라미터 입력 변수의 쓰임새를 살펴서 몇 개의 파라미터의 조합을 가지고 테스트 케이스를 생성할 것인지에 대해 설명하도록 하겠다.

2. 본 론

2.1 테스트 조합 강도

소프트웨어 단위 테스트 시에 입력으로 들어가는 인자들의 모든 테스트 케이스를 적용하기에는 시간과 비용이 들기 때문에 테스트 케이스의 수를 줄이면서 커버리지를 만족시킬 수 있는 방법이 필요하게 된다. 이를 줄이기 위해 직교배열과 같이 테스트 인자의 상호 관계에 따라 테스트의 교호강도의 수를 정하여 테스트 케이스를 줄이는 것이 중요하다. 일반적인 함수에서 교호작용의 수가 2가 되어도 문제가 없다는 것을 밝힌 기존연구들이 있다. 본 논문에서는 소스코드 상에 포함되어 있는 입력인자들의 관계를 살펴봄으로서 테스트 시 몇 개의 인자의 조합을 가지고 테스트 케이스를 생성할 것인지를 결정하는 방법에 대해 이야기 한다. 이를 위하여 소스코드를 분석하고 표1에서 제시하는 방법에 의해 교호강도 수를 결정하게 된다. 테스터는 테스트 케이스를 효과적으로 줄일 수 있게 되고 이로 인해 단위 테스트 시에 들어가는 노력과 비용이 감소될 수 있고 테스트 케이스를 작성하는 과정을 자동화함으로서 빠르게 테스트를 진행할 수 있게 된다.

<표 1> 조함 강도 결정 테이블

패턴 번호	형식	교호강도 수 계산법
1	A∧B	1 증가
2	A∨B	증가 없음
3	A←B	1 증가
4	A⊃B	1 증가
5	1∧2 (A∧B)∨C	1 증가
6	1∧3 (A∧B)←C	2 증가
7	1∧4 (A∧B)∨(A⊃B)	3 증가
8	2∧3 (A∨B)∨(A←B)	1 증가
9	2∧4 (A∨B)⊃C	3 증가
10	3∧4 (A←B)⊃C	3 증가
11	return⊃A	인자의 개수 만큼 더한다.
12	출력에 관계된 인자	증가 없음
13	A∧B∧C∧D 같은 경우와 같이 상호작용이 일어나는 인자의 수가 지속적으로 늘어날 경우	더해지는 인자의 수 만큼 교호 강도 수를 증가시킨다.

2.2 정의

테스트 인자간의 관계는 동시사용, 선택적사용, 종속관계, 포함관계(∧,∨,←,⊃)가 있을 수 있다. 함수의 파라미터에는 입력인자와 출력인자가 있게 되는데 함수의 출력에 영향을 주게 되는 입력인자의 상호관계가 중요하게 된다.

1)동시 사용: 동시 사용이란 소스코드의 한 문장(문장이란 일반적으로 ; 로 끝나는 코드)안 에서 입력으로 들어가는 인자들이 같이 사용되는 경우를 이야기 한다. 한 문자에서 같이 인자가 사용되는 것은 두 인자 사이의 관계가 함수 출력에 영향을 줄 수 있다. 표기는 ∧으로 한다.

2)선택적 사용: 선택적 사용이란 소스코드의 한 문장 안에서 입력인자가 둘 이상 사용 되었을 경우를 이야기 한다. 이때는 인자들 간에 서로 영향을 끼치지 않게 되기 때문에 이 인자들의 조합을 테스트할 필요가 없게 된다. 표기는 ∨로 한다.

3)종속관계: 테스트 인자들이 종속관계에 있다는 것은 소스코드 내에서 출력에 영향을 줄 수 있다는 것이다. 영향을 주기 위해서는 출력에 영향을 주는 변수와 관련된 곳에 사용되었을 경우이다. 소스 상에서 대괄호안의 연속된 문장에서 인자들이 사용되게 될 때 종속관계에 있다고 할 수 있다. 표기는 ←으로 한다.

4) 포함관계 : 테스트 인자 하나가 다른 인자의 사용 이 또다른 인자의 사용에 영향을 준다는 것이다. 이는 하나의 인자가 사용되고 그 다음 오는 문장에 앞의 인자 사용문장에 포함되는 경우를 이야기 한다. 표기는 ⊃으로 한다.

2.3 조합 수 결정 방법

위의 4가지의 기본적인 조합의 경우를 가지고 소스를 분석하게 된다. 소스분석을 할 때 우선 4가지의 기본 조합이 있는지를 살펴보고 있다면 그 다음으로 조합 수를 증가시키게 된다. 이 때 다시 기본적인 조합에 또 다른 조합이 있는지를 살펴야 한다. 이러한 방식으로 소스를 분석할 수 있는데 나올 수 있는 교호강도의 경우의 수는 매우 많아질 수 있다. 하지만 도면 1에 나온 것처럼 조합 수를 증가시켜야 할 경우는 1,3,4번 패턴이 관련된 경우와 11번과 같이 함수의 리턴 값에 직접 영향을 주는 경우가 있다고 할 수 있다. 재귀적으로 1,3,4번 패턴이 관련된 경우에 테이블의 패턴대로 계산해 주면된다. 이러한 패턴들을 계속적으로 소스를 분석하여 조합 수를 구할 수 있다. 그림1은 예제 코드를 보여준다.

<그림 1> 예제 소스

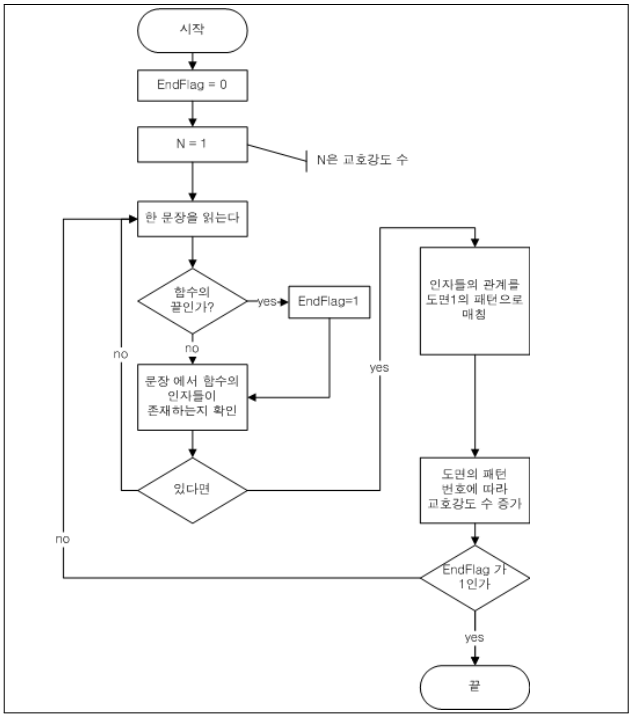
```
void test(int a,int b,int c,int d)
{
int x=0,y=0;
if ( a > 0 )
{
x = 2;
}
else
{
x = 5;
}

if ( b > 0)
{
y = 1 + x;
}

if ( c > 0)
{
if ( d > 0)
{
output(x);
}
else
{
output(10);
}
}
else
{
output(1/(y-6));
}
}
```

test 함수에서 사용된 a,b,c,d를 도면1의 표의 패턴 번호에 맞추게 되면 a와 b는 연관관계가 없는 것을 알 수 있다. c와 d의 경우에 패턴 2와 매핑이 된다. 하지만 이 경우에 c와 d는 관계가 없기 때문에 초기의 교호강도 수인 2로 결정되어서 총 4개의 테스트인자들의 모든 조합을 고려하지 않고 2개의 조합만 가지고 테스트 케이스를 생성하여 실행해도 모든 커버리지를 만족할 수 있게 된다. 이 경우 모든 조합을 사용하게 되면 총 5의 5승의 개수만큼의 테스트 케이스가 만들어지게 된다. 하지만 본 발명의 방법을 이용하여 인자의 교호강도 수인 2로 하여 테스트 케이스를 만들게 되면 30가지의 테스트 케이스만 생성되게 된다.

2.3 자동 교호 강도 결정 알고리즘

그림 2는 테스트 교호강도 수를 결정하는 방법의 순서도이다. 각 인자가 무엇인지를 인식한 후 교호강도의 수를 2로 초기화하는데 이는 최소한의 강도수가 2가 되어야 하기 때문이다. 이후 계속적으로 소스를 분석하면서 표1에 있는 패턴이 검출되는지를 살피게 된다.

<그림 2> 교호강도 수 결정 순서도



3. 결 론

본 논문에서는 교호작용을 통하여 기존의 테스트 케이스를 줄이는 방법들에 대해 살펴보고 교호작용의 문제점에 대해 살핀 후 그를 해결하기 위한 방안을 제시하였다. 테스트 케이스는 테스트 작업의 입력으로 들어가는 파라미터들을 말한다는 가정하에 이 입력들의 관계를 살핌으로써 교호작용의 강도를 자동으로 선택할 수 있을 것이다.

향후 이를 이용한 자동 테스트 케이스 생성 방안에 대해 연구하고 구현할 계획이다.

[참 고 문 헌]

[1] D. Richard Kuhn, Dolores R. Wallace, Albert M. Gallo Jr. ,"Software Fault Interactions and Implications for Software Testing," IEEE transactions on Software Engineering, vol. 30, no. 6, June 2004.

[2] David M. Cohen , Siddhartha R. Dalal, Michael L. Fredman, Gardner C. Patton, "The AETG System: An Approach to Testing Based on Combinatorial Design ," IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 23, NO. 7, JULY 1997

[3] Lei, Y and Tai, K. C., In-Parameter-Order: A Test Generating Strategy for Pairwise Test- ing, IEEE Trans. on Software Engineering, 2002, 28(1), 1-3.