

변화 추적에 의한 효율적인 점진 평가 방법

한정란
 협성대학교 경영정보학과
 e-mail:jlhan@uhs.ac.kr

An Efficient Incremental Evaluation Technique Using the Trace of Change

Junglan Han
 Dept of Management and Information System, Hyupsung University

요 약

프로그램 개발 단계에서 소요되는 비용을 최소화하기 위해 점진 평가 방법이 사용되고 있다. 점진 평가는 프로그램을 모두 다시 평가하는 대신 수정한 부분과 그 부분에 영향 받는 부분만을 다시 평가하는 방법으로 프로그램의 실행 효율성을 향상시킬 수 있는 중요 분야이다.

본 논문에서는 자바 같은 언어에서 확장된 종속차트의 종속 링크를 사용해 변수의 값이 변경되었을 때 그 부분에 영향 받는 부분을 찾아내는 변화 추적 과정을 통하여 효율적인 점진 평가를 수행할 수 있는 방법을 제시한다. 모의실험을 통해 점진 평가의 효율성을 분석한다.

1. 서론

프로그램의 생산성을 향상시키기 위해 프로그램 개발 단계에서 소요되는 비용을 최소화하려는 연구가 필요하다. 점진 평가 방법은, 프로그램을 수정할 경우, 전체 프로그램을 다시 평가하는 대신 수정한 부분과 그 부분에 의해 영향 받게 될 부분을 분석하여 수정한 부분과 그 부분에 영향 받는 부분만을 다시 평가하는 방법이다. 프로그램 개발 환경의 실행 효율성을 향상시키기 위해 점진 평가 방법을 사용하고 있다.

본 논문에서는 점진 평가를 수행하기 위해 확장된 종속차트(dependency chart)[2]를 사용하여 변화된 부분과 그 부분에 영향을 받아 변경되는 부분을 찾아내는 변화 추적 과정을 통해 자바 같은 객체 지향 언어에서 점진 평가를 수행할 수 있는 방법을 제시한다. 예제 프로그램을 통해 점진 평가를 효율적으로 수행하는 방법을 제시하고 모의실험을 통해 점진 평가의 효율성을 분석한다.

2. 변화 추적 과정

점진 평가를 수행하려면 변경된 변수의 변화를 추적하는 변화 추적 과정을 거치면서 이 변화에 영향 받는 필수 변수들만을 평가하는 과정을 거치게 된다.

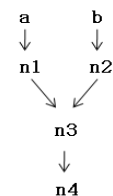
어떤 변수 값이 변경되면 그 변수에 의존하는 변수도 변하게 되는데 이러한 값의 변화를 이끄는 후속자(successor)로 파악시키면서 점진 평가를 수행한다. 그러나 기존의 방법과는 달리 종속성을 나타내기 위한 유향(directed) 그래프인 확장된 종속차트[2]를 사용하여 종속성을 구성할 때 변수의 값을 중심으로 종속성을 나타내면 값의 변화를 파악시키는 과정이 별도로 필요하지 않고 종속성을 나타내는 포인터를 통해 값의 변화를 추적하여 영향 받는 변수들을 바로 찾아낼 수 있다.

(그림 1)에 예제 프로그램이 나와 있고 (그림 2)에 종속차트에서 변수 값의 종속성을 나타내는 종속 링크가 표시되어 있다.

```
class Trace {
  int a = 5, b = 20;    --- ①
  void methodTR() {
    int n1, n2, n3, n4;
    n1 = a + 10;      --- ②
```

```
    n2 = b - 10;      --- ③
    n3 = n1 + n2;    --- ④
    if (n3 > 20)
      n4 = n3 / 2;    --- ⑤
    else n4 = n3 * 2; --- ⑥
    System.out.println("n4 = " + n4);
  }
  public static void main(String args[]) {
    Trace obj;
    obj = new Trace();
    obj.methodTR();    --- ⑦
  }
```

(그림 1) 자바 프로그램



(그림 2) 종속 링크

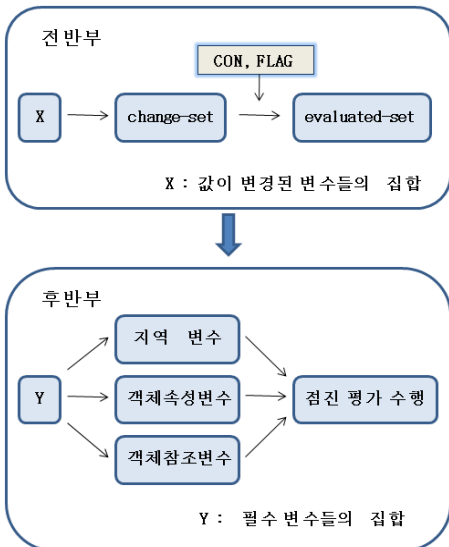
(그림 1)의 프로그램을 실행하면 $n4 = 12$ 라고 출력되는데 만일 a나 b 값을 변경했을 때 $n4$ 값도 바뀌게 된다. 예로서 ①번 명령문에서 a값을 0으로 변경할 경우 $n4$ 값은 40으로 바뀌게 된다. $n4$ 가 바뀌는 변화 과정을 (그림 2)의 종속 링크를 통해 알 수 있다. (그림 2)에서 a를 찾아서 DFS(Depth First Search)로 링크를 따라가 방문되는 각 변수가 a의 값이 변할 때 변경되는 변수들이다. 즉 a가 변경되면 ②번 명령문에서 $n1$ 이 변경되고 ④번 명령문에서 $n3$ 변수가 변경되고 최종적으로 ⑤번이나 ⑥번 명령문에서 $n4$ 까지 값이 바뀌게 됨을 알 수 있다.

3. 점진 평가 방법

프로그램을 올바르게 실행하여 정확한 결과를 표시하려면 프로그램에서 사용된 변수들의 값의 변화를 추적하여 계산하고 그 값을 출력하는 과정을 거치게 된다. 점진 평가에서는 변수 값을 수정하는 명령문이 나오면 종속차트의

종속링크를 통해 그 변수 값에 영향 받아 변하게 될 변수들을 추적하여 영향 받는 명령문들을 다시 평가한다.

점진 평가를 수행하는 과정은 (그림 3)에서처럼 두 단계로 진행된다. 첫 번째 단계인 전반부에서는 점진 평가에서 중요한 변수가 되는 필수 속성을 찾아내는 과정이다. 먼저 값이 변경된 변수에 대해 그 변수 값이 변함에 따라 값이 변경될 변수들을 찾아야 한다. (그림 2)에서처럼 종속 링크를 참조하여 값이 변경될 변수들을 찾아낸다. 이러한 변수들로 change-set[2]을 구성한다. 조건문이나 반복문에 속한 명령문의 경우 제어식의 값에 따라 명령문의 실행 여부가 결정되고 만일 change-set에 속한 변수 중 이러한 명령문에 소속되어 있을 경우 변경될 변수이긴 하지만 실제로 실행될 때 수행되지 않는 변수가 될 수 있다. 실제로 수행될 필수 변수인지 판별하기 위해 종속 차트에 CON과 FLAG 값을 지정하여 제어문에 속한 명령문의 평가 여부를 미리 결정하게 된다. CON과 FLAG 값에 따라 실행될 필요가 있는 필수 변수만을 찾아내게 된다. 이러한 필수 변수로 evaluated-set을 구성한다. 즉, change-set에 있는 모든 변수에 대해 평가될 필요성이 있는지를 적절하게 검사해서 evaluated-set을 구성하게 된다[1]. 각 변수의 CON 필드와 FLAG 필드를 조사하여 CON 필드에 있는 속성 값에 따라서 evaluated-set이 구성되고 evaluated-set에 속한 필수 변수만 평가하게 된다.



(그림 3) 점진 평가 과정

두 번째 단계인 후반부에서는 객체를 처리하기 위해 새롭게 확장된 부분으로 evaluated-set에 속한 필수변수가 지역변수인 경우 종속차트를 참조하여 점진 평가를 수행한다. 만일 evaluated-set에 속한 필수변수가 객체 속성변수인 경우 종속 차트에서 해당 속성을 갖는 객체를 찾아 그 값을 변경해 주어야 한다. (그림 1)에서 a나 b를 변경한 경우 a나 b를 객체 속성 변수로 갖는 obj의 객체속성 변수를 변경해주어야 한다. 객체이름을 변경한 경우 해당 객체의 속성변수가 변경되도록 점진 평가를 수행해야 한다.

4. 점진 평가의 효율성 분석

자바로 작성된 네 가지 유형의 프로그램에 대해 모의 실험을 통해 점진 평가의 효율성을 분석한다.

성적을 계산하는 프로그램(P1), 행렬의 값을 비교하는 프로그램(P2), 이자 계산 프로그램(P3), 선형리스트 구현 프로그램(P4)에 대해 실험하고 네 가지 명령문인 배정문, 조건문, 반복문, 메소드 호출문에 대해 변수 값을 변경하였고 그 변경으로 점진 평가하는 명령문의 비율을 비교하여 변경하는 명령문유형의 따라 점진 평가의 효율성을 분석한다. 전체 프로그램을 평가하는 비율은 100%이고 점진

평가했을 때 평가되는 비율에 따라 점진 평가의 효율성을 분석할 수 있다.

표 1에서 P1과 P2 프로그램의 경우 반복문이 실행 결과를 얻는 핵심구문이 되고 자료 수에 비례하여 반복적으로 실행하여 결과를 얻으므로 소량의 자료에 대한 평균 점진 실행비율은 각각 27%, 38% 이다. 반면 대량의 자료에 대한 평균 점진 실행 비율은 70%, 96%로 자료 수에 비례하여 점진 평가 비율이 높아짐을 알 수 있다. P2의 비율이 더 높은 이유는 P1보다 값을 비교하는 조건문에 더 큰 영향을 받기 때문이다.

<표 1> 점진 평가 결과 (단위:%)

프로그램 유형	배정문	조건문	반복문	호출문	평균
P1	28	13	32	36	27
P1	90	5	91	92	70
P2	35	33	40	44	38
P2	94	95	97	97	96
P3	7	8	79	73	42
P3	10	15	90	85	50
P4	11	19	91	84	51
P4	7	11	92	85	49
평균	35	25	77	75	53

P3과 P4 프로그램의 경우도 핵심 구문이 반복문이지만 P1과 P2와는 달리 배정문에 있는 변수를 수정하더라도 반복문에 있는 변수에 영향을 주지 않는 경우도 있으므로 점진 실행 비율이 자료 수에 비례하여 영향을 미치지 않음을 알 수 있다.

실험을 통해 프로그램의 명령문 변경유형 중 반복문의 변경은 점진 평가의 효율성을 방해하는 요소가 되고 다른 명령문 유형을 변경했을 때 반복문에 영향을 미치는 경우 점진 실행의 효율성에 나쁜 영향을 주게 된다.

결론적으로 배정문의 변경이 반복문까지 영향을 주는 경우도 있지만 영향을 주지 않는 경우도 있으므로 53%의 점진 실행 효율성이 나타난다.

5. 결론

본 논문에서는 변수의 값의 변화에 영향 받아 변화가 발생하는 부분을 추적하기 위해 확장된 종속 차트(dependency chart)를 사용하여 객체 지향언어인 자바 같은 언어에서 점진 평가를 수행할 수 있는 효율적인 점진 평가 방법을 제시하였다.

네 가지 프로그램 유형에 대해서 점진평가의 효율성을 평가하는데 네 가지 명령문인 배정문, 조건문, 반복문, 메소드 호출문에 대해 변경한 후 다시 평가될 명령문 수를 중심으로 점진 평가의 효율성을 검토해 본 결과 이들 변경 유형들이 반복문에 영향을 줄 경우, 점진 평가 비율이 79%~97%로 다소 효율성이 떨어지지만 네 가지 유형을 평균적으로 고려할 때 53%의 실행 효율성을 나타내어 프로그램 전체를 평가하는 것 보다 훨씬 효율적이라는 사실을 알 수 있다.

참고문헌

- [1] 한정란 “작용 식 기반 점진 해석” Ph. D Thesis 이화여대 1999.
- [2] 한정란 “확장된 종속차트를 사용한 효율적인 점진 평가 방법” 인터넷정보학회 논문지, 제10권 2호, pp.75~84, 2009.
- [3] 한정란 “객체 지향 언어를 위한 의미 명세” 인터넷정보학회 논문지, 제8권 5호, pp.35~43, 2007.
- [4] Han junglan “Action Equations for Object Oriented Programming Language” In Proceeding of ICUT pp.101~108, 2007.