<Homework 4> Solve the 5 Problems Chapter 7

20201635 컴퓨터공학과 전찬

#1. Assume the following rules of associativity and precedence for expressions and show the order of evaluation of the following expressions.

1. (((a \* b)1 – 1)2 + c)3

2. (((a \* (b - 1)1)2 / c)3 mod d)4

3. (((a – b)1 / c)5 & (((d \* e)2 / a)3 - 3)4)6

4. ((-a)1 or ((c = d)2 and e)3)4

5. (((a > b)1 xor c)3 or (d <= 17)2)4

6. (–(a + b)1)2

#2. Show the order of evaluation of the following expressions with no precedence rule, right to left associative rule.

1. (a \* (b – (1 + c)1)2)3

2. (a \* ((b - 1)1 / (c mod d)2)3)4

3. ((a - b)5 / c (& (d \* (e / (a – 3)1)2)3)4)6

4. (–(a or (c = (d and e)1)2)3)4

5. (a > (b xor (c or (d <= 17)1)2)3)4

6. (–(a + b)1)2

#3. Let the function fun and its usage be defined, what are the values of sum1 and sum2

1. if the operands in the expressions are evaluated left to right?

left to right 이기 때문에, i의 값은 sum1을 계산할 때에 변하지 않으며(10 이며), j의 값은 sum2를 계산할 때에 변할 것(14가 될 것)임을 파악할 수 있다.

따라서 sum1 = 5 + 3\*14 – 1 = 46, sum2 = 3\*14 – 1 + 14/2 = 48 이다.

2. if the operands in the expressions are evaluated right to left?

위와 반대로 i가 변하며, j가 변하지 않기 때문에

sum1 = 7 + 3\*14 – 1 = 48, sum2 = 3\*14 – 1 + 5 = 46 이 된다.

#4. Consider the C program, what’s the value of x after the assignment statement in main

1. operands are evaluated left to right

x = x + fun(&x); 에 대해, left to right 이므로, x = 3 + 4 = 9 가 될 것임을 파악할 수 있다.

2. operands are evaluated right to left.

x = x + fun(&x); 에 대해, right to left 이므로, x = 8 + 4 = 12 가 될 것임을 파악할 수 있다. fun(&x)가 먼저 실행되기 때문에, x의 값을 3 + 5 = 8 로 바꾸기 때문이다.

#5. Consider the C program and function fun, explain the results.

우선 위 프로그램을 실행시켜보면,

With the function call on the right, b is: 40

With the function call on the left, b is: 40

와 같은 결과가 나온다. 따라서 두 경우에서 동일하게 b가 40이 된다는 것인데, 이를 분석해보면 아래와 같다.

- b = a + fun(); 에 대해 fun() 이 먼저 실행되며, fun()의 return type이 정의되지 않았기 때문에, + C 언어에서 자동적으로 fun()의 return value = a로 설정한다. 따라서 b = a + 20 형태이다. 더 나아가, fun() 에서 전역 변수 a의 값을 20으로 update했기 때문에 b = 20 + 20 이 된다.

- b = fun() + a; 에 대해 위와 동일하게 fun()이 먼저 실행되며 return value = a가 되며 b = 20 + a, b = 20 + 20 이 되어 b = 40이 된다. 따라서 left call, right call 모두 b의 값은 40 이 된다.

위를 통해서 파악할 수 있는 점은, + binary operator 에 대해서 left or right에 function call이 존재한다면, 해당 function 을 먼저 수행함을 파악할 수 있다.