

Theory of Multi-Paradigm Programming Languages

Lecture 1-1

Chapter 1 Principles of Programming Languages

Overview

Prof. Moonkun Lee, Ph.D.

Division of Computer Science and Engineering
Jeonbuk National University, Republic of Korea

Contents

- 제1장 프로그래밍언어의 원리
 - 1.1 프로그래밍언어의 기본적인 쟁점
 - 1.2 프로그래밍언어 정의
 - 1.3 프로그래밍언어의 표현 수준
(Level of Expression)
 - 1.4 프로그래밍언어의 패러다임
 - 1.5 결어

1.1

프로그래밍언어의 기본적인 쟁점

절차중심언어

[L#]	코드
1	int factorial(int n)
2	{
3	int n_f = 1;
4	
5	if (n<0)
6	{
7	printf("No factorial for negative integer! \n");
8	exit();
9	}
10	while (n>0)
11	{
12	n_f = n_f * n;
13	n = n - 1;
14	}
15	return n_f;
16	}

<프로그램 1-1> C: Factorial 함수

[L#]	코드	실행 경로			
		1	2	3	4
1	int factorial(int n)	n=3			
2	{				
3	int n_f = 1;	n_f=1			
4					
5	if (n<0)	(3<0)?			
6	{				
7	printf("No factorial for negative int!\n");				
8	exit();				
9	}				
10	while (n>0)	(3>0)?	(2>0)?	(1>0)?	(0>0)?
11	{				
12	n_f = n_f * n;	n_f=3	n_f=6	n_f=6	
13	n = n - 1;	n=2	n=1	n=0	
14	}				
15	return n_f;	n_f=6			
16	}				

<프로그램 1-2> C: Factorial(3) 실행 단계

[L#]	코드	프로그램 상태(n, n_f)와 전이			
		1	2	3	4
1	int factorial(int n)	$S_0 = (3,0)$			
2	{				
3	int n_f = 1;	$S_1 = (3,1)$			
4					
5	if (n<0)				
6	{				
7	printf("No ... \n");				
8	exit();				
9	}				
10	while (n>0)				
11	{				
12	n_f = n_f * n;	$S_2 = (3,3)$	$S_4 = (2,6)$	$S_6 = (1,6)$	
13	n = n - 1;	$S_3 = (2,3)$	$S_5 = (1,6)$	$S_7 = (0,6)$	
14	}				
15	return n_f;	$S_f = (0,6)$			
16	}				

<프로그램 1-3> C: Factorial(3) 실행 단계

비절차중심언어: 함수언어

[L#]	코드
1	int factorial(int n)
2	{
3	if (n<0)
4	{
5	printf("No factorial for negative integer! \n");
6	exit();
7	}
8	else
9	return fact(n);
10	}
11	
12	int fact (int n)
13	{
14	if (n==1 n==0)
15	return 1;
16	else
17	return n*fact(n-1);
18	}

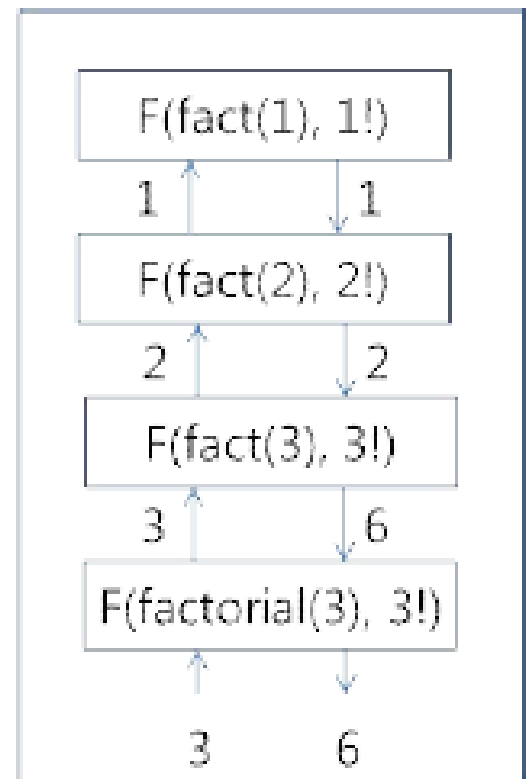
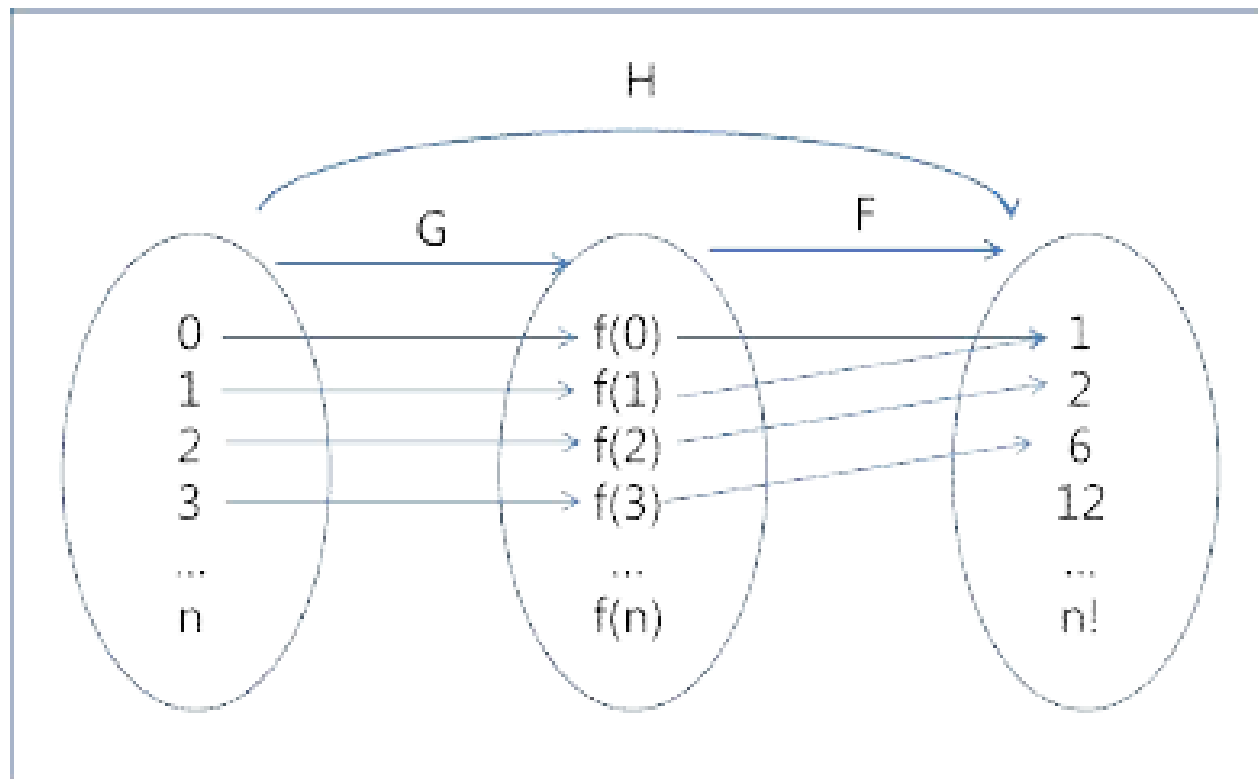
<프로그램 1-4> C: Factorial/Fact 재귀함수

[L#]	코드	함수 호출			
		factorial(3)	fact(3)	fact(2)	fact(1)
1	int factorial(int n)	n = 3			
2	{				
3	if (n<0)	(3 < 0)?			
4	{				
5	printf("No ...! \n");				
6	exit();				
7	}				
8	else				
9	return fact(n);	fact(3)			
10	}				
11					
12	int fact (int n)		n = 3	n = 2	n = 1
13	{				
14	if (n==1 n==0)		(3==1 3==0)?	(2==1 2==0)?	(1==1 1==0)?
15	return 1;				1
16	else				
17	return n*fact(n-1);		3 * fact(2)	2 * fact(1)	
18	}				

<프로그램 1-5> C: Factorial (3) /Fact (3) 재귀함수 실행

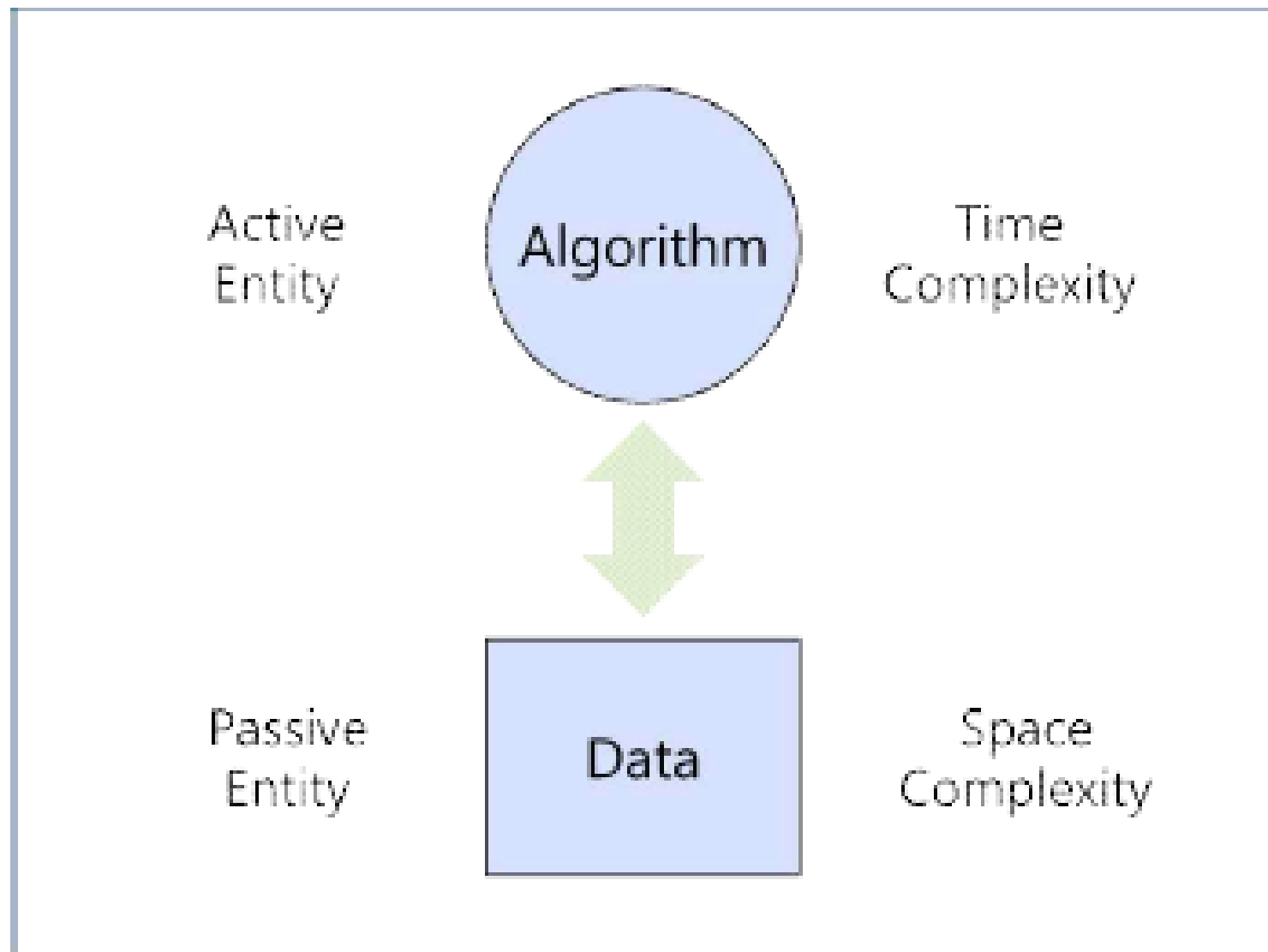
[L#]	코드	함수 영역			
		F(factorial(3),6)	F(fact(3),6)	F(fact(2),2)	F(fact(1),1)
1	int factorial(int n)	n = 3			
2	{				
3	if (n<0)				
4	{				
5	printf("No ...! \n");				
6	exit();				
7	}				
8	else				
9	return fact(n);	6			
10	}				
11					
12	int fact (int n)		n = 3	n = 2	n = 1
13	{				
14	if (n==1 n==0)				
15	return 1;				1
16	else				
17	return n* fact (n-1);		6	2	
18	}				

<프로그램 1-5-1> C: Factorial(3)/Fact(3) 재귀함수 영역 (Domain)

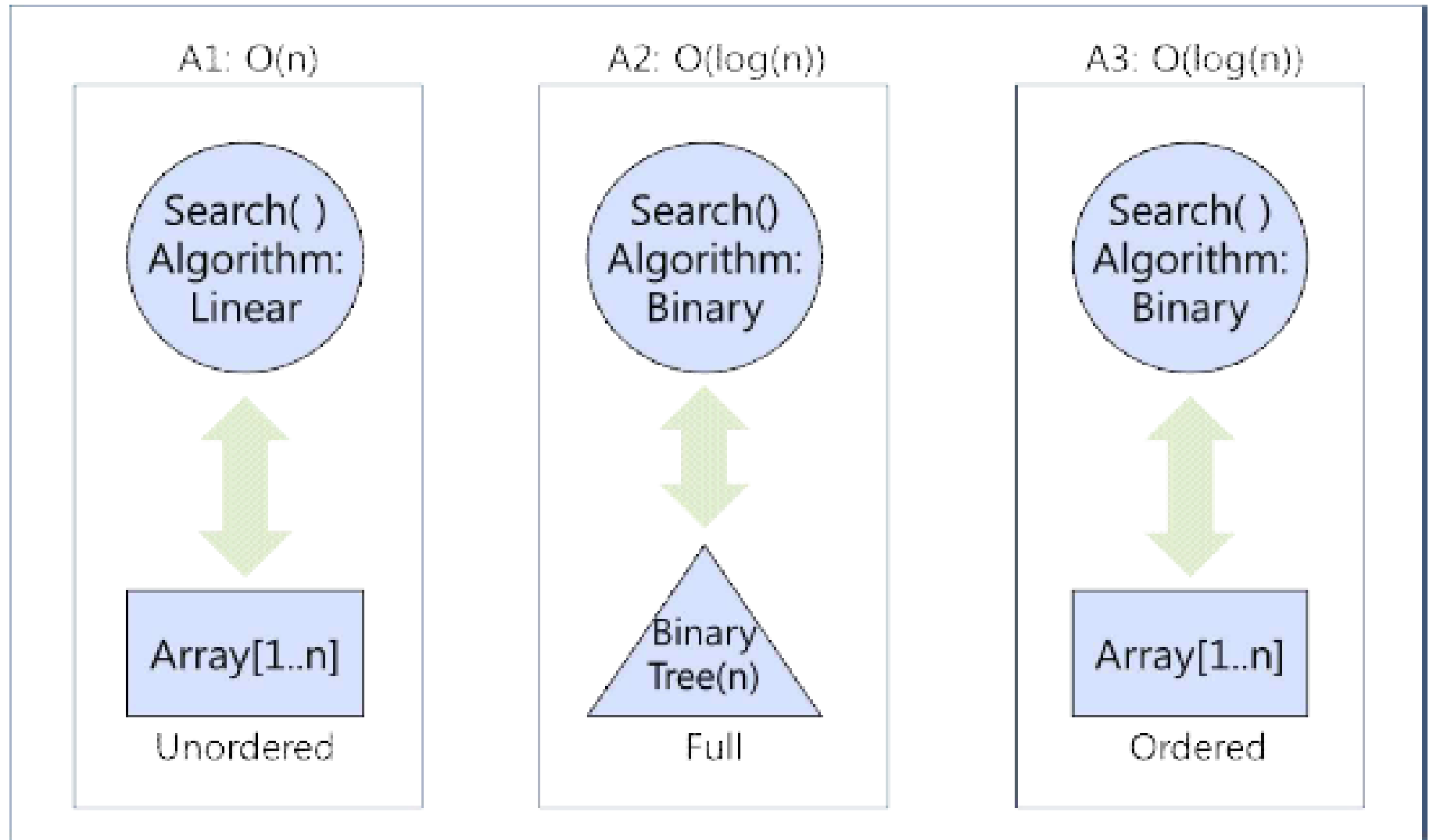


<그림 1-1> Factorial/Fact 함수 영역 및 부분 경우들(Instance)

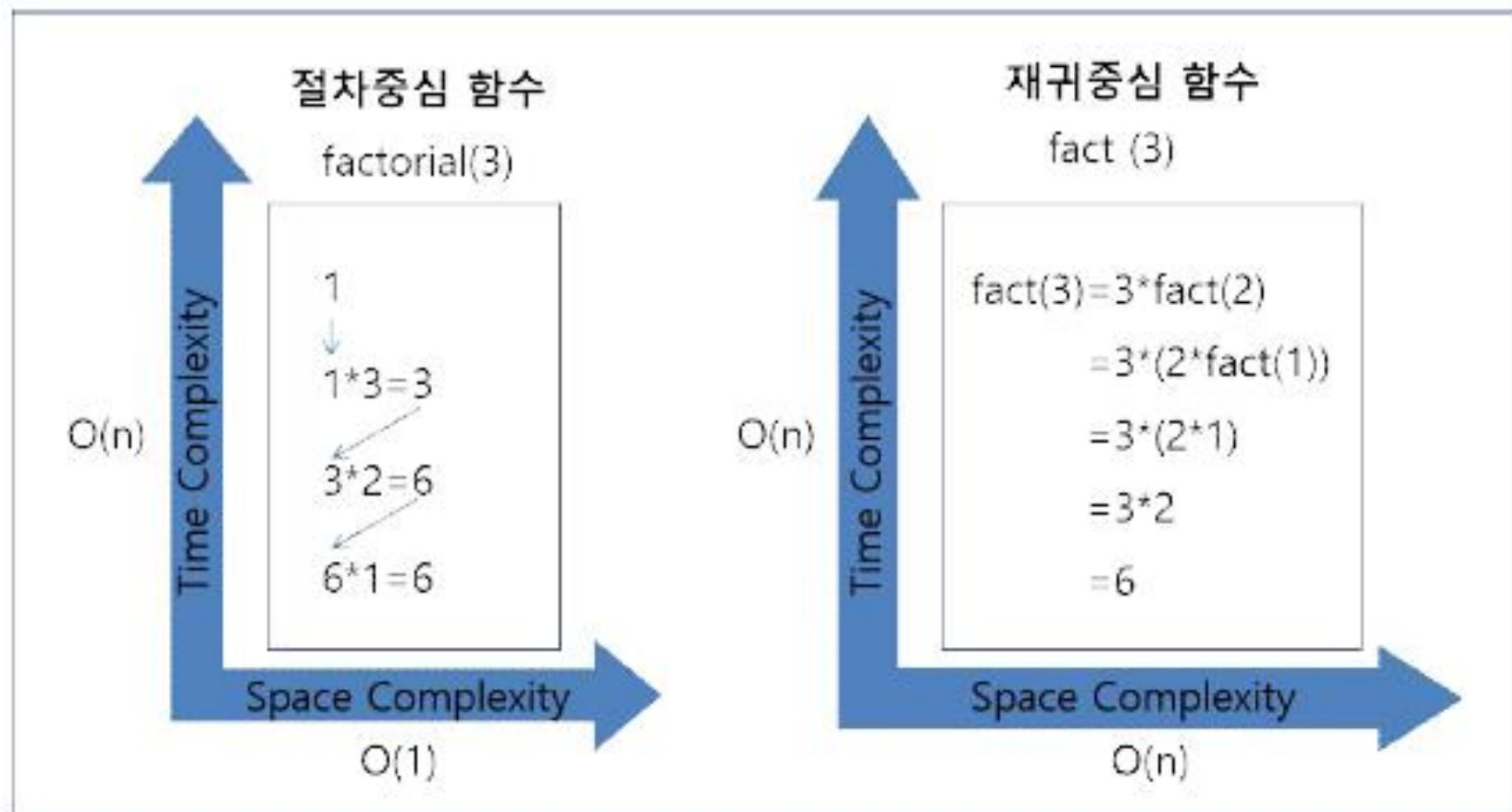
시. 공간복잡도 (TIME/SPACE COMPLEXITY)



<그림 1-2> Algorithm과 Data의 관계



<그림 1-3> Time Complexity: Search Algorithm

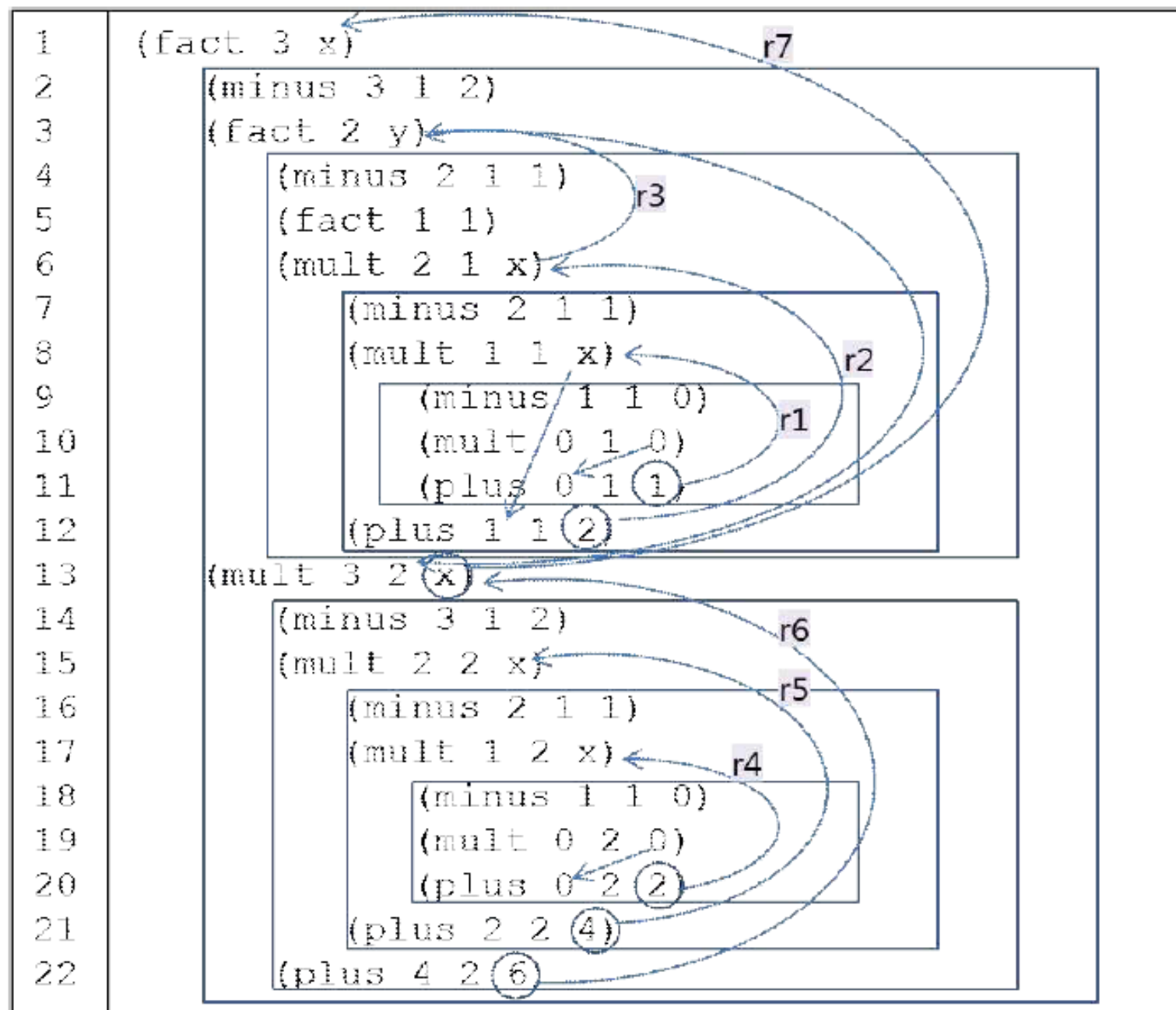


<그림 1-4> 절차 함수와 재귀함수의 시공간복잡도

비절차중심언어: 논리언어

[L#]	코드
1	(infer (mult 0 y 0))
2	(infer (mult X Y Z) from
3	(minus X 1 V)
4	(mult V Y W)
5	(plus W Y Z))
6	(infer (fact 0 1))
7	(infer (fact 1 1))
8	(infer (fact n x) from
9	(minus n 1 m)
10	(fact m y)
11	(mult n y x))
12	(infer? (fact 3 x) (print x))

<프로그램 1-6> Prolog: Factorial Axion 및 Induction Rule Clause들

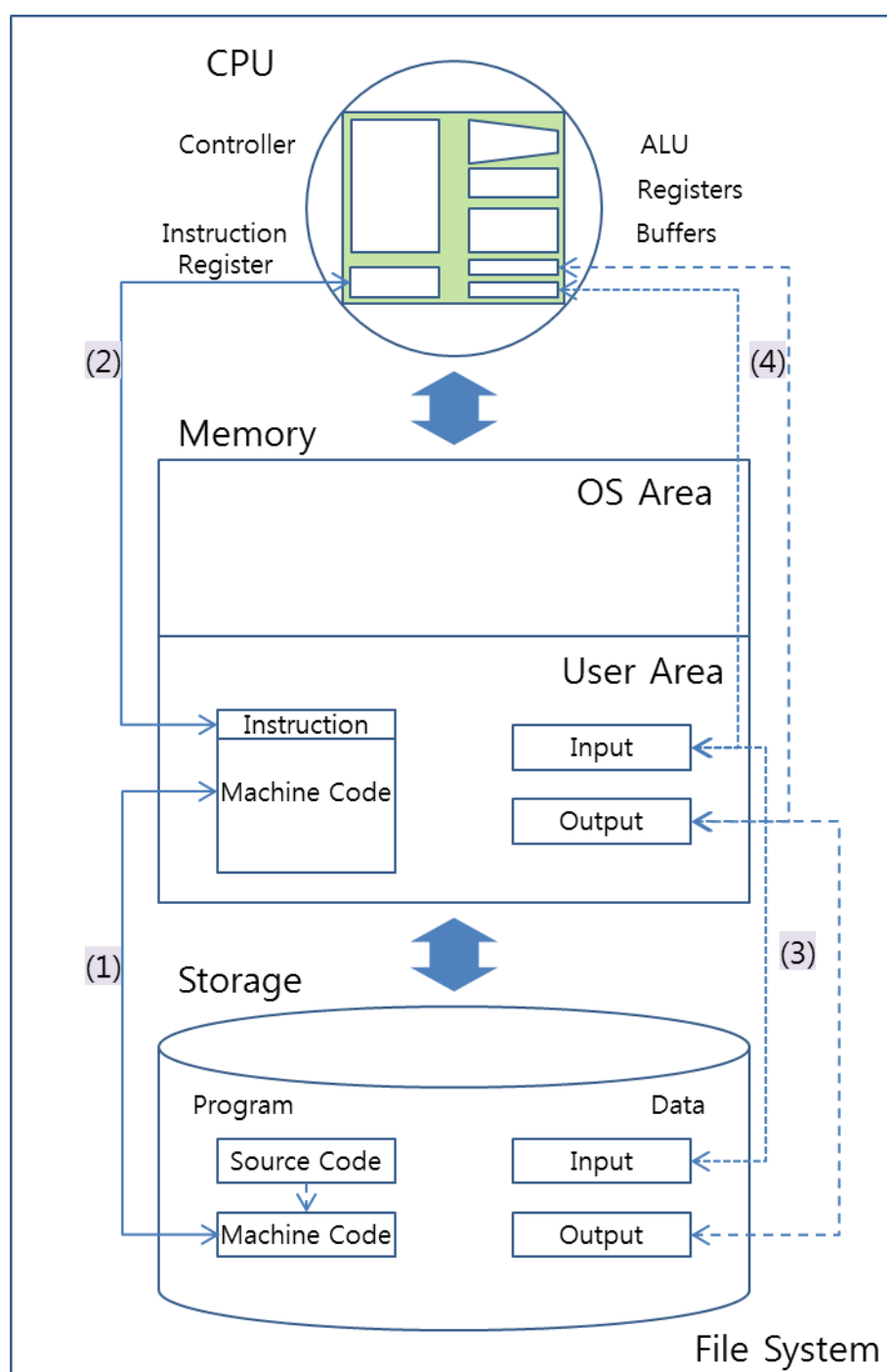


<그림 1-5> Prolog: (fact 3 x) 실행

[L#]	코드
12	(infer? (fact 3 x) (print x))
13	6
14	<i>Satisfied</i>

<프로그램 1-7> Prolog, Factorial Query와 출력

프로그램 실행 체계 및 방법



1.2

프로그래밍언어 정의

Main Concepts/Notations

- 1.2.1 프로그래밍언어 (Programming Languages)
- 1.2.2 Computer Program
- 1.2.3 알고리즘(Algorithm)
- 1.2.4 구문론(Syntax)
- 1.2.5 구문구성도 (Syntax/Parse Tree)
- 1.2.6 정형언어 (Formal Language)
- 1.2.7 정형언어 (Formal Language) 유형
- 1.2.8 유도트리 (Derivation Tree)
- 1.2.9 모호성 (Ambiguity)
- 1.2.10 의미론(Semantics)
- 1.2.11 일치(Satisfaction)

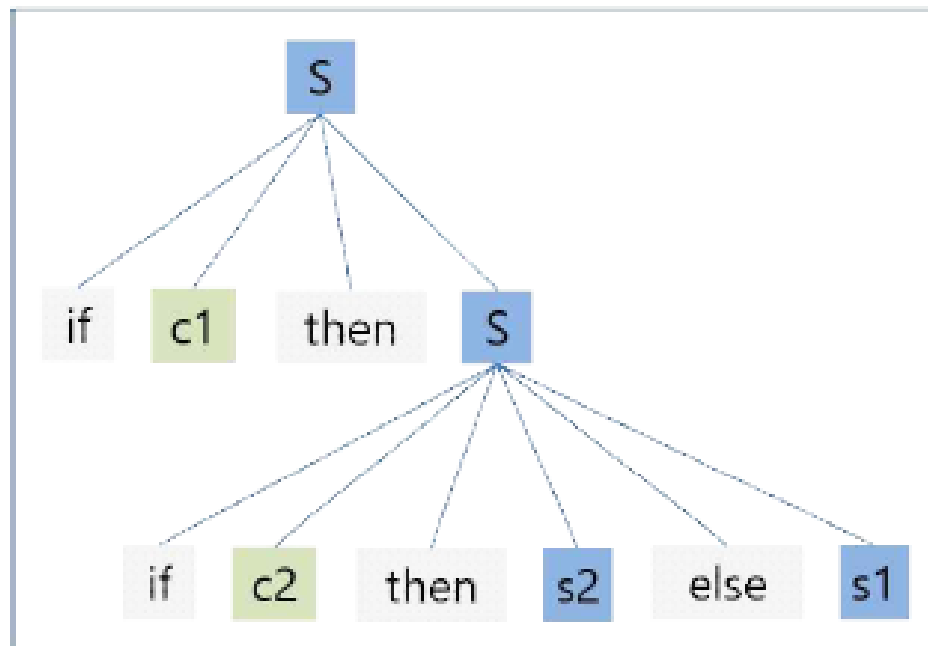
모호성 (AMBIGUITY)

$S \rightarrow \text{if } C \text{ then } S \text{ else } S$

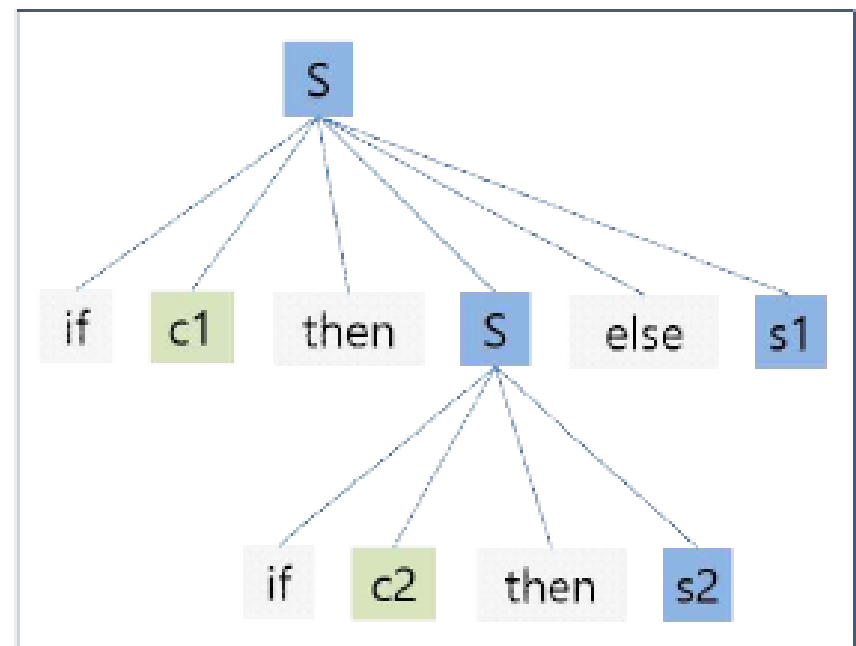
$S \rightarrow \text{if } C \text{ then } S$

| if c1 then if c2 then s2 else s1

Parse Tree A



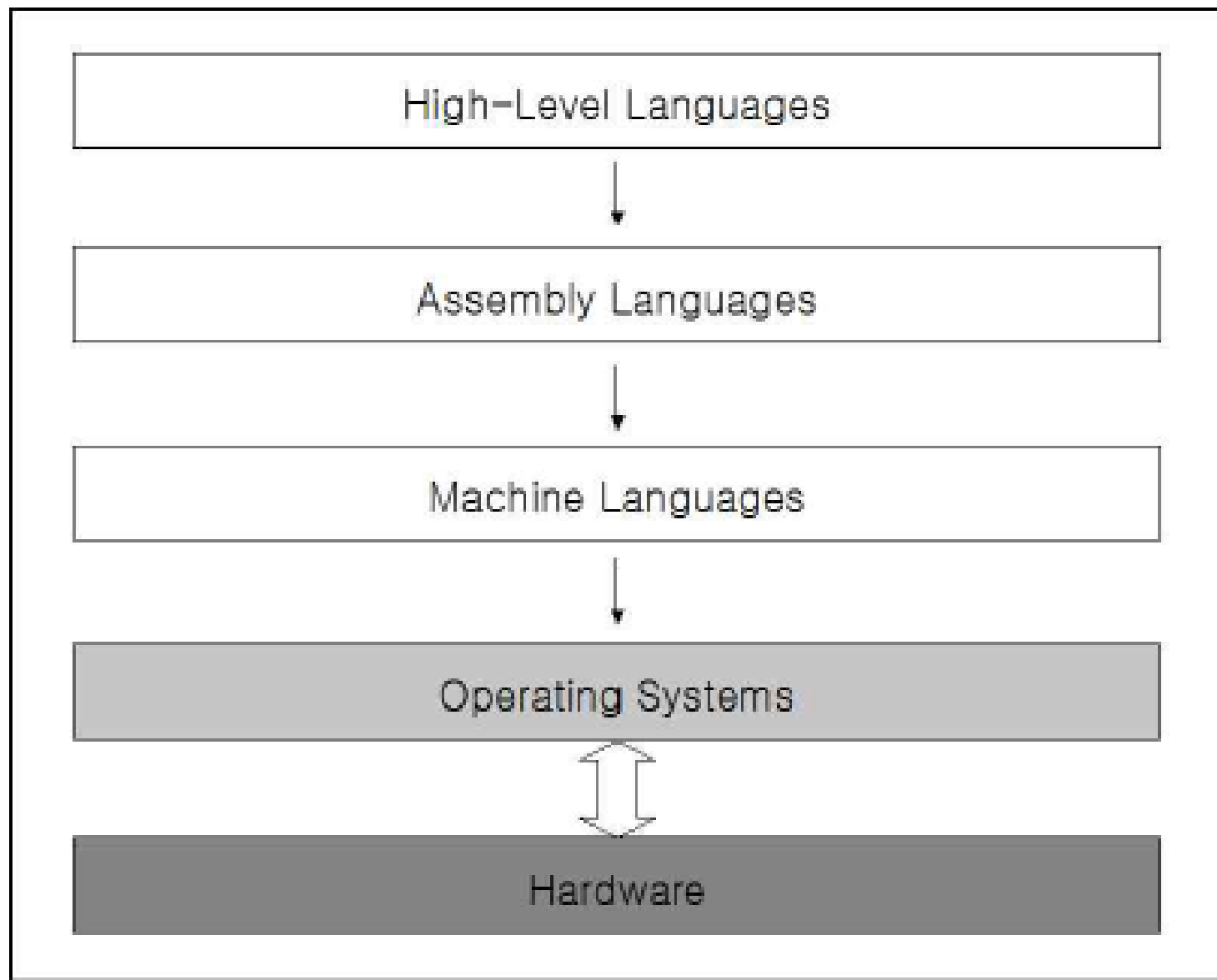
Parse Tree B



<그림 1-14> if-문 문법에 의해서 생성된 두 개의 Parse Tree A, B

1.3

프로그래밍언어의 표현 수준



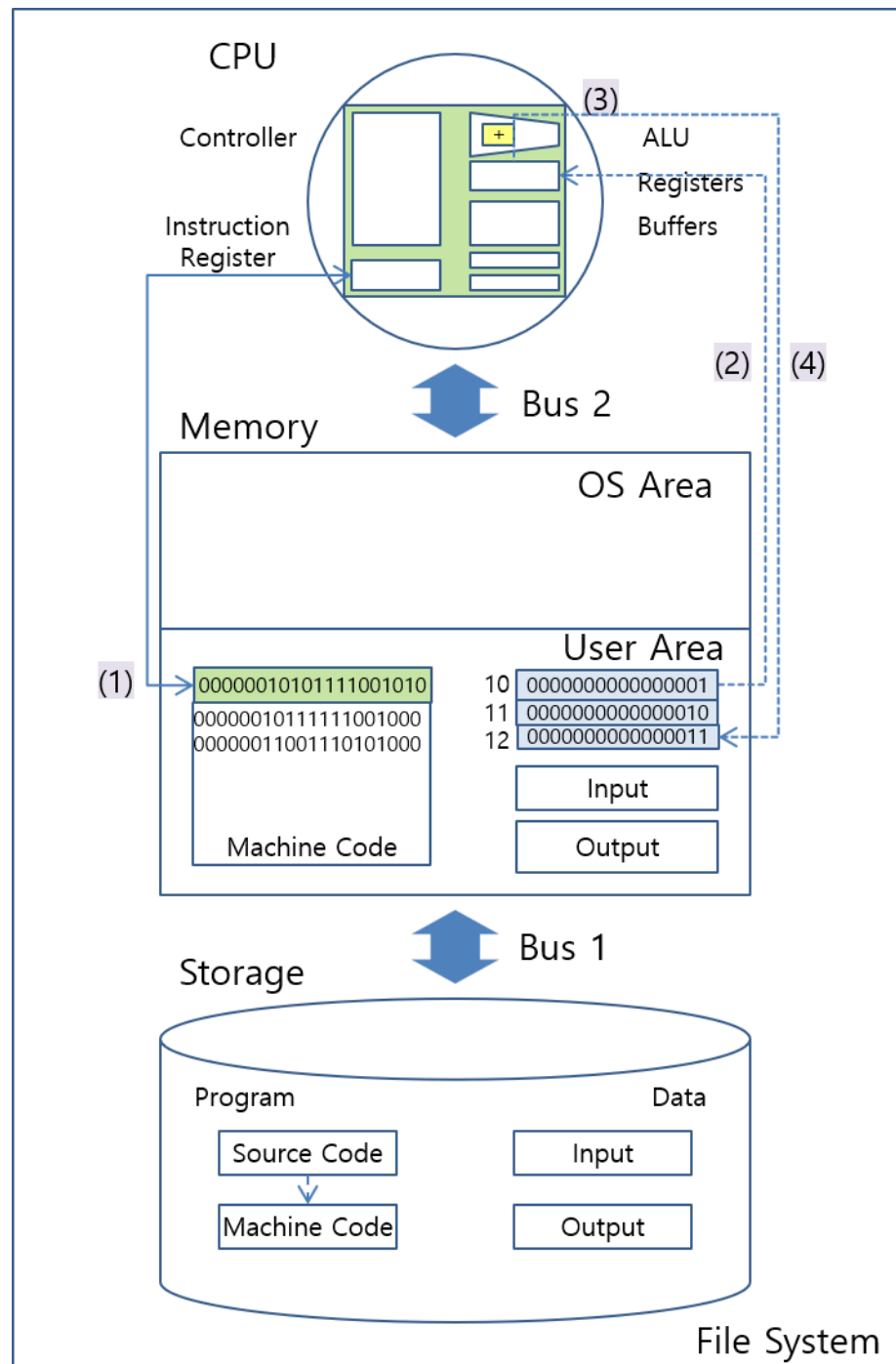
<그림 1-15> 프로그래밍언어의 계층구조

기계언어 (MACHINE LANGUAGE)

[L#]	Machine 코드
1	00000010101111001010
2	00000010111111001000
3	00000011001110101000

<프로그램 1-8> 기계언어 예제

"memory의 주소값 10과 11에
있는 수를 더해
그 결과를 주소값 12에 저장



어셈블리어언어 (ASSEMBLY LANGUAGE)

[L#]	Machine 코드	어셈블리 코드
1	00000010101111001010	LOAD M[10] R1;
2	00000010111111001000	LOAD M[11] R2;
3	00000011001110101000	ADD R1, R2, R3;
4		STORE R3 M[12] ;

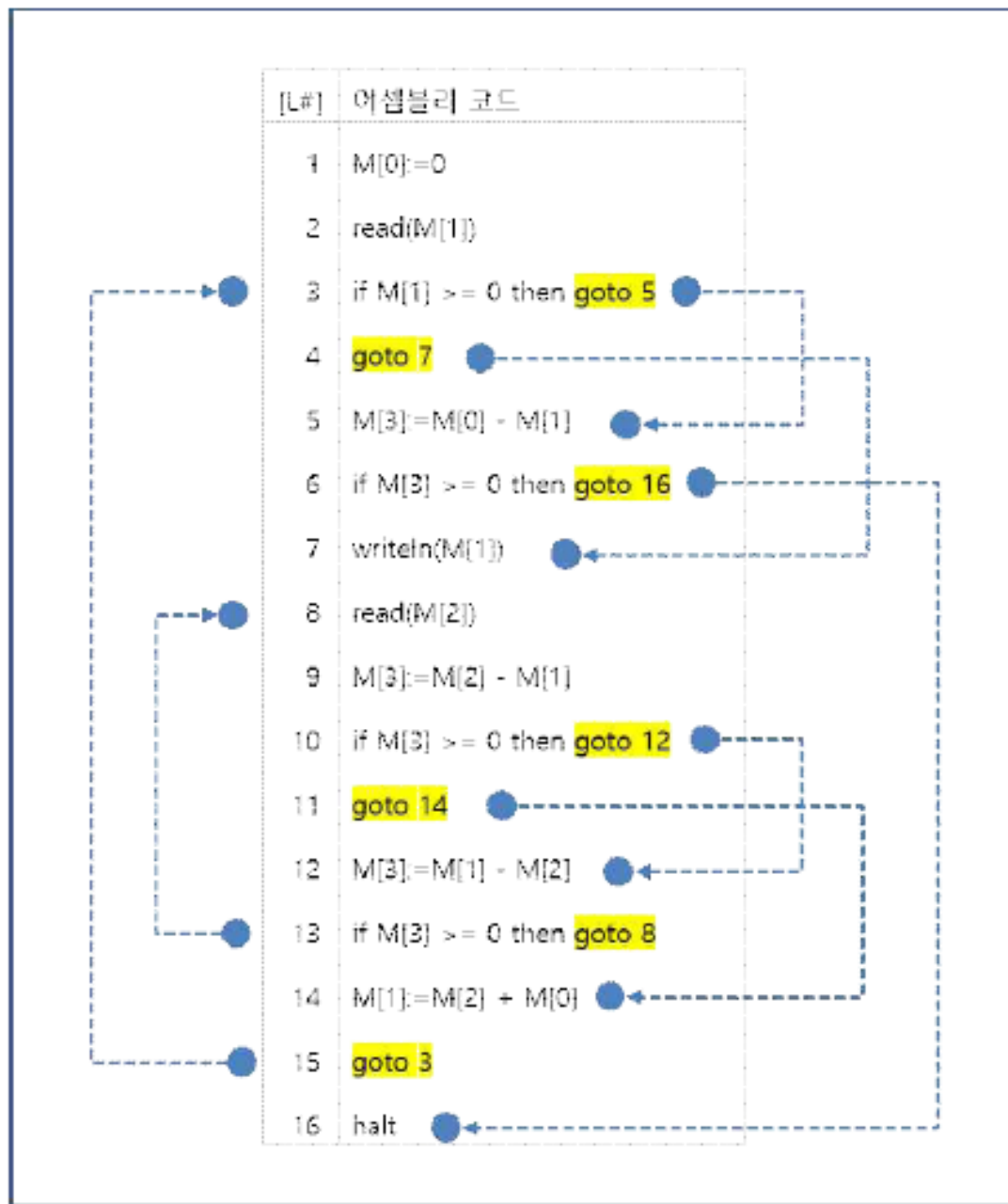
<프로그램 1-9> 어셈블리어 예제

[L#]	어셈블리 코드
1	M[0] :=0
2	read(M[1])
3	if M[1] >= 0 then goto 5
4	goto 7
5	M[3] :=M[0] - M[1]
6	if M[3] >= 0 then goto 16
7	writeln(M[1])
8	read(M[2])
9	M[3] :=M[2] - M[1]
10	if M[3] >= 0 then goto 12
11	goto 14
12	M[3] :=M[1] - M[2]
13	if M[3] >= 0 then goto 8
14	M[1] :=M[2] + M[0]
15	goto 3
16	halt

<프로그램 1-10> 어셈블리 프로그램 예제

입력	1 1 2 2 2 3 1 4 4 0
출력	1 2 3 1 4

어셈블리 코드	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	
입력	1	1	2	2	2	3	1	4	4	0	
1 M[0] := 0											
2 read (M[1])	0/?/?/?										
3 if M[1] >= 0 then goto 5	0/1/?/?										
4 goto 7	T			T ↓			T ↓	T ↓	T ↓		T ↓
5 M[3] := M[0] - M[1]											
6 if M[3] >= 0 then goto 16	0/1/?/-1			0/2/2/-2			0/3/3/-3	0/1/1/-1	0/4/4/-4		0/0/0/0
7 writeln(M[1])	F			F			F	F	F		T ↓
8 read (M[2])	0/1/?/-1			0/2/2/-2			0/3/3/-3	0/1/1/-1	0/4/4/-4		
9 M[3] := M[2] - M[1]	→	0/1/1/-1	0/1/2/0	0/2/2/-2	0/2/2/0	0/2/3/0	0/3/1/-3	0/1/4/-1	0/4/4/-4	0/4/0/0	
10 if M[3] >= 0 then goto 12		0/1/1/0	0/1/2/1	0/2/2/0	0/2/2/0	0/2/3/-1	0/3/1/-2	0/1/4/-3	0/4/4/0	0/4/0/-4	
11 goto 14		T ↓	T	T ↓	T ↓	F	F	F	T ↓	F	
12 M[3] := M[1] - M[2]						↓	↓	↓		↓	
13 if M[3] >= 0 then goto 8		0/1/1/0	0/1/2/-1	0/2/2/0	0/2/2/0				0/4/4/0		
14 M[1] := M[2] + M[0]		T ↑	F	T ↑	T ↑				T ↑		
15 goto 3			0/2/2/-1			0/3/3/-1	0/1/1/-2	0/4/4/-3		0/0/0/-4	
16 halt			↑			↑	↑	↑		↑	halt
출력	1			2			3	1	4		

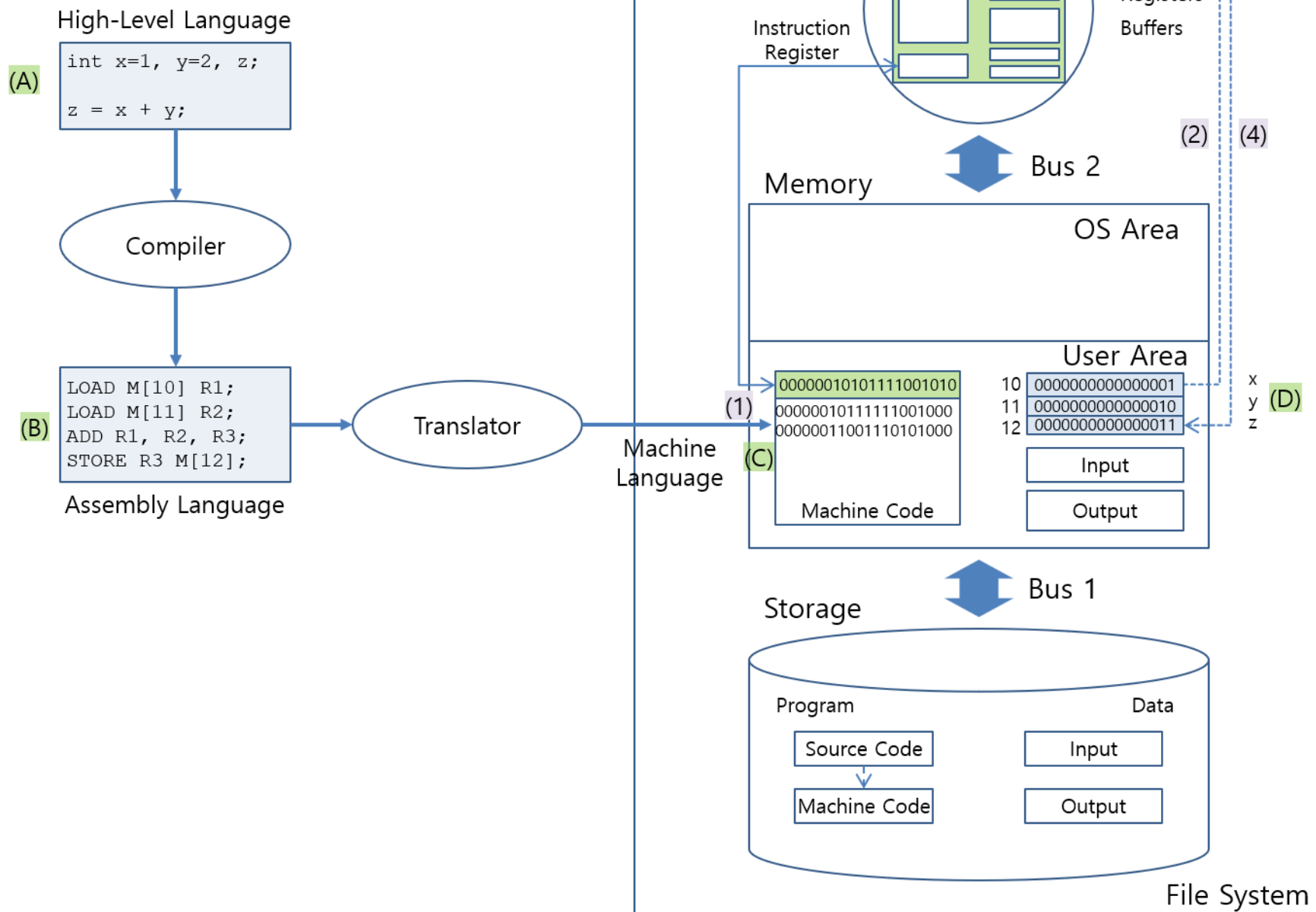


<그림 1-17> 스파게티 코드

상위언어 (HIGH-LEVEL LANGUAGE)

[L#]	Machine 코드	어셈블리 코드	상위언어
1	00000010101111001010	LOAD M[10] R1;	int x=1, y=2, z; z = x + y;
2	00000010111111001000	LOAD M[11] R2;	
3	00000011001110101000	ADD R1, R2, R3; STORE R3 M[12] ;	

<프로그램 1-13> 단계별 언어 예제



1.4 프로그래밍언어의 패러다임

1.5 결어