Discrete Mathematics Assignment

2013210061 채윤병

컴퓨터 프로그램은 논리적인 문제를 해결하기 위한 다양한 명령들의 집합이다. 컴퓨터 프로그램은 일단 다양한 기능(function)으로 구성된다. 그러한 기능은 간단한 덧셈이 될 수도 있고, 복잡한 공식의 알고리즘이 될 수도 있다. 그러한 기능들은 또 다른 기능의 일부가 되기도 한다. 기능들은 명령들로 이루어져 있기 때문에 명령으로 기능을 구성하고 기능들은 서로 여러가지 관계를 맺으면서 하나의 프로그램을 만드는 것이다. 특정한 기능을 갖는 프로그램은 오타마타와 관련이 있다. 오토마타 이론은 계산 능력이 있는 추상 기계와 그 기계를 이용해서 풀 수 있는 문제들을 연구하는 분야이다. 오토마타는 유한한 상태를 갖고, 입력을 받아 입력에 따라 상태를 전이하며 출력을 내놓는 계산 문제를 해결하는 알고리즘 구조를 갖는다. 계산 문제는 오토마타의 기능에 맞게 True or False 답이 도출되는 질문으로 환산되는데, 이 때 이러한 논리를 기계의 체계로 보기 위해서 언어는 오토마타와 관련이 깊다.

따라서 언어를 통해 프로그램을 작성할 때는 위의 문단을 생각하며 작성해야 한다. 그러기 위해선 언어는 어떤 특징을 가져야 한다. 즉 첫째로, 언어는 어떠한 명령과 기능을 지시할 수 있는 능력이 있어야 한다. 언어가 명령을 지시하는 designator가 없다면 명령들을 표현하는 프로그램을 표현할 수 없다. 같은 정수를 범위로 갖는다고 하더라도 다른 값을 가리키고 있다면 다른 변수를 쓰기 때문에 각각의 변수도 가리킬 수 있어야 한다. 또 어떤 기능은 어떤 기능을 포함하고 어떤 명령도 포함하는데 그런 상하관계까지 지정할 수 있다면 더 섬세한 프로그램을 만들 수 있을 것이다.

두번째로, 언어는 논리를 표현할 수 있어야 한다. 참, 거짓, 부정이나 조건 등이 없다면 언어로 논리를 다루는 데는 무리가 있다. 컴퓨터가 계산할 결정 가능한 문제는 곧 True or False를 결과 값으로 갖는다. 그 말은 컴퓨터는 논리를 계산한다는 말인데 논리를 표현할 수 없는데 논리를 계산한다는 것은 불가능하다.

세번째로 언어로 표현된 어떤 문장은 의미를 가지고 있어야 하고 또 문법을 가지고 있어야 한다. First Order Logic에서 모든 기호들이 어떤 의미를 서술하는 것처럼 프로그램을 서술하는 언어도 어떤 의미를 가져야 한다. 의미를 갖지 않는다는 것은 논리도 알고리즘도 표현하지 않는다는 것이기 때문이다. 또 컴퓨터 언어는 하나의 입력에 대해 다양한 해석이 존재하지 않아야 하기 때문에 문법을 가져야 한다. 컴퓨터의 문법은 또한 위의 두 원칙들을 명확하게 표현할 수 있어야 한다.

언어는 그 특징을 잘 나타내기 위해 지원해야할 기능도 있다. 언어는 기능(function), 명령, 프로그램의 관계를 잘 나타내기 위한 기능이 필요하다. 프로그램은 명령을 포함할 수 있지만 명령은 프로그램을 포함할 수 없는 것처럼 ((a,b)가 관계 R에 포함되지만 (b,a)가 관계 R에 포함되지는 않는 경우) 논리적 오류도 확인할 수 있어야 한다. 또 언어는 모든 계산 가능한 문제들을 표현하기 위한 기능들을 가져야 한다. 계산 가능한 논리적인 문제를 다루기 위해 쓰는데 그 자체를 다룰 수 없다면 그 역할을 충분히 못 하기 때문이다.