**프롤로그(PROLOG)**

프롤로그(PROLOG)[[1]](#footnote-1)는 술어 논리식을 프로그램이며 증명하는 것을 계산한다는 것으로 간주하는 관점에서 새로운 계산의 기술형태를 취하고 있다. 즉, 프로그램 자체는 논리식의 모양으로 만들어지고, 그 프로그램을 실행하는 처리계가 그 증명기로 되어 있다.

**1. Data Type**

Prolog의 가장 기본적인 Data type은 Term이다. Prolog의 모든 문장은 Term으로부터 구성된다. Term은 상수, 변수, 혹은 구조로 이루어져 있다.

1) 상수

상수는 Atom과 Number로 구성된다. Atom은 고유의 의미를 갖지 않는 이름이다. 소문자로 시작하는 문자, 숫자, 밑줄이나 특수문자의 조합으로 나타내어진다. 숫자로만 구성된 Atom의 경우 Number와 비슷하지만 이는 산술적인 계산을 실행하지 못한다. 반면 Number은 정수와 실수로 구성되어 있으며 PROLOG 프로그래밍에서는 정수와 실수를 따로 구분하지 않는다.

Ex) skku, abc, 123

2) 변수(Variable)

Atom이 소문자로 시작된다면 변수(Variable)는 대문자나 밑줄(\_)로 시작하는 문자, 숫자, 밑줄의 조합으로 구성된다. 일반적인 언어에서와 달리 변수는 어떤 값을 저장하는 메모리상의 공간을 의미하지 않는다. PROLOG에서 변수를 사용하는 경우, 변수는 instantiate되거나 not instantiate되거나 한다. instantiate가 되었다는 것은 변수가 뜻하는 대상이 존재한다는 의미이다.

Ex) X, Y, Programming, SKKU

3) 구조(Structure)

Term의 구조는 functor와 parameters로 이루어져 있으며 Atom으로 이루어진 functor를 앞에 쓰고 parameters를 괄호 안에 나열하여 구성한다. 아래의

예시에서 parent는 atom으로 구성된 functor이며, tom과 jerry는 Term을 구성하는 parameter가 된다. Term은 functor와 parameter의 개수에 따라 구별된다.

Ex) parent(tom, jerry)

**2. Control sequence and Function**

PROLOG에서는 사실(Fact)과 규칙(Rule)들을 제공하여 데이터베이스를 만들고, 이 데이터베이스에 질의를 함으로써 프로그램을 수행하게 된다. PROLOG 프로그래밍은 대상과 그들 간의 관계에 대한 사실(Facts) 선언, 규칙(Rules)의 정의, 물음(Questions)으로 구성되어 있다. PROLOG 프로그래밍은 사실들과 규칙들로 이루어져 있고, 이것들로 물음에 대한 대답을 한다.

1) 사실(Fact)

하나의 사실은 head와 그에 딸린 arguments로 구성되는 서술식으로 표현 되는데, 예를 들어 "John은 Mary를 좋아한다."라는 사실은 likes(john, mary).로 표현된다. likes는 head이고 john과 mary는 arguments이다. argument의 개수는 head에 따라 달라질 수 있다. “John은 남자이다”라는 사실은 male(john).으로 하나의 argument만 갖는다. 또 argument의 순서 또한 중요하다. likes(john, mary).는 John이 Mary를 좋아하는 것이지만 likes(mary, john).은 Mary가 John을 좋아하는 것이다. 순서 사용은 프로그래머의 선택이나, 하나의 프로그램 안에서는 일관된 순서를 갖는 것이 이해도를 높일 수 있을 것이다.

2) 물음(Question)

PROLOG에서는 물음은 사실(Facts)과 비슷하게 표현하지만, 그 앞에 물음표(?)와 하이픈(-)울 붙인다. 예를 들어 ?- likes(john, mary).라는 물음은 “John이 Mary를 좋아하는가?”라는 뜻을 나타낸다. 이런 물음을 하면 데이터베이스에 있는 사실들을 찾아 물음에 대한 답을 해준다. 물음에 일치하는 사실을 찾으면 yes, 찾지 못한 경우는 no로 답한다. 여기서 no라는 것은 물음에 부합하는 것이 아무 것도 없다는 뜻이며, false를 의미하는 것은 아니다.

예를 들어, likes(john, mary). likes(mary, tom). 이라는 사실이 있을 때 ?- likes(john, mary). 은 yes, ?- likes(tom, mary).는 no라는 답을 해준다.

3) 규칙(Rule)

PROLOG에서의 규칙은 예를 들어 father(X,Y):-parent(X,Y),male(X).와 같이 표현된다. 이 식은 만약 X와 Y가 parent관계이고, 이와 동시에 X가 male이면 X와 Y가 father관계라는 것을 의미한다. ":-"는 만약 ...하다면을, 그리고(and)는 ","로 또는(or)은 ";"로 표시된다.

**3. PROLOG의 장단점**

다른 프로그래밍 언어와 달리 PROLOG는 프로그램 논리가 관계, 사실 및 규칙으로 표현되는 선언적 프로그래밍 언어이다. 추론 기구를 간결하게 표현할 수 있기 때문에 인공지능이나 계산 언어학 분야, 특히 프롤로그가 만들어진 목적이었던 자연언어 처리 분야에서 많이 사용된다.

1) 장점

PROLOG는 최초의 논리적 프로그래밍 언어 중 하나이며 구현에 많은 비용이 들지 않기 때문에 오늘날 인공지능 분야에서 많이 사용되고 있다. 뿐만 아니라 데이터베이스 검색, 음성 제어 시스템과 같은 규칙 기반의 논리적 쿼리로부터 데이터를 추출하는 작업에 가장 적합하다. 즉, PROLOG는 우리가 많이 사용하는 언어인 C, JAVA 등에서 구현하기가 매우 어려운 논리적 작업을 쉽게 수행 할 수 있다.

2) 단점

PROLOG 컴파일러는 다른 일반적인 프로그래밍 언어와 달리 구문의 효율성과 정확성을 검사하는 것은 정말 어렵다. 따라서 작성된 사실이나 규칙이 논리적으로 올바르게 작성된 것처럼 보이지만 실제로 프로그램을 실행하면 생각했던 것과는 다르게 실행되는 경우가 있고 그것의 정확성을 검증하고 확인할 수 없는 경우가 종종 발생한다.

PROLOG는 술어를 사용하기 때문에 프로그래밍이 복잡하고 이해가 어려워질 수 있다. 또 표현방법이 몇 가지 되지 않기 때문에 복잡한 내용을 한 번에 표현할 수 있는 표기법이 없다는 것이 단점이다.

1. https://ko.wikipedia.org/wiki/프롤로그\_(프로그래밍\_언어) [↑](#footnote-ref-1)