

명제 논리

원자적 문장(*atomic sentence*): 명제 기호(*proposition symbol*)로 구성

|
참 또는 거짓 중 하나의 값만 가질 수 있음

$P, Q, R, W_{1,3}, FacingEast$

|
웜푸스가 [1,3]에 있다는 명제

두가지 특수한 명제

True : 항상 참인 명제

False: 항상 거짓인 명제

복합 문장

원자적 문장을 괄호와 아래의 논리적 연결어 5가지를 사용하여 구성

- \neg (not) $\neg W_{1,3}$
- \wedge (and) $W_{1,3} \wedge P_{3,1}$
- \vee (or) $(W_{1,3} \wedge P_{3,1}) \vee W_{2,2}$
- \Rightarrow (implies) $(W_{1,3} \wedge P_{3,1}) \Rightarrow \neg W_{2,2}$
- \Leftrightarrow (if and only if) $\blacklozenge W_{1,3} \Leftrightarrow \neg W_{2,2}$

명제 논리의 문법



John Backus (IBM)

National Medal of Science (1975)

ACM Turing Award (1977)

Backus-Naur form (BNF):

$$\textit{Sentence} \rightarrow \textit{AtomicSentence} \mid \textit{ComplexSentence}$$
$$\textit{AtomicSentence} \rightarrow \textit{True} \mid \textit{False} \mid P \mid Q \mid R \mid \dots$$
$$\textit{ComplexSentence} \rightarrow (\textit{Sentence})$$
$$\mid \neg \textit{Sentence}$$
$$\mid \textit{Sentence} \wedge \textit{Sentence}$$
$$\mid \textit{Sentence} \vee \textit{Sentence}$$
$$\mid \textit{Sentence} \Rightarrow \textit{Sentence}$$
$$\mid \textit{Sentence} \Leftrightarrow \textit{Sentence}$$

OPERATOR PRECEDENCE : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

$\neg A \vee B \wedge C \Rightarrow D$ is equivalent to

$$((\neg A) \vee (B \wedge C)) \Rightarrow D$$


Peter Naur (U. Copenhagen)

ACM Turing Award (2005)

명제 논리의 장단점

프로그래밍 언어는 기본적인 사실들로 부터 다른 사실을 도출하는
메커니즘 부재

프로그래밍 언어는 부분적인 정보를 다루기 위한 표현력이 부족

◆ 명제 논리를 이용하면 위의 두가지 이슈 해결 가능

◆ 구성적 특성(compositionality): 문장의 의미는 그 구성 요소의 의미에
의해 결정

$\neg \text{rain} \vee \neg \text{outside} \vee \text{wet}$

♠ 다수의 객체가 있는 환경을 묘사하기에는 표현력이 부족함

$B_{1,1} \Leftrightarrow (P_{1,2} \vee P_{2,1})$

$B_{1,2} \Leftrightarrow (P_{1,1} \vee P_{1,3} \vee P_{2,2})$

⋮

// 구덩이와 인접한 칸에는 바람이 분다.

♠ 명제 논리는 세상에 사실만 존재한다고 가정

1차 논리 (First-Order Logic)

- ◆ 객체와 그들간의 관계를 기반으로 논리 구축
 - 객체: people, houses, cars, trees, colors, days, ...
 - 관계:
 - ◆ unary properties such as big, windy, ...
 - ◆ n -ary properties such as bigger than, parent of, on, owns, ...
 - 함수: square of, best friend, age, ...
- ◆ 일부 또는 모든 객체에 대한 사실을 표현할 수 있음

1차 논리 구성 요소

◆ 논리 기호

- 연결자: $\wedge, \vee, \Rightarrow, \Leftrightarrow, \neg$
- 괄호 (,) 와 구두점 ,
- 상등: $=$
- 한정자(또는 양화사): \forall (전칭 한정, universal quantification),
 \exists (존재 한정, existential quantification)
- 변수: $x, y, z, \dots; x_1, x_2, \dots$

◆ 비논리 기호

- 상수 기호: Socrates, Turing, 1, earth, ...

- 술어 기호: *true, false*

Father(x, y) // x 는 y 의 아버지이다

Female(x) // x 는 여성이다

- 함수 기호:

gcd(x, y) // x, y 의 최대 공약수

FatherOf(x) // x 의 아버지

1차 논리의 구문

$Sentence \rightarrow AtomicSentence \mid ComplexSentence$

$AtomicSentence \rightarrow Predicate \mid Predicate(Term, \dots) \mid Term = Term$

$ComplexSentence \rightarrow (Sentence)$

$\mid \neg Sentence$

$\mid Sentence \wedge Sentence$

$\mid Sentence \vee Sentence$

$\mid Sentence \Rightarrow Sentence$

$\mid Sentence \Leftrightarrow Sentence$

$\mid Quantifier Variable, \dots Sentence$

$Term \rightarrow Function(Term, \dots)$

$\mid Constant$

$\mid Variable$

$Quantifier \rightarrow \forall \mid \exists$

$Constant \rightarrow A \mid X_1 \mid John \mid \dots$

$Variable \rightarrow a \mid x \mid s \mid \dots$

$Predicate \rightarrow True \mid False \mid After \mid Loves \mid Raining \mid \dots$

$Function \rightarrow Mother \mid LeftLeg \mid \dots$

OPERATOR PRECEDENCE : $\neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow$

영어 문장을 1차 논리로 변환

1. Everyone makes mistakes.

$$\forall x \exists y \text{ Human}(x) \Rightarrow (\text{Does}(x, y) \wedge \text{Mistake}(y))$$

2. One person is an ancestor of another if he is a parent of the other or he is an ancestor of an ancestor of the other.

$$\forall x \forall y \left(\text{Parent}(x, y) \vee \exists z (\text{Ancestor}(x, z) \wedge \text{Ancestor}(z, y)) \right) \Rightarrow \text{Ancestor}(x, y)$$