1-1 논리와 명제

명제의 정의

명제(proposition)는 **참**(**true**; T) 이나 **거짓**(**false**; F)을 나타내는(declare) **문장**이다.

(예) p: 철수가 저녁을 먹는다.
q: 철수가 TV를 본다.
r: 철수가 저녁을 먹으면, TV를 본다.
s: 철수가 착하다.
p, q, r은 명제이나, s는 명제가 아니다.

복합명제(Compounded Proposition)

기존의 명제와 논리연산자(logical operator)를 이용하여 새로운 명제를 만든다.
($\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$)

예) $\neg p$, $p \wedge q$, $p \vee q$, $p \rightarrow (\neg q)$, ...

복합명제 만들기

1. 일진 앞쓰기(unary prefix) 연산자($\neg$) 명제 하나 앞에 연산자를 쓴다.
   $\neg p$: 명제 p를 부정한다. $\neg p$ = $\overline{p}$ 로 쓰기도 한다,
   (예) $\neg p$: 철수가 저녁을 못 먹는다.
   $\neg q$: 철수가 TV를 본다.
2. 이진 사이쓰기(binary infix)연산자($\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$)두개의 명제 사이에 연산자를 넣는다.
   1. $p \wedge q$: 명제 p이고 (and) 명제 q이다.
      (예) $p \wedge q$: 철수가 저녁을 먹고(*and*), TV를 본다.
   2. $p \vee q$: 명제 p 이거나(or) 명제 q이다.
      (예) $p \vee q$: 철수가 저녁을 먹거나(*or*) TV를 본다.
   3. $p \rightarrow q$: (If) 가정명제(hypothesis) p이면,(then) 결론명제(conclusion) q이다.
      (예) $p \rightarrow q$: 철수가 저녁을 먹으면,(*then*) 철수는 TV를 본다.
      $p \rightarrow (\neg q)$: 철수가 저녁을 먹으면,(*then*) 철수는 TV를 못 본다.
   4. $p \leftrightarrow q$: (If, and only if) 명제 p이면, 명제 q이고. 명제 q이면 명제 p이다.
      $(p \rightarrow q) \wedge (q \rightarrow p)$
      (예) $p \leftrightarrow q$: 철수가 저녁을 먹으면 TV를 보고, TV를 보면 저녁을 먹는다.

논리 연산의 진리표(Truth table) $\neg p$, $p \wedge q$, $p \vee q$, $p \rightarrow q$, $p \leftrightarrow q$.

p	$\neg p$
T	F
F	T

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

p	q	$p \rightarrow q$ [1]
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

피연산자(operand)가 한 개(unary)인 경우 피연산자의 참과 거짓이 (T), (F)의 두 가지 경우를 가지므로 2 = 2^1개의 진리표 줄(raw)이 필요하고, 피연산자가 두 개(binary)인 경우 (T, T), (T, F), (F, T), (F, F)의 4가지 경우를 가지므로 진리표에서 2^2 = 4개의 줄이 필요하다. 피연산자(operand)가 n 개(unary)인 경우 2^n개의 줄이 필요하다.

---

1) $p \rightarrow q$가 F인 경우는 가정 p가 T이고 결론 q가 F인 경우 하나뿐이고, 가정이 F이면 결론에 관계없이 $p \rightarrow q$ T이다.

연산자 결합도의 우선순위(precedence of operator)

서로 다른 연산자가 괄호 없이 쓰였을 때, 결합하는 우선순서를 말하며 (¬, ∧, ∨, →, ↔)의 순서로 왼쪽에 연산자가 오른쪽 연산자보다 더 빨리 결합한다.

(예) $p \vee q \wedge \neg r \rightarrow s$ = $(p \vee (q \wedge (\neg r))) \rightarrow s$

명제의 동등함(Equivalence of propositions)

(정의 1.1) 두 명제 p와 q가 같은 논리 값(T 혹은 F)을 가질 때, 명제 p와 q가 동등하다(equivalent)라 부르고, $p \equiv q$라고 쓰고 결합 순위는 가장 낮다. 즉 (¬, ∧, ∨, →, ↔, ≡)낮은 순서의 결합 우선순위를 가진다.

(정의 1.2) 명재 p가 항상 참(T)일 경우 **항진명제**(tautology)라고 부르고, $p \equiv$ T라 쓰고, 항상 거짓(F) 일 경우 **모순명제**(contradiction)라고 부르고, $p \equiv$ F라 쓴다.

(예) $\neg \mathbf{F} \equiv p \vee \mathbf{T} \equiv p \vee \neg p \equiv$ T. $\neg \mathbf{T} \equiv p \wedge \mathbf{F} \equiv p \wedge \neg p \equiv$ F.

나머지의 경우(contingency)는 그냥 명제 p라고만 쓴다.

논리식의 동등

$p \vee \mathbf{F} \equiv p \vee p \equiv p \vee (p \wedge q) \equiv \neg\neg p \equiv p \wedge \mathbf{T} \equiv p \wedge p \equiv p \wedge (p \vee q) \equiv p.$

$(p \vee q) \vee r \equiv p \vee (q \vee r),$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r).$ **결합법칙**

$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r), p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r).$ **분배법칙**

위의 논리식의 동등함은 교과서 25p의 표 6에 잘 정리되어 있다.

∨와 ∧에 결합법칙(associative law)이 성립한다.

$(p \vee q) \vee r \equiv p \vee (q \vee r)$이므로 $(p \vee q) \vee r \equiv p \vee (q \vee r) \equiv p \vee q \vee r$이라 쓸 수 있다.

$n \geq 1$일 때, $(\cdots((p_1 \vee p_2) \vee p_3) \cdots) \vee p_n$[2] $\equiv p_1 \vee (\cdots \vee (p_{n-2} \vee (p_{n-1} \vee p_n)) \cdots))$[3]

$\equiv p_1 \vee p_2 \vee \cdots \vee p_n$[4] $\equiv \bigvee_{i=1}^{n} p_i \equiv \bigvee_{i \in N_n}$[5]$p_i$ (단 $N_n = \{1, 2, \cdots, n\}$)

$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$이므로 $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r) \equiv p \wedge q \wedge r$이라 쓸 수 있다.

$n \geq 1$일 때, $(\cdots((p_1 \wedge p_2) \wedge p_3) \cdots) \wedge p_n$[6] $\equiv p_1 \wedge (\cdots \vee (p_{n-2} \wedge (p_{n-1} \wedge p_n)) \cdots))$[7]

$\equiv p_1 \wedge p_2 \wedge \cdots \wedge p_n$[8] $\equiv \bigwedge_{i=1}^{n} p_i \equiv \bigwedge_{i \in N_n}$[9]$p_i$ (단 $N_n = \{1, 2, \cdots, n\}$)

조건명제(Implication) $p \rightarrow q$의 진리표를 살펴보자.

가정 p와 결론 q의 4가지 경우의 수 (T, T), (T, F), (F, T), (F, F) 중 $p \rightarrow q$가 F인 경우는

2) 이진 연산 ∨이 왼쪽우선결합(Left associative) 이라한다.
3) 이진 연산 ∨이 오른쪽우선결합(Right associative) 이라한다.
4) 연산 ∨이 n개의 피연산자(operand)를 가지는 n진(n-ary)연산 이라한다.
5) Indexed set notation이라 하고, $N_n = \{1, 2, \ldots, n\}$를 indexed set이라 부른다.
6) 이진 연산 ∧이 왼쪽우선결합(Left associative) 이라한다.
7) 이진 연산 ∧이 오른쪽우선결합(Right associative) 이라한다.
8) 연산 ∧이 n개의 피연산자(operand)를 가지는 n진(n-ary)연산 이라한다.
9) Indexed set notation이라 하고, $N_n = \{1, 2, \ldots, n\}$를 indexed set이라 부른다.

(T, F) 하나뿐이고 나머지는 모두 T이다. 가정 p가 T일 때, 결론q가 T이면 조건 $p \to q$도 T이고, 결론q가 F이면, 조건 $p \to q$도 F인 것은 적절해 보아나, 가정 p가 F일 때, 결론q가 T이건 F이건, 항상 조건 $p \to q$도 T인 것이 조건명제 $p \to q$에서 주의 하여 살펴봐야 할 점이다.
위의 저녁과 TV 예에서 $\neg p \to \neg q$라는 조건명제를 생각하면, 어머니께서 "철수야 너 TV 안 보면 저녁 못 먹는다!"라고 회유하셨을 경우, 철수가 착하게 저녁을 먹어도 TV를 볼 수도 있고, 못 볼 수도 있다는 뜻이다.

$p \to q$는 p가 F이거나 q가 T인 경우에 T이므로 $p \to q \ \equiv \ \neg p \ \vee \ q$이다.

<u>De Morgan의 법칙</u>
명제 p, q에 합 $p \vee q$과 곱 $p \wedge q$의 역 $\neg(p \vee q)$와 $\neg(p \wedge q)$을 진리표를 이용해 알아보자.

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	F

$$\neg(p \vee q) \ \equiv \ \neg p \wedge \neg q$$

p	q	$p \wedge q$	$\neg(p \wedge q)$	$\neg p$	$\neg q$	$\neg p \vee \neg q$
T	T	T	F	F	F	F
T	F	F	T	F	T	T
F	T	F	T	T	F	T
F	F	F	T	T	T	T

$$\neg(p \wedge q) \ \equiv \ \neg p \vee \neg q$$

연산 $\vee$와 $\wedge$가 모두 결합법칙을 만족시키므로 De Morgan의 법칙도 일반화 할 수 있다.
$n \geqq 1$일 때, $\neg(p_1 \vee p_2 \vee \cdots \vee p_n) \ \equiv \ \neg p_1 \wedge \neg p_2 \wedge \cdots \wedge \neg p_n$. $\quad \neg\left(\bigvee_{i \in N_n} p_i\right) \ \equiv \ \bigwedge_{i \in N_n} \neg p_i$

$\neg(p_1 \wedge p_2 \wedge \cdots \wedge p_n) \ \equiv \ \neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n$. $\quad \neg\left(\bigwedge_{i \in N_n} p_i\right) \ \equiv \ \bigvee_{i \in N_n} \neg p_i$