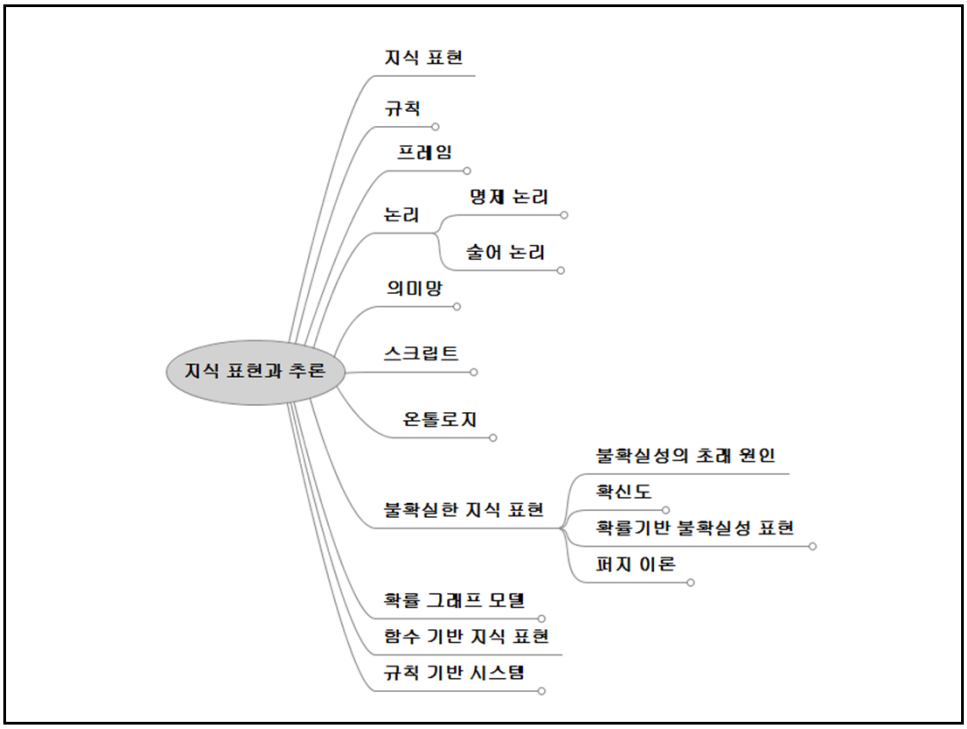
지식표현과 추론 - I

충북대학교 소프트웨어학과
이건명



1. 지식 표현

❖ **데이터 피라미드**

- **데이터** (data)
  - 특정 분야에서 **관측된 아직 가공되는 않은 것**
  - 사실인 것처럼 관측되지만 **오류**나 **잡음**을 포함 가능
- **정보** (information)
  - 데이터를 **가공**하여 어떤 **목적**이나 **의미**를 갖도록 한 것
- **지식** (knowledge)
  - 정보를 **취합**하고 **분석**하여 얻은 대상에 대해 사람이 **이해**한 것
- **지혜** (wisdom)
  - 경험과 학습을 통해서 얻은 지식보다 높은 수준의 **통찰**

지식 표현

❖ **지식**(知識, knowledge)

- **경험**이나 **교육**을 통해 얻어진 **전문적인 이해**(understanding)와 체계화된 **문제 해결 능력**
- 어떤 주제나 분야에 대한 **이론적** 또는 **실제적인 이해**, 또는 현재 알려진 **사실**과 **정보**의 모음
- **암묵지**(暗黙知, tacit knowledge)
  - 형식을 갖추어 표현하기 어려운, 학습과 경험을 통해 쌓은 지식
- **형식지**(形式知, explicit knowledge)
  - 비교적 쉽게 형식을 갖추어 표현될 수 있는 지식
- **절차적 지식**(procedural knowledge)
  - 문제해결의 절차 기술
- **선언적 지식**(declarative knowledge)
  - 어떤 대상의 성질, 특성이나 관계 서술
- **컴퓨터를 통한 지식 표현 및 처리**
  - 프로그램이 쉽게 처리할 수 있도록 **정형화된 형태**로 표현
  - 규칙, 프레임, 논리, 의미망, 스크립트, 수치적 함수 등

2. 규칙

❖ **규칙 (rule)**
- **'~이면, ~이다'** 또는 **'~하면, ~하다'**와 같은 **조건부의 지식**을 표현하는 **IF-THEN 형태**의 문장
- 직관적
- 이해하기 쉬움

❖ **규칙 획득 및 표현**
- **예.** 신호등이 녹색일 때는 건널목을 안전하게 건널 수 있고, 빨간색일 때는 길을 건너지 말아야 한다
- **대상**, **속성**, **행동** 또는 **판단**의 정보 추출
  - 대상 : 신호등
  - 속성 : 녹색, 빨간색
  - 행동/판단 : 건넌다, 멈춘다.
- **표현**
  - **IF** 신호등이 녹색이다 **THEN** 행동은 건넌다
  - **IF** 신호등이 빨간색이다 **THEN** 행동은 멈춘다

규칙

❖ **규칙 (rule)**
- **IF** 신호등이 녹색이다 **THEN** 행동은 건넌다
- **IF** 신호등이 빨간색이다 **THEN** 행동은 멈춘다

- **IF** trafficLight = green **THEN** action = cross
- **IF** trafficLight= red **THEN** action = stop

- **IF 부분**
  - 주어진 정보나 사실에 대응될 조건
  - **조건부**(conditional part, antecedent)

- **THEN 부분**
  - 조건부가 만족될 때의 판단이나 행동
  - **결론부**(conclusion, consequent)

규칙

❖ **규칙의 구성**

- **조건부**
  - 둘 이상의 조건을 **AND** 또는 **OR**로 **결합**하여 구성 가능
    - **IF** <조건1> **AND** <조건2> **AND** <조건3> **THEN** <결론>
    - **IF** <조건1> **OR** <조건2> **OR** <조건3> **THEN** <결론>

- **결론부**
  - 여러 개의 판단 또는 행동 포함 가능
    - **IF** <조건>
      THEN <결론1>
      AND <결론2>
      AND <결론3>

규칙

❖ **규칙을 통한 지식 표현**

- **인과관계**
  - **원인**을 조건부에 **결과**는 결론부에 표현
    - **IF** 연료통이 빈다
      THEN 차가 멈춘다

- **추천**
  - **상황**을 조건부에 기술하고 이에 따른 **추천 내용**을 결론부에 표현
    - **IF**여름철이다 **AND** 날이 흐리다
      THEN 우산을 가지고 가라

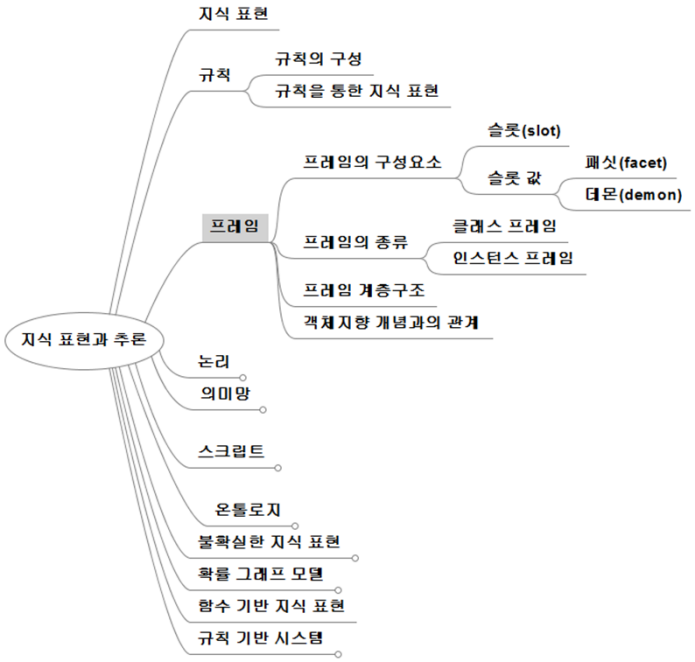
- **지시**
  - **상황**을 조건부에 기술하고 이에 따른 **지시 내용**을 결론부에 표현
    - **IF** 차가 멈추었다 **AND** 연료통이 비었다
      THEN 주유를 한다

규칙

❖ **규칙을 통한 지식 표현**

- **전략 (strategy)**
  - **일련의 규칙**들로 표현
  - 이전 단계의 판정 결과에 따라 다음 단계에 고려할 규칙이 결정
    - **IF** 차가 멈추었다
      THEN 연료통을 확인한다 **AND** 단계1을 끝낸다
    - **IF** 단계1이 끝났다 **AND** 연료통은 충분히 찼다
      THEN 배터리를 확인한다 **AND** 단계2를 끝낸다

- **휴리스틱 (heuristic)**
  - **경험적인 지식**을 표현하는 것
  - 전문가적 견해는 최적을 항상 보장하는 것이 아니고 **일반적으로 바람직한 것**을 표현
    - **IF** 시료가 액체이다 **AND** 시료의 PH가 6미만이다 **AND** 냄새가 시큼하다
      THEN 시료는 아세트산이다



3. 프레임

❖ **프레임**(frame)

- 민스키(M. Minsky, 1927~2016)가 제안한 지식표현 방법
- 특정 **객체** 또는 **개념**에 대한 **전형적인 지식**을 **슬롯**(slot)**의 집합**으로 표현하는 것

- 예. **컴퓨터**를 표현한 프레임

frame-name	**Computer**	
frame-type	Class	
CPU	default	Intel
	data-type	string
	require	Intel AMD ARM SPARC
OS	default	Windows
	data-type	string
memory	data-type	integer
warranty	default	3years
HDD	default	1TB
price	data-type	integer
stock	default	in-stock

(**frame**
(**frame-name** Computer)
(**frame-type** class)
(**CPU** (default Intel)
(data-type string)
(require (Intel AMD ARM SPARC)))
(**OS** (default Windows)
(data-type string)
(**memory** (data-type integer))
(**warranty** (default 3years)
(**HDD** (default 1TB))
(**price** (data-type integer))
(**stock** (default in-stock)))

프레임

❖ **프레임의 구성요소**

- **슬롯(slot)**
  - 객체의 **속성**(attribute)을 기술하는 것
  - **슬롯 이름**(slot name)과 **슬롯 값**(slot value)으로 구성
    - 슬롯 이름: 속성 이름
    - 슬롯 값: 속성의 값
  - **슬롯 값(slot value)**
    - 복수 개의 **패싯**(facet)과 **데몬**(demon)으로 구성

프레임

❖ **프레임의 구성요소 – Cont.**

- **패싯 (facet)**
  - '측면' 또는 '양상'을 의미
  - 속성에 대한 부가적인 정보를 지정하기 위해 사용
  - **패싯 이름**과 **패싯 값**의 쌍으로 구성

  - **패싯 이름**
    - **value** : **속성값 (**수, 문자열, 다른 프레임의 포인터 등**)**
    - **data-type** : 속성값의 **자료형**
    - **default** : 디폴트값(속성값이 주어지지 않을 때 사용되는 **초기값**)
    - **require** : 슬롯에 **들어갈 수 있는 값**이 만족해야 할 **제약조건**

CPU	default	Intel
	data-type	string
	require	Intel AMD ARM SPARC
memory	data-type	integer

프레임

❖ **프레임의 구성요소 – Cont.**

- **데몬(demon)**
  - 지정된 **조건**을 만족할 때 **실행할 절차적 지식**(procedure)을 기술
  - 슬롯 값으로 **데몬 실행조건**과 **데몬 이름**의 쌍

  - 데몬의 **실행조건**의 예
    - **if_needed** : 슬롯 값을 알아야 할 때(즉, **사용**하려고 할 때)
    - **if_added** : 슬롯 값이 **추가**될 때
    - **if_removed** : 슬롯 값이 **제거**될 때
    - **if_modified** : 슬롯 값이 **수정**될 때

HDD	value	512GB
price	if-needed	look-up-the-list
stock	if-needed	ask-for-vendor

데몬 실행조건 데몬 이름

프레임

❖ **프레임의 종류**
- **클래스(class) 프레임**
  - 부류(class)에 대한 정보 표현
- **인스턴스(instance) 프레임**
  - 특정 객체에 대한 정보 표현

❖ **프레임 계층구조(hierachy)**
- **상위 프레임**
  - 클래스를 나타내는 프레임
- **하위 프레임**
  - 하위 클래스 프레임 또는 상위 클래스 프레임의 객체
  - 상위 프레임을 **상속**(inheritance) 받음

프레임

❖ **프레임 표현의 예 : 컴퓨터**
- **클래스 프레임 Computer**

(**frame**
  (**frame-name** Computer)
  (**frame-type** class)
  (**CPU** (**default** Intel)
    (**data-type** string)
    (**require** (Intel AMD ARM SPARC)))
  (**OS** (**default** Windows)
    (**data-type** string)
  (**memory** (**data-type** integer))
  (**warranty** (**default** 3years)
  (**HDD** (**default** 1TB))
  (**price** (**data-type** integer))
  (**stock** (**default** in-stock)))

frame-name	Computer	
frame-type	Class	
CPU	default	Intel
	data-type	string
	require	Intel AMD ARM SPARC
OS	default	Windows
	data-type	string
memory	data-type	integer
warranty	default	3years
HDD	default	1TB
price	data-type	integer
stock	default	in-stock

프레임

❖ **프레임 표현의 예 : 컴퓨터**
- **인스턴스 프레임 Ultra-Slim-Notebook**

frame-name	Computer	
frame-type	Class	
CPU	default	Intel
	data-type	string
	require	Intel AMD ARM SPARC
OS	default	Windows
	data-type	string
memory	data-type	integer
warranty	default	3years
HDD	default	1TB
price	data-type	integer
stock	default	in-stock

(**frame**
(**frame-name** Ultra-Slim-Notebook)
(**frame-type** **instance** (class Computer))
(**CPU** (**value** ARM))
(**OS** (**value** Android))
(**memory** (**value** 4G))
(**HDD** (**value** 512GB))
(**price** (**if-needed** look-up-the-list)
(**stock** (**if-needed** ask-for-vendor)))

(**warranty** 3years) ?

frame-name	Ultra-Slim-Notebook	
frame-type	**instance** (class Computer)	
CPU	value	ARM
OS	value	Android
memory	value	4G
warranty	value	**3years**
HDD	value	512GB
price	if-needed	look-up-the-list
stock	if-needed	ask-for-vendor

데몬

프레임

❖ **프레임과 규칙을 결합한 지식 표현**
- **프레임**은 특정 **개념**이나 **대상**에 대한 **속성들** 표현
  - 관련된 속성들을 하나의 덩어리로 관리
- **규칙**을 사용하여 **조건적인 지식** 표현
  - **데몬**에 **규칙 사용**
  - 또는 **규칙**의 **조건부**나 **결론부**에서 **프레임** 사용
- 대부분의 규칙기반 시스템에서 **객체**(object) 개념 사용
  - 객체의 표현에 프레임 사용 가능

프레임

❖ **프레임** vs **클래스와 객체**

- **클래스와 객체**
  - 소프트웨어 개발에 있어서 **모듈화**, **재사용성** 및 **유지보수**의 용이성을 고려한 프로그래밍 개념
  - **정보은닉** 등 **정보 접근**에 대한 제한 메커니즘

- **프레임**
  - 사람이 특정 **대상**에 대해 갖는 **지식의 표현**을 목표
  - 슬롯의 특정 상황에 따라 **자동**으로 **호출**되는 **데몬** 개념

4. 논리

❖ **논리**(論理, logic)

- **말로 표현**된 **문장**들에 대한 **타당한 추론**을 위해,
  기호를 사용하여 문장들을 **표현**하고
  기호의 **조작**을 통해 문장들의 **참 또는 거짓**을 **판정**하는 분야

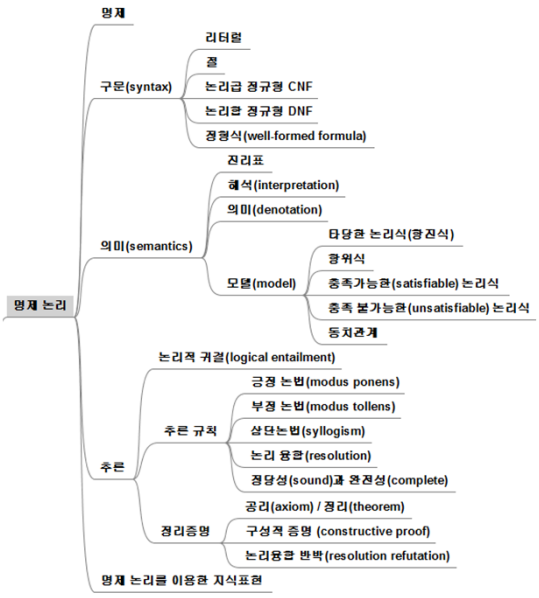
❖ **논리학의 역사**

- **아리스토텔레스**(Aristotle, BC384-BC322)
  - 기호의 대수적 조작을 통해 추론을 하는 **삼단 논법**(syllogism) 도입

- **부울**(George Boule, 1815-1864)
  - **명제 논리**(propositional logic)의 이론적 기초 확립

- **프리게**(Gottlob Frege, 1848-1925)
  - **술어 논리**(predicate logic)의 이론적 기초를 확립

4.1 명제 논리



4.1 명제 논리

❖ **명제 논리 (propositional logic)**

- **명제**(命題, proposition)
  - 참, 거짓을 분명하게 판정할 수 있는 문장
    - 아리스토텔레스는 플라톤의 제자이다. (명제)
    - 1+1 = 3. (명제)
    - 일어나서 아침 먹자. (명제 아님)

- 명제를 **P, Q**등과 같은 **기호로 표현**
- 명제 기호의 **진리값**(truth value)을 사용하여 명제들에 의해 표현되는 문장들의 진리값 결정
- 문장 자체의 **내용**에 대해는 **무관심**, 문장의 **진리값**에만 **관심**

명제 논리

❖ **기본 명제**(primitive proposition)
- 하나의 진술(statement)로 이루어진 최소 단위의 명제

❖ **복합 명제**(compound proposition)
- 기본 명제들이 결합되어 만들어진 명제

❖ 예.
- 알렉산더는 아시아를 넘본다 ⇨ P
- 징기스칸은 유럽을 넘본다 ⇨ Q
- 알렉산더는 아시아를 넘보고, 징기스칸은 유럽을 넘본다 ⇨ P ∧ Q

명제 논리의 구문

❖ **논리식**(logical expression)
- 명제를 기호로 표현한 형식
- **명제기호**, 참과 거짓을 나타내는 **T**와 **F**, 명제 기호를 연결하는 **논리기호**인 ¬, ∨, ∧, →, ≡를 사용하여 구성

논리 기호	이름	논리식	의미
¬	부정(negation)	$\neg P$	P가아님
∨	논리합(disjunction)	$P \vee Q$	P 또는 Q
∧	논리곱(conjunction)	$P \wedge Q$	P 그리고 Q
→	함의(implication)	$P \rightarrow Q$	P 이면 Q
≡	동치(equivalence)	$P \equiv Q$	$(P \rightarrow Q) \wedge (Q \rightarrow P)$

명제 논리의 구문

❖ **리터럴**(literal)
- 명제 기호 P와 명제 기호의 $\neg P$ 부정

❖ **절**(clause)
- 리터럴들이 **논리합**으로만 연결되거나 **논리곱**으로 연결된 논리식

$P \vee Q \vee \neg R$ (논리합 절)

$P \wedge Q \wedge \neg R$ (논리곱 절)

❖ **논리곱 정규형 (conjunctive normal form, CNF)**
- 논리합 절들이 논리곱으로 연결되어 있는 논리식

$$(P \vee Q \vee \neg R) \wedge (\neg Q \vee R \vee S) \wedge (P \vee R \vee S)$$

❖ **논리합 정규형 (disjunctive normal form, DNF)**
- 논리곱 절들이 논리합으로 연결되어 있는 논리식

$$(P \wedge Q \wedge \neg R) \vee (\neg Q \wedge R \wedge S) \vee (P \wedge R \wedge S)$$

명제 논리의 구문

❖ **정형식**(well-formed formula, **wff**)
- 논리에서 문법에 맞는 논리식

- **명제 논리에 대한 정형식**

(1) 진리값 T, F와 명제 기호들 $P, Q, R, \cdots$은 정형식이다.

(2) p와 q가 정형식이면, 논리 기호를 사용하여 구성되는 논리식 $\neg p$, $p \vee q$, $p \wedge q$, $p \rightarrow q$, $p \equiv q$도 정형식이다.

(3) (1)과 (2)에 의해 정의되는 논리식만 정형식이다.

$(P \wedge Q) \rightarrow \neg P$

$P \rightarrow \neg P$

$P \vee P \rightarrow P$

$(P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P)$

명제 논리의 의미

❖ **진리표**(truth table)
- 논리기호에 따라 참, 거짓 값을 결합하는 방법을 나타낸 표

P	Q	$\neg P$	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \equiv Q$
F	F	T	F	F	T	T
F	T	T	T	F	T	F
T	F	F	T	F	F	F
T	T	F	T	T	T	T

❖ 논리식의 **해석**(interpretation)
- **논리식**의 **진리값**을 **결정**하는 것
  - P = T, Q = F, R = T
    - $(P \vee \neg Q) \wedge (Q \vee \neg R)$ = F
- 우선 각 **명제기호**의 **진리값** 결정 필요
  - 명제 기호에 "명제"를 대응시키고, 해당 명제의 진리값을 결정
    - 예. $\boldsymbol{P}$ => "**토마토는 과일이다**", P = F
  - **대응된 명제**를 명제 기호의 **외연**(外延) 또는 **의미**(denotation)라 함

명제 논리의 의미

❖ 논리식의 **모델**(model)
- 논리식의 **명제기호**에 **참값**(T) 또는 **거짓값**(F)을 **할당**한 것
  - $(P \vee \neg Q) \wedge (Q \rightarrow \neg R)$: P = T, Q = F, R = T

- 모델이 주어지면, **진리표**를 사용하여 논리식의 진리값 결정, 즉 **해석** 가능

P	Q	$\neg P$	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \equiv Q$
F	F	T	F	F	T	T
F	T	T	T	F	T	F
T	F	F	T	F	F	F
T	T	F	T	T	T	T

- $\boldsymbol{n}$**개**의 **명제기호**가 **논리식**에 사용된다면, 각각 T 또는 F값을 가질 수 있기 때문에, 총 $\boldsymbol{2^n}$**개**의 **모델**이 존재

명제 논리의 의미

❖ **타당한 논리식** (**valid** logical expression)
- 모든 가능한 모델에 대해서 **항상 참**(T)인 논리식
- **항진식**(恒眞式, tautology)
  - 예. $P \vee \neg P$
    P = T인 경우 : $P \vee \neg P$ = T
    P = F인 경우 : $P \vee \neg P$ = T

❖ **항위식**(恒僞式, contradiction)
- 모든 가능한 모델에 대해서 **항상 거짓**이 되는 논리식
  - 예. $P \wedge \neg P$
    P = T인 경우 : $P \wedge \neg P$ = F
    P = F인 경우 : $P \wedge \neg P$ = F

명제 논리의 의미

❖ **충족가능한**(satisfiable) 논리식
- 참으로 만들 수 있는 모델이 하나라도 있는 논리식
  - 예. $(P \vee \neg Q) \wedge (Q \vee \neg R)$
    P = T, Q = T, R = F

❖ **충족불가능한**(unsatisfiable) 논리식
- 참으로 만들 수 있는 모델이 전혀 없는 논리식
- **항위식**인 논리식
  - 예. $P \wedge \neg P$

명제 논리의 의미

❖ **동치관계**(equivalence relation)의 논리식
- **어떠한 모델**에 대해서도 **같은 값**을 갖는 두 논리식

(1) $\neg(\neg p) \equiv p$
(2) $p \vee \mathrm{F} \equiv p,\ \ p \wedge \mathrm{T} \equiv p$
(3) $p \vee \neg p \equiv \mathrm{T},\ \ p \wedge \neg p \equiv \mathrm{F}$
(4) $\neg(p \wedge q) \equiv \neg q \vee \neg q,\ \neg(p \vee q) \equiv \neg q \wedge \neg q$
(5) $p \rightarrow q \equiv \neg p \vee q$
(6) $p \vee (q \vee r) \equiv (p \vee q) \vee r,\ p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$
(7) $p \vee (q \wedge q) \equiv (p \vee q) \wedge (p \vee q),\ p \wedge (q \vee q) \equiv (p \wedge q) \vee (p \wedge q)$

명제 논리의 의미

❖ **동치관계를 이용한 논리식의 변환**
- 논리식의 동치관계를 이용하면
  임의의 논리식을 **논리곱 정규형**(CNF)과 같은 정형식으로 변환

$$
\begin{aligned}
& p \wedge (q \rightarrow r) \rightarrow p \wedge q \\
& \equiv \neg(p \wedge (\neg q \vee r)) \vee (p \wedge q) \\
& \equiv (\neg p \vee \neg(\neg q \vee r)) \vee (p \wedge q) \\
& \equiv (\neg p \vee (q \wedge \neg r)) \vee (p \wedge q) \\
& \equiv ((\neg p \vee q) \wedge (\neg p \vee \neg r)) \vee (p \wedge q) \\
& \equiv ((\neg p \vee q) \vee (p \wedge q)) \wedge ((\neg p \vee \neg r) \vee (p \wedge q)) \\
& \equiv ((\neg p \vee q) \vee p) \wedge ((\neg p \vee q) \vee q)) \wedge ((\neg p \vee \neg r) \vee p) \wedge ((\neg p \vee \neg r) \vee q)) \\
& \equiv (T \vee q) \wedge (\neg p \vee q) \wedge (T \vee \neg r) \wedge (\neg p \vee \neg r \vee q) \\
& \equiv (\neg p \vee q) \wedge (\neg p \vee q \vee \neg r)
\end{aligned}
$$

명제 논리의 의미

❖ **논리적 귀결**(logical entailment)

- Δ : 정형식(wff)의 집합. $\Delta = \{P, P \rightarrow Q\}$
  ω : 정형식. $\omega = Q$

- **Δ**에 있는 모든 정형식을 **참**(T)으로 만드는 **모델**이, **ω**를 **참**(T)으로 만든다
  = Δ는 ω를 **논리적으로 귀결**한다(**logically entail**)
  = ω는 Δ를 **논리적으로 따른다**(**logically follow**)
  = ω는 Δ의 **논리적 결론**(**logical consequence**)이다

  - Δ가 참이면, ω도 참이다
- **추론**
  - 참으로 알려진 Δ로 부터, 알려지지 않은 참인 ω를 찾는 것

명제 논리의 추론

❖ **추론**(推論, inference)

- **귀납적 추론**(inductive inference)
  - 관측된 복수의 **사실들**을 **일반화**(generalization)하여 **일반적인 패턴** 또는 **명제**를 도출하는 것
- **연역적 추론**(deductive inference)
  - **참**인 **사실들** 또는 **명제들**로 부터 **새로운** 참인 **사실** 또는 **명제**를 도출하는 것

- **논리**에서의 **추론**
  - 함의(→)의 논리적 관계를 이용하여 **새로운 논리식**을 **유도**해 내는 것
  - 함의 $\boldsymbol{p \rightarrow q}$
    - $\boldsymbol{p}$: **전제**(premise)
    - $\boldsymbol{q}$: **결론**(conclusion, consequence)

명제 논리의 추론

❖ **추론규칙**(inference rule)

- 참인 논리식들이 **논리적으로 귀결**하는 **새로운 논리식**을 만들어내는 **기계적**으로 **적용**되는 규칙

- **긍정 논법** (modus ponens)

$$\frac{p \rightarrow q \quad p}{q}$$

새이다 → 날 수 있다
새이다

날 수 있다

$p \rightarrow q, p \vdash q$

- **부정 논법** (modus tollens)

$$\frac{p \rightarrow q \quad \neg q}{\neg p}$$

새이다 → 날 수 있다
날 수 없다

새가 아니다

$p \rightarrow q, \neg q \vdash \neg p$

- **삼단 논법** (syllogism)

$$\frac{p \rightarrow q \quad q \rightarrow r}{p \rightarrow r}$$

새이다 → 날개가 있다
날개가 있다 → 날 수 있다

새이다 → 날 수 있다

$p \rightarrow q, q \rightarrow r \vdash p \rightarrow r$

명제 논리의 추론

❖ **추론규칙** – cont.

- **논리 융합 (resolution)**
  - 일반화된 추론규칙
    - 긍정 논법, 부정 논법, 삼단 논법의 규칙을 포함한 추론 규칙
  - **두 개**의 **논리합절**이 **같은 기호**의 **긍정**과 **부정**의 **리터럴**을 서로 포함하고 있을 때, 해당 리터럴들을 제외한 **나머지 리터럴들**의 **논리합절**을 만들어 내는 것

$$P \vee q, \neg P \vee r \vdash q \vee r$$

논리융합식(resolvent)

명제 논리의 추론

❖ **추론 규칙의 정당성과 완전성**

- **추론 규칙의 정당성 (sound)**
  - 추론 규칙에 의해 생성된 논리식은 주어진 논리식들이 논리적으로 귀결하는 것이다.
  - 즉, 추론 규칙이 만들어 낸 것은 항상 참이다.

$$\Delta \vdash \omega \rightarrow \Delta \models \omega$$

- **추론 규칙의 완전성 (complete)**
  - 주어진 논리식들이 **논리적으로 귀결하는 것**들은 **추론 규칙**이 **찾아 낼 수 있다**.

명제 논리의 추론

❖ **정리증명**(theorem proving)

- **공리**(axiom)
  - 추론을 할 때, 참인 것으로 주어지는 논리식
- **정리**(theorem)
  - 공리들에 추론 규칙을 적용하여 얻어지는 논리식
- **정리 증명**
  - 공리들을 사용하여 정리가 참인 것을 보이는 것
  - **구성적 증명**(**constructive proof**)
    - 공리들에 추론 규칙들을 적용하여 증명을 만들어 보이는 증명
  - **논리융합 반박**(**resolution refutation**)
    - 증명할 정리를 부정(negation)한 다음, 논리융합 방법을 적용하여 모순이 발생하는 것을 보여서, 정리가 참임을 증명하는 방법

명제 논리의 추론

❖ **논리융합 반박을 이용한 정리증명의 예**

- **공리**

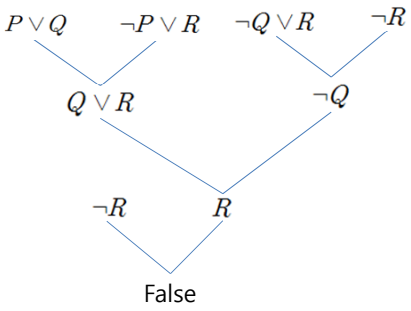
$P \vee Q$

$P \rightarrow R \quad \neg P \vee R$

$Q \rightarrow R \quad \neg Q \vee R$

- **정리**

$R \quad \Rightarrow \quad \neg R$



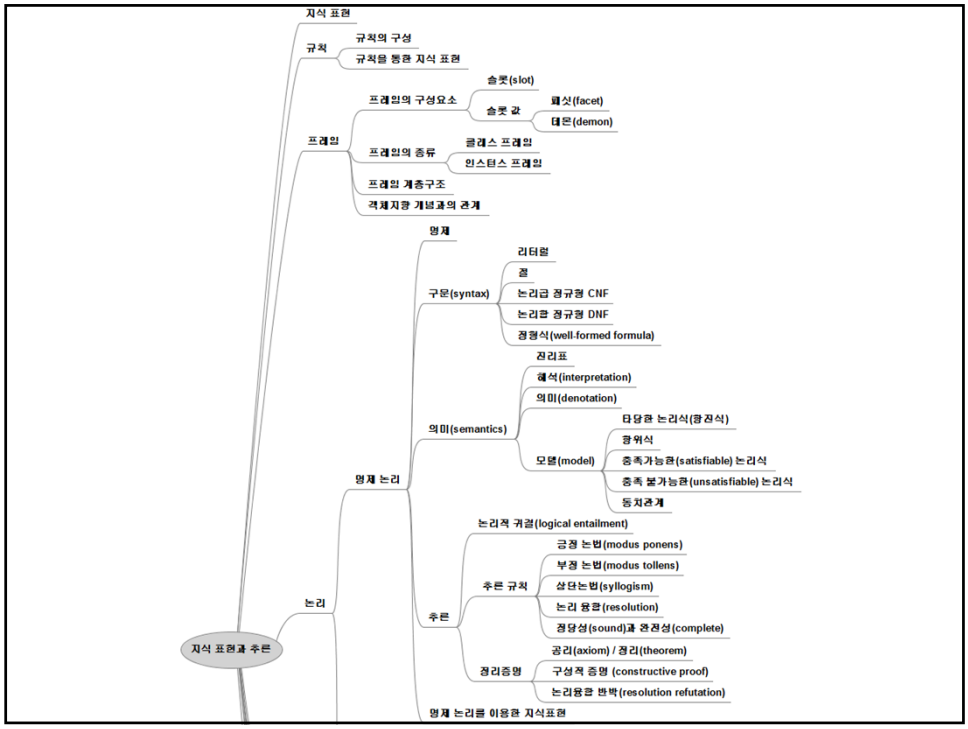
명제 논리의 지식표현

❖ **명제 논리를 이용한 지식 표현**

- 문장으로 표현된 지식으로부터 **기본 명제**들을 **추출**
- 각 명제에 대해 **명제기호 부여**
- 기본 명제들의 **논리적 연결 관계**를 참고하여 대응되는 명제 기호들을 **논리기호**로 **연결**하여 논리식 구성

❖ **명제 논리로 표현된 지식에 대한 추론**

- **명제 기호**가 나타내는 명제의 **의미**와는 **무관**
- 대수적인 **기호 연산**을 통해서 **추론** 수행


지식 표현과 추론
지식 표현
규칙
규칙의 구성
규칙을 통한 지식 표현
프레임
프레임의 구성요소
슬롯(slot)
슬롯 값
패싯(facet)
데몬(demon)
프레임의 종류
클래스 프레임
인스턴스 프레임
프레임 계층구조
객체지향 개념과의 관계
논리
명제 논리
명제
구문(syntax)
리터럴
절
논리곱 정규형 CNF
논리합 정규형 DNF
정형식(well-formed formula)
의미(semantics)
진리표
해석(interpretation)
의미(denotation)
모델(model)
타당한 논리식(항진식)
항위식
충족가능한(satisfiable) 논리식
충족 불가능한(unsatisfiable) 논리식
동치관계
추론
논리적 귀결(logical entailment)
추론 규칙
긍정 논법(modus ponens)
부정 논법(modus tollens)
삼단논법(syllogism)
논리 융합(resolution)
정당성(sound)과 완전성(complete)
정리증명
공리(axiom) / 정리(theorem)
구성적 증명 (constructive proof)
논리융합 반박(resolution refutation)
명제 논리를 이용한 지식표현