

지식표현(Knowledge Representation)

- 지식은 인공지능에서 가장 핵심
- 지식표현 연구는 지식을 체계적으로 조직, 저장하고 이를 효율적으로 이용하도록 하는 방법의 연구
- 문제 영역이나 문제해결의 효율성을 위해 적절한 지식표현 방법을 선택하는 것이 매우 중요하다.

- 지식표현의 종류
 - 논리(**Logic**)
 - 의미망(**Semantic Net**)
 - 프레임(**Frame**)
 - 규칙(**Rule**)
 - 객체지향 표현기법(**Object-Oriented Representation**)

논리(Logic)

● 논리

- 수학, 논리학에서 사용된 명제논리나 서술논리 사용
“If x is a bird, then x has wings”란 사실(규칙)의 표현
⇒ $(\forall x) \{Is-a(x, Bird) \rightarrow has(x, Wings)\}$
⇒ 정형공식(wff: well-formed formular) 이해

□ 장점

- 수학적 근거를 바탕으로 논리개념을 자연스럽게 표현
- 지식의 정형화 영역에 적합(정리 증명 : **theorem proving**)
- 지식의 첨가와 삭제가 용이하고 단순

□ 단점

- 절차적, 결정적 지식표현이 어렵다.
- 사실의 구성법칙이 부족하므로 실세계의 복잡한 구조를 표현하기 어렵다.

□ 항(Term)과 기초공식(Atomic Formula)

- 항

- 1) 상수, 변수는 항
- 2) 함수 f 가 항 x 를 인자로 가지면 $f(x)$ 는 항
- 3) 1), 2)에 의해 구성되는 것은 모두 항

- 기초공식

- 항을 인자로 가지는 서술어(predicate)는 모두 기초공식이다.
- Ex) Woman(MARY), Married(father(JOHN), mother(JOHN))

□ 정형공식(Wff : well formed formular)

- 1) 기초공식 F 는 wff
- 2) F, G 가 wff면, $(F \vee G)$, $(F \wedge G)$, $\sim F$, $(F \rightarrow G)$, $(F \leftrightarrow G)$ 도 wff
- 3) F 가 wff면, $(\forall x)F$, $(\exists x)F$ 도 wff
- 4) 1), 2), 3)의 과정에 의해서만 구성되는 것은 모두 wff

비교흡수 부정

● 비교흡수 부정(Resolution Refutation)

- 두개의 기초절(부모절(parent clause)이라 함)에서

$$P_1 \vee P_2 \vee \dots \vee P_N \text{ 과 } \sim P_1 \vee Q_2 \vee \dots \vee Q_M$$

- 이 두개의 부모절(parent clause)을 논리합을 취해서 새로운 비교흡수절(resolvent)을 생성(비교흡수)

$$(P_1 \vee \sim P_1) \vee (P_2 \vee \dots \vee P_N \vee Q_2 \vee \dots \vee Q_M)$$

- 비교흡수 부정 방법

- 비교흡수 부정에서의 모든 절은 논리합으로만 된 정형공식
- 정형공식 집합 **S**에 특정 정형공식 **X**가 논리적으로 따름(logically follow)을 증명하기 위한 것
- $P = S \cup \{\sim X\} \Rightarrow P$ 에 대해 비교흡수 수행 \Rightarrow 비교흡수절 R_i 생성 $\Rightarrow P \cup R_i$ 에 대해 비교흡수 반복 \Rightarrow 모순(NIL)이 생성되면 종결

● 비교흡수부정의 이론

□ 가정: **S**에 **X**가 논리적으로 따른다고 가정

⇒ **S**를 만족하는 모든 해석은 **X**를 만족(**satisfy**)함

⇒ **S**를 만족하는 모든 해석은 **S** ∪ **{X}**를 만족함(**satisfiable**)

⇒ **S**를 만족하는 모든 해석은 **~X**를 만족하지 않음

⇒ **S**를 만족하는 모든 해석은 **S** ∪ **{~X}**를 불만족(**unsatisfiable**)

□ 즉, 불만족인 집합내에는 어떤 절 **C_i**에 대해, **C_i**와 **~C_i**가 공존하는 상황(모순)으로 볼 수 있다.

⇒ 어떤 해석이 주어져도 **C_i**와 **~C_i**를 동시에 만족할 수 없다

⇒ 이러한 절들에 대한 비교흡수는 결국 **NIL**(모순) 생성

⇒ 이는 **S**에 논리적으로 따르는 **X**를 부정(**~X**)한 결과는 모순을 초래하므로 결국 **X**는 **S**를 논리적으로 따른다

비교흡수를 위한 정형공식의 절 변환

- 1) Implication(\rightarrow) 제거 $\text{Ex}) A \rightarrow B \equiv \sim A \vee B$
- 2) Negation(\sim) 영역 축소 $\text{Ex}) \sim(A \vee B) \equiv \sim A \wedge \sim B$
- 3) 각 한정기호에 고유한 변수를 가지도록 변수 표준화
 $\text{Ex}) (\forall x)[P(x) \rightarrow (\exists x)Q(x)] \equiv (\forall x)[P(x) \rightarrow (\exists y)Q(y)]$
- 4) 존재한정기호(\exists) 제거
 - $(\forall y)[(\exists x)P(x, y)]$: x 는 y 에 종속되어 결정
 - x 를 y 에 대한 어떤 함수로 표현: $g(y)$: Skolem 함수
 - $(\forall y)[P(g(y), y)]$ 로 변환
- 5) Prenix 형으로 변환: 모든 전체한정기호(\forall)를 정형공식 앞으로 내어 영역을 전체공식에 미치도록 함
- 6) 정형공식을 논리곱 정규형으로 변환
 - $\text{Ex}) X_1 \vee (X_2 \wedge X_3) \equiv (X_1 \vee X_2) \wedge (X_1 \vee X_3)$
- 7) 전체한정기호를 모두 생략.
- 8) 논리곱을 생략. $\text{Ex}) X_1 \wedge X_2 \equiv \{X_1, X_2\}$
- 9) 각 절에서 같은 변수명이 없도록 조정

예)

$$(\forall x) \{P(x) \Rightarrow \{(\forall y)[P(y) \Rightarrow P(f(x, y))] \wedge \sim (\forall y)[Q(x, y) \Rightarrow P(y)]\}\}$$

$$1) (\forall x) \{\sim P(x) \vee \{(\forall y)[\sim P(y) \vee P(f(x, y))] \wedge \sim (\forall y)[\sim Q(x, y) \vee P(y)]\}\}$$

$$2) (\forall x) \{\sim P(x) \vee \{(\forall y)[\sim P(y) \vee P(f(x, y))] \wedge (\exists y)[Q(x, y) \wedge \sim P(y)]\}\}$$

$$3) (\forall x) \{\sim P(x) \vee \{(\forall y)[\sim P(y) \vee P(f(x, y))] \wedge (\exists w)[Q(x, w) \wedge \sim P(w)]\}\}$$

$$4) (\forall x) \{\sim P(x) \vee \{(\forall y)[\sim P(y) \vee P(f(x, y))] \wedge [Q(x, g(x)) \wedge \sim P(g(x))]\}\}$$

$$5) (\forall x)(\forall y) \{\sim P(x) \vee \{[\sim P(y) \vee P(f(x, y))] \wedge [Q(x, g(x)) \wedge \sim P(g(x))]\}\}$$

$$6) (\forall x)(\forall y) \{[\sim P(x) \vee \sim P(y) \vee P(f(x, y))] \wedge [\sim P(x) \vee Q(x, g(x))] \\ \wedge [\sim P(x) \vee \sim P(g(x))]\}$$

$$7) [\sim P(x) \vee \sim P(y) \vee P(f(x, y))] \wedge [\sim P(x) \vee Q(x, g(x))] \wedge [\sim P(x) \vee \sim P(g(x))]$$

$$8) \{[\sim P(x) \vee \sim P(y) \vee P(f(x, y))], [\sim P(x) \vee Q(x, g(x))], [\sim P(x) \vee \sim P(g(x))]\}$$

$$9) \{[\sim P(x_1) \vee \sim P(y) \vee P(f(x_1, y))], [\sim P(x_2) \vee Q(x_2, g(x_2))], [\sim P(x_3) \vee \sim P(g(x_3))]\}$$

● 실세계 문제

□ 알고있는 사실

- 읽을 수 있으면 글을 안다. $\rightarrow (\forall x)[R(x) \Rightarrow L(x)]$
- 돌고래는 글을 모른다. $\rightarrow (\forall x)[D(x) \Rightarrow \sim L(x)]$
- 어떤 돌고래는 지능이 있다. $\rightarrow (\exists x)[D(x) \wedge I(x)]$

□ 증명해야 할 사실

- 지능이 있는 어떤 동물은 읽을 수 없다. $\rightarrow (\exists x)[I(x) \wedge \sim R(x)]$

□ 정형공식의 절 변환

- $\sim R(x) \vee L(x), \sim D(y) \vee \sim L(y)$
- $(\exists x)[D(x) \wedge I(x)] \rightarrow x$ 가 종속되는 변수가 없다. \rightarrow Skolem 함수는 없음 \rightarrow Skolem 상수화 $\rightarrow x$ 를 임의의 상수 **A**로 대치 \rightarrow
 $D(A) \wedge I(A) \rightarrow D(A), I(A)$

□ 목표공식의 부정에서 목표절 생성

- $(\exists x)[I(x) \wedge \sim R(x)] \rightarrow$ 부정 $\rightarrow \sim(\exists x)[I(x) \wedge \sim R(x)] \rightarrow (\forall x)[\sim I(x) \vee R(x)]$
 $\rightarrow \sim I(x) \vee R(x) \rightarrow \sim I(z) \vee R(z)$

□ 비교흡수 부정의 적용

● 답의 유도

- 존재를 나타내는 변수가 무엇인가?
- 기초집합 **S**에 논리적으로 따르는 $(\exists x)W(x)$ 에서 **x**가 구체적으로 무엇인가를 유도

- 비교흡수 부정 방법을 이용한 답 유도과정
 - 1) 비교 흡수 부정과정에 의한 트리 생성
 - 2) 목표절의 **Skolem** 함수의 변수는 새로운 이름으로 대치
 - 3) 부정된 목표절과 이것의 부정된 절을 논리합하여 기초절에 추가 →
항진명제 → 기초절에 항상 참인 절을 추가해도 무관
 - 4) 1)의 트리를 바탕으로 수정된 증명 트리 생성
 - 5) 증명트리의 뿌리노드의 절이 답이 된다.

의미망

● 의미망(Semantic Network)

- 지식, 인간의 기억, 실세계를 망 구조로 표현
- 노드에는 객체, 개념, 사건 등을 표현
- 링크는 노드들간의 관계를 묘사
 - 구체, 절차, 인과, 부분 등의 객체 관계 표현
 - isa는 성질 계승(property inheritance) 링크

- 장점
 - 매우 복잡한 개념이나 인과 관계 표현에 용이
- 단점
 - 지식량이 커지면 복잡해짐 → 조작이 어려움

프레임

● 프레임(Frame)

- 의미망 한 종류로서 객체와 그 속성의 구조적 기술
- 프레임 객체 구조 내에 슬롯이라는 속성 묘사에 중점
- 데이터와 프로시저를 하나의 구조로 묶는다.
- 프레임들은 계층적으로 구성
- 슬롯(slot)은 객체의 속성과 속성값을 채우는 칸
 - 디폴트값, 프레임 포인터, 규칙, 프로시저
 - 프로시저는 슬롯 값 요구, 변경, 제거될 때 자동으로 작동되는 일종의 demon
- 패싯(facet)은 슬롯 값을 다양하게 줄 수 있는 키
 - Value, Default, Range, If-added, If-needed 등
- 프레임 표현
 - <프레임 이름>-<슬롯 이름>-<패싯 이름>-<값>

● 프레임 예

(FRAME Canary

(Is-a	(Value Bird))
(Color	(Value Yellow))
(Can	(Default Sing))
(Breed	(Range Africa India))
(Length	(If-added Calculate-width (If-removed Erase-width-weight))
(Width	(if-added Calculate-weight))
(Weight	(If-needed Calculate-weight)))

□ 장점

- 지식 표현이 일반적이고 자연스러우며 강력한 방법

□ 단점

- 복잡성 때문에 지식작성이 어렵다.

규칙

● 규칙(Rule)

- 가정(if-part, LHS)과 결론(then-part, RHS)의 문장으로 표현
Ex) If (A and B) then (C) $(A, B) \rightarrow (C)$
- 결정이나 결론이 요구되는 영역에 유용
- 규칙기반 시스템에서 사실집합과 규칙집합으로 구분되어 규칙의 가정부분이 사실집합의 일부와 부합될 때 규칙의 결론부분이 실행된다.

- 장점
 - 모듈화. 독립적으로 추가, 삭제 변경 용이
 - 특정 표현 방법에 따라 구조를 달리할 수 있다.
 - 결정, 결론이 요구되는 영역에 적합
- 단점
 - 문제풀이에 많은 경비 소요, 제어가 복잡

객체지향 개념

● 객체지향 (Object-Oriented) 개념

- 클래스와 객체, 인스턴스
- 클래스의 계층구조
- 계승, 다중계승, 재사용성
- 메시지, 메쏘드
- 캡슐화, 정보 은닉
- 객체 모델링
- 객체지향 언어의 장점
 - 현실세계의 개념적 개체는 단일 개념의 객체로 묘사 가능
 - 데이터 사이에 존재하는 일반화와 집단화를 쉽게 표현
 - **multimedia** 데이터 처리가 용이
 - 시스템 설계 및 구축시 생산성 향상
 - 동시 처리를 자연스럽게 지원
 - 편리한 사용자 인터페이스 지원

□참고문헌 및 강의자료 제공

인공지능: 개념 및 응용 Artificial Intelligence: Concepts and Applications (2001년 2월 개정판)