Knowledge Representation

05026010 Intelligent Systems

Dr. Anantaporn Srisawat

Outline

- การแสดงความรู้ (Knowledge Representation)
- เพรดิเคตโลจิก (Predicate Logic)
- ~~กฎการอนุมาน (Rule of inference)~~
- ~~การแทนค่าและการทำให้เท่ากัน (Substitution and Unification)~~
- รีโซลูชัน (Resolution)
- การปฏิเสธแบบรีโซลูชัน (Resolution Refutation)

Knowledge Representation

- เป็นวิธีการในการแสดงความรู้ในการแก้ปัญหาให้อยู่ในรูปแบบที่คอมพิวเตอร์สามารถนำไปประมวลผลได้
- อยู่ในรูปของประโยคและสอดคล้องกันในแง่ของไวยากรณ์ (Syntax) และความหมาย (Semantic)
- ตัวอย่างปัญหา Water Jug
  - ไวยากรณ์ คือ ส่วนเงื่อนไข → ส่วนข้อสรุป เช่น (X<4, Y) → (4,y)
  - แสดงความหมาย เช่น (4, 3)

Knowledge Representation

- ความรู้มีความซับซ้อน ไม่ใช่ตัวเลข
- เป็นวัตถุ (object) และความจริง (fact) ที่มีความสัมพันธ์กัน
- เช่น ความรู้ "Book is on the table"
  - มี object 2 ตัว คือ book และ table
  - มี on แสดงถึงความสัมพันธ์กัน
  - ON(book, chair) : Book is on the chair
  - IN(chair, room) : Chair is in the room
  - UNDER(chair, window) : Chair is under the window

Predicate Logic

- predicate logic
- ไวยากรณ์ของ predicate และความหมาย
- ตัวเชื่อมทางตรรกศาสตร์
- ตัวบ่งปริมาณ

predicate logic

- คือข้อความทางตรรกศาสตร์ที่แทนความรู้โดยอาศัยหลักทฤษฎีทางตรรกศาสตร์
- มีชื่อเรียกอื่น คือ predicate calculus
- การใช้ predicate logic ในการแก้ปัญหาจะใช้ความเป็นเหตุผลที่เลียนแบบความคิดของมนุษย์ในการค้นหาคำตอบของปัญหา
- ขั้นตอนในการทำงานของหลักตรรกศาสตร์
  - เปลี่ยนข้อเท็จจริงให้อยู่ในรูปแบบของข้อความทางตรรกศาสตร์หรือ predicate logic แล้วนำไปเก็บเป็นฐานความรู้
  - สร้างกฎการอนุมานทางตรรกศาสตร์ เพื่อแก้ปัญหา

ไวยากรณ์ของ predicate และความหมาย

- การแทนประโยคด้วย predicate logic สามารถใช้อักขระได้ดังนี้
  - ตัวอักษรภาษาอังกฤษได้แก่ A-Z และ a-z
  - ตัวเลขตั้งแต่ 0-9
  - เครื่องหมาย '_' (underscore)
- ส่วนประกอบของ predicate logic
  - สัญลักษณ์แสดงเพรดิเคต (predicate symbols) ใช้สายอักขระตั้งแต่ 1 ตัวขึ้นไปและขึ้นต้นด้วยตัวอักษรใหญ่ เช่น P, Q, R
  - สัญลักษณ์แสดงตัวแปร (variable symbols) ใช้สายอักขระตั้งแต่ 1 ตัวขึ้นไปและขึ้นต้นด้วยตัวอักษรเล็ก เช่น x, y, z
  - สัญลักษณ์แสดงฟังก์ชัน (function symbols) ใช้สายอักขระตั้งแต่ 1 ตัวขึ้นไปและขึ้นต้นด้วยตัวอักษรเล็ก เช่น f, g, h
  - สัญลักษณ์แสดงค่าคงที่ (constant symbols) ใช้สายอักขระตั้งแต่ 1 ตัวขึ้นไปและขึ้นต้นด้วยตัวอักษรใหญ่ เช่น A, B, C

ไวยากรณ์ของ predicate และความหมาย

- การแทนประโยคหรือความรู้ด้วยเพรดิเคตที่เหมาะสมจะเรียกว่า well form formulas wff(s)
- Well form formulas คือเพรดิเคตที่สามารถนำไปหาความสัมพันธ์ระหว่างเพรดิเคตที่เกี่ยวข้องกัน
- ตัวอย่างของการแปลงประโยคต่างๆให้อยู่ในรูปของเพรดิเคต
  - Macus is a man. แทนด้วยเพรดิเคต **Marcusman**
  - Socrates is a man. แทนด้วยเพรดิเคต **Socratesman**
  - ความสัมพันธ์ของทั้ง 2 เพรดิเคตนี้คืออักขระ 3 ตัวสุดท้ายต้องเป็น man
  - ถ้ามีเพรดิเคตใหม่ **Plutoman** จะหมายความว่า "Pluto is a man."
  - ถ้ามีประโยค "Susan is a woman." แทนด้วยเพรดิเคต Susanwoman
  - หรืออาจจะได้ว่า Susanwo is a man.

ไวยากรณ์ของ predicate และความหมาย

- การแทนเพรดิเคตด้วยวิธีนี้ไม่เป็น well form formulas (wffs) เนื่องจากการตีความหมายของเพรดิเคตนั้นไม่ชัดเจน จึงควรกำหนดเพรดิเคตใหม่ ดังนี้
- Marcus is a man. แทนด้วยเพรดิเคต MAN(Marcus)
- Socrates is a man. แทนด้วยเพรดิเคต MAN(Socrates)
- Pluto is a man. แทนด้วยเพรดิเคต MAN(Pluto)
- Susan is a woman. แทนด้วยเพรดิเคต WOMAN(Susan)
- x+y แทนด้วยเพรดิเคต PLUS(x,y)
- x is greater than 5. แทนด้วยเพรดิเคต GREATER(x,5)

ไวยากรณ์ของ predicate และความหมาย

FATHER(SOMCHAI, SOMYING)

- เรียกว่า *สูตรอะตอม (atomic formular)*
- เป็น wffs ที่เล็กที่สุด
- FATHER คือ สัญลักษณ์เพรดิเคต ที่แสดงความสัมพันธ์ "พ่อ"
- มีอาร์กิวเมนต์ (argument) 2 ตัว คือ 'SOMCHAI' และ 'SOMYING' ซึ่งเป็นค่าคงที่
- อาร์กิวเมนต์แต่ละตัวต้องคั่นด้วยเครื่องหมาย ',' และถูกคลุมด้วยวงเล็บ
- ความหมายคือ SOMCHAI เป็นพ่อของ SOMYING
- สามารถใช้ตัวแปรเพื่อแทนถึงค่าคงที่ใดๆ ได้ เช่น FATHER(x,y) เมื่อ x และ y แทนตัวแปร ซึ่งหมายความว่า x เป็นพ่อของ y

ไวยากรณ์ของ predicate และความหมาย

HAS_MONEY(SOMCHAI, salary(SOMCHAI))

- HAS_MONEY คือ สัญลักษณ์เพรดิเคต ที่แสดงความสัมพันธ์ "มีเงิน"
- Salary เป็นฟังก์ชัน ที่ระบบหาพจน์จาก 'SOMCHAI' ซึ่งพจน์นี้อาจเป็นค่าคงที่ค่าหนึ่ง เช่น 20,000
- ถ้ากำหนดให้ salary(SOMCHAI)= 20,000 สูตรอะตอมนี้จะหมายความว่า "สมชายมีเงินเท่ากับ 20,000 บาท"

ตัวเชื่อมทางตรรกศาสตร์ (Logical Connectives)

- เป็นสิ่งที่ใช้สำหรับการเชื่อมประโยคทางตรรกศาสตร์เข้าด้วยกัน
- ตัวเชื่อมทางตรรกศาสตร์ที่ใช้ทั่วไป ได้แก่

ตัวเชื่อม	ความหมาย	สัญลักษณ์
NOT	นิเสธ (ไม่)	~ tild
AND	และ	$\wedge$
OR	หรือ	$\vee$
IMPLICATION	ถ้า...แล้ว	$\Rightarrow$

ตารางค่าความจริง

P	Q	$P \vee Q$	$P \wedge Q$	$P \Rightarrow Q$	~P
T	T	T	T	T	F
F	T	T	F	T	T
T	F	T	F	F	F
F	F	F	F	T	T

ตัวเชื่อมทางตรรกศาสตร์ (Logical Connectives)

- John did not read a book.
  - $\sim$READ(JOHN,BOOK)
- Bobby is not a man.
  - $\sim$MAN(BOBBY)
- John lives in a blue house.
  - LIVE(JOHN, HOUSE1)$\wedge$COLOR(HOUSE1,BLUE)
- My house is brown and big.
  - BROWN(MY_HOUSE)$\wedge$BIG(MY_HOUSE)
- Tom buys a book or a video.
  - BUY(TOM, BOOK)$\vee$BUY(TOM,VIDEO)
- If the car belongs to John then it is black.
  - OWNS(JOHN,CAR1)$\rightarrow$COLOR(CAR1,BLACK)

ตัวบ่งปริมาณ (Quantifier)

- ตัวบ่งปริมาณใน predicate logic มี 2 แบบ คือ
  - ตัวบ่งปริมาณเอกภพ (Universal quantifier) หมายถึง for all แทนด้วยเครื่องหมาย$\forall$
  - ตัวบ่งปริมาณมีอยู่ (Existential quantifier) หมายถึง for some แทนด้วยเครื่องหมาย $\exists$
- Any basketball players are tall.
  - $\forall$x(BASKETBALL_PLAYER(x)$\rightarrow$TALL(x))
- All elephant are gray.
  - $\forall$x(ELEPHANT(x)$\rightarrow$COLOR(x,GRAY)) $\sim\exists x$
- There is a person who like durian.
  - $\exists$x(PERSON(x)$\rightarrow$LIKE(x,DURIAN)) $\forall x$
- Nobody likes taxes.
  - $\sim\exists$x(LIKE(x,TAXES))

ตัวบ่งปริมาณ (Quantifier)

- การหาค่าความจริงของ wffs ที่มีตัวบ่งปริมาณปรากฏอยู่อาจหาค่าความจริงไม่ได้
- $\forall$x(P(x)) ให้โดเมนของ x เป็นเลขจำนวนจริง
  - ไม่สามารถสรุปได้ เนื่องจากไม่สามารถนำเลขจำนวนจริงมาทดสอบกับสูตรนี้ได้จนหมด
  - ถ้าจำนวนจริงที่นำมาทดสอบยังให้ค่าเป็นจริงอยู่ก็ยังคงสรุปค่าความจริงไม่ได้
  - แต่สามารถสรุปได้ว่าสูตรนี้เป็นเท็จ ถ้าสามารถหาเลขจำนวนจริงตัวใดตัวหนึ่งมาทำให้สูตรเป็นเท็จ
- $\exists$x(P(x)) ให้โดเมนของ x เป็นเลขจำนวนจริง
  - ถ้าสามารถหาจำนวนจริงมาอย่างน้อยหนึ่งตัวที่ทำให้สูตรนี้เป็นจริง แสดงว่าสรุปได้ว่าสูตรนี้มีค่าความจริงเป็นจริง
- First-order predicate logic คือเพรดิเคตที่ไม่มีตัวบ่งปริมาณของสัญลักษณ์แสดงเพรดิเคตหรือสัญลักษณ์แสดงฟังก์ชัน

กฎการอนุมาน (Rule of inference)

- การอนุมาน คือการนำความรู้ที่เขียนอยู่ในรูป predicate logic มาหาข้อเท็จจริงใหม่หรือผลสรุปที่แฝงอยู่ในความรู้นั้นได้
- เรียกกฎที่ใช้อนุมานเพื่อหาความรู้ใหม่ว่า *กฎการอนุมาน*

กฎโมดัสโพเน็นส์ (Modus Ponens): W1 $\Rightarrow$ W2

W1

W2

กฎเจาะจงตัวแปรเอกภพ (Universal Specialization): $\forall$x (W(x))

W(A)

- เรียกสูตรใหม่ที่เกิดขึ้นว่า *ทฤษฎี (theorem)* และลำดับของกฎการอนุมานที่ใช้ในการสร้างทฤษฎีเรียกว่า *การพิสูจน์ (proof)* ของทฤษฎีนั้น

กฎการอนุมาน (Rule of inference)

- จากสูตร 2 ตัวคือ $\forall x\ (W1(x) \Rightarrow W2(x))$ และ W1(A)
- สามารถอนุมานได้ว่า W2(A) เป็นจริงถ้าสูตร 2 ตัวบนเป็นจริง
- ทฤษฎีคือ W2(A) ส่วนการพิสูจน์คือลำดับของกฎการอนุมาน ดังนี้

ใช้กฎเจาะจงตัวแปรเอกภพ	$\underline{\forall x\ (W1(x) \Rightarrow W2(x))}$
ได้ว่า	$W1(A) \Rightarrow W2(A)$
จากนั้นใช้กฎโมดัสโพเน็นส์	$\underline{W1(A)}$
ได้ว่า	W2(A)

- ปัญหาคือ จะรู้ได้อย่างไรว่าจะต้องแทนตัวแปรด้วยค่าคงที่ตัวใด จึงจะทำให้สามารถอนุมานต่อไปได้
- ถ้าแทนตัวแปร x ด้วยค่าคงที่ B จะไม่สามารถอนุมานต่อไปได้

การแทนค่า (Substitution)

- *การแทนค่า (substitution)* คือการแทนพจน์ *(term)* ให้กับตัวแปร
- *พจน์* หมายถึง ค่าคงที่ ฟังก์ชันและตัวแปร
- สูตรที่ได้จากการทำการแทนค่าพจน์ในตัวแปรของสูตรใดๆ เรียกว่า *ตัวอย่างการแทนค่า (substitution instance)* ของสูตรนั้นๆ
- ตัวอย่าง สูตร P(x, f(y), B) สามารถสร้างตัวอย่างการแทนค่าได้เป็น
  - ตัวอย่างการแทนค่า 1: P(z, f(w), B)
  - ตัวอย่างการแทนค่า 2: P(C, f(A), B)
- การแทนค่าสามารถเขียนให้อยู่ในรูปของเซตคู่ลำดับ ดังนี้

$$s=\{t_1/v_1, t_2/v_2, \ldots, t_n/v_n\}$$

โดยที่คู่ลำดับ t_i/v_i หมายถึงพจน์ t_i ถูกแทนค่าให้กับตัวแปร v_i

การแทนค่า (Substitution)

- ตัวอย่าง สูตร P(x, f(y), B) สามารถสร้างตัวอย่างการแทนค่าได้เป็น
  - ตัวอย่างการแทนค่า 1: P(z, f(w), B)
  - ตัวอย่างการแทนค่า 2: P(C, f(A), B)
- เมื่อใช้การแทนค่า s1 และ s2 ในตัวอย่างข้างต้น จะได้ว่า

  s1={z/x, w/y}

  s2={C/x, A/y}
- สามารถเขียนสูตรที่ได้จากการแทนค่า s กับสูตร E ด้วย Es ดังนี้

  P(z, f(w), B) = P(x, f(y), B)s1

  P(C, f(A), B) = P(x, f(y), B)s2

การทำให้เท่ากัน (Unification)

- สูตร E1 และ E2 สามารถ*ทำให้เท่ากัน (unify)* ถ้ามีการแทนค่า s ที่ทำให้ E1s=E2s และในกรณีนี้จะเรียก s ว่าเป็น *ตัวทำให้เท่ากัน (unifier)* ของ E1 และ E2
- P(x, f(y), B) และ P(x, f(B), B)
  - ทำให้เท่ากันโดยกำหนดให้ s={A/x, B/y} เป็น unifier
  - ผลของการทำให้เท่ากันคือ P(A, f(B), B)
- สูตร 2 ตัวใดๆ จะมี unifier มากกว่าหนึ่งตัว
- แต่ที่สนใจคือ unifier ที่ใช้แทนค่าไม่มากเกินความจำเป็น ซึ่งเรียก unifier นี้ว่า *mgu (most general unifier)*
- g เป็น *mgu* ของ E1 และ E2 ก็ต่อเมื่อ ถ้ามี s ที่เป็น unifier อื่นของ E1 และ E2 แล้ว จะต้องมีตัว unifier s' ที่ทำให้ E1s=E1gs' และ E2s=E2gs'
- mgu ของ P(x, f(y), B) และ P(x, f(B), B) คือ {B/y}

การทำให้เท่ากัน (Unification)

```
Algorithm: Unify(L1,L2)

2. IF L1 or L2 are both variables or constants THEN
     IF L1=L2 THEN return NIL
     ELSE IF L1 is a variable THEN
          IF L1 occurs in L2 THEN return {FAIL} ELSE
            return {L2/L1}
     ELSE IF L2 is a variable THEN
          IF L2 occurs in L1 THEN return {FAIL} ELSE
            return {L1/L2}
     ELSE return {FAIL}
3. IF the predicate or function symbols of L1 and L2
     are not identical
   THEN return {FAIL}
4. IF L1 and L2 have a different number of arguments
   THEN return {FAIL}
5. SUBST := NIL
6. FOR i := 1 TO number of arguments in L1 DO
   5.1 S := Unify(ith argument of L1, ith argument of L2)
   5.2 IF S contains FAIL THEN return {FAIL}
   5.3 IF S <> NIL THEN
          5.3.1 Apply S to the remainder of both L1 and
                L2
          5.3.2 SUBST := SUBST ∪  S
6. return SUBST
```


การทำให้เท่ากัน (Unification)

- กำหนดให้ L1=P(A, x, h(g(z))) และ L2=P(z, h(y), h(y)) ต้องการทำให้ L1=L2 โดยเรียกใช้อัลกอริทึม Unify(L1,L2)

การทำให้เท่ากัน (Unification)

- กำหนดให้ L1=P(A, x, h(g(z))) และ L2=P(z, h(y), h(y)) ต้องการทำให้ L1=L2 โดยเรียกใช้อัลกอริทึม Unify(L1,L2)

Resolution

- Resolution เป็นกฎการอนุมานที่ใช้กับสูตรทุกตัวที่เป็นอนุประโยค (clause) ที่นิยมใช้ได้อย่างกว้างขวาง
- อนุประโยค คือสูตรอะตอมหลายตัวที่มาเชื่อมกันด้วยเครื่องหมาย ∨ และสูตรอะตอมบางตัวอาจมีเครื่องหมายนิเสธอยู่หรือไม่ก็ได้
- เช่น $P(x) \vee Q(x,y) \vee \sim R(A)$
- ก่อนที่จะทำการรีโซลูชัน ต้องทำการแปลงสูตรต่างๆ ให้อยู่ในรูปของอนุประโยคก่อน

การแปลงสูตรให้เป็นอนุประโยค

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

1. เปลี่ยนรูปของ $X \Rightarrow Y$ ให้เป็น $\sim X \vee Y$

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

2. ลดขอบเขตของนิเสธให้ครอบคลุมน้อยที่สุด
   - ทำการกระจายนิเสธเข้าไปข้างในบริเวณที่มันคลุมอยู่ โดยมีกฎดังนี้
     - $\sim(\sim P)=P$
     - $\sim(P\vee Q)= \sim P \wedge \sim Q$
     - $\sim(P\wedge Q)= \sim P \vee \sim Q$
     - $\sim((\forall x)P(x))=(\exists x)(\sim P(x))$
     - $\sim((\exists x)P(x))=(\forall x)(\sim P(x))$

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

การแปลงสูตรให้เป็นอนุประโยค

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

3. เปลี่ยนชื่อตัวแปรตามขอบเขตของตัวบ่งปริมาณ
   - ถ้าพบว่ามีตัวแปรซ้ำกันให้เปลี่ยนชื่อตัวใดตัวหนึ่ง

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

4. กำจัดตัวบ่งปริมาณมีอยู่
   - แทนค่าตัวแปรด้วยฟังก์ชันสคอเล็ม (skolem function) ซึ่งเป็นฟังก์ชันที่แทนค่าตัวแปรตัวหนึ่งด้วยฟังก์ชันของตัวแปรอื่นๆ ที่ตัวแปรนั้นขึ้นอยู่กับมัน
   - $(\exists a)P(a)$ จะได้ $P(c)$
   - $(\forall a)\{(\exists b)P(a,b)\}$ จะได้ $(\forall a)[P(a,f(a))]$
   - $(\exists a)(\forall b)(\forall c)((\exists d)P(a,b,c,d)$ จะได้ $(\forall b)(\forall c)((\exists d)P(x,b,c,f(b,c))$

"Everybody likes something"

$\forall(x)\exists(y)[Person(x)\wedge Likes(x,y)]$

$\forall(x)[Person(x)\wedge Likes(x,f(x))]$

Every ∃ quantified variable can be replace by Skolem Function

ยกมา

การแปลงสูตรให้เป็นอนุประโยค

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

4. กำจัดตัวบ่งปริมาณมีอยู่ (ต่อ)

$(\forall x)\{\ldots(\exists w)[Q(x,w)\wedge\sim P(w)]$ อ่านได้ว่าสำหรับ x ทุกตัวจะมี w บางตัวที่ทำให้ $Q(x,w)$ และ $\sim P(w)$ เป็นจริง หมายความว่า w ขึ้นอยู่กับ x หรือเป็นฟังก์ชันของ x ดังนั้น $w=g(x)$ เมื่อ g เป็นฟังก์ชันสคอเล็ม

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

5. แปลงให้อยู่ในรูปแบบพรีเน็กซ์ (prenex form) คือย้ายตัวบ่งปริมาณเอกภพทุกตัวมาอยู่หน้าสุด

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

การแปลงสูตรให้เป็นอนุประโยค

$1 \vee (2 \wedge 3)$

$(\forall x)(\forall y)\{\underbrace{\sim P(x)}_{1}\vee\{\underbrace{[\sim P(y)\vee P(f(x,y))]}_{2}\wedge\underbrace{[Q(x,g(x))\wedge\sim P(g(x))]}_{3}\}\}$

6. จัดรูปของพรีเน็กซ์ให้อยู่ในรูป conjunctive normal form
   - รูปที่สูตรทุกตัวเชื่อมกันด้วยเครื่องหมาย $\wedge$ แต่ภายในสูตรมีแต่เครื่องหมาย $\vee$ โดยใช้ความสมมูลของสูตร $P\vee(Q\wedge R)=(P \vee Q)\wedge(P\vee R)$

$(1\vee 2)\wedge(1\vee 3)$

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

7. ตัดตัวบ่งปริมาณเอกภพออกให้หมด
8. แยกแต่ละสูตรที่เชื่อมด้วย $\wedge$ ออกจากกันเป็นอนุประโยคต่างๆ

(1) $\sim P(x)\vee\sim P(y)\vee P(f(x,y))$

(2) $\sim P(x)\vee Q(x,g(x))$

(3) $\sim P(x)\vee\sim P(g(x))$

การแปลงสูตรให้เป็นอนุประโยค

(1) ~P(x)∨~P(y)∨P(f(x,y))

(2) ~P(x)∨Q(x,g(x))

(3) ~P(x)∨~P(g(x))

9. เปลี่ยนชื่อตัวแปรที่ซ้ำกันระหว่างอนุประโยค

(1) ~P(x1)∨~P(y)∨P(f(x1,y))

(2) ~P(x2)∨Q(x2,g(x2))

(3) ~P(x3)∨~P(g(x3))

Resolution

P1 ∨ P2 ∨ ··· ∨ Pn	} parent clauses
~P1 ∨ Q2 ∨ ··· ∨ Qm	
P2 ∨ ··· ∨ Pn ∨ Q2 ∨ ··· ∨ Qm	← resolvent

ตัวอย่างของการทำรีโซลูชัน

Parent clause	Resolvent
P และ ~P ∨Q	Q
P ∨Q และ ~P ∨ Q	Q
P ∨Q และ ~P ∨ ~Q	Q ∨ ~Q และ P ∨~P
~P และ P	NIL
~P ∨ Q และ ~Q ∨ R	~P ∨ R

Resolution Refutation

- การพิสูจน์แบบหักล้าง (resolution refutation) เป็นการหาค่าความจริงของคำตอบ
- เริ่มจากสมมติให้ค่าความจริงของเพรดิเคตคำตอบมีค่าเป็นปฏิเสธ
- นำเพรดิเคตนั้นไปหักล้างกับข้อเท็จจริงที่มีอยู่ในฐานความรู้ด้วยวิธีรีโซลูชัน
- ถ้าเพรดิเคตคำตอบสามารถหักล้างข้อเท็จจริงต่างๆ จนไม่เกิด resolvent แสดงว่าเพรดิเคตที่ต้องการพิสูจน์มีค่าความจริงเป็นจริง
- ขั้นตอนในการทำ resolution refutation
  1. แปลงสูตรต่างๆ ให้อยู่ในรูปอนุประโยค (กฎ) จริง
  2. เปลี่ยนสูตรหรือเพรดิเคตที่ต้องการพิสูจน์ให้อยู่ในรูปของนิเสธและเพิ่มเข้าไปเป็นกฎใหม่
  3. ทำ resolution กับอนุประโยคต่างๆ รวมทั้งกฎใหม่

Resolution Refutation: Example

- กำหนดข้อเท็จจริงดังนี้
  1. All birds are animals.
  2. Tweety is a bird.
  3. All animals die.
- ต้องการหาว่า Tweety dies or not.
- เปลี่ยนข้อเท็จจริงให้อยู่ในรูปของเพรดิเคต

case

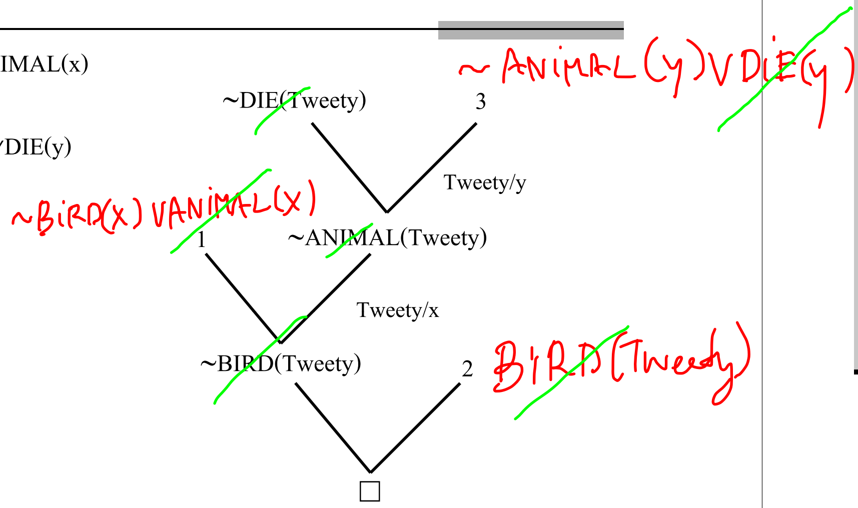
1. (∀x)BIRD(x) ⟹ ANIMAL(x) — ~BIRD(x)∨ANIMAL(x)
2. BIRD(Tweety) — BIRD(Tweety)
3. (∀y)ANIMAL(y) ⟹DIE(y) — ~ANIMAL(y)∨DIE(y)
4. DIE(Tweety) — ~DIE(Tweety)

Q

อนุประโยค จากจริง

Resolution Refutation: Example

1. ~BIRD(x)∨ANIMAL(x)
2. BIRD(Tweety)
3. ~ANIMAL(y)∨DIE(y)
4. ~DIE(Tweety)



Resolution Refutation: Example

- Given facts:
  1. Caesar was a man.
  2. Caesar was a Roman.
  3. Caesar was born in 50 A.D.
  4. All men are bachelors.
  5. All Romans died when earthquake occurred in 81 A.D.
  6. No bachelor lives longer than 100 years.
  7. Now it is 2010.
- ต้องการหาคำตอบ Is Caesar dead now?

Your H/W

เอกสารอ้างอิง

- เอกสารประกอบการสอนวิชาปัญญาประดิษฐ์ โดย ศ.ดร.บุญเสริม กิจศิริกุล
- หนังสือ "ปัญญาประดิษฐ์" โดย รศ.ดร. บุญเจริญ ศิริเนาวกุล