EXPERT SYSTEMS
Biểu diễn tri thức

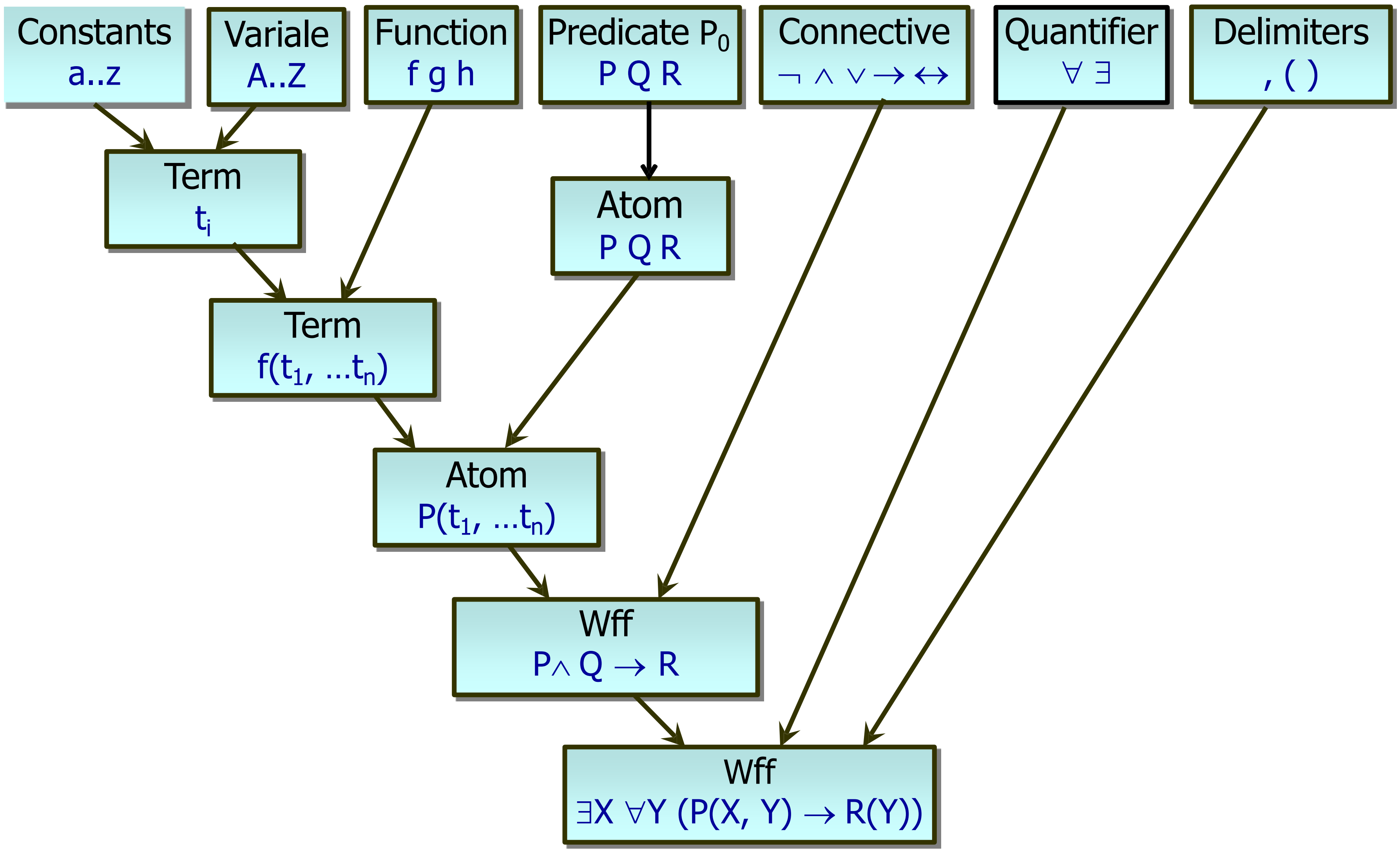
Khái niệm

– Ngôn ngữ vị từ bậc một (first–order predicate language) sử dụng một bảng ký hiệu đặc biệt và các khái niệm.

- hạng (term)
- nguyên tử (atom)
- trực kiện (literal) hay mệnh đề
- công thức chuẩn (WFF well formed formula)

để xây dựng các biểu thức đúng (correct expressions)

Alphabet



Constants
a..z
Variale
A..Z
Function
f g h
Predicate P_0
P Q R
Connective
$\neg \wedge \vee \rightarrow \leftrightarrow$
Quantifier
$\forall \exists$
Delimiters
, ( )
Term
t_i
Atom
P Q R
Term
$f(t_1, \ldots t_n)$
Atom
$P(t_1, \ldots t_n)$
Wff
$P \wedge Q \rightarrow R$
Wff
$\exists X \forall Y (P(X, Y) \rightarrow R(Y))$

Bảng ký hiệu (Alphabet)

- Bảng ký hiệu để xây dựng các biểu thức đúng gồm :
  - Các *dấu phân cách* (separator signs) :

    dấu phẩy (**,**), dấu mở đóng ngoặc **(   )**
  - Các *hằng* (constant) :

    có dạng chuỗi sử dụng các chữ cái in thường **a..z**

    Ví dụ : a, block
  - Các *biến* (variable) :

    có dạng chuỗi sử dụng các chữ cái in hoa **A..Z**

    Ví dụ : X, NAME
  - Các *vị từ* (predicate) :

    được viết tương tự các *biến*, sử dụng các chữ cái in hoa **A..Z**

    Ví dụ : ISRAINING, ON(table), P(X, blue), BETWEEN(X, Y, Z)

    – Khi cần thao tác trên một vị từ nào đó, cần phải ghi rõ bậc (arite) hay số các đối (argument) của vị từ đó.

    – Khi bậc có giá trị cố định là 0, vị từ còn được gọi là mệnh đề (proposition). E.g. ISRAINING, EMPTY là các mệnh đề.

Các hàm (function)

- Các hàm :
  - có cách viết tương tự các hằng
  - sử dụng các chữ in thường **a..z**
  - Mỗi hàm có bậc (hay số lượng các đối) cố định, là một số nguyên dương

Ví dụ :

  - f(X), weight(elephan), successor(M, N)
    là các hàm có bậc lần lượt là 1, 1, và 2

Người ta quy ước rằng :

  - Các hằng là những hàm bậc không (nil)
  - Ví dụ : a, elephan, block là các hằng

Function Symbols

- Function symbols
  - function_name(arg_1, arg_2, …, arg_n)
  - Identifies the object referred to by a tuple of objects
  - May be defined implicitly through other functions, or explicitly through tables
- Function names begin with a lowercase letter or are expressed with a symbol
  - $F = \{f, g, h, \ldots\} = F_0 \cup F_1 \cup F_2 \cup \ldots$
- Function arities:
  - F_0: function symbols of arity 0 (constants): a, b, max, jim
  - F_1: function symbols of arity 1 (one argument)
  - F_2: function symbols of arity 2 (two arguments)
  - ...

Functions Examples

- A function is used to express complex names
  - age(max) — Max's age
  - password(claire) — Claire's password
- A function may be nested
  - Max's age's double — double(age(max))
  - father(mother(max)) — Max's mother's father
  - starship(son(dr_crusher)) — Dr_Crusher's son's starship
- A function is never a predicate
  - Can't nest predicates — TALL(TALL(max))
- Function symbols of arity >1
  - youngestChild(max, ann) — Max and Ann's youngest child
  - *(5, +(2, 4)) — 30
- A predicate forms a sentence, while a function names an individual

Bảng ký hiệu (Alphabet)

- Các phép nối logic (logical connector) :

  $\neg$, $\wedge$, $\vee$, $\rightarrow$ và $\leftrightarrow$
  tương ứng với các phép phủ định, và, hoặc, kéo theo và kéo theo lẫn nhau (tương đương)

- Các dấu lượng tử (quantifier)

  $\exists$ lượng tử tồn tại (existential quantifier)

  $\forall$ lượng tử toàn thể (universal quantifier)

Hạng/hạng tử (term)

- Hạng được tạo thành từ hai luật sau :
  - Các hằng và các biến là các hạng
  - Nếu f là một hàm có bậc $n \geq 1$
    và nếu $t_1, \ldots, t_n$ đều là các hạng,
    thì hàm $f(t_1, \ldots, t_n)$ cũng là một hạng
- Ví dụ các hàm sau đây đều là các hạng :
  - successor(X, Y), weight(b), successor(b, wight(Z))
- Nhưng các hàm sau đây không phải là hạng :
  - P(X, blue) vì P là vị từ
  - weight (P(b))
    vì P(b) không phải là hạng (vị từ không làm đối cho hàm)

Nguyên tử (atom)

- Nguyên tử được tạo thành từ hai luật sau :
  - Các mệnh đề (vị từ bậc 0) là các nguyên tử
  - Nếu P là một vị từ bậc n ($n \geq 1$) nếu $t_1, \ldots, t_n$ đều là các hạng, thì $P(t_1, \ldots, t_n)$ cũng là một nguyên tử
- Ví dụ các vị từ sau đây là các nguyên tử :
  - P(X, blue), EMPTY, BETWEEN(table, X, sill(window))
- Còn :
  - successor (X, Y, sill (window)) không phải nguyên tử, mà là các hàm

Câu nguyên tử- Atomic Sentence

- A atomic sentence:
  - Expresses a claim that is either true or false
  - Formed by a single predicate followed by one or more arguments
- Example:
  - Max is tall

    TALL(max)
  - A is larger than B

    LARGER(A, B)
  - B is not larger than A

    ¬LARGER(B, A)
  - C is smaller than D, or D is not smaller than C SMALLER(C, D), ¬SMALLER(D, C)
  - A is between B and E:

    BETWEEN(A, B, E)

Vị từ- Predicates

- Predicate symbols :
  - PREDICATE(arg_1, arg_2, …, arg_n)
  - A (determinate) property possessed by an object: Shape, Size
  - A (determinate) relationship among objects:
    Shape relationship, size relationship, positional relationship…
  - The number of arguments is called the predicate's arity
  - The order of the arguments is important
- Predicates have names beginning with an uppercase letter or are represented by an operator symbol
  - $P = P_0 \cup P_1 \cup P_2 \cup \ldots$
- Predicate arities:

  P_0: predicate symbols of arity 0 (constants: proposition) : P, Q, R, …

  P_1: predicate symbols of arity 1 (one argument)

  P_2: predicate symbols of arity 2 (two arguments) **…**

Công thức chỉnh

- Các công thức chỉnh (WFF-well formed formula) được tạo thành từ ba luật sau :
  - Các nguyên tử là các WFF.
  - Nếu G và H biểu diễn các WFF, thì $(\neg G)$, $(G \wedge H)$, $(G \vee H)$, $(G \rightarrow H)$ và $(G \leftrightarrow H)$ cũng là các WFF do được tạo thành từ các phép nối logic giữa G và H.
- Nếu G là WFF và X là một biến, thì $(\exists X)G$ và $(\forall X)G$ cũng là các WFF.

  $(\exists X)G$ được đọc là tồn tại một biến X sao cho G được thoả mãn.

  $(\forall X)G$ được đọc là với mọi biến X thì G đều được thoả mãn.

- ***Ví dụ*** **:**

  Các công thức sau đây là chỉnh :

  $(\exists X)\ (\forall Y)\ ((P(X, Y) \wedge\ Q(X, Y) \rightarrow R(X))$

  $((\neg(P(a) \rightarrow P(b))) \rightarrow \neg P(b))$

  còn $(\neg(f(a))$: *phủ định của một hàm*,

  và f $(P(a))$: *hàm có đối là một vị tư*, đều không phải là *WFF*

Well-formed Formula (WFF)

- Any atomic sentence is a WFF
- If A are B are wffs then so are

  $\neg A$

  $A \wedge B$

  $A \vee B$

  $A \rightarrow B$

  $A \leftrightarrow B$

- B is a cube or B is large (a large cube):

  CUBE(B) $\vee$ LARGE(B)

- E and C are in the same row and E is in back of B:

  SAMEROW(E, C) $\wedge$ BACKOF(E, B)

Ví dụ

- Các công thức sau đây là chỉnh :
  - $(\exists X)\ (\forall Y)\ ((P(X, Y) \vee Q(X, Y) \rightarrow R(X))$
  - $((\neg(P(a) \rightarrow P(b))) \rightarrow \neg P(b))$
- Còn các công thức sau đây không chỉnh :
  - $(\neg(f(a))$: phủ định của một hàm,
  - $f\ (P(a))$: hàm có đối là một vị từ
- Chú ý :
  - WFF được gọi là một trực kiện (literal) hay trị đúng nếu nó là một nguyên tử hay có dạng $(\neg G)$, với G là một nguyên tử
  - Trong một WFF, trước hoặc sau các ký tự nối, ký tự phân cách, các hằng, các biến, các hàm, các vị từ, người ta có thể đặt tùy ý các dấu cách (space hay blank)

Các bước xây dựng WFF

- Cho một phát biểu (sự kiện) trong ngôn ngữ tự nhiên :
  - (Chơ ai) đi mô cũng nhớ về Hà Tĩnh
- Xác định các miền đối tượng :
  - Người : cu Tý, cu Tèo, cái Nơ… $X \in D1$
  - Quê : Hà Tĩnh, Hà Tây, Hà Giang… $Y \in D2$
- Xác định các quan hệ để xây dựng các mệnh đề
  - Cu Tý quê Hà Tĩnh : QUÊ(cutý, hàtĩnh), QUÊ(X, Y)
  - Cu Tý xa quê Hà Tĩnh : XAQUÊ(cutý, hàtĩnh), XAQUÊ(X, Y)
  - Cu Tý nhớ quê Hà Tĩnh :
    NHỚQUÊ(cutý, hàtĩnh), NHỚQUÊ(X, Y)
- Xây dựng WFF

  → ∀X, ∃Y (QUÊ(X, Y) ∧ XAQUÊ(X, Y) → NHỚQUÊ(X, Y))

Một số nhận xét

- Quy ước trong một WFF :
  - Một biến được lượng tử hóa sẽ xuất hiện ngay sau ∃ hay ∀
  - Phạm vi lượng tử hóa của biến kể từ vị trí xuất hiện trở đi
  - Có thể có các biến tự do (free variable),
    là các biến không được lượng tử hóa
  - Ví dụ : P(X) và (∃Y) Q(X, Y) có chứa biến tự do X
- Logic vị từ được gọi là «bậc một» (first–order) :
  - Trong WFF không định nghĩa lượng tử cho vị từ hay cho hàm
  - Ví dụ : (∀P)P(a) và (∀f) (∀f) (∀X) P(f (X), b)
    không phải là những vị từ bậc một,
    mà có bậc cao hơn (higher-order)

Một số nhận xét

- Tri thức diễn tả theo ngôn ngữ tự nhiên hay toán học không phải luôn luôn dễ dàng chuyển đổi thành các WFF trong lôgic vị từ bậc một
- Chẳng hạn, để diễn tả rằng :
  «Nếu hai vật y chang nhau thì chúng có cùng tính chất», người ta có thể viết :

  $(\forall P)\ (\forall X)\ (\forall Y)\ (EQUAL(X, Y) \rightarrow (P(X) \leftrightarrow P(Y)))$
- Nhưng biểu thức trên không phải là logic vị từ bậc một vì có lượng tử $\forall$ áp dụng cho một ký tự vị từ là P
- Trong lôgic vị từ bậc một, sự kiện trên được viết :

  $(\forall X)\ (\forall Y)\ (SAME\_P(X, Y) \rightarrow (P(X) \leftrightarrow P(Y)))$, hoặc

  $(\forall X)\ (\forall Y)\ (SAME\_P(X, Y) \rightarrow (HAVE(X, p) \leftrightarrow HAVE\ (Y, p)))$

Equality

- Equality indicates that two terms refer to the same object =(A, B)
  - A and B are identical
  - Usually, written in infix form A = B
  - The equality symbol "=" is an (in-fix) shorthand
  - FATHER(jane) = jim, or =(FATHER(jane), jim)
  - E.g. Jim is Jane's and John's father
- Equality by reference and equality by value
  - Sometimes the distinction between referring to the same object and referring to two objects that are identical (indistinguishable) can be important
- $E \neq E$ E is not identical to iteself

Bài tập: Chuyển thành vị từ

- Ai đủ 18 tuổi mới được phép lái xe
- Gái đủ 18 tuổi, trai 20 tuổi mới được phép lập gia đình
- Kiểm tra hồ sơ
  - Nhập học tại các trường ĐH, CĐ
  - Sản phẩm
  - Quy trình công nghệ...

Sau đó thực hiện

1. **Xác định không gian các sự kiện, nhân vật thật liên quan**
2. **Tìm các hằng, biến, hàm và/hoặc vị từ tương ứng với các phát biểu**
3. **Gán nghĩa cho tứng thành phần để kiểm tra tính đúng đắn**
4. **Nhận kết quả**

Mệnh đề - Predicate Logic

- There is a predicate logic for each basis B=(F, P) of function and predicate symbols
- Terms formed on basis B are called B-terms: the set of all B-terms is denoted T_B
- Formulas formed on basis B are called B-formulas: the set of all B-formulas is denoted WFF_B
- Formulas with all variables bound to a quantifier are called closed formulas
- Formulas with no quantifier are called quantifier free formulas
- Formulas with no quantifier and no variable are called ground formulas

Example: Family Relationships

- Objects: people
- Properties: gender, …
  - Expressed as unary predicates:

    MALE(X), FEMALE(Y)   hoặc SEX(X, male), SEX(Y, female)
- Relations: parenthood, brotherhood, marriage
  - Expressed through binary predicates

    PARENT(X, Y), BROTHER(X, Y*), …*
- Functions: motherhood, fatherhood
  - Because every person has exactly one mother and one father:

    mother(X), father(Y)
  - There may also be a relation

    mother-of(X, Y), father-of(X, Y)

Atomic Sentences Translation

- Function translation:
  - Brando is Nancy's favorite actor:

    brando = favoriteactor(nancy)
  - Sean is his own favorite actor

    sean = favoriteactor(sean)
- Sentences translation:
  - Nancy's favorite actor is better than Max's favorite actor:

    BETTERACTOR(favoriteactor(nancy), favoriteactor(max))

Examples Atomic Sentences

FATHER(jack, john), MOTHER(jill, john), SISTER(jane, john)

PARENTS(jack, jill, john, jane)

MARRIED(jack, jill)

MARRIED(father-of(john), mother-of(john))

MARRIED(father-of(john), mother-of(jane))

FATHER(jack, john) $\wedge$ MOTHER(Jill, john) $\wedge$ SISTER(jane, john)

$\neg$ SISTER(john, jane)

PARENTS(jack, jill, john, jane) $\wedge$ MARRIED(jack, jill)

PARENTS(jack, jill, john, jane) $\rightarrow$ MARRIED(jack, jill)

OLDER-THAN(jane, john) $\vee$ OLDER-THAN(john, jane)

OLDER(father-of(john), 30) $\vee$ OLDER (mother-of(john), 20)

Avoid Ambiguity

- Both C and E are cubes
  $CUBE(c) \wedge CUBE(e)$
- Either C or E is a cube
  $CUBE(c) \vee CUBE(e)$
- Neither C nor E is a cube
  $\neg(CUBE(c) \vee CUBE(e))$
- Both A or B and C
  $A \vee B \wedge C$
- Either A or both B and C
  $A \vee (B \wedge C)$
- Either both A and B or C
  $A \wedge B \vee C$
- Both A and either B or C
  $A \wedge (B \vee C)$
- Either both Max is at home and Claire is tall or Carl is happy
  $(HOME(max) \wedge TALL(claire)) \vee HAPPY(carl)$

Lượng tử- Quantifiers

- Quantifiers can be used to express properties of collections of objects
- Quantifiers eliminates the need to explicitly enumerate all objects
- The universal quantifier, represented by the symbol ∀ means "for every" or "for all"
- The existential quantifier, represented by the symbol ∃ means "there exists"
- Limitations of predicate logic – *most* quantifier

Universal Quantifiers $\forall$ (for all)

- $\forall X\ P(x)$ states that
  - a predicate P is holds for all objects X in the universe (domain) under discourse
  - The sentence is true if and only if all the individual sentences where the variable X is replaced by the individual objects it can stand for are true
- Example
  - All dogs are happy

    $\forall\ X\ (DOG(X) \rightarrow HAPPY(X))$
  - No dog is happy

    All dogs are unhappy

    $\forall\ X\ (DOG(X) \rightarrow \neg HAPPY(X))$

Usage of Universal Qualification

- Universal quantification is frequently used to make statements like:
  - "all humans are mortal"
  - "all cats are mammals"
  - "all birds can fly" …
- This can be expressed through sentences like

$\forall X$ HUMAN(X) $\rightarrow$ MORTAL(X)

$\forall X$ CAT(X) $\rightarrow$ MAMMAL(X)

$\forall X$ BIRD(X) $\rightarrow$ CAN-FLY(X)

- These sentences are equivalent to the explicit sentence about individuals

HUMAN(john) $\rightarrow$ MORTAL(john) $\wedge$

HUMAN(jane) $\rightarrow$ MORTAL(jane) $\wedge$

HUMAN(jill) $\rightarrow$ MORTAL(jill) $\wedge$ …

Existential Quantifier ∃ (there exists)

- $\exists X\ P(x)$ states that
  - a predicate P holds for some objects in the universe
  - The sentence is **`true`** if and only if there is at least one **`true`** individual sentence where the variable X is replaced by the individual objects it can stand for
- Example:
  - Some dogs are happy

    $\exists X\ (DOG(X) \land HAPPY(X))$
  - Some dogs are not happy

    $\exists X\ (DOG(X) \land \neg HAPPY(X))$

Usage of Existential Quantification

- Existential quantification is used to make statements like:
  - "some humans are computer scientists"
  - "john has a sister who is a computer scientist"
  - "some birds can't fly" …
- This can be expressed through sentences like

  $\exists X$ HUMAN(X) $\wedge$ COMPUTER-SCIENTIST(X)

  $\exists X$ SISTER(X, john) $\wedge$ COMPUTER-SCIENTIST(X)

  $\exists X$ BIRD(X) $\wedge$ $\neg$CAN-FLY(X)
- These sentences are equivalent to the explicit sentence about individuals
  - HUMAN(john) $\wedge$ $\neg$ COMPUTER-SCIENTIST(john) $\vee$
  - HUMAN(jane) $\wedge$ COMPUTER-SCIENTIST(jane) $\vee$
  - HUMAN(jill) $\wedge$ $\neg$COMPUTER-SCIENTIST(jill) $\vee$ . . .

Multiple Quantifiers

- More complex sentences can be formulated by multiple variables and by nesting quantifiers
  - The order of quantification is important
  - Variables must be introduced by quantifiers, and belong to the innermost quantifier that mention them
- Examples

$\forall X, \forall Y \; PARENT(X, Y) \rightarrow CHILD(Y, X)$

$\forall X \; HUMAN(X) \wedge \exists Y \; MOTHER(Y, X) \quad \Rightarrow \forall X \; \exists Y \; HUMAN(X) \wedge MOTHER(Y, X)$

$\forall X \; HUMAN(X) \wedge \exists Y \; LOVES(X, Y) \quad \Rightarrow \forall X \; \exists Y \; HUMAN(X) \wedge LOVES(X, Y)$

$\exists X \; HUMAN(X) \wedge \forall Y \; LOVES(X, Y) \quad \Rightarrow \forall X \; \forall Y \; HUMAN(X) \wedge LOVES(X, Y)$

$\exists X \; HUMAN(X) \wedge \forall Y \; LOVES(Y, X) \quad \Rightarrow \exists X \; \forall Y \; HUMAN(X) \wedge LOVES(Y, X)$

- They can translate to the form:

Q<V> M<V>, with Q: quantifiers, M: Matrix, wffs including <V>
V: Variable

Connections between ∀ and ∃

- All statements made with one quantifier can be converted into equivalent statements with the other quantifier by using negation

  $\forall$ is a conjunction over all objects under discourse

  $\exists$ is a disjunction over all objects under discourse

- De Morgan's rules apply to quantified sentences

  $\forall X \neg P(X) \equiv \neg \exists X\ P(X)$　　$\neg \forall X\ P(X) \equiv \exists X \neg P(X)$

  $\forall X\ P(X) \equiv \neg \exists X \neg P(X)$　　$\neg \forall X \neg P(X) \equiv \exists X\ P(X)$

- Strictly speaking, only one quantifier is necessary
  - Using both is more convenient

Family Relationships

- One's mother is one's sibling's mother

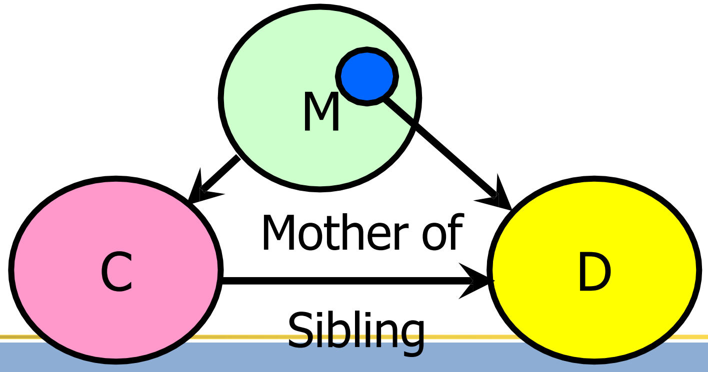
  $\forall M, C, D\ MOTHER(M, C) \wedge SIBLING(C, D) \rightarrow MOTHER(M, D)$
- Brothers are siblings

  $\forall X, Y\ BROTHER(X, Y) \rightarrow SIBLING(X, Y)$
- Sibling is transitive

  $\forall X, Y, Z\ SIBLING(X, Y) \wedge SIBLING(Y, Z) \rightarrow SIBLING(X, Z)$
- One's mother is one's sibling's mother

  $\forall M, C\ MOTHER(M, C) \wedge SIBLING(C, D) \rightarrow MOTHER(M, D)$
- A first cousin is a child of a parent's sibling

  $\forall C, D\ FIRSTCOUSIN(C, D) \leftrightarrow$
  $\exists P, PS\ PARENT(P, D) \wedge SIBLING(P, PS) \wedge PARENT(PS, C)$



Bài tập

- Tìm mệnh đề tương ứng đặt vào dấu ?

Người này là mẹ của người kia khĩ một người là phụ nữ và là thân sinh của người kia

$\forall M, \forall C\ MOTHER(C) = M \leftrightarrow FEMALE(M) \wedge PARENT(M, C)$

Người này là chồng của người kia khĩ người này là đàn ông và người kia là vợ của người này

$\forall W, \forall H\ HUSBAND(H, W) \leftrightarrow MALE(H) \wedge SPOUSE(W, H)$

1) ?

$\forall X\ MALE(X) \leftrightarrow \neg FEMALE(X)$

2) ?

$\forall G, \forall C\ GRANDPARENT(G, C) \leftrightarrow \exists P\ PARENT(G, P) \wedge PARENT(P, C)$

3) ?

$\forall X, \forall Y\ SIBLING(X, Y) \leftrightarrow$

$\neg(X=Y) \wedge \exists P\ PARENT(P, X) \wedge PARENT(P, Y)$

Translating English

- Every gardener likes the sun

  $\forall X\ \mathrm{GARDENER}(X) \rightarrow \mathrm{LIKES}(X, \mathrm{sun})$
- You can fool some of the people all of the time

  $\exists X\ \forall T\ (\mathrm{PERSON}(X) \wedge \mathrm{TIME}(T)) \rightarrow \mathrm{CAN\text{-}FOOL}(X, T)$
- You can fool all of the people some of the time

  $\forall X\ \exists T\ (\mathrm{PERSON}(X) \wedge \mathrm{TIME}(T) \rightarrow \mathrm{CAN\text{-}FOOL}(X, T)$
- All purple mushrooms are poisonous

  $\forall X\ (\mathrm{MUSHROOM}(X) \wedge \mathrm{PURPLE}(X)) \rightarrow \mathrm{POISONOUS}(X)$
- No purple mushroom is poisonous

  $\neg(\exists X)\ \mathrm{PURPLE}(X) \wedge \mathrm{MUSHROOM}(X) \wedge \mathrm{POISONOUS}(X)$

  or, equivalently,

  $(\forall X)\ (\mathrm{MUSHROOM}(X) \wedge \mathrm{PURPLE}(X)) \rightarrow \neg\mathrm{POISONOUS}(X)$

Translating English

- There are exactly two purple mushrooms

  $(\exists X)(\exists Y)\ MUSHROOM(X) \wedge PURPLE(X) \wedge MUSHROOM(Y) \wedge PURPLE(Y)$
  $\wedge \neg(X=Y) \wedge (\forall Z)\ (MUSHROOM(Z) \wedge PURPLE(Z))$
  $\rightarrow$
  $((X=Z) \vee (Y=Z))$

- Bob is not tall

  $\neg TALL(bob)$

- X is above Y if X is on directly on top of Y or else there is a pile of one or more other objects directly on top of one another starting with X and ending with Y

  $(\forall X)(\forall Y)\ ABOVE(X, Y) \leftrightarrow (ON(X, Y) \vee (\exists Z)\ (ON(X, Z) \wedge ABOVE(Z, Y)))$

Colonel West Example

- Assume:
  - It is a crime for an American to sell weapons to a hostile nation. The country Nono, an enemy of America, has some missiles, and all its missiles were sold to it by Colonel West, who is an American
  - Constants: nono, america, west
  - Predicates: CRIMINAL, AMERICAN, WEAPON, HOSTILE, NATION, ENEMY, MISSILE, OWNS, SELLS
- Then
  - It is a crime for an American to sell weapons to a hostile nation

$\Rightarrow$ If any american X sells any weapon Y to any hostile nation Z, then that american X is a criminal

$$\forall X,Y,Z \quad (\mathrm{AMERICAN}(X) \wedge \mathrm{WEAPON}(Y) \wedge \mathrm{NATION}(Z) \wedge \mathrm{HOSTILE}(Z) \wedge \mathrm{SELLS}(X, Z, Y) \rightarrow \mathrm{CRIMINAL}(X))$$

Other Translating

- Nono's missiles
  - Nono has some missiles

    $\exists X\ (MISSILE(X) \wedge OWNS(nono,\ X))$
  - All of Nono's missiles were sold to it by West

  $\Rightarrow$ If X is a missile owned by Nono then West sold X to Nono

    $\forall X\ ((MISSILE(X) \wedge OWNS(nono, X)) \rightarrow SELLS(west, nono,\ X))$
- The other facts
  - An enemy of America is hostile

    $\forall X\ (ENEMY(X, america) \rightarrow HOSTILE(X))$
  - West is an American

    $AMERICAN(west)$
  - Nono is an enemy of America

    $NATION(nono) \wedge NATION(america) \wedge ENEMY(nono,\ america)$

Other Examples

- Love
  - There is a girl who is loved by every boy

  $\Rightarrow$ There is a girl X and if Y is a boy then Y loves her

  $\exists X\,(GIRL(X) \wedge \forall Y(BOY(Y) \rightarrow LOVES(Y,X)))$

  - Every boy loves some girl
  - For every boy Y there exists a girl X that he loves

  $\forall Y(BOY(Y) \rightarrow \exists X\,(GIRL(X) \wedge LOVES(Y,X)))$

- Socrates Example
  - All men are mortal

  $\forall X\,(MAN(X) \rightarrow MORTAL(X))$

  - Socrates is a man

  MAN(socrates)

  - Socrates is mortal

  MORTAL(socrates)

Curiosity Example

- Assume:

  Jack owns a dog

  Every dog owner is an animal lover

  No animal lover kills an animal

  Either Jack or Curiosity killed the cat

  The cat's name is Tuna

  Then

  - Constants: jack, curiosity, tuna.
  - Predicates: OWNS, DOG, ANIMALLOVER, KILLS, ANIMAL

Curiosity Example

- Dog owners
  - Jack owns a dog

  $\Rightarrow$ Some dog is owed by Jack

  $\exists X\ (DOG(X) \wedge OWNS(jack,\ X))$

  - Every dog owner is an animal lover

  $\forall X\ ((\exists Y\ (DOG(Y) \wedge OWNS(X,Y))) \rightarrow ANIMALLOVER(X))$

- Animal Lovers
  - No animal lover kills an animal

  $\Rightarrow$ An animal lover does not kill an animal

  $\Rightarrow$ For any animal lover X and any animal Y, then Y is not killed by X

  $\forall X, Y\ ((ANIMALLOVER(X) \wedge ANIMAL(Y)) \rightarrow \neg KILLS(X,Y))$

Curiosity Example

- Cat Knowledge
  - The cat's name is Tuna

    CAT(tuna)
  - Either Jack or Curiosity killed the cat

    $KILLS(jack, tuna) \vee KILLS(curiosity, tuna)$
  - All cats are animals

    $\forall X\ CAT(X) \rightarrow ANIMAL(X)$

Bài tập

- Cho các câu sau :
  1. Marcus là một người
  2. Marcus là người xứ Pompeii
  3. Mọi người Pompeii đều là người La Mã
  4. Ceasar là một kẻ cầm quyền
  5. Người La Mã hoặc là trung thành với Ceasar, hoặc là thù ghét Ceasar
  6. Mỗi người đều trung thành với một người nào đó
  7. Nhân dân chỉ muốn giết những kẻ cầm quyền mà họ không trung thành
  8. Marcus muốn giết Ceasar
- Biểu diễn các câu trên thành các wff's

Biểu diễn wff's trong lôgic vị từ

- Marcus là một người

  MAN(marcus)
- Marcus là một người Pompeii

  POMPEIAN(marcus)
- Mọi người Pompeian đều là người La Mã

  $\forall x : POMPEIAN(X) \rightarrow ROMAN(X)$
- Caesar là một kẻ cầm quyền
  (giả sử không có sự trùng tên)

  RULER(caesar)
- Người La Mã hoặc là trung thành với Caesar,
  hoặc là thù ghét Caesar

  $\forall x : ROMAN(X) \rightarrow LOYALTO(X, caesar) \vee HATE(X, caesar)$
- Tuy nhiên khi sử dụng với nghĩa hoặc có loại trừ, có thể viết lại :

  $\forall x : ROMAN(X) \rightarrow ((LOYALTO(X, caesar) \wedge HATE(X, caesar)) \wedge (\neg LOYALTO(X, caesar) \wedge HATE(X, caesar)))$

Biểu diễn wff's trong lôgic vị từ

- Mọi người đều trung thành với một người nào đó

  $\forall$X, $\exists$Y LOYALTO(X, Y) hoặc $\exists$Y, $\forall$X LOYALTO(X, Y)

- Nhân dân chỉ muốn giết những kẻ cầm quyền mà họ không trung thành

  $\forall$X, $\forall$Y PERSON(X) $\wedge$ RULER(Y) $\wedge$ TRYASSASSINATE(X, Y) $\rightarrow$ $\neg$LOYALTO(X, Y)

  Mệnh đề này tỏ ra mập mờ (ambiguous). Có phải chỉ những kẻ cầm quyền mà nhân dân muốn giết là những kẻ họ không trung thành (theo nghĩa đã biểu trưng) hay chỉ những điều mà nhân dân có ý định là giết những kẻ cầm quyền mà họ không trung thành ?

- Marcus muốn giết Ceresar
  - TRYASSASSINATE(marcus, caesar)
- Câu hỏi : Marcus có trung thành với Caesar không ?

Bài tập

- Cho các câu sau :
  - Marcus là một người
  - Marcus là người Pompeii
  - Marcus sinh năm 40 trong công nguyên (A.D.: Anno Domini)
  - Mọi người đều (ai cũng phải) chết
  - Tất cả mọi người dân Pompeii đều bị chết
    vì núi lửa phun vào năm 79 A.D.
  - Không có người nào (không ai) sống nhiều hơn 150 tuổi
  - Bây giờ là năm 2007
  - Còn sống có nghĩa là không chết
  - Nếu ai đó chết, thì người ấy có thể chết ở mọi thời điểm sau đó
  - Marcus còn sống không ?
- Biểu diễn các câu trên thành các wff's

Ví dụ

Marcus là người

MAN(marcus)

- Marcus là người Pompeii

POMPEIAN (marcus)

- Marcus sinh năm 40 trong công nguyên (A.D.: Anno Domini)

BORN(marcus, 40)

- Mọi người đều (ai cũng phải) chết

$\forall$X MAN(X) $\varnothing$ MORTAL(X)

- Tất cả mọi người dân Pompeii đều bị chết vì núi lửa phun vào năm 79 A.D.

$\forall$X ERUPTED(vocalno, 79) $\wedge$ POMPEIAN(X) $\varnothing$ DIED(X, 79)

Ví dụ

- Không có người nào sống nhiều hơn 150 tuổi

  ∀X ∀T1 ∀T2
  MAN(X) ∧ BORN(X, T1) ∧ GT(T2 - T1, 150) ∅ DIED(X, T2)

  - Bây giờ là năm 2007

  now = 2007

  - Còn sống có nghĩa là không chết

  ∀X ∀T (ALIVE(X, T) ∅ ¬DEAD(X, T))
  ∧ ((DIED(X, T) ∅ ¬ALIVE(X, T))

  - Nếu ai đó chết,
    thì người ấy có thể chết ở mọi thời điểm sau đó

  ∀X ∀T1 ∀T2 DIED(X, T1) ∧ GT(T2, T1) ∅ DEAD(X, T2)

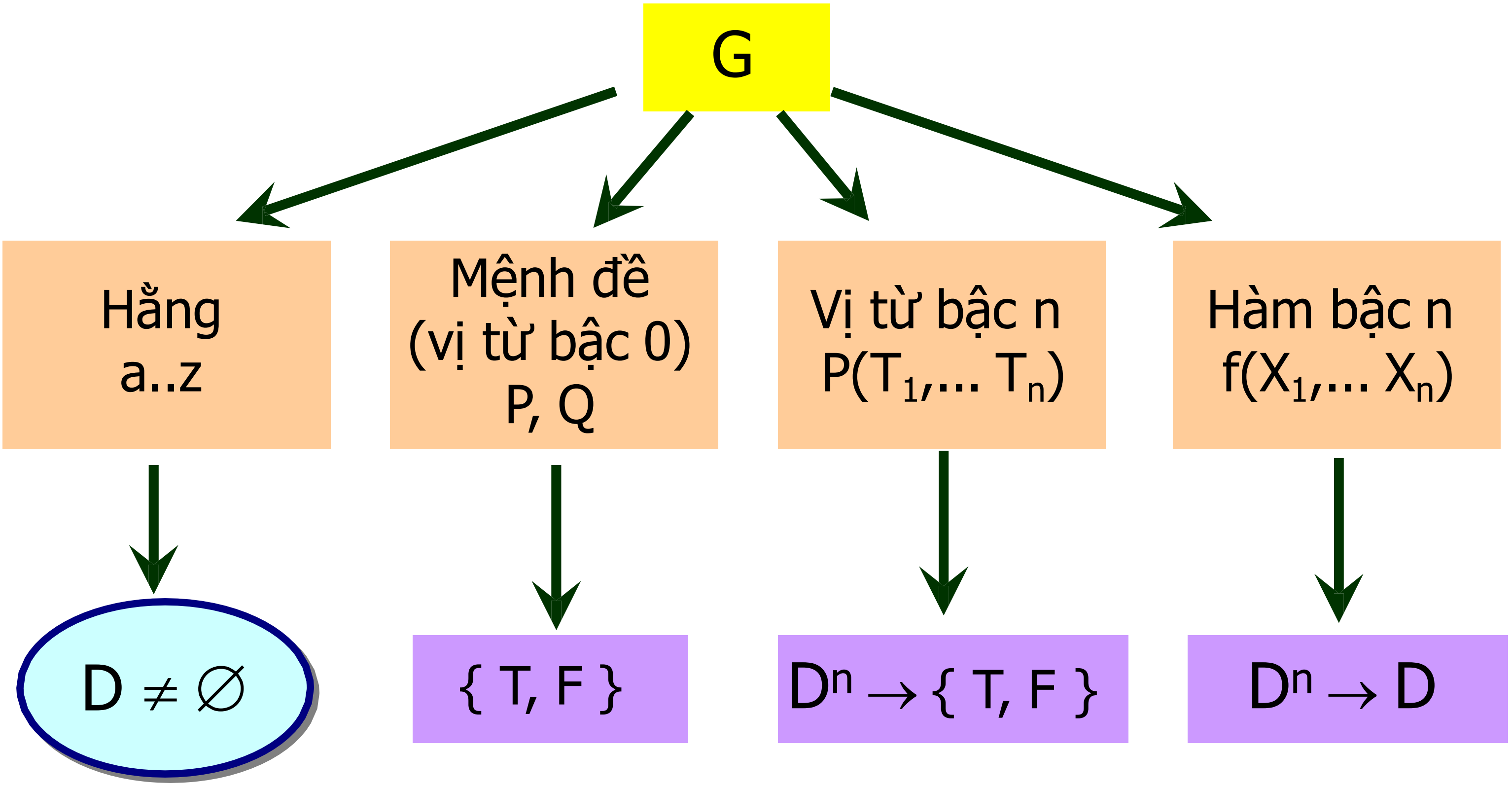
Diễn giải (Interpretation)

- Cho G là một WFF, một diễn giải của G, ký hiệu I, được xác định từ năm bước sau đây :
  - Chọn miền diễn giải (Interpretation Domain) là các tập hợp khác rỗng, ký hiệu $D \neq \varnothing$
  - Gán (Assignation) cho mỗi hằng của G một phần tử của D_i
  - Gán cho mỗi mệnh đề (hay vị từ bậc 0) một giá trị true (T) hoặc false (F)
  - Gán cho mỗi vị từ bậc n ($n \geq 1$) ánh xạ từ D^n lên { T, F }:

    $P(T_1, \ldots T_n) : D^n \rightarrow \{ T, F \}$
  - Gán cho mỗi hàm bậc n ($n \geq 1$) ánh xạ từ D^n lên D:

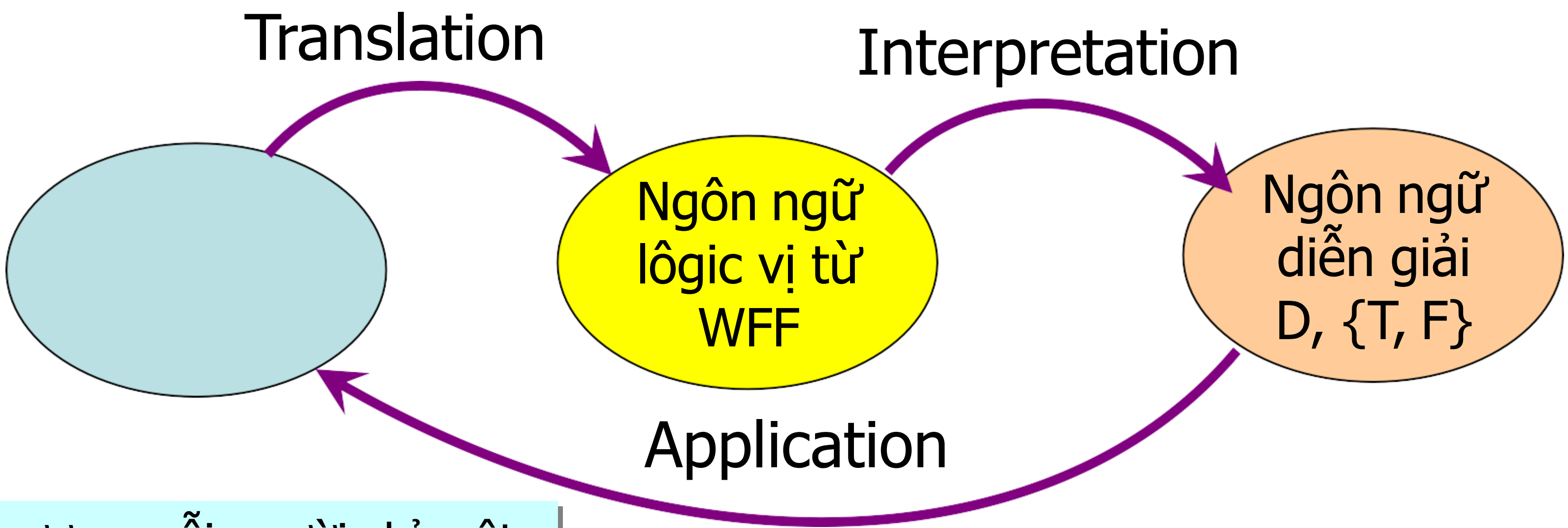
    $f(X_1, \ldots X_n) : D^n \rightarrow D$

Mô hình diễn giải I từ G lên D



- Khi một WFF G có giá trị là T theo một diễn giải I, người ta nói rằng diễn giải I là một mô hình của G

Quan hệ Translation-Interpretation



Quê hương mỗi người chỉ một,
như là chỉ một mẹ thôi

T

$$\forall X \exists Y \exists Z\ (QUE(X,Y) \wedge (QUE(X,Z) \rightarrow EQ(Y,Z))$$
$$\leftrightarrow$$
$$\forall T \exists U \exists V\ (FEMALE(U) \wedge (MOTHER(U,T))$$
$$\wedge\ (FEMALE(V) \wedge (MOTHER(V,T)) \rightarrow EQ(U,V))$$

Tính giá trị của wff theo diễn giải

- Cho G là một WFF và một diễn giải I trên một miền D
- Khi đó :
  - Nếu G là một mệnh đề, giá trị gán cho G qua I là I(G)
  - Nếu G là một trực kiện, G nhận một giá trị T hay F
  - Nếu G có dạng (∀X)G' :
    - I(G) = T nếu I(G') = T cho mọi giá trị của biến X trong D
    - I(G) = F nếu không phải
  - Nếu G có dạng (∃X)G' :
    - I(G) = T nếu I(G') = T với ít nhất một giá trị của X trong D
    - I(G) = F nếu không phải

Tính giá trị của wff theo diễn giải

- Nếu G có dạng ($\neg$G′) :
  - I(G) = T nếu I(G′) = F trong D
  - I(G) = F nếu I(G′) = T trong D
- Nếu G có dạng (G′ $\wedge$ G″), hoặc (G′ $\vee$ G″), hoặc (G′ $\rightarrow$ G″), hoặc (G′ $\leftrightarrow$ G″), khi đó :

G′	G″	(G′ $\wedge$ G″)	(G′ $\vee$ G″)	G′ $\rightarrow$ G″	G′ $\leftrightarrow$ G″
F	F	F	F	T	T
F	T	F	T	T	F
T	F	F	T	F	F
T	T	T	T	T	T

Tính hợp thức và tính nhất quán

- Một WFF được gọi là :
  - *Hằng đúng* (tautology), hay *hợp thức* (valid) nếu và chỉ nếu mọi diễn giải đều cho giá trị T
  - Nếu không, được gọi là *không hợp thức* (non–valid)
- Một WFF được gọi là :
  - *Mâu thuẫn* (contradiction), hay *không nhất quán* (inconsistent) nếu và chỉ nếu với mọi diễn giải đều cho giá trị F
  - Nếu không, được gọi là *nhất quán* (consistent)
- Quy ước :

  →⅃ biểu diễn một WFF hợp thức-hằng đúng

  →∇ biểu diễn một WFF mâu thuẫn-không nhất quán

Công thức tương đương

- Cho G và H là hai WFF
- G và H được gọi là tương đương, $G \equiv H$:
  - Nếu và chỉ nếu G và H có cùng giá trị (T hoặc F) cho mọi diễn giải I, I(G) = I(H)
- Ví dụ :
  - $(P(a) \rightarrow Q(b)) \equiv ((\neg P(a) \vee Q(b))$
- Có thể dùng bảng chân lý để kiểm tra tính tương đương của các WFF

G, H, K là các *WFF* bất kỳ,
– G(X), H(X) là các *WFF* với X là biến tự do,
– □ biểu diễn một *WFF* hợp thức,
∇ biểu diễn một *WFF* không nhất quán.

Bảng các công thức tương đương

Công thức tương đương		*Tên công thức*
$(G \to H)$	$((\neg G) \vee H)$	
$(G \leftrightarrow H)$	$((G \to H) \wedge (H \to G))$	
$(\neg(\neg G))$	G	
$(\neg(G \wedge H))$	$((\neg G) \vee (\neg H))$	Luật De Morgan
$(\neg(G \vee H))$	$((\neg G) \wedge (\neg H))$	
$((G \wedge (H \vee K))$	$((G \wedge H) \vee (G \wedge K))$	Luật phân phối
$((G \vee (H \wedge K))$	$((G \vee H) \wedge (G \vee K))$	
$(G \wedge H)$	$(H \wedge G)$	Luật giao hoán
$(G \vee H)$	$(H \vee G)$	
$((G \vee H) \vee K)$	$(G \vee (H \vee K))$	Luật kết hợp cho phép loại bỏ dấu ngoặc
$((G \wedge H) \wedge K))$	$(G \wedge (H \wedge K))$	
$(G \to H)$	$((\neg H) \to (\neg G))$	Luật đối vị

Bảng các công thức tương đương

Công thức tương đương		*Tên công thức*
$(G \wedge \square)$	$\mathbf{G}$	
$(G \wedge \nabla)$	∇	
$(G \vee \square)$	$\square$	
$(G \vee \nabla)$	G	
$(G \vee (\neg G))$	$\square$	
$(G \wedge (\neg G))$	∇	
$(\forall X)(G(X))$	$(\forall Y)(G(Y))$	*Luật dùng chung các biến*
$(\exists X)(G(X))$	$(\exists Y)(G(Y))$	
$\neg((\forall X)G(X))$	$(\exists Y)(\neg G(Y))$	
$\neg((\exists X)G(X))$	$(\forall Y)(\neg G(Y))$	
$(\forall X)(G(X) \wedge H(X))$	$((\forall X)G(X) \wedge (\forall Y)H(Y))$	
$(\exists X)(G(X)) \vee H(X))$	$((\exists X)G(X) \vee (\exists Y)H(Y))$	

Biến đổi các WFF loại bỏ lượng tử $\forall$ và $\exists$

- Standardize Variables
  - Make sure that you are not using the same variable name twice in a single sentence (unless you really meant to)

    Eg.

    $(\forall X\ P(X)) \vee (\exists X\ Q(X))$ becomes $(\forall X\ P(X)) \vee (\exists Z\ Q(Z))$
- Move all quantifiers left, but keep them in order!
  - Eg.$(\forall X\ P(X) \vee \exists Y\ Q(Y))$ becomes $(\forall X\ \exists Y\ P(X) \vee Q(Y))$
- Translation into the form:

Q<V> M<V>

with Q: quantifiers, in order $\forall$s and then $\exists$s
M: Matrix: wffs including V, <V>: Variable

Biến đổi các WFF

- Skolemize: Eliminate Existential Quantifiers
  - Existential quantifiers can be eliminated by the introduction of a new constant that does not appear elsewhere
- in the database
  - SUBST{ X|a, a$\in$D } /* *Substitution*
  - Eg.

$\exists$X P(X) becomes P(a)

"*Có người vào lớp muộn*"

becomes "*Cu Tý vào lớp muộn*"

$\exists$X P(X) $\vee$ Q(X) becomes P(a) $\vee$ Q(a)

"*Chàng tìm đồng kia bãi nọ*"

becomes

"*Chàng tìm đồng dưới đồng trên*"

Biến đổi các WFF

- Skolemize: Drop Existential Quantifiers
  - One possible complication occurs if we also have Universal quantifiers… Consider

    $\forall X\ PERSON(X) \rightarrow \exists Y\ (HEART(Y) \wedge HAS(X,\ Y))$

    becomes by SUBST{Y|h}:

    $\forall X\ PERSON(X) \rightarrow HEART(h) \wedge HAS(X,\ h)$
  - Instead we have to create a new (Skolem) function to map from a person to their HEART(f(x))

    $\forall X\ \exists Y\ PERSON(X) \rightarrow (HEART(Y) \wedge HAS(X,\ Y))$ becomes:

    $\forall X\ PERSON(X) \rightarrow HEART(f(X)) \wedge HAS(X,\ f(X))$

    Eg., in general:

    $\forall X\ \forall Y\ \forall Z\ \ldots\ \exists T\ \ldots\ P(X,\ Y,\ Z, \ldots T, \ldots)$ becomes:

    $\forall X\ \forall Y\ \forall Z\ \ldots\ P(X,\ Y,\ Z, \ldots f(X,\ Y,\ Z, \ldots), \ldots)$
- Drop all Universal Quantifiers in the end:
  - $\forall X\ \forall Y\ \forall Z \ldots\ P(X,\ Y,\ Z, \ldots)$ becomes: $P(X,\ Y,\ Z, \ldots)$

Biến đổi các WFF

- Move the $\leftrightarrow$:
  - $P \leftrightarrow Q$ becomes $(P \rightarrow Q) \wedge (Q \rightarrow P)$
- Distribute $\wedge$ over $\vee$
  - $(A \wedge B) \vee C$ becomes $(A \vee C) \wedge (A \vee B)$
  - Just like distribution in arithmetic
  - $(5 + 4) * 6$ becomes $(5 * 6) + (4 * 6)$
- Flatten nested conjunctions and disjunctions
  - $(A \wedge B) \wedge C$ becomes $(A \wedge B \wedge C)$
  - $(A \vee B) \vee C$ becomes $(A \vee B \vee C)$

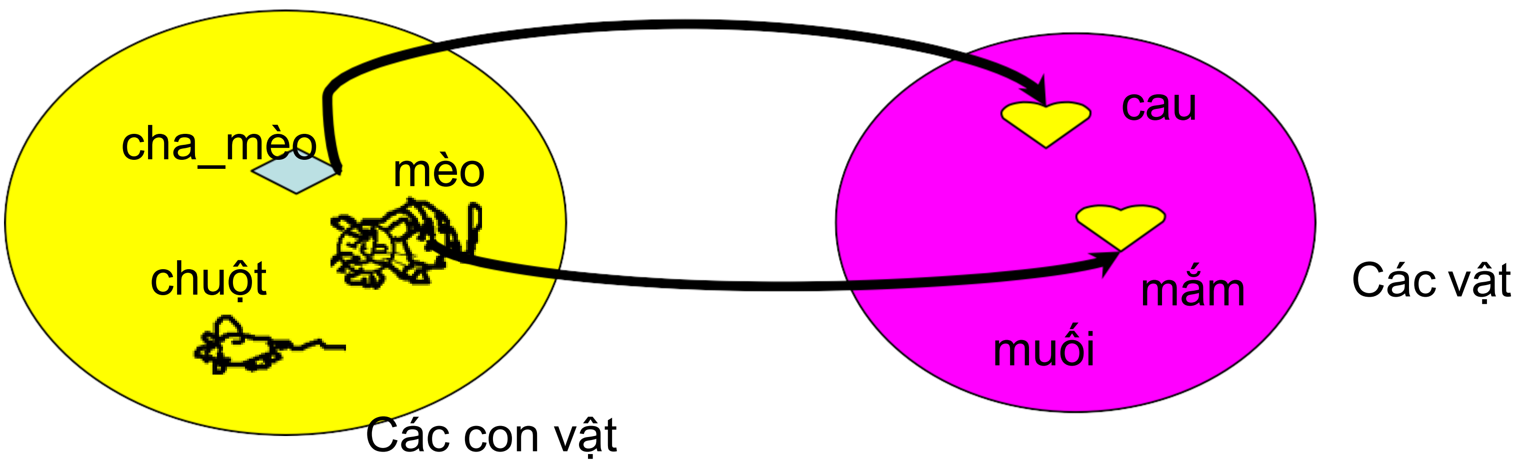
- Biến đổi các WFF :
  - Standardize Variables
  - Move the $\leftrightarrow$
  - Move Quantifiers left
  - Skolemize:
    - Eliminate Existential Quantifiers
    - Drop Universal Quantifiers
  - Distribute $\wedge$ over $\vee$
  - Flatten nested conjunctions and disjunctions
- Attention:
  - In Proof Theory by Robinson Resolution, they need to before:
    - Eliminate Implications
    - Move $\neg$ inwards

Xây dựng cơ sở luật cho XPS

- Sự kiện 1 :
  - Con mèo mà trèo cây cau
    Hỏi thăm chú chuột đi đâu vắng nhà
    Chú chuột đi chợ đường xa
    Mua mắm, mua muối về giỗ cha con mèo
- Hỏi :
  - Mèo có ăn được (gặp) chuột không rứa ?
- Sự kiện 2 :
  - Ông Trăng mà lấy bà Trời
    Tháng Năm đi cưới, tháng Mười nộp cheo
    Làng xã làm thịt một con mèo
    Làng ăn không hết làng treo cột đình
    Ông Xã đánh trống thình thình
    Bao nhiêu con nít ra đình gặm xương
- Hỏi :
  - Cu Tý có gặm được xương không vậy ?

Tạo không gian : các miền xác định

- Sự kiện 1 :
  - Con mèo mà trèo cây cau : TREO(X, Y)
  - Hỏi thăm chú chuột đi đâu vắng nhà :THAM(X, Y), VANGNHA(X)
  - Chú chuột đi chợ đường xa
  - Mua mắm, mua muối về giỗ cha con mèo



Gợi ý dùng luật :

$$MEO(X) \wedge CHUOT(Y) \wedge THAM(X, Y) \wedge COONHA(Y) \rightarrow ANTHIT(X,Y)$$

Bài tập mệnh đề logic

- Xác định mệnh đề logic vị từ tương ứng với các phát biểu sau:
- a) Người này là mẹ của người khi người này là phụ nữ và là thân sinh của người kia.
- b) Người này là chồng của người kia khi và chỉ khi người này là đàn ông và người kia là vợ của người này
- c) Người G là ông bà của người C khi và chỉ khi người G là thân sinh của người P và người P là thân sinh của người C
- d) Người X và người Y là anh chị em ruột khi và chỉ khi người P đồng thời là thân sinh của người X và người Y.

Bài tập mệnh đề logic

- Xác định mệnh đề logic vị từ tương ứng với các phát biểu sau:
- a) Người này là mẹ của người khi người này là phụ nữ và là thân sinh của người kia.
- $\forall M, \forall C \; MOTHER(C) = M \leftrightarrow FEMALE(M) \wedge PARENT(M, C)$
- b) Người này là chồng của người kia khi và chỉ khi người này là đàn ông và người kia là vợ của người này
- $\forall H, \forall W \; HUSBAND(H, W) \leftrightarrow MALE(H) \wedge SPOUSE(W, H)$

Bài tập mệnh đề logic

- Xác định mệnh đề logic vị từ tương ứng với các phát biểu sau:
- c) Người G là ông bà của người C khi và chỉ khi người G là thân sinh của người P và người P là thân sinh của người C
- $\forall G, \forall C$ GRANDPARENT(G, C) $\leftrightarrow \exists P$ PARENT(G, P) $\wedge$ PARENT(P, C)
- d) Người X và người Y là anh chị em ruột khi và chỉ khi người P đồng thời là thân sinh của người X và người Y.
- $\forall X, \forall Y$ SIBLING(X, Y) $\leftrightarrow \neg(X=Y) \wedge \exists P$ PARENT(P, X) $\wedge$ PARENT(P, Y)

Q&A