Procedure Production

1. DATA <- Initial database
2. until DATA satisfies the termination condition do:
3. Begin
4. Select some rule, R, from the set of Rules that can be applied to DATA
5. DATA <- Result of applying R to DATA
6. end

FOPL

A formal language in which a wide variety of statements can be expressed

- Define the language
- How it is used to represent statements
- How inferences are drawn
- How statements are deduced

Introduction to Predicate Calculus

Syntax: alphabets of symbols and how they are put together to form legitimate expression (*wff*)

Elementary components:
predicate symbol, variable symbol, function symbol, constant symbol

Alongwith parenthesis, brackets, commas
connectives
Quantifiers

WRITE(Nilsson, Principles……)

For all/ There exists WRITE(x,y)

Jonh's brother and Jonhn's sister are sibling to each other

SIBLING (brother(JOHN), sister(JOHN))

The House is Yellow.

John lives in a yellow house.

Connectives: AND, OR, Not

LIVES(JOHN, H1) ^ COLOR(H1, YELLOW)

==========================

And: ^ Not: ~/¬ Or: v

(∀x) ELEPHANT(x) => COLOR (x, GREY)

(∃y) PERSON(y) ^ **WROTE(y, GITA)**

Unification: substitution instance, composition

Substitution instance: of an expression is obtained by substituting terms for variables in that expression. E = P[x,f(y),B]

E = P[x,f(y),B]

E1 = P[z,f(w),B] s1 = {z/x, w/y} alphabetic variant (E1 = Es1)
E2 = P[x,f(A),B] s2 = {A/y}
E3 = P[g(z),f(A),B] s3 = {g(z)/x, A/y}
E4 = P[C,f(A),B] s4= {C/x, A/y} ground instance (E4= Es4)

s = {t1/v1, t2/v2,…, ti/vi,…, tn/vn}

To denote a substitution: Es
P[z,f(w),B] = P[x,f(y),B] s1
Composition of two substitutions : Es1s2
s1 s2 ??
{g(x,y)/z} {A/x, B/y, C/w, D/z}
E = (x, y, z) ------ s = {g(A,B)/z, A/x, B/y, C/w}
Substitution obtained by applying s2 to the terms of s1 and then adding any pairs of s2 having variables not occurring among variables of s1.

Unification: substitution instance, composition

Substitution instance: of an expression is obtained by substituting terms for variables in that expression. E = P[x,f(y),B]

E = P[x,f(y),B]

E1 = P[z,f(w),B] s1 = {z/x, w/y} alphabetic variant (E1 = Es1)
E2 = P[x,f(A),B] s2 = {A/y}
E3 = P[g(z),f(A),B] s3 = {g(z)/x, A/y}
E4 = P[C,f(A),B] s4= {C/x, A/y} ground instance (E4=Es4)

s = {t1/v1, t2/v2,…, ti/vi,…, tn/vn}

To denote a substitution: Es
P[z,f(w),B] = P[x,f(y),B] s1
Composition of two substitutions : Es1s2
s1 s2 ??
{g(x,y)/z} {A/x, B/y, C/w, D/z}
E = (x, y, z) ------ s = {g(A,B)/z, A/x, B/y, C/w}
Substitution obtained by applying s2 to the terms of s1 and then adding any pairs of s2 having variables not occurring among variables of s1.

Unification

Unification is done to match certain sub expressions in a given set of expressions.

Let our database contains:
(1)(∀x) [W1(x) => W2(x)]
(2) W1(A)
From 1 and 2 we can derive W2(A). *But how??*
To do that, we substitute A for x which makes W1(A) and W1(x) identical

Unification: Finding substitution of terms for variables to make expressions identical
(e.g., E1 = W1(A), E2 = W1(x)
To make, E1 = E2, I need some substitution, here it is {A/x})

I can substitute a **variable** by a **term** , where,
terms: var/ constant/ function

Constraints: x by A --- throughout ~~Replace x by f(x)~~
{term / var} = {Mother(x)/ x} ??

Rules of Inference, Theorems, Proofs

Rules of Inference can be applied to certain wffs and sets of wffs to produce new wffs.

1.Modus ponens

Produce W2 from (i) W1 and (ii) W1=>W2

2.Universal specialization

Produce W(A) from (i) (∀x) W(x)

3. Using both,

Produces W2(A) from

(i) (∀x) [W1(x) => W2(x)] and (ii) W1(A)

Our Aim

Given the **Database**

1. Wff1
2. Wff2
3. Wff3
4. Wff4

5. Prove wff5

a)Wff1, wff2 ---- wffx (which two can be combined, is supported by **Operators/ Rules/— here , Rules of inference**)
b)Wff3, wff4 --- wffy

……………..

m)Wff7, wffx ---- wff5 /
n) Wffx, wffy --- wff5 say, *wff5 is my derived wff*
(whether I will first apply a) then b) or the opposite or some other… will be governed by **Control Strategy**)

~ (~X)
~(X1 ^ X2)   ~X1 U ~X2
X1 ^ (X2 U X3)
X1 ^ X2

Contrapositive: X1 => X2   ~X2 => ~X1

~(∃y) P(y)

(∀x) [P(x) and Q(x)]
(∀x) [P(x)] and (∀y) [Q(y)]
For every set x, there is a set y, such that cardinality of y is greater than that of x.
(∀x) SET(x) =>(∃y) SET(Y) ^
GREATER(cardinality(SET(X),cardinality(SET(Y))
CARD(x,u)   CARD(y,v)

Predicate Calculus

John Lives in a Yellow House

COLOR(HOUSE ???
LIVE(JOHN, HOUSE1) ^ COLOR (HOUSE1, YELLOW)
HOUSE(JOHN,YELLOW)

John Plays Chess or Badminton
PLAYS(JOHN,CHESS) U PLAYS(JOHN,BADMINTON)

If the car belongs to John, then it is yellow
BELONGS (JOHN, CAR1) => COLOR (CAR1, YELLOW)

A => B is equivalent to ~A U B

Proposition Calculus: as *terms* we never used variables

Sets of Literals	Most Common Substitution instances
i. {P(x), P(A)}	P(A) {A/x} ~~{x/A}~~ ti/vi
ii. {P[f(x),y,g(y)], P[f(x),z,g(x)]}	P[f(x), x, g(x)] {x/y, x/z}
iii. {P[f(x,g(A,y)), g(A,y)], P[f(x,z),z]}	P[f(x,g(A,y)), g(A,y)]

The disagreement set of E is the set D obtained by comparing each symbol of all expressions in E from left to right and extracting from E the subexpression whose first symbols do not agree.

Unification Algorithm

1. Set $k = 0$, $mgu_k = \{\}$
2. If the set $Emgu_k$ is a singleton then stop; mgu_k is an mgu of E. Otherwise, find the disagreement set D_k of $Emgu_k$
3. If there is a var v and term t in D_k such that v does not occur in t, put $mgu_{k+1} = mgu_k\{t/v\}$, set $k = k+1$, and return to step 2. Otherwise, stop, E is not unifiable.

E = {P(x,z,y), P(w,u,w), P(A,u,u)}

mgu ??

Unification: substitution instance, composition

Es' = Es1s2 = Es2s1 ??? Is it Correct ?
s1s2 = s2s1 wrong (commutative property will not hold here)
(s1,s2)s3 = s1(s2s3) correct (Associative)

Unifiable:
{E1, E2, E3….}
A set {Ei} s
E1s = E2s = E3s …… Ens = E

{Ei} is unifiable, s is unifier

s = {A/x, B/y} unifies
{Ei} = {P[x,f(y),B], P[x,f(B),B] = P[A,f(B),B]
---- is s the simplest unifier?

s is NOT mgu
g of {Ei}
s {Ei} {Ei}s
{Ei}s = {Ei}gs'