Contents

Tabla de Conversion

- Si el v esta en la premisa

If AvB
Then
C =

if A
then
C
If B
Then
C

- Si el ^ esta en la conclusion

If P
Then
A^B =

if P
then
A
If P
Then
B

Si el ^ esta en la conclusion

If P
Then
A(); =
B();

if P
then
A();
If P
Then
B();

- Si el v esta en la conclusión

If P
Then
AvB

Una regla
If P
Then
Q
Expresa una implicación tautológica: P ⇒ Q

/*
p=”estudio” q=”apruebo”
p →q = si estudio entonces apruebo
La implicación puede ser true o false

P = me saco 51 q=aprubo la materia
p ⇒q = si me saco 51 entonces apruebo la materia
La implicación es tautológica (siempre true)
*/

Entonces la regla
If P
Then
AvB
Sera: p ⇒(AvB)

Escribiendo sin la doble línea
p →(AvB) = // ∝→β = ¬ ∝vβ
=¬pv(AvB)
=¬pvAvB

Nota. Cuando los conectivos son los mismos se quita los paréntesis, cason contrario no, se usa la distribución.

=¬(p^¬A)vB= // ¬ ∝vβ =∝→β
=(p^¬A) →B

If p^¬A
Then
B

Pero también se puedo haber hecho esto
¬pvAvB= Conmutativa
=¬pvBvA Ley de morgan
=¬(p^¬B)vA // ¬ ∝vβ =∝→β
=(p^¬B) →A

If p^¬B
Then
A

Resumiendo.

If P
Then
AvB =

if P ^ ¬A
then
B
If P ^ ¬B
Then
A

If P
Then
AvBvC

if P ^ ¬B^ ¬C
then
A
If P ^ ¬A^¬C
Then
B
If P ^ ¬A^¬B
Then
C

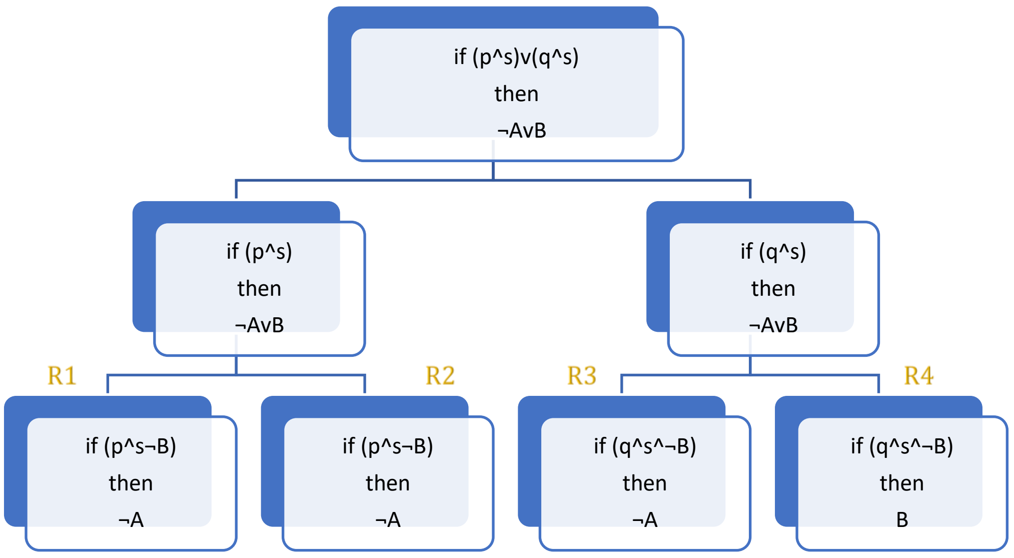
(.) Convertir a reglas esenciales las siguientes regla R
R) if (¬p→q)^s
Then
A→¬B

<u>Solucion:</u>
Lo que hace es convertir la premisa a FND y la conclusión a FNC, para luego aplicar la tabla de conversión (usando el arbol)

Premisa

		Conclusion	
=(¬p→q)^s	∝→β = ¬ ∝vβ	= A→¬B	∝→β = ¬ ∝vβ
=(pvq)^s	Distributividad	= ¬AvB	
=(p^s)v(q^s)	FND de 2 minterm		

La regla R queda
R) if (p^s)v(q^s)
Then
¬AvB



R = {R1, R2, R3, R4}

UNIFICACION DE UNA VARIBLE

La asignacion en un lenguaje procedimental
X = 55
X = 80 //x=~~55~~ 80
La asignación puede cambiar su valor con la asignación (=)

Cuando la variable se unifica con un valor, la variable **es** ese valor
xU = 80 //xU = x unificada a 80

Ahora x ya no puede unificarse a otro valor
xU = 90 //ERROR

Un hecho sintácticamente tiene la forma de una conclusión de regla: var=valor
La Base de hechos (BH) es una lista de hechos, por ejemplo:
BH = {x=30, Edad=25, Presion=Alta}
(La base de hecho no tiene hechos repetidos)

La BH también nos dice:
- Que variables están unificadas, e.g. x, Edad, Presion.
- ¿Qué variables no están unificadas?

Recuerde.
- Si una variable esta en la BH, la variable esta unificada.
- Si una variable no esta en la BH, la variable NO esta unificada y vale null.

BH = [p=alta, z=0]

BH.add(temp=baja) //lo inserta sin problemas

BH = [p=alta, z=0, temp=baja]

BH.add(z=0) //ya esta en la BH, no pasa nada

BH.add(p=media) //no se puede puede modificar(contrdiccion), halt=Error=Exception

BH = [p=alta, z=0, temp=baja]

MODUS PONENS

p →q   true   //si me traes chocoflan entonces voy al cine
p   .   true
q

if p
then   BH = [., ., …]
va=valor

R1) if (nubes $\overset{True}{\neq}$ estratos ^ $\overset{true}{x<20}$)
Then
Presion=baja   ⇒ Presion=baja

BH = [nubes=cirros, x=10]

R2) if ¬p^q^s
Then
¬m
BH = [s, t, q, ¬p]

R3) if ¬p^q^s
Then   //no hay **q**
q
BH = [s, t, r, ¬p]

R4) if ¬p^q^m
Then
z
BH = [s, t, q, ¬m]   //no hay m ≠¬m

SHORT – CUT (Corto Circuito)

v [| | |]   v.length
0 1 2 3

```
I=0;
While(I < v.length && v[i] != 0) {
	I++;
}
```

~~While(v[i] != 0 && i < v.length) {~~ //esto es un ERROR
~~I++;~~
~~}~~

El MP es un SE, testea la premisa sin usar short cut.
Si alguna variable de la premisa no esta unificada, no se testea el MP
Ejemplo:

R) if Q>0 ^ P=a   //P no esta unificada, NI siquiera testeo
Then   //$R \not\Rightarrow_{MP}$
M=si
BH = [Q=20, s=a, ..]

R) if $\overset{false}{p<0}$ ^ $\overset{true}{z<>b}$ //la premisa no se valida (no es true)
Then   //$R \not\Rightarrow_{MP}$
M=si
BH = [p=4, z=c, ..]
Como las varibles de la premisa (p y q) estan unificadas, testeo el MP

El que mas confunde es el "v"
R) if Q<>0 v z<>0 v P=a   //z no esta unificada, NO TESTEO el MP
Then   //$R \not\Rightarrow_{MP}$
Tiempo=frio
BH = [Q=8, P=a, ..]

EL MODUS TOLLENS

Si la BH niega la conclusión, entonces se concluye la negación de la premisa.

If P

Then $\qquad$ R $_{MT}$ ⇒ ¬P

Q

BH = [¬Q, ...]

P →Q $\qquad$ // P →Q ≡ ¬Q → ¬P

BH: P . $\quad$ (MP)

Q

P →Q $\qquad$ ¬Q → ¬P

BH: ¬Q . $\qquad$ BH: ¬Q .

¬p $\qquad$ ¬p

Usaremos el MT Algebraicamente (con letritas)

(.) Realizar el MT con la regla R y la BH dada.

R) if ¬A

Then

¬P $\qquad$ // Antes de hacer el MT, verifique que la conclusión este negada x por un hecho

BH = [P, ...]

Hay MT

R$_{MT}$⇒A $\qquad$ //¬¬A=A

Pero el MT tiene un problema

(.) Realizar el MT entre la regla R y la BH dada.

R) if P^¬Q

Then

¬S

BH = [S, ...]

R $_{MT}$⇒ ¬(P^¬Q) ≡ ¬PvQ

Pero ¬PvQ no puede insertarse (add) a la BH

/*

Recordemos que la BH solo tiene literales (letra o negación de letra)

BH = [p, q, s, ¬r, ...]

*/

Disparo(shoot) de una regla

Se dice que una regla R dispara una conclusión c, anotado: R⇒c

Si lo arrojado por la regla R, puede insertarse a la BH

En el ejemplo anterior, se hizo el MT, pero lo obtenido (¬PvQ) no puede insertarse a la BH.

∴ R $_{MT}$⇒

R) if ¬s ^ t ^ ¬q

Then

p

BH = [¬p, ...]

R $_{MT}$⇒ ¬ (¬s ^ t ^ ¬q) ≡ s v ¬t v q

(.)

R) if ¬p

Then

w

BH = [¬w, ...]

R $_{MT}$⇒ ¬ (¬p) ≡ p $\qquad$ //esto lo puede add a la BH

Silogismo disyuntivo (SD)

p v q $\qquad$ p v ¬s v t $\qquad$ q v ~~¬s~~ v p

BH: ¬p . $\qquad$ BH: s, ¬t . $\qquad$ BH: q, ~~s~~ .

q $\qquad$ p $\qquad$ q v p

Solucion:

R⇒¬(q ^ ¬s ^ t) ≡ ¬q v s v ¬t

Usando Silogismo Disyuntivo

¬q v s v ¬t

BH: s, q, t

s //Esto se add a la BH

(.) Haga el MT con

R) if ¬A ^ P ^ Q

Then

w

BH: [¬w, ¬P, ¬A]

<u>**Solucion**</u>

R⇒ (¬A ^ P ^ Q) ≡ A v ¬P v ¬Q

BH: [¬w, ¬P, ¬A]

¬P, ¬Q //Esto no puede add a la BH

∴ R $_{MT}$ ⇏

Metodo Rapido (MT + SD)

//Solo funciona para reglas simples

/*

If minterm

Then

c

*/

<u>**Ejemplos:**</u>

(.) Usando el MT rapido

R) if ¬A ^ P ^ s ^ w

Then

C

BH = [¬c, ¬A, s, w]

<u>**Solucion:**</u>

Hemos validado todos los literales de la premisa, <u>EXCEPTO UNO</u> (p), son validas.

∴ Concluyo la negacion del que falto validar: R $_{MT}$⇒¬p

(.) Usando el MT rapido

R) if ¬p ^ q ^ ¬s

Then

¬s

BH = [¬s, q]

<u>**Solucion:**</u>

Para aplicar el MT, SOLO UNO debe quedarse sin validar, pero aquí hay 2 (¬p y ¬s)

∴ R $_{MT}$⇏

(.) Si se validan todos

R) if ¬p ^ ¬q ^ ¬s

Then

¬r

BH = [r, ¬p, ¬s, ¬q]

<u>**Solucion:**</u>

Todos los literales de la premisa han sido validados (No hay ninguno sin validar)

∴ R $_{MT}$⇏

MT usando Factores Boole

(Utilizamos el MT)

If p

Then

~~x=3~~

BH = [x<>3, ...]

Solucion

No se puede colocar x ≠ 3, debe ser: var = valor

¿Cuando hay contradiccion entre una conclusión de una regla y un hecho?

Hay contradicción, cuando la conclusión y un hecho tienen la misma variable, pero diferentes valores.

Ejemplo:

R) if ¬p

Then

Var = valor1

BH = [var = valor2]

<u>**Solucion:**</u>

var1 ≠ var2

Negacion de los OPPREL's (operadores relacionales)

$\neg(A\neq B) \equiv A=B$ $\neg(A<B) \equiv A\geq B$ $\neg(A\geq B) \equiv A<B$
$\neg(A=B) \equiv A\neq B$ $\neg(A>B) \equiv A\leq B$ $\neg(A\leq B) \equiv A>B$

(.) Usando el MT, haga el MT con la regla R y la BH dada

R) if p>0 ^ R<>a ^ z<>10
Then
Q=b
BH = [~~Q=c~~, p=20, z=5]

Solucion
El literal (Factor Boole), R<>a no se valida con la BH
$R_{MT} \Rightarrow \neg(R\neq a) \equiv R=a$
$\therefore R_{MT} \Longrightarrow R=a$

Sin embargo, muchas veces el MT puede arrojar factores Boole, que no pueden add a la BH
Por ejemplo:

R) if Q>0 ^ P=alta ^ Temp>100
Then
x=20
BH = [~~x=10~~, Q=50, P=alta]

Solucion
Usando el MT
$R_{MT} \Rightarrow \neg(Temp>100) \equiv Temp\leq 100$, esto no es un hecho
$\therefore$ No se puede add a la BH

/*
Recuerde. Un hecho es: var=valor
*/

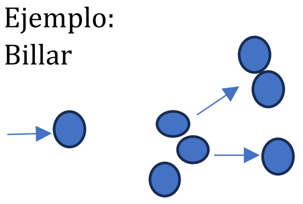
$\therefore R_{MT} \not\Longrightarrow$

Al tener el Mt muchos inconvenientes, los SE solo usan el MP.
Para subsanar esto, utilizan algoritmos del motor de inferencia, llamando Backward-Chaining (BWC).

Algoritmos de encadenamiento (chaining algoritm's)

//Algoritmos de redaccion en cadenas
Se los llama asi porque el disparo de una regla, puede hacer disparar otras reglas.
Ejemplo:

Billar



Reglas
$R1 \Longrightarrow h_1$
h_2 $R2 \Longrightarrow h_3$
$R4 \Longrightarrow h_4$

Existen 2 "criterios" para el encadenamiento.

- **FWC (en profundidad)**
- **BWC (en anchura)**

Forward Chainind (FWC)