REGLAS DE INFERENCIA

1. Modus Ponens (M.P.)

$p \supset q$

p

$\therefore q$

2. Modus Tollens (M.T.)

$p \supset q$

$\sim q$

$\therefore \sim p$

3. Silogismo Hipotético (H.S.)

$p \supset q$

$q \supset r$

$\therefore p \supset r$

4. Silogismo Disyuntivo (D.S.)

$p \vee q$

$\sim p$

$\therefore q$

5. Dilema Constructivo (C.D.)

$(p \supset q) \cdot (r \supset s)$

$p \vee r$

$\therefore q \vee s$

6. Dilema Destructivo (D.D.)

$(p \supset q) \cdot (r \supset s)$

$\sim q \vee \sim s$

$\therefore \sim p \vee \sim r$

7. Simplificación (Simp.)

$p \cdot q$

$\therefore p$

8. Conjunción (Conj.)

p

q

$\therefore p \cdot q$

9. Adición (Ad.)

p

$\therefore p \vee q$

Regla de Reemplazo: Cualquiera de las siguientes expresiones lógicamente equivalentes puede reemplazar a la otra en donde ocurran:

10. Teoremas de De Morgan (De M.):	$\sim(p \cdot q) \equiv (\sim p \vee \sim q)$
	$\sim(p \vee q) \equiv (\sim p \cdot \sim q)$
11. Conmutación (Conm.):	$(p \vee q) \equiv (q \vee p)$
	$(p \cdot q) \equiv (q \cdot p)$
12. Asociación (Asoc.):	$[p \vee (q \vee r)] \equiv [(p \vee q) \vee r]$
	$[p \cdot (q \cdot r)] \equiv [(p \cdot q) \cdot r]$
13. Distribución (Dist.):	$[p \cdot (q \vee r)] \equiv [(p \cdot q) \vee (p \cdot r)]$
	$[p \vee (q \cdot r)] \equiv [(p \vee q) \cdot (p \vee r)]$
14. Doble Negación (D.N.):	$p \equiv \sim\sim p$
15. Transposición (Trans.):	$(p \supset q) \equiv (\sim q \supset \sim p)$
16. Implicación Material (Impl.):	$(p \supset q) \equiv (\sim p \vee q)$
17. Equivalencia Material (Equiv.):	$(p \equiv q) \equiv [(p \supset q) \cdot (q \supset p)]$
	$(p \equiv q) \equiv [(p \cdot q) \vee (\sim p \cdot \sim q)]$
18. Exportación (Exp.):	$[(p \cdot q) \supset r] \equiv [(p \supset (q \supset r)]$
19. Tautología (Taut.):	$p \equiv (p \vee p)$
	$p \equiv (p \cdot p)$

Algoritmo: Conversión a forma clausal

1. *Eliminar las implicaciones*, $\rightarrow$, usando el hecho de que a $\rightarrow$ b es equivalente a $\neg$a v b. Realizando esa transformación en las fbf dadas anteriormente obtenemos:

   $\forall$x: $\neg$[romano(x) ^ conoce(x,Marco)] v

   [odia(x,César) v ($\forall$y: $\neg$($\exists$z: odia(y,z)) v cree_loco(x,y))]

2. *Reducir el ámbito de las negaciones*, $\neg$, a un único término, usando el hecho de que $\neg$($\neg$p) = p, las leyes de Morgan [según las cuales $\neg$(a ^ b) = $\neg$a v $\neg$b y $\neg$(a v b) = $\neg$a ^ $\neg$b], y las correspondencias normales entre cuantificadores [$\neg\forall$x: P(x) = $\exists$x: $\neg$P(x) y $\neg\exists$x: P(x) = $\forall$x: $\neg$P(x)]. Realizando esta transformación en las fbf del paso 1 se obtiene:

   $\forall$x: [$\neg$romano(x) v $\neg$conoce(x,Marco)] v

   [odia(x,César) v ($\forall$y: $\forall$z: $\neg$odia(y,z)) v cree_loco(x,y))]

3. *Normalizar las variables* de forma que cada cuantificador esté ligado a una variable. Puesto que las variables son solo nombres sin un valor concreto, este proceso no puede afectar al valor de verdad de la fbf. Por ejemplo, la fórmula

   $\forall$x: P(x) v $\forall$x: Q(x)

   se convertiría en

   $\forall$x: P(x) v $\forall$y: Q(y)

   Este paso es una preparación para el siguiente.

4. *Mover todos los cuantificadores a la izquierda de la fórmula sin cambiar su orden relativo*. Esto es posible gracias a que no existe ningún conflicto entre los nombres de las variables. Realizando esta operación sobre la fórmula del paso 2, se obtiene:

   $\forall$x: $\forall$y: $\forall$z: [$\neg$romano(x) v $\neg$conoce(x,Marco)] v

   [odia(x,César) v ($\neg$odia(y,z)) v cree_loco(x,y))]

   En este momento la fórmula es lo que se conoce como **fórmula normal prenex**.

   Consiste en un prefijo de cuantificadores seguido por una matriz que está libre de cuantificadores.

5. *Eliminar los cuantificadores existenciales*. En una fórmula donde se incluye una variable cuantificada existencialmente se afirma que existe un valor que puede sustituir a la variable, y que hace verdadera la fórmula. Es posible eliminar el cuantificador sustituyendo la variable por una referencia a una función que produzca el valor deseado. Puesto que no se conoce necesariamente la forma de producir ese valor, se debe crear un nuevo nombre de función para cada sustitución. No se hace ninguna afirmación sobre esas funciones excepto que deben existir. Así, por ejemplo, la fórmula:

   $\exists$y: Presidente(y)

   puede transformarse en la fórmula

   Presidente(S1)

   Donde S1 es una función sin argumentos que produce de algún modo un valor que satisface el predicado Presidente.

   Si surgen cuantificadores existenciales dentro del ámbito de cuantificadores universales, los valores que satisfagan el predicado pueden depender de los valores de las variables cuantificadas universalmente. Por ejemplo, en la fórmula:

   $\forall$x: $\exists$y: padre_de(y,x)

   el valor de y que satisface padre_de depende del valor concreto de x. Por lo tanto, se deben generar funciones con el mismo número de argumentos que el número de cuantificadores universales que afecten a la expresión que se esté tratando. Así, este ejemplo se transformaría en:

   $\forall$x: padre_de(S2(x),x)

Estas funciones que generamos se llaman **funciones de Skolem**. Aquellas que no tienen argumentos se llaman a veces **constantes de Skolem**.

6. *Eliminar el prefijo*. En este punto, *todas las variables que quedan están cuantificadas universalmente*, por lo que el prefijo puede ser simplemente ignorado, y cualquier procedimiento de demostración que usemos puede suponer simplemente que cualquier variable con la que se encuentre, está cuantificada universalmente. Ahora la fórmula producida en el paso 4 aparece como:

   [¬romano(x) v ¬conoce(x,Marco)] v

   [odia(x,César) v (¬odia(y,z)) v cree_loco(x,y))]

7. *Convertir la matriz en una conjunción de disyunciones*. Como en este ejemplo no aparece ninguna conectiva Y, basta con utilizar la propiedad asociativa de la conectiva lógica O [es decir, a v (b v c) = (a v b) v c] y quitar simplemente los paréntesis, para obtener:

   ¬romano(x) v ¬conoce(x,Marco) v

   odia(x,César) v ¬odia(y,z) v cree_loco(x,y)

   Sin embargo, con frecuencia es también necesario utilizar la propiedad distributiva [(a ^ b) v c = (a v c) ^ (b v c)]. Por ejemplo, la fórmula:

   (invierno ^ llevarbotas) v (verano ^ llevarsandalias)

   se convierte después de una aplicación de la regla en:

   [invierno v (verano ^ llevarsandalias)]

   ^ [llevarbotas v (verano ^ llevarsandalias)]

   y, después de una segunda aplicación, que es necesaria, porque aún quedan conjunciones unidas por la conectiva O, en

   (invierno v verano) ^

   (invierno v llevarsandalias) ^

   (lelvarbotas v verano) ^

   (llevarbotas v llevarsandalias)

8. *Crear una cláusula por cada conjunción*. Para que una fbf sea cierta, todas las cláusulas que se han generado a partir de ella deben ser ciertas. Cuando se trabaja con varias fórmulas bien formadas, es posible combinar el conjunto de cláusulas generadas por cada una de ellas para representar los mismos hechos que representaban las fórmulas originales.

9. *Normalizar las variables* que aparecen en el conjunto de cláusulas generadas en el paso 8. Con esto se pretende que no haya dos cláusulas que hagan referencia a la misma variable, para lo cual es necesario renombrar a las variables adecuadamente. Esta transformación se apoya en el hecho:

   $(\forall x: P(x) \wedge Q(x)) = \forall x: P(x) \wedge \forall x: Q(x)$

   Así, puesto que cada cláusula es una conjunción separada y todas las variables están cuantificadas universalmente, no es necesario dar valor a una variable cuantificada universalmente (es decir, sustituirla por un valor concreto). Pero, en general, queremos mantener las cláusulas en su forma más general durante tanto tiempo como sea posible. Al instanciar una variable, queremos conocer el mínimo número de sustituciones que deben realizarse para preservar el valor de verdad del sistema.