Capítulo 1

Lógica Proposicional

La Lógica se ocupa de los métodos del razonamiento. Uno de los objetivos fundamentales es sistematizar y codificar principios de los razonamientos válidos con el objeto de formar o construir argumentaciones o deducciones que sean correstas. Un argumento o deducción consta esencialmente de un conjunto de sentencias (afirmaciones) que forman lo que se llama premisas o hipótesis de las cuales otra sentencia, llamada conclusión, es deducida o inferida. Nosotros nos ocuparemos de analizar en cierto detalle las deducciones. Por ejemplo, si queremos deducir alguna propiedad acerca de los números naturales en la deducción (o también llamada demostración o prueba) de tal propiedad , emplearemos sentencias o propiedades que sabemos a priori que son verdaderas y posiblemente también hipótesis adicionales que no necesariamente son verdaderas. Supongamos que queremos deducir la siguiente propiedad elemental acerca de los números naturales:

$$\text{Para todo } n, m \in \mathbb{N}, \text{ si } n = m + 1, \text{ entonces } m \leq n.$$

En la formulación de esta propiedad interviene la sentencia $n = m + 1$ que obviamente no es necesariamente cierta para todos los números naturales. Pero es evidente que la formulación anterior no afirma que $n = m + 1$ es válida, afirma que en caso de que estemos en presencia de un par de naturales n y m, cumpliendo la propiedad de que $n = m + 1$, entonces será verdadero que $m \leq n$. En la demostración de la anterior propiedad utilizaremos la *hipótesis* ”$n = m + 1\tau$ también propiedades conocidas anteriormente.

Para analizar la estructura de las deducciones necesitamos fijar un lenguaje en donde trabajar. Si por ejemplo, queremos estudiar el idioma Español, necesitaremos conocer primero el alfabeto el cual nos permitirá construir palabras y posteriormente frases cada vez más complejas. De igual forma nosotros deberemos establecer un alfabeto que constará de ciertos símbolos a partir de los cuales y por medio de ciertas reglas construiremos lo que llamaremos fórmulas del lenguaje. En Español toda palabra se puede ver como una secuencia finita de símbolos. Nuestras fórmulas también serán secuencias finitas de símbolos de nuestro lenguaje.

1.1. Lenguaje Proposicional

En el lenguaje ordinario nos encontramos constantemente con sentencias que han sido formadas uniendo frases más pequeñas por medio de ciertas palabras, como las palabras *no, y, o,* y por *si...... entonces (o implica)....., si y sólo si,* etc. Estas palabras son llamadas conectivos proposicionales o conectivos lógicos. Nuestra procupación es estudiar la estructura de sentencias en donde aparecen estos conectivos.

Ahora vamos a definir el lenguaje $\mathcal{L}$ de la Lógica Proposicional Clásica. Este lenguaje es un conjunto de símbolos con los cuales formamos cadenas de elementos de $\mathcal{L}$. Las cadenas no se construyen de una manera arbitraria. Daremos reglas precisas para la formación de dichas cadenas, las cuales serán llamadas fórmulas.

Lenguaje. El lenguaje $\mathcal{L}$ del Cálculo Proposicional Clásico (CPC) consta de las siguientes partes:

1. Un conjunto finito o numerable de símbolos Var, llamados *símbolos proposicionales o variables.* Los elementos de Var serán denotados por p, q, r, etc.

2. Un conjunto de símbolos $\{\neg, \vee, \wedge, \rightarrow\}$, llamados *conectivos proposicionales y cuya interpretación es*

   $\neg$ se interpreta como *negación*
   $\vee$ se interpreta como *o*
   $\wedge$ se interpreta como *y*
   $\rightarrow$ se interpreta como *si...., entonces...*

3. Los símbolos auxiliares (y).

Los símbolos auxiliares sirven para separar distintas componentes de las fórmulas. No deben ser entendidos como símbolos del lenguaje.

Teniendo definido el lenguaje ahora corresponde decir como *unimos* estos símbolos para formar cadenas finitas de símbolos. Las reglas de formación dadas en la siguiente definición nos dice como debemos esto.

Definición 1.1 El conjunto de las ***fórmulas*** Fm es el menor conjunto de cadenas de símbolos de $\mathcal{L}$ que se obtiene aplicando algunas de las siguientes reglas:

1. Toda variable p es un fórmula, es decir, $Var \subseteq Fm$.

2. Si $A \in Fm$, entonces $\neg A \in Fm$,

3. Si $A, B \in Fm$, entonces $A \vee B$, $A \wedge B$ y $A \rightarrow B \in Fm$.

Ejemplo 1.2 Las siguientes cadenas de símbolos son fórmulas

$$p \rightarrow (r \vee \neg q)$$

$$\neg (q \rightarrow p) \vee (p \wedge \neg q) .$$

Las cadenas de símbolos

$$\neg pq \rightarrow q$$

$$\neg (q \rightarrow) \vee \neg p.$$

no son fórmulas pues no están construidas por medio de las reglas dadas en la definición anterior.

En algunos textos de lógica es usual definir el conectivo $\leftrightarrow$, llamado *bi-implicación* y el cual se interpreta como *si y sólo si*. Nosotros lo definiremos a partir de los implicación $\rightarrow$ y de la conjunción $\wedge$ como

$$p \leftrightarrow q = (p \rightarrow q) \wedge (q \rightarrow p) .$$

Notemos que existe una cantidad infinita de fórmulas. Por ejemplo, con la variable p y el símbolo $\neg$ podemos generar infinitas fórmulas de la siguiente forma:

$$p, \neg p, \neg\neg p, \neg\neg\neg p, \ldots\ldots$$

Nuestro próximo objetivo es describir un método inductivo que nos permita asegurar que ciertas propiedades son válidas en todas las fórmulas. Para esto definiremos primero la longitud de una fórmula:

Definición 1.3 La longitud de una fórmula A, en símbolos $long\,(A)$, se define inductivamente como sigue*:*

1. Si $A = p \in Var$, entonces $long\,(p) = 1$.

2. Si $A = \neg B$, con $B \in Fm$, entonces $long\,(A) = long\,(B) + 1$.

3. Si $A = B * C$, donde $*$ es alguno de los conectivos binarios $\vee, \wedge$ *o* $\rightarrow$, entonces $long\,(A) = long\,(B) + long\,(C) + 1$.

Ejemplo 1.4 Determinemos la longitud de la fórmula $A = p \rightarrow (q \vee r)$.

Aplicando la definición anterior tenemos que:

$$\begin{aligned} long\,(A) &= long\,(p) + long\,(q \vee r) + 1 \\ &= long\,(p) + [long\,(q) + long\,(r) + 1] + 1 \\ &= 1 + [1 + 1 + 1] + 1 = 6. \end{aligned}$$

Definición 1.5 Un subconjunto S de cadenas finitas de elementos de $\mathcal{L}$ se dice ***cerrado por los conectivos*** si cumple las siguientes condiciones:

C1 Si $C \in S$, entonces $\neg C \in S$,

C2 Si $C, F \in S$, entonces $C * F \in S$, donde $*$ es algunos de los conectivos binarios $\vee, \wedge$ o $\rightarrow$,

Teorema 1.6 *Sea S un subconjunto de cadenas finitas de elementos de $\mathcal{L}$ cerrado por los conectivos. Si $Var \subseteq S$, entonces $Fm \subseteq S$.*

Demostración. La demostración es por inducción sobre la longitud de las fórmulas. Sea $A \in Fm$. Supongamos que $long\,(A) = 1$. Entonces $A = p$ con p una variable. Por hipótesis $Var \subseteq S$. Por lo tanto $A \in S$.

Supongamos que para toda fórmula A de longitud $long\,(A) < n$, se tiene que $A \in S$. Sea A tal que $long\,(A) = n$. Analizamos los siguientes casos.

Si $A = \neg B$, con $B \in Fm$. Como $long\,(A) = long\,(B) + 1 = n$, entonces $long\,(B) < n$. Luego $B \in S$, y como S es cerrado por conectivos, $\neg B \in S$.

Supongamos que $A = B * C$, donde $*$ es algunos de los conectivos binarios $\vee, \wedge$ o $\to$. Como $long\,(A) = long\,(B) + long\,(C) + 1 = n$, entonces $long\,(B)$, $long\,(C) < n$. Por hipótesis inductiva, $B, C \in S$, y como S es cerrado por los conectivos $B * C \in S$. Por lo tanto $Fm \subseteq S$. ■

El teorema anterior nos dice que dado un lenjuage proposicional $\mathcal{L}$ y un conjunto de variables Var, el conjunto Fm es el *menor conjunto de cadenas de símbolos de* $\mathcal{L}$ *tal que* $Var \subseteq Fm$ *y está cerrado bajo los conectivos.* Este teorema nos permite probar enunciados sobre fórmulas. Analicemos esto en más detalle.

Supongamos que queremos demostrar que todas las fórmulas de $\mathcal{L}$ tienen cierta propiedad P. Consideremos el conjunto

$$\mathcal{C} = \{A \in Fm : A \text{ tiene la propiedad } P\}.$$

Es claro que $\mathcal{C} \subseteq Fm$. Si logramos probar que:

$$Var \subseteq \mathcal{C} \text{ y que } \mathcal{C} \text{ está cerrado bajo los conectivos,}$$

entonces aplicando el Teorema anterior, obtenemos que

$$\mathcal{C} = Fm.$$

Es decir que todas las fórmulas tendrían la propiedad P.

1.1.1. Subfórmulas

Una subfórmula de una fórmula es una subcadena de símbolos de la cadena que define a la fórmula. Más precisamente:

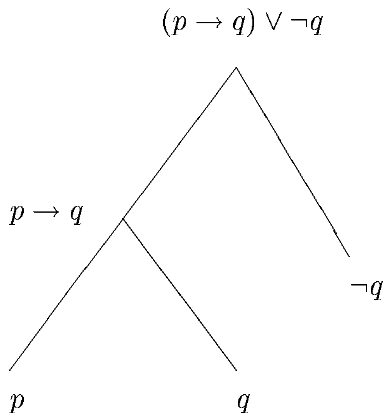
Definición 1.7 Sea A una fórmula. El conjunto de las *subfórmulas* de A se define recursivamente por:

- $Sf\,(p) = \{p\}$, si p es una variable,
- $Sf\,(\neg A) = Sf\,(A) \cup \{\neg A\}$,
- $Sf\,(A * B) = Sf\,(A) \cup Sf\,(B) \cup \{A * B\}$, donde $*$ es cualquiera de los conectivos binarios $\to$, $\vee$ o $\wedge$.

Ejemplo 1.8 Determinemos el conjunto de las subfórmulas de $A = (p \to q) \vee \neg q$.

$$\begin{aligned} Sf\,(A) &= Sf\,((p \to q)) \cup Sf\,(\neg q) \cup \{A\} \\ &= Sf\,(p) \cup Sf\,(q) \cup \{p \to q\} \cup Sf\,(q) \cup \{\neg q\} \cup \{A\} \\ &= \{p\} \cup \{q\} \cup \{p \to q\} \cup \{\neg q\} \cup \{A\}. \end{aligned}$$

El conjunto de todas las subfórmulas de una fórmula dada también puede ser determinado construyendo lo que se llama el *árbol genealógico* de la fórmula. Este árbol representa el proceso de construcción de una fórmula. El árbol genealógico de la fórmula A del ejemplo anterior se muestra en la siguiente figura . Las subfórmulas de A son las fórmulas que figuran en los nodos del árbol genealógico de A.


$(p \rightarrow q) \vee \neg q$
$p \rightarrow q$
$\neg q$
p
q

Capítulo 2

Semántica Proposicional

En el contexto en que estamos trabajando las sentencias tienen la cualidad de que pueden ser clasificadas de dos formas distintas. Por un lado podemos hablar de sentencias válidas o verdaderas. Por ejemplo, si decimos que el número 2 es par, es ciertamente veradadera. Pero también podemos hablar sobre sentencias falsas, como por ejemplo, 2 es menor que 1. Ciudado, no siempre una frase puede ser clasificada en estas dos categorías. Existen muchos ejemplos en donde no es posible afirmar si una sentencia es verdadera o falsa. Por ejemplo, la sentencia *es mañana lloverà* no es posible afirmar si es falsa o verdadera. Nosotros nos ocuparemos solamente de sentencias a las que siempre podamos algún valor de verdad (falso o verdadero).

2.1. Valuaciones

Vamos a definir de una forma precisa como asignar a las fórmulas una interpretación o un valor de verdad. De ahora en adelante vamos a representar el concepto de *verdadero* con el símbolo 1 y *falso* con el símbolo 0. Otras notaciones usuales son, V o $\top$ para verdadero, y F o $\perp$ para falso. Para ello vamos a definir primero las tablas de verdad.

De acuerdo a las interpretaciones dadas a los conectivos lógicos $\vee, \wedge, \rightarrow$ y $\neg$, podemos dar las siguientes tablas, llamadas **tablas de verdad** de los conectivos conectivos

$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

$\rightarrow$	0	1
0	1	1
1	0	1

x	$\neg x$
0	1
1	0

Las tablas anteriores deben ser entendidas como definiciones. Indicaremos con B al conjunto $\{0, 1\}$ dotado de las operaciones binarias $\vee, \wedge, \rightarrow$ y $\neg$ definidas por las tablas anteriores. El lector familiarizado con las álgebras de Boole reconocerá que B es el álgebra de Boole de dos elementos.

Las anteriores tablas nos permiten asignar un valor de verdad a las fórmulas. Cada fórmula es una secuencia finita de símbolos, en donde intervienen variables proposicionales y los conectivos lógicos. Si los valores de verdad de las variables es conocido, entonces el valor de verdad de la fórmula puede ser conocido construyendo la tabla de verdad de la fórmula. Como en las fórmulas solo intervienen una cantidad finita de variables, entonces podemos construir una tabla en donde aparezcan todas las posibilidades de asignación de valores a las variables. Aplicando las definiciones dadas en las tablas anteriores podemos determinar de una forma mecánica el valor de verdad de una fórmula dada.

Ejemplo 2.1 Consideremos la siguiente fórmula $A = (p \vee q) \rightarrow p$. En esta fórmula intervienen solo dos variables. La tabla asociada a la fórmula tendra $2^2 = 4$ filas.

p	q	$p \vee q$	$(p \vee q) \rightarrow p$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	1	1

En el ejemplo anterior hemos construido una tabla que nos permite ver todos los posibles valores de verdad que se le pueden asignar a la fórmula A. Observemos que para cada elección de un valor de verdad para p y q obtenemos un valor de verdad para A. Si pensamos con un poco más de detalle vemos que estamos en presencia de una función del conjunto de variables $\{p, q\}$ en el conjunto $B = \{0, 1\}$. Esta clase de funciones son llamadas valuaciones, valoraciones o asignaciones. De acuerdo a esto, la tabla de verdad de una formula no es más que una descripción ordenada de todas las posibles valuaciones para las variables que ocurren en la fórmula. A continuación vamos a definir exactamente la noción de valuación.

Definición 2.2 Una ***valuación*** o ***asignación*** es una función $v : Var \rightarrow \{0, 1\}$ que cumple las siguientes propiedades para todo par $A, B \in Fm$

V1 $v(\neg A) = \neg v(A)$,

V2 $v(A \vee B) = v(A) \vee v(B)$,

V3 $v(A \wedge B) = v(A) \wedge v(B)$,

V4 $v(A \rightarrow B) = v(A) \rightarrow v(B)$.

Una valuación queda determinada conociendo exactamente lo que asigna a cada variable proposicional, como lo demuestra el sigiuente resultado.

Teorema 2.3 *Sea $v : Var \rightarrow \{0, 1\}$ una función. Entonces existe una única valuación $\bar{v} : Fm \rightarrow B$ tal que $v(p) = \bar{v}(p)$ para toda variable p. Recíprocamente, toda valuación $\bar{v}$ queda determinada por una función $v : Var \rightarrow B$.*

Demostración. Consideremos la función $v : Var \rightarrow \{0, 1\}$. Definimos inductivamente la valuación $\bar{v} : Fm \rightarrow B$ como sigue:

1. $v(p) = \bar{v}(p)$, para cada variable p.

2. Para para todo par $A, B \in Fm$:

   a) $\bar{v}(\neg A) = \neg \bar{v}(A)$,

   b) $\bar{v}(A * B) = \bar{v}(A) * \bar{v}(B)$, siendo $*$ uno de los conectivos lógicos $\vee, \wedge$ o $\rightarrow$.

Es claro por la propia definición de $\bar{v}$ que $v\,(p) = \bar{v}\,(p)$ y que $\bar{v}$ es una valuación. Veamos la unicidad. Supongamos que existe otra valuación w tal que $w\,(p) = v\,(p)$. Consideremos el conjunto

$$X = \{A \in Fm : \bar{v}\,(A) = w\,(A)\}\,.$$

Es claro que $Var \subseteq X$. Además, si $A, B \in X$, entonces $\neg A \in X$ y $A * B \in X$. Por lo tanto X por el Teorema 1.6 $Fm = X$. Es decir, $\bar{v}\,(A) = w\,(A)$ para toda $A \in Fm$. ∎

De acuerdo al Teorema anterior toda valuación $\bar{v}$ queda determinada por la restricción al conjunto Var. De esta forma es costumbre identificar a las valuaciones con funciones $v : Var \to \{0, 1\}$.

Ejemplo 2.4 Consideremos la fórmula $A = (p \vee q) \to p$. La función v definida por

$$v\,(x) = \begin{cases} 1 & x = p \\ \\ 0 & x \neq p \end{cases}$$

es una valuación. El valor, o la interpretación, de la fórmula A bajo esta valuación se obtiene aplicando la definición anterior, es decir:

$$\begin{aligned} v\,(A) &= v\,((p \vee q) \to p) = v\,(p \vee q) \to v\,(p) \\ &= (v\,(p) \vee v\,(q)) \to v\,(p) \\ &= (1 \vee 0) \to 1 = 1 \to 1 = 1. \end{aligned}$$

Una fórmula puede ser verdadera bajo una valuación y falsa bajo otra valuación.

Ejemplo 2.5 En el ejemplo anterior A es verdadera bajo v, pero es falsa bajo la interpretación w definida por

$$w\,(x) = \begin{cases} 0 & x = p \\ \\ 1 & x \neq p \end{cases}$$

En los dos ejemplos anteriores construimos valuaciones primero para las variables que ocurren en la fórmula y despues, por medio de la definición, extendimos la valuación a toda fórmula. Esto es cierto en general. Es decir, para conocer el valor en una fórmula bajo una valuación se necesita conocer que valores asigna la valuación a las variables. Esta afirmación la probaremos en la siguiente proposición.

Si A es una fórmula escribiremos $A\,(p_1, p_2, ..., p_n)$ o $var\,(A) \subseteq \{p_1, p_2, ..., p_n\}$ para indicar que las variables de A están en el conjunto $\{p_1, p_2, ..., p_n\}$.

Proposición 2.6 *Sea A una fórmula tal que $var\,(A) \subseteq \{p_1, p_2, ..., p_n\}$. Si v y w son dos valuaciones tal que*

$$v\,(p_i) = w\,(p_i)$$

para toda $p_i \in \{p_1, p_2, ..., p_n\}$, entonces

$$v\,(A) = w\,(A)\,.$$

Demostración. La prueba se hace por inducción sobre la longitud de la fórmula A.

1. Para $n = 1$, entonces $A = p \in Var$. Luego, por hipótesis, $v(p) = w(p)$.

2. Supongamos que la Proposición vale para toda fórmula de longitud $k < n$.

Estudiemos el caso para $\neg A$. Como $long(A) < long(\neg A) = long(A) + 1$, entonces $v(A) = w(A)$. Luego, $v(\neg A) = \neg v(A) = \neg w(A) = w(\neg A)$. Por lo tanto, vale para $\neg A$.

3. Sean A y B son fórmulas.

Por hipótesis, $v(A) = w(A)$ y $v(B) = w(B)$, pues $long(A), long(B) < long(A * B)$. Entonces, $v(A * B) = v(A) * v(B) = w(A) * w(B) = w(A * B)$. ■

Observación. El resultado anterior afirma que para conocer el valor que asigna una valuación a una fórmula $A(p_1, ..., p_n)$ es suficiente conocer los valores $v(p_1), ..., v(p_n)$. Como toda función definida sobre un conjunto finito queda determinada por la imágen de la función, entonces una valuación v sobre un conjunto finito de variables queda determinada fijando una secuencia finita de ceros y unos de la siguiente manera:

$$v = (v(p_1), ..., v(p_n)) = (e_1, ..., e_n),$$

donde $e_i = 0$ o $e_i = 1$.

Ejemplo 2.7 Consideremos la fórmula $A(p_1, p_2, p_3, p_4) = ((p_1 \to p_2) \wedge \neg p_3) \to \neg p_4$. La secuencia $v = (1, 0, 0, 1)$ es un ejemplo de una valuación. El valor de A bajo esta valuación es

$$\begin{aligned} v(A) = v(((p_1 \to p_2) \wedge \neg p_3) \to \neg p_4) &= ((1 \to 0) \wedge \neg 0) \to \neg 1 \\ &= (0 \wedge 1) \to 0 = 0 \to 0 = 1. \end{aligned}$$

Un importante subconjunto de fórmulas es el de las fórmulas que son válidas bajo cualquier valuación.

Definición 2.8 Una ***tautología*** es una fórmula A que es verdadera bajo cualquier valuación. Es decir, A es una tautología si y sólo si para toda valuación v, $v(A) = 1$. Una ***contradicción*** es una fórmula que es falsa bajo cualquier valuación. Una ***contingencia*** es una fórmula que no es una tautología ni una contradicción.

Las tautologías son aquellas fórmulas que son verdaderas por la configuración de sus símbolos, independientemente de la valuación que se considere. Igual consideración vale para las contradicciones. Notemos que una fórmula A es una tautología si y sólo si su negación $\neg A$ es una contradicción.

Notemos que A es una tautología si en la tabla de verdad todos los elementos de la columna correspondiente a la fórmula A son unos.

Vamos a utlizar el símbolo

$$\models A$$

para indicar que A es una tautología.

Ejemplo 2.9 Veamos que $\models \neg p \vee p$. Debemos comprobar que para toda valuación v, $v(\neg p \vee p) = 1$. Tenemos los siguientes casos:

- Si $v(p) = 1$, entonces $v(\neg p) = \neg v(p) = \neg 1 = 0$. Por lo tanto,

$$v(\neg p \vee p) = \neg v(p) \vee v(p) = 0 \vee 1 = 1.$$

- Si $v(p) = 0$, entonces $v(\neg p) = \neg v(p) = \neg 0 = 1$. Por lo tanto,

$$v(\neg p \vee p) = \neg v(p) \vee v(p) = 1 \vee 0 = 1.$$

Como no hay otros casos para analizar concluimos que la fórmula $\neg p \vee p$ es una tautología.

En la siguiente Proposición damos una lista de fórmulas que son tautologías. Más adelante tendremos oportunidad de dar otras tautologías.

Proposición 2.10 *Para toda fórmulas A, B y C, las siguientes son tautologías.*

1. $A \to A$
2. $A \to (B \to A)$
3. $(A \to B) \to ((B \to C) \to (A \to C))$
4. $(\neg A \to A) \to A$
5. $\neg A \to (A \to B)$
6. $((A \to B) \to A) \to A$
7. $((A \to B) \wedge A) \to B$
8. $A \to A\,(\vee B)$
9. $(A \wedge B \to)\,A$

Más adelante veremos algunos métodos para determinar cuando una fórmula es una tautología.

2.2. Equivalencia lógica

Consideremos las fórmulas $\neg p \vee p$ y $q \to q$ y escribamos sus tablas.

p	$p \vee \neg p$
0	1
1	1

p	$p \to p$
0	1
1	1

Como podemos observar la tabla de $\neg p \vee p$ es idéntica a la tabla de $p \to p$. Es sencillo comprobar que la fórmula $p \wedge \neg p \to p$ también tiene la misma tabla. Es decir, fórmulas distintas pueden tener la misma tabla de verdad. Esta idea la formalizamos en la siguiente definición.

Definición 2.11 Dos fórmulas A y B se dicen ***equivalentes***, en símbolos $A \equiv B$, si y sólo si para toda valuación v, $v(A) = v(B)$.

Es sencillo comprobar que $\equiv$ es una relación de equivalencia en el conjunto de las fórmulas Fm. Es decir, se cumplen las tres propiedades siguientes:

Reflexiva: $A \equiv A$
Simétrica: Si $A \equiv B$, entonces $B \equiv A$, y
Transitiva: Si $A \equiv B$ y $B \equiv C$, entonces $A \equiv C$.

Lema 2.12 *Sean A y B fórmulas. Entonces $A \equiv B$ si y sólo si $v(A \leftrightarrow B) = 1$ para toda valuación v.*

***Demostración*..** Si $A \equiv B$, entonces para cualquier valuación v, $v(A) = v(B)$. Por lo tanto, $v(A \to B) = v(A) \to v(A) = 1$ y $v(B \to A) = v(B) \to v(B) = 1$. Por lo tanto, $v(A \leftrightarrow B) = 1$.

Supongamos que $v(A \leftrightarrow B) = 1$. Es decir, $v(A \to B) = 1$ y $v(B \to A) = 1$. Si suponemos que $v(A) \neq v(B)$, entonces se pueden dar los siguientes casos:

1. $v(A) = 1$ y $v(B) = 0$. En este caso, $v(A \to B) = 1 \to 0 = 0$.
2. Si $v(A) = 0$ y $v(B) = 1$, entonces $v(B \to A) = 1 \to 0 = 0$.

En cualquier caso, obtenemos una contradicción. Por lo tanto, $v(A) = v(B)$. ■

A continuación damos una lista de fórmulas lógicamente equivalentes. Las demostraciones son sencillas y se dejan como ejercicios.

Proposición 2.13 *Las siguientes son fórmulas lógicamente equivalentes:*

1. $A \equiv \neg\neg A$
2. $A \equiv A \wedge A \qquad A \equiv A \vee A$
3. $A \wedge B \equiv B \wedge A \quad A \vee B \equiv B \vee A$
4. $A \vee \neg A \equiv A \to A$
5. $A \to B \equiv \neg B \to \neg A$
6. $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
7. $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
8. $\neg(A \vee B) \equiv \neg A \wedge \neg B$
9. $\neg(A \wedge B) \equiv \neg A \vee \neg B$
10. $A \vee B \equiv \neg(\neg A \wedge \neg B)$
11. $A \wedge B \equiv \neg(\neg A \vee \neg B)$

Consideremos dos fórmulas arbitrarias A y B. Es sencillo comprobar que las siguientes equivalencias son válidas:

$$A \wedge \neg A \equiv B \wedge \neg B \tag{2.1}$$

$$A \vee \neg A \equiv B \vee \neg B \tag{2.2}$$

Observemos que para cualquier fórmula A la fórmula $A \wedge \neg A$ es insatisfacible, y que la fórmula $A \vee \neg A$ es una tautología. Es conveniente utilizar símbolos especiales para este tipo de fórmulas. Denotaremos con $\perp$ a cualquier fórmula del tipo $A \wedge \neg A$, y denotaremos con $\top$ a cualquier fórmula del tipo $A \vee \neg A$. En otras palabras, toda contradicción se identificará con el símbolo $\perp$ y toda tautología se identificará con el símbolo $\top$.

2.3. Sustituciones

Definición 2.14 Una *sustitución* es cualquier función

$$e : Fm \to Fm$$

tal que cumpla las siguientes condiciones:

1. $e(\neg A) = \neg e(A)$,
2. $e(A \wedge B) = e(A) \wedge e(B)$,
3. $e(A \vee B) = e(A) \vee e(B)$,
4. $e(A \to B) = e(A) \to e(B)$.

Es posible probar que toda función $f : Var \to Fm$ se puede extender a una sustitución $e : Fm \to Fm$ tal que $f(p) = e(p)$ para toda variable p. En consecuencia para definir una sustitución sobre un fórmula $A(p_1, p_2, ..., p_n)$ solo es necesario definir la sustitución sobre el conjunto de variables $\{p_1, p_2, ..., p_n\}$ y despues extender la sustitución a la fórmula A aplicando la definición.

Ejemplo 2.15 Sea $A = p \to (q \to p)$. Entonces las siguientes funciones son sustituciones:
$e_1 : Var \to Fm$ definida por

$$e_1(x) = \begin{cases} q \to p & \text{si } x = p \\ q & \text{si } x \neq p \end{cases}$$

$e_2 : Var \to Fm$ definida por

$$e_2(x) = \begin{cases} q & \text{si } x = p \\ p \to r & \text{si } x \neq p \end{cases}$$

Lema 2.16 *Sea $e : Fm \to Fm$ una sustitución y $v : Fm \to \{0,1\}$ una valuación. Entonces la composición de v con e*

$$v' = v \circ e : Fm \to \{0,1\}$$

es una valuación.

Demostración. Ejercicio. ■

Lema 2.17 *Sean $A, B \in Fm$. Entonces,*

$$A \equiv B \Leftrightarrow e(A) \equiv e(B),$$

para toda sustitución e.

Demostración. Ejercicio. ■

Lema 2.18 *Sean $A, B \in Fm$.*

1. *Si A es una tautología, entonces $e(A)$ es una tautología, para cualquier sustitución e.*
2. *Si $A \equiv$, entonces $e(A) \equiv e(B)$, para cualquier sustitución e.*

Demostración. Ejercicio. ■

2.4. Formas Normales.

En esta sección probaremos que toda fórmula A es lógicamente equivalente a una fórmula B en donde solo figuran los conectivos $\vee, \wedge$ y $\neg$.

Definición 2.19 Un ***literal*** es una variable proposicional o la negación de una variable proposicional. El conjunto $\{p, \neg p\}$ se llama un par complementario de literales.

Definición 2.20 Una ***cláusula*** es un fórmula del tipo $C = \bigvee l_i$, con l_i literales. *Una* ***cláusula dual*** *es* una fórmula del tipo $C = \bigwedge l_i$, donde l_i son literales.

La siguiente es un ejemplo de cláusula

$$C_1 = p \vee \neg q \vee r$$

Definición 2.21 Sea A una fórmula con $var(A) = \{p_1, p_2, ..., p_n\}$.

1. Diremos que A está en la ***forma normal conjuntiva (fnc)*** si es una conjunción de cláusulas, es decir: A es de la forma

$$A = \bigwedge C_j = \bigwedge_{j=1}^{n} \bigvee_{i=1}^{m} l_{ij},$$

donde l_{ij} son literales.

2. Diremos que A está en la ***forma normal disyuntiva (fnd)*** si es una disyunción de clausulas duales, es decir: A está en la forma
$$A = \bigvee_{j=1}^{n} \bigwedge_{i=1}^{m} l_{ij},$$
donde l_{ij} son literales.

Ejemplo 2.22 La fórmula
$$(\neg p \vee q \vee r) \wedge (q \vee \neg r) \wedge \neg p$$
está en la fnc. La fórmula
$$(p \wedge q \wedge \neg r) \vee (q \wedge \neg r) \vee (\neg p \wedge r)$$
está en la fnd. En cámbio las fórmulas
$$(p \wedge q \wedge \neg r) \vee ((q \wedge \neg r) \vee p) \vee (\neg p \wedge r)$$
y
$$\neg (p \wedge q \wedge \neg r) \vee (\neg p \wedge r)$$
no está en la fnd ni en la fnc.

Ahora vamos dar un método para obtener, a partir de a una fórmula dada, otra fórmula equivalente escrita en la fnd o en la fnc.

Consideremos una fórmula A. Para obtener una fórmula B que esté en la fnd o en la fnc y que $A \equiv B$ utilizamos las equivalencias dadas en la Proposición 2.13.

Los pasos a seguir para convertir una fórmula en la **fnc** son los siguientes:

1. Utilizar la equivalencia
$$A \rightarrow B \equiv \neg A \vee B$$
para eliminar el conectivo $\rightarrow$.

2. Utilizar las leyes de De Morgan para poner la negación inmediatamente ántes de las fórmulas.
$$\neg (A \vee B) \equiv \neg A \wedge \neg B$$
$$\neg (A \wedge B) \equiv \neg A \vee \neg B$$

3. Utilizar
$$\neg\neg A = A$$
para eliminar la doble negación.

4. Utilizar las leyes distributivas para eliminar las conjunciones entre disyunciones:
$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$$
$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C) .$$

5. Utilizar las leyes

$$A \equiv A \vee A$$
$$A \equiv A \wedge A$$

para eliminar repeticiones de fórmulas.

Ejemplo 2.23 Determinar una fórmula equivalente a la fórmula A que este en la forma normal disyuntiva.

$$A = \neg (p \rightarrow (q \wedge r)) \rightarrow (\neg (p \vee q) \wedge r) .$$

$$\begin{aligned} \neg (p \rightarrow (q \wedge r)) \rightarrow (\neg (p \vee q) \wedge r) &\equiv \neg (\neg (p \rightarrow (q \wedge r)) \vee (\neg (p \vee q) \wedge r)) \\ &\equiv (p \rightarrow (q \wedge r)) \vee (\neg (p \vee q) \wedge r) \\ &\equiv (\neg p \vee (q \wedge r)) \vee (\neg (p \vee q) \wedge r) \\ &\equiv \neg p \vee (q \wedge r) \vee (\neg p \wedge \neg q \wedge r) . \end{aligned}$$

Teorema 2.24 *Toda fórmula A es equivalente a una fórmula en la forma normal disyuntiva y a una fórmula en la forma normal conjuntiva.*

2.5. Consecuencia semántica

En forma cotidiana utilizamos argumentos para aseverar cosas o hechos. Un argumento es una deducción que permite llegar a una conclusión a partir de un cierto conjunto de premisas o hipótesis. Nuestro objetivo es intentar formalizar el concepto de argumentación desde un punto de vista semántico. Para motivar la definición de relación de consecuencia vamos a analizar un ejemplo:

Supongamos que tenemos tres conjuntos X, Y y Z tal que $X \subseteq Y$ e $Y \subseteq Z$. Por teoría elemental de conjuntos sabemos que es posible probar que $X \subseteq Z$. Esto lo podemos escribir también de la siguiente forma:

Si las inclusiones $X \subseteq Y$ e $Y \subseteq Z$ son verdaderas, entonces es verdadera la inclusión $X \subseteq Z$.

Lo anterior es un ejemplo elemental de lo que es una argumentación. Las inclusiones $X \subseteq Y$ e $Y \subseteq Z$ son las hipótesis y la inclusión $X \subseteq Z$ es la conclusión. Si denotamos por A a la proposición $X \subseteq Y$, B a la proposición $Y \subseteq Z$ y C a la proposición $X \subseteq Z$, entonces la anterior argumentación se puede escribir como:

Si proposiciones A y B son verdaderas, entonces es verdadera la proposición C.

Usando la noción de asignación o valuación, la anterior argumentación se puede interpretar como sigue:

Para cada valuación v tal que $v(A) = 1$ y $v(B) = 1$ entonces $v(C) = 1$.

Para simboilizar la anterior situación utilizaremos el símbolo $\models$ (ya utilizado para denotar a las tautologías) de consecuencia semántica. Luego la situación planteada en nuestro ejemplo queda simbolizada como:

$$\{A, B\} \models C.$$

Veamos otro ejemplo.

Ejemplo 2.25 Probemos que si fórmulas $p \to q$, $q \to r$ son verdaderas bajo una valuación, entonces la fórmula $p \to r$ es verdadera bajo la misma valuación. Es decir, queremos probar que

$$\{p \to q, q \to r\} \models p \to r.$$

Para comprobar este hecho vamos a construir una tabla de verdad para las fórmulas $(p \to q)$, $(q \to r)$ y para la conclusión $p \to r$. En aquellas filas donde se asigne el valor 1 a la premisa debe también asigna el valor 1 a la conclusión.

p	q	r	$p \to q$	$q \to r$	$p \to r$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	0	
0	1	1	1	1	1
1	0	0	0	1	
1	0	1	1	1	1
1	1	0	0	0	
1	1	1	1	1	1

En los casos donde el conjunto de hipótesis toma el valor 1 la consecuencia también toma el valor 1. Por lo tanto el argumento es verdadero.

Definición 2.26 Diremos que una valuación v ***satisface*** una fórmula A si $v(A) = 1$.

Diremos que una valuación v *satisface* un conjunto de fórmulas Γ si $v(A) = 1$ para toda fórmula $A \in \Gamma$. En este caso escribiremos $v(\Gamma) = 1$.

Una fórmula o un conjunto de fórmulas se dice *satisfacible* si existe alguna valuación que la (lo) satisfaga. Diremos que es *insatisfacible* en caso contrario.

Observemos que una fórmula es satisfacible si y sólo si es una contingencia o una tautología, y es insatisfacible si y sólo si es una contradicción.

Ejemplo 2.27 Estudiar si alguno de los siguientes conjuntos es satisfacible:

1. $S_1 = \{\neg p \vee q, q \to \neg p, p\}$. Si suponemos que S_1 es satisfacible, entonces debe existir una valuación v tal que
$$v(\neg p \vee q) = 1, v(q \to \neg p) = 1 \text{ y } v(p) = 1.$$
Ahora, $v(p) = 1$ implica que $1 = v(\neg p \vee q) = 0 \vee v(q)$. Por lo tanto $v(q) = 1$. Como $v(p) = 1$ y $v(q) = 1$, entonces $v(q \to \neg p) = 1 \to 0 = 0$, lo que es una contradicción. Por lo tanto no puede existir una valuación que satisfaga a las tres fórmulas simultaneamente, es decir el conjunto S_1 es insatisfacible.

2. $S_2 = \{p \to (q \to \neg r), p \wedge (q \vee \neg r)\}$. Supongamos que v es una valuación tal que
$$v(p \to (q \to \neg r)) = 1 \; y \; v(p \wedge (q \vee \neg r)) = 1.$$
De la última igualdad deducimos que $v(p) = 1 \; y \; v(q \vee \neg r) = 1$. Entonces,
$$v(p \to (q \to \neg r)) = 1 \to v(q \to \neg r) = 1.$$

Por lo tanto debe ocurrir que $v(q \to \neg r) = 1$. Si ahora suponemos que $v(q) = 1$, entonces deber ocurrir que $v(\neg r) = 1$, es decir, $v(r) = 0$. Por lo tanto, la valuación v definida por $v(p) = 1$, $v(q) = 1$ y $v(r) = 0$ satisface el conjunto S_2. Es decir, S_2 es satisfacible.

Observemos que una valuación v que satisface a un conjunto de fórmulas $\{A_1, A_2, ..., A_n\}$ es una fila de la tabla de verdad de la fórmula $A_1 \wedge A_2 \wedge ... \wedge A_n$ que de el valor 1.

Ahora estamos en condiciones de formalizar el concepto de consecuencia.

Definición 2.28 Sea $\Gamma \cup \{A\} \subseteq Fm$. Diremos que A es ***consecuencia semántica*** de Γ, en símbolos

$$\Gamma \vDash A,$$

si para toda valuación v que satisfaga a Γ, también satisface a la fórmula A. Esto es:

$$\Gamma \vDash A \Leftrightarrow \textit{para toda valuación } v\textit{, tal que } v(\Gamma) = 1\textit{, entonces } v(A) = 1.$$

De la definición anterior resulta que

$$\emptyset \vDash A \text{ si y sólo si } A \text{ es una tautología.}$$

En los siguientes resultados damos algunas propiedades que serán de utilidad más adelante.

Proposición 2.29 *Las siguientes propiedades son válidas para todo par de conjuntos* $\Gamma, \Delta \subseteq Fm$.

1. *Si* $A \in \Gamma$, *entonces* $\Gamma \vDash A$.
2. *Si* $\Gamma \vDash A$ *y* $\Gamma \subseteq \Delta$, *entonces* $\Delta \vDash A$.
3. *Si* $\Gamma \vDash A$ *y* $A \vDash B$, *entonces* $\Gamma \vDash B$,

Demostración. Ejercicio. ■

El siguiente resultado, conocido como Teorema de la Deducción (versión semántica), pone de manifiesto la relación entre el conectivo binario $\to$ y la noción de consecuencia.

Teorema 2.30 (de la Deducción) *Sea el conjunto* $\Gamma \cup \{A, B\} \subseteq Fm$. *Entonces*

$$\Gamma \cup \{A\} \vDash B \textit{ si y sólo si } \Gamma \vDash A \to B.$$

Demostración. Ejercicio. ■

Corolario 2.31 *Para todo par de fórmulas* A, B *se tiene que*

$$A \vDash B \textit{ si y sólo si } \vDash A \to B.$$

Corolario 2.32 *Para todo conjunto* $\Gamma \cup \{A\} \subseteq Fm$,

$$\Gamma \vDash A \textit{ si y sólo si } \Gamma \cup \{\neg A\} \textit{ es insatisfacible.}$$

Demostración. Ejercicio. ■

Proposición 2.33 *Consideremos las fórmulas $A_1, ..., A_n, A$. Entonces las siguientes condiciones son equivalentes:*

1. $\{A_1, ..., A_n\} \models A$.
2. $\{A_1 \wedge ... \wedge A_n\} \models A$.
3. $\models (A_1 \wedge ... \wedge A_n) \rightarrow A$.

2.6. Teorema de Compacidad

Cuando desarrollamos un argumento por lo general partimos de un conjunto finito de hipótesis. Durante la argumentación utilizamos este conjunto finito de hipótesis u otras sentencias que deducimos a partir del conjunto de hipótesis por medio de alguna regla de deducción. Cualquier argumentación usual consta de una cantidad finita de pasos. Es decir, las argumentaciones son esencialmente procedimientos finitarios. El próximo teorema que probaremos afirma que una fórmula A es consecuencia de un conjunto de fórmulas Γ si y sólo si existe un subconjunto finito Γ_0 de Γ tal que A es consecuencia de Γ_0. Este teorema se lo conoce como teorema de Compacidad y es uno de los resultados más importantes de la Lógica Matemática. Existen diversas versiones de este teorema, aunque todas son equivalentes.

Teorema 2.34 *Sea $\Gamma \subseteq Fm$. Entonces Γ es satisfacible si y sólo si todo subconjunto finito de Γ es satisfacible.*

Demostración. Si Γ es satisfacible por una valuación v, entonces es claro que cualquier subconjunto de Γ es también satisfacible. En particular todo subconjunto finito es satisfacible.

Probemos la otra dirección. Supongamos que todo subconjunto finito de Γ es satisfacible.

Como el conjunto de todas las variables es un conjunto numerable, entonces podemos enumerar todas las variables que ocurren en alguna fórmula de Γ. Supongamos que tal conjunto es:

$$Var(\Gamma) = \{p_1, p_2, ..., p_n, ...\} .$$

Vamos a probar que existe una sucesión $(e_n)_{n\in\mathbb{N}}$ con $e_i \in \{0, 1\}$ tal que la valuación v definida por

$$v(p_n) = e_n$$

satisface a Γ, es decir, $v(\Gamma) = 1$.
La prueba es por inducción sobre n.

Primero debemos definir e_0. Consideremos los siguientes casos:

a_0. Para todo subconjunto finito Δ de Γ existe una valuación v_Δ tal que $v_\Delta(\Delta) = 1$ y $v_\Delta(p_0) = 0$. En este caso definimos

$$e_0 = 0.$$

b_0. Existe un subconjunto finito Δ_0 de Γ tal que para toda valuación v_{Δ_0} que cumple que $v_{\Delta_0}(\Delta_0) = 1$, entonces $v_{\Delta_0}(p_0) = 1$. En este caso definimos:

$$e_0 = 1.$$

Observemos que si se cumple el caso b_0 entonces también se cumple la siguiente condición:

Para todo subconjunto finito Δ de Γ existe una valuación v_Δ tal que $v_\Delta(\Delta_0) = 1$ y $v_{\Delta_0}(p_0) = 1$.

En efecto. Sea Δ un subconjunto finito de Γ. Por la condición b_0, existe un subconjunto Δ_0 finito de Γ. Entonces el conjunto $\Delta \cup \Delta_0$ es también finito. Por la hipótesis del teorema (todo conjunto finito es satisfacible) existe alguna valuación v tal que $v(\Delta \cup \Delta_0) = 1$. Entonces, $v(\Delta) = 1$ y $v(\Delta_0) = 1$. Ahora, por b_0, $v(p_0) = 1$.

Por lo tanto, por la definición de e_0 podemos asegurar que se cumple la siguiente propiedad:

(R_0) *Para todo conjunto finito Δ de Γ existe una valuación v_Δ tal que $v_\Delta(\Delta) = 1$ y $v_\Delta(p_0) = e_0$.*

Hipótesis Inductiva (HI)

Suponemos que hemos definido la sucesión de elementos de $\{0, 1\}$

$$e_0, e_1, ..., e_n$$

de tal forma que se cumple la propiedad (R_n) siguiente:

(R_n) *Para todo conjunto Δ finito de Γ existe una valuación v_Δ tal que $v_\Delta(\Delta) = 1$, $v_\Delta(p_0) = e_0$, $v_\Delta(p_1) = e_1, ..., v_\Delta(p_{n-1}) = e_{n-1}$ y $v_\Delta(p_n) = e_n$.*

Para definir e_{n+1} tenemos dos posibles casos:

a_{n+1} Para todo subconjunto finito Δ de Γ existe una valuación v_Δ tal que $v_\Delta(\Delta) = 1$ y $v_\Delta(p_0) = e_0$, $v_\Delta(p_1) = e_1, ..., v_\Delta(p_n) = e_n$ *y* $v_\Delta(p_{n+1}) = 0$. En este caso definimos

$$e_{n+1} = 0.$$

b_{n+1} Existe un subconjunto finito Δ_{n+1} de Γ tal que para toda valuación $v_{\Delta_{n+1}}$ tal que $v_{\Delta_{n+1}}(\Delta_{n+1}) = 1, v_\Delta(p_0) = e_0, v_\Delta(p_1) = e_1, ...$ y $v_\Delta(p_n) = e_n$, entonces satisface a p_{n+1}, es decir, $v_{\Delta_0}(p_{n+1}) = 1$. En este caso definimos:

$$e_{n+1} = 1.$$

Probemos que se cumple la propiedad (R_{n+1}).

Sea Δ un subconjunto finito de Γ. Si se cumple el caso a_{n+1}, entonces existe una valuación v_Δ tal que $v_\Delta(\Delta) = 1$ y además $v_\Delta(\Delta) = 1, v_\Delta(p_0) = e_0, v_\Delta(p_1) = e_1, ..., v_\Delta(p_n) = e_n$ y $v_\Delta(p_{n+1}) = 0$.

Supongamos que se cumple el caso b_{n+1}. Consideremos el subconjunto finito Δ_{n+1} cuya existencia asegura b_{n+1}. Como $\Delta \cup \Delta_{n+1}$ es también finito, entonces por la propiedad (R_n) existe una valuación v tal que $v(\Delta \cup \Delta_{n+1}) = 1$ y $, v(p_0) = e_0$, $v(p_1) = e_1, ...$ y $v(p_n) = e_n$. Como $v(\Delta_{n+1}) = 1$, entonces por b_{n+1}, $v(p_{n+1}) = 1$. Con esto hemos probado que se cumple la propiedad (R_{n+1}).

Por lo tanto hemos definido inductivamente la sucesión $(e_n)_{n\in\mathbb{N}}$, con $e_n \in \{0,1\}$ y probado que la propiedad (R_n) se verifica para todo $n \geq 0$.

Ahora definimos la valuación v como

$$v(p_n) = e_n.$$

Comprobemos que $v(\Gamma) = 1$.

Sea $A \in \Gamma$. Supongamos que el conjunto de variables de A está incluido en el conjunto $\{p_0, p_1, ..., p_k\}$. Por la propiedad (R_k) y por el hecho que el conjunto $\{A\}$ es finito, existe una valuación w tal que $w(A) = 1$ y $w(p_i) = e_i$ para $0 \leq i \leq k$. Ahora, como $v(p_i) = w(p_i)$para $0 \leq i \leq k$, entonces por el Lema 2.6 deducimos que $w(A) = v(A)$. Por lo tanto, $v(A) = 1$.

Con esto hemos probado que $v(\Gamma) = 1$. ■

El siguiente Corolario es de mucha utilidad en el capítulo correspondiente a Resolución.

Corolario 2.35 *Un conjunto de fórmulas Γ es insatisfacible si y sólo si existe un subconjunto finito de Γ insatisfacible.*

Teorema 2.36 *Sea $\Gamma \cup \{A\}$ un conjunto de fórmulas. Entonces $\Gamma \vDash A$ si y sólo si existe un subconjunto finito Γ_0 de Γ tal que $\Gamma_0 \vDash A$.*

Demostración. Supongamos que $\Gamma \vDash A$. Por el Corolario 2.32, $\Gamma \vDash A$ es equivalente a que el conjunto $\Gamma \cup \{\neg A\}$ sea insatisfacible y por el Corolario anterior esto es equivalente a decir que existe un subconjunto finito Γ_0 de $\Gamma \cup \{\neg A\}$ que es insatisfacible. Nuevamente por el Corolario 2.32 deducimos que $\Gamma_0 \vDash A$. ■

Capítulo 3

Resolución Proposicional

En el Capítulo anterior estudiamos como asignar valores de verdad a fórmulas. Es decir, trabajamos desde un punto de vista semántico. Ahora estamos interesados en estudiar como deducir mecánicamente fórmulas a partir de un conjunto de fórmulas. Nuestro objetivo es determinar algún procedimiento *mecánico* de deducción de fórmulas. Existen varios métodos de deducción. El más conocido desde el punto de vista de la matemática es el método axiomático. A partir de un conjunto de axiomas, alguna regla de deducción y una noción formal de deducción obtenemos otras fórmulas llamadas teoremas. Otro posible enfoque es el método de deducción natural y/o de Gentzen. En este caso tenemos predominan las reglas en vez de los axiomas. Un tercer método es el de resolución, el cual estudiaremos en este curso. El método de resolución se basa en la idea que para probar que una fórmula es válida podemos probar que su negación es una fórmula contradictoria. Esta clase de métodos de deducción se conocen generalmente como métodos de deducción por refutación. Una de las ventajas del método de resolución radica en el hecho que solo se utiliza una regla. Los fundamentos teóricas de la Programación Lógica (PROLOG) están basadas en este método.

Para motivar la idea de este método, recordemos que si A y B son fórmulas, entonces

$$\{A\} \vDash B \;\Leftrightarrow \text{el conjunto } \{A, \neg B\} \text{ es insatisfacible.} \tag{3.1}$$

y

$$B \text{ es una tautología} \Leftrightarrow \neg B \text{ es una contradicción.}$$

El argumento anterior se extiende cuando tenemos conjuntos finitos de fórmulas como hipótesis de la siguiente forma.

$$\{A_1, ..., A_n\} \vDash B \Leftrightarrow \{A_1, ..., A_n, \neg B\} \text{ es insatisfacible.}$$

Las observaciones anteriores afirman que para saber si una deducción $\{A_1, ..., A_n\} \vDash B$ es válida es suficiente probar que el conjunto $\{A_1, ..., A_n, \neg B\}$ es insatisfacible. Este forma de deducción se conoce como *deducción por refutación.* Esta manera de deducción es la base del método de resolución.

Ejemplo 3.1 Supongamos que $\Sigma = \{p \to q, \neg p \to r, q \vee r \to s\}$. Queremos saber si $\Sigma \vDash s$. Esto es equivalente a probar que $\Sigma \cup \{\neg s\}$ es insatisfacible. Transformamos cada fórmula de Σ en una fórmula que esté en la f.n.c. Entonces

$$\Sigma = \{\neg p \vee q, p \vee r, \neg (q \vee r) \vee s\} = \{\neg p \vee q, p \vee r, (\neg q \vee s) \wedge (\neg r \vee s)\}.$$

Ahora es sencillo comprobar que el conjunto

$$\Sigma \cup \{\neg s\} = \{\neg p \vee q, p \vee r, \neg q \vee s, \neg r \vee s, \neg s\}$$

es insatisfacible. Por lo tanto, podemos asegurar que $\Sigma \vDash s$.

Notemos que probar que una fórmula A es insatisfacible es equivalente a probar que $A \models \perp$.

3.1. Cláusulas

Cuando trabajamos con fórmulas escritas en la forma normal conjuntiva $A = C_1 \wedge \ldots \wedge C_n$, donde C_i son disyunciones de literales, el orden en que están escritas las fórmulas $C_1, ..., C_n$ es irrelevante. De igual forma, en una disyunción de literales $C = l_1 \vee l_2 \vee \ldots \vee l_k$ el orden en que aparecen los literales l_i es irrelevante. Esto sugiere tratar a las fórmulas escritas en la forma normal conjuntiva como un *conjunto* de fórmulas, donde cada fórmula del conjunto es a su vez un conjunto de literales. Teniendo en cuenta este hecho vamos a redefinir la noción de cláusula o mejor dicho vamos a escribir una cláusula como un conjunto de literales y a las fórmulas escritas en la forma normal conjuntiva como un conjunto de cláusulas:

- Una cláusula $C = l_1 \vee l_2 \vee \ldots \vee l_m$ se escribirá como un conjunto de finito de literales $C = \{l_1, l_2, \ldots, l_k\}$.
- Una cláusula C es **unitaria** si solo tiene un literal, es decir, si $C = \{l\}$.
- Una fórmula A que está escrita en la forma normal conjuntiva

  $$A = \bigwedge_{j=1}^{n} C_j = \bigwedge_{j=1}^{n} \bigvee_{i=1}^{m} l_{ij}$$

  se puede escribir como un conjunto de cláusulas de la siguiente manera

  $$cl\,(A) = \{C_1, C_2, ..., C_n\}.$$

  donde $cl\,(A)$ denota la forma clausular asociada a la fórmula A. Como cada cláusula se puede escribir como un conjunto de literales entonces también podemos escribir

  $$cl\,(A) = \{C_1, C_2, ..., C_n\} = \{\{l_{11}, l_{12}, ..., l_{1m_1}\}, ..., \{l_{n1}, l_{n2}, ..., l_{nm_k}\}\}.$$

Por una cuestión de comodidad notacional, en muchos casos no escribiremos a las cláusulas como un conjunto de literales.

Definición 3.2 Diremos que una fórmula A está escrita en la forma ***clausular*** si esta escrita como un conjunto de cláusulas.

Observación. Para escribir una fórmula A es la forma clausular primero debemos pasarla a la forma normal conjuntiva.

Ejemplo 3.3 La fórmula $A = (p \vee \neg q) \wedge (\neg p \vee q \vee r) \wedge \neg r$ tiene la siguiente forma clausular asociada

$$cl\,(A) = \{p \vee \neg q, \neg p \vee q \vee r, \neg r\}\,,$$

o

$$cl\,(A) = \{\{p, \neg q,\}\,, \{\neg p, q, r\}\,, \{\neg r\}\}$$

si escribimos todo en notación de conjuntos.

Definición 3.4 Sea $C = \{l_1, l_1, ..., l_n\}$ una cláusula. Diremos que C es ***satisfacible*** por una valuación v si la fórmula $l_1 \vee l_1 \vee ... \vee l_n$ es satisfacible por v. De igual forma, una fórmula A escrita en la forma clausular $A = \{C_1, ..., C_n\}$ es satisfacible por una valuación v si la fórmula $C_1 \wedge C_2 \wedge ... \wedge C_n$ es satisfacible por v.

Notemos que una cláusula $C = \{l_1, l_1, ..., l_n\}$ es satisfacible por una valuación v si $v\,(l_i) = 1$ para algun literal l_i que interviene en C.

El siguiente resultado es inmediato.

Lema 3.5 *Sea A una fórmula. Entonces A es satisfacible si y sólo si la forma clausular asociada $cl\,(A) = \{C_1, ..., C_n\}$ es satisfacible.*

Observación. Como una cláusula es un *conjunto* de literales $C = \{l_1, l_1, ..., l_n\}$, entonces también debemos admitir la presencia del *conjunto vacío de literales* $\{\ \}$. La cláusula que no tiene literales se llama *cláusula vacía.* Luego, podemos tener un conjunto de cláusulas que solo tenga la cláusula vacía $S = \{\{\ \}\}\,.$

Nuestro próximo objetivo es probar que el conjunto vacío de cláusulas es satisfacible y que el conjunto de cláusulas que solo tiene la cláusula vacía es insatisfacible.

Sea S un conjunto de cláusulas y l un literal de S. Definimos el conjunto

$$S^l = \{C - \{l^c\} : l \notin C,\ C \in S\}\,.$$

Ejemplo 3.6 Consideremos el siguiente conjunto de cláusulas

$$S = \{r \vee \neg q, r \vee p, \neg r \vee p \vee \neg q, \neg q \vee \neg p\}\,.$$

Para el literal r el conjunto S^r es

$$S^r = \{C - \{\neg r\} : r \notin C\} = \{p \vee \neg q, \neg q \vee \neg p\}\,.$$

El conjunto S^r es satisfacible por la valuación $v\,(p) = 1$ y $v\,(q) = 0$. Por lo tanto, el conjunto S es también satisfacible (no necesariamente por la misma valuación).

El siguiente resultado prueba que la cláusula vacía $\{\ \}$ es insatisfacible.

Lema 3.7 *Sea S un conjunto de cláusulas. Entonces S es satisfacible si y sólo si S^l o S^{l^c} es satisfacible.*

Demostración. $\Rightarrow$) Supongamos que S es satisfacible. Entonces existe una valuación v tal que $v(S) = 1$. Supongamos que $v(l^c) = 0$, es decir, $v(l) = 1$. Probemos que $v\left(S^l\right) = 1$. Consideremos una cláusula arbitraria $C \in S^l$. Por lo definición de las cláusulas de S^l, $C \cup \{l^c\} \in S$ o $C \in S$. Entonces $v(C \cup \{l^c\}) = 1$ o $v(C) = 1$. Como $v(l^c) = 0$, entonces debe ocurrir que $v(C) = 1$.

Si $v(l^c) = 1$, entonces se prueba de forma similar que $v\left(S^{l^c}\right) = 1$.

$\Leftarrow$) Supongamos que S^l es satisfacible por una valuación v. Debemos determinar una valuación w tal que $w(S) = 1$. Consideremos la valuación w definida por

$$w(k) = \begin{cases} v(k) & \text{si} \quad k \neq l, l^c \\ \\ 1 & \text{si} \quad k = l. \end{cases}$$

Como en toda cláusula $C \in S^l, l, l^c \notin C$, entonces $w\left(S^l\right) = v\left(S^l\right) = 1$. Comprobemos que $w(S) = 1$. Sea $C \in S$. Si $l \in C$, entonces $w(C) = 1$. Si $l \notin C$, entonces $C - \{l^c\} \in S^l$, y en consecuencia $v(C - \{l^c\}) = 1$. Luego existe un literal $k \neq l^c$ tal que $k \in C$ y $v(k) = 1$. Por la definición de w tenemos que $v(k) = w(k)$ y por lo tanto $w(C) = 1$. Esto implica que $w(S) = 1$. ■

Corolario 3.8 *Sea S un conjunto de cláusulas. Entonces S es insatisfacible si y sólo si S^l y S^{l^c} son insatisfacibles.*

Corolario 3.9 *Sea S un conjunto de cláusulas. Supongamos que la cláusula unitaria $C = \{l\} \in S$. Entonces S es satisfacible si y sólo si S^l es satisfacible.*

Observación. Supongamos tener el conjunto de cláusulas unitarias $S = \{\{l\}, \{l^c\}\}$. De acuerdo al Corolario anterior, S es satisfacible si y solo si

$$S^l = \{C - \{l^c\} : l \notin C\} = \{\{l^c\} - \{l^c\}\} = \{\{\ \}\}$$

es satisfacible. Pero, como S es insatisfacible entonces deducimos que

$$S^l = \{\{\ \}\}$$

es insatisfacible. Por lo tanto, el conjunto de cláusulas que *tiene solo la cláusula vacía es insatisfacible.*

Lema 3.10 *Sea S un conjunto de cláusulas. Sea C una cláusula de S tal que $l, l^c \in C$. Entonces S es satisfacible si y sólo si $S - \{C\}$ es satisfacible.*

Demostración. Es sencilla y se deja al lector. ■

El conjunto de cláusulas $S = \{l \vee l^c\}$ siempre es satisfacible, pues para cualquier valuación v, tenemos que $v(l) = 1$ o $v(l^c) = 1$. Por el Lema anterior S es satisfacible si y solo si $S - \{l \vee l^c\} = \emptyset$ es satisfacible. En consecuencia podemos afirmar que el conjunto vacío $S = \emptyset$ de cláusulas es *siempre satisfacible.*

Podemos resumir lo expuesto anteriormente diciendo que el conjunto de cláusulas que solo tiene la cláusula vacía, es decir $S = \{\{\ \}\}$ es insatisfacible, y que el conjunto de cláusulas vacío $S = \emptyset$ es siempre satisfacible.

Sabemos que toda fórmula insatisfacible, como por ejemplo $A \wedge \neg A$, donde A es una fórmula, es insatisfacible. Recordemos que hemos convenido en denotar a cualquier fórmula insatisfacible con el símbolo $\perp$. De ahora en más vamos también a simbolizar por $\perp$ a la cláusula vacía $\{\ \}$.

3.2. Resolvente

Las fórmulas que están en la forma clausular tienen algunas propiedades especiales. Por ejemplo, en ciertos casos una fórmula puede ser transformada en otra eliminando algunas cláusulas de tal forma que la satisfacibilidad no cambia. Veamos esto con un ejemplo.

Supongamos que tenemos una fórmula en la forma clausular

$$A = \{p \vee \neg q, q \vee r\} .$$

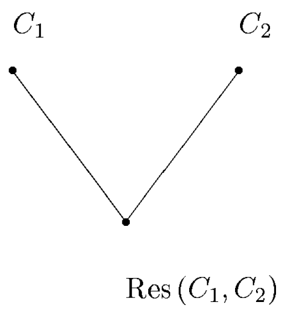
Sea v una valuación. Si $v(q) = 1$, entonces para que $v(p \vee \neg q) = 1$ debe ser $v(p) = 1$, es decir la satisfacibilidad de la cláusula $p \vee \neg q$ depende de la satisfacibilidad de la variable p. Si $v(q) = 0$, entonces la satisfacibilidad de la cláusula $q \vee r$ depende de la satisfacibilidad de la variable r. Por lo tanto, la fórmula $A = \{p \vee \neg q, q \vee r\}$ es satisfacible si y sólo si las cláusulas $p \vee \neg q$ y $q \vee r$ son satisfacibles si y sólo si p es satisfacible o r es satisfacible. Además, si la cláusula $p \vee r$ es falsa, entonces la fórmula A es falsa. En otras palábras, la satisfacibilidad de A depende de la satisfacibilidad de la clausula $p \vee r$. En cierta manera, las cláusulas $p \vee \neg q$ y $q \vee r$ se pueden reducir a la cláusula $p \vee r$. Vamos a precisar esta idea en la siguiente definición.

Recordemos que si C es una cláusula y l un literal, escribimos $l \in C$ para indicar que l aparece en C. El literal l^c indica el *literal complementario* de l.

Definición 3.11 Sean C_1 y C_2 dos cláusulas. Supongamos que $l \in C_1$ y que $l^c \in C_2$. La ***resolvente*** de C_1 y C_2 es la cláusula

$$\text{Res}(C_1, C_2) = (C_1 - \{l\}) \cup \{C_2 - \{l^c\}\} .$$

Es común asociar a la regla de resolución un árbol como lo muestra la siguiente figura .



Si determinamos la resolvente de las cláusulas unitarias $\{p, \neg p\}$ obtenemos la *cláusula vacía.* Es decir,

$$\mathrm{Res}\,(p, \neg p) = \{\ \} = \bot.$$

Ejemplo 3.12 Determinar la resolvente de $C_1 = p \vee \neg q$ y $C_2 = q \vee r$.

$$\mathrm{Res}\,(p \vee \neg q, q \vee r) = \{p\} \cup \{r\} = \{p, r\}\,.$$

o también podemos escribir

$$\mathrm{Res}\,(p \vee \neg q, q \vee r) = p \vee r.$$

La determinación de la resolvente de un par de cláusulas C_1 y C_2 se conoce como *regla de resolución.* Esta regla de resolución nos permite definir una noción de deducción. Es decir una noción que nos dice exactamente como podemos determinar mecánicamente una cláusula a partir de un conjunto finito de fórmulas aplicando la regla de resolución.

Definición 3.13 Sea S un conjunto de cláusulas y C una cláusula. Una *deducción por resolución* de C a partir de S es una sucesión finita de cláusulas

$$C_1, C_2, ..., C_n$$

tal que

1. La última cláusula C_n es C,

2. Para todo $1 \leq i \leq n$, se cumple que

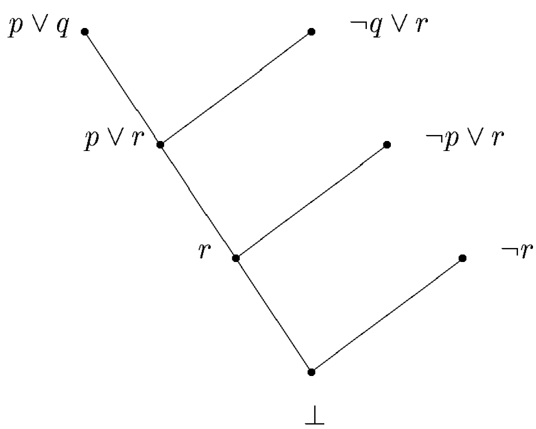
a) $C_i \in S$, o

b) C_i se obtiene por la regla de resolución de dos anteriores. Es decir, existen C_j, C_k, $j, k < n$, tal que
$$C_i = \mathrm{Res}\,(C_j, C_k)\,.$$

Escribiremos
$$S \vdash_R C,$$
cuando exista una deducción por resolución de una cláusula C a partir de un conjunto de cláusulas S . Notemos que el símbolo $\vdash_R$ define una relación entre conjunto de cláusulas y cláusulas. Como caso particular de deducción por resoluición tenemos la deducción de la cláusula vacía $\perp$.

Definición 3.14 Sea S un conjunto de cláusulas. Una ***refutación*** de S es una deducción por resolución de la cláusula $\perp$ a partir S. En símbolos,
$$S \vdash_R \perp.$$

Ejemplo 3.15 El siguiente árbol es una refutación del conjunto de cláusulas
$$S = \{p \vee q, \neg q \vee r, \neg p \vee r, \neg r\}\,.$$



En una deducción por resolución una cláusula puede ser utilizada más de una vez.

Como podemos observar en la definición de la deducción por resolución solo aparece un conjunto *finito de fórmulas.* Es decir, la noción de deducción es una noción esencialmente finitaria. Esto lo podemos precisar en la siguiente versión del teorema de compacidad.

Teorema 3.16 (de Compacidad) *Sea S un conjunto de cláusulas y C una cláusula. Entonces $S \vdash_R C$ si y sólo si existe un subconjunto finito S_0 de S tal que $S_0 \vdash_R C$.*

Demostración. $\Rightarrow$) Supongamos que $S \vdash_R C$. Entonces por la definición de $\vdash_R$ existe un conjunto finito de cláusulas $C_0, C_1, ..., C_n = C$ tal que para cada $1 \leq i \leq n$, $C_i \in S$ o existen dos cláusulas C_j, C_k con $j, k < n$ tal que $\mathrm{Re}\, s\,(C_k, C_j) = C_i$. Luego, el conjunto $S_0 = \{C_i \in S$ tal que intervienen en la sucesión $C_0, C_1, ..., C_n\}$ es tal que $S_0 \vdash_R C$.

La dirección $\Leftarrow$ es inmediata. ■

En la siguiente proposición damos una serie de propiedades de la relación $\vdash_R$ cuya prueba queda a cargo del lector.

Proposición 3.17 *Sean S, S' conjuntos de cláusulas y C, D cláusulas. Entonces*

1. *Si $C \in S$, entonces $S \vdash_R C$.*
2. *Si $S \vdash_R C$, entonces $S \cup \{D\} \vdash_R C$ para cualquier cláusula D. En particular, $S \cup S' \vdash_R C$ para cualquier conjunto de cláusulas S'.*
3. *Si $S \vdash_R C$ y $\{C\} \vdash_R D$, entonces$S \vdash_R D$.*

Definición 3.18 Sea S un conjunto de cláusulas. Definimos el conjunto $\mathcal{R}(S)$ como

$$\mathcal{R}(S) = S \cup \{C : \text{existen } C_1, C_2 \in S \text{ tal que } C = \mathrm{Res}\,(C_1, C_2)\}\,.$$

Definimos inductivamente el conjunto $\mathcal{R}^n(S)$ como*:*

$$\begin{aligned} \mathcal{R}^0(S) &= \mathcal{R}(S) \\ \mathcal{R}^{n+1}(S) &= \mathcal{R}(\mathcal{R}^n(S))\,. \end{aligned}$$

Finalmente, definimos

$$\mathcal{R}^*(S) = \bigcup_{n \geq 0} \mathcal{R}^n(S)\,.$$

Entonces es inmediato comprobar que:

$$C \in \mathcal{R}^*(S) \Leftrightarrow C \in \mathcal{R}^n(S)\,, \text{ para algún } n \geq 0.$$

Con la notación introducida en la definición anterior podemos escribir que

$$S \vdash_R C \Leftrightarrow C \in \mathcal{R}^*(S)\,.$$

Más adelante veremos el teorema de completitud para el cálculo por resolución. Este teorema afirma que una fórmula A es una tautología si y solo si una forma clausular de $\neg A$ tiene una refutación. En símbolos,

$$\vDash A \Leftrightarrow \{\neg A\} \text{ es insatisfacible } \Leftrightarrow cl\,(\neg A) \vdash_R \bot.$$

El teorema de completitud se extiende a conjunto finitos de fórmulas del siguiente modo.
Sea $\Gamma = \{A_1, ..., A_n\}$ un conjunto finito de fórmulas y A una fórmula. Entonces

$$\{A_1, ..., A_n\} \vDash A \Leftrightarrow \{A_1, ..., A_n, \neg A\} \text{ es insatisfacible } \Leftrightarrow cl\,\{A_1, ..., A_n, \neg A\} \vdash_R \bot,$$

donde $cl\,\{A_1, ..., A_n, \neg A\}$ significa que cada una de las fórmulas del conjunto $\{A_1, ..., A_n, \neg A\}$ están en la forma cláusular.

Ejemplo 3.19 Determinar si $\{p \to q, \neg p \to r, q \vee r \to s\} \vDash s$. Para poder aplicar resolución primero debemos pasar todas las fórmulas a la forma cláusular.

$p \to q \equiv \neg p \vee q$
$\neg p \to r \equiv p \vee r$
$q \vee r \to s \equiv \neg (q \vee r) \vee s \equiv (\neg q \wedge \neg r) \vee s \equiv (\neg q \vee s) \wedge (\neg r \vee s)\,.$
Verifiquemos ahora si

$$\left\{\underbrace{\neg p \vee q}_{1}, \underbrace{p \vee r}_{2}, \underbrace{\neg q \vee s}_{3}, \underbrace{\neg r \vee s}_{4}, \underbrace{\neg s}_{5}\right\} \vdash_R \perp.$$

6. $\mathrm{Res}\,(1,2) = q \vee r$
7. $\mathrm{Res}\,(6,3) = r \vee s$
8. $\mathrm{Res}\,(7,4) = s$
9. $\mathrm{Res}\,(8,5) = \perp$

3.3. Algorítmo de Davis-Putnam

Determinar si una cláusula se deduce por resolución de un conjunto finito de cláusulas o si existe una refutación de un conjunto finito de cláusulas puede resultar una tarea dificil, especialmente si el conjunto tiene una gran cantidad de cláusulas. Existen diversos mecanismos que permiten sistematizar el proceso de resolución. Ahora veremos un procedimiento o algorítmo debido a Davis y Putnam que permite obtener una deducción por resolución.

Sea S un conjunto finito de cláusulas y sea $\{p_1, p_2, ..., p_n\}$ el conjunto de todas las variables proposicionales que aparecen en las cláusulas de S. El algorítmo de Davis-Putnam consiste en los siguientes pasos:

- Eliminar del conjunto S todas las cláusulas en donde aparezca simultaneamente un literal l y su literal complementario l^c.
- Elegir una variable p_i de alguna de las cláusulas y elegir el subconjunto T_{p_i} de cláusulas de S en donde aparezca la variable p_i.
- Determinar el conjunto R_{p_i} de todas las cláusulas que son resolventes de cláusulas de la forma $D \cup \{p_i\}$ y $D' \cup \{\neg p_i\}$.
- Formar el conjunto de cláusulas $S_{p_i} = (S - T_{p_i}) \cup R_{p_i}$.
- Elegir otra variable p_{i+1} de S_{p_i} y continuar el procedimiento.

Como la cantidad de variables que ocurren en un conjunto finito de cláusulas S es finito, supongamos que tal número sea n, entonces en $n+1$ pasos se eliminan todas las variables.

Ahora vamos a dar una descripción precisa del algorítmo.
ALGORÍTMO DE DP :Sea S un conjunto finito de cláusulas y sea $\{p_1, p_2, ..., p_n\}$ el conjunto de todas las variables proposicionales que ocurren en las cláusulas de S.

1. Sea $S_1 = S$.
2. Sea $i = 1$
3. Loop hasta $i = n + 1$
4. Sea $S_i' = S_i - \{C\}$ donde en C aparecen un literal l y su complementario l^c.
5. $T_i = \{C \in S_i' : p_i \in C \text{ o } \neg p_i \in C\}$.
6. Sea $R_i = \{D : \text{ existen } C_1 \cup \{p_i\}, C_2 \cup \{\neg p_i\} \in T_i \text{ y } D = \text{Res}\,(C_1 \cup \{p_i\}, C_2 \cup \{\neg p_i\})\}$.
7. Sea $S_{i+1} = (S_i' - T_i) \cup R_i$.
8. Sea $i = i + 1$
9. Fin Loop
10. Sea S_{n+1}

El conjunto S_{n+1} del algorítmo de DP se llama conjunto final o conjunto solución, y puede ser solo alguno de los dos siguientes tipos:

$$S_{n+1} = \{\bot\},$$

o

$$S_{n+1} = \emptyset.$$

En el primer caso tenemos una refutación del conjunto S, es decir, $S \vdash_R \bot$. En el segundo caso $S \nvdash_R \bot$ pues ya sabemos que el conjunto vacío de cláusulas es siempre satisfacible.

Ejemplo 3.20 Aplicar el algorítmo de Davis-Putnam al siguiente conjunto de cláusulas.

$$S = \left\{ \begin{array}{l} p \vee \neg r, q \vee \neg r, q \vee \neg s, \neg p \vee r, \neg q \vee \neg t, \\ \neg q \vee r \vee t, p \vee s \vee \neg t, \neg p \vee q \vee r, q \vee r \vee s \vee t \end{array} \right\}$$

1. $S_1' = S$. Elegimos una variable, por ejemplo p. Determinamos el conjunto T_1 y numeramos sus cláusulas para facilitar los cálculos del conjunto R_1.

$$T_1 = \left\{ \underbrace{p \vee \neg r}_{1}, \underbrace{\neg p \vee r}_{2}, \underbrace{p \vee s \vee \neg t}_{3}, \underbrace{\neg p \vee q \vee r}_{4} \right\}.$$

$(1,2) \quad \neg r \vee r \equiv \top$

$(1,4) \quad \neg r \vee q \vee r \equiv \top$

$(2,3) \quad r \vee s \vee \neg t$

$(3,4) \quad s \vee \neg t \vee q \vee r$

$R_1 = \{r \vee s \vee \neg t, s \vee \neg t \vee q \vee r\}$

2. $S_2 = (S_1' - T_1) \cup R_1 = \left\{ \begin{array}{l} q \vee \neg s, q \vee \neg r, \neg q \vee \neg t, \neg q \vee r \vee t, \\ q \vee r \vee s \vee t, r \vee s \vee \neg t, s \vee \neg t \vee q \vee r \end{array} \right\}$

   $S_2' = S_2$. Elegimos la variable q.

   $$T_2 = \left\{ \underbrace{q \vee \neg s}_{1}, \underbrace{q \vee \neg r}_{2}, \underbrace{\neg q \vee \neg t}_{3}, \underbrace{\neg q \vee r \vee t}_{4}, \underbrace{q \vee r \vee s \vee t}_{5}, \underbrace{s \vee \neg t \vee q \vee r}_{6} \right\}$$

   $(1,3) \quad \neg s \vee \neg t$

   $(1,4) \quad \neg s \vee r \vee t$

   $(2,3) \quad \neg r \vee \neg t$

   $(2,4) \quad \neg r \vee r \vee t \equiv \top$

   $(3,5) \quad \neg t \vee r \vee s \vee t \equiv \top$

   $(3,6) \quad \neg t \vee s \vee r$

   $(4,5) \quad r \vee t \vee s$

   $(4,6) \quad r \vee t \vee \neg t \equiv \top$

   $R_2 = \{\neg s \vee \neg t, \neg s \vee r \vee t, \neg r \vee \neg t, \neg t \vee s \vee r, r \vee t \vee s\}$

3. $S_3 = (S_2' - T_2) \cup R_2 = \{r \vee s \vee \neg t, \neg s \vee \neg t, \neg s \vee r \vee t, \neg r \vee \neg t, \neg t \vee s \vee r, r \vee t \vee s\}$

   Elegimos la variable r.

   $$T_3 = \left\{ \underbrace{r \vee s \vee \neg t}_{1}, \underbrace{\neg s \vee r \vee t}_{2}, \underbrace{\neg r \vee \neg t}_{3}, \underbrace{\neg t \vee s \vee r}_{4}, \underbrace{r \vee t \vee s}_{5} \right\}$$

   $(1,3) \quad s \vee \neg t$

   $(2,3) \quad \neg s \vee t \vee \neg t \equiv \top$

   $(3,4) \quad \neg t \vee s$

   $(3,5) \quad \neg t \vee t \vee s \equiv \top$

   $R_3 = \{s \vee \neg t, \neg t \vee s\}$

4. $S_4 = (S_3' - T_3) \cup R_3 = \{\neg s \vee \neg t, s \vee \neg t, \neg t \vee s\}$

   $S_4 = S_4'$. Elegimos la variable s

   $$T_4 = S_4 = \left\{ \underbrace{\neg s \vee \neg t}_{1}, \underbrace{s \vee \neg t}_{2}, \underbrace{\neg t \vee s}_{3} \right\}$$

   $(1,2) \quad \neg t$

   $(1,3) \quad \neg t$

   $R_4 = \{\neg t\}$

5. $S_5 = (S_4 - T_4) \cup \{R_4\} = \{\neg t\}$

   Elegimos la variable t

   $T_5 = S_5 = \{\neg t\}$ y $R_5 = \emptyset$

6. $S_6 = (S_5 - T_5) \cup R_5 = \emptyset$.

   Como no hemos obtenido la cláusula $\perp$, entonces podemos asegurar que $S \nvdash_R \perp$.

3.4. Corrección y Completitud

En el apartado anterior hemos estudiado como deducir cláusulas a partir de un conjunto de cláusulas utilizando resolución. Ahora nos dedicaremos a estudiar la relación entre la noción de refutación de una cláusula y la noción de tautología. Veremos que A es una tautología, entonces existe una refutación de la cláusula asociada a $\neg A$. Recíprocamente, si existe una refutación para la cláusula asociada a $\neg A$, entonces A es una tautología. Si denotamos con $cl\,(A)$ la cláusula asociada a una fórmula A, entonces nuestro próximo objetivo es probar que:

$$\models A \quad \Rightarrow \quad cl\,(\neg A) \vdash_R \perp \tag{3.2}$$

y

$$cl\,(\neg A) \vdash_R \perp \quad \Rightarrow \quad \models A \tag{3.3}$$

El resto de esta sección vamos a dedicarnos a probar que las aserciones (3.2) y (3.3) son válidas.

En general, es posible probar resultados más generales. Dado un conjunto de fórmulas $\Sigma \cup \{A\}$ probaremos que:

1. Corrección:

   - Si $cl\,(\Gamma) \vdash_R cl\,(A)$, entonces $\Gamma \models A$.

2. Completitud:

   - Si $\Gamma \models A$, entonces $cl\,(\Gamma \cup \neg A) \vdash_R \perp$.

Para poder probar estos teoremas debemos primero dar unos resultados preliminares.

Teorema 3.21 *Sean C_1 y C_2 dos cláusulas. Entonces C_1 y C_2 son satisfacibles si y sólo si $C = \mathrm{Res}\,(C_1, C_2)$ es satisfacible.*

Demostración $\Rightarrow$) Supongamos que C_1 y C_2 son satisfacibles. Entonces existe una valuación v tal que $v\,(C_1) = 1$ y $v\,(C_2) = 1$. Sea l un literal tal que $l \in C_1$ y $l^c \in C_2$. Como $\{l, l^c\}$ es un par complemenetario, entonces $v\,(l) = 1$ o $v\,(l^c) = 1$.

Supongamos que $v\,(l) = 1$. Entonces $v\,(l^c) = 0$. Como $v\,(C_2) = 1$, entonces debe existir otro literal $l' \in C_2$ distinto de l^c tal que $v\,(l') = 1$. Por contrucción de la resolvente $C = \mathrm{Res}\,(C_1, C_2)$ debe ocurrir que $l' \in C$. Luego $v\,(C) = 1$.

El caso $v\,(l^c) = 1$ se analiza similarmente.

$\Leftarrow$) Supongamos que $C = \mathrm{Res}\,(C_1, C_2)$ es satisfacible. Es decir, existe una valuación v tal que $v\,(C) = 1$. Como $l, l^c \notin C$, entonces existe un literal $l' \in C$ distinto de l y de l^c tal que $v\,(l') = 1$. Este literal debe pertenecer a C_1 o a C_2. Supongamos que $l' \in C_1$. Entonces $v\,(C_1) = 1$. La valuación v no necesariamente está definida en l y l^c (pues $l, l^c \notin C$). Extendemos v a una valuación w del siguiente modo:

$$w(k) = \begin{cases} 1 & \text{si} \quad k = l^c \\ \\ v(k) & \text{si} \quad k \neq l^c \end{cases}.$$

Entonces, $w(C_2) = 1$ y $w(C_1) = v(C_1) = 1$. Por lo tanto, C_1 y C_2 son satisfacibles por la valuación w.

El caso $l' \in C_2$ se analiza en forma similar. ■

Teorema 3.22 (de Corrección) *Sea S un conjunto de cláusulas y C una cláusula. Si*

$$S \vdash_R C, \text{ entonces } S \vDash C.$$

Demostración Supongamos que $S \vdash_R C$. La prueba es por inducción sobre la longitud de la deducción por resolución de C a partir de S.
Si $n = 1$, entonces $C_1 = C$, y en consecuencia $C \in S$. Por lo tanto, $S \vDash C$.
Supongamos como hipótesis inductiva, que el teorema es verdadero para toda deducción de longitud $k < n$.
Sea

$$C_1, C_2, ..., C_{n-1}, C_n = C$$

una deducción de longitud n. Tenemos los siguientes casos.

1. Si $C_n = C \in S$, entonces $S \vDash C$.

2. Si $C_n = C \notin S$, entonces existen cláusulas C_i, C_j, con $i, j < n$, tales que $C = \text{Res}(C_i, C_j)$. Las secuencias

   $$C_1, C_2, ..., C_i, \qquad y \qquad C_1, C_2, ..., C_j,$$

   son deducciones de C_i y C_j a partir de S, es decir, $S \vdash_R C_i$ y $S \vdash_R C_j$. Como $i, j < n$, entonces por Hipótesis inductiva,

   $$S \vDash C_i \text{ y } S \vDash C_j.$$

   Consideremos una valuación v tal que $v(S) = 1$. Entonces, $v(C_i) = 1$ y $v(C_j) = 1$. Por el Lema anterior, $v(C) = 1$, pues $C = \text{Res}(C_i, C_j)$. Por lo tanto, $S \vDash C$. ■

Ejemplo 3.23 Consideremos el siguiente conjunto de cláusulas $S = \{\neg p \vee q, \neg q \vee r\}$. Como $\text{Res}(\neg p \vee q, \neg q \vee r) = \neg p \vee r$, entonces $S \vdash_R C$. Según el Teorema de Corrección, toda valuación que satisfaga al conjunto S también satisface a C. Construimos en una tabla todas las posibles valuaciones sobre el conjunto S.

p	q	r	$\neg p \vee q$	$\neg q \vee r$		$\neg p \vee r$
0	0	0	1	1		1
0	0	1	1	1		1
0	1	0	1			1
0	1	1	1	1		1
1	0	0		1		
1	0	1		1		1
1	1	0	1			
1	1	1	1	1		1

Observemos que toda fila que asigna el 1 a todas las cláusulas de S también asigna 1 a la cláusula C.

Corolario 3.24 *Sea S es el conjunto de cláusulas. Si $S \vdash_R \bot$, entonces S es insatisfacible.*

Corolario 3.25 *Sea A una fórmula. Si $cl\,(\neg A) \vdash_R \bot$, entonces A es una tautología.*

El anterior Corolario nos dice que si A es una fórmula que está en la forma normal conjuntiva y $cl\,(\neg A) \vdash_R \bot$, entonces $\neg A$ es insatisfacible, o lo que es lo mismo decir, A es una tautología.

Ejemplo 3.26 Determinar por resolución si la fórmula A es una tautología.

$$A = ((p \to q) \wedge (q \to r)) \to (p \to r)$$

Primero pasamos a la fnc la fórmula $\neg A$.

$$\begin{aligned} \neg A &= \neg\left[((p \to q) \wedge (q \to r)) \to (p \to r)\right] \\ &\equiv \neg\left[((\neg p \vee q) \wedge (\neg q \vee r)) \to (\neg p \vee r)\right] \\ &\equiv (\neg p \vee q) \wedge (\neg q \vee r) \wedge p \wedge \neg r. \end{aligned}$$

La forma clausular de $\neg A$ es

$$cl\,(\neg A) = \left\{ \underbrace{\neg p \vee q}_{1}, \underbrace{\neg q \vee r}_{2}, \underbrace{p}_{3}, \underbrace{\neg r}_{4} \right\}.$$

Entonces obtenemos la siguiente deducción por resolución:

5. $\neg p \vee r$ de 1 y 2.

6. r de 5 y 3.

7. $\bot$ de 6 y 4.

Por lo tanto, como $cl\,(\neg A) \vdash_R \bot$, entonces A es una tautología.

Para probar el Teorema de Completitud necesitamos el siguiente Lema.

Lema 3.27 *Sea S un conjunto de cláusulas y l un literal. Sea*

$$S\,(l) = \{C \in \mathcal{R}^*\,(S) : l, l^c \notin C\} = \{S \vdash_R C : l, l^c \notin C\}.$$

Si S es insatisfacible, entonces $S\,(l)$ es insatisfacible.

Demostración. Recordemos que $\mathcal{R}^*\,(S) = \bigcup\limits_{n \geq 0} \mathcal{R}^n\,(S)$. Asumimos que S es insatisfacible y que $S\,(l)$ es satisfacible para algún literal l y por una valuación v. Como $l, l^c \notin S\,(l)$, entonces v no está definida en los literales l y l^c. Consideremos las extensiones v_1 y v_2 de v definidas por:

$$v_1\,(k) = \begin{cases} v\,(k) & \text{si} \quad k \neq l, l^c \\ 1 & \text{si} \quad k = l \end{cases}$$

y

$$v_2(k) = \begin{cases} v(k) & \text{si} \quad k \neq l, l^c \\ \\ 1 & \text{si} \quad k = l^c \end{cases}.$$

Como S es insatisfacible, entonces $v_1(S) = 0$ y $v_2(S) = 0$. Luego existen cláusulas $C_1, C_2 \in S$ tales que:

$$v_1(C_1) = 0 \text{ y } v_2(C_2) = 0.$$

Como $v_1(l) = 1$, entonces $l \notin C_1$. Si $l^c \notin C_1$, entonces $C_1 \in S(l)$, pero como $v_1(S(l)) = v(S(l)) = 1$, entonces $v(C_1) = 1$, lo que es un absurdo. Por lo tanto, $l^c \in C_1$. Razonando de igual forma, podemos afirmar que $l \in C_2$. Luego, $C = \text{Res}(C_1, C_2)$ y $l, l^c \notin C$. Por lo tanto, $C \in S(l)$, y como $S(l)$ es satisfacible por v, $v(C) = 1$. Como en la prueba del Lema 3.21, existe un literal k distinto de l y l^c tal que $v(k) = 1$ y $k \in C_1$ o $k \in C_2$. Si $k \in C_1$, entonces $v_1(C_1) = 1$, lo que es un absurdo. Si $k \in C_2$, entonces $v_2(C_2) = 1$ lo que también es un absurdo. Por lo tanto, $S(l)$ es insatisfacible. ■

Observación. Recordemos que en la formulación del algorítmo de DP, definimos los conjuntos S_{i+1} como $S_{i+1} = (S_i' - T_i) \cup R_i$. Utilizando la notación del Lema anterior, es fácil comprobar que

$$S_{i+1} = S_i'(l_i).$$

Esta notación será utilizada en el próximo resultado.

Teorema 3.28 *Sea S un conjunto de cláusulas.*

Si S es insatisfacible entonces $S \vdash_R \perp$.

Demostración. Sea S es insatisfacible. Por el Teorema de Compacidad, es suficiente suponer que S es finito.

Sea $\{p_1, ..., p_n\}$ el conjunto de todas las variables que ocurren en alguna cláusula de S. Como S es finito, este conjunto es finito.

Para probar que $S \vdash_R \perp$, aplicaremos el algorítmo de D-P y probaremos, por inducción sobre n, que para cada $n \geq 1$ el conjunto S_n es insatisfacible. En consecuencia tendremos que el conjunto final S_{n+1} es insatisfacible. Como S_{n+1} no tiene variables, la única posibilidad es que $S_{n+1} = \{\perp\}$. Con esto se deduce que $\perp \in \mathcal{R}^*(S)$, es decir, $S \vdash_R \perp$.

El caso $n = 0$, tenemos que $S_1 = S$, y por hipótesis S es insatisfacible.

Si suponemos que $S_i = S_{i-1}'(l_{i-1})$ es insatisfacible para todo $i \leq n$. Entonces por el Lema 3.27, $S_{n+1} = S_n(l_n)$ es insatisfacible. Por lo tanto, S_{n+1} es insatisfacible y con esto concluímos que $S \vdash_R \perp$.
■

Con el anterior teorema hemos justificado el procedimiento que permite averiguar cuando una fórmula es una tautología. Calculamos la forma clausular de la negación de la fórmula y determinamos por resolución si existe una refutación del conjunto de cláusulas.

El razonamiento anterior se extiende a conjuntos de fórmulas arbitrarios de la siguiente manera.

Sea Γ un conjunto de fórmulas. La *forma clausular* de Γ es el conjunto de cláusulas $cl(\Gamma) = \{cl(A) : A \in \Gamma\}$. Podemos denotar que un conjunto de fórmulas Γ es insatisfacible por $\Gamma \vDash \perp$. Recordemos que toda fórmula A es equivalente a su forma clausular $cl\,(A)$. Por lo tanto, A es insatisfacible si y sólo si $cl\,(A)$ es insatisfacible. De igual manera, un conjunto Γ es insatisfacible si y sólo si $cl\,(\Gamma)$ es insatisfacible.

Resumiendo, si $\Gamma \cup \{A\}$ es un conjunto de fórmulas, entonces valen las siguientes equivalencias:

$$\begin{aligned} \Gamma \models A &\Leftrightarrow \Gamma \cup \{\neg A\} \models \bot \\ &\Leftrightarrow cl\,(\Gamma \cup \{\neg A\}) \models \bot \\ &\Leftrightarrow cl\,(\Gamma \cup \{\neg A\}) \vdash_R \bot. \end{aligned}$$

Ejemplo 3.29 Determinar si la deducción $\{p \rightarrow (q \rightarrow r)\,, p \rightarrow q, p\} \models r$ es válida.

Por la observación anterior, tenemos las siguientes equivalencias:

$$\begin{aligned} \{p \rightarrow (q \rightarrow r)\,, p \rightarrow q, p\} \models r &\Leftrightarrow \{p \rightarrow (q \rightarrow r)\,, p \rightarrow q, p, \neg r\} \models \bot \\ &\Leftrightarrow cl\,(\{p \rightarrow (q \rightarrow r)\,, p \rightarrow q, p, \neg r\}) \models \bot \\ &\Leftrightarrow cl\,(\{p \rightarrow (q \rightarrow r)\,, p \rightarrow q, p, \neg r\}) \vdash_R \bot \\ &\Leftrightarrow \{\neg p \vee \neg q \vee r, \neg p \vee q, p, \neg r\} \vdash_R \bot. \end{aligned}$$

Queda como ejercicio para el lector comprobar la última refutación.

Capítulo 4

Cálculo de Predicados

4.1. Introducción

La lógica proposicional es deficiente para interpretar los razonamientos matemáticos. Existen muchas proposiciones que no pueden ser expresadas solamente con la herramientas que nos poporciona la Lógica Proposicional. Esto se debe a que cuando estudiamos la validez de un argumento no analizamos la estructura interna de las variables proposicionales. Por ejemplo, consideremos la siguiente deducción:

Todo número racional es un número real
7 es un número racional
entonces, 7 es un número real.

Este razonamiento es claramente válido, pero no es posible justificarlo con las técnicas vistas hasta ahora. En Lógica Proposicional analizamos solamente la estructura de las fórmulas o sentencias en términos de las sentencias que las componen.. En la anterior deducción necesitamos un análisis más fino de los componentes que intervienen.

4.2. Lenguajes de Primer Orden

En algunas ramas de la matemática se trabaja con estructuras que constan de un conjunto de elementos, operaciones definidas en el conjunto y/o relaciones definidas en el conjunto. Por ejemplo, un grupo es un conjunto dotado de una operación binaria, denotada por $\circ$, junto con un conjunto de axiomas. Los anillos son conjuntos dotados de dos operaciones binarias, denotadas por $+$ y $\circ$, que cumplen determinados axiomas. Un grafo es un conjunto junto con una relación binaria (no una función) verificando determinadas condiciones. En todas estas estructuras matemáticas las proposiciones o sentencias no pueden ser expresadas únicamente con el lenguaje proposicional. Los lenguajes de primer orden corrigen estas falencias. En los lenguajes de primer orden, además de los símbolos del lenguaje proposicional, también tenemos símbolos para denotar expresiones como *para todo x* y *existe un x*, símbolos para denotar relaciones, símbolos para denotar funciones, símbolos para denotar elementos distinguidos del conjunto o constantes y algunos símbolos auxiliares. Por ejemplo, en matemáticas estamos acostumbrados a trabajar con expresiones como

Todos los números naturales son reales
Existen algunos número enteros divisibles por 2
Existen números naturales menores que 10

Todas estas expresiones pueden ser formalizadas en un lenguaje más amplio que el lenguaje proposicional. Para poder escribir simbolicamente expresiones tales como *existen.....* o *para todo....*, necesitamos los símbolos especiales $\exists$ (cuantificador existencial) y $\forall$ (cuantificador universal). Por ejemplo la expresión

Todos los números naturales son reales

se podría escribir

$$\forall x: \text{ si } x \text{ es natural,entonces } x \text{ es real.}$$

De igual manera, estamos acostumbrados a tratar con relaciones entre objetos. Por ejemplo, la propiedad

si x, y son enteros positivos tal que $x \leq y$, entonces existe un entero positivo z tal que $y = x + z$,

puede ser escrita en términos lógicos como

$$\forall x \forall y \left(x \in \mathbb{Z}^+ \wedge y \in \mathbb{Z}^+ \wedge x \leq y \rightarrow \exists z \in \mathbb{Z}^+ \left(y = x + z \right) \right).$$

En este ejemplo necesitamos utilizar el símbolo de relación *menor o igual* $\leq$ y un símbolo de función binario $+$.

Como en el caso proposicional, vamos a definir lo que es un lenguaje de primer orden especificando sus símbolos y que sucesiones finitas de estos símbolos serán las fórmulas.

Definición 4.1 Un lenjuage de primer orden $\mathcal{L}$ consta de los siguientes conjuntos

1. *Símbolos Lógicos*

   - Un conjunto numerable de *variables,* denotado por Var. Las variables las simbolizaremos por $x_1, x_2, ..$ o x, y, z, etc.
   - *Conectivos proposicionales* $\vee, \wedge, \rightarrow$ y $\neg$.
   - *Cuantificadores* $\forall$ *y* $\exists$.
   - *Símbolos auxiliares (* y *).*

2. Un conjunto de símbolos $\mathcal{C}$ llamado *símbolos de constantes.* Usualmente los símbolos de constantes serán denotados por a, b, c, etc. Puede ser $\mathcal{C} = \emptyset$.

3. Un conjunto de símbolos $\mathcal{F}$ llamado *símbolos de funciones n-arias.* Los símbolos de función serán denotados por f, g, h, etc. Puede ser que $\mathcal{F} = \emptyset$.Cada símbolo de función tiene asociado un número natural $n \geq 1$ llamado la aridad de símbolo. Si la aridad de un símbolo f es n diremos que f es un n-ario.

4. Un conjunto de símbolos $\mathcal{R}$ llamado *símbolos de predicados n-arios o relaciones n-arias*. Los predicados serán denotados por P, R, etc. Puede que $\mathcal{R} = \emptyset$. Cada símbolo de relación tiene asociado un número natural $n \geq 1$ llamado la aridad de símbolo. Si la aridad de un símbolo R es n diremos que R es un n-ario.

Los símbolos lógicos y los símbolos auxiliares son comunes a cualquier lenguaje de primer orden. Los símbolos correspondientes a los apartados 2., 3. y 4. son los símbolos *propios* del lenguaje. Es decir, los lenguaje de primer orden quedan caracterizados por sus símbolos propios. Debido a esto, cuando hagamos referencia a un lenguaje de primer orden $\mathcal{L}$ escribiremos $\mathcal{L} = \langle \mathcal{R}, \mathcal{F}, \mathcal{C} \rangle$.

Es claro que existen muchos lenguajes de primer orden. Cada uno depende de los símbolos que se incluyan.

Veamos algunos ejemplos de lenguajes de primer orden.

Ejemplo 4.2 Consideremos el lenguaje $\mathcal{L} = \langle \mathcal{R}, \mathcal{F}, \mathcal{C} \rangle$, donde

$$\begin{aligned} \mathcal{R} &= \{R, P\}\text{ , donde } R \text{ es un símbolo de relación binario y } P \text{ unario.} \\ \mathcal{F} &= \emptyset \\ \mathcal{C} &= \{a\} \end{aligned}$$

Todavía no hemos definido lo que son fórmulas en un lenguaje de primer orden, pero podemos utilizar nuestros conocimientos de lógica elemental para dar un par de ejemplos de enunciados en este lenguaje:

$$\forall x \exists y \left(R\left(x, y\right) \rightarrow P\left(a\right) \wedge P\left(x\right)\right),$$

$$\forall x \left(P\left(x\right) \rightarrow R\left(x, y\right)\right) \vee \neg R\left(x, a\right).$$

Ejemplo 4.3 Sea $\mathcal{L}$ un lenguaje de primer orden que tiene un símbolo de predicado binario R, un símbolo de función binaria g y un símbolo de constante c. Como ejemplos de enunciados en este lenguaje tenemos a

$$\exists y \left(R\left(x, g\left(x, y\right)\right) \rightarrow R\left(x, c\right)\right)$$

$$\forall x \forall y \exists z R\left(g\left(x, y\right), z\right).$$

Muchas estructuras matemáticas conocidas tienen asociados un lenguaje de primer orden.

Ejemplo 4.4 El lenguaje formal de la teoría de números , simbolizado por $\mathcal{L}_{\mathbb{N}}$, puede ser definido como sigue:

1. Símbolos de constantes: 0 y 1,

2. Dos símbolos de función binarios: $+$ y $\times$

3. Dos símbolos de relación binarios: $<$ y $|$.

   Cada uno de estos símbolos intenta representar la idea usual que tienen en el uso matemático: 0 y 1 son constantes, $+$ y $\times$ son los símbolos para las operaciones de adicción o suma y producto, y los símbolos $<$ y $|$ son usados para denotar las relaciones menor o igual y divide, respectivamente.

Ejemplo 4.5 Otra lenguaje para la teoría de números, es el siguiente

1. Símbols de constante: 0

2. Un símbolo de función unario: S

3. Dos símbolos de funcion binarios: $+$ y E

   En este caso 0 representa al cero, S la función sucesor, es decir, $S(n) = n + 1$, y E la función $E(n, m) = n^m$.

De igual manera que hicimos en lógica proposicional, debemos definir el concepto de fórmula. En general, una fórmula en un lenguaje de primer orden $\mathcal{L}$ es una sucesión finita de símbolos del lenguaje. Obviamente, no toda sucesión será una fórmula. Para precisar exactamente qué sucesiones de símbolos serán considerados como fórmulas debemos dar reglas de formación para las mismas. Primero definiremos los términos del lenguaje, despues las fórmulas atómicas y por último las fórmulas del lenguaje $\mathcal{L}$.

En las siguientes definiciones suponemos que hemos fijado un lenguaje de primer orden $\mathcal{L}$.

Definición 4.6 El conjunto de todos los *términos* de $\mathcal{L}$, en símbolos $Ter(\mathcal{L})$, es el menor subconjunto de sucesiones finitas de elementos de $\mathcal{L}$ que verifican las siguientes condiciones:

1. $Var \cup \mathcal{C} \subseteq Ter(\mathcal{L})$, es decir, toda variable y toda constante es un término.

2. Si $t_1, t_2, ..., t_n$ son términos y f es un símbolo de función de aridad n, entonces $f(t_1, t_2, ..., t_n) \in Ter(\mathcal{L})$.

Ejemplo 4.7 Consideremos un lenguaje $\mathcal{L}$ con un símbolo relacional binario R, un símbolo de función unario f, y una constante c. Los siguientes son ejemplos de términos en el lenguaje $\mathcal{L}$:

$$c,\ f(x),\ f(f(x)),\ f(c),\ f(f(c)).$$

Definición 4.8 El conjunto de todas las *fórmulas atómicas* de $\mathcal{L}$, en símbolos $At(\mathcal{L})$, es el menor subconjunto de sucesiones finitas de elementos de $\mathcal{L}$ que verifican las siguientes condiciones:

1. $Ter(\mathcal{L}) \subseteq At(\mathcal{L})$, es decir, todo término es una fórmula atómica.

2. Si $t_1, t_2, ..., t_n$ son términos y R es un símbolo de relación de aridad n, entonces $R(t_1, t_2, ..., t_n) \in At(\mathcal{L})$.

Ejemplo 4.9 En el mismo lenguaje del ejemplo anterior, las siguientes expresiones son ejemplos de fórmulas atómicas:

$$c,\ f(x),\ f(f(x)),\ f(c),\ f(f(c))$$
$$R(x, y),\ R(f(x), y),\ R(c, f(f(c))).$$

Definición 4.10 El conjunto de todas las *fórmulas* de $\mathcal{L}$, en símbolos $Fm(\mathcal{L})$, es el menor subconjunto de sucesiones finitas de elementos de $\mathcal{L}$ que verifica las siguientes condiciones:

1. $At(\mathcal{L}) \subseteq Fm(\mathcal{L})$
2. Si $A, B \in Fm(\mathcal{L})$, entonces $A \vee B$, $A \wedge B$, $A \to B$, $\neg A \in Fm(\mathcal{L})$.
3. Si $A \in Fm(\mathcal{L})$, entonces $\forall x A$, $\exists x A \in Fm(\mathcal{L})$, siendo x cualquier variable.

Veamos ahora algunos ejemplos de lenguajes de primer orden y de fórmulas en dichos lenguajes.

Ejemplo 4.11 Consideremos el lenguaje $\mathcal{L} = \langle \mathcal{R}, \mathcal{F}, \mathcal{C} \rangle$ donde

$$\begin{array}{rcl} \mathcal{C} & = & \{a\} \\ \mathcal{R} & = & \{R\}\text{, donde } R \text{ es un símbolo de predicado binario, y} \\ \mathcal{F} & = & \emptyset \end{array}$$

En este caso el conjunto de todos los términos es $Ter(\mathcal{L}) = Var \cup \mathcal{C}$.

Como ejemplos de fórmulas atómicas tenemos

$$At(\mathcal{L}) = Var \cup \mathcal{C} \cup \{R(c,c), R(x,c), R(x,y), ...\}$$

Ejemplos de fórmulas son

$$\forall x \forall y (R(x,y) \to R(y,x))$$

$$\forall x (R(c,x) \to \neg R(x,c)).$$

Ejemplo 4.12 Consideremos ahora un lenguaje con dos constantes a, b, dos símbolos de predicados binarios R y P, un símbolo de función f unario y un símbolo de función g binario. Entonces

$$Ter(\mathcal{L}) = Var \cup \{a, b, f(a), f(b), g(a,b), g(f(a), b), ...\}$$

$$At(\mathcal{L}) = Ter(\mathcal{L}) \cup \{R(a,b), P(a,a), R(f(a), g(a,b)), ...\}$$

Ejemplos de fórmulas en este lenguaje serían

$$\forall x \forall y ((R(x,y) \wedge f(x)) \to P(g(x,y), a) \wedge \exists y (P(y, f(a,x)))).$$

$$\neg \exists x (P(x,a) \wedge \forall x (R(b,x))).$$

4.2.1. Subfórmulas

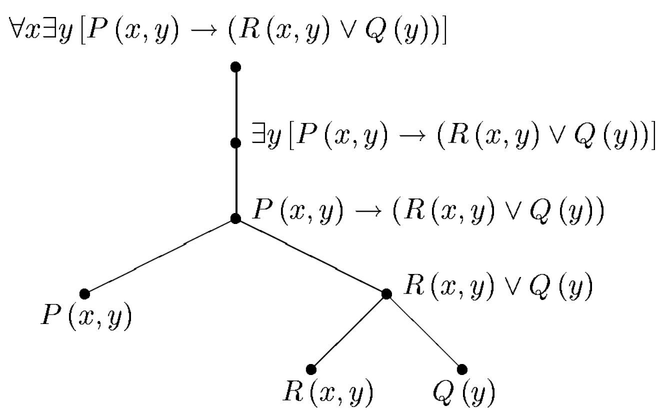
El concepto de subfórmula en un lenguaje de primer orden $\mathcal{L}$ se define de manera similar al concepto de subfórmula definido en el cálculo proposicional.

Definición 4.13 Sea $\mathcal{L}$ un lenguaje *y* $A \in Fm(\mathcal{L})$. El conjunto de las *subfórmulas* de A se define recursivamente por:

- $Sf(A) = \{A\}$, si A es una fórmula atómica,
- $Sf(\neg A) = Sf(A) \cup \{\neg A\}$,

- $Sf(A * B) = Sf(A) \cup Sf(B) \cup \{A * B\}$, donde $*$ es cualquiera de los conectivos binarios $\rightarrow$, $\vee$ o $\wedge$.
- $Sf(\forall x A) = Sf(A) \cup \{\forall x A\}$.
- $Sf(\exists x A) = Sf(A) \cup \{\exists x A\}$.

Para determinar el conjunto de subfórmulas de una fórmula A podemos determinar su árbol genealógico. Por ejemplo, el árbol genealógico de la fórmula $\forall x \exists y\, [P(x,y) \rightarrow (R(x,y) \vee Q(y))]$ es



Alcance de un cuantificador

El alcance o radio de un cuantificador $\forall x$ o $\exists x$ es la fórmula afectada por el cuantificador. Es decir, si $\forall x A$ (o $\exists x A$) es una fórmula, entonces A es el alcance del cuantificador $\forall$ (o $\exists$).

Por ejemplo, en la fórmula

$$(\forall x (A(x) \wedge P(x,y)) \Rightarrow \exists z R(z)) \vee R(a,x)$$

el alcance del cuantificador $\forall x$ es la fórmula $A(x) \wedge P(x,y)$, y el alcance del cuantificador $\exists z$ es $R(z)$.

Variables libres y ligadas

Una ocurrencia de una variable x en una fórmula A se dice *ligada* o *acotada* si está dentro del alcance de un cuantificador. En caso contrario, se dice *libre*. Una definición formal de este concepto es la siguiente.

Definición 4.14 Sea x una variable Diremos x *ocurre libre* en una fórmula A de $\mathcal{L}$ si:

1. Si A es atómica, entonces x ocurre libre en A si y sólo si x es una variable de A.

2. Si A is $(\neg B)$, entonces x ocurre libre en A si y sólo si x ocurre libre en B.
3. Si A is $(B \to C)$, entonces x ocurre libre en A si y sólo si x ocurre libre en B o en C.
4. Si A is $\forall y\, B$, entonces x ocurre libre en A si y sólo si x es diferente de y y x ocurre libre en B.
5. Si A is $\exists y\, B$, entonces x ocurre libre en A si y sólo si x es diferente de y y x ocurre libre en B.

Las partes 4. y 5. de la definición anterior dice que una ocurrencia de una variable x es acotada si está dentro del alcance de una cuantificador $\forall x$ o de un cuantificador $\exists x$.

Ejemplo 4.15 En la fórmula

$$\forall x A(x) \wedge B(z, y)$$

la variable x es acotada y las variables y y z son libres.

Definición 4.16 Una fórmula A es *cerrada* o es una *sentencia* cuando no tiene variables libres.

Por ejemplo, la fórmula

$$\forall x \exists y\, ((P(x, y) \wedge \neg R(x)) \to P(y))$$

es una sentencia. La fórmula

$$(R(a, b) \wedge f(a)) \to f(b)$$

donde a y b son constantes es también una sentencia. La fórmula

$$\exists x \exists y\, (A(x, y) \wedge P(a, z))$$

no es una sentencia pues la variable z es libre.

Las sentencias son las fórmulas en un lenguaje dado que despues de interpretarlas tiene sentido preguntarse si son verdaderas o falsas.

Observación. Una misma fórmula puede tener variables con apariciones libres y ligadas. Por ejemplo en la fórmula

$$(\forall x \exists y\, (R(x, y) \to P(y))) \vee \neg Q(x, f(x))$$

la variable x tiene una aparación ligada y dos libres.

Dada una fórmula A escribiremos $A(x_1, x_2, ..., x_n)$ para indicar que las variables libres de A, si existen, están en el conjunto $\{x_1, x_2, ..., x_n\}$. También simbolizaremos $Vl(A)$ al conjunto de las variables libres, si existen, de la fórmula A.

Definición 4.17 La *clausura universal* de una fórmula $A(x_1, ..., x_n)$ es la sentencia

$$\forall x_1 \forall x_2 ... \forall x_n A(x_1, x_2, ..., x_n).$$

La *clausura existencial* de $A(x_1, ..., x_n)$ es la sentencia

$$\exists x_1 \exists x_2 ... \exists x_n A(x_1, x_2, ..., x_n).$$

Sustituciones. Si A es una fórmula, x una variable de A y t un término, la *sustitución* de x por t en A, en símbolos $A(x/t)$, es la fórmula que se obtiene al reemplazar en A cada aparición libre de x por el término t.

Por ejemplo, si c es una constante, entonces

$$\forall x R(x,y)(y/c) \text{ es } \forall x R(x,c).$$

$$[\forall x P(x) \to Q(x)](x/c) \text{ es } (\forall x P(x)) \to Q(c).$$

En la última fórmula solo sustituimos en $Q(x)$, pues la otra aparición de x es ligada.

Si $A(x_1,...,x_n)$ es una fórmula $t_1,...,t_n$ son términos distintos, entonces la *sustitución simultánea* de las variables $x_1,...,x_n$ por lo términos $t_1,...,t_n$, en símbolos

$$A(x_1/t_1,...,x_n/t_n),$$

es la fórmula que se obtiene al reemplazar cada aparición libre de x_i por t_i. Podemos dar una definición más formal de la noción de sustitución por medio de la siguisnte definición.

Definición 4.18 Sean t y h términos y x una variable. La *sustitución de la variable x por el término h en t es el término* $t(x/h)$ (también denotado $t\binom{x}{h}$) definido recursivamente como sigue:

1. *Si $t = x$, entonces $t(x/h) = h$.*
2. *Si $t = y$ con y una variable distinta de x, entonces $t(x/h) = y$.*
3. *Si $t = c$, donde c es una constante, entonces $t(x/h) = c$.*
4. *Si $t = f(t_1,...,t_n)$, donde f es un símbolo de función n-ario y $t_{1,}...,t_n$ son términos, entonces $t(x/h) = f(t_1/h,...,t_n/h)$.*

Para el caso de fórmulas la definición de sustitución de una varieable x por un término h en una fórmula A es la siguiente.

Definición 4.19 Sea A *una fórmula,* h un término y x una variable. La *sustitución de la variable x por el término h en A es la fórmula* $A(x/h)$ (también denotado $A\binom{x}{h}$) definida recursivamente como sigue:

1. *Si $A = R(t_1,...,t_n)$, donde R e un símbolo de predicado n-ario, , entonces $A(x/h) = R(t_1/h,...,t_n/h)$.*
2. *Si $A = \neg B$, entonces $A(x/h) = (\neg B)(x/h) = \neg B(x/h)$.*
3. *Si $A = B * C$, donde $*$ es $\vee, \wedge$ o $\to$, entonces $A(x/h) = (B * C)(x/h) = B(x/h) * C(x/h)$.*
4. *Si $A = \exists x B$, entonces $A(x/h) = \exists x B$.*
5. *Si $A = \exists y B$, entonces $A(x/h) = \exists y B(x/h)$, siendo x una variable libre en B.*
6. *Si $A = \forall x B$, entonces $A(x/h) = \forall x B$.*
7. *Si $A = \forall y B$, entonces $A(x/h) = \forall y B(x/h)$, siendo x una variable libre en B.*

Notemos que en el punto 4. y 6. en la anterior definición la fórmula A queda invariante pues las sustituciones se hacen sobre variables libres y no sobre variables afectadas por un cuantificador.

Definición 4.20 Un término t se dice *libre para una variable* x en una fórmula A si ninguna ocurrencia libre de x en A está dentro del alcance de un cuantificador $\forall y$ o $\exists y$, donde y es una variable de t.

Si t es libre para x en A, entonces el término t se puede sustituir en todas las instancias de la variable x sin que alguna variable y de t quede dentro del alcance de un cuantificador.

Ejemplo 4.21 Consideremos la fórmula

$$A = \forall x \left(B\left(x\right) \rightarrow C\left(y\right)\right).$$

El término $t = f\left(x\right)$, donde f es un símbolo de función unario, no es libre para y en A, pues cuando sustituimos y por $f\left(x\right)$ en A, obteniendo la fórmula $\forall x \left(B\left(x\right) \rightarrow C\left(f\left(x\right)\right)\right)$, la variable x queda bajo el alcance de $\forall x$ en A. El término $t = f\left(z\right)$ es libre para y en A.

Veamos otro ejemplo. Sea la fórmula

$$\forall x A\left(x, y\right) \rightarrow \forall z B\left(z, x\right).$$

El término $t = f\left(x, w\right)$ no es libre para y, pues al sustituir en la fórmula incial nos queda la fórmula $\forall x A\left(x, f\left(x, w\right)\right) \rightarrow \forall z B\left(z, x\right)$. El término $d = f\left(y, z\right)$ es libre para y en A, y el término y es libre para x.

Las definiciones anteriores se generalizan al caso de sustituciones de variables $x_1, ..., x_n$ por términos $t_1, ..., t_n$, en donde se supone que $x_i \neq x_j$ y x_i no es una variable de t_i. Una sustitución de este tipo se puede denotar como una sucesión finita de la forma

$$e = \left(x_1/t_1, ..., x_n/t_n\right).$$

Ejemplo 4.22 Consideremos la fórmula

$$A = \forall x \exists z \left[\left(P\left(x, y\right) \vee R\left(w\right) \vee Q\left(z\right)\right) \rightarrow \left(P\left(x, y\right) \vee \neg R\left(z\right)\right)\right]$$

y la sustitución

$$e = \left(x/c, y/f\left(a, b\right), z/y, w/f\left(a, a\right)\right),$$

donde f es un símbolo de función binario y a, b y c son constantes. Entonces la sustitución e aplicada a la fórmula A, en símbolos $e\left(A\right)$, es

$$e\left(A\right) = \forall x \exists z \left[\left(P\left(x, f\left(a, b\right)\right) \vee R\left(f\left(a, a\right)\right) \vee Q\left(z\right)\right) \rightarrow \left(P\left(x, f\left(a, b\right)\right) \vee \neg R\left(z\right)\right)\right].$$

Notemos que las variables x e z quedan inalteradas pues son ligadas en A.

4.3. Semántica de Primer Orden

Dado un lenguaje de primer orden $\mathcal{L} = \langle \mathcal{R}, \mathcal{F}, \mathcal{C} \rangle$, una fórmula A en dicho lenguaje es una sucesión finita de símbolos que no tiene una interpretación fija. Para interpretar las fórmulas de primer orden vamos a definir los modelos o interpretaciones del lenguaje. En el caso proposicional las interpretaciones o modelos eran las valuaciones sobre el conjunto $\{0, 1\}$. En el caso de primer orden los modelos son estructuras más complicadas, pues no solo debemos definir cómo interpretamos los símbolos $\vee, \wedge, \rightarrow$ y $\neg$, sino también debemos definir cómo interpretar los símbolos relacionales, funcionales, constantes y los cuantificadores $\forall$ y $\exists$.

Las fórmulas de un lenguaje de primer orden se interpretan sobre conjuntos, o también llamados dominios. Para entender mejor cómo se debe definir una interpretación vamos a dar primero un ejemplo de una fórmula en un lenguaje de primer orden con y posibles interpretaciones.

Ejemplo 4.23 Consideremos un lenguaje de primer orden $\mathcal{L}$ con una sola constante c y un símbolo de predicado binario R. Consideremos las siguientes fórmulas en este lenguaje

$$A = \forall y \exists x R\left(y, x\right).$$

$$B = \forall y \exists x R\left(x, y\right)$$

La fórmula A nos dice que para cualquier y existe un x tal que y está relacionado con x. Consideremos ahora el conjunto de los naturales $\mathbb{N} \cup \{0\}$ y sea $<$ la relación de orden estricto entre números naturales. La relación $<$ interpreta la relación R y el cero 0 interpreta la constante c. La fórmula A es válida sobre la terna $\langle \mathbb{N}, <, \{0\} \rangle$, pues para todo elemento $y \in \mathbb{N}$ existe un elemento $x \in \mathbb{N}$ tal que $x < y$. Por ejemplo, el elemento $y = x + 1$ cumple la condición de que $x < y$. En este caso podemos escribir

$$\langle \mathbb{N}, <, \{0\} \rangle \vDash \forall y \exists x R\left(y, x\right).$$

La fórmula B no es válida sobre la estructura $\langle \mathbb{N}, <, \{0\} \rangle$, pues existe un elemento y tal que para todo $x \not< y$. El elemento $y = 0$ no tiene antecesor. En este caso podemos escribir

$$\langle \mathbb{N}, <, \{0\} \rangle \nvDash \forall y \exists x R\left(x, y\right).$$

Si cambiamos de dominio, por ejemplo tomamos la estructura $\langle \mathbb{Z}, <, \{0\} \rangle$, entonces en este caso tenemos que

$$\langle \mathbb{Z}, <, \{0\} \rangle \vDash \forall y \exists x R\left(y, x\right)$$

y

$$\langle \mathbb{Z}, < \rangle \vDash \forall y \exists x R\left(x, y\right).$$

De acuerdo a este ejemplo, un lenguaje de primer orden puede tener muchos modelos o interpretaciones. Algunas fórmulas pueden ser verdaderas en algunos modelos y falsas en otro. Vamos ahora a decir exactamente que es un modelo o interpretación de un lenguaje.

Definición 4.24 Sea $\mathcal{L} = \langle \mathcal{R}, \mathcal{F}, \mathcal{C} \rangle$ un lenguaje de primer orden. Una ***interpretación*** o un ***modelo*** $\mathcal{M}$ en el lenguaje $\mathcal{L}$ es una estructura

$$\mathcal{M} = \left\langle D, \mathcal{R}^D, \mathcal{F}^D, \mathcal{C}^D \right\rangle$$

donde

1. D es el *dominio o universo de la interpretación.* Es un conjunto no vacío donde las variables del lenguaje toman valores.

2. $\mathcal{R}^D$ es un conjunto de relaciones n-arias definidas sobre D tal que para cada símbolo de relación n-ario $R \in \mathcal{R}$ de $\mathcal{L}$ existe una relación $R^D \subseteq D^n$ que es asignada a R.

3. $\mathcal{F}^D$ es un conjunto de funciones n-arias definidas sobre D tal que para cada símbolo de función n-ario $f \in \mathcal{F}$ de $\mathcal{L}$ existe una función $f^{\ D} : D^n \to D$ que es asignada a f.

4. $\mathcal{C}^D$ es un conjunto de elementos distinguidos de D tal que para cada símbolo de constante $c \in \mathcal{C}$ de $\mathcal{L}$ existe un elemento $c^D \in D$ tal que es asignado a c.

Dada una interpretación o modelo $\mathcal{M}$ es usual escribir el dominio D como $|\mathcal{M}|$. También podemos dar la definición de modelo como una estructura $\mathcal{M} = \left\langle D, \mathcal{R}^D, \mathcal{F}^D, \mathcal{C}^D \right\rangle$ junto con una función $I : \mathcal{L} \to D$ tal que cumpla las condiciones:

1. $I : \mathcal{R} \to \mathcal{R}^D : I(R) = R^D$.
2. $I : \mathcal{F} \to \mathcal{F}^D : I(f) = f^D$
3. $I : \mathcal{C} \to \mathcal{C}^D$: $I(c) = c^D$

En otras palábras, la función I asigna a cada símbolo de relación $R \in \mathcal{R}$ una relación R^D sobre D, a cada símbolo de función $f \in \mathcal{F}$ una función f^D definida sobre D, y a cada símbolo de constante $c \in \mathcal{C}$, un elemento distinguido $c^D \in D$.

Ejemplo 4.25 Sea $\mathcal{L}$ un lenguaje con un símbolo relacional unario R. En cualquier modelo $\mathcal{M} = \left\langle D, \left\{ R^D \right\} \right\rangle$ de este lenguaje la interpretación del símbolo relacional R corresponde a subconjuntos del dominio D, es decir $R^D \subseteq D$.

Ejemplo 4.26 Sea $\mathcal{L}$ un lenguaje con un símbolo relacional binario R y una constante c. Cualquier modelo $\mathcal{M}$ de este lenguaje deber tener definido en el dominio una relación binaria R^D y un elemento distinguido c^D tal que: $I(R) = R^D$ y $I(c) = c^D$. Por ejemplo, en el conjunto $D = \{a, b, c, d\}$ definimos la relación binaria

$$S = \{(a, a), (a, b), (a, c), (a, d), (b, c)\}$$

y supongamos que a es el elemento distinguido. Entonces

$$I(R) = R^D = S \text{ y } I(c) = c^D = a.$$

En este modelo podemos interpretar fórmulas en el lenguaje $\mathcal{L}$. Por ejemplo, intentemos interpretar las siguientes fórmulas:

1. $A = \forall x \exists y R(x, y)$
2. $B = \forall x R(a, x)$

La fórmula A no es verdadera en este modelo, pues el elemento d no está relacionado con ningún elemento. La fórmula B si es verdadera en $\mathcal{M}$ pues la interpretación de la constante c está relacionada con todo elemento de D.

4.3.1. Concepto de verdad

Ahora definiremos formalmente el concepto de fórmula válida en un modelo bajo una valuación, fórmula válida en un modelo, y finalmente fórmula válida en cualquier modelo. Primero definiremos la noción de valuación o asignación sobre un modelo.

Cuando queremos interpretar una fórmula en un modelo dado debemos asignar valores a las variables que ocurren en la fórmula. Por ejemplo en el conjunto de los números reales $\mathbb{R}$ consideremos la siguiente desigualdad:

$$x^2 - 4 < 0. \tag{4.1}$$

Sabemos que esta desigualdad es verdadera solo en el intervalo real $(-2, 2)$. Cualquier otro valor fuera de este intervalo no verifica la desigualdad. En otras palábras, para ciertas asignaciones de valores de $\mathbb{R}$ a la variable x la desigualdad es válida. Si queremos ser un poco más formal, podemos considerar un lenguaje que tenga un símbolo relacional binario R, dos símbolos de función binarios f y g y dos constantes a y b. Un modelo en este lenguaje debería ser de la forma $\left\langle D, \left\{R^D\right\}, \left\{f^D, g^D\right\}, \{a, b\}\right\rangle$. Para nuestro ejemplo $\mathcal{M} = \langle \mathbb{R}, \{<\}, \{\times, -\}, \{0, 4\}\rangle$ es un modelo donde $<= R^{\mathbb{R}}$, $\times = f^{\mathbb{R}}$, $- = g^{\mathbb{R}}$, $a^{\mathbb{R}} = 0$ y $b^{\mathbb{R}} = 4$. La desigualdad (4.1) se puede escribir como la fórmula con una variable libre x siguiente:

$$A(x) = R(g(f(x, x), 4), 0).$$

Luego la fórmula $A(x)$ es verdadera en el modelo $\mathcal{M}$ solo en los casos donde la variable x se le asigna valores del intervalo $(-2, 2)$.

La discusión anterior nos lleva a definir la noción de valuación o asignación de un modelo como una función que asigna a las variables elementos del dominio del modelo.

En todo lo que sigue supondremos que tenemos fijado un lenguaje de primer orden $\mathcal{L} = \langle \mathcal{R}, \mathcal{F}, \mathcal{C}\rangle$.

Definición 4.27 Sea $\mathcal{M} = \left\langle D, R^D, \mathcal{F}^D, \mathcal{C}^D\right\rangle$ un modelo del lenguaje $\mathcal{L}$. Una ***valuación o asignación*** v es una función

$$v : Var \to D.$$

Como el conjunto de variables Var es un conjunto numerable, es decir $Var = \{p_1, ..., p_n, ...\}$, una valuación $v : Var \to D$ puede ser descripta por medio de su imagen. En otras palabras, una valuación v puede ser definida como una sucesión

$$v = (a_1, ..., a_n, ..)$$

de elementos del dominio D, donde $v(p_i) = p_i^D = a_i$. También es muy utilizada la notación vectorial $\vec{a} = (a_1, ..., a_n, ...)$.

Ahora definiremos el valor de un término t bajo una valuación $\vec{a} = (a_1, ..., a_n, ...)$.

Definición 4.28 El *valor* o la *denotación* de un término $t(x_1, ..., x_n)$ bajo una valuación $\vec{a} = (a_1, ..., a_n, ...)$, en símbolos $t^D[\vec{a}]$ es un elemento del dominio D que se define recursivamente como:

1. Si t_i es una variable x_i entonces

$$x_i^D[\vec{a}] = a_i$$

1. *Si t es una constante $c \in \mathcal{C}$ y c^D es la interpretación de c en D, entonces*

$$c^D[\vec{a}] = c^D.$$

2. *Si $t = f(t_1, ..., t_n)$, donde $t_1, t_2, ..., t_n \in Ter(\mathcal{L})$ y $f \in \mathcal{F}$ es un símbolo de función n-ario, entonces*

$$t^D[\vec{a}] = f^D(t_1, t_2, ..., t_n)[\vec{a}] = f^D\left(t_1^D[\vec{a}], t_2^D[\vec{a}], ..., t_n^D[\vec{a}]\right).$$

donde $f^D : D^n \to D$ es la función n-$aria$ que interpreta el símbolo de función f, y t_i^D son las interpretaciones de los términos t_i sobre D.

Observación. Sea $\mathcal{M}$ un modelo. Para una valuación $\vec{a} = (a_1, ..., a_n, ..)$ y un término $t(x_1, ..., x_n)$, $t^D[\vec{a}]$ es el elemento del dominio D que se obtiene sustituyendo cada ocurrencia de la variable x_i por el elemento a_i de D en t.

Ejemplo 4.29 Sea $t = f(x_1, g(x_2, x_3))$, donde f y g son símbolos de función binarios y sea el modelo $\mathcal{M} = \langle \mathbb{R}, f^{\mathbb{R}}, g^{\mathbb{R}} \rangle$ donde $f^{\mathbb{R}} = +$ y $g^{\mathbb{R}} = \times$. Entonces para cualquier valuación $\vec{a} = (a_1, a_2, a_3, ...)$

$$\begin{aligned} t^D[\vec{a}] &= f^{\mathbb{R}}\left(x_1[\vec{a}], g^{\mathbb{R}}(x_2, x_3)[\vec{a}]\right) = f^{\mathbb{R}}\left(x_1[\vec{a}], g^{\mathbb{R}}\left(x_2^{\mathbb{R}}[\vec{a}], x_3^{\mathbb{R}}[\vec{a}]\right)\right) \\ &= f^{\mathbb{R}}\left(a_1, g^{\mathbb{R}}(a_2, a_3)\right) = a_1 + g^{\mathbb{R}}(a_2, a_3) = a_1 + (a_2 \times a_3). \end{aligned}$$

Ejemplo 4.30 Consideremos un lenguaje con un símbolo de función unario f y un símbolo de función binario g. Consideremos los términos $t_1(x, y) = f(g(x, y))$ y $t_2 = g(f(x), f(y))$. Sea el modelo $\mathcal{M} = \langle \mathbb{Z}_3, f^{\mathbb{Z}_3}, \oplus \rangle$ donde $\mathbb{Z}_3 = \{0, 1, 2\}$ y , $f^{\mathbb{Z}_3}$ y , $g^{\mathbb{Z}_3} = \oplus$ están definidas por las tablas:

x	, $f^{\mathbb{Z}_3}(x)$
0	0
1	2
2	2

$\oplus$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Para cualquier valuación $\vec{a} = (a_1, a_2) \in \mathbb{Z}_3 \times \mathbb{Z}_3$:

$$\begin{aligned} t_1^{\mathbb{Z}_3}(x, y)[a_1, a_2] &= f^{\mathbb{Z}_3}\left(g^{\mathbb{Z}_3}(x, y)\right)[a_1, a_2] &&= f^{\mathbb{Z}_3}(a_1 \oplus a_2). \\ t_2^{\mathbb{Z}_3}(x, y)[a_1, a_2] &= g^{\mathbb{Z}_3}\left(f^{\mathbb{Z}_3}(x), f^{\mathbb{Z}_3}(y)\right)[a_1, a_2] &&= f^{\mathbb{Z}_3}(a_1) \oplus, f^{\mathbb{Z}_3}(a_2) \end{aligned}$$

Por ejemplo, si $a = (1, 2)$, entonces

$$t_1^{\mathbb{Z}_3}[1, 2] = f^{\mathbb{Z}_3}(1 \oplus 2) = f^{\mathbb{Z}_3}(0) = 0,$$

y

$$t_2^{\mathbb{Z}_3}[1, 2] = f^{\mathbb{Z}_3}(1) \oplus f^{\mathbb{Z}_3}(2) = 2 \oplus 2 = 1.$$

Definición 4.31 Sea $\mathcal{M}$ un modelo y $\vec{a}$ valuación. Sea x_i una variable y $d \in D$. La valuación $\vec{a}_d^{x_i}$ es la valuación definida por:

$$\vec{a}_d^{x_i}(y) = \begin{cases} y^D[\vec{a}] & \text{si} \quad y \neq x_i \\ \\ d & \text{si} \quad y = x_i \end{cases}.$$

La valuación $\vec{a}_d^{x_i}$ es igual a la valuación $\vec{a}$, excepto tal vez, en el valor que asigna a la variable x. Si $\vec{a}$ está escrita como una secuencia $\vec{a} = (a_1, ..., a_{i-1}, a_i, a_{i+1}, ...)$ entonces

$$\vec{a}_d^{x_i} = (a_1, ..., a_{i-1}, d, a_{i+1}, ...)\,.$$

Si $\vec{a}$ es una valuación y x es una variable, entonces vamos a escribir $[d, \vec{a}]$ si el orden en que ocurre la variable no es importante.
Observación. Sea a una valuación. Entonces, si $x, y \in Var$, $x \neq y$, entonces $(\vec{a}_a^x)_b^y = (\vec{a}_b^y)_a^x$.

Ahora definiremos la validez o satisfacción de una fórmula en un modelo bajo una valuación en términos de la relación de satisfacción $\models$.

Definición 4.32 Sea $\mathcal{M}$ un modelo y $A(x_1, ..., x_n)$ una fórmula cuyas variables libres se encuentran entre las del conjunto $\{x_1, ..., x_n\}$. Sea $\vec{a}$ una valuación. Diremos que la fórmula $A(x_1, ..., x_n)$ es *válida* en $\mathcal{M}$ *bajo* $\vec{a}$, o que $\vec{a}$ *satisface* la fórmula $A(x_1, ..., x_n)$, en símbolos

$$\mathcal{M} \vDash A(x_1, ..., x_n)[\vec{a}]\,,$$

si se cumplen las siguientes cláusulas:

1. *Si A is $t_1 = t_2$, con t_1 y $t_2 \in Ter(\mathcal{L})$, entonces*
$$\mathfrak{M} \models A[\vec{a}] \Leftrightarrow t_1^D = \mathbf{t}_2^D.$$

2. Si $A \in At(\mathcal{L})$, es decir $A = R(t_1, ..., t_n)$, donde $t_i(x_1, ..., x_n) \in Ter(\mathcal{L})$ y R es un símbolo de predicado n-ario, entonces
$$\mathcal{M} \vDash R(t_1, ..., t_n)[\vec{a}] \Leftrightarrow \left(t_1^D[\vec{a}], ..., t_n^D[\vec{a}]\right) \in R^{\mathcal{M}}.$$

3. Si $A = \neg B$, entonces,
$$\mathcal{M} \vDash A[\vec{a}] \Leftrightarrow \mathcal{M} \nvDash B[\vec{a}]\,.$$

4. Si $A = B \wedge C$, entonces
$$\mathcal{M} \vDash (B \wedge C)[\vec{a}] \Leftrightarrow\ \mathcal{M} \vDash B[\vec{a}] \text{ y } \mathcal{M} \vDash C[\vec{a}]\,.$$

5. Si $A = B \vee C$, entonces
$$\mathcal{M} \vDash (B \vee C)[\vec{a}] \Leftrightarrow\ \mathcal{M} \vDash B[\vec{a}] \text{ o } \mathcal{M} \vDash C[\vec{a}]\,.$$

6. Si $A = B \rightarrow C$, entonces

$$\mathcal{M} \vDash (B \rightarrow C)[\vec{a}] \Leftrightarrow \mathcal{M} \vDash \neg B[\vec{a}] \text{ o } \mathcal{M} \vDash C[\vec{a}].$$

7. Si $A(x_1, ..., x_n) = \forall x B(x, x_1, ..., x_n)$, entonces

$$\mathcal{M} \vDash \forall x B(x, x_1, ..., x_n)[\vec{a}] \Leftrightarrow \text{para todo } d \in D,\ \mathcal{M} \vDash B[d, \vec{a}],$$

   donde d es el elemento de D que se asigna a la variable x.

8. Si $A(x_1, ..., x_n) = \exists x B(x, x_1, ..., x_n)$, entonces

$$\mathcal{M} \vDash \exists x B(x, x_1, ..., x_n)[\vec{a}] \Leftrightarrow \text{existe } d \in D,\ \mathcal{M} \vDash B(x, x_1, ..., x_n)[d, \vec{a}],$$

   donde d es el elemento de D que se asigna a la variable x.

Observación. Consideremos una fórmula $A(x_1, ..., x_n)$ donde $x_1, ..., x_n$ son todas las variables libres que ocurren en A. Más adelante probaremos un resultado, llamado Lema de Coincidencia, que afirma que si $\vec{a}$ y $\vec{b}$ son dos valuaciones definidas sobre un modelo $\mathcal{M}$ que coinciden sobre las variables de A (es decir, si $a_i = b_i$, para $1 \leq i \leq n$), entonces

$$\mathcal{M} \vDash A[\vec{a}] \Leftrightarrow \mathcal{M} \vDash A[\vec{b}].$$

En otras palabras, la validez de la fórmula A bajo una valuación $\vec{a}$ solo depende de los valores que dicha valuación asigna a las variables libres de A. En consecuencia, si $\vec{a} = (a_1, ..., a_n, ...)$, con $a_i \in D$, entonces podemos escribir $\mathcal{M} \vDash A[a_1, ..., a_n]$ en vez de $\mathcal{M} \vDash A[\vec{a}]$, pues los restantes valores a_i, con $i \geq n+1$ no son necesarios para evaluar la fórmula A.

Resumiendo, si $A(x_1, ..., x_n) = \forall x_{n+1} B(x_1, ..., x_n, x_{n+1})$, entonces

$$\begin{aligned} \mathcal{M} \vDash A(x_1, ..., x_n)[a_1, ..., a_n] &\Leftrightarrow \mathcal{M} \vDash \forall x_{n+1} B(x_1, ..., x_n, x_{n+1})[a_1, ..., a_n] \\ &\Leftrightarrow \mathcal{M} \vDash B(x_1, ..., x_n, x_{n+1})[a_1, ..., a_n, d], \text{ para todo } d \in D, \end{aligned}$$

y si $A(x_1, ..., x_n) = \exists x_{n+1} B(x_1, ..., x_n, x_{n+1})$, entonces

$$\begin{aligned} \mathcal{M} \vDash A(x_1, ..., x_n)[a_1, ..., a_n] &\Leftrightarrow \mathcal{M} \vDash \exists x_{n+1} B(x_1, ..., x_n, x_{n+1})[a_1, ..., a_n] \\ &\Leftrightarrow \mathcal{M} \vDash \exists (x_1, ..., x_n, x_{n+1})[a_1, ..., a_n, d], \text{ para algún } d \in D. \end{aligned}$$

Las fórmulas con n-variables libres se comportan como una ecuación en n-variables. Si queremos saber cúales son las valuaciones definidas sobre un modelo $\mathcal{M}$ que satisfacen a una fórmula $A(x_1, ..., x_n)$ debemos determinar cúales son las n-uplas $(a_1, ..., a_n) \in D^n$ que satisfacen a $A(x_1, ..., x_n)$, pues cada n-upla se identifica con una valuación.

El conjunto de todas las n-uplas $(a_1, ..., a_n)$ que satisfacen una fórmula $A(x_1, ..., x_n)$ define una relación n-aria A^D en D del siguiente modo:

$$A^D = \{(a_1, ..., a_n) : \mathcal{M} \vDash A(x_1, ..., x_n)[a_1, ..., a_n]\}.$$

Este conjunto se llama el *conjunto de soluciones de la fórmula* $A(x_1, ..., x_n)$. Por ejemplo el conjunto de soluciones de la fórmula $A(x) = x^2 - 4 < 0$ en el modelo $\mathcal{M} = \langle \mathbb{R}, \{<\}, \{+, .\} \rangle$ es el conjunto $A^D = (-2, 2)$.

Con esta notación podemos afirmar que una valuación $(a_1, ..., a_n)$ satisface a una fórmula $A(x_1, ..., x_n)$ si y solo si $(a_1, ..., a_n) \in A^D$, en símbolos

$$\mathcal{M} \vDash A[\vec{a}] \Leftrightarrow \vec{a} = (a_1, ..., a_n) \in A^D.$$

Aquellas fórmulas que son satisfechas por cualquier valuación $(a_1, ..., a_n)$, es decir cuando

$$A^D = D^{n,}$$

son las fórmulas *válidas* en el modelo $\mathcal{M}$.

Definición 4.33 Una fórmula A es ***válida en un modelo*** $\mathcal{M}$, en símbolos $\mathcal{M} \vDash A$, si y solo si es válida bajo cualquier valuación definida sobre el modelo. Es decir,

$$\mathcal{M} \vDash A \Leftrightarrow \mathcal{M} \vDash A[\vec{a}], \text{ para cualquier valuación } \vec{a}.$$

Diremos que A es ***válida*** o ***lógicamente válida***, si es válida en cualquier modelo. Diremos que una fórmula A *es* ***satisfacible*** si existe algún modelo $\mathcal{M}$ tal que $\mathcal{M} \vDash A$.

Ejemplo 4.34 Consideremos el modelo

$$\mathcal{M} = \langle D, P^D, Q^D, R^D \rangle$$

donde $D = \{1, 2, 3, 4\}$, $P^D = \{1, 2\}$, $Q^D = \{2, 3, 4\}$, y $R^D = \{(1, 2), (1, 3), (2, 4)\}$. Estudiemos si existen valuaciones definidas en $\mathcal{M}$ que satisfagan a la fórmula $A(x) = \exists y R(x, y)$, y verifiquemos si es válida en el modelo.

Observemos que la fórmula A tienen una sola variable libre. Debemos considerar, inicialmente, valuaciones definidas sobre la variable que aparece libre, es decir, estudiemos si existen valuaciones $v = (a)$ que satisfagan la fórmula $A(x)$, o lo que es lo mismo, debemos determinar la relación unaria

$$A^D = \{a \in D : \mathcal{M} \vDash A(x)[a]\}.$$

Si $a = 1$, tenemos que

$$\begin{aligned} \mathcal{M} \vDash A(x)[1] &\Leftrightarrow \mathcal{M} \vDash \exists y R(x, y)[1] \\ &\Leftrightarrow \text{existe algún } d \in D \text{ tal que } \mathcal{M} \vDash R(x, y)[1, d] \\ &\Leftrightarrow \text{existe algún } d \in D \text{ tal que } (1, d) \in R^D. \end{aligned}$$

Es claro que los valores $d = 2$ y $d = 3$ satisacen la última condición. Por lo tanto, la valuación $a = 1$ satisface a la fórmula.

Si $a = 2$, entonces

$$\mathcal{M} \vDash A(x)[2] \quad \Leftrightarrow \quad \text{existe algún } d \in D \text{ tal que } (2, d) \in R^D.$$

El elemento $d = 4$ cumple la condición anterior y por lo tanto la valuación $\vec{a} = (2)$ satisface la fórmula.

Si $a = 3$, tenemos que

$$\mathcal{M} \vDash A(x)[3] \quad \Leftrightarrow \quad \text{existe algún } d \in D \text{ tal que } (3, d) \in R^D.$$

Si miramos la definición de R^D observamos que no hay elementos $d \in D$ tal que $(3, d) \in R^D$. Por lo tanto la valuación $a = 3$ no satisface a la fórmula. De igual manera se deduce que $a = 4$ no satisface a $A(x)$. De acuerdo a este análisis la relación determinada por $A(x)$ (en este caso es un conjunto) es

$$A^D = \{1, 2\}.$$

Como $A^D \neq D$, es decir, existen valuaciones que no satisfacen a la fórmula, entonces $A(x)$ no es válida en $\mathcal{M}$.

Ejercicio 4.3.1 Utilizando el mismo modelo del ejemplo anterior, estudiar la satisfacibilidad y validez de las siguientes fórmulas.

1. $B(y) = \forall x ((Px \vee Qx) \rightarrow R(x, y))$
2. $C = \forall x \exists y (Px \rightarrow R(x, y) \vee Q(y, x))$.

Ejemplo 4.35 Consideremos la fórmula

$$A(x) = \forall y \forall z [(R(y, x) \wedge R(z, x)) \rightarrow R(y, z)].$$

Sea $\mathcal{M} = \langle D, R^D \rangle$ un modelo donde $D = \{a, b\}$ y $R^D = \{(a, a), (b, b), (b, a)\}$. Determinemos el conjunto A^D. En este caso como el dominio del modelo es pequeño podemos construir una tabla

y	z	x	$R(y, x)$	$R(z, x)$	$R(y, z)$
a	a	a	1	1	1
a	a	b			
a	b	a	1	1	
a	b	b		1	
b	a	a	1	1	
b	a	b	1		1
b	b	a	1	1	1
b	b	b	1	1	

De acuerdo con la tabla anterior, la fórmula $A(x)$ es válida solo para $x = a$. En consecuencia, $A^D = \{a\}$ y por lo tanto no es una fórmula válida.

Ejemplo 4.36 Consideremos un lenguaje de primer orden con una constante c y un símbolo de relación binario R. En dicho lenguaje consideremos la fórmula

$$A = \forall x R(x, c).$$

Sea $\mathcal{M}$ cualquier modelo, donde D es su dominio. Entonces

$$\begin{aligned} \mathcal{M} \vDash \forall x R(c, x) \quad &\Leftrightarrow \quad \text{para todo elemento } d \in D, \mathcal{M} \vDash R(c, x)[d] \\ &\Leftrightarrow \quad \text{para todo elemento } d \in D, \ (c^D, d) \in R^D. \end{aligned}$$

Consideremos ahora los siguientes modelos

$$\mathcal{M}_1 = \langle \mathbb{N}, \leq, \{0\} \rangle \quad \mathcal{M}_2 = \langle \mathbb{N}, \leq, \{1\} \rangle \quad \mathcal{M}_3 = \langle \mathbb{Z}, \leq, \{0\} \rangle.$$

Entonces de acuerdo al análisis dado anteriormente, tenemos que

$$\begin{array}{rcl}
\langle \mathbb{N}, \leq, \{0\} \rangle \vDash \forall x R(x, c) & \Leftrightarrow & \text{para todo elemento } d \in \mathbb{N},\ \left(c^D, d\right) \in R^D \\
& \Leftrightarrow & \text{para todo elemento } d \in \mathbb{N},\ c^{\mathbb{N}} = 0 \leq d \\
\\
\langle \mathbb{N}, \leq, \{2\} \rangle \nvDash \forall x R(x, c) & \Leftrightarrow & \text{existe un elemento } d \in \mathbb{N},\ \left(c^D, d\right) \notin R^D \\
& \Leftrightarrow & \text{existe un elemento } d \in \mathbb{N},\ c^{\mathbb{N}} = 2 \nleq d \\
& \Leftrightarrow & \text{existe } d = 1, \text{ tal que } 2 \nleq 1 \\
\\
\langle \mathbb{Z}, \leq, \{0\} \rangle \nvDash \forall x R(x, c) & \Leftrightarrow & \text{existe un elemento } d \in \mathbb{Z},\ \left(c^D, d\right) \notin R^D \\
& \Leftrightarrow & \text{existe un elemento } d \in \mathbb{Z},\ c^{\mathbb{N}} = 0 \nleq d \\
& \Leftrightarrow & \text{existe } d = -1, \text{ tal que } 0 \nleq -1
\end{array}$$

La fórmula $\forall x R(x, c)$ es satisfacible pues existe al menos el modelo $\mathcal{M}_1 = \langle \mathbb{N}, \leq, \{0\} \rangle$ tal que $\mathcal{M}_1 \vDash \forall x R(x, c)$. Pero esta fórmula no es válida pues existe el modelo $\mathcal{M}_2 = \langle \mathbb{N}, \leq, \{2\} \rangle$ donde $\mathcal{M}_2 \nvDash \forall x R(x, c)$.

El siguiente resultado justifica la utilización de valuaciones con un número finito de valores en una fórmula con n-variables.

Lema 4.37 (de Coincidencia) *Sean $\vec{a}$ y $\vec{b}$ dos valuaciones definidas sobre un modelo $\mathcal{M}$. Entonces para cada término $t(x_1, ..., x_n)$ y para cada fórmula $A(x_1, ..., x_n)$:*

1. *Si $a_i = b_i$ para toda variable de t, entonces*

$$t^D[\vec{a}] = t^D[\vec{b}].$$

2. *Si $a_i = b_i$ para toda variable libre de A, entonces*

$$\mathcal{M} \vDash A[\vec{a}] \Leftrightarrow \mathcal{M} \vDash A[\vec{b}].$$

Demostración. 1. La prueba es por inducción sobre la construcción del término $t.El$ primer caso a considerar es cuando el término es una variable x_i.
Supongamos que $t = x_i \in Var$. Entonces por hipótesis, $t^D[\vec{a}] = a_i = b_i = t^D[\vec{b}]$.
Consideremos ahora el caso en que t es una constante c. Si $t = c \in \mathcal{C}$, entonces $t^D[\vec{a}] = c^D = t^D[\vec{b}]$.
Ahora veamos el caso en que t es un término $f(t_1, ..., t_n)$ donde t_i son términos y f es un símbolo de función n-ario. En este caso utilizaremos la hipótesis inductiva para cada uno de los términos t_i. De acuerdo con esto, $t_i[\vec{a}] = t_i[\vec{b}]$ para cada término t_i. Entonces

$$\begin{aligned}
t^D[\vec{a}] &= f^D(t_1, ..., t_n)[\vec{a}] = f^D(t_1[\vec{a}], ..., t_n[\vec{a}]) \\
&= f^D\left(t_1[\vec{b}], ..., t_n[\vec{b}]\right) = f^D(t_1, ..., t_n)[\vec{b}] = t^D[\vec{b}].
\end{aligned}$$

2. Sea $A(x_1, ..., x_n)$ donde $x_1, ..., x_n$ son todas las variables libres de A. La prueba es por inducción sobre la longitud de la fórmula. Probaremos solo el caso de fórmulas atómicas y fórmulas cuantificadas universalmente. Los otros casos quedan como ejercicios.

Supongamos que $A = R(t_1, ..., t_n)$, donde R es un símbolo de relación n-ario y $t_1, ..., t_n$ son términos. Entonces

$$\begin{aligned}
\mathcal{M} \vDash R(t_1, ..., t_n)[\vec{a}] &\Leftrightarrow \left(t_1^D[\vec{a}], ..., t_n^D[\vec{a}]\right) \in R^D \\
&\overset{HI}{\Leftrightarrow} \left(t_1^D[\vec{b}], ..., t_n^D[\vec{b}]\right) \in R^D \\
&\Leftrightarrow \mathcal{M} \vDash R(t_1, ..., t_n)[\vec{b}]
\end{aligned}$$

Supongamos que $A(x_1, ..., x_n) = \forall y B(y, x_1, ..., x_n)$, donde $x_1, ..., x_n$ son todas las variables libres que ocurren en B. Luego,

$$\begin{aligned}
\mathcal{M} \vDash \forall y B(y, x_1, ..., x_n)[\vec{a}] &\Leftrightarrow \text{para todo } d \in D, \mathcal{M} \vDash B(y, x_1, ..., x_n)[d, \vec{a}] \\
&\Leftrightarrow \text{para todo } d \in D, \mathcal{M} \vDash B(y, x_1, ..., x_n)[d, \vec{b}] \\
&\Leftrightarrow \mathcal{M} \vDash \forall y B(y, x_1, ..., x_n)[\vec{b}].
\end{aligned}$$

■

Lema 4.38 *Sea t un término cerrado. Para cualquier par de valuaciones $\vec{a}$ y $\vec{b}$ definidas sobre un modelo $\mathcal{M}$,*

$$t^D[\vec{a}] = t^D[\vec{b}].$$

Si A es una sentencia, entonces para cualquier $\vec{a}$ y $\vec{b}$ definidas sobre un modelo $\mathcal{M}$,

$$\mathcal{M} \vDash A[a] \Leftrightarrow \mathcal{M} \vDash A[\vec{b}].$$

Demostración. Es inmediata por el Lema de Coincidencia. ■

Observaciones.

1. El Lema de Coincidencia nos afirma que la validez de fórmulas en un modelo solo depende de las variables libres que tenga la fórmula. En consecuencia, cuando trabajemos con fórmulas $A(x_1, ..., x_n)$ donde $x_1, ..., x_n$ son todas las variables libres, una asignación para dicha fórmula es suficiente describirla por medio de la secuencia finita $\vec{a} = (a_1, ..., a_n)$, donde a_i es el elemento de D que se le asigna a la variable x_i.

2. Sea t un término cerrado y $\mathcal{M}$ un modelo. Como la validez de t no depende de las valuaciones, entonces podemos escribir t^D en vez de $t^D[\vec{a}]$. De igual forma, si A es una sentencia, podemos escribir $\mathcal{M} \vDash A$ en vez de $\mathcal{M} \vDash A[\vec{a}]$.

3. Como hemos afirmado anteriormente, la validez de una sentencia depende de su estructura interna y no de las variables que tenga. Por ello, teniendo en cuenta la observación 2. anterior, la definición de validez de una sentencia A en un modelo $\mathcal{M}$ puede ser precisada de la siguiente forma:

   a) Si $A = R(t_1, ..., t_n)$, donde R es un símbolo de relación n-ario y $t_1, ..., t_n$ son términos cerrados, entonces
   $$\mathcal{M} \vDash R(t_1, ..., t_n) \Leftrightarrow \left(t_1^D, ..., t_n^D\right) \in R^D.$$

   b) Si $A = \neg B$, entonces
   $$\mathcal{M} \vDash \neg B \Leftrightarrow \mathcal{M} \nvDash B.$$

c) Si $A = B \wedge C$, entonces

$$\mathcal{M} \vDash B \wedge C \Leftrightarrow \mathcal{M} \vDash B \text{ y } \mathcal{M} \vDash C.$$

d) Si $A = B \vee C$, entonces

$$\mathcal{M} \vDash B \vee C \Leftrightarrow \mathcal{M} \vDash B \text{ o } \mathcal{M} \vDash C.$$

e) Si $A = B \rightarrow C$, entonces

$$\mathcal{M} \vDash B \rightarrow C \Leftrightarrow \mathcal{M} \vDash \neg A \text{ o } \mathcal{M} \vDash C.$$

f) Si $A = \forall x B(x)$, donde B tiene a lo sumo una única variable libre x, entonces

$$\mathcal{M} \vDash \forall x B(x) \Leftrightarrow \text{para cada } a \in D, \mathcal{M} \vDash B(x)[a].$$

g) Si $A = \exists x B(x)$, donde B tiene a lo sumo una única variable libre x, entonces

$$\mathcal{M} \vDash \exists x B(x) \Leftrightarrow \text{existe un } a \in D, \mathcal{M} \vDash B(x)[a].$$

Sea $A(x_1, ..., x_n)$ una fórmula donde $x_1, ..., x_n$ son sus variables libres. El próximo resultado afirma que la fórmula $A(x_1, ..., x_n)$ es válida en un modelo $\mathcal{M}$ si y solo si es válida su clausura universal.

Lema 4.39 *Sea $A(x_1, ..., x_n)$ una fórmula y $\mathcal{M}$ un modelo. Entonces*

$$\mathcal{M} \vDash A(x_1, ..., x_n) \Leftrightarrow \mathcal{M} \vDash \forall x_1 ... \forall x_n A(x_1, ..., x_n).$$

***Demostración*.** Probaremos solo el caso $A(x)$. Es decir cuando A tiene una sola variable libre.

$$\begin{aligned} \mathcal{M} \vDash \forall x A(x) &\Leftrightarrow \text{para toda valuación } a,\ \mathcal{M} \vDash A(x)[a] \\ &\Leftrightarrow \mathcal{M} \vDash A(x) \end{aligned}$$

■

El lema anterior nos dice que la validez de una fórmula en un modelo es lo mismo que la validez de su clausura universal en el modelo. Por este hecho, cuando hablemos de validez en modelos solo es necesario hacerlo para sentencias.

Lema 4.40 (de Sustitución) *Sea $\mathcal{M}$ un modelo. Sea t un término libre para una variable x. Sea a una valuación tal que $t^D[a] = b$. Entonces*

1. *Para todo término r,*

$$r^D(x/t)[a] = r^D(x)[b].$$

2. *Para toda fórmula A con t un término libre para x en A*

$$\mathcal{M} \vDash A(x/t)[a] \Leftrightarrow \mathcal{M} \vDash A(x)[b].$$

Demostración. 1. La prueba es por inducción sobre la longitud del término r.

Si $r = x$, entonces $r(x/t) = t$. Luego

$$r^D(x/t)[a] = t^D[a] = b$$

y

$$r^D(x)[b] = x^D[b] = b.$$

Si $r = y$, con y una variable distinta de x, entonces $r(x/t) = r$. En consecuencia,

$$\begin{aligned} r^D((x/t))[a] &= r^D[a] = y^D[a] = b. \\ r(x)[b] &= y^D[b] = b. \end{aligned}$$

Si $r = c$, con c una constante, entonces $r(x/t) = c$

$$r^D(x/t)[a] = c^D[a] = c^D = c^D[b] = t^D[b].$$

Sea $r = f(t_1, ..., t_n)$, donde f es un símbolo de función n-ario y $t_1, ..., t_n$ son términos que cumplen las hipótesis del Lema. Entonces:

$$\begin{aligned} r^D(x/t)[a] = c^D = f^D(t_1(x/t), ..., t_n(x/t))[a] &= f^D(t_1(x/t)[a], ..., t_n(x/t)[a]) \\ &= f^D(t_1(x)[b], ..., t_n(x)[b]) = f^D(t_1(x), ..., t_n(x))[b] \\ &= r^D(x)[b]. \end{aligned}$$

2. Sea A una fórmula y t un término libre para x en A. La prueba es por indución sobre la longitud de la fórmula.

Sea $A = R(t_1, ..., t_n)$, donde R es un símbolo de relación n-ario y $t_1, ..., t_n$ son términos. Entonces

$$\begin{aligned} \mathcal{M} \vDash R(t_1(x/t), ..., t_n(x/t))[a] &\Leftrightarrow (t_1(x/t)[a], ..., t_n(x/t)[a]) \in R^D \\ &\Leftrightarrow (t_1(x)[b], ..., t_n(x)[b]) \in R^D \\ &\Leftrightarrow \mathcal{M} \vDash R(t_1(x), ..., t_n(x))[b]. \end{aligned}$$

Sea $A = \forall y B(y, x)$, donde suponemos que x es libre en A. Entonces

$$\begin{aligned} \mathcal{M} \vDash \forall y B(y, x/t)[a] &\Leftrightarrow \text{para cualquier } d \in D, \mathcal{M} \vDash B(y, x/t)[d, a] \\ &\Leftrightarrow \text{para cualquier } d \in D, \mathcal{M} \vDash B(y, x)[d, b] \\ &\quad\ \mathcal{M} \vDash \forall y B(y, x)[b]. \end{aligned}$$

Como t es libre para x en A, entonces la variable y no puede ocurrir en t.
Si x no es libre en A, entonces A es una sentencia y por lo tanto $\mathcal{M} \vDash A[a] \Leftrightarrow \mathcal{M} \vDash A[b]$.

El caso $A = \exists y B(y, x)$ se analiza de manera similar y queda como ejercicio. ■

Vamos a finalizar esta sección dando algunas fórmulas válidas. Recordemos que una fórmula es válida si y solo si es válida en cualquier modelo.

Lema 4.41 *Sean A y B fórmulas con a lo sumo una variable libre x y sea c una constante. Entonces las siguientes fórmulas son válidas.*

1. $\forall x A(x) \rightarrow A(x)$
2. $A(x) \rightarrow \exists x A(x)$,
3. $\forall x A(x) \rightarrow A(x/c)$,
4. $A(x/c) \rightarrow \exists x A(x)$,
5. $\forall x A(x) \rightarrow \exists x A(x)$,
6. $\forall x \forall x A(x) \rightarrow \forall x A(x)$
7. $\exists x A(x) \rightarrow \exists x \exists x A(x)$
8. $\forall x (A(x) \rightarrow B(x)) \rightarrow (\forall x A(x) \rightarrow \forall x B(x))$,
9. $\forall x (A(x) \rightarrow B(x)) \rightarrow (\exists x A(x) \rightarrow \exists x B(x))$,
10. $\forall x (A(x) \wedge B(x)) \leftrightarrow (\forall x A(x) \wedge \forall x B(x))$,
11. $\exists x (A(x) \vee B(x)) \leftrightarrow \exists x A(x) \vee \exists x B(x)$,
12. $\forall x A(x) \wedge \exists x B(x) \rightarrow \exists x (A(x) \wedge B(x))$,
13. $\forall x (A(x) \vee B(x)) \rightarrow (\forall x A(x) \vee \exists x B(x))$,
14. $\forall x A(x) \leftrightarrow \neg \exists x \neg A(x)$,
15. $\exists x A(x) \leftrightarrow \neg \forall x \neg A(x)$,
16. $\neg \forall x A(x) \leftrightarrow \exists x \neg A(x)$,
17. $\neg \exists x A(x) \leftrightarrow \forall x \neg A(x)$,

Demostración. Probaremos a modo de ejemplo 3. y 8. Las demás pruebas quedan como ejercicio.

3. Sea $\mathcal{M}$ un modelo cualquiera. Debemos probar que $\mathcal{M} \vDash \forall x A(x) \rightarrow A(x/c)$. Es decir, debemos probar que si $\mathcal{M} \vDash \forall x A(x)$, entonces $\mathcal{M} \vDash A(x/c)$. Supongamos entonces que $\mathcal{M} \vDash \forall x A(x)$, es decir, para cualquier $d \in D$, $\mathcal{M} \vDash A(x)[d]$. En particular vale para $c^{\mathcal{M}}$, $\mathcal{M} \vDash A(x)\left[c^D\right]$. Pero por el Lema de Sustitución, tenemos las siguientes equivalencias,

$$\mathcal{M} \vDash A(x/c)[d] \Leftrightarrow \mathcal{M} \vDash A(x)\left[c^D[d]\right] \Leftrightarrow \mathcal{M} \vDash A(x)\left[c^D\right].$$

Por lo tanto, la sentencia es válida.

8. Sea $\mathcal{M}$ un modelo. Probar que $\mathcal{M} \vDash \forall x (A(x) \rightarrow B(x)) \rightarrow (\forall x A(x) \rightarrow \forall x B(x))$ es equivalente a probar que si $\mathcal{M} \vDash \forall x (A(x) \rightarrow B(x))$ y $\mathcal{M} \vDash \forall x A(x)$, entonces $\mathcal{M} \vDash \forall x B(x)$. Sea $a \in D$ arbitrario. Por la suposición, $\mathcal{M} \vDash A(x) \rightarrow B(x)[a]$ y $\mathcal{M} \vDash A(x)[a]$. Entonces, $\mathcal{M} \vDash B(x)[a]$. Como a es arbitrario, $\mathcal{M} \vDash \forall x B(x)$. ■

Lema 4.42 *Sea $A(x,y)$ una fórmula con a lo sumo dos variables libres. Entonces la sentencia*

$$\exists x \forall y A(x,y) \rightarrow \forall y \exists x A(x,y)$$

es válida.

Demostración. Ejercicio. ■

4.3.2. Equivalencia lógica

De forma similar a lo hecho en el caso proposicional, ahora definiremos el concepto de fórmulas equivalentes. Supondremos que hemos fijado un lenguaje de primer orden $\mathcal{L}$.

Definición 4.43 Diremos que las fórmulas A y B con las mismas variables libres son ***equivalentes***, en símbolos $A \equiv B$, si y solo para todo modelo $\mathcal{M}$,

$$\mathcal{M} \vDash A \Leftrightarrow \mathcal{M} \vDash B.$$

Es decir, A y B son equivalentes si son válidas en los mismos modelos.

Es inmediato comprobar que

$$A \equiv B \Leftrightarrow \text{para todo modelo } \mathcal{M}, \mathcal{M} \vDash A \leftrightarrow B.$$

Proposición 4.44 *Para cualquier fórmula A se verifica que:*

1. $\neg\forall xA \equiv \exists x\neg A$.
2. $\neg\exists xA \equiv \forall x\neg A$.
3. $\forall xA \equiv \neg\exists x\neg A$
4. $\exists xA \equiv \neg\forall\neg A$.

Demostración. Se deduce por el Lema 4.41. ■

Proposición 4.45 (Cambio de variables) *Sea A una fórmula. Entonces para toda variable y que no aparece en A valen las siguientes equivalencias:*

1. $\forall xA(x) \equiv \forall yA(x/y)$.
2. $\exists xA(x) \equiv \exists yA(x/y)$.

Demostración. Probamos 1. Sea $A(x)$ una fórmula con una sola variable libre. Consideremos un modelo $\mathcal{M}$. Primero veremos, por inducción que $A(x)[a] = A(x/y)[a]$, para cualquier $a \in D$ y para $y \notin Var(A(x))$. Analizamos algunos casos solamente.

Si $A(x)$ es un término $t(x)$ tal que $t(x) = x$, entonces $t(x/y) = y$. Luego, $t^D(x)[a] = x^D[a] = a$ y $t^D(x/y)[a] = y^D[a] = a$. Por lo tanto, $t^D(x)[a] = t^D(x/y)[a]$.

Si $A(x) = R(t_1(x), ..., t_n(x))$, donde R es un símbolo de relación n-ario y t_i son términos. Entonces

$$\begin{aligned} R^D(t_1(x), ..., t_n(x))[a] &= R^D\left(t_1^D(x)[a], ..., t_n^D(x)[a]\right) \\ &= R^D\left(t_1^D(x/y)[a], ..., t_n^D(x/y)[a]\right) \\ &= R^D(t_1(x/y), ..., t_n(x/y))[a]. \end{aligned}$$

Por lo tanto,

$$\begin{aligned} \mathcal{M} \vDash \forall xA(x) &\Leftrightarrow \text{para cada } a \in D,\ \mathcal{M} \vDash A(x)[a] \\ &\overset{HI}{\Leftrightarrow} \text{para cada } a \in D,\ \mathcal{M} \vDash A(x/y)[a], \text{ con } y \notin Var(A(x)) \\ &\Leftrightarrow \mathcal{M} \vDash \forall xA(x/y) \end{aligned}$$

■

Lema 4.46 *Sean A y B fórmulas tal que x no es una variable de B. Entonces*

1. $\forall x\,(A \vee B) \equiv (\forall x A) \vee B$,

2. $\exists x\,(A \wedge B) \equiv (\exists x A) \wedge B$.

Demostración. Ejercicio. ■

4.4. Consecuencia semántica

Cuando estudiamos el Cálculo Proposicional definimos la relación de consecuencia como una relación entre conjuntos de fórmulas y fórmulas. Ahora definiremos una noción similar para la lógica de Primer Orden.

Definición 4.47 Sea Γ un conjunto de sentencias en un lenguaje $\mathcal{L}$. Diremos que Γ es *satisfacible* si existe un modelo $\mathcal{M}$ tal que $\mathcal{M} \vDash A$ para toda sentencia $A \in \Gamma$. Para abreviar escribiremos $\mathcal{M} \vDash \Gamma$. Un conjunto es *insatisfacible* si no es satisfacible.

Diremos que una fórmula A es *consecuencia* de un conjunto de sentencias Γ, en símbolos $\Gamma \vDash A$, si para cada modelo $\mathcal{M}$ tal que $\mathcal{M} \vDash \Gamma$, entonces $\mathcal{M} \vDash A$.

Ejemplo 4.48 El conjunto de sentencias

$$\Gamma = \{\forall x \exists y\,(P(x) \to R(x,y)),\ \forall x \forall y\,[\neg R(x,y) \to (\neg P(x) \vee \neg P(y))]\}.$$

es satisfacible, pues por ejemplo, en el modelo $\mathcal{M} = \langle D, P^D, R^D \rangle$, donde $D = \{1,2,3\}$, $P^D = \{1,2\}$ y $R^D = \{(1,1),(2,1),(1,2)\}$, las sentencias son válidas.

Ejemplo 4.49 El conjunto de sentencias

$$\Gamma = \{\exists x\,(P(x) \wedge Q(x)),\ \forall x\,(P(x) \to R(x,x)),\ \forall x\,(Q(x) \to \neg R(x,x))\}$$

es insatisfacible. Para comprobar esta afirmación razonaremos por contradicción. Supongamos que existe un modelo $\mathcal{M}$ que satisface a las tres sentencias simultaneamente. Como satisface a la primer sentencia existe un $a \in D$ tal que $\mathcal{M} \vDash P(a)$ y $\mathcal{M} \vDash Q(a)$. La segunda sentencia debe también ser satisfecha por el elemento a, es decir $\mathcal{M} \vDash P(a) \to R(a,a)$. Por lo tanto, como $\mathcal{M} \vDash P(a)$ entonces

$$\mathcal{M} \vDash R(a,a).$$

Pero también el elemento a debe satisfacer la tercer sentencia; $\mathcal{M} \vDash Q(a) \to \neg R(a,a)$, y como $\mathcal{M} \vDash Q(a)$ entonces $\mathcal{M} \vDash \neg R(a,a)$, lo que es una contradicción. En consecuencia, el conjunto de sentencias Γ es insatisfacible.

Ejemplo 4.50 La sentencia $\forall x\,(\neg Q(x) \to \exists y R(x,y))$ es consecuencia del conjunto de sentencias

$$\Gamma = \{\forall x\,(P(x) \vee Q(x)),\ \forall x\,(P(x) \to \exists y R(x,y))\}.$$

En efecto. Supongamos que existe un modelo $\mathcal{M}$ tal que satisface al conjunto de sentencias Γ. Probemos que $\mathcal{M} \vDash \forall(\neg Q(x) \to \exists y R(x,y))$. Sea $a \in D$ arbitrario. Debemos comprobar que $\mathcal{M} \vDash \neg Q(x) \to \exists y R(x,y)[a]$, es decir, si $\mathcal{M} \vDash \neg Q(x)[a]$, entonces $\mathcal{M} \vDash \exists y R(x,y)[a]$. Ahora, si $\mathcal{M} \vDash \neg Q(x)[a]$ entonces $\mathcal{M} \nvDash Q(x)[a]$. Como $\mathcal{M} \vDash P(x) \vee Q(x)[a]$, entonces $\mathcal{M} \vDash P(x)[a]$. Por último como en $\mathcal{M}$ la sentencia $\forall x\,(P(x) \to \exists y R(x,y))$ es válida, entonces $\mathcal{M} \vDash \exists y R(x,y)[a]$.

La relación de consecuencia semántica tiene algunas propiedades comunes con la relación de consecuencia semántica definida en lógica proposicional. Esta coincidencia no es casual. Muchas de las propiedades de la relación son propiedades de tipo universales. Es decir, no dependen de la lógica en que uno esté trabajando. Dependen más de la propia definición de la relación en términos de los modelos.

Lema 4.51 *Sea Γ un conjunto de sentencias y A una sentencia. Entonces*

1. *Si $A \in \Gamma$, entonces $\Gamma \models A$.*
2. *Si $\Gamma \models A$ y $\Gamma \subseteq \Delta$, donde Δ es un conjunto de sentencias, entonces $\Delta \models A$.*
3. *Si $\Gamma \models A$ y $A \models B$ entonces $\Gamma \models B$.*

Demostración. Ejercicio. ■

Teorema 4.52 (de la Deducción) *Sea $\Gamma \cup \{A, B\}$ un conjunto de sentencias. Entonces,*

$$\Gamma \cup \{A\} \models B \Leftrightarrow \Gamma \models A \rightarrow B.$$

Demostración. Ejercicio. ■

Lema 4.53 *Sea $\Gamma \cup \{A\}$ un conjunto de sentencias. Entonces,*

$$\Gamma \models A \textit{ si y solo si } \Gamma \cup \{\neg A\} \textit{ es insatisfacible.}$$

Demostración. Ejercicio. ■

Capítulo 5

Teoría de Herbrand

La semántica de la lógica proposicional está basada en el conjunto $\{0,1\}$, lo que hace que la teoría de modelos de la lógica proposicional sea relativamente simple. Esto no ocurre cuando trabajamos en lógica de predicados. Una de las dificultades en lógica de predicados la encontramos en que la definición de modelos está basada en conjuntos arbitrarios. En esta sección vamos a desarrollar una teoría para construir modelos de fórmulas o conjuntos de fórmulas en una forma canónica. Por medio de estos modelos es posible conocer, por lo menos en una forma teórica, si un conjunto de cláusulas es satisfacible o no.

Primero estudiaremos las formas normales y la forma cláusular de una fórmula. Posteriormente daremos un resultado debido a Thoralf Skolem y finalmente definiremos los modelos de Herbrand. Toda la teoría desarrollada en este capítulo es de fundamental importancia para la Programación Lógica.

5.1. Formas Normales

Vamos a generalizar algunos conceptos definidos en Lógica Proposicional, como la noción de literal y cláusula.

Definición 5.1 Un ***literal*** es una fórmula atómica o la negación de una fórmula atómica.

Por ejemplo, en el lenguaje con un símbolo de constante c, un símbolo de predicado binario P y un símbolo de función unario f, las siguientes son literales:

$$l_1 = P(a,a)$$
$$l_2 = \neg P(a, f(a))$$

Un literal será denotado por l. La negación del literal l será denotada por l^c.

Definición 5.2 Una fórmula está en la ***forma normal conjuntiva*** si es una conjunción de disyunciones de literales. Una fórmula está en la ***forma normal conjuntiva prenexa***, o directamente en forma prenexa, si es de la forma

$$Q_1 x_1 .... Q_n x_n M(x_1, ..., x_n),$$

donde Q_i son cuantificadores y $M(x_1, ..., x_n)$ es una fórmula libre de cuantificadores que está en la forma normal conjuntiva.

Por ejemplo, la fórmula

$$\forall x \exists y \left(\left(P\left(x,a\right) \vee Q\left(y\right) \right) \wedge \neg P\left(a,b\right) \vee Q\left(f\left(f\left(a\right)\right)\right) \right)$$

está en la forma prenexa.

La secuencia de cuantificadores $Q_1x_1....Q_nx_n$ se llama el *prefijo* de la fórmula, y la fórmula $M\ (x_1, ..., x_n)$ se llama la *matriz* de la fórmula.

Más adelante comprobaremos que toda fórmula es equivalente a una fórmula escrita en la forma prenexa.

Ahora definiremos la noción de cláusula. Una cláusula es una sentencia escrita en la forma prenexa pero que en el prefijo *solo* aparecen cuantificadores universales y además, la matriz es una disyunción de literales. Más adelante veremos el motivo para estudiar este tipo de sentencias.

Definición 5.3 Una ***cláusula*** es una fórmula de la forma

$$\forall x_1....\forall x_n \left(l_1 \vee l_2 \vee .... \vee l_n\right)$$

donde l_i son literales. Una fórmula esta en la ***forma normal de Skolem (o también forma clausular)*** si es una conjunción de cláusulas. Es decir, una fórmula esta en la forma clausular si es de la forma

$$\forall x_1....\forall x_n \left(C_1 \wedge ... \wedge C_n\right)$$

donde C_i son cláusulas, y $x_1, ..., x_n$ son todas las variables que ocurren en la conjunción $C_1 \wedge ... \wedge C_n$.

Nota. Como toda variable que ocurre en una cláusula siempre está cuantificada universalmente, usualmente omitiremos los cuantificadores.

Una fórmula A que está escrita en la forma normal de Skolem también puede ser escrita como un conjunto de cláusulas. Es decir, si

$$A = \forall x_1....\forall x_n \left(C_1 \wedge ... \wedge C_n\right)$$

entonces podemos escribir

$$\{C_1, C_2, ..., C_n\}\,.$$

En este caso diremos que la fórmula A está escrita en la forma ***clausular.*** De ahora en más no haremos distinción entre una forma normal de Skolem y la forma clausular asociada.

Ejemplo 5.4 La fórmula

$$A = \forall x \forall y \forall z \left[\left(P\left(x,y\right) \vee Q\left(x, f\left(z\right)\right) \right) \wedge \left(f\left(x,c\right) \vee \neg P\left(x,z\right) \right) \right]$$

está escrita en la forma normal de Skolem. En notación de conjuntos, la fórmula A queda escrita como $A = \left\{ P\left(x,y\right) \vee Q\left(x, f\left(z\right)\right),\ \left(f\left(x,c\right) \vee \neg P\left(x,z\right)\right) \right\}$.

Cuando estudiamos resolución proposicional probamos que el conjunto vacio de cláusulas $S = \emptyset$ es siempre satisfacible. Es decir, siempre existe un modelo $\mathcal{M}$ tal que $\mathcal{M} \vDash S$. En lógica de predicados podemos probar exatamente lo mismo. Por lo tanto, en el resto del capítulo el conjunto vacío de clásusulas $S = \emptyset$ será siempre satisfacible. Similarmente, el conjunto de literales vacio o cláusula vacia será simbolizado por $\perp$ y se prueba que $S = \{\perp\}$ es un conjunto de cláusulas insatisfacibles.

Ahora veremos un procedimiento que permite determinar una fórmula A' escrita en la forma prenexa equivalente a una fórmula A dada. Recordemos que para cualquier par de fórmulas A y B valen las equivalencias $A \vee \neg A \equiv B \vee \neg B = \top$ y $A \wedge \neg A \equiv B \wedge \neg B \equiv \perp$.

- PROCEDIMIENTO PARA CONVERTIR UNA FÓRMULA EN UNA FÓRMULA EQUIVALENTE ESCRITA EN LA FORMA PRENEXA.

  1. Renombrar variables si es necesario. Para ello recordar las equivalencias
     a) $\forall x A(x) \equiv \forall y A(y)$ y $\exists x A(x) \equiv \exists y A(y)$, donde y es una variable que no aparece en A.
  2. Usar la equivalencia $A \to B \equiv \neg A \vee B$ para eliminar el conectivo $\to$.
  3. Usar las equivalencias
     a) $\neg(A \wedge B) \equiv \neg A \vee \neg B$
     b) $\neg(A \vee B) \equiv \neg A \wedge \neg B$
     c) $\neg\neg A \equiv A$
     d) $\neg\forall x A(x) \equiv \exists x \neg A(x)$
     e) $\neg\exists x A(x) \equiv \forall x \neg A(x)$

     para poner la negación $\neg$ inmediatamente antes de una fórmula atómica.
  4. Escribir todos los cuantificadores al principio. Para ello recordar las siguientes equivalencias:
     a) $\forall x(A(x) \wedge B) \equiv \forall x A(x) \wedge B$, donde $x \notin Vl(B)$.
     b) $\forall x(A(x) \vee B) \equiv \forall x A(x) \vee B$, donde $x \notin Vl(B)$.
     c) $\exists x(A(x) \vee B) \equiv \exists x A(x) \vee B$, donde $x \notin Vl(B)$.
     d) $\exists x(A(x) \vee B) \equiv \exists x A(x) \vee B$, donde $x \notin Vl(B)$.
  5. Usar las leyes distributivas:
     a) $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$.
     b) $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$.

Ejemplo 5.5 Convertir a la forma prenexa la siguiente fórmula

$$\forall x(P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x)).$$

1. $\forall x(P(x) \to Q(x)) \to (\forall y P(y) \to \forall z Q(z))$

2. $\neg\forall x(\neg P(x) \vee Q(x)) \vee (\neg\forall y P(y) \vee \forall z Q(z))$

3. $\exists x(P(x) \wedge \neg Q(x)) \vee \exists y \neg P(y) \vee \forall z Q(z)$

4. $\exists x \exists y \forall z((P(x) \wedge \neg Q(x)) \vee \neg P(y) \vee Q(z))$

5. $\exists x \exists y \forall z \left(\underbrace{(P(x) \vee \neg P(y) \vee Q(z))}_{C_1} \wedge \underbrace{(\neg Q(x) \vee \neg P(y) \vee Q(z))}_{C_2} \right)$

6. $\exists x \exists y \forall z (C_1 \wedge C_2)$

La fórmula obtenida en el ejemplo anterior está en la forma prenexa pero no es una fórmula en la forma cláusular. Para obtener, dada una fórmula A, una fórmula en la forma cláusular debemos emplear un método para eliminar los cuantificadores existenciales. La fórmula que se obtiene **no** es equivalente a la fórmula original, pero sí es satisfacible siy sólo si la original lo es. Este requerimiento es suficiente para nuestros propositos, como veremos más adelante.

Supongamos tener una fórmula $A = \forall x \exists y P(x, y)$ que está en la forma prenexa. Queremos eliminar el cuantificador existencial $\exists y$. La fórmula dice que para todo elemento x existe un elemento y tal que hacen que el predicado $P(x, y)$ sea válido. Es decir, para cada elemento x existe, al menos, un elemento y asociado tal que hacen válido el predicado $P(x, y)$. Ahora, supongamos tener una función $y = f(x)$. La definición de función nos dice que dado el elemento x existe un *único* elemento y asociado a x. Es decir, una función $y = f(x)$ es un caso particular de relación. Las dos nociones tienen algo en común. En el caso de la fórmula A pueden existir muchos elementos asociados con x. En el caso de la función $y = f(x)$ existe un único elemento. Si sustituimos en la fórmula A la variable y por $f(x)$ tendremos la fórmula

$$A' = \forall x P(x, f(x)).$$

Esta fórmula expresa la idea que para todo elemento x existe un único elemento $f(x)$ tal que hacen válido el predicado $P(x, f(x))$. Las fórmulas A y A' no son equivalentes, pero sí se cumple lo siguiente

existe un modelo $\mathcal{M}$ de A si y sólo si existe un modelo $\mathcal{M}'$ de A'.

En otras palabras,

A es satisfacible sí y solo sí A' es satisfacible.

Sean A y A' dos fórmulas. Escribiremos $A \approx A'$ para denotar que A es satisfacible si y sólo si A' es satisfacible.

En el próximo resultado no solo vamos a probar que una fórmula A es satisfacible si y sólo si existe una forma normal de Skolem satisfacible, también vamos a dar un método que nos permite deteminar una forma normal de Skolem.

Teorema 5.6 (de Skolem) *Sea A una sentencia. Entonces existe un fórmula A' en la forma normal de Skolem $A \approx A'$.*

Demostración. Para construir la fórmula A' tal que $A \approx A'$ debemos seguir los siguientes pasos.

1. Determinar una fórmula B que esté en la forma prenexa equivalente a la fórmula A.

2. Para eliminar los cuantificadores existenciales se deben seguir los siguientes pasos.

Supongamos que $\exists x$ es un cuantificador existencial que ocurre en B y sean $y_1, y_2, ..., y_n$ las variables que ocurren cuantificadas universalmente antes del cuantificador $\exists x$. Es decir, tenemos la siguiente fórmula

$$\forall y_1 \forall y_2 ... \forall y_n \exists x\, M(x, y_1, y_2, ..., y_n)$$

Considerar un nuevo símbolo de función f de aridad n. Para eliminar $\exists x$ reemplazar toda ocurrencia de la variable x por $f(y_1, y_2, ..., y_n)$. Entonces nos queda la fórmula

$$A' = \forall y_1 \forall y_2 ... \forall y_n\, M(f(y_1, y_2, ..., y_n), y_1, y_2, ..., y_n).$$

Si la fórmula B es del tipo

$$\exists x M\left(x\right)$$

es decir, no hay cuantificadores universales antes de $\exists x$, entonces sustituimos la variable x por una nueva constante a. Entonces nos queda la fórmula

$$A' = M\left(a\right).$$

La fórmula A' determinada es la fórmula buscada.
Ahora debemos probar que $A \approx A'$.

Supongamos que

$$A = \forall y_1 \forall y_2 ... \forall y_n \exists x\, P\left(x, y_1, y_2, ..., y_n\right),$$

donde P es un predicado de aridad $n+1$. Supongamos que $\mathcal{M}$ es un modelo de A. Es decir,

$$\mathcal{M} \vDash \forall y_1 \forall y_2 ... \forall y_n \exists x\, P\left(x, y_1, y_2, ..., y_n\right).$$

Debemos probar que existe un modelo $\mathcal{M}'$ tal que

$$\mathcal{M}' \vDash \forall y_1 \forall y_2 ... \forall y_n\ P\left(f\left(y_1, y_2, ..., y_n\right), y_1, y_2, ..., y_n\right)$$

donde f es un nuevo símbolo de función de aridad n.

Como $\mathcal{M} \vDash A$, entonces para toda secuencia $a_1, a_2, ..., a_n \in D$ se verifica que

$$\mathcal{M} \vDash\ \exists x\, P\left(x, y_1, y_2, ..., y_n\right)\left[a_1, a_2, ..., a_n\right]$$

Entonces, eiste un elemento $d \in D$ tal que

$$\mathcal{M} \vDash\ P\left(x, y_1, y_2, ..., y_n\right)\left[d, a_1, a_2, ..., a_n\right],$$

es decir

$$\left(d, a_1, a_2, ..., a_n\right) \in P^D \tag{5.1}$$

Definimos un nuevo modelo $\mathcal{M}'$ agregando al modelo $\mathcal{M}$ una nueva función n-aria $f^D : D^n \to D$ definida por

$$f^D\left(a_1, a_2, ..., a_n\right) = d$$

donde $a_1, a_2, ..., a_n$ es una secuencia arbitraria y d es el elemento tal que $\left(a_1, a_2, ..., a_n, d_{n+1}\right) \in P^D$. Tal elemento existe por (5.1).

Probemos ahora que

$$\mathcal{M}' \vDash \forall y_1 \forall y_2 ... \forall y_n\ P\left(f\left(y_1, y_2, ..., y_n\right), y_1, y_2, ..., y_n\right).$$

Es decir, debemos probar que para cualquier secuencia $a_1, a_2, ..., a_n \in D$,

$$\mathcal{M}' \vDash P\left(f\left(y_1, y_2, ..., y_n\right), y_1, y_2, ..., y_n\right)\left[a_1, a_2, ..., a_n\right],$$

lo que es equivalente a probar que para cualquier secuencia $a_1, a_2, ..., a_n \in D$,

$$\left(f\left(a_1, a_2, ..., a_n\right), a_1, a_2, ..., a_n\right) = \left(d, a_1, a_2, ..., a_n\right) \in P^D,$$

pero esto es inmediato por (5.1). Como la secuencia $a_1, a_2, ..., a_n$ es arbitraria, entonces tenemos que

$$\mathcal{M}' \vDash \forall y_1 \forall y_2 ... \forall y_n \ P\left(f\left(y_1, y_2, ..., y_n\right), y_1, y_2, ..., y_n\right).$$

La prueba en la otra dirección es sencilla y se deja a cargo del lector. ■

Si A es una fórmula, entonces vamos a denotar con $Cl\left(A\right)$ al conjunto de cláusulas de una forma normal de Skolem asociada a la fórmula A. Notemos que la forma clausular de una fórmula no es necesariamente única.

Inmediatamente obtenemos el siguiente corolario el cual nos da un método para determinar la validez de una fórmula.

Corolario 5.7 *Sea A una fórmula. Entonces A es insatisfacible si y sólo si cualquier forma clausular asociada $Cl\left(A\right)$ es insatisfacible.*

Ejemplo 5.8 Dada la siguiente fórmula

$$A = \exists x \forall y P\left(x, y\right) \rightarrow \forall y \exists x P\left(x, y\right),$$

determinar una fórmula en la forma clausular A' tal que $A \approx A'$.

Primeros convertimos A en la forma prenexa.

1. $\exists x \forall y P\left(x, y\right) \rightarrow \forall y \exists x P\left(x, y\right)$
2. $\exists x \forall y P\left(x, y\right) \rightarrow \forall w \exists z P\left(z, w\right)$
3. $\neg \exists x \forall y P\left(x, y\right) \vee \forall w \exists z P\left(z, w\right)$
4. $\forall x \exists y \neg P\left(x, y\right) \vee \forall w \exists z P\left(z, w\right)$
5. $\forall x \exists y \forall w \exists z \neg P\left(x, y\right) \vee P\left(z, w\right)$

Ahora reemplazamos los cuantificadores existenciales por apropiadas funciones de Skolem para obtener la deseada fórmula A'

$$\forall x \forall w \neg P\left(x, f\left(x\right)\right) \vee P\left(g\left(x, w\right), w\right).$$

Alternativamente, cuando llegamos al paso 4 anterior: $\forall x \exists y \neg P\left(x, y\right) \vee \forall w \exists z P\left(z, w\right)$ eliminamos los cuantificadores existenciales por apropiadas funciones de Skolem, obteniendo otra fórmula A''

$$\forall x \neg P\left(x, f\left(x\right)\right) \vee \forall w P\left(g\left(w\right), w\right).$$

De esta forma empleamos funciones de Skolem de menor aridad.

Ejemplo 5.9 Convertir la siguiente fórmula a una fórmula en la forma clausular.

$$A = \forall x\left(P\left(x\right) \rightarrow Q\left(x\right)\right) \rightarrow \left(\forall x P\left(x\right) \rightarrow \forall x Q\left(x\right)\right).$$

1. $\forall x\left(P\left(x\right) \rightarrow Q\left(x\right)\right) \rightarrow \forall x P\left(x\right) \rightarrow \forall x Q\left(x\right).$

2. $\forall x\,(P\,(x) \to Q\,(x)) \to \forall y P\,(y) \to \forall z Q\,(z)\,.$
3. $\exists x\,(P\,(x) \wedge \neg Q\,(x)) \vee \exists y \neg P\,(y) \vee \forall z Q\,(z)\,.$
4. $(P\,(a) \wedge \neg Q\,(a)) \vee \neg P\,(b) \vee \forall z Q\,(z)\,.$
5. $\forall z\,((P\,(a) \wedge \neg Q\,(a)) \vee \neg P\,(b) \vee Q\,(z))\,.$
6. $\forall z\,((P\,(a) \vee \neg P\,(b) \vee Q\,(z)) \vee (\neg Q\,(a) \vee \neg P\,(b) \vee Q\,(z)))\,.$

La forma clausular asociada es $Cl\,(A) = \{P\,(a) \vee \neg P\,(b) \vee Q\,(z)\,, \neg Q\,(a) \vee \neg P\,(b) \vee Q\,(z)\}.$

Ejemplo 5.10 Estudiar si la fórmula $A = \forall x\,(P\,(x) \to Q\,(x)) \to (\exists x P\,(x) \to \exists x Q\,(x))$ es lógicamente válida. Recordemos que A es lógicamente válida si y sólo si $\neg A$ es insatisfacible. Por el Corolario anterior $\neg A$ es insatisfacible si y sólo si cualquier forma clausular $Cl\,(\neg A)$ es insatisfacible. Aplicamos esta última equivalencia para determinar si A es válida. Determinemos $Cl\,(\neg A)$

1. $\neg A = \neg\,[\forall x\,(P\,(x) \to Q\,(x)) \to (\exists x P\,(x) \to \exists x Q\,(x))]$
2. $\forall x\,(\neg P\,(x) \vee Q\,(x)) \wedge \exists y P\,(y) \wedge \neg \exists z Q\,(z)$
3. $\forall x\,(\neg P\,(x) \vee Q\,(x)) \wedge P\,(a) \wedge \forall z \neg Q\,(z)$
4. $\forall x \forall z\,[(\neg P\,(x) \vee Q\,(x)) \wedge P\,(a) \wedge \neg Q\,(z)]$

Luego, $Cl\,(\neg A) = \{\neg P\,(x) \vee Q\,(x)\,, P\,(a)\,, \neg Q\,(z)\}$. En busca de una contradicción suponemos que existe un modelo $\mathcal{M}$ tal que $\mathcal{M} \vDash Cl\,(\neg A)$. Entonces se debe cumplir que $\mathcal{M} \vDash \neg P\,(x) \vee Q\,(x)$, $\mathcal{M} \vDash P\,(a)$ y $\mathcal{M} \vDash \neg Q\,(z)$, lo que es evidentemente una contradicción. Por lo tanto $Cl\,(\neg A)$ es insatisfacible y en consecuencia A es válida.

5.2. Modelos de Herbrand

El objetivo de esta sección es introducir un tipo particular de modelo, llamado modelo de Herbrand, que es especialmente importante en la semántica de la programación lógica. Estos modelos están construidos a partir del conjunto de términos cerrados (términos sin variables) del lenguaje considerado, a partir del conjunto de términos que ocurren en un conjunto de fórmulas.

El resultado más importante que probaremos en esta parte dice que un conjunto de cláusulas tienen un modelo si y sólo si tiene un modelo de Herbrand. A partir de este resultado se deduce que un conjunto de cláusulas es insatisfacible si y sólo si no tiene modelos de Herbrand. Para ciertos casos de fórmulas esto nos dá un procedimiento para averiguar si una fórmula es una fórmula lógicamente válida.

Definición 5.11 Sea $\mathcal{L}$ un lenguaje de primer orden. El conjunto de los términos cerrados de $\mathcal{L}$ *se* llama el universo de *Herbrand* y lo simbolizaremos con $U\,(\mathcal{L})$. Es decir,

1. si $c \in \mathcal{C}$, entonces $c \in U\,(\mathcal{L})$
2. si $t_1, t_2, ..., t_n \in U\,(\mathcal{L})$ y f es un símbolo de función n-ario, entonces $f\,(t_1, t_2, ..., t_n) \in U\,(\mathcal{L})\,.$

Si $\mathcal{C} = \emptyset$, entonces iniciamos el universo de Herbrand $U(\mathcal{L})$ con un nuevo símbolo de constante c.

Los elementos de $U(\mathcal{L})$ también son llamados *ground términos.*

Definición 5.12 Sea $\mathcal{L}$ un lenguaje de primer orden. La *base de Herbrand,* en símbolos $B(\mathcal{L})$, es el conjunto de las fórmulas atómicas cerradas de $\mathcal{L}$ siguiente:

$$B(\mathcal{L}) = \{P(t_1, t_2, ..., t_n) : t_i \in U(\mathcal{L})\},$$

donde P es un predicado n-ario.

Los elementos de $B(\mathcal{L})$ se denominan *ground fórmulas atómicas.* Las definiciones anteriores también se aplican a conjuntos de cláusulas.

Ahora definimos conceptos análogos pero referidos a un conjunto de cláusulas.

Definición 5.13 Sea S un conjunto de cláusulas. ***El universo de Herbrand*** de S, en símbolos $U(S)$, es el conjunto de términos cerrados de S. Es decir,

1. Si c es un símbolo de constante que aparece en S, entonces $c \in U(S)$.
2. Si $t_1, t_2, ..., t_n \in U(S)$ y f es un símbolo de función n-ario que aparece en S, entonces $f(t_1, t_2, ..., t_n) \in U(S)$.

La ***base de Herbrand*** de S, en símbolos $B(S)$, es el conjunto de las fórmulas atómicas cerradas siguiente:

$$B(S) = \{P(t_1, ..., t_n) : t_i \in U(S)\},$$

donde $P(t_1, ..., t_n)$ es un predicado n-ario que ocurre en S.

Es importante remarcar que la base de Herbrand está formada por *todas* las instancias de fórmulas atómicas de S donde los términos son tomados del universo de Herbrand.

Ejemplo 5.14 Sea $\mathcal{L}$ un lenguaje de primer orden que tiene un símbolo de constante c, un predicado binario P, un predicado unario Q y sin símbolos de función. Entonces

$$\begin{array}{rcl} U(\mathcal{L}) & = & \{c\} \\ B(\mathcal{L}) & = & \{P(c,c), Q(c)\}. \end{array}$$

Ejemplo 5.15 Sea $\mathcal{L}$ un lenguaje de primer orden que tiene dos símbolos de constante c y d, un predicado binario P, un predicado unario Q y símbolo de función unario f. Entonces

$$\begin{array}{rcl} U(\mathcal{L}) & = & \{c, d, f(c), f(d), f(f(c)), f(f(d)), ...\} \\ \\ B(\mathcal{L}) & = & \{P(c,c), Q(c), P(c,d), Q(f(c)), P(f(f(c)), d), ...\}. \end{array}$$

Ejemplo 5.16 Consideremos los siguientes conjuntos de cláusulas

$$S_1 = \{P(a) \vee \neg P(b), \neg Q(z) \vee \neg P(b)\}$$

$$S_2 = \{\neg P(a, f(x, y)) \vee P(b, f(x, y))\}$$

Entonces

$$\begin{aligned} U(S_1) &= \{a, b\} \\ B(S_1) &= \{P(a), P(b), Q(a), Q(b)\} \\ \\ U(S_2) &= \{a, b, f(a, a), f(a, b), f(b, b), f(f(a, b), a), ...\} \\ B(S_2) &= \{P(a, b), P(a, a), P(f(a, b), b), P(a, f(a, b)), ...\} \end{aligned}$$

Definición 5.17 *Una **estructura de Herbrand*** en un lenguaje $\mathcal{L} = \langle \mathcal{C}, \mathcal{R}, \mathcal{F} \rangle$ es un modelo $\mathcal{M}_H = \left\langle D, \mathcal{C}^D, \mathcal{R}^D, \mathcal{F}^D \right\rangle$ donde

1. $D = U(\mathcal{L})$
2. Para cada $c \in \mathcal{C}$, entonces $c^D = c$
3. Para cada $f \in \mathcal{F}$, $f^D(t_1, t_2, ..., t_n) = f\left(t_1^D, t_2^D, ..., t_n^D\right)$.

Es decir, una estructura de Herbrand en el lenguaje $\mathcal{L}$ es un modelo cuyo universo es el universo de Herbrand y donde la interpretación de los términos es fija. Podemos elegir libremente la interpretación de los símbolos relacionales fijando algún subconjunto de fórmulas atómicas. En una estructura de Herbrand la validez de fórmulas queda determinada por un subconjunto de la base de Herbrand. De acuerdo a la elección del subconjunto tendremos que ciertas fórmulas pueden ser válidas o no. Más precisamente, si $P(t_1, ..., t_n) \in B(\mathcal{L})$ e $Y \subseteq B(\mathcal{L})$, entonces

$$\mathcal{M}_H \vDash P(t_1, ..., t_n) \Leftrightarrow P(t_1, ..., t_n) \in Y.$$

Observemos que la fórmula atómica $P(t_1, ..., t_n)$ es *cerrada,* es decir, no tiene variables. En los argumentos de los términos que ocurren en la fórmula atómica solo pueden ocurrir constantes. Las fórmulas atómicas válidas en una estructura de Herbrand son las fórmulas atómicas tal que existe alguna instancia cerrada de ella que pertenecen al conjunto elegido. Con esta definición, se determina la validez de una fórmula en forma recursiva.

Definición 5.18 Sea A una fórmula o un conjunto de cláusulas. Una estructura de Herbrand $\mathcal{M}$ es un ***modelo de Herbrand para*** A si es un modelo de A, es decir si $\mathcal{M} \vDash A$.

Veamos algunos ejemplos.

Ejemplo 5.19 Sea $\mathcal{L}$ un lenguaje de primer orden con tres símbolos de constantes a, b y c, dos predicados unarios P y Q y un predicado binario R. Entonces

$$U(\mathcal{L}) = \{a, b, c\}$$

$$B(\mathcal{L}) = \left\{ \begin{array}{l} P(a), P(b), P(c), Q(a), Q(b), Q(c), R(a, a), R(b, b) \\ R(c, c), R(a, b), R(b, a), R(a, c), R(c, a), R(b, c), R(c, b) \end{array} \right\}$$

Un ejemplo de estructura de Herbrand es

$$\mathcal{M}_1(H) = \left\langle U(\mathcal{L}), \mathcal{C}^D, \mathcal{R}^D, \mathcal{F}^D \right\rangle$$

donde

$$a^{U(\mathcal{L})} = a, \quad b^{U(\mathcal{L})} = b, \quad c^{U(\mathcal{L})} = c,$$

Fijando un subconjunto Y de $B(\mathcal{L})$ obtenemos un modelo de Herbrand. Por ejemplo, podemos tomar $Y_1 = \{P(a), R(a,b), R(a,c)\}$.

A modo de ejemplo comprobemos si la fórmula

$$A = \forall x (P(x) \vee \neg Q(x))$$

es válida en el modelo de Herbrand $\mathcal{M}_1(H)$. Debemos probar que

$$\begin{aligned}
\mathcal{M}_1(H) \vDash \forall x (P(x) \vee \neg Q(x)) \quad &\Leftrightarrow \quad \text{para todo } d \in U(\mathcal{L}),\ \mathcal{M}_1(H) \vDash P(x) \vee \neg Q(x)[d] \\
&\Leftrightarrow \quad \text{para todo } d \in U(\mathcal{L}),\ \mathcal{M}_1(H) \vDash P(x)[d] \qquad o \\
& \quad \text{para todo } d \in U(\mathcal{L}),\ \mathcal{M}_1(H) \vDash \neg Q(x)[d]
\end{aligned}$$

Si $d = a$, $\mathcal{M}_1(H) \vDash P(x)[a]$, pues $P(a) \in Y_1$. Entonces $\mathcal{M}_1(H) \vDash P(x) \vee \neg Q(x)[a]$
Si $d = b$, $\mathcal{M}_1(H) \vDash \neg Q(x)[b]$, pues $Q(b) \notin Y_1$. Entonces, $\mathcal{M}_1(H) \vDash P(x) \vee \neg Q(x)[b]$.
Si $d = c$, $\mathcal{M}_1(H) \vDash P(x) \vee \neg Q(x)[c]$ pues $Q(c) \notin Y_1$. Por lo tanto, $\mathcal{M}_1(H) \vDash A$.

Otra interpretación se podría dar fijando el subconjunto $Y_2 = \{P(c), Q(a), Q(c), R(a,b)\}$.

Ejemplo 5.20 Consideremos la siguiente cláusula $S = \{\neg P(a, f(x,y)) \vee P(b, f(x,y))\}$. Entonces,

$$\begin{aligned}
U(S) &= \{a, b, f(a,a), f(a,b), f(b,b), f(f(a,b), a), ...\} \\
B(S) &= \{P(a,b), P(a,a), P(f(a,b), b), P(a, f(a,b)), ...\}.
\end{aligned}$$

Para definir un modelo de Herbrand fijamos un subconjunto de la base de Herbrand $B(S)$. Por ejemplo,

$$Y = \{P(b, f(a,a)), P(b, f(a,b)), P(b, f(b,a)), P(b, f(b,b))\}.$$

Con esta interpretación se verifica que

$$\mathcal{M}(H) \vDash \neg P(a, f(x,y)) \vee P(b, f(x,y)).$$

es decir,

$$\mathcal{M}(H) \vDash \forall x \forall y (\neg P(a, f(x,y)) \vee P(b, f(x,y)))$$

Queda como ejercicio comprobar esta afirmación.

Ahora probaremos el teorema más importante de esta sección. Recordemos que si Γ es un conjunto de fórmulas y $\mathcal{M}$ es un modelo, la notación $\mathcal{M} \vDash \Gamma$ significa que todas las fómulas de Γ son válidas en el modelo $\mathcal{M}$, es decir $\mathcal{M} \vDash A$ para toda $A \in \Gamma$.

Teorema 5.21 (de Herbrand) *Sea S un conjunto de cláusulas. Entonces S tiene un modelo si y sólo si$S$ tiene un modelo de Herbrand.*

Demostración. Supongamos que $\mathcal{M}$ es un modelo de S. Como $S = \{C_1, C_2, ..., C_n, ...\}$, entonces $\mathcal{M} \vDash C_i$ para todo $i = 1, ..., n$. Debemos determinar un modelo de Herbrand $\mathcal{M}(H)$ tal que $\mathcal{M}(H) \vDash C_i$, para todo $i = 1, ..., n$. El modelo de Herbrand queda fijado si tomamos un subconjunto de la base de Herbrand $B(S)$ de S. Consideremos el siguiente subconjunto Y de $B(S)$

$$\begin{aligned} Y &= \{P(t_1, ..., t_n) \in B(S) : \ \mathcal{M} \vDash P(t_1, ..., t_n)\} \\ &= \left\{P(t_1, ..., t_n) \in B(S) : \left(t_1^D, ..., t_n^D\right) \in P^D\right\}. \end{aligned}$$

El conjunto Y es el subconjunto de la base $B(S)$ formado por todas las fórmulas atómicas válidas en el modelo,

$$\mathcal{M}(H) \vDash P(t_1, ..., t_n) \Leftrightarrow \mathcal{M} \vDash P(t_1, ..., t_n) \text{ y } P(t_1, .., t_n) \in Y.$$

Observemos, que inicialmente, no todas las fórmulas de $B(S)$ tienen que ser válidas en el modelo $\mathcal{M}$. Consideremos una cláusula $C_i \in S$. Como $C_i = \bigvee l_{ij}$, donde l_{ij} son literales, entonces l_{ij}

$$\begin{aligned} \mathcal{M} \vDash C_i &\Leftrightarrow \mathcal{M} \vDash l_{ij}, \text{ para algún } l_{ij} \in C_i \\ &\Leftrightarrow \mathcal{M}(H) \vDash l_{ij} \\ &\Leftrightarrow \mathcal{M}(H) \vDash C_i \end{aligned}$$

Como lo anterior es válido para cualquier cláusula $C_i \in S$, entonces $\mathcal{M}(H) \vDash S$.

La recíproca es inmediata. ■

Observación. En general, el teorema anterior no es válido para fórmulas. Por ejemplo, supongamos tener la fórmula

$$A = P(a) \wedge \exists x \neg P(x).$$

El universo de Herbrand de A es

$$U(A) = \{a\}$$

y la base de Herbrand de A es

$$B(A) = \{P(a)\}.$$

Los posibles modelos de Herbrand de A se determinan conociendo todos los posibles subconjuntos de $B(A)$. En este caso solo tenemos dos posibles subconjuntos

$$Y_1 = \emptyset$$

$$Y_2 = \{P(a)\}.$$

Es decir, podemos definir dos posibles modelos de Herbrand $\mathcal{M}_1(H)$ y $\mathcal{M}_2(H)$, donde

$$\mathcal{M}_1(H) \nvDash P(a)$$

$$\mathcal{M}_2(H) \vDash P(a).$$

En el primer modelo, como $\mathcal{M}_1(H) \nvDash P(a)$, entonces $\mathcal{M}_1(H) \nvDash A$. En el segundo modelo, $\mathcal{M}_2(H) \vDash P(a)$ y en consecuencia $\mathcal{M}_2(H) \nvDash \neg P(a)$. Esto implica que, $\mathcal{M}_2(H) \nvDash \exists x \neg P(x)$. Es decir,

$\mathcal{M}_2(H) \nvDash A$. Por lo tanto A no tiene modelos de Herbrand. Pero veamos ahora que A si tiene un modelo. Sea $\mathcal{M}$ el modelo donde el universo es $D = \{0, 1\}$, $a^D = 0$ y $P^D = \{0\}$ la interpretación del predicado P. Como $0 \in P^D$, entonces

$$\mathcal{M} \vDash P(a).$$

Como $1 \notin P^D$, entonces

$$\mathcal{M} \nvDash P(1),$$

es decir

$$\mathcal{M} \vDash \exists x \neg P(x).$$

Por lo tanto,

$$\mathcal{M} \vDash A.$$

Del Teorema de Herbrand y del Teorema de Skolem podemos deducir el siguiente fundamental resultado el cual nos proporciona un método para determinar si una sentencia A es lógicamente válida.

Teorema 5.22 *Sea A una sentencia . Entonces las siguientes condiciones son equivalentes:*

1. *A es válida,*
2. *$\neg A$ es insatisfacible*
3. *Cualquier forma clausular $Cl(\neg A)$ de A, es insatisfacible.*
4. *Cualquier forma clausular $Cl(\neg A)$ de A no tiene modelos de Herbrand.*

Ejemplo 5.23 Sean P y Q dos predicados unarios. Ya sabemos que la fórmula $A = \forall x (P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x))$ es una fórmula lógicamente válida. Podemos llegar a la misma conclusión aplicando el teorema anterior. Para ello consideremos la forma normal conjuntiva de la negación de A :

$$\begin{array}{rcl} \neg A & = & \neg [\forall x (P(x) \to Q(x)) \to (\forall x P(x) \to \forall x Q(x))] \\ & \equiv & \forall x (\neg P(x) \vee Q(x)) \wedge \forall x P(x) \wedge \neg \forall x Q(x) \\ & \equiv & \forall x (\neg P(x) \vee Q(x)) \wedge \forall y P(x) \wedge \neg \forall z Q(x) \\ & \equiv & \forall x (\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \exists z \neg Q(z) \end{array}$$

Una forma clausular de $\neg A$ es

$$Cl(\neg A) = \{\neg P(x) \vee Q(x), P(y), \neg Q(a)\}.$$

Ahora determinamos el unicverso de Herbrand y la base de Herbrand de $Cl(\neg A)$:

$$\begin{array}{rcl} U(Cl(\neg A)) & = & \{a\} \\ \\ B(Cl(\neg A)) & = & \{P(a), Q(a)\}. \end{array}$$

Los posibles modelos de Herbrand para $Cl(\neg A)$ están determinados por los subconjuntos de $B(Cl(\neg A))$: $Y_1 = \emptyset$, $Y_2 = \{P(a)\}$, $Y_2 = \{P(b)\}$ y $Y_4 = \{P(a), Q(a)\}$. Ahora es sencillo comprobar que ninguna estructura de Herbrand es un modelo de Herbrand para $Cl(\neg A)$. Por lo tanto, $\neg A$ es insatisfacible, o lo que es lo mismo, A es válida.

5.3. P-satisfacibilidad

Definición 5.24 Una cláusula cerrada es una cláusula sin variables. Es decir, es una cláusula donde en los argumentos de los símbolos de predicados solo aparecen términos cerrados. Un conjunto de cláusulas cerradas es un conjunto de cláusulas cerradas.

Si S es un conjunto de cláusulas y P es un conjunto de términos cerrados, entonces $P(S)$ es el conjunto de instancias de cláusulas de S donde las variables libres que ocurren en S han sido sustituidas por términos cerrados del conjunto P.

Ejemplo 5.25 El siguiente es un conjunto de cláusulas cerradas

$$S = \{P(a,b),\ R(b) \vee Q(f(a))\}.$$

Ejemplo 5.26 Sea $S = \{P(a,x) \vee \neg Q(x),\ R(b) \vee Q(x),\ \neg R(a) \vee P(x,y)\}$ un conjunto de cláusulas. El conjunto de constantes que ocurren en S es $\{a,b\}$. Recordemos que toda constante es un término. Si $P = \{a\}$, entonces

$$P(S) = \{P(a,a) \vee \neg Q(a),\ R(b) \vee Q(a),\ \neg R(a) \vee P(a,a)\}.$$

En un conjunto de cláusulas cerradas las fórmulas atómicas que ocurren en S pueden ser tratadas como variables proposicionales. De esta manera podemos relacionar la satisfacibilidad de un conjunto de cláusulas cerradas con la satisfacibilidad de un conjunto de variables proposicionales. En lo que sigue precisaremos esta relación.

Definición 5.27 Sea S un conjunto de cláusulas cerradas. Diremos que S es ***proposicionalmente satisfacible**, o que es p-**satisfacible**,* si S es satisfacible como un conjunto de cláusulas proposicionales, donde las fórmulas atómicas cerradas $P(t_1,...,t_n)$ que ocurren en S son tratadas como variables proposicionales.

Ejemplo 5.28 Consideremos el siguiente conjunto de cláusulas cerradas

$$S = \{P(a,b) \vee R(b),\ \neg R(b) \vee Q(a),\ R(b) \vee \neg Q(a)\}.$$

Las fórmulas atómicas que ocurren en S son

$$P(a,b),\ R(b),\ Q(a).$$

Llamando $p_1 = P(a,b)$, $p_2 = R(b)$ y $p_3 = Q(a)$, entonces S puede ser considerado como un conjunto de cláusulas proposicionales $S' = \{p_1 \vee p_2,\ \neg p_2 \vee p_3,\ p_2 \vee \neg p_3\}$. Es sencillo comprobar que la valuación v definida por $v(p_1) = 1$, $v(p_2) = 1$ y $v(p_3) = 1$ satisface al conjunto S'. Por lo tanto, el conjunto de cláusulas cerradas S es p-satisfacible.

Lema 5.29 *Sea S un conjunto de cláusulas cerradas. Entonces S es p-satisfacible si y sólo si tiene un modelo de Herbrand.*

Demostración. $\Rightarrow$) Sea S un conjunto de cláusulas cerradas p-satisfacible. Luego existe una valuación v tal que para cada fórmula atómica $P(t_1,...,t_n)$ que ocurre en S, $v(P(t_1,...,t_n)) = 1$ o $v(P(t_1,...,t_n)) = 0$. Consideremos el subconjunto Y de $B(S)$ siguiente

$$Y = \{P(t_1,...,t_n) : v(P(t_1,...,t_n)) = 1\}.$$

Es claro que el modelo de Herbrand $\mathcal{M}_Y(S)$ generado por el conjunto Y es un modelo de S, es decir, $\mathcal{M}_Y(S) \vDash S$.

$\Leftarrow$) Supongamos que S tiene un modelo de Herbrand $\mathcal{M}_Y(S)$ para un subconjunto $Y \subseteq B(S)$. Definimos la valuación v por

$$v(P(t_1,...,t_n)) = 1 \Leftrightarrow P(t_1,...,t_n) \in Y.$$

Es claro que S bajo esta valuación es p-satisfacible. ■

Ejemplo 5.30 Consideremos el siguiente conjunto de cláusulas

$$S = \{P(a,b) \vee R(b), \neg R(b) \vee Q(a), R(b) \vee \neg Q(a)\}.$$

Determinemos el universo de Herbrand y la base de Herbrand de S:

$$U(S) = \{a,b\}$$

$$B(S) = \{P(a,a), P(a,b), P(b,a), P(b,b), Q(a), Q(b), R(a), R(b)\}.$$

Como hemos visto en el ejemplo anterior, el conjunto S es p-satisfacible. Ahora, de acuerdo con el Lema anterior, S debe tener un modelo de Herbrand. Es decir debe existir, al menos, un subconjunto Y de la base de Herbrand tal que $\mathcal{M}_Y(H) \vDash S$. Observemos que el conjunto

$$Y_1 = \{R(b), Q(a)\}$$

define una estructura de Herbrand $\mathcal{M}_{Y_1}(H)$ tal que $\mathcal{M}_{Y_1}(H) \vDash S$. Notemos también que el conjunto $Y_2 = \{P(a,b), R(b), Q(a)\}$ también define un modelo de Herbrand para S. Este es el modelo de Herbrand que está asociado a la valuación v definida en el ejemplo anterior.

Corolario 5.31 *Sea S un conjunto de cláusulas (no necesariamente cerradas). Entonces las siguientes condiciones son equivalentes:*

1. *S tiene un modelo de Herbrand.*
2. *$U(S)(S)$ es p-satisfacible (donde $U(S)(S)$ denota el conjunto de todas las instancias cerradas de claúsulas de S sobre el universo $U(S)$).*
3. *Todo subconjunto finito $S_0 \subseteq U(S)(S)$ es p-satisfacible.*

Demostración. $1 \Rightarrow 2$. Asumimos que S tiene un modelo de Herbrand $\mathcal{M}_Y(S)$ para algún subonjunto de fórmulas atómicas cerradas $Y \subseteq B(S)$. Luego el modelo $\mathcal{M}_Y(S)$ satisface también el conjunto $U(S)(S)$ de todas las instancias cerradas de S sobre $U(S)$, es decir, $\mathcal{M}_Y(S) \vDash U(S)(S)$. Por el Lema anterior se sigue que $U(S)(S)$ es p-satisfacible.

$2 \Rightarrow 1$. Si $U(S)(S)$ es p-satisfacible, entonces existe una valuación v tal que $v(C_i) = 1$ para toda cláusula cerrada de $U(S)(S)$. Es decir, para cada cláusula cerrada C_i de $U(S)(S)$ existe una fórmula atómica cerrada $P(t_1, ..., t_n)$ ($P(t_1, ..., t_n)$ es un elemento de $B(S)$) tal que $v(P(t_1, ..., t_n)) = 1$ o $v(\neg P(t_1, ..., t_n)) = 1$. Entonces es claro que el conjunto

$$Y = \{P(t_1, ..., t_n) \in B(S) : v(P(t_1, ..., t_n)) = 1\}$$

define un modelo de Herbrand $\mathcal{M}_Y(S)$ de S.

Las pruebas de $2 \Rightarrow 3$ y $3 \Rightarrow 2$ se siguen por el Teorema de Compacidad para la lógica Proposicional. ■

Considerando a un conjunto de cláusulas cerradas como un conjunto de variables proposicionales podemos aplicar lo visto para resolución proposicional a este caso. En el siguiente resultado resumimos todo lo visto hasta el momento para determinar la insatisfacibilidad de un conjunto de cláusulas. Este resultado se aplica a una fórmula tA pero utilizando alguna forma clausular $Cl(\neg A)$ asociada a $\neg A$.

Teorema 5.32 *Sea S un conjunto de cláusulas. Entonces las siguientes condiciones son equivalentes:*

1. *S es insatisfacible.*
2. *S no tiene modelos de Herbrand.*
3. *$U(S)(S)$ es p-insatisfacible.*
4. *Existe un subconjunto finito $S_0 \subseteq U(S)(S)$ p-insatisfacible.*
5. *Existe un subconjunto finito $S_0 \subseteq U(S)(S)$ tal que $S_0 \vdash_R \bot$.*

De acuerdo con los resultados anteriores si un conjunto de cláusulas S no tiene un modelo de Herbrand, entonces no tiene modelos, es decir, S es insatisfacible. Pero si S es insatisfacible entonces la fórmula A que dió origen al conjunto de cláusulas S es insatisfacible, lo que es equivalente a decir que $\neg A$ es una fórmula válida. Por lo tanto, un método para saber si una fórmula A es válida es considerar la negación $\neg A$, determinar una forma clausular asociada $Cl(\neg A)$ y estudiar si tiene modelos de Herbrand. Si no tiene modelos de Herbrand entonces $\neg A$ será insatisfacible. Este método no es eficiente, pues, por ejemplo, si un conjunto de cláusulas tiene un símbolo de función entonces el universo de Herbrand es infinito y en consecuencia no podremos determinar todas las posibles estructuras de Herbrand. En estos casos debemos modificar el método o aplicar algún razonamiento que nos permita decir que un conjunto de cláusulas es insatisfacible sin necesidad de considerar todos los posibles modelos de Herbrand.

Ejemplo 5.33 Comprobar que el siguiente conjunto de cláusulas es insatisfacible.

$$S = \{P(x) \vee \neg Q(x, a), P(a) \vee Q(y, a), \neg P(y)\}.$$

Determinemos los posibles modelos de Herbrand

$$\begin{aligned} U(S) &= \{a\} \\ B(S) &= \{P(a), Q(a, a)\} \end{aligned}$$

. Las posibles estruturas de Herbrand quedan definidas por medio de los siguientes subconjuntos

$$Y_1 = \{P(a), Q(a,a)\} \qquad \mathcal{M}_1 \vDash P(a) \wedge Q(a,a)$$

$$Y_2 = \{Q(a,a)\} \qquad \mathcal{M}_2 \vDash Q(a,a) \text{ y } \mathcal{M}_2 \vDash \neg P(a)$$

$$Y_3 = \{P(a)\} \qquad \mathcal{M}_3 \vDash P(a) \text{ y } \mathcal{M}_3 \vDash \neg Q(a,a)$$

$$Y_4 = \{\emptyset\} \qquad \mathcal{M}_4 = \neg P(a) \wedge \neg Q(a,a)$$

Es claro que para todo $i = 1, .., 4$

$$\mathcal{M}_i \nvDash (P(x) \vee \neg Q(x,a)) \wedge (P(a) \vee Q(y,a)) \wedge \neg P(y),$$

Por lo tanto S es insatisfacible. En consecuencia, la fórmula

$$A = \neg \forall x \forall y ((P(x) \vee \neg Q(x,a)) \wedge (P(a) \vee Q(y,a)) \wedge \neg P(y))$$

es una fórmula válida.

Ahora utilizaremos la noción de p-satisfacible para arribar a la misma conclusión. Primero debemos determinar el conjunto $U(S)(S)$:

$$U(S)(S) = \{P(a) \vee \neg Q(a,a), P(a) \vee Q(a,a), \neg P(a)\}.$$

Si llamamos $p_1 = P(a)$, $p_2 = Q(a,a)$, entonces el conjunto $U(S)(S)$ da lugar al siguiente conjunto de cláusulas proposicionales:

$$S' = \{p_1 \vee \neg p_2, p_1 \vee p_2, \neg p_1\}.$$

Este conjunto es insatisfacible, pues si existe una valuación proposicional v tal que $v(S') = 1$, entonces $v(p_1 \vee \neg p_2) = 1$, $v(p_1 \vee p_2) = 1$ y $v(\neg p_1) = 1$. Es decir, $v(p_1) = 0$, lo que implica por la primera igualdad que $v(p_2) = 0$, pero por la segunda igualdad $v(p_1 \vee p_2) = 0$, lo que es un absurdo. Por lo tanto S' es insatisfacible.

Ejemplo 5.34 Probemos, aplicando el Corolario 5.31, que el conjunto de cláusulas

$$S = \{\neg R(x,y) \vee \neg R(y,x), R(f(x), f(y))\}$$

es insatisfacible. Consideremos el universo y la base de Herbrand de S :

$U(S) = \{a, f(a), f(f(a)), ...\}$

$B(S) = \{R(a,a), R(f(a), f(a)), R(f(f(a)), f(f(a))), ..., R(a, f(a)), R(f(a), a), ...\}.$

Consideremos también $U(S)(S)$

$$U(S)(S) = \{\neg R(a,a), R(f(a), f(a)), R(a, f(a)) \vee \neg R(f(a), a), ...\}.$$

Observemos que el subconjunto finito de $U(S)(S)$

$$S_0 = \{\neg R(f(a), f(a)), R(f(a), f(a))\}$$

es insatisfacible. Por lo tanto, aplicando el Corolario 5.31, podemos asegurar que S *es insatisfacible.*

Ejemplo 5.35 Probar que el siguiente conjunto de cláusulas es insatisfacible.

$$S = \{P(x), \neg P(x) \vee Q(f(x)), \neg Q(f(a))\}.$$

Determinemos el universo y la base de Herbrand de S :

$$\begin{aligned} U(S) &= \{a, f(a), f(f(a)), ...\} \\ B(S) &= \{P(a), P(f(a)), Q(a), Q(f(a)), P(f(f(a))), ...\} \end{aligned}$$

En este caso como $B(S)$ es infinito no podemos determinar todos los posibles modelos de Herbrand. Por la misma razón tampoco es sencillo aplicar la noción de p-satisfacible, pero si podemos emplear el siguiente razonamiento:

Supongamos que $\mathcal{M}$ es un modelo de S, es decir

$$\mathcal{M} \vDash S.$$

Entonces

$$\mathcal{M} \vDash P(x) \wedge (\neg P(x) \vee Q(f(x))) \wedge \neg Q(f(a)).$$

Pero, como $\mathcal{M} \vDash \neg Q(f(a))$, entonces $\mathcal{M} \nvDash Q(f(a))$. Es decir, $\mathcal{M} \nvDash Q(f(x))$.

Además, como $\mathcal{M} \vDash \neg P(x) \vee Q(f(x))$, entonces $\mathcal{M} \vDash \neg P(x)$, o lo que es lo mismo $\mathcal{M} \nvDash P(x)$.

En conclusión, cualquier modelo que satisface $(\neg P(x) \vee Q(f(x))) \wedge \neg Q(f(a))$ no satisface $P(x)$. Por lo tanto S es insatisfacible.

También podemos intentar determinar algún subconjunto finito S_0 de $U(S)(S)$ que admita una refutación (es decir, si $S_0 \vdash_R \perp$). Por ejemplo, el conjunto

$$S_0 = \{P(a), \neg P(a) \vee Q(f(a)), \neg Q(f(a))\}$$

admite una refutación y en consecuencia S es insatisfacible.

Capítulo 6

Resolución

En la sección anterior hemos visto que un conjunto finito de cláusulas S es insatisfacible si y sólo si no tiene modelos de Herbrand. En los casos donde la base de Herbrand es un conjunto finito se puede determinar todos los posibles modelos de Herbrand de S. Pero en muchos casos esto no es posible. Ahora vamos a generalizar el método de resolución visto para el cálculo proposicional a la lógica de predicados.

6.1. Unificación

Ahora necesitamos ampliar el método de resolución a cualquier conjunto de cláusulas. En el caso general, el método de resolución incluye un proceso de cámbio de variables como veremos más adelante. Discutamos un ejemplo

Ejemplo 6.1 Consideremos las dos cláusulas siguientes

$$\begin{aligned} C_1 &= \neg P(a, y) \vee Q(x, y) \\ C_2 &= P(x, y) \vee R(x, b). \end{aligned}$$

En este caso no podemos determinar directamente la resolvente de C_1 y C_2 pues los literales $\neg P(a, y)$ y $P(x, y)$ no son exactamente complementarios. Pero si sustituimos la variable x por la constante a en las cláusulas C_1 y C_2 obtenemos las cláusulas

$$\begin{aligned} C_1' &= \neg P(a, y) \vee Q(a, y) \\ C_2' &= P(a, y) \vee R(a, b). \end{aligned}$$

En este caso si podemos determinar la resolvente de C_1' y C_2':

$$\text{Res}\left(C_1, C_2'\right) = Q(a, y) \vee R(a, y).$$

El proceso de sustitución del ejemplo anterior se justifica por el hecho de que las cláusulas son fórmulas cuantificadas universalmente, es decir

$$\begin{aligned} C_1 &= \forall x \forall y \left(\neg P(a, y) \vee Q(x, y)\right) \\ C_2 &= \forall x \forall y \left(P(x, y) \vee R(x, b)\right), \end{aligned}$$

y por lo tanto $C_2' = P(a,y) \vee R(a,b)$ es una instancia de C_2.

Ahora vamos a introducir formalmente la noción de sustitución. ‘ás adelante definiremos el proceso de resolución general en donde el proceso de unificación juega un importante papel.

Definición 6.2 Una ***sustitución*** es un conjunto finito de reemplazos simultaneos de variables por términos. Es decir, una sustitución e es un conjunto finito

$$e = \{x_1/t_1, ..., x_n/t_n\},$$

donde x_i son variables distintas y t_i son términos distintos.

Sea $A = A(x_1, ..., x_n)$ un término, un literal, una cláusula o un conjunto de cláusulas, donde $x_1, ..., x_n$ son todas las variables que ocurren en A. La expresión que resulta despues de aplicar una sustitución $e = \{x_1/t_1, ..., x_n/t_n\}$ a A es:

$$Ae = A(x_1, ..., x_n)\, e = A(x_1/t_1, ..., x_n/t_n).$$

Por ejemplo, si

$$A(x,y) = P(x) \vee Q(x, g(y), a)$$

y

$$e = \{x/f(b), y/b\}$$

entonces

$$Ae = P(f(b)) \vee Q(f(b), g(b), a).$$

Definición 6.3 Sean

$$e_1 = \{x_1/t_1, ..., x_n/t_n\}$$

y

$$e_2 = \{y_1/s_1, ..., y_k/s_k\}$$

dos sustituciones. La ***composición*** de e_1 con e_2 es la sustitución e_1e_2 definida por

$$e_1e_2 = \{x_i/t_ie_2 : x_i \neq t_ie_2\} \cup \{y_j/s_j : y_j \neq x_i,\, i = 1, ..., n\}.$$

Ejemplo 6.4 Consideremos las siguientes sustituciones

$$e_1 = \{x/f(y), y/f(a), z/u\}$$
$$e_2 = \{y/g(a), u/z, v/f(f(a))\}.$$

Determinemos e_1e_2.

$$e_1e_2 = \{x/f(g(a)), y/f(a), u/z, v/f(f(a))\}.$$

En la composición anterior se ha eliminado las sustitución z/z pues es redundante y la sustitución $y/g(a)$ pues la variable y aparece como variable en la primer sustitución.

Definición 6.5 Sea $A = \{l_1, ..., l_n\}$ un conjunto de literales con el mismo símbolo relacional. Una sustitución e se dice un ***unificador*** de A si

$$le = l_2 e = \cdots = l_n e.$$

Si existe un unificador de A entonces decimos que A es unificable.

Un unificador u se dice el *unificador más general (umg)* para A si cualquier otro unificador e *p*uede ser obtenido de u por medio de una composición con una sustitución λ. Es decir: u es el umg para A si y solo si para cualquier otro unificador e existe una sustitución λ tal que $e = u\lambda$.

Ejemplo 6.6 Consideremos las fórmulas atómicas $P(x, y)$ y $P(a, f(z))$. Un unificador más general es la sustitución $e = \{x/a, y/f(z)\}$.

Observación. Si l_1 y l_2 son dos literales unificables, entonces los dos son positivos (es decir son fórmulas atómicas) o los dos son negativos. Remarquemos que los literales l_1 y l_2 deben tener el mismo símbolo relacional. Es decir, para que l_1 y l_2 sean unificables deben ser de la forma:

$$l_1 = P(t_1, ..., t_n) \quad \text{y} \quad l_2 = P(s_1, ..., s_n)$$

o de la forma

$$\neg l_1 = P(t_1, ..., t_n) \quad \text{y} \quad \neg l_2 = P(s_1, ..., s_n)$$

donde $t_1, ..., t_n, s_1, ..., s_n$ son términos.

Ahora estamos interesados en determinar un método para encontrar un unificador más general de un conjunto de fórmulas atómicas. El algorítmo que expondremos se debe a Robinson, y se conoce con el nombre de algorítmo de unificación de Robinson.

Definición 6.7 Sean A y B dos literales considerados como una sucesión de símbolos. Sea k la posición en ámbas secuencias de símbolos comenzando por la izquierda en donde las secuencias difieren en un par de términos $\{t, t'\}$, donde $t \in A$ y $t' \in B$. El par $\{t, t'\}$ se llama el k-par no coincidente.

Ejemplo 6.8 Consideremos las fórmulas atómicas $P(x, g(y), b)$ y $P(a, z, b)$. Entonces $\{x, a\}$ es el 1-par no coincidente y $\{g(y), z\}$ es el 2-par no coincidente.

Algorítmo de Unificación. Sean A y B dos literales

1. Iniciamos el algorítmo con $A_0 = A$ y $B_0 = B$.

2. Supongamos que A_i y B_i han sido construidas. Sea $\{t, t'\}$ un k-par no coincidente de A_i y B_i. Si uno de los términos es una variable x_{i+1} y el otro término es un término t_{i+1} tal que $x_i + 1 \notin Var(t_{i+1})$ entonces definimos

$$e_i = \{x_{i+1}/t_{i+1}\}$$

y calculamos

$$A_{i+1} = A_i e_i$$
$$B_{i+1} = B_i e_i.$$

Si esto no es posible, entonces los literales A y B no son unificables.

Si $A_n = B_n$, entonces los literales son unificables y el unificador más general es

$$u = e_i e_{i+1} \ldots . e_n.$$

Ejemplo 6.9 Determinar si es posible unificar las siguientes fórmulas atómicas.

$$A = P\left(g\left(y\right), f\left(x\right), h\left(x\right), y\right)$$
$$B = P\left(x, f\left(g\left(z\right)\right), w, z\right).$$

El primer par no coincidente es $\{x, g\left(y\right)\}$. Como uno de los términos es una variable y el otro es un término que no contiene la variable x, entonces podemos definir la sustitución:

$$e_1 = \{x/g\left(y\right)\}$$

Luego,

$$A_1 = Ae_1 = P\left(g\left(y\right), f\left(g\left(y\right)\right), h\left(g\left(y\right)\right), y\right)$$
$$B_1 = Be_1 = P\left(g\left(y\right), f\left(g\left(z\right)\right), w, z\right).$$

El segundo par no coincidente es $\{y, z\}$. Entonces definimos la sustitución

$$e_2 = \{y/z\}.$$

En consecuencia obtenemos

$$A_2 = A_1 e_2 = P\left(g\left(z\right), f\left(g\left(z\right)\right), h\left(g\left(z\right)\right), z\right)$$
$$B_2 = B_1 e_2 = P\left(g\left(z\right), f\left(g\left(z\right)\right), w, z\right).$$

El tercer par no coincidente es $\{w, h\left(g\left(z\right)\right)\}$. Entonces definimos la sustitución

$$e_3 = \{w/h\left(g\left(z\right)\right)\}.$$

$$A_3 = A_2 e_3 = P\left(g\left(z\right), f\left(g\left(z\right)\right), h\left(g\left(z\right)\right), z\right)$$
$$B_3 = B_2 e_3 = P\left(g\left(z\right), f\left(g\left(z\right)\right), h\left(g\left(z\right)\right), z\right).$$

Observemos que

$$A_3 = B_3$$

Por lo tanto, la sustitución más general para A y B es

$$u = e_1 e_2 e_3 = \{x/g\left(y\right), y/z, w/h\left(g\left(z\right)\right)\}.$$

Ejemplo 6.10 El algorítmo de Robinson puede ser aplicado a más de dos literales. Por ejemplo, intentemos unificar el conjunto de literales $\{R\left(x, f\left(w\right), f\left(z\right)\right), R\left(f\left(y\right), z, f\left(z\right)\right), R\left(w, u, f\left(u\right)\right)\}$.

$$\left\{\begin{array}{l} R\left(x, f\left(w\right), f\left(z\right)\right) \\ R(f(y), z, f\left(z\right)) \\ R\left(w, u, f\left(u\right)\right) \end{array}\right. \xrightarrow{x/f(y)} \left\{\begin{array}{l} R\left(f\left(y\right), f\left(w\right), f\left(z\right)\right) \\ R\left(f\left(y\right), z, f\left(z\right)\right) \\ R\left(w, u, f\left(u\right)\right) \end{array}\right. \xrightarrow{w/f(y)} \left\{\begin{array}{l} R\left(f(y), f^2(y), f\left(z\right)\right) \\ R\left(f(y), z, f\left(z\right)\right) \\ R\left(f(y), u, f\left(u\right)\right) \end{array}\right. \xrightarrow{z/f^2(y)}$$

$$\left\{\begin{array}{l} R\left(f(y), f^2(y), f^3(y)\right) \\ R\left(f(y), f^2(y), f^3(y)\right) \\ R\left(f(y), u, f\left(u\right)\right) \end{array}\right. \xrightarrow{u/f^2(y)} \left\{\begin{array}{l} R\left(f(y), f^2(y), f^3(y)\right) \\ R\left(f(y), f^2(y), f^3(y)\right) \\ R\left(f(y), f^2(y), f^3(y)\right) \end{array}\right.$$

6.2. Resolución con unificación

Ahora vamos a redefinir resolución incorporando el proceso de unificación como parte de la regla.

Definición 6.11 Sean C_1 y C_2 dos cláusulas.

1. Supongamos que existen sustituciones $\theta_1 : Var(C_1) \rightarrow Var$ y $\theta_2 : Var(C_2) \rightarrow Var$ tal que las cláusulas $C_1\theta_1 = C_1' \cup \{l_1, ..., l_n\}$ y $C_2\theta_2 = C_2' \cup \{\neg l_1', ..., \neg l_m'\}$ no tienen variables en común.

2. Supongamos que los conjuntos de literales
$$L_1\theta_1 = \{l_1, ..., l_n\} \subseteq C_1\theta_1$$
y
$$L_2\theta_2 = \left\{\neg l_1', ..., \neg l_m'\right\} \subseteq C_2\theta_2$$
son tales que el conjunto
$$L_1\theta_1 \cup \neg L_2\theta_2 = \left\{l_1, ..., l_n, l_1', ..., l_m'\right\}$$
es unificable por un unificador más general u, es decir:
$$l_1 u = l_2 u = \cdots = l_n u = l_1' u = \cdots = l_m' u = l.$$
Entonces la *resolvente* de C_1 y C_2 es la nueva cláusula
$$\mathrm{Res}(C_1, C_2) = (C_1\theta_1 u - \{l\}) \cup (C_2\theta_2 u - \{l^c\}),$$
donde $L_1\theta_2 u = \{l\}$ *y* $L_2\theta_2 u = \{l^c\}$.

En la definición anterior se pide que existan sustituciones θ_1 y θ_2 tal que $C_1\theta_1$ y $C_2\theta_2$ no tengan variables en común para que no se determinen sustituciones inconsistentes. Recordemos que las cláusulas son implicitamente cuantificadas y ántes de comenzar a trabajar se pueden renombrar las variables. Esta condición es necesaria, pues por ejemplo, el conjunto de cláusulas $\{C_1 = P(x), C_2 = \neg P(f(x))\}$ es insatisfacible (y refutable) pero estas cláusulas no pueden ser unificadas sin primero renombrar las variables.

Ejemplo 6.12 Determinar la resolvente de

$$\begin{aligned} C_1 &= P(f(x)) \vee \neg Q(z) \vee P(z) \\ C_2 &= \neg P(x) \vee R(g(x), a) \end{aligned}$$

El primer paso es renombrar las variables que ocurren en las cláusulas de tal manera que tengan variables diferentes. En este caso podemos dejar fijas las variables de C_1 y cambiar las variables de C_2 haciendo la sustitución $\theta_2 = (x/w)$. Dejar fijas las variables de C_1 es lo mismo que aplicar la sustitución $\theta_1 = (\ \ )$, es decir la sustitución que no cámbia ninguna variable. Luego:

$$\begin{aligned} C_1 &= P(f(x)) \vee \neg Q(z) \vee P(z) \\ C_2\theta_2 &= \neg P(w) \vee R(g(w), a) \end{aligned}$$

Ahora determinaremos el unificador más general del conjunto de literales $\{P(f(x)), P(z), P(w)\}$. Tomando las dos primeras fórmulas atómicas $P(f(x)), P(z)$, determinamos el unificador $u_1 = (z/f(x))$ y cuyo resultado es $P(f(x))u_1 = P(z)u_1 = P(f(x))$. Ahora con esta fórmula y la fórmula $P(w)$ determinamos un unificador $u_2 = (w/f(x))$. En consecuencia, el unificador más general es $u = u_1u_2 = \{z/f(x), w/f(x)\}$. Aplicando esta sustitución a C_1 y $C_2\theta_2$ obtenemos

$$\begin{aligned} C_1u &= P(f(x)) \vee \neg Q(f(x)) \\ C_2\theta_2u &= \neg P(f(x)) \vee R(g(f(x), a)) \end{aligned}$$

Por lo tanto, la resolvente de las cláusulas es

$$\mathrm{Res}(C_1, C_2) = \neg Q(f(x)) \vee R(g(f(x), a)).$$

Ejemplo 6.13 Determinar la resolvente de

$$\begin{aligned} C_1 &= P(f(x), g(y)) \vee Q(x, y) \\ C_2 &= \neg P(f(f(a)), g(y)) \vee Q(f(a), g(y)). \end{aligned}$$

Primero observemos que las cláusulas C_1 y C_2 tienen la variable y en común. Luego renombramos las variables de C_2 cambiando la variable y por z. Entonces resulta

$$\begin{aligned} C_1 &= P(f(x), g(y)) \vee Q(x, y) \\ C_2 &= \neg P(f(f(a)), g(z)) \vee Q(f(a), g(z)). \end{aligned}$$

Determinemos el unificador más general para las fórmulas atómicas $P(f(x), g(y))$ y $P(f(f(a)), g(z))$. Entonces*:*

$$\{x, f(a)\}$$
$$e_1 = \{x/f(a)\}$$

$$\begin{aligned} P(f(x), g(y))e_1 &= P(f(f(a)), g(y)) \\ P(f(f(a)), g(z))e_1 &= P(f(f(a)), g(z)). \end{aligned}$$

$$\{y, z\}$$
$$e_2 = \{y/z\}$$

$$\begin{aligned} P(f(f(a)), g(y))e_2 &= P(f(f(a)), g(z)) \\ P(f(f(a)), g(z))e_2 &= P(f(f(a)), g(z)). \end{aligned}$$

Por lo tanto el unificador más general es

$$u = \{x/f(a), y/z\}.$$

Aplicando u a las cláusulas obtenemos

$$\begin{aligned} C_1u &= P(f(f(a)), g(z)) \vee Q(f(a), z) \\ C_2u &= \neg P(f(f(a)), g(z)) \vee Q(f(a), g(z)), \end{aligned}$$

y resolviendo obtenemos

$$\mathrm{Res}(C_1, C_2) = Q(f(a), z) \vee Q(f(a), g(z)).$$

La noción de deducción por resolución en cálculo de predicados es similar al caso proposicional.

Definición 6.14 Sea S un conjunto de cláusulas y C una cláusula. Una ***deducción por resolución*** de C a partir de S, en símbolos $S \vdash_R C$, es una sucesión finita de cláusulas

$$C_1, C_2, ..., C_n = C$$

tales que

1. $C_i \in S$, o
2. Existen cláusulas C_j, C_k, con $j, k < i < n$ tal que $\mathrm{Res}\,(C_j, C_k) = C_i$.

Una resolución de la cláusula $\perp$ de un conjunto de cláusulas S se dice una ***refutación*** de S.

Para probar que el método de resolución es correcto, es decir para probar que si $S \cup \{C\}$ es un conjunto de cláusulas tal que $S \vdash_R C$, entonces $S \vDash C$, probaremos primero el siguiente resultado auxiliar.

Lema 6.15 *Sean C_1 y C_2 dos cláusulas y supongamos que C es la resolvente de C_1 y C_2. Si C_1 y C_2 son satisfacibles, entonces C es satisfacible.*

Demostración. Supongamos que existe un modelo $\mathcal{M}$ tal que $\mathcal{M} \vDash C_1$ y $\mathcal{M} \vDash C_2$. Supongamos que

$$C_1 = C_1' \cup \{l_1, ..., l_n\} = C_1' \cup L_1 \text{ y } C_2 = C_2' \cup \left\{\neg l_1', ..., \neg l_m'\right\} = C_2' \cup L_2.$$

Sean θ_1 y θ_2 las sustituciones tales que $Var\,(C_1\theta_1) \cap Var\,(C_2\theta_2) = \emptyset$, y sea u el unificador más general tal que $L_1\theta_1 u = l$ y $L_2\theta_2 u = l^c$. Entonces,

$$C = \mathrm{Res}\,(C_1, C_2) = (C_1\theta_1 u - \{l\}) \cup (C_2\theta_2 u - \{l^c\}) = C_1'\theta_1 u \cup C_2'\theta_2 u.$$

Como $\mathcal{M} \vDash C_1$ y $\mathcal{M} \vDash C_2$, entonces $\mathcal{M} \vDash C_1'\theta_1 u \cup \{l\}$ y $\mathcal{M} \vDash C_2'\theta_2 u \cup \{l^c\}$ (pues recordemos que las cláusulas C_1 y C_2 son fórmulas cuantificadas universalmente). Si $\mathcal{M} \vDash l$, entonces $\mathcal{M} \nvDash l^c$. En consecuencia, $\mathcal{M} \vDash C_2'\theta_2 u$, y por lo tanto $\mathcal{M} \vDash C_1'\theta_1 u \cup C_2'\theta_2 u$. El caso $\mathcal{M} \vDash l^c$ se analiza en forma similar.■

Teorema 6.16 (de Corrección) *Sea $S \cup \{C\}$ un conjunto de cláusulas. Si $S \vdash_R C$, entonces $S \vDash C$.*

Demostración. La prueba se hace por inducción sobre la longitud de la demostración por resolución de C a partir S aplicando el lema anterior. La prueba es similar al caso proposicional y queda como ejercicio. ■

6.3. Completitud

Sea S un conjunto de cláusulas y sea $U(S)$ el universo de Herbrand de S. Recordemos que para cada subconjunto $P \subseteq U(S)$, el conjunto $P(S)$ es el conjunto de todas las *cláusulas cerradas* que se obtienen reemplazando las variables que ocurren en S por términos del subconjunto $P \subseteq U(S)$, es decir por elementos del universo de Herbrand de S.

La prueba del teorema de completitud en el caso general se basa en el siguiente hecho: si S es un conjunto de cláusulas y $U(S)$ es su universo de Herbrand, entonces si existe algún conjunto de instancias cerradas de S (es decir, existe $P(S)$ para algún $P \subseteq U(S)$) tal que $P(S) \vdash_R \perp$, entonces $S \vdash_R \perp$. Es decir, para determinar si $S \vdash_R \perp$ es suficiente hacerlo para un conjunto de instancias cerradas de cláusulas de S.

Consideremos un conjunto de cláusulas S. Vamos a extender la notación $\mathcal{R}(S)$ utilizada en el caso proposicional de la siguiente forma:

$$
\begin{aligned}
\mathcal{R}(S) &= S \cup \{C : C = \operatorname{Res}(C_1, C_2), \text{ para un par de cláusulas } C_1, C_2 \in S\} \\
\mathcal{R}^0(S) &= S \\
\mathcal{R}^{n+1}(S) &= \mathcal{R}(\mathcal{R}^n(S)) \\
\mathcal{R}^*(S) &= \bigcup_{n \geq 0} \mathcal{R}^n(S)
\end{aligned}
$$

Dado un conjunto de cláusulas $S \cup \{C\}$, es inmediato comprobar que:

$$S \vdash_R C \Leftrightarrow C \in \mathcal{R}^*(S) \Leftrightarrow \text{existe un } n \geq 0 \text{ tal que } C \in \mathcal{R}^n(S).$$

El siguiente resultado, conocido como *Lifting Lema*, se utiliza para transformar una refutación de cláusulas proposicionales en una refutación de cláusulas en lógica de predicados.

Lema 6.17 (Lifting Lema) *Sea S un conjunto de cláusulas y sea $U(S)$ el universo de Herbrand de S. Si $P \subseteq U(S)$, entonces*

$$\mathcal{R}(P(S)) \subseteq P(\mathcal{R}(S)).$$

Demostración. Sea $C' \in \mathcal{R}(P(S)) = P(S) \cup \operatorname{Res}(P(S))$. Si $C' \in P(S)$, entonces como $S \subseteq \mathcal{R}(S) = S \cup \operatorname{Res}(S)$, entonces $P(S) \subseteq P(\mathcal{R}(S))$. Por lo tanto, $C' \in P(\mathcal{R}(S))$. Supongamos que $C' \notin P(S)$. Entonces $C' \in \operatorname{Res}(P(S))$. Es decir, existen dos cláusulas $C_1, C_2 \in S$, sustituciones cerradas $\alpha_1 : Var(C_1) \to U(S)$ y $\alpha_2 : Var(C_2) \to U(S)$, y literales $l \in C_1\alpha_1$ y $l^c \in C_2\alpha_2$ tal que

$$C' = \operatorname{Res}(C_1\alpha_1, C_2\alpha_2) = (C_1\alpha_1 - \{l\} \cup C_2\alpha_2 - \{l^c\}).$$

Observemos que el literal l es una instancia de literales $L_1 = \{l_1, ..., l_n\} \subseteq C_1$ y l^c es una instancia de literales $L_2 = \{\neg l'_1, ..., \neg l'_m\}$. Es decir, $L_1\alpha_1 = l$ y $L_2\alpha_2 = l^c$. Las sustituciones α_1 y α_2 están dadas por

$$
\begin{aligned}
\alpha_1 &= (x_1/s_1, ..., x_k/s_k) \qquad \text{donde } Var(C_1) = \{x_1, ..., x_k\} \\
\alpha_2 &= (y_1/t_1, ..., y_m/t_m) \qquad \text{donde } Var(C_2) = \{y_1, ..., y_m\}.
\end{aligned}
$$

Consideremos las sustituciones $\theta_1 : Var(C_1) \to Var$ y $\theta_2 : Var(C_2) \to Var$ tal que hacen que las cláusulas $C_1\theta_1$ y $C_2\theta_2$ no tengan variables en común. Las sustituciones θ_1 y θ_2 están definidas como

$$\begin{aligned}\theta_1 &= (x_1/u_1, ..., x_k/u_k)\\ \theta_2 &= (y_1/v_1, ..., y_m/v_m).\end{aligned}$$

Consideremos también la sustitución

$$u = (u_1/s_1, ..., u_k/s_k, v_1/t_1, ..., v_m/t_m).$$

Observemos que

$$\begin{aligned}C_1\alpha_1 &= C_1\theta_1 u\\ C_2\alpha_2 &= C_2\theta_2 u\end{aligned}$$

$$\begin{aligned}l &= L_1\alpha_1 &&= L_1\theta_1 u\\ l^c &= L_1\alpha_2 &&= L_2\theta_2 u.\end{aligned}$$

Como la sustitución u es un unificador para el conjunto de literales

$$L\theta_1 \cup \neg L_2\theta_2,$$

donde $\neg L_2\theta_2$ es el conjunto de literales negados de $L_2\theta_2$, entonces por el Algorítmo de Robinson podemos asegurar la existencia de un unificados más general u_0, y una resolvente C de C_1 y C_2 donde

$$C = \mathrm{Res}(C_1, C_2) = (C_1\theta_1 - \{l_1, ..., l_n\})\, u_0 \cup \left(C_2\theta_2 - \left\{\neg l'_1, ..., \neg l'_m\right\}\right) u_0.$$

Como u_0 es un unificador más general, entonces existe una sustitución λ tal que

$$u = u_0.\lambda$$

Veamos que $C' = C\lambda$:

$$\begin{aligned}C' &= \mathrm{Res}(C_1\alpha_1, C_2\alpha_2) &&= (C_1\alpha_1 - \{l\}) \cup (C_2\alpha_2 - \{l^c\})\\ &= (C_1\alpha_1 - L_1\alpha_1) \cup (C_2\alpha_2 - L_2\alpha_2) &&= (C_1\theta_1 u - L_1\theta_1 u) \cup (C_2\theta_2 u - L_2\theta_2 u)\\ &= [(C_1\theta_1 - L_1\theta_1) \cup (C_2\theta_2 - L_2\theta_2)]\, u &&= [(C_1\theta_1 - L_1\theta_1) \cup (C_2\theta_2 - L_2\theta_2)]\, u_0\lambda\\ &= ([(C_1\theta_1 - L_1\theta_1) \cup (C_2\theta_2 - L_2\theta_2)]\, u_0)\, \lambda &&= C\lambda\end{aligned}$$

Por lo tanto $C \in P(\mathcal{R}(S))$. ■

Ejemplo 6.18 Analicemos con un ejemplo los pasos del Lema anterior. Consideremos las siguientes cláusulas

$$\begin{aligned}C_1 &= P(x) \vee P(f(y)) \vee P(f(z)) \vee Q(x)\\ C_2 &= \neg P(f(u)) \vee \neg P(w) \vee R(u)\end{aligned}$$

y las instancias cerradas de C_1 y C_2

$$\begin{aligned}C'_1 &= P(f(a)) \vee Q(f(a))\\ C'_2 &= \neg P(f(a)) \vee R(a).\end{aligned}$$

La resolvente de C_1' y C_2' es

$$C' = \operatorname{Res}\left(C_1', C_2'\right) = Q\left(f\left(a\right)\right) \vee R\left(a\right).$$

Según el Lemma anterior, la cláusula C' debe ser una instancia cerrada de una cláusula C tal que $C = \operatorname{Res}\left(C_1, C_2\right)$. Veamos como obtenemos la cláusula C.

Primero observemos que C_1 y C_2 tienen variables distintas. Por lo tanto $\theta_1 = \theta_2 = (\ )$. Notemos que las sustituciones que α_1 y α_2 que trasforman C_1 y C_2 en C_1' y C_2', respectivamente, están dadas por:

$$\begin{array}{rcl} \alpha_1 & = & \left(x/f\left(a\right), y/a, z/a\right) \\ \alpha_2 & = & \left(w/f\left(a\right), u/a\right) \end{array}$$

Observemos también que

$$\begin{array}{rcl} L_1 & = & \left\{P\left(x\right), P\left(f\left(y\right), P\left(f\left(z\right)\right)\right)\right\} \\ L_2 & = & \left\{\neg P\left(f\left(u\right)\right), \neg P\left(w\right)\right\}. \end{array}$$

Como $\theta_1 = \theta_2 = (\ )$, entonces

$$u = \alpha_1\alpha_2 = \left(x/f\left(a\right), y/f\left(a\right), z/a, w/f\left(a\right), u/a\right).$$

Luego

$$\begin{array}{rcl} L_1\alpha_1 & = & L_1u \\ L_2\alpha_2 & = & L_2u \\ & & \\ L_1u & = & \neg L_2u. \end{array}$$

Como u es un unificador de $L_1 \cup \neg L_2$, entonces por el Algorítmo de Robinson podemos determinar un unificador más general u_0. Es sencillo comprobar que u_0 está definido por:

$$u_0 = \left(x/f\left(y\right), z/y, w/f\left(y\right), u/y\right),$$

y que la sustitución

$$\lambda = \left(y/a, z/a\right)$$

es tal que

$$u = u_0\lambda.$$

Luego, la cláusula C buscada es la cláusula definida por

$$\begin{array}{rcl} C & = & \operatorname{Res}\left(C_1, C_2\right) \qquad = \left(C_1u_0 - L_1u_0\right) \cup \left(C_2u_0 - L_2u_0\right) \\ & = & Q\left(f\left(y\right)\right) \vee R\left(y\right). \end{array}$$

El siguiente Lema es una generalización del lema anterior.

Lema 6.19 *Para todo conjunto de cláusulas S*

$$\mathcal{R}^n\left(P\left(S\right)\right) \subseteq P\left(\mathcal{R}^n\left(S\right)\right).$$

Por lo tanto, $\mathcal{R}^*\left(P\left(S\right)\right) \subseteq P\left(\mathcal{R}^*\left(S\right)\right)$.

Demostración. La prueba es por inducción sobre n.

El caso $n=1$ es el Lema anterior. Suponemos que vale para n y probemos que vale para $n+1$:

$$\begin{array}{rclcl} \mathcal{R}^{n+1}\left(P\left(S\right)\right) & = & \mathcal{R}\left(\mathcal{R}^{n}\left(P\left(S\right)\right)\right) & & \\ & \overset{HI}{\subseteq} & \mathcal{R}\left(P\left(\mathcal{R}^{n}\left(S\right)\right)\right) & & \\ & \overset{\text{Lema anterior}}{\subseteq} & P\left(\mathcal{R}\left(\mathcal{R}^{n}\left(S\right)\right)\right) & = & P\left(\mathcal{R}^{n+1}\left(S\right)\right). \end{array}$$

■

Teorema 6.20 *Sea S un conjunto de cláusulas. Entonces $S\vdash_R \bot$ si y solo si S es insatisfacible.*

Demostración. Ya sabemos que si $S\vDash C$ entonces $S\vdash_R C$.

Supongamos ahora que S es insatisfacible. Por el Corolario 5.31, existe un subconjunto $P\subseteq U\left(S\right)$ tal que $P\left(S\right)$ es p-insatisfacible. Por el Teorema de Completitud para el caso proposicional $P\left(S\right)\vdash_R \bot$. Es decir, $\bot\in\mathcal{R}^{n}\left(P\left(S\right)\right)$. Del Lema anterior deducimos que $\bot\in P\left(\mathcal{R}^{n}\left(S\right)\right)$, para algún $n\geq 0$. Por lo tanto $S\vdash_R \bot$. ■

Resumimos los resultados estudiados hasta ahora en el siguiente corolario.

Corolario 6.21 *Sea S un conjunto de cláusulas. Entonces las siguientes condiciones son equivalentes:*

1. *S es insatisfacible.*
2. *S no tiene modelos de Herbrand.*
3. *$U\left(S\right)\left(S\right)$ es p-insatisfacible.*
4. *Existe un subconjunto finito $S_0\subseteq U\left(S\right)\left(S\right)$ p-insatisfacible.*
5. *Existe un subconjunto finito $S_0\subseteq U\left(S\right)\left(S\right)$ tal que $S_0\vdash_R \bot$.*
6. *$S\vdash_R \bot$.*

Vamos a finalizar el capítulo con un ejemplo.

Ejemplo 6.22 Probar que el siguiente conjunto de cláusulas es insatisfacible.

$$S=\left\{\begin{array}{l}\neg P\left(x\right)\vee Q\left(x\right)\vee R\left(x,f\left(x\right)\right),\neg P\left(x\right)\vee Q\left(x\right)\vee S\left(f\left(x\right)\right),T\left(a\right),P\left(a\right),\\ \neg R\left(a,z\right)\vee T\left(z\right),\neg T\left(x\right)\vee\neg Q\left(x\right),\neg T\left(y\right)\vee\neg S\left(y\right)\end{array}\right\}.$$

Determinemos una refutación de S. En cada paso escribiremos cual es la sustitución empleada.

1. $C_8=\mathrm{Res}\left(C_3,C_6\right)=\mathrm{Res}\left(T\left(a\right),\neg T\left(x\right)\vee\neg Q\left(x\right)\right)=\neg Q\left(a\right)$, $e_1=\left(x/a\right)$.
2. $C_9=\mathrm{Res}\left(C_2,C_4\right)=\mathrm{Res}\left(\neg P\left(x\right)\vee Q\left(x\right)\vee S\left(f\left(x\right)\right),P\left(a\right)\right)=Q\left(a\right)\vee S\left(f\left(a\right)\right)$ $e_2=\left(x/a\right)$
3. $C_{10}=\mathrm{Res}\left(C_8,C_9\right)=\mathrm{Res}\left(\neg Q\left(a\right),Q\left(a\right)\vee S\left(f\left(a\right)\right)\right)=S\left(f\left(a\right)\right).$

4. $C_{11} = \text{Res}\left(C_1, C_4\right) = \text{Res}\left(\neg P\left(x\right) \vee Q\left(x\right) \vee R\left(x, f\left(x\right)\right), P\left(a\right)\right) = Q\left(a\right) \vee R\left(a, f\left(a\right)\right)$, $e = \left(x/a\right).$
5. $C_{12} = \text{Res}\left(C_8, C_{11}\right) = \text{Res}\left(\neg Q\left(a\right), Q\left(a\right) \vee R\left(a, f\left(a\right)\right)\right) = R\left(a, f\left(a\right)\right).$
6. $C_{13} = \text{Res}\left(C_5, C_{12}\right) = \text{Res}\left(\neg R\left(a, z\right) \vee T\left(z\right), R\left(a, f\left(a\right)\right)\right) = T\left(f\left(a\right)\right)$, $\quad e = \left(z/f\left(a\right)\right).$
7. $C_{14} = \text{Res}\left(C_7, C_{13}\right) = \text{Res}\left(\neg T\left(y\right) \vee \neg S\left(y\right), T\left(f\left(a\right)\right)\right) = \neg S\left(f\left(a\right)\right)$, $\quad e = \left(y/f\left(a\right)\right).$
8. $C_{15} = \text{Res}\left(C_{10}, C_{14}\right) = \text{Res}\left(S\left(f\left(a\right)\right), \neg S\left(f\left(a\right)\right)\right) = \perp.$

Por lo tanto, $S \vdash_R \perp$, y en consecuencia S es insatisfacible.

También se puede demostrar que S es insatisfacible utilizando la noción de p-satisfacible. Consideremos el conjunto de cláusulas cerradas que se obtiene a partir de S haciendo la sustitución $e = \left(x/a, z/f\left(a\right), y/f\left(a\right)\right):$

$$S_0 = Se = \left\{ \begin{array}{l} \neg P\left(a\right) \vee Q\left(a\right) \vee R\left(a, f\left(a\right)\right), \neg P\left(a\right) \vee Q\left(a\right) \vee S\left(f\left(a\right)\right), T\left(a\right), P\left(a\right), \\ \neg R\left(a, f\left(a\right)\right) \vee T\left(f\left(a\right)\right), \neg T\left(a\right) \vee \neg Q\left(a\right), \neg T\left(f(a)\right) \vee \neg S\left(f(a)\right) \end{array} \right\}$$

Es sencillo comprobar que S_0 es p-insatisfacible. También se puede llegar al mismo resultado probando que $S_0 \vdash_R \perp$.

Capítulo 7

Refinamientos de Resolución

Uno de los avances fundamentales del método de resolución es que solo se aplica una regla. A pesar de que el método de resolución es muy poderoso existen problemas de tipo combinatorio. En general, cuando intentamos aplicar resolución a un conjunto de cláusulas existen muchas maneras de seleccionar dos cláusulas para producir una resolvente. En un conjunto finito pero con un gran número de cláusulas las posibilidades de elección hacen que el método de resolución sea difícil de implementar. Para solucionar este problema se pueden implementar refinamientos o modificaciones del método original que limiten la busqueda. Existen varios tipos de refinamientos, dependiendo del problema a atacar. Algunos se aplican a cualquier conjunto de cláusulas, como la resolución lineal, y otros se aplican a determinados conjuntos de cláusulas, como la resolución unitaria que solo se aplica a las cláusulas de Horn.

Los métodos de resolución que veremos a continuación son reformulaciones del método original y están basados en dos ideas: *estrategias y restricciones* de resolución. Las estrategias son reglas que nos dicen como buscar las cláusulas a resolver. Es decir, nos dicen, aplicando una determinada regla, como buscar las cláusulas a resolver. Un ejemplo de estrategia es la resolución unitaria. En este alguna de las posibles cláusulas a resolver debe ser unitaria (con un solo literal). Obviamente esto no siempre es posible. Las restricciones se refieren a restricciones en la elección de las cláusulas a resolver, dependiendo de su forma sintáctica. En este caso el número de posibles elecciones es menor que el número total de elecciones sin restricción.

Observación. Para cada refinamiento que estudiemos deberemos probar un Teorema de Correctitud y un Teorema de Completitud. Pero, como todos los refinamientos son casos particulares del método de resolución estudiado anteriormente, el Teorema de Correctitud es inmediato en cada caso. Para probar el correspondiente Teorema de Completitud utilizaremos el Lema de Lifting. Recordemos que dicho Lema nos permite trasladar una refutación de un conunto de cláusulas proposicionales en una refutación de cláusulas en lógica de predicados. Por lo tanto, en cada caso solo es necesario probar el Teorema de Completitud para el caso proposicional. Utilizando la completitud proposicional y el Lema de Lifting obtenemos el Teorema de Completitud para el caso general.

Comenzaremos dando algunas definiciones y resultados generales que luego se aplicarán a los distintos refinamientos que expondremos.

La siguiente definición introduce un tipo particular de subconjuntos de cláusulas construidos a partir de un conjunto de cláusulas y un literal que permite analizar si el conjunto de cláusulas es satisfacible.

Sea S un conjunto de cláusulas y l un literal de S. definimos el conjunto

$$S^l = \{C - \{l^c\} : l \notin C\}.$$

Es decir, el conjunto S^l es el conjunto de todas las cláusulas C de S que no contienen a los literales l y l^c y a todas las cláusulas de la forma $C \cup \{l^c\} \in S$ donde $l \notin C$. Si la cláusula unitaria $\{l^c\} \in S$, entonces $\perp \in S^l$.

Lema 7.1 *Sea S un conjunto de cláusulas. Entonces S es satisfacible si y solo si S^l o S^{l^c} es satisfacible.*

Demostración. $\Rightarrow$) Supngamos que S es satisfacible. Entonces existe una valuación v tal que $v(S) = 1$. Supongamos que $v(l^c) = 0$, es decir, $v(l) = 1$. Probemos que $v\left(S^l\right) = 1$. Consideremos una cláusula arbitraria $C \in S^l$. Por lo definición de las cláusulas de S^l, $C \cup \{l^c\} \in S$ o $C \in S$. Entonces $v(C \cup \{l^c\}) = 1$ o $v(C) = 1$. Como $v(l^c) = 0$, entonces debe ocurrir que $v(C) = 1$.

Si $v(l^c) = 1$, entonces se prueba de forma similar que $v\left(S^{l^c}\right) = 1$.

$\Leftarrow$) Supongamos que S^l es satisfacible por una valuación v. Debemos determinar una valuación w tal que $w(S) = 1$. Consideremos la valuación w definida por

$$w(k) = \begin{cases} v(k) & \text{si} \quad k \neq l, l^c \\ \\ 1 & \text{si} \quad k = l. \end{cases}$$

Como en toda cláusula $C \in S^l, l, l^c \notin C$, entonces $w\left(S^l\right) = v\left(S^l\right) = 1$. Comprobemos que $w(S) = 1$. Sea $C \in S$. Si $l \in C$, entonces $w(C) = 1$. Si $l \notin C$, entonces $C - \{l^c\} \in S^l$, y en consecuencia $v(C - \{l^c\}) = 1$. Luego existe un literal $k \neq l^c$ tal que $k \in C$ y $v(k) = 1$. Por la definición de w tenemos que $v(k) = w(k)$ y por lo tanto $w(C) = 1$. Esto implica que $w(S) = 1$. ■

Corolario 7.2 *Sea S un conjunto de cláusulas. Entonces S es insatisfacible si y solo si S^l y S^{l^c} son insatisfacibles.*

7.1. Resolución Lineal

Ahora estudiaremos uno de los refinamientos del método de resolución llamado resolución lineal.

Definición 7.3 Sea S un conjunto de cláusulas insatisfacible. Un subconjunto $U \subseteq S$ es un conjunto ***soporte*** de S, si $S - U$ es un conjunto satisfacible.

Que un subconjunto U sea un conjunto soporte de un conjunto de cláusulas insatisfacible S significa que podemos aislar la causa de insatisfacibilidad de S.

Ejemplo 7.4 Consideremos el siguiente conjunto de cláusulas insatisfacible

$$S = \{P(x) \vee \neg Q(x, a), P(a) \vee Q(y, a), \neg P(y)\}.$$

Un conjunto soporte de S es, por ejemplo, $U = \{P(x) \vee \neg Q(x, a)\}$.

Definición 7.5 Un conjunto de cláusulas S es un ***conjunto insatisfacible minimal*** si es insatisfacible pero todo subconjunto propio de S es satisfacible. En decir, S es insatisfacible minimal si y solo si para toda cláusula $C \in S$ el conjunto $U = \{C\}$ es un soporte de S.

Lema 7.6 *Sea S un conjunto insatisfacible. Entonces existe un subconjunto insatisfacible minimal S' de S. Además, si U es un conjunto soporte de S, entonces $U \cap S'$ es un conjunto soporte de S'.*

Demostración. Sea S insatisfacible. Por el Teorema de Compacidad, existe algún subconjunto finito de S insatisfacible. De todos los posibles subconjuntos finitos insatisfacibles elegimos el que tenga el menor número de cláusulas. Luego, si S' es tal subconjunto entonces es claro que es minimal.

Sea U un conjunto soporte de S. Veamos primero que $S' \cap U \neq \emptyset$. Si suponemos que $S' \cap U = \emptyset$, entonces $S' \subset S - U$ y como $S - U$ es un conjunto satisfacible, entonces S' es satisfacible, lo que es una contradicción. Por lo tanto $S' \cap U \neq \emptyset$. El conjunto $S' - (S' \cap U)$ es un subconjunto propio de S', y como S' es insatisfacible minimal, entonces $S' - (S' \cap U)$ es satisfacible, es decir, $S' \cap U$ es un conjunto soporte de S'. ■

Definición 7.7 Sea S un conjunto de cláusulas y C una cláusula. Una deducción por ***resolución lineal*** de la cláusula C a partir del conjunto S, en símbolos $S \vdash_l C$, es una secuencia finita de pares ordenados de cláusulas de la forma:

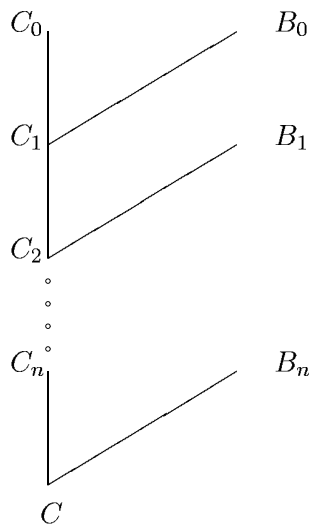
$$(C_0, B_0), (C_1, B_1), ..., (C_n, B_n)$$

tal que

1. $C = C_{n+1}$
2. C_0 y B_i son cláusulas de S o algún C_j con $j < i$,
3. Cada C_{i+1}, $i \leq n$, es una resolvente de C_i y B_i.

Como es usual, diremos C es linealmente deducible de S si existe una resolución lineal $S \vdash_l C$. Si $C = \perp$, diremos que existe una *refutación lineal* de S.

Una descripción gráfica de una deducción por resolución lineal se dá en la siguiente figura:



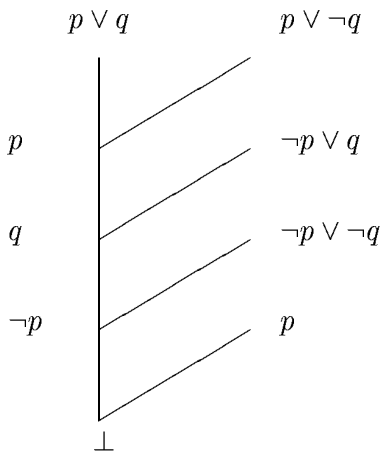
Los elementos de S son llamados *cláusulas de entrada (o input cláusulas)*, las cláusulas C_i se denominan cláusulas del *centro*, y las cláusulas B_i se llaman cláusulas *marginales.*

Es claro que una resolución lineal es un caso particular del método general de resolución.

Ejemplo 7.8 Dado el conjunto de cláusulas

$$S = \{p \vee q, p \vee \neg q, \neg p \vee q, \neg p \vee \neg q\}$$

la siguiente figura corresponde una refutación lineal de S se dá en la siguiente figura



Observemos que la cláusula $C = \{p\}$ es utilizada en el último paso.

El próximo paso es probar que el método de resolución lineal es completo. La prueba de correctitud, es decir, la prueba que si S es un conjunto de cláusulas y C una cláusula tal que $S \vdash_l C$ implica que

$S \vDash C$ es inmediata, pues la resolución lineal es un caso particular del método general de resolución.

Definición 7.9 Sea S un conjunto de cláusulas y U un conjunto soporte de S. Diremos que existe una resolución lineal de C a partir de S con soporte U si $C_0 \in U$.

Teorema 7.10 *Sea S un conjunto de cláusulas insatisfacible con soporte U. Entonces existe una refutación lineal de S con soporte U.*

Demostración (optativa). Sea S un conjunto de cláusulas insatisfacible. Por el Teorema de Compacidad podemos suponer que S es un conjunto insatisfacible finito. Consideremos un conjunto insatisfacible minimal S'. En consecuencia todo subconjunto propio de S' es satisfacible. En particular, para todo conjunto unitario $\{C\}$, con $C \in S'$, el conjunto $S' - \{C\}$ es satisfacible. Es decir, todo conjunto unitario $\{C\}$, con $C \in S'$, es un conjunto soporte de S'. Vamos a probar que existe una refutación lineal de S' con conjunto soporte $U = \{C\}$, donde C es cualquier cláusula de S'.

Sea $|S'| = n$ el número de literales positivos que ocurren en S'. Probaremos la existencia de la refutación con soporte $\{C\}$ de S' por inducción sobre n.

Si $n = 0$, entonces $S' = \{\{\perp\}\}$ y $C = \{\perp\}$. Luego, es claro que $S \vdash_l^{\{C\}} \perp$.

Supongamos que $n > 0$. Consideremos los siguientes casos:

a) Supongamos que $|C| = 1$. Es decir, $C = \{l\}$, con l un literal positivo. Consideremos el conjunto

$$(S')^l = \left\{D - \{l^c\} : l \notin D,\ D \in S'\right\}.$$

Como S' es insatisfacible, entonces $(S')^l$ es insatisfacible. Consideremos un subconjunto S'' de $(S')^l$ insatisfacible minimal.

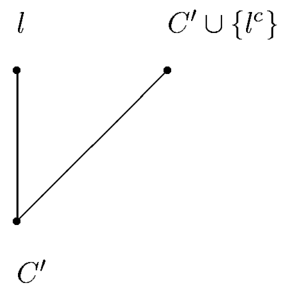
Comprobemos que existe una cláusula $C' \in S''$ tal que

$$C' \cup \{l^c\} \in S'.$$

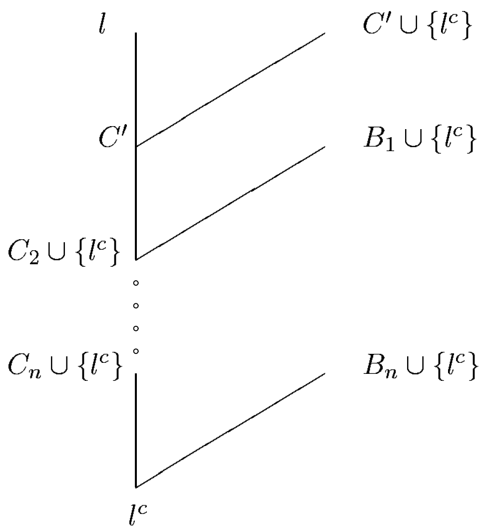
Supongamos que no existe tal cláusula. es decir, para toda cláusula $C' \in S''$, $C' \cup \{l^c\} \notin S'$. Como $C' \in S'' \subset (S')^l$, entonces $l \notin C'$. Luego, $l, l^c \notin C'$. Por lo tanto $S'' \subset S' - \{l\}$, pero esto es una contradicción puesto que S' es un conjunto insatisfacible minimal. Por lo tanto, existe alguna cláusula $C' \in S''$ tal que $C' \cup \{l^c\} \in S'$. Como $S'' \subset (S')^l$ y $\left|(S')^l\right| < n$, entonces $|S''| < n$. Por hipótesis inductiva, existe una refutación lineal de S'' con soporte C'. Es decir, existe una sucesión

$$\left(C', B_0\right), \left(C_1, B_1\right), ..., \left(C_n, B_n\right), C_{n+1} = \perp. \tag{7.1}$$

A partir de ésta resolución lineal podemos construir una refutación lineal de S' con soporte $C = \{l\}$ de la siguiente forma. El primer paso de la nueva refutación es la sucesión $C = \{l\}$, $C' \cup \{l^c\}$, $\mathrm{Res}\left(\{l\}, C' \cup \{l^c\}\right) = C'$ como muestra la figura



Ahora unimos la anterior resolución con la refutación (7.1) pero tomando las cláusulas en S' en vez de S''. Es decir, para cada cláusula b_i, $0 \leq i \leq n$, tomamos la cláusula $B_i \cup \{l^c\} \in S'$, y para cada cláusula C_i, $1 \leq i \leq n$, tomamos la cláusula $C_i \cup \{l^c\}$. De esta forma a partir de la refutación (7.1) de S'' construimos una deducción por resolución lineal de la cláusula $\{l^c\}$ a partir de S'. Gráficamente



A la secuencia de la figura anterior le unimos como último paso la deducción, l, $\mathrm{Res}\,(l, l^c) = \perp$. Por lo tanto, hemos construido una refutación lineal de S' con conjunto soporte $C = \{l\}$.

b) Supongamos ahora que $|C| > 1$. Entonces la cláusula C tiene más de un literal positivo. Consideremos un literal arbitrario $l \in C$ y definimos la cláusula $C' = C - \{l\}$. Consideremos el conjunto de cláusulas

$$\left(S'\right)^{l^c} = \left\{D - \{l\} : l^c \notin D,\ D \in S'\right\}.$$

Como S' es insatisfacible, entonces $\left(S'\right)^{l^c}$ es insatisfacible. Observemos que $C' = C - \{l\} \in \left(S'\right)^{l^c}$, pues en caso contrario, como $l \notin C'$ entonces $l^c \in C'$. Entonces $l, l^c \in C$ y C sería una cláusula válida.

Esto implicaría que S' no es insatisfacible minimal (pues $S' - \{C\}$ sería insatisfacible).

Probemos que $(S')^{l^c} - \{C\}$ es un conjunto satisfacible. Como S' es insatisfacible minimal, entonces S' -$\{C\}$ es satisfacible. Sea v una valuación tal que $v(S' - \{C\}) = 1$. Como $C \in S'$ y S' es insatisfacible, entonces $v(C) = 0$ y en consecuencia $v(l) = 0$ (pues $l \in C$). Probemos que toda cláusula $D \in (S')^{l^c}$ tal que $B \neq C$, $v(D) = 1$. Sea $D \in (S')^{l^c}$. Entonces $l, l^c \notin D$ y $D \cup \{l\} \in S'$. Luego, $D \cup \{l\} \in S' - \{C\}$ y en consecuencia $v(D \cup \{l\}) = 1$, y puesto que $v(l) = 0$, concluímos que $v(D) = 1$. Con esto probamos que $(S')^{l^c} - \{C\}$ es satisfacible.

Consideremos ahora un conjunto $S'' \subseteq (S')^{l^c}$ insatisfacible minimal. Como $C' = C - \{l\} \in (S')^{l^c}$ y S'' es minimal, entonces $C' \in S''$ (pues en caso contrario $S'' \subseteq (S')^{l^c} - \{C\}$ y S'' no sería insatisfacible minimal). Por lo tanto, por hipótesis inductiva, existe una refutación lineal a partir de S'' con soporte $\{C'\}$, es decir existe una sucesión

$$\left(C', B_0\right), (C_1, B_1), ..., (C_n, B_n), \perp.$$

A partir de ésta sucesión construímos una nueva deducción lineal agregando a cada una de las cláusulas anterires el literal l

$$\left(C' \cup \{l\}, B_0 \cup \{l\}\right), (C_1 \cup \{l\}, B_1 \cup \{l\}), ..., (C_n \cup \{l\}, B_n \cup \{l\}), l. \tag{7.2}$$

Luego tenemos una resolución lineal de l con soporte $C = C' \cup \{l\}$ a partir de S'.

Por otra parte, observemos que el conjunto

$$\left(S' - \{C\}\right) \cup \{l\}$$

es insatisfacible y S' -$\{C\}$ es satisfacible. Aplicando el caso anterior a), podemos afirmar que existe una refutación lineal de $(S' - \{C\}) \cup \{l\}$ con soporte $\{l\}$:

$$(\{l\}, A_0), (H_1, A_1), ..., (A_n, H_n), \perp \tag{7.3}$$

Por último, de (7.2) y (7.3) obtenemos una refutación de S' basada en C. ■

7.2. Cláusulas de Horn

En esta sección vamos a estudiar refinamientos del método de resolución para un conjunto particular de cláusulas. Esta clase particular de cláusulas son las cláusulas que intervienen en Prolog.

Definición 7.11 Una ***cláusula de Horn*** es una cláusula C que contiene como máximo un literal positivo. Es decir: C es una cláusula de Horn si tiene alguna de las siguientes formas:

1. $\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n \vee Q$
2. Q
3. $\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n$.

donde $P_1, ..., P_n$ y Q son fórmulas atómicas.

Una cláusula de Horn C escrita con los cuantificadores tiene alguna de las siguientes formas

1. $\forall x_1 ... \forall x_n (\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n \vee Q) \equiv \forall x_1 ... \forall x_n ((P_1 \wedge P_2 \wedge \ldots \wedge P_n) \to Q)$
2. $\forall x_1 ... \forall x_n Q$
3. $\forall x_1 ... \forall x_n (\neg P_1 \vee \neg P_2 \vee \ldots \vee \neg P_n) \equiv \forall x_1 ... \forall x_n \neg (P_1 \wedge P_2 \wedge \ldots \wedge P_n)$
$$\equiv \neg (\exists x_1 ... \exists x_n (P_1 \wedge P_2 \wedge \ldots \wedge P_n)).$$

Las cláusulas del tipo 1. se llaman ***reglas***, las cláusulas del tipo 2. se llaman ***hechos* (facts) o cláusulas unitarias.** Cláusulas de programas son cláusulas del tipo 1. o 2.. Una ***lógica de programas*** (o un Prolog programa) es un conjunto de cláusulas de programas. Una cláusula del tipo 3. se denomina una ***cláusula objetivo (goal cláusula).***

Sea P una lógica de programas y G una cláusula objetivo. El problema básico en programación lógica es determinar si
$$P \vDash \neg G.$$
Recordemos que $P \vDash \neg G$ significa que si $\mathcal{M}$ es un modelo tal que $\mathcal{M} \vDash P$ entonces $\mathcal{M} \vDash \neg G$. Recordemos también que

$$P \vDash \neg G \text{ si y solo si } P \cup \{G\} \text{ es insatisfacible si y solo si } P \cup \{G\} \vdash_R \perp.$$

Por lo tanto, para probar que $\neg G$ se deduce semánticamente de la lógica de programas P debemos construir una deducción por resolución del conjunto $P \cup \{G\}$.

Por otra parte, ya sabemos que si S es un conjunto de cláusulas, entonces

$$S \text{ es satisfacible si y solo si tiene un modelo de Herbrand.}$$

El próximo resultado nos dice que toda lógica de programas P tiene un modelo de Herbrand.

Teorema 7.12 *Sea P una lógica de programas. Entonces P tiene un modelo de Herbrand, es decir, P siempre es satisfacible.*

Demostración. Consideremos el universo de Herbrand $U(P)$ y la base de Herbrand $B(P)$. Recordemos que todos los posibles modelos de Herbrand que satisfacen el conjunto de cláusulas P se determinan encontrando todos los subconjuntos de $B(P)$. Consideremos $Y = B(P) = \{P_i(t_1, ..., t_n) : P_i$ es un predicado n-ario c
Veamos que este conjunto define un modelo de Herbrand $M(H)$ tal que

$$M(H) \vDash P.$$

Debemos probar que para toda cláusula $C_i \in P$,

$$M(H) \vDash C_i.$$

Como P es una lógica de programas, entonces las cláusulas C_i de P tienen alguna de las siguientes formas
$$C_i = (P_1 \wedge P_2 \wedge \ldots \wedge P_n) \to Q$$

o

$$C_i = Q$$

donde P_i , Q son fórmulas atómicas positivas. Como $P_i, Q \in B(P)$, entonces $M(H) \vDash P_1 \wedge P_2 \wedge \ldots \wedge P_n$ y $M(H) \vDash Q$. Esto implica que

$$M(H) \vDash (P_1 \wedge P_2 \wedge \ldots \wedge P_n) \rightarrow Q \text{ y } M(H) \vDash Q.$$

Por lo tanto, para toda cláusula $C_i \in P$, $M(P) \vDash C_i$. ■

En el caso de cláusulas de programa proposicionales se puede demostrar un resultado similar al anterior.

Teorema 7.13 *Sea P una lógica de programas proposicionales. Entonces P es satisfacible.*

Demostración. Ejercicio. ■

Es importante destacar que si un conjunto de cláusulas de Horn es insatisfacible entonces debe existir una cláusula objetivo y una cláusula hecho en el conjunto de cláusulas, como se prueba a continuación.

Lema 7.14 *Sea S un conjunto de cláusulas de Horn. Si S es insatisfacible entonces existe al menos una cláusula objetivo y al menos un cláusula hecho.*

Demostración. Supongamos que $S = P \cup H$ donde P es una lógica de programas y H es un conjunto de todas las cláusulas objetivos. Por el Teorema 7.12 el conjunto P es satisfacible por un modelo de Herbrand.Luego si S es insatisfacible debe ocurrir que $H \neq \emptyset$, pues en caso contrario $S = P$ lo que implica que S es satisfacible. Por lo tanto existe al menos una cláusula objetivo en S.

Consideremos ahora la estructura de Herbrand $\mathcal{M}_\emptyset$ generada por el conjunto $Y = \emptyset \subset B(S)$. Sabemos que en tal estructura es válida la negación de toda fórmula atómica cerrada $P(t_1, ..., t_n)$, donde $P(t_1, ..., t_n)$ es una fórmula atómica que ocurre en alguna cláusula de S. Es decir, $\mathcal{M}_\emptyset \vDash \neg P(t_1, ..., t_n)$. En esta estructura es válida toda cláusula de a forma $\neg l_1 \vee \ldots \vee \neg l_n \vee l$ y toda cláusla de la forma $\neg l_1 \vee \ldots \vee \neg l_n$. Por lo tanto en $\mathcal{M}_\emptyset$ es válida toda regla y toda cláusula objetiva. Como S es insatisfacible entonces debe existir alguna cláusula de la forma $C = P(t_1, ..., t_n)$ tal que $\mathcal{M}_\emptyset \nvDash P(t_1, ..., t_n)$. En consecuencia S contiene al menos un hecho. ■

7.3. Resolución para cláusulas de Horn

En el caso de cláusulas de Horn podemos dar otros refinamientos del método de resolución más adecuados.

Ya sabemos que el método de resolución lineal es completo para cualquier conjunto de cláusulas. En particular para cláusulas de Horn. Este hecho lo establecemos en el siguiente teorema.

Teorema 7.15 *Sea S un conjunto de cláusulas de Horn. Si S es insatisfacible entonces existe una refutación lineal de S.*

Consideremos ahora un conjunto de cláusulas de Horn de la forma $P \cup \{G\}$, donde P es una lógica de programas y G una cláusula objetivo. Si queremos estudiar si de P se deduce semánticamente $\neg G$, es decir si $P \vDash \neg G$, entonces podemos estudiar si $P \cup \{G\} \vdash_R \bot$. En particular, podemos estudiar si existe una refutación lineal de $P \cup \{G\}$. En la deducción por resolución lineal de la cláusula $\bot$ a partir de $P \cup \{G\}$ se puede comenzar con la cláusula objetivo G. Esto genera un método de resolución lineal, llamado input resolución, que es el método básico en Prolog.

Definición 7.16 Sea P un conjunto de cláusulas de programa y G una cláusula objetivo. Una ***input refutación*** de $P \cup \{G\}$ es una refutación lineal de $P \cup \{G\}$ comenzando con la cláusula G (la input cláusula) y donde todas las cláusulas de entrada son cláusulas de P.

Observación. En general, el método de input resolución no es completo para cualquier conjunto de cláusulas. Solo es completo para conjuntos de cláusulas de Horn.

Teorema 7.17 *Sea P un conjunto de cláusulas de programa y G una cláusula objetivo. Si $S = P \cup \{G\}$ es insatisfacible, entonces existe una input refutación de $P \cup \{G\}$ comenzando con G.*

Demostración. Primero observemos que cuando resolvemos una cláusula objetivo $G = \neg p_1 \vee \cdots \vee \neg p_n$ la otra cláusula que interviene en la resolvente solo puede ser una cláusula de programa, es decir una cláusula de la forma $\neg p_1 \vee \cdots \vee \neg p_n \vee p$ o una cláusula de la forma $C = p$. Por lo tanto, la resolvente contiene únicamente literales negativos (variables proposicionales en el caso proposicional o fórmulas atómicas en el caso de predicados). Es decir, la resolvente es otra cláusula objetivo. Por lo tanto, en una input refutación la resolvente de un par de cláusulas siempre es una cláusula objetivo y las cláusulas de entradas son cláusulas de programa.

Veamos ahora la prueba del teorema. Por el teorema de Compacidad podemos suponer que P es un conjunto finito de cláusulas de programa. La prueba es por inducción sobre la cantidad de literales positivos de $P \cup \{G\}$. Como $P \cup \{G\}$ es insatisfacible, entonces por el Lema 7.14 existe al menos una cláusula hecho (es decir, un literal positivo) $C = l$ en P. Por el Lema 7.1 (o por el Corolario 7.2), el conjunto de cláusulas

$$S^l = (P \cup \{G\})^l = \{D - \{l^c\} : l \notin D,\, D \in S\} = P^l \cup G^l$$

es un conjunto insatisfacible. Notemos que como G es una cláusula objetivo (y en consecuencia no tiene el literal positivo l), entonces $G^l = G - \{l^c\}$. Como P es satisfacible (por el Lema 7.13), entonces P^l es satisfacible por la misma valuación que satisface a P. Como S^l contiene menos literales que S, entonces por la hipótesis inductiva existe una input refutación de $S^l = P^l \cup G^l$ comenzando por la cláusula objetivo $G^l = G - \{l^c\}$:

$$\left(G^l, B_0\right), (C_1, B_1), ..., (C_n, B_n), C_{n+1} = \bot. \tag{7.4}$$

Ahora construimos una input refutación a partir de cláusulas de S de la siguiente forma:

El primer paso es añadir el literal l^c a todas las cláusulas de la deducción anterior (7.4) obteniendo una secuencia de la forma

$$\left(G^l \cup \{l^c\} = G, B_0 \cup \{l^c\}\right), (C_1 \cup \{l^c\}, B_1 \cup \{l^c\}), ..., (C_n \cup \{l^c\}, B_n \cup \{l^c\}), C_{n+1} = l^c.$$

Como segundo paso agregamos al final de la secuencia anterior la deducción

$$l^c, \quad l, \quad \mathrm{Res}\,(l^c, l) = \perp.$$

Con lo cual obtenemos una input refutación de $P \cup \{G\}$ comenzando con G. ■

Ahora veremos otro refinamiento de resolución para cláusulas de Horn llamado resolución unitaria.

Definición 7.18 Sea $S \cup \{C\}$ un conjunto de cláusulas de Horn. Una deducción por ***resolución unitaria*** de C a partir de S es una deducción por resolución de C a partir de S donde cada paso de resolución interviene una cláusula unitaria.

Teorema 7.19 *Sea S un conjunto de cláusulas de Horn. Si S es insatisfacible entonces existe una refutación unitaria de S.*

Demostración. Ejercicio. ■

Sea P una lógica de programas y $G = \neg R_1 \vee \neg R_2 ... \vee \neg R_n$ una cláusula objetivo. Volvamos al problema de cuando $P \vDash \neg G$. Por el Teorema 7.12 la lógica de programas P siempre tiene un modelo $\mathcal{M}\,(H)$, justamente el modelo de Herbrand de P construido con la base de Herbrand $B\,(P)$. Como $P \vDash \neg G$ es equivalente a decir que $P \cup \{G\}$ es insatisfacible, entonces debe ocurrir que

$$\mathcal{M}\,(H) \nvDash G$$

o lo que es equivalente a decir

$$\mathcal{M}\,(H) \vDash \neg G = R_1 \wedge R_2 \wedge ... \wedge R_n.$$

Recordemos que la cláusula G está cuantificada universalmente, es decir,

$$G = \forall x_1 ... \forall x_n \left(\neg R_1 \vee \neg R_2 ... \vee \neg R_n\right).$$

Por lo tanto, $\mathcal{M}\,(H) \vDash \neg G = R_1 \wedge R_2 \wedge ... \wedge R_n$ es equivalente a

$$\mathcal{M}\,(H) \vDash \exists x_1 .... \exists x_k \left(R_1 \wedge R_2 \wedge ... \wedge R_n\right).$$

Entonces debe existir una sucesión finita $a_1, a_2, ..., a_k$ de elementos de $U\,(P)$ tal que satisfaga la fórmula $\exists x_1 .... \exists x_k \left(R_1 \wedge R_2 \wedge ... \wedge R_n\right)$, en símbolos

$$\mathcal{M}\,(H) \vDash R_1 \wedge R_2 \wedge ... \wedge R_n \left[a_1, a_2, ..., a_k\right].$$

Como cada elemento a_i de la sucesión $a_1, a_2, ..., a_k$ es un término sín variables y cada variable x_i que aparece en la fórmula $\exists x_1 .... \exists x_k \left(R_1 \wedge R_2 \wedge ... \wedge R_n\right)$ es reemplazada por el término a_i, entonces estamos en presencia de una sustitución e definida por

$$e = \{x_1/a_1, ..., x_k/a_k\}.$$

Teniendo en cuenta esto, podemos escribir

$$\mathcal{M}(H) \vDash (R_1 \wedge R_2 \wedge ... \wedge R_n)\, e.$$

Es resúmen:

$$P \vDash \neg G \quad \Leftrightarrow \quad \text{existe una sustitución } e \text{ con dominio en las variables de } G$$
$$\text{y rango en el universo de Herbrand de } P \text{ tal que } P \vDash \neg Ge.$$

Notemos que la fórmula resultante $\neg Ge$ es cerrada, pues todas las variables de G son sustituidas por elementos de $U(P)$.

Definición 7.20 Sea P un programa y $G = \neg R_1 \vee \neg R_2 ... \vee \neg R_n$ una goal cláusula. Diremos que una sustitución e para las variables de G es una ***sustitución de respuesta correcta*** *(src)* si

$$P \vDash (R_1 \wedge R_2 \wedge ... \wedge R_n)\, e$$

Teniendo en cuenta esta definición y las consideraciones hechas anteriormente, podemos afirmar que

$$P \vDash \neg G \Leftrightarrow P \cup \{G\} \vdash_R \bot \Leftrightarrow \text{existe una src } e \text{ tal que } P \vDash \neg Ge.$$

Ejemplo 7.21 Consideremos la lógica de programas

$$P = \left\{ \underbrace{\neg P(x,y) \vee Q(x,y)}_{(C_1)}, \underbrace{\neg P(x,z) \vee \neg Q(z,x) \vee Q(x,y)}_{(C_2)}, \underbrace{P(b,a)}_{(C_3)}, \underbrace{P(a,b)}_{(C_4)} \right\}$$

y la cláusula objetivo

$$G = \neg Q(y,b) \vee \neg Q(b,z) .$$

Determinar si $P \vDash \neg G$, o lo que es lo mismo, determinar una sustitución de respuesta correcta e tal que $P \vDash \neg Ge$.

Primero vamos a construir una input refutación de $P \cup \{G\}$. Recordemos que se deben cambiar las variables en los casos necesarios.

1. Consideremos la cláusula $C_2 = \neg P(x,z) \vee \neg Q(z,x) \vee Q(x,y)$. Resolveremos con la cláusula G, pero ántes cambiamos las variables de C_2, obteniendo

$$C_2 = \neg P(x,w) \vee \neg Q(w,x) \vee Q(x,v) .$$

   En este caso $L_1 = \{Q(x,v)\}$ y $\neg L_2 = \{Q(y,b), Q(b,z)\}$. El unificador más general para $L_1 \cup \neg L_2$ es $\{x/b, v/b, y/b, z/b\}$:

$$C_5 = \text{Res}(\neg P(b,w) \vee \neg Q(w,b) \vee Q(b,b), \neg Q(b,b)) = \neg P(b,w) \vee \neg Q(w,b) .$$

2. Resolvemos ahora C_5 con $C_3 = P(b,a)$

$$C_6 = \text{Res}(\neg P(b,w) \vee \neg Q(w,b), P(b,a)) \underset{w/a}{=} \neg Q(a,b) .$$

3. Resolvemos C_6 con C_1, para lo cual hacemos la sustitución $\{x/a, y/b\}$, obteniendo

$$C_7 = \mathrm{Res}\left(\neg P(x,y) \vee Q(x,y), \neg Q(a,b)\right) = \neg P(a,b).$$

4. Por último, resolvemos C_7 con $C_4 = P(a,b)$ la sustitución $e_4 = \{x/b, y/z\}$, obteniendo la cláusula

$$C_6 = \mathrm{Res}\left(\neg P(a,b), P(a,b)\right) = \perp$$

Por lo tanto, $P \vDash \neg G$. La sustitución de respuesta correcta es

$$u = \{y/b, z/b\}.$$

Como $\neg Ge = \neg\left(\neg Q(y,b) \vee \neg Q(b,z)\right)u = \left(Q(y,b) \wedge Q(b,z)\right)u = Q(b,b)$, entonces,

$$P \vDash \exists y \exists z Q(y,b) \wedge Q(b,z).$$

Ejemplo 7.22 Ahora aplicaremos resolución unitaria para obtener una refutación del conjunto de cláusulas $P \cup \{G\}$ del ejemplo anterior.

1. Resolvemos $C_1 = \neg P(x,y) \vee Q(x,y)$ con $C_3 = P(b,a)$. Haciendo la sustitución x/b, y/a obtenemos

$$C_5 = \mathrm{Res}\left(\neg P(x,y) \vee Q(x,y), P(b,a)\right) = Q(b,a).$$

2. Resolvemos $C_2 = \neg P(x,z) \vee \neg Q(z,x) \vee Q(x,y)$ con C_5, Haciendo la sustitución z/b, x/a obtenemos

$$C_6 = \mathrm{Res}\left(\neg P(x,z) \vee \neg Q(z,x) \vee Q(x,y), Q(b,a)\right) = \neg P(a,b) \vee Q(a,y).$$

3. Resolvemos C_6 con C_4

$$C_7 = \mathrm{Res}\left(\neg P(a,b) \vee Q(a,y), P(a,b)\right) = Q(a,y).$$

4. Resolvemos C_7 con $G = \neg Q(y,b) \vee \neg Q(b,z)$. En este caso primero cambiamos las variables de C_7 y obtenemos $C_7 = Q(a,w)$. Empleamos la sustitución y/a, w/b, obteniendo

$$C_8 = \mathrm{Res}\left(\neg Q(y,b) \vee \neg Q(b,z), Q(a,w)\right) = \neg Q(b,z).$$

5. Resolvemos C_8 con C_1 con la sustitución x/b, z/y, obteniendo

$$C_9 = \mathrm{Res}\left(\neg P(x,y) \vee Q(x,y), \neg Q(b,z)\right) = \neg P(b,y).$$

6. Por último, resolvemos C_9 con C_3 empleando la sustitución y/a

$$C_9 = \mathrm{Res}\left(\neg P(b,y), P(b,a)\right) = \perp.$$

Por lo tanto, existe un refutación unitaria de $P \cup \{G\}$.