Cálculo Proposicional

3. La consecuencia lógica y la deducción

Como ya se comentó, la lógica estudia las formas de razonamiento, permite analizar si las argumentaciones o razonamientos propuestos son correctos o no. Argumentar consiste en deducir una conclusión a partir de un conjunto de premisas que se tienen por verdaderas. Un argumento, por lo tanto, estará compuesto de algunas premisas y de una conclusión. Un argumento puede ser válido o no serlo. Consideremos, por ejemplo, el siguiente argumento:

Premisa 1) Si estudio entonces aprobaré

Premisa 2) No he estudiado

Conclusión: No aprobaré.

Considerando este argumento desde la intuición, parece no ser válido porque, aun siendo verdaderas las premisas 1) y 2), la conclusión no tiene por qué serlo, pues es posible que apruebe sin estudiar (copiando, por ejemplo). Así pues, aunque sea verdad que si estudio aprobaré, y aunque sea también verdad que no he estudiado, a partir de ahí no se puede deducir con absoluta certeza que vaya a desaprobar, por lo que el argumento no es válido y puede considerarse mentiroso.

No ocurre lo mismo en el argumento presentado en el próximo ejemplo:

Premisa 1) Si Alicia llega tarde a casa, será castigada

Premisa 2) Alicia ha llegado tarde a casa

Conclusión: Alicia será castigada

Este argumento, en cambio, es válido porque si las premisas son verdaderas, y suponemos que lo son, entonces, necesariamente, la conclusión debe ser verdadera, ya que la única forma de que Alicia no sea castigada es que, o bien la premisa 1) sea falsa, o lo sea la premisa 2); pero si ambas premisas son verdaderas, la conclusión también lo será.

De lo dicho podemos concluir la siguiente definición de argumento correcto o válido:

Definición 1 *Un argumento se considera **válido** si y sólo si, bajo cualquier asignación de valores de verdad a las variables proposicionales, no es posible que las premisas sean verdaderas y la conclusión falsa.*

Un argumento es *inválido* o incorrecto si es posible asignar valores de verdad a las variables proposicionales que aparecen en él, de tal manera que las premisas tomen el valor de verdad verdadero ($\top$) y la conclusión tome el valor falso ($\perp$).

Este concepto es formalizado por la *consecuencia lógica*, ésta es la noción más importante considerada en lógica.

Sea S un conjunto de fórmulas, $S \subseteq \mathbf{Form}$, y sea $P \in \mathbf{Form}$ una fórmula, para cada asignación de valores de verdad a las variables proposicionales, algunas de las fórmulas de S podrán resultar verdaderas y otras falsas, y lo mismo ocurre con P. Intuitivamente parece natural pedir que para que P sea una consecuencia lógica de S, se cumpla que *cualquier* asignación de valores de verdad que haga verdaderas a *todas* las fórmulas de S también haga verdadera a P.

Esta idea intuitiva es la que tomaremos como definición semántica de la relación de consecuencia. Antes, comenzaremos por definir los siguientes conceptos:

Definición 2 *Diremos que una **valuación v satisface a una fórmula** $P \in \mathbf{Form}$ si dicha valuación le asigna a P el valor verdadero, $v(P) = \top$, y diremos que v **satisface a un conjunto** $S \subseteq \mathbf{Form}$ si v satisface a todas las fórmulas de S, esto es, si $v(Q) = \top$ para toda $Q \in S$.*

Definición 3 *Una fórmula (o un conjunto de fórmulas) se dice **satisfacible** si existe una valuación que la (o lo) satisfaga, e **insatisfacible** en caso contrario.*

Observación 1: Toda valuación v satisface (trivialmente) al conjunto vacío.

Observación 2: Un conjunto finito y no vacío de fórmulas es satisfacible si y sólo si la conjunción de las fórmulas que lo componen es satisfacible. Sea $S = \{Q_1, Q_2, ..., Q_n\}$ entonces

S es saticfacible si y sólo si para alguna valuación v, tenemos que $v(Q_1 \wedge Q_2 \wedge ... \wedge Q_n) = \top$

Observación 3: Una fórmula $P \in \mathbf{Form}$ tal que $var(P) = \{x_1, x_2, \ldots x_n\}$ es satisfacible si y sólo si existe una n-upla $(a_1, a_2, \ldots, a_n) \in \mathbf{B^n}$ tal que $T_p(a_1, a_2, \ldots, a_n) = \top$.

De las observaciones precedentes, resulta que tenemos un procedimiento efectivo para poder determinar si un conjunto finito y no vacío de fórmulas es o no satisfacible: hacemos primero la conjunción de las fórmulas del conjunto, y luego vamos construyendo la tabla de verdad de esta conjunción, hasta obtener una fila en la cual el valor de la última columna sea $\top$. Si encontramos tal fila, el conjunto es *satisfacible*. Si en la última columna de la tabla figuran solamente $\perp$, entonces el conjunto es *insatisfacible*.

Observación 4: Es claro que una fórmula es satisfacible si y sólo si es una tautología o una contingencia. Luego, las *fórmulas insatisfacibles* coinciden con las contradicciones (o falsedades).

Utilizando este concepto, diremos que un argumento es válido si todas las valuaciones que satisfacen a las premisas también satisfacen a la conclusión; en cambio es inválido si existen valuaciones que satisfacen a las premisas pero no satisfacen a la conclusión.

Definición 4 *Sean $S \subseteq \mathbf{Form}$ y $P \in \mathbf{Form}$. Diremos que P es **consecuencia de** S, y escribiremos $S \models P$, en el caso que toda valuación v que satisface a S también satisface a P. Esto es,*

$$\text{si } v(Q) = \top \text{ para toda } Q \in S, \text{ entonces } v(P) = \top$$

Indicaremos con $\mathbf{Con}(S)$ al conjunto de las consecuencias de S:

$$\mathbf{Con}(S) = \{P \in \mathbf{Form} \ / \ S \models P\}$$

Es fácil comprobar que la relación de consecuencia es *transitiva*, es decir, si $Q \models P$ y $P \models R$, entonces $Q \models R$.

Observación 5: De la definición anterior resulta que P es consecuencia del conjunto vacío, $\emptyset \models P$, si y sólo si P es una tautología. Luego, $\mathbf{Con}(\emptyset)$ *es el conjunto de todas las tautologías*.

El resultado siguiente pone de manifiesto la relación existente entre el conectivo binario $\rightarrow$ (condicional) y la noción de consecuencia.

Teorema 1 (Teorema de la Deducción) *(forma semántica) Sean las fórmulas P, Q y el conjunto $S \subseteq$* $\mathbf{Form}$. *Entonces*

$$Q \in \mathbf{Con}(S \cup \{P\}) \textit{ si y sólo si } (P \rightarrow Q) \in \mathbf{Con}(S)$$

Demostración: Vamos a probarlo por el absurdo, para esto suponemos que se cumple que $Q \in \mathbf{Con}(S \cup \{P\})$ y que $(P \rightarrow Q) \notin \mathbf{Con}(S)$; entonces si v es una valuación que satisface a S, sabemos que $v((P \rightarrow Q)) = \bot$. Por otro lado, sabemos que el condicional es falso, solo si $v(P) = \top$ y $v(Q) = \bot$, lo que implica que v satisface a $S \cup \{P\}$ y no satisface a Q; esto significa que $Q \notin \mathbf{Con}(S \cup \{P\})$.

Esto muestra que si $P \rightarrow Q \notin \mathbf{Con}(S)$, entonces $Q \notin \mathbf{Con}(S \cup \{P\})$, lo que contradice la hipótesis.

Para probar la recíproca, supongamos que $(P \rightarrow Q) \in \mathbf{Con}(S)$, y que v es una valuación que satisface a $S \cup \{P\}$. Entonces, sabiendo que $(P \rightarrow Q) \equiv (\neg P \vee Q)$, podemos asegurar que $v(P \rightarrow Q) = v(\neg P \vee Q) = \neg v(P) \vee v(Q)$. Por otra parte, se sabe que el operador $\vee$ solo es falso cuando ambas proposiciones son falsas, y si tenemos la fórmula $(p_0 \vee \bot)$ el valor de verdad que le corresponda a la misma solo depende de $v(p_0)$ ya que si $v(p_0) = \top$ la fórmula será verdadera y si $v(p_0) = \bot$ esta será falsa. A partir de estas consideraciones, analicemos cuál sería el valor de verdad de Q:

$$v(Q) = \bot \vee v(Q) = \neg v(P) \vee v(Q) = v(P \rightarrow Q) = \top$$

lo que muestra que $Q \in \mathbf{Con}(S \cup \{P\})$ c.q.d.■

Tomando $S = \emptyset$ en el teorema anterior y teniendo en cuenta la Observación 5, se obtiene el siguiente corolario:

Corolario 1 *Para todo par de fórmulas P y Q se tiene que $\{P\} \models Q$ si y sólo si $(P \rightarrow Q)$ es una tautología.*

Si analizamos las argumentaciones teniendo en cuenta el concepto de consecuencia definido, vemos que un argumento será válido o correcto cuando la conclusión pertenezca a las consecuencias de las premisas.

Intuitivamente, todos los argumentos pueden convertirse en un condicional, ya que, después de todo, lo que un argumento está afirmando es que si las premisas son verdaderas, entonces la conclusión también lo es, o dicho de otro modo:

$$(P_1 \wedge P_2 \wedge ... \wedge P_n) \rightarrow C$$

Es decir, un argumento puede verse como un condicional, en el que el antecedente es la conjunción de todas las premisas $(P_1 \wedge P_2 \wedge ... \wedge P_n)$ y el consecuente es la conclusión C. Como sabemos, la tabla de verdad del condicional nos dice que éste sólo es falso cuando el antecedente es verdadero y el consecuente falso, y verdadero en el resto de casos. Esto coincide completamente con la definición de argumento válido, según la cual, una argumento será válido exactamente en los mismos casos en que el condicional que le corresponde lo sea. Como un condicional no puede ser verdadero si el antecedente es verdadero y el consecuente falso, un argumento no podrá ser correcto o válido si las premisas son verdaderas y la conclusión falsa.

No siempre es fácil averiguar intuitivamente si un argumento es correcto o no, por lo que en ocasiones es necesario recurrir a métodos más fiables que la intuición.

Si se considera el Teorema de la Deducción, la Observación 2 y el Corolario 1, resulta claro que podemos convertir cualquier argumento en un condicional, donde las premisas serán las fórmulas del conjunto S y la conclusión la fórmula P. De esta manera, podremos usar el método de las tablas de verdad para averiguar si un argumento dado es válido o no. Evidentemente, un argumento sólo será válido cuando el condicional correspondiente sea una tautología y no será válido en el resto de casos (si es una contradicción o si es una contingencia). Analicemos lo dicho considerando los ejemplos presentados al principio.

Para evaluar el primer ejemplo:

Premisa 1) Si estudio entonces aprobaré

Premisa 2) No he estudiado

Conclusión: No aprobaré.

Lo primero que se debe hacer para decidir si el argumento es válido o no, es formalizarlo:

Formalización de la *premisa* 1): $(p \rightarrow q)$ (si estudio entonces aprobaré)

Formalización de la *premisa* 2): $\neg\, p$ (no he estudiado)

Formalización de la *conclusión*: $\neg\, q$ (no aprobaré)

En segundo lugar, tenemos que convertir el argumento en un condicional. Como hemos visto, el antecedente del condicional estará formado por la conjunción de todas las premisas, y el consecuente por la conclusión, de modo que obtenemos lo siguiente:

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

Habiendo obtenido el condicional que le corresponde al argumento del ejemplo, es momento de hacer su tabla de verdad (que quedará a cargo del lector) y de analizarla; la tabla de verdad de esta fórmula nos revela que el condicional analizado es una contingencia, lo que significa que puede ser verdadero o no; las valuaciones que asignan $\perp$ a ambas variables proposicionales, hacen al condicional falso. Es decir, que es posible que sus premisas sean verdaderas y su conclusión falsa. Por lo tanto el argumento correspondiente no será correcto, como se dedujo intuitivamente en el análisis anterior.

Procedamos del mismo modo con el otro argumento propuesto como ejemplo:

Premisa 1) Si Alicia llega tarde a casa, será castigada

Premisa 2) Alicia ha llegado tarde a casa

Conclusión: Alicia será castigada

Como en el caso anterior, obtenemos el condicional que le corresponde al argumento que vamos a evaluar, que, tras formalizar cada una de las premisas y la conclusión, quedará como sigue:

$$(((p \rightarrow q) \wedge p) \rightarrow q)$$

La realización de la tabla de verdad correspondiente, que queda a cargo del lector, nos indica que la fórmula evaluada es una *tautología*, por lo tanto, podemos concluir que el argumento correspondiente es válido, y la tabla de verdad correspondiente es la *prueba* de su validez.

El siguiente teorema muestra la relación entre consecuencia y satisfacibilidad; su demostración es inmediata a partir de las respectivas definiciones:

Teorema 2 *Para todo conjunto $S \subseteq \mathbf{Form}$ y para toda fórmula $P \in \mathbf{Form}$, se tiene que $S \models P$ si y sólo si $S \cup \{\neg P\}$ es insatisfacible.*

Notar que si en este teorema hacemos $S = \varnothing$, obtenemos que *una fórmula P es una tautología si y sólo si $\neg P$ es insatisfacible*. En efecto, sabemos que si $\varnothing \models P$, entonces P es una tautología, y por otro lado, por el teorema anterior podemos afirmar que $\varnothing \cup \{\neg P\}$ es insatisfacible; es decir, $\neg P$ es una contradicción.

El Teorema 2 nos muestra otra manera de determinar si una fórmula P es o no consecuencia de un conjunto finito de fórmulas S. Para hacerlo, agregamos la fórmula $\neg P$ al conjunto S y, como establece la Observación 2, se realiza la tabla de verdad de la conjunción de las fórmulas de este nuevo conjunto, para establecer si la misma es satisfacible o no, lo que permite determinar si P es consecuencia de S o no.

Si bien la construcción de una tabla de verdad es un procedimiento efectivo para determinar si una fórmula dada es o no satisfacible, porque se hace un número finito de pasos bien determinados, este número puede ser muy grande y por lo tanto el método puede resultar impracticable. Para convencerse de esto, basta observar que si en P figuran n variables proposicionales, su tabla de verdad tendrá 2^n filas y $n + 1$ columnas, y podría ocurrir que el valor $\top$ apareciese sólo al final de la última fila.

A continuación describiremos un procedimiento, alternativo a las tablas de verdad, para determinar si una fórmula o un conjunto finito y no vacío de fórmulas es o no satisfacible. Este método, aunque también requiere una cantidad de pasos del orden del anterior, es más natural, pues es más próximo a la forma en que razonamos habitualmente y, más importante aún, es un método que puede generalizarse para el cálculo de predicados, lo que no ocurre con las tablas de verdad.

La idea básica de este método se remonta a los trabajos del matemático alemán G. Gentzen publicados en 1934. Este apunte seguirá la presentación del método publicada en 1968 por R. M. Smullyan, quien tuvo en cuenta ideas introducidas por E. W. Beth y K. Hintikka a mediados de la década del 50 [1]

Comenzaremos por el siguiente lema, cuya demostración resulta inmediata de la Definición 1 y de las tablas que representan el comportamiento de los conectivos del apunte de *Semántica del Cálculo Proposicional*:

Lema 1 *Sean P y Q fórmulas y v una valuación. Entonces se tienen las siguientes propiedades:*

($R_\neg$) *v satisface a $\neg\neg P$ si y sólo si v satisface a P,*

($R_\vee$) *v satisface a $(P \vee Q)$ si y sólo si v satisface a P o v satisface a Q,*

($R_{\neg\vee}$) *v satisface a $\neg(P \vee Q)$ si y sólo si v satisface a $\neg P$ y v satisface a $\neg Q$,*

($R_\wedge$) *v satisface a $(P \wedge Q)$ si y sólo si v satisface a P y v satisface a Q,*

($R_{\neg\wedge}$) *v satisface a $\neg(P \wedge Q)$ si y sólo si v satisface a $\neg P$ o v satisface a $\neg Q$,*

($R_\rightarrow$) *v satisface a $(P \rightarrow Q)$ si y sólo si v satisface a $\neg P$ o v satisface a Q y*

($R_{\neg\rightarrow}$) *v satisface a $\neg(P \rightarrow Q)$ si y sólo si v satisface a P y v satisface a $\neg Q$.*

[1] R.M. Smullyan, *First Order Logic*, Springer Verlag, Berlín-New York, 1968.

Las propiedades enunciadas en el lema anterior nos permiten definir reglas sintácticas que describen el comportamiento de los conectivos con respecto a la satisfacibilidad, y se pueden esquematizar en la forma siguiente:

$$\boldsymbol{R}_{\neg} \qquad \frac{\neg\neg P}{P}$$

$$\boldsymbol{R}_{\vee} \qquad \frac{(P \vee Q)}{P \mid Q} \qquad\qquad \boldsymbol{R}_{\neg\vee} \qquad \frac{\neg(P \vee Q)}{\neg P,\ \neg Q}$$

$$\boldsymbol{R}_{\wedge} \qquad \frac{(P \wedge Q)}{P,\ Q} \qquad\qquad \boldsymbol{R}_{\neg\wedge} \qquad \frac{\neg(P \wedge Q)}{\neg P \mid \neg Q}$$

$$\boldsymbol{R}_{\to} \qquad \frac{(P \to Q)}{\neg P \mid Q} \qquad\qquad \boldsymbol{R}_{\neg\to} \qquad \frac{\neg(P \to Q)}{P,\ \neg Q}$$

La fórmula que figura encima de la línea horizontal en cada una de las reglas se llama la *premisa* de dicha regla, y la o las fórmulas que figuran debajo, *la o las conclusiones* de la regla. Desde el punto de vista semántico, la veracidad de una conclusión depende de la veracidad de su premisa; es decir que si la premisa es satisfacible, alguna (o ambas) de sus conclusiones también lo es.

Analizando cada regla se puede ver que, en caso de haber dos conclusiones, se da uno de dos casos: están separadas por una coma, o por una barra vertical, esto tiene que ver con su satisfacibilidad. Cuando la premisa es verdadera y *ambas* conclusiones son verdaderas, entonces se representan separadas por una coma; en cambio, cuando *al menos una* de las conclusiones es verdadera, se representan separadas por la barra vertical.

Observación 6: En todos los casos se tiene que las conclusiones son o bien subfórmulas o bien negaciones de subfórmulas de la correspondiente premisa.

Podemos clasificar a estas reglas en dos tipos: las que llamaremos de tipo A, que tienen una única conclusión o dos conclusiones separadas por una coma, y las que llamaremos de tipo B, que tienen dos conclusiones separadas por una barra vertical.

Las reglas de tipo A son $R_{\neg}$, $R_{\neg\vee}$, $R_{\wedge}$ y $R_{\neg\to}$, y las de tipo B son $R_{\vee}$, $R_{\neg\wedge}$ y $R_{\to}$.

Las reglas de tipo A significan que una valuación satisface a la premisa si y sólo si satisface a las conclusiones, y las de tipo B, que una valuación satisface a la premisa si y sólo si satisface a por lo menos una de las conclusiones.

Lema 2 *Toda fórmula que no es una variable proposicional o negación de una variable proposicional, es de la forma de la premisa de una (y sólo una) de las reglas $R_{\neg}$, $R_{\vee}$, $R_{\neg\vee}$, $R_{\wedge}$, $R_{\neg\wedge}$, $R_{\to}$ y $R_{\neg\to}$.*

Demostración: Sea P una fórmula. Si P no es una variable proposicional, entonces del Corolario 1 (del apunte de *El lenguaje del Cálculo Proposicional*) resulta que se puede presentar uno y sólo uno de los siguientes casos:

(i) $P = \neg T$,

(ii) $P = (Q \vee R)$,

(iii) $P = (Q \wedge R)$ o

(iv) $P = (Q \to R)$.

En los casos **(ii)**, **(iii)** y **(iv)** se ve que P tiene de la forma de la premisa de las reglas $R_\vee$, $R_\wedge$ y $R_\rightarrow$, respectivamente, con lo que el teorema se cumple.

Consideremos el caso **(i)**. Si T es una variable proposicional, entonces P es la negación de una variable proposicional. En caso contrario, otra vez por el Corolario 1 (del apunte de *El lenguaje del Cálculo Proposicional*) resulta que se puede presentar uno, y sólo uno, de los casos siguientes:

(a) $P = \neg T = \neg\neg Q$,

(b) $P = \neg T = \neg(Q \vee R)$,

(c) $P = \neg T = \neg(Q \wedge R)$ o

(d) $P = \neg T = \neg(Q \rightarrow R)$,

de donde resulta que P es de la forma de la premisa de las reglas $R_\neg$, $R_{\neg\vee}$, $R_{\neg\wedge}$ y $R_{\neg\rightarrow}$. c.q.d. ■

La alternativa al método de las tablas de verdad para determinar si una fórmula, o un conjunto de fórmulas, es satisfacible consiste en la aplicación sistemática de las reglas definidas anteriormente para los distintos conectivos. En este punto cabe notar que durante el proceso, no se tomará en cuenta el valor de verdad de la fórmula analizada (como si lo hace el método de la tablas de verdad), aquí solo se buscará la regla cuya premisa coincida con la fórmula en cuestión y se aplicará la misma sin más. Este tipo de métodos que no consideran la verdad o falsedad de las fórmulas se denominan *sintácticos*.

Antes de describir el método formalmente, daremos un par de ejemplos que se resolverán intuitivamente.

Como primer ejemplo, utilizaremos las reglas para averiguar si la fórmula

$$P = (((p_0 \vee p_1) \rightarrow p_2) \rightarrow ((p_0 \rightarrow p_2) \wedge (p_1 \rightarrow p_2))) \tag{1}$$

es o no una tautología.

Para ello se usará el método del absurdo; se sabe que P es una tautología si y sólo si $\neg P$ es insatisfacible. Entonces supongamos que $\neg P$ sea satisfacible, entonces existiría una valuación v que la satisface, es decir $v(\neg P) = \top$. Veamos si esto es posible.

Como P es de la forma $(Q \rightarrow R)$, con $Q = ((p_0 \vee p_1) \rightarrow p_2)$ y $R = ((p_0 \rightarrow p_2) \wedge (p_1 \rightarrow p_2))$ la regla $R_{\neg\rightarrow}$ es aplicable a $\neg P$, lo que implica que $((p_0 \vee p_1) \rightarrow p_2)$ y $\neg((p_0 \rightarrow p_2) \wedge (p_1 \rightarrow p_2))$ deberían ser satisfacibles por la valuación v.

El siguiente árbol nos ayudará a visualizar la situación:

$$\begin{array}{c} \neg(((p_0 \vee p_1) \rightarrow p_2) \rightarrow ((p_0 \rightarrow p_2) \wedge (p_1 \rightarrow p_2))) \\ | \\ ((p_0 \vee p_1) \rightarrow p_2) \\ | \\ \neg((p_0 \rightarrow p_2) \wedge (p_1 \rightarrow p_2)) \end{array}$$

Si ahora se considera la fórmula que aparece en el segundo nodo, $((p_0 \vee p_1) \rightarrow p_2)$, vemos que la regla $R_\rightarrow$ es aplicable y por lo tanto v debería satisfacer a $\neg(p_0 \vee p_1)$ o a p_2. Para tener en cuenta ambas posibilidades, agregamos a nuestro árbol dos nuevos nodos, con estas dos fórmulas, que deben estar a la misma altura:

$$\neg(((p_0 \vee p_1) \to p_2) \to ((p_0 \to p_2) \wedge (p_1 \to p_2)))$$

$$((p_0 \vee p_1) \to p_2)$$

$$\neg((p_0 \to p_2) \wedge (p_1 \to p_2))$$

$$\neg(p_0 \vee p_1) \qquad\qquad p_2$$

Al seguir considerando los nodos que van apareciendo en el árbol, se ve que la regla $R_{\neg\wedge}$ es aplicable a la fórmula $\neg((p_0 \to p_2) \wedge (p_1 \to p_2))$, por lo que v debería satisfacer a $\neg(p_0 \to p_2)$ o a $\neg(p_1 \to p_2)$. Esta reglas se aplicará a una fórmula que está por encima de la bifurcación en el árbol, por lo tanto, para seguir teniendo en cuenta todas las posibilidades, agregamos las fórmulas que surgen en la conclusión, como nuevos nodos terminales a *cada uno* de los nodos terminales existentes, obteniendo el siguiente árbol:

$$\neg(((p_0 \vee p_1) \to p_2) \to ((p_0 \to p_2) \wedge (p_1 \to p_2)))$$

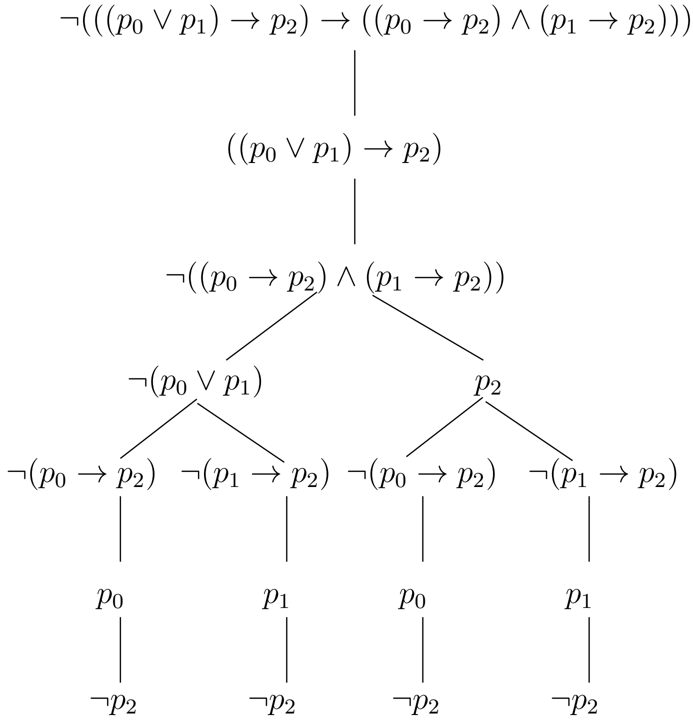
$$((p_0 \vee p_1) \to p_2)$$

$$\neg((p_0 \to p_2) \wedge (p_1 \to p_2))$$

$$\neg(p_0 \vee p_1) \qquad\qquad p_2$$

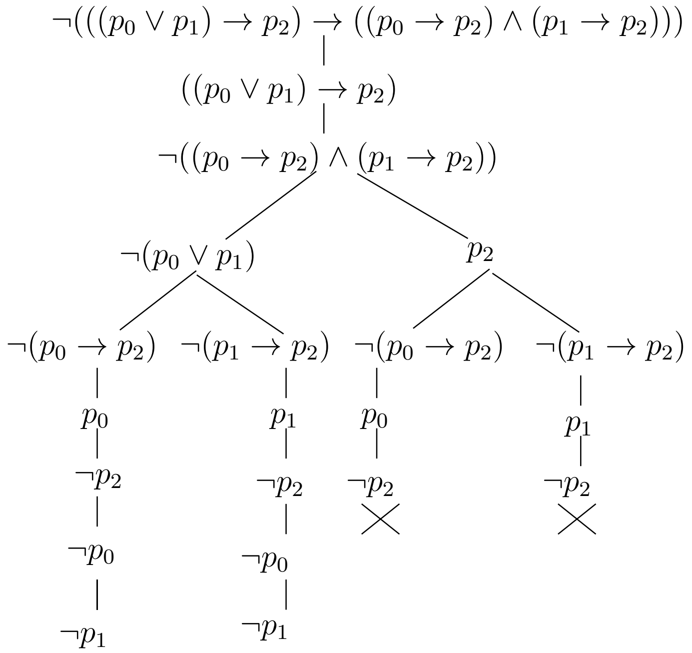
$$\neg(p_0 \to p_2) \quad \neg(p_1 \to p_2) \quad \neg(p_0 \to p_2) \quad \neg(p_1 \to p_2)$$

Continuando con este proceso, la regla $R_{\neg\to}$ es aplicable tanto a $\neg(p_0 \to p_2)$ como a $\neg(p_1 \to p_2)$. Por lo tanto, deberemos agregar sucesivamente las conclusiones de esta regla a los nodos terminales que se encuentran debajo de las respectivas fórmulas, obteniendo:

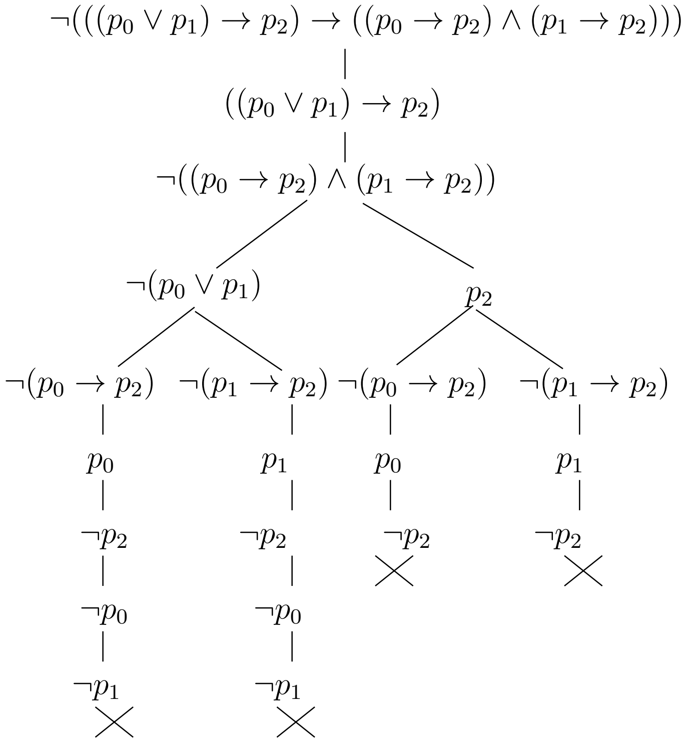


Observemos que en las dos ramas de la derecha del árbol que estamos construyendo, si las seguimos desde la raíz hasta el nodo hoja, figuran las fórmulas p_2 y $\neg p_2$. Como no puede haber ninguna valuación que satisfaga simultáneamente ambas fórmulas, los caminos indicados por estas dos ramas se nos "cierran", por lo que seguiremos aplicando las reglas sólo a las fórmulas que figuran en las dos ramas izquierdas. Indicamos este hecho colocando la marca $\times$ debajo de los nodos terminales de las dos ramas derechas.

Para continuar, se aplica la regla $R_{\neg\vee}$ a la fórmula $\neg(p_0 \vee p_1)$, por lo que debemos agregar sucesivamente a $\neg p_0$ y $\neg p_1$ como nuevos nodos a cada uno de los nodos terminales que figuran debajo de $\neg(p_0 \vee p_1)$, obteniendo:



En la primera rama de la izquierda figuran p_0 y $\neg p_0$, y en la segunda, p_1 y $\neg p_1$, por lo que debemos poner una marca $\times$ debajo de los nodos terminales de cada una de ellas, obteniendo:



Ahora en todos los nodos terminales del árbol figura la marca $\times$, lo que significa que todos los posibles caminos a seguir se nos han "cerrado" o "bloqueado". Luego, partiendo de la hipótesis que v satisface a la fórmula $\neg P$ y examinando los valores que v le debería asignar a todas las fórmulas que podemos derivar de $\neg P$ por aplicación de las reglas, siempre llegamos a una contradicción. Por lo tanto podemos concluir que no existe una valuación que satisfaga a $\neg P$, y por ende, que P *es una tautología*.

Como segundo ejemplo de aplicación de las reglas, veamos si la fórmula:

$$Q = ((p_0 \wedge p_1) \to \neg(p_0 \vee \neg p_1)) \tag{2}$$

es o no una tautología. Como en el caso de la fórmula P del ejemplo anterior, debemos averiguar si $\neg\, Q$ es o no satisfacible. Supongamos entonces que exista una valuación v tal que $v(\neg Q) = \top$. Como la regla $R_{\neg\to}$ es aplicable a $\neg Q$, v debe satisfacer a $(p_0 \wedge p_1)$ y a $\neg\neg(p_0 \vee \neg p_1)$. Podemos entonces presentar el árbol:

$$\begin{array}{c} \neg((p_0 \wedge p_1) \to \neg(p_0 \vee \neg p_1)) \\ | \\ (p_0 \wedge p_1) \\ | \\ \neg\neg(p_0 \vee \neg p_1) \end{array}$$

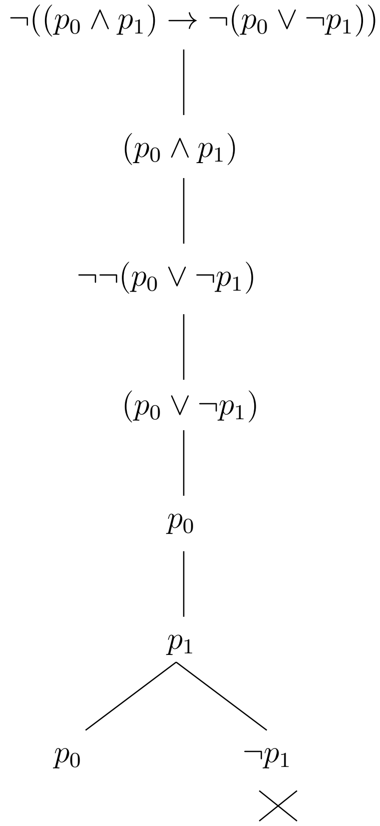
Aplicando la regla $R_\neg$ a la fórmula $\neg\neg(p_0 \vee \neg p_1)$, podemos agregar $(p_0 \vee \neg p_1)$ como nuevo nodo terminal:

$$\begin{array}{c}
\neg((p_0 \wedge p_1) \rightarrow \neg(p_0 \vee \neg p_1)) \\
| \\
(p_0 \wedge p_1) \\
| \\
\neg\neg(p_0 \vee \neg p_1) \\
| \\
(p_0 \vee \neg p_1)
\end{array}$$

y aplicando la regla $R_\wedge$ a $(p_0 \wedge p_1)$, obtenemos:

$$\begin{array}{c}
\neg((p_0 \wedge p_1) \rightarrow \neg(p_0 \vee \neg p_1)) \\
| \\
(p_0 \wedge p_1) \\
| \\
\neg\neg(p_0 \vee \neg p_1) \\
| \\
(p_0 \vee \neg p_1) \\
| \\
p_0 \\
| \\
p_1
\end{array}$$

Finalmente, aplicando la regla $R_\vee$ a $(p_0 \vee \neg p_1)$, nos queda:



La rama de la derecha aparece cerrada por la marca $\times$. En la rama de la izquierda todas las fórmulas que figuran arriba de su nodo terminal p_0 (y que no son variables proposicionales o negaciones de variables proposicionales), fueron utilizadas como premisa de alguna regla, por lo tanto se puede decir que está saturada, es decir que se ha *completado* el procedimiento.

Sea v una valuación tal que $v(p_0) = v(p_1) = \top$ (una tal valuación existe por el Corolario 1 del apunte de *La Semántica del Cálculo Proposicional*). Entonces "subiendo" por la rama de la izquierda, la que no esta cerrada, se puede ver que v satisface cada una de las fórmulas que figuran en la misma, en particular al nodo inicial o raíz de este árbol. Luego $v(\neg Q) = \top$. Esto significa que $\neg Q$ es satisfacible, y por lo tanto que Q *no es una tautología*.

A continuación formalizaremos la construcción de los árboles que vimos en los dos ejemplos anteriores[2].

Para abreviar, en lo sucesivo diremos que una fórmula P es de tipo A (o de tipo B), en lugar de decir que es de la forma de la premisa de una regla de tipo A (o de tipo B), y llamaremos *conclusiones de* P a las conclusiones de la única regla que tiene por premisa a P.

Definición 5 *Sea $\mathcal{A}$ un árbol cuyos nodos son fórmulas. Un árbol $\mathcal{A}'$ se dice una* ***extensión inmediata de*** *$\mathcal{A}$ si $\mathcal{A}'$ se obtiene por aplicación de una de las reglas $R_\neg$, $R_\vee$, $R_{\neg\vee}$, $R_\wedge$, $R_{\neg\wedge}$, $R_\rightarrow$ o $R_{\neg\rightarrow}$ a una fórmula P que figura en $\mathcal{A}$ del siguiente modo:*

a) *Si la fórmula es de tipo A, entonces $\mathcal{A}'$ se obtiene agregando un nuevo nodo terminal con una de las conclusiones de P como único sucesor del nodo terminal de una rama de $\mathcal{A}$ en donde figure P.*

b) *Si la fórmula es de tipo B, entonces $\mathcal{A}'$ se obtiene agregando dos nuevos nodos terminales, correspondientes a cada una de las conclusiones de P, como sucesores del nodo terminal de una rama de $\mathcal{A}$ en la que figure P.*

[2]En el Apéndice de este tema se dan todas las nociones sobre los árboles que usaremos en lo que sigue.

Definición 6 *Un árbol de fórmulas es un* ***árbol de refutación de una fórmula*** P *si y sólo si se obtiene realizando un número finito de extensiones inmediatas a partir del árbol cuyo único nodo es la fórmula* P.

Definición 7 *Un* ***árbol de refutación de un conjunto finito*** S ***de fórmulas*** *es un árbol de refutación de la conjunción de las fórmulas de* S.

Una rama de un árbol de fórmulas se dice ***cerrada*** si contiene a la vez a una fórmula y a su negación. En caso contrario, la rama se dice ***abierta***. Un árbol se dice ***cerrado*** si todas sus ramas son cerradas. Una rama de un árbol de fórmulas se dice ***saturada*** si el conjunto formado por las fórmulas de sus nodos es ***saturado*** en el sentido de la siguiente definición:

Definición 8 *Un conjunto* S *de fórmulas se dice* **saturado** *si y sólo si satisface las siguientes condiciones:*

(S0) *Una fórmula y su negación no pueden estar simultáneamente en* S.

(S1) *Si una fórmula de tipo A está en* S*, entonces sus dos conclusiones están en* S.

(S2) *Si una fórmula de tipo B está en* S*, al menos una de sus conclusiones está en* S.

Definición 9 *Un árbol de refutación se dice* **completo** *si cada una de sus ramas es cerrada o saturada.*

Notar que, en particular, *todo árbol de refutación cerrado es un árbol completo*.

En los dos ejemplos desarrollados anteriormente obtuvimos árboles de refutación completos de las fórmulas definidas en (1) y (2). El primero es cerrado, mientras que el segundo tiene una rama cerrada y la otra saturada.

Observemos que, si analizamos el grado de complejidad de las reglas utilizadas en los árboles de refutación, el grado de complejidad de las conclusiones no es necesariamente menor que el grado de complejidad de las premisas. Por ejemplo, si consideramos la regla $R \to$, la premisa $P = (p_0 \to p_1)$, tiene $comp(P) = 1$ y la complejidad de sus conclusiones, $\neg p_0$ y p_1, es $comp(p_1) = 0$ y $comp(\neg p_0) = comp(P) = 1$. Esto hace que, al momento de tratar de demostrar alguna propiedad sobre las mismas, no podamos argumentar por inducción en el grado de complejidad al pasar de una fórmula a sus conclusiones.

Lo que sí disminuye al pasar de una fórmula a sus conclusiones es la longitud. Pero la longitud no es adecuada para razonamientos inductivos, pues *las variables proposicionales pueden tener longitud arbitrariamente grande*. En efecto, recordemos que p_n abrevia al símbolo p seguido de n barras $|$, por lo que $long(p_n) = n + 1$. Para evitar este inconveniente, se puede modificar la definición de longitud para fórmulas, asignándole a las variables proposicionales *longitud 1*. Estas consideraciones conducen a la definición de otro tipo de longitud que nos será más útil:

Definición 10 *La* **longitud modificada** *de una fórmula* P*, que denotamos con* $long^*(P)$*, se define como el grado de complejidad de* P *más el número de variables que figuran en* P*, contadas tantas veces como aparezcan repetidas.*

En otras palabras, la longitud modificada es la función:

$$long^* : \mathbf{Form} \longmapsto \mathbb{N}$$

caracterizada por las siguientes propiedades:

(LM1) Para toda variable proposicional p_n, $long^*(p_n) = 1$,

(LM2) Para toda fórmula P, $long^*(\neg P) = 1 + long^*(P)$,

(LM3) Para todo par de fórmulas P y Q, $long^*(P \vee Q) = long^*(P \wedge Q) = long^*(P \rightarrow Q) = 1 + long^*(P) + long^*(Q)$.

El siguiente lema es el análogo del Corolario 1 (del apunte de *El lenguaje del Cálculo Proposicional*)

Lema 3 *Para toda fórmula P se tiene que:*

(i) *Si $long^*(P) = 1$, entonces P es una variable proposicional,*

(ii) *Si $long^*(P) = 2$, entonces P es la negación de una variable proposicional,*

(iii) *Si $long^*(P) \geq 3$, entonces P es de tipo A o B, y la longitud modificada de sus conclusiones es menor que $long^*(P)$.*

Demostración: Resulta de la definición de longitud modificada y de examinar cada una de las reglas $R_\neg$, $R_\vee$, $R_{\neg\vee}$, $R_\wedge$, $R_{\neg\wedge}$, $R_\rightarrow$ o $R_{\neg\rightarrow}$. c.q.d ■

Con esta definición de longitud modificada, estamos ahora en condiciones de probar el siguiente lema:

Lema 4 *Toda fórmula P tiene por lo menos un árbol de refutación completo.*

Demostración: Por inducción en la longitud modificada de P.

Si $long^*(P) = 1$ o $long^*(P) = 2$, entonces por el Lema 3 es una variable proposicional o la negación de una variable proposicional, y por consiguiente, su árbol de refutación tiene un único nodo que es P, y es un árbol de refutación completo de P ya que su única rama es saturada.

Supongamos que $long^*(P) \geq 3$, y que toda fórmula Q tal que $long^*(Q) < long^*(P)$ admite un árbol de refutación completo.

Por el Lema 3, P es de tipo A o de tipo B, y la longitud modificada de sus conclusiones es menor que $long^*(P)$.

Supongamos primero que P es de tipo A, y sea Q una conclusión de P. Sea $\mathcal{A}_1$ el árbol que se obtiene poniendo Q debajo de P, y luego desarrollando un árbol completo para Q (que existe por la hipótesis inductiva). Como $\mathcal{A}_1$ se puede obtener por un número finito de extensiones inmediatas a partir del nodo inicial P, es un árbol de refutación de P. Si $\mathcal{A}_1$ es cerrado, o si Q es la única conclusión de P (esto es, $P = \neg\neg Q$), entonces $\mathcal{A}_1$ es un árbol de refutación completo de P. En caso contrario, sea R la otra conclusión de P, esta se agrega como nuevo nodo terminal de todas las ramas abiertas de $\mathcal{A}_1$. Llamemos $\mathcal{A}_2$ al árbol que se obtiene desarrollando a partir de cada uno de estos nuevos nodos un árbol de refutación completo para R (que existe por la hipótesis inductiva). Como $\mathcal{A}_2$ se puede obtener haciendo un número finito de extensiones inmediatas a partir de P, es un árbol de refutación de P. Veamos que es completo. Si $\mathcal{A}_2$ es cerrado, no hay nada que probar. En caso de que no sea cerrado, supongamos que S es una rama abierta de $\mathcal{A}_2$, y sea $S_1 = S \cap \mathcal{A}_1$. Sea X una fórmula de tipo A o B en S. Si $X = P$, entonces por la construcción de $\mathcal{A}_2$, sus dos conclusiones Q y R están en S. Si $X \neq P$ y $X \in S_1$, entonces como $S_1 - \{P\}$ es una rama abierta de un árbol completo de Q, sus conclusiones (si es de tipo A), o una de ellas (si es de tipo B) deben estar en $S_1 \subseteq S$. Si $X \in (S - S_1)$, se puede aplicar el mismo argumento, pues $S - S_1$ es una rama abierta de un árbol de refutación completo de R. Luego en cualquier caso S es una rama completa de $\mathcal{A}_2$.

Supongamos a continuación que P es de tipo B, y sean Q y R sus dos conclusiones. Llamemos $\mathcal{A}$ al árbol que se obtiene del siguiente modo: partiendo del nodo inicial P, se agregan dos nodos terminales Q y R, y luego se desarrollan árboles completos con origen en cada uno de estos nodos, que existen por la hipótesis inductiva. Como $\mathcal{A}$ se puede obtener de P haciendo un número finito de extensiones inmediatas, es un árbol de refutación de P. Si es cerrado, entonces es completo. Supongamos que S es una rama abierta de $\mathcal{A}$, y que X es una fórmula de tipo A o B que está en S. Si $X = P$, entonces por la construcción de $\mathcal{A}$, una de sus conclusiones está en S. Si $X \neq P$, la demostración se termina observando que $S - \{P\}$ es una rama abierta de un árbol completo para Q o de un árbol completo para R. c.q.d. ■

Lema 5 *Si P es una fórmula satisfacible, entonces todo árbol de refutación de P tiene al menos una rama abierta.*

Demostración: Sea P satisfacible probaremos, por inducción en la altura, que todo árbol de refutación de P tiene por lo menos una rama abierta.

Si la altura del árbol es cero [3], entonces tiene un único nodo y por lo tanto una única rama formada por la fórmula P; dado que P es la única fórmula que aparece en esta rama, queda claro que su negación no pertenece a la rama, por lo cual ésta está *abierta*. Luego, siendo P satisfacible, su árbol de refutación tiene una rama abierta.

Hipótesis inductiva: suponemos que todo árbol de refutación de P de altura n tiene por lo menos una rama satisfacible abierta.

Sea $\mathcal{A}$ un árbol de refutación de P de altura $n+1$, y sea $\mathcal{A}'$ el árbol que se obtiene eliminando todos los nodos del nivel $n+1$ de $\mathcal{A}$. Es claro que $\mathcal{A}'$ es un árbol de refutación de P de altura n. Luego, por la hipótesis inductiva, $\mathcal{A}'$ tiene por lo menos una rama abierta. Elijamos una que llamaremos S. Podría ser que S fuera también una rama de $\mathcal{A}$, y en tal caso ya encontramos una rama abierta de $\mathcal{A}$.

En caso contrario el nodo terminal N de S en $\mathcal{A}'$ no sería un nodo terminal de $\mathcal{A}$ y habría dos posibilidades: o hay un único nodo terminal N_1 de $\mathcal{A}$ que sigue a N, o hay una bifurcación en dos nodos terminales de $\mathcal{A}$, N_1 y N_2.

En el primer caso N_1 es una de las conclusiones de una fórmula Q de tipo A que pertenece a S. Como toda valuación que satisface a Q también satisface a N_1, resulta que $S \cup \{N_1\}$ es una rama satisfacible de $\mathcal{A}$.

En el segundo caso los nodos N_1 y N_2 son las conclusiones de una fórmula R de tipo B perteneciente a S. Como toda valuación que satisface a R también debe satisfacer a N_1 o a N_2 resulta que por lo menos una de las dos ramas de $\mathcal{A}$, $S \cup \{N_1\}$ o $S \cup \{N_2\}$, es satisfacible.

Vemos así que $\mathcal{A}$ tiene una rama satisfacible, y como las ramas satisfacibles no pueden ser cerradas, resulta que $\mathcal{A}$ tiene una rama abierta. c.q.d. ■

Toda rama abierta de un árbol de refutación completo es saturada. La importancia de los conjuntos saturados está dada por el siguiente lema:

Lema 6 *Todo conjunto saturado de fórmulas es satisfacible.*

[3] Según la definición de altura dada en el apéndice, $alt(\mathcal{A})$.

Demostración: Sea S un conjunto saturado de fórmulas. Definimos la función $f : \mathbf{Var} \longmapsto \mathbf{B}$, donde $\mathbf{Var}$ denota el conjunto de las variables proposicionales, de la siguiente forma:

$$f(p_n) = \begin{cases} \top & \text{si} \quad p_n \in S \\ \bot & \text{si} \quad p_n \notin S \end{cases}$$

Si $v = v_f$ es la valuación que extiende a f, vamos a ver que $v(P) = \top$ para toda fórmula $P \in S$, lo que probará que S es satisfacible.

Haremos la demostración por inducción sobre la longitud modificada de P.

Si $P \in S$ y $long^*(P) = 1$, entonces P es una variable proposicional, digamos $P = p_n$, y por la definición de v sabemos que $v(P) = v(p_n) = f(p_n) = \top$.

Si la $long^*(P) = 2$ sabemos que P es la negación de una variable proposicional, entonces $P = \neg p_n$. Como $\neg p_n \in S$, por **(S0)** de la definición de conjunto saturado, resulta que $p_n \notin S$ y por lo tanto $f(p_n) = \bot$, por consiguiente $v(P) = \top$.

Supongamos que $long^*(P) \geq 3$ y que $v(Q) = \top$ para toda fórmula Q tal que $Q \in S$ y $long^*(Q) < long^*(P)$.

Sabemos, por el Lema 3, que P es de tipo A o de tipo B. En el primer caso, de la propiedad **(S1)** resulta que sus conclusiones están en S y su longitud modificada es menor que $long^*(P)$, por lo tanto según la hipótesis inductiva v les asigna el valor $\top$, lo que implica que $v(P) = \top$. Si P es de tipo B, por **(S2)** resulta que al menos una de sus conclusiones está en S y también tiene menor $long^*$ que P, por hipótesis inductiva v le asigna el valor $\top$, lo que implica que $v(P) = \top$.

Luego $v(P) = \top$ para toda fórmula $P \in S$. c.q.d. ■

Observación 7: En la demostración del lema anterior sólo se usó el siguiente caso particular de la propiedad **(S0)**:

(S0p) Una variable proposicional y su negación no pueden estar simultáneamente en S.

Luego se tiene que un conjunto S es saturado si y sólo si satisface las propiedades **(S0p)**, **(S1)** y **(S2)**. En efecto todo conjunto saturado satisface **(S0p)**, que es un caso particular de **(S0)**. Supongamos a continuación que S satisface **(S0p)**, **(S1)** y **(S2)**. Entonces por la demostración del Lema 6 resulta que existe una valuación v tal que $v(P) = \top$ para todo $P \in S$, lo que implica que S debe cumplir **(S0)**.

Los lemas anteriores se pueden resumir en el siguiente teorema:

Teorema 3 *Las siguientes propiedades son equivalentes para todo conjunto finito y no vacío $S \subseteq \mathbf{Form}$:*

(i) *S es satisfacible.*

(ii) *Todo árbol de refutación de S tiene por lo menos una rama abierta.*

(iii) *Existe un árbol de refutación de S completo con una rama abierta.*

Demostración: La primer propiedad **(i)** implica la segunda **(ii)**: esto es resultado del Lema 5.

La segunda propiedad **(ii)** implica a la tercera **(iii)**: Por el Lema 4 existe un árbol de refutación de S completo y por la hipótesis **(ii)** éste debe tener por lo menos una rama abierta.

La tercera propiedad **(iii)** implica la primera **(i)**: Sea $\mathcal{A}$ un árbol de refutación completo de S. Una rama abierta R de $\mathcal{A}$ constituye un conjunto saturado de fórmulas, y por el Lema 6 R es satisfacible. En particular la fórmula del nodo inicial de $\mathcal{A}$ es satisfacible, esto es, la conjunción de las fórmulas de S es satisfacible. Luego, por la Observación 2, resulta que S es satisfacible. c.q.d. ■

Corolario 2 *Las siguientes propiedades son equivalentes para todo conjunto finito y no vacío* $S \subseteq \mathbf{Form}$*:*

(i) *S es insatisfacible.*

(ii) *S tiene un árbol de refutación cerrado.*

(iii) *Todo árbol de refutación completo de S es cerrado.*

Corolario 3 *Las siguientes propiedades son equivalentes para toda fórmula P.*

(i) *P es un tautología.*

(ii) *$\neg P$ tiene un árbol de refutación cerrado.*

(iii) *Todo árbol de refutación completo de $\neg P$ es cerrado.*

Los dos últimos corolarios nos dan un método alternativo al de las tablas de verdad para determinar si un conjunto finito es o no satisfacible, o si una fórmula es o no una tautología. Más aún, en caso de encontrar un árbol completo con una rama abierta, asignando a las variables proposicionales los valores $\top$ y $\bot$ de acuerdo con lo especificado en la demostración del Lema 6 podemos definir una valuación que satisface a S. Queda como ejercicio usar este procedimiento para encontrar una valuación que satisface a la fórmula dada en el ejemplo (2).

Una sencilla inducción a partir del Teorema 1 muestra que una fórmula P es consecuencia de un conjunto finito $S = \{Q_1, \ldots, Q_n\} \subseteq \mathbf{Form}$ si y sólo si $(Q_1 \to (Q_2 \to (\ldots \to (Q_n \to P))\ldots) \in \mathbf{Con}(\emptyset)$, esto es si y sólo si $(Q_1 \to (Q_2 \to (\ldots \to (Q_n \to P))\ldots)$ es una tautología. Recordando que una fórmula es una tautología si y sólo si su negación es insatisfacible y teniendo en cuenta que:

$$\neg(Q_1 \to (Q_2 \to (\ldots \to (Q_n \to P))\ldots) \equiv Q_1 \wedge Q_2 \wedge \ldots \wedge Q_n \wedge \neg P$$

podemos dar la siguiente caracterización de las consecuencias de un conjunto finito S de fórmulas:

a) Si $S = \emptyset$, entonces $P \in \mathbf{Con}(S)$ si y sólo si la fórmula $\neg P$ tiene un árbol de refutación cerrado.

b) Si $S = \{Q_1, \ldots, Q_n\}$ entonces $P \in \mathbf{Con}(S)$ si y sólo si la fórmula $Q_1 \wedge \ldots \wedge Q_n \wedge \neg P$ tiene un árbol de refutación cerrado.

Como se mencionó anteriormente, que una fórmula P sea consecuencia de un conjunto de fórmulas S, asegura que P es una conclusión válida a partir de las premisas en S. Si consideramos esto desde el punto de vista *sintáctico*, podemos decir que la fórmula P se puede deducir a partir de las fórmulas en S. La noción de deducción es puramente sintáctica y se corresponde a la idea intuitiva de demostrar una proposición a partir de ciertas hipótesis, utilizando reglas de inferencia.

Si analizamos los árboles de refutación, podemos observar que son objetos sintácticos, es decir construidos a partir de fórmulas, siguiendo reglas en las que no intervienen explícitamente valores de verdad. En este caso las reglas de inferencia son las reglas definidas anteriormente para los árboles de refutación y se formaliza la idea de *demostrar* utilizando el método de “ reducción al absurdo”, habitual en matemáticas. Se parte de suponer que la fórmula de la raíz (la negación de fórmula analizada) es satisfacible y un árbol cerrado contradice esa hipótesis.

Estas observaciones sugieren las siguientes definiciones:

Definición 11 ***Una deducción de una fórmula*** P ***a partir de un conjunto de fórmulas*** S *es un árbol de refutación cerrado de la conjunción de* $\neg P$ *con todas las fórmulas de* S.

Definición 12 *Una fórmula* P *se dice* ***deducible*** *a partir de* $S \subseteq \mathbf{Form}$ *si existe una deducción de* P *a partir de* S. *El conjunto de las fórmulas deducibles a partir de* S *será denotado por* $\mathbf{Ded}(S)$.

Notar que cuando $S = \emptyset$ un árbol de refutación cerrado de $\neg P$ constituye una deducción de P.

Las condiciones **a)** y **b)** señaladas anteriormente muestran que para todo conjunto finito de fórmulas S se tiene que $\mathbf{Con}(S) = \mathbf{Ded}(S)$. Esto quiere decir que toda fórmula que se pueda deducir sintácticamente a partir de las fórmulas de S ($S \vdash P$), pertenece al conjunto de consecuencias de S ($S \models P$).

Luego, para los conjuntos finitos de fórmulas hemos reducido la noción *semántica* de consecuencia a la noción *sintáctica* de demostración (deducción). Vamos a ver ahora que esta reducción también es posible para conjuntos infinitos.

Teorema 4 **(Teorema de la Compacidad)** *Sea* S *un conjunto infinito de fórmulas. Si todo subconjunto finito de* S *es satisfacible entonces* S *es satisfacible.*

Demostración: Sea $S = \{Q_0, Q_1, \ldots, Q_n, Q_{n+1}, \ldots\}$ un conjunto infinito de fórmulas. Suponemos que todo subconjunto finito de S es satisfacible.

Sea $\mathcal{A}_0$ un árbol de refutación completo de Q_0. Como $\{Q_0\}$ es un subconjunto finito de S, $\mathcal{A}_0$ debe tener por lo menos una rama abierta. Sea $\mathcal{A}_1$ un árbol completo obtenido a partir de agregar Q_1 como nuevo nodo terminal a todas las ramas abiertas de $\mathcal{A}_0$. Notemos que $\mathcal{A}_1$ es un árbol de refutación del subconjunto finito $\{Q_0, Q_1\} \subseteq S$, luego tiene por lo menos una rama abierta. Sea $\mathcal{A}_2$ un árbol completo obtenido a partir de agregar la fórmula Q_2 como nuevo nodo terminal en cada rama abierta de $\mathcal{A}_1$. Entonces $\mathcal{A}_2$ es un árbol de refutación de $\{Q_0, Q_1, Q_2\}$ y por lo menos tiene una rama abierta. Podemos proseguir el procedimiento indicado para conseguir una sucesión infinita de árboles de refutación completos $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n, \mathcal{A}_{n+1}, \ldots$ de modo que $\mathcal{A}_{n+1}$ se obtiene completando el árbol construido con el agregado de la fórmula Q_{n+1} como nuevo nodo terminal de cada una de las ramas abiertas de $\mathcal{A}_n$. Como $\mathcal{A}_{n+1}$ es un árbol de refutación del conjunto finito $\{Q_0, Q_1, \ldots, Q_n, Q_{n+1}\} \subseteq S$, debe tener ramas abiertas y por lo tanto el procedimiento no se detiene.

Entonces $\mathcal{A} = \bigcup_{n\geq 0} \mathcal{A}_n$ es un árbol binario infinito, y por el Lema de König (*Todo árbol infinito y finitamente generado tiene al menos una rama infinita* - ver Apéndice) tiene una rama infinita que llamaremos R.

Para cada n, llamaremos $R_n = R \cap \mathcal{A}_n$. R_n es una rama de $\mathcal{A}_n$ que debe ser abierta pues de lo contrario no se hubiese prolongado a R_{n+1}. Entonces, como R es la unión de todas las R_n ($R = \bigcup_{n\geq 0} R_n$), resulta que R no es cerrada. Veamos que R es saturada.

Supongamos que una fórmula Q de tipo A está en R. Entonces $Q \in R_n$ para algún n. Como R_n es una rama abierta de $\mathcal{A}_n$, y este árbol es completo, R_n es saturada. Luego ambas conclusiones de Q están en $R_n \subseteq R$. Un argumento similar prueba que si una fórmula Q de tipo B está en R entonces una de sus conclusiones también lo está. Por lo tanto resulta que R es una rama abierta que además es saturada.

El Lema 6 asegura que un conjuto saturado es satisfacible, por lo tanto siendo R saturada, el conjunto de fórmulas en R es un conjunto satisfacible. Por otro lado, si consideramos la forma en que se construyó cada árbol $\mathcal{A}_n$, resulta que $S \subseteq R$ por lo cual también S es satisfacible. c.q.d. ■

Corolario 4 *Sea* S *un conjunto de fórmulas. Una fórmula* P *es consecuencia de* S *si y sólo si* P *es consecuencia de un subconjunto finito de* S.

Demostración: Como cualesquiera que sean los conjuntos S y R sabemos que $S \subseteq R$ implica que $\mathbf{Con}(S) \subseteq \mathbf{Con}(R)$, resulta que toda consecuencia de un subconjunto finito de S es también consecuencia de S.

Para probar la recíproca sea $P \in \mathbf{Con}(S)$. Por el Teorema 2, $S \cup \{\neg P\}$ es insatisfacible, y por el Teorema de Compacidad (Teorema 4), existe un subconjunto finito $F \subseteq S \cup \{\neg P\}$ tal que F es insatisfacible. Entonces $G = F - \{\neg P\}$ es un subconjunto finito de S (eventualmente vacío), y otra vez por el Teorema 2, se tiene que $P \in \mathbf{Con}(G) \subseteq \mathbf{Con}(S)$. c.q.d. ■

Observación 8: En algunos textos se da el nombre de Teorema de Compacidad al Corolario 4. En realidad el Teorema 4 y el Corolario 4 son *equivalentes* en el sentido que uno se deduce del otro. En efecto, ya vimos como el Corolario 4 se obtiene a partir de Teorema 4. Veamos ahora cómo se puede demostrar el Teorema 4 a partir del Corolario 4. Para ello supongamos cierto el Corolario 4 y sean S un conjunto de fórmulas insatisfacible y $P \in S$ (notemos que como S es insatisfacible , no puede ser vacío). Por el Teorema 4, $\neg P \in \mathbf{Con}(S - \{P\})$, y por el Corolario 4 existe un subconjunto finito $F \subseteq S - \{P\}$ tal que $\neg P \in \mathbf{Con}(F)$. Luego, otra vez por el Teorema 2, $F \cup \{\neg\neg P\}$ es insatisfacible y por lo tanto también lo es $F \cup \{P\}$. Luego hemos encontrado un subconjunto finito de S que es insatisfacible. Por lo tanto, si todo subconjunto finito de S es satisfacible, debe ser S satisfacible.

Finalmente extenderemos la equivalencia entre consecuencia y deducibilidad a conjuntos arbitrarios de fórmulas:

Corolario 5 *Para todo conjunto S de fórmulas* $\mathbf{Con}(S) = \mathbf{Ded}(S)$.

Entonces, para cualquier conjunto de fórmulas S, finito o infinito, toda fórmula que se pueda deducir sintácticamente a partir de las fórmulas del mismo, $S \vdash P$, pertenece al conjunto de sus consecuencias, $S \models P$.

Reconocimientos

El presente apunte se realizó tomando como base el Apunte de *Cálculo Proposicional* del **Dr. Roberto Cignoli** de la Universidad de Buenos Aires; el Libro *Lógica para Matemáticos* de **A. Hamilton** de Editorial Paraninfo; el libro *Lógica para informática* de **Claudia Pons, Ricardo Rosenfeld y Clara Smith** de Facultad de Informática de la Universidad Nacional de La plata (UNLP); libro Lógica Proposicional de **Felipe Garrido Bernabeu**, profesor de en I.E.S La Fola, de Ibi, Alicante (https://antesdelascenizas.com); y notas de clase del *Curso de Lógica y Computación* dictado por el **Dr. Guillermo Martínez** en la Universidad Nacional de San Luis.

Apéndice: Árboles y Lema de König

Llamaremos *árbol* a un conjunto parcialmente ordenado $\mathcal{A}$ con primer elemento a_0 y tal que para todo $a \in \mathcal{A}$, el conjunto de los antecesores de a, esto es, $\{x \in \mathcal{A} \mid x < a\}$, es finito y totalmente ordenado (con el orden heredado de $\mathcal{A}$).

Para cada $a \in \mathcal{A}$ definimos el *nivel de* a, que denotaremos como $niv(a)$, como el número de antecesores de a. Observemos que $niv(a_0) = 0$ puesto que el primer elemento no tiene antecesores.

Diremos que a es un *antecesor inmediato de* b y que b es un *sucesor inmediato* de a, en el caso en que $a < b$ y no existe $c \in \mathcal{A}$ tal que $a < c < b$. Observemos que en un árbol todo elemento distinto del primero tiene un único antecesor inmediato, pero que en cambio no hay restricciones sobre el número de sucesores inmediatos.

Notemos que si b es un sucesor inmediato de a se tiene que $niv(b) = niv(a) + 1$, pero en general esta igualdad no implica que b sea un sucesor inmediato de a.

Un elemento $a \in \mathcal{A}$ se dice *terminal* si no tiene sucesores, *simple* si tiene un único sucesor inmediato, y un *punto de ramificación* si tiene más de un sucesor inmediato.

Un árbol se dice *finitamente generado* si todo elemento tiene un número finito de sucesores inmediatos (considerando que los puntos terminales tienen cero sucesores inmediatos).

Un árbol se dice *finito* si tiene un número finito de elementos.

Se verifica fácilmente que el conjunto $\mathbb{N}$ de los números naturales con su orden natural es un ejemplo de un árbol finitamente generado que no es finito.

Intuitivamente, el número $niv(a)$ mide *la distancia* del elemento a al origen a_0. Por eso resulta natural definir la *altura del árbol* $\mathcal{A}$, que indicaremos con $alt(\mathcal{A})$ como el mayor de los niveles de los elementos de $\mathcal{A}$.

$$alt(\mathcal{A}) = sup\{niv(a) \mid a \in \mathcal{A}\}$$

Como $niv(a)$ es un número natural para cada $a \in \mathcal{A}$, resulta que $alt(\mathcal{A}) = k < \infty$ si y sólo si existe un $b \in \mathcal{A}$ tal que $niv(b) = k$ y $niv(a) \leq k$ para todo $a \in \mathcal{A}$, y resulta que $alt(\mathcal{A}) = \infty$ si y sólo si para todo número natural n existe $a_n \in \mathcal{A}$ tal que $niv(a_n) = n$.

Es claro que todo árbol finito tiene altura finita. El conjunto cuyos elementos son el conjunto vacío y los subconjuntos unitarios de $\mathbb{N}$ (es decir, los conjuntos cuyo único elemento es un número natural), ordenado por inclusión, es un ejemplo de árbol infinito de altura finita (más precisamente, de altura uno).

La demostración del lema siguiente es muy fácil, y la dejamos a cargo del lector:

Lema 1 *Todo árbol finitamente generado y de altura finita es finito*

Una *rama de un árbol* $\mathcal{A}$ es un subconjunto totalmente ordenado maximal de $\mathcal{A}$, esto es, una rama de $\mathcal{A}$ es un subconjunto $R \subseteq \mathcal{A}$ que satisface estas dos propiedades:

(R1) R es totalmente ordenado con el orden heredado de $\mathcal{A}$,

(R2) Si S es tal que $R \subseteq S \subseteq \mathcal{A}$, entonces S no es totalmente ordenado con el orden heredado de $\mathcal{A}$.

Si a es un elemento terminal de $\mathcal{A}$, es claro que

$$Rm(a) = \{x \in \mathcal{A} \mid x \leq a\}$$

es una rama de $\mathcal{A}$, que se llama la *rama determinada por* a.

No es difícil demostrar que todas las ramas de un árbol finito $\mathcal{A}$ están determinadas por un elemento terminal de $\mathcal{A}$.

Si $\mathcal{A}$ es un árbol infinito, entonces sus ramas infinitas son los subconjuntos $\{a_0, a_1, \ldots, a_n, \ldots\} \subseteq \mathcal{A}$ tales que $a_n < a_{n+1}$ y $niv(a_n) = n$ para todo $n \in \mathbb{N}$.

El siguiente lema es un resultado clásico de la teoría de árboles, que usamos para probar el Teorema de Compacidad del Cálculo Proposicional.

Lema 2 (D. K önig, 1926) *Todo árbol infinito y finitamente generado tiene al menos una rama infinita.*

Demostración: Sea $\mathcal{A}$ un árbol finitamente generado pero con infinitos elementos. Diremos que un elemento de $\mathcal{A}$ es *bueno* si tiene infinitos sucesores, y que es *malo* en caso contrario. Como $\mathcal{A}$ es infinito, el primer elemento a_0 es bueno. Observemos ahora que todo elemento bueno debe tener al menos un sucesor inmediato bueno. En efecto, como cada elemento tiene sólo un número finito de sucesores inmediatos, si todos ellos fuesen malos el número total de sucesores sería finito, y el elemento no podría ser bueno. Por lo tanto, como a_0 es bueno, debe tener por lo menos un sucesor inmediato bueno. Elijamos uno, llamémoslo a_1. Como a_1 es bueno, debe tener por lo menos un sucesor inmediato bueno. Elijamos uno, llamémoslo a_2. Continuando de esta forma definiremos una sucesión infinita de elementos de $\mathcal{A}$ tal que $a_n < a_{n+1}$ y $niv(a_n) = n$ para todo $n \in \mathbb{N}$,. c.q.d. ■

Observación El hecho de que $\mathcal{A}$ sea finitamente generado es esencial para la validez del Lema de König (Lema 2), como lo muestra el ejemplo de los subconjuntos unitarios de $\mathbb{N}$ ya mencionado.