# Deducción natural

La derivación natural hace uso de varias reglas de inferencia.

| Introduce conectiva | | Elimina conectiva | |
|---|---|---|---|
| Nombre | Regla | Nombre | Regla |
| $\neg I$ | Teorema de la deducción | $\neg E$ | $\neg\neg A \vdash A$ |
| $\Rightarrow I$ | Teorema de la deducción | $\Rightarrow E$ | $A, A \Rightarrow B \vdash B$ |
| $\wedge I$ | $A, B \vdash A \wedge B$ | $\wedge E$ | $A \wedge B \vdash A$<br>$A \wedge B \vdash B$ |
| $\vee I$ | $A \vdash B \vee A$ | $\vee E$ | Teorema de la deducción |
| $\Leftrightarrow I$ | $A \Rightarrow B$,<br>$B \Rightarrow A \vdash A \Rightarrow B$ | $\Leftrightarrow E$ | $A \Leftrightarrow B \vdash A \Rightarrow B$<br>$A \Leftrightarrow B \vdash B \Rightarrow A$ |

Existen dos reglas para cada conectiva lógica, una para introducir la conectiva y otra para eliminarla.

Probar que $P \wedge Q \vdash (P \vee Q) \wedge (Q \vee R)$

| | | |
|---|---|---|
| 1 | $(P \wedge Q)$ | |
| 2 | P | $\wedge E, 1$ |
| 3 | Q | $\wedge E, 1$ |
| 4 | $P \vee R$ | $\vee I, 2$ |
| 5 | $Q \vee R$ | $\vee I, 3$ |
| 6 | $(P \vee Q) \wedge (Q \vee R)$ | $\wedge I, 4, 5$ |

**Implementación del teorema de la deducción.**

Tres reglas de la tabla hacen uso del teorema de deducción. $\neg I$ , $\Rightarrow I$ , $\vee E$ .
En todos los casos hay que crear una derivación subordinada y en el caso de $\vee E$ dos derivaciones subordinadas.
Las subderivaciones pueden ser anidadas.
La derivación que crea la subderivación es la *derivación de llamada*.
Todas las subderivaciones presentan un supuesto adicional que se desecha una vez que la subderivación está terminada.
El siguiente ejemplo prueba el silogismo hipotético.

| | | |
|---|---|---|
| 1 | $P \Rightarrow Q$ | |
| 2 | $Q \Rightarrow R$ | |
| 3 | P | |
| 4 | $P \Rightarrow Q$ | R,1 |
| 5 | Q | $\Rightarrow E, 3, 4$ |
| 6 | $Q \Rightarrow R$ | R,2 |
| 7 | R | $\Rightarrow E, 5, 6$ |
| 8 | $P \Rightarrow R$ | $\Rightarrow I, 3-7$ |

Derivación del Modus Tollens en deducción natural.

| | | |
|---|---|---|
| 1 | $P \Rightarrow Q$ | |
| 2 | $\neg Q$ | |
| 3 | P | |
| 4 | $P \Rightarrow Q$ | R,1 |
| 5 | Q | $\Rightarrow E, 3, 4$ |
| 6 | $\neg Q$ | R,2 |
| 7 | $Q \wedge \neg Q$ | $\wedge I, 5, 6$ |
| 8 | $\neg P$ | $\neg I, 3-7$ |

Derivación por el silogismo disyuntivo.

| | | |
|---|---|---|
| 1 | $P \vee Q$ | |
| 2 | $\neg Q$ | |
| 3 | P | |
| 4 | P | R,3 |
| 5 | Q | |
| 6 | $\neg P$ | |
| 7 | $Q \wedge \neg Q$ | $\wedge I, 2, 5$ |
| 8 | $\neg\neg P$ | $\neg I, 6-7$ |
| 9 | P | $\neg E, 8$ |
| 10 | P | $\vee E, 1, 3-4, 5-9$ |

**Resolución.**

La prueba del teorema de resolución es un método para hacer derivaciones que tiene las siguientes propiedades:
Las únicas expresiones que se permiten en la prueba del teorema de la resolución son disyunción de literales. Una disyunción de literales se llama *cláusula*. Todas las expresiones deben ser cláusulas.
La resolución se ajusta al principio de refutación, que demuestra que la negación de la conclusión es inconsistente con las premisas.
Hay esencialmente una sola regla de inferencia, la resolución.

Ejemplo del Modus Ponens:

Se puede expresar como: $P, P \Rightarrow Q \vDash Q$,
La negación de la conclusión se añade a las premisas: $P, P \Rightarrow Q, \neg Q$,
Se debe demostrar que es imposible encontrar asignación alguna que pueda satisfacer todas esas proposiciones, *es decir que las proposiciones son inconsistentes*. Hay que convertir todas las expresiones en disyunciones: $P, \neg P \vee Q, \neg Q$.
Dos cláusulas se pueden *resolver* si y solo si contienen dos literales complementarios.
Dando origen a una nueva cláusula llamada resolverte.

Si los literales complementarios son $P$ $y$ $\neg P$ se dice que la resolución radica en P.
Las cláusulas que dan origen al resolvente se llaman cláusulas padre.
El resolverte sobre P es la disyunción de todos los literales de las cláusulas padre, excepto $P$ $y$ $\neg P$ se omiten en el resolverte.

Hallar el resolverte de: $P \vee \neg Q \vee R$ y $\neg S \vee Q$.

Las dos cláusulas se pueden resolver sobre Q porque Q es negativa en la primera cláusula y positiva en la segunda.

El resolverte es la disyunción de $P \vee R$ con $\neg S$, por lo tanto,

$P \vee R \vee \neg S$

Se pueden tratar las cláusulas como conjuntos: $C = A \cup B - \{P, \neg P\}$

Para probar el ***Modus Ponens*** se usan las siguientes derivaciones:

| | |
|---|---|
| 1. P | Premisa |
| 2. $\neg P \vee Q$ | Premisa |
| 3. $\neg Q$ | Negación de la conclusión |
| 4. Q | Resolvente de 1, 2 |
| 5. F | Resolvente de 3, 4 |

Para el ***silogismo hipotético*** se tiene:

| | |
|---|---|
| 1. $\neg P \vee Q$ | Premisa |
| 2. $\neg Q \vee R$ | Premisa |
| 3. P | Derivada de la negación de la conclusión |
| 4. $\neg R$ | Derivada de la negación de la conclusión |
| 5. Q | Resolvente de 1, 3 |
| 6. $\neg Q$ | Resolvente de 2, 4 |
| 7. F | Resolvente de 5, 6 |

Se suele utilizar la *resolución unitaria* que consiste en que todas las resoluciones implican al menos una cláusula unitaria.
Otra estrategia es la del conjunto de apoyo:

> Se dividen todas las cláusulas en dos conjuntos, el conjunto de apoyo y el auxiliar.
> El conjunto auxiliar está formado de manera que las expresiones que contiene no son contradictorias.
> Las premisas se usan como conjunto auxiliar inicial.
> Y la negación de la conclusión como el conjunto de apoyo inicial.
> Cada resolución toma al menos una cláusula del conjunto de apoyo.
> Entonces el resolvente se añade al conjunto de apoyo.

La estrategia del conjunto de apoyo es completa, pero la resolución unitaria no.
Si no hay cláusula unitaria se hace imposible la resolución unitaria.
La resolución de las cláusulas unitarias se puede modificar de forma que no se permita que haya cláusulas unitarias para resolver a partir de otras cláusulas.
Esto da lugar a una estrategia de *preferencia unitaria*.
A veces se aplica una regla adicional en la prueba del teorema de la resolución, es la factorización.
La factorización permite eliminar ciertas resoluciones no eficientes a la vez que se mantiene la condición de completo.

# Tablas semánticas, Tableaux

## Introducción

Si desea comprobar que una fórmula es ***consecuencia*** de otras, niéguela e incorpórela a esas otras. Si resulta insatisfacible este nuevo conjunto, efectivamente existía aquella relación de consecuencia.

La ***satisfacibilidad*** de las cláusulas se define como:

- Una cláusula $C = \{l_1, \ldots, l_n\}$ es satisfacible si y sólo si la fórmula $(l_1 \vee \ldots \vee l_n)$ es satisfacible.
- La cláusula vacía {} es insatisfacible.
- Un conjunto de cláusulas $\{C_1, \ldots, C_k\}$ es satisfacible si y sólo si existe una misma interpretación que satisface a cada cláusula del conjunto.

Recuerde que una cláusula es una disyunción implícita. Es, por lo tanto satisfacible si existe un literal que se hace verdadero bajo cierta asignación. En la cláusula vacía no existe ningún literal que pueda hacerla verdadera, luego es insatisfacible. Denotaremos a la cláusula vacía como {} o como $\square$.

Para satisfacer un conjunto de cláusulas basta encontrar una interpretación que satisface al menos un literal en cada cláusula.

## Tableaux

Las tablas semánticas también se denominan tablas analíticas y más comúnmente tableaux.

Es una estrategia deductiva por refutación, las tablas semánticas proporcionan un medio sintáctico de investigar la satisfacibilidad de un conjunto.

Si desea comprobar que una fórmula es consecuencia de otras, niéguela e incorpórela a esas otras. Si resulta insatisfacible este nuevo conjunto, efectivamente existía aquella relación de consecuencia.

**Constatación sintáctica de la insatisfacibilidad**. De nuevo, si el conjunto de partida $\Phi \cup \{\neg\Psi\}$ era insatisfacible, en algún momento del cálculo se evidencia claramente.

Intuitivamente, las fórmulas del conjunto inicial se estructuran como árbol. En particular como un árbol muy lineal, con una sola rama. Existen dos tipos de reglas de inferencia y ambas añaden nodos al árbol: una, linealmente, y otra produciendo una bifurcación binaria. La satisfacibilidad de un árbol se puede comprobar en cualquier momento, considerando la satisfacibilidad de cada rama. Existen indicadores sintácticos que lo explicitan.

La clave de estas operaciones consiste en la preservación de la satisfacibilidad. Cada aplicación de una regla amplía el árbol sin modificar la satisfacibilidad. Si el árbol original, las fórmulas de partida, eran satisfacibles así resultará el árbol ampliado en todo momento. Y si eran insatisfacibles, se llegará a constatar sintácticamente en algún momento.

## Notación uniforme

Considere la fórmula $(p \wedge \neg q)$. Una interpretación la satisface si y sólo si satisface a sus componentes conjuntivos. Es decir,

$(p \wedge \neg q)$ es satisfacible si y sólo si $\{p, \neg q\}$ es satisfacible.

La fórmula propuesta era evidentemente conjuntiva. Otra fórmula como $\neg(\neg p \vee q)$ se puede fácilmente reescribir de forma equivalente como una fórmula conjuntiva. En particular como $(\neg\neg p \wedge \neg q)$. Y se satisfaría, de nuevo, si y sólo si se satisfacen simultáneamente sus componentes.

Dualmente, una fórmula disyuntiva como $(p \vee \neg q)$ se satisface si y sólo si se satisface alguna de sus dos fórmulas inmediatas. Otras fórmulas, no explícitamente disyuntivas como $(p \rightarrow \neg q)$ también se puede reescribir equivalentemente como $(\neg p \vee \neg q)$.

Vamos a fijar el desarrollo que se persigue con esta introducción. Observe la tabla en figura. 1.21.

Cada fórmula α de la izquierda es equivalente a la conjunción de sus componentes $\alpha_1$ y $\alpha_2$. Y cada fórmula $\beta$, a la disyunción de sus componentes $\beta_1$, $\beta_2$.

Tanto en un caso como en otro, las componentes son subfórmulas de la fórmula principal. Y a partir sólo de ellas y de sus negaciones se construye una conjunción o una disyunción equivalente a la fórmula dada.

| Conjuntiva | | | Disyuntiva | | |
|---|---|---|---|---|---|
| $\alpha$ | $\alpha_1$ | $\alpha_2$ | $\beta$ | $\beta_1$ | $\beta_2$ |
| $X \wedge Y$ | $X$ | $Y$ | $\neg(X \wedge Y)$ | $\neg X$ | $\neg Y$ |
| $\neg(X \vee Y)$ | $\neg X$ | $\neg Y$ | $X \vee Y$ | $X$ | $Y$ |
| $\neg(X \rightarrow Y)$ | $X$ | $\neg Y$ | $X \rightarrow Y$ | $\neg X$ | $Y$ |
| $\neg(X \leftarrow Y)$ | $\neg X$ | $Y$ | $X \leftarrow Y$ | $X$ | $\neg Y$ |
| $\neg(X \uparrow Y)$ | $X$ | $Y$ | $X \uparrow Y$ | $\neg X$ | $\neg Y$ |
| $X \downarrow Y$ | $\neg X$ | $\neg Y$ | $\neg(X \downarrow Y)$ | $X$ | $Y$ |
| $X \nrightarrow Y$ | $X$ | $\neg Y$ | $\neg(X \nrightarrow Y)$ | $\neg X$ | $Y$ |
| $X \nleftarrow Y$ | $\neg X$ | $Y$ | $\neg(X \nleftarrow Y)$ | $X$ | $\neg Y$ |

En las tres primeras líneas se utilizan las conectivas básicas que se fijaron en el alfabeto. Las cinco líneas restantes utilizan otras conectivas binarias que se podían haber incorporado igualmente al alfabeto. Si se asume esta notación, incluso utilizando todas estas conectivas, cada fórmulas es simplemente de tipo $\alpha$ o de tipo $\beta$. Y, recursivamente, cada una de sus subfórmulas es de uno de estos dos tipos.

Ni el bicondicional ni su negación, la disyunción exclusiva, se pueden escribir como una conjunción o disyunción de sus subfórmulas negadas o no. Por tanto, no se considerarán conectivas primarias del lenguaje sino abreviaturas.

**Tableaux: definición**

**Ejemplo 1.68.** Observe la figura 1.22 considerando que inicialmente sólo consta del nodo 1.

Esto es así porque el conjunto de fórmulas iniciales analizado consta de una única fórmula: $\neg(p \rightarrow (q \wedge r))$. Luego se construye un árbol A que consta de ese único nodo.

Puede comprobarse que la fórmula analizada es de tipo $\alpha$, en concreto $\neg(X \rightarrow Y)$. Expandamos el árbol $A$ con dos nodos más: $\alpha_1$ (nodo 2) y $\alpha_2$ (nodo 3). El resultado es otro árbol $A'$: una rama compuesta por tres fórmulas que son satisfacibles por las mismas interpretaciones.

La fórmula del nodo 3 puede aún expandirse. Es una fórmula de tipo $\beta$ (disyuntiva), y por tanto para su satisfación basta que una de las dos componentes se satisfaga. Representemos ese hecho bifurcando el árbol en ese punto terminal. El resultado es el árbol $A''$ final que se observa en la figura (fig. 1.22).

Con las consideraciones anteriores, la fórmula inicial se satisface si y sólo si se satisfacen todas las de la rama 1-4 o las de la rama 1-5. Cuando se recorren, tanto una como otra, no hay señales evidentes de que tales fórmulas no puedan ser satisfechas a la vez. De hecho, para que la rama 1-4 se satisfaga basta considerar $q$ falsa y $p$ verdadero. Y en la rama 1-5, $r$ falso y $p$ verdadero.

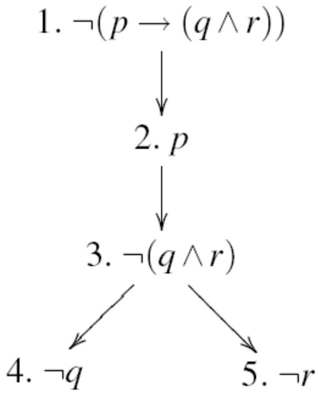


Figura 1.22: Tableau de $\neg(p \to (q \wedge r))$

**Ejemplo 1.69** La figura 1.23 inicialmente constaba sólo de los nodos 1 y 2. Se pretende analizar la satisfacibilidad del conjunto formado por esas dos fórmulas.
Nos situaremos en el nodo 2, por ser el extremo actual de ese árbol. Y aplicaremos allí la expansión de una de esas dos fórmulas. En concreto, escogemos expandir la fórmula 2, podía haber sido la 1. La fórmula 2 es de tipo $\alpha$ y produce los nodos 3 y 4.
Nos situamos en el nodo 4 y expandimos la fórmula 1, que es de tipo $\beta$. Como la satisfacción de esta fórmula puede venir por un lado o por otro, se produce la bifurcación de los nodos 5 y 6. En este punto, las fórmulas iniciales son simultáneamente satisfacibles si y sólo si lo son todas las fórmulas de la rama 1-5 o todas las fórmulas de la rama 1-6.
Recorriendo las fórmulas de la rama 1-5 se observa que una de las fórmulas es $p$ mientras otra es $\neg p$. Luego la satisfacibilidad simultánea de todas las fórmulas de esa rama queda descartada, se marca su extremo. Comprobamos la otra rama, el conjunto inicial es satisfacible si y sólo si lo es esta rama.
Como aún se pueden expandir más fórmulas, nos situamos en 6 y lo hacemos. En concreto se expande la propia fórmula 6, que produce los nodos 7 y 8 por ser de tipo conjuntivo. De nuevo, recorriendo esta rama se encuentra uno con $r$ y $\neg r$ y la rama queda cerrada.
El conjunto formado por las dos fórmulas iniciales no es satisfacible.

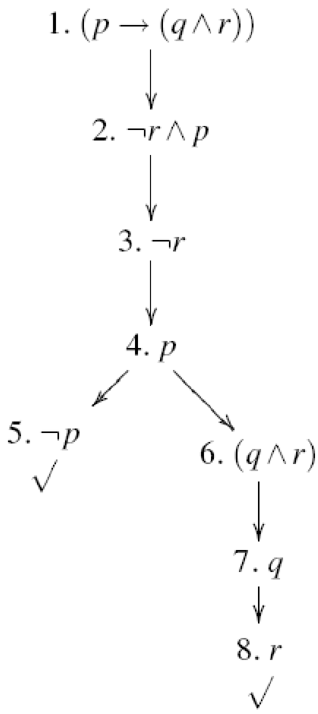


Figura 1.23: Tableau de $\{p \to (q \wedge r), \neg r \wedge p\}$

**Definición 1.70.** Tableau de un conjunto de fórmulas. Sea $\{\varphi_1, \dots, \varphi_n\}$ un conjunto de fórmulas. Entonces el árbol formado por la única rama $\varphi_1 \dots \varphi_n$ es un tableau A para este conjunto.

Si A es un tableau para ese conjunto y $A'$ resulta de aplicar a A alguna de las reglas de expansión de tableaux, entonces $A'$ es un tableau para ese conjunto.

Falta precisar cuáles son las reglas de expansión, dónde y cómo se puede expandir un árbol y si existe un final determinado para este proceso.

Las reglas de expansión son las que se muestran en la tabla 2.5. Dado un árbol, se escoge una rama cualquiera del mismo y una fórmula de esa rama que pueda expandirse, que no sea un literal. La expansión se produce en el terminal de esa rama: una bifurcación si la fórmula escogida era disyuntiva o la secuencia de dos nodos si era conjuntiva.

**Definición 1.71.** Tableau cerrado Una rama se dice cerrada si ocurren en ella tanto una fórmula X como una fórmula ¬X, o si ocurre la fórmula $\perp$. Está atómicamente cerrado si la fórmula X es atómica o si ocurre $\perp$.

Un árbol se ha cerrado si se han cerrado todas sus ramas. Cuando todos los cierres son atómicos se dice que el árbol se ha cerrado de forma atómica.

**Tabla 2.5 Reglas de expansión de un Tableau.**

| | |
|---|---|
| Conectivas monarias | $\dfrac{\neg\neg X}{X}$ $\quad \dfrac{\neg\top}{\perp}$ $\quad \dfrac{\neg\perp}{\top}$ |
| Conectivas binarias | $\dfrac{\alpha}{\alpha_1}$ $\quad \dfrac{\beta}{\beta_1 \mid \beta_2}$ |
| Cuantificador universal | $\dfrac{\gamma}{\gamma(t)}$ (para cualquier término *t* cerrado de $L^{par}$ ) |
| Cuantificador existencial | $\dfrac{\delta}{\delta(p)}$ (para cualquier parámetro *p* nuevo) |

Observe que las reglas de expansión son no deterministas. Se puede escoger la rama y la fórmula siguientes que se van a expandir. Esta opción obliga a plantearse si existen estrategias de elección, heurísticas, más eficientes que otras. En efecto, por regla general, trate de expandir todas las fórmulas $\alpha$ primero.

Previa a esta preocupación por la complejidad, conviene fijar si el proceso para en algún momento.

No hay nada en las definiciones que impida expandir la misma fórmula α una y otra vez sobre una rama, puesto que no se elimina. Afortunadamente, para la lógica proposicional abordada, basta utilizar una única vez cada fórmula para garantizar el cierre del tableau (si es que debe cerrarse). No ocurría así en la Resolución. El sistema descrito es consistente y completo en los mismo términos con que se enunció para Resolución: un conjunto es insatisfacible si y sólo si existe un tableau cerrado del mismo.

## Tablas semánticas

## Fórmulas proposicionales

Recordemos las propiedades básicas de los tableaux para fórmulas proposicionales:

1. se utilizan para decidir la satisfacibilidad de un conjunto de fórmulas; indirectamente, para decidir la relación de consecuencia entre una fórmula y un conjunto: niegue aquélla e incorpórela al conjunto inicial analizado

2. se construye un primer árbol, con una sola rama, que consta de tantos nodos como fórmulas haya en el conjunto inicial

3. las ramas se pueden bifurcar o ampliar linealmente; los nodos añadidos son subfórmulas adecuadas negadas o no de una fórmula en esa rama

4. una rama es satisfacible si lo es el conjunto de todas sus fórmulas; obviamente, si entre ellas se encuentran tanto una fórmula como su negación, la rama es insatisfacible

5. un árbol es satisfacible si lo es alguna de sus ramas

6. el árbol inicial es tan satisfacible como los sucesivos árboles ampliados; así, si se detecta que alguno de ellos es insatisfacible, también lo era el conjunto inicial de fórmulas

**Ejemplo 2.71** Observe el árbol de la figura 2.7. Muestra que, de la hipótesis $p \to (q \wedge r)$, es consecuencia lógica $\neg r \to \neg p$. Más apropiadamente, que esto segundo se deduce de aquello primero.

Pero en este sistema consistente todo lo que se deduce es consecuencia.

[Nodos 1-2] El árbol inicial lo constituyen el nodo 1 (única hipótesis) y el nodo 2 (negación de la supuesta conclusión). Situados en el nodo más bajo de esta rama (el 2), tratamos de ampliarla descomponiendo adecuadamente alguna de las fórmulas de esa rama. Estratégicamente, se recomienda descomponer primero las fórmulas α (conjuntivas) y después las β (disyuntivas).

[Nodos 3-5] Optamos por utilizar la fórmula 2 (α, conjuntiva): da lugar a una expansión lineal de esa rama (nodos 3 y 4). Y seguimos expandiendo: estamos en el nodo 4 y podemos expandir esa rama utilizando cualquiera de las fórmulas de la misma. Entre las diversas opciones, se opta por utilizar la propia fórmula 4: la 'doble negación' se puede expandir como 'sin negación' (nodo 5).

[Nodos 6-7] En el nodo 5 se decide expandir la fórmula 1 (β, disyuntiva), que bifurca la rama (nodos 6 y 7). La rama del nodo 6 se puede cerrar inmediatamente: entre las fórmulas de esa rama (1-6) hay tanto una fórmula (la 6) como su negación (en 5).

[Nodos 8-9] En la rama del nodo 7 no se observa aún este tipo de contradicción. Pero todavía se pueden expandir fórmulas de esa rama; en concreto, la propia fórmula 7 (de tipo α). Se obtienen los nodos 8 y 9. Y el nodo 9 es la negación de la fórmula del nodo 3, en esa misma rama desde la raíz.

El árbol se ha cerrado. Luego el conjunto de partida ($\varphi$ en nodo 1, $\neg\psi$ en nodo 2) era insatisfacible.

Así que se ha demostrado que $\varphi \vdash \psi$, es decir, que $\varphi \vDash \psi$.

$$
\begin{tikzcd}
& 1.\ (p \to (q \wedge r)) \arrow[d] & \\
& 2.\ \neg(\neg r \to \neg p) \arrow[d, "\alpha_1,2"] & \\
& 3.\ \neg r \arrow[d, "\alpha_2,2"] & \\
& 4.\ \neg\neg p \arrow[d, "\neg\neg,4"] & \\
& 5.\ p \arrow[dl, "\beta_1,1"'] \arrow[dr, "\beta_2,1"] & \\
6.\ \neg p & & 7.\ (q \wedge r) \arrow[dd, "\alpha_1,7"] \\
\perp,5 & & \\
& & 8.\ q \arrow[d, "\alpha_2,8"] \\
& & 9.\ r \\
& & \perp,3
\end{tikzcd}
$$

Figura 2.7: Tableau que confirma que $p \to (q \wedge r) \vdash \neg r \to \neg p$

**Ejemplo 2.72** El árbol de la figura 2.8 es igual al del ejemplo previo figura 2.7. Compare las fórmulas de uno y otro: todas las fórmulas del segundo ejemplo se obtienen, por sustitución uniforme, de las del primero. Observe cómo se produce el cierre del segundo ejemplo: donde antes se cerraban en fórmulas atómicas, ahora se pueden cerrar en sus fórmulas sustituyentes.

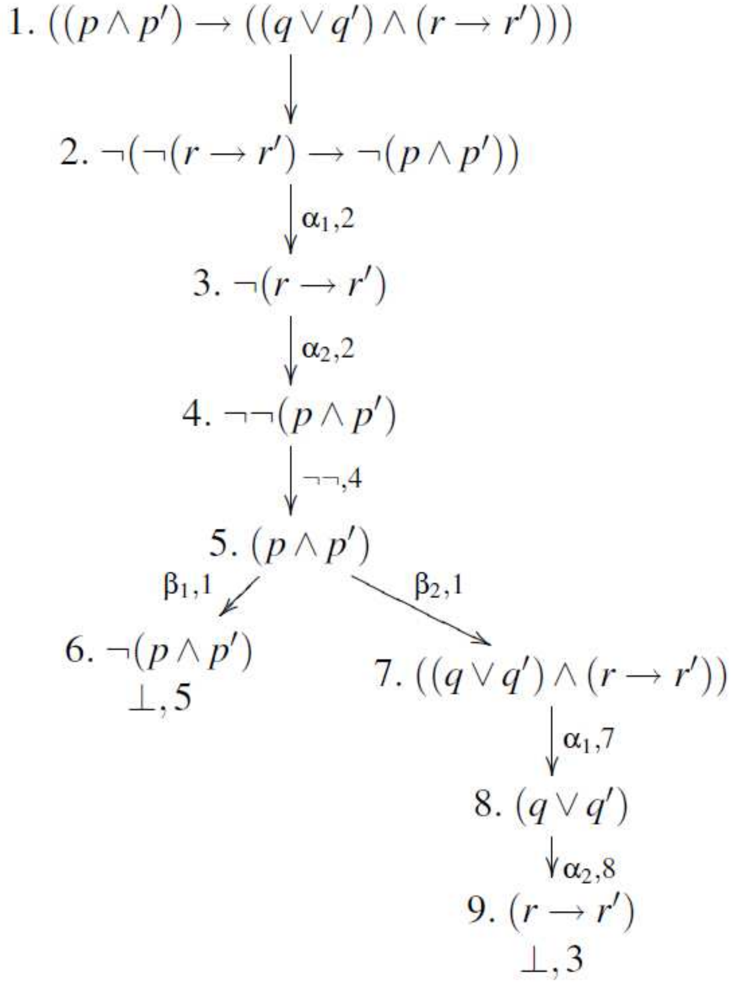


Figura 2.8: $(p \wedge p') \rightarrow ((q \vee q') \wedge (r \rightarrow r')) \vdash \neg(r \rightarrow r') \rightarrow \neg(p \wedge p')$

El árbol de la figura 2.9 es ya una primera incursión en lógica de predicados. Sin embargo, no se van a necesitar aquí reglas adicionales: también es una instancia por sustitución del primer árbol considerado. Compruebe qué fórmula ha sustituido a qué proposición.

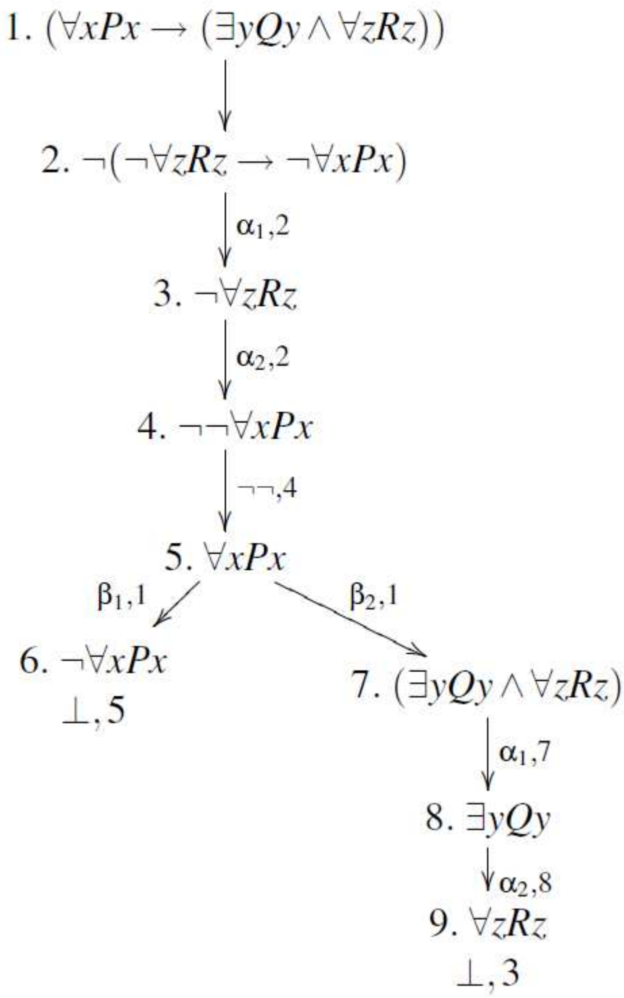


Figura 2.9: $\forall xPx \to (\exists yQy \wedge \forall zRz) \vdash \neg\forall zRz \to \neg\forall xPx$

## Notación uniforme

En lógica de predicados, no todas las relaciones de consecuencia entre fórmulas tienen raíz puramente conectiva proposicional. Por eso se requieren nuevas reglas de inferencia. En concreto, en este sistema, se añadirán dos nuevas reglas:

1. una, para manipular fórmulas universalmente cuantificadas (tipo $\gamma$)

2. otra, para manipular fórmulas existencialmente cuantificadas (tipo $\delta$)

Así, toda fórmula en este sistema es intrínsecamente conjuntiva ($\alpha$), disyuntiva ($\beta$), universal ($\gamma$), existencial ($\delta$) o una doble negación (ó $\perp$ ó $\top$).

La tabla 2.4 resume las fórmulas que se consideran intrínsecamente universales ($\gamma$) y existenciales ($\delta$), junto a su expansión. Todas ellas producen una expansión del árbol en un sólo nodo. No producen, por tanto, bifurcación del árbol.

**Tabla 2.4 Notificación uniforme, fórmulas cuantificadotas.**

| Universales | | Existenciales | |
|---|---|---|---|
| $\gamma$ | $\gamma(t)$ | $\delta$ | $\delta(a)$ |
| $\forall xX$ | $X(t)$ | $\exists xX$ | $X(a)$ |
| $\neg\exists xX$ | $X(t)$ | $\neg\forall xX$ | $X(a)$ |

**Ejemplo 2.73** Considere la fórmula:

$\neg\forall x(\forall z\neg(Pz \to Qx) \to \exists wRwa)$

es de tipo existencial (δ). Su subfórmula inmediata:

$(\forall z\neg(Pz \to Qx) \to \exists wRwa)$

es de tipo (β). Sus subfórmulas inmediatas:

$\forall z\neg(Pz \to Qx)$, $\exists wRwa$

son, respectivamente, (γ) y (δ). Y la subfórmula:

$\neg(Pz \to Qx)$

es de tipo (α).

## Reglas de expansión γ y δ

En ambos tipos de fórmulas la expansión aporta un sólo nodo. En concreto, la subfórmula inmediata:

La fórmula que resta cuando se omite el cuantificador principal. Más precisamente, una instancia por sustitución de esta subfórmula. El párrafo siguiente expone cuáles pueden ser las cadenas sustituyentes.

Más adelante se determinará qué se debe sustituir y con qué restricciones.

**Parámetros**. Cada lenguaje de primer orden fija sus propias constantes y funciones. Si se pretende que el lenguaje sirva, por ejemplo, para razonar sobre números naturales, debe incluir una constante (que se asignará al 0) y una función (la función sucesor).

Las demostraciones sobre fórmulas de un lenguaje suelen requerir, como herramienta, el uso de constantes auxiliares. A estas constantes adicionales se les denomina ’parámetros’. Así, las demostraciones sobre fórmulas de un cierto lenguaje L se describen usando fórmulas del lenguaje $L^{par}$, extensión del L por adición de estas constantes.

Las cadenas sustituyentes en la expansión serán, según los casos, parámetros o términos cerrados, términos de $L^{par}$ que no incluyen variables.

**Regla de expansión δ**. Las fórmulas δ son del tipo $\exists x\phi$ ó $\neg\forall x\phi$. Su expansión es un único nodo de la forma $\phi(x/p)$ o $\neg\phi(x/p)$ respectivamente, donde todas las apariciones libres en $\phi$ de la variable del cuantificador se han sustituido por el mismo parámetro *p*. Este parámetro, esta constante auxiliar, debe ser nueva en el árbol: no puede figurar en ninguna fórmula previa (realmente, basta que sea nueva en la rama).

De una fórmula como $\exists xPx$ no tiene por qué ser consecuencia $P_a$. Sin embargo, si $\exists xPx$ es satisfacible, también lo será $P_a$. Y estamos trabajando en un sistema que trata de decidir la satisfacibilidad de un conjunto de fórmulas.

Cada instanciación debe hacerse sobre una constante nueva. De lo contrario, esta constante tendría unas propiedades (fijadas en otras fórmulas, donde aparece) que pueden modificar (innecesariamente) la decisión final sobre la satisfabilidad del conjunto.

Este cuidado se mantiene en toda manipulación de fórmulas de primer orden. Si no se considera, se puede llegar a afirmar (erróneamente) que 'siempre que hay alguien que tiene la propiedad $P$ y hay alguien que tiene la propiedad $Q$, entonces alguien tiene ambas propiedades'. Un razonamiento erróneo en deducción natural sería: "puesto que alguien tiene la propiedad P, llamémosle a, luego $P_a$. Puesto que alguien tienen la propiedad $Q$, llamémosle a, luego $Q_a$. Entonces, $P_a \wedge Q_a$. Luego alguien tiene (a la vez) la propiedad $P$ y la $Q$".

**Regla de expansión γ.** Las fórmulas γ son del tipo $\forall x\phi$ ó $\neg\exists x\phi$. Su expansión es un único nodo de la forma $\phi(x/p)$ o $\neg\phi(x/p)$ respectivamente, donde todas las apariciones libres de la variable del cuantificador se han sustituido por el mismo término $t$. Este término debe ser cerrado: no debe incluir variables, sólo constantes y funciones de $L$ o constantes auxiliares.

La particularización de una fórmula universal no requiere de ningún cuidado especial: lo que es verdadero para todo elemento del universo lo será para el que represente al término $t$.

Todas las reglas de expansión se recogen en la tabla 2.5.

**Tabla 2.5 Reglas de expansión de un Tableau.**

| | |
|---|---|
| Conectivas monarias | $\frac{\neg\neg X}{X}$ $\frac{\neg\top}{\bot}$ $\frac{\neg\bot}{\top}$ |
| Conectivas binarias | $\frac{\alpha}{\alpha_1}$ $\frac{\beta}{\beta_1 \mid \beta_2}$ |
| Cuantificador universal | $\frac{\gamma}{\gamma(t)}$ (para cualquier término $t$ cerrado de $L^{par}$) |
| Cuantificador existencial | $\frac{\delta}{\delta(p)}$ (para cualquier parámetro $p$ nuevo) |

### 2.4.4 Ejemplos

**Ejemplo 2.74.** El árbol de la figura 2.10 muestra que:

$\forall xPx \vee \exists yQy \vdash \exists y\forall x(Px \vee Qy)$

[Nodos 1-2]Respectivamente, hipótesis y negación de la supuesta conclusión.

[Nodos 3-4] Expansión (β) de la fórmula 1. Estratégicamente, siempre es preferible expandir primero las fórmulas proposicionales (α y β), luego las existenciales (δ) y finalmente las universales (γ) para intentar cerrar.

[Nodo 5] Expansión de la fórmula 2 (universal, γ), porque no existía ninguna fórmula existencial en la rama 1-3. Se puede instanciar $y$ por cualquier parámetro.

[Nodo 6] Expansión de la fórmula 5 (existencial, δ). El parámetro de la instanciación debe ser nuevo: b.

[Nodos 7-8]Expansión de la fórmula 6 (α)

[Nodo 9]Aprovechando que ya existe ¬Pb en el nodo 8, expandiremos la fórmula 3 (universal).

Como se puede escoger cualquier término, elegiremos instanciarla para b, cerrando esta rama.

[Nodos 10-14] Se procede de manera análoga (indicada en el árbol). Tan sólo merece reseñar la elección de parámetros. El de la línea 10 debía ser un parámetro nuevo en la rama. Se ha escogido incluso nuevo en el árbol (c).

[Nodos 11-14]El parámetro de la línea 11 proviene de una sustitución universal: podía ser cualquiera.

Como ya existía $Q_c$ previamente, se escoge asimismo $Q_c$. El parámetro de la línea 12 debía ser nuevo: d. La elección correcta de la instancia universal en la línea 11 acaba favoreciendo el cierre de esta última rama en 14.

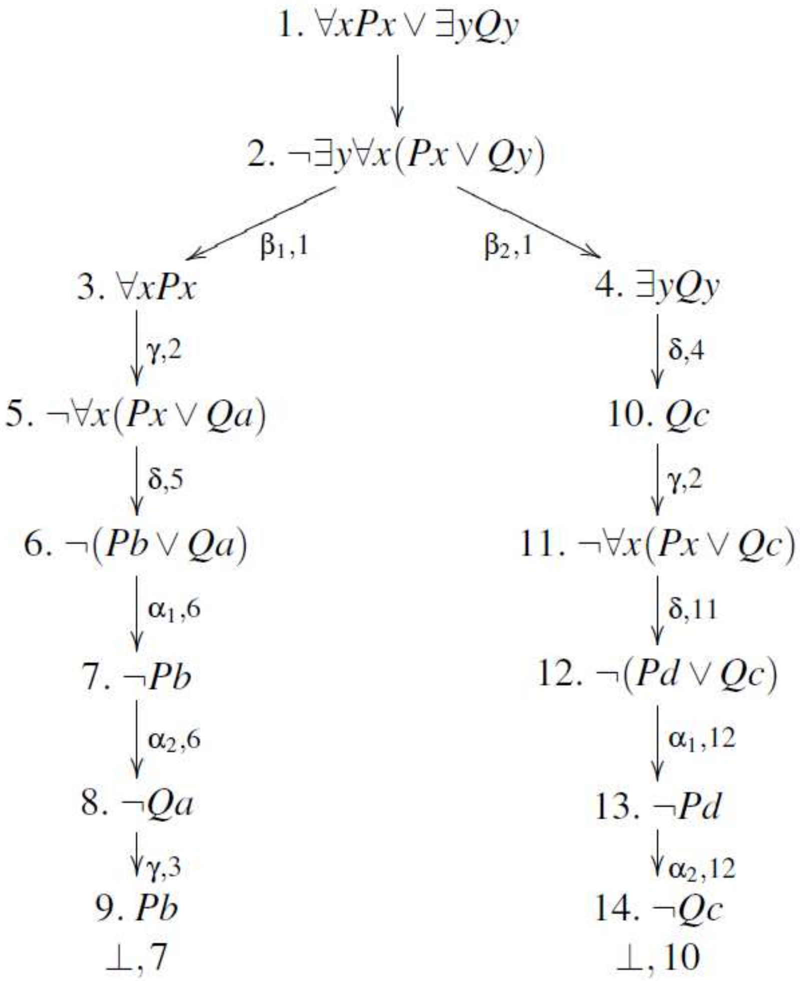


Figura 2.10: Tableau que confirma que $\forall x Px \lor \exists y Qy \vdash \exists y \forall x(Px \lor Qy)$

**Ejemplo 2.75** La figura 2.11 confirma que

$\forall x \exists y(Rxy \rightarrow Qy), \forall x \forall y Rxy \vdash \exists z Qz$

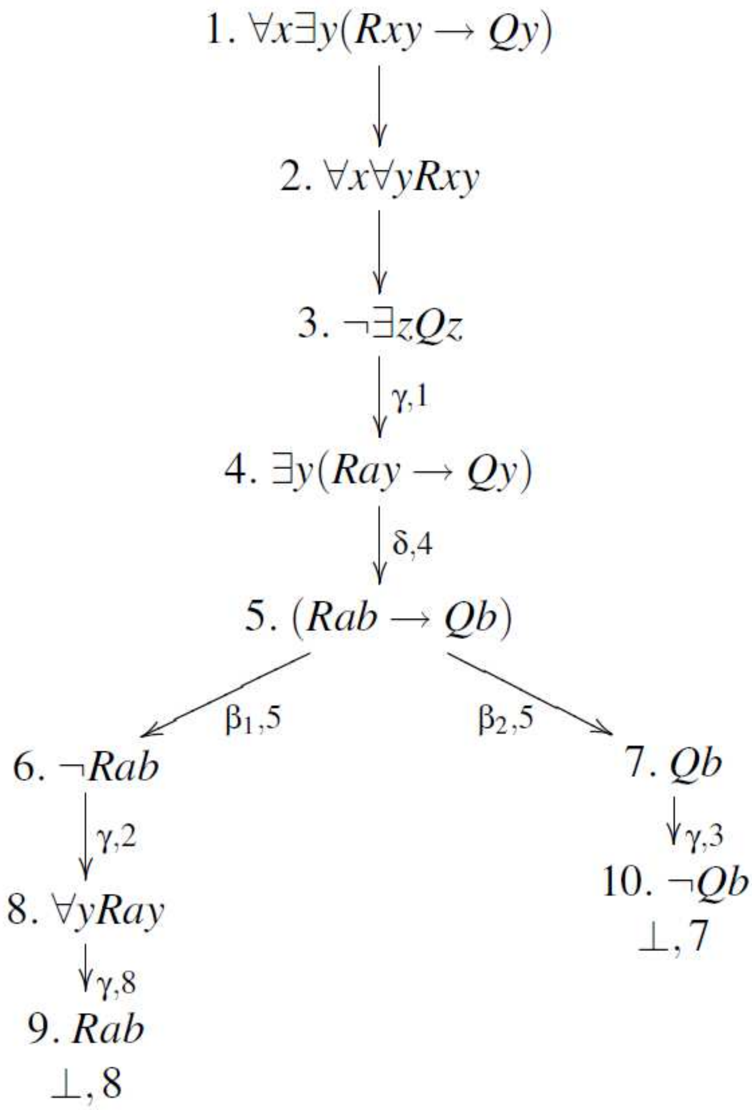


Figura 2.11: Tableau que confirma que $\forall x \exists y(Rxy \rightarrow Qy), \forall x \forall y Rxy \vdash \exists z Qz$