Apuntes de
Lógica Matemática

Saúl Rodríguez Martín
Miguel Muñoz Pérez
Eduardo González Vaquero

7 de julio de 2020

Apuntes basados en las clases de Luis Fernando Llana Díaz de la asignatura Lógica Matemática del grado en Matemáticas de la Universidad Complutense de Madrid. Redactados durante el curso académico 2019-2020.
Se agradece el envío de erratas a saulro01@ucm.es.
Gracias a Alberto Acosta por señalar numerosas erratas.

Índice general

1 Lógica proposicional

1.1. Introducción

Antes de comenzar con definiciones rigurosas, conviene dar una breve explicación sobre la motivación de la lógica proposicional, a la que dedicaremos este capítulo. Un aspecto común de los lenguajes naturales (como el español o el chino) es la ambigüedad de ciertas expresiones, lo cual dificulta las intenciones usuales de la ciencia, a saber, la precisión en la comunicación y la distinción clara de aquello que se comunica. De aquí nace una necesidad evidente, que es cubierta por los lenguajes artificiales y, entre ellos, la lógica formal.

Debemos aclarar entonces cuál es nuestro objeto de estudio. Fijémonos en lo siguiente: existen ciertos elementos del lenguaje que no cambian si los traducimos a otro idioma, por ejemplo. Nos referimos a lo que se denomina como *conectivas lógicas*, es decir, 'no' (negación), 'o' (disyunción), 'y' (conjunción), 'si... entonces' (implicación material) y 'si y solo si' (equivalencia). Obviamente, podemos usar múltiples expresiones distintas para referirnos a cada una de ellas, pero su función en el lenguaje natural no cambia. Por esto mismo podemos formalizarlas y estudiarlas desde el punto de vista matemático.

Ahora bien, a fin de sistematizar nuestro lenguaje, introducimos una serie de símbolos que se referirán a las entidades correspondientes del lenguaje natural. Conviene puntualizar de nuevo que estas traducciones, es decir, estas 'asignaciones de significados' *no* son únicas. Esto lo veremos más adelante. Entonces tenemos: $\neg$ (negación), $\vee$ (disyunción), $\wedge$ (conjunción), $\rightarrow$ (implicación material) y $\leftrightarrow$ (equivalencia).

Los otros elementos básicos que constituirán nuestro lenguaje formal son las *proposiciones*. Con el fin de no entrar en cuestiones semánticas que exceden nuestro estudio, establecemos la siguiente:

Definición 1.1. Una proposición es un enunciado que puede ser verdadero o falso.

Como con las conectivas, introducimos una serie de *símbolos de proposición* para referirnos formalmente a las proposiciones. Denotamos por SP al conjunto de todos estos símbolos.

Así, el enunciado 'Todos los solteros son hombres no casados' es una proposición, mientras que '¿Va a llover mañana?' no lo es. Por otro lado, 'César conquistó las Galias y combatió contra Vercingétorix' admite una descomposición como conjunción de dos proposiciones. Evidentemente, esta reducción a proposiciones más simples tiene que parar eventualmente; así llegamos a lo que denominamos *proposiciones atómicas*. Esto nos permitirá, una vez determinada la verdad o falsedad de estas proposiciones atómicas, hallar la de todas las proposiciones que se compongan de ellas mediante la aplicación de las conectivas lógicas (antiguamente llamadas *proposiciones moleculares*).

Las proposiciones atómicas se representan entonces por los elementos de SP. Añadimos, por motivos que serán aclarados más adelante, dos elementos $\top$ y $\bot$, que se refieren a la *proposición idénticamente verdadera* y a la *proposición idénticamente falsa*, respectivamente[1]. Es decir,

Definición 1.2. Definimos el conjunto de las proposiciones atómicas como:

$$AT := SP \cup \{\top, \bot\}$$

Definición 1.3. (Alfabeto) Dado un conjunto de símbolos de proposiciones SP, se denomina alfabeto a:

$$A_{SP} = SP \cup \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,), \top, \bot\},$$

obtenido añadiendo a SP los símbolos $\top$ y $\bot$, las conectivas lógicas $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, y los paréntesis, $(,)$.

Definición 1.4. (Cierre de Kleene) Dado un conjunto de símbolos A, definimos el cierre de Kleene de A, A^*, como el conjunto de las concatenaciones de elementos de A junto con ϵ, el *espacio vacío* o en blanco.

Ejemplo 1.5. Dado $A = \{a, b\}$, le corresponde

$$A^* = \{\epsilon, a, b, ab, ba, aab, baa, aba, ...\}$$

Ejemplo 1.6. A^*_{SP} será el conjunto de todas las concatenaciones de elementos de A_{SP}, junto con ϵ.

Una vez que disponemos de los elementos básicos de la sintaxis del lenguaje y sus sucesivas combinaciones, nos interesa definir las expresiones que están 'bien formadas', es decir, queremos excluir de nuestro lenguaje concatenaciones de símbolos como '$)) \leftrightarrow \varphi$'.

Definición 1.7. (Proposiciones posibles) Se define el conjunto de proposiciones posibles, $PROP_{SP}$, como la intersección de todos los subconjuntos de A^*_{SP} que verifican:

[1] En realidad, como veremos, al igual que $\neg, \wedge, \vee, \rightarrow$ y $\leftrightarrow$ son conectivas binarias, $\top$ y $\bot$ pueden considerarse conectivas 0-arias.

1. $AT = SP \cup \{\top, \bot\} \subseteq PROP_{SP}$.
2. Si $\phi \in PROP_{SP}$, entonces $(\neg\phi) \in PROP_{SP}$.
3. Si $\phi_1, \phi_2 \in PROP_{SP}$ y $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$, entonces $(\phi_1 \square\, \phi_2) \in PROP_{SP}$.

Es fácil comprobar que $PROP_{SP}$ también cumple estas tres propiedades, por tanto decimos que es el mínimo subconjunto que las verifica. También podemos obtener $PROP_{SP}$ de forma constructiva:

Proposición 1.8. *Se define $P_0 := AT$ y para cada $n \in \mathbb{N}$:*

$$P_n := P_{n-1} \cup \{(\neg\phi) : \phi \in P_{n-1}\} \cup \{(\phi_1 \square\, \phi_2) : \phi_1, \phi_2 \in P_{n-1}, \square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}\}.$$

Entonces $P := \bigcup_{n=0}^{\infty} P_n$ coincide con $PROP_{SP}$.

Demostración. Comenzamos comprobando $P \subseteq PROP_{SP}$. Para ello, comprobamos por inducción que $P_n \subseteq PROP_{SP}$ para todo $n \in \mathbb{N}$.

Está claro que $P_0 \subseteq PROP_{SP}$, ya que $PROP_{SP}$ cumple la propiedad 1. Ahora, suponiendo que $P_n \subseteq PROP_{SP}$, por las propiedades 2 y 3, $\{(\neg\phi) : \phi \in P_{n-1}\} \cup \{(\phi_1 \square\, \phi_2) : \phi_1, \phi_2 \in P_{n-1}, \square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}\}$ también estarán contenidos en $PROP_{SP}$. Por tanto, $P_{n+1} \subseteq PROP_{SP}$, y hemos acabado la inducción.

Veamos ahora que $PROP_{SP} \subseteq P$. Para ello nos basta comprobar que P cumple las tres propiedades de la definición 1.7. La propiedad 1 es directa, pasamos a comprobar las propiedades 2 y 3.

Dado $\varphi \in P$, existe $n \in \mathbb{N} \cup \{0\}$ tal que $\varphi \in P_n$ luego $(\neg\varphi) \in P_{n+1}$ por lo que $(\neg\varphi) \in P$.

Sean $\varphi_1, \varphi_2 \in P$ y sea $\square \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. Tomamos $n \in \mathbb{N}$ tal que $\varphi_1, \varphi_2 \in P_n$ (para cada uno existe al menos un natural: basta tomar el mayor). Luego $(\phi_1 \square\, \phi_2) \in P_{n+1}$ y por lo tanto también pertenece a P.

$\square$

A partir de ahora, usaremos para mayor brevedad el símbolo $\square$ como intercambiable por $\wedge, \vee, \rightarrow$ o $\leftrightarrow$ a no ser que se indique lo contrario.

Básicamente, el resultado anterior nos dice que podemos construir cualquier proposición partiendo de proposiciones atómicas y creando proposiciones más complicadas a partir de ellas usando las conectivas. Esta forma de verlo da lugar al siguiente concepto:

Definición 1.9. (Secuencia de formación) Definimos una *secuencia de formación* para cierto $\varphi \in A_{SP}^*$ como una sucesión finita $\langle\varphi_1, \ldots, \varphi_n\rangle$ de elementos de A_{SP}^* tal que $\varphi_n = \varphi$ y para cada $i \leq n$, se da alguna de estas tres opciones:

- $\varphi_i \in AT$.
- $\varphi_i = (\neg\varphi_j)$, para cierto $j < i$.
- $\varphi_i = (\varphi_j \Box \varphi_k)$, para ciertos $j, k < i$.

No es difícil comprobar que toda proposición puede construirse mediante una secuencia de formación, y recíprocamente si φ tiene una secuencia de formación, entonces $\varphi \in PROP_{SP}$.

1.2. Inducción estructural y recursión

Análogamente al método de inducción en los naturales, el método de inducción estructural nos servirá para demostrar que todas las fórmulas cumplen una *propiedad*[2] en dos pasos: demostrar que todas las proposiciones atómicas cumplen dicha propiedad, y demostrar que si dos proposiciones φ_1, φ_2 cumplen una propiedad, entonces $(\neg\varphi_1)$ y $(\varphi_1 \Box \varphi_2)$ también la cumplen.

Usaremos la notación $P(\varphi)$ para decir que la fórmula φ cumple la propiedad P.

Proposición 1.10. *(Inducción estructural) Supongamos que queremos probar $P(\varphi)$ para todo $\varphi \in PROP_{SP}$ (siendo P una propiedad cualquiera). Entonces basta comprobar que:*

1. *$P(\varphi)$ para toda $\varphi \in AT$.*
2. *Si $P(\varphi)$ entonces $P((\neg\varphi))$.*
3. *Si $P(\varphi_1)$ y $P(\varphi_2)$ entonces $P((\varphi_1 \Box \varphi_2))$.*

Demostración. Si una propiedad P cumple 1, 2 y 3, entonces el conjunto de proposiciones que cumplen P cumple las tres propiedades de 1.7, por tanto contiene a $PROP_{SP}$.

$\Box$

Ejemplo 1.11. Definamos $P(\varphi)$ como la propiedad de $|\varphi|_) = |\varphi|_($, siendo $|\varphi|_)$ el número de símbolos) en la fórmula φ y $|\varphi|_($ el número de ([3]. Vamos a comprobar por inducción estructural que todas las proposiciones cumplen esta propiedad:

1. Para cualquier $\varphi \in AT$ se verifica trivialmente $|\varphi|_) = |\varphi|_(= 0$, luego $P(\varphi)$.

[2] No hemos definido rigurosamente el concepto de "propiedad", pero esencialmente significa algo que las proposiciones pueden cumplir o no. Una forma rigurosa de definir propiedad consiste en definirla como un subconjunto $P \subseteq PROP_{SP}$, de modo que decimos que una proposición cumple la propiedad P si y solo si pertenece a P.

[3] En general, siendo $*$ un símbolo, podemos denotar $|\varphi|_*$ al número de símbolos $*$ en la fórmula φ.

2. Supongamos $P(\varphi)$ con $\varphi \in PROP_{SP}$, entonces $|\varphi|_{(} = |\varphi|_{)}$ y por tanto:

$$|(\neg\varphi)|_{(} = |\varphi|_{(} + 1 = |\varphi|_{)} + 1 = |(\neg\varphi)|_{)}$$

   luego $P((\neg\varphi))$.

3. Supongamos $P(\varphi), P(\psi)$ con $\varphi, \psi \in PROP_{SP}$, entonces:

$$|(\varphi\Box\psi)|_{(} = |\varphi|_{(} + |\psi|_{(} + 1 = |\varphi|_{)} + |\psi|_{)} + 1 = |(\varphi\Box\psi)|_{)}$$

   luego $P((\varphi\Box\psi))$.

Definición 1.12. (Prefijos) Sea A un alfabeto y sea $w \in A^*$. Decimos que w' es prefijo de w si existe w'' tal que $w = w'w''$, es decir, si $w = w_1w_2...w_n$ entonces existe $0 \leq k \leq n$ tal que $w' = w_1...w_k$.
Decimos además que un prefijo w' es prefijo propio de w si es distinto de w y ϵ.

Ejemplo 1.13. Sea $A := \{a, b\}$ y sea $w := aababb$. Entonces sus prefijos son $\epsilon, a, aa, aab, aaba, aabab, aababb$, correspondientes a $k = 0, 1, 2, 3, 4, 5, 6$, respectivamente, y sus prefijos propios son $a, aa, aab, aaba, aabab$.

Proposición 1.14. *Si φ' es prefijo propio de φ, entonces $|\varphi'|_{)} < |\varphi'|_{(}$.*

Demostración. Se demuestra por inducción estructural.

1. Si $\varphi \in AT$ no tiene prefijos propios luego se cumple la afirmación.

2. Sea $\varphi \in PROP_{SP}$ cumpliendo la propiedad del enunciado y sea φ' un prefijo propio de $(\neg\varphi)$. Existen cuatro casos:

   a) $\varphi' = ($, entonces $|\varphi'|_{(} = 1 > 0 = |\varphi'|_{)}$-
   b) $\varphi' = (\neg$, ocurre lo mismo que antes.
   c) $\varphi' = (\neg\varphi''$ con φ'' prefijo propio de φ entonces por inducción se verifica $|\varphi''|_{)} < |\varphi''|_{(}$. Ahora bien, $|\varphi'|_{(} = |\varphi''|_{(} + 1$ y $|\varphi'|_{)} = |\varphi''|_{)}$ luego:

$$|\varphi'|_{)} < |\varphi'|_{(}$$

   d) $\varphi' = (\neg\varphi$, que es directo.

3. Se hace de forma similar al apartado anterior. Basta probar con los prefijos $($, $(\varphi_1'$, $(\varphi_1$, $(\varphi_1\Box$, $(\varphi_1\Box\varphi_2'$ y $(\varphi_1\Box\varphi_2$.

□

En próximas secciones será útil, dado un conjunto A, definir una función $H : PROP_{SP} \to A$ conociendo su restricción a AT, y suponiendo que podemos calcular $H((\neg\varphi))$ y $H((\varphi\Box\psi))$ en función de $H(\varphi)$ y $H(\psi)$. Si la función H queda bien definida así, diremos que está *definida de forma recursiva*. La siguiente proposición nos garantiza que podemos definir funciones de esta forma:

Proposición 1.15. *El esquema de definición recursiva da como resultado una única función, es decir, dadas:*

1. $H_{AT} : AT \to A$.
2. $H_{\neg} : A \to A$.
3. $H_{\square} : A \times A \to A$.

existe una única función $H : PROP_{SP} \to A$ *tal que:*

1. $H(\varphi) = H_{AT}(\varphi)$, *para toda* $\varphi \in AT$.
2. $H((\neg\varphi)) = H_{\neg}(H(\varphi))$.
3. $H((\varphi_1 \square \varphi_2)) = H_{\square}(H(\varphi_1), H(\varphi_2))$, *para cada* $\square \in \{\wedge, \vee, \to, \leftrightarrow\}$.

Demostración. Se omite la demostración. $\square$

Ejemplo 1.16. Sea $H : PROP_{SP} \to \mathbb{N}$ la función $|.|_($ definida anteriormente. Podemos definirla de manera recursiva:

1. $H(\varphi) = 0$, para toda $\varphi \in AT$.
2. $H((\neg\varphi)) = H(\varphi) + 1$.
3. $H((\varphi_1 \square \varphi_2)) = H(\varphi_1) + H(\varphi_2) + 1$.

Nótese como en ocasiones omitimos las funciones auxiliares y usamos la notación H para referirnos a H, H_{AT}, $H_{\neg}$ y $H_{\square}$. Este abuso de notación no suele dar lugar a confusión.

Ejemplo 1.17. Podemos definir recursivamente una función que nos lleve cada fórmula al conjunto de todas sus subfórmulas, es decir, $SUB : PROP_{SP} \to \mathcal{P}(PROP_{SP})$[4] tal que:

1. $SUB(\varphi) = \{\varphi\}$, para toda $\varphi \in AT$.
2. $SUB((\neg\varphi)) = SUB(\varphi) \cup \{(\neg\varphi)\}$.
3. $SUB((\varphi_1 \square \varphi_2)) = SUB(\varphi_1) \cup SUB(\varphi_2) \cup \{(\varphi_1 \square \varphi_2)\}$.

1.3. Eliminación de paréntesis

Para conseguir una notación más compacta, podemos establecer una notación que use menos paréntesis cuando nos convenga. Sin embargo, la omisión de paréntesis puede dar lugar a ambigüedades en las fórmulas, por ejemplo,

$$p \to q \to r$$

podría simbolizar dos fórmulas distintas:

[4] Dado un conjunto X, denotamos por $\mathcal{P}(X)$ a su potencia, es decir, el conjunto de subconjuntos de X.

- $((p \to q) \to r)$
- $(p \to (q \to r))$

Para evitar este tipo de situaciones, necesitamos las siguientes reglas de omisión de paréntesis:

1. Los paréntesis externos pueden omitirse. Esto no da lugar a ambigüedad.
2. Las conectivas se aplicarán en este orden: $\neg, \wedge, \vee, \to, \leftrightarrow$.
3. Cuando hay varias conectivas del mismo tipo seguidas, se asocia siempre empezando por la izquierda.

Algunos ejemplos de aplicación de estas reglas:

$$\begin{array}{ccc}
\neg p \vee p & \text{es} & ((\neg p) \vee p) \\
\neg p \wedge q \vee s \to t \leftrightarrow r & \text{es} & (((((\neg p) \wedge q) \vee s) \to t) \leftrightarrow r) \\
p \to q \to s & \text{es} & ((p \to q) \to s) \\
p \wedge q \leftrightarrow \neg s \to t \vee r & \text{es} & ((p \wedge q) \leftrightarrow ((\neg s) \to (t \vee r))) \\
t \leftrightarrow p \to \neg q \vee p \to s & \text{es} & (t \leftrightarrow ((p \to ((\neg q) \vee p)) \to s))
\end{array}$$

Algo a tener en cuenta al leer libros sobre lógica es que las reglas que adoptamos no son universales y varían de autor a autor (por ejemplo, en muchos casos se asocia por la derecha en vez de por la izquierda).

Comprobaremos cuando veamos semántica que las conectivas $\wedge$ y $\vee$ son asociativas y conmutativas, por tanto, en ese sentido, no importa el orden en que se asocien. De modo que, cuando una conectiva se repita varias veces seguidas, podemos usar la siguiente notación:

$$\bigwedge_{i \in I} p_i := p_{i_1} \wedge \cdots \wedge p_{i_n}$$

$$\bigvee_{i \in I} p_i := p_{i_1} \vee \cdots \vee p_{i_n},$$

siendo $I = \{i_1, \ldots, i_n\}$ un conjunto finito de índices.
Establecemos además el siguiente convenio,

$$\bigwedge_{i \in \emptyset} p_i := \top$$

$$\bigvee_{i \in \emptyset} p_i := \bot,$$

que será consistente con el comportamiento semántico de $\vee$ y $\wedge$.

1.4. Asignaciones de verdad

Hasta ahora hemos venido considerando cuestiones sintácticas sobre las fórmulas. Lo que queremos ahora es abordar su semántica, esto es, su comportamiento cuando les damos una interpretación determinada. Nos interesa, por tanto, estudiar qué significa que una proposición se siga de otra independientemente de la traducción que hagamos al lenguaje natural, esto es, la noción de *consecuencia lógica*. Así, por ejemplo, queremos que q se siga de $p \wedge (p \to q)$ independientemente de cómo se traduzcan p y q informalmente. Es decir, exigimos que, sea cual sea la traducción de $p \wedge (p \to q)$, si es verdadera entonces la de q tiene que serlo también. Siguiendo la definición 1.1,

Definición 1.18. Definimos el conjunto de valores de verdad, $Bool = \{V, F\}$, donde V es *la verdad* y F es *la falsedad*[5].

Una vez que disponemos de $Bool$ podemos definir la aplicación que nos lleva cada proposición a su valor de verdad correspondiente. Vamos a definir esta aplicación de forma recursiva, es decir, primero lo haremos para las fórmulas atómicas y después describiremos reglas para deducirla en el resto de fórmulas.

Definición 1.19. (Valoración) Sea el conjunto $Bool = \{V, F\}$. Una valoración o evaluación es una función $v : SP \to Bool$.

Un alfabeto de n letras tiene 2^n valoraciones distintas.

Para definir la valoración de proposiciones, vamos a necesitar una forma de interpretar las conectivas. Con este fin, a cada conectiva le asociaremos una aplicación, que definiremos caso a caso en formato de tabla de verdad:

1. $v_\neg : Bool \to Bool$;

φ	$v_\neg(\varphi)$
V	F
F	V

2. $v_\wedge, v_\vee, v_\to, v_\leftrightarrow : Bool \times Bool \to Bool$;

φ_1	φ_2	$v_\wedge(\varphi_1, \varphi_2)$	$v_\vee(\varphi_1, \varphi_2)$	$v_\to(\varphi_1, \varphi_2)$	$v_\leftrightarrow(\varphi_1, \varphi_2)$
V	V	V	V	V	V
V	F	F	V	F	F
F	V	F	V	V	F
F	F	F	F	V	V

[5]Nada nos impide considerar casos en que $Bool$ sea infinito numerable, o incluso no numerable, pero la lógica concebida clásicamente es bivalente, esto es, solo consta de dos valores de verdad.

Decimos que la conectiva binaria $\wedge$ es un operador conmutativo ya que $v_\wedge(V,F) = v_\wedge(F,V)$. $\vee$ y $\leftrightarrow$ también son conmutativas.

Definición 1.20. (Extensión) Dada la valoración $v : SP \to Bool$, definimos recursivamente su *extensión*, $\hat{v} : PROP_{SP} \to Bool$:

1. $\hat{v}(\bot) = F$.
2. $\hat{v}(\top) = V$.[6]
3. $\hat{v}(\varphi) = v(\varphi)$, para toda $\varphi \in SP$.
4. $\hat{v}((\neg\varphi)) = v_\neg(\hat{v}(\varphi))$.
5. $\hat{v}((\varphi_1 \square \varphi_2)) = v_\square(\hat{v}(\varphi_1), \hat{v}(\varphi_2))$.

La propia definición nos da un método para calcular los valores de verdad de fórmulas en función de los de sus símbolos:

Ejemplo 1.21. Encontremos el valor de verdad de $p \wedge \neg q \to r \vee s$, siendo v una valoración que cumple $v(p) = v(q) = V, v(r) = v(s) = F$:

1. $\hat{v}(\neg q) = v_\neg(\hat{v}(q)) = v_\neg(V) = F$.
2. $\hat{v}(p \wedge \neg q) = v_\wedge(\hat{v}(p), \hat{v}(\neg q)) = v_\wedge(V,F) = F$.
3. $\hat{v}(r \vee s) = v_\vee(\hat{v}(r), \hat{v}(s)) = v_\vee(F,F) = F$.
4. Finalmente, $\hat{v}(p \wedge \neg q \to r \vee s) = v_\to(\hat{v}(p \wedge \neg q), \hat{v}(r \vee s)) = v_\to(F,F) = V$.

Definición 1.22. (satisfactibilidad) Dada $v : SP \to Bool$ y $\varphi \in PROP_{SP}$ se dice que v satisface φ, y se escribe como $v \vDash \varphi$, si y solo si $\hat{v}(\varphi) = V$. Si $\hat{v}(\varphi) = F$, escribimos $v \nvDash \varphi$.

En virtud de la definición anterior, podemos clasificar cada fórmula φ como:

- *Satisfactible*, si existe alguna valoración v tal que $v \vDash \varphi$.
  - *Tautología*, si $v \vDash \varphi$ para toda valoración v.
  - *Contingencia*, si es satisfactible pero no tautología.
- *Contradicción*, si no es satisfactible.

Ejemplo 1.23. La fórmula $p \wedge q \to r \leftrightarrow (p \to q) \to r$ es contingencia. Invitamos al lector a que realice la tabla de verdad correspondiente.

Definición 1.24. (Conjunto de modelos) Dado $\Phi \subseteq PROP_{SP}$, el conjunto de modelos de Φ se define como:

$$Mod(\Phi) := \{v : SP \to Bool \mid \text{para toda } \varphi \in \Phi \ v \vDash \varphi\}$$

[6] Aquí es donde hemos dado sentido a la definición que dimos de $\top$ y $\bot$ en 1.2.

Definición 1.25. (Satisfactibilidad) Sea $\Phi \subseteq PROP_{SP}$.

Φ se dice satisfactible, y se escribe $Sat(\Phi)$, si $Mod(\Phi) \neq \emptyset$.

Φ se dice insatisfactible, y se escribe $Insat(\Phi)$, si $Mod(\Phi) = \emptyset$.

Ahora llegamos a la definición del concepto que ha motivado nuestro estudio inicial de la semántica.

Definición 1.26. (Consecuencia lógica) Sea $\varphi \in PROP_{SP}$, $\Phi \subseteq PROP_SP$. Decimos que φ es consecuencia lógica de Φ, $\Phi \vDash \varphi$, si y solo si $Mod(\Phi) \subseteq Mod(\{\varphi\})$.

Esto equivale a decir que toda asignación de verdad que satisface todos los elementos de Φ también satisface φ.

De ahora en adelante, escribimos $Mod(\varphi)$ en vez de $Mod(\{\varphi\})$.

Ejemplo 1.27. Dado Φ tal que $Insat(\Phi)$, se sigue de la anterior definición que $\Phi \vDash \varphi$, para cualquier φ.

A continuación presentamos algunos resultados básicos de semántica.

Proposición 1.28.

1. *$Mod(\emptyset) = \{v | v : SP \to Bool\}$.*
2. *Si $\Phi = \emptyset$ y $\Phi \vDash \varphi$, entonces φ es tautología.*
3. *$Mod(PROP_{SP}) = \emptyset$.*
4. *$Mod(\Phi) \cap Mod(\Psi) = Mod(\Phi \cup \Psi)$.*
5. *$Mod(\Phi) \cup Mod(\Psi) \subseteq Mod(\Phi \cap \Psi)$.*

Demostración.

1. En efecto, toda valoración satisface todas las proposiciones del conjunto vacío.
2. $Mod(\emptyset)$ tiene a todas las valoraciones, por tanto si $Mod(\emptyset) \subseteq Mod(\varphi)$, cualquier valoración satisface φ.
3. Dada una valoración v, y cualquier proposición φ, no podemos tener a la vez $\hat{v}(\varphi) = T$ y $\hat{v}(\neg\varphi) = T$. Por tanto v no satisface todos los elementos de $PROP_{SP}$.
4. Comprobamos los dos contenidos:
   $\subseteq$: Si una valoración v está en $Mod(\Phi) \cap Mod(\Psi)$, satisface todos los elementos de Φ y de Ψ, por tanto satisface todos los elementos de $\Phi \cup \Psi$.
   $\supseteq$: Si v está en $Mod(\Phi \cup \Psi)$, v satisface todos los elementos de $\Phi \cup \Psi$, por tanto satisface todos los elementos de Φ y todos los de Ψ, es decir, $v \in Mod(\Phi) \cap Mod(\Psi)$.

5. Dada $v \in Mod(\Phi) \cup Mod(\Psi)$, o bien v verifica todos los elementos de Φ o los de Ψ. En ambos casos, v verifica los elementos de $\Phi \cap \Psi$.

En 5, el contenido opuesto no se cumple. Poniendo por ejemplo $\Phi = \{p\}, \Psi = \{p \wedge p\}$, siendo $p \in SP$, tenemos $\Phi \cap \Psi = \emptyset$, por tanto $Mod(\Phi \cap \Psi)$ contiene a todas las valoraciones, pero cualquier valoración v que cumpla $v(p) = F$ no estará en $Mod(\Phi) \cup Mod(\Psi)$. $\square$

Proposición 1.29. $\Phi \cup \{\varphi\} \vDash \psi$ *si y solo si* $\Phi \vDash \varphi \to \psi$.

Demostración. Veamos que $Mod(\Phi) \subseteq Mod(\varphi \to \psi)$ suponiendo que $Mod(\Phi \cup \{\varphi\}) \subseteq Mod(\psi)$. Sea $v \in Mod(\Phi)$. Si $\hat{v}(\varphi) = V$ entonces $v \in Mod(\varphi)$. Por hipótesis, $v \in Mod(\psi)$, luego $\hat{v}(\psi) = V$ y entonces $\hat{v}(\varphi \to \psi) = V$, es decir, $v \in Mod(\varphi \to \psi)$. Si $\hat{v}(\varphi) = F$, $\hat{v}(\varphi \to \psi) = V$ y de nuevo $v \in Mod(\varphi \to \psi)$.

Recíprocamente, si $v \in Mod(\Phi \cup \{\varphi\})$, entonces $\hat{v}(\varphi) = V$. Por hipótesis, al ser $v \in Mod(\Phi)$, $v \in Mod(\varphi \to \psi)$, es decir, $\hat{v}(\varphi \to \psi) = V$, luego $\hat{v}(\psi) = V$ y $v \in Mod(\psi)$. $\square$

Proposición 1.30. $\Phi \vDash \varphi$ *si y solo si* $Insat(\Phi \cup \{\neg\varphi\})$

Demostración. Supongamos que $\Phi \vDash \varphi$, esto es, que $Mod(\Phi) \subset Mod(\varphi)$. Sea $v \in Mod(\Phi \cup \{\neg\varphi\})$. Entonces, por definición, $v \vDash \chi$, para toda $\chi \in \Phi \cup \{\neg\varphi\}$. Luego en especial, $v \vDash \neg\varphi$, lo que contradice que $v \in Mod(\varphi)$ (¿por qué?).

Recíprocamente, si $Mod(\Phi \cup \{\neg\varphi\}) = \emptyset$, sea $v \in Mod(\Phi)$. Entonces $v \notin Mod(\neg\varphi)$, por tanto $\hat{v}(\neg\varphi) = F$, pero como $\hat{v}(\neg\varphi) = v_\neg(\hat{v}(\varphi))$, tenemos que $F = v_\neg(\hat{v}(\varphi))$, es decir, $\hat{v}(\varphi) = V$, así que $v \in Mod(\varphi)$. Como esto lo cumple cualquier $v \in Mod(\Phi)$, tenemos, como queríamos, que $Mod(\Phi) \subseteq Mod(\varphi)$

$\square$

De ahora en adelante, escribiremos $\Phi, \varphi \vDash \psi$ en vez de $\Phi \cup \{\varphi\} \vDash \psi$.

Dada una valoración v, para cualesquiera fórmulas φ_1, φ_2 en que aparecen los símbolos $p_1, \ldots, p_n$ y $q_1, \ldots, q_m$, respectivamente, está claro, por cómo hemos definido las valoraciones, que $\hat{v}(\varphi_1)$ depende de $v(p_1), \ldots, v(p_n)$, y $\hat{v}(\varphi_2)$ depende de $v(q_1), \ldots, v(q_m)$[7]. De modo que para comprobar que $\varphi_1 \vDash \varphi_2$ nos basta con comprobar que, para cada una de las posibles valoraciones de $p_1, \ldots, p_n, q_1, \ldots, q_m$, $\hat{v}(\varphi_2) = V$ si $\hat{v}(\varphi_1) = V$. Este es nuestro primer procedimiento para demostrar consecuencias lógicas entre fórmulas, que resumiremos gráficamente en tablas de verdad.

Ejemplo 1.31. Consideremos $\Phi = \{\neg p \wedge q \to r,\ \neg p,\ \neg r\}$, $\varphi = \neg q$. Para demostrar que $\Phi \vDash \varphi$ basta realizar la tabla de verdad correspondiente y comparar los valores de verdad de las premisas del conjunto Φ con los de φ. En Φ y φ aparecen los símbolos p, q y r, por tanto tenemos la siguiente tabla de verdad:

[7] Este hecho se probará y generalizará para lógica de primer orden en el lema de coincidencia, 2.41.

p	q	r	$(\neg p \wedge q) \to r$	$\neg p$	$\neg r$	$\neg q$
V	V	V	V	F	F	F
V	V	F	V	F	V	F
V	F	V	V	F	F	V
V	F	F	V	F	V	V
F	V	V	V	V	F	F
F	V	F	F	V	V	F
F	F	V	V	V	F	V
F	F	F	V	V	V	V

En efecto, solo hay un caso en que se cumplen todas las proposiciones de Φ (la última fila) y en ese caso se cumple φ. Por tanto, $\Phi \vDash \varphi$.

1.5. Equivalencia lógica

El método descrito de las tablas de verdad aumenta de complejidad a medida que lo hace el número de proposiciones que estudiamos. Por ello, investigamos relaciones formales entre las fórmulas que simplifiquen tales cuestiones. Comenzamos por las proposiciones atómicas:

Definición 1.32. (Equivalencia lógica) Sean $\varphi, \psi \in PROP_{SP}$. φ se dice lógicamente equivalente a ψ, $\varphi \sim \psi$, si para toda valoración v, $\hat{v}(\varphi) = \hat{v}(\psi)$.

Esto equivale a decir que $\varphi_1 \vDash \varphi_2$ y $\varphi_2 \vDash \varphi_1$.

Proposición 1.33. *$\sim$ es relación de equivalencia. Además, $\sim$ es congruencia respecto de las conectivas lógicas, es decir:*

1. *Si $\varphi \sim \psi$, entonces $\neg\varphi \sim \neg\psi$.*
2. *Si $\varphi_1 \sim \varphi_2$ y $\psi_1 \sim \psi_2$, entonces $\varphi_1 \Box \psi_1 \sim \varphi_2 \Box \psi_2$.*

Demostración. Que es relación de equivalencia es directo.

Supongamos $\varphi \sim \psi$, y sea v cualquier valoración. Entonces

$$\hat{v}(\neg\varphi) = v_\neg(\hat{v}(\varphi)) = v_\neg(\hat{v}(\psi)) = \hat{v}(\neg\psi).$$

De igual manera, supongamos $\varphi_1 \sim \varphi_2$ y $\psi_1 \sim \psi_2$, sea v cualquier valoración. Entonces

$$\hat{v}(\varphi_1 \Box \psi_1) = v_\Box(\hat{v}(\varphi_1), \hat{v}(\psi_1)) = v_\Box(\hat{v}(\varphi_2), \hat{v}(\psi_2)) = \hat{v}(\varphi_2 \Box \psi_2).$$

$\Box$

Las siguientes reglas nos ayudarán a demostrar equivalencias entre fórmulas:

Teorema 1.34. *(Leyes de equivalencia lógica)*

1. *Conmutatividad:*

$$p \wedge q \sim q \wedge p, \quad p \vee q \sim q \vee p, \quad p \leftrightarrow q \sim q \leftrightarrow p$$

2. *Asociatividad:*

$$(p \vee q) \vee r \sim p \vee (q \vee r), \quad (p \wedge q) \wedge r \sim p \wedge (q \wedge r)$$

3. *Idempotencia:*

$$p \wedge p \sim p, \quad p \vee p \sim p$$

4. *Distributiva:*

$$p \wedge (q \vee r) \sim (p \wedge q) \vee (p \wedge r), \quad p \vee (q \wedge r) \sim (p \vee q) \wedge (p \vee r)$$

5. *Absorción:*

$$p \wedge (p \vee q) \sim p, \quad p \vee (p \wedge q) \sim p$$

6. *Cero y unidad:*

$$p \wedge \top \sim p, \quad p \vee \top \sim \top, \quad p \wedge \bot \sim \bot, \quad p \vee \bot \sim p$$

7. *Contradicción:*

$$p \wedge \neg p \sim \bot$$

8. *Tercio excluso:*

$$p \vee \neg p \sim \top$$

9. *Negación:*

$$\neg\neg p \sim p$$

10. *Leyes de De Morgan:*

$$\neg(p \wedge q) \sim \neg p \vee \neg q, \quad \neg(p \vee q) \sim \neg p \wedge \neg q$$

11. *Simplificación de los condicionales:*

$$p \to q \sim \neg p \vee q, \quad p \leftrightarrow q \sim (p \to q) \wedge (q \to p)$$

Demostración. Las demostraciones de equivalencia consisten simplemente en comprobar que coinciden las tablas de verdad. Comprobamos las tres primeras:

1. Conmutatividad:

p	q	$p \wedge q$	$q \wedge p$	$p \vee q$	$q \vee p$	$p \leftrightarrow q$	$q \leftrightarrow p$
V	V	V	V	V	V	V	V
V	F	F	F	V	V	F	F
F	V	F	F	V	V	F	F
F	F	F	F	F	F	V	V

2. Asociatividad:

p	q	r	$(p \vee q) \vee r$	$p \vee (q \vee r)$	$(p \wedge q) \wedge r$	$p \wedge (q \wedge r)$
V	V	V	V	V	V	V
V	V	F	V	V	F	F
V	F	V	V	V	F	F
V	F	F	V	V	F	F
F	V	V	V	V	F	F
F	V	F	V	V	F	F
F	F	V	V	V	F	F
F	F	F	F	F	F	F

3. Idempotencia:

p	$p \wedge p$	$p \vee p$
V	V	V
F	F	F

□

1.6. Sustitución

Pasamos a estudiar los lemas de sustitución, que entre otras cosas nos permitirán generalizar las leyes de equivalencia lógica. Para ello es necesario estudiar el concepto de *sustitución*:

Definición 1.35. (Sustitución) Sean $\varphi, \psi \in PROP_{SP}$ y sea $p \in SP$. Llamamos $\psi[\varphi/p]$ a la fórmula que resulta de sustituir en ψ cada aparición de p por φ. Dados φ y p, podemos definir $\psi[\varphi/p]$ recursivamente del siguiente modo:

1. $p[\varphi/p] = \varphi$.
2. $q[\varphi/p] = q$, para cualquier $q \in SP \setminus \{p\}$.

3. $\top[\varphi/p] = \top$.

4. $\bot[\varphi/p] = \bot$.

5. $(\neg\psi)[\varphi/p] = \neg(\psi[\varphi/p])$.

6. $(\psi_1 \Box \psi_2)[\varphi/p] = (\psi_1[\varphi/p] \Box \psi_2[\varphi/p])$.

Ejemplo 1.36. Si usamos las fórmulas $\psi = p \wedge q \to p$ y $\varphi = p_1 \vee \neg p_2$, al sustituir p por φ en ψ obtenemos la fórmula:

$$\psi[\varphi/p] = (p_1 \vee \neg p_2) \wedge q \to (p_1 \vee \neg p_2).$$

Introducimos la siguiente notación que será de utilidad: dada una función $f : A \to B$ y dados $a \in A$, $b \in B$, denotamos respectivamente por $f[b/a](x)$ a $f(x)$, si $x \neq a$, y a b, si $x = a$.

Ejemplo 1.37. Sea $X \in \{V, F\}$. Si con la notación anterior usamos como función una valoración v, $v[X/p]$ es una valoración de verdad que cumple:

- $v[X/p](q) = v(q)$ si $q \neq p$
- $v[X/p](p) = X$.

La siguiente proposición muestra que obtenemos el mismo resultado tanto si sustituimos y posteriormente asignamos un valor de verdad como si hacemos el proceso inverso:

Proposición 1.38. *Sean $\varphi, \psi \in PROP_{SP}$, $p \in SP$ y $v : SP \to Bool$ valoración. Entonces:*

$$\hat{v}(\psi[\varphi/p]) = \widehat{v[\hat{v}(\varphi)/p]}(\psi)$$

Demostración. Procedemos por inducción estructural. Dividimos las proposiciones atómicas en cuatro casos como en la definición 1.35:

1. Si $\psi = p$ entonces, por definición de sustitución $\psi[\varphi/p] = \varphi$. Por tanto, $\hat{v}(\psi[\varphi/p]) = \hat{v}(\varphi)$ y siguiendo la notación que hemos explicado arriba, $\widehat{v[\hat{v}(\varphi)/p]}(\psi) = \widehat{v[\hat{v}(\varphi)/p]}(p) = v[\hat{v}(\varphi)/p](p) = \hat{v}(\varphi)$.

2. Si $\psi = q \in SP$ y $q \neq p$ entonces $\psi[\varphi/p] = q$, con lo que $\hat{v}(\psi[\varphi/p]) = \hat{v}(q)$ y, como antes, $\widehat{v[\hat{v}(\varphi)/p]}(q) = v[\hat{v}(\varphi)/p](q) = \hat{v}(q)$.

3. Si $\psi = \top$, $\psi[\varphi/p] = \top$, y $\widehat{v[\hat{v}(\varphi)/p]}(\top) = v(\top) = V$.

4. Si $\psi = \bot$, $\psi[\varphi/p] = \bot$, y $\widehat{v[\hat{v}(\varphi)/p]}(\bot) = v(\bot) = F$.

5. Si $\psi = \neg\psi_1$, entonces $\hat{v}((\neg\psi_1)[\varphi/p]) = \hat{v}(\neg(\psi_1[\varphi/p])) = v_\neg(\hat{v}(\psi_1[\varphi/p]))$, por las definiciones de sustitución y extensión de la valoración v. Aplicando la hipótesis de inducción, lo anterior es igual a

$$v_\neg(\widehat{v[\hat{v}(\varphi)/p]})(\psi_1) = \widehat{v[\hat{v}(\varphi)/p]}(\neg\psi_1).$$

6. Si $\psi = \psi_1 \Box \psi_2$, entonces

$$\hat{v}(\psi[\varphi/p]) = \hat{v}((\psi_1[\varphi/p] \ \Box \ \psi_2[\varphi/p])) = v_{\Box}(\hat{v}(\psi_1[\varphi/p]),\ \hat{v}(\psi_2[\varphi/p])).$$

Aplicando la hipótesis de inducción, esto es igual a

$$v_{\Box}(\widehat{v[\hat{v}(\varphi)/p]}(\psi_1),\ \widehat{v[\hat{v}(\varphi)/p]}(\psi_2)) = \widehat{v[\hat{v}(\varphi)/p]}(\psi_1 \Box \psi_2).$$

□

Lema 1.39. *(De sustitución) Sean $\varphi, \varphi', \psi \in PROP_{SP}$, $p \in SP$. Si $\varphi \sim \varphi'$, $\psi[\varphi/p] \sim \psi[\varphi'/p]$.*

Demostración. Sea v valoración. Entonces, al ser $\hat{v}(\varphi) = \hat{v}(\varphi')$, se obtiene que

$$\hat{v}(\psi[\varphi/p]) = \widehat{v[\hat{v}(\varphi)/p]}(\psi) = \widehat{v[\hat{v}(\varphi')/p]}(\psi) = \hat{v}(\psi[\varphi'/p])$$

Como esto sucede para cualquier valoración v, $\psi[\varphi/p] \sim \psi[\varphi'/p]$. □

Acabamos de ver que sustituir fórmulas equivalentes en una misma fórmula nos lleva a fórmulas equivalentes. A continuación vemos que sustituir la misma fórmula en fórmulas equivalentes da como resultado fórmulas equivalentes.

Lema 1.40. *(De sustitución) Sean $\varphi, \psi_1, \psi_2 \in PROP_{SP}$, $p \in SP$. Si $\psi_1 \sim \psi_2$, entonces $\psi_1[\varphi/p] \sim \psi_2[\varphi/p]$.*

Demostración. Sea v cualquier valoración. Tenemos que:

$$\hat{v}(\psi_1[\varphi/p]) = \widehat{v[\hat{v}(\varphi)/p]}(\psi_1) = \widehat{v[\hat{v}(\varphi)/p]}(\psi_2) = \hat{v}(\psi_2[\varphi/p])$$

Lo que nos da el resultado. □

El lema 1.40 nos permite probar una versión más general de las leyes de equivalencia lógica, 1.34, en las que en vez de símbolos de proposición podemos sustituir cualquier fórmula. En el siguiente ejemplo probamos la primera:

Ejemplo 1.41. Consideremos las dos fórmulas $\psi_1 = p \vee q$ y $\psi_2 = q \vee p$. Sabemos que son equivalentes, luego si consideramos $\varphi, \chi \in PROP_{SP}$ tenemos que, por el lema 1.40, $\psi_1[\varphi/p]$ y $\psi_2[\varphi/p]$, es decir, $\varphi \vee q$ y $q \vee \varphi$, son equivalentes. Aplicando una vez más 1.40 para sustituir q por χ, obtenemos que $\varphi \vee \chi$ y $\chi \vee \varphi$ son equivalentes.

Ejemplo 1.42. Supongamos que queremos demostrar la equivalencia de estas dos fórmulas:

- $p_1 \vee p_2 \vee p_3 \to p_4 \wedge (p_5 \vee p_6)$
- $\neg((p_4 \wedge p_5) \vee (p_4 \wedge p_6)) \to \neg(p_1 \vee p_2 \vee p_3)$

Desde luego podríamos hacerlo usando tablas de verdad, pero está claro que tardaríamos demasiado. Una forma más sencilla de hacerlo usando las leyes de equivalencia y los lemas de sustitución es comprobar las dos equivalencias lógicas:

1. $p_1 \vee p_2 \vee p_3 \to p_4 \wedge (p_5 \vee p_6) \sim p_1 \vee p_2 \vee p_3 \to (p_4 \wedge p_5) \vee (p_4 \wedge p_6)$
2. $p_1 \vee p_2 \vee p_3 \to (p_4 \wedge p_5) \vee (p_4 \wedge p_6) \sim \neg((p_4 \wedge p_5) \vee (p_4 \wedge p_6)) \to \neg(p_1 \vee p_2 \vee p_3)$

En efecto:

- 1 se obtiene aplicando el lema 1.39 con $\psi = p_1 \vee p_2 \vee p_3 \to p$, $\varphi_1 = p_4 \wedge (p_5 \vee p_6)$,$\varphi_2 = (p_4 \wedge p_5) \vee (p_4 \wedge p_6)$.
- 2 se obtiene en dos pasos:
  Aplicando el lema 1.40 con $\psi_1 = p \to q$, $\psi_2 = \neg q \to \neg p$ y $p = p_1 \vee p_2 \vee p_3$, obtenemos: $p_1 \vee p_2 \vee p_3 \to q \sim \neg q \to \neg(p_1 \vee p_2 \vee p_3)$.
  Y aplicando de nuevo el lema 1.40 con $\psi_1 = p_1 \vee p_2 \vee p_3 \to q$, $\psi_2 = \neg q \to \neg(p_1 \vee p_2 \vee p_3)$ y $q = (p_4 \wedge p_5) \vee (p_4 \wedge p_6)$, obtenemos 2.

1.7. Completitud funcional

Hasta ahora solo hemos tratado con cinco conectivas, $\neg, \wedge, \vee, \to, \leftrightarrow$. En esta sección vamos a considerar de forma más general los símbolos de conectiva y su interpretación semántica. Dado un símbolo de conectiva \$, decimos que \$ es n-aria si con ella se forman fórmulas con n símbolos de proposición, de la forma $\$(p_1, \ldots, p_n)$. Por ejemplo, las conectivas $\wedge, \vee, \to, \leftrightarrow$ son binarias (2-arias) y la conectiva $\neg$ es 1-aria. Esto puede resultar extraño ya que hasta ahora hemos usado expresiones de tipo $(p_1 \Box p_2)$, en vez de $\Box(p_1, p_2)$. Sin embargo, también se puede formular la lógica proposicional usando esta nueva notación[8].

En 1.20 definimos recursivamente una serie de asignaciones de verdad que correspondían a los símbolos de las conectivas lógicas, $\neg, \wedge, \vee, \to, \leftrightarrow$. En general, a cualquier tabla de verdad dada por $v_\$: Bool^n \to Bool$ podemos asignarle una conectiva n-aria, $\$(p_1, \ldots, p_n)$[9]. En general supondremos que $n \geq 1$, pero según esta definición, podríamos interpretar los símbolos $\top, \perp$ como conectivas 0-arias, dadas por las tablas de verdad:

$\top$	$\perp$
V	F

[8]De hecho, la conocida *notación polaca* prescinde de los paréntesis y escribe las conectivas de forma $\$p_1 \ldots p_n$, sin que esa falta de paréntesis dé problemas en la estructura del lenguaje.

[9]Aunque sintácticamente escribiremos las conectivas como $\$(p_1, \ldots, p_n)$, semánticamente el valor de verdad de $\$(p_1, \ldots, p_n)$ solo dependerá de los valores de verdad de $p_1, \ldots, p_n$, no de $p_1, \ldots, p_n$, al igual que con las conectivas conocidas anteriormente.

Analizando con mayor detenimiento las conectivas binarias, es fácil observar que existen 16 de ellas distintas, una asociada a cada tabla de verdad[10]. Por ejemplo, podemos definir mediante tablas de verdad las conectivas:

p	q	$p \veebar q$	$p \uparrow q$	$p \downarrow q$
V	V	F	F	F
V	F	V	V	F
F	V	V	V	F
F	F	F	V	V

$\veebar$ se conoce como *disyunción excluyente*, mientras que $\uparrow$ y $\downarrow$ se llaman *NAND* y *NOR*, respectivamente.

Sin embargo, en cierto sentido, estas nuevas conectivas son redundantes, ya que podemos expresarlas en función de nuestras cinco conectivas iniciales:

- $p \veebar q \sim \neg(p \leftrightarrow q)$
- $p \uparrow q \sim \neg(p \wedge q)$
- $p \downarrow q \sim \neg(p \vee q)$

Es natural, por tanto, preguntarse si cualquier conectiva n-aria será expresable en función de nuestras conectivas, como lo son $\veebar, \downarrow$ y $\uparrow$. Para formalizar esta cuestión, introducimos dos nociones de gran importancia:

Definición 1.43. Una conectiva n-aria \$ dada por $v_\$: Bool^n \rightarrow Bool$ se dice expresable en términos de un conjunto de conectivas C si existe una fórmula φ que tiene por conectivas a elementos de C y por símbolos de proposición a ciertos $p_1, ..., p_k$, tal que $\varphi \sim \$(p_1, ..., p_k)$.

Definición 1.44. (Completitud funcional) Un conjunto de conectivas C se dice funcionalmente completo si cualquier conectiva \$ dada por $v_\$: Bool^n \rightarrow Bool$ es expresable en términos de C.

Ejemplo 1.45. Si C es funcionalmente completo, cualquier proposición φ con símbolos $p_1, \ldots, p_n$ y cualesquiera conectivas $\$_1, \ldots, \$_m$ es equivalente a una expresión con símbolos $p_1, \ldots, p_n$ y conectivas en C.

En efecto, φ tiene una tabla de verdad en función de $p_1, \ldots, p_n$, de modo que se puede expresar como una conectiva n-aria $\$_\varphi$ con esa misma tabla de verdad, es decir, $\$_\varphi(p_1, \ldots, p_n) \sim \varphi$. Ahora, como C es funcionalmente completo, $\$_\varphi$ es expresable en términos de C, por tanto hay una expresión ψ de conectivas elementos de C y símbolos de proposición $p_1, \ldots, p_n$ tal que $\psi \sim \$_\varphi \sim \varphi$.

Ahora ya podemos dar respuesta a la pregunta que nos formulamos antes:

[10] Decimos que dos conectivas n-arias son distintas cuando tienen distinta tabla de verdad. Para cada n, hay 2^{2^n} conectivas n-arias distintas.

Proposición 1.46. $\{\neg, \wedge, \vee\}$ *es funcionalmente completo.*

Demostración. Sea \$ conectiva n-aria. Procedemos por inducción sobre n.

Si $n = 1$, es fácil comprobar que tenemos cuatro conectivas posibles, y que éstas corresponden a la identidad (es decir, $\$(p) \sim p$), $\neg$, $\top \sim p \vee \neg p$ y $\perp \sim p \wedge \neg p$, con lo que se verifica el enunciado.

Supongamos que es válido para $n-1$ y veámoslo para n. Sea $\$(p_1, ..., p_n)$ una conectiva dada por la siguiente tabla de verdad:

p_n	p_1 ... p_{n-1}	$\$(p_1, ..., p_n)$
V	... ...	$X(1)$
...	... ...	...
V	... ...	$X(2^{n-1})$
F	... ...	$X(2^{n-1}+1)$
...	... ...	...
F	... ...	$X(2^n)$

donde $X(i)$ denota el valor de verdad correspondiente a la fila i-ésima. De esta forma obtenemos dos tablas de verdad, cada una con 2^{n-1} filas: aquella en la que p_n toma V como valor de verdad (las primeras 2^{n-1} filas de la tabla anterior) y aquella en la que toma el valor F (las filas restantes). Llamamos $\$_1(p_1, ..., p_{n-1})$ y $\$_2(p_1, ..., p_{n-1})$ a las conectivas inducidas por estas dos tablas de verdad.

Entonces tenemos:

$$\$(p_1, ..., p_n) \sim (p_n \wedge \$_1(p_1, ..., p_{n-1})) \vee (\neg p_n \wedge \$_2(p_1, ..., p_{n-1})).$$

Aplicando la hipótesis de inducción, existen fórmulas φ_1, φ_2 construidas a partir de $\{\neg, \wedge, \vee\}$ tales que $\varphi_1 \sim \$_1(p_1, ..., p_{n-1})$ y $\varphi_2 \sim \$_2(p_1, ..., p_{n-1})$, con lo que obtenemos que $\$(p_1, ..., p_n)$ es expresable en términos de $\{\neg, \wedge, \vee\}$. $\square$

Definición 1.47. (Formas normales) Decimos que una fórmula está en *Forma Normal Conjuntiva* (FNC) si es de la forma:

$$\bigwedge_{i \leq n} \bigvee_{j \leq m_i} \varphi_{ij}$$

y se dice que está en *Forma Normal Disyuntiva* (FND) si es de la forma:

$$\bigvee_{i \leq n} \bigwedge_{j \leq m_i} \varphi_{ij}$$

donde cada φ_{ij} es una proposición atómica o la negación de una proposición atómica. Ya veremos en un ejercicio que, para cada fórmula φ, existen dos fórmulas φ^c, φ^d en FNC y en FND, respectivamente, tales que $\varphi \sim \varphi^c$ y $\varphi \sim \varphi^d$.

Corolario 1.48.

1. $\{\neg, \wedge\}$ *es funcionalmente completo.*
2. $\{\neg, \vee\}$ *es funcionalmente completo.*
3. $\{\rightarrow, \perp\}$ *es funcionalmente completo.*
4. $\{\uparrow\}$ *es funcionalmente completo.*
5. $\{\downarrow\}$ *es funcionalmente completo.*

Demostración.

1. Sabemos, por 1.46, que $\{\neg, \wedge, \vee\}$ es funcionalmente completo. Basta ver, entonces, que $\vee$ es expresable en términos de $\neg$ y $\wedge$. Esto es evidente por las leyes de De Morgan (1.34):

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

2. Análogo al anterior, usando la otra ley de De Morgan:

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

3. Basta ver que $\neg, \vee$ son expresables en términos de $\rightarrow, \perp$. Esto se sigue de estas afirmaciones que se pueden comprobar con tablas de verdad:

$$\neg p \sim p \rightarrow \perp.$$

$$p \vee q \sim (p \rightarrow \perp) \rightarrow q.$$

4. Expresamos $\neg, \wedge$ en términos de $\uparrow$:

$$\neg p \sim p \uparrow p.$$

$$p \wedge q \sim (p \uparrow q) \uparrow (p \uparrow q).$$

5. Expresamos $\neg, \vee$ en términos de $\downarrow$:

$$\neg p \sim p \downarrow p$$

$$p \vee q \sim (p \downarrow q) \downarrow (p \downarrow q).$$

□

1.8. Método de los *tableaux*

Dado un conjunto finito de fórmulas Ψ, nos interesa encontrar un método sistemático para determinar si $Insat(\Psi)$. Este procedimiento nos permitirá también deducir si, dados un conjunto de fórmulas $\Phi = \{\varphi_1, ..., \varphi_n\}$ y una fórmula φ, $\Phi \vDash \varphi$, pues por 1.30 esto es equivalente a $Insat(\Phi \cup \{\neg\varphi\})$ o, alternativamente, a que $\bigwedge_{i=1}^{n} \varphi_i \wedge \neg\varphi \sim \perp$. Es decir, aplicaremos el método a $\Psi := \Phi \cup \{\neg\varphi\}$.

1.8.1. Nociones básicas y ejemplos

Nuestro objetivo es tomar los datos anteriores como vértices de un grafo y construir un *árbol*, es decir, un diagrama (llamado '*tableau*') en el que conectaremos las distintas fórmulas mediante aristas, formando *ramas* y de modo que, dados dos elementos del árbol, si pertenecen a la misma rama entonces se encuentran en conjunción y, si son de ramas distintas, se encuentran en disyunción. De esta forma, y a partir de una serie de reglas de construcción, el problema de la insatisfactibilidad de Ψ quedará reducido a comprobar que un *tableau* suyo tiene todas las ramas *cerradas*, esto es, contienen una contradicción.

Construiremos nuestro diagrama de árbol, cuyos vértices serán proposiciones, de arriba hacia abajo. Llamamos al vértice de arriba del todo *raíz*[11]. Llamaremos *hojas* a los vértices v tales que no haya ningún vértice v' debajo de v y unido por una arista a v. Llamaremos *rama* a una sucesión de vértices $\langle v_1, \ldots, v_n \rangle$ en que:

- v_1 es la raíz.
- v_{i+1} está unido por una arista a v_i y debajo de v_i para todo i.
- v_n es una hoja.

Por ejemplo, en el diagrama:

$$\begin{tikzcd}[arrows=-]
 & r \arrow[d] & \\
 & p \vee q \arrow[dl] \arrow[dr] & \\
p & & q
\end{tikzcd}$$

tendríamos estas dos ramas:

[11]Todos estos conceptos, que por ahora se van a explicar de forma intuitiva, quedarán mejor formalizados en lógica de primer orden, en las definiciones 2.64 a 2.66.

$$
\begin{array}{ccc}
r & \qquad & r \\
| & & | \\
p \lor q & & p \lor q \\
| & & | \\
q & & p
\end{array}
$$

Al hacer el *tableau* podemos encontrarnos fórmulas que se pueden simplificar, llamadas *σ-fórmulas* o *simplificables*. Estas simplificaciones se podrán hacer de acuerdo a la siguiente tabla:

σ	σ_1
$\neg\top$	$\bot$
$\neg\bot$	$\top$
$\neg\neg\varphi$	φ

Está claro que, en cada fila, $\sigma \sim \sigma_1$.

Ahora bien, en el proceso de extender el *tableau*, nos conviene distinguir dos tipos de fórmulas, que llamaremos *α-fórmulas* y *β-fórmulas*. En las siguientes tablas, la primera columna servirá como definición de α-fórmulas y β-fórmulas, es decir, habrá cuatro tipos de α-fórmulas y cuatro tipos de β-fórmulas. En las otras dos columnas aparecerán sus descomposiciones, que usaremos al construir los *tableaux*.

- *α-fórmulas*:

α	α_1	α_2
$\varphi_1 \land \varphi_2$	φ_1	φ_2
$\neg(\varphi_1 \lor \varphi_2)$	$\neg\varphi_1$	$\neg\varphi_2$
$\neg(\varphi_1 \to \varphi_2)$	φ_1	$\neg\varphi_2$
$\varphi_1 \leftrightarrow \varphi_2$	$\varphi_1 \to \varphi_2$	$\varphi_2 \to \varphi_1$

- *β-fórmulas*:

β	β_1	β_2
$\varphi_1 \lor \varphi_2$	φ_1	φ_2
$\neg(\varphi_1 \land \varphi_2)$	$\neg\varphi_1$	$\neg\varphi_2$
$\varphi_1 \to \varphi_2$	$\neg\varphi_1$	φ_2
$\neg(\varphi_1 \leftrightarrow \varphi_2)$	$\neg(\varphi_1 \to \varphi_2)$	$\neg(\varphi_2 \to \varphi_1)$

Podemos comprobar que para cada fila de la primera tabla, se cumple $\alpha \sim \alpha_1 \wedge \alpha_2$, y para cada fila de la segunda tabla, se cumple $\beta \sim \beta_1 \vee \beta_2$.

No es difícil probar el siguiente resultado, que justifica la anterior clasificación:

Proposición 1.49. *Sea $\varphi \in PROP_{SP}$. Entonces φ verifica una de las siguientes condiciones:*

1. *Es de la forma p, $\neg p$, $\bot$ o $\top$, con $p \in SP$.*
2. *Es σ-fórmula.*
3. *Es α-fórmula.*
4. *Es β-fórmula.*

Debido a esto, nuestro método nos permitirá descomponer un conjunto de proposiciones múltiples veces hasta obtener proposiciones de tipo p_i, $\neg p_i$, $\bot$ o $\top$, con $p_i \in SP$.

Ahora pasamos a describir las reglas de construcción de *tableaux* asociados a un conjunto Ψ de proposiciones.

- *Regla inicial*, R_{ini}: Sean $\varphi_1, ..., \varphi_n \in \Psi$. Entonces

$$\begin{array}{c} \varphi_1 \\ | \\ \varphi_2 \\ \vdots \\ \varphi_n \end{array}$$

  es *tableau* de Ψ.

- *Regla de hipótesis*, R_{hip}: Sea

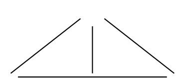
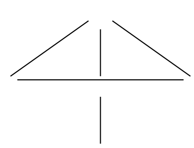


  tableau de Ψ. Entonces



$$\varphi$$

es *tableau* de Ψ, con $\varphi \in \Psi$.
Es decir, siempre podemos añadir un elemento de Ψ a una rama.

- *Regla* σ, R_σ: Sea

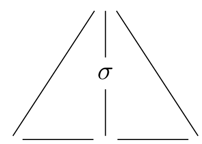
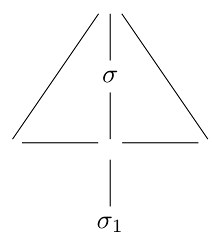


tableau de Ψ, donde σ es fórmula simplificable y σ_1 la versión simplificada de σ. Entonces



es *tableau* de Ψ.
Es decir, siempre podemos añadir a una rama una versión simplificada de cualquier elemento de la rama y seguimos teniendo un *tableau* de Ψ.

- *Regla* α, R_α: Sea

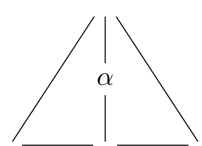
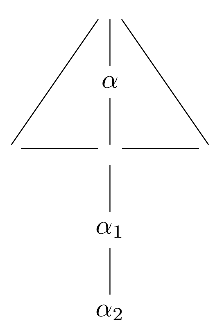


tableau de Ψ, donde α es una α-fórmula y α_1, α_2 su descomposición. Entonces



es *tableau* de Ψ.
Es decir, si hay una α-fórmula α en una rama, podemos añadir a la rama los dos elementos de su descomposición, α_1 y α_2.

- *Regla* β, R_β: Sea

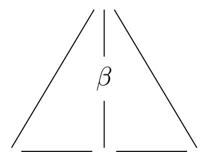
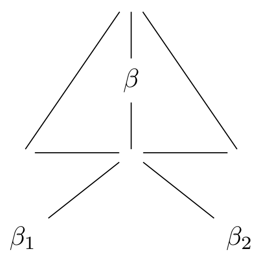


tableau de Ψ, donde β es una β-fórmula y β_1,β_2 su descomposición. Entonces



es *tableau* de Ψ.
Es decir, si hay una β-fórmula de descomposición β_1, β_2 en una rama, podemos dividir esa rama en dos ramificaciones, una que contiene a β_1 y otra que contiene a β_2.

Precisemos las nociones que describimos al inicio de esta sección. Dado un *tableau* T y una rama r, consideremos los siguientes conjuntos:

$$\Gamma_T := \{\varphi \in PROP_{SP} | \varphi \text{ está en } T\}$$

$$\Gamma_r := \{\varphi \in PROP_{SP} | \varphi \text{ está en } r\}$$

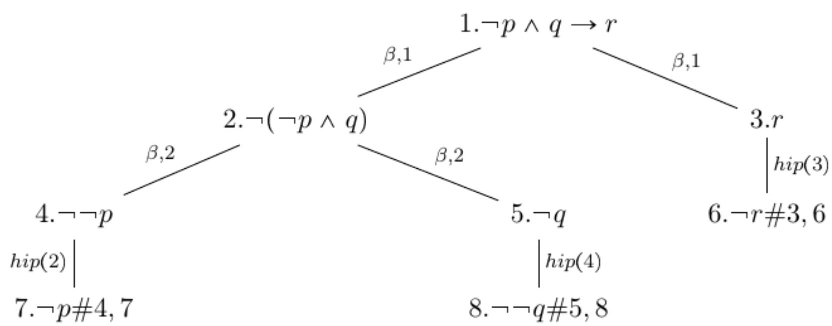
Esto nos permite establecer la siguiente

Definición 1.50. Una rama r de un *tableau* T se dice *cerrada* si se da al menos uno de los siguientes casos:

- $\bot \in \Gamma_r$.
- Existe φ tal que $\varphi, \neg\varphi \in \Gamma_r$.

Si todas las ramas de T son cerradas, T se dice *cerrado*. Si existe un *tableau* cerrado para $\Phi \cup \{\neg\varphi\}$ lo denotamos por $\Phi \vdash_{tb} \varphi$.
Decimos que una rama es *abierta* cuando no es cerrada, y decimos que un tablero es *abierto* cuando no es cerrada.

Ejemplo 1.51. Supongamos que queremos comprobar $\{\neg p \wedge q \to r, \neg p, \neg r\} \vDash \neg q$. Como veremos en breve, para ello nos basta encontrar un *tableau* cerrado para $\Psi = \{\neg p \wedge q \to r, \neg p, \neg r, \neg\neg q\}$. Enumeremos las premisas del siguiente modo: (1) $\neg p \wedge q \to r$, (2) $\neg p$, (3) $\neg r$, (4) $\neg\neg q$. La siguiente figura muestra un *tableau* que obtenemos de Ψ, y a continuación explicamos cómo se ha desarrollado a partir de las reglas de construcción:



Recordemos que el *tableau* se construye de arriba hacia abajo. Por la regla R_{ini}, el grafo de un solo vértice $\neg p \wedge q \to r$ es un *tableau* de Ψ, ya que $\neg p \wedge q \to r \in \Psi$.
Ahora bien, $\neg p \wedge q \to r$ es una β-fórmula de tipo $\beta = \varphi_1 \to \varphi_2$, de modo que, atendiendo a la tabla de las β-fórmulas, se descompone en las fórmulas $\beta_1 = \neg\varphi_1 = \neg(\neg p \wedge q)$ y $\beta_2 = \varphi_2 = r$.
Teniendo la descomposición de la β-fórmula, podemos aplicar la regla R_β, y obtenemos un *tableau* con dos ramificaciones, una con $\neg(\neg p \wedge q)$ y otra con r:

- Comenzamos con la segunda rama, que es la más sencilla. Usando la regla R_{hip}, podemos añadir a la rama cualquier elemento de Ψ, de modo que escogemos $\neg r$. Ahora bien, en esta rama tenemos las proposiciones $\neg p \wedge q \to r$, r y $\neg r$. En concreto, tenemos las proposiciones r y $\neg r$. Como en la rama hay dos proposiciones de tipo φ y $\neg\varphi$, es cerrada y hemos acabado con ella. Simbolizaremos que una rama está cerrada escribiendo # y señalando los vértices en los que se encuentran las proposiciones contradictorias.

- Fijémonos ahora la rama de $\neg(\neg p \wedge q)$. De nuevo, tenemos una β-fórmula. Usando la tabla como antes, vemos que se descompone en $\beta_1 = \neg\neg p$, $\beta_2 = \neg q$. De modo que, por la regla R_β, esta rama se divide a su vez en otras dos:

  - La primera rama tiene los elementos $\neg p \wedge q \to r$, $\neg(\neg p \wedge q)$ y $\neg\neg p$. Usando la regla R_{hip}, podemos añadir a esta rama el elemento $\neg p$ de Ψ. Ahora, dos de los elementos de esta rama son $\neg\neg p$ y $\neg p$. Es decir, tenemos las fórmulas φ y $\neg\varphi$, siendo $\varphi = \neg p$. Por tanto esta rama está cerrada.

- La segunda rama tiene los elementos $\neg p \wedge q \to r$, $\neg(\neg p \wedge q)$ y $\neg q$. Usando la regla R_{hip}, podemos añadir a esta rama el elemento $\neg\neg q$ de Ψ. Ahora, dos de los elementos de esta rama son $\neg\neg q$ y $\neg q$. Como la rama contiene una fórmula y su negación, es cerrada.

Como todas las ramas son cerradas, el *tableau* es cerrado.

Vemos a continuación otro ejemplo en el que no se cumple que $\Phi \vDash \varphi$. Es decir, hay una valoración que cumple Φ y $\neg\varphi$. En estos casos, veremos que no se puede conseguir un *tableau* cerrado de $\Phi \cup \{\neg\varphi\}$. Al intentar obtener un *tableau* cerrado, la complicación del método aumentará considerablemente pero también nos dará indicaciones sobre qué valoración de verdad satisface a Φ y $\neg\varphi$.

Ejemplo 1.52. Sea $\varphi = \neg(p_3 \to p_6)$ y Φ formado por estas seis proposiciones:

1. $p_1 \to (p_2 \vee p_3)$
2. $\neg p_1 \to (p_3 \vee p_4)$
3. $p_1 \to \neg p_6$
4. $\neg p_3 \to (p_4 \to p_1)$
5. $\neg(p_2 \wedge p_5)$
6. $p_2 \to p_5$

En la siguiente figura intentamos conseguir un *tableau* cerrado de $\Phi \cup \{\neg\varphi\}$.

Como podemos ver, no conseguimos cerrar las ramas. En la rama que hemos indicado con flechas, no parece haber forma de encontrar una contradicción, y además tenemos en esta rama las proposiciones $\neg p_1$, $\neg p_2$, p_3 y p_6. En este caso podemos imaginar que la rama no se puede cerrar y hay una valoración que cumple los elementos de esa rama. En efecto, solo tenemos 4 opciones en función de los valores de verdad que asociemos a p_4 y p_5, y encontramos que la valoración dada por $v(p_1) = v(p_2) = v(p_5) = F, v(p_3) = v(p_4) = v(p_6) = V$ cumple Φ y $\neg\varphi$. Es decir, $\Phi \nvDash \varphi$. Como veremos en breve, esto significa que no podemos encontrar un *tableau* cerrado para $\Phi \cup \{\neg\varphi\}$.

$$
\begin{tikzcd}
1.\neg\neg(p_3 \to p_6) & \\
2.p_3 \to p_6 \arrow[u, "\sigma{,}1"] \arrow[dr, dash, "\beta{,}2"] & \\
4.p_6 \arrow[u, "\beta{,}2"] & 3.\neg p_3 \\
5.p_1 \to \neg p_6 \arrow[u, "hip(3)"] \arrow[dr, dash, "\beta{,}5"] & \\
7.\neg p_1 \arrow[u, "\beta{,}5"] & 6.\neg p_6 \# 4, 6 \\
8.\neg p_1 \to (p_3 \vee p_4) \arrow[u, "hip(2)"] \arrow[dr, dash, "\beta{,}8"] & \\
10.p_3 \vee p_4 \arrow[u, "\beta{,}8"] & 9.\neg\neg p_1 \# 7, 9 \\
11.\neg p_3 \to (p_4 \to p_1) \arrow[u, "hip(4)"] \arrow[dr, dash, "\beta{,}11"] & \\
13.\neg\neg p_3 \arrow[u, "\beta{,}11"] & 12.p_4 \to p_1 \\
14.p_1 \to (p_2 \vee p_3) \arrow[u, "hip(1)"] \arrow[dr, dash, "\beta{,}14"] & \\
16.\neg p_1 \arrow[u, "\beta{,}14"] & 15.p_2 \vee p_3 \\
17.\neg(p_2 \wedge p_5) \arrow[u, "hip(5)"] \arrow[dr, dash, "\beta{,}17"] & \\
19.\neg p_2 \arrow[u, "\beta{,}17"] & 18.\neg p_5 \\
20.p_2 \to p_5 \arrow[u, "hip(6)"] \arrow[dr, dash, "\beta{,}20"] & \\
22.\neg p_2 \arrow[u, "\beta{,}20"] & 21.p_5 \\
23.p_3 \arrow[u, "\sigma{,}13"] &
\end{tikzcd}
$$

El último ejemplo muestra que hay muchas formas de intentar conseguir un *tableau* cerrado para un conjunto de fórmulas. Además, puede que dos formas distintas den lugar a *tableaux* con distinto número de proposiciones. Para obtener *tableaux* pequeños, nos conviene usar las reglas de construcción con la siguiente prioridad:

1. R_σ
2. R_α
3. R_β

Es decir, nos conviene aplicar siempre que sea posible la regla R_σ. Cuando no sea posible, aplicaremos la regla R_α. Si esta tampoco es posible, usaremos R_β.

Recordemos que lo que motivó esta sección fue, dados un conjunto de fórmulas Φ y una fórmula φ, el demostrar que $\Phi \vDash \varphi$, usando el método de los *tableaux*. Por tanto, para que el método nos permita alcanzar este objetivo tiene que cumplir dos propiedades:

- *Completitud:* $\Phi \vDash \varphi$ implica $\Phi \vdash_{tb} \varphi$. Es decir, si se cumple que $\Phi \vDash \varphi$, entonces podemos encontrar un *tableau* cerrado para $\Phi \cup \{\neg\varphi\}$.
- *Corrección:* $\Phi \vdash_{tb} \varphi$ implica $\Phi \vDash \varphi$. Es decir, si encontramos un *tableau* cerrado para $\Phi \cup \{\neg\varphi\}$, entonces se cumple que $\Phi \vDash \varphi$.

1.8.2. Corrección

Comenzamos demostrando la propiedad de corrección. El siguiente resultado condensa las ideas expuestas hasta ahora:

Teorema 1.53. *Sean $\Phi \subseteq PROP_{SP}$ tal que $Sat(\Phi)$ y T tableau finito para Φ. Entonces existe una rama de T, r, tal que $Sat(\Gamma_r)$. Es más: si una valoración v cumple $v \vDash \Phi$, hay una rama r que cumple $v \vDash \Gamma_r$.*

Demostración. Todo *tableau* finito T se construye partiendo de
$$\begin{array}{c} \varphi_1 \\ | \\ \varphi_2 \\ \vdots \\ \varphi_k \end{array}$$
y aplicando reglas de construcción de *tableaux* $R_1, ..., R_n$. Nos interesa hacer inducción sobre el número de reglas empleadas, n.

Así pues, sea una valoración v tal que $v \vDash \Phi$.

Si $n = 0$, por la regla inicial tenemos una única rama r tal que $\Gamma_r = \{\varphi_1, ..., \varphi_k\} \subseteq \Phi$. Como $v \vDash \Phi$, tenemos que $v \vDash \Gamma_r$.

Ahora sea $n > 0$, supongamos que el resultado se cumple para $n - 1$.

Sea Φ satisfactible, y sea T un *tableau* obtenido aplicando la regla inicial y otras reglas $R_1, \ldots, R_n$. Es decir, T se obtiene aplicando la regla R_n al *tableau* T' obtenido aplicando las reglas $R_1, \ldots, R_{n-1}$. Por hipótesis de inducción, tenemos una rama r de T' tal que $v \vDash \Gamma_r$. Al aplicar la regla R_n extendemos una rama de T' para obtener T.

Si la rama que extendemos no es r, entonces r también es una rama de T, por tanto, como dice el enunciado, $v \vDash \Gamma_r$ para una rama r de T.

Si la rama que extendemos es r, distinguimos los siguientes casos:

- $R_n = R_\sigma$. Para cierta fórmula $\sigma \in \Gamma_r$, añadimos σ_1 tal que $\sigma \sim \sigma_1$. Por tanto, como $v \vDash \Gamma_r$, en particular $v \vDash \sigma$ y así $v \vDash \sigma_1$. Entonces $v \vDash \Gamma_r \cup \{\sigma_1\}$, es decir, la rama r' formada al añadir σ_1 a r es la buscada.

- $R_n = R_\alpha$. Para cierta fórmula $\alpha \in \Gamma_r$, añadimos α_1, α_2 tales que $\alpha \sim \alpha_1 \wedge \alpha_2$. Por tanto, como $v \vDash \Gamma_r$, en particular $v \vDash \alpha$ y así $v \vDash \alpha_1$ y $v \vDash \alpha_2$. Entonces $v \vDash \Gamma_r \cup \{\alpha_1, \alpha_2\}$, es decir, la rama r' formada al añadir α_1 y α_2 a r es la buscada.

- $R_n = R_\beta$. Para cierta fórmula $\beta \in \Gamma_r$, añadimos β_1, β_2 tales que $\beta \sim \beta_1 \vee \beta_2$, formando dos nuevas ramas. Como $v \vDash \Gamma_r$, en particular $v \vDash \beta$. Esto quiere decir que o bien $v \vDash \beta_1$ o bien $v \vDash \beta_2$. En el primer caso, $v \vDash \Gamma_r \cup \{\beta_1\}$, por tanto la rama r_1 formada añadiendo β_1 a r cumple que $v \vDash \Gamma_{r_1}$. En el segundo caso, $v \vDash \Gamma_r \cup \{\beta_2\}$, por tanto la rama r_2 formada añadiendo β_2 a r cumple que $v \vDash \Gamma_{r_2}$. En ambos casos, tenemos una rama r' de T tal que $v \vDash r'$, por tanto se cumple el enunciado.

- $R_n = R_{hip}$. Añadimos a la rama r cierta $\varphi \in \Phi$. Por hipótesis, tenemos que $v \vDash \Phi$ y $v \vDash \Gamma_r$, luego en especial $v \vDash \Gamma_r \cup \{\varphi\}$. Es decir, la rama r' formada al añadir a r la hipótesis cumple $v \vDash r'$.

$\square$

Corolario 1.54. *Sea $\Phi \subseteq PROP_{SP}$. Si Φ tiene un tableau cerrado, entonces $Insat(\Phi)$.*

Demostración. Se trata de la forma contrapositiva del enunciado 1.53. $\square$

Corolario 1.55. *(Corrección) Sean $\Phi \subseteq PROP_{SP}, \varphi \in PROP_{SP}$. Si $\Phi \vdash_{tb} \varphi$, entonces $\Phi \vDash \varphi$.*

Demostración. Si $\Phi \vdash_{tb} \varphi$ entonces $\Phi \cup \{\neg\varphi\}$ tiene un *tableau* cerrado. Por 1.54 se sigue que $Insat(\Phi \cup \{\neg\varphi\})$, es decir, aplicando 1.30, $\Phi \vDash \varphi$. $\square$

1.8.3. Completitud

Ahora pasamos a estudiar la completitud. Para ello es necesario introducir previamente varias nociones:

Definición 1.56. $\Phi \subseteq PROP_{SP}$ se dice *coherente* si no existe φ tal que $\varphi, \neg\varphi \in \Phi$ y $\perp \notin \Phi$.

Definición 1.57. $\Phi \subseteq PROP_{SP}$ se dice *saturado* si, usando la notación de las σ, α y β-fórmulas:

- $\sigma \in \Phi$ implica que $\sigma_1 \in \Phi$.
- $\alpha \in \Phi$ implica que $\alpha_1, \alpha_2 \in \Phi$.
- $\beta \in \Phi$ implica que $\beta_1 \in \Phi$ o $\beta_2 \in \Phi$.

Por ejemplo, si a un *tableau* T no le podemos aplicar más reglas de forma que se añadan nuevas proposiciones al *tableau*, entonces Γ_T es saturado.

Definición 1.58. $\Phi \subseteq PROP_{SP}$ se dice *de Hintikka* si es coherente y saturado.

En relación a los *tableaux*, nos gustaría poder tratar la 'complejidad' de las distintas fórmulas. Esto nos lleva a la siguiente:

Definición 1.59. Definimos la norma $||.|| : PROP_{SP} \to \mathbb{N}$ dada recursivamente por:

- $||\varphi|| = 0$, para toda $\varphi \in AT$.
- $||\neg\varphi|| = 1 + ||\varphi||$.
- $||\varphi_1 \wedge \varphi_2|| = 1 + ||\varphi_1|| + ||\varphi_2||$.
- $||\varphi_1 \vee \varphi_2|| = 1 + ||\varphi_1|| + ||\varphi_2||$.
- $||\varphi_1 \to \varphi_2|| = 2 + ||\varphi_1|| + ||\varphi_2||$.
- $||\varphi_1 \leftrightarrow \varphi_2|| = 5 + 2||\varphi_1|| + 2||\varphi_2||$.

La siguiente proposición es fácil de demostrar por casos y justifica nuestro tratamiento de las descomposiciones de las σ, α y β-fórmulas como expresiones reducidas de las fórmulas originales:

Proposición 1.60. *Sean σ, α y β σ, α y β-fórmulas, respectivamente, con $\sigma \sim \sigma_1$, $\alpha \sim \alpha_1 \wedge \alpha_2$, $\beta \sim \beta_1 \vee \beta_2$. Entonces:*

1. $||\sigma|| > ||\sigma_1||$.
2. $||\alpha|| > ||\alpha_1||$ *y* $||\alpha|| > ||\alpha_2||$.
3. $||\beta|| > ||\beta_1||$ *y* $||\beta|| > ||\beta_2||$.

Definición 1.61. (Valoración de Hintikka) Sea $H \subseteq PROP_{SP}$ de Hintikka. Definimos la valoración $v_H : SP \to \{V, F\}$ como:

- $v_H(p) = V$, si $p \in H$.
- $v_H(p) = F$, si $\neg p \in H$.

En caso de que $p, \neg p \notin H$, va a ser para nuestros propósitos indiferente el valor que toma v_H. Para que v_H esté bien definida, podemos suponer por ejemplo $v_H(p) = V$ si $p, \neg p \notin H$.

Proposición 1.62. *Si $H \subseteq PROP_{SP}$ es de Hintikka, $v_H \vDash H$. En particular, H es satisfactible.*

Demostración. Tenemos que demostrar que para toda $\varphi \in H$, $v_H \vDash \varphi$. Para ello procedemos por inducción sobre $||\varphi||$. Es necesario distinguir dos casos base primero.

Si $||\varphi|| = 0$ entonces $\varphi \in AT$ y, al ser H coherente, $\varphi \in SP$ o $\varphi = \top$. Si se da lo primero, entonces se da el resultado por la definición de v_H. Lo segundo es trivial, porque toda valoración satisface a $\top$.

Si $||\varphi|| = 1$, por la coherencia de H tenemos las siguientes posibilidades:

$\varphi = \neg p,\ \neg\bot,\ p \vee \bot,\ \bot \vee p,\ p \vee \top,\ \top \vee p,\ p \wedge \top,\ \top \wedge p,\ p \wedge q,\ p \vee q,\ \top \vee \top,\ \top \wedge \top,\ \top \vee \bot,\ \bot \vee \top,$

para $p, q \in SP$. En el primer caso obtenemos que $v_H(p) = F$ y por tanto que $\hat{v}_H(\neg p) = V$. En el segundo caso está claro que $\hat{v}_H(\neg\bot) = V$. El resto de casos se deduce de forma similar.

Sea φ con $||\varphi|| > 1$, supongamos que el resultado se cumple para todo $n < ||\varphi||$. Por la proposición 1.49 tenemos las siguientes posibilidades:

- $\varphi = \sigma$ es σ-fórmula, con $\sigma \sim \sigma_1$. Al ser H saturado, $\sigma_1 \in H$. Sabemos de 1.60 que $||\sigma|| > ||\sigma_1||$, luego podemos aplicar la hipótesis de inducción y así $\hat{v}_H(\sigma_1) = V$, de lo que se sigue que $\hat{v}_H(\sigma_1) = \hat{v}_H(\sigma) = V$.

- $\varphi = \alpha$ es α-fórmula, con $\alpha \sim \alpha_1 \wedge \alpha_2$. Al ser H saturado, $\alpha_1, \alpha_2 \in H$ y, por 1.60, $||\alpha|| > ||\alpha_1||$ y $||\alpha|| > ||\alpha_2||$. Aplicando la hipótesis de inducción, $\hat{v}_H(\alpha_1) = \hat{v}_H(\alpha_2) = V$, luego $\hat{v}_H(\alpha) = \hat{v}_H(\alpha_1 \wedge \alpha_2) = V$.

- $\varphi = \beta$ es β-fórmula, con $\beta \sim \beta_1 \vee \beta_2$. Por 1.60, $||\beta|| > ||\beta_1||$ y $||\beta|| > ||\beta_2||$. Además, como H es saturado, o bien $\beta_1 \in H$ y, por hipótesis de inducción, $\hat{v}(\beta_1) = V$, o bien $\beta_2 \in H$ y, por hipótesis de inducción, $\hat{v}(\beta_2) = V$. En ambos casos, $\hat{v}(\beta) = \hat{v}(\beta_1 \vee \beta_2) = V$.

$\square$

Corolario 1.63. *Sean $\Phi \subseteq PROP_{SP}$, T tableau abierto de Φ. Si:*

1. *Existe una rama abierta de T, r, tal que $\Phi \subseteq \Gamma_r$.*

2. Γ_r *es de Hintikka.*

entonces Φ *es satisfactible.*

Demostración. Como Γ_r es de Hintikka, por 1.62 es satisfactible, por tanto como $\Phi \subseteq \Gamma_r$, Φ es satisfactible.

$\square$

Esto motiva la siguiente definición:

Definición 1.64. Sean $\Phi \subseteq PROP_{SP}$, T *tableau* de Φ. T se dice *completo* si toda rama abierta de T, r, verifica:

- $\Phi \subseteq \Gamma_r$.
- Γ_r es de Hintikka.

Proposición 1.65. *Sea* $\Phi \subseteq PROP_{SP}$ *finito. Entonces existe un tableau completo para* Φ*.*

Demostración. En 2.85 probaremos una generalización de este teorema para lógica de primer orden.

$\square$

Corolario 1.66. *(Completitud) Sea* $\Phi \subseteq PROP_{SP}$ *finito. Si* $\Phi \vDash \varphi$ *entonces* $\Phi \vdash_{tb} \varphi$*.*

Demostración. $\Phi \vDash \varphi$ equivale por 1.30 a $Insat(\Phi \cup \{\neg\varphi\})$. Se sigue de 1.65 que podemos considerar un *tableau* T completo para $\Phi \cup \{\neg\varphi\}$. Veamos que T no puede tener ramas abiertas:

Si T tiene una rama abierta r, por definición de *tableau* completo, $\Phi \cup \{\neg\varphi\} \subseteq \Gamma_r$ y Γ_r es de Hintikka. Pero entonces, por el corolario 1.63, $\Phi \cup \{\neg\varphi\}$ sería satisfactible, lo cual contradice el enunciado.

Por tanto el *tableau* es cerrado , es decir, $\Phi \vdash_{tb} \varphi$. $\square$

2 | Lógica de primer orden

2.1. Introducción

En el anterior capítulo, construimos con éxito un lenguaje formal que nos permitía traducir frases informales del español a expresiones formales, además de formalizar los conceptos de implicación y equivalencia lógica. Sin embargo, se puede reprochar que la lógica proposicional es demasiado *simple* en el sentido siguiente:

Consideremos el silogismo:

Todo hombre es mortal
Sócrates es un hombre.

Sócrates es mortal $\quad \therefore$

En este caso, podríamos pensar que la consecuencia lógica se trata de una del tipo $p \wedge (p \to q) \to q$. Pero, por otro lado, parece evidente que depende de elementos más básicos que los símbolos de proposición. Sería, entonces, más conveniente una formalización del tipo:

Para todo x, si *x es hombre* entonces *es mortal*
Sócrates es hombre.

Sócrates es mortal $\quad \therefore$

Hemos empleado los términos 'hombre', 'mortal' y 'para todo' en un sentido puramente formal. Como ocurría con las proposiciones, existen múltiples frases y expresiones informales distintas que corresponden al silogismo que acabamos de exponer.

A continuación vamos a definir, igual que en lógica proposicional, el alfabeto que usaremos para construir fórmulas en los lenguajes de primer orden. Pero además, en este caso, habrá múltiples lenguajes de primer orden, y cada uno tendrá unos ciertos elementos que lo caractericen, que resumimos en el concepto de 'signatura':

Definición 2.1. Una *signatura* S es una tupla $\langle Ct_S, Fn_S, Pd_S \rangle$ donde:

- Ct_S es el conjunto de símbolos de constante.
- Fn_S es el conjunto de símbolos de función con determinada aridad[1].
- Pd_S es el conjunto de símbolos de predicado con determinada aridad.

Dado un símbolo Γ de función o predicado, denotamos por $\Gamma|_n$ que es n-ario.

Ejemplo 2.2. La signatura $Nat := \langle \{0\}, \{+|_2, s|_1\}, \{<|_2\} \rangle$ para los números naturales. Es decir:

- Hay un símbolo de constante, 0.
- Los símbolos de función serán el símbolo de función binaria $+$, que llamaremos 'suma', y el símbolo de función 1-aria, s, que llamaremos 'sucesor'.
- Tenemos un símbolo de predicado 2-ario, $<$.

A veces en la signatura Nat incluiremos un otro símbolo de función 2-aria, $*$, que llamaremos 'producto'.

Definición 2.3. Dada la signatura S, definimos el alfabeto asociado como:

$$A_S := Ct_S \cup Fn_S \cup Pd_S \cup Var \cup \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \top, \bot\} \cup \{(,)\} \cup \{\exists, \forall\} \cup \{\doteq\},$$

donde:

- Ct_S, Fn_S, Pd_S vienen dados por 2.1.
- Var son los símbolos de variable: $\{x, y, z, x_1, \dots\}$.
- $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \top, \bot, (,)$ son los símbolos de conectiva y los paréntesis, al igual que en lógica proposicional.
- $\forall$ (para todo) y $\exists$ (existe) son los *cuantificadores lógicos*. Llamamos a $\forall$ *cuantificador universal* y a $\exists$ *cuantificador existencial*.
- $\doteq$, el símbolo de igualdad. Usamos $\doteq$ en vez de $=$ para distinguirlo de la igualdad de fórmulas. Por ejemplo, si escribimos $\varphi = t \doteq s$ queremos decir que φ es igual a $t \doteq s$.

Como ocurría con las proposiciones, nos interesa distinguir las expresiones del alfabeto anterior que están bien formadas. Para ello, primero necesitaremos definir los *términos*, que podemos interpretar como las expresiones que usaremos para nombrar objetos, y después las *fórmulas*, expresiones que usaremos para denotar afirmaciones sobre los objetos.

[1] Por ahora, nos referimos por *aridad* de un símbolo de función o de predicado como el número de argumentos que admite. Más adelante especificaremos lo que significa esta idea.

Definición 2.4. Dada la signatura S, el *conjunto de términos* de S, $TERM_S$, es el menor subconjunto de A_S^* que verifica:[2]

1. $Ct_S \subseteq TERM_S$.

2. $Var \subseteq TERM_S$.

3. Si $f|_n \in Fn_S$ y $t_1, .., t_n \in TERM_S$, entonces $f(t_1, ..., t_n) \in TERM_S$.

Si f es una función 2-aria, a veces usaremos la notación tradicional xfy en vez de $f(x, y)$. Por ejemplo, en Nat diremos $x + y$ en vez de $+(x, y)$.

Ejemplo 2.5. Siguiendo con la signatura Nat, algunos ejemplos de elementos de $TERM_{NAT}$ serían $0, s(s(0)), x, y, z, +(s(s(0)), s(0))$ y $s(+(x, s(s(s(0)))))$.

La siguiente proposición nos da una definición constructiva de $TERM_S$:

Proposición 2.6. *Sea $S = \langle Ct_S, Fn_S, Pd_S \rangle$ signatura. Definimos los conjuntos:*

$$T_S^0 := Ct_S \cup Var$$

$$T_S^{n+1} := T_S^n \cup \bigcup_{k \in \mathbb{N}} \{f(t_1, \ldots, t_k) \mid t_1, \ldots, t_k \in T_S^n,\ f|_k \in Fn_S\}$$

Entonces $\bigcup_{i \in \mathbb{N}} T_S^i = TERM_S$.

Demostración. La demostración es análoga a la de 1.8. □

Ahora podemos construir fórmulas a partir de estos elementos básicos que ya hemos definido. Comenzamos por

Definición 2.7. Dada la signatura S, el *conjunto de fórmulas atómicas* de S, $FORMAT_S$, es el menor subconjunto de A_S^* que verifica:

1. Si $t_1, t_2 \in TERM_S$, $t_1 \doteq t_2 \in FORMAT_S$.

2. Si $R|_n \in Pd_S$ y $t_1, \ldots, t_n \in TERM_S$, $R(t_1, \ldots, t_n) \in FORMAT_S$.

3. $\top, \bot \in FORMAT_S$.

Si R es un predicado 2-ario, a veces usaremos la notación tradicional xRy en vez de $R(x, y)$. Por ejemplo, en Nat diremos $x < y$ en vez de $< (x, y)$.

Definición 2.8. Dada la signatura S, el *conjunto de fórmulas* de S, $FORM_S$, es el menor subconjunto de A_S^* que verifica:

1. $FORMAT_S \subseteq FORM_S$.

2. Si $\varphi_1, \varphi_2 \in FORM_S$, $(\neg\varphi_1), (\varphi_1 \Box \varphi_2) \in FORM_S$.

3. Si $x \in Var$ y $\varphi \in FORM_S$, $(Qx\,\varphi) \in FORM_S$, siendo $Q \in \{\forall, \exists\}$

[2] Recordemos que A_S^* es el cierre de Kleene de A_S, como definimos en 1.4.

A partir de ahora, usaremos para mayor brevedad el símbolo Q como intercambiable por $\forall$ o $\exists$ a no ser que se indique lo contrario.

Damos una definición constructiva de $FORM_S$:

Proposición 2.9. *Sea $S = \langle Ct_S, Fn_S, Pd_S\rangle$ signatura. Definimos los conjuntos:*

$$F_S^0 := FORMAT_S$$

$$F_S^{n+1} := F_S^n \cup \{(\neg\varphi) \mid \varphi \in F_S^n\} \cup \{(\varphi \Box \psi) \mid \varphi, \psi \in F_S^n\} \cup \{(Qx\,\varphi) \mid x \in Var,\ \varphi \in F_S^n\}$$

Entonces $\bigcup\limits_{i\in\mathbb{N}} F_S^i = FORM_S$.

Demostración. De nuevo, la demostración es análoga a la de 1.8. $\square$

2.2. Inducción estructural y recursión

Tal y como ocurría con la lógica proposicional, la recursión y la inducción estructural son las dos principales herramientas empleadas en las definiciones y demostraciones de la lógica de primer orden. Sin embargo, ahora tenemos que tratar separadamente los conjuntos $TERM_S$ y $FORM_S$ asociados a cierta signatura S.

Comenzamos con los teoremas de inducción estructural y recursión para $TERM_S$, dada una signatura $S = \langle Ct_S, Fn_S, Pd_S\rangle$:

Proposición 2.10. *(Inducción estructural) Sean $S = \langle Ct_S, Fn_S, Pd_S\rangle$ una signatura y P una propiedad. Si se cumple que:*

1. *P es válida para todo elemento de T_S^0.*
2. *Sea $f|_k \in Fn_S$. Si P es válida para todo elemento de T_S^n, entonces es válida para $f(t_1, \dots t_k)$, con $t_1, \dots, t_k \in T_S^n$.*

Entonces P es válida para todo elemento de $TERM_S$.

Demostración. Si la propiedad P cumple 1, 2 y 3, entonces el conjunto de términos que cumplen P cumple las tres propiedades de la definición 2.4, por tanto contiene a $TERM_S$. $\square$

Proposición 2.11. *El esquema de definición recursiva da como resultado una única función, es decir, dadas:*

1. *$F_0 : Ct_S \cup Var \to A$.*
2. *$F_f : A^k \to A$, para cada función $f|_k \in Fn_S$.*

existe una única función $F : TERM_S \to A$ tal que:

1. *$F(t) = F_0(t)$, para todo $t \in Ct_S \cup Var$.*

2. $F(f(t_1, \dots, t_k)) = F_f(F(t_1), \dots, F(t_k))$*, para cada* $f|_k \in Fn_S$ *y* $t_1, \dots, t_k \in TERM_S$*.*

Demostración. Se omite la demostración. $\square$

Ejemplo 2.12. Una función que nos será de utilidad será var, que nos lleva cada elemento de $TERM_S$ al conjunto de variables que aparecen en él. La definimos recursivamente como:

1. Caso base: $var_0 : Ct_S \cup Var \to \mathcal{P}(Var)$, dada por $var_0(c) = \emptyset$ si $c \in Ct_S$ y $var(x) = \{x\}$ si $x \in Var$.
2. Caso recursivo: Dado $f|_k \in Fn_S$, $var_f : (\mathcal{P}(Var))^k \to \mathcal{P}(Var)$, dada por $var_f(f(t_1, \dots t_k)) = \bigcup\limits_{i=1}^{k} var(t_i)$.

Ahora enunciamos los teoremas de inducción estructural y recursión para $FORM_S$, dada una signatura $S = \langle Ct_S, Fn_S, Pd_S \rangle$:

Proposición 2.13. *(Inducción estructural) Sean* $S = \langle Ct_S, Fn_S, Pd_S \rangle$ *una signatura y* P *una propiedad. Si se cumple que:*

1. P *es válida para todo elemento de* $FORMAT_S$*.*
2. *Si* $\varphi_1, \varphi_2 \in FORM_S$ *y se cumplen* $P(\varphi_1), P(\varphi_2)$*, entonces tenemos* $P((\neg\varphi_1))$ *y* $P((\varphi_1 \square \varphi_2))$*.*
3. *Si* $x \in Var$ *y* $\varphi \in FORM_S$*, y se cumple* $P(\varphi)$*, entonces tenemos* $P((Qx\varphi))$*.*

Entonces P *es válida para todo elemento de* $FORM_S$*.*

Demostración. Si la propiedad P cumple 1, 2 y 3, entonces el conjunto de términos que cumplen P cumple las tres propiedades de la definición 2.8, por tanto contiene a $FORM_S$. $\square$

Proposición 2.14. *El esquema de definición recursiva da como resultado una única función, es decir, dadas:*

1. $F_{AT} : FORMAT_S \to A$*.*
2. $F_\neg : A \to A$*.*
3. $F_\square : A \times A \to A$*.*
4. $F_Q : Var \times A \to A$*.*

existe una única función $F : FORM_S \to A$ *tal que:*

1. $F(\varphi) = F_{AT}(\varphi)$*, para toda* $\varphi \in FORMAT_S$*.*
2. $F((\neg\varphi)) = F_\neg(F(\varphi))$*.*
3. $F((\varphi_1 \square \varphi_2)) = F_\square(F(\varphi_1), F(\varphi_2))$*.*

4. $F((Qx\,\varphi)) = F_Q(x, F(\varphi))$.

Demostración. Se omite la demostración. $\square$

Ejemplo 2.15. Extendamos la función var al conjunto $FORM_S$. Por comodidad, omitimos las funciones auxiliares:

1. Caso base:
   - $var(\top) = var(\bot) = \emptyset$.
   - Sean $p|_k \in Pd_S$, $t_1, \dots t_n \in TERM_S$. Entonces $var(p(t_1, \dots t_n)) = \bigcup_{i=1}^{k} var(t_i)$.
   - Sean $t, s \in TERM_S$. Entonces $var(t \doteq s) = var(t) \cup var(s)$.
2. Caso recursivo:
   - $var((\neg\varphi)) = var(\varphi)$.
   - $var((\varphi \square \psi)) = var(\varphi) \cup var(\psi)$.
   - $var((Qx\,\varphi)) = \{x\} \cup var(\varphi)$.

2.3. Eliminación de paréntesis

Al igual que en lógica proposicional, nos serán útiles unas reglas de omisión de paréntesis:

1. Los paréntesis externos pueden omitirse. Esto no da lugar a ambigüedad.
2. Conectivas y cuantificadores se aplicarán en este orden: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$.
3. Cuando hay varias conectivas del mismo tipo seguidas, se asocia siempre por la izquierda.

Un par de ejemplos de aplicación de estas reglas en fórmulas con cuantificadores:

$$\begin{array}{rcl} \forall x\,\varphi \rightarrow \psi & \text{es} & (\forall x\,(\varphi \rightarrow \psi)) \\ \exists x\,\varphi \wedge \forall y \psi & \text{es} & (\exists x\,(\varphi \wedge (\forall y \psi))) \end{array}$$

De nuevo, las reglas que adoptamos no son universales y varían de autor a autor (por ejemplo, en muchos casos los cuentificadores tienen máxima prioridad, en vez de mínima prioridad como supondremos en este manual).

2.4. Variables libres y ligadas

Consideremos las siguientes fórmulas de primer orden:

- $\forall x\, x \doteq 3$
- $x \doteq 3$

La primera fórmula se puede traducir informalmente como 'para todo x, x es igual a 3'. Intuitivamente, podemos decir que el cuantificador $\forall$ está afectando al significado de x. Sin embargo, en la segunda fórmula, x no aparece afectada por ningún cuantificador. En general, diremos que una variable es *ligada* en una fórmula φ si siempre aparece afectada por un cuantificador, como en el primer ejemplo, y diremos que es una variable *libre* en φ si aparece alguna vez sin estar afectada por un cuantificador. Formalicemos estos conceptos:

Definición 2.16. Sea $S = \langle Ct_S, Fn_S, Pd_S \rangle$ signatura. Definimos recursivamente la función $lib : FORM_S \to \mathcal{P}(Var)$, que nos lleva cada fórmula al conjunto de sus variables libres:

1. Caso base: Sea $\varphi \in FORMAT_S$. Entonces, $lib(\varphi) = var(\varphi)$.
2. Caso recursivo:
   - $lib(\neg\varphi) = lib(\varphi)$.
   - $lib(\varphi \Box \psi) = lib(\varphi) \cup lib(\psi)$.
   - $lib(Qx\, \varphi) = lib(\varphi) \setminus \{x\}$.

Definición 2.17. Sea $S = \langle Ct_S, Fn_S, Pd_S \rangle$ signatura. Definimos el *conjunto de sentencias*, $SENT_S$, como el formado por aquellas $\varphi \in FORM_S$ tales que $lib(\varphi) = \emptyset$.

2.5. Álgebras e interpretaciones

Hasta ahora hemos venido considerando cuestiones sintácticas sobre las fórmulas. Lo que queremos ahora es abordar su semántica, esto es, su comportamiento cuando les damos una interpretación determinada. Comenzamos definiendo la estructura de la que tomamos el significado de los símbolos de una signatura:

Definición 2.18. Sea $S = \langle Ct_S, Fn_S, Pd_S \rangle$ una signatura. Una *S-álgebra* o *S-estructura* es una tupla

$$\mathfrak{A} := \langle A, \{c^{\mathfrak{A}} \mid c \in Ct_S\}, \{f^{\mathfrak{A}} \mid f \in Fn_S\}, \{p^{\mathfrak{A}} \mid p \in Pd_S\} \rangle$$

De modo que:

- A, el *conjunto soporte*, verifica que $A \neq \emptyset$.

- Si $c \in Ct_S$, $c^{\mathfrak{A}} \in A$.
- Si $f|_k \in Fn_S$, $f^{\mathfrak{A}} : A^k \to A$.
- Si $p|_k \in Pd_S$, $p^{\mathfrak{A}} : A^k \to Bool$.

Por convenio, un símbolo f de función 0-aria tendrá asociada una función $f^{\mathfrak{A}} : \{\emptyset\} \to A$. Es decir, será una función desde el conjunto de un elemento $\{\emptyset\}$ en A. Está claro que estas funciones pueden identificarse con elementos de A, identificando f con $f(\emptyset)$. De modo que las funciones 0-arias van a llevar a cabo esencialmente la misma función que las constantes, y de hecho, aunque aquí usaremos constantes, toda la lógica de primer orden se puede desarrollar sin constantes usando funciones 0-arias.

De igual forma, un símbolo p de predicado 0-ario tendrá asociada una función $p^{\mathfrak{A}} : \{\emptyset\} \to Bool$. Es decir, igual que con las funciones 0-arias, podemos asociar a cada predicado 0-ario un valor de $Bool$, V o F. En esto se asemejarán a los símbolos de proposición de lógica proposicional, a los cuales asignábamos (mediante las asignaciones de verdad) un valor de $Bool$.

Ejemplo 2.19. Consideremos la signatura $Nat := \langle\{0\}, \{+|_2, *|_2, s|_1\}, \{< |_2\}\rangle$ que presentamos previamente. La forma natural (aunque no la única) de asignar significados a estos símbolos sería la S-álgebra:

$$\mathfrak{A}_{Nat} := \langle \mathbb{N}, \{0^{\mathfrak{A}_{Nat}}\}, \{+^{\mathfrak{A}_{Nat}}, *^{\mathfrak{A}_{Nat}}, s^{\mathfrak{A}_{Nat}}\}, \{<^{\mathfrak{A}_{Nat}}\}\rangle, \text{ donde:}$$

- $\mathbb{N}$ es el conjunto de los números naturales: $\{0, 1, 2, \dots\}$.
- $0^{\mathfrak{A}_{Nat}}$ será el número natural 0.
- $+^{\mathfrak{A}_{Nat}}$ será la función suma en los naturales, $+ : \mathbb{N}^2 \to \mathbb{N}; (m, n) \mapsto m+n$.

  $*^{\mathfrak{A}_{Nat}}$ será la función producto en los naturales, $* : \mathbb{N}^2 \to \mathbb{N}; (m, n) \mapsto m * n$.
  $s^{\mathfrak{A}_{Nat}}$ será la función sucesor en los naturales, $s : \mathbb{N} \to \mathbb{N}; n \mapsto n + 1$.
- $<^{\mathfrak{A}_{Nat}}$: $\mathbb{N}^2 \to Bool$ será la función que cumpla $<^{\mathfrak{A}_{Nat}} (m, n) = V$ si $m < n$, y $<^{\mathfrak{A}_{Nat}} (m, n) = F$ en caso contrario. Es decir, $<^{\mathfrak{A}_{Nat}} (m, n)$ será verdadero si y solo si $m < n$.

Es importante distinguir los símbolos de sus objetos asociados, ya que en algunos casos, como 0, suma y producto en este ejemplo, representamos igual ambas cosas. En el siguiente ejemplo habrá una clara distinción entre los símbolos y sus objetos asociados.

Ejemplo 2.20. Vamos a ver otro ejemplo de álgebra para la signatura Nat. Sea $\mathfrak{A} = \langle\{\triangle, \bigcirc\}, \{0^{\mathfrak{A}}\}, \{+^{\mathfrak{A}}, *^{\mathfrak{A}}, s^{\mathfrak{A}}\}, \{<^{\mathfrak{A}}\}\rangle$ tal que:

- $0^{\mathfrak{A}} = \triangle$.

- $s^{\mathfrak{A}} : \{\triangle, \bigcirc\} \to \{\triangle, \bigcirc\}$, $x \mapsto \triangle$.
- $*^{\mathfrak{A}}, +^{\mathfrak{A}} : \{\triangle, \bigcirc\}^2 \to \{\triangle, \bigcirc\}$, $(x, y) \mapsto \bigcirc$.
- $<^{\mathfrak{A}} : \{\triangle, \bigcirc\}^2 \to Bool$, $(x, y) \mapsto V$.

Como veremos en interpretaciones, la expresión

$$*(+(s(0), s(0)), s(s(s(0))))$$

en $\mathfrak{A}$ se refiere a $\bigcirc$ ya que, desarrollando, esa expresión corresponde a:

$$\begin{aligned}
&*^{\mathfrak{A}}\ (+^{\mathfrak{A}}(s^{\mathfrak{A}}(0^{\mathfrak{A}}), s^{\mathfrak{A}}(0^{\mathfrak{A}})), s^{\mathfrak{A}}(s^{\mathfrak{A}}(s^{\mathfrak{A}}(0^{\mathfrak{A}}))))\\
&= *^{\mathfrak{A}}(+^{\mathfrak{A}}(s^{\mathfrak{A}}(\triangle), s^{\mathfrak{A}}(\triangle)), s^{\mathfrak{A}}(s^{\mathfrak{A}}(s^{\mathfrak{A}}(\triangle))))\\
&= *^{\mathfrak{A}}(+^{\mathfrak{A}}(\triangle, \triangle), s^{\mathfrak{A}}(s^{\mathfrak{A}}(\triangle)))\\
&= *^{\mathfrak{A}}(\bigcirc, s^{\mathfrak{A}}(\triangle))\\
&= *^{\mathfrak{A}}(\bigcirc, \triangle)\\
&= \bigcirc
\end{aligned}$$

, mientras que en $\mathfrak{A}_{Nat}$ podemos ver de forma similar que la expresión se refiere a 6.
Sin embargo, si intentamos ver a qué se refiere el término $+(s(x), s(0))$, tiene una variable que todavía no sabemos como interpretar. Informalmente, las variables suelen referirse a objetos que no hemos determinado. En nuestra segunda álgebra, x podría referirse a $\triangle$ o a $\bigcirc$. En ambos casos, $+(s(x), s(0))$ se refiere a $\bigcirc$ en $\mathfrak{A}$. Sin embargo, en $\mathfrak{A}_{Nat}$, el valor que tome la expresión dependerá del valor que tome la x.

Lo anterior muestra que es necesario definir una función especial para dotar de significado a las variables:

Definición 2.21. Sea S signatura. Una *S-interpretación* es una tupla $\mathfrak{I} := \langle \mathfrak{A}, \sigma \rangle$ con:

- $\mathfrak{A}$ es *S-álgebra* de soporte A.
- $\sigma : Var \to A$.

Ejemplo 2.22. Así, en el ejemplo anterior, tomando una Nat-interpretación $\sigma : Var \to \mathbb{N}$, podemos interpretar $+(s(x), s(0))$: si por ejemplo $\sigma(x) = 7$, estamos 'asignando' a la variable x el valor 7, por tanto a la expresión total le asignaríamos el valor:

$$\begin{aligned}
&+^{\mathfrak{A}_{Nat}}\ (s^{\mathfrak{A}_{Nat}}(\sigma(x)), s^{\mathfrak{A}_{Nat}}(0^{\mathfrak{A}_{Nat}}))\\
&= +^{\mathfrak{A}_{Nat}}(s^{\mathfrak{A}_{Nat}}(7), s^{\mathfrak{A}_{Nat}}(0))\\
&= +^{\mathfrak{A}_{Nat}}(8, 1)\\
&= 9
\end{aligned}$$

Ahora extendemos este concepto a términos, como ya hemos hecho intuitivamente en los dos últimos ejemplos:

Definición 2.23. Sea $S = \langle Ct_S, Fn_S, Pd_S \rangle$ signatura, $\mathfrak{I} := \langle \mathfrak{A}, \sigma \rangle$ S-interpretación. Definimos recursivamente la *interpretación de los términos* $t \in TERM_S$, $t^{\mathfrak{I}}$, como:

- Caso base:
  - Si $c \in Ct_S$, $c^{\mathfrak{I}} = c^{\mathfrak{A}}$.
  - Si $x \in Var$, $x^{\mathfrak{I}} = \sigma(x)$.
- Caso recursivo:
  - Si $f|_k \in Fn_S$, $t_1, \ldots, t_k \in TERM_S$, $f(t_1, \ldots, t_k)^{\mathfrak{I}} = f^{\mathfrak{A}}(t_1^{\mathfrak{I}}, \ldots, t_k^{\mathfrak{I}})$.

Antes de pasar a la interpretación de fórmulas, conviene establecer la siguiente notación: dada la S-interpretación $\mathfrak{I} := \langle \mathfrak{A}, \sigma \rangle$, por el símbolo $\mathfrak{I}[a/x]$ designamos la S-interpretación determinada por $\langle \mathfrak{A}, \sigma[a/x] \rangle$.[3]

Definición 2.24. Sea $S = \langle Ct_S, Fn_S, Pd_S \rangle$, $\mathfrak{I} := \langle \mathfrak{A}, \sigma \rangle$ S-interpretación, con A conjunto soporte. Definimos recursivamente la *interpretación de las fórmulas* $\varphi \in FORM_S$, $\varphi^{\mathfrak{I}}$, como:

- Caso base:
  - Si $\varphi = \top$, $\varphi^{\mathfrak{I}} = V$.
  - Si $\varphi = \bot$, $\varphi^{\mathfrak{I}} = F$.
  - Si $\varphi = t \doteq s$, con $t, s \in TERM_S$; entonces $\varphi^{\mathfrak{I}} = V$ si $t^{\mathfrak{I}} = s^{\mathfrak{I}}$ y $\varphi^{\mathfrak{I}} = F$ en otro caso.
  - Si $\varphi = p(t_1, \ldots, t_k)$, con $t_1, \ldots, t_k \in TERM_S$, $p|_k \in Pd_S$; entonces $\varphi^{\mathfrak{I}} = V$ si $p^{\mathfrak{A}}(t_1^{\mathfrak{I}}, \ldots, t_k^{\mathfrak{I}}) = V$ y $\varphi^{\mathfrak{I}} = F$ en otro caso.
- Caso recursivo:
  - Si $\varphi = \neg\varphi_1$, $\varphi^{\mathfrak{I}} = v_{\neg}(\varphi_1^{\mathfrak{I}})$.
  - Si $\varphi = \varphi_1 \Box \varphi_2$, $\varphi^{\mathfrak{I}} = v_{\Box}(\varphi_1^{\mathfrak{I}}, \varphi_2^{\mathfrak{I}})$.
  - Si $\varphi = \forall x\, \varphi_1$, con $x \in Var$; entonces $\varphi^{\mathfrak{I}} = V$, si para todo $a \in A$, $\varphi_1^{\mathfrak{I}[a/x]} = V$; y $\varphi^{\mathfrak{I}} = F$ en otro caso.
  - Si $\varphi = \exists x\, \varphi_1$, con $x \in Var$; entonces $\varphi^{\mathfrak{I}} = V$, si existe $a \in A$ tal que $\varphi_1^{\mathfrak{I}[a/x]} = V$; y $\varphi^{\mathfrak{I}} = F$ en otro caso.

Describamos el concepto de *consecuencia lógica* para lógica de primer orden:

Definición 2.25. Sea S signatura. Sean $\Phi \subseteq FORM_S$, $\varphi \in FORM_S$, $\mathfrak{I}$ S-interpretación. Decimos que $\mathfrak{I}$ *satisface* φ, $\mathfrak{I} \vDash \varphi$, si $\varphi^{\mathfrak{I}} = V$. Análogamemente, $\mathcal{A}$ *satisface* Φ, $\mathfrak{I} \vDash \Phi$, si $\psi^{\mathfrak{I}} = V$, para cada $\psi \in \Phi$.

[3] Recordemos que denotamos respectivamente por $f[b/a](x)$ a $f(x)$, si $x \neq a$, y a b, si $x = a$.

Definición 2.26. Sea S signatura. Dados $\Phi \subseteq FORM_S$ y $\varphi \in FORM_S$, decimos que φ es *consecuencia lógica* de Φ, $\Phi \models \varphi$, si toda S-interpretación $\mathfrak{I}$ tal que $\mathfrak{I} \models \Phi$ verifica $\mathfrak{I} \models \varphi$.

Está claro que $\emptyset \models \varphi$ es equivalente a que toda interpretación verifica φ. En estos casos, podemos escribir $\models \varphi$ en vez de $\emptyset \models \varphi$.

Definición 2.27. Sea S signatura. $\Phi \subseteq FORM_S$ se dice *satisfactible* si existe una S-interpretación $\mathfrak{I}$ tal que $\mathfrak{I} \models \Phi$. Φ se dice *insatisfactible* si no existe tal S-interpretación $\mathfrak{I}$.

Fijada una signatura S, podemos clasificar cada fórmula φ como:

- *Satisfactible*, si existe alguna S-interpretación $\mathfrak{I}$ tal que $\mathfrak{I} \models \varphi$.
    - *Tautología*, si $\mathfrak{I} \models \varphi$ para toda S-interpretación $\mathfrak{I}$, es decir, $\models \varphi$
    - *Contingencia*, si es satisfactible pero no tautología.
- *Contradicción*, si no es satisfactible.

Veamos algunos ejemplos:

Ejemplo 2.28. La fórmula $\varphi = \exists x\, x \doteq x$ siempre es una tautología. En esta afirmación juega un papel importante que el conjunto soporte, A, no es vacío. Por tanto dada una signatura S y una S-interpretación $\mathfrak{I}$, existe un elemento $a \in A$. De modo que llamando $\varphi_1 = x \doteq x$, tenemos que $\varphi_1^{\mathfrak{I}[a/x]} = V$, ya que se cumple que $a = a$. Como hay un elemento $a \in A$ tal que $\varphi_1^{\mathfrak{I}[a/x]} = V$, tenemos que $\varphi^{\mathfrak{I}} = V$. De hecho, como esto sucede para todo elemento $a \in A$, también será una tautología la fórmula $\forall x\, x \doteq x$.

Ejemplo 2.29. Consideremos la signatura $S := \langle \emptyset, \emptyset, \{R|_2\} \rangle$ y las fórmulas

$$\varphi := \exists x\, \forall y\, R(x,y)$$

$$\psi := \forall y\, \exists x\, R(x,y)$$

Veamos que $\varphi \models \psi$. Para ello, vamos a ver qué significan φ y ψ en términos del conjunto soporte.

Sea $\mathfrak{I}$ S-interpretación de soporte A tal que $\mathfrak{I} \models \varphi$. Entonces, existe un $a \in A$ tal que $\mathfrak{I}[a/x] \models \forall y\, R(x,y)$. Por tanto, para todo $b \in A$ se cumple que $\mathfrak{I}[a/x][b/y] \models R(x,y)$.

Es decir, $\mathfrak{I} \models \varphi$ significa que existe un elemento $a \in A$ que está relacionado con todos los elementos del conjunto soporte A mediante la relación R.

Ahora, sea $\mathfrak{I}$ S-interpretación de soporte A tal que $\mathfrak{I} \models \psi$. Entonces, para todo $c \in A$, $\mathfrak{I}[c/y] \models \exists x\, R(x,y)$. De la misma forma, existe $d \in A$ tal que $\mathfrak{I}[c/y][d/x] \models R(x,y)$.

Es decir, $\mathfrak{I} \vDash \psi$ significa que para todo elemento $c \in A$ existe algún elemento d del conjunto soporte A tal que d está relacionado con c.

Claramente, si tenemos el primer caso tenemos el segundo, porque para todo elemento $c \in A$, a está relacionado con c, por tanto existe algún elemento relacionado con c. Por tanto, $\varphi \vDash \psi$.

Veamos que la implicación recíproca, $\psi \vDash \varphi$, no es cierta en general.

Para ello damos el siguiente contraejemplo: sea el S-álgebra $\mathfrak{A} := \langle A, \emptyset, \{R^{\mathfrak{A}}\}\rangle$, con $A := \{1, 2, 3\}$ y $R^{\mathfrak{A}} : A^2 \to Bool$ dada por $(1, 2), (2, 3), (3, 1) \mapsto V$, $(x, y) \mapsto F$ en otro caso[4].

Sea $\mathcal{G} = \langle \mathfrak{A}, \sigma\rangle$ una S-interpretación. Evidentemente, para todo $a \in A$ existe $b \in A$ tal que $\mathcal{G}[a/y][b/x] \vDash R(x, y)$: si $a = 1$, $b = 3$; si $a = 2$, $b = 1$; si $a = 3$, $b = 2$. Por otro lado, es imposible que exista $c \in A$ tal que, para todo $d \in A$, $\mathfrak{I}[c/x][d/y] \vDash R(x, y)$, ya que ningún elemento está relacionado con todos. Por tanto $\mathcal{G}$ satisface ψ pero no satisface φ.

Ejemplo 2.30. $\varphi_{=\infty} := (\exists z\, \forall x\, \neg f(x) \doteq z) \wedge (\forall x\, \forall y\, f(x) \doteq f(y) \to x \doteq y)$ es una contingencia. Nótese que nos dice que f es (símbolo de) una función inyectiva y no sobreyectiva. En concreto, esto implica que el cardinal del conjunto soporte de cualquier S-álgebra inducida por una S-interpretación $\mathfrak{I}$ que satisface $\varphi_{=\infty}$ tiene que ser infinito. Esto se debe a que el símbolo de función f tiene asociada una función $f^{\mathfrak{A}} : A \to A$ que es inyectiva y no sobreyectiva, lo cual no puede pasar si A es finito.

Ejemplo 2.31. $\varphi_{=2} := (\exists x\, \exists y\, \neg x \doteq y) \wedge \forall z\, (z \doteq y \vee z \doteq x)$ es una contingencia. Nos dice, en concreto, que el cardinal del conjunto soporte de cualquier S-álgebra inducida por una S-interpretación tiene que ser igual a 2.

2.5.1. Axiomas de Peano

Recordemos la signatura Nat definida en 2.2. Este lenguaje fue introducido por Peano para describir los números naturales usando los símbolos de suma, producto, sucesor y menor o igual. Para describir los naturales podrían considerarse estos axiomas:

1. $\forall x\, \neg s(x) \doteq 0$
2. $\forall x\, \forall y\; s(x) \doteq s(y) \to x \doteq y$
3. $\forall x\; x + 0 \doteq x$
4. $\forall x\, \forall y\; x + s(y) \doteq s(x + y)$

[4]En ocasiones es más cómoda la notación conjuntista, es decir, en vez de dar $R^{\mathfrak{A}}$ como una función $R^{\mathfrak{A}} : A^n \to Bool$, podemos darla como el subconjunto $\{a \in \mathfrak{A}^n \mid R^{\mathfrak{A}}(a) = V\}$. Por ejemplo, en este caso, $R^{\mathfrak{A}}$ sería el conjunto $\{(1, 2), (2, 3), (3, 1)\}$

5. $\forall x\, x * 0 \doteq 0$

6. $\forall x \forall y\, x * s(y) \doteq (x * y) + x$

7. $\forall x \forall y\, (x < y \leftrightarrow \exists z\, x + s(z) \doteq y)$

Llamamos Φ_N al conjunto de las anteriores sentencias. Es fácil comprobar que en la Nat-álgebra $\mathfrak{A}_{Nat}$, introducida en 2.19, cualquier interpretación satisface los axiomas de Peano. Sin embargo, también hay otras estructuras con esta propiedad. De hecho, como veremos en 4.50, cualquier conjunto de sentencias que se cumplan en los naturales también se cumplirán en alguna estructura de conjunto soporte no numerable.

2.6. Sustitución

El ejemplo 2.29 muestra que la determinación de la consecuencia lógica en particular y de la satisfactibilidad en general es una tarea costosa. Motivados por este hecho, investigamos relaciones sintácticas que nos permitan trabajar más cómodamente. Comenzamos por el concepto de *sustitución*, que ya estuvo presente a lo largo del anterior capítulo.

De ahora en adelante, denotamos vectores de elementos de un determinado conjunto con una raya horizontal sobre una letra. Normalmente empleamos la misma letra para referirnos a los elementos de tal vector. Por ejemplo, dada la signatura S y el conjunto soporte de cierta S-álgebra, A, designamos $\bar{a} := (a_1, \dots, a_n)$ con $a_i \in A$, para todo $i = 1, \dots, n$.

Comenzamos definiendo la sustitución para términos:

Definición 2.32. Sea S signatura. Sean $t \in TERM_S$, $\bar{x} := (x_1, \dots, x_n)$ vector de Var y $\bar{s} := (s_1, \dots, s_n)$ vector de $TERM_S$. Definimos la *sustitución de $\bar{x}$ por $\bar{s}$ en t* como:

- Si $t \in Ct_S$, $t[\bar{s}/\bar{x}] = t$.
- Si $t \in Var$, $t[\bar{s}/\bar{x}] = s_i$ si $t = x_i$, para cierto i, y $t[\bar{s}/\bar{x}] = t$ si $t \neq x_i$, para todo i.
- $f(t_1, \dots t_n)[\bar{s}/\bar{x}] = f(t_1[\bar{s}/\bar{x}], \dots, t_n[\bar{s}/\bar{x}])$.

Pasamos a la sustitución para fórmulas:

Definición 2.33. Sea S signatura. Sean $\varphi \in FORM_S$, $\bar{x} := (x_1, \dots, x_n)$ vector de Var y $\bar{s} := (s_1, \dots, s_n)$ vector de $TERM_S$. Definimos la *sustitución de $\bar{x}$ por $\bar{s}$ en φ* como:

- Caso base:
  - $\top[\bar{s}/\bar{x}] = \top$.
  - $\bot[\bar{s}/\bar{x}] = \bot$.

- $(t_1 \doteq t_2)[\bar{s}/\bar{x}] = (t_1[\bar{s}/\bar{x}] \doteq t_2[\bar{s}/\bar{x}])$.
- Si $p|_k \in Pd_S$ y $t_1, \ldots, t_k \in TERM_S$, $p(t_1, \ldots, t_k)[\bar{s}/\bar{x}] = p(t_1[\bar{s}/\bar{x}], \ldots, t_k[\bar{s}/\bar{x}])$.

- Caso recursivo:
  - $(\neg\varphi)[\bar{s}/\bar{x}] = (\neg\varphi[\bar{s}/\bar{x}])$.
  - $(\varphi_1 \Box \varphi_2)[\bar{s}/\bar{x}] = (\varphi_1[\bar{s}/\bar{x}] \Box \varphi_2[\bar{s}/\bar{x}])$.
  - $(Qx\ \varphi)[\bar{s}/\bar{x}]$ se define por casos:
    - $Qx\ \varphi[\bar{s}/\bar{x}]$, si $x \notin \bar{x}$ y $x \notin \bigcup_{i=1}^n var(s_i)$.
    - $Qx\ \varphi[(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)/(x_1, \ldots, x_{i-1}, x_{i+1} \ldots, x_n)]$, si $x = x_i \in \bar{x}$ y $x \notin \bigcup_{i=1}^n var(s_i)$.
    - $Qz\ \varphi[z/x][\bar{s}/\bar{x}]$ si $x \in \bigcup_{i=1}^n var(s_i)$, siendo z una variable nueva, es decir, que no está en $\bar{x}$, $\bigcup_{i=1}^n var(s_i)$ o φ.

Si queremos sustituir un solo elemento, no un vector con varios, podemos omitir la raya horizontal sobre la letra (aunque también puede interpretarse como un vector de un elemento).

En ocasiones haremos uso de la siguiente notación:

$$f(t_1, \ldots t_n)[\bar{s}/\bar{x}] = f(t_1, \ldots t_n)[s_1/x_1, \ldots, s_n/x_n].$$

Ejemplo 2.34. Si llamamos $a := h(x)$ y $b := f(x, y)$, entonces

$$f(x, y)[h(x)/x, f(x, y)/y] = f(h(x), f(x, y)).$$

Nota 2.35. Vemos que el caso recursivo $Qx\varphi$ de la sustitución se divide en tres subcasos. El primer caso se da cuando no hay ningún problema. El segundo caso se introduce para no tener que sustituir variables ligadas.
En el tercer caso, introducimos una variable nueva. Si no lo hiciéramos así, podría darse que al hacer la sustitución, una variable libre pasara a ser una variable ligada, un fenómeno conocido como *colisión*. Por ejemplo, consideremos la fórmula

$$\psi = \exists x\ \ y < x.$$

Si queremos sustituir y por x en esa fórmula, no nos gustaría obtener $\exists x\ \ x < x$. Esta fórmula tiene un significado claramente distinto a la anterior. Por eso en estos casos cambiamos la variable x por una nueva variable z, obteniendo

$$\psi[x/y] = \exists z\ \ (y < x)[z/x][x/y] = \exists z\ \ x < z$$

2.7. Lema de coincidencia

Más adelante nos será útil el estudiar las relaciones entre diferentes interpretaciones y signaturas. Para ello nos será útil el lema de coincidencia, además de para demostrar el lema de sustitución de la próxima sección.

Comencemos con la siguiente

Definición 2.36. Sea $S = \langle Ct_S, Fn_S, Pd_S\rangle$ signatura. Definimos recursivamente el *vocabulario para términos*, *voc*, como:

- Caso base:
  - $voc(c) = \{c\}$, para todo $c \in Ct_S$.
  - $voc(x) = \emptyset$, para todo $x \in Var$.
- Caso recursivo: $voc(f(t_1, \ldots, t_k)) = \{f\} \cup \bigcup_{i=1}^{k} voc(t_i)$, con $f|_k \in Fn_S$, $t_1 \ldots t_k \in TERM_S$.

Definición 2.37. Sea $S = \langle Ct_S, Fn_S, Pd_S\rangle$ signatura. Definimos recursivamente el *vocabulario para fórmulas*, *voc*, como:

- Caso base:
  - $voc(p(t_1, \ldots, t_k)) = \{p\} \cup \bigcup_{i=1}^{k} voc(t_i)$, con $p|_k \in Pd_S$, $t_1 \ldots t_k \in TERM_S$.
  - $voc(t \doteq s) = voc(t) \cup voc(s)$, con $t, s \in TERM_S$.
  - $voc(\top) = voc(\bot) = \emptyset$.
- Caso recursivo:
  - $voc(\neg\varphi) = voc(\varphi)$, con $\varphi \in FORM_S$.
  - $voc(\varphi \Box \psi) = voc(\varphi) \cup voc(\psi)$, con $\varphi, \psi \in FORM_S$.
  - $voc(Qx\, \varphi) = voc(\varphi)$, con $\varphi \in FORM_S$ y $x \in Var$.

Es decir, el vocabulario de una expresión, sea término o fórmula, es el conjunto de elementos de la signatura (símbolos de constante, función y predicado) que aparecen en ella.

Ahora consideramos:

Definición 2.38. Sean $S_1 = \langle Ct_{S_1}, Fn_{S_1}, Pd_{S_1}\rangle$, $S_2 = \langle Ct_{S_2}, Fn_{S_2}, Pd_{S_2}\rangle$ signaturas. Definimos la *signatura unión* y la *signatura intersección* respectivamente como:

$$S_1 \cup S_2 := \langle Ct_{S_1} \cup Ct_{S_2}, Fn_{S_1} \cup Fn_{S_2}, Pd_{S_1} \cup Pd_{S_2}\rangle,$$

$$S_1 \cap S_2 := \langle Ct_{S_1} \cap Ct_{S_2}, Fn_{S_1} \cap Fn_{S_2}, Pd_{S_1} \cap Pd_{S_2}\rangle.$$

Ya que podemos obtener nuevas signaturas a partir de otras, es decir, nuevos conjuntos de símbolos, nos interesa estudiar qué ocurre con las interpretaciones, es decir, con los significados de los símbolos. Esto motiva la noción de *coincidencia*:

Definición 2.39. Sean S_1, S_2 signaturas, $\mathfrak{I}_1 = \langle \mathfrak{A}_1, \sigma_1 \rangle$ una S_1-interpretación y $\mathfrak{I}_2 = \langle \mathfrak{A}_2, \sigma_2 \rangle$ una S_2-interpretación con el mismo conjunto soporte. Consideremos $S := S_1 \cap S_2$, $t \in TERM_S$. Decimos que $\mathfrak{I}_1$ *y* $\mathfrak{I}_2$ *coinciden en* t, $\mathfrak{I}_1 \sim_t \mathfrak{I}_2$, si:

1. $c^{\mathfrak{I}_1} = c^{\mathfrak{I}_2}$, para todo $c \in voc(t)$, $c \in Ct_S$.
2. $x^{\mathfrak{I}_1} = x^{\mathfrak{I}_2}$, para todo $x \in var(t)$.
3. $f^{\mathfrak{A}_1} = f^{\mathfrak{A}_2}$, para todo $f \in voc(t)$, $f \in Fn_S$.

Definición 2.40. Sean S_1, S_2 signaturas, $\mathfrak{I}_1 = \langle \mathfrak{A}_1, \sigma_1 \rangle$ una S_1-interpretación y $\mathfrak{I}_2 = \langle \mathfrak{A}_2, \sigma_2 \rangle$ una S_2-interpretación con el mismo conjunto soporte. Consideremos $S := S_1 \cap S_2$, $\varphi \in FORM_S$. Decimos que $\mathfrak{I}_1$ *y* $\mathfrak{I}_2$ *coinciden en* φ, $\mathfrak{I}_1 \sim_\varphi \mathfrak{I}_2$, si:

1. $c^{\mathfrak{I}_1} = c^{\mathfrak{I}_2}$, para todo $c \in voc(\varphi)$, $c \in Cts_S$.
2. $\sigma_1(x) = \sigma_2(x)$, para todo $x \in lib(\varphi)$.
3. $f^{\mathfrak{A}_1} = f^{\mathfrak{A}_2}$, para todo $f \in voc(\varphi)$, $f \in Fn_S$.
4. $p^{\mathfrak{A}_1} = p^{\mathfrak{A}_2}$, para todo $p \in voc(\varphi)$, $p \in Pd_S$.

Ya estamos en disposición de enunciar el resultado que da nombre a esta sección:

Teorema 2.41. *(Lema de coincidencia) Sean S_1, S_2 signaturas, $\mathfrak{I}_1 = \langle \mathfrak{A}_1, \sigma_1 \rangle$ una S_1-interpretación y $\mathfrak{I}_2 = \langle \mathfrak{A}_2, \sigma_2 \rangle$ una S_2-interpretación con el mismo conjunto soporte. Consideremos $S := S_1 \cap S_2$.*

1. *Si $\mathfrak{I}_1 \sim_t \mathfrak{I}_2$, con $t \in TERM_S$, entonces $t^{\mathfrak{I}_1} = t^{\mathfrak{I}_2}$.*
2. *Si $\mathfrak{I}_1 \sim_\varphi \mathfrak{I}_2$, con $\varphi \in FORM_S$, entonces $\varphi^{\mathfrak{I}_1} = \varphi^{\mathfrak{I}_2}$.*

Es decir, la primera parte nos dice que el significado de un término bajo una interpretación solo depende de los significados de las constantes, variables y funciones que aparecen en ella. La segunda nos dice algo parecido para fórmulas, solo que además nos dice que el significado de una fórmula **no** depende del significado de sus variables ligadas, solo importan las libres.

Demostración.

1. Inducción estructural. Caso base:
   - Si $t = p$ con $p \in Ct_S$, entonces $p \in voc(t)$, por tanto como $\mathfrak{I}_1 \sim_t \mathfrak{I}_2$, tenemos que $p^{\mathfrak{I}_1} = p^{\mathfrak{I}_2}$.
   - Si $t = x$ con $x \in var$, entonces $x \in var(t)$, por tanto como $\mathfrak{I}_1 \sim_t \mathfrak{I}_2$, se cumple $x^{\mathfrak{I}_1} = x^{\mathfrak{I}_2}$.

Caso inductivo: $t = f(t_1, \dots, t_n)$, para $t_1, \dots, t_n \in TERM_S$. Como $voc(t_i) \subseteq voc(t)$ y $var(t_i) \subseteq var(t)$, se cumple que $\mathfrak{I}_1 \sim_{t_i} \mathfrak{I}_2, i = 1, \dots, n$.
Por tanto por inducción tenemos $t_i^{\mathfrak{I}_1} = t_i^{\mathfrak{I}_2}$, de modo que:

$$f(t_1, \dots, t_n)^{\mathfrak{I}_1} = f^{\mathfrak{A}_1}(t_1^{\mathfrak{I}_1}, \dots, t_n^{\mathfrak{I}_1}) = f^{\mathfrak{A}_2}(t_2^{\mathfrak{I}_2}, \dots, t_n^{\mathfrak{I}_2}) = f(t_1, \dots, t_n)^{\mathfrak{I}_2},$$

Donde hemos usado que $f \in voc(t)$, por tanto $f^{\mathfrak{A}_1} = f^{\mathfrak{A}_2}$.

2. De nuevo, inducción estructural. Caso base:

   - φ es $\top$ o $\bot$. Caso obvio.
   - $\varphi = t_1 \doteq t_2$. En este caso, $voc(t_i) \subseteq voc(\varphi)$ y $var(t_i) \subseteq var(\varphi)$. De modo que $\mathfrak{I}_1 \sim_{t_i} \mathfrak{I}_2, i = 1, 2$, y por el apartado anterior se cumple que $t_i^{\mathfrak{I}_1} = t_i^{\mathfrak{I}_2}, i = 1, 2$. Por tanto $\varphi^{\mathfrak{I}_1}$ es V si y solo si $t_1^{\mathfrak{I}_1} = t_2^{\mathfrak{I}_1}$ si y solo si $t_1^{\mathfrak{I}_2} = t_2^{\mathfrak{I}_2}$ si y solo si $\varphi^{\mathfrak{I}_2}$.
   - $\varphi = p(t_1, \dots, t_k)$. En este caso,igual que en el anterior tenemos $t_i^{\mathfrak{I}_1} = t_i^{\mathfrak{I}_2}$. Además, $p \in voc(\varphi)$, por tanto $p^{\mathfrak{A}_1} = p^{\mathfrak{A}_2}$. De modo que $\varphi^{\mathfrak{I}_1}$ es V si y solo si $p^{\mathfrak{A}_1}(t_1^{\mathfrak{I}_1}, \dots, t_n^{\mathfrak{I}_1}) = V$ si y solo si $p^{\mathfrak{A}_2}(t_1^{\mathfrak{I}_2}, \dots, t_n^{\mathfrak{I}_2}) = V$ si y solo si $\varphi^{\mathfrak{I}_2}$ es V.

   Caso inductivo:

   - $\varphi = \neg\varphi_1$. Entonces $voc(\varphi) = voc(\varphi_1)$ y $lib(\varphi) = lib(\varphi_1)$. Por tanto, $\mathfrak{I}_1 \sim_{\varphi_1} \mathfrak{I}_2$, y por hipótesis de inducción, $\varphi_1^{\mathfrak{I}_1} = \varphi_1^{\mathfrak{I}_2}$. De modo que $\varphi^{\mathfrak{I}_1} = v_\neg(\varphi_1^{\mathfrak{I}_1}) = v_\neg(\varphi_1^{\mathfrak{I}_2}) = \varphi^{\mathfrak{I}_2}$.
   - Si $\varphi = \varphi_1 \Box \varphi_2$. Igual que en el caso anterior, tenemos que $\varphi_i^{\mathfrak{I}_1} = \varphi_i^{\mathfrak{I}_2}, i = 1, 2$. De modo que: $\varphi^{\mathfrak{I}_1} = v_\Box(\varphi_1^{\mathfrak{I}_1}, \varphi_2^{\mathfrak{I}_1}) = v_\Box(\varphi_1^{\mathfrak{I}_2}, \varphi_2^{\mathfrak{I}_2}) = \varphi^{\mathfrak{I}_2}$.
   - $\varphi = \exists x\varphi_1$. Como $\mathfrak{I}_1 \sim_\varphi \mathfrak{I}_2$, y tenemos $voc(\varphi) = voc(\varphi_1)$, $lib(\varphi) = lib(\varphi_1) \setminus \{x\}$, sabemos que $\mathfrak{I}_1[a/x] \sim_{\varphi_1} \mathfrak{I}_2[a/x]$, ya que la única diferencia de interpretaciones de $\mathfrak{I}_1$ y $\mathfrak{I}_2$ en φ_1 que no se da en φ es en la variable x, y $\mathfrak{I}_1[a/x]$ y $\mathfrak{I}_2[a/x]$ coinciden en la variable x. Ahora, $\varphi^{\mathfrak{I}_1}$ es V si y solo si existe $a \in A$ tal que $\varphi_1^{\mathfrak{I}_1[a/x]} = V$. Pero esto equivale a que exista $a \in A$ tal que $\varphi_1^{\mathfrak{I}_2[a/x]} = V$, es decir, a que $\varphi^{\mathfrak{I}_2}$ sea V.
   - $\varphi = \forall x\varphi_1$. Igual que en el caso anterior, sabemos que $\mathfrak{I}_1[a/x] \sim_{\varphi_1} \mathfrak{I}_2[a/x]$. Ahora, $\varphi^{\mathfrak{I}_1}$ es V si y solo si para todo $a \in A$ se cumple $\varphi_1^{\mathfrak{I}_1[a/x]} = V$. Pero esto equivale a que para todo $a \in A$ se cumpla $\varphi_1^{\mathfrak{I}_2[a/x]} = V$, lo cual equivale a que $\varphi^{\mathfrak{I}_2}$ sea V.

$\square$

Nota 2.42. Sean $t \in TERM_S, \varphi \in FORM_S$ tales que existen $x_1, \dots, x_n \in Var$ de modo que $var(t) \subseteq \{x_1, \dots, x_n\}$ y $lib(\varphi) \subseteq \{x_1, \dots, x_n\}$. El Lema de Coincidencia nos dice que el significado que tome cada variable que no pertenezca a $\{x_1, \dots, x_n\}$ no afectará a los significados de t y φ.
Es decir, dada una interpretación $\mathfrak{I} = \langle \mathfrak{A}, \sigma \rangle$, $\varphi^{\mathfrak{I}}$ solo depende de $\mathfrak{A}$ y de $\sigma(x_1), \dots, \sigma(x_n)$.

Ejemplo 2.43. En el ejemplo anterior, si φ es una fórmula cerrada, $lib(\varphi) = \emptyset$ y entonces podemos tomar $n = 0$. Es decir, el significado de φ depende solo de el álgebra, no de la interpretación.

2.8. Isomorfía

Seguimos nuestro estudio de las relaciones entre signaturas e interpretaciones. Introducimos a continuación un concepto empleado constantemente en matemáticas. En nuestro caso, captura la idea de una aplicación que 'preserva el significado' entre dos álgebras de la misma signatura:

Definición 2.44. Sea $S = \langle Ct_S, Fn_S, Pd_S \rangle$ signatura. Sean $\mathfrak{A}_1$ y $\mathfrak{A}_2$ S-álgebras de conjuntos soporte respectivos A_1 y A_2. Una aplicación $h : A_1 \to A_2$ es *isomorfismo de S-álgebras* si:

- h es biyectiva.
- $h(c^{\mathfrak{A}_1}) = c^{\mathfrak{A}_2}$, para todo $c \in Ct_S$.
- $h(f^{\mathfrak{A}_1}(a_1, \dots, a_n)) = f^{\mathfrak{A}_2}(h(a_1), \dots, h(a_n))$, con $f|_n \in Fn_S$, $a_1, \dots, a_n \in A_1$.
- $p^{\mathfrak{A}_1}(a_1, \dots, a_n) = p^{\mathfrak{A}_2}(h(a_1), \dots, h(a_n))$, con $p|_n \in Pn_S$, $a_1, \dots, a_n \in A_1$.

Si existe un isomorfismo entre $\mathfrak{A}_1$ y $\mathfrak{A}_2$, decimos que son *isomorfas* y lo denotamos por $\mathfrak{A}_1 \approx \mathfrak{A}_2$.

Definición 2.45. Sea $S = \langle Ct_S, Fn_S, Pd_S \rangle$ signatura. Sean $\mathfrak{I}_1 = \langle \mathfrak{A}_1, \sigma_1 \rangle$ y $\mathfrak{I}_2 = \langle \mathfrak{A}_2, \sigma_2 \rangle$ S-intepretaciones. Una aplicación h es *isomorfismo de S-interpretaciones* si:

- h es isomorfismo entre las S-álgebras $\mathfrak{A}_1$ y $\mathfrak{A}_2$.
- $h(\sigma_1(x)) = \sigma_2(x)$, para todo $x \in Var$.

Cuando existe un isomorfismo entre $\mathfrak{I}_1$ y $\mathfrak{I}_2$, decimos que son *isomorfas* y lo simbolizamos por $\mathfrak{I}_1 \approx \mathfrak{I}_2$.

Ejemplo 2.46. Consideremos la signatura característica de los grupos, $S := \langle \{e\}, \{*|_2\}, \emptyset \rangle$. Sean los grupos $(\mathbb{Z}_2, +)$ y $(A, \cdot)$ dado por $A = \{a, b\}$ con:

$\cdot$	a	b
a	a	b
b	b	a

Entonces tenemos dos S-álgebras:

$$\mathfrak{A}_1 = \langle \mathbb{Z}_2, \{0, 1\}, \{+\}, \emptyset \rangle$$

$$\mathfrak{A}_2 = \langle A, \{a, b\}, \{\cdot\}, \emptyset\rangle$$

Entonces, la aplicación $h : \mathbb{Z}_2 \to A$ dada por $h(0) = a$ y $h(1) = b$ es un isomorfismo, es decir, $h : \mathfrak{A}_1 \approx \mathfrak{A}_2$.

Este lema técnico nos permitirá probar el siguiente teorema:

Lema 2.47. *Sea $S = \langle Ct_S, Fn_S, Pd_S\rangle$ signatura. Sean $\mathfrak{I}_1 = \langle \mathfrak{A}_1, \sigma_1\rangle$ y $\mathfrak{I}_2 = \langle \mathfrak{A}_2, \sigma_2\rangle$ S-interpretaciones de conjuntos soporte A_1, A_2 tales que existe $h : \mathfrak{I}_1 \approx \mathfrak{I}_2$. Sea $a \in A_1$.*
Entonces, $\mathfrak{I}_1[a/x] \approx \mathfrak{I}_2[h(a)/x]$. De hecho, la misma función $h : A_1 \to A_2$ es isomorfismo entre $\mathfrak{I}_1[a/x]$ y $\mathfrak{I}_2[h(a)/x]$.

Demostración. Está claro que se cumple $h(\sigma_1(y)) = \sigma_2(y)$ para todas las variables excepto x, ya que si $y \in Var \setminus \{x\}$, $h(\sigma_1[a/x](y)) = h(\sigma_1(y)) = \sigma_2(y) = \sigma_2[h(a)/x](y)$.
Respecto a la variable x, $h(\sigma_1[a/x](x)) = h(a) = \sigma_2[h(a)/x](x)$.
Queda entonces comprobar que h es isomorfismo entre las álgebras de $\mathfrak{I}_1$ y $\mathfrak{I}_2$. Pero esto es directo, ya que estas álgebras son $\mathfrak{A}_1$ y $\mathfrak{A}_2$. □

El siguiente resultado muestra que los isomorfismos entre S-interpretaciones se extienden a las fórmulas y los términos de tal signatura S.

Teorema 2.48. *(Lema de Isomorfía) Sea $S = \langle Ct_S, Fn_S, Pd_S\rangle$ signatura. Sea h isomorfismo entre las S-interpretaciones $\mathfrak{I}_1$ y $\mathfrak{I}_2$. Entonces:*

1. *Para todo $t \in TERM_S$, $h(t^{\mathfrak{I}_1}) = t^{\mathfrak{I}_2}$.*
2. *Para toda $\varphi \in FORM_S$, $\varphi^{\mathfrak{I}_1} = \varphi^{\mathfrak{I}_2}$, y por tanto $\mathfrak{I}_1 \vDash \varphi$ si y solo si $\mathfrak{I}_2 \vDash \varphi$.*

Demostración.

1. Por inducción estructural sobre $t \in TERM_S$:
   - Caso base:
     - Si $t = c$, con $c \in Ct_S$, entonces $h(t^{\mathfrak{I}_1}) = h(c^{\mathfrak{I}_1}) = c^{\mathfrak{I}_2} = t^{\mathfrak{I}_2}$.
     - Si $t = x$, con $x \in Var$, entonces $h(t^{\mathfrak{I}_1}) = h(x^{\mathfrak{I}_1}) = h(\sigma_1(x)) = \sigma_2(x) = x^{\mathfrak{I}_2} = t^{\mathfrak{I}_2}$.
   - Caso recursivo: Si $t = f(t_1, \ldots, t_n)$, con $f|_n \in Fn_S$ y $t_1, \ldots, t_n \in TERM_S$ cumplen el enunciado, entonces:
     $h(f(t_1, \ldots, t_n)^{\mathfrak{I}_1}) = h(f^{\mathfrak{A}_1}(t_1^{\mathfrak{I}_1}, \ldots, t_n^{\mathfrak{I}_1})) = f^{\mathfrak{A}_2}(h(t_1^{\mathfrak{I}_1}), \ldots, h(t_n^{\mathfrak{I}_1})) = f^{\mathfrak{A}_2}(t_1^{\mathfrak{I}_2}, \ldots, t_n^{\mathfrak{I}_2}) = f(t_1, \ldots, t_n)^{\mathfrak{I}_2}$
2. Por inducción estructural sobre $\varphi \in FORM_S$:
   - Caso base:
     - Si $\varphi = \top$, entonces $\varphi^{\mathfrak{I}_1} = \top^{\mathfrak{I}_1} = V = \top^{\mathfrak{I}_2} = \varphi^{\mathfrak{I}_2}$ y análogo con $\bot$.
     - Si $\varphi = t \doteq s$ para $t, s \in TERM_S$, entonces $\mathfrak{I}_1 \vDash \varphi$ si y solo si $t^{\mathfrak{I}_1} = s^{\mathfrak{I}_1}$ si y solo si (por ser h inyectiva) $h(t^{\mathfrak{I}_1}) = h(s^{\mathfrak{I}_1})$ si y solo si $t^{\mathfrak{I}_2} = s^{\mathfrak{I}_2}$ si y solo si $\mathfrak{I}_1 \vDash \varphi$.

- Si $\varphi = p(t_1, \dots, t_n)$, con $p|_n \in Pd_S$ y $t_1, \dots, t_n \in TERM_S$ cumplen el enunciado, entonces $p(t_1, \dots, t_n)^{\mathfrak{I}_1} = V$, si y solo si $p^{\mathfrak{A}_1}(t_1^{\mathfrak{I}_1}, \dots, t_n^{\mathfrak{I}_1}) = V$ si y solo si $p^{\mathfrak{A}_2}(h(t_1^{\mathfrak{I}_1}), \dots, h(t_n^{\mathfrak{I}_1})) = V$ si y solo si $p^{\mathfrak{A}_2}(t_1^{\mathfrak{I}_2}, \dots, t_n^{\mathfrak{I}_2}) = V$ si y solo si $p(t_1, \dots, t_n)^{\mathfrak{I}_2} = V$.

- Caso recursivo:
  - $\varphi = \neg\varphi_1$. Entonces, $\varphi^{\mathfrak{I}_1}$ es V si y solo si $v_\neg(\varphi_1^{\mathfrak{I}_1})$ es V si y solo si $v_\neg(\varphi_1^{\mathfrak{I}_2})$ es V si y solo si $\varphi^{\mathfrak{I}_2}$ es V.
  - $\varphi = \varphi_1 \Box \varphi_2$. Entonces, $\varphi^{\mathfrak{I}_1}$ es V si y solo si $v_\Box(\varphi_1^{\mathfrak{I}_1}, \varphi_2^{\mathfrak{I}_1})$ es V si y solo si $v_\Box(\varphi_1^{\mathfrak{I}_2}, \varphi_1^{\mathfrak{I}_2})$ es V si y solo si $\varphi^{\mathfrak{I}_2}$ es V.
  - $\varphi = \exists x\, \varphi_1$. Entonces, $\varphi^{\mathfrak{I}_1}$ es V si y solo si existe a que cumple $\varphi_1^{\mathfrak{I}_1[a/x]} = V$. Como por hipótesis inductiva el enunciado se cumple para φ_1, usando el lema 2.47 tenemos que $\varphi_1^{\mathfrak{I}_1[a/x]} = V$ equivale a $\varphi_1^{\mathfrak{I}_2[a/x]} = V$. Y que exista a que cumpla esto es equivalente a que $\varphi^{\mathfrak{I}_2}$ sea V, como queríamos.
  - $\varphi = \forall x\, \varphi_1$. Entonces, $\varphi^{\mathfrak{I}_1}$ es V si y solo si para todo a se cumple $\varphi_1^{\mathfrak{I}_1[a/x]} = V$. Como por hipótesis inductiva el enunciado se cumple para φ_1, usando el lema 2.47 tenemos que $\varphi_1^{\mathfrak{I}_1[a/x]} = V$ equivale a $\varphi_1^{\mathfrak{I}_2[a/x]} = V$. Y que esto se cumpla para todo a es equivalente a que $\varphi^{\mathfrak{I}_2}$ sea V, como queríamos.

$\square$

2.9. Lema de sustitución

Proseguimos nuestro estudio de la sustitución. A continuación obtenemos el resultado análogo al que ya demostramos en lógica proposicional.

Teorema 2.49. *(Lema de sustitución) Sean $S = \langle Ct_S, Fn_S, Pd_S \rangle$ signatura, $\mathfrak{I}$ S-interpretación de conjunto soporte A. Sean $\bar{x} := (x_1, \dots, x_n)$ y $\bar{t} := (t_1, \dots, t_n)$ vectores de Var y $TERM_S$, respectivamente. Denotemos $\bar{t}^{\mathfrak{I}} := (t_1^{\mathfrak{I}}, \dots, t_n^{\mathfrak{I}})$ y $\mathcal{J} := \mathfrak{I}[\bar{t}^{\mathfrak{I}}/\bar{x}]$. Entonces:*

1. *Si $s \in TERM_S$, $(s[\bar{t}/\bar{x}])^{\mathfrak{I}} = s^{\mathcal{J}}$.*
2. *Si $\varphi \in FORM_S$, $(\varphi[\bar{t}/\bar{x}])^{\mathfrak{I}} = \varphi^{\mathcal{J}}$, es decir, $\mathfrak{I} \vDash \varphi[\bar{t}/\bar{x}]$ si y solo si $\mathcal{J} \vDash \varphi$.*

Demostración.

1. Por inducción estructural.
   - Caso base:
     - Si $s = c$, con $c \in Ct_S$, entonces $s = s[\bar{t}/\bar{x}] = c$, y se cumple $c^{\mathfrak{I}} = c^{\mathcal{J}}$, ya que $\mathfrak{I}$ y $\mathcal{J}$ tienen la misma álgebra.
     - Sea $s = x$, con $x \in Var$. Si x no está en $\bar{x}$, entonces $x[\bar{t}/\bar{x}] = x$, y como $x^{\mathfrak{I}} = x^{\mathcal{J}}$, tenemos el resultado. Si $x = x_i$, con x_i el elemento i-ésimo de $\bar{x}$, $(x[\bar{t}/\bar{x}])^{\mathfrak{I}} = t_i^{\mathfrak{I}} = x^{\mathfrak{I}[\bar{t}/\bar{x}]} = x^{\mathcal{J}}$.

- Caso recursivo: Sea $s = f(r_1, \ldots, r_k)$, con $f|_k \in Fn_S$ y $r_1, \ldots, r_k \in TERM_S$. Entonces, $(f(r_1, \ldots, r_k)[\bar{t}/\bar{x}])^{\mathfrak{I}} = (f(r_1[\bar{t}/\bar{x}], \ldots, r_k[\bar{t}/\bar{x}]))^{\mathfrak{I}} = f^{\mathfrak{A}}((r_1[\bar{t}/\bar{x}])^{\mathfrak{I}}, \ldots, (r_k[\bar{t}/\bar{x}])^{\mathfrak{I}})$ y aplicando la hipótesis de inducción, lo anterior es igual $f^{\mathfrak{A}}(r_1^{\mathcal{J}}, \ldots, r_k^{\mathcal{J}}) = f(r_1, \ldots, r_k)^{\mathcal{J}}$.

2. Por inducción estructural.

   - Caso base:

     - Si $\varphi = \top$, $(\varphi[\bar{t}/\bar{x}])^{\mathfrak{I}} = V = \varphi^{\mathcal{J}}$. Análogo con $\bot$.
     - Si $\varphi = r \doteq s$, con $r, s \in TERM_S$. Entonces $((r \doteq s)[\bar{t}/\bar{x}])^{\mathfrak{I}} = (r[\bar{t}/\bar{x}] \doteq s[\bar{t}/\bar{x}])^{\mathfrak{I}}$, que es cierto si y solo si $r[\bar{t}/\bar{x}])^{\mathfrak{I}} = s[\bar{t}/\bar{x}]^{\mathfrak{I}}$ si y solo si, por hipótesis de inducción, $r^{\mathcal{J}} = s^{\mathcal{J}}$, es decir, si y solo si $\mathcal{J} \vDash r \doteq s$.
     - Si $\varphi = p(r_1, \ldots, r_k)$, con $p|_k \in Pd_S$, $r_1, \ldots r_k \in TERM_S$, entonces $(p(r_1, \ldots, r_k)[\bar{t}/\bar{x}])^{\mathfrak{I}} = (p(r_1[\bar{t}/\bar{x}], \ldots, r_k[\bar{t}/\bar{x}]))^{\mathfrak{I}}$ es igual a $p^{\mathfrak{A}}((r_1[\bar{t}/\bar{x}])^{\mathfrak{I}}, \ldots, (r_k[\bar{t}/\bar{x}])^{\mathfrak{I}})$, que por hipótesis de inducción es igual a $p^{\mathcal{A}}(r_1^{\mathcal{J}}, \ldots, r_k^{\mathcal{J}}) = p(r_1, \ldots, r_k)^{\mathcal{J}}$.

   - Caso inductivo:

     - Si $\varphi = \neg\psi$, entonces $((\neg\psi)[\bar{t}/\bar{x}])^{\mathfrak{I}} = (\neg\psi[\bar{t}/\bar{x}])^{\mathfrak{I}} = v_{\neg}((\psi[\bar{t}/\bar{x}])^{\mathfrak{I}})$, que por hipótesis de inducción es $v_{\neg}(\psi^{\mathcal{J}}) = (\neg\psi)^{\mathcal{J}}$.
     - Si $\varphi = \psi\Box\chi$, entonces $((\psi\Box\chi)[\bar{t}/\bar{x}])^{\mathfrak{I}} = ((\psi[\bar{t}/\bar{x}])\Box(\chi[\bar{t}/\bar{x}]))^{\mathfrak{I}} = v_{\Box}((\psi[\bar{t}/\bar{x}])^{\mathfrak{I}}, (\chi[\bar{t}/\bar{x}])^{\mathfrak{I}})$, que por hipótesis de inducción es $v_{\Box}(\psi^{\mathcal{J}}, \chi^{\mathcal{J}}) = (\psi\Box\chi)^{\mathcal{J}}$.
     - Veamos el caso $\forall x\, \varphi$. Supongamos que x no es ninguno de los elementos del vector $\bar{x}$, ya que si esto fuera cierto y x correspondiera a la variable i-ésima de $\bar{x}$, x_i, por cómo hemos definido sustitución tendríamos que

       $$(\forall x\ \varphi)[\bar{t}/\bar{x}] = (\forall x\ \varphi)[(t_1, \ldots t_{i-1}, t_{i+1}, \ldots, t_n)/(x_1, \ldots x_{i-1}, x_{i+1}, \ldots, x_n)],$$

       es decir, nuestra sustitución es igual a otra sustitución en que $x \notin \bar{x}$.
       Tenemos entonces dos casos, $x \in var(t_i)$ para algún $i = 1, \ldots, n$ o no. El primero resulta ser más complicado y nos limitamos a hacer ese:
       Supongamos $x \in var(t_i)$ para cierto $1 \leq i \leq n$, con t_i el i-ésimo elemento de $\bar{t}$. Entonces, introduciendo la variable nueva z,

       $$(\forall x\, \varphi)[\bar{t}/\bar{x}] = \forall z\, \varphi[z/x][\bar{t}/\bar{x}]$$

       Entonces, $\mathfrak{I} \vDash \forall z\, \varphi[z/x][\bar{t}/\bar{x}]$ es equivalente a que para todo $a \in A$, $\mathfrak{I}[a/z] \vDash \varphi[z/x][\bar{t}/\bar{x}]$. Aplicando la hipótesis de inducción a $\mathfrak{I}[a/z]$ y a φ, lo anterior equivale a que para todo $a \in A$,

       $$(\mathfrak{I}[a/z])[\bar{t}^{\mathfrak{I}[a/z]}/\bar{x}] \vDash \varphi[z/x]$$

Nótese que, al ser z variable nueva, $z \notin var(t_i)$, para todo $1 \leq i \leq n$ y por 2.41, $t_i^{\mathfrak{I}[a/z]} = t_i^{\mathfrak{I}}$. Lo anterior es entonces equivalente a que para todo $a \in A$:

$$(\mathfrak{I}[a/z])[\overline{t}^{\mathfrak{I}}/\bar{x}] \vDash \varphi[z/x]$$

Por ser z nueva, se sigue también que es distinta de todo elemento de $\bar{x}$. De modo que $(\mathfrak{I}[a/z])[\overline{t}^{\mathfrak{I}}/\bar{x}] = \mathfrak{I}[\overline{t}^{\mathfrak{I}}/\bar{x}][a/z] = \mathcal{J}[a/z]$, y entonces lo anterior es cierto si y solo si para todo $a \in A$,

$$\mathcal{J}[a/z] \vDash \varphi[z/x]$$

De nuevo, por hipótesis de inducción sobre $\mathcal{J}[a/z]$ y φ, lo anterior se da si y solo si, para todo $a \in A$,

$$\mathcal{J}[a/z][z^{\mathcal{J}[a/z]}/x] \vDash \varphi$$

esto es, usando que $z^{\mathcal{J}[a/z]} = a$,

$$(\mathcal{J}[a/z])[a/x] \vDash \varphi$$

Por ser z variable nueva, $\mathcal{J}[a/z] \sim_{\varphi} \mathcal{J}$ y empleando 2.41, $\varphi^{\mathcal{J}[a/z]} = \varphi^{\mathcal{J}}$. Finalmente, lo anterior es equivalente a que para todo $a \in A$,

$$\mathcal{J}[a/x] \vDash \varphi$$

es decir, $\mathcal{J} \vDash \forall x \, \varphi$. El caso $\exists x \, \varphi$ es análogo.

□

2.10. Equivalencia lógica. Forma prenexa

Algunos de los ejemplos precedentes muestran que operar con interpretaciones no es tarea cómoda. Por ello, debemos investigar relaciones entre los signos, que nos faciliten un futuro estudio de su semántica. De nuevo, la noción principal para este propósito es la de *equivalencia lógica*:

Definición 2.50. Sea S signatura, $\varphi, \psi \in FORM_S$. φ y ψ son *lógicamente equivalentes*, $\varphi \sim \psi$, si para toda S-intepretación $\mathfrak{I}$, $\varphi^{\mathfrak{I}} = \psi^{\mathfrak{I}}$, es decir, $\mathfrak{I} \vDash \varphi$ si y solo si $\mathfrak{I} \vDash \psi$[5].

Proposición 2.51. *Sea S signatura, $\varphi, \psi \in FORM_S$. Entonces son equivalentes:*

1. $\varphi \sim \psi$.
2. $\vDash \varphi \leftrightarrow \psi$.

[5]No es difícil demostrar que $\sim$ es reflexiva, simétrica y transitiva, es decir, es relación de equivalencia sobre $FORM_S$.

3. $\varphi \vDash \psi$ *y* $\psi \vDash \varphi$.

Demostración.

(1) $\Longrightarrow$ (2): Si $\varphi \sim \psi$, $\varphi^{\mathfrak{I}} = \psi^{\mathfrak{I}}$ para toda S-interpretación $\mathfrak{I}$, por tanto toda interpretación $\mathfrak{I}$ cumple $\mathfrak{I} \vDash \varphi \leftrightarrow \psi$, como queríamos.

(2) $\Longrightarrow$ (3): Si $\vDash \varphi \leftrightarrow \psi$, toda interpretación $\mathfrak{I}$ solo puede satisfacer φ y ψ o no satisfacer φ ni ψ. Por tanto, si una interpretación satisface φ, satisface ψ y viceversa.

(3) $\Longrightarrow$ (1): Si $\varphi \vDash \psi$ y $\psi \vDash \varphi$, entonces una interpretación satisface ψ si satisface φ y viceversa, por tanto satisface ψ si y solo si satisface φ. □

Evidentemente, las leyes de equivalencia del cálculo proposicional siguen siendo válidas en lógica de primer orden. Además, introducimos nuevas leyes en las que aparezcen cuantificadores:

Teorema 2.52. *(Leyes de equivalencia lógica con cuantificadores)*

1. $Qx\,\varphi \sim Qu\,\varphi[u/x]$, *siendo* u *variable nueva.*
2. *Si* $\varphi \sim \psi$, *entonces* $Qx\,\varphi \sim Qx\,\psi$.
3. $\neg\forall x\,\varphi \sim \exists x\,\neg\varphi, \quad \neg\exists x\,\varphi \sim \forall x\,\neg\varphi$.
4. $\forall x\,\varphi \sim \neg\exists x\,\neg\varphi, \quad \exists x\,\varphi \sim \neg\forall x\neg\varphi$.
5. *Si* $x \neq y$, *entonces* $Qx\,Qy\,\varphi \sim Qy\,Qx\,\varphi$.
6. $\forall x\,(\varphi \wedge \psi) \sim (\forall x\,\varphi) \wedge (\forall x\,\psi), \quad \exists x\,(\varphi \vee \psi) \sim (\exists x\,\varphi) \vee (\exists x\,\psi)$.
7. *Si* $x \notin lib(\psi)$, $Qx\,\psi \sim \psi$.
8. *Si* $x \notin lib(\psi)$, $(\forall x\,\varphi) \wedge \psi \sim \forall x\,(\varphi \wedge \psi), \quad (\exists x\,\varphi) \vee \psi \sim \exists x\,(\varphi \vee \psi)$.
9. *Si* $x \notin lib(\psi)$, $\forall x\,(\varphi \vee \psi) \sim (\forall x\,\varphi) \vee (\forall x\,\psi), \quad \exists x\,(\varphi \wedge \psi) \sim (\exists x\,\varphi) \wedge (\exists x\,\psi)$.
10. *Si* $x \notin lib(\psi)$, $(\forall x\,\varphi) \rightarrow \psi \sim \exists x\,(\varphi \rightarrow \psi), \quad (\exists x\,\varphi) \rightarrow \psi \sim \forall x(\varphi \rightarrow \psi)$.

Demostración.

1. Demostremos el caso $\forall$, el $\exists$ se hace de forma análoga. Sea $\mathfrak{I}$ S-interpretación de conjunto soporte A. $\mathfrak{I} \vDash \forall u\ \varphi[u/x]$ es por definición equivalente a que, para todo $a \in A$,
$$\mathfrak{I}[a/u] \vDash \varphi[u/x].$$
Como $u^{\mathfrak{I}[a/u]} = a$, aplicando el lema de sustitución 2.49 lo anterior es cierto si y solo si, para todo $a \in A$,
$$(\mathfrak{I}[a/u])[a/x] \vDash \varphi$$

Al ser u variable nueva, $(\mathfrak{I}[a/u])[a/x] \sim_\varphi \mathfrak{I}[a/x]$ y 2.41 nos dice que lo anterior equivale a que, para todo $a \in A$,

$$\mathfrak{I}[a/x] \vDash \varphi,$$

que es la definición de $\mathfrak{I} \vDash \forall x\, \varphi$.

2. Sea $\mathfrak{I}$ S-interpretación tal que $\mathfrak{I} \vDash \exists x\, \varphi$, es decir, tal que existe $a \in A$ con $\mathfrak{I}[a/x] \vDash \varphi$, y al ser $\varphi \sim \psi$, lo anterior se da si y solo si existe $a \in A$ con $\mathfrak{I}[a/x] \vDash \psi$, es decir, la definición de $\mathfrak{I} \vDash \exists x\, \psi$. El caso $\forall$ es análogo.

3. Sea $\mathfrak{I}$ S-interpretación tal que $\mathfrak{I} \vDash \neg\forall x\, \varphi$, es decir, $\mathfrak{I} \nvDash \forall x\, \varphi$. Equivalentemente, existe $a \in A$ tal que $\mathfrak{I}[a/x] \vDash \neg\varphi$, que es la definición de $\mathfrak{I} \vDash \exists x\, \neg\varphi$. El caso $\exists$ es análogo.

4. $\varphi \sim \neg\neg\varphi$ luego, aplicando (2), $\forall x\, \varphi \sim \forall x\, \neg\neg\varphi$ y, por (3), $\forall x\, \neg\neg\varphi \sim \neg\exists x\, \neg\varphi$. El caso $\exists$ es análogo.

5. $\mathfrak{I} \vDash \exists x\ \exists y\ \varphi$ si y solo si existe $a \in A$ tal que $\mathfrak{I}[a/x] \vDash \exists y\ \varphi$ si y solo si existen $a, b \in A$ tal que $\mathfrak{I}[a/x][b/y] \vDash \varphi$. Como $x \neq y$, $\mathfrak{I}[a/x][b/y] = \mathfrak{I}[b/y][a/x]$, por tanto lo anterior equivale a que existan $a, b \in A$ tales que $\mathfrak{I}[b/y][a/x] \vDash \varphi$. Igual que antes, vemos que esto es equivalente a que $\mathfrak{I} \vDash \exists y\ \exists x\ \varphi$. El caso $\forall$ es análogo.

6. Sea $\mathfrak{I}$ S-interpretación tal que $\mathfrak{I} \vDash (\forall x\, \varphi) \wedge (\forall x\, \psi)$. Esto, por 2.51, es equivalente a que $\mathfrak{I} \vDash \forall x\, \varphi$ y $\mathfrak{I} \vDash \forall x\, \psi$. Entonces, en el primer caso tenemos que, para todo $a \in A$, $\mathfrak{I}[a/x] \vDash \varphi$ y en el segundo que, para todo $b \in A$, $\mathfrak{I}[b/x] \vDash \psi$. De esta forma, lo anterior es equivalente a que, para todo $c \in A$, $\mathfrak{I}[c/x] \vDash \varphi$ y $\mathfrak{I}[c/x] \vDash \psi$, es decir, $\mathfrak{I}[c/x] \vDash \varphi \wedge \psi$, es decir, $\mathfrak{I} \vDash \varphi \wedge \psi$. El otro caso se puede probar a partir del primero, ya que:

$$\exists x\, (\varphi \vee \psi) \sim \neg\neg(\exists x\, (\varphi \vee \psi)) \sim \neg(\forall x\, \neg(\varphi \vee \psi)) \sim \neg(\forall x\, (\neg\varphi \wedge \neg\psi)) \sim$$
$$\sim \neg((\forall x\, \neg\varphi) \wedge (\forall x \neg\psi)) \sim \neg(\forall x\, \neg\varphi) \vee \neg(\forall x \neg\psi) \sim (\exists x\, \neg\neg\varphi) \vee (\exists x\, \neg\neg\psi) \sim \exists x \varphi \vee \exists x \psi,$$

donde hemos usado las leyes de equivalencia de lógica proposicional, el primer apartado de este ejercicio y la regla (4) para negar cuantificadores.

7. En este caso, solo es necesario notar que $\mathfrak{I}[a/x] \sim_\psi \mathfrak{I}$ y aplicar el Lema de Coincidencia, 2.41.

8. Es una consecuencia directa de los apartados 6 y 7.

9. Comencemos con el caso $\forall$.
Queremos demostrar que $\mathfrak{I} \vDash \forall x\, \varphi \vee \psi$ si y solo si $\mathfrak{I} \vDash \forall x\, \varphi \vee \forall x \psi$
$\Longrightarrow$) Si $\mathfrak{I} \vDash \forall x\, \varphi \vee \psi$, para todo $a \in A$ se cumple $\mathfrak{I}[a/x] \vDash \varphi \vee \psi$. Dividimos en casos:

    - $\mathfrak{I} \vDash \forall x\, \varphi$. En este caso, es obvio que se cumple $\mathfrak{I} \vDash \forall x\, \varphi \vee \forall x \psi$.

- En caso contrario, existe $a \in A$ tal que $\mathfrak{I}[a/x] \nvDash \varphi$. Entonces para ese a tiene que cumplirse $\mathfrak{I}[a/x] \vDash \psi$. Pero entonces por la regla (7), $\mathfrak{I}[a/x] \vDash \forall x \psi$, por tanto $\mathfrak{I} \vDash \forall x\, \varphi \vee \forall x \psi$.

$\Longleftarrow$) $\mathfrak{I} \vDash \forall x \psi \vee \forall x \psi$. Tenemos dos casos:

- $\mathfrak{I} \vDash \forall x \psi$, es decir, para todo $a \in A$, $\mathfrak{I}[a/x] \vDash \psi$. Esto implica que para todo $a \in A$, $\mathfrak{I}[a/x] \vDash \varphi \vee \psi$, es decir, $\mathfrak{I} \vDash \forall x(\varphi \vee \psi)$.
- $\mathfrak{I} \vDash \forall x \varphi$, es decir, para todo $a \in A$, $\mathfrak{I}[a/x] \vDash \varphi$. Esto implica que para todo $a \in A$, $\mathfrak{I}[a/x] \vDash \varphi \vee \psi$, es decir, $\mathfrak{I} \vDash \forall x(\varphi \vee \psi)$.

Pasamos a la segunda parte. Esta se puede hacer de forma análoga a la anterior, aunque es más rápido deducirla de la anterior:

$$\begin{aligned}&\exists x\,(\varphi \wedge \psi) \sim \neg\neg\exists x(\varphi \wedge \psi) \sim \neg\forall x(\neg\varphi \vee \neg\psi) \sim \\ &\quad \sim \neg((\forall x \neg\varphi) \vee (\forall x \neg\psi)) \sim (\neg\forall x \neg\varphi) \wedge (\neg\forall x \neg\psi) \sim (\exists x \varphi) \wedge (\exists x \psi).\end{aligned}$$

Hemos podido usar la parte anterior de la regla ya que $x \in lib(\psi)$ si y solo si $x \in lib(\neg\psi)$.

10. Estas últimas reglas se pueden deducir de las anteriores:

$$\begin{aligned}&(\forall x\, \varphi) \to \psi \sim \neg(\forall x\, \varphi) \vee \psi \sim \\ &\qquad\qquad \sim (\exists x\, \neg\varphi) \vee \psi \sim \exists x\,(\neg\varphi \vee \psi) \sim \exists x\, \varphi \to \psi.\end{aligned}$$

$$\begin{aligned}&(\exists x\, \varphi) \to \psi \sim \neg(\exists x\, \varphi) \vee \psi \sim (\forall x\, \neg\varphi) \vee \psi \sim \\ &\qquad\qquad \sim (\forall x\, \neg\varphi) \vee (\forall x\, \psi) \sim \forall x(\neg\varphi \vee \psi) \sim \forall x(\varphi \to \psi).\end{aligned}$$

□

Las leyes de equivalencia como 6, 8, 9 o 10 permiten escribir los cuantificadores *al principio* de las fórmulas. Esto nos lleva a la siguiente

Definición 2.53. Sea S signatura. Se dice que $\varphi \in FORM_S$ *está en forma prenexa* si se puede escribir como: $Q_1 x_1, \ldots, Q_n x_n\, \psi$, con x_i variable y Q_i cuantificador, para cada $1 \leq i \leq n$, y ψ libre de cuantificadores. ψ recibe el nombre de *núcleo.*

Ahora bien, ¿toda fórmula tiene una fórmula equivalente en forma prenexa? Antes de dar una respuesta afirmativa a esta pregunta necesitamos el siguiente lema:

Lema 2.54. *Sea S signatura. Sean $\varphi, \psi \in FORM_S$, $x \in lib(\psi)$, $u \in Var$ tal que u una variable nueva. Entonces*

$$(Qx\, \varphi) \vee \psi \sim Qu\,(\varphi[u/x] \vee \psi).$$

Demostración. Es consecuencia de las reglas de equivalencia:

$$(\forall x\, \varphi) \lor \psi \sim (\forall u\, \varphi[u/x]) \lor \psi \sim (\forall u\, \varphi[u/x]) \lor (\forall u\, \psi) \sim \forall u\, \varphi[u/x] \lor \psi$$

$$(\exists x\, \varphi) \lor \psi \sim (\exists u\, \varphi[u/x]) \lor \psi \sim (\exists u\, \varphi[u/x]) \lor (\exists u\, \psi) \sim \exists u\, \varphi[u/x] \lor \psi,$$

donde en el caso $\forall$ hemos usado que u no aparece libre en ψ. □

Proposición 2.55. *Sea S signatura. Entonces, para cada $\varphi \in FORM_S$ existe $\psi \in FORM_S$ en forma prenexa tal que $\varphi \sim \psi$.*

Demostración. La demostración se lleva a cabo por inducción estructural. Es rutinaria salvo por el paso inductivo en fórmulas, que demostramos para $\neg$ y $\lor$ [6].

($\neg$) Sea $\varphi = \neg\varphi_1$. Por hipótesis de inducción, existe $\psi \in FORM_S$ en forma prenexa tal que $\varphi_1 \sim \psi$, con $\psi = Q_1 x_1 \ldots Q_n x_n\, \chi$. Entonces $\varphi \sim \neg\varphi_1 \sim \neg\psi \sim Q'_1 x_1 \ldots Q'_n x_n\, \neg\chi$, donde $Q'_i = \forall$ si $Q_i = \exists$ y $Q'_i = \exists$ si $Q'_i = \forall$, para cada $1 \le i \le n$.

($\lor$) Sea $\varphi = \varphi_1 \lor \varphi_2$. Por hipótesis de inducción, existen los respectivos núcleos χ_1, χ_2, tales que $\varphi_1 \sim Q^1_1 x_1 \ldots Q^1_n x_n\, \chi_1$ y $\varphi_2 \sim Q^2_1 y_1 \ldots Q^2_m y_m\, \chi_2$. Consideremos los vectores de variables nuevas $\bar{u} = (u_1, \ldots, u_n)$ y $\bar{v} = (v_1, \ldots, v_m)$. Entonces, $\varphi_1 \sim Q^1_1 u_1 \ldots Q^1_n u_n\, \chi_1[\bar{u}/\bar{x}]$ y $\varphi_2 \sim Q^2_1 v_1 \ldots Q^2_m v_m\, \chi_2[\bar{v}/\bar{y}]$. Ahora, como $\bar{u}$ y $\bar{v}$ son variables nuevas, aplicando repetidamente el lema 2.54, obtenemos:

$$\begin{array}{rcl}
\varphi_1 \lor \varphi_2 & \sim & Q^1_1 x_1 (Q^1_2 x_2 \ldots Q^1_n x_n\, \chi_1) \lor Q^2_1 y_1 \ldots Q^2_m y_m\, \chi_2 \\
& \sim & Q^1_1 u_1 (Q^1_2 x_2 \ldots Q^1_n x_n\, \chi_1[u_1/x_1] \lor Q^2_1 y_1 \ldots Q^2_m y_m\, \chi_2) \\
& \vdots & \\
& \sim & Q^1_1 u_1 \ldots Q^1_n u_n (\chi_1[\bar{u}/\bar{x}] \lor Q^2_1 y_1 \ldots Q^2_m y_m\, \chi_2) \\
& \sim & Q^1_1 u_1 \ldots Q^1_n u_n (Q^2_1 y_1 \ldots Q^2_m y_m\, \chi_2 \lor \chi_1[\bar{u}/\bar{x}]) \\
& \sim & Q^1_1 u_1 \ldots Q^1_n u_n Q^2_1 v_1 (Q^2_2 y_2 \ldots Q^2_m y_m\, \chi_2[v_1/y_1] \lor \chi_1[\bar{u}/\bar{x}]) \\
& \vdots & \\
& \sim & Q^1_1 u_1 \ldots Q^1_n u_n Q^2_1 v_1 \ldots Q^2_m v_m (\chi_2[\bar{v}/\bar{y}] \lor \chi_1[\bar{u}/\bar{x}]),
\end{array}$$

y ya tenemos una forma prenexa. □

2.11. Método de los *tableaux*

En esta sección desarrollaremos el método de los *tableaux* para lógica de primer orden, análogo al que ya presentamos en el anterior capítulo. Por ello, remitimos toda motivación informal acerca de este procedimiento a lo que ya dijimos allí. Para aprender métodos de este tipo siempre son de ayuda los ejemplos, que hemos incluido al final de la siguiente subsección.

[6] En realidad no hace falta demostrar el paso inductivo para más conectivas ya que el resto de conectivas pueden expresarse en términos de $\neg$ y $\lor$, como vimos en 1.48

2.11.1. Nociones básicas y ejemplos

Como veremos, para aplicar una regla de construcción de *tableaux* necesitaremos unas constantes especiales, llamadas *auxiliares*. Por tanto, supondremos la existencia de un conjunto de estas constantes, $C_A = \{c_0, c_1, \dots\}$, en nuestra signatura.

Siempre que tengamos una fórmula existencial en un *tableaux* podremos seleccionar un elemento del conjunto C_A que la verifica. Más adelante veremos un ejemplo.

Las fórmulas que pueden aparecer ahora en nuestros *tableaux* obviamente incluyen a las que lo hacían en los del cálculo proposicional, pero se incorporan algunas nuevas. Es decir, además de las α, β, σ-fórmulas, definimos:

- γ-*fórmulas* (también llamadas *universales*):

γ	$\gamma(t)$
$\forall x\, \varphi$	$\varphi[t/x]$
$\neg\exists x\, \varphi$	$\neg\varphi[t/x]$

  Siendo $t \in TERM_S$.

- δ-*fórmulas* (también llamadas *existenciales*):

δ	$\delta(c)$
$\exists x\, \varphi$	$\varphi[c/x]$
$\neg\forall x\, \varphi$	$\neg\varphi[c/x]$

  Siendo $c \in C_A$.

- *Axiomas de igualdad*, EQ_S:

(RF) $\forall x\, x \doteq x$.

(IM) $\forall x\, \forall y\, x \doteq y \to y \doteq x$.

(TR) $\forall x\, \forall y\, \forall z\, x \doteq y \wedge y \doteq z \to x \doteq z$.

(ST$_1$) Sea $f|_n \in Fn_S$. Entonces,

$$\forall x_1, \dots, \forall x_n\, \forall y_1, \dots, \forall y_n \bigwedge_{i=1}^{n} x_i \doteq y_i \to f(x_1, \dots, x_n) \doteq f(y_1, \dots, y_n).$$

(ST$_2$) Sea $p|_n \in Pd_S$. Entonces,

$$\forall x_1, \dots, \forall x_n\, \forall y_1, \dots, \forall y_n \bigwedge_{i=1}^{n} x_i \doteq y_i \to (p(x_1, \dots, x_n) \to p(y_1, \dots, y_n)).$$

Los nombres de las reglas RF, IM, TR se deben a que a estas propiedades se conocen como reflexiva, simétrica y transitiva. En el caso de las dos últimas se debe a que son reglas de sustitución.

Un resultado análogo al correspondiente de lógica proposicional es el siguiente:

Proposición 2.56. *Sean S signatura, $\varphi \in FORM_S$. Entonces φ verifica una de las siguientes condiciones:*

1. *Es de la forma ψ, $\neg\psi$, $\perp$ o $\top$, con $\psi \in FORMAT_S$.*
2. *Es σ-fórmula.*
3. *Es α-fórmula.*
4. *Es β-fórmula.*
5. *Es γ-fórmula.*
6. *Es δ-fórmula.*

Esto asegura que nuestro método nos permitirá descomponer fórmulas arbitrarias en otras de la forma $\perp$, $\top$ ψ o $\neg\psi$, con $\psi \in FORMAT_S$.

Ahora describimos las reglas para la construcción de *tableaux* asociados a un conjunto de fórmulas Σ:

- Las reglas $R_{ini}, R_{hip}, R_\sigma, R_\alpha, R_\beta$.
- *Regla* γ, R_γ: Sea

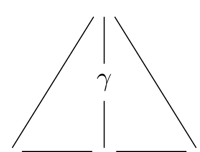
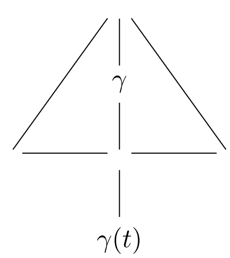


tableau de Σ. Entonces



es *tableau* de Σ, donde γ es γ-fórmula y $\gamma(t)$ su versión simplificada.
Es decir, siempre podemos añadir a una rama abierta una versión simplificada de cualquier γ-fórmula suya y seguimos teniendo un *tableau* de Σ.

- *Regla* δ, R_δ: Sea

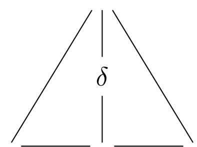
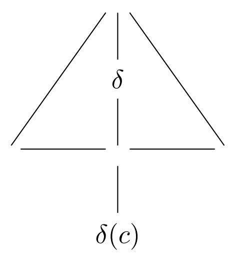


tableau de Σ. Entonces



es *tableau* de Σ, donde δ es δ-fórmula y $\delta(c)$ su versión simplificada.
Es decir, siempre que tengamos una rama abierta que contenga una δ-fórmula, δ, la podemos extender con su versión simplificada $\delta(c)$, siendo $c \in C_A$ nueva en tal rama, y seguimos teniendo un *tableau* de Σ.

- Dado un axioma de igualdad, $\theta \in EQ_S$, la *regla* θ: Sea

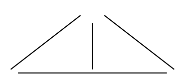
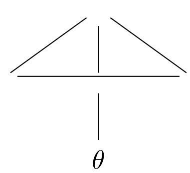


tableau de Σ. Entonces



Es decir, siempre podemos introducir cada axioma de igualdad θ a una rama.

El siguiente ejemplo muestra por qué, al aplicar la regla δ, la constante auxiliar introducida en la rama debe ser nueva en ella.

Ejemplo 2.57. Consideremos el siguiente razonamiento:

Alguien mató a Trotsky
Stalin pagó a alguien.

Stalin pagó a alguien que mató a Trotsky $\quad\therefore$

Lo formalizamos como:

$$\begin{array}{ll}
\varphi_1\colon \exists x\, M(x, Trotsky) & \\
\varphi_2\colon \exists x\, P(Stalin, x). & \\
\hline
\psi\colon \exists x\, M(x, Trotsky) \wedge P(Stalin, x) & \therefore
\end{array}$$

Es decir, tenemos el conjunto $\Sigma = \{\varphi_1, \varphi_2, \neg\psi\}$.

$$\begin{tikzcd}
1.\neg\exists x M(x, Trotsky) \wedge P(Stalin, x) \arrow[d, "hip(1)"'] & \\
2.\exists x M(x, Trotsky) \arrow[d, "hip(2)"'] & \\
3.\exists x P(Stalin, x) \arrow[d, "\delta{,}2"'] & \\
4.M(Mercader, Trotsky) \arrow[d, "\delta{,}3"'] & \\
5.P(Stalin, Mercader) \arrow[d, "\gamma{,}1"'] & \\
6.\neg P(Stalin, Mercader) \wedge M(Mercader, Trotsky) \arrow[d, "\beta{,}6"'] \arrow[dr, "\beta{,}6"] & \\
7.\neg P(Stalin, Mercader)\#5,7 & 8.\neg M(Mercader, Trotsky)\#4,8
\end{tikzcd}$$

En el punto 4 hemos aplicado correctamente una regla δ introduciendo la constante auxiliar $Mercader$, y en el 5 hemos vuelto a introducirla por la misma regla, es decir, hemos introducido una constante auxiliar no nueva en la rama. Esto nos ha llevado al absurdo de que la deducción original era válida.

Ejemplo 2.58. Un ejemplo sencillo. Veamos si es válida la siguiente deducción:

Algunos niños son buenos
Todo lo bueno se come.

Algunos niños se comen $\quad \therefore$

Que formalizamos como:

φ_1: $\exists x\,(ni(x) \wedge bu(x))$
φ_2: $\forall x\,(bu(x) \rightarrow co(x))$.

ψ: $\exists x\,(ni(x) \wedge co(x)) \quad \therefore$

Es decir, en este caso, $\Sigma = \{\varphi_1, \varphi_2, \neg\psi\}$. Veamos un tableau cerrado de Σ:

$$
\begin{tikzcd}
1. \exists x(ni(x) \wedge bu(x)) \arrow[d] & \\
2. \forall x(bu(x) \rightarrow co(x)) \arrow[d] & \\
3. \neg\exists x(ni(x) \wedge co(x)) \arrow[d, "\delta{,}1"'] & \\
4. ni(Pepe) \wedge bu(Pepe) \arrow[d, "\alpha{,}4"'] & \text{(Donde } Pepe \in C_A\text{)} \\
5. ni(Pepe) \arrow[d] & \\
6. bu(Pepe) \arrow[d, "\gamma{,}2{,}pepe/x"] & \\
7. bu(Pepe) \rightarrow co(Pepe) \arrow[d, "\beta{,}7"] \arrow[dr, "\beta{,}7"] & \\
9. co(Pepe) \arrow[d, "\gamma{,}3{,}pepe/x"] & \neg bu(Pepe)\#6,8 \\
10. \neg(ni(pepe) \wedge co(Pepe)) \arrow[d, "\beta{,}10"] \arrow[dr, "\beta{,}10"] & \\
12. \neg co(Pepe)\#9,12 & 11. \neg ni(Pepe)\#5,11
\end{tikzcd}
$$

2.11.2. Corrección

Tal y como hicimos en el método de los *tableaux* para lógica proposicional, dados $\Phi \subseteq FORM_S$ y $\varphi \in FORM_S$, escribimos $\Phi \vdash_{tb} \varphi$ cuando existe un *tableau* cerrado para $\Phi \cup \{\neg\varphi\}$.

Recordemos del capítulo anterior que decimos del método de los *tableaux* que tiene *corrección* si $\Phi \vdash_{tb} \varphi$ implica $\Phi \vDash \varphi$, y decimos que tiene *completitud* si $\Phi \vDash \varphi$ implica $\Phi \vdash_{tb} \varphi$.
Vamos a comprobar que el método de los *tableaux* de primer orden tiene ambas propiedades, empezando por la corrección.

Comencemos con el siguiente:

Lema 2.59. *Para todo $\theta \in EQ_S$, $\vDash \theta$.*

Demostración.

(RF) $\forall x\, x \doteq x$ es tautología ya que dada una interpretación $\mathfrak{I}$, se cumple $\mathfrak{I}[a/x] \vDash x \doteq x$ para todo $a \in A$, ya que obviamente $a = a$.

(IM) $\forall x\, \forall y\, x \doteq y \to y \doteq x$ es tautología ya que, dada una interpretación $\mathfrak{I}$, se cumple $\mathfrak{I}[a/x][b/y] \vDash x \doteq y \to y \doteq x$ para todos $a, b \in A$, ya que si $a = b$, $\mathfrak{I}[a/x][b/y]$ verifica $x \doteq y$ y $y \doteq x$, por tanto verifica $x \doteq y \to y \doteq x$. Si $a \neq b$, $\mathfrak{I}[a/x][b/y]$ no verifican $x \doteq y$ ni $y \doteq x$, por tanto de nuevo verifica $x \doteq y \to y \doteq x$.

(TR) $\forall x\, \forall t\, \forall z\, x \doteq y \wedge y \doteq z \to x \doteq z$ es una tautología. Este caso se demuestra de forma similar a los anteriores, comprobando que $\mathfrak{I}[a/x][b/y][c/z]$ siempre verifica $x \doteq y \wedge y \doteq z \to x \doteq z$. Se puede hacer como el apartado anterior dividiendo en los casos $a \neq b$, $a = b \neq c$, $a = b = c$.

(ST_1) Sea $f|_n \in Fn_S$. Entonces,

$$\forall x_1, \ldots, \forall x_n\, \forall y_1, \ldots, \forall y_n \bigwedge_{i=1}^{n} x_i \doteq y_i \to f(x_1, \ldots, x_n) \doteq f(y_1, \ldots, y_n)$$

es una tautología. Se omite la demostración.

(ST_2) Sea $p|_n \in Pd_S$. Entonces,

$$\forall x_1, \ldots, \forall x_n\, \forall y_1, \ldots, \forall y_n \bigwedge_{i=1}^{n} x_i \doteq y_i \to p(x_1, \ldots, x_n) \to p(y_1, \ldots, y_n)$$

es una tautología. Se omite la demostración.

□

El siguiente resultado es fundamental, nos permite extender una rama satisfactible del *tableaux* a otra que también lo es:

Proposición 2.60. *Sea S signatura. Sean $\Phi \subseteq FORM_S$ y $\mathfrak{I}$ S-interpretación tales que $\mathfrak{I} \vDash \Phi$. Entonces existe una S-interpretación $\mathfrak{I}'$ tal que $\mathfrak{I} \sim_\varphi \mathfrak{I}'$ para toda $\varphi \in \Phi$ de modo que:*

1. *Sea $\sigma \in \Phi$ fórmula simplificable. Entonces $\mathfrak{I}' \vDash \Phi \cup \{\sigma_1\}$, con $\sigma \sim \sigma_1$ la simplificación de σ.*
2. *Sea $\alpha \in \Phi$ una α-fórmula. Entonces $\mathfrak{I}' \vDash \Phi \cup \{\alpha_1, \alpha_2\}$, con $\alpha \sim \alpha_1 \wedge \alpha_2$ la descomposición de α.*
3. *Sea $\beta \in \Phi$ una β-fórmula. Entonces $\mathfrak{I}' \vDash \Phi \cup \{\beta_1\}$ o $\mathfrak{I}' \vDash \Phi \cup \{\beta_2\}$, con $\beta \sim \beta_1 \vee \beta_2$ la descomposición de β.*
4. *Sean $\gamma \in \Phi$ γ-fórmula, $t \in TERM_S$. Entonces $\mathfrak{I}' \vDash \Phi \cup \{\gamma(t)\}$.*
5. *Sean $\delta \in \Phi$ δ-fórmula, $c \in C_A$ constante auxiliar nueva en Φ. Entonces $\mathfrak{I}' \vDash \Phi \cup \{\delta(c)\}$.*
6. *Para cada $\theta \in EQ_S$, $\mathfrak{I}' \vDash \Phi \cup \{\theta\}$.*

Demostración.
(1), (2) y (3) son análogos a los de lógica proposicional. Basta hacer que $\mathfrak{I} = \mathfrak{I}'$ y el hecho de que:

$$\sigma \sim \sigma_1, \quad \alpha \sim \alpha_1 \wedge \alpha_2, \quad \beta \sim \beta_1 \vee \beta_2.$$

Veamos (4). Sea $\gamma \in \Phi$ y supongamos, sin pérdida de generalidad, que γ es de la forma $\forall x\, \varphi$ (si fuese de la forma $\neg\exists x\, \varphi$, usamos que esto es equivalente a $\forall x\, \neg\varphi$ y hacemos $\psi := \neg\varphi$). Tomemos $\mathfrak{I}' = \mathfrak{I}$.
Consideremos el conjunto soporte de la interpretación $\mathfrak{I}$, A. Sea $t \in TERM_S$, y sea $a := t^{\mathfrak{I}} \in A$. Dado que $\forall x\, \varphi \in \Phi$, por hipótesis, $\mathfrak{I} \vDash \forall x\, \varphi$, y por tanto, $\mathfrak{I}[b/x] \vDash \varphi$ para todo $b \in A$, en especial, $\mathfrak{I}[a/x] \vDash \varphi$. Por el Lema de Sustitución, $\mathfrak{I} \vDash \varphi[t/x]$, y notando que $\varphi[t/x] = \gamma(t)$, obtenemos el resultado.

Veamos (5). Sea $\delta \in \Phi$ y supongamos que $\delta = \exists x\, \varphi$, para $\varphi \in FORM_S$ (en caso de que δ fuese de la forma $\neg\forall x\, \varphi$, entonces sería equivalente a $\exists x\, \neg\varphi$ y, haciendo que $\psi := \neg\varphi$, habríamos terminado). Sea $c \in C_A$ constante auxiliar nueva en Φ.
Consideremos de nuevo el conjunto soporte de $\mathfrak{I} := \langle \mathfrak{A}, \sigma \rangle$, A. Como $\exists x\, \varphi \in \Phi$, por hipótesis, $\mathfrak{I} \vDash \exists x\, \varphi$, luego existe $a \in A$ tal que $\mathfrak{I}[a/x] \vDash \varphi$. Ahora definimos $\mathfrak{A}'$ como $\mathfrak{A}$ excepto que $c^{\mathfrak{A}'} = a$. Definamos $\mathfrak{I}' := \langle \mathfrak{A}', \sigma \rangle$.
Al ser c constante nueva para Φ, es decir, para cada elemento de Φ, tenemos que $\mathfrak{I} \sim_\psi \mathfrak{I}'$, para cada $\psi \in \Phi$, por tanto se cumple $\mathfrak{I} \vDash \psi$ para toda $\psi \in \Phi$. Por el Lema de Sustitución,

$$(\varphi[c/x])^{\mathfrak{I}'} = \varphi^{\mathfrak{I}'[c^{\mathfrak{I}'}/x]} = \varphi^{\mathfrak{I}'[a/x]}.$$

Y dado que $\mathfrak{I}'[a/x] \sim_{\varphi} \mathfrak{I}[a/x]$ por ser c nueva, aplicando una vez más el Lema de Sustitución,

$$\varphi^{\mathfrak{I}'[a/x]} = \varphi^{\mathfrak{I}[a/x]} = V.$$

Lo que nos da el resultado.

Finalmente, (6) se deduce de 2.59. $\square$

Del siguiente resultado se deducirá como corolario la corrección del método de los *tableaux* de primer orden:

Teorema 2.61. *Sea S signatura. Sea $\Phi \subseteq FORM_S$ conjunto de fórmulas satisfactible y T un tableau finito para Φ. Entonces T es abierto y existe una rama abierta de T, r, tal que Γ_r es satisfactible.*

Demostración. Demostremos que existe una S-interpretación $\mathfrak{I}$ y una tal rama r de modo que $\mathfrak{I} \models \Phi$ y $\mathfrak{I} \models \Gamma_r$.

Al ser T finito, se ha construido en un número finito de reglas $R_{ini}, R_1, \ldots, R_n$, dando lugar la sucesión de *tableaux* $T_0, T_1, \ldots, T_n = T$. Por tanto, procedemos por inducción sobre n.

Si $n = 0$, entonces T_0 es un *tableau* inicial, es decir, tiene la forma $\begin{array}{c} \varphi_1 \\ | \\ \varphi_2 \\ \vdots \\ \varphi_k \end{array}$.

Entonces r es la única rama disponible, y $\Gamma_r = \{\varphi_1, \ldots, \varphi_k\} \subseteq \Phi$, por lo que, dado que Φ es satisfactible, Γ_r lo es. Por tanto, hemos encontrado una S-interpretación que verifica lo deseado.

Sea $n > 0$ y supongamos que el resultado es cierto para el *tableau* T_{n-1} y veamos que lo es para T_n. Al aplicar la regla R_n a T_{n-1}, extendemos una rama r. Además, por hipótesis de inducción, hay una rama r' tal que $\Gamma_{r'}$ es satisfactible.

Si $r \neq r'$ entonces r' es rama de T_n y, como hemos dicho, verifica que $\Gamma_{r'}$ es satisfactible, con lo que obtenemos el resultado.

Si $r = r'$, debemos distinguir los siguientes casos, dependiendo de qué tipo de regla es R_n:

1. R_n es una regla inicial, de tipo σ, α o β. La demostración es análoga a la que vimos en lógica proposicional.
2. R_n es una regla δ. Sea Γ_r^k el conjunto de fórmulas de la rama r en el árbol k-ésimo. Entonces existe $\delta \in \Gamma_r^{n-1}$ tal que $\Gamma_r^n = \Gamma_r^{n-1} \cup \{\delta(c)\}$, con $c \in C_A$

nueva en r. Por hipótesis de inducción, existe una S-interpretación $\mathfrak{I}$ tal que $\mathfrak{I} \models \Phi$ y $\mathfrak{I} \models \Gamma_r^{n-1}$.

Por 2.60, existe $\mathfrak{I}'$ tal que $\mathfrak{I}' \models \Gamma_r^{n-1} \cup \{\delta(c)\}$, es decir, tal que $\mathfrak{I}' \models \Gamma_r^n$. Al ser c nueva en r, $c \notin voc(\psi)$, para cada $\psi \in \Phi$, por tanto como en la prueba de 2.60 vimos que $\mathfrak{I}'$ solo difiere de $\mathfrak{I}$ en c, $\mathfrak{I}' \sim_\psi \mathfrak{I}$, y por tanto $\mathfrak{I}' \models \psi$, es decir, $\mathfrak{I}' \models \Phi$.

3. R_n es una regla γ. Existe $\gamma \in \Gamma_r^{n-1}$ tal que $\Gamma_r^n = \Gamma_r^{n-1} \cup \{\gamma(t)\}$, siendo $t \in TERM_S$. Por hipótesis, existe una S-interpretación $\mathfrak{I}$ con $\mathfrak{I} \models \Phi$ y $\mathfrak{I} \models \Gamma_r^{n-1}$.

   Por la prueba de 2.60, la misma $\mathfrak{I}$ cumple $\mathfrak{I} \models \Gamma_r^{n-1} \cup \{\gamma(t)\}$, por tanto $\mathfrak{I} \models \Gamma_r^n$.

4. R_n es una regla θ, con $\theta \in EQ_S$. Entonces, $\Gamma_r^n = \Gamma_r^{n-1} \cup \{\theta\}$. Por hipótesis de inducción, existe una S-interpretación $\mathfrak{I}$ tal que $\mathfrak{I} \models \Phi$ y, $\mathfrak{I} \models \Gamma_r^{n-1}$. Por tanto, $\mathfrak{I} \models \Gamma_r^n$.

□

Corolario 2.62. *Sea S signatura. Si $\Phi \subseteq FORM_S$ y Φ tiene un tableau cerrado, entonces Φ es insatisfactible.*

Demostración. Basta notar que este enunciado es la forma contrapositiva de 2.61. □

Finalmente, obtenemos el resultado al que hemos dedicado esta subsección:

Corolario 2.63. *(Corrección) Sean S signatura, $\Phi \subseteq FORM_S$, $\varphi \in FORM_S$. Entonces, si $\Phi \vdash_{tb} \varphi$, entonces $\Phi \models \varphi$.*

Demostración. Si $\Phi \vdash_{tb} \varphi$, entonces $\Phi \cup \{\neg\varphi\}$ tiene un *tableau* cerrado y, por 2.62, $\Phi \cup \{\neg\varphi\}$ es insatisfactible. Por el resultado análogo a 1.30 para lógica de primer orden, esto implica que $\Phi \models \varphi$. □

2.11.3. Completitud. Conceptos Básicos

El estudio de la completitud va a ser más complicado que en lógica proposicional. Para poder estudiarla, vamos a necesitar introducir una serie de nociones fundamentales de teoría de grafos:

Definiciones 2.64.

1) Un *grafo no dirigido* es un par ordenado $T = \langle N, E \rangle$, donde N es el conjunto de *nodos* y E es el conjunto de *aristas*, que son pares sin ordenar de nodos distintos.

2) Un camino es una sucesión de (≥ 1) aristas $(a_1, \ldots, a_n)$ de forma que a_i comparte uno de sus vértices con a_{i-1} y el otro con a_{i+1} para $i = 2, 3, \ldots, n-1$. Llamamos a n *longitud del camino*.

3) Decimos que el camino es *simple* si cada vértice del camino solo aparece hasta dos veces, una al final de una arista y otra al principio de la siguiente. Decimos que un camino es un *ciclo* si es simple y el primer vértice de la primera arista es el segundo vértice de la última arista.

4) Decimos que un grafo es *acíclico* si no tiene ciclos. Decimos que un grafo es *conexo* si para cada par de nodos hay un camino entre ellos.

5) Un *árbol (libre)* es un grafo no dirigido conexo y acíclico. Equivalentemente, es un grafo tal que entre cada par de sus nodos hay un solo camino simple que los une.

6) Sea $G = \langle V, E\rangle$ grafo no dirigido. Se llama *grado* de $v \in V$, $deg(v)$, al número de elementos de E que inciden sobre él.

Se tienen las siguientes propiedades:

Proposición 2.65. *Sea $T = \langle N, E\rangle$ un grafo:*

1. *Si T es un árbol y le quitamos un elemento de E, T deja de ser conexo.*

2. *Si T tiene finitos nodos, T es un árbol si y solo si $|E| = |N| - 1$.*

Demostración. Se omite la demostración. $\square$

Definiciones 2.66.

1) Un árbol $T = \langle N, E\rangle$ se llama *árbol con raíz* si se selecciona un elemento especial de N, llamado *raíz* de T.

2) Sea $T = \langle N, E\rangle$ árbol con raíz n_0. Se llama *altura* de $n \in N$, $h(n)$, a la longitud del único camino simple que une n_0 con n.

3) Sea $T = \langle N, E\rangle$ árbol de raíz n_0. Se dice que $n_1 \in N$ es *hijo de n* si hay una arista que conecta n y n_1 y $h(n_1) = h(n) + 1$. Se dice que n_1 y n_2 son *hermanos* si son hijos del mismo nodo.

4) Sea $T = \langle N, E\rangle$ árbol de raíz n_0. Llamamos *rama finita* a un camino simple desde n_0 que no está contenido en ningún otro camino simple desde n_0. Una *rama infinita* será una sucesión de aristas $a_1, a_2, \ldots$ tal que a_i comparte un vértice con a_{i-1} y el otro con a_{i+1} para $i \geq 2$, que parte desde n_0 y en que no se repiten vértices (es decir, como un camino simple desde n_0 pero con infinitos vértices).

De ahora en adelante consideraremos árboles con una raíz determinada.

Definición 2.67. Sean $T_1 := \langle N_1, E_1\rangle$, $T_2 := \langle N_2, E_2\rangle$ dos árboles con la misma raíz. Decimos que T_1 *está incluido en* T_2, $T_1 \leq T_2$, si:

- $N_1 \subseteq N_2$.
- $E_1 \subseteq E_2$.

Definición 2.68. Sean $T_0, \dots, T_k, \dots$ árboles tales que $T_0 \leq \cdots \leq T_k \leq \dots$. Definimos:

$$\lim_{n \in \mathbb{N}} T_n := \langle N, E \rangle$$

siendo $N := \bigcup_{n \in \mathbb{N}} N_n$ y $E := \bigcup_{n \in \mathbb{N}} E_n$ y con la misma raíz que la de T_0.

Proposición 2.69. *Sea $T_0 \leq \cdots \leq T_k \leq \dots$ sucesión de árboles. Sea $T := \lim_{n \in \mathbb{N}} T_n$. Entonces:*

1. *$T_i \leq T$, para todo $i \in \mathbb{N}$.*

2. *T es un árbol.*

Demostración.

1. Se sigue de las definiciones anteriores.

2. Comencemos demostrando que $T = \langle T, N \rangle$ es conexo. Sean $n_1, n_2 \in N$. Como $N := \bigcup_{n \in \mathbb{N}} N_n$ y $N_i \subseteq N_j$, para todo $i \leq j$, existe $k \in \mathbb{N}$ tal que $n_1, n_2 \in N_k$. Al ser $T_k = \langle N_k, E_k \rangle$ conexo, existe un camino en T_k que une n_1 y n_2. Al ser $E_k \subseteq E$ por definición, también es un camino en T, lo que nos da la conexión de T.

   Supongamos que existen dos nodos $n_1, n_2 \in N$ tales que existen dos caminos simples distintos $e_1, \dots, e_l$ y $f_1, \dots, f_k$ que los conectan. Es claro, razonando igual que antes, que existe $i \in \mathbb{N}$ tal que $n_1, n_2 \in N_i$ y $e_1, \dots, e_l, f_1, \dots, f_k \in E_i$ y, al ser $T_i = \langle N_i, E_i \rangle$ árbol, $e_1, \dots, e_l$ y $f_1, \dots, f_k$ son iguales, lo que implica que T sea acíclico.

$\square$

Nótese que nuestros *tableaux* son árboles con raíz. Además, se cumplirá que las ramas cerradas serán siempre finitas, ya que una rama cerrada se forma cuando dos fórmulas en la rama están en contradicción y ya no hace falta seguir extendiendo la rama.

Necesitamos el siguiente resultado técnico de la teoría de grafos:

Teorema 2.70. *(Lema de König para árboles) Sea T un árbol con raíz con infinitos nodos tal que cada nodo tiene finitos hijos. Entonces T tiene una rama infinita.*

Demostración. Supongamos que tenemos una ordenación de los vértices, $(v_n)_{n \in \mathbb{N}}$. La raíz del árbol será v_{i_0}. Dados dos nodos a y b distintos, decimos que b es descendiente de a si el único camino simple c_b desde v_0 hasta b pasa por a. Como $b \neq a$, en c_b tenemos un vértice siguiente a a, que por tanto es hijo de a. De modo que cualquier descendiente de un vértice es uno de sus hijos o descendiente de uno de sus hijos.

Entonces, v_{i_0} tiene infinitos descendientes (todos los vértices salvo v_0). Estos infinitos descendientes contienen a los finitos hijos de v_{i_0} y a sus descendientes.

Por tanto alguno de sus hijos tiene que tener infinitos descendientes. Sea v_{i_1} el menor de ellos. Como v_{i_1} tiene infinitos descendientes, de la misma forma que antes escogemos el menor de sus hijos, v_{i_2}, que tiene infinitos descendientes.

De este modo creamos una sucesión de vértices $(v_{i_n})_{n\in\mathbb{N}}$. Como por definición v_{i_n} tiene altura n, no se repiten vértices. Por tanto la sucesión de aristas a_n que unen v_{i_n} con $v_{i_{n+1}}$ forman una rama infinita.

□

Proposición 2.71. *Si T es tableau cerrado, es finito.*

Demostración. Usaremos reducción al absurdo. Sea T un *tableaux* infinito cerrado. Observemos que, en todo *tableau*, el número máximo de hijos de cada nodo es dos: los que se pueden obtener por la aplicación de una regla β. Por tanto, por el lema de König 2,70 el *tableau* contiene una rama infinita. Esto contradice el hecho de que todas las ramas cerradas son finitas. □

2.11.4. Conjuntos de Hintikka para lógica de primer orden

Para continuar con el teorema de completitud, lo siguiente es definir el análogo a los conjuntos de Hintikka que vimos en la lógica proposicional, pero ahora para la lógica de primer orden.

Lo primero que necesitamos es el concepto de *término adecuado* para un conjunto de fórmulas $\Phi \subseteq FORM_S$

Definición 2.72. Sea S una signatura, $t \in TERM_S$ y $\Phi \subseteq FORM_S$. Diremos que t es *adecuado* para Φ si se cumple:

- $voc(t) \subseteq voc(\Phi) := \bigcup_{\varphi\in\Phi} voc(\varphi)$
- $var(t) \subseteq lib(\Phi) := \bigcup_{\varphi\in\Phi} lib(\varphi)$

Puede darse el caso de que no encontremos ningún término adecuado para un conjunto de fórmulas. Por ejemplo consideremos el conjunto de fórmulas

$$\Phi = \{\exists x\ (p(x) \to q(x))\}$$

En este caso $(voc(\Phi) \cap Ct_S) \cup lib(\Phi) = \emptyset$, por tanto no podemos construir términos adecuados. En este caso, como en el método de los *tableaux* nos serán necesarios términos adecuados para desarrollar el algoritmo, tomaremos como término adecuado cualquier constante auxiliar $c \in C_A$ nueva.

Definición 2.73 (Conjunto de Hintikka)**.** Sea S una signatura, $\Phi \in FORM_S$ y $T_\Phi := \{t \mid t \in TERM_S \text{ adecuado para } \Phi\}$. Diremos que Φ es de *Hintikka* si verifica todas estas propiedades:

- Φ es coherente:
  - $\bot \notin \Phi$

- no existe $\varphi \in \Phi$ tal que $\neg\varphi \in \Phi$.

- Φ es α-saturado: si $\alpha \in \Phi$ es una α-fórmula, $\alpha_1, \alpha_2 \in \Phi$.
- Φ es β-saturado: si $\beta \in \Phi$ es una β-fórmula, $\beta_1 \in \Phi$ o $\beta_2 \in \Phi$.
- Φ es σ-saturado: si $\sigma \in \Phi$ es una σ-fórmula, $\sigma_1 \in \Phi$.
- Φ es γ-saturado: si $\gamma \in \Phi$ es una γ-fórmula y $t \in T_\Phi$, entonces $\gamma(t) \in \Phi$.
- Φ es δ-saturado: si $\delta \in \Phi$ es una δ-fórmula, entonces existe una constante auxiliar $c \in C_A$ tal que $\delta(c) \in \Phi$.
- Φ es θ-saturado: si $\theta \in \mathrm{EQ}_S$ es un axioma de igualdad, entonces $\theta \in \Phi$.

Consideremos a continuación una signatura cualquiera S. Nuestra intención es demostrar que todo conjunto $\Phi \subseteq FORM_S$ de Hintikka es satisfactible. Es decir, tenemos que crear una interpretación, con una álgebra $\mathfrak{T}$ y una asignación de variables σ_Φ, que haga ciertas todas las fórmulas de Φ. El primer paso para ello es encontrar un conjunto soporte para nuestra álgebra.
Una opción a considerar es el *álgebra libre de términos*, una álgebra conocida que tiene como como conjunto soporte a $TERM_S$. Sin embargo, esta álgebra no es suficiente: Supongamos que tenemos la signatura de la aritmética. Según esta signatura, los términos $s(0) + 0$ y $s(0)$ son distintos, pero representan el mismo elemento. Eso se traduce en que en nuesto álgebra necesitamos establecer una relación de equivalencia de términos. Esa relación la denotaremos como $\equiv_\Phi$, la idea es que $t \equiv_\Phi s$ si $\Phi \models t \doteq s$.

Definición 2.74. Dada S una signatura, $\Phi \subseteq FORM_S$ y $s, t \in TERM_S$ definimos la equivalencia de términos como:

$$t \equiv_\Phi s \quad \text{si y solo si} \quad t \doteq s \in \Phi$$

Proposición 2.75. *Sea S signatura, $\Phi \subseteq FORM_S$. Si Φ es de Hintikka, entonces $\equiv_\Phi \subseteq T_\Phi \times T_\Phi$ es relación de equivalencia.*

Demostración. Probamos las propiedades reflexiva, simétrica y transitiva.

1. *Propiedad reflexiva.* Dado $t \in T_\Phi$ hay que probar que $t \equiv_\Phi t$. Utilizaremos el axioma de reflexión (RF), que está en Φ por ser de $Hintikka$.

$$\gamma := (\forall x\ x \doteq x) \in \Phi$$

   Es una γ-fórmula, luego por ser Φ de $Hintikka$, $\gamma(t) \in \Phi$, esto es:

$$t \doteq t\ \in \Phi$$

   lo que equivale a que $t \equiv_\Phi t$, que es el resultado deseado.

2. *Propiedad simétrica.* Sean $t, s \in T_\Phi$ tales que $t \equiv_\Phi s$, es decir, $t \doteq s \in \Phi$. Al ser Φ de Hintikka contiene al axioma de simetría (IM):

$$\gamma \ := \ (\forall x \, \forall y \, x \doteq y \to y \doteq x)$$

que es una γ-fórmula, por lo que $\gamma(t) \in \Phi$:

$$\gamma(t) \ = \ (\forall y \, t \doteq y \to y \doteq t)$$

que es de nuevo otra γ-fórmula. Se aplica nuevamente la misma regla para s y se tiene que:

$$\gamma(t)(s) \ = \ (t \doteq s \to s \doteq t) \in \Phi$$

que es una β-fórmula, luego se cumple:

$$\neg t \doteq s \in \Phi \ \ o \ \ s \doteq t \in \Phi$$

Como se sabe ya que $t \doteq s \in \Phi$ no queda otra que $s \doteq t \in \Phi$, que equivale a que $s \equiv_\Phi t$, como queríamos probar.

3. *Propiedad transitiva.* Dados $t, s, r \in T_\Phi$ tal que $t \equiv_\Phi s$ y $s \equiv_\Phi r$, hay que demostrar que $t \equiv_\Phi r$. El axioma de transitividad está en Φ, luego:

$$\gamma \ := \ (\forall x \forall y \forall z \ x \doteq y \land y \doteq z \to x \doteq z) \in \Phi$$

que es una γ-fórmula. Aplicando los cambios de las variables por los términos y atendiendo a que Φ es γ-saturado por ser de Hintikka se tiene:

$$\gamma(t)(r)(s) \ = \ (t \doteq s \land s \doteq r \to t \doteq r) \in \Phi$$

que es una β-fórmula. Entonces:

$$\neg(t \doteq s \land s \doteq r) \in \Phi \ \ o \ \ t \doteq r \in \Phi$$

Veamos que la primera no se cumple: de ser así, y al ser una β-fórmula, se tendría que:

$$\neg(t \doteq s) \in \Phi \ \ o \ \ \neg(s \doteq r) \in \Phi$$

pero esto es imposible por la coherencia de Φ, pues tanto $t \doteq s$ como $s \doteq r$ están en Φ. Se deduce que necesariamente $t \doteq r \in \Phi$ y entonces $t \equiv_\Phi r$.

$\square$

Dada una relación de equivalencia $\equiv_\Phi$ podemos crear de forma natural una partición del conjunto considerando el conjunto cociente, es decir, el conjunto de clases de equivalencia inducido por $\equiv_\Phi$. Dado $t \in T_\Phi$, se tiene su clase de equivalencia:

$$[t] := \{s \mid s \in T_\Phi, t \equiv_\Phi s\}$$

y entonces el conjunto cociente es:

$$T_\Phi / \equiv_\Phi \quad = \quad \{[t] \mid t \in T_\Phi\}$$

El soporte del álgebra que vamos a construir es precisamente este conjunto cociente.
Ahora hay que definir el valor de las constantes, las funciones y los símbolos de predicado en el conjunto cociente.

- Si $c \in Ct_S$ definimos

$$c^{\mathfrak{T}_\Phi} = \begin{cases} [c] & \text{si } c \in T_\Phi \\ \text{arbitrario}^7 & \text{si no} \end{cases}$$

- Si $f|_k \in Fn_S$ y $t_1, \ldots t_k \in T_\Phi$

$$f^{\mathfrak{T}_\Phi}([t_1], \ldots [t_k]) = \begin{cases} [f(t_1, \ldots t_k)] & \text{si } f(t_1, \ldots t_k) \in T_\Phi \\ \text{arbitrario} & \text{si no} \end{cases}$$

- Si $p|_k \in Pd_S$ y $t_1, \ldots t_k \in T_\Phi$

$$p^{\mathfrak{T}_\Phi}([t_1], \ldots [t_k]) = \begin{cases} V & \text{si } p(t_1, \ldots t_k) \in \Phi \\ F & \text{si no} \end{cases}$$

Veamos a continuación que el valor de una clase está bien definido, es decir, no depende de los representantes escogidos:

Proposición 2.76. *Sea S una signatura, $\Phi \subseteq FORM_S$ un conjunto de fórmulas de Hintikka, $t_1, \ldots t_k, s_1, \ldots, s_k \in T_\Phi$ términos tales que $[t_i] = [s_i]$ para $1 \leq i \leq k$.*

- *Si $f|_k \in voc(\Phi)$ entonces $[f(t_1, \ldots t_k)] = [f(s_1, \ldots, s_k)]$*
- *Si $p|_k \in voc(\Phi)$ entonces $p(t_1, \ldots t_k) \in \Phi$ si y solo si $p(s_1, \ldots, s_k) \in \Phi$*

Demostración. En primer lugar veamos el caso de los símbolos de función. Puesto que Φ es de Hintikka tenemos que el axioma de igualdad

$$\gamma_0 = \forall x_1 \ldots \forall x_k \forall y_1 \ldots \forall y_k \; x_1 \doteq y_1 \wedge \cdots \wedge x_k \doteq y_k \rightarrow f(x_1, \ldots, x_k) \doteq f(y_1, \ldots, y_k) \in \Phi$$

Definimos ahora las siguientes fórmulas, que estarán en Φ por ser Φ de Hintikka:

$$\begin{aligned} &\gamma_1 = \gamma_0(t_1), \quad \gamma_2 = \gamma_1(t_2), \quad \ldots \quad \gamma_k = \gamma_{k-1}(t_k) \\ &\gamma_{k+1} = \gamma_k(s_1), \; \ldots \; , \gamma_{2k} = \gamma_{2k-1}(s_k) \\ &\gamma_{2k} \; = \; t_1 \doteq s_1 \wedge \cdots \wedge t_k \doteq s_k \rightarrow f(t_1, \ldots, t_k) \doteq f(s_1, \ldots, s_k) \end{aligned}$$

Resulta que γ_{2k} es una β-fórmula. Por tanto $\neg(t_1 \doteq s_1 \wedge \cdots \wedge t_k \doteq s_k) \in \Phi$ o $f(t_1, \ldots, t_k) \doteq f(s_1, \ldots, s_k) \in \Phi$. Hay 2 casos:

7 Por el lema de coincidencia, para nuestros propósitos no importará el valor que tenga.

- $\neg(t_1 \doteq s_1 \wedge \cdots \wedge t_k \doteq s_k) \in \Phi$. Esta es una β-fórmula. En este caso, aplicando la regla β repetidamente llegamos a que alguna de las fórmulas $\neg(t_i \doteq s_i)$ está en Φ. Por otro lado, $[t_i] = [s_i]$ significa $t_i \doteq s_i \in \Phi$. Esto contradice que Φ sea de Hintikka, al tener $\neg(t_i \doteq s_i), t_i \doteq s_i \in \Phi$, por tanto este caso no puede darse.

- $f(t_1, \ldots, t_k) \doteq f(s_1, \ldots, s_k) \in \Phi$. Esta es la definición de $f(t_1, \ldots, t_k) \equiv_\Phi f(s_1, \ldots, s_k)$, por tanto $[f(t_1, \ldots, t_k)] = [f(s_1, \ldots, s_k)]$.

Pasamos ahora a los símbolos de predicado. Nos vale probar que si $p(t_1, \ldots t_k) \in \Phi$, entonces $p(s_1, \ldots, s_k) \in \Phi$. En efecto, la implicación recíproca es consecuencia de la implicación directa al intercambiar $t_1, \ldots, t_k$ por $s_1, \ldots, s_k$ respectivamente.

Puesto que Φ es de Hintikka contiene el axioma de igualdad

$$\gamma_0 = \forall x_1 \ldots \forall x_k \forall y_1 \ldots \forall y_k \, x_1 \doteq y_1 \wedge \cdots \wedge x_k \doteq y_k \rightarrow (p(x_1, \ldots, x_k) \rightarrow p(y_1, \ldots, y_k) \in \Phi).$$

De forma similar a la vista los símbolos de función, llegamos a que la siguiente fórmula está en Φ:

$$\gamma_{2k} \ = \ t_1 \doteq s_1 \wedge \cdots \wedge t_k \doteq s_k \rightarrow (p(t_1, \ldots, t_k) \rightarrow p(s_1, \ldots, s_k))$$

Por tanto, $\neg(t_1 \doteq s_1 \wedge \cdots \wedge t_k \doteq s_k) \in \Phi$ o $p(t_1, \ldots, t_k) \rightarrow p(s_1, \ldots, s_k) \in \Phi$. Hay 2 casos. En el primer caso, igual que con los símbolos de función llegamos a un absurdo. En el segundo caso, tenemos que $p(t_1, \ldots, t_k) \rightarrow p(s_1, \ldots, s_k) \in \Phi$. Esto implica, que o bien $\neg(p(t_1, \ldots, t_k)) \in \Phi$ en cuyo caso $p(t_1, \ldots, t_k) \notin \Phi$, o bien $p(s_1, \ldots, s_k) \in \Phi$. En ambos casos se cumple la implicación que buscamos. $\square$

Ya podemos definir el álgebra que buscamos:

$$\mathfrak{T}_\Phi \ := \ \left\langle T_\Phi / \equiv_\Phi, \{c^{\mathfrak{T}_\Phi} := [c] \mid c \in Ct_S\}, \{f^{\mathfrak{T}_\Phi} \mid f|_k \in Fn_S\}, \{p^{\mathfrak{T}_\Phi} \mid p|_k \in Pd_S\} \right\rangle$$

Además, consideremos la asignación de variables:

$$\sigma_\Phi := \begin{cases} [x] & \text{si } x \in T_\Phi \\ \text{arbitrario} & \text{si no} \end{cases}$$

Y tomamos la interpretación:

$$\mathfrak{I}_\Phi := \langle \mathfrak{T}_\Phi, \sigma_\Phi \rangle$$

Veamos que $\mathfrak{I}_\Phi \models \Phi$. En primer lugar tenemos la siguiente

Proposición 2.77. *Sea S una signatura, $\Phi \subseteq FORM_S$ un conjunto de fórmulas de Hintikka y $t \in T_\Phi$. Entonces $t^{\mathfrak{I}_\Phi} = [t]$.*

Demostración. Se puede hacer por inducción sobre la longitud de los términos. Para longitud 1, es directo de la definición en constantes y variables. Supongamos que se cumple para palabras de longitud $\leq n$, siendo $n > 1$.
Ahora, si t es un término de longitud n, será de la forma $f(t_1, \ldots, t_k)$, donde

los t_i tienen longitudes $< n$. Por tanto, por hipótesis de inducción, se cumple $t_i^{\mathfrak{I}_\Phi} = [t_i]$ para $i = 1, \ldots, k$, ya que $t_i \in T_\Phi$ para todo i. Por tanto:

$$t^{\mathfrak{I}_\Phi} = f^{\mathfrak{I}_\Phi}([t_1], \ldots, [t_k]) = [f(t_1, \ldots, t_k)],$$

lo cual completa la inducción. $\square$

Ahora tenemos que demostrar $\mathfrak{I}_\Phi \models \varphi$ para cada $\varphi \in \Phi$. Para esto no basta con inducción estructural, sino que nos interesa extender la norma que vimos en lógica proposicional:

Definición 2.78. Sea S una signatura. Definimos la *norma* de una fórmula como:

$$\| \cdot \| : FORM_S \to \mathbb{N}$$

de forma recursiva como sigue:

- Caso base: $\|\varphi\| = 0$ si φ es atómica.
- Casos recursivos:
  - $\|\neg\varphi\| = 1 + \|\varphi\|$
  - $\|\varphi \wedge \psi\| = 1 + \|\varphi\| + \|\psi\|$
  - $\|\varphi \vee \psi\| = 1 + \|\varphi\| + \|\psi\|$
  - $\|\varphi \to \psi\| = 2 + \|\varphi\| + \|\psi\|$
  - $\|\varphi \leftrightarrow \psi\| = 5 + 2\|\varphi\| + 2\|\psi\|$
  - $\|\forall x\, \varphi\| = 1 + \|\varphi\|$
  - $\|\exists x\, \varphi\| = 1 + \|\varphi\|$

Conviene hacer notar que:

- $\|\varphi \to \psi\| = \|\neg\varphi \vee \psi\|$
- $\|\varphi \leftrightarrow \psi\| = \|(\varphi \to \psi) \wedge (\psi \to \varphi)\|$

Veamos los siguientes resultados auxiliares:

Lema 2.79.

- *Si α es una α-fórmula, $\|\alpha\| > \|\alpha_1\|$ y $\|\alpha\| > \|\alpha_2\|$*
- *Si β es una β-fórmula, $\|\beta\| > \|\beta_1\|$ y $\|\beta\| > \|\beta_2\|$*
- *Si σ es una σ-fórmula, $\|\sigma\| > \|\sigma_1\|$.*
- *Si $\varphi \in FORM_S$ es una fórmula y $t \in TERM_S$ y $x \in var$, $\|\varphi[t/x]\| = \|\varphi\|$*

Demostración. Las tres primeras son fáciles de obtener por casos. La última se demuestra por una inducción estructural rutinaria en φ. Intuitivamente este lema quiere decir que la norma de una fórmula no depende de los términos involucrados. $\square$

Lema 2.80. *Sea S una signatura, $\Phi \subseteq FORM_S$ de Hintikka y $\varphi \in FORMAT_S$. Entonces, si $\varphi \in \Phi$ se cumple $\mathfrak{I}_\Phi \models \varphi$. De hecho la implicación recíproca también se cumple excepto si $\varphi = \top$.*

Demostración.

- $\varphi = \top$. Trivial
- $\varphi = t \doteq s$. En este caso $\varphi \in \Phi$ si y solo si $t \equiv_\Phi s$, que es lo mismo que $[t] = [s]$. Por la proposición 2.77 esto equivale a $t^{\mathfrak{I}_\Phi} = s^{\mathfrak{I}_\Phi}$, que equivale a $\mathfrak{I}_\Phi \models t \doteq s$.
- $\varphi = p(t_1, \ldots, t_k)$ para $p|_k \in Pd_S$ y $t_1, \ldots, t_k \in T_\Phi$.

  Por definición $p(t_1, \ldots, t_k) \in \Phi$ si y solo si $p^{\mathfrak{T}_\Phi}([t_1], \ldots, [t_k]) = V$. Por tanto tenemos

$$p^{\mathfrak{T}_\Phi}([t_1], \ldots, [t_k]) = p^{\mathfrak{T}_\Phi}(t_1^{\mathfrak{I}_\Phi}, \ldots, t_k^{\mathfrak{I}_\Phi}) = (p(t_1, \ldots, t_k))^{\mathfrak{I}_\Phi}$$

  Puesto que $(p(t_1, \ldots, t_k))^{\mathfrak{I}_\Phi} = V$ es la definición de $\mathfrak{I}_\Phi \models p(t_1, \ldots, t_k) \in \Phi$, tenemos $p(t_1, \ldots, t_k) \in \Phi$ si y solo si $\mathfrak{I}_\Phi \models p(t_1, \ldots, t_k) \in \Phi$.

$\square$

Ahora podemos demostrar el resultado que buscamos.

Teorema 2.81. *Sea S una signatura y $\Phi \subseteq FORM_S$ un conjunto de fórmulas de Hintikka, entonces $\mathfrak{I}_\Phi \models \Phi$.*

Demostración. Sea $\varphi \in \Phi$, probaremos $\mathfrak{I}_\Phi \models \varphi$ por inducción sobre $\|\varphi\|$.

$\|\varphi\| = 0$ Es una de las implicaciones del lema 2.80.

$\|\varphi\| > 0$ En ese caso tenemos varias opciones

- $\varphi = \alpha$ es una α-fórmula. Al ser Φ un conjunto α-saturado, tenemos $\alpha_1, \alpha_2 \in \Phi$. Puesto que $\|\alpha\| > \|\alpha_1\|$ y $\|\alpha\| > \|\alpha_2\|$ podemos aplicar hipótesis de inducción y tenemos $\mathfrak{I}_\Phi \models \alpha_1$ y $\mathfrak{I}_\Phi \models \alpha_2$. Puesto que $\varphi = \alpha \sim \alpha_1 \wedge \alpha_2$ tenemos $\mathfrak{I}_\Phi \models \varphi$.
- $\varphi = \beta$ es una β-fórmula. En este caso, $\beta_1 \in \Phi$ o $\beta_2 \in \Phi$. Esto implica por hipótesis de inducción que $\mathfrak{I}_\Phi \models \beta_1$ o $\mathfrak{I}_\Phi \models \beta_2$. Como $\beta \sim \beta_1 \vee \beta_2$, en ambos casos se cumple que $\mathfrak{I}_\Phi \models \beta$.
- $\varphi = \sigma$ es una σ-fórmula. Similar a las anteriores.
- $\varphi = \gamma$ es una γ-fórmula. Hay 2 casos
  - $\varphi = \forall x\, \psi$. Al ser Φ saturado, tenemos que para todo término $t \in T_\Phi$ se verifica $\psi[t/x] \in \Phi$. Además,

$$||\forall x \psi|| = 1 + ||\psi|| > ||\psi|| = ||\psi[t/x]||,$$

    por tanto por hipótesis de inducción, para todo $t \in T_\Phi$ se cumple $\mathfrak{I}_\Phi \vDash \psi[t/x]$. Esto equivale por el lema de sustitución a que

$\mathfrak{I}_\Phi[t^{\mathfrak{I}_\Phi}/x] \vDash \psi$, es decir, $\mathfrak{I}_\Phi[[t]/x] \vDash \psi$.
Como esto se cumple para cualquier elemento y cada clase de equivalencia $a \in T_\Phi/\equiv_\Phi$ contiene al menos un elemento, tenemos que para todo $a \in T_\Phi/\equiv_\Phi$ se cumple que $\mathfrak{I}_\Phi[a/x] \vDash \psi$. Esto es la definición de que $\mathfrak{I} \vDash \forall x\psi$, y hemos acabado.

- $\varphi = \neg\exists x\,\psi$. Puesto que Φ es saturado tenemos que para todo término $t \in T_\Phi$ se verifica $\neg\psi[t/x] \in \Phi$. Por otro lado

$$\|\neg\exists x\,\psi\| = 2 + \|\psi\| = 1 + \|\neg\psi\| > \|\neg\psi\| = \|\neg\psi[t/x]\|,$$

Por tanto podemos aplicar inducción a los términos $\neg\psi[t/x]$ para $t \in T_\Phi$. Por lo que podemos decir que para cada $t \in T_\Phi$ tenemos $\mathfrak{I}_\Phi \models \neg\psi[t/x]$ y, aplicando el lema de sustitución, tenemos $\mathfrak{I}_\Phi[t^{\mathfrak{I}_\Phi}/x] \models \neg\psi$. Puesto que $t^{\mathfrak{I}_\Phi} = [t]$,
tenemos que para cada $[t] \in T_\Phi/\equiv_\Phi$ se verifica $\mathfrak{I}_\Phi[[t]/x] \models \neg\psi$. Puesto que cada elemento de $t \in T_\Phi$ está en una y solo una clase de equivalencia, podemos decir que para cada $a \in T_\Phi/\equiv_\Phi$ se verifica $\mathfrak{I}_\Phi[a/x] \models \neg\psi$, que es la definición de que $\mathfrak{I}_\Phi \models \forall x\,\neg\psi$. Puesto que $\forall x\,\neg\psi \sim \neg\exists x\,\psi$ tenemos el resultado.

- $\varphi = \delta$ es una δ-fórmula. 2 casos.
  - $\varphi = \exists x\psi$. En este caso, al ser Φ saturado, existe una constante auxiliar $c \in C_A$ tal que $\psi[c/x] \in \Phi$. Además, tenemos que

$$||\exists x\psi|| = 1 + ||\psi|| > ||\psi|| = ||\psi[c/x]||,$$

por tanto por hipótesis de inducción, se cumple que $\mathfrak{I}_\Phi \vDash \psi[c/x]$. Por el lema de sustitución, tenemos que $\mathfrak{I}_\Phi[c^{\mathfrak{I}_\Phi}/x] \vDash \psi$. Es decir, existe un elemento a del conjunto soporte tal que $\mathfrak{I}_\Phi[a/x] \vDash \psi$. Esto es la definición de $\mathfrak{I}_\Phi \vDash \exists x\psi$,y hemos acabado.

  - $\varphi = \neg\forall x\psi$. Caso similar al anterior.

$\square$

2.11.5. *Tableau* canónico. Teorema de compacidad

Volvemos a considerar el concepto de *tableau completo*, análogo al visto en lógica proposicional:

Definición 2.82. Sean S una signatura y $\Phi \subseteq FORM_S$. Un *tableau* T es *completo para* Φ si cada rama abierta r suya verifica:

1. $\Phi \subseteq \Gamma_r$
2. Γ_r es de Hintikka.

La principal diferencia que tenemos respecto a la lógica proposicional es que en lógica de primer orden hemos visto que, a la hora de hacer un razonamiento $\Phi \models \varphi$, el conjunto Φ puede ser infinito o bien, incluso si el conjunto de fórmulas es finito, la aplicación de reglas de tipo γ hace que el conjunto de fórmulas que tengamos que considerar es infinito, dado que el conjunto de términos suele ser infinito. Esto hace necesario considerar árboles infinitos para construir un *tableau* completo, lo que llamaremos *tableau canónico*. En general, es un árbol infinito y por tanto no podremos construirlo en un conjunto finito de pasos. Podemos, en cambio, decir cómo se construye cada uno de esos pasos. El límite de los árboles construidos mediante esos pasos será tal *tableau* canónico.

De ahora en adelante, para la construcción del *tableau* canónico, nos vamos a restringir a signaturas numerables, es decir, aquéllas en que el conjunto de todos los símbolos es numerable.[8]

A continuación obtenemos un resultado relativo al cierre de Kleene:

Lema 2.83. *Si un conjunto A es numerable, A^* es numerable.*

Demostración. Consideremos $A_0 = \{\epsilon\}$, donde ϵ es la cadena vacía. Sea:

$$A_n = A_{n-1} \cup \{a\omega \mid a \in A \ \omega \in A_{n-1}\}^9$$

Puesto que A es numerable, es fácil ver por inducción que A_i es numerable para cada $i \in \mathbb{N}$. Por otro lado, $A^* = \bigcup_{i\in\mathbb{N}} A_i$, por tanto, A^* es numerable. □

Como consecuencia, el conjunto de términos y fórmulas de una signatura numerable es numerable.
Consideremos lo siguiente:

1. S una signatura numerable.
2. El conjunto de constantes auxiliares, que supondremos también numerable:
$$C_A = \{c_0, c_1, \ldots, c_k, \ldots\}$$
3. $\overline{S} := S \cup C_A$, que también es numerable.
4. El conjunto de términos, considerando también el conjunto de constantes auxiliares. Por lo dicho antes, es numerable. Además, consideramos una enumeración:
$$TERM_{\overline{S}} = \{t_0, t_1, \ldots, t_k, \ldots\}$$
5. El conjunto de γ-fórmulas. Puesto que es un subconjunto de fórmulas, también es numerable y consideramos una numeración suya:
$$G := \{\gamma_0, \gamma_1, \ldots, \gamma_k, \ldots\}$$

[8] En la práctica, necesitaremos algo más fuerte: que la signatura sea efectivamente enumerable, es decir, que haya un algoritmo que permita obtener una numeración de los símbolos. Más información en la sección 4.4.

[9] $a\omega$ denota la concatenación de a y ω

6. El conjunto de axiomas de igualdad. Consideramos una numeración de estas fórmulas,
$$\Theta := \{\theta_0, \theta_1, \dots, \theta_k, \dots\}$$

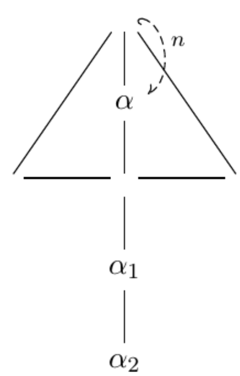
7. El conjunto de fórmulas Φ para las que queremos construir el *tableau*. Una numeración suya es:
$$\Phi := \{\varphi_0, \varphi_1, \dots, \varphi_k, \dots\}$$

Pasamos ahora a construir el *tableau* canónico, teniendo en cuenta todos los anteriores elementos.
Dado un *tableau* finito, vamos a definir varias formas de extenderlo. Esto dará lugar a lo que se llaman *extensiones*. Enumerémoslas a continuación.

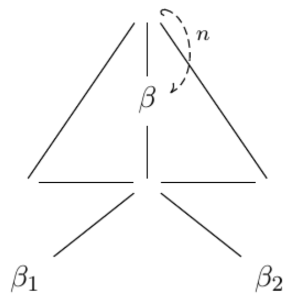
ABSD-extensión

Esta extensión la denotaremos como $\mathbf{ABSD}(T, n)$, donde T es un tableau y n es un entero. Esta extensión desplegará por orden todas las ramas que a profundidad n tengan una fórmula de tipo α, β, σ o δ. En el siguiente orden:

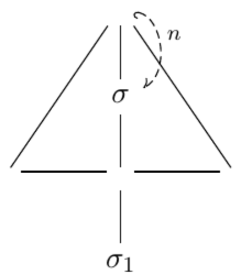
1. Fórmulas de tipo α. En cada rama abierta r que tenga a profundidad menor o igual que n una fórmula de tipo α no desplegada, aplicamos a esa fórmula la regla α:



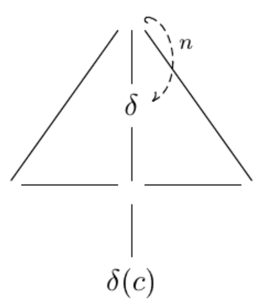
2. Fórmulas de tipo β. En cada rama abierta r que tenga a profundidad menor o igual que n una fórmula de tipo β no desplegada, aplicamos a esa fórmula la regla β:



3. Fórmulas de tipo σ. En cada rama abierta r que tenga a profundidad menor o igual que n una fórmula de tipo σ no desplegada, aplicamos a esa fórmula la regla σ:

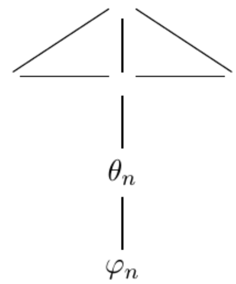


4. Fórmulas de tipo δ. En cada rama abierta r que tenga a profundidad menor o igual que n una fórmula de tipo δ no desplegada, aplicamos a esa fórmula la regla δ:



EF-extensión

Esta extensión la denominamos $\mathbf{EF}(T, n)$. Consiste en añadir a cada rama abierta r de T las fórmulas $\varphi_n \in \Phi$ (según la numeración del conjunto de fórmulas Φ) y la fórmula $\theta_n \in EQ_S$ (según la numeración de las fórmulas de los axiomas de la igualdad).

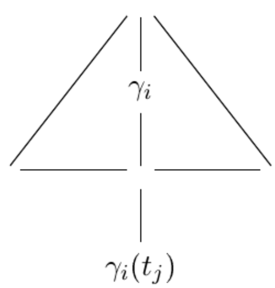


G-extensión

Esta extensión se denota $\mathbf{G}(T, n)$. Consiste en extender cada rama abierta r de T con la fórmula $\gamma_i(t_j)$ donde el par $(i, j) \in \mathbb{N} \times \mathbb{N}$ es el menor par según el siguiente buen orden de $\mathbb{N} \times \mathbb{N}$

$$(i_1, j_1) < (i_2, j_2) \text{ sii } i_1 + j_1 < i_2 + j_2 \text{ o bien } i_1 + j_1 = i_2 + j_2 \text{ e } i_1 < i_2$$

tal que $\gamma_i \in \Gamma_r$ está a altura $\leq n$, $\gamma_i(t_j) \notin \Gamma_r$ y t_j es adecuado para Γ_r. Si no hay ningún término adecuado para Γ_r, aplicamos la regla considerando como término adecuado la primera constante auxiliar.



Por ejemplo, supongamos $\gamma_0, \gamma_2 \in \Gamma_r$ y $\gamma_1 \notin \Gamma_r$. Las fórmulas se van poniendo en el siguiente orden en las ramas: $\gamma_0(t_0)$, $\gamma_0(t_1)$, $\gamma_0(t_2)$ y $\gamma_2(t_0)$, donde hemos omitido las fórmulas con γ_1 ya que $\gamma_1 \notin \Gamma_r$.

Veamos finalmente cómo se construye el *tableau* canónico de una secuencia de fórmulas $\Phi = \{\varphi_1, \varphi_2, \dots\}$. Definimos la siguiente secuencia de árboles:

$$T_0 = \begin{array}{c} \varphi_0 \\ | \\ \theta_0 \end{array}$$

Luego definimos para $n \geq 1$:

$$\begin{aligned} T_n^1 &= \mathbf{ABSD}(T_{n-1}, n) \\ T_n^2 &= \mathbf{G}(T_n^1, n) \\ T_n &= \mathbf{EF}(T_n^2, n) \end{aligned}$$

Tenemos así una secuencia de árboles

$$T_0 \subseteq T_1 \subseteq \ldots T_k \subseteq T_{k+1} \subseteq \ldots$$

El *tableau* canónico de un conjunto de fórmulas Φ, que denotaremos por T_Φ, se construye como se indica a continuación. En la secuencia anterior se cumple una de las siguientes condiciones:

- Existe n tal que $T_n = T_{n+1}$ y T_n no tiene fórmulas a altura $> n$. Eso significa, por como están definidas las reglas, que $T_k = T_{k+1}$ para todo $k \geq n$. El *tableau* canónico es $T_\Phi = T_n$

- No se da el caso anterior. Entonces tenemos una secuencia infinita y el *tableau* canónico es el límite de todos los árboles

$$T_\Phi = \bigcup_{i \in \mathbb{N}} T_i$$

Proposición 2.84. *Sea S una signatura, $\Phi \subseteq FORM_S$, r una rama del tableau canónico T_Φ y $\varphi \in \Gamma_r$ está a profundidad k. Entonces ha sido introducida en un paso $l < k$.*

Demostración. Lo probamos por inducción. En el caso 0 está claro que se cumple. Supongamos que se cumple para $n-1$. Sea φ fórmula en r a profundidad k, y supongamos que la hemos introducido en el paso $l \geq k$. Pues bien, la fórmula ψ que está encima de φ, que está a altura $k-1$, ha tenido que ser introducida en la rama en un paso $\leq k-2$ por hipótesis de inducción. Esto significa que desde el *tableau* T_{l-2} a T_{l-1} (es decir, en el paso $l-1$) no se ha añadido ningún elemento a esa rama. El resto se reduce a comprobar regla a regla que si una rama tiene longitud $l-1$ y no hemos añadido ninguna fórmula al aplicar la regla a esa rama en el paso $l-1$, entonces tampoco se añade ninguna fórmula al aplicar esa regla a la rama en el paso l, lo cual es una contradicción. □

Teorema 2.85. *Sea S una signaturas y $\Phi \subseteq FORM_S$. El tableau canónico T_Φ es completo.*

Demostración. Sea r una rama abierta de T_Φ. Hemos de demostrar $\Phi \subseteq \Gamma_r$ y Γ_r es de Hintikka.
Lo primero es fácil, dada $\varphi \in \Phi$, existe un $k \in \mathbb{N}$ tal que $\varphi = \varphi_k$. Esa fórmula ha sido introducida en r en el árbol T_k al aplicar la regla $\mathbf{EF}(T_k^2, n)$, de modo que ya tenemos que $\Phi \subseteq \Gamma_r$.
Para ver que Γ_r es de Hintikka, comprobamos primero que no puede darse $\perp \in \Gamma_r$ o $\varphi, \neg\varphi \in \Gamma_r$ para ninguna φ, ya que en ese caso r sería una rama cerrada. Veamos que se cumplen el resto condiciones de la definición de Hintikka.

α-saturación. Tomemos $\alpha \in \Gamma_r$. Esa fórmula estará a una profundidad k del árbol. Esa fórmula ha sido expandida en el el arbol T_k. Por tanto $\alpha_1, \alpha_2 \in \Gamma_r$.

β-saturación, σ-saturación, δ-saturación. Similar al anterior.

γ-saturación. Sea $\gamma \in \Gamma_r$ y $t \in TERM_{\overline{S}}$ un término adecuado para Γ_r. Existe $n \in \mathbb{N}$ de forma que γ está en la rama r en todos los *tableaux* T_k con $k \geq n$.

Además existen $i \in \mathbb{N}$ tal que $\gamma = \gamma_i$ (en la numeración de las γ-fórmulas) y $j \in \mathbb{N}$ tal que $t = t_j$ (en la numeración de los términos). Llamamos l a la posición que ocupa el par (i, j) en nuestro buen orden de $\mathbb{N} \times \mathbb{N}$.

Por otro lado, como t es adecuado para Γ_r y $voc(t)$ y $var(t)$ son finitos, hay cierto m tal que $voc(t) \subseteq voc(\Gamma_{r_m})$ y $var(t) \subseteq lib(\Gamma_{r_m})$, es decir, t es adecuado para Γ_{r_m}, siendo r_m la rama del *tableau* T_m que da lugar a r.

Tras haber definido n, m y l, está claro que en el *tableau* T_{n+m} se cumple que t es adecuado para $\Gamma_{r_{n+m}}$ y $\gamma \in \Gamma_{r_{n+m}}$. De modo que, tras aplicar l veces más la regla **G**, es decir, en el *tableau* T_{n+m+l}, tenemos que haber añadido en cierto punto la fórmula $\gamma(t)$. Por tanto para todo término t adecuado para Γ_r, $\gamma(t)$ estará incluida en la rama, como queríamos.

$\square$

Corolario 2.86. *(Completitud) Sea S signatura y $\Phi \subseteq FORM_S$ tal que $Insat(\Phi)$. Entonces:*

a) Φ tiene un tableau cerrado finito.

b) Sea $\varphi \in FORM_S$ tal que $\Phi \models \varphi$. Entonces $\Phi \vdash_{tb} \varphi$.

Demostración. El segundo apartado es aplicación inmediata del primero al conjunto $\Psi = \Phi \cup \{\neg\varphi\}$. Así que demostraremos *b*). Construimos el *tableau* canónico T_Φ. Puesto que T_Φ es completo para Φ, si T_Φ tuviera una rama r abierta tendríamos

- $\Phi \subseteq \Gamma_r$
- Γ_r es de Hintikka y por tanto satisfactible.

Por tanto tendríamos que Φ es satisfactible. Por tanto T_Φ ha de ser cerrado. Por el lema 2.71, el tableau es finito. $\square$

Teorema 2.87. *(Teorema de compacidad) Sea S una signatura, $\Phi \subseteq FORM_S$.*

a) Si $Insat(\Phi)$, existe $\Phi_{fin} \subseteq \Phi$ finito tal que $Insat(\Phi_{fin})$.

b) Si todo conjunto finito $\Phi_{fin} \subseteq \Phi$ verifica $Sat(\Phi_{fin})$, entonces $Sat(\Phi)$.

Demostración. Puesto que Φ es insatisfactible tiene un *tableau* cerrado T. Al ser cerrado, es finito. Consideramos entonces el conjunto $\Phi_{fin} = \Phi \cap \Gamma_T$[10]. Veamos que verifica las propiedades buscadas:

- Es finito, al ser T finito.
- $\Phi_{fin} \subseteq \Phi$.
- Insat(Φ_{fin}) puesto que T también es un *tableau* para Φ_{fin}.

El segundo apartado es simplemente la afirmación contrarrecíproca del primero. □

2.11.6. Ejemplo de *tableau* canónico abierto

Veamos un ejemplo de *Tableau* canónico abierto. Este ejemplo no tiene funciones, ya que cuando hay funciones el *Tableau* resultante es infinito, al haber infinitos términos que sustituir en las reglas **G**. El tableaux siguiente corresponde al conjunto de fórmulas

$$\Phi = \{\forall x\, P(x) \vee Q(x), \neg(\forall x\, P(x) \vee \forall x\, Q(x))\} = \{\varphi_0, \varphi_1\}$$

Veremos que este tableau es satisfactible, de modo que al acabar de desarrollarlo tendremos ramas abiertas que nos permitirán encontrar modelos que satisfagan Φ.

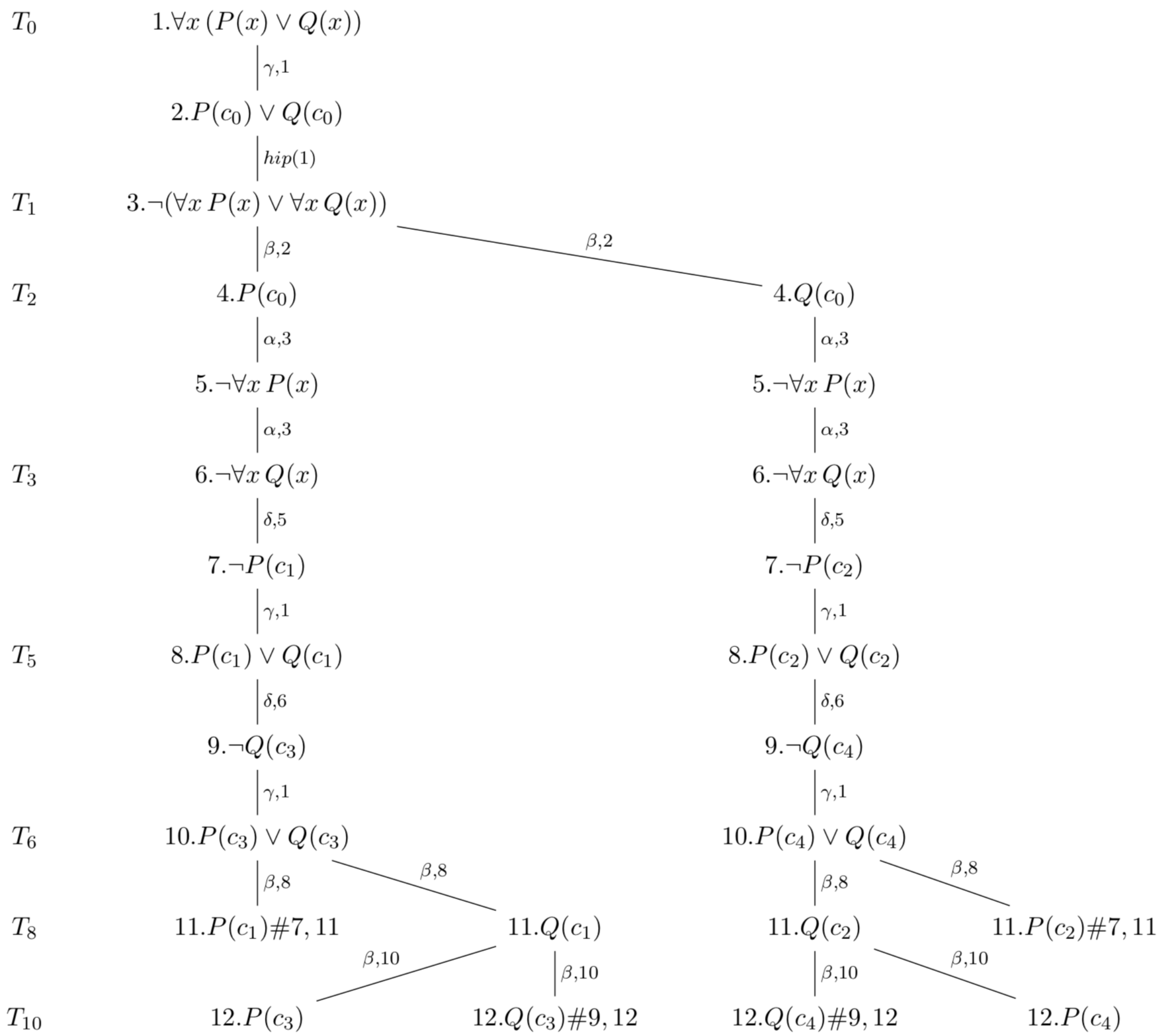
Para simplificar el *tableau*, al desarrollarlo se van a ignorar los axiomas de igualdad. Veremos que en este caso esto no va a ser un problema ya que vamos a poder encontrar los modelos igualmente.

También para simplificar, vamos a enumerar las fórmulas según su altura. Esto dará lugar a que haya varias fórmulas numeradas igual, pero no causará ambiguedades ya que en cada rama solo habrá una fórmula con cada número.

Otra anotación importante es que en la versión teórica del algoritmo, tenemos una numeración predefinida de los términos y de las γ-fórmulas. Sin embargo, en la práctica obviamente solo haremos *tableaux* finitos. Como da igual cambiar el orden de finitos términos, podemos establecer el orden de los finitos términos o γ-fórmulas que usemos según nos convenga.

Se muestra primero el *tableau* construido y a continuación explicamos su construcción paso a paso.

[10] Γ_T es el conjunto de fórmulas que aparecen en el *tableau* T.



Comenzamos con la construcción:

- T_0 Inicializamos el *tableau* con la primera fórmula de Φ, φ_0. En el algoritmo completo tendríamos que tener el primer axioma de igualdad, pero como dijimos en la introducción vamos a omitirlos para simplificar.

- T_1

  - **ABSD**$(T_0, 1)$ No hay ninguna fórmula α, β, σ o δ que desarrollar a altura 1, de modo que seguimos.
  - $\mathbf{G}(T_1^1)$ Tenemos una fórmula γ a altura ≤ 1: φ_0. Como vamos a ir numerando las γ-fórmulas sobre la marcha, esta será la fórmula γ_0. Ahora buscamos términos adecuados para Γ_r. Sin embargo, vemos

que no hay variables libres ni símbolos de constante en Γ_r, por tanto no se puede construir ningún término adecuado para Γ_r. Como no hay términos adecuados, usaremos la primera constante auxiliar, c_0. De modo que añadimos la fórmula $\gamma_0(c_0)$, que es $P(c_0) \vee Q(c_0)$.

- **EF**$(T_1^2, 1)$ Añadimos la segunda fórmula de Φ, φ_1. Como ya no quedan más fórmulas en Φ y no estamos usando axiomas de igualdad, no nos hará falta volver a usar la regla **EF** a partir de ahora.

Habiendo aplicado todas las reglas, ya tenemos el *tableau* T_1. Comenzamos la siguiente fase para extender el *tableau* a T_2.

- T_2
  - **ABSD**$(T_1, 2)$ Tenemos una β-fórmula en altura 2, $P(c_0) \vee Q(c_0)$. Por tanto, le aplicamos la regla β, dividiendo así el *tableau* en dos ramas, r_1, con la fórmula $P(c_0)$, y r_2, con la fórmula $Q(c_0)$.
  - **G**(T_2^1) Como ya tenemos una constante, ya tenemos algún término adecuado: c_0. Ahora, la única γ-fórmula a altura ≤ 2 es $\varphi_0 = \gamma_0$, pero ya hemos aplicado la regla a esa fórmula con el único término adecuado, c_0. Por tanto no añadimos nada en la regla **G**. Queda completado el *tableau* T_2.
- T_3
  - **ABSD**$(T_2, 1)$ En las ramas r_1 y r_2 tenemos en altura 3 la α-fórmula, $\neg(\forall x\, P(x) \vee \forall x\, Q(x))$. Por tanto aplicamos la α-fórmula en ambas ramas, añadiendo las fórmulas $\neg\forall x\, P(x)$ y $\neg\forall x\, Q(x)$.
  - **G**(T_3^1) Se da la misma situación que al construir T_2, no añadimos fórmulas en la regla **G**.
- T_4
  - **ABSD**$(T_3, 4)$ No hay fórmulas $\alpha, \beta, \gamma, \delta$ a altura 4 no extendidas.
  - **G**(T_4^1) Igual que antes no añadimos fórmulas en la regla **G**.

Como podemos observar, en el *tableau* T_4 no hemos añadido ninguna fórmula nueva. Eso no significa que hayamos acabado, ya que quedan fórmulas a profundidad > 4. Como tenemos fórmulas de altura hasta 6, tendríamos que llegar a T_6 sin ningún cambio para concluir que ya tenemos el *tableau* canónico. Pero veremos que ese no es el caso.

- T_5
  - **ABSD**$(T_4, 5)$ En r_1 y r_2 tenemos a altura 5 la δ-fórmula $\neg\forall x\, P(x)$. Por tanto, en r_1 añadimos la fórmula sustituyendo la primera constante auxiliar que no hemos usado, c_1. Es decir, prolongamos r_1 con la fórmula $\neg P(c_1)$. Ahora, en r_2, usamos la primera constante auxiliar que no habíamos usado hasta ahora, c_2, por tanto añadimos a la

rama la fórmula $\neg P(c_2)$.

- $\mathbf{G}(T_5^1)$ Seguimos teniendo una sola fórmula γ a altura ≤ 5, que es $\gamma_0 = \varphi_0$.

  Al extender la rama r_1, hemos aumentado su vocabulario con un símbolo de constante nuevo: c_1. Por tanto tenemos un nuevo término, c_1. Como $\gamma_0(c_1)$ no está en r_1, añadimos $\gamma_0(c_1) = P(c_1) \vee Q(c_1)$ a r_1.

  En r_2, análogamente a r_1, tenemos el nuevo término c_2, por tanto añadimos $\gamma_0(c_2) = P(c_2) \vee Q(c_2)$.

- T_6
  - $\mathbf{ABSD}(T_5, 6)$ Hay una δ-fórmula a altura 6 que extender en cada rama, $\neg\forall x\, Q(x)$. Razonando como en T_5, a la rama r_1 le añadimos la fórmula $\neg Q(c_3)$ y a r_2 le añadimos la fórmula $\neg Q(c_4)$.
  - $\mathbf{G}(T_6^1)$ Igual que hicimos en T_5, γ_0 sigue siendo nuestra única γ-fórmula, y tenemos los nuevos términos c_3 y c_4 en r_1 y r_2 respectivamente, por tanto añadimos a r_1 la fórmula $\gamma_0(c_3)$ y añadimos a r_2 $\gamma_0(c_4)$.
- T_7
  - $\mathbf{ABSD}(T_7, 8)$ Ninguna fórmula que extender a altura 7.
  - $\mathbf{G}(T_8^1)$ De nuevo, en ambas ramas hemos agotado el vocabulario y ya no nos quedan términos adecuados que aplicar a γ_0, por tanto no añadimos ninguna fórmula.
- T_8
  - $\mathbf{ABSD}(T_7, 8)$ En r_1, tenemos una β-fórmula a altura 8, $P(c_1) \vee Q(c_1)$. De modo que obtenemos dos nuevas ramas. La de la izquierda, que se forma añadiendo $P(c_1)$, es una rama cerrada ya que contiene las fórmulas $P(c_1)$ y $\neg P(c_1)$. La de la derecha se forma añadiendo $Q(c_1)$.

    En r_2, de igual forma, aplicamos la regla β obteniendo dos ramas, una de las cuales se forma añadiendo $Q(c_2)$. La otra se forma añadiendo $P(c_2)$ y queda cerrada al contener a $P(c_2)$ y $\neg P(c_2)$.
  - $\mathbf{G}(T_8^1)$Igual que en T_7, no añadimos nada.
- T_9 Al igual que en T_4 y T_7, ninguna de las reglas añade ninguna fórmula.
- T_{10}
  - $\mathbf{ABSD}(T_9, 10)$ En la rama de la izquierda, aplicamos la regla β a $P(c_3) \vee Q(c_3)$, obteniendo dos ramas. Una de ellas queda cerrada por contener a $Q(c_3)$ y a $\neg Q(c_3)$. En la rama de la derecha hay una situación similar.

- $\mathbf{G}(T_{10}^1)$ Ninguna nueva γ-fórmula ni vocabulario nuevo en ninguna rama.

- T_{11} Ninguna regla añade nada nuevo.
- T_{12} De nuevo, ninguna regla añade nada nuevo.

  Sin embargo, nuestro *tableaux* no tiene fórmulas a altura > 12. Como además se cumple que $T_{11} = T_{12}$, hemos acabado y ya tenemos nuestro *tableau* canónico.

Pues bien, ya tenemos el *tableau* canónico asociado a Φ. Nos quedan dos ramas abiertas. Llamamos al conjunto de fórmulas de la rama abierta de la izquierda Φ_1 y al conjunto de fórmulas de la rama abierta de la derecha Φ_2. Pues bien, como las ramas son abiertas en el *tableau* canónico, Φ_1 y Φ_2 son de Hintikka. Por tanto, conforme a lo visto en la subsección 2.11.4, podemos crear modelos que satisfacen Φ_1 y Φ_2. Comenzamos con Φ_1.

En Φ_1, los únicos símbolos de constante o variable que aparecen son c_0, c_1 y c_3. Como no hay símbolos de función, estos son nuestros únicos términos. Además, como no han aparecido igualdades en el *tableau*, el conjunto soporte coincidirá con el conjunto de términos. Por tanto tenemos nuestro conjunto soporte:

$$T = \{c_0, c_1, c_3\}$$

Ahora, en esta rama solo tenemos dos predicados, P y Q, ambos predicados 1-arios. Para ver cuáles son sus interpretaciones, recordemos que dado un término t, definimos $P(t) = V$ si y solo si $P(t)$ aparece en la rama. Por tanto, P será el conjunto de términos tales que $P(t)$ aparece en la rama. Las apariciones de P en la rama son $P(c_0)$ en altura 4 y $P(c_3)$ en altura 12, por tanto:

$$P = \{c_0, c_3\}$$

De igual forma, la única aparición de Q en la rama es $Q(c_1)$ en altura 11, por tanto:

$$Q = \{c_1\}$$

El caso de Φ_2 es similar al de Φ_1. El conjunto soporte coincide con el conjunto de términos, en este caso como solo aparecen c_0, c_2, c_4:

$$T = \{c_0, c_2, c_4\}$$

Y P solo aparece como $P(c_4)$, por tanto $P = \{c_4\}$. Q aparece como $Q(c_0)$ y $Q(c_2)$, por tanto $Q = \{c_0, c_2\}$.

3 | Deducción natural

3.1. Introducción

En los anteriores capítulos hemos desarrollado unos sistemas formales fundamentados en semántica y valores de verdad, es decir, a cada fórmula le asociamos un valor de verdad, V o F, que se puede obtener mediante interpretaciones. Sin embargo, si uno de los objetivos de la lógica es formalizar razonamientos, nos interesa estudiar también un sistema formal centrado en cómo deducir unas proposiciones de otras, independientemente de la noción de verdad asociada a ellas.

Por ejemplo, supongamos que queremos demostrar un teorema, que vendrá expresado por una fórmula ψ, y para deducirlo necesitamos unas fórmulas $\varphi_1, \varphi_2, \ldots, \varphi_n$. Lo que queremos es una serie de reglas que nos digan cuándo una fórmula se deduce de otras, de modo que partiendo de unos axiomas y aplicando repetidamente estas reglas, podamos llegar en finitos pasos a que ψ se deduce de $\varphi_1, \varphi_2, \ldots, \varphi_n$.

De modo que nuestra intención ahora consiste en prestar atención a las *reglas de inferencia*. Veremos a lo largo de este capítulo un sistema que solo conste de reglas de inferencia, así que comencemos por la siguiente

Definición 3.1. Una *secuencia* es un par $\Gamma \quad \psi$ donde Γ es un conjunto finito de fórmulas y ψ es una fórmula que se denominan, respectivamente, *conjunto de premisas* y *conclusión*. Normalmente, cuando Γ es finito, denotamos la secuencia $\{\varphi_1, \ldots, \varphi_n\} \quad \psi$ como $\varphi_1, \ldots, \varphi_n \quad \psi$ y escribiremos $\Gamma, \varphi \quad \psi$ en vez de $\Gamma \cup \{\varphi\} \quad \psi$.

Todas las reglas de nuestro sistema tendrán la forma:

$$\frac{\Gamma_1 \quad \varphi_1, \ldots, \Gamma_n \quad \varphi_n}{\Gamma \quad \varphi}$$

que intuitivamente significa: 'se demuestra $\Gamma \quad \varphi$ si se demuestra cada secuencia $\Gamma_1 \quad \varphi_1, \ldots, \Gamma_n \quad \varphi_n$'.

A lo largo de la exposición, veremos que existen reglas que no necesitan de premisas, que llamaremos *axiomas*. En la siguiente sección exponemos las reglas y axiomas *básicos*, y de ellos deduciremos después las reglas *derivadas*.

3.2. Reglas básicas

(**SP**) Axioma del supuesto: *si* $\varphi \in \Gamma$, *entonces* $\Gamma \quad \varphi$ *es un axioma.*

$$(\mathbf{SP}) \quad \frac{}{\Gamma \quad \varphi} \quad \varphi \in \Gamma$$

(**FS**) Fortalecimiento del supuesto: *si* $\Delta \quad \varphi$ *y* Γ *contiene a* Δ, *se deduce* $\Gamma \quad \varphi$.

$$(\mathbf{FS}) \quad \frac{\Delta \quad \varphi}{\Gamma \quad \varphi} \quad \Delta \subseteq \Gamma$$

(**RC**) Razonamiento en cadena: *podemos demostrar un resultado intermedio* ψ *y luego usarlo.*

$$(\mathbf{RC}) \ \frac{\Gamma \quad \psi \qquad \Gamma, \psi \quad \varphi}{\Gamma \quad \varphi}$$

($\wedge$**A**) Conjunción en el antecedente:

$$(\wedge\mathbf{A}) \quad \frac{\Gamma, \varphi, \psi \quad \chi}{\Gamma, \varphi \wedge \psi \quad \chi}$$

($\wedge$**C**) Conjunción en el consecuente.

$$(\wedge\mathbf{C}) \ \frac{\Gamma \quad \psi \qquad \Gamma \quad \varphi}{\Gamma \quad \psi \wedge \varphi}$$

($\rightarrow$ **A**) Implicación en el antecedente: *si tenemos una implicación en el antecedente, podemos intentar demostrar la premisa y entonces suponer el consecuente.*

$$(\rightarrow \mathbf{A}) \ \frac{\Gamma, \varphi \rightarrow \psi \quad \varphi \qquad \Gamma, \psi \quad \chi}{\Gamma, \varphi \rightarrow \psi \quad \chi}$$

($\rightarrow$ **C**) Implicación en el consecuente: *ponemos el antecedente de la implicación en las premisas.*

$$(\rightarrow \mathbf{C}) \ \frac{\Gamma, \varphi \quad \psi}{\Gamma \quad \varphi \rightarrow \psi}$$

($\vee$**A**) Disyunción en el antecedente: *si tenemos una distinción de casos hay que demostrar cada uno de los casos.*

$$(\vee\mathbf{A}) \ \frac{\Gamma, \varphi \quad \chi \qquad \Gamma, \psi \quad \chi}{\Gamma, \varphi \vee \psi \quad \chi}$$

($\vee$**C**$_{1,2}$) Disyunción en el consecuente: *basta demostrar uno de los elementos de la disyunción.* En realidad son dos reglas:

$$(\vee\mathbf{C}_1) \quad \frac{\Gamma \quad \varphi}{\Gamma \quad \varphi \vee \psi} \qquad\qquad (\vee\mathbf{C}_2) \quad \frac{\Gamma \quad \psi}{\Gamma \quad \varphi \vee \psi}$$

($\neg$**A**) Negación en el antecedente.

$$(\neg\mathbf{A}) \; \frac{\Gamma, \neg\varphi \quad \varphi}{\Gamma, \neg\varphi \quad \bot}$$

($\neg$**C**) Negación en el consecuente.

$$(\neg\mathbf{C}) \; \frac{\Gamma, \varphi \quad \bot}{\Gamma \quad \neg\varphi}$$

($\bot$**A**) Axioma de contradicción en el antecente: *si en el antecedente tenemos una contradicción podemos deducir cualquier cosa.*

$$(\bot\mathbf{A}) \; \frac{}{\Gamma, \bot \quad \varphi}$$

(**DN**!) Doble negación.

$$(\mathbf{DN}!) \; \frac{\Gamma \quad \neg\neg\psi}{\Gamma \quad \psi}$$

Esta regla se dice *no constructiva*. La lógica intuicionista, por ejemplo, no la acepta. Escribiremos signos '!' en toda regla derivada que use (**DN**!) en su demostración.

($\forall$**A**) Fórmula universal en el antecente: *la fórmula universal se puede aplicar a cualquier término.*

$$(\forall\mathbf{A}) \quad \frac{\Gamma, \forall x\varphi, \varphi[t/x] \quad \psi}{\Gamma, \forall x\varphi \quad \psi} \qquad t \in TERM_{\overline{S}}$$

($\forall$**C**$*$) Fórmula universal en el consecuente: *demostramos la fórmula para un elemento genérico.*

$$(\forall\mathbf{C}*) \quad \frac{\Gamma \quad \varphi[c/x]}{\Gamma \quad \forall x\varphi} \qquad c \in C_A \text{ nueva}$$

($\exists$**A**$*$) Fórmula existencial en el antecente: *sabemos que existe un elemento que cumple la propiedad, lo nombramos.*

$$(\exists \mathbf{A}*) \qquad \frac{\Gamma, \varphi[c/x] \quad \psi}{\Gamma, \exists x \varphi \quad \psi} \qquad c \in C_A \text{ nueva}$$

(**∃C**) Fórmula existencial en el consecuente: *basta encontrar un elemento que cumple la propiedad.*

$$(\exists \mathbf{C}) \qquad \frac{\Gamma \quad \varphi[t/x]}{\Gamma \quad \exists x \varphi} \qquad t \in TERM_{\overline{S}}$$

(**ID**) Axioma de identidad.

$$(\mathbf{ID}) \qquad \frac{}{\Gamma \quad t \doteq t} \qquad t \in TERM_{\overline{S}}$$

(**SUST**) Sustitución: *si quiero demostrar la propiedad para un término t y sabemos que es igual a s, basta demostrar la propiedad para s*

$$(\mathbf{SUST}) \qquad \frac{\Gamma \quad \varphi[s/x]}{\Gamma, t \doteq s \quad \varphi[t/x]} \qquad t, s \in TERM_{\overline{S}}$$

Las propiedades simétrica y transitiva pueden deducirse de esta regla.

3.3. Árboles de deducción

Una vez descritas las reglas básicas y dada una secuencia, para verificar si tal secuencia es o no correcta la descomponemos en las reglas básicas conocidas, es decir, revertimos la aplicación de las reglas básicas. La siguiente definición aclara estas nociones:

Definición 3.2. Un *árbol de deducción* para una secuencia $\Gamma \quad \psi$ es un árbol de forma que los nodos están etiquetados con secuencias, la raíz es $\Gamma \quad \psi$ y, dados un nodo $\Gamma_0 \quad \psi_0$ y sus hijos:

$$\begin{array}{ccc} & \Gamma_0 \quad \psi_0 & \\ \swarrow & & \searrow \\ \Gamma_1 \quad \psi_1 & \cdots & \Gamma_n \quad \psi_n \end{array}$$

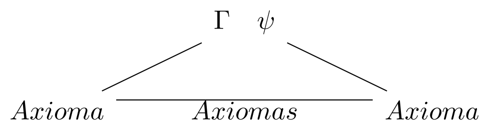
se cumple que

$$\frac{\Gamma_1 \quad \psi_1 \quad \cdots \quad \Gamma_n \quad \psi_n}{\Gamma_0 \quad \psi_0}$$

es una regla.

Definición 3.3.

1. Una secuencia $\Gamma \quad \psi$ se dice *formalmente deducible*, $\vdash \Gamma \quad \psi$, si existe un árbol de deducción para $\Gamma \quad \psi$ con axiomas en las hojas. Si $\Gamma = \emptyset$, en vez de $\Gamma \quad \psi$ escribimos $\vdash \psi$.

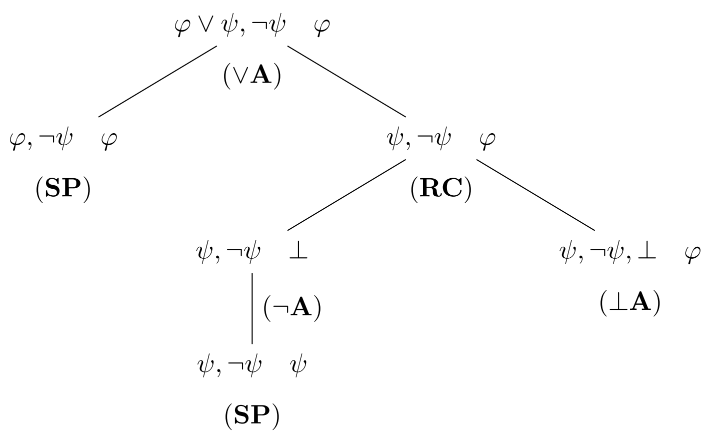


2. Si Φ es un conjunto de fórmulas (no necesariamente finito) diremos que una fórmula φ es *formalmente deducible a partir de* Φ, y denotado por $\Phi \vdash \varphi$, si existe un subconjunto finito $\Gamma \subset \Phi$ tal que $\vdash \Gamma \quad \varphi$

$\Phi \vdash \varphi$ y $\vdash \Phi \quad \varphi$ son equivalentes si Φ es finito. $\Phi \vdash \varphi$ permite que Φ sea infinito y con $\vdash \Phi \quad \varphi$ se asume la finitud de Φ.

Veamos un primer ejemplo detallado de árbol de deducción:

Ejemplo 3.4.



En este ejemplo, queremos llegar a que $\vdash \Gamma \quad \varphi$, siendo $\Gamma = \{\varphi \vee \psi, \neg\psi\}$. Como tenemos una disyunción en el antecedente, $\varphi \vee \psi$, podemos dividir por casos según la regla $(\vee\mathbf{A})$.
En una rama, tenemos $\varphi, \neg\psi \quad \varphi$. Como la conclusión es una premisa, esta secuencia es un axioma por $(\mathbf{SP})$.
En la otra rama, tenemos $\psi, \neg\psi \quad \varphi$. Está claro que $\psi, \neg\psi$ tiene que llevar a una contradicción, pero no hay ninguna regla que nos permita acabar directamente. En este caso podemos usar la regla $(\mathbf{RC})$, demostrando primero el resultado intermedio $\perp$ y luego demostrando φ usando $\perp$. Esto lleva a una nueva división en ramas.
En la primera rama, tenemos $\psi, \neg\psi \quad \perp$. Como tenemos la fórmula $\neg\psi$ en el antecedente y $\perp$ en la conclusión, aplicamos la regla $(\neg\mathbf{A})$. Así, llegamos a la

secuencia $\psi, \neg\psi \quad \psi$, que es un axioma (**SP**).
En la segunda rama, tenemos $\psi, \neg\psi, \bot \quad \varphi$. Esta regla es un axioma ($\bot$**A**), y hemos acabado.

Ejemplo 3.5.

$$\forall x\, p(x) \to q(x) \quad \big(\forall x\, p(x)\big) \to \big(\forall x\, q(x)\big)$$

$(\to \mathbf{C})$

$$\forall x\, p(x) \to q(x), \forall x\, p(x) \quad \forall x\, q(x)$$

$(\forall \mathbf{C}*)$

$$\forall x\, p(x) \to q(x), \forall x\, p(x) \quad q(c)$$

$(\forall \mathbf{A})$

$$\forall x\, p(x) \to q(x), \forall x\, p(x), p(c) \to q(c) \quad q(c)$$

$(\forall \mathbf{A})$

$$\forall x\, p(x) \to q(x), \forall x\, p(x), p(c) \to q(c), p(c) \quad q(c)$$

$(\mathbf{FS})$

$$p(c) \to q(c), p(c) \quad q(c)$$

$(\to \mathbf{A})$

$$p(c) \to q(c), p(c) \quad p(c) \qquad\qquad p(c), q(c) \quad q(c)$$

$(\mathbf{SP}) \qquad\qquad (\mathbf{SP})$

3.4. Reglas derivadas

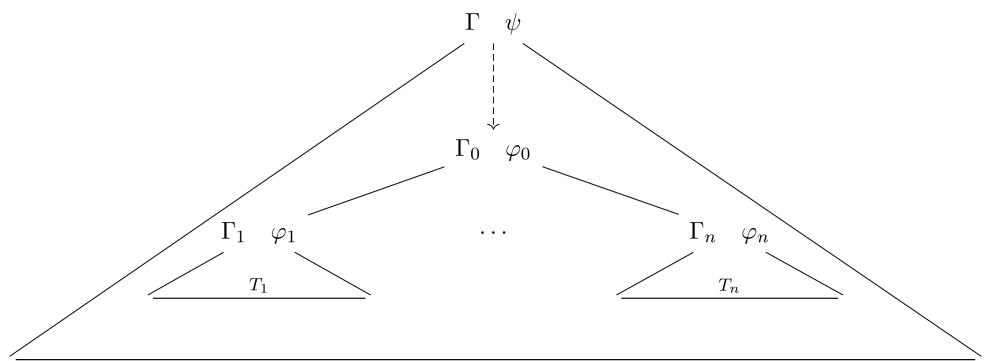
Al crear árboles de deducción, hay una serie de razonamientos que se aplican muy frecuentemente pero que no vienen expresados por ninguna regla básica. Por ejemplo, en el ejemplo 3.4 para deducir la secuencia $\psi, \neg\psi \quad \varphi$, que parece tan obvia como un axioma, hemos necesitados tres nodos extra. Podemos resumir razonamientos frecuentes de este tipo en reglas que no son básicas pero se pueden deducir de ellas:

Definición 3.6. Una regla

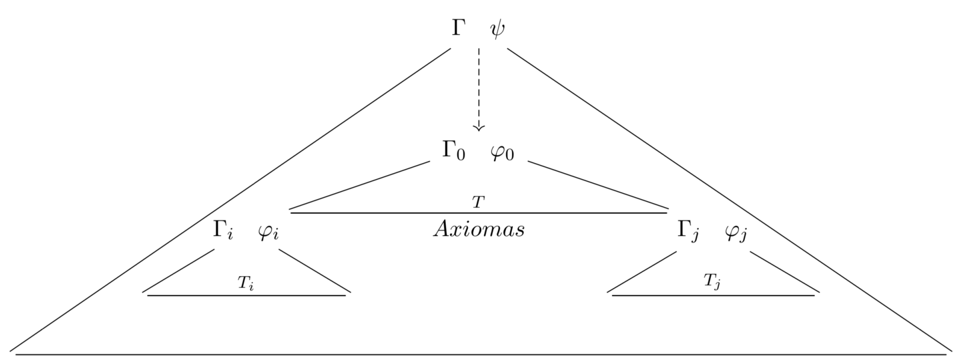
$$\frac{\Gamma_1 \quad \varphi_1 \quad \cdots \quad \Gamma_n \quad \varphi_n}{\Gamma \quad \varphi}$$

con $n \geq 0$ es derivable si existe un árbol de deducción para $\Gamma \quad \varphi$ cuyas hojas son axiomas o de la forma $\Gamma_i \quad \varphi_i, i = 1, \ldots, n$.

Ahora, si tenemos un árbol de deducción T para $\Gamma \quad \varphi$ que usa reglas básicas y derivadas, también existirá un árbol de deducción para $\Gamma \quad \varphi$ que solo usa reglas básicas. Informalmente, para obtener este nuevo árbol, basta con buscar en el árbol un nodo $\Gamma_0 \quad \varphi_0$ que se ha obtenido de sus hijos mediante una regla derivada. De modo que fijándonos en ese nodo y sus hijos, el árbol será algo así:



donde T_i es el árbol de deducción que se ha usado para llegar a $\Gamma_i \quad \varphi_i$. Ahora, simplemente, cambiamos el grafo formado por el nodo y sus n hijos por el árbol de deducción de la regla derivada. Cuando en una hoja del árbol de la regla derivada aparece una secuencia $\Gamma_i \quad \varphi_i$, añadimos debajo de ella el árbol de deducción T_i. De modo que esquemáticamente obtenemos un árbol así:



donde en el árbol T solo se usan reglas básicas y axiomas. Se puede demostrar que sustituyendo de este modo todas las reglas derivadas acabamos obteniendo un árbol en el que solo hay reglas básicas y axiomas.

De modo que por lo que acabamos de ver, si tenemos un árbol para $\Gamma \quad \varphi$ que usa reglas básicas y derivadas, se cumple que $\Gamma \quad \varphi$ es formalmente deducible. Esto quiere decir que podemos usar sin problemas reglas derivadas que conocemos en los árboles de deducción.

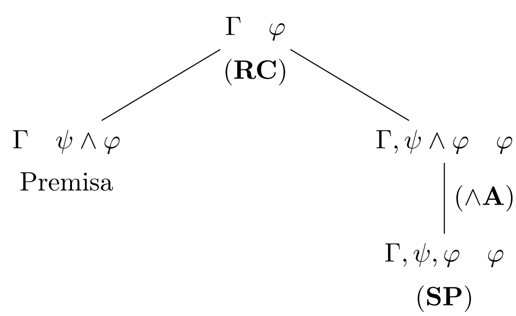
3.4.1. Repertorio de reglas derivadas

En esta sección expondremos muchas reglas derivadas que permiten ver cómo razonamientos que hacemos habitualmente se expresan en el cálculo de secuencias. Para casi todas las reglas derivadas incluímos sus árboles de deducción.

- Eliminación de una conjunción: *siempre podemos demostrar un resultado más fuerte*. Son dos reglas simétricas.

$$(\mathbf{E}\wedge) \qquad \frac{\Gamma \quad \psi \wedge \varphi}{\Gamma \quad \varphi} \qquad \frac{\Gamma \quad \varphi \wedge \psi}{\Gamma \quad \varphi}$$

Demostración.



□

- Modus Ponens

$$(\mathbf{MP}) \qquad \frac{\Gamma \quad \varphi \qquad \Gamma \quad \varphi \to \psi}{\Gamma \quad \psi}$$

Demostración.

$$\Gamma \quad \psi$$
(**RC**)

- $\Gamma \quad \varphi$ — Premisa
- $\Gamma, \varphi \quad \psi$ (**RC**)
  - $\Gamma, \varphi, \varphi \to \psi \quad \psi$ ($\to$ **A**)
    - $\Gamma, \varphi, \varphi \to \psi \quad \varphi$ — SP
    - $\Gamma, \psi \quad \psi$ — SP
  - $\Gamma, \varphi \quad \varphi \to \psi$ (**FS**)
    - $\Gamma \quad \varphi \to \psi$ — Premisa

□

- Variante de la conjunción en el consecuente.

$$(\wedge \mathbf{C}') \qquad \frac{}{\Gamma, \varphi, \psi \quad \varphi \wedge \psi}$$

Demostración.

$$\Gamma, \varphi, \psi \quad \varphi \wedge \psi$$
($\wedge$**C**)

- $\Gamma, \varphi, \psi \quad \varphi$ (**SP**)
- $\Gamma, \varphi, \psi \quad \psi$ (**SP**)

□

- Variante de la disyunción en el consecuente.

$$(\vee \mathbf{C}') \qquad \frac{}{\Gamma, \varphi \quad \varphi \vee \psi} \qquad \frac{}{\Gamma, \psi \quad \varphi \vee \psi}$$

Demostración.

$$\Gamma, \varphi \quad \varphi \vee \psi$$
($\vee$**C**)

- $\Gamma, \varphi \quad \varphi$ (**SP**)

□

- Reducción al absurdo: *suponer la negación de la conclusión y llegar a una contradicción.*

$$(\mathbf{RA}!) \qquad \frac{\Gamma, \neg\varphi \quad \bot}{\Gamma \quad \varphi}$$

Demostración.

$\Gamma \quad \varphi$

$(\mathbf{DN}!)$

$\Gamma \quad \neg\neg\varphi$

$(\neg\mathbf{C})$

$\Gamma, \neg\varphi \quad \bot$

Premisa

□

- Reglas de la contradicción.

$$(\mathbf{CT}_1) \qquad \frac{\Gamma \quad \psi \qquad \Gamma \quad \neg\psi}{\Gamma \quad \varphi}$$

Demostración.

$\Gamma \quad \varphi$

$(\mathbf{RC})$

$\Gamma, \bot \quad \varphi$ $\qquad$ $\Gamma \quad \bot$

$(\bot\ \mathbf{A})$ $\qquad$ $(\mathbf{RC})$

$\Gamma, \neg\psi \quad \bot$ $\qquad$ $\Gamma \quad \neg\psi$

$(\neg\mathbf{A})$ $\qquad$ Premisa

$\Gamma, \neg\psi \quad \psi$

$(\mathbf{FS})$

$\Gamma \quad \psi$

Premisa

□

$$(\mathbf{CT}_2) \qquad \frac{}{\Gamma, \varphi, \neg\varphi \quad \psi}$$

Demostración.

$\Gamma, \varphi, \neg\varphi \quad \psi$

$(\mathbf{CT}_1)$

$\Gamma, \varphi, \neg\varphi, \quad \neg\varphi$ $\qquad$ $\Gamma, \varphi, \neg\varphi, \quad \varphi$

$(\mathbf{SP})$ $\qquad$ $(\mathbf{SP})$

□

- Reglas de la doble negación.

$$(\neg\neg\mathbf{C}) \quad \frac{\Gamma \quad \varphi}{\Gamma \quad \neg\neg\varphi} \qquad (\neg\neg\mathbf{A}!) \quad \frac{\Gamma, \varphi \quad \psi}{\Gamma, \neg\neg\varphi \quad \psi}$$

Demostración.

$\Gamma \quad \neg\neg\varphi$

$(\mathbf{RC})$

$\Gamma \quad \varphi$ $\qquad$ $\Gamma, \varphi \quad \neg\neg\varphi$

Premisa $\qquad$ $(\neg\mathbf{C})$

$\Gamma, \varphi, \neg\varphi \quad \bot$

$(\mathbf{CT}_2)$

$\Gamma, \neg\neg\varphi \quad \psi$

$(\mathbf{RC})$

$\Gamma, \neg\neg\varphi \quad \varphi$ $\qquad$ $\Gamma, \neg\neg\varphi, \varphi \quad \psi$

$(\mathbf{DN}!)$ $\qquad$ $(\mathbf{FS})$

$\Gamma, \neg\neg\varphi \quad \neg\neg\varphi$ $\qquad$ $\Gamma, \varphi \quad \psi$

$(\mathbf{SP})$ $\qquad$ Premisa

□

- Reglas de la contraposición.

$$(\mathbf{CP}_1) \quad \frac{\Gamma, \psi \quad \varphi}{\Gamma, \neg\varphi \quad \neg\psi} \qquad (\mathbf{CP}_2) \quad \frac{\Gamma, \varphi \quad \neg\psi}{\Gamma, \psi \quad \neg\varphi}$$

Demostración.

$$\begin{array}{c}
\Gamma, \neg\varphi \quad \neg\psi \\
\Big| \; (\neg\mathbf{C}) \\
\Gamma, \neg\varphi, \psi \quad \bot \\
(\mathbf{RC}) \\
\begin{array}{cc}
\Gamma, \neg\varphi, \psi \quad \varphi & \Gamma, \neg\varphi, \psi, \varphi \quad \bot \\
\Big| \; (\mathbf{FS}) & (\mathbf{CT}_2) \\
\Gamma, \psi \quad \varphi & \\
\text{Premisa} &
\end{array}
\end{array}$$

□

$$(\mathbf{CP}_3!) \quad \frac{\Gamma, \neg\varphi \quad \neg\psi}{\Gamma, \psi \quad \varphi} \qquad (\mathbf{CP}_4!) \quad \frac{\Gamma, \neg\varphi \quad \psi}{\Gamma, \neg\psi \quad \varphi}$$

Demostración.

$$\begin{array}{c}
\Gamma, \psi \quad \varphi \\
\Big| \; (\mathbf{DN}!) \\
\Gamma, \psi \quad \neg\neg\varphi \\
\Big| \; (\mathbf{CP}_2) \\
\Gamma, \neg\varphi \quad \neg\psi \\
\text{Premisa}
\end{array}$$

□

- Otras reglas no constructivas.

$$(\mathbf{VC}!) \quad \frac{\Gamma, \neg\varphi \quad \psi}{\Gamma \quad \varphi \vee \psi} \qquad \frac{\Gamma, \neg\psi \quad \varphi}{\Gamma \quad \varphi \vee \psi}$$

Demostración.

$$\Gamma \quad \varphi \vee \psi$$

$$\Big| \; (\mathbf{RA}!)$$

$$\Gamma, \neg(\varphi \vee \psi) \quad \bot$$

$$\Big| \; (\neg\mathbf{A})$$

$$\Gamma, \neg(\varphi \vee \psi) \quad \varphi \vee \psi$$

$$\Big| \; (\vee\mathbf{C})$$

$$\Gamma, \neg(\varphi \vee \psi) \quad \varphi$$

$$\Big| \; (\mathbf{CP}_4)$$

$$\Gamma, \neg\varphi \quad \varphi \vee \psi$$

$$\Big| \; (\vee\mathbf{C})$$

$$\Gamma, \neg\varphi \quad \psi$$

Premisa

$\square$

(**TE**!) Tercio excluido $\dfrac{}{\Gamma \quad \varphi \vee \neg\varphi}$

Demostración.

$$\Gamma \quad \varphi \vee \neg\varphi$$

$$\Big| \; (\vee\mathbf{C}!)$$

$$\Gamma, \neg\varphi \quad \neg\varphi$$

(**SP**)

$\square$

(**DC**!) Distinción de casos $\dfrac{\Gamma, \varphi \quad \psi \qquad \Gamma, \neg\varphi \quad \psi}{\Gamma \quad \psi}$

Demostración.

$$\dfrac{\underset{(\mathbf{TE}!)}{\Gamma \quad \varphi \vee \neg\varphi} \qquad \dfrac{\underset{\text{Premisa}}{\Gamma, \varphi \quad \psi} \qquad \underset{\text{Premisa}}{\Gamma, \neg\varphi \quad \psi}}{\Gamma, \varphi \vee \neg\varphi \quad \psi}(\vee\mathbf{A}!)}{\Gamma \quad \psi}(\mathbf{RC})$$

□

$$(\to \mathbf{A}!) \qquad \dfrac{\Gamma, \neg\varphi \quad \chi \qquad \Gamma, \psi \quad \chi}{\Gamma, \varphi \to \psi \quad \chi}$$

Demostración.

$$\dfrac{\dfrac{\underset{\text{Premisa}}{\Gamma, \neg\varphi \quad \chi}}{\Gamma, \neg\varphi, \varphi \to \psi \quad \chi}(\mathbf{FS}) \qquad \dfrac{\underset{(\mathbf{SP})}{\Gamma, \varphi, \varphi \to \psi \quad \varphi} \qquad \dfrac{\underset{\text{Premisa}}{\Gamma, \psi \quad \chi}}{\Gamma, \varphi, \psi \quad \chi}(\mathbf{FS})}{\Gamma, \varphi, \varphi \to \psi \quad \chi}(\to \mathbf{A})}{\Gamma, \varphi \to \psi \quad \chi}(\mathbf{DC}!)$$

□

$$(\neg \to \mathbf{A}!) \qquad \dfrac{\Gamma, \varphi, \neg\psi \quad \chi}{\Gamma, \neg(\varphi \to \psi) \quad \chi}$$

Demostración.

$$\Gamma, \neg(\varphi \to \psi) \quad \chi$$

$$\Big| \; (\mathbf{CP}_4!)$$

$$\Gamma, \neg\chi \quad \varphi \to \psi$$

$$\Big| \; (\to \mathbf{C})$$

$$\Gamma, \neg\chi, \varphi \quad \psi$$

$$\Big| \; (\mathbf{CP}_4!)$$

$$\Gamma, \varphi, \neg\psi \quad \chi$$

Premisa

□

- Leyes de De Morgan.

$$(\mathbf{DM}_1) \qquad \frac{}{\Gamma, \neg\varphi \vee \neg\psi \quad \neg(\varphi \wedge \psi)}$$

Demostración.

$$\Gamma, \neg\varphi \vee \neg\psi \quad \neg(\varphi \wedge \psi)$$

$$\Big| \; (\mathbf{CP}_2)$$

$$\Gamma, \varphi \wedge \psi \quad \neg(\neg\varphi \vee \neg\psi)$$

$$\Big| \; (\wedge\mathbf{A})$$

$$\Gamma, \varphi, \psi \quad \neg(\neg\varphi \vee \neg\psi)$$

$$\Big| \; (\neg\mathbf{C})$$

$$\Gamma, \varphi, \psi, \neg\varphi \vee \neg\varphi \quad \bot$$

$$(\vee\mathbf{A})$$

$$\Gamma, \varphi, \psi, \neg\varphi \quad \bot \qquad\qquad \Gamma, \varphi, \psi, \neg\psi \quad \bot$$

$$(\mathbf{CT}_2) \qquad\qquad (\mathbf{CT}_2)$$

□

$$(\mathbf{DM}_2) \qquad \frac{}{\Gamma, \neg(\varphi \wedge \psi) \quad \neg\varphi \vee \neg\psi}$$

Demostración.

$$
\begin{array}{c}
\Gamma, \neg(\varphi \wedge \psi) \quad \neg\varphi \vee \neg\psi \\
\Big| \; (\mathbf{CP}_4) \\
\Gamma, \neg(\neg\varphi \vee \neg\psi) \quad \varphi \wedge \psi \\
(\wedge\mathbf{C})
\end{array}
$$

$$
\begin{array}{cc}
\Gamma, \neg(\neg\varphi \vee \neg\psi) \quad \varphi & \Gamma, \neg(\neg\varphi \vee \neg\psi) \quad \psi \\
\Big| \; (\mathbf{CP}_4!) & \Big| \; (\mathbf{CP}_4!) \\
\Gamma, \neg\varphi \quad \neg\varphi \vee \neg\psi & \Gamma, \neg\psi \quad \neg\varphi \vee \neg\psi \\
\Big| \; (\vee\mathbf{C}) & \Big| \; (\vee\mathbf{C}) \\
\Gamma, \neg\varphi \quad \neg\varphi & \Gamma, \neg\varphi \quad \neg\psi \\
(\mathbf{SP}) & (\mathbf{SP})
\end{array}
$$

$\square$

$$(\mathbf{DM}_3) \qquad \frac{}{\Gamma, \neg\varphi \wedge \neg\psi \quad \neg(\varphi \vee \psi)}$$

$$(\mathbf{DM}_4) \qquad \frac{}{\Gamma, \neg(\varphi \vee \psi) \quad \neg\varphi \wedge \neg\psi}$$

- Ley de Pierce

$$\frac{}{\Gamma \quad ((\varphi \to \psi) \to \varphi) \to \varphi}$$

- Reglas de ejemplificación. ($t \in TERM_{\overline{S}}$)

$$(\mathbf{E}\forall_1) \quad \frac{}{\Gamma, \forall x \ \varphi \quad \varphi[t/x]} \qquad (\mathbf{E}\forall_2) \quad \frac{\Gamma \quad \forall x \ \varphi}{\Gamma \quad \varphi[t/x]}$$

$$(\mathbf{E}\forall_3) \quad \frac{\Gamma, \varphi[t/x] \quad \psi}{\Gamma, \forall x \ \varphi \quad \psi}$$

Demostración.

$$
\begin{array}{c}
\Gamma, \forall x\varphi \quad \varphi[t/x] \\
\Big| \; (\forall\mathbf{A}) \\
\Gamma, \varphi, \varphi[t/x] \quad \varphi[t/x] \\
(\mathbf{SP})
\end{array}
$$

$$\Gamma \quad \varphi[t/x]$$
$(\mathbf{RC})$

$$\Gamma\forall x\varphi \qquad\qquad \Gamma, \forall x\varphi \quad \varphi[t/x]$$
Premisa $\qquad\qquad$ $(\mathbf{E}\forall_1)$

$$\Gamma, \forall x\varphi \quad \psi$$
$(\mathbf{RC})$

$$\Gamma, \forall x\varphi \quad \varphi[t/x] \qquad\qquad \Gamma, \forall x\varphi, \varphi[t/x] \quad \psi$$
$(\mathbf{E}\forall_1)$ $\qquad\qquad$ $(\mathbf{RS})$

$$\Gamma, \varphi[t/x] \quad \psi$$
Premisa

□

$$(\mathbf{E}\exists_1) \quad \frac{}{\Gamma, \varphi[t/x] \quad \exists x\ \varphi} \qquad\qquad (\mathbf{E}\exists_2) \quad \frac{\Gamma, \exists x\ \varphi \quad \psi}{\Gamma, \varphi[t/x] \quad \psi}$$

Demostración.

$$\Gamma, \varphi[t/x] \quad \exists x\varphi$$
$(\exists\mathbf{C})$

$$\Gamma, \varphi[t/x] \quad \varphi[t/x]$$
$(\mathbf{SP})$

$$\Gamma, \varphi[t/x] \quad \psi$$
$(\mathbf{RC})$

$$\Gamma, \varphi[t/x]\exists x\varphi \quad \psi \qquad\qquad \Gamma, \varphi[t/x] \quad \exists x\varphi$$
$(\mathbf{FS})$ $\qquad\qquad$ $(\mathbf{E}\exists_1)$

$$\Gamma, \exists x\varphi \quad \psi$$
Premisa

□

- Leyes de dualidad.

$$(\mathbf{SDM}_1) \quad \frac{}{\Gamma, \exists x \neg\varphi \quad \neg\forall x\ \varphi} \qquad (\mathbf{SDM}_2) \quad \frac{}{\Gamma, \neg\forall x\ \varphi \quad \exists x \neg\varphi}$$

$$(\mathbf{SDM}_3) \quad \frac{}{\Gamma, \forall x \neg\varphi \quad \neg\exists x\ \varphi} \qquad (\mathbf{SDM}_4) \quad \frac{}{\Gamma, \neg\exists x\ \varphi \quad \forall x \neg\varphi}$$

Demostración. Se demuestran la segunda y la cuarta.

$$\begin{array}{c}
\Gamma, \neg\forall x\ \varphi \quad \exists x \neg\varphi \\
\Big| (\mathbf{CP}_4!) \\
\Gamma, \neg\exists x \neg\varphi \quad \forall x \varphi \\
\Big| (\forall \mathbf{C^*}) \\
\Gamma, \neg\exists x \neg\varphi \quad \varphi[c/x] \\
\Big| (\mathbf{CP}_4!) \\
\Gamma, \neg\varphi[c/x] \quad \exists x \neg\varphi \\
(\mathbf{E}\exists_1)
\end{array}$$

$$\begin{array}{c}
\Gamma, \neg\exists x \varphi \quad \forall x \neg\varphi \\
\Big| (\forall \mathbf{C^*}) \\
\Gamma, \neg\exists x \varphi \quad \neg\varphi[c/x] \\
\Big| (\mathbf{CP}_1!) \\
\Gamma, \neg\varphi[c/x] \quad \exists x \varphi \\
(\mathbf{E}\exists_1)
\end{array}$$

□

- Sustitución generalizada

$$(\mathbf{SUST}_n) \quad \frac{\Gamma \quad \varphi[t_1/x_1, ..., t_n/x_n]}{\Gamma, t_1 \doteq s_1, ..., t_n \doteq s_n \quad \varphi[s_1/x_1, ..., s_n/x_n]}$$

Demostración. Por inducción sobre n.
(POR COMPLETAR) □

3.5. Ejemplo de uso de deducción natural

Como ejemplo, intentaremos demostrar que se cumple la siguiente frase:

No existe el barbero que sea el barbero que afeite a los barberos que no se afeitan a sí mismos.

Para formalizar esto, usaremos los predicados $p|_1$ y $a|_2$, donde $p(x)$ significa 'x es barbero' y $a(x, y)$ significa 'x afeita a y'.

Entonces, la conclusión a la que queremos llegar es:

$$\neg\exists x b(x) \land (\forall y b(y) \to (a(x, y) \leftrightarrow \neg a(y, y)))$$

Y no tenemos ninguna premisa. Pasamos directamente al árbol de deducción:

$$\neg\exists x b(x) \land (\forall y b(y) \rightarrow (a(x,y) \leftrightarrow \neg a(y,y)))$$

$$(\neg\mathbf{C})$$

$$\exists x b(x) \land (\forall y b(y) \rightarrow (a(x,y) \leftrightarrow \neg a(y,y))) \quad \perp$$

$$(\exists\mathbf{A})$$

$$b(c) \land (\forall y b(y) \rightarrow (a(c,y) \leftrightarrow \neg a(y,y))) \quad \perp$$

$$(\land\mathbf{A})$$

$$b(c), \forall y b(y) \rightarrow (a(c,y) \leftrightarrow \neg a(y,y)) \quad \perp$$

$$(\forall\mathbf{A})$$

$$b(c), \forall y b(y) \rightarrow (a(c,y) \leftrightarrow \neg a(y,y)), b(c) \rightarrow (a(c,c) \leftrightarrow \neg a(c,c)) \quad \perp$$

$$(\mathbf{FS})$$

$$b(c), b(c) \rightarrow (a(c,c) \leftrightarrow \neg a(c,c)) \quad \perp$$

$$(\rightarrow\mathbf{A})$$

$$b(c), b(c) \rightarrow (a(c,c) \leftrightarrow \neg a(c,c)) \quad b(c) \qquad b(c), a(c,c) \leftrightarrow \neg a(c,c \quad \perp$$

$$(\mathbf{SP})$$

$$(\land\mathbf{A})$$

$$b(c), \neg a(c,c) \rightarrow a(c,c), a(c,c) \rightarrow \neg a(c,c) \quad \perp$$

$$(\mathbf{FS})$$

$$\neg a(c,c) \rightarrow a(c,c), a(c,c) \rightarrow \neg a(c,c) \quad \perp$$

$$(\rightarrow\mathbf{A}!)$$

$$\neg a(c,c) \rightarrow a(c,c), \neg a(c,c) \quad \perp \qquad \neg a(c,c) \rightarrow a(c,c), \neg a(c,c) \quad \perp$$

$$(\rightarrow\mathbf{A}!) \qquad (\rightarrow\mathbf{A}!)$$

$$\neg\neg a(c,c), \neg a(c,c) \quad \perp \qquad \neg a(c,c), a(c,c) \quad \perp \qquad \neg\neg a(c,c), \neg a(c,c) \quad \perp \qquad \neg a(c,c), a(c,c) \quad \perp$$

$$(\mathbf{CT}_2) \qquad (\mathbf{CT}_2) \qquad (\mathbf{CT}_2) \qquad (\mathbf{CT}_2)$$

Como no hemos definido reglas para dobles implicaciones, hemos tratado la doble implicación como si fuera la conjunción de las implicaciones en los dos sentidos.

3.6. Corrección

Ya hemos definido anteriormente lo que significa que φ sea formalmente deducible a partir de Φ, $\Phi \vdash \varphi$. Vamos a comparar ahora esto con la semántica que hemos visto en el tema anterior.

Para que la deducción natural sea compatible con la semántica, nos interesa ver que Φ implica φ si y solo si φ es formalmente deducible a partir de Φ. Expresándolo formalmente, queremos ver que $\Gamma \models \varphi$ si y solo si $\Gamma \vdash \varphi$. Esto se divide en dos implicaciones:

- Corrección: $\Phi \vdash \varphi$ implica $\Phi \models \varphi$.
- Completitud: $\Phi \models \varphi$ implica $\Phi \vdash \varphi$.

En esta sección se demostrará la corrección de la deducción natural, y la siguiente sección se dedicará a la completitud.

Definición 3.7. Diremos que una secuencia $\Gamma \quad \varphi$ es *correcta* si se cumple $\Gamma \models \varphi$. Es decir, informalmente, si 'las premisas implican la conclusión'.

Definición 3.8. Una regla

$$\frac{\Gamma_1 \quad \varphi_1, \ldots, \Gamma_n \quad \varphi_n}{\Gamma \quad \varphi}$$

se dice *correcta* si la secuencia $\Gamma \quad \varphi$ es correcta siempre que lo sean $\Gamma_1 \quad \varphi_1, \ldots, \Gamma_n \quad \varphi_n$. Es decir, si se cumplen $\Gamma_1 \models \varphi_1, \ldots, \Gamma_n \models \varphi_n$, entonces se cumple $\Gamma \models \varphi$.

Conviene hacer notar que un axioma

$$\frac{}{\Gamma \quad \varphi}$$

es correcto si $\Gamma \models \varphi$.

Proposición 3.9. *Todas las reglas y axiomas básicos son correctos.*

Demostración. Se comprueba directamente. Haremos algunas demostraciones a modo de ejemplo.

$$(\mathbf{RC})\ \frac{\Gamma \quad \psi \qquad \Gamma, \psi \quad \varphi}{\Gamma \quad \varphi}$$

Supongamos que $\Gamma \models \psi$, $\Gamma, \psi \models \varphi$. Sea una interpretación $\mathfrak{I}$ tal que $\mathfrak{I} \models \Gamma$ (si no existe tal interpretación, el enunciado es trivial). Al ser $\Gamma \models \psi$, $\mathfrak{I} \models \psi$ y por tanto $\mathfrak{I} \models \Gamma \cup \{\psi\}$, y como $\Gamma, \psi \models \varphi$, $\mathfrak{I} \models \varphi$. Como esto lo cumple cualquier modelo $\mathfrak{I}$, tenemos que $\Gamma \models \varphi$.

$$(\to \mathbf{A})\ \frac{\Gamma, \varphi \to \psi \quad \varphi \qquad \Gamma, \psi \quad \chi}{\Gamma, \varphi \to \psi \quad \chi}$$

Supongamos que $\Gamma, \varphi \to \psi \vDash \varphi$, $\Gamma, \psi \vDash \chi$. Sea $\mathfrak{I}$ tal que $\mathfrak{I} \vDash \Gamma, \varphi \to \psi$. Por tanto, $\mathfrak{I} \vDash \varphi$ y $\mathfrak{I} \vDash \varphi \to \psi$, es decir, $\mathfrak{I} \vDash \psi$. Por otro lado, como también $\mathfrak{I} \vDash \Gamma$, entonces $\mathfrak{I} \vDash \Gamma \cup \{\psi\}$, con lo que $\mathfrak{I} \vDash \chi$. Al ser $\mathfrak{I}$ arbitrario, $\Gamma \cup \{\varphi \to \psi\} \vDash \chi$.

$$(\forall \mathbf{A}) \quad \frac{\Gamma, \forall x\varphi, \varphi[t/x] \quad \psi}{\Gamma, \forall x\varphi \quad \psi} \quad t \in TERM_{\overline{S}}$$

Sea $\mathfrak{I} \vDash \Gamma \cup \{\forall x\, \varphi\}$. Entonces, para todo $a \in A$ (con A conjunto soporte de $\mathfrak{I}$), $\mathfrak{I}[a/x] \vDash \varphi$ y, por ser $t^{\mathfrak{I}} \in A$, tenemos que $\mathfrak{I}[t^{\mathfrak{I}}/x] \vDash \varphi$. Por el Lema de Sustitución, $\mathfrak{I} \vDash \varphi[t/x]$. Por tanto, $\mathfrak{I} \vDash \Gamma \cup \{\forall x\, \varphi, \varphi[t/x]\}$ y, como suponemos que $\Gamma, \forall x\, \varphi, \varphi[t/x] \vDash \psi$, obtenemos que $\mathfrak{I} \vDash \psi$, lo que nos da el resultado.

$$(\forall \mathbf{C}*) \quad \frac{\Gamma \quad \varphi[c/x]}{\Gamma \quad \forall x\varphi} \quad c \text{ variable nueva}$$

Sea $\mathfrak{I}$ interpretación tal que $\mathfrak{I} \vDash \Gamma$. Como suponemos $\Gamma \quad \varphi[c/x]$ correcta, tenemos $\mathfrak{I} \vDash \varphi[c/x]$. Ahora queremos demostrar $\mathfrak{I} \vDash \forall x\varphi$. Es decir, dada $a \in A$ cualquiera, queremos demostrar $\mathfrak{I}[a/x] \vDash \varphi$.

Sea $\mathfrak{I}' = \mathfrak{I}[a/c]$, es decir, igual que $\mathfrak{I}$ salvo que $c^{\mathfrak{I}'} = a$. Como c no está en ninguna fórmula de Γ y $\mathfrak{I} \vDash \varphi \quad \forall \varphi \in \Gamma$, por el lema de coincidencia 2.41 se cumplirá que $\mathfrak{I}' \vDash \varphi \ \ \forall \varphi \in \Gamma$.

Por tanto, como estamos suponiendo que $\Gamma \vDash \varphi[c/x]$, tenemos que $\mathfrak{I}' \vDash \varphi[c/x]$. Equivalentemente, $\mathfrak{I}'[c^{\mathfrak{I}'}/x] \vDash \varphi$, es decir, $\mathfrak{I}'[a/x] \vDash \varphi$.
De modo que, como $c \notin voc(\varphi)$ y por tanto $\mathfrak{I} \sim_{\varphi} \mathfrak{I}'$, por el lema de coincidencia también se cumple que $\mathfrak{I}[a/x] \vDash \varphi$, como queríamos. □

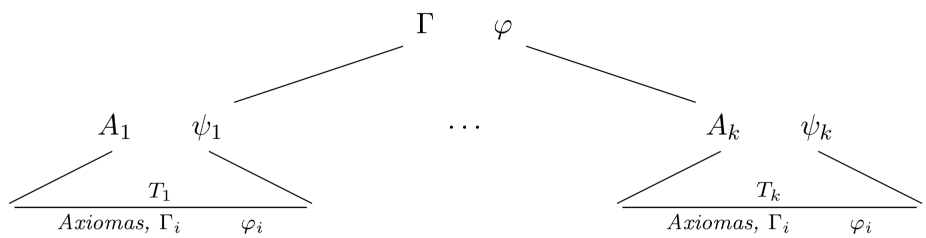
Ahora que tenemos que las reglas básicas son correctas, el último paso que nos queda para demostrar la corrección es demostrar que las reglas derivables también son correctas.

Proposición 3.10. *Sea* $(\mathbf{R})\ \frac{\Gamma_1 \quad \varphi_1 \ldots \Gamma_n \quad \varphi_n}{\Gamma \quad \Phi}$ *una regla derivada usando reglas correctas. Entonces* $(\mathbf{R})$ *es correcta.*

Demostración. Ya que $(\mathbf{R})$ es derivada, existe un árbol de deducción que tiene a $\Gamma \quad \varphi$ en la raíz y axiomas o elementos de $\{\Gamma_1 \quad \varphi_1, \ldots, \Gamma_n \quad \varphi_n\}$ en las hojas. Vamos a demostrar el resultado por inducción en m, siendo m la profundidad del árbol de deducción.

- $m = 1$; Entonces el árbol solo tiene un nodo, la secuencia $\Gamma \quad \varphi$. Como esta secuencia es a la vez una hoja, o bien es $\Gamma_i \quad \varphi_i$ para algún $i \in \{1, \ldots, n\}$, en cuyo caso el enunciado se cumple trivialmente, o bien $\frac{}{\Gamma \quad \varphi}$ es un axioma correcto, en cuyo caso tanto se cumple $\Gamma \vDash \varphi$ y también se cumple el enunciado.

- Sea $m > 1$ y supongamos que el árbol de deducción para (**R**) tiene altura m. Entonces es de la forma:



donde

$$(\text{R'})\ \frac{A_1 \quad \psi_1 \qquad\quad A_k \quad \psi_k}{\Gamma \quad \varphi}$$

es una regla correcta, y cada $A_l \quad \psi_l$ tiene un árbol de deducción T_l de altura $< m$.

Como el árbol de deducción T_l tiene altura $< m$, y tiene en la raíz a $A_l \quad \psi_l$ y en las hojas axiomas o elementos de $\{\Gamma_1 \quad \varphi_1, \dots, \Gamma_n \quad \varphi_n\}$, por hipótesis de inducción tenemos que la regla

$$(R_l)\ \frac{\Gamma_1 \quad \varphi_1 \qquad\quad \Gamma_n \quad \varphi_n}{A_l \quad \psi_l}$$

es correcta, para $l = 1, \dots, k$.

De modo que si suponemos que $\Gamma_1 \quad \varphi_1, \dots, \Gamma_n \quad \varphi_n$ son correctas, entonces $A_l \quad \psi_l$ es correcta para $l = 1, \dots, k$. Por tanto, como ($\mathbf{R}'$) es correcta, $\Gamma \quad \varphi$ también es correcta. Por tanto (**R**) es correcta.

□

Corolario 3.11. *Toda regla derivada es correcta.*

Demostración. Directo usando 3.9 y 3.10. □

Corolario 3.12. *(Corrección) Si $\Phi \vdash \varphi$, entonces $\Phi \vDash \varphi$.*

Demostración. Por definición, si $\Phi \vdash \varphi$, hay un subconjunto finito $\Gamma \subseteq \Phi$ tal que $\vdash \Gamma \quad \varphi$. Es decir, $\frac{}{\Gamma \quad \varphi}$ es un axioma derivado, y por 3.11 es correcto. De modo que se cumple $\Gamma \vDash \varphi$. Esto, a su vez, implica que $\Phi \vDash \varphi$. □

3.7. Completitud

Como en el caso de los *tableaux* de primer orden, la completitud va a ser más complicada de demostrar que la corrección.

Definición 3.13. Decimos que un conjunto Φ es *consistente*, $con(\Phi)$, si $\Phi \nvdash \bot$, es decir, no existe un subconjunto finito $\Gamma \subseteq \Phi$ tal que $\vdash \Gamma \quad \bot$.

El resultado que nos llevará la gran mayoría de la sección será demostrar el teorema 3.17, que dice que si un conjunto Φ es consistente, entonces es satisfactible. Una vez demostrado este teorema, la completitud se deducirá sin mucho esfuerzo. Comenzamos con el siguiente lema:

Lema 3.14. *Las siguientes reglas son derivadas:*

1. *Si α es α-fórmula,*
$$(\alpha)\frac{\Gamma, \alpha_1, \alpha_2, \alpha \quad \chi}{\Gamma, \alpha \qquad \chi}$$
2. *Si β es β-fórmula,*
$$(\beta)\frac{\Gamma, \beta_1, \beta \quad \chi \qquad \Gamma, \beta_2, \beta \quad \chi}{\Gamma, \beta \quad \chi}$$
3. *Si σ es σ-fórmula,*
$$(\sigma)\frac{\Gamma, \sigma, \sigma_1 \quad \chi}{\Gamma, \sigma \quad \chi}$$
4. *SI γ es γ-fórmula y $t \in TERM_S$*
$$(\gamma)\frac{\Gamma, \gamma(t), \gamma \quad \chi}{\Gamma, \gamma \quad \chi}$$
5. *Si δ es δ-fórmula y $c \in C_A$ es nueva,*
$$(\delta)\frac{\Gamma, \delta(c), \delta \quad \chi}{\Gamma, \delta \quad \chi}$$
6. *Si $\theta \in EQ_S$ es θ-fórmula,*
$$(\theta)\frac{\Gamma, \theta \quad \chi}{\Gamma \quad \chi}$$

Demostración. No se incluye. En cada apartado es necesario dividir en casos y crear árboles de deducción para cada caso. □

Proposición 3.15. *Sea T tableau finito para Φ. Sean:*

$$\Gamma := \{\varphi \in \Phi \mid \varphi \text{ aparece en } T\}$$

$$\Gamma_r := \{\varphi \mid \varphi \text{ aparece en la rama } r\}$$

Si para toda rama r de T se cumple que, dada la fórmula χ, $\vdash \Gamma_r \quad \chi$, entonces $\vdash \Gamma \quad \chi$.

Demostración. Sea T *tableau* finito para Φ. Entonces se puede construir en un número finito de pasos $T_0, \ldots, T_n$ mediante la aplicación de las reglas $R_1, \ldots, R_n$.

Realizamos la demostración por inducción sobre n. Denotamos $\Gamma_k := \{\varphi \in \Phi \mid \varphi \text{ aparece en } T_k\}$.

Si $n = 0$ entonces tenemos un *tableaux* inicial $T_0 = \begin{array}{c} \varphi_1 \\ | \\ \varphi_2 \\ \vdots \\ \varphi_k \end{array}$, con $\{\varphi_1, \ldots, \varphi_k\} \subseteq \Phi$. Por tanto, solo tenemos una rama r que además verifica que $\Gamma = \Gamma_r$, de lo que se sigue el resultado.

Sea $n \geq 1$, supongamos que el resultado se cumple para T_{n-1}. Para ver que se cumple en T_n, tenemos que distinguir casos dependiendo de la regla que sea R_n. En todos los casos, usaremos la letra r para nombrar las ramas de T_{n-1} y s para nombrar a las ramas de T_n.

- Si $R_n = R_\alpha$, entonces en T_{n-1} tenemos una rama r_0 conteniendo la α-fórmula α, y esa rama se extiende añadiendo α_1 y α_2 para formar la rama s_0 de T_n.

  Supongamos ahora que $\vdash \Gamma_s \quad \chi$ para toda rama s de T_n y queremos llegar a que se cumple $\vdash \Gamma_n \quad \chi$. Veamos primero que para toda rama r en T_{n-1}, tenemos $\vdash \Gamma_r \quad \chi$. Para ello basta comprobar que $\vdash \Gamma_{r_0} \quad \chi$, ya que el resto de ramas r que no son r_0 no se han extendido y son también ramas de T_n.

  Pero sabemos que $\vdash \Gamma_{s_0} \quad \chi$. Además, como $\alpha \in \Gamma_{r_0}$ y $\Gamma_{s_0} = \Gamma_{r_0} \cup \{\alpha_1, \alpha_2\}$, por el apartado 1 de 3.14 la regla $\frac{\Gamma_{s_0} \quad \chi}{\Gamma_{r_0} \quad \chi}$ es derivada. Por tanto como se cumple $\vdash \Gamma_{s_0} \quad \chi$, también se cumple $\vdash \Gamma_{r_0} \quad \chi$.

  Como tenemos $\vdash \Gamma_r \quad \chi$ para toda rama r de T_{n-1}, por hipótesis de inducción $\vdash \Gamma_{n-1} \quad \chi$. Como $\Gamma_{n-1} \subseteq \Gamma_n$, esto implica $\vdash \Gamma_n \quad \chi$.

- Si $R_n = R_\beta$, entonces en T_{n-1} tenemos una rama r_0 conteniendo la β-fórmula β, y esa misma rama se extiende añadiendo β_1 y β_2 para formar las ramas s_1 y s_2 de T_n, respectivamente.

  Supongamos ahora que $\vdash \Gamma_s \quad \chi$ para toda rama s de T_n y queremos llegar a que se cumple $\vdash \Gamma_n \quad \chi$. Veamos primero que para toda rama r en T_{n-1}, tenemos $\vdash \Gamma_r \quad \chi$. Para ello basta comprobar que $\vdash \Gamma_{r_0} \quad \chi$, ya que el resto de ramas r que no son r_0 no se han extendido y son también ramas de T_n.

Pero sabemos que $\vdash \Gamma_{s_1} \quad \chi$ y $\vdash \Gamma_{s_2} \quad \chi$. Además, como $\beta \in \Gamma_{r_0}$ y $\Gamma_{s_1} = \Gamma_{r_0} \cup \{\beta_1\}, \Gamma_{s_2} = \Gamma_{r_0} \cup \{\beta_2\}$ por el apartado 2 de 3.14 la regla $\frac{\Gamma_{s_1} \quad \chi \qquad \Gamma_{s_2} \quad \chi}{\Gamma_{r_0} \quad \chi}$ es derivada. Por tanto como se cumplen $\vdash \Gamma_{s_1} \quad \chi$ y $\vdash \Gamma_{s_2} \quad \chi$, también se cumple $\vdash \Gamma_{r_0} \quad \chi$.

Como tenemos $\vdash \Gamma_r \quad \chi$ para toda rama r de T_{n-1}, por hipótesis de inducción $\vdash \Gamma_{n-1} \quad \chi$. Como $\Gamma_{n-1} \subseteq \Gamma_n$, esto implica $\vdash \Gamma_n \quad \chi$.

- Si $R_n = R_\sigma$, entonces en T_{n-1} tenemos una rama r_0 conteniendo la σ-fórmula σ, y esa misma rama se extiende añadiendo σ_1 para formar la rama s_0 de T_n.

  Supongamos ahora que $\vdash \Gamma_s \quad \chi$ para toda rama s de T_n y queremos llegar a que se cumple $\vdash \Gamma_n \quad \chi$. Veamos primero que para toda rama r en T_{n-1}, tenemos $\vdash \Gamma_r \quad \chi$. Para ello basta comprobar que $\vdash \Gamma_{r_0} \quad \chi$, ya que el resto de ramas r que no son r_0 no se han extendido y son también ramas de T_n.

  Pero sabemos que $\vdash \Gamma_{s_0} \quad \chi$. Además, como $\sigma \in \Gamma_{r_0}$ y $\Gamma_{s_0} = \Gamma_{r_0} \cup \{\sigma_1\}$, por el apartado 3 de 3.14 la regla $\frac{\Gamma_{s_0} \quad \chi}{\Gamma_{r_0} \quad \chi}$ es derivada. Por tanto como se cumple $\vdash \Gamma_{s_0} \quad \chi$, también se cumple $\vdash \Gamma_{r_0} \quad \chi$.

  Como tenemos $\vdash \Gamma_r \quad \chi$ para toda rama r de T_{n-1}, por hipótesis de inducción $\vdash \Gamma_{n-1} \quad \chi$. Como $\Gamma_{n-1} \subseteq \Gamma_n$, esto implica $\vdash \Gamma_n \quad \chi$.

- Las reglas γ, δ siguen el mismo razonamiento que la σ cambiando σ por γ/δ y σ_1 por $\gamma(t)/\delta(c)$ respectivamente, y usando los apartados 4/5 de 3.14.

  La regla θ también se hace de forma similar, viendo que si en la regla añadimos una fórmula θ a una rama r_0 de T_{n-1} para obtener una rama s_0 de T_n, entonces por el apartado 6 de 3.14, la regla $\frac{\Gamma_{s_0} \quad \chi}{\Gamma_{r_0} \quad \chi}$ es derivada.

- Si $R_n = R_{hip}$, entonces en T_{n-1} tenemos una rama r_0, a la que añadimos la fórmula φ de Φ para formar la rama s_0 de T_n.

  Supongamos ahora que $\vdash \Gamma_s \quad \chi$ para toda rama s de T_n y queremos llegar a que se cumple $\vdash \Gamma_n \quad \chi$. Veamos primero que para toda rama r en T_{n-1}, tenemos $\vdash \Gamma_r \quad \varphi \to \chi$. Dos casos:

  - $r \neq r_0$. Entonces r también es una rama de T_n, por tanto sabemos $\vdash \Gamma_r \quad \chi$. Por (**FS**), obtenemos $\vdash \Gamma_r, \varphi \quad \chi$, y por ($\to$ **C**), llegamos a que $\vdash \Gamma_r \quad \varphi \to \chi$.

- $r = r_0$. Sabemos que $\vdash \Gamma_{s_0} \quad \chi$, por tanto por ($\to$ **C**), tenemos que $\vdash \Gamma_{s_0} \setminus \{\varphi\} \quad \varphi \to \chi$. Como $\Gamma_{s_0} \setminus \{\varphi\} \subseteq \Gamma_{r_0}$, por (**FS**) tenemos que $\vdash \Gamma_{r_0} \quad \varphi \to \chi$.

De modo que como tenemos que $\vdash \Gamma_r \quad \varphi \to \chi$ para toda rama r de T_{n-1}, por hipótesis de inducción con la fórmula $\varphi \to \chi$ tenemos que $\vdash \Gamma_{n-1} \quad \varphi \to \chi$. Como $\Gamma_{n-1} \subseteq \Gamma_n$, tenemos $\vdash \Gamma_n \quad \varphi \to \chi$. Como además $\vdash \Gamma_n \quad \varphi$ es un axioma del supuesto, por (**MP**) deducimos que $\vdash \Gamma_n \quad \chi$, como queríamos.

□

Corolario 3.16. *Si el conjunto de fórmulas Φ tiene un tableau cerrado, entonces $\Phi \vdash \bot$.*

Demostración. Sea T un *tableau* cerrado para Φ. Sea r rama de T. Si $\bot \in \Gamma_r$, entonces $\vdash \Gamma_r \quad \bot$, por (**SP**). Si existe cierta φ tal que $\varphi, \neg\varphi \in \Gamma_r$, entonces, por ($\mathbf{CT_2}$), $\vdash \Gamma_r \quad \bot$.

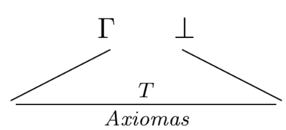
De modo que como para cada rama r se cumple $\vdash \Gamma_r \quad \bot$, por 3.15 tenemos $\vdash \Gamma \quad \bot$, con $\Gamma := \{\varphi \in \Phi \mid \varphi \text{ está en } T\}$. Como Γ es subconjunto finito de Φ, $\Phi \vdash \bot$. □

Teorema 3.17. *Si $con(\Phi)$, entonces $Sat(\Phi)$.*

Demostración. Es la forma contrarrecíproca del resultado anterior. □

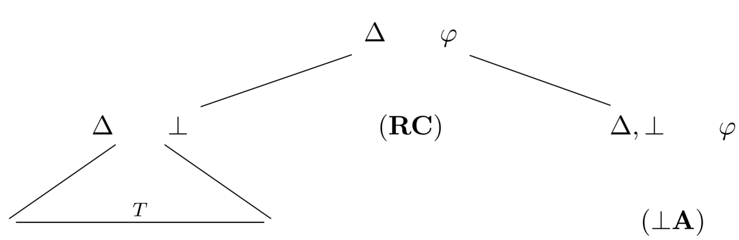
Corolario 3.18. *(Completitud) Si $\Phi \vDash \varphi$, entonces $\Phi \vdash \varphi$.*

Demostración. Si $\Phi \vDash \varphi$ entonces $Insat(\Phi \cup \{\neg\varphi\})$, es decir, por 3.17 es falso que $con(\Phi \cup \{\neg\varphi\})$, y entonces existe un subconjunto finito $\Gamma \subseteq \Phi \cup \{\neg\varphi\}$ tal que $\vdash \Gamma \quad \bot$. Tenemos, por tanto,



Veamos si existe un subconjunto finito $\Delta \subseteq \Phi$ tal que $\vdash \Delta \quad \varphi$.

- Si $\neg\varphi \notin \Gamma$, tomando $\Delta := \Gamma$ obtenemos:



- Si $\neg\varphi \in \Delta$, tomando $\Delta := \Gamma \setminus \{\neg\varphi\}$ obtenemos:

$$
\begin{array}{cc}
\Delta & \varphi \\
\multicolumn{2}{c}{\Big|\,(\mathbf{DN!})} \\
\Delta & \neg\neg\varphi \\
\multicolumn{2}{c}{\Big|\,(\neg\mathbf{C})} \\
\Delta, \neg\varphi & \bot \\
\multicolumn{2}{c}{T} \\
\hline
\multicolumn{2}{c}{\mathit{Axiomas}}
\end{array}
$$

de modo que en ambos casos se cumple el enunciado. $\square$

De modo que hemos visto que si S es signatura numerable y $\Phi \subseteq FORM_S$ es insatisfactible, entonces Φ tiene *tableau* cerrado y es, por tanto, inconsistente. De hecho, este resultado también se puede demostrar cuando S no es numerable.[1]

Finalmente, demos una visión global de lo que hemos estudiado en estas últimas subsecciones. Partiendo de los dos pares de conceptos *deducción formal-consistencia* y *corrección-satisfactibilidad* hemos establecido un nexo entre las perspectivas sintáctica y semántica, respectivamente representadas por cada par de tales conceptos.

Este nexo consiste en los Teoremas de Corrección y Completitud, que nos permiten pasar 'de un lado a otro': El teorema de corrección nos permite traducir algunas afirmaciones sintácticas a semánticas, y el de completitud nos permite traducir afirmaciones semánticas a sintácticas.

A continuación mostramos la correspondencia que hemos definido:

$$
\begin{array}{rrcl}
 & \text{Sintaxis} & & \text{Semántica} \\
 & \text{Fórmulas} & \Longleftrightarrow & \text{Modelos} \\
\Phi \vdash \varphi & (\Phi \vdash_{tb} \varphi) & \Longleftrightarrow & \Phi \models \varphi \\
 & con(\Phi) & \Longleftrightarrow & sat(\Phi) \\
 & \vdash \varphi \leftrightarrow \psi & \Longleftrightarrow & \varphi \sim \psi \\
 & \vdash \varphi & \Longleftrightarrow & \models \varphi
\end{array}
$$

[1] Este resultado se puede encontrar en el capítulo 5 del libro *Mathematical Logic*, de los autores H.-D. Ebbinghaus, J. Flum y W. Thomas.

4 Alcance y limitaciones de la lógica de primer orden

Vamos a dedicar esta sección a estudiar las nociones y resultados más importantes de este manual.

4.1. Teorema de Compacidad y modelos

Comencemos demostrando una versión más general de 2.87, que se convertirá en una de nuestras herramientas básicas en las siguientes secciones:

Teorema 4.1. *(Teorema de compacidad) Sea S signatura, $\Phi \subseteq PROP_S$, $\varphi \in PROP_S$. Entonces,*

a) $\Phi \vDash \varphi$ si y solo si existe $\Phi_0 \subseteq_{fin} \Phi$ tal que $\Phi_0 \vDash \varphi$.

b) $sat(\Phi)$ si y solo si para todo $\Phi_0 \subseteq_{fin} \Phi$, se cumple $sat(\Phi_0)$.

Demostración. Demostramos solo el caso de signatura S numerable, que es donde hemos estudiado los teoremas de corrección y completitud. El caso no numerable también se puede demostrar.[1]

a) La implicación $\Longleftarrow$ es trivial. Veamos $\Longrightarrow$. Si $\Phi \vDash \varphi$, $\Phi \vdash \varphi$, por tanto hay $\Phi_0 \subseteq \Phi$ finito tal que $\vdash \Phi_0 \quad \varphi$, de modo que $\Phi_0 \vdash \varphi$ y por tanto $\Phi_0 \vDash \varphi$.

b) La implicación $\Longrightarrow$ es trivial. Veamos $\Longleftarrow$ en su forma contrapositiva. Si $insat(\Phi)$, $\Phi \vdash \bot$, por tanto hay $\Phi_0 \subseteq \Phi$ finito tal que $\vdash \Phi_0 \quad \bot$, luego $\Phi_0 \vdash \bot$ y por tanto $\Phi_0 \vDash \bot$, es decir, $Insat(\Phi_0)$.

$\square$

Recordemos que una sentencia es una fórmula de primer orden en que ninguna variable aparece libre. Pues bien, a partir de ahora vamos a trabajar con sentencias. Dada una signatura S, definimos $Sent_S := \{\varphi \in FORM_S \mid lib(\varphi) = \emptyset\}$.

[1] Una prueba se puede encontrar en *Logic and Structure*, Dirk Van Dalen, Ed. Springer

Ahora supongamos que tenemos una signatura S, una S-álgebra $\mathfrak{A}$ con conjunto soporte A y una sentencia $\varphi \in Sent_S$, y sean $\sigma_1, \sigma_2 : Var \to A$ funciones que asignan valores a las variables. Con ellas podemos definir dos interpretaciones, $\mathfrak{I}_1 := \langle \mathfrak{A}, \sigma_1 \rangle$ y , $\mathfrak{I}_2 := \langle \mathfrak{A}, \sigma_2 \rangle$. Entonces, por el Lema de Coincidencia 2.41, como $lib(\varphi) = \emptyset$, tenemos que $\varphi^{\mathfrak{I}_1} = \varphi^{\mathfrak{I}_2}$.

Es decir, que una interpretación $\mathfrak{I} := \langle \mathfrak{A}, \sigma \rangle$ verifique o no una sentencia solo depende de $\mathfrak{A}$, no de σ. Por tanto, es natural introducir la siguiente definición:

Definición 4.2. Dadas una signatura S, una S-álgebra $\mathfrak{A}$ y una sentencia $\varphi \in Sent_S$, decimos que $\mathfrak{A} \vDash \varphi$, $\mathfrak{A}$ *es un modelo de* φ, cuando para alguna asignación $\sigma : Var \to A$ la S-interpretación $\mathfrak{I} := \langle \mathfrak{A}, \sigma \rangle$ verifica φ.

Definición 4.3. Si $\Phi \subseteq Sent_S$, definimos la siguiente clase:

$$Mod_S(\Phi) = \{\mathfrak{A} \mid \mathfrak{A} \ S\text{-álgebra y } \mathfrak{A} \vDash \varphi \text{ para toda } \varphi \in \Phi\}.$$

Así, decimos que una álgebra $\mathfrak{A}$ es modelo de Φ si $\mathfrak{A} \vDash \varphi$ para toda $\varphi \in \Phi$.

Teorema 4.4. *Si $\Phi \subseteq Sent_S$ admite modelos finitos arbitrariamente grandes, entonces admite algún modelo infinito.*

Demostración. Para cada n, tenemos un modelo $\mathfrak{A}_\mathfrak{n}$ de soporte A_n con $|A_n| > n$. Ahora vamos a definir para cada $n \geq 2$ una sentencia φ_n que dice 'hay al menos n elementos en el conjunto soporte':
$\varphi_2 = \exists x \exists y \neg x \doteq y$
$\varphi_3 = \exists x \exists y \exists z \neg x \doteq y \wedge \neg x \doteq z \wedge \neg y \doteq z$
$\vdots$
$\varphi_n = \exists x_1 \exists x_2 \ldots \exists x_n \bigwedge_{i=1}^{n} \bigwedge_{j=i+1}^{n} \neg x_i \doteq x_j$
$\vdots$

No es difícil verificar que una estructura es modelo de φ_n si y solo si su soporte tiene al menos n elementos. Ahora definimos el conjunto de sentencias $\hat{\Phi} := \Phi \cup \{\varphi_n\}_{n \geq 2}$. Si comprobamos que $\hat{\Phi}$ es satisfactible habremos acabado, ya que cualquier modelo de $\hat{\Phi}$ tiene que tener infinitos elementos en su soporte. Pero por el Teorema de Compacidad, para esto nos basta comprobar que cualquier subconjunto finito de $\hat{\Phi}$ es satisfactible.

Sea $\Phi_0 \subseteq \hat{\Phi}$ finito. Entonces podemos definir $n_0 := \max\{n \in \mathbb{N} | \varphi_n \in \Phi_0\}$. Y está claro que $\mathfrak{A}_{n_0}$ es modelo de Φ_0, ya que es modelo de Φ y de φ_n para cualquier $n \leq n_0$. Por tanto Φ_0 es satisfactible. □

4.2. Teoremas de Löwenheim-Skolem

Nos disponemos ahora a ver los teoremas de Löwenheim-Skolem que, dado un conjunto de sentencias Φ, nos garantizan la existencia de modelos para Φ de cardinales grandes (versión ascendente del teorema) o pequeños (versión descendente del teorema). En los siguientes teoremas, con el 'cardinal de una signatura' nos referimos al cardinal del conjunto de constantes, funciones y predicados de la signatura, y con el 'cardinal de un modelo' nos referimos al cardinal de su conjunto soporte.

Teorema 4.5. *(Teorema de Löwenheim-Skolem, caso numerable descendente) Sea S signatura numerable, $\Phi \subseteq Sent_S$ satisfactible. Entonces Φ tiene un modelo a lo sumo numerable.*

Demostración. Como S es numerable y Φ es satisfactible, el *tableau* canónico para Φ tendrá alguna rama abierta r, de modo que $\Phi \subseteq \Gamma_r$ y Γ_r es de Hintikka. Entonces cogemos la interpretación $\mathfrak{I}_{\Gamma_r}$, como está definida en 2.11.4.

Esta interpretación satisface Γ_r, de modo que como $\Phi \subseteq \Gamma_r$, también satisface Φ. Es decir, su álgebra $\mathfrak{T}_{\Gamma_r}$ es modelo de Φ.

Además, su conjunto soporte es $T_{\Gamma_r}/\equiv_{\Gamma_r}$. Pero $|T_{\Gamma_r}/\equiv_{\Gamma_r}| \leq |T_{\Gamma_r}| \leq |FORM_S|$. Como $FORM_S$ es numerable, este es el modelo con soporte numerable que buscábamos. □

Pasamos a ver la versión general del teorema anterior, de la cual no se incluye demostración.

Teorema 4.6. *(Teorema de Löwenheim-Skolem descendente) Sea S signatura, $\Phi \subseteq Sent_S$ satisfactible. Entonces Φ tiene un modelo de cardinal $\leq |FORM_S|$.*

Nota 4.7. Para cada signatura con conjunto de símbolos de constante C, podemos considerar el conjunto de fórmulas que nos dicen que las constantes son distintas dos a dos, $\Phi_C = \{\neg c_1 \doteq c_2\}_{c_1,c_2 \in C; c_1 \neq c_2}$. Entonces, cualquier modelo de Φ tiene que tener cardinal $\geq |C|$.

Esto nos dice que el teorema 4.6 no es mejorable:

Supongamos que tenemos una signatura S sin funciones ni predicados y con un conjunto infinito C de constantes. Entonces, se puede comprobar que $|FORM_S| = |C|$. Si ahora consideramos el conjunto de sentencias Φ_C, es satisfactible (basta elegir un modelo $\mathfrak{A}$ cuyo conjunto soporte sea el propio conjunto de constantes y definir $c^{\mathfrak{A}} = c$ para toda constante c), sin embargo, cualquier modelo de Φ_C tiene cardinal superior a $|C| = |FORM_S|$. De modo que en este caso, la desigualdad de 4.6 no puede ser estricta.

Ejemplo 4.8. Sea la signatura $S = \langle 0, 1, \cdot, +, < \rangle$ y el álgebra habitual de los reales, $\mathbb{R}^< = \langle \mathbb{R}; 0, 1; \cdot, +; \leq \rangle$.

Entonces podemos definir el conjunto de sentencias que cumple $\mathbb{R}^<$:

$$Th(\mathbb{R}^<) := \{\varphi \in Sent_S \mid \mathbb{R}^< \vDash \varphi\})$$

El teorema 4.6 nos dice que de hecho hay un modelo numerable $\mathbb{R}^!$ que también verifica $Th(\mathbb{R}^<)$, es decir, cualquier sentencia que cumpla $\mathbb{R}^<$ también la cumple este modelo numerable $\mathbb{R}^!$.

Pasamos directamente a ver la versión ascendente del Teorema de Löwenheim-Skolem, que sí podremos demostrar (usando el caso no numerable del teorema de compacidad).

Teorema 4.9. *(Teorema de Löwenheim-Skolem ascendente) Sea S signatura, $\Phi \subseteq Sent_S$ de forma que existe algún modelo infinito de Φ, y sea κ un cardinal. Entonces Φ tiene un modelo de cardinal $\geq \kappa$.*

Demostración. Sea K un conjunto de cardinal κ (por ejemplo, $K = \kappa$). Consideramos una nueva signatura que resulta de añadir a S una constante por cada elemento de K:

$$S' = S \cup \{c_a; a \in K\} \qquad \text{donde } c_a \neq c_b \text{ si } a \neq b.$$

Sea $\Phi' \subseteq FORM_{S'}$ definido como $\Phi \cup \{\neg c_a \doteq c_b | a, b \in K, a \neq b\}$. Veamos primero que Φ' es satisfactible. Para ello, claro, nos basta el teorema de compacidad 4.1.

De modo que sea $\Phi_0 \subseteq \Phi'$ finito, veamos que es satisfactible. Sean $a_1, \ldots, a_k$ los elementos a de K tales que c_a aparece en alguna fórmula de Φ_0. Entonces, llamando $\Phi_1 = \Phi \cup \{\neg c_{a_i} \doteq c_{a_j} | i, j \in \{1, \ldots, n\}, i \neq j\}$, tenemos que $\Phi_0 \subseteq \Phi_1$.

Pero Φ_1 es satisfactible, ya que como Φ tiene un modelo infinito $\mathfrak{A}$ en S, podemos elegir el modelo $\mathfrak{A}$ para Φ_1 en S', asignando k elementos distintos del conjunto soporte a $c_{a_1}, \ldots, c_{a_k}$ y asignando los valores que queramos al resto de constantes.

Por tanto, por el Teorema de Compacidad, ya tenemos que Φ' es satisfactible en la signatura S'. Sea $\mathfrak{A}_0$ una S'-álgebra que satisface Φ'. Entonces, el cardinal de $\mathfrak{A}_0$ es $\geq \kappa$, ya que tenemos κ constantes distintas y Φ' afirma que representan distintos elementos del conjunto soporte. Hemos encontrado una S'-álgebra que satisface Φ de cardinal $\geq \kappa$. Para probar el enunciado, queremos una S-álgebra que cumpla lo mismo.

Para ello basta definir la S-álgebra $\mathfrak{A}_1$, que tiene el mismo soporte que $\mathfrak{A}_0$ y asigna a cada constante, función y predicado de S lo mismo que $\mathfrak{A}_0$. Como $\mathfrak{A}_0$ es modelo de Φ, $\mathfrak{A}_1$ también será modelo de Φ. Por tanto, ya hemos encontrado una $S-$álgebra que satisface Φ, y de cardinal $\geq \kappa$, como queríamos. $\square$

Teorema 4.10. *(Teorema de Löwenheim-Skolem-Tarski) Sean S signatura, $\Phi \subseteq Sent_S$ con modelos infinitos y κ cardinal tal que $\kappa \geq |FORM_S|$. Entonces existe $\mathfrak{A} \in Mod(\Phi)$ tal que $|\mathfrak{A}| = \kappa$.*

Demostración. Usando la misma notación que en el teorema anterior, podemos obtener una S'-álgebra de cardinal κ que consiste en añadir constantes a Φ y un conjunto de sentencias $\Phi' \subseteq FORM_{S'}$ satisfactible y tal que cualquier modelo de Φ' tiene cardinal $\geq \kappa$.

Pero entonces, por el Teorema de Löwenheim descendente, 4.6, tenemos que hay una S'-álgebra $\mathfrak{A}$ que satisface Φ' y de cardinal $\leq \kappa$. Como es modelo de Φ', tiene cardinal $\geq \kappa$, por tanto $\mathfrak{A}$ tiene cardinal exactamente κ.

Ahora, igual que en el teorema anterior, creamos una S-álgebra $\mathfrak{A}_1$ con el mismo conjunto soporte que $\mathfrak{A}$ y que asigna a cada constante, función y predicado de S lo mismo que $\mathfrak{A}$. $\mathfrak{A}_1$ es un modelo de Φ de cardinal κ, como queríamos. $\square$

4.3. Axiomatizabilidad

Definiciones 4.11. Dada S signatura, sea $\mathcal{K}$ clase de S-álgebras.

- $\mathcal{K}$ se dice *clase elemental (o finitamente axiomatizable)* si existe $\Phi \subseteq Sent_S$ finito tal que $\mathcal{K} = Mod(\Phi)$.
- $\mathcal{K}$ se dice *clase* Δ*-elemental (o axiomatizable)* si existe $\Phi \subseteq Sent_S$ tal que $\mathcal{K} = Mod(\Phi)$.

Ejemplo 4.12. Sea $G := \langle \cdot, e \rangle$. Sea

$$\Phi_G := \begin{cases} \forall x \forall y \forall z \, (x \cdot (y \cdot z)) = ((x \cdot y) \cdot z) \\ \forall x \, x \cdot e \doteq x \wedge e \cdot x \doteq x \\ \forall x \exists y \, x \cdot y \doteq e \wedge y \cdot x \doteq e \end{cases}$$

Sea $\mathcal{K} := \{\mathfrak{A} \mid \mathfrak{A} \text{ es grupo}\}$. Entonces es claro que $\mathcal{K} = Mod(\Phi_G)$, por tanto $\mathcal{K}$ es elemental.

Ejemplo 4.13. Sea $C := \langle 0, 1, +|_2, \cdot|_2 \rangle$ y sea Φ_C el conjunto de las sentencias que expresan los axiomas de cuerpos. Si queremos expresar la propiedad de 'ser de característica 0' necesitamos las sentencias $\Phi_C \cup \{\neg \chi_p \mid p \text{ es primo}\}$, siendo

$$\chi_p := 1 \underbrace{+ \cdots +}_{p \text{ veces}} 1 \doteq 0.$$

Por tanto, solo podemos afirmar que $\mathcal{K} := \{\mathfrak{A} \mid \mathfrak{A} \text{ es cuerpo de característica } 0\}$ es Δ-elemental. De hecho, veremos que no es elemental.

Ejemplo 4.14. $\mathcal{K} := \{\mathfrak{A} \mid \mathfrak{A} \text{ es cuerpo algebraicamente cerrado}\}$ es clase Δ-elemental.

Conviene hacer notar lo siguiente. Dada S signatura, dada la clase elemental $\mathcal{K}$ y el conjunto finito $\Phi := \{\varphi_1, \ldots, \varphi_n\} \subseteq Sent_S$ tal que $\mathcal{K} = Mod(\Phi)$, sea $\varphi := \varphi_1 \wedge \cdots \wedge \varphi_n$. Entonces $A \in Mod(\Phi)$ si y solo si $A \in Mod(\varphi)$. Por tanto, hemos probado que

Proposición 4.15. *Dada S signatura, la clase de S-estructuras $\mathcal{K}$ es elemental si y solo si existe $\varphi \in Sent_S$ tal que $\mathcal{K} = Mod(\varphi)$.*

Proposición 4.16. *Sea S signatura.*

1. *Si $\mathcal{K}$ es elemental y existe $\Phi \subseteq Sent_S$ tal que $Mod(\Phi) = \mathcal{K}$, entonces existe $\Phi_0 \subseteq \Phi$ finito tal que $Mod(\Phi_0) = \mathcal{K}$.*

2. *$\mathcal{K}$ es elemental si y solo si $\mathcal{K}$ y $\mathcal{K}^{\complement}$ son Δ-elementales.*

Demostración.

(1) Si $\mathcal{K}$ es elemental, existe $\varphi \in Sent_S$ tal que $\mathcal{K} = Mod(\varphi)$. Como, por otro lado, $\mathcal{K} = Mod(\Phi)$, $\Phi \models \varphi$ y, por compacidad, existe $\Phi_0 \subseteq \Phi$ finito tal que $\Phi_0 \models \varphi$, luego

$$\mathcal{K} = Mod(\varphi) \subseteq Mod(\Phi_0) \subseteq Mod(\Phi) = \mathcal{K},$$

con lo que $\mathcal{K} = Mod(\Phi_0)$.

(2) Si $\mathcal{K}$ es elemental, es claramente Δ-elemental y, además, existe $\varphi \in Sent_S$ tal que $\mathcal{K} = Mod(\varphi)$, con lo que $Mod(\neg\varphi) = \mathcal{K}^{\complement}$, lo que nos da que $\mathcal{K}^{\complement}$ es elemental y por tanto Δ-elemental.

Recíprocamente, si $\mathcal{K}$ y $\mathcal{K}^{\complement}$ son Δ-elementales, entonces existen $\Phi, \Psi \subseteq Sent_S$ tales que

$$Mod(\Phi) = \mathcal{K} \text{ y } Mod(\Psi) = \mathcal{K}^{\complement}$$

Vemos que $Mod(\Phi \cup \Psi) = Mod(\Phi) \cap Mod(\Psi) = \mathcal{K} \cap \mathcal{K}^{\complement} = \emptyset$, con lo que $\Phi \cup \Psi$ es insatisfactible. Por compacidad, existe un subconjunto finito de $\Phi \cup \Psi$ de la forma $\Phi_0 \cup \Psi_0$, con $\Phi_0 \subseteq \Phi$ y $\Psi_0 \subseteq \Psi$ finitos, que es insatisfactible, esto es, $Mod(\Phi_0 \cup \Psi_0) = \emptyset$. Por otro lado,

$$\mathcal{K} = Mod(\Phi) \subseteq Mod(\Phi_0) \text{ y } \mathcal{K}^{\complement} = Mod(\Psi) \subseteq Mod(\Psi_0).$$

Esto junto a que $Mod(\Phi_0) \cap Mod(\Psi_0) = \emptyset$ nos permite deducir que $Mod(\Phi_0) \cap \mathcal{K}^{\complement} = \emptyset$, es decir, $Mod(\Phi_0) \subseteq \mathcal{K}$. Por tanto $Mod(\Phi_0) = \mathcal{K}$, lo cual conlleva que $\mathcal{K}$ es elemental, como queríamos ver.

□

Ejemplo 4.17. Consideremos de nuevo la signatura C, los axiomas de cuerpos, Φ_C, y las fórmulas χ_p, $p \in \mathbb{N}$. Definamos $\Psi := \Phi_C \cup \{\neg\chi_p \mid p \in \mathbb{N}\}$ y $\mathcal{K} := \{\mathfrak{A} \mid \mathfrak{A}$ es cuerpo de característica $0\}$. Ya vimos que $\mathcal{K} = Mod(\Psi)$ y, por tanto, que $\mathcal{K}$ es clase Δ-elemental. Veamos que no es finitamente axiomatizable.

Si fuera finitamente axiomatizable, por 4.16 existiría cierto $\Psi_0 \subseteq \Psi$ finito tal que $\mathcal{K} = Mod(\Psi_0)$. Por ser finito, existe p primo tal que $\Psi_0 \subseteq \Phi_p := \Phi_C \cup \{\neg\chi_q \mid q \leq p\}$. Consideremos el cuerpo $\mathbb{F}_t$ de t elementos, con $t > p$ primo. Este cuerpo no es de característica 0 y satisface Φ_p, lo que contradice la suposición de que $\mathcal{K} = Mod(\Psi_0)$.

Ejemplo 4.18. En las condiciones anteriores se puede demostrar, por compacidad, que si $\Psi \models \sigma$, con $\sigma \in Sent_C$, entonces σ es satisfactible en todos los cuerpos de característica mayor que p, para cierto primo suficientemente grande.

De aquí se sigue que, si $f, g \in \mathbb{Z}[x]$ son coprimos en $\mathbb{Q}[x]$, entonces son coprimos en $\mathbb{Z}_p[x]$, para cierto primo p suficientemente grande.

Definiciones 4.19.

- Sea S una signatura, $\mathfrak{A}$ una S-álgebra. Definimos *la teoría de* $\mathfrak{A}$ como:

$$Th(\mathfrak{A}) := \{\sigma \in Sent_S \mid \mathfrak{A} \vDash \sigma\}$$

- Dos S-álgebras $\mathfrak{A}, \mathfrak{B}$ se dicen *elementalmente equivalentes*, $\mathfrak{A} \equiv \mathfrak{B}$ si $Th(\mathfrak{A}) = Th(\mathfrak{B})$.

Proposición 4.20. *Sea S signatura y sean $\mathfrak{A}, \mathfrak{B}$ dos S-álgebras. Si $\mathfrak{A} \approx \mathfrak{B}$, entonces $\mathfrak{A} \equiv \mathfrak{B}$.*

Demostración. Directo por el Lema de Isomorfía, 2.48. □

Teorema 4.21. *Sea S signatura, $\mathfrak{A}$ S-álgebra de cardinal infinito. Entonces existe una S-álgebra $\mathfrak{A}^*$ no isomorfa a $\mathfrak{A}$ tal que $\mathfrak{A}^* \equiv \mathfrak{A}$.*

Demostración. Por ser $\mathfrak{A}$ infinita, $\mathfrak{A} \vDash Th(\mathfrak{A})$ y entonces $Th(\mathfrak{A})$ tiene modelos infinitos.

Sea κ cardinal tal que $\kappa > |\mathfrak{A}|$ y $\kappa > |FORM_S|$, por ejemplo, $\kappa = 2^{max(|\mathfrak{A}|,|FORM_S|)}$. Por el Teorema de Löwenheim-Skolem-Tarski, $Th(\mathfrak{A})$ tiene modelos de cardinal κ. Seleccionemos uno de estos modelos, $\mathfrak{A}^*$. Veamos que $Th(\mathfrak{A}) = Th(\mathfrak{A}^*)$:

Como $\mathfrak{A}^* \vDash Th(\mathfrak{A})$, dada $\sigma \in Th(\mathfrak{A})$, $\mathfrak{A}^* \vDash \sigma$ y por tanto $\sigma \in Th(\mathfrak{A}^*)$, luego $Th(\mathfrak{A}) \subseteq Th(\mathfrak{A}^*)$.

Recíprocamente, dada $\sigma \in Th(\mathfrak{A}^*)$, $\neg\sigma \notin Th(\mathfrak{A}^*)$ y por tanto $\neg\sigma \notin Th(\mathfrak{A})$, con lo que $\mathfrak{A} \nvDash \neg\sigma$ y $\mathfrak{A} \vDash \sigma$ y finalmente $\sigma \in Th(\mathfrak{A})$.

Además, $\mathfrak{A}$ y $\mathfrak{A}^*$ no son isomorfas, ya que tienen distinto cardinal. □

Definiciones 4.22. Sean S signatura, $\mathfrak{A}$ S-álgebra:

- $Elem(\mathfrak{A}) := \{\mathfrak{B}\ S\text{-álgebra} \mid \mathfrak{B} \equiv \mathfrak{A}\}$.
- $Iso(\mathfrak{A}) := \{\mathfrak{B}\ S\text{-álgebra} \mid \mathfrak{B} \approx \mathfrak{A}\}$.
- $\mathfrak{A}$ se dice *(finitamente) axiomatizable* si $Iso(\mathfrak{A})$ es (finitamente) axiomatizable.

Nótese que de 4.20 se sigue que $Iso(\mathfrak{A}) \subseteq Elem(\mathfrak{A})$, para cada S-álgebra $\mathfrak{A}$.

Teorema 4.23. *Sea S signatura, $\mathfrak{A}$ S-álgebra. Entonces, $Elem(\mathfrak{A})$ es la menor clase axiomatizable de S-álgebras a la que pertenece $\mathfrak{A}$.*

Demostración. $Elem(\mathfrak{A})$ es una clase axiomatizable de S-álgebras, definida por el conjunto de proposiciones $Th(\mathfrak{A})$. Para comprobar esto, por una parte está claro que $Elem(\mathfrak{A}) \subseteq Mod(Th(\mathfrak{A}))$, y por otra sea $\mathfrak{A}' \in Mod(Th(\mathfrak{A}))$, veamos que $\mathfrak{A} \equiv \mathfrak{A}'$, es decir, $Th(\mathfrak{A}) = Th(\mathfrak{A}')$.

Está claro que $Th(\mathfrak{A}) \subseteq Th(\mathfrak{A}')$ ya que $\mathfrak{A}'$ es modelo de $Th(\mathfrak{A})$. Por otra parte, sea $\varphi \in Th(\mathfrak{A}')$. Entonces $\mathfrak{A}' \vDash \varphi$, por tanto $\mathfrak{A}' \nvDash \neg\varphi$, por tanto por el otro contenido tenemos que $\mathfrak{A} \models \varphi$, por tanto $\mathfrak{A} \models \varphi$.

Además, cualquier clase axiomatizable $\mathcal{K}$ de S-álgebras que contenga a $\mathfrak{A}$ contendrá a $Elem(\mathfrak{A})$, ya que las S-álgebras de $Elem(\mathfrak{A})$ verifican las mismas sentencias que $\mathfrak{A}$, por tanto como $\mathfrak{A}$ verifica el conjunto Φ de axiomas de $\mathcal{K}$, cualquier otro elemento de $Elem(\mathfrak{A})$ también verifica Φ. □

Corolario 4.24. *Sea S signatura, $\mathfrak{A}$ S-álgebra infinita. Entonces $\mathfrak{A}$ no es axiomatizable.*

Demostración. Por ser $\mathfrak{A}$ infinita, 4.21 nos dice que existe $\mathfrak{A}^*$ S-álgebra no isomorfa a $\mathfrak{A}$ tal que $\mathfrak{A} \equiv \mathfrak{A}^*$. Es decir, $Iso(\mathfrak{A}) \subsetneq Elem(\mathfrak{A})$, por tanto como $Elem(\mathfrak{A})$ es la menor clase axiomatizable que contiene a $\mathfrak{A}$, $Iso(\mathfrak{A})$ no es axiomatizable. □

Proposición 4.25. *Sea S signatura, $\mathcal{K}$ clase de S-álgebras y $\mathfrak{A} \in \mathcal{K}$. Si existe una S-álgebra $\mathfrak{B} \notin \mathcal{K}$ tal que $\mathfrak{A} \equiv \mathfrak{B}$, entonces $\mathcal{K}$ no es axiomatizable.*

Demostración. Supongamos que exista $\Phi \subseteq Sent_S$ tal que $\mathcal{K} = Mod(\Phi)$. Como $\mathfrak{A} \in \mathcal{K}$, $\mathfrak{A} \models \Phi$, y como $\mathfrak{B} \equiv \mathfrak{A}$, $\mathfrak{B} \models \Phi$. Entonces $\mathfrak{B} \in \mathcal{K}$, lo que es imposible. □

Ejemplo 4.26. Un cuerpo ordenado C se dice *arquimediano* si para cada $x \in C$, existe $n \in \mathbb{N}$ tal que $n > x$. $\mathbb{R}$ es cuerpo arquimediano. Veamos que la clase de los cuerpos arquimedianos no es axiomatizable.

Sea $S := \langle 0, 1; +, \cdot; < \rangle$ y sea $\mathcal{K}$ tal clase. Supongamos que existe $\Phi \subseteq Sent_S$ tal que $\mathcal{K} = Mod(\Phi)$. Consideremos:

$\Psi := \Phi \cup \{1 < c, 1+1 < c, \dots\}$, con c constante que no usamos en los axiomas.

Entonces Ψ no puede ser satisfactible. Supongamos que $\mathfrak{A}$ S-álgebra tal que $\mathfrak{A} \models \Psi$. Entonces, $\mathfrak{A} \models \Phi$, por tanto $\mathfrak{A}$ es arquimediano, pero $\mathfrak{A} \models 1 \underbrace{+ \cdots +}_{n \text{ veces}} 1 < c$ para todo n, lo cual contradice que $\mathfrak{A}$ sea arquimediano. Por tanto Ψ no es satisfactible, y por el teorema de compacidad tiene un subconjunto finito Ψ_0 insatisfactible.

Sea k el mayor entero tal que

$$\Psi_0 \subseteq \Phi \cup \{1 < c, 1 + 1 < c, \dots, 1 \underbrace{+ \cdots +}_{k \text{ veces}} 1 < c\}$$

Tomando como modelo al usual de $\mathbb{R}$ y haciendo $c = k+1$ obtenemos que Ψ_0 es satisfactible, y tenemos una constradicción. Por tanto, efectivamente, los cuerpos arquimedianos no pueden ser axiomatizados por un conjunto de sentencias de primer orden.

4.4. Teorías y enumerabilidad efectiva

4.4.1. Ideas básicas

En esta sección vamos a hacer referencia constantemente al concepto intuitivo de *algoritmo* o *procedimiento efectivo*. Un algoritmo es una serie de reglas o instrucciones de longitud finita y que se puedan ejecutar mecánicamente, es decir, de forma precisa, sin ambiguedades, no abiertas a interpretaciones. El ejemplo más claro de algoritmos son los programas que ejecutan los ordenadores.

Así, por ejemplo, *para sumar dos números naturales a y b, exprésalos como una hilera de barras, una a continuación de la otra, cada barra correspondiendo a una unidad, y cuenta el número total de barras* es un algoritmo, pero *para sumar dos números naturales a y b, reflexiona hasta llegar a la solución* no lo es.

Simbolizaremos los algoritmos mediante cajas, que representan 'máquinas' que ejecutan las instrucciones. Cuando decimos que un algoritmo da una respuesta o *output*, nos referiremos a que lo hace en finitos pasos, es decir, un algoritmo no puede darnos información que requiera ejecutar las instrucciones durante infinitos pasos.

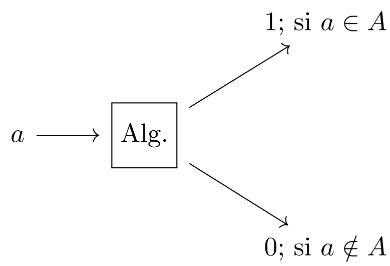
Nosotros emplearemos aquí la definición informal, y por tanto apelamos al lector para que sea especialmente cuidadoso leyendo los siguientes resultados y sus demostraciones. Hemos marcado con un asterisco todo aquello que emplee estas nociones intuitivas.[2]

***Definición 4.27.** Un conjunto A se dice *efectivamente enumerable* si existe un algoritmo que enumera sus elementos.

$$\boxed{\text{Algoritmo}} \longrightarrow a_1, a_2, \ldots \qquad (\text{Donde } A = \{a_n\}_{n\in\mathbb{N}})$$

***Definición 4.28.** Un conjunto A se dice *decidible* si existe un algoritmo el cual, dado un elemento cualquiera, nos indica si pertenece o no a A.

[2]La definición formal de algoritmo lleva inevitablemente a hablar de *programa*: el principio que establece la equivalencia de ambos conceptos es la *Tesis de Church-Turing*.



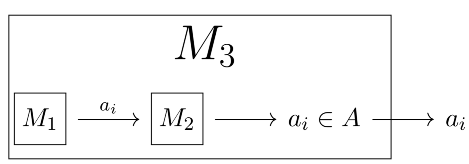
Ejemplo 4.29. Todo conjunto finito es efectivamente enumerable.

Ejemplo 4.30. El conjunto de los enteros pares es decidible.

La demostración del siguiente resultado muestra cómo realizar demostraciones empleando el concepto de algoritmo.

***Proposición 4.31.** *Sea U efectivamente enumerable. Si $A \subseteq U$ es decidible, entonces es efectivamente enumerable.*

Demostración. Sea M_1 un algoritmo que enumere U. Sea M_2 un algoritmo que decida los elementos de A. Consideremos el algoritmo M_3 que consiste en escribir en una lista aquellos elementos enumerados por M_1 que M_2 decida como elementos de A. De esta forma, M_3 enumera A. □



***Teorema 4.32.** *Sea U efectivamente enumerable, sea $A \subseteq U$. A es decidible si y solo si A y $A^{\complement}$ son efectivamente enumerables.*

Demostración.

($\Longrightarrow$) Por la anterior proposición, al ser A decidible, A es efectivamente enumerable. Ahora bien, por ser U también efectivamente enumerable, el algoritmo que, dado un elemento de la lista de U que no esté en la de A lo añade a una lista, enumera a $A^{\complement}$.

($\Longleftarrow$) Sean M_1, M_2 que enumeran, respectivamente, a A y $A^{\complement}$. Como, dado un elemento u de la enumeración de U, $u \in A$ o $u \in A^{\complement}$, se sigue que u será enumerado por M_1 o por M_2. Por tanto, el algoritmo M_3 que, decide si el elemento u está en las enumeraciones dadas por M_1 y M_2 decide si u está en A. Por tanto, A es decidible.

□

Nota 4.33. Existen conjuntos efectivamente enumerables no decidibles. Una forma de entender esto es que si un conjunto A es efectivamente enumerable por un algoritmo M_1, entonces hay un algoritmo M_2 que responde $a \in A$ si un elemento a está en A: en efecto, basta con que este algoritmo ejecute los pasos de M_1 hasta que llegue al elemento a de A, y entonces M_2 dice que $a \in A$.

Sin embargo, no tiene por qué haber un algoritmo que nos diga que un elemento b *no* está en A: no podemos ejecutar el algoritmo M_1 durante 'infinitos pasos' para que compare b a todos los elementos de A y deduzca que b no está en A. De ahí que en el teorema 4.32 necesitemos tanto que A como que $A^{\complement}$ sean efectivamente enumerables.

4.4.2. Teorema de Enumeración

Veamos los resultados que se siguen de las nociones que hemos introducido.

Durante el resto de la sección 4.4, consideraremos una signatura S y un conjunto de variables Var que sean **efectivamente enumerables**. Esto nos permitirá usar, además de algunos resultados que hemos presentado en las secciones anteriores, los teoremas de corrección y completitud.

***Lema 4.34.** *Sea S signatura efectivamente enumerable. Entonces:*

- *$TERM_S$ es efectivamente enumerable.*
- *$FORM_S$ es efectivamente enumerable.*
- *$Sent_S$ es efectivamente enumerable.*

Demostración. Se omite. □

***Lema 4.35.** *Si A es un conjunto efectivamente enumerable, $\mathcal{P}_{fin}A$, el conjunto de subconjuntos finitos de A, también lo es.*

Demostración. Se omite. □

***Proposición 4.36.** *Si la signatura S es efectivamente enumerable, el conjunto de árboles de deducción de las fórmulas asociadas es efectivamente enumerable.*

Sketch de demostración. Sabemos de los dos anteriores resultados que $Sent_S$ y $\mathcal{P}_{fin}(Sent_s)$ son efectivamente enumerables por los algoritmos a_1 y a_2, respectivamente. Tenemos por tanto las enumeraciones $a_1 : \varphi_1, \varphi_2, \ldots$ y $a_2 : \Gamma_1, \Gamma_2, \ldots$

Esbocemos el siguiente algoritmo, que consiste en un bucle. En la etapa n-ésima consideramos las secuencias del tipo $\Gamma_i \quad \varphi_j$, con $i, j \leq n$. A continuación, construimos todos los árboles de deducción de profundidad $\leq n$ que tienen por raíz a $\Gamma_i \quad \varphi_j$. Este conjunto es claramente finito.

De este modo, el conjunto de árboles de deducción es efectivamente enumerable. □

Definición 4.37. Sea S signatura. Dado $\Phi \subseteq FORM_S$, definimos la *teoría de* Φ como:

$$Th(\Phi) := \{\varphi \in Sent_S \mid \Phi \vdash \varphi\}$$

***Teorema 4.38.** *(Teorema de Enumeración) Sea S signatura efectivamente enumerable. Si $\Phi \subseteq Sent_S$ es efectivamente enumerable, $Th(\Phi)$ es efectivamente enumerable.*

Sketch de demostración. Consideremos las enumeraciones respectivas de Φ y del conjunto de los árboles de deducción, $a_1 : \varphi_1, \varphi_2, \ldots$ y $a_2 : A_1, A_2, \ldots$.

Construyamos otro algoritmo con un bucle. En la etapa n-ésima consideramos los árboles de deducción ya enumerados, de modo que cada uno de los árboles A_i tiene por raíz a una secuencia $\Gamma_i \quad \psi_i$, con $i \leq n$. De este modo, si $\Gamma_i \subseteq \{\varphi_1, \ldots, \varphi_n\}$ y A_i tiene axiomas en las hojas, incorporamos ψ_i a nuestra enumeración.

Veamos que así conseguimos enumerar $Th(\Phi)$. Sea $\sigma \in Th(\Phi)$. Entonces, $\Phi \models \sigma$ y $\Phi \vdash \sigma$. De modo que existe $\Gamma \subseteq \Phi$ finito tal que $\vdash \Gamma \quad \sigma$. Por ser finito, tenemos que existen los enteros: (i) k_1 tal que $\Gamma \subseteq \{\varphi_1, \ldots, \varphi_{k_1}\}$, (ii) k_2, que es la profundidad del árbol de deducción para $\vdash \Gamma \quad \sigma$ y (iii) k_3, que es la posición del árbol de deducción en la enumeración a_2. Tomando $k = \text{máx}\{k_1, k_2, k_3\}$, enumeramos σ en la etapa k-ésima del algoritmo que hemos descrito. □

4.4.3. Teorías

Cuando pensamos informalmente en una *teoría*, solemos referirnos a un conjunto de principios básicos y a los resultados que se obtienen de ellos y/u otros resultados. Es decir, nos referimos a un conjunto de proposiciones que es *cerrado bajo* consecuencia lógica. Esta misma idea es la que se tiene en la definición formal:

Definición 4.39. Sea S signatura. $T \subseteq Sent_S$ es *teoría* si para cada φ, si $T \models \varphi$ entonces $\varphi \in T$.

Ejemplo 4.40. Dado $\Phi \subseteq Sent_S$, $Th(\Phi)$ es teoría.

Ejemplo 4.41. Dada S signatura, $\mathfrak{A}$ S-álgebra, $Th(\mathfrak{A})$ es teoría.

***Definiciones 4.42.**

- Una teoría T es *finitamente axiomatizable* si existe $\Phi \subseteq Sent_S$ finito tal que $T = Th(\Phi)$.
- Una teoría T es *efectivamente axiomatizable* si existe $\Phi \subseteq Sent_S$ decidible tal que $T = Th(\Phi)$.
- Una teoría T es *completa* si para toda $\varphi \in Sent_S$, $\varphi \in T$ o $\neg\varphi \in T$.

Lema 4.43. *Sea S signatura. Una teoría $T \subseteq Sent_S$ es insatisfactible si y solo si $T = Sent_S$.*

Demostración. $Sent_S$ es insatisfactible, ya que contiene por ejemplo a la fórmula $\exists x \neg x \doteq x$, con lo que esa implicación es trivial. Recíprocamente, si T es insatisfactible, entonces para toda $\sigma \in Sent_S$, $T \models \sigma$, y por tanto $\sigma \in T$. □

Proposición 4.44. *Sean S signatura, $\mathfrak{A}$ S-álgebra. Entonces:*

1. *$Th(\mathfrak{A})$ es completa.*
2. *Si la teoría T es completa y $\mathfrak{A} \in Mod(T)$, entonces $T = Th(\mathfrak{A})$.*
3. *T es completa si y solo si para cualesquiera $\mathfrak{A}, \mathfrak{B} \in Mod(T)$, $\mathfrak{A} \equiv \mathfrak{B}$.*

Demostración.

(1) Sea $\varphi \in Sent_S$. Si $\mathfrak{A} \vDash \varphi$, $\varphi \in Th(\mathfrak{A})$; si $\mathfrak{A} \nvDash \varphi$, entonces $\mathfrak{A} \vDash \neg\varphi$ y $\neg\varphi \in Th(\mathfrak{A})$.

(2) Sea $\mathfrak{A} \in Mod(T)$, entonces $\mathfrak{A}$ verifica todo elemento de T, por tanto $T \subseteq Th(\mathfrak{A})$. Por otro lado, si $\varphi \in Th(\mathfrak{A})$, entonces $\mathfrak{A} \vDash \varphi$, $\mathfrak{A} \nvDash \neg\varphi$ y $\neg\varphi \notin Th(\mathfrak{A})$, de lo que se sigue que $\neg\varphi \notin T$ y $\varphi \in T$.

(3) Si T es completa y $\mathfrak{A}, \mathfrak{B} \in Mod(T)$ entonces, por el apartado (2), $Th(\mathfrak{A}) = T = Th(\mathfrak{B})$, de lo que se sigue que $\mathfrak{A} \equiv \mathfrak{B}$.

Veamos la implicación recíproca. Supongamos que T no es completa, es decir, existe $\varphi \in Sent_S$ tal que $\varphi, \neg\varphi \notin T$. Al ser T teoría, $T \nvDash \varphi, \neg\varphi$ y por tanto existen dos S-estructuras $\mathfrak{A}, \mathfrak{B} \in Mod(T)$ tales que $\mathfrak{A} \nvDash \varphi$ y $\mathfrak{B} \nvDash \neg\varphi$. Pero entonces $\mathfrak{B} \vDash \varphi$, con lo que $\varphi \notin Th(\mathfrak{A})$ y $\varphi \in Th(\mathfrak{B})$. Ergo, $Th(\mathfrak{A}) \neq Th(\mathfrak{B})$, es decir, $\mathfrak{A} \not\equiv \mathfrak{B}$.

□

En lo que queda de sección vamos a probar las siguientes implicaciones, siendo T una teoría:

$$\begin{array}{ccc} T\ decidible & & T\ finitamente\ axiomatizable \\ \Downarrow\ \Uparrow\ {\scriptstyle T\ completa} & & \Downarrow\ \not\Uparrow \\ T\ efectivamente\ enumerable & \Longleftrightarrow & T\ efectivamente\ axiomatizable \end{array}$$

***Teorema 4.45.** *Sea S signatura, T teoría. T es efectivamente axiomatizable si y solo si es efectivamente enumerable.*

Demostración.

($\Longrightarrow$) Si T es efectivamente axiomatizable, existe $\Phi \subseteq Sent_S$ decidible tal que $Th(\Phi) = T$. Por 4.36 tenemos un algoritmo que enumera los árboles de deducción, $a_1 : A_1, A_2, \ldots$.

Construyamos ahora un procedimiento que enumere T. Para ello, nos basta encontrar todas las fórmulas φ tales que existe un árbol de deducción en cuya raíz hay una secuencia $\Gamma \quad \varphi$, con $\Gamma \subseteq \Phi$ finito. En cada etapa n, consideramos el árbol A_n, cuya raíz llamamos $\Gamma_i \quad \varphi_i$. Se comprueba:

- Si A_i tiene axiomas en las hojas, lo que se puede comprobar de manera efectiva ya que se puede comprobar que el conjunto de axiomas es decidible.
- Si $\Gamma_i \subseteq \Phi$, lo que es posible comprobar mediante un algoritmo, por ser Γ_i finito y Φ decidible (basta ir tomando cada elemento de Γ_i y decidiendo si se encuentra o no en Φ).

Y en caso de que se cumplan ambas condiciones, añadimos φ a T.

($\Longleftarrow$) Al ser T efectivamente enumerable, tenemos un algoritmo que lo enumera, $a_2 : \varphi_1, \varphi_2, \ldots$. Llamando $\psi_n = \bigwedge_{i_1}^n \varphi_i$, sea $\Phi := \{\psi_n\}_{n\in\mathbb{N}}$. Nos bastará con demostrar que Φ es decidible y que $T = Th(\Phi)$. Comencemos viendo que $T = Th(\Phi)$:

- $T \subseteq Th(\Phi)$: Dada $\varphi \in T$, existe cierto n entero positivo tal que $\varphi = \varphi_n$. Por tanto, $\psi_n \vDash \varphi$ y entonces $\varphi \in Th(\Phi)$.
- $Th(\Phi) \subseteq T$: Sea $\varphi \in Th(\Phi)$. Esto implica que $\Phi \vDash \varphi$. Por el teorema de compacidad, hay un subconjunto finito $\Gamma \subseteq \Phi$ tal que $\Gamma \vDash \varphi$. Si la última fórmula de Φ que está en Γ es ψ_k, entonces $\Gamma \subseteq \{\psi_1, \ldots, \psi_k\}$. Por tanto, $\{\psi_1, \ldots, \psi_k\} \vDash \varphi$.

  Pero sabemos que $T \vDash \psi_i$ para todo i, ya que $T \supseteq \varphi_1, \ldots, \varphi_i \vDash \psi_i$. Por tanto, como $T \vDash \{\psi_1, \ldots, \psi_k\}$, tenemos que $T \vDash \varphi$, y como T es una teoría, $\varphi \in T$.

Veamos ahora que Φ es decidible. Como $\psi_i = \varphi_1 \wedge \cdots \wedge \varphi_i$, la longitud de ψ_i es $\geq i$ para todo i.

Ahora, sea $\chi \in Sent_S$ cualquiera de longitud l. Ponemos en marcha el algoritmo que enumera T, y en el paso n comprobamos si $\chi = \varphi_1 \wedge \cdots \wedge \varphi_n$. Si en para algún n se cumple esta igualdad, hemos acabado, $\chi \in \Phi$. Si llegamos a $n = l + 1$ y no se ha cumplido la igualdad en ningún paso, acabamos el algoritmo y concluímos que $\chi \notin \Phi$, ya que hemos comprobado que $\chi \neq \psi_n$ para $n = 1, \ldots, l$ y además $\chi \neq \psi_n$ para $n > l$, ya que la longitud de χ es l y la longitud de ψ_n es $> l$.

□

***Proposición 4.46.** *Sea S signatura, T teoría. Si T es efectivamente enumerable y completa, entonces es decidible.*

Demostración. Si T es insatisfactible, entonces $T = Sent_S$ y por tanto es decidible. Supongamos entonces que T es satisfactible.

Al ser efectivamente enumerable, existe el algoritmo $a_1 : \varphi_1, \varphi_2, \ldots$. Sea $\psi \in Sent_S$. Entonces, por ser T completa, $\psi \in T$ o $\neg\psi \in T$. Por tanto, en la enumeración que da a_1, terminaremos encontrando ψ o $\neg\psi$, lo que nos permite decidir si ψ está o no en T. □

Ejemplo 4.47. Veamos un ejemplo de teoría efectivamente axiomatizable pero no finitamente axiomatizable.

Consideremos el conjunto Φ_{C_0} de los axiomas de cuerpos de característica 0. Entonces Φ_{C_0} es decidible (es fácil) y por tanto $T := Th(\Phi_{C_0})$ es efectivamente axiomatizable, pero veamos que no es finitamente axiomatizable.

Si fuera finitamente axiomatizable, $T = Th(\{\varphi_1, \dots, \varphi_n\})$, entonces tendríamos que la fórmula $\varphi = \varphi_1 \wedge \cdots \wedge \varphi_n$ se cumple exactamente en los cuerpos de característica 0. Pero esto significaría que $\neg\varphi$ se cumpliría en los cuerpos de característica finita (de hecho, en cualquier estructura que no fuera un cuerpo de característica 0). Sin embargo, se puede demostrar de forma similar a 4.4 que si un conjunto de fórmulas tiene de modelos cuerpos de característica arbitrariamente grande, entonces tiene de modelo algún cuerpo de característica 0, de modo que tenemos una contradicción.

4.5. Categoricidad

Finalizamos este capítulo con uno de los conceptos más importantes de Teoría de Modelos.

La noción de categoricidad viene motivada por el papel que juegan las teorías respecto de los objetos que tratan. Es decir, a simple vista, una teoría puede tener diversos modelos, y los axiomas que la componen parecen no describir más que la forma en que se comportan ciertos elementos. Sin embargo, existen teorías de las cuales solo existe un modelo salvo isomorfismo y, por tanto, en estos casos sí que podemos afirmar que los axiomas *caracterizan* los objetos que estudian. Esto queda condensado en la siguiente

Definición 4.48. Sea S signatura. Una teoría T se dice *categórica* si $\mathfrak{A} \approx \mathfrak{B}$ para cualesquiera S-álgebras $\mathfrak{A}, \mathfrak{B} \in Mod(T)$.

Proposición 4.49. *Sea S signatura, T teoría. Si T es categórica, es completa.*

Demostración. Sean $\mathfrak{A}, \mathfrak{B} \in Mod(T)$ S-álgebras. Entonces, por categoricidad, $\mathfrak{A} \approx \mathfrak{B}$, y por tanto, $\mathfrak{A} \equiv \mathfrak{B}$. Como esto se cumple para cualesquiera $\mathfrak{A}, \mathfrak{B} \in Mod(T)$, por 4.44 T es completa. $\square$

Proposición 4.50. *Sea S signatura, sea T teoría. Si T tiene modelos infinitos, no es categórica.*

Demostración. Sea la S-álgebra infinita $\mathfrak{A} \models T$ y llamamos $\kappa = |\mathfrak{A}|$. Por ser κ infinito, 4.21 nos dice que existe una S-álgebra $\mathfrak{A}^*$ no isomorfa a $\mathfrak{A}$ y tal que $\mathfrak{A} \equiv \mathfrak{A}^*$. Por tanto, $\mathfrak{A}^* \models T$, y T no es categórica. $\square$

Esta es una proposición bastante sorprendente que nos hace darnos cuenta de las limitaciones de la lógica de primer orden. Nos dice que cualquier estructura infinita en la que pensemos, por ejemplo los naturales $\mathbb{N}$ con su buen orden, el grupo de los enteros $\mathbb{Z}$, o el cuerpo ordenado $\mathbb{R}$, ninguna de ellas puede *caracterizarse* mediante axiomas de primer orden.

Definición 4.51. Sea S signatura. Una teoría T se dice *κ-categórica* si $\mathfrak{A} \approx \mathfrak{B}$, para cualesquiera S-álgebras $\mathfrak{A}, \mathfrak{B} \in Mod(T)$ tales que $|\mathfrak{A}| = |\mathfrak{B}| = \kappa$.

Teorema 4.52. *Sea S signatura. Si T es una teoría satisfactible, sin modelos finitos y κ-categórica para algún cardinal $\kappa \geq |FORM_S| \geq \aleph_0 = |\mathbb{N}|$, entonces es completa.*

Demostración. Supongamos que T no es completa. Entonces existe $\sigma \in Sent_S$ tal que $\sigma, \neg\sigma \notin T$, es decir, tal que $T \not\models \sigma, \neg\sigma$ y entonces $T \cup \{\sigma\}$ y $T \cup \{\neg\sigma\}$ son satisfactibles con modelos infinitos (ya que T no tiene modelos finitos) y por 4.10 existen dos S-álgebras $\mathfrak{A}, \mathfrak{B}$ tales que $\mathfrak{A} \models T \cup \{\sigma\}$ y $\mathfrak{B} \models T \cup \{\neg\sigma\}$ y de modo que $\kappa = |\mathfrak{A}| = |\mathfrak{B}|$. Entonces, al ser $\mathfrak{A} \models T$ y $\mathfrak{B} \models T$, y por κ-categoricidad, $\mathfrak{A} \approx \mathfrak{B}$, lo que es imposible, ya que $\mathfrak{A} \models \sigma$ y $\mathfrak{B} \models \neg\sigma$. $\square$

Corolario 4.53. *Sea S signatura numerable. Si T es una teoría satisfactible, sin modelos finitos y κ-categórica para algún cardinal $\kappa \geq \aleph_0$, entonces es completa.*

Demostración. El caso con S numerable del teorema anterior $\square$