

LÓGICA COMPUTACIONAL

CAPÍTULO 1: LÓGICA DE PROPOSICIONES

DAVID RODRÍGUEZ HERNÁNDEZ

FECHA DE REVISIÓN: **08 – Octubre - 2007**

ZAMORA (CURSO 2007/2008)

david.rgh@gmail.com

Nota importante:

Este documento no pretende reemplazar al material propuesto por la UNED para la asignatura Lógica Computacional.

Su finalidad es presentar de una forma esquematizada los contenidos de la asignatura, para facilitar el estudio de la misma. Es conveniente disponer de la bibliografía propuesta por la Universidad para su estudio completo.

Cualquier sugerencia, comentario o corrección sobre este documento, envíelo a david.rgh@gmail.com para poder realizar los cambios necesarios.

Nota adicional:

Este documento no está finalizado. Aconsejo revisar periódicamente el portal **WAINU** (<http://wainu.ii.uned.es>) para actualizaciones.

David Rguez. Hdez.

Alumno de 2º ciclo Ingeniería Informática (UNED)

SINTAXIS:

Alfabeto: Conjunto de símbolos. Puede ser finito o infinito:

Finito $\rightarrow A_1 = \{\$, s, 5\}$ Infinito $\rightarrow A_2 = \{p_1, p_2, p_3, p_4 \dots\}$

Expresión: Secuencia FINITA de símbolos. Si se compone sobre el abecedario A_1 entonces se dice que es una expresión sobre el alfabeto A_1 . A los símbolos se les puede llamar también caracteres y a las expresiones, cadenas.

Al conjunto infinito de expresiones (finitas) sobre A se le nombra A^* .

Lenguaje: Es el conjunto de expresiones. Si es sobre A, entonces también lo es sobre cualquier subconjunto de A^* (incluyendo al propio A^*). Puede ser infinito, pero mencionando alguna propiedad común (por ejemplo: "contiene al menos una "s").

Lenguaje de lógica de proposiciones:

Proporciona los símbolos necesarios para representar proposiciones sobre el mundo. Requiere tener letras proposicionales.

Alfabeto de la lógica de proposiciones: Comprende las letras proposicionales, los símbolos lógicos y los símbolos de puntuación.

$A = \{p_0, p_1, \dots, \perp, \top, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\}$ Si hay pocas letras proposicionales, se pueden utilizar p,r,s,t...

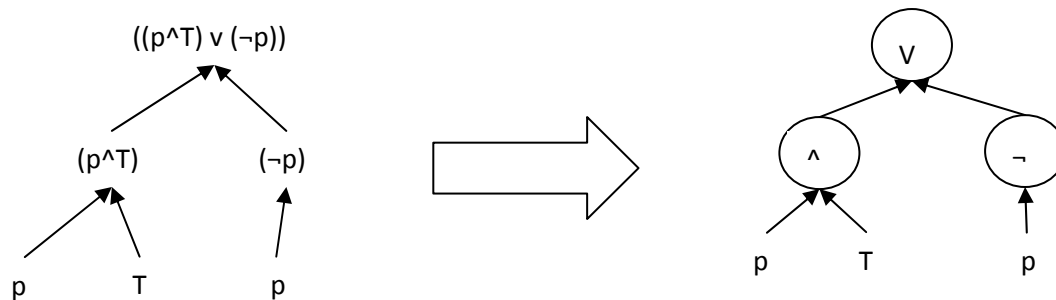
Se leen de la siguiente forma:

\perp	Falso	false
\top	Verdadero	true
$\neg p$	no p	$\sim p$
$p \wedge q$	p y q	p&q
$p \vee q$	p o q	
$p \rightarrow q$	si p, entonces q	$p \supset q$ $p \Rightarrow q$
$p \leftrightarrow q$	p si, y solo si, q	$p \equiv q$ $p \Leftrightarrow q$

Fórmulas: Son las expresiones que consideramos como aceptables en el lenguaje.

Form: Conjunto infinito de fórmulas.

Árboles sintácticos: Representaciones gráficas del desglose (o composición) de una fórmula.



Fórmula atómica: Expresión compuesta por una letra proposicional o por constantes (verdadero/falso).

Un conjunto X de fórmulas verifican que:

1. Las fórmulas atómicas pertenecen a X.
2. Si una fórmula no atómica pertenece a X, entonces su negación también.
3. Si dos fórmulas no atómicas pertenecen a X, entonces su composición por una conectiva también.

Lenguaje de lógica de proposiciones: (conjunto Form) Contiene todas las fórmulas de nuestro lenguaje (SÓLO ESTAS) y es el menor de los conjuntos que verifican las propiedades anteriores.

Metalenguaje: Letras que representan fórmulas completas. Se suele utilizar el alfabeto griego.

Principio de inducción estructural (demostración inductiva):

1. Caso base: Toda fórmula atómica tiene la propiedad P.
2. Si una fórmula tiene P, su negación también.
3. Si dos fórmulas tienen P (ambas), entonces su composición por una conectiva también.

Análisis sintáctico único: Cada fórmula proposicional pertenece a una Y SOLO UNA de estas categorías:

- Es atómica.
- Es una negación de otra fórmula única.
- Es una composición de dos fórmulas únicas para una determinada conectiva.

Si se trata de la tercera categoría, cada una de esas dos fórmulas se denominan subfórmulas inmediatas (no necesariamente atómicas).

Definiciones recursivas: Son aquellas definiciones que se basan sobre las aportaciones de las subfórmulas inmediatas.

Principio de Recursión Estructural: Para una determinada elección de las funciones previas, la función resultante es única y está bien definida.

Ej: $f_{at} : Form_Atom \rightarrow X$ $f_{\neg} : X \rightarrow X$ $f_{*} : X^2 \rightarrow X$ (para cada conectiva binaria).

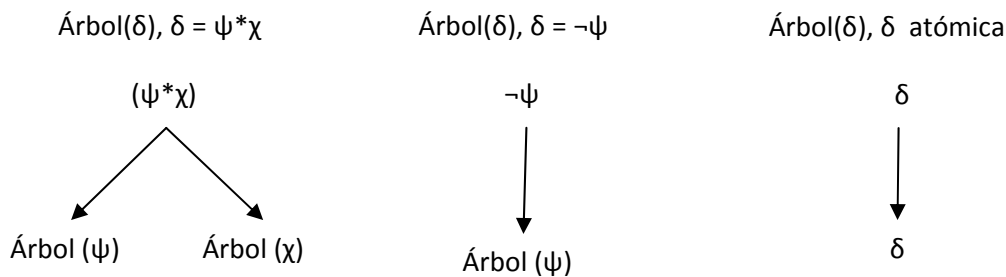
Rango: Proporciona la longitud de la mayor rama del árbol sintáctico de la fórmula.

Número de símbolos: Proporciona el número de fórmulas atómicas de la fórmula.

Número de aparición de conectivas: Proporciona el número de veces que aparece una determinada conectiva en la fórmula.

Principio de inducción completa: Si se demuestra que para una fórmula dada, todas las fórmulas con rango menor que el rango de dicha fórmula tienen la propiedad P, entonces esa fórmula también tiene la propiedad P.

Definición recursiva de árbol sintáctico:



Subfórmulas (Subform): Es el conjunto de TODAS sus subfórmulas.

Sustitución uniforme: Permite escribir una fórmula a partir de otra estableciendo una equivalencia para cada letra proposicional. Es muy útil. Se debe aplicar UNIFORMEMENTE la PRIMERA vez de forma SIMULTÁNEA (no se puede aplicar primero unas y luego otras, ya que podría dar lugar a bucles cerrados).

Instancia, por sustitución, de una fórmula:

Si tenemos φ y $\sigma_{atom} : Form_Atom \mapsto Form$ (sustitución para fórmula atómica) entonces la sustitución uniforme será:

$$(\varphi)^\sigma = \begin{cases} ((\psi)^\sigma * (\chi)^\sigma); & \text{para } \varphi = (\psi * \chi) \\ (\neg(\psi)^\sigma); & \text{para } \varphi = (\neg\varphi) \\ \sigma_{atom}(\varphi); & \text{para } \varphi \text{ atómica} \end{cases}$$

Si se trata de una constante, su instancia por sustitución de σ es ella misma.

Si no se define sustitución para todas (infinitas) las letras proposiciones, entonces la instancia para aquella letra no definida, es ella misma.

Eliminar paréntesis: La eliminación de paréntesis puede suponer ambigüedades en algunas fórmulas. Para organizarlos, tenemos 2 posibilidades:

- **Orden de precedencia:** Se establecen prioridades para los distintos símbolos. Lo más usual es (de mayor prioridad a menor prioridad): Negación, conjunción/disyunción y condicional/bicondicional.
- **Notación prefija:** Al igual que en una operación matemática podemos pasar de la notación infija a notación prefija (o notación polaca):

$$(3 + 5) \rightarrow (+ 35)$$

También lo podemos pasar en una fórmula proposicional:

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

SEMÁNTICA

Sistema lógico: Se utilizan para representar declaraciones sobre el mundo y para operar sobre dichas representaciones.

Expresión declarativa: Enuncia un estado de cosas. Por ejemplo, “*llueve*”. Una expresión declarativa **NO** es una pregunta ni una orden. A partir de dos o más expresiones declarativas, se pueden construir nuevas. Ejemplo: “*llueve*” y “*hace frío*”.

Los sistemas lógicos representan de una forma formal estas expresiones declarativas. Es un lenguaje pobre que intenta representar uno rico y variado, por lo que tiene su dificultad. El ejemplo anterior se representaría como $p \wedge q$

Cada expresión declarativa tiene un grado de verdad, y su conjunto tiene un grado de verdad que es dependiente de los anteriores. La lógica proposicional intenta captar las dependencias entre los valores de verdades según la conectiva.

Así pues, se puede averiguar el valor de verdad de expresiones complejas a partir de los valores de sus componentes.

Existen cuatro cuestiones muy importantes sobre los valores de verdad:

- **Satisfabilidad:** Una fórmula es satisfactible si, dada una interpretación, todas las declaraciones son verdaderas. Si no existe ninguna interpretación que lo cumpla, es **insatisfactible**.

- **Validez:** Aquellas expresiones que SIEMPRE son verdaderas.
- **Consecuencia lógica:** Algunas declaraciones pueden ser deducibles a partir del resto.
- **Equivalencia:** Algunas declaraciones son equivalentes. Ej: $p \rightarrow q$ es equivalente a $(\neg q) \rightarrow (\neg p)$

Valores de verdad de fórmulas atómicas: Existen dos valores: verdadero y falso (1 y 0).

Asignación: Es una función del conjunto de fórmulas atómicas en el conjunto de valores de verdad, con $v_{atom}(\perp) = 0$ y $v_{atom}(T) = 1$

En otras palabras, una asignación es una combinación de valores para el conjunto de letras proposicionales. Para n letras existen 2^n asignaciones.

Valores de verdad de las conectivas binarias básicas:

p	q	$p \wedge q$	$p \vee q$	$p \rightarrow q$	$p \leftrightarrow q$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Valores de verdad de fórmulas complejas: El valor de verdad de una fórmula compuesta depende sólo del valor de sus subfórmulas inmediatas y de su conectiva.

Toda fórmula puede evaluarse y las complejas tienen sus correspondencias con valores de verdad. Se descartan aquellas correspondencias que evalúen como verdadera a una fórmula y a su negación.

Interpretación: Es una correspondencia aceptable. $v : Form \mapsto \{0,1\}$ es una función que asigna a cada fórmula un valor de verdad y verifica lo siguiente:

- $v(\perp) = 0$
- $v(T) = 1$
- $v(\neg \ell) = \neg(v(\ell))$
- $v(\ell * \psi) = v(\ell) * v(\psi)$ (para cada conectiva binaria).

Aplicando el principio de recursión estructural:

$$v(\varphi) \begin{cases} v_*(\varphi) = v(\psi) * v(\chi), \text{ para } \varphi = (\psi * \chi) \\ v_{\neg}(\varphi) = \neg v(\psi), \text{ para } \varphi = (\neg \psi) \\ v_{atom}(\varphi), \text{ para } \varphi \text{ atómica} \end{cases}$$

Si el alfabeto tiene muchas letras proposicionales y sólo se utilizan unas pocas, existirán varias asignaciones que coincidirán para esas letras (aunque diferirían en el resto) y todas darán el mismo resultado, ya que las fórmulas sólo dependen de esas pocas.

Si $v_1(p_k) = v_2(p_k)$ para todo $p_k \in \varphi$ entonces $v_1(\varphi) = v_2(\varphi)$

Tablas de verdad: Estas tablas sirven para conocer cómo se comporta una fórmula globalmente. Contiene todas las asignaciones posibles. La estructura es como la presentada anteriormente para las conectivas básicas. Puede compactarse teniendo en mente la figura del árbol.

Tautologías: Aquellas fórmulas que son verdaderas para toda interpretación.

Contradicciones: Aquellas fórmulas que son falsas para toda interpretación.

Contingentes: Aquellas fórmulas que toman valores verdaderos o falsos para distintas interpretaciones.

CONCEPTOS SEMÁNTICOS BÁSICOS

Satisfacibilidad: Una fórmula es satisfacible si existe al menos una interpretación que la haga verdadera. v satisface a φ si $v(\varphi_k) = 1$ para todo $\varphi_k \in \Phi$, siendo $\Phi = \{\varphi_1, \dots, \varphi_n\}$

Si no existe ninguna interpretación posible, entonces es insatisfacible.

Si el conjunto Φ es satisfacible: si eliminamos una fórmula sigue siendo satisfacible; si añadimos una tautología sigue siendo satisfacible; si añadimos una contradicción, será insatisfacible.

Si el conjunto Φ es insatisfacible: si añadimos una fórmula o eliminamos una tautología, seguirá siendo insatisfacible.

El resto de casos posibles, el resultado es indeterminado.

Para demostrar si una fórmula es satisfacible, basta con encontrar UNA interpretación que cumpla la condición. Para demostrar si no lo es, hay que comprobar TODAS las interpretaciones. Con pocas letras no es problema; pero si hay, por ejemplo, 30 letras, existirán 1.073.741.824 interpretaciones distintas.

Φ es satisfacible si y solo si $(\varphi_1 \wedge \dots \wedge \varphi_n)$ es satisfacible.

Φ es insatisfacible si y solo si $(\varphi_1 \wedge \dots \wedge \varphi_n)$ es una contradicción.

Validez: Una fórmula es válida si es verdadera frente a cualquier interpretación. Las **tautologías** siempre son válidas. La validez divide a las fórmulas en *fórmulas tautológicas* y *fórmulas contingentes*. Es conveniente tener en cuenta las siguientes observaciones:

- Si se niega una fórmula insatisfacible resulta una fórmula tautológica.
- Si se niega una fórmula tautológica resulta una fórmula insatisfacible.
- Si se niega una fórmula contingente resulta otra fórmula contingente.
- Si se niega una fórmula satisfacible, **NO** resulta una fórmula tautológica (puede ser - tanto satisfacible como insatisfacible).

Notación \rightarrow Para indicar que una fórmula φ es válida, se realiza así: $\models \varphi$ (la barra vertical y las dos horizontales van juntas; en este documento no puedo reproducir el símbolo exacto).

Para decidir si una fórmula es válida o no, hay que recorrer toda la tabla de verdad. Si detectamos una sola interpretación que no satisfaga la fórmula, entonces no es válida. Si TODAS las interpretaciones la satisfacen, entonces sí es válida.

$\models \varphi \leftrightarrow \neg \varphi$ es insatisfacible.

Preservación por sustitución: Gracias a la **sustitución uniforme**, podemos preservar la validez sustituyendo cada componente por su correspondiente. Ejemplo:

$\varphi = ((p \wedge q) \rightarrow (q \vee r))$ es una tautología. Si sustituimos de esta forma:

$\sigma(p) = p \quad \sigma(q) = (p \wedge r) \quad \sigma(r) = (\neg r)$ entonces

$(\varphi)^\sigma = ((p \wedge (p \wedge r)) \rightarrow ((p \wedge r) \vee (\neg r)))$ TAMBIÉN es una tautología.

Consecuencia lógica: Una fórmula ψ es consecuencia de un conjunto $\Phi = \{\varphi_1, \dots, \varphi_n\}$ si toda interpretación que satisface a Φ también satisface a ψ . Si un conjunto de hipótesis son verdaderas, entonces su consecuencia lógica también lo es.

Notación $\rightarrow \Phi \models \psi$ (es consecuencia lógica) $\Phi \not\models \psi$ (no es consecuencia lógica).

Ejemplo: Si tenemos la fórmula $p \vee q$, podemos ver que ésta es consecuencia lógica de p y de q , ya que cuando ambas tienen valor 1, la fórmula también: $\{p, q\} \models (p \vee q)$

Para comprobar la consecuencia lógica, hay que comprobar una por una cada interpretación. Si AL MENOS UN interpretación satisface al conjunto, PERO NO a ψ , entonces la consecuencia tampoco está satisfecha. El procedimiento es el siguiente:

1. Construir tabla de verdad.
2. Señalar líneas verdaderas de cada hipótesis $(\varphi_1, \varphi_2, \dots, \varphi_n)$.

3. Determinar intersección entre ellas.
4. Valorar la consecuencia lógica en esas líneas comunes. Si es verdadera en todas ellas, entonces $\{\varphi_1, \varphi_2, \dots, \varphi_n\} \models \psi$

La consecuencia lógica tiene las siguientes propiedades:

- Reflexividad: $\varphi \models \varphi$
- Transitividad: Si $\Phi \models \psi$ y $\psi \models \chi$ entonces $\Phi \models \chi$
- Monotonía: Sea $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ Si $\Phi \models \psi$ y $\Phi \subset \Omega$ entonces $\Omega \models \psi$

Consecuencias lógicas de conjuntos insatisfactibles: Aunque pueda parecer contradictorio, cualquier fórmula puede ser consecuencia lógica de un conjunto insatisfactible, teniendo en cuenta que la definición SÓLO alude a las interpretaciones que satisfacen el conjunto (si no existe ninguna, no por ello deja de cumplirse la condición; puesto que TODAS esas interpretaciones (en este caso, 0) satisfacen también la fórmula (como hay 0 interpretaciones, consideramos que la condición se cumple)).

Si un conjunto es satisfactible y una fórmula es su consecuencia lógica, entonces la negación de dicha fórmula no lo es.

Consecuencia lógica y validez: La fórmula condicional construida con los conjuntos de todas las hipótesis como antecedente y la consecuencia como consecuente, es SIEMPRE una tautología. ¿Por qué? La fórmula sólo sería falsa si el antecedente fuera verdadero y el consecuente falso. Pero dicha situación no se dará, ya que partimos de que el consecuente es una **consecuencia lógica** del antecedente. Por tanto:

$$(\{\varphi_1, \dots, \varphi_n\}) \leftrightarrow ((\varphi_1, \dots, \varphi_n) \rightarrow \psi \text{ es tautología})$$

Consecuencia lógica y satisfabilidad: Si un conjunto de fórmulas tiene a ψ como consecuencia lógica, entonces ese conjunto más la negación de ψ es un conjunto **insatisfactible** (y viceversa) (recuérdese que un conjunto es insatisfactible si la **conjunción** de las fórmulas que lo componen es también insatisfactible).

De consecuencia lógica a insatisfactibilidad: Teniendo en cuenta lo anterior:

Si $\{\varphi_1, \varphi_2, \varphi_3\} \models r$ entonces $\{\varphi_1, \varphi_2, \varphi_3, \neg r\}$ (o, lo que es lo mismo, $\varphi_1 \wedge \dots \wedge \varphi_n \wedge \neg r$) es insatisfactible.

De insatisfactibilidad a consecuencia lógica: Siguiendo la directriz anterior:

Si $\{\varphi_1, \dots, \varphi_n, \neg r\}$ es insatisfactible, entonces $\{\varphi_1, \dots, \varphi_n\} \models r$

Combinando ambos apartados, tenemos que si un conjunto es insatisfactible, cualquier componente es consecuencia lógica del conjunto restante.

Equivalencia:

Se dice que dos fórmulas φ y ψ son equivalentes si cada una es consecuencia lógica de la otra; es decir, debe cumplir que $\varphi \models \psi$ y que $\psi \models \varphi$. Esto implica que ambas fórmulas tienen exactamente la misma tabla de verdad; es decir, son *dos formas de expresar 'lo mismo'*.

Notación $\rightarrow \varphi \equiv \psi$ (son equivalentes) $\varphi \not\equiv \psi$ (no son equivalentes (**cuidado:** el símbolo es con **tres** líneas horizontales, no con dos como aparece en este documento)).

La equivalencia tiene las siguientes propiedades:

- Reflexividad: $\varphi \equiv \varphi$
- Simetría: $(\varphi \equiv \psi) \rightarrow (\psi \equiv \varphi)$
- Transitividad: Si $\varphi \equiv \psi$ y $\psi \equiv \chi$ entonces $\varphi \equiv \chi$

Las equivalencias dividen las fórmulas en **clases** (grupos de fórmulas equivalentes). Las clases se nombran como C_k , siendo 'k' el número correspondiente al binario leído en columna.

Para dos letras proposicionales hay un total de 16 clases posibles. Todas y cada una de las fórmulas que se pueden formar con esas dos letras están catalogadas en una de esas 16 clases. Toda clase tiene su complementaria (aquella en la que 1's y 0's se intercambian en el número binario).

Cualquier fórmula de las 16 clases puede expresarse con las 4 conectivas básicas.

Conjunto de conectivas completo: Sobre dicho conjunto se puede representar cualquier otra conectiva.

Ejemplos: $\{\neg, \wedge\}, \{\neg, \vee\}, \{\neg, \rightarrow\}, \{\uparrow\}, \{\downarrow\}$

Tabla de equivalencias básica: Puede consultarse en la página 38 del libro básico.

Situación y reemplazo: Tomemos que φ y ψ son equivalentes si y sólo si $\varphi \leftrightarrow \psi$ es una tautología. Si se aplica la misma sustitución uniforme, entonces sus sustituciones son también equivalentes entre sí.

Esto es así porque la sustitución uniforme **preserva la validez**. Es decir, si $\varphi \leftrightarrow \psi$ es válida (recordemos que las tautologías SIEMPRE son válidas), entonces $(\varphi \leftrightarrow \psi)^\sigma$ también es válida, por lo que $\varphi^\sigma \leftrightarrow \psi^\sigma$ sigue siendo una tautología y, atendiendo a la definición inicial:

$$\varphi^\sigma \equiv \psi^\sigma$$

Reemplazo: Sean φ y ψ dos fórmulas equivalentes. Si φ es parte de una fórmula X (ya sea con una o más apariciones) y reemplazamos una o varias de dichas apariciones (no necesariamente todas por ψ , entonces la fórmula resultante será equivalente a la inicial.

Formas normales: Se dividen en 3 clases:

- **Disyuntiva:** Si es de la forma $\varphi_1 \vee \dots \vee \varphi_n$ donde cada componente es una conjunción de literales (un literal es una letra proposicional o su negación). Es contradictorio si y sólo si cada conjunción suya contiene un literal y su negación simultáneamente. Ej: $(p \wedge \neg p) \vee (\neg q \wedge q)$ es contradicción.
- **Conjuntiva:** Si es de la forma $\varphi_1 \wedge \dots \wedge \varphi_n$ donde cada componente es una disyunción de literales. Es tautología si y sólo si cada disyunción suya contiene un literal y su negación simultáneamente. Ej: $(p \vee \neg p) \wedge (\neg q \vee q)$ es tautología.
- **Clausulada:** Cada componente de una conjunción la denominamos **cláusula** (sea literal o sea un conjunto). Recordando que los grupos de conjunciones o disyunciones pueden expresarse como conjuntos, tenemos:

$\varphi_1 \wedge \dots \wedge \varphi_n$ donde cada cláusula $\varphi_k = (\varphi_k^1 \vee \dots \vee \varphi_k^m)$ entonces se puede expresar como $\{\varphi_1, \dots, \varphi_k\} = \{\{\varphi_1^1, \dots, \varphi_1^m\}, \dots, \{\varphi_k^1, \dots, \varphi_k^m\}\}$

Equivalencia, consecuencia lógica, validez y satisfactibilidad: La equivalencia tiene conexiones formales con el resto de términos:

- $\varphi \equiv \psi$ si y sólo si $\varphi \models \psi$ y $\psi \models \varphi$
- $\varphi \equiv \psi$ si y sólo si $(\varphi \rightarrow \psi)$ y $(\psi \rightarrow \varphi)$ son tautologías.
- $\varphi \equiv \psi$ si y sólo si $(\varphi \leftrightarrow \psi)$ es tautología (deducción de la anterior).
- $\varphi \equiv \psi$ si y sólo si $(\neg(\varphi \leftrightarrow \psi))$ es insatisfactible.