**Texto

Descripción generada automáticamente con confianza bajaFacultad de Filosofía,**

**Educación y**

**Ciencias Humanas**

**Unidad 2**

**Sintaxis y semántica de lógica de conectores (LC)**

**1. Sintaxis de LC**

|  |  |  |
| --- | --- | --- |
| **Alfabeto**  Conjunto completo de símbolos que componen el lenguaje de la *LC* | | |
| **Nombre de los símbolos** | | **Símbolos** |
| Letras oracionales | | *P, Q, R, S, T, P’, Q’, R’, S’, T’, P’’, Q’’, …* |
| Conectores u operadores lógicos | Negación |  |
| Conjunción |  |
| Disyunción |  |
| Condicional |  |
| Bicondicional |  |
| Símbolos de jerarquía | |  |
| Separador de premisas | | , |
| Indicador de conclusión | |  |

|  |
| --- |
| **Fórmulas en LC**  Secuencias de símbolos que pueden formarse con los símbolos del alfabeto de la LC a través de la aplicación recursiva de reglas de interpretación del lenguaje |
| 1. **Fórmula atómica**   Formada por una sola letra oracional. Por ejemplo:  *P, Q’, R, S’’,* etc.   1. **Fórmula compuesta**   Compuesta por una o más letras oracionales y, al menos, un conector lógico. Por ejemplo:  , etc. |

|  |
| --- |
| **Reglas de formación de fórmulas en *LC***  Conjunto de reglas que para la construcción de fórmulas bien formadas en LC (fbf). |
| 1. Toda letra oracional es una fbf. 2. Si es una fbf, entonces es una fbf. 3. Si y son fbf’s, entonces son fbfs. ***[1]*** 4. Nada más es una fbf. ***[2]*** |

***[1]*** Cada introducción de un operador diádico – ­– debe venir acompañada por los símbolos de jerarquía – ( ) – para señalar su alcance, es decir, las dos fórmulas sobre las que opera. En cambio, la introducción del operador monádico – – señala que su alcance se aplica solo a la fórmula que está inmediatamente a su derecha y no demanda la introducción de paréntesis. Los nombres diádico y monádicoindican la cantidad de fórmulas sobre las que opera cada conector.

***[2]*** Toda secuencia de símbolos de LC que no siga *rf1-rf4* para construirse, será una expresión sin sentido en LC. No se le atribuye, ni siquiera, el nombre de fórmula. Por ello, cuando se hable de fórmulas se estará hablando exclusivamente de fbfs.

**2. Árboles sintácticos**

Si la formación toda fbf en LCdebe seguir las reglas *rf1-rf4*, entonces es posible rastrear todos los pasos seguidos en ella, en tanto serán aplicaciones de alguna de dichas reglas.

|  |
| --- |
| ***Def. 1. Árbol sintáctico***  Representación retrospectiva y única de la construcción de una fórmula |

A continuación, se detallará cómo se elabora el árbol sintáctico de la fórmula:

|  |
| --- |
| **Paso 1.** Determinar el **operador principal** de la fórmula   * Si, partiendo desde la izquierda, el primer símbolo de la fórmula es una **negación**, este es el operador principal (**ojo**: los paréntesis cuentan como símbolos) * Si no, el operador principal será un operador **diádico**. Para determinarlo hay que contar, desde la izquierda, los paréntesis de apertura – **(** – y los paréntesis de cierre – **)** –: **cuando el número de paréntesis de apertura sea mayor por uno al de paréntesis de cierre** (1-0, 2-1, etc**.**)**, el primer operador diádico que aparezca a la derecha será el operador principal**. |
| En el ejemplo, el operador principal es la “”. |
| **Paso 2.** Eliminar el operador principal   * Si el operador principal es la **negación**, se debe eliminar y consignar la fórmula resultante encima de la fórmula original. Entre ellas, se debe trazar una línea horizontal que indica el alcance de la negación. * Si el operador principal es **diádico**, se debe eliminar **junto con los paréntesis externos** de la fórmula analizada. A continuación, hay que consignar las dos fórmulas resultantes, en extremos opuestos, encima de la fórmula analizada. Finalmente, se debe trazar una línea horizontal debajo de estas dos fórmulas y sobre la fórmula analizada para indicar el alcance del operador principal de esta. |
| En el ejemplo, la eliminación del operador diádico “” sería así:    Esta representación retrospectiva muestra que el último paso en la construcción de la fórmula original fue la aplicación de *rf3* para introducir el operador principal “” |
| **Paso 3.** Si todas las fórmulas resultantes son fórmulas atómicas, el árbol sintáctico está terminado. Si no, se deberá aplicar nuevamente los pasos 1 y 2 con cada fórmula resultante o, como se llamará a partir de ahora, subfórmula**.** |
| En el ejemplo, ninguna de las subfórmulas que se alcanzan en el paso 2 es atómica, así que se aplicará nuevamente los pasos 1 y 2 con cada una de ellas:      Para llegar las subfórmulas “” y “” se aplicó *rf2*. Sin embargo, la subfórmula aún no es atómica, así que se aplicará los pasos 1 y 2 con ella:        Para llegar a la subfórmula “” se aplicó *rf3.* Las subfórmulas alcanzadas son ahora, en su totalidad, fórmulas atómicas y, por tanto, son fbfs por *rf1.* No hay más pasos que aplicar, así que este es el árbol sintáctico de “”. |

Ahora se aplicará los mismos pasos a la siguiente fórmula, pero más intuitivamente:

**Paso 1.** El operador principal es la “”

**Paso 2.** Se elimina la “”:

**Paso 3:** las subfórmulas “” y“” no son atómicas, así que se aplicarán los **pasos 1 y 2** en cada una de ellas:

**Paso 3 (otra vez):** las subfórmulas “”, “” y “” no son atómicas, así que se aplicarán los **pasos 1 y 2** en cada una de ellas:

**Paso 3 (una vez más):** las subfórmulas “” y “” no son atómicas, así que se aplicarán los **pasos 1 y 2** en cada una de ellas:

**Paso 3 (por última vez):** todas las subfórmulas resultantes son atómicas: el árbol sintáctico de “”está completo.

Finalmente, un ejemplo en el que todos los pasos ya se aplicaron:

*P*

En los ejercicios para determinar el árbol sintáctico de una fórmula en LCse pedirá consignar, adicionalmente, el operador principal de . En este caso, el operador principal es . En este punto, cabe señalar que la cantidad total de subfórmulas de es la cantidad de fórmulas que aparecen en su árbol semántico, incluyendo a . En el último árbol, queda en evidencia que tiene 16 subfórmulas.

Ahora bien, si no es posible aplicar los pasos 1 o 2 a cualquier subfórmula en el desarrollo de un árbol sintáctico, esto significará que la secuencia de símbolos que se está analizando está mal organizada. Así, desarrollar un árbol sintáctico para una secuencia de símbolos de LC permitirá determinar, también, si esta es o no una fórmula en LC. A continuación, se decidirá si la siguiente secuencia de símbolos es una fbf o no:

**Paso 1:** el operador principal de la fórmula es la “”

**Paso 2:** eliminemos la “”:

**Paso 3:** Las subfórmulas “” y “” no son atómicas; por ello, se intentará aplicar lospasos 1 y 2 con cada una de ellas:

Las subfórmulas “” y “” son atómicas. Por su lado, la subfórmula “” no es una fórmula atómica y, por tanto, no se puede decir que el árbol esté terminado. Sin embargo, tampoco es una fórmula compuesta y, por ello, no puede analizarse más siguiendo los pasos 1 y 2. Por ello, la fórmula “” no es una fbf.

Finalmente, se hará un ejemplo de este tipo de ejercicio de manera directa. Se determinará si “” es una fbf o no:

La subfórmula “” no es atómica ni puede analizarse por los pasos 1 y 2. Por ello, “” no es una fbf. En este tipo de ejercicios no será necesario consignar el operador principal de la fórmula original. Basta con concluir que la secuencia no es una fbf.

**3. Grados de complejidad**

Se puede crear una escala de complejidad sintáctica para las fórmulas de la LC siguiendo estas indicaciones:

* Una fórmula atómica tiene siempre grado de complejidad 0.
* Una fórmula compleja formada a través de una negación antecedida a una fórmula de grado de complejidad tiene un grado de complejidad
* Una fórmula compleja formada a través de la vinculación, por medio de un conector diádico, de dos fórmulas de grados de complejidad y , respectivamente, tiene grado , si , o , si .

**Ejemplos**

|  |  |
| --- | --- |
| **Fórmulas de grado 0** |  |
| **Fórmulas de grado 1** |  |
| **Fórmulas de grado 2** |  |
| **Fórmulas de grado 3** |  |
| **Fórmulas de grado 4** |  |

Haciendo uso del alfabeto y las reglas de formación, se puede elaborar fórmulas de cualquier grado de complejidad en LC.

**4. Semántica de LC**

La sintaxis expuesta contiene los conectores suficientes para que, al interpretarla en términos funcionales, ella exprese todas las funciones diádicas posibles para un rango de dos valores, y . Por ello, el conjunto de los cinco conectores expuesto es llamado funcionalmente adecuado y la LC es un sistema funcionalmente completo[[1]](#footnote-1). Esta es la razón por la que, como se verá en la unidad siguiente, es posible programar un método (tablas de verdad) para calcular mecánicamente todos los argumentos válidos (o inválidos) y todos los conjuntos de oraciones consistentes de la LC. Sin embargo, este material se ceñirá exclusivamente a dos tareas: la interpretación del lenguaje formal descrito en la sección anterior según una semántica de valores de verdad o modelos, y el cálculo de valores y modelos.

Para interpretar las fórmulas de la LC será necesario contar tanto con una función de asignación de denotación al léxico llamada estructura como con conjunto de reglas de interpretación para cualquier fórmulaen términos de funciones de verdad o modelos. En el caso de la función de asignación, el dominio es el léxico de letras oracionales y, el rango, un universo de dos valores, y . Por su parte, las reglas de interpretación tomarán como dominio a los valores de verdady de una o dos fórmulas y tendrán como rango, otro valor o .

**4.1. Estructuras**

Esta función toma las letras oracionales de una fórmula y les asigna uno de los valores o .Se presenta en un formato tabular que tiene dos filas y tantas columnas como letras oracionales haya en . En las celdas de la primera fila, se ordenan alfabéticamente las letras oracionales y, en las de la segunda, se consigna un valor aleatorio para cada letra.

**Ejemplos**

1.

|  |
| --- |
| *P* |
| V |

2.

|  |
| --- |
| *P* |
| F |

3.

|  |  |
| --- | --- |
| *P* | *Q* |
| F | F |

4.

|  |  |  |
| --- | --- | --- |
| *P* | *Q* | *R* |
| V | F | V |

El formato tabular de estructuras para la LC son las estructuras matemáticas más sencillas que se utilizan en la lógica clásica[[2]](#footnote-2).

**4.2. Reglas de interpretación**

Para cualquier fórmula , las siguientes reglas son funciones que permiten calcular un valor o a partir de cualquiera de las estructuras posibles según el léxico que posea. En términos de modelos, lo que permiten calcular estas reglas son todas las que son o no modelos de . Debe recordarse que(“ es *V* según ”) significaexactamente lo mismo que (“ es modelo de ” o “ satisface a ”); del mismo modo que (“ es *F* según ”) es solo una forma distinta de decir (“ no es modelo de ” o “ no satisface a ”).

**4.2.1 Para fórmulas atómicas**

1. *sii es V según*

**4.2.2 Para fórmulas moleculares**

1. *sii*
2. *sii* y
3. *sii* o
4. *sii* o
5. *sii* y , o y

De acuerdo con estas reglas, el valor de cualquier fórmula compuesta se calculará funcionalmente siempre a partir de una estructura que asigne valores a sus fórmulas atómicas previamente. Por ello, el cálculo semántico de la LC es recursivo*,* al igual que su formación sintáctica, ya que se desarrolla composicionalmente según los símbolos que utiliza cada fórmula. Esto evidencia que los lenguajes formales son absolutamente transparentes en sus funciones lógicas al nivel de su estructura sintáctica. Los lenguajes naturales no poseen esta transparencia[[3]](#footnote-3).

**5. Cálculo de valores**

Este tipo de cálculo se puede desarrollar a través de dos métodos: el arbóreo y el lineal.

**5.1 Método arbóreo**

Como se vio en la sección 2, existe un único árbol sintáctico para toda fórmula bien formada en LC. Estos árboles culminan su desarrollo cuando todas las fórmulas al extremo de sus ramas son atómicas. Este análisis sintáctico ascendente puede complementarse con un cálculo semántico descendente de valores. Para calcular arbóreamente el valor de verdad de cualquier fórmula , se debe seguir tres pasos:

**Paso 1.** Construir el árbol sintáctico de

**Paso 2.** Consignar los valores asignados al léxico de en una estructura *U* al lado derecho de todas las subfórmulas atómicas en los extremos de las ramas del árbol, antecedidas por el símbolo “”.

**Paso 3.** Calcular los valores de todas las subfórmulas compuestas de según *ri2-ri6*.

**Ejemplos**. Calcular el valor de verdad de las siguientes fórmulas según las estructuras *U* indicadas por el método arbóreo

1.

|  |  |
| --- | --- |
|  |  |
| V | F |

**Paso 1**

**Paso 2**

**Paso 3**

Cada paso descendente desde las ramas representará la aplicación de una de las reglas de interpretación *ri2-ri6*. Primero, según *ri3*, es F:

A continuación, según *ri2*:

Por lo tanto, la estructura que asigna los valores y a las letras oracionales y , respectivamente, asigna el valor a la fórmula compuesta .

2.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| V | F | V |

**Paso 1**

**Pasos 2 y 3**

F

*V*

*V*

De arriba hacia abajo, las reglas que han sido aplicadas en cada paso son *ri2-ri4-ri2-ri5*, en ese orden. Así, es en la estructura que asigna los valores *,* y a , y , respectivamente.

**5.2 Método lineal**

Este tipo de cálculo deja de lado la representación de la interpretación semántica superpuesta al análisis sintáctico de una fórmula en favor de un formato mucho más directo y austero. Los pasos para realizar un cálculo lineal de valores para cualquier fórmula son los siguientes:

**Paso 1.** Consignar una estructura *U* en su formato tabular y extender una columna a la derecha de la tabla. En la celda que está a la derecha de las letras oracionales, consignar la fórmula .

**Paso 2.** Copiar los valores asignados por *U*a cada letra oracional en la celda que está debajo de la fórmula , exactamente debajo de cada aparición suya en esta.

**Paso 3.** Calcular los valores de las subfórmulas compuestas de según *ri2-ri6*. El orden que debe seguirse es ascendente en términos de la complejidad sintáctica de las fórmulas y se corresponde con el del cálculo arbóreo.

**Ejemplos**. Calcular el valor de verdad de las siguientes fórmulas según las estructuras *U* indicadas por el método lineal.

1.

|  |  |
| --- | --- |
|  |  |
| V | F |

**Paso 1**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| V | F |  |

**Paso 2**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| V | F | V F |

**Paso 3**

En primer lugar, se debe calcular el valor de según *r3*:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| V | F | V F F |

A continuación, se debe aplicar *ri2* como último paso:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| V | F | V V F F |

No se debe olvidar que el valor de en *U* es siempre el último en consignarse en un cálculo lineal y está siempre debajo del operador principal de .

2.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| V | F | V |

**Pasos 1 y 2**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| V | F | V | V V F |

**Paso 3**

El orden que se debe seguir está dado siempre por los valores que ya están disponibles en ese momento. Así, aún no se puede deducir la valuación de ni la de , pero sí la de :

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| V | F | V | V F V F |

A continuación, se puede calcular el valor de ; luego, el de ; y, finalmente, el de , es decir, el de :

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| V | F | V | F V V F V V F |

**6. Cálculo de modelos**

Debido al paralelismo entre las semánticas de valores de verdad y la de modelos, los métodos arbóreo y lineal de cálculo de valores sirven también para determinar si una estructura es modelo de una fórmula o no; en otras palabras, permiten saber si cualquier satisface o no a . Por ello, pueden plantearse un cálculo inverso de valores o cálculo de modelos para una fórmula que opera según *ri1-ri6*, aunque solo para fórmulas que son satisfacibles, es decir, aquellas para las cuales hay al menos una estructura que les asigna el valor . Estos son los pasos que deben seguirse para realizar un cálculo de modelos para cualquier fórmula en un formato lineal:

**Paso 1.** En una tabla de cálculo lineal de valores con la celda debajo de las letras oracionales de la estructura *U* en blanco, consignar el valor *V* debajo del operador principal de .

**Paso 2.** Siguiendo las reglas de interpretación *ri2-ri6* en orden descendente de complejidad sintáctica, consignar los valores que podría tener cada subfórmula para satisfacer el valor V asignado a . Es posible que haya más de un valor que cumpla con este requisito, ya que algunas fórmulas tienen más de una estructura *U* que los modela.

**Paso 3.** Cuando todas las subfórmulas atómicas tengan un valor asignado, es importante cotejar que no haya una contradicción en los valores asignados a las letras oracionales, es decir, que una misma letra no tenga los valores *V* y *F* en sus apariciones en . De ser el caso, volver sobre los pasos y realizar las correcciones correspondientes. De no serlo, copiar los valores en la celda debajo de las letras oracionales de la estructura *U* y este será el modelo buscado.

**Recomendación.** Apenas se asigne un valor a una subfórmula atómica de , copiar el mismo valor debajo de todas las apariciones de la misma letra oracional en . A continuación, seguir con el paso 2 intentando mantener la consistencia en la asignación de valores. De no ser posible, volver sobre los pasos y realizar las correcciones correspondientes.

**Ejemplo.** Calcular una estructura que sea modelo para cada una de las siguientes fórmulas.

1.

**Paso 1**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  | V |

**Paso 2**

Por ejemplo, una opción sería asignar los valores *V* y *F* a las subfórmulas y , respectivamente:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  | V V F |

En tanto ambas subfórmulas tienen como operador principal la negación, se puede deducir que las subfórmulas a la que operan tendrán los valores *F* y *V*,respectivamente:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  | V F V F V |

A continuación, aplicando la recomendación, se debe copiar el valor *V* asignado a la segunda aparición de *T* en debajo de su primera aparición.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  | V F V V F V |

Finalmente, para que fuese F, *P* podría ser *V* y podría ser F. En este caso, ya que *T* es *V*, *S* tendría que ser *F*:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  | V V F F F V V F V |

**Paso 3**

Ya que no hay inconsistencias en la asignación de valores, se puede copiar los valores de las fórmulas atómicas de en la celda debajo de las letras oracionales de la estructura *U* que es modelo de :

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| V | F | V | V V F F F V V F V |

**Glosario de conceptos**

* Árbol sintáctico
* Operador principal
* Subfórmula
* Grados de complejidad
* Cálculo de valores
* Cálculo de modelos

**Lecturas sugeridas**

* Haack, S. (1982). *Filosofía de las lógicas*. Madrid: Cátedra. Capítulo 3: Conectivas de oraciones, pp. 48-58.
* Smith, P. (2010). *An Introduction to Formal Logic*. Cambridge: Cambridge University Press. Capítulo 8. The syntax of PL y 9. The semantics of PL., pp. 63 -81.

1. Si bien el conjunto de cinco conectores expuesto aquí es suficiente, no es necesario para ser adecuado funcionalmente. En realidad, lenguajes formales con los conectores “” y “”, o “” y “”, que son más económicos sintácticamente, también son completos funcionalmente. Siguiendo este mismo camino, se puede definir un lenguaje que es adecuado en términos suficientes y necesarios que consta de un único conector lógico “……” que puede leerse como “no es cierto que tanto … como … ”. [↑](#footnote-ref-1)
2. El atomismo lógico, una corriente filosófica que se desarrolló desde finales del siglo XIX hasta mediados del siglo XX, desarrolló, en gran medida, la lógica de conectores y la adecuó a una semántica de estados posibles del mundo, análoga a la de modelos, pero con un compromiso ontológico mucho mayor que el de las estructuras matemáticas . Según el atomismo, el universo se define como el conjunto de hechos que son el caso; a continuación, si un hecho expresado por una fórmula es el caso, se dice que esta es verdadera; si no es el caso, se dice que es falsa. Así, un estado posible del universo (*EPM*) está compuesto de hechos atómicos expresados por las letras oracionales de la LC que los atomistas llaman letras proposicionales. Los atomistas añadían que las entidades a las que se le asignan propiamente los valores de verdad no son las letras oracionales, sino las proposiciones, es decir, entidades mentales que se definen como significados de oraciones. [↑](#footnote-ref-2)
3. La lógica tradicional (desde Aristóteles hasta Leibniz) entiende la forma lógica en tanto patrones sintácticos de lenguajes naturales que tan solo se esquematizan en un lenguaje formal. Sin embargo, dicho enfoque incurre en la equivocación de asumir a los lenguajes naturales como transparentes en su funcionalidad lógica. La lógica moderna , en cambio, la interpreta en términos de funciones (a partir de las publicaciones del *Análisis matemático de la lógica* deG. Boole, en 1846, y el *Begriffschrift* de G. Frege, en 1876). El enfoque de la semántica de modelos (a partir de la publicación de la *Introducción a la lógica y a la metodología de las ciencias deductivas* deTarski, en 1941), más precisamente, entiende a la forma lógica en términos de funciones veritativas o modelares. [↑](#footnote-ref-3)