Asignatura: Lógica 3 Curso 2004-2005
Profesor: Juan José Acero 20 – 25 de Octubre del 2004

TEMA I: LOS CONCEPTOS FUNDAMENTALES DE LA TEORÍA DE LA COMPUTABILIDAD

1. El concepto de algoritmo. Los matemáticos y la búsqueda de soluciones a clases de problemas: métodos que permitan hallar soluciones de una forma *sistemática*. *Algoritmos* como *procedimientos generales*: (i) procedimiento aplicable a cada elemento de una *clase* de problemas; (ii) procedimiento cuya ejecución se puede especificar hasta sus últimos detalles; (iii) de forma no *ambigua* (e. d., no queda hueco que la imaginación creativa pueda rellenar); (iv) que consta de un número *finito* de pasos. Algoritmos que finalizan y algoritmos que no finalizan (p. ej., localizar un número de teléfono en una guía telefónica, dividir un número por otro). Algoritmos que finalizan *vs.* algoritmos finitos (p. ej., el algoritmo para calcular n^m).

2. Algoritmos y lenguajes. Disponer de un procedimiento (general) equivale a disponer de medios, de equipos o equipamiento, para llevar a cabo operaciones con cosas concretas (p. ej., una máquina de calcular, una guía telefónica, las fichas de los alumnos matriculados en una asignatura). Nuestra perspectiva es *teórica*: los materiales con que se ha de trabajar son signos. *Algoritmos como procedimientos para modificar filas* (cadenas, ristras) *de signos*. Alfabetos (o vocabularios), filas de signos y palabras. La palabra vacía: □. El alfabeto **U** = {I, □}. El alfabeto **B** = {0, 1, 2…, 9}. Palabras de **B**.

3. Un ejemplo: el algoritmo de las subfórmulas. El alfabeto de la lógica proposicional. El alfabeto, las filas de signos y las fórmulas de la lógica proposicional. El algoritmo que permite obtener las subfórmulas de una fórmula dada.

4. Los conceptos básicos de la Teoría de la Constructibilidad: (1) Los conceptos de computabilidad y enumerabilidad. *Funciones computables*: funciones aritméticas para las cuales existen algoritmos que finalizan (p. ej. $x+y$, $x \cdot y$, x^y, etc.). Desde la perspectiva que nosotros adoptamos. Una función computable *f* tiene como dominio el conjunto de todas las

palabras sobre el alfabeto **B** y como valores palabras (W_1, W_2, …) formadas con los elementos de **B**. La función *f* es computable si (y sólo si) existe un procedimiento general –un *algoritmo*– con ayuda del cual, para toda palabra *W* de **B** puede obtenerse efectivamente el valor de *f*(*W*). Sea *f* una función computable cuyo dominio es el conjunto de los números naturales. Un conjunto *M* de palabras de un alfabeto A es *enumerable* (o bien *f enumera M*) si, y sólo si, *M* es el conjunto de los valores *f*(0), *f*(1), *f*(2), etc. (*M* = {*f*(0), *f*(1), *f*(2),…}.) De otra forma: *M* es *enumerable*, si hay una función computable que asigna a **todo** número natural un elemento de *M* de manera que ningún elemento de M queda sin enumerar.

5. Los conceptos básicos de la Teoría de la Constructibilidad: (1) El concepto de decidibilidad. *Conjuntos decidibles* (p. ej., el conjunto de los granadinos nacidos el 24 de Octubre del 2003, el conjunto de las fórmulas de la lógica proposicional que son tautologías, etc.). La idea general: hay dos conjuntos M_1 y M_2; y hay también un procedimiento general para determinar si un elemento de M_1 es también elemento de M_2. Desde nuestra perspectiva, M_1 y M_2 son conjuntos de palabras: M_2 es *decidible con relación a* M_1 si existe un algoritmo que finaliza por medio del cual puede determinarse efectivamente si una palabra cualquiera de M_1 es o no miembro de M_2. El concepto de *procedimiento decisorio*. Un conjunto *M* de palabras de un alfabeto **U** es decidible si *M* es decidible con respecto al conjunto de todas las palabras de **U**. **TEOREMA I: Todo conjunto finito es decidible. TEOREMA II: Sean *A*, *B* y *C* conjuntos de palabras de un alfabeto U. Entonces si *B* es decidible con relación a *C* y *A* está contenido en *B*, entonces *A* es decidible con relación a B si, y sólo si, A es decidible con relación a C. TEOREMA III: Sean *A* y *B* conjuntos de palabras de un alfabeto U. Si *B* es decidible y *A* está contenido en *B*, entonces *A* es decidible con relación a *B* si, y sólo si, *A* es decidible.**

6. Los conceptos básicos de la Teoría de la Constructibilidad: (1) El concepto de conjunto generable. El concepto de deducibilidad. Conjuntos *generables*. Ejemplos de conjuntos generables: el lenguaje $a^n b^n$; el lenguaje monario de los números naturales({I, II, III,....}); el lenguaje binario de los números naturales ({0, 1, 10, 11, 100,...}). **TEOREMA IV: Un conjunto *M* de palabras de un alfabeto U es generable si, y sólo si, *M* es enumerable**. **TEOREMA V: Sea *M* un conjunto de palabras de un alfabeto U y sea *M** el complemento de *M* con respecto a U. Entonces *M* es decidible si, y sólo si, *M* y *M** son generables.**

7. Numeraciones de Gödel. Palabras *vs.* números. Un solo alfabeto es necesario: el alfabeto **U** = {I, □}. La idea de una numeración de Gödel: trabajar tan sólo con palabras que representen números y así manejar únicamente funciones computables. *G* es una *numeración de Gödel* si, y sólo si: (i) Si $W_1 \neq W_2$, entonces $G(W_1) \neq G(W_2)$; (ii) existe un algoritmo tal que para cualquier palabra *W*, el algoritmo encuentra el valor de *G*(*W*) —el *número de Gödel* de *W*— en un número finito de pasos; (iii) si *G*(*W*) es el número de Gödel de una palabra *W*, hay un algoritmo que termina que construye la palabra W que tiene *G*(*W*) a partir de *G*(*W*).

APÉNDICES

(1) DEMOSTRACIÓN DEL TEOREMA IV

1ª Parte. Supongamos que *M* es enumerable: *M* es el contradomino (o dominio de valores) de una función computable *f* cuyo dominio es un subconjunto de los números naturales. Por lo tanto, hay un algoritmo *R* con ayuda del cual podemos calcular *f(n)* para cualquier número natural *n*. *M* es generable mediante el siguiente sistema de reglas:

(1) El sistema que genera los números naturales.
(2) Para cada número natural *n*, calcúlese el valor de *f(n)* aplicando *R*.

2ª Parte. Supongamos que *M* es generable. Hay un sistema de reglas *G* que genera el conjunto *M*. Consideremos la longitud de las deducciones que puedan hacerse mediante G: es decir, el número de líneas de cada deducción. Ahora consideramos, si es que las hay, las deducciones de longitud *k*. El número de ellas es finito. A continuación las ordenamos todas por medio de algún procedimiento algorítmico (quizás lexicográfico). (P. ej., si hay deducciones de longitud 1, hacemos una lista alfabética
de su última palabra.) Supongamos que hay *m* de ellas y que son las siguientes:

$$W_{0,k}, W_{1,k}, W_{2,k}, ..., W_{m,k}.$$

Entonces, definimos una función *f* así:

$$f(0) = W_{0,k} \qquad f(1) = W_{1,k} \qquad f(2) = W_{2,k} \qquad \ldots\ldots \quad f(m) = W_{m,k}$$

A continuación, consideramos las deducciones de longitud k+1. Supongamos que hay j de ellas y que son las siguientes:

$$W_{0,k+1}, W_{1,k+1}, W_{2,k+1}, ..., W_{j,k+1}.$$

Entonces seguimos definiendo *f* del siguiente modo:

$$f(m+1) = W_{0,k+1} \qquad f(m+2) = W_{1,k+1} \qquad \ldots\ldots \quad f(m+j) = W_{j,k+1}.$$

Si no hay deducciones de longitud k+1, pasamos a considerar las que tienen longitud k+2; y así sucesivamente.

(2) DEMOSTRACIÓN DEL TEOREMA V

1ª Parte. Supongamos que *M* es decidible. En primer lugar, generamos el conjunto *M'* , es decir, el conjunto de todas las palabras del alfabeto *U*. Si $U = \{a_1, ... a_n\}$, la gramática *G* que la genera es la siguiente:

R_0 : Escriba a_1
R_1: Añada a_2
.
.
.
R_{n-1}: Añada a_n

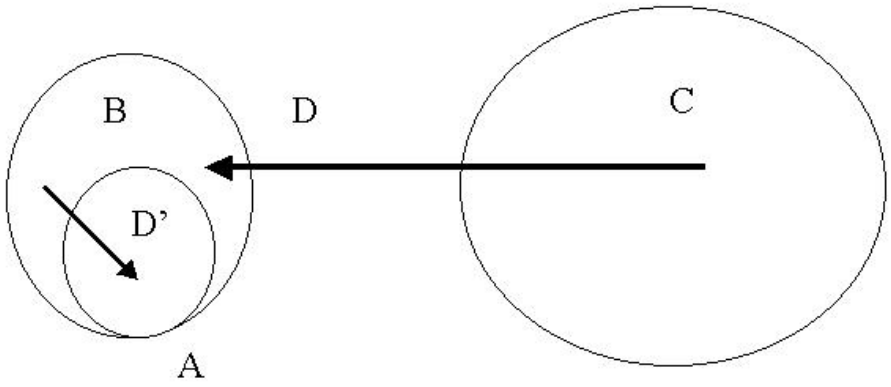
Puesto que *M* es decidible, hay un procedimiento decisorio *D* que determina si un miembro de *M'* es o no miembro de *M*. Pues bien, a R_0, R_1, R_{n-1} añadimos esta otra regla:

R_n : Para cualquier palabra *W* generada con el primer sistema, aplique *D* a *W*, de forma que si *W* es miembro de *M*, escriba *M*; y si *W* no es miembro de *M*, no haga nada.

2ª Parte. Supongamos que *M* y *M** son generables. Por el **TEMA III**, *M* y *M** son enumerables. Por lo tanto, hay funciones computables, *f* y *g*, tales que $M = \{f(0), f(1), ...\}$ y $M^* = \{g(0), g(1), ...\}$. Sea *W* una palabra cualquiera de *M'*. Hay un procedimiento decisorio *D* que en un número finito de pasos determina si *W* es o no miembro de *M*. *D* es el siguiente procedimiento:

> Compútese $f(0)$, $g(0)$, $f(1)$, $g(1)$, $f(2)$, $g(2)$, …., uno por uno y tras cada proceso de computación de un $f(n)$ o de un $g(n)$ compruebe si el resultado es *W*. Si $f(n) = W$, *W* es miembro de *M*; si $g(n) = W$, *W* no es miembro de *M*, sino de *M**. Además, *D* es un algoritmo que termina.

Teorema II: Si A, B y C son conjuntos de palabras de V,
B es decidible con respecto a C y A está contenido en B,
A es decidible con respecto a B si, y sólo si,
A es decidible con respecto a C

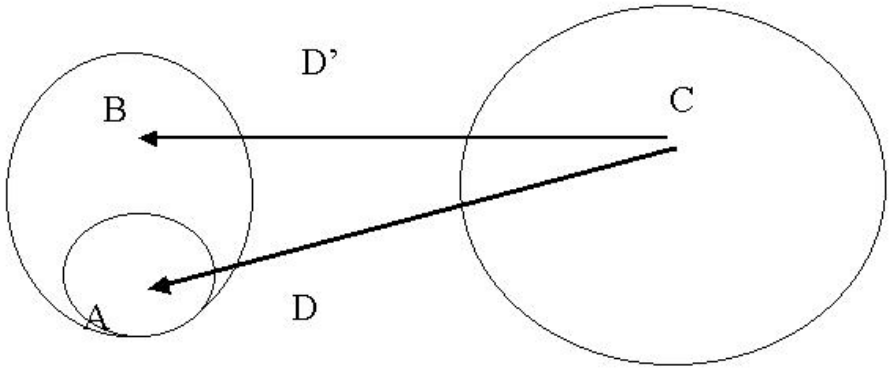


Demostración.

(1) Supongamos primero que A es decidible con respecto a B. Es decir, hay un procedimiento decisorio D tal que D determina en un número finito de pasos si un *x* cualquiera de B es miembro de A. Otro tanto ocurre con B y C: hay un procedimiento análogo D' que determina en un número finito de pasos si un *x* cualquiera de C es o no miembro de B. Y sabemos también que todo miembro de A es miembro de B. Sea *y* un elemento cualquiera de C. D' nos dice si *y* es o no miembro de B; si *y* no lo es, tampoco será *y* miembro de A. Y si D' determina que *y* es miembro de B, entonces D determina

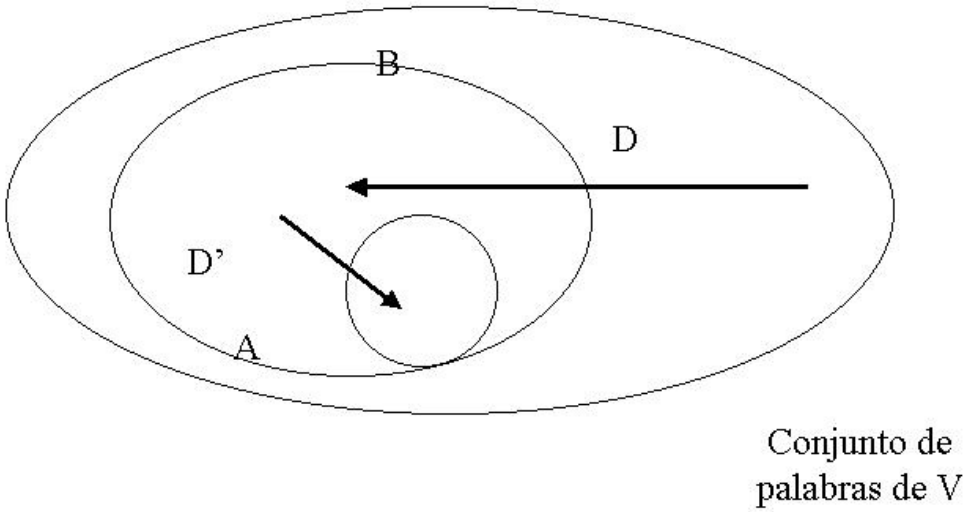
también si *y* es miembro de A. Por lo tanto, de cualquier *y* de C podemos saber en un número finito de pasos si es o no miembro de A.

Teorema II Si A, B y C son conjuntos de palabras de V,
B es decidible con respecto a C y A está contenido en B,
A es decidible con respecto a B si, y sólo si,
A es decidible con respecto a C



(2) Supongamos ahora que A es decidible con respecto a C (el conjunto de todas las palabras de V) : habrá un procedimiento decisorio D tal que D determina en un número finito de pasos si un *x* cualquiera de C es miembro de A. Otro tanto ocurre con B y C: hay un procedimiento análogo D' que determina en un número finito de pasos si un *x* cualquiera de C es o no miembro de B. Y sabemos también que todo miembro de A es miembro de B. Sea *y* una palabra cualquiera de B. ¿Podemos determinar en un número finito de pasos si *y* es miembro de A? Sí. En efecto, si *y* se obtiene por aplicación del procedimiento D, *y* es miembro de A; en caso contrario, *y* no es miembro de A. El procedimiento que buscamos D* es el siguiente. Si D' dice 'Sí' a *y*, entones hay que consultar D y entonces sabremos si *y* está o no en A. Si D' dice 'No' a *y*, entonces *y* no está en A.

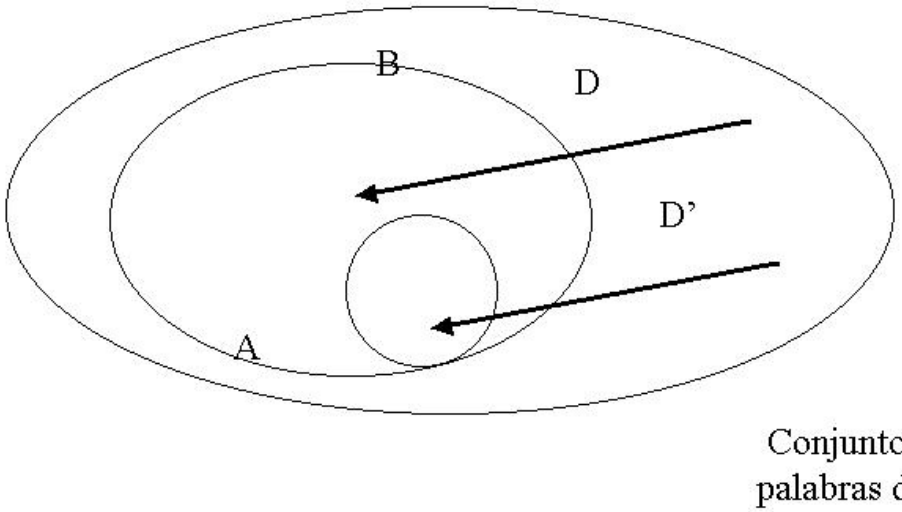
Teorema III: Si A y B son conjuntos de palabras de V, B es un conjunto decidible y A esta contenido en B, entonces A es decidible con respecto a B si, y sólo si, A es decidible



Demostración

(1) Supongamos que A es decidible con respecto a B. Entonces hay un procedimiento decisorio D' que determina si cualquier palabra de *x* de B es o no miembro de A. Por otra parte, sabemos que B es decidible con respecto al conjunto C (formado por todas las palabras de V). Es decir, hay un procedimiento decisorio que determina en un número finito de pasos si un elemento cualquiera de C, *x*, es o no miembro de B. A es decidible (es decir, decidible con respecto a C). Pues sea *y* un miembro de C. Entonces D determina si es o no miembro de B. Y en caso de serlo, D' determina si *y* es miembro de A. El procedimiento buscado utiliza, primero, D y luego, si todavía no se tiene la respuesta completa, emplea D'.

Teorema III: Si A y B son conjuntos de palabras de V, B es un conjunto decidible y A esta contenido en B, entonces A es decidible con respecto a B si, y sólo si, A es decidible



(2) Supongamos que A es decidible (con respecto al conjunto C, de todas las palabras de V). Por lo tanto, hay un procedimiento decisorio D' que determina si un elemento cualquiera *y* de C es o no elemento de A. También B es decidible (con respecto al conjunto C, de todas las palabras de V). Por lo tanto, hay un procedimiento decisorio D que determina si un elemento cualquiera de C *x* es o no elemento de B. ¿Hay un procedimiento decisorio para determinar si una palabra cualquiera de B, *y*, es o no elemento de A? Sí; pues *y* es elemento de C y D' determina si y es o no elemento de A. Si, según D', *y* no es elemento de A, veamos qué dice D acerca de *y*. Si *y* es elemento de B, el procedimiento deseado es D': cuando *y* no está en A, D' nos dice que no está; y si lo está, también D' nos lo dice.

Problemas

1. Haciendo uso del Teorema V, demuestre que el conjunto de las tautologías de la lógica proposicional y el conjunto complemento de éste son generables.

2. Demuestre que el conjunto de los números naturales en notación binaria (N = {0, 1, 10, 11, 101,...} es generable. **Nota 1**: Tenga en cuenta que delante de un '1' no puede ir un '0' a menos que haya un '1' delante de ese '0'. **Nota 2**: A la hora de escribir la gramática G que genere N, puede hacer uso, además de '0' y '1', de cualquiera otros símbolos ('X', 'Y', etc. Auxiliares).

3. Demuestre que el conjunto $L = \{a^n b^n\}$ es enumerable. Podemos suponer que n = 0, 1, 2, ...

4. Demuestre que el conjunto $L = \{a^n b^n\}$ es decidible. El vocabulario de las palabras de L es V = {a,b}.

5. Discuta con detalle la posibilidad de enumerar el conjunto de las tautologías de la lógica proposicional.

6. Suponga que F es un conjunto finito. Por otra parte, sabemos que el conjunto **N** de los números naturales es infinito. Por lo tanto F tiene menos elementos que **N**. ¿Es esto una dificultad para que F sea enumerable? Explique su respuesta.