CONJUNTOS RECURSIVAMENTE ENUMERABLES.-

Def.- Un conjunto $A \subseteq N$ es recursivamente enumerable (r.e.) si

$A = \varnothing$ o

existe una función P-computable, sobreyectiva, total

$ef : N \rightarrow A$

Esto significa que por medio de una máquina podemos encontrar todo elemento del conjunto contando :

ef(0), ef(1), ef(2), ...

Si deseamos demostrar que un conjunto A es r.e. escribimos un programa ***P*** EFA que :

1) Siempre termina
2) El resultado está siempre en A
3) Todo elemento de A es el resultado de <EFA,n> para algún n

Decimos que EFA **genera** el conjunto A

Algunos conjuntos r.e. :

- N
- $\{n \in N$ / n es primo $\}$
- conjuntos definidos inductivamente
- lenguaje generado por una gramática

A partir de la definición se deduce inmediatamente :

Teorema 1.- Un conjunto r.e. es numerable.

Teorema 2.- La unión y la intersección de dos conjuntos r.e. es también r.e..

Dem.- Lo demostraremos para la unión.

A r.e. , B r.e. $\rightarrow$ A U B es r.e.

- Si ambos conjuntos son vacíos, la unión es vacía y se cumple por def.

- Si uno de los dos conjuntos es vacío, la unión coincide con el otro, y es r.e. por hipótesis

- Supongamos que ninguno de los dos conjuntos es vacío :

Como A es r.e. , existe $EFA : N \rightarrow A$, sobreyectiva total

Como B es r.e. , existe EFB : N $\rightarrow$ B , sobreyectiva total

Construiremos un programa para EF(AUB)

```
PROGRAM(X0)
     X1 := X0 MOD 2;
     X2 := XO DIV 2;
     IF X1 /= 0 THEN
          X3 := EFA(X2)
     ELSE X3 := EFB(X2)
     FI
RESULT(X3)
```

Observemos que para el programa anterior se cumple :

1) Siempre para

2) El resultado está en A o en B, o sea, está en A U B

3) Consideremos un elemento cualquiera x de A U B

- $x \in A$, existe n t.q. EFA(n) = x, se cumple :
  EF(AUB)(2n+1) = x

- $x \in B$, existe m t.q. EFB(m) = x, se cumple :
  EF(AUB)(2m) = x

Por lo tanto, A U B es r.e..

Ejercicio.- Demostrar el resultado para la intersección.

- Un conjunto numerable y no r.e.

La mayoría de los conjuntos numerables que hemos visto son también r.e.. Consideremos un conjunto que no lo es :

TOT = $\{ i \in N / \phi_i$ es total $\}$

TOT es numerable, ya que está contenido en el conjunto de programas.

Teorema 3.- TOT no es r.e.

Dem.- Supongamos que TOT es r.e. Claramente TOT no es vacío, entonces debiera existir una función de enumeración :

$$EFTOT : N \rightarrow TOT$$

Consideremos la siguiente función :

$$g(x) = EFTOT(x)(x) + 1$$

g es computable, ya que EFTOT lo es
g es total , por razón similar

pero g difiere de toda función en la enumeración (por construcción, difiere en al menos un valor)

Por lo tanto, no puede existir EFTOT, TOT no es r.e.

Teorema 4.- Un conjunto A es r.e. sii existe una función computable, parcial $f : N \rightarrow N$, tal que $A = dom(f)$

Dem.

$\Rightarrow$) Supongamos que A es r.e., existe una función de enumeración $f : N \rightarrow A$, con macro EFA

Si $A = \Phi$, f es la función vacía

En caso contrario, considerar :

```
PROGRAM(X0)
     X1 := 0;
     WHILE EFA(X1) ≠X0 DO X1 := SUC(X1) FI
RESULT(X1)
```

Claramente este programa entra en loop sii la entrada no pertenece a A. Sea f la función computada por este programa. Se cumple : $dom(f) = A$.

$\Leftarrow$) Sea $f : N \rightarrow N$, computable, $dom(f) = A$

Sea q el índice del programa que computa f

Si $A = \emptyset$, A es r.e. por definición

Supongamos $A \neq \emptyset$, $a \in A$, consideremos el siguiente programa :

```
PROGRAM(<X1,X2>)
     X3 := EVAL-PROG-STEP (q,X1,X2);
     IF FST(X3) ≠ 0
          THEN X5 := X1
```

```
            ELSE X5 := a
        FI
    RESULT(X5)
```

1) Este programa siempre para (recordar que eval-prog-step siempre para)

2) El resultado está en A. En efecto, el resultado es , o bien a que pertenece a A, o bien, un elemento de dom(f). (por definición, Ix(q) computa f).

3) Para todo elemento de A existe alguna entrada que lo genera.
En efecto, si $x \in A$, $x \in$ dom(f), existe un natural m tal que Ix(q) para frente a x en a lo sumo m pasos. El programa anterior, con entrada <x,m>, da como resultado x.

Este resultado nos lleva a otro modo de caracterizar un conjunto r.e.

Teorema 5.- Un conjunto $A \subseteq N$ es r.e.
sii existe una función P-computable
$f_A : N \rightarrow N$ definida por :

$$f_A(x) = \begin{cases} 1 & \text{si } x \in A \\ \text{indefinida en caso contrario} & \end{cases}$$

Dem.-
$\Rightarrow$) De acuerdo al teorema anterior, existe una función ϕ_q tal que
$A = \text{dom}(\phi_q)$. Entonces el programa :

```
PROGRAM(X0)
    X1 := EVAL-PROG (q,X0);
    X2 := 1
RESULT(X2)
```

computa la función f_A

$\Leftarrow$) Inmediato de acuerdo al teorema anterior

CONJUNTOS DECIDIBLES

Def.- Un conjunto A es decidible si su función característica C_A es computable.

$$C_A(x) = \begin{cases} 1 \text{ si } x \in A \\ 0 \text{ en caso contrario} \end{cases}$$

Teorema 6.- Si A y B son decidibles, también lo son :

C(A) (complemento de A)

A U B

$A \cap B$

Dem.- Es inmediato construir los programas que computan las funciones características en cada caso.

Teorema 7.- Todo conjunto decidible es r.e..

Dem.- Sea A decidible, q el índice del programa que computa C_A

Considere el siguiente programa :

```
PROGRAM(X0)
     X1 := EVAL-PROG(q,X0);
     WHILE X1 = 0 DO SKIP END
RESULT(X1)
```

la función que computa vale 1 si $x \in A$ y es indefinida en caso contrario. Por el teorema 5, A es r.e..

Sea $K = \{ i \in N \,/\, \langle Ix(i),i\rangle \downarrow \}$

Teorema 8 - K es r.e. pero no es decidible

Dem.-

1) K no es decidible

$$C_K(i) = \begin{cases} 1 & \text{si } \langle Ix(i),i\rangle \downarrow \\ 0 & \text{en caso contrario} \end{cases}$$

es la función θ , que sabemos que no es computable

2) K es r.e.

Consideremos el siguiente programa :

```
PROGRAM(X0)
     X1 := EVAL-PROG(XO,XO);
     X1 := 1
RESULT(X1)
```

La función que evalúa vale 1 si $x \in K$ y es indefinida en caso contrario. Por teorema 5, K es r.e.

Existe una interesante relación entre conjuntos decidibles y conjuntos r.e., que se establece en el teorema siguiente :

Teorema 9 .- Si A y C(A) (complemento de A) son ambos r.e., entonces ambos son decidibles.

Dem.- Si $A = \Phi$ o $A = N$, inmediato

Si no, sean MA y MCA expresiones que generan A y C(A).

El programa siguiente computa la función característica de A.

```
PROGRAM(X0)
     X1 := 0;
     WHILE MA(X1) /= X0 & MCA(X1) /= X0 DO
          X1 := SUC(X1)
     END;
     IF MA(X1) = X0
          THEN X2 := 1
          ELSE X2 := 0
     FI
RESULT(X2)
```

Un programa análogo (cambiando la sentencia IF) computa la función característica de C(A).

Como consecuencia de este teorema y el teorema 8, tenemos que el conjunto:

$C(K) = \{ i \in N / \langle Ix(i),i \rangle \uparrow \}$ no es r.e.

Def.- Un **conjunto I de índices** (códigos de programas en P) **respeta funciones** si $i \in I$ y $\phi_i = \phi_j$ implica $j \in I$.

Ejemplos :

$I1 = \{ i \in N / \langle Ix(i),n \rangle \downarrow$ en a lo sumo m pasos $\}$ no respeta funciones

$I2 = \{ i \in N / \phi_i (x) = x + 1 \}$ respeta funciones

Teorema 10.- (Teorema de Rice). Sea I un conjunto de índices que respeta funciones y tal que $I \neq \emptyset$, $I \neq N$. Se cumple que I es indecidible.

Dem.- Sea I que respeta funciones, $q \in I$. Supongamos que ningún índice para la función vacía pertenece a I. (Esto no nos hace perder generalidad, si los índices de la función vacía pertenecen a I tomamos C(I)).

Sea MH la macro en P que dado n computa el código h(n) del siguiente programa H(n) :

```
PROGRAM(X0)
     X1 := EVAL-PROG(n,n);
     X1 := EVAL-PROG(q,X0)
RESULT(X1)
```

Notar que $h(n) \in I$ sii $<Ix(n),n> \downarrow$

Supongamos que I es decidible, sea MI una macro que computa C_I.

El siguiente programa :

```
PROGRAM(X0)
     X1 := MH(X0);
     X1 := MI(X1)
RESULT(X1)
```

computa la función θ !!

Práctico

Ejercicio 1.-

Sea $R = \{ x \in N / (\exists y \in N) < Ix(x),y> \Rightarrow K \}$

a) Demostrar que R es recursivamente enumerable.

b) Es R decidible ? Demostrar.

Ejercicio 2.-

¿Cuáles de los siguientes conjuntos son recursivamente enumerables ?

a) $\{ x \in N / \ \Phi_x (x) = 1 \}$

b) $\{ x \in N / \ \Phi_x \neq id_N \}$

c) $\{x \in N / \ \Phi_x = id_N \}$

Ejercicio 3.-

Sea $f(p,q) = 1$ si $\forall x (< Ix(p),x >\downarrow \ \Rightarrow < Ix(q),x >\downarrow)$ $(dom(\Phi_p) \subseteq dom(\Phi_q))$
indefinido en caso contrario

¿ Es f computable ? Demostrar.