Lógica Superior:

Computabilidad e Incompletitud

Profesor: Eduardo Barrio
UBA - Filosofía
2do cuatrimestre de 2015

Recursividad: conjuntos y relaciones

Lo que hemos hecho:

¿Qué funciones son computables?

Funciones recursivas Básicas: z(x), s(x) and id_k

Composición

Recursión primitiva

La clase de las funciones p.r.

Minimización

La clase de las funciones recursivas

Computabilidad: Recursividad

Recursividad: conjuntos y relaciones

Lo que vamos a hacer

Generalizar
- la noción de *función recursiva*
- La noción de *función primitivo recursivo* (o p.r.)

No simplemente a funciones, sino a conjuntos y relaciones.

----- *Los conjuntos pueden ser recursivos.*

Computabilidad: Recursividad

Recursividad: conjuntos y relaciones

Lo que debería comprenderse al finalizar el capítulo 7

a) Lo que significa que una relación (conjunto) sea una relación recursiva

b) Lo que significa que una relación (conjunto) sea una relación primitiva recursiva

c) Aprender algunas técnicas básicas para mostrar que una relación es recursiva (p.r)

Los conjuntos computables son los conjuntos recursivos

Recursividad: conjuntos y relaciones

Lo que debería comprenderse al finalizar el capítulo 7

Las tres técnicas más importantes son:

a) definición por casos

b) bounded minimización y maximización

c) minimización/maximización con p.r. bound

Los conjuntos computables son los conjuntos recursivos

Recursividad: conjuntos y relaciones

¿Para qué se usan estas técnicas?

Las tres técnicas más importantes son:

a) definición por casos

b) bounded minimización y maximización

c) minimización/maximización con p.r. bound

Usamos estas técnicas para probar que:

1) toda función Turing-computable function es recursiva) (cap.8)

2) No hay un método efectivo para decidir la noción de *implicación lógica* (Cap. 11)

3) Los resultados de Gödel (Cap. 15-18)

Recursividad: conjuntos y relaciones

Conjuntos y Relaciones p.r.

La función característica de un conjunto de números P es la función f tal que:

1. - Si m $\in$ P, f(m) = V
2. - Si m n $\notin$ P, f(m) = F

La función característica de una relación numérica R es la función f tal que:

3. - Si m tiene R con n, f(m,n) = V
4. - Si m no tiene R con n, f(m,n) = F

Recursividad: conjuntos y relaciones

Propiedades efectivamente decidibles:

Decibilidad es a los conjuntos (y relaciones) como computabilidad es a las funciones.

- Funciones: efectivamente computables, p.r y recursivas.
- Conjuntos:
  - efectivamente decidibles,
  - primitivos recursivos
  - recusivamente decidibles.

Recursividad: conjuntos y relaciones

Propiedades efectivamente decidibles:

1. Un conjunto es efectivamente decidible ssi su función característica es total y efectivamente computable.

S es decidible sss su función característica

C(m) = Yes, si m $\in$ S.

No, si m $\notin$ S.

- La relación de *divisibilidad* es el conjunto de pares <m, n>, tal que m divide a n.

- La relación de *divisibilidad* es **decidible** porque dado m y n, hay un algoritmo tal que nos permite determinar en un tiempo finito si m divide a n. (sin resto).

Recursividad: conjuntos y relaciones

Resultados

S es efectivamente decidible ssi su función característica es efectivamente computable.

S es recusivamente decidible (recursivo) ssi su función característica es recursiva.

S es primitivo recursivo (p.r) ssi su función caracterítica es p.r.

Recursividad: conjuntos y relaciones

Resultados acerca de los conjuntos

Ya que las funciones recursivas son efectivamente computables, los conjuntos recursivos son efectivamente decidibles.

Tesis de Church: todas las funciones efectivamente computables son recursivas implica que todos los conjuntos efectivamente decidibles son recursivos.

Todas estas nociones pueden ser generalizadas a relaciones n-adicas

Recursividad: conjuntos y relaciones

Técnicas para mostrar que una relación es recursiva p.r

(i) mostrar que su función característica es p.r.

(ii) Substitución

(iii) Aplicar operaciones lógicas

a. Negación
b. Conjunción
c. Disyunción
d. Bounded Quantification
    i. Universal
    ii. Existencial

(iv) definición por casos (sin overlap)

(v) bounded maximización y minimización

Recursividad: conjuntos y relaciones

Definición de ***minimización acotada:***

Bounded recursión:

Es igual que la recursión ordinaria con una cláusula extra que se agrega la cual requiere que la función que está siendo definida sea más pequeña que la que ya se obtuvo.

$\forall k \leq x \; (x + 0 = x)$
$\forall k \leq x \; (x + Sy = S(x+y))$

(Full unbounded) recursión conduce a la función suma.

Recursividad: conjuntos y relaciones

Definición de ***minimización acotada:***

Bounded recursión:

Operador de minimización (acotado): el menor t (de los ns) tal que Px.

($\mu t < n$) (t es P) = f(x,n) = ($\mu t < n$) (t es primo y $t < n$).

$\text{Min}_{t \leq n}\, f(t, x_1, \ldots, x_n)$ = mínimo $t \leq n$ tal que $f(t, x_1, \ldots, x_n)$ es verdadero si existe tal t

0, si no existe,

Recursividad: conjuntos y relaciones

Definición de ***minimización acotada:***

- **Unbounded searches:** permitir que algún proceso sea iterado hasta que una condición dada se satisfaga (donde no hay un límite a priori en el número de computaciones que pueden ser ejecutadas)

- **Paralelismo:**

○ Máquinas de Turing que no se detienen para ciertas entradas

○ El resultado indefinido de una función parcial.

○ Una búsqueda no acotada de un valor de una función no puede ser definido usando operaciones p.r.

(Full unbounded) minimización conduce a funciones recursivas.

Recursividad: conjuntos y relaciones

Definición de ***minimización no acotada:***

Eliminar Bound

f(x) = el menor y tal que y + x = 10

Para cada x > 10, f(x) es indefinido. Sin embargo, f es computable.

Min (f (x, y)): el valor más pequeño a la equación (f (x, y)) = 0, si tal valor existe, si no existe, indefinido.

Min $f(t, x_1, \ldots, x_n)$

= mínimo t tal que $f(t, x_1, \ldots, x_n)$ es verdadero

si existe tal t

Indefinida, si no existe,

Recursividad: conjuntos y relaciones

Definición de ***minimización no acotada:***

Para computar Mn[f](x), se computa f(x, 0), f(x, 1), f(x, 2), ... , hasta

que se encuentra un y tal que f(x, y) = 0.

Si hay tal y, entonces Mn[f](x) = el menor y;

Si no hay tal y, entonces Mn[f](x) es indefinido.

Podríamos no ser capaces de contestar si la equación (f (x, y)) = 0 tiene solución.

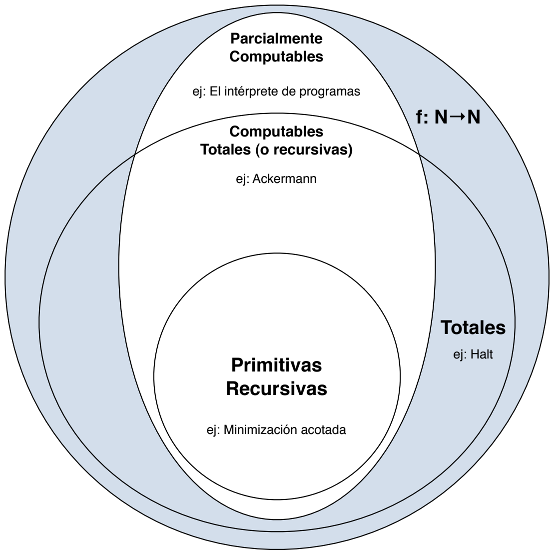
Una búsqueda infinita implica la compleción de una tarea infinita.

Ejemplo: definir la función raíz tal que:

$$\text{raíz}(x) = \sqrt{x}.$$

Para valores cuya raíz cuadrada no es entera, la función no está definida.

$\text{raíz}(x) = \sqrt{x} = \mu t \{t \in N \text{ tal que } t^2 = x\}$



- **Primitivas Recursivas** (programas FOR)
  - Funciones iniciales
  - Composición
  - Recursión primitiva
- **Computables** (programas que terminan)
  - Funciones iniciales
  - Composición
  - Recursión primitiva
  - *Minimización garantizada*
- **Parcialmente Computables** (todos los programas posibles en el lenguaje $\mathcal{S}$)
  - Funciones iniciales
  - Composición
  - Recursión primitiva
  - *Minimización no acotada no garantizada*

Funciones de N → N

En color, las funciones no computables

En blanco, las funciones computables.

Las funciones totales y las computables **no coinciden**

Recursividad: conjuntos y relaciones

Propiedades Semi-decidibles

S es semi-decidible ssi su función característica C(m) =

C(m) es una función parcial.

Yes, si $m \in S$.

Indetermin, si $m \notin S$.

Hay un procedimiento mecánico que reconoce los miembros de S, pero no hay tal cosa para los que no lo son.

Todo conjunto decidible es semidecidible.

Hay conjuntos semidecidible que no son decidibles.

Sus funciones características nunca responden NO.

Recursividad: conjuntos y relaciones

Irresolubilidad del *halting problem*

La relación: *aplicar las instrucciones codificadas por m al input n, STOP* es semidecidible, pero no decidible.

Recursividad: conjuntos y relaciones

TEOREMA DE KLEENE: 7.6

Un conjunto (o relación) es **decidible** ssi ese conjunto y su complemento son semidecidibles.

Proof: Si Rx y $\neg Rx$ son ambas semirecursiva, digamos $Rx \leftrightarrow \exists y\ S{+}xy$ y

$\neg Rx \leftrightarrow \exists y\ S{-}\,xy$, entonces la relación $S{*}$ dada por $S{*}xy \leftrightarrow (S{+}xy \vee S{-}xy)$ es recursiva, y si f es la función definida dejando que $f\,(x)$ sea al menos el y tal que $S{*}xy$, entonces f es una función recursiva total.

Pero, entonces, tenemos $Rx \leftrightarrow S{+}x\,f\,(x)$, mostrando que R se obtiene sustituyendo una función recursiva total en una relación recursiva, y que es por lo tanto recursiva.

Conjuntos recursivamente enumerables (re)

Corolario 8.8: Existen conjuntos recursivamente enumerables que no son recursivos

Un conjunto (o relación) es **decidible** ssi ese conjunto y su complemento son semidecidibles

Un conjunto es r.e. ssi la función f :

f(x) = 1 si x pertenece a A
f(x) = indefinido si x no pertenece a A

es computable (la función f es recursiva.

Que un conjunto sea r e es equivalente a que el conjunto sea generable por una MT.

Un conjunto es recursivo (es decidible), si tanto él como su complemento son recursivamente enumerables.

Indecidible ssi no recursivo

Equivalencia entre MTs y funciones recursivas

Si una función parcial puede calcularse con una máquina de Turing, decimos que es computable por una MT.

Para mostrar esta equivalencia es necesario demostrar que

1) Teorema: Toda función recursiva parcial es computable por una máquina de Turing

2) Teorema: Todo proceso computacional realizado por una máquina de Turing es en realidad el cálculo de una función recursiva parcial