Repaso

martes, 10 de junio de 2025 19:01

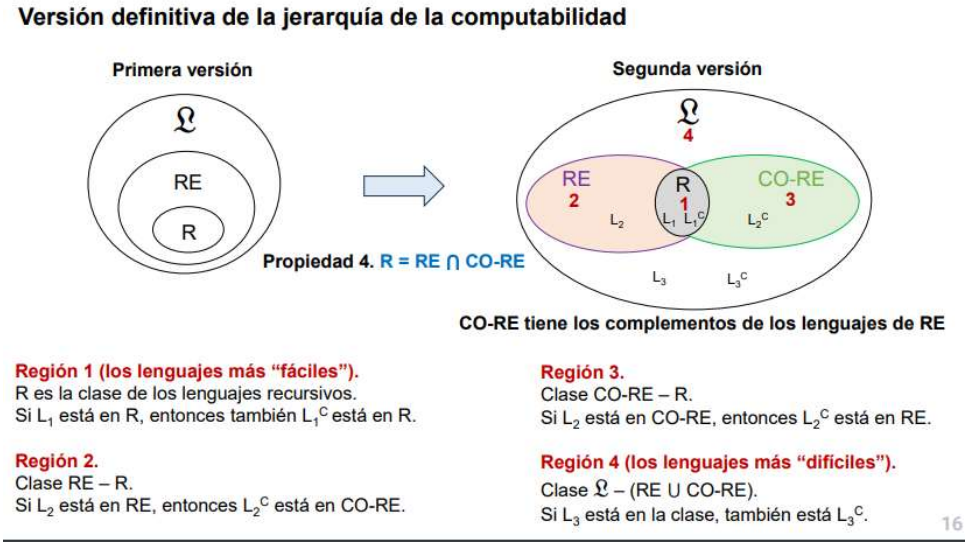
Computabilidad:

Tesis de church-turing: todo lo compuatble se puede hacer con una mt
En un paso, mt cambia el sìmbolo, cambia el estado y se mueve a izq o der

Clase 2:

R siempre paran(decidibles)
Re en los positivos para siempre, en los negativos puede loopear(computables)
L no tienen mt (prblemas)

Propiedades:
Si un lengaje es r o re, tambien lo serà su complemento
Lo mismo con uniòn e interseccìon en r, es poner primero una y despues la otra
En re es màs dificil, tengo que paralelizar.



Clase 3: diagonalizaciòn

Antes probamaos pertenencia a un conjunto, construyendo màquinas.
No nos sirve para probar que algo no està
Entra conceptualmente mètodo constructivo y no constructivo, no diagonalizar en si

Codificar cualquier sìmbolo con enteros. Orden canònico. Truco para no tener que decir hasta màquinas de x cantidad de icntas acepto . No entendi, min 15
Puedo ordenar las mt. Por logitudes y si hay empate por diccionario
No toda cadena es una mt.

HALTING PROBLEM:
M para a partir de w?
De los màs dificiles de RE, RE-R.
Puedo decidir si una mt para si me aseguran que se mueve en un espacio acotado.
Del hp habò bastante.

Clase 4: reducciones

Funciòn computable en forma total:

Reduce u lenguaje si ed adentro va a adetro y afuera va a afuera.ùtil para probar no pertenencia.
Para probar pertenencia lo màs natural es encontrar una mt.

TEOREMA

Caso 1

Si $L_1 \leq L_2$ entonces ($L_2 \in R \rightarrow L_1 \in R$)
O bien:
Si $L_1 \leq L_2$ entonces ($L_1 \notin R \rightarrow L_2 \notin R$)

Caso 2

Si $L_1 \leq L_2$ entonces ($L_2 \in RE \rightarrow L_1 \in RE$)
O bien:
Si $L_1 \leq L_2$ entonces ($L_1 \notin RE \rightarrow L_2 \notin RE$)

Reeducciones en el parcial van a ser elementales, naa muy complicado.
La propiedad màs importante es componer, transitvidad.
La de los complementos tb importa
NO es simèrica la reducciòn.

Mt limitadas no creo que entre.

COMPLEJIDAD
CLASE 5:
Complejdad siempre en R:decidibes.
Funciones que varìan segùn el tamaño de la entrada.
Poli, exp lo importante. Poli eficiente, exponencial no eficiente. N
Una màquina tarda t(n) si hace a lo sumo t(n) pasos.
Hablamos de O(n)
A clase son los lenguajes cuya unica forma de resoluion es por fuerza bruta.

Tesis fuerte de Church-Turing (robustez de la clase P)
Si un lenguaje es decidible en tiempo poly(n) por un modelo computacional **razonable (realizable),** también es decidible en tiempo poly(n) por una MT **(¿hasta que las máquinas cuánticas sean una realidad?).**

Clase np: con un certificado puede decidir si una cadena està en el lenguaje o no.
Todo conceptual en los parciales: puede ser explicar por que no estarìa en p/np algo. Por que sat complemento no estarìa en np? Por ejemplo
Sat, hamilton hay que saberlos ch tb

CLASE 6
Min 49, no entendì pero es importante sobre np complet.
Propiedades: està en np y que todos los lengujes de np se reducen a np.
Iso no tiene mt que se resuelve en tiempo eficiente y algo màs, dij que era importante min51 aprox.
Reducciones polinomiales: tarda tiempo poli.
Ahora a las reducciones :
Funciòn total computable
Adentro adentro afuera afuera
Ahora se suma computable EFICIENTEMENTE
Lo de la derecha es tan o màs difìcil. A la izquierda lo màs fàcil en la reducciones.

Como probar NPC min 56 aprox me medio dormì
Todo par de lenguajes de npc conocidos los puedo reducir de uno al otro,

CLase 7 temporal:

Estaba en otra

CLase 7b espacial:

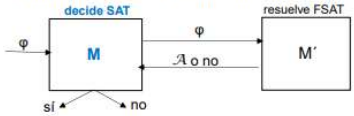
Todo lo de la clase rara no entra.
Complejidad espacial:
Saco del calculo la cinta de entrada.
Tiempo t(n) implica espacio s(n). Espacio s(n) implica tiempo màs grande.
Porque con timpo t(n) sabes que no te vas a mover màs de n pasos.
Se usan punteros (variables que referencian a una posiciòn. Entonces ocupa log)
Todo lo que estè e logspace va a ser tratable.

Clases 7 (parte 3), 8 y 9. Misceláneas de complejidad computacional

Problemas de búsqueda

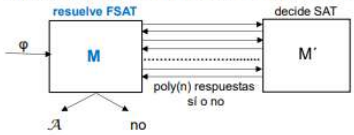
- Las clases P y NP correspondientes a los **problemas de búsqueda** se denominan **FP** y **FNP**.
- **Ejemplo. Problemas FSAT (búsqueda) y SAT (decisión).**

Claramente, FSAT es tan o más difícil que SAT:



A partir de M´ se puede construir M, tal que: si M´ resuelve FSAT en tiempo poly(n), entonces M decide SAT en tiempo poly(n).

Menos intuitivo, también se cumple que SAT es tan o más difícil que FSAT:



A partir de M' se puede construir M, tal que: si M´ decide SAT en tiempo poly(n), entonces M resuelve FSAT en tiempo poly(n).

- Esto se cumple **para todos los lenguajes NP-completos**.

64

Esto de la imagen es importante, 1:14

Parte 3: verificaciòn

CLases 10 y 11:
Toda la parte semàntica.
Un programa s no es correcto solo, es correcto respecto de precondicion, postcondiciòn.
Correctitud parcial y total. Parcial: SI TERMINA el programa, termina como quiero
Se prueba por separado verificacion parcial y total.
Se prueban axiomaticamente con mètodos distintos

3.1. Axiomática para la correctitud parcial (Método H)

1. **Axioma del skip (SKIP)** $\{p\}\ \text{skip}\ \{p\}$

2. **Axioma de la asignación (ASI)** $\{p[x|e]\}\ x := e\ \{p\}$

   p[x|e] denota la sustitución en el predicado p de toda ocurrencia libre de la variable entera x libre por la expresión entera e

3. **Regla de la secuencia (SEC)**
$$\frac{\{p\}\ S_1\ \{r\},\ \{r\}\ S_2\ \{q\}}{\{p\}\ S_1\ ;\ S_2\ \{q\}}$$

4. **Regla del condicional (COND)**
$$\frac{\{p \wedge B\}\ S_1\ \{q\},\ \{p \wedge \neg B\}\ S_2\ \{q\}}{\{p\}\ \text{if B then } S_1 \text{ else } S_2 \text{ fi}\ \{q\}}$$

5. **Regla de la repetición (REP)**
$$\frac{\{p \wedge B\}\ S\ \{p\}}{\{p\}\ \text{while B do S od}\ \{p \wedge \neg B\}}$$

   El predicado p es un predicado invariante del while

6. **Regla de consecuencia (CONS)**
$$\frac{p \rightarrow p_1,\ \{p_1\}\ S\ \{q_1\},\ q_1 \rightarrow q}{\{p\}\ S\ \{q\}}$$

La regla formaliza la posibilidad de reemplazar un predicado p_1 por otro que lo implique, y/o un predicado q_1 por otro al cual implique

76

Regla REP*:
$$\frac{\langle p \wedge B\rangle\ S\ \langle p\rangle,\quad \langle p \wedge B \wedge (t = Z)\rangle\ S\ \langle t < Z\rangle,\quad p \rightarrow t \geq 0}{\langle p\rangle\ \text{while B do S od}\ \langle p \wedge \neg B\rangle}$$

Regla REP*:
$$\frac{\langle p \wedge B\rangle\ S\ \langle p\rangle,\quad \langle p \wedge B \wedge (t = Z)\rangle\ S}{\langle p\rangle\ \text{while B do S od}\ \langle p}$$

No entra lo de concurrencia

CLASE 12:SENSATEZ