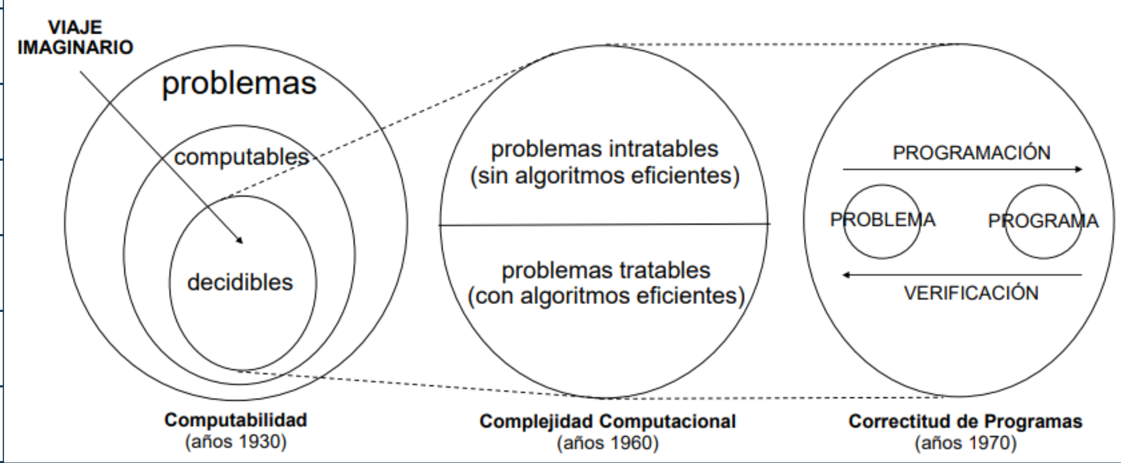
Clase 1-intro

martes, 11 de marzo de 2025 18:58

Programas computables no son todos, y a su vez no deciibles (no todos pueden resolverse siempre, que nunca quedan loopeando)



Los intratables son decidibles pero o por tiempo o por espacio no se pueden resolver. Es más fácil verificar una solución que verificarla

La ia es difícil de verificar

Máquina de turing

Modelo muy simple de computadora.

Frente a un estímulo, da una respuesta.

Una maquina de turing es una maquina que acpeta o reconoce un lenguaje formado por las cadenas que acepta.

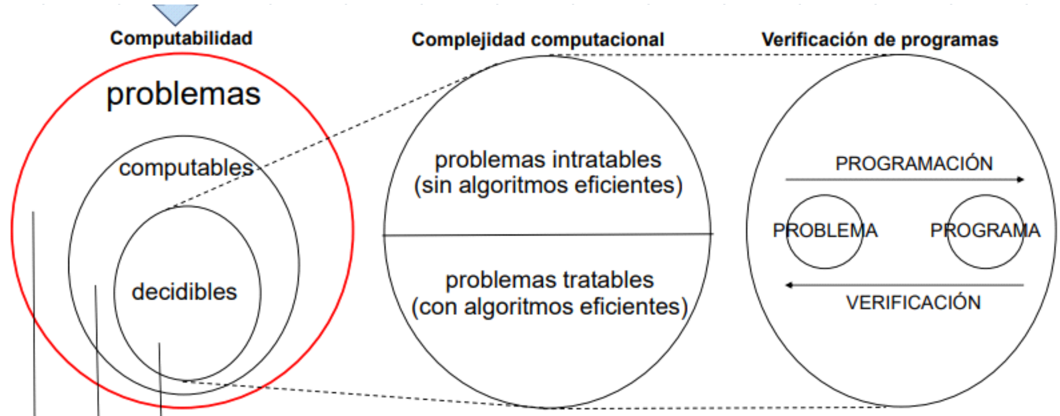
Problema de decisión=lenguaje (para la materia)

Muchos problemas no solo dan sí o no sino que quedan loopeando y no dan respuestas. Si es si dice que si, el tema es si es no que a veces queda en loop

Son computables pero NO decidibles

Ej: halting problem: no hay una maquina que para todo programa de entrada me pueda decir si es decidible o no (si para o no)

Hay problemas a los que no se les puede hace una MT. NO pueden ni detectarme casos positivos. Son NO computables.



Cuentan con MT que los resuelven **totalmente** (casos positivos y negativos)

Cuentan con MT que los resuelven **parcialmente** (con algunos casos negativos pueden loopear)

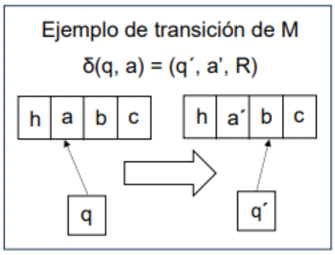
No cuentan con MT que los resuelvan (no resuelven ni siquiera todos los casos positivos)

22

Todo dispositivo computacional físicamente realizable puede ser simulado por una MT.

Formalmente, una MT M es una tupla (Q, Γ, δ, q_0, q_A, q_R):

- Q es el conjunto de **estados** de M.
- Γ es el **alfabeto** (conjunto de símbolos) de M.
  Γ incluye al símbolo blanco B.
  Las cadenas de entrada no admiten blancos.
- q_0 es el **estado inicial** de M.
- q_A y q_R son los **estados finales** de M (de aceptación y de rechazo).
- δ es la **función de transición** de M (especifica su comportamiento):



Dado un estado corriente de Q, y un símbolo corriente de Γ, la MT M:

- pasa eventualmente a un nuevo estado de Q,
- modifica eventualmente el símbolo corriente de Γ,
- y se mueve un lugar a la derecha (R), a la izquierda (L), o no se mueve (S).

La máquina para si en algún momento alcanza el estado q_A o q_R.

Máquinas de t de varias cintas pueden
La MT en un solo paso puede modificar el estado corriente,
los símbolos corrientes de todas las cintas,
y moverse distinto en cada cinta. Más en detalle, en un paso:
1. Lee un estado y una tupla de símbolos (los apuntados por los cabezales).
2. Modifica eventualmente el estado corriente.
3. Modifica eventualmente los símbolos corrientes de las cintas.
4. Se mueve independientemente en cada cinta (a la derecha, a la izquierda, o no se mueve).

Teorema:
Para toda MT M1 con varias cintas, existe una MT M2 equivalente (acepta el mismo lenguaje) con una cinta (ver idea de prueba en la hoja siguiente).

Pierdo tiempo cuadrático. Tengo que ir y volver

- Equivalencias de MT: cuando aceptan = lenguaje (resuelven= problema).
- Dos modelos de MT son equivalentes si dada una MT de un modelo existe una MT equivalente del otro.

Σ es un alfabeto o conjunto de símbolos. Σ = {w1 , w2 , w3 , w4 , …}

Es alfabeto universal

Σ* es el lenguaje o conjunto de cadenas de símbolos generado a partir de Σ. Σ* = {λ, w1 , w2 , w3 , …, w1w1 , w1w2 , w1w3 , …, w1w1w1 , w1w1w2 , …}

Σ* es infinito. Sus cadenas son finitas. λ es la cadena vacía.

Todas las cadenas que puedo formar con sigma

CADA CADENA ES FINITA, NO INFINITA

TODO LO QUE UNA MT LEE ES FINITO SIEMPREEEE

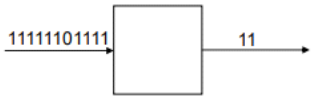
Todos los lenguajes L que consideramos son subconjuntos de sigma*

Otras visiones de MT

Visión de MT calculadora (el caso que vimos al comienzo, el más general, para **problemas de búsqueda**)

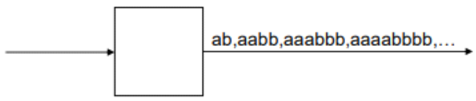
Problema: construir una MT que reste 2 números codificados en unario (todos 1) y separados por un cero.
Por ejemplo, dada la entrada 1111101111, hay que obtener la salida 11 (6 – 4 = 2).
Idea general: tachar el primer 1 antes del 0, luego el primer 1 después del 0, luego el segundo 1 antes del 0, luego el segundo 1 después del 0, y así hasta tachar al final el 0.

Comentario: en este caso, además del estado final, interesa el contenido final de la cinta.



Visión de MT generadora (genera en una cinta de salida todas las cadenas del lenguaje que acepta)

Problema: construir una MT que genere todas las cadenas de la forma $a^n b^n$, con n ≥ 1. Es decir, en una cinta especial de salida debe **generar las cadenas ab, aabb, aaabbb, aaaabbbb, etc.**
Idea general:
(1) i := 1
(2) imprimir i veces *a*, imprimir i veces b, e imprimir una coma
(3) i := i + 1 y volver a (2)



Teorema: Existe una MT M que acepta un lenguaje L sii existe una MT M´ que genera el lenguaje L.

Teorema: Existe una MT M que acepta un lenguaje L sii existe una MT M´ que genera el lenguaje L

Cómo se mantiene el contador con una mt??????