CC

Funcția lui Ackermann (este recursivă, dar nu este primitiv recursivă)

$F : N \times N \to N$

$A(0,n) = n+1$

$A(m+1,0) = A(m,1)$

$A(m+1,n+1) = A(m, A(m+1,n))$

O functie primitiv recursiva este o functie definita din functile de baza:

- Succesor
- Constanta 0
- Proiectie

si care se folosteste de operatii:

- Compunere
- Recursie primitiva

f (x_1,..xn), h (x1,...xn+2), g(x1,... xn+1)

G e recursie primita daca :

- f(x1,...xn) = g(x1,...,xn,0)
- g(x1,...,xn,y + 1) = h(x1,...,xn,y,g(x1,...,xn,y)
-

• $f(x,y) = x + y$ este primitivă recursivă

$f(x,0) = S(x)$

$f(x,y+1) = S(f(x,n))$

Introducem si notiunea de partial recursiv cu operatia de minimizare.
Daca avem f cu n+1 parametrii, definim g cu n parametrii care este cel mai mic y pt care f(x1,..,xn,y) = 0 sau nedefinita daca y nu exista

Masina turing

Avem S multimea de starti, sigma alfabetul si ro: Sxsigma -> S functia de tranzitie

Masina poate scrie peste simbolul pe care l-a citit si asa avem
ro: S X Sigma -> S X Sigma X {left, right, pe_loc}

Cu o masina touring cu o singuraa banda infinita la dreapta putem simula o masina touring infinita in ambele directi cu n benzi. Pentru asta facem alfabetul ca fiecare litera sa fie un tuplu de litere pt fiecare din cele n benzi + literele de la stanga si dreapta pozitie de inceput + offset

Exista o masina touring universala care poate simula orice masina touring given the code and input

Functii necalculabile

Functia universala :

- $U(x) = \begin{cases} 0 \text{ dacă } M_x(x) = 1 & \text{(mașina x pe intrarea x dă 1)} \\ 1 \text{ altfel} \end{cases}$

$U : N \to N$

Calculez $U(n) = M_n(n)$

$\Rightarrow M_n$ calculează U

$U(n) = 0$ dacă $M_n(n) = 1$

1 altfel

Orice functie partial recursiva poate fi calculata de o MT (echivalenta)

Problema opririi este necalculabila

Nu avem algo pt Wang tiles (ala cu dominourile de acopera un plan) ca daca am avea am avea algo pt HALT

S(n) -> mr de pasi pe care-i poate face o MT pana se opreste. nu putem calcula asta ca daca am putea am avea algo pt HALT (am stii in maxim cati pasi s-ar opri orice MT)

O multime este recursiva daca avem o fct f care ne zice 1/0 daca un element apartine sau nu multimii
E recursiv enumerabila daca ne zice 1 daca aprtine si nu returneaza altfel
Daca o multime este recursiv enumerabila si complementara miultimii e recursiv enumerabila, atunci multimea e recursiva

Ne referim la multimi recursiv enumerabile cand ne intereseaza doar elementele care apartin multimii nu si cele care nu

DTIME (f) = $\{A \mid A$ poate fi rezolvată de o MT a.î. $M(x)$ face $O(f(|x|))$ pași

O porblema de decizie este in clasa NP daca avem un algoritm care ne poate verifica in timp polinomial daca problema noastra accepta sau nu o solutie. Dar nu avem algoritm polinomial care sa ne dea solutiile efectiv.

O problema A se reduce la B daca avem o fct f care ne duce un x din A in B si B accepta f(x)

O problema A e NP completa daca A e NP si orice problema din Np se reduce la A
Daca A nu e din NP este NP hard

Probleme NP
- TSP
- SAT
- Colorare graf
- Sudoku

Primele 3 sunt chiar NP complete

Un automat nedeterministic poate fi simulat de unul deterministic dar cu mai multe stari

TSP si SAT sunt np complete

Se poate demonstra ca 3-SAT e np completa
Si putem reduce SAT la 3-SAT

Φ_0

$x \vee y \vee \bar{z} \vee t$

$x \vee \bar{y} \vee \bar{z} \vee \bar{t} \vee m$

$g(\Phi_0)$

$x \vee y \vee \alpha$ → variabilă nouă

$\bar{\alpha} \vee \bar{z} \vee t$

$x \vee \bar{y} \vee \bar{z} \vee \alpha$

$\bar{\alpha} \vee \bar{t} \vee m$

$x \vee \bar{y} \vee \beta$

$\bar{\beta} \vee \bar{z} \vee \alpha$

$\bar{\alpha} \vee \bar{t} \vee m$

Pt o clauze cu 4 variabile bagam 2 cu cate 3 variabile. Introducem o variabila intermediara care in prima clauza e pozitiva si in a doua e negata

2-SAT e P

Integer linear programing, clique (se da un graf, daca avem sau nu miunim K noduri conectate toate intre ele), vertex cover sunt NPC

Orice problema din P o putem reduce la NP

Dem: Fie M o MT polinomială care decide pb A

$f: x \to f(x)$ aî $x \in A \iff f(x) \in B$

Fie $x_0 \in B$

$x_1 \notin B$

Alg pt funcția f:

INPUT x

rulez $M(x)$ adică $x \in A$

Dacă $M(x) = 1 \Rightarrow$ return x_0

Altfel return x_1 $x \notin A$

Dacă $x \in A$ atunci $f(x) = x_0 \in B$

$x \notin A$ atunci $f(x) = x_1 \notin B$

NPC-urile se reduc unele la altele

O formula cu cat are mai multe clauze cu atat e probabilistic mai putin probabil sa fie satisfiabila

Pentru SAT avem algoritmul Davis Putnam care ne rezolva unele probleme. E un algoritm de backtracking:
- Literar pur (x apare doar pozitiv sau doar negat0v -> alegem direct valoarea pt el)
- Literar unitar (x apare doar intr-i clauza -> alegem direct valoarea pt el)

Avem de asemenea si algormitlul de rezolutie. Daca stim C sau C1 si !C sau C2 atunci stim C1 sau C2

O multime se numeste p selectiva daca avem o functie calculabila in timp polinomila a.i

- f (x,y) = fie x fie y
- Daca x sau y aparitn A atunci f(x,y) apartine a

Problema lui matiavelichi e recursiv enumerabila