Formalne metode

u softverskom inženjerstvu

15 Klase problema

ETFBL 24-25

Dunja Vrbaški

- Laki i teški problemi
- Jako laki, laki, teški i jako teški problemi

- Klase složenosti
- Da li ima poklapanja
- Da li ima svođenja

- Heuristike

Vreme izvršavanja izračunavanja Tjuringove mašine M koja kao ulaz dobija podatak x jednako je dužini niza konfiguracija koje predstavljaju to izračunavanje.

Deterministička Tjuringova mašina M radi u vremenu f(n), ako je za bilo koji ulazni podatak x vreme izvršavanja izračunavanja mašine najviše f(|x|).

Nedeterministička Tjuringovu mašina M radi u vremenu f(n), ako je za bilo koji ulazni podatak x vreme izvršavanja bilo kog izračunavanja mašine najviše f(|x|).

Funkcija f je vremenska granica složenosti za M.

Slično smo definisali i za prostornu složenost.

T:

Za datu <u>determinističku</u> Tjuringovu mašnu M sa <u>k-traka</u> i vremenskom granicom složenosti f(n) može se konstruisati <u>deterministička</u> Tjuringova mašina M_0 sa <u>jednom</u> trakom koja simulira rad mašine M i ima vremensku granicu složenosti $\boldsymbol{O(f(n)^2)}$.

Za datu <u>nedeterminističku</u> Tjuringovu mašinu M sa <u>k-traka</u> i vremenskom granicom složenosti f(n) može se konstruisati <u>deterministička</u> Tjuringova mašina M_0 sa <u>jednom</u> trakom koja simulira rad mašine M i ima vremensku granicu složenosti $\boldsymbol{O(c^{f(n)})}$, za c > 1 koje zavisi od mašine M.

Klasa složenosti je skup problema sa zajedničkom vremenskom ili prostornom granicom.

TIME(f(n)) je skup problema za koje postoje <u>determinističke</u> Tjuringove mašine koje ih odlučuju, a za koje je vremenska granica složenosti f(n).

NTIME(f(n)) je skup problema za koje postoje <u>nedeterminističke</u> Tjuringove mašine koje ih odlučuju, a za koje je vremenska granica složenosti f(n).

SPACE(f(n)) je skup problema za koje postoje <u>determinističke</u> Tjuringove mašine koje ih odlučuju, a za koje je prostorna granica složenosti f(n).

NSPACE(f(n)) je skup problema za koje postoje <u>nedeterminističke</u> Tjuringove mašine koje ih odlučuju, a za koje je prostorna granica složenosti f(n).

Nedeterminističke klase složenosti

- broj kandidata za rešenje veliki
- kada se kandidat izabere, onda je problem njegovog testiranja, (verifikacije, provere) u okviru determinističke klase problema.

Sudoku?

Iskazne formule?

Putujući trgovac?

Množenje brojeva, sortiranje?

$L = \mathrm{SPACE}(O(\log n))$

$NL = \mathrm{NSPACE}(O(\log n))$

$P = \cup_i \mathrm{TIME}(n^i)$

$NP = \cup_i \mathrm{NTIME}(n^i)$

$PSPACE = \cup_i \mathrm{SPACE}(n^i)$

$NPSPACE = \cup_i \mathrm{NSPACE}(n^i)$

$EXP = \cup_i \mathrm{TIME}(2^{n^i})$

$NEXP = \cup_i \mathrm{NTIME}(2^{n^i})$

$EXPSPACE = \cup_i \mathrm{SPACE}(2^{n^i})$

$2-EXP = \cup_i \mathrm{TIME}(2^{2^{n^i}})$

$2-NEXP = \cup_i \mathrm{NTIME}(2^{2^{n^i}})$

TIME = DTIME

Da li se klase poklapaju?

Ko je pravi podskup, a ko ne?

Zašto je to važno?

Nešto se zna, a nešto ne.

Da li je

- P = NP
- P = PSPACE
- L = NL
- EXP = NEXP

$P = \cup_i TIME(n^i)$

$NP = \cup_i NTIME(n^i)$

Problemi za koje je vremenska granica složenosti programa koji ih rešavaju neka polinomijalna funkcija.

- granica između praktično izračunljivih problema i onih koji su to samo u principu.
- Dokaz da su ove klase jednake - ogromne razmere
- dokazano: TIME(O(n)) je pravi podskup od NTIME(O(n)) - sugeriše da važi ono što intuitivno mislimo, u šta verujemo i ponašamo se u skladu sa tim
- Ako bi važilo da su klase jednake važilo bi i da je EXP = NEXP.

Problem **A se redukuje na problem B**, u oznaci **A ≤ B**, ako postoji izračunljiva funkcija f takva da je A(x) tačno ako i samo ako je tačno i B(f(x)).

Funkcija f se tada naziva funkcija redukcije.

Ima smisla samo ako je složenost izračunavanja funkcije redukcije zanemarljiva
u odnosu na složenost problema B.

Funkcija redukcije f problema A na problem B je efikasna, a problem A je **efikasno reducibilan** na problem B, u oznaci $A \leq_{ef} B$, ako je složenost funkcije f u klasi L.

Da li postoji Hamiltonov put u grafu, tj. put koji kroz svaki čvor grafa prolazi tačno jednom se efikasno redukuje na problem SAT koji se odnosi na ispitivanje da li je proizvoljna klasična iskazna formula zadovoljiva.

$(x_1 \lor \neg x_2) \land (x_2 \lor x_3) \land (\neg x_1 \lor \neg x_3)$

Pitanje: Postoji li takva dodela vrednosti promenljivama x_1,x_2,x_3 da cela formula bude istinita?

$A \leq_{ef} B$

- složenost problema A je odozgo ograničena zbirom složenosti problema B i funkcije redukcije f.
- najpre se x preslika u f(x), a zatim se primeni program za utvrđivanje da li je B(f(x))
- => Znamo f i B, imamo gornju za A

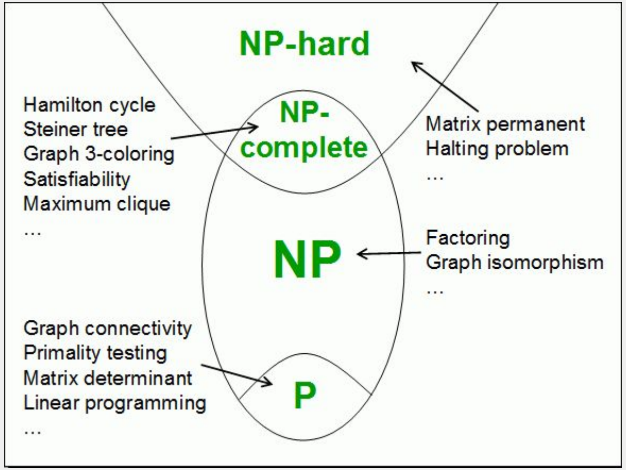
- Sa druge strane, ako je poznato da je složenost problema A veća od nekog zadatog nivoa, onda se kontrapozicijom može odrediti i jedna donja granica složenosti problema B

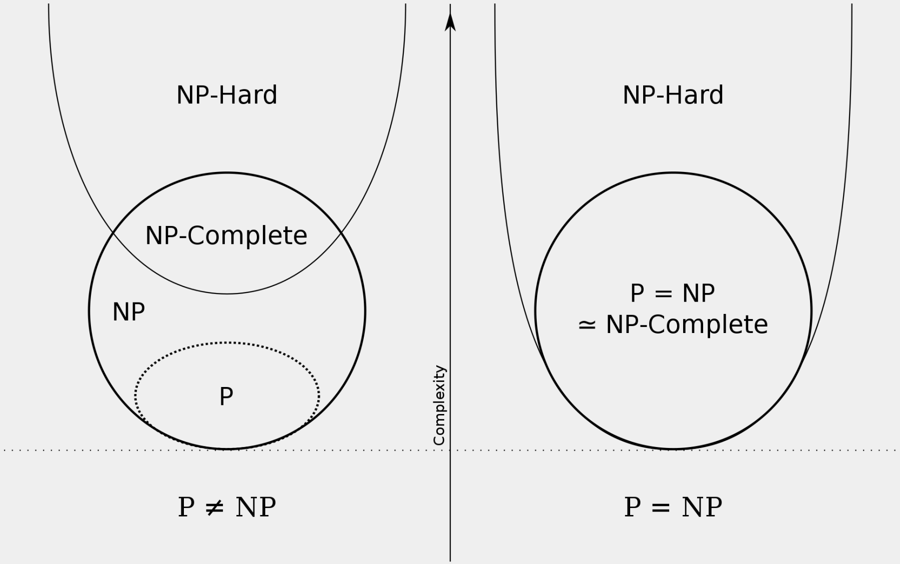
Neka je B problem i C klasa složenosti.

Problem B je **C-težak,** u oznaci $C \leq_{ef} B$**,** ako je za svaki problem $A \in C$ ispunjeno $A \leq_{ef} B$.

Problem B je **C-kompletan** ako je $C \leq_{ef} B$ i $B \in C$.

Šta je onda NP-Hard i NP-Complete?


NP-hard
Hamilton cycle
Steiner tree
Graph 3-coloring
Satisfiability
Maximum clique
...
NP-
complete
Matrix permanent
Halting problem
...
NP
Factoring
Graph isomorphism
...
Graph connectivity
Primality testing
Matrix determinant
Linear programming
...
P



Pojam kompletnog problema je značajan pošto svaki takav problem predstavlja klasu u odnosu na koju je kompletan.

Problem SAT - skup svih zadovoljivih klasičnih iskaznih formula.

T: SAT je NP.

T: SAT je NP-complete.

Dokaz: Potrebno je pokazati da se svaki problem $B \in NP$ može u polinomijalnom vremenu redukovati na SAT.

Ako bi SAT bio i P onda bi sledilo da je i P=NP