TÖL403G GREINING REIKNIRITA

25. Neðri mörk

Hjálmtýr Hafsteinsson

Vor 2022

Í þessum fyrirlestri

- Neðri mörk á tímaflækju verkefna
  - Ákvörðunartré
  - Andstæðingsrök

Aukaefni (DC 12)

- *NP*-erfið (*NP-hard*) verkefni
  - Flækjustigsflokkar (*complexity classes*)
  - *CircuitSAT* verkefnið
  - Önnur *NP-erfið* verkefni

12.1 − 12.3, 12.4

Neðri mörk (*lower bounds*)

- Höfum hingað til verið að finna reiknirit sem leysa verkefni á tilteknum tíma
- Skoðum nú leiðir til að sýna að sum verkefni er ekki hægt að leysa á tilteknum tíma

Höfum verkefni P:

Viljum vita besta tíma sem hægt er að leysa P á

Hvert reiknirit A gefur okkur efri mörk á því

$$T_A(n) = \max_{|X|=n} T_A(X)$$

X er inntak í P af stærð n

Ef við getum sannað neðri mörk $T_L(n)$ þá vitum við betur hvar við stöndum

Ef neðri mörkin ($T_L(n)$) eru jöfn efri mörkum reiknirits A ($T_A(n)$) þá er reikniritið það besta mögulega (*optimal*)

Efri mörk

$T_A(n)$

Bil?

$T_L(n)$

Neðri mörk

Aðferðir til að finna neðri mörk

- Við höfum tvær almennar aðferðir til að sanna neðri mörk fyrir verkefni:

Ákvörðunartré (*decision trees*)

Notar fjölda mögulegra útkoma til
að rökstyðja nauðsynlega tímaflækju

Andstæðingsrök (*adversary argument*)

Notar andstæðing sem velur slæmt inntak
þ.a. öll reiknirit muni taka langan tíma

Athugið að nú erum við að vinna með verkefnin sjálf, ekki einstök reiknirit

Rökin sem við notum þurfa að eiga við um öll möguleg reiknirit fyrir verkefnið

Þetta getur oft reynst erfitt, því við erum vön að "hugsa í lausnum"

þ.e. reikniritum!

Ákvörðunartré

- Getum litið á ákvörðunartré sem eina gerð reiknilíkans

Oftast tvíundartré   Hnútar eru samanburður (ákvörðun)

Hvert lauf er útkoma í verkefninu

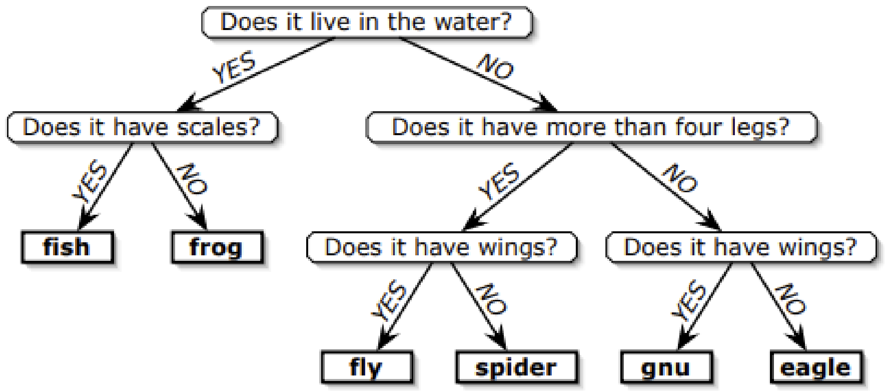
Allar mögulegar útkomur þurfa að koma fyrir sem lauf í trénu

Dæmi:

Leikurinn "Finna dýrið"

Þurfum að geta greint á milli allra mögulegra útkoma

Aðeins 6 möguleg dýr!



Hæð trésins samsvarar fjölda samanburða sem þarf að lágmarki að gera

Dæmi um ákvörðunartré

- Röðun: Gefinn listi $x_1, x_2, \ldots, x_n$ af n ólíkum tölum, finna umröðun π þ.a.

$$x_{\pi(1)} < x_{\pi(2)} < \ldots < x_{\pi(n)}$$

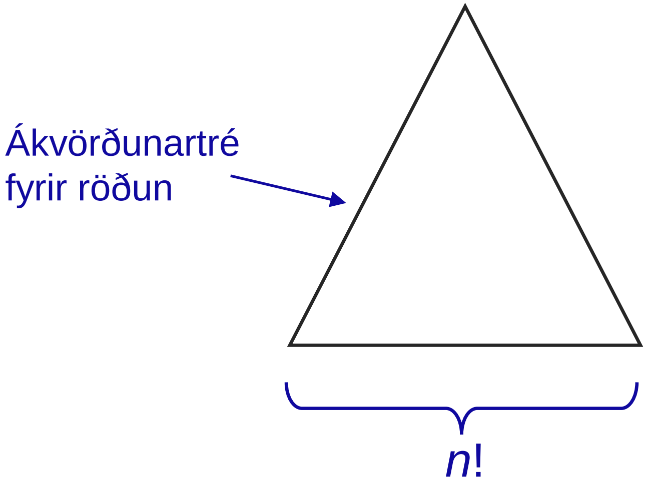
Það eru $n!$ ólíkar umraðanir á n stökum, svo fjöldi laufa í ákvörðunartrénu er $n!$

Þetta er tvíundartré, því hver ákvörðun gefur aðeins tvær niðurstöður (já/nei, satt/ósatt, ...)

Tvíundartré með $n!$ lauf hlýtur að hafa hæð > $\log_2(n!)$

Getum skrifað $n! \approx \left(\frac{n}{e}\right)^n$

Fáum þá neðri mörkin $\left\lceil \log_2\left(\frac{n}{e}\right)^n \right\rceil = \lceil n \cdot \ln n - n \cdot \ln e \rceil \quad = \Omega(n \cdot \log n)$



Þessi neðri mörk gilda aðeins fyrir röðun með samanburðum!

Andstæðingsrök

- Höfum andstæðing sem velur inntakið þannig að reiknirit taki sem lengstan tíma
  - Þurfum að smíða þennan andstæðing til að ráða við öll reiknirit

Dæmi: Finna stærsta gildið í $[x_1, x_2, ..., x_n]$

Andstæðingur:

Byrjar með $x_i = i$ og svarar reikniritinu í samræmi við það

Ath. x_n verður ekki merkt, því $x_i = i$

Í hvert sinn sem andstæðingurinn svarar $x_i < x_j$ þá merkir hann x_i sem stak sem getur ekki verið stærst

Ef reikniritið gerir færri en $n-1$ samanburð, þá er a.m.k. eitt stak x_k ($\neq x_n$) sem er ómerkt

Andstæðingurinn getur því breytt gildi x_k yfir í $n+1$ og látið það verða stærsta stakið án þess að lenda í mótsögn við sjálfan sig (fyrri útkomur)

Reikniritið verður því að framkvæma $n-1$ samanburð til þess að tryggja að svarið sé rétt. Annars getur andstæðingurinn klekkt á því!

Ath:
Andstæðingurinn gefur sér ekkert um röð samanburða

Finna minnsta og stærsta stak

- Höfum óraðaðan lista X með n stökum $[x_1, x_2, ..., x_n]$
- Finna minnsta og stærsta stakið

Reiknirit:

$\left\lfloor \frac{n}{2} \right\rfloor$ - Bera saman hliðstæð stök

Setja minna stakið í S og það stærra í L

$2 \cdot \left\lceil \frac{n}{2} \right\rceil - 1$
- Finna minnsta stakið í S
- Finna stærsta stakið í L

Stök í S koma ekki til greina sem stærstu stök

Stök í L koma ekki til greina sem minnstu stök

$\left\lceil \frac{3n}{2} \right\rceil - 2$ samanburðir

Þetta eru þá efri mörk á fjölda samanburða

Er til betra reiknirit?

Neðri mörk á verkefninu

Andstæðingur:

Í upphafi eru öll stök bæði með **+** og **−**

Merkja öll stök með **+** ef þau gætu verið stærsta og með **−** ef þau gætu verið minnsta

Ef reiknirit spyr um tvö stök með **+** og **−** þá velur andstæðingurinn annað til að vera minna stakið, tekur **+**-inn af því og **−**-inn af hinu stakinu

Í öllum öðrum tilfellum svarar andstæðingurinn þannig að aðeins þarf að taka eitt merki af stökunum

t.d. ef **+/−** vs **+** eða **−** vs **+/−** þá verður það að **−** vs **+**

Í upphafi eru 2*n* merki

Í lokin má aðeins eitt stak vera merkt **+** og aðeins eitt stak merkt **−**

Þarf að eyða $2n - 2$ merkjum

Það eru aðeins $\lfloor n/2 \rfloor$ samanburðir sem eyða tveimur merkjum, allir aðrir eyða aðeins einu

Andstæðingurinn neyðir öll reiknirit til að gera a.m.k. $2n - 2 - \lfloor n/2 \rfloor \;=\; \left\lceil \frac{3n}{2} \right\rceil - 2$ samanburði

Efri og neðri mörk eru því þau sömu!

Ekki hægt að bæta reikniritið

Æfingar

- Gefin eru 8 stök [12, 11, 8, 1, 13, 4, 7, 3]. Notið reikniritið hér á undan til að finna bæði stærsta og minnsta stakið. Hvað eru notaðir margir samanburðir?

- En ef við finnum stærsta og minnsta sjálfstætt?

NP-erfið verkefni

- Nær öll reikniritin hingað til hafa verið margliðutíma
  - þ.e. með tímaflækju $O(n^k)$ fyrir fasta k

Er hægt að leysa öll verkefni á margliðutíma?

NEI!

- Sum verkefni eru <u>óleysanleg</u>!

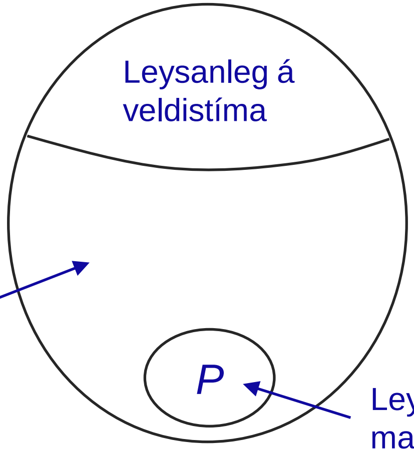
Til dæmis Stöðvunarverkefnið (*Halting problem*)

Mun forritið *P* hætta keyrslu?
Notum hornalínuröksemd
(*diagonal argument*)

- Aðeins hægt að leysa sum verkefni með veldistímareikniritum

Slík verkefni kallast <u>torleysanleg</u> (*intractable*)

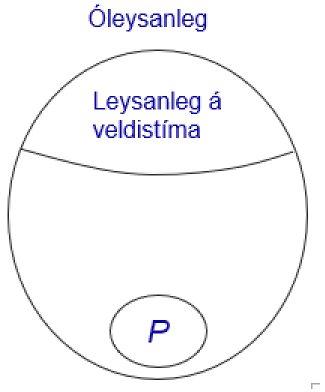
Óleysanleg



Hér er stór hópur verkefna.
Ekki vitað hvort þau eru í *P*
eða ekki

Leysanleg á
margliðutíma

Flækjustigsflokkar

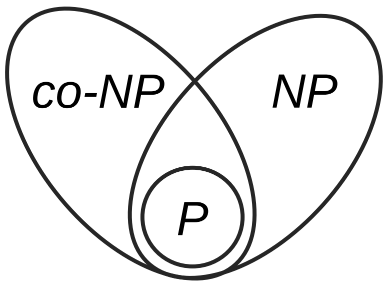


- Reynum að finna eitthvað skipulag á þessum "milli"-verkefnum

P er mengi þeirra ákvörðunarverkefna sem hægt er að leysa á margliðutíma

NP er mengi þeirra ákvörðunarverkefna, þ.a. ef svarið er **JÁ** þá er til staðfesting á því sem hægt er að sannreyna á margliðutíma

$co\text{-}NP$ er mengi þeirra ákvörðunarverkefna, þ.a. ef svarið er **NEI** þá er til staðfesting á því sem hægt er að sannreyna á margliðutíma



Líkleg mynd

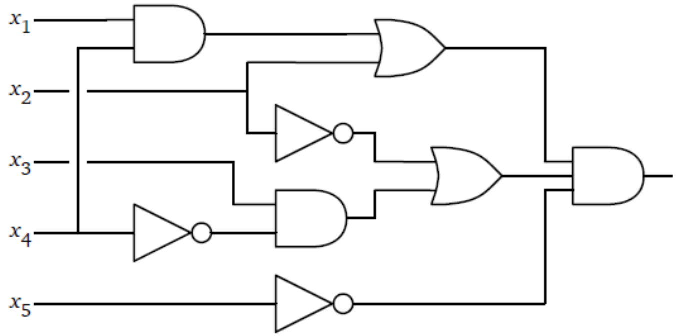
Við vitum ekki hvort $P = NP$

Við vitum ekki hvort $co\text{-}NP = NP$

Við vitum reyndar að $co\text{-}P = P$, svo ef $co\text{-}NP \neq NP$, þá gildir að $P \neq NP$

Circuit-SAT verkefnið

- Gefin rökrás með *n* inntaksvírum og einum úttaksvír
- Er hægt að gefa inntaksvírunum gildi (0/1, af/á, ósatt/satt ...) þannig að úttakið sé 1 (á, satt, ...)?



Engum hefur tekist að leysa þetta verkefni á almennan hátt, nema með því að prófa alla 2^n möguleikana á inntakinu

Svo það er líklega ekki í *P*

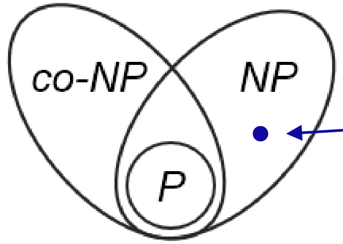
Þetta verkefni er í *NP*:

Ef við fáum gildi á *n* inntaksvírunum, þá getum við rakið okkur í gegnum rökrásina og staðfest að úttakið sé 1

Tekur augljóslega aðeins margliðutíma í stærð rökrásarinnar (líklega línulegan)

Verkefnið er líklega ekki í *co-NP*:

Hvernig staðfestum við á einfaldan hátt að ekkert mögulegt inntak gefi 1?



Líklegur staður

NP-erfið verkefni

- Verkefni Π er *NP-erfitt* (*NP-hard*) ef margliðutíma reiknirit fyrir Π gæfi margliðutíma reiknirit fyrir <u>öll verkefni í *NP*</u>

$\boxed{\text{Verkefni } \Pi \text{ er } \textit{NP-erfitt} \iff \text{Ef } \Pi \text{ er leysanlegt á margliðutíma þá er } P = NP}$

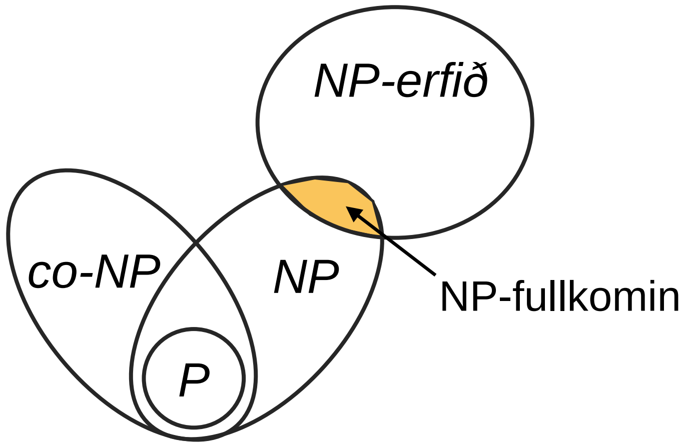
Þetta gerir það ennþá ólíklegra að *P* = *NP*

Lengi verið reynt að finna margliðutíma reiknirit fyrir þessi verkefni og ekki tekist, kannski vegna þess að það er ekki hægt!?

Sum *NP-erfið* verkefni eru líka í *NP*

Þau verkefni eru kölluð <u>*NP-fullkomin*</u> (*NP-complete*)

Þetta eru þá "erfiðustu" verkefnin í *NP*



NP-fullkomin verkefni

- Fyrsta *NP-fullkomna* verkefnið: *Circuit-SAT*

Sannað af Stephen Cook árið 1971

Kóðar framkvæmd Turing vélar sem rökrás, þ.a. vélin samþykkir inntak þþaa rökrásin sé fullnægjanleg

Sönnum að önnur verkefni séu *NP-fullkomin* (eða *NP-erfið*) með umritunum (*reductions*)

Umritum þekkt *NP-erfitt* verkefni *X* yfir í okkar verkefni *A*

Þá gætum við leyst *X* á margliðutíma ef við gætum leyst okkar verkefni á margliðutíma

og þá væri *P* = *NP*

Við tökum inntak *x* í *X*, umritum við það í inntak fyrir *A*, *a*=*f*(*x*)
Þannig að *x* er **já**-inntak í *X* þþaa *a* er **já**-inntak í *A*

Ef við gætum nú leyst *A* á margliðutíma þá getum við notað það reiknirit til að leysa *X* á margliðutíma - **mótsögn!**

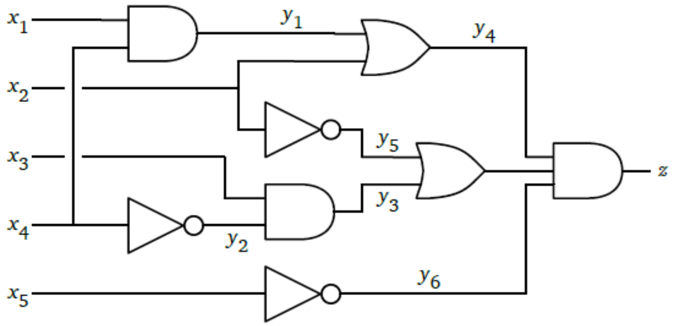
Umritun (*reduction*)

Getum umritað *Circuit-SAT* yfir í *SAT*

Verkefnið *SAT* spyr hvort <u>rökyrðing</u> sé fullnægjanleg

Til dæmis: $(a \vee b \vee c \vee \bar{d}) \Leftrightarrow ((b \wedge \bar{c}) \vee \overline{(\bar{a} \Rightarrow d)} \vee (c \neq a \wedge b))$

Umritum rökrás *K* yfir í rökyrðingu Φ:



$$(y_1 = x_1 \wedge x_4) \wedge (y_2 = \overline{x_4}) \wedge (y_3 = x_3 \wedge y_2) \wedge (y_4 = y_1 \vee x_2) \wedge$$
$$(y_5 = \overline{x_2}) \wedge (y_6 = \overline{x_5}) \wedge (y_7 = y_3 \vee y_5) \wedge (z = y_4 \wedge y_7 \wedge y_6) \wedge z$$

Þurfum að sýna:

- *K* er fullnægjanleg þþaa Φ er fullnægjanleg
- Umritunin tekur aðeins margliðutíma

Ef *SAT* leysanlegt á margliðutíma þá
höfum við margliðutímareiknirit fyrir *Circuit-SAT*:

<u>CIRCUITSAT(K):</u>
  transcribe K into a boolean formula Φ
  return SAT(Φ)   ⟨⟨★★★★MAGIC★★★★⟩⟩

Dæmi um NP-erfið verkefni

- Þúsundir verkefna eru *NP-erfið*
  - Mörg þeirra eru *NP-fullkomin*
- Oft erfitt að greina á milli auðveldra verkefna og NP-erfiðra

"*<u>Lengsti</u> einfaldi vegur á milli tveggja hnúta í neti*" er *NP-fullkomið* verkefni

"*<u>Stysti</u> einfaldi vegur á milli tveggja hnúta í neti*" er í *P*

Hamilton-hringur fer í gegnum <u>alla hnúta</u> nets nákvæmlega einu sinni

að ákvarða hvort net *G* hafi Hamilton-hring er *NP-fullkomið*

Euler-hringur fer í gegnum <u>allar stikur</u> nets nákvæmlega einu sinni

G hefur Euler-hring þþaa allir hnútar hafi jafntölu gráðu

Fleiri *NP-fullkomin* verkefni

- Klíka (*Clique*)

  Hefur net G klíku (mengi hnúta sem allir eru tengdir) af stærð k?

- Farandsalaverkefnið (*TSP*)

  Eitt gagnlegasta verkefnið (t.d. tæming póstkassa, logsuða vélmenna, ...)

  Er til Hamilton-hringur í netinu G sem er léttari en k?

- Hnútalitun (*Vertex coloring*)

  Notkun: lausn Sudoku þrauta, úthlutun gista í þýðendum, ...

  Er hægt að lita hnúta G með k litum þ.a. allar stikur séu á milli hnúta með ólíkan lit?

- Hlutmengissumma (*Subset sum*)

  Gefið mengi X af heiltölum og heiltalan T, er til hlutmengi í X með summuna T?

  Sáum gervi-margliðutíma reiknirit í kafla 3, tími $O(nT)$

NP-erfiðir leikir

- Mjög margir leikir eða þrautir eru NP-erfiðir

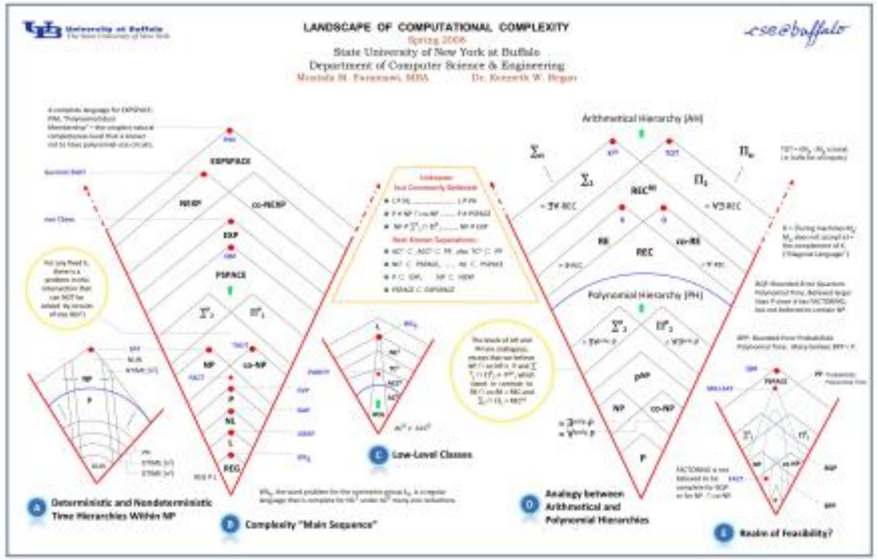
Þetta eru þá almennar ákvörðunarútgáfur af leikjunum

  - Minesweeper (leikur í Windows)
  - Tetris
  - Sudoku
  - Klondike (spilakapall)
  - Pac-Man
  - Super Mario Bros.
  - Candy Crush Saga
  - 2048
  - Lemmings
  - Dammtafl (*checkers*, *draughts*)
  - ...

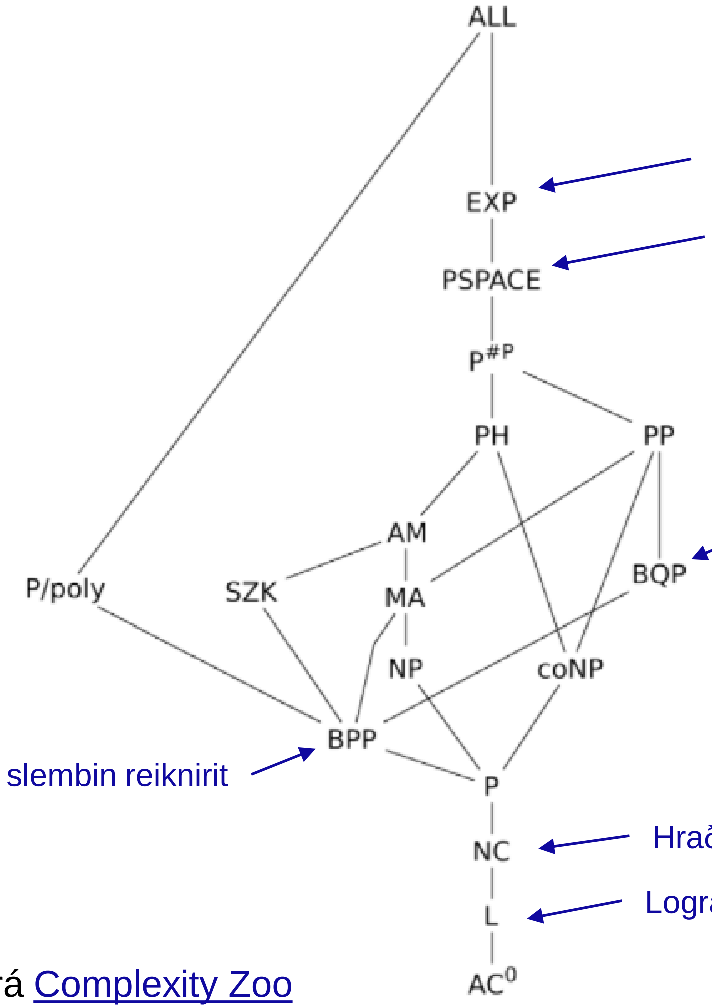
Listi af NP-fullkomnum verkefnum

Aðrir flækjustigsflokkar

- Til mikið af flækjustigsflokkum
  - Erfiðari: *PSPACE*, *EXP-TIME*, ...
  - Léttari (inni í *P*): *NC*, *L*, ...



[Áhugavert veggspjald]



Veldistíma

Margliðu magn minnis

Hraðvirk skammta (*quantum*) reiknirit

Margliðutíma slembin reiknirit

Hraðvirk samhliða reiknirit

Logra magn minnis

Mynd frá [Complexity Zoo]

[Mun ítarlegri mynd]

Fyrirlestraæfingar

1. Það þarf $n - 1$ samanburð til að finna minnsta stakið í n-staka óröðuðum lista. Hvað þarf marga samanburði til að finna næstminnsta stakið?

2. Sýnið margliðutíma reiknirit til að ákvarða hvort til sé Hamilton-vegur í óhringuðu stefnuneti (*DAG*) (almenna verkefnið er *NP-fullkomið*)

3. Verkefnið 3-Klíka (*3-Clique*) er að ákvarða hvort netið G hafi klíku af stærð 3. Er 3-Klíka *NP-erfitt*?