Bonyolultságelmélet

Monday 26th September, 2016, 19:19

Az **NP** osztály

Az **NP** osztályba mindazok a problémák tartoznak, melyek eldönthetők polinom időigényű nemdeterminisztikus programmal.

Azaz, $L \in \mathbf{NP}$, ha van olyan M polinom időkorlátos nemdeterminisztikus program, melyre

- ha $x \in L$, akkor M-nek létezik elfogadó futása x-en, és
- ha $x \notin L$, akkor M-nek minden futása elutasítja x-et.

Példa

Hamilton-Út, Sat és $3 -$ Színezés **NP**-beli problémák.

Mivel minden $f(n)$ időigényű program felfogható $f(n)$ időigényű nemdeterminisztikus programként is, így nyilván $\mathbf{P} \subseteq \mathbf{NP}$.

Polinomidőben verifikálhatóság

Az előző példákban az algoritmusok mind a következő sémára épültek fel:

- nemdeterminisztikusan generáltak valami „bizonyítékot” arra, hogy az input a problémának egy IGEN példánya, majd
- determinisztikusan ellenőrizték, hogy tényleg jó bizonyítékot sikerült-e generálni.

Ez a séma pontosan karakterizálja az **NP** osztályt!

Polinomidőben verifikálhatóság

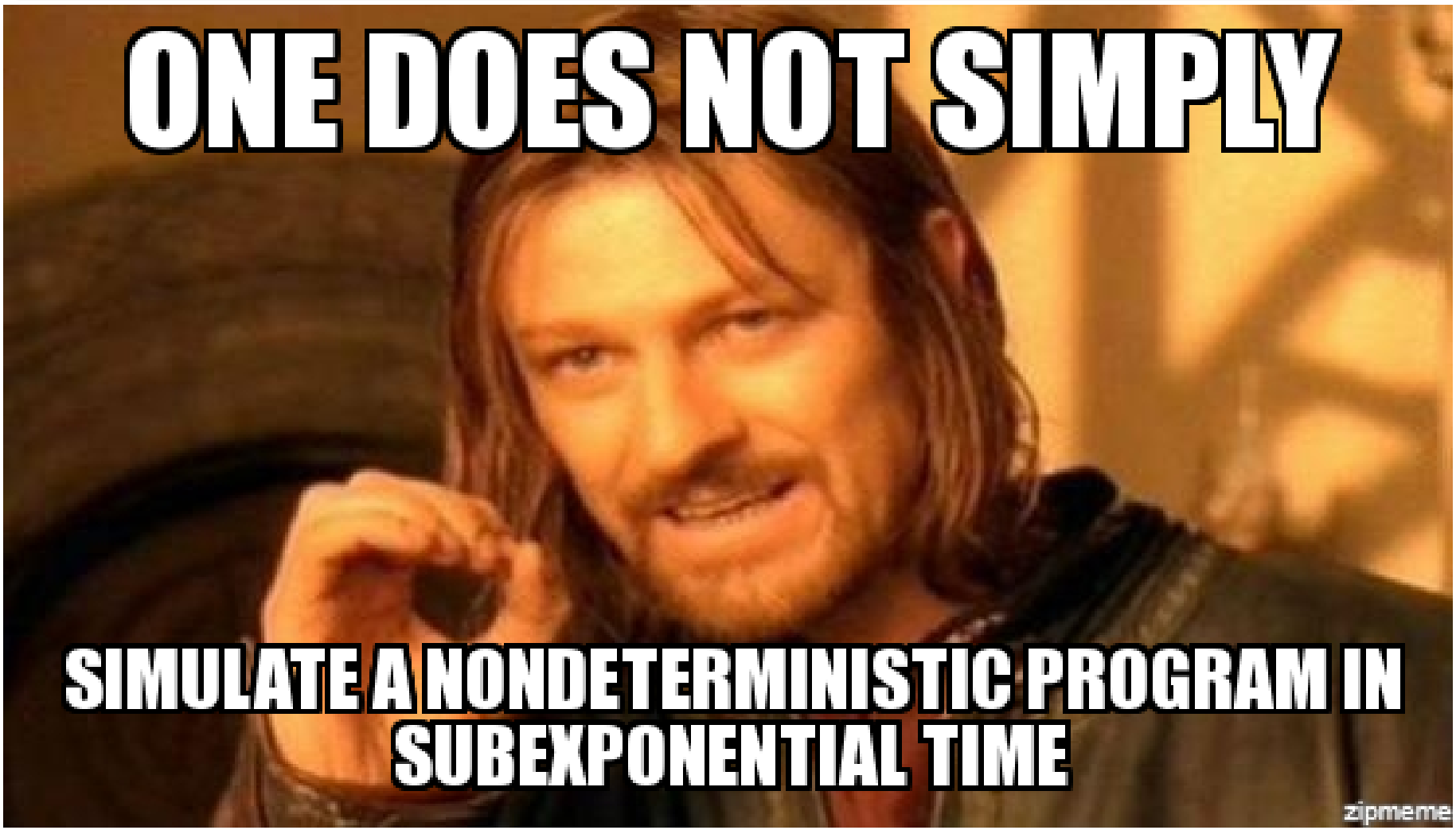
Tétel

Egy L probléma pontosan akkor van az $\mathbf{NP}$ osztályban, ha létezik egy olyan $K \subseteq \mathbb{N}^* \times \mathbb{N}^*$ reláció „inputok" és „tanúk" közt, melyre a következők fennállnak:

- létezik olyan k konstans, melyre ha $(x, y) \in K$, akkor $|y| \leq |x|^k$ (azaz az egyes inputokhoz tartozó tanúk „rövidek");
- K polinom időben eldönthető (azaz van hatékony algoritmus, mely az x, y párra eldönti, hogy y az x-hez tartozó tanú-e);
- $x \in L$ pontosan akkor, ha $\exists y : \ (x, y) \in K$ (azok az inputok a probléma IGEN példányai, melyekre van tanú).

A HAMILTON-ÚT problémánál egy gráfhoz tartozó tanú pl. maga egy Hamilton-út: lineáris méretű és könnyű ellenőrizni, hogy tényleg Hamilton-út vagy sem, és – nyilván – pontosan az IGEN példányokra van ilyen tanú.

Hatékonyak ezek a „polinomidejű” algoritmusok?



Nehézség, teljesség

Legyen $\mathcal{C}$ (eldöntési) problémák egy osztálya.

- Az A probléma $\boldsymbol{\mathcal{C}}$-nehéz, ha $\mathcal{C}$ minden eleme visszavezethető rá.
- Ha még $\mathrm{A} \in \mathcal{C}$ is, akkor A $\boldsymbol{\mathcal{C}}$-teljes.

Észrevétel

Ha egy $\mathbf{NP}$-teljes probléma polinomidőben eldönthető, akkor (és csak akkor) $\mathbf{P} = \mathbf{NP}$.
(Hiszen akkor $\mathbf{NP}$ bármelyik problémája eldönthető egy polinomidejű visszavezetés és egy, a fenti problémát eldöntő polinomidejű algoritmus kompozíciójával, tehát polinomidőben.)

$\mathcal{C}$-nehéz problémák keresése

Észrevétel

A hatékony visszavezetés tranzitív.

Ezért hogy megmutassuk egy A probléma $\mathcal{C}$-nehézségét, mindössze vissza kell rá vezetnünk egy másik, ismerten $\mathcal{C}$-nehéz B problémát. Hisz ekkor

- minden $\mathcal{C}$-beli visszavezethető B-re,
- B visszavezethető A-ra,

ezért (tranzitivitás!) minden $\mathcal{C}$-beli is visszavezethető A-ra.

Csakhogy...

...az első $\mathcal{C}$-nehéz problémát hogy kapjuk?

Visszavezetésre való zártság

Definíció

Azt mondjuk, hogy egy $\mathcal{C}$ bonyolultsági osztály zárt a visszavezetésre, ha valahányszor $A \leq_{\mathcal{P}} B$ és $B \in \mathcal{C}$, mindig teljesül $A \in \mathcal{C}$ is.

Állítás

Az eddig látott bonyolultsági osztályok ($\mathbf{P}$, $\mathbf{NP}$, $\mathbf{R}$, $\mathbf{RE}$) zártak a visszavezetésre.

Ha $\mathcal{C}$ zárt a visszavezetésre és A egy $\mathcal{C}$-teljes probléma, akkor $\mathcal{C}$-ben pontosan az A-ra visszavezethető problémák vannak ($\mathcal{C} = \{B : B \leq_{\mathcal{P}} A\}$). Tehát A reprezentálja az egész osztályt!

Megjegyzés

$\mathbf{P}$ bármely két nemtriviális A, B elemére igaz, hogy $A \leq_{\mathcal{P}} B$.

Első **RE**-teljes problémánk

MEGÁLLÁS **RE**-teljes.

Legyen $\mathrm{A} \in \mathbf{RE}$ egy tetszőleges probléma, amit felismer az M RAM-program. Akkor tetszőleges x inputra

$$x \in A \;\Leftrightarrow\; (M; x) \in \text{MEGÁLLÁS},$$

hiszen ha M az A problémát ismeri fel, akkor pontosan az IGEN példányain áll meg (ACCEPT választ adva), így az $x \mapsto (M; x)$ leképezés egy (hatékony) visszavezetése A-nak a MEGÁLLÁS problémára.

Így ha $\text{MEGÁLLÁS} \leq \mathrm{A}$ valamely A problémára, akkor A is **RE**-nehéz:

- MINDENEN MEGÁLLÁS
- ÜRES INPUTON MEGÁLLÁS
- EKVIVALENCIA
- ...

RE-n kívül

Ha $\mathcal{C}, \mathcal{C}'$ zártak a visszavezetésre, A pedig egy $\mathcal{C}$-nehéz és $\mathcal{C}'$-beli probléma, akkor $\mathcal{C} \subseteq \mathcal{C}'$.

Láttuk, hogy $\textsc{Megállás} \leq \textsc{Ekvivalencia}$.
Az is igaz, hogy $\overline{\textsc{Megállás}} \leq \textsc{Ekvivalencia}$, tehát $\textsc{Ekvivalencia}$ $\mathbf{RE}$-nehéz és co$\mathbf{RE}$-nehéz is egyszerre.
Mivel $\mathbf{RE} \neq \mathrm{co}\mathbf{RE}$ (emiatt persze egyikük sem lehet része a másiknak, hisz ha pl. $\mathbf{RE} \subseteq \mathrm{co}\mathbf{RE}$, akkor $\mathrm{co}\mathbf{RE} \subseteq \mathrm{coco}\mathbf{RE} = \mathbf{RE}$, mely esetben egyenlőek), ez azt jelenti, hogy $\textsc{Ekvivalencia}$ nem $\mathbf{RE} \cup \mathbf{coRE}$-beli probléma!
Azaz sem olyan algoritmus nincs, mely felismeri, ha két program ekvivalens, sem olyan, mely felismeri, ha két program nem ekvivalens.

Logikai hálózatok

Definíció

Hálózat: körmentes irányított gráf,
a csúcsok címkézettek: $\wedge$, $\vee$, $\neg$, igaz, hamis, x_i
$\wedge$, $\vee$ címke esetén a csúcs befoka 2,
$\neg$ címke esetén a csúcs befoka 1,
igaz, hamis, x_i címke esetén a csúcs befoka 0.

Általában megköveteljük még, hogy pontosan egy csúcs kifoka legyen 0.

Amennyiben a változók közül az $x_1, \ldots, x_n$ fordulnak elő a hálózatban (legfeljebb), akkor a hálózat kiszámít egy $\{\mathrm{igaz}, \mathrm{hamis}\}^n \to \{\mathrm{igaz}, \mathrm{hamis}\}$ Boole-függvényt.

Hálózat-Kiértékelés

- Adott: változómentes hálózat.
- Kérdés: igaz-e az értéke?

Hálózat-Kielégíthetőség

- Adott: hálózat.
- Kérdés: a változóknak lehet-e úgy értéket adni, hogy a hálózat értéke igaz legyen?

Sat

- Adott: egy konjunktív normálformájú (ítéletkalkulus-beli) formula.
- Kérdés: kielégíthető-e?

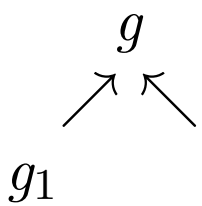
Állítás

$$\textsc{Hálózat-Kielégíthetőség} \leq_{\mathcal{P}} \textsc{Sat}.$$

A visszavezetés

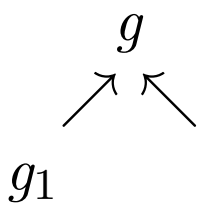
Minden g csúcsnak feleljen meg a g változó.

- $\boldsymbol{g}$ címkéje $\boldsymbol{x}$ $\mapsto$ $x \leftrightarrow g$, azaz $(\neg x \vee g) \wedge (x \vee \neg g)$
- $\boldsymbol{g}$ címkéje $\mathbf{igaz}$ $\mapsto$ g
- $\boldsymbol{g}$ címkéje $\mathbf{hamis}$ $\mapsto$ $\neg g$
- $\boldsymbol{g}$ címkéje $\vee$:



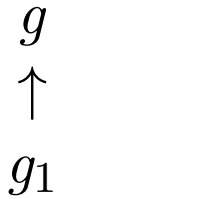
$\mapsto$ $g \leftrightarrow (g_1 \vee g_2)$, azaz $(\neg g \vee g_1 \vee g_2) \wedge (\neg g_1 \vee g) \wedge (\neg g_2 \vee g)$

- $\boldsymbol{g}$ címkéje $\wedge$:



$\mapsto$ $g \leftrightarrow (g_1 \wedge g_2)$, azaz $(\neg g \vee g_1) \wedge (\neg g \vee g_2) \wedge (\neg g_1 \vee \neg g_2 \vee g)$

- $\boldsymbol{g}$ címkéje $\neg$:



$\mapsto$ $g \leftrightarrow (\neg g_1)$, azaz $(\neg g \vee \neg g_1) \wedge (g_1 \vee g)$

- $\boldsymbol{g}$ kimenő kapu $\mapsto$ g

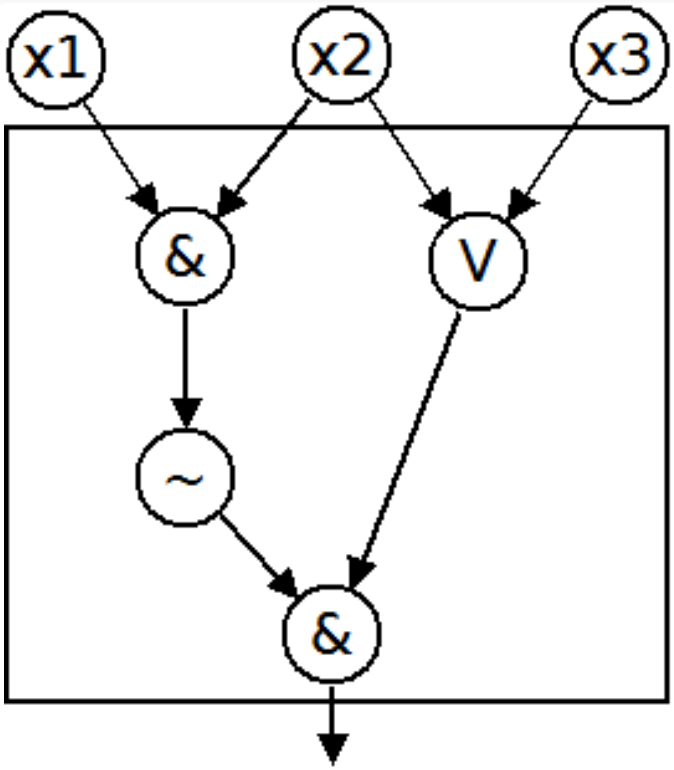
Hálózat-Kielégíthetőség $\leq_{\mathcal{P}}$ Sat

A keresett formula: a fenti formulák konjunkciója.

Adott hálózathoz a formula elkészíthető lineáris időben.
Továbbá a hálózat akkor és csak akkor kielégíthető, ha a formula az.

Hálózat-Kielégíthetőség $\leq_{\mathcal{P}}$ Sat példa

Példa



$(g_1 \leftrightarrow x_1) \wedge (g_2 \leftrightarrow x_2) \wedge (g_3 \leftrightarrow x_3) \wedge (g_4 \leftrightarrow (g_1 \wedge g_2)) \wedge (g_5 \leftrightarrow (g_2 \vee g_3)) \wedge (g_6 \leftrightarrow (\neg g_4)) \wedge (g_7 \leftrightarrow (g_5 \wedge g_6)) \wedge g_7.$

CNF alakban:

$(\neg g_1 \vee x_1) \wedge (g_1 \vee \neg x_1) \wedge (\neg g_2 \vee x_2) \wedge (g_2 \vee \neg x_2) \wedge (\neg g_3 \vee x_3) \wedge (g_3 \vee \neg x_3) \wedge (\neg g_4 \vee g_1) \wedge (\neg g_4 \vee g_2) \wedge (g_4 \vee \neg g_1 \vee \neg g_2) \wedge (\neg g_5 \vee g_2 \vee g_3) \wedge (g_5 \vee \neg g_2) \wedge (g_5 \vee \neg g_3) \wedge (\neg g_6 \vee \neg g_4) \wedge (g_6 \vee g_4) \wedge (\neg g_7 \vee g_5) \wedge (\neg g_7 \vee g_6) \wedge (g_7 \vee \neg g_5 \vee \neg g_6) \wedge g_7.$

Első kiszámítható teljes problémáink

Emlékezzünk: egy $\mathcal{C}$-beli probléma $\mathcal{C}$-teljes, ha minden $\mathcal{C}$-beli probléma visszavezethető rá.

Eddig még csak eldönthetetlen, **RE**-teljes problémákkal találkoztunk: a MEGÁLLÁS ilyen volt.

A HÁLÓZAT-KIÉRTÉKELÉS probléma **P**-beli.

Tétel

A HÁLÓZAT-KIELÉGÍTHETŐSÉG probléma **NP**-teljes.

Következmény (Cook tétele)

A SAT probléma is **NP**-teljes.

Sat **NP**-teljességéről

Mivel a Sat probléma **NP**-teljes,

- ha polinomidőben megoldható lenne, akkor $\mathbf{P} = \mathbf{NP}$. (Ebben persze lehet hinni. De eddig még senki nem oldotta meg polinomidőben. . .)
- Nem ismert rá szubexponenciális algoritmus.
- Ilyenkor bevethetők (a teljesség igénye nélkül):
  - heurisztikák
  - randomizált algoritmusok alkalmazása
  - a követelmények relaxálása. . . (pl. nem az összes, de minél több klóz egyszerre történő kielégítése)
  - . . . majd a relaxált követelmények approximálása
  - vagy olyan speciális esetek keresése, melyre van hatékony algoritmus.

Összefoglalás

- Megismertük a polinomidőben verifikálhatóságot és láttuk, hogy pontosan az **NP**-beli problémák ilyenek.
- Megismertük a $\mathcal{C}$-nehézség és $\mathcal{C}$-teljesség fogalmát.
- Láttuk, hogy $\mathbf{P} = \mathbf{NP}$ pontosan akkor igaz, ha valamelyik **NP**-teljes probléma megoldható polinomidőben.
- Láttuk, hogy a hatékony visszavezetés tranzitív.
- Megismertük a visszavezetésre való zártságot és láttuk, hogy az eddigi osztályaink zártak a visszavezetésre.
- Láttuk, hogy egy visszavezetésre zárt osztályt karakterizálnak a teljes problémái.
- Láttuk, hogy a megállási probléma **RE**-teljes.
- Láttuk, hogy a programok ekvivalenciája nehezebb: nincs $\mathbf{RE} \cup \text{co}\mathbf{RE}$-ben sem.
- Megismertük a logikai hálózatokat, a Hálózat-Kiértékelés, Hálózat-kielégíthetőség és Sat :P problémákat.
- Láttuk, hogy Hálózat-Kiértékelés **P**-ben van.
- Cook tétele: Hálózat-Kielégíthetőség **NP**-teljes.
- Láttuk, hogy Hálózat-Kielégíthetőség $\leq$ Sat. Így Sat is **NP**-teljes.