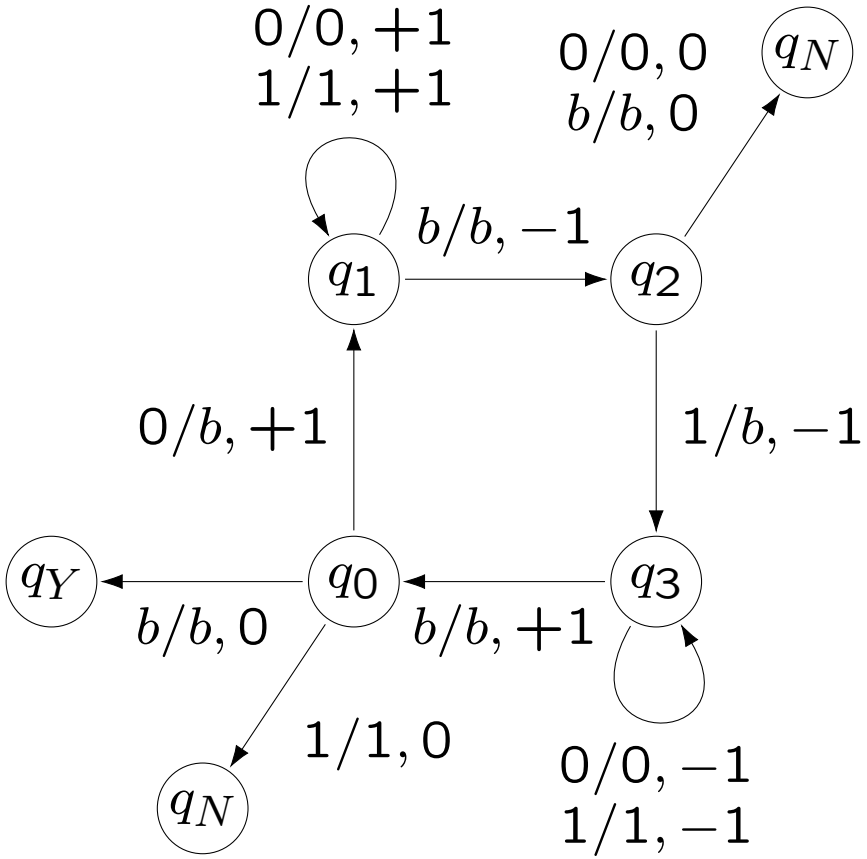
Twaalfde college complexiteit

7 mei 2019

NP-volledigheid IV
Cook-Levin
Savitch

Turing machine $\{0^n1^n \mid n \geq 0\}$



q_0 op eerste symbool links, schrap 0
q_1 ga naar rechts, tot b
q_2 op laatste symbool rechts, schrap 1
q_3 ga naar links, tot b

	0	1	
q_0	$q_1, b, +1$	$q_N, 1, 0$	$q_Y, b, 0$
q_1	$q_1, 0, +1$	$q_1, 1, +1$	$q_2, b, -1$
q_2	$q_N, 0, 0$	$q_3, b, -1$	$q_N, b, 0$
q_3	$q_3, 0, -1$	$q_3, 1, -1$	$q_0, b, +1$

deterministisch: functie
$\delta : Q \times \{0, 1, b\} \mapsto$
$Q \times \{0, 1, b\} \times \{-1, 0, +1\}$

In 1971 bewees Stephen Cook op een directe manier (dus door een reductie te geven van alle problemen uit $\mathcal{NP}$ naar SAT) dat SAT NP-volledig is.

Stelling
Gegeven een *willekeurig* probleem $P \in \mathcal{NP}$. Dan is P reduceerbaar tot SAT: $P \leq_{\mathsf{P}}$ SAT.

Sindsdien is met behulp van de reductiemethode van zeer veel bekende problemen aangetoond dat ze NP-volledig zijn (allereerst door Richard Karp, 1972).
Bijvoorbeeld voor enige voorbeeldproblemen:

$$\begin{aligned} \text{SAT} \leq_{\mathsf{P}} \text{3SAT} \leq_{\mathsf{P}} \text{Kliek} \leq_{\mathsf{P}} \text{VC} \\ \text{3SAT} \leq_{\mathsf{P}} \text{HC2} \leq_{\mathsf{P}} \text{TSP} \\ \text{SAT} \leq_{\mathsf{P}} \text{3Kleur} \leq_{\mathsf{P}} \text{4Kleur} \end{aligned}$$

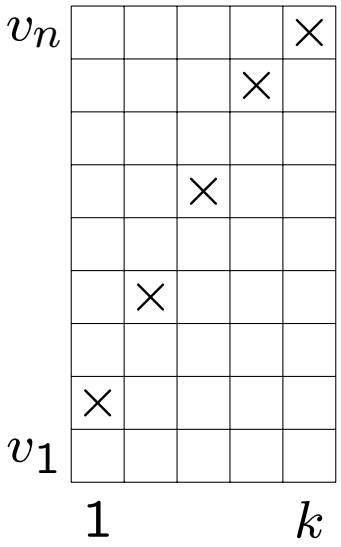
Stelling (Stephen Cook, 1971; Leonid Levin, 1973)

SAT $\in \mathcal{NPC}$.

Stephen Cook, The complexity of theorem proving procedures. Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151–158, 1971. doi:10.1145/800157.805047

Kliek: Zijn er k-knopen in $G = (V, E)$, onderling verbonden?

Kliek $\leq_P$ SAT



variabelen x_{iv} $\quad 1 \leq i \leq k,\ v \in V$

true $\iff$ v is de i-de knoop in de kliek

clauses

elke kolom i bevat een keuze

$(\bigvee_{v \in V} x_{iv})$

maar nooit twee in dezelfde rij of kolom

$\left(\neg x_{iv} \vee \neg x_{jv}\right)$, $\quad (\neg x_{iu} \vee \neg x_{iv})$

$1 \leq i < j \leq k,\ v \in V$ $\quad$ $1 \leq i \leq k,\ u, v \in V,\ u \neq v$

knopen zonder lijn mogen niet samen gekozen

$\left(\neg x_{iu} \vee \neg x_{jv}\right)$ $\quad$ als $(u, v) \notin E$

$1 \leq i < j \leq k,\ u, v \in V$

Omdat $k \leq |V|$ zijn dit $O(|V|^2)$ variabelen en $O(|V|^4)$ clauses.

Schets van het bewijs

1. Omdat $P \in \mathcal{NP}$, is er een niet-deterministisch algoritme A (een niet-deterministische Turingmachine) voor P. Verder is A polynomiaal begrensd ($p(|x|)$ op invoer x).
2. Dit algoritme zal voor elke invoer x van P gemodelleerd worden als een logische formule $\phi = T(x)$ in CNF: deze ϕ beschrijft de gehele berekening van A, werkend op x.
3. De formule ϕ is weliswaar lang, maar heeft een polynomiale lengte (in $|x|$).
4. Voor een ja-instantie x vergt de executie hooguit $N = p(|x|)$ stappen (A is polynomiaal begrensd).
5. Een waarmakende waardering voor ϕ correspondeert precies met een executie van A die een "ja" produceert. Dus er is een waardering die ϕ waarmaakt $\Leftrightarrow$ x is een ja-instantie voor P.

Een NDTM-programma (algemene beschrijving*) bevat:

- Γ: een eindige verzameling tape-symbolen (waaronder invoer-alfabet Σ en blanco). Voorbeeld: $\Gamma = \{0, 1, B\}$.

- Q: een eindige verzameling toestanden, waaronder een begintoestand q_0 en twee eindtoestanden q_Y en q_N. Voorbeeld: $Q = \{q_0, q_1, q_2, q_3, q_Y, q_N\}$.

- $\delta \subseteq (Q \setminus \{q_Y, q_N\}) \times \Gamma \times Q \times \Gamma \times \{-1, 0, 1\}$ een transitie-relatie die bepaalt wat er kan gebeuren als in een bepaalde toestand een bepaald karakter wordt gelezen.
  extra: in toestand q_Y blijft de TM in die toestand (op dezelde plek).

In de begintoestand staat de invoerstring x op plek 1 tot en met $|x|$ en de rest van de tape is blanco. Het programma start in toestand q_0 met de lees- en schrijfkop op positie 1.

*equivalent met ons model, waarin het niet-deterministische gedeelte is samengebracht in Fase 1

Boolese variabelen in ϕ:

${Q_i}^q$: op tijdstip i is de machine in toestand $q \in Q$, $0 \leq i \leq N$

H_{ij}: op tijdstip i scant de machine cel j, $0 \leq i \leq N$, $-N \leq j \leq N$

${S_{ij}}^a$: op tijdstip i bevat cel j symbool $a \in \Gamma$, $0 \leq i \leq N$, $-N \leq j \leq N$

De formule ϕ is een conjunctie van:

- ${Q_0}^{q_0} \wedge H_{01}$

  de machine start in toestand q_0, op positie 1 — 2 clauses

- ${S_{0j}}^{x_j}$, $1 \leq j \leq |x|$ ${S_{0j}}^{B}$, $-N \leq j \leq 0$ of $|x| < j \leq N$

  x op de posities 1 t/m $|x|$; rest bevat B ... — $2N + 1$ clauses

- $Q_N^{q_Y}$

  op tijdstip N stopt de berekening in de ja-toestand

- $\left(\bigvee_{q\in Q}\ {Q_i}^q\right), \quad \left(\neg {Q_i}^p \vee \neg {Q_i}^q\right)$ $\quad 0 \le i \le N$, $p, q \in Q, p \neq q$

  te allen tijde in precies één toestand $(N+1)\frac{|Q|(|Q|+1)}{2}$ clauses

- $\left(\bigvee_{a\in\Gamma} {S_{ij}}^a\right), \quad \left(\neg {S_{ij}}^a \vee \neg {S_{ij}}^b\right)$ $\quad 0 \le i \le N,\ -N \le j \le N$, $a, b \in \Gamma, a \neq b$

  elke cel precies één symbool $(N+1)(2N+1)\frac{|\Gamma|(|\Gamma|+1)}{2}$ clauses

- $\left(\bigvee_{0\le j\le N}\ H_{ij}\right), \quad \left(\neg H_{ij} \vee \neg H_{ik}\right)$ $\quad 0 \le i \le N$, $-N \le j < k \le N$

  scant precies één cel $(N+1)(1+N(2N+1))$ clauses

- $Q_i^{\,p} \wedge H_{ij} \wedge S_{ij}^{\,a} \longrightarrow$
  $\bigvee_{(p,a,q,b,d)\in\delta} \left(Q_{i+1}^{\,q} \wedge H_{i+1,j+d} \wedge S_{i+1,j}^{\,b}\right)$

  elke stap van de machine verloopt volgens de transitie-relatie δ

  (nog wel "even" omschrijven naar CNF . . .)

- $S_{ij}^{\,a} \wedge \neg H_{ij} \longrightarrow S_{i+1,j}^{\,a}$

  een cel die op tijdstip i niet gescand wordt bevat op tijdstip $i+1$ hetzelfde symbool

- . . .

Een waarmakende waardering correspondeert zo precies met een echte executie van de niet-deterministische Turingmachine die eindigt in "ja" na een polynomiaal (nl. N) aantal stappen.

Nog wel het aantal variabelen en clausules tellen . . .

Voor de liefhebber.

$a \wedge b \to c \iff \neg a \vee \neg b \vee c$ (makkelijk)

$a \wedge b \wedge c \to \bigvee_{i \in I}(d_i \wedge e_i \wedge f_i) \overset{*}{\iff}$
$(\neg a \vee \neg b \vee \neg c \vee \bigvee_{i \in I} z_i)$
$\wedge \bigwedge_{i \in I}(\neg z_i \vee d_i) \wedge \bigwedge_{i \in I}(\neg z_i \vee e_i) \wedge \bigwedge_{i \in I}(\neg z_i \vee f_i)$
met verse variabelen z_i voor $i \in I$.

Als a, b, c alledrie waar zijn, moet er een z_i waar zijn. Vanwege de andere clauses moeten nu d_i, e_i en f_i waar zijn.

Gegeven een beslissingsprobleem P: $\overline{P}$ is dan het probleem dat dezelfde invoerverzameling heeft als P, maar dat precies het tegengestelde vraagt als P. Er geldt dus: x is een ja-instantie van $\overline{P} \iff x$ is een nee-instantie van P.

Definitie: co-$\mathcal{P} = \{P : \overline{P} \in \mathcal{P}\}$ en co-$\mathcal{NP} = \{P : \overline{P} \in \mathcal{NP}\}$

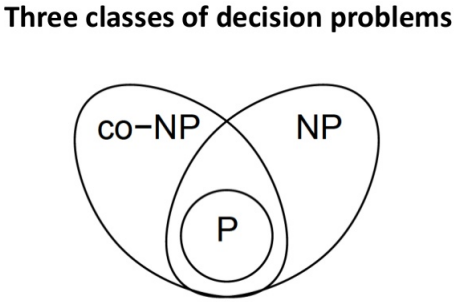
Stelling (opgave 58):

. $\mathcal{P} \subseteq$ co-$\mathcal{NP}$

. $\mathcal{P} =$ co-$\mathcal{P}$

. als $\mathcal{NP} \neq$ co-$\mathcal{NP}$ dan $\mathcal{P} \neq \mathcal{NP}$

Vraag: is $\mathcal{P} = \mathcal{NP} \cap$ co-$\mathcal{NP}$?



$\mathcal{P}$ is de verzameling beslissingsproblemen die in polynomiale *tijd* op te lossen zijn (dus met een deterministisch algoritme).

$\mathcal{PSPACE}$ is de verzameling beslissingsproblemen die in polynomiale *ruimte* (dus met gebruik van een polynomiale hoeveelheid geheugen) op te lossen zijn (deterministisch).

Het is duidelijk dat geldt: $\mathcal{P} \subseteq \mathcal{PSPACE}$
Maar er geldt ook: $\mathcal{NP} \subseteq \mathcal{PSPACE}$
Alles bij elkaar:
$\mathcal{P} \subseteq \mathcal{NP} \subseteq \mathcal{PSPACE} = \mathcal{NPSPACE} \subseteq \mathcal{EXPTIME}^*$

En ook: co-$\mathcal{NP} \subseteq \mathcal{PSPACE}$

$^*\mathcal{EXPTIME}$ (of $\mathcal{EXP}$): in exponentiële tijd oplosbaar, dus deterministisch

Stelling. NSPACE($s(n)$) $\subseteq$ SPACE($s^2(n)$)

Kunnen we een ja-toestand bereiken?
In $s(n)$ ruimte. Met $|Q|s(n) \cdot |\Gamma|^{s(n)}$ mogelijke configuraties (toestand, positie, tape inhoud) zijn dat dus maximaal exponentieel veel stappen, want herhalingen zijn nutteloos.

Dat kunnen we recursief oplossen:

reach(ini,fin,1) = step(ini,fin)

reach(ini,fin,k+1)
  for each configuration mid
    test reach(ini,mid,k) $\wedge$ reach(mid,fin,k)

recursiestapel van $s(n)$ configuraties, elk ter grootte $s(n)$

Gevolg. $\mathcal{PSPACE} = \mathcal{NPSPACE}$

Walter J. Savitch, Relationships between nondeterministic and deterministic tape complexities, Journal of Computer and System Sciences 4 (1970) 177–192, doi:10.1016/S0022-0000(70)80006-X

Open vraag: is $\mathcal{NP} =$ co-$\mathcal{NP}$?

Lemma. $\mathcal{NPSPACE} =$ co-$\mathcal{NPSPACE}$

Maar voor ruimte weten we de relatie.

Dat volgt uit het resultaat van Savitch, want voor deterministische klassen $\mathcal{X}$ geldt $\mathcal{X} =$ co-$\mathcal{X}$.

Het kan zelfs zonder extra ruimte; dus in het bijzonder geldt het voor TM's die geen extra tape mogen gebruiken buiten hun invoer (*linear bounded automaton* $s(n) = n$).

Stelling. NSPACE$(s(n))$ = co-NSPACE$(s(n))$

Neil Immerman (1988, SIAM J. Comput)
Róbert Szelepcsényi (1988, Acta Informatica)

QSAT

Gegeven een logische formule $\phi = \phi(x_1, x_2, \ldots, x_n)$ in CNF. Is de volgende formule waar (met n oneven)?

$$\exists x_1 \; \forall x_2 \; \exists x_3 \; \forall x_4 \; \cdots \; \forall x_{n-1} \; \exists x_n \; \phi(x_1, x_2, \ldots, x_n)$$

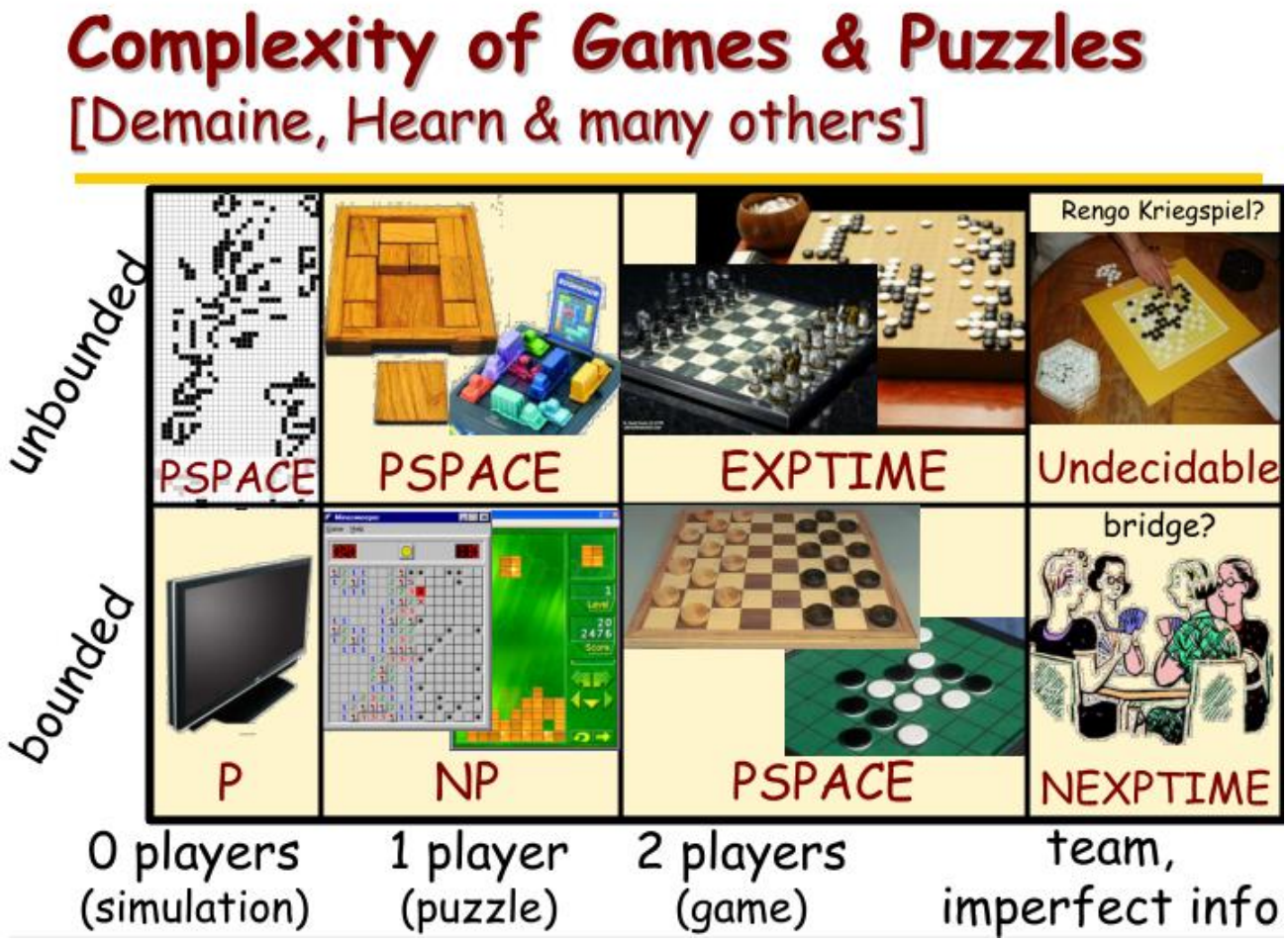
Relatie met 2-persoonsspelen:

A kiest een waarheidswaarde voor x_1, dan B voor x_2, dan A voor x_3, etcetera. Kan A ervoor zorgen dat ϕ wordt waargemaakt, wat B ook doet?

Voorbeeld 1: $(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \longrightarrow$ ja: A wint

Voorbeeld 2: $(x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \longrightarrow$ nee: A kan niet winnen

Er geldt: QSAT $\in \mathcal{PSPACE}$ (zelfs: PSPACE-volledig)


Complexity of Games & Puzzles
[Demaine, Hearn & many others]
unbounded
bounded
PSPACE
PSPACE
EXPTIME
Undecidable
Rengo Kriegspiel?
P
NP
PSPACE
NEXPTIME
bridge?
0 players
(simulation)
1 player
(puzzle)
2 players
(game)
team,
imperfect info

De wonderlijke relatie tussen "gegeneraliseerde" spellen en complexiteitsklassen, nogmaals in een tabel.

unbounded space	**PSPACE**	**PSPACE** NPSPACE	**EXPTIME** APSPACE
bounded time	**P**	**NP**	**PSPACE** AP
	deterministic *simulation*	nondeterministic *puzzle*	alternating *game*
players	zero	one	two

nondeterminism: er is een keuze die tot succes leidt
alternation: er is een keuze / alle keuzes leiden tot succes (afhankelijk van de toestand)*

*maar dat behandelen we hier niet; het is mooi geweest

- Volgende college:
  dinsdag 14 mei, 11.00 – 12.45, zaal 174

- Eerstvolgende werkcollege:
  dinsdag 14 mei, 13.30 – 15.15, zaal 174
  Opgaven 58, 59, 61, 65

- Vierde en laatste huiswerkopgave:

  * deadline: dinsdag 14 mei; LaTeX; print → college

  * www.liacs.leidenuniv.nl/~graafjmde/COMP/