

Meer reducties, SAT en klassen

Mark van den Bergh
16 mei 2025



Universiteit
Leiden

- polynomiale reductie T van P naar Q : $P \leq_P Q$
 - T beeldt elke invoer x van P af op een invoer $T(x)$ van Q
 - constructie van $T(x)$ uit x is polynomiaal (in $|x|$)
 - reductie-eigenschap: x is ja-instantie voor $P \Leftrightarrow T(x)$ is ja-instantie voor Q
 - hoop voorbeelden
- Q is NP-hard (NP-moeilijk) $\Leftrightarrow P \leq_P Q$ voor alle $P \in \mathcal{NP}$
- Q is NP-volledig ($\mathcal{NPC}$) als Q NP-hard is en $Q \in \mathcal{NP}$
- NP-volledigheidsbewijs voor Q : bewijs dat $Q \in \mathcal{NP}$ en dat $P \leq_P Q$ voor $P \in \mathcal{NPC}$
- SAT is het “oer- $\mathcal{NPC}$ -probleem”

Vandaag

- Two-Dots
- SAT solvers
- Meer complexiteitsklassen
- Quantum

FUN 2016: <https://www2.idsia.ch/cms/fun16/>

Proceedings: <https://drops.dagstuhl.de/opus/portals/lipics/index.php?semnr=16004>

Two-Dots is NP-Complete: https://drops.dagstuhl.de/opus/frontdoor.php?source_opus=5883

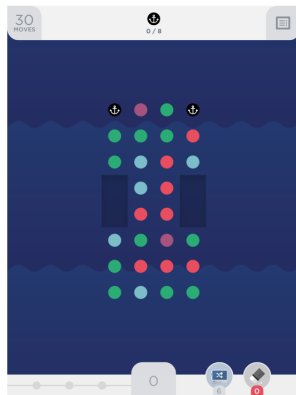
Two-Dots: <https://www.dots.co/twodots/>

Online alternatief: <https://plays.org/game/two-dots/>

Two-Dots



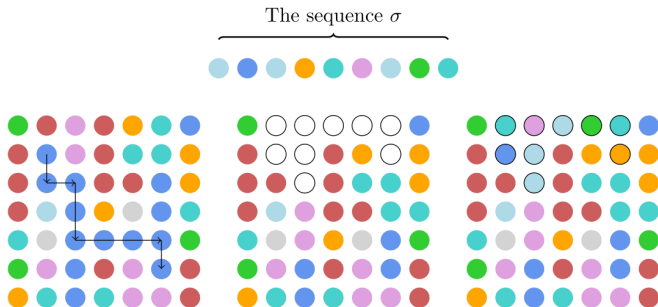
(a) Level One of Two Dots.



(b) A more advanced level of Two Dots.

Doel: gegeven k zetten, verzamel x

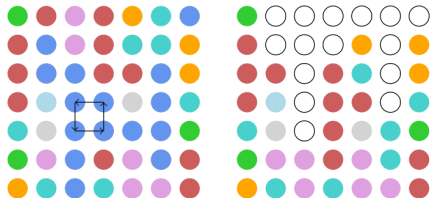
Two-Dots



■ **Figure 2** A depiction of the regular move. The first panel shows the set of locations that are committed to the move, the second panel shows the voids created, and the third panel shows how the voids are filled in accordance with σ .

Reguliere zet: orthogonaal pad, verzamelt pad

Two-Dots

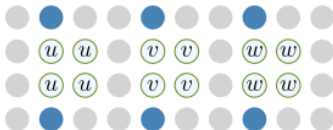


■ **Figure 3** A depiction of the square move, which has the effect of eliminating all the blue dots from the board. The example is rather similar to the above, but note the difference in the number of voids created. The process for filling up the voids is identical, and is therefore omitted.

Speciale zet: 2×2 vierkant, verzamelt hele kleur

Two-Dots en Kliek

Kliek: gegeven een graaf $G = (V, E)$ en een getal k . Bevat G een kliek (volledige deelgraaf) van grootte k ?



(a) The Vertex Gadgets.



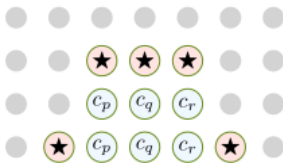
(b) A Tester Gadget for the edges (u, v) and (w, v) .

■ **Figure 4** The vertex and tester gadgets used in the reduction from CLIQUE. The colors α and β are depicted, respectively, by blue and red.

Two-Dots: $2k + \binom{k}{2}$ zetten, doel is $2k$ blauw en $4\binom{k}{2}$ rood

Two-Dots en Exact Cover: zonder de vierkante zet

Exact Cover: gegeven een verzameling $U = \{u_1, \dots, u_n\}$ en een verzameling van verzamelingen $F = \{S_1, \dots, S_m\}$ (waar $|S_i| = 3$ voor alle i). Bestaat er een deelverzameling $G \subseteq F$ zodat elk element van U precies één keer voorkomt in de elementen van G ?
(Dit is NP-volledig.)



■ **Figure 5** The gadget corresponding to a set $S = \{u_p, u_q, u_r\}$, from an instance of X3C.

Two-Dots: $n + \frac{n}{3}$ zetten, doel is 2 van elke c_i en $\frac{5n}{3}$ ★

Met verschillende reducties kan worden aangetoond dat zelfs beperkte versies NP-volledig zijn:

- Slechts één minimaal doel (vanaf 3SAT)
- Slechts drie kleuren (vanaf Exact Cover)
- Slechts twee kolommen (vanaf 3SAT)

On the Complexity of Two Dots for Narrow Boards and Few Colors
<https://drops.dagstuhl.de/opus/volltexte/2018/8798/>
(FUN 2018)

SAT-solvers zijn algoritmen of programma's die SAT-instanties oplossen, d.w.z., bepalen of ze ja- of nee-instanties zijn.

Een triviaal algoritme (exhaustive search) is te geven in $O(2^n \times |\phi|)$. Maar: kan het beter?

Handbook of Satisfiability. Armin Biere, Marijn Heule, Hans van Maaren en Toby Walsh.

<https://books.google.nl/books?isbn=160750376X> (2009)

<https://catalogue.leidenuniv.nl> (tweede editie, 2021)

SAT-conferenties en -wedstrijden

SAT 2024

Home Dates and Calls



<http://satisfiability.org/SAT24/>



<https://satcompetition.github.io/2024/>

Als in een partiële waarheidstoekenning alle literalen in een clause op één na onwaar zijn gemaakt, dan moet de laatste wel waar zijn.

Herhaaldelijk toepassen van deze regel heet *unit propagation*.

Voorbeeld:

$$(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge \neg x_1$$

Door de laatste clause moet $x_1 = \text{False}$ worden toegekend.

Hierdoor moet in de eerste clause $x_2 = \text{True}$ worden toegekend.

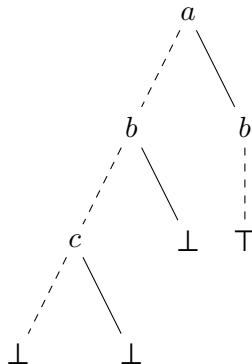
Tenslotte moet hierdoor in de tweede clause $x_3 = \text{False}$ worden toegekend.

Davis–Putnam–Logemann–Loveland (DPLL)

Backtracken m.b.v. unit propagation.

Voorbeeld:

$$\begin{aligned} & (a \vee \neg b \vee d) \\ & \wedge (a \vee \neg b \vee e) \\ & \wedge (\neg b \vee \neg d \vee \neg e) \\ & \wedge (a \vee b \vee c \vee d) \\ & \wedge (a \vee b \vee c \vee \neg d) \\ & \wedge (a \vee b \vee \neg c \vee e) \\ & \wedge (a \vee b \vee \neg c \vee \neg e) \end{aligned}$$



Conflict-driven clause learning (CDCL)

Nieuwe clause toevoegen wanneer er een conflict ontstaat.

Voorbeeld vorige slide: toewijzing $a = b = c = \text{False}$ veroorzaakt conflict (d kan geen goede toewijzing krijgen), dus voeg nieuwe clause $(a \vee b \vee c)$ toe.

Eigenlijk: houd netwerk van keuzes en implicaties bij, en kies nieuwe clause slim.

Ga vervolgens terug naar de laatste keuze waarbij een variabele in de nieuwe clause betrokken was (*non-chronological backjumping*).

- Goede heuristieken om te beslissen welke variabele nu op True/False te proberen.
- Slimme datastructuren voor snelle unit propagation.
- Af en toe clauses vergeten die aantoonbaar niet meer nuttig zijn.
- Herstarten uit (te) moeilijke gebieden.

Ondergrens SAT?

Gigantische verbeteringen, zowel algoritmisch als implementatietechnisch, maar:

Wat is de ondergrens?

Even tussendoor: niet-polynomiaal en exponentieel

Polynomiaal: $O(n^k)$ ($k \geq 0$)

Exponentieel: $O(c^n)$ ($c > 1$)

Als $\mathcal{P} \neq \mathcal{NP}$: NP-volledige problemen niet polynomiaal oplosbaar.
Wil echter niet zeggen dat ze per se exponentieel zijn.

Quasi-polynomiaal: groter dan polynomiaal, maar nog niet exponentieel. $2^{O((\log n)^c)}$ (constante $c > 0$)

Bijvoorbeeld: graafisomorfisme (Babai, 2017)

Subexponentieel: meerdere definities, maar bijv. $2^{O(\sqrt{n})}$.

Exponential Time Hypothesis

Russell Impagliazzo en Ramamohan Paturi (1999/2001):

Hypothese (Exponential Time Hypothesis (ETH))

Er bestaat een constante $c > 1$ zodat $3\text{SAT} \in \Omega(c^n)$.

M.a.w.: 3SAT heeft in de worst case een exponentiële ondergrens.

Hypothese (Strong Exponential Time Hypothesis (SETH))

Voor elke constante $c < 2$ bestaat er een k zodat $k\text{SAT} \in \Omega(c^n)$.

M.a.w.: er bestaat (asymptotisch gezien) geen beter algoritme voor algemene SAT-instanties dan exhaustive search.

$\text{SETH} \Rightarrow \text{ETH} \Rightarrow \mathcal{P} \neq \mathcal{NP}$, maar ook interessante gevolgen als $\neg\text{SETH}$ of $\neg\text{ETH}$

Beide nog niet bewezen of ontkracht.

Een recent (2016) overzicht van voortgang over $\mathcal{P}$ vs $\mathcal{NP}$:

<https://www.scottaaronson.com/papers/pnp.pdf>

(En ook o.a. Busy Beaver (2020):

<https://www.scottaaronson.com/papers/bb.pdf>)

Definitie

Een probleem is in $\mathcal{P}$ als er een deterministische Turing machine bestaat met tapealfabet $\{0, 1\}$ die voor iedere gegeven instantie van het probleem in een aantal stappen dat hoogstens polynomiaal is in de grootte van de invoer een oplossing genereert.

Definitie

Een probleem is in **PSPACE** als er een deterministische Turing machine bestaat met tapealfabet $\{0, 1\}$ die voor iedere gegeven instantie van het probleem **op een stuk tape** dat hoogstens polynomiaal is in de grootte van de invoer een oplossing genereert.

Definitie

Een probleem is in **PSPACE** als er een deterministische Turing machine bestaat met tapealfabet $\{0, 1\}$ die voor iedere gegeven instantie van het probleem **op een stuk tape** dat hoogstens polynomiaal is in de grootte van de invoer een oplossing genereert.

Definitie

Een probleem is in **NPSPACE** als er een niet-deterministische Turing machine bestaat met tapealfabet $\{0, 1\}$ die voor iedere gegeven instantie van het probleem op een stuk tape dat hoogstens polynomiaal is in de grootte van de invoer een oplossing *kan* genereren.

Definitie

Een probleem is in **PSPACE** als er een deterministische Turing machine bestaat met tapealfabet $\{0, 1\}$ die voor iedere gegeven instantie van het probleem **op een stuk tape** dat hoogstens polynomiaal is in de grootte van de invoer een oplossing genereert.

Definitie

Een probleem is in **NPSPACE** als er een niet-deterministische Turing machine bestaat met tapealfabet $\{0, 1\}$ die voor iedere gegeven instantie van het probleem op een stuk tape dat hoogstens polynomiaal is in de grootte van de invoer een oplossing *kan* genereren.

Net zo: **EXSPACE** en **NEXSPACE**.

Stelling

$\mathcal{P} \subseteq \text{PSPACE}$.

Stelling

$\mathcal{P} \subseteq \text{PSPACE}$.

Stelling

$\mathcal{NP} \subseteq \text{NPSPACE}$.

Stelling

$\mathcal{P} \subseteq \text{PSPACE}$.

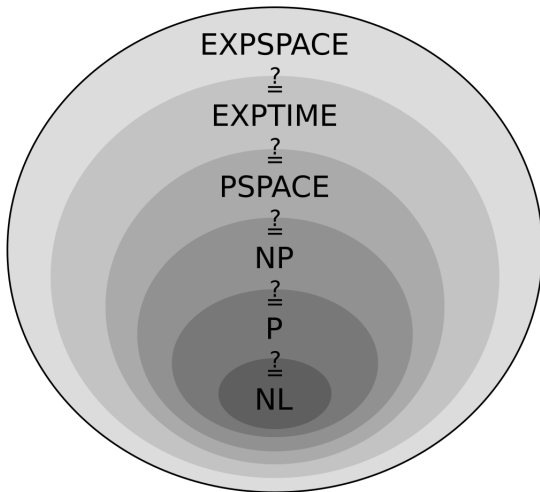
Stelling

$\mathcal{NP} \subseteq \text{NPSPACE}$.

Stelling (Savitch, 1970)

$\text{PSPACE} = \text{NPSPACE}$ en $\text{EXPSPACE} = \text{NEXPSPACE}$.

Ordering



Voor iedere klasse $\mathcal{C}$ is er een concept van een reductie $\leq_{\mathcal{C}}$ die de definiërende eigenschappen van de klasse behoudt. D.w.z. als $Q \in \mathcal{C}$ en $P \leq_{\mathcal{C}} Q$, dan ook $P \in \mathcal{C}$. Dergelijke reducties zijn ook transitief.

Definitie

Als $P \leq_{\mathcal{C}} Q$ voor alle problemen $P \in \mathcal{C}$, dan noemen we Q een $\mathcal{C}$ -moeilijk probleem. Als tevens $Q \in \mathcal{C}$, dan is Q een $\mathcal{C}$ -volledig probleem.

Gegeven een reguliere expressie r over een alfabet Σ . Geldt $L(r) = \Sigma^*$?

Gegeven een reguliere expressie r over een alfabet Σ . Geldt $L(r) = \Sigma^*$?

Gegeven een expressie in eerste-orde logica zoals

$$\exists x_1 \forall x_2 \exists x_3 : (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$

Is de expressie waar?

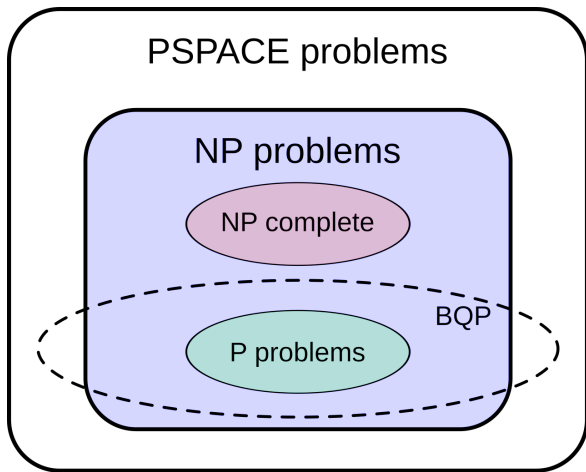
Gegeven een reguliere expressie r over een alfabet Σ . Geldt $L(r) = \Sigma^*$?

Gegeven een expressie in eerste-orde logica zoals

$$\exists x_1 \forall x_2 \exists x_3 : (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$$

Is de expressie waar?

Gegeven een rush hour puzzel (op een $n \times m$ bord, met k autootjes waarvan één rode). Kan het rode autootje worden bevrijd?



Werkcollege: zodadelijk, 13u15 – 15u00, BW.0.29, BW.0.30
Opgave 68, Opgave 4 van tentamen 2016, Opgave 4 van
hertentamen 2016

Huiswerk:

- huiswerk 3 wordt nagekeken, volgende week prijsuitreiking
- huiswerk 4 inleveren uiterlijk 19 mei

Website: <https://liacs.leidenuniv.nl/~graafjmde/COMP>

De komende twee weken

- maandag 19 mei: deadline HW4
- vrijdag 23 mei: tentamenoefening en prijsuitreiking HW3
- maandag 26 mei: (hopelijk) feedback HW4
- woensdag 28 mei: tentamen