2. Cook's theorem and the complexity variants of SAT

- **NP, NPH, NPC**
- **Lemmaer**
- **Cook's**
- **3SAT og 2SAT**

NP er en anden kompleksitetsklasse, som er formelt defineret ved:

$$\forall x \in \{0,1\}^*: x \in L \Leftrightarrow [\exists y \in \{0,1\}^*: |y| \leq p(|x|) \wedge \langle x, y\rangle \in L'] \quad , \quad L' \in \boldsymbol{P} \quad , \quad p \text{ er } poly.$$

Dvs. vi vil kunne tjekke om en given løsning er en løsning i polynomiel tid – ydermere må denne løsning maks. have længden polynomisk af probleminstansen.

Dermed kunne vi afgøre om $x \in L$ for $L \in \boldsymbol{NP}$ ved at lave en exhaustive search, hvor der for alle mulige $2^{p(|x|)+1} - 1$ værdier af y tjekkes om $\langle x, y\rangle \in L'$.

NP-hårde sprog er de sprog, der har ret lille sandsynlighed for at ligge i P. Formelt kan vi sige

$$\forall L' \in \boldsymbol{NP}: L' \leq L \Rightarrow L \in NPH$$

NP-hårde sprog ligger ikke nødvendigvis i NP, men de der gør er ret interessante, idet der er en del naturlige problemer, vi ønsker at løse, hvor vi ved, vi ikke kan finde en effektiv løsning (hvis $P \neq NP$). Problemer i både NP og NPH kaldes **NP-complete.**

Lemma 7: Hvis L_1 er NPH og $L_1 \leq L_2$ så er L_2 NPH.

Bevis: Transitivitet gælder, og idet alle problemer i NP reducerer til L_1 reducerer disse også til L_2, og def. for NPH er opfyldt.

Lemma 7 giver os en anledning til at finde et problem, som er NPC, hvor vi vha. lemma 7 kan reducere andre mulige NPC-problemer til, således vi ikke behøver fører et langt, akavet bevis, hver gang vi vil vise, at et problem er NPC. Cook's sætning gør nøjagtigt dette. Først etablerer vi nogle lemmaer til senere brug:

Lemma 8: For enhver boolsk funktion $f: \{0,1\}^n \to \{0,1\}^m$ er der et kredsløb C således at $\forall x \in \{0,1\}^n: C(x) = f(x)$.

Lemma 9: Vi er givet en TM M, som kører i tid højst $p(n) > n$ på input af længde n, hvor p er et pol.

Givet ethvert fixed input længde n er der et kredsløb C_n af størrelse (antal gates) højst $O(p(n)^2)$ således at $\forall x \in \{0,1\}^n: C_n(x) = 1 \Leftrightarrow M\ acc.\ x$.

Mapning $1^n \to C_n$ er polynomisk tid beregnelig.

Bevis: Vi kigger på positioner i: $-p(x)$ til $p(x)$ og tider $t \in \{0, \dots, p(n)\}$

Cell-state-vektor $c_{t,i}$ indeholder information om symbolet på plads i til tid t, om hovedet peger på plads i og *hvis* der peges på i noterer vi tilstanden for TM til tiden t.

Vi kan bestemme $c_{t,i} = h(c_{t-1,i-1}, c_{t-1,i}, c_{t-1,i+1})$ (ud fra tidligere tid).

Lemma 8 giver os et kredsløb D der kan beregne h.

Kredsløb E, hvor $E(b)$ er en cell-state-vektor, hvor cellen indeholder 0 eller 1, og hovedet ikke peger på.

Kredsløb F svarer $F(y) = 1$, hvis y er en cell-state-vektor, hvor hovedet er på, og TM er i en accept-tilstand.

Cook's sætning giver os et problem SAT, som er NP-hårdt (og ligger i NP). Dermed ligger det i NPC. Vi viser at CircuitSAT er NP-hårdt og reducerer til SAT, hvormed Cook's sætning er vist. CircuitSAT generaliserer SAT, idet CNF'er er formularer, som er kredsløb.

CircuitSAT: Givet et boolsk kredsløb C, er der så en vektor x, således at $C(x) = 1$?

Theorem 11: $CIRCUIT\ SAT \in \boldsymbol{NPC}$

Bevis: CSAT er tydeligvis i NP, så vi skal vise, at det er NP-hårdt. Givet et tilfældigt sprog $L \in NP$, skal vi vise at $\forall x \in L \Leftrightarrow r(x) \in CSAT$.

$$\forall x \in \{0,1\}^*: x \in L \Leftrightarrow [\exists y \in \{0,1\}^*: |y| \leq p(|x|) \wedge \langle x, y\rangle \in L'] \quad , \quad L' \in \boldsymbol{P} \quad , \quad p\ er\ poly.$$

Instanser af L skal mappes til instanser af CSAT, dvs. beskrivelser af kredsløb. Givet input x, så er $r(x)$ en beskrivelse af kredsløbet $C \equiv D_0 \vee D_1 \vee \ldots \vee D_{p(|x|)}$.

Vi lader M være en TM, der afgør L' i polynomisk tid. Lemma 9 giver os et kredsløb C_n, som for alle z, y hvor $|\langle z, y\rangle| = n$ giver $C_n(\langle z, y\rangle) = 1 \Leftrightarrow M\ acc. \langle z, y\rangle$. Vi lader $n = 2(|x| + |y| + 1)$.

På C_n hardwirer vi input x ved at udskifte ulige inputgates med x_i, lige med 0, og de to sidste med 1:

$$x_1 0 x_2 0 \ldots 0 x_n 011$$

Dette kalder vi D_i, som kan afgøre om et input y af længde i er en løsning til x: Hver D_i tager i boolske input og evaluerer til 1 på input $y \in \{0,1\}^i \Leftrightarrow \langle x, y\rangle \in L'$.

Dvs. reduktionen skal konstruere $p(|x|)$ underkredsløb D_i, som kombineres med ORs og outputter C. Vha. Polynomial Church-Turing sætning kan vi se reduktionen er polynomisk tids beregnelig.

Nu har vi vist, at CSAT er NP-hårdt, men Cook's sætning siger, at SAT er NP-hårdt.

Theorem 12: $CSAT \leq SAT$

Givet et 1-output kredsløb C, definerer vi $f = r(C)$, således at f har en tilfredsstillende sandhedstilordning $\Leftrightarrow C$ har. r oversætter i polynomiel tid: $x \in CSAT \Leftrightarrow r(x) \in SAT$. Vi oversætter gates til klausuler:

AND: $(g \Leftrightarrow h_1 \wedge h_2)$

$$\equiv \big(g \Rightarrow (h_1 \wedge h_2)\big) \wedge \big((h_1 \wedge h_2) \Rightarrow g\big)$$
$$\equiv \big(\neg g \vee (h_1 \wedge h_2)\big) \wedge (\neg(h_1 \wedge h_2) \vee g)$$
$$\equiv (\neg g \vee h_1) \wedge (\neg g \vee h_2) \wedge (\neg h_1 \vee \neg h_2 \vee g)$$

OR: $(g \Leftrightarrow h_1 \vee h_2) \equiv (\neg g \vee h_1 \vee h_2) \wedge (g \vee \neg h_1) \wedge (g \vee \neg h_2)$

NOT: $(g \Leftrightarrow \neg h) \equiv (\neg g \vee \neg h) \wedge (g \vee h)$ – klausulerne opfyldes kun, hvis g og h er forskellige.

COPY: $(g \Leftrightarrow h) \equiv (\neg g \vee h) \wedge (g \vee \neg h)$ – klausulerne opfyldes kun, hvis g og h er ens.

CONST-0: $0 \rightarrow (\neg g)$ CONST-1: $1 \rightarrow (g)$ OUTPUT: $g \rightarrow (g)$ er outputtet af C.

Ovenstående klausuler ANDes sammen til f. Output-klausulen sikrer, at SAT-formlen kun går op, når vores CSAT går op.

3SAT er en variant af SAT, hvor vi siger, at der skal være præcis 3 literaler i hver klausul.

$CSAT \leq SAT \leq 3SAT \in NPC$:

Oversættelsen fra CircuitSAT til SAT giver os allerede stort set det, vi vil have, idet ingen af kredsløbene fra CSAT bliver oversat til klausuler med mere end 3 literaler. De klausuer, hvor der er mindre end 3 literaler kopierer vi simpelthen bare en af de andre literaler i klausulen et passende antal gange, således at der nu er 3 literaler i hver klausul.

2SAT er en variant af SAT, hvor vi siger, at der skal være præcis 2 literaler i hver klausul.

$2SAT \in P$:

Givet en 2CNF-formel Φ, definer grafen $G(\Phi)$ ved at *knuder* er variabler og deres negeringer. For hver klausul $(\alpha \vee \beta)$ tilføjes der *kanter*: $(\neg\alpha, \beta)$ og $(\neg\beta, \alpha)$.

Lemma: Φ kan opfyldes $\Leftrightarrow \forall x$ er der ikke både en sti $x \rightarrow \neg x$ og $\neg x \rightarrow x$ (ingen cykler).

Bevis: ⇒ modbevis: Antag, der for variabel x findes en sådan cykel, og antag vi har en vilk. sandheds. T.

Antag $T(x) = t$rue. Pga. stien og $T(x) = true$ mens $T(\neg x) = false$, så må der et sted på stien være en kant (α, β), hvor $T(\alpha) = true, T(\beta) = false$, da denne kant er en del af $G(\Phi)$ findes klausulen $(\neg\alpha \vee \beta)$ i Φ. Denne klausul tilfredsstilles ikke.

⇐: Konstruer T ud fra grafen, således ingen kant går fra true til false. Kig ikke på stier $x \to \neg x$.

Sæt $T(x) = 1$, sæt $T(\alpha) = 1$ hvis $x \to \alpha$. Negeringer af disse sættes til false (der er stier $\neg\alpha \to \neg x$).

Veldefineret: Hvis der var stier fra α til både β og $\neg\beta$, var der også stier fra disse til $\neg\alpha$, og dermed tilbage igen til α, hvilket er en modsigelse.

Dette gentages indtil alle knuder har en sandhedsværdi. Der kan ikke gå en kant fra true til false, da alle efterfølgere af en true-knude sættes til true, og alle efterfølgere til en false-knude sættes til false.

Denne algoritme er polynomiel, da det ikke tager mere end polynomiel tid at konstruere grafen, og vi efterfølgende kan finde ud af, om der er en sandhedstilordning ved at tjekke efter cykler.

NAESAT: Givet en 3CNF-formel, eksisterer der en sandhedstilordning T, således at alle literaler i klausulerne ikke er ens?

MAX-2-SAT: Givet en 2CNF-formel og et tal k, eksisterer der en sandhedstilordning, som tilfredsstiller mindst k klausuler i formlen?