

Notation

$\mathcal{B} = \{0, 1\} = \{\text{false}, \text{true}\} = \{f, t\} = \{\top, \perp\}$
zweielementige boolesche Algebra mit Operationen

- ▶ Konjunktion ("und") \wedge
- ▶ Disjunktion ("oder") \vee
- ▶ Negation ("nicht") \neg
- ▶ Exklusiv-Oder \oplus
- ▶ ...

$\mathcal{B}^n = \{0, 1\}^n$: boolesche Algebra der Bitvektoren der Länge n
 $\mathbf{b} = (b_1, b_2, \dots, b_n)$ mit
Operationen $\wedge, \vee, \neg, \oplus, \dots$ komponentenweise

(alias Teilmengen von $\{1, 2, \dots, n\}$ mit $\cap, \cup, c, \Delta, \dots$).

Variable, Literale, Klauseln

$X_n = \{x_1, x_2, \dots, x_n\}$: Menge von (booleschen) Variablen

$\bar{X}_n = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$: negierte Variable

$U_n = X_n \cup \bar{X}_n$: Literale

Negation von negierten Literalen: $\bar{\bar{x}} = x$

Klausel (über X_n): Menge $C = \{u_1, u_2, \dots, u_s\} \subseteq U_n$,
wobei komplementäre Paare x_i, \bar{x}_i nicht auftreten sollen

Klauseln verstehen sich als *Disjunktion ihrer Literale*

$$C = u_1 \vee u_2 \vee \dots \vee u_s$$

Für $s = 0$ wird die leere Klausel $C = \emptyset$ mit false identifiziert (= neutrales Element bezgl. Disjunktion), oft auch als \square notiert.

$\mathcal{C}(X_n)$ Menge der Klauseln über X_n

Hamming-Abstand

$\|\mathbf{b}\|$: Anzahl der Komponenten $b_i = 1$ in \mathbf{b} (HAMMING-Gewicht)

$d(\mathbf{b}, \mathbf{c}) = \|\mathbf{b} \oplus \mathbf{c}\|$ HAMMING-Distanz von Bitvektoren

$\mathcal{B}_k^n = \{\mathbf{b} \in \mathcal{B}^n; \|\mathbf{b}\| = k\}$ ($0 \leq k \leq n$)
 $\#\mathcal{B}_k^n = \binom{n}{k}$

$\mathcal{B}_{\leq k}^n = \{\mathbf{b} \in \mathcal{B}^n; \|\mathbf{b}\| \leq k\}$ ($0 \leq k \leq n$)
HAMMING-Kugel vom Radius k um den Nullvektor

n -stellige boolesche Funktionen : $f : \mathcal{B}^n \rightarrow \mathcal{B}$

Formeln (KNF)

Formel (in konjunktiver Normalform, über X_n) :

Menge $F = \{C_1, C_2, \dots, C_t\}$ von verschiedenen Klauseln über X_n ,
zu verstehen als *Konjunktion ihrer Klauseln*

$$F = C_1 \wedge C_2 \wedge \dots \wedge C_t$$

Für $t = 0$ wird die leere Formel $F = \emptyset$ mit true identifiziert
(= neutrales Element bezgl. Konjunktion)

$\mathcal{F}(X_n)$: Menge der Formeln über X_n

Bewertungen

Jedes Element $\mathbf{b} = (b_1, b_2, \dots, b_n) \in \mathcal{B}^n$ definiert eine Bewertung der Variablen

$$\phi_{\mathbf{b}} : X_n \rightarrow \mathcal{B} : x_i \mapsto b_i \quad (1 \leq i \leq n)$$

Fortsetzung zu einer Bewertung der Literale:

$$\phi_{\mathbf{b}} : U_n \rightarrow \mathcal{B} \quad \text{mittels} \quad \phi_{\mathbf{b}}(\bar{x}) = \overline{\phi_{\mathbf{b}}(x)} \quad (x \in X_n)$$

Fortsetzung zu einer Bewertung der Klauseln:

$$\phi_{\mathbf{b}} : \mathcal{C}(X_n) \rightarrow \mathcal{B} : \phi_{\mathbf{b}}(u_1 \vee u_2 \vee \dots \vee u_s) = \phi_{\mathbf{b}}(u_1) \vee \phi_{\mathbf{b}}(u_2) \vee \dots \vee \phi_{\mathbf{b}}(u_s)$$

Fortsetzung zu einer Bewertung der Formeln:

$$\phi_{\mathbf{b}} : \mathcal{F}(X_n) \rightarrow \mathcal{B} : \phi_{\mathbf{b}}(C_1 \wedge C_2 \wedge \dots \wedge C_t) = \phi_{\mathbf{b}}(C_1) \wedge \phi_{\mathbf{b}}(C_2) \wedge \dots \wedge \phi_{\mathbf{b}}(C_t)$$

Das Problem SAT (Satisfiability)

- ▶ Instanzen: boolesche Formeln $F \in \mathcal{F}(X_n)$ ($n =$ Instanzengröße)
- ▶ Entscheide, ob F erfüllbar ist oder nicht, d.h. ob es ein $\mathbf{b} \in \mathcal{B}^n$ gibt mit $\phi_{\mathbf{b}}(F) = \langle F \rangle(\mathbf{b}) = \text{true}$ oder nicht.

k -SAT: wie SAT, aber mit höchstens k Literalen pro Klausel

- ▶ Für SAT und k -SAT für $k \geq 3$ sind keine effizienten Algorithmen bekannt!
2-SAT ist effizient entscheidbar.
- ▶ SAT ist das zentrale Problem unter den "NP-vollständigen Problemen"

Funktionale Vollständigkeit

- ▶ Jede boolesche Formel $F \in \mathcal{F}(X_n)$ definiert eine n -stellige boolesche Funktion

$$\langle F \rangle : \mathcal{B}^n \rightarrow \mathcal{B} : \mathbf{b} \mapsto \phi_{\mathbf{b}}(F)$$

- ▶ Zu jeder n -stelligen booleschen Funktion $f : \mathcal{B}^n \rightarrow \mathcal{B}$ gibt es eine Formel $F \in \mathcal{F}(X_n)$ mit $\langle F \rangle = f$.

Partielle Bewertungen

P.B. von X_n sind Funktionen $\psi : U_n \rightarrow \mathcal{B} \cup U_n$ mit

$$\psi(u) \in \{u, 0, 1\} \quad \text{und} \quad \psi(\bar{u}) = \overline{\psi(u)} \quad (u \in U_n)$$

Fortsetzung zu einer partiellen Bewertung der Klauseln:

$$\begin{aligned} \psi : \mathcal{C}(X_n) &\rightarrow \mathcal{C}X_n \cup \{t\} : \\ \psi(u_1 \vee u_2 \vee \dots \vee u_s) &= \psi(u_1) \vee \psi(u_2) \vee \dots \vee \psi(u_s) \end{aligned}$$

Fortsetzung zu einer partiellen Bewertung der Formeln:

$$\begin{aligned} \psi : \mathcal{F}(X_n) &\rightarrow \mathcal{F}(X_n) : \\ \psi(C_1 \wedge C_2 \wedge \dots \wedge C_t) &= \psi(C_1) \wedge \psi(C_2) \wedge \dots \wedge \psi(C_t) \end{aligned}$$

Wachstum von $T(n)$

Definiere rekursive Folge $(t_k(n))_{n \geq 0}$ mittels

$$t_k(n) = t_k(n-1) + t_k(n-2) + \dots + t_k(n-k) \quad (n \geq k)$$

und mit Anfangswerten $t_k(0), t_k(1), \dots, t_k(k-1)$, für die $T(j) \leq t_k(j)$ ($0 \leq j < k$) ist.

Dann gilt

$$T(n) \leq t_k(n) \quad \text{für alle } n \geq 0.$$

Resultat

Berechnungsaufwand $T(n)$ der *divide-and-conquer*-Methode in Anhängigkeit von der Anzahl der Variablen n , angewendet auf k -SAT-Formeln, verhält sich wie

$$T(n) \leq t_k(n) \sim \alpha_k^n \quad (n \rightarrow \infty).$$

Genauere Betrachtung liefert für k -SAT-Formeln sogar

$$T(n) \leq t_{k-1}(n) \sim \alpha_{k-1}^n \quad (n \rightarrow \infty)$$

Für das NP-vollständige Problem 3-SAT hat man also die Schranke mit dem Goldenen Schnitt ϕ :

$$T(n) \leq t_2(n) \sim \phi^n = (1.6181\dots)^n \quad (n \rightarrow \infty)$$

Analytisches

Wir werden sehen: es gilt

$$t_k(n) \sim \alpha_k^n \quad \text{für } n \rightarrow \infty,$$

wobei α_k die (eindeutige) im Intervall $0 < x < 2$ liegende reelle Nullstelle des Polynoms

$$X^k - X^{k-1} - X^{k-2} - \dots - X - 1$$

ist. Beispielsweise gilt

$$k = 1 : \quad \alpha_1 = 1$$

$$k = 2 : \quad \alpha_2 = \phi = \frac{1 + \sqrt{5}}{2} = 1.6181\dots$$

$$k = 3 : \quad \alpha_3 = 1.8393\dots$$

$$k = 4 : \quad \alpha_4 = 1.9276\dots$$

$$k = 5 : \quad \alpha_5 = 1.966\dots$$

relativer Aufwand

Auch die *divide-and-conquer*-Methode hat exponentielles Verhalten in der Variablenanzahl n , aber der relative Aufwand (gemessen an der *brute-force*-Methode) verringert sich exponentiell!

Einige Werte für $k = 3$:

| n | $\left(\frac{\phi}{2}\right)^n$ |
|-----|---------------------------------|
| 5 | 0.3465678100... |
| 10 | 0.1201092469... |
| 20 | 0.0144262311... |
| 50 | 0.0000249966... |
| 100 | $0.6248\dots \cdot 10^{-9}$ |

Literaturhinweis: SCHÖNING, Algorithmik, MONIEN-SPECKENMEYER-Algorithmus.