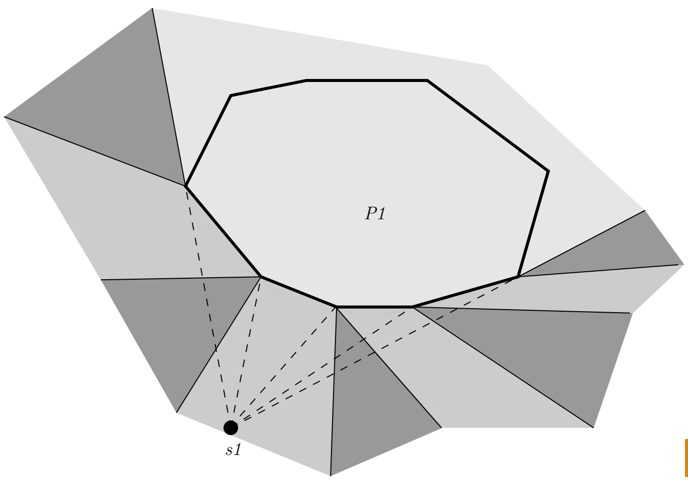
Offline Bewegungsplanung: TPP und Wege in 3D

Elmar Langetepe
University of Bonn

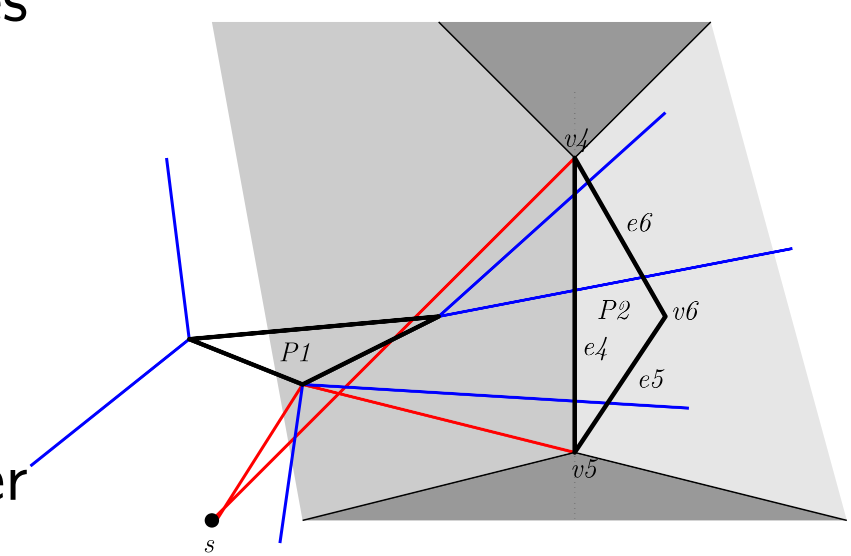
Berechnung $S_1, \ldots, S_k$: Alg. 1.11

- Queries: Backward
- Berechnung: Forward
- SPM für P_1
- Sichtbare konvexe Kette/Baum der Strahlen: $O(n_1)$
- Disjunkte Reflektionen!!



Berechnung $S_1, \ldots, S_k$: Alg. 1.11

- SPM von P_i aus SPM of $P_{i-1}, \ldots, P_1$
- Letztes Segment des Kürzesten Weges von s zu Knoten von P_i
- Query: Nutze SPM $P_{i-1}, \ldots, P_1$
- Laufzeit: $O\left(n_i(i-1)\log\frac{N_{i-1}}{i-1}\right)$ mit $N_j := \sum_{l=1}^{j} n_l$
- Sichtbare konvexe Kette/Baum der Strahlen
- Disjunkt wegen konvexer Kette!!



Analyse Berechnung $S_1, \ldots, S_k$: Theorem 1.34

Rekursiv: $P_2, \ldots, P_k$

Gesamtlaufzeit:

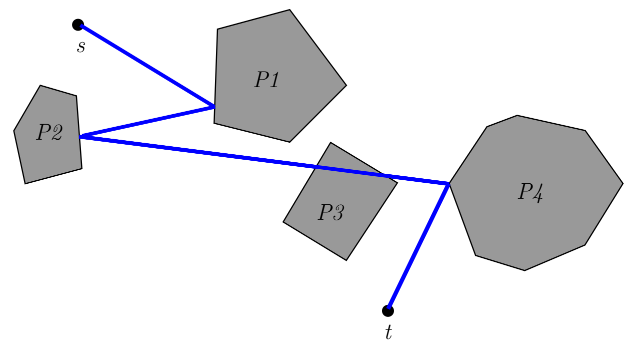
$$\sum_{i=2}^{k} n_i (i-1) \log \frac{N_{i-1}}{i-1}$$

$$n_i (i-1) \log \frac{N_{i-1}}{i-1} \quad \leq \quad n_i k \log \frac{n}{k}$$

Gesamtlaufzeit:

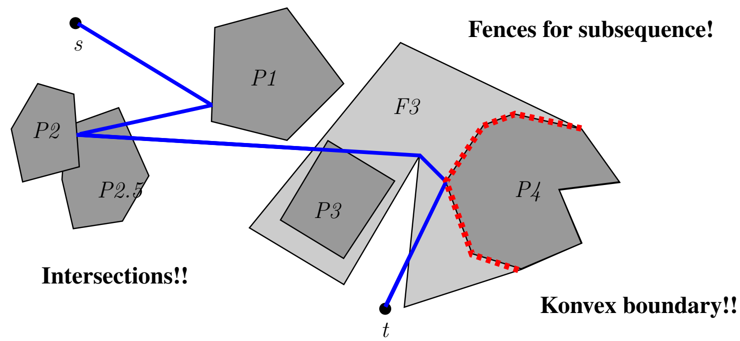
$$O\left(kn \log \frac{n}{k}\right)!$$

Zusammenfassung!!



- Einfache Version:
- Disjunkte, konvexe Polygone, keine Zäune
- $O(nk \log \frac{n}{k})$
- Build(Query): $O(nk \log \frac{n}{k})$
- Komplexität: $O(n)$
- Query (festes s): $O(k \log \frac{n}{k})$
- Theorem 1.34

Erweiterung!

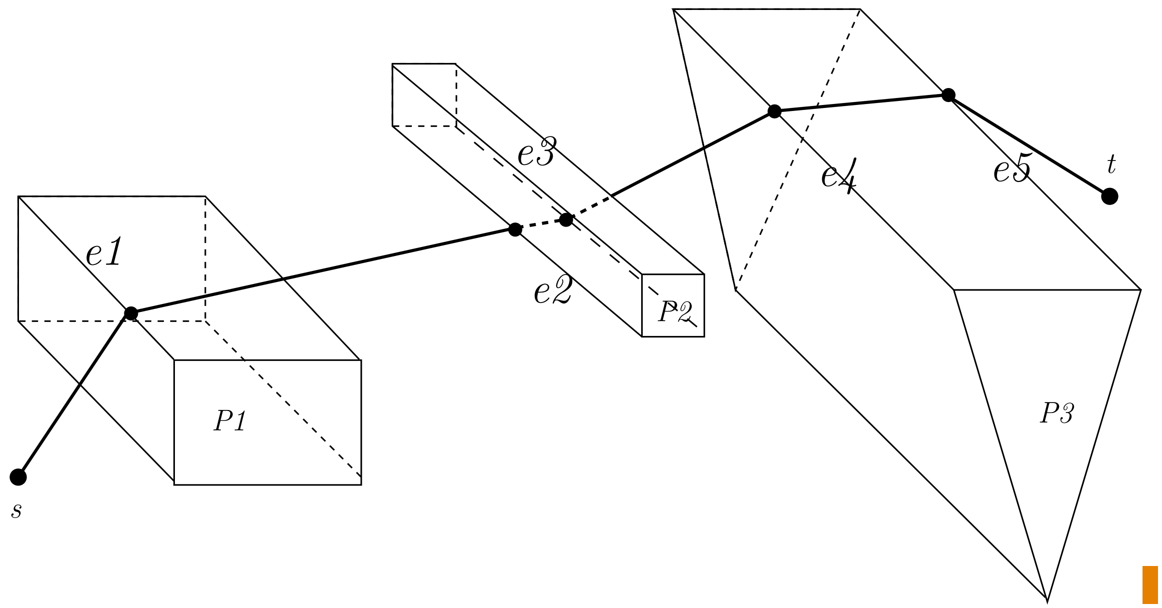


- Komplexe Version:
- Nicht-disjunkte, konvexe Polygone, Zäune
- $O(nk^2 \log n)$ insgesamt
- Build(Query): $O(nk^2 \log n)$
- Komplexität: $O(kn)$
- Query (festes s): $O(kn)$

- Lemma 1.35: Reflexionsbereich ist Baum
- Theorem 1.36/Theorem 1.37

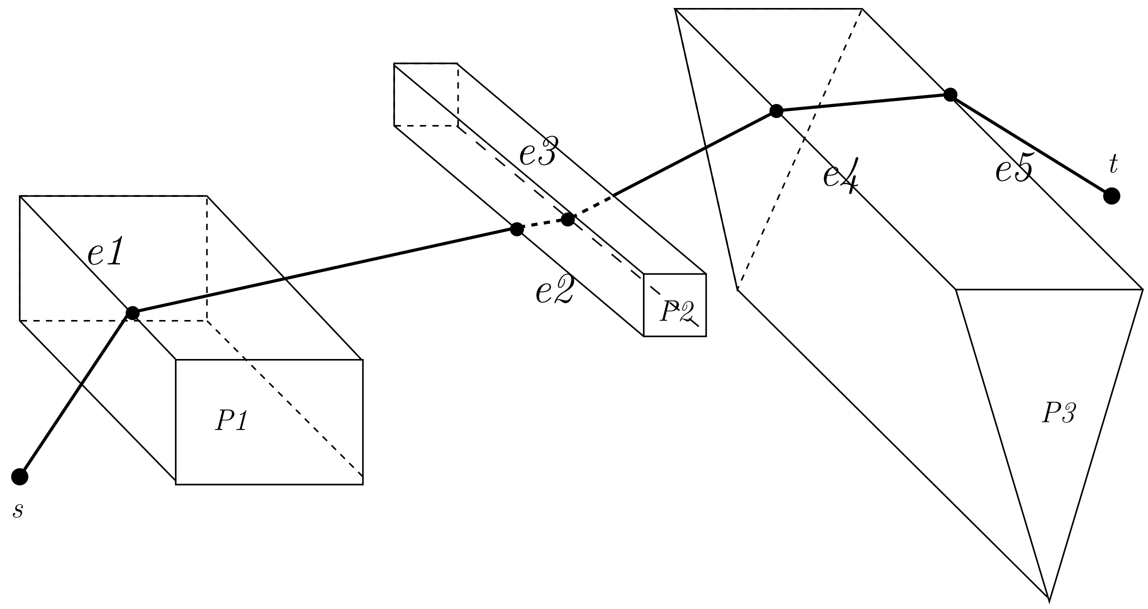
Polyeder-Szene in 3D

- Startpunkt s, Zielpunkt t
- Menge von Polyedern
- Kürzester Weg von s nach t: NP-hard



Polyeder-Szene in 3D

- Teilprobleme
- 1) Kantenreihenfolge
- 2) Verschiebung auf der Kante
- Bereits 1) ist NP hard



Kantenreihenfolge: NP hard

- Entscheidungsproblem, E-Problem mit Schranke, Optimierungsproblem
- $S \subset \Omega$: $x \in \Omega$: gilt $x \in S$?
- Problem S ist NP-vollständig

1. Liegt in NP
2. Jedes andere Problem $S' \in NP$ läßt sich auf S in polynomieller Zeit auf S *reduzieren*

- NP hard: Nur zweiter Teil (Optimierungsprobleme)
- 3-SAT ist NP vollständig (Cook)

Kantenreihenfolge: NP hard

- Reduktion $S' \subset \Omega'$ auf $S \subset \Omega$
- Funktion $f : \Omega' \to \Omega$

1. $\forall x' \in \Omega'$: $f(x')$ in polynomieller Zeit $(|x'|)$
2. $\forall x' \in \Omega'$: $f(x') \in S \Leftrightarrow x' \in S'$

- 3-SAT NP vollständig
- 3-SAT reduzieren auf Kantenreihenfolge

3-SAT reduzieren auf Kantenreihenfolge

$$\alpha = \bigwedge_{i=1}^{m} (L_{i_1} \vee L_{i_2} \vee L_{i_3}) \text{ mit } L_{i_j} \in \{X_k, \neg X_k\}$$

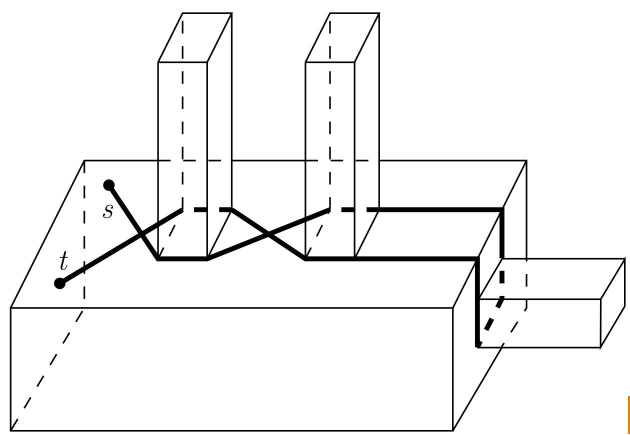
m Klauseln mit n Variablen: Erfüllbarkeit?

Konstruiere Parcours P_α, so dass:

- Kürzester Weg (Kantenfolge) von s nach t erzeugt Belegung w
- w erfüllt $\alpha \Rightarrow$ fertig!
- w erfüllt α nicht $\Rightarrow$ kein w erfüllt α

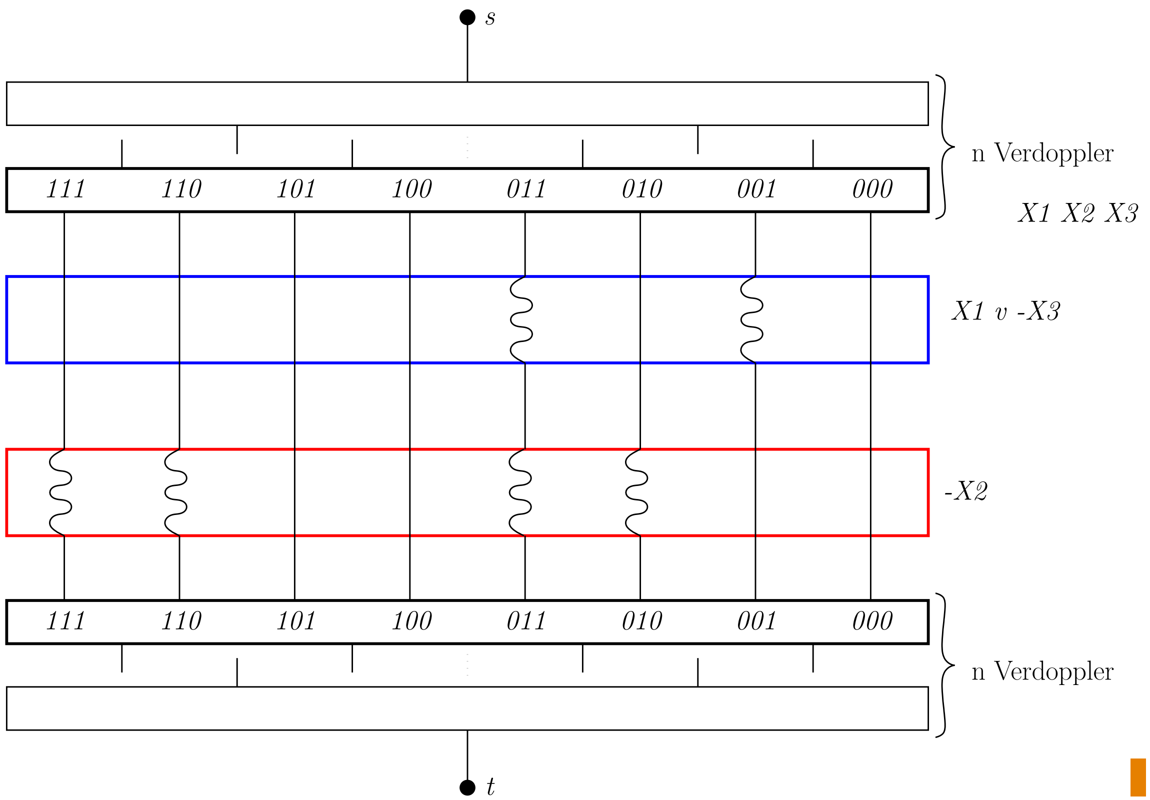
Parcours in $O(p(mn))$ erzeugen

- 2^n Belegungen der n Variablen
- 2^n *geodätisch* Kürzeste Wege
- Eine davon wird die Kürzeste sein
- Ergibt Variablen-Belegung nach Kantenreihenfolge



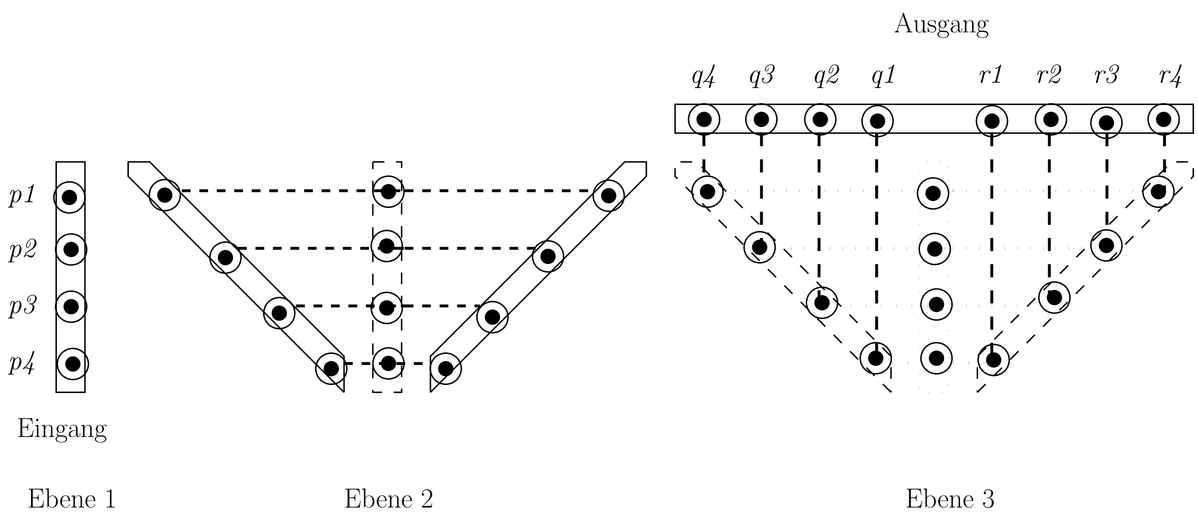
Parcours erzeugen: Prinzip

Beispiel: $(X1 \vee \neg X3) \wedge (\neg X2)$



Komponenten: Verdoppler!

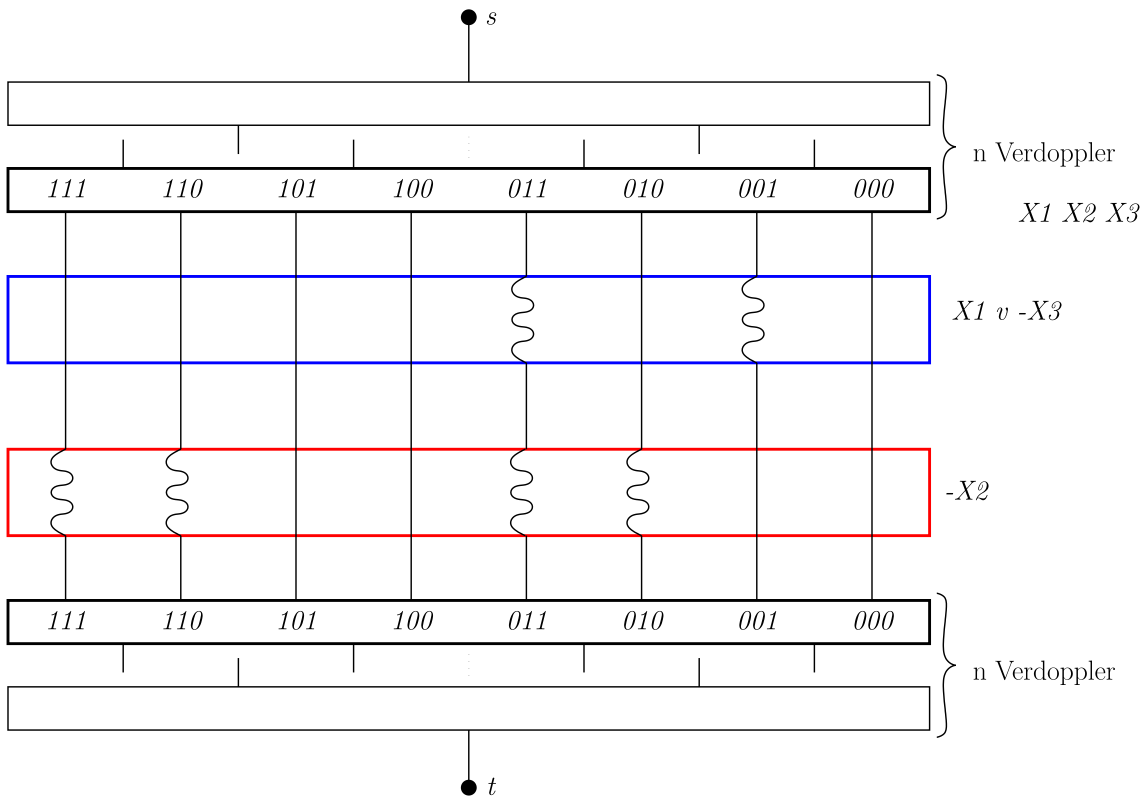
Dünne Platten mit Schlitzen eng hintereinander!



Sukzessive 2^n ungefähr gleichlange Wege erzeugen!

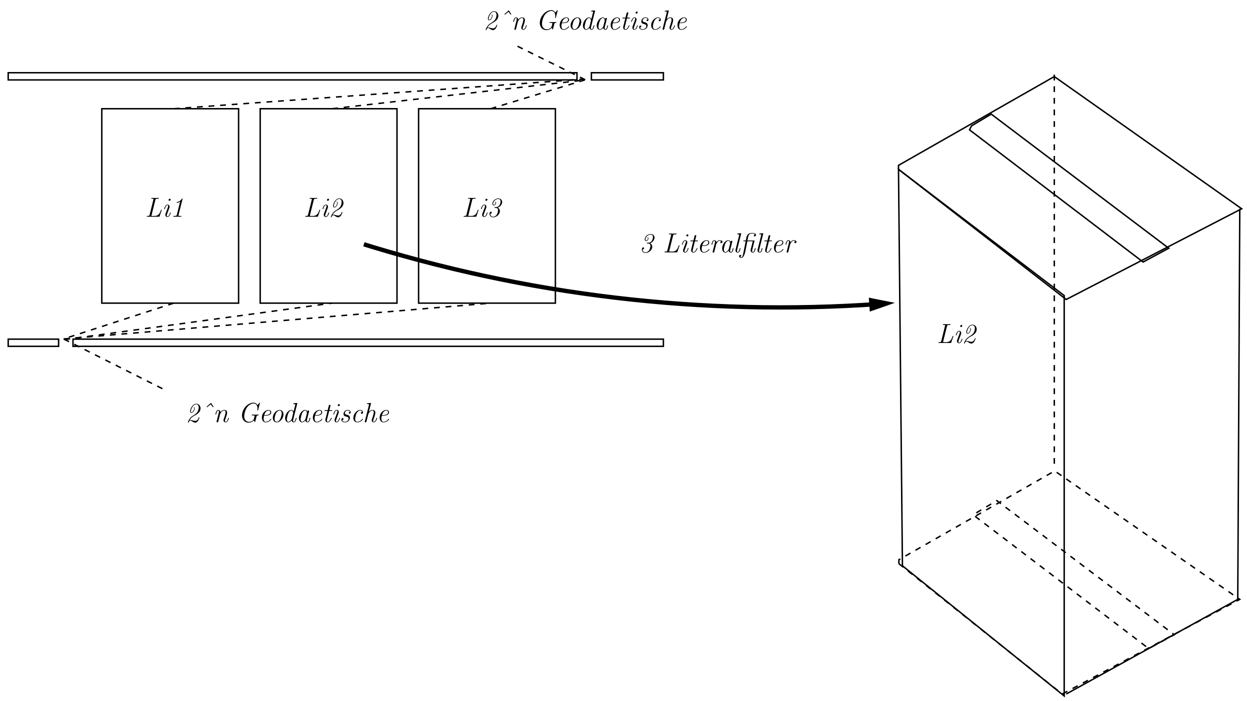
Kantenreihenfolge ist gleich!

Gesamtprinzip: Nacheinander Klauseln!



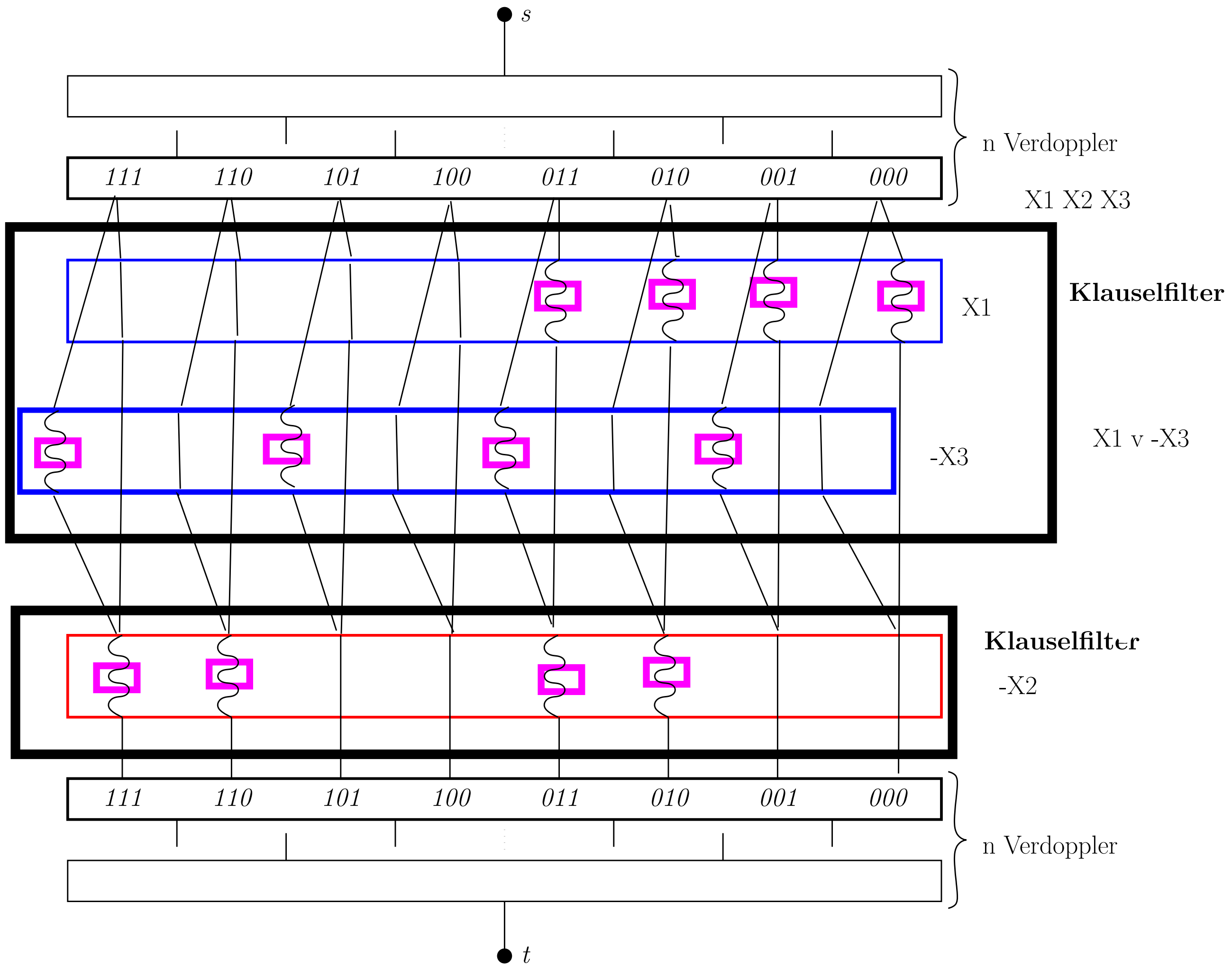
Komponenten: Klauselfilter

Sukzessive durch die Klauseln schicken! Auf Literale aufteilen! Dünne Platten!



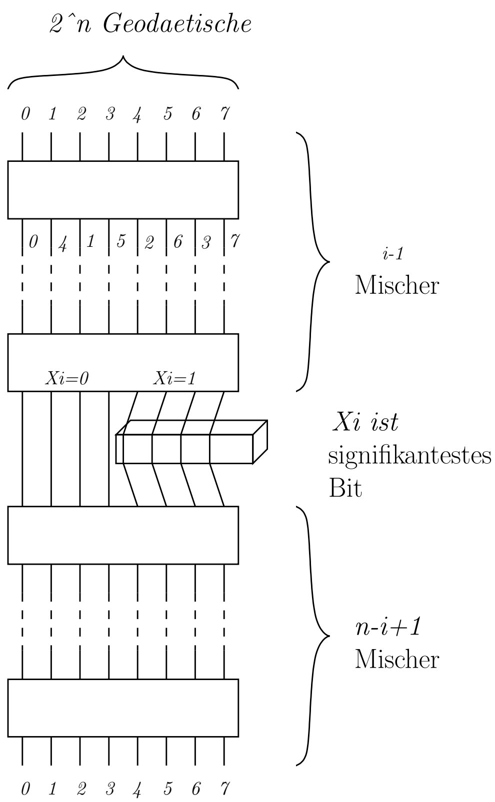
Gleich lang, bis auf das, was in den Literalfiltern passiert!

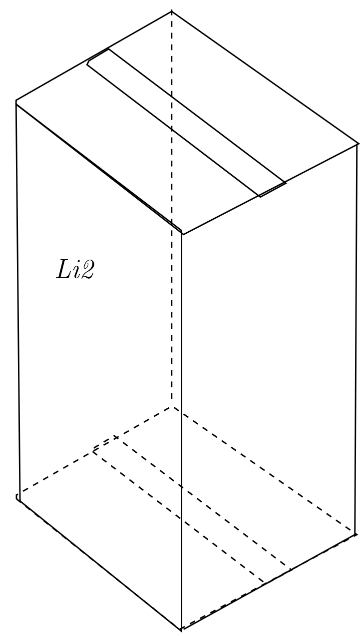
Gesamtprinzip: Einzelne Literale



Komponenten: Literalfilter

Wege für signifikantes Bit sammeln!

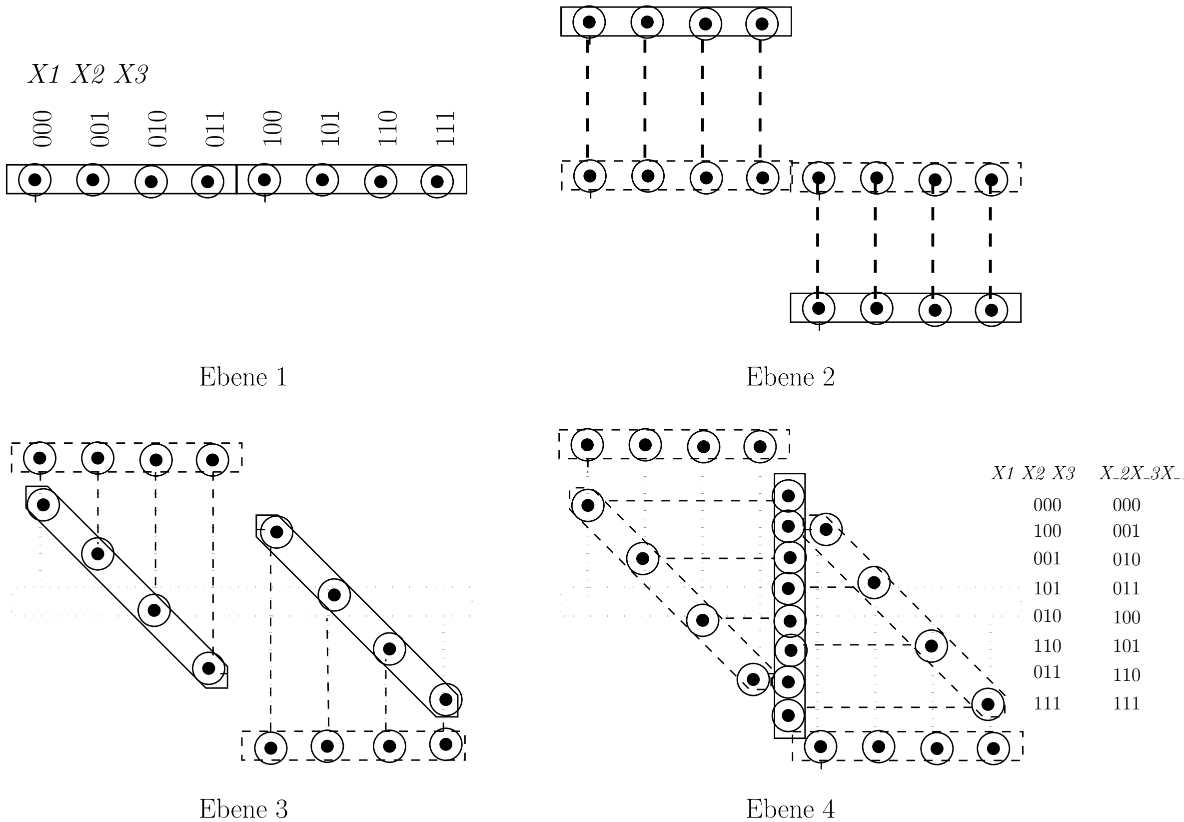




Falls X_i dann $X_i = 0$ verlängern! Falls $\neg X_i$ dann $X_i = 1$ verlängern!

n Mischer pro Literalfilter!!

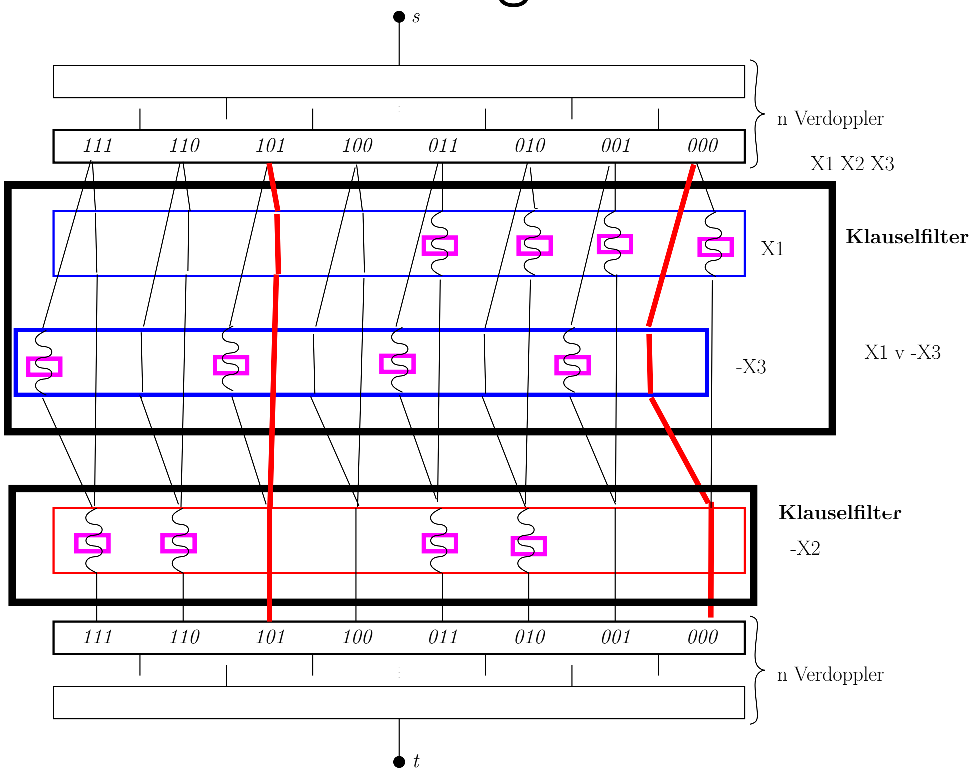
Ein Mischer erzeugt Bitverschiebung der Wege um 1! Alle bleiben gleich lang!!



Kürzeste Wege Alg. für P_α

Ein Weg, der nicht verlängert wird entspricht genau einer Belegung, die die Formel erfüllt!!!

Das kann man der Kantenreihenfolge entnehmen!!



Konstruktion insgesamt!!

- $2n$ Verdoppler: $O(n)$ Kanten
- m Klauselfilter: je Klauselfilter
  - 3 Literalfilter
  - n Mischer je Filter
- Insgesamt $O(mn)$ Kanten
- In polynomieller Zeit konstruierbar

Ergebnis!!!

Theorem 1.38 (Canny/Reif): Bestimmung der optimalen Kantenfolge bei der Berechnung Kürzester Wege in polyedrischer Szene in 3D ist NP hart.

Beweis!!