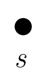
Offline Bewegungsplanung: Polyeder

Elmar Langetepe
University of Bonn

Polyeder-Szene in 3D

Polyeder-Szene in 3D

- Startpunkt s,



Polyeder-Szene in 3D

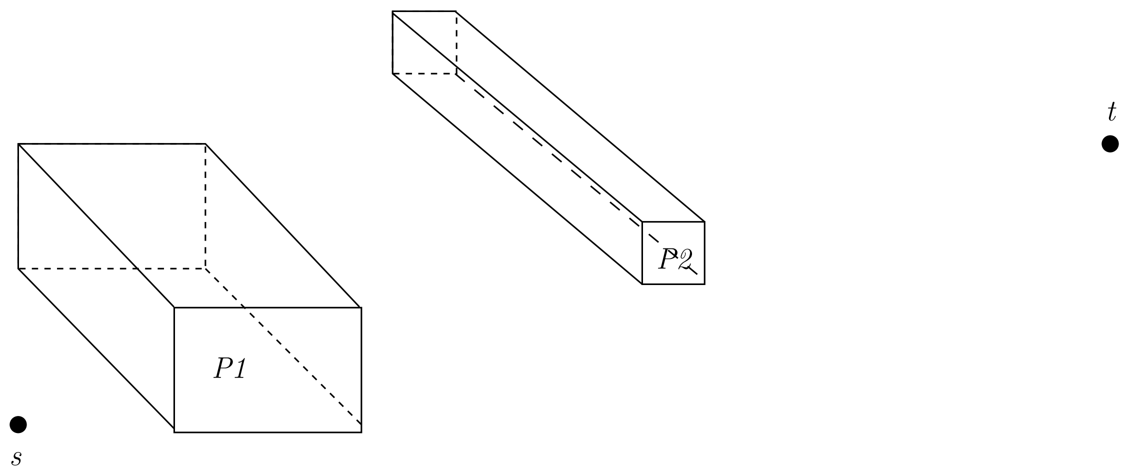
- Startpunkt s, Zielpunkt t

Polyeder-Szene in 3D

- Startpunkt s, Zielpunkt t
- Menge von Polyedern

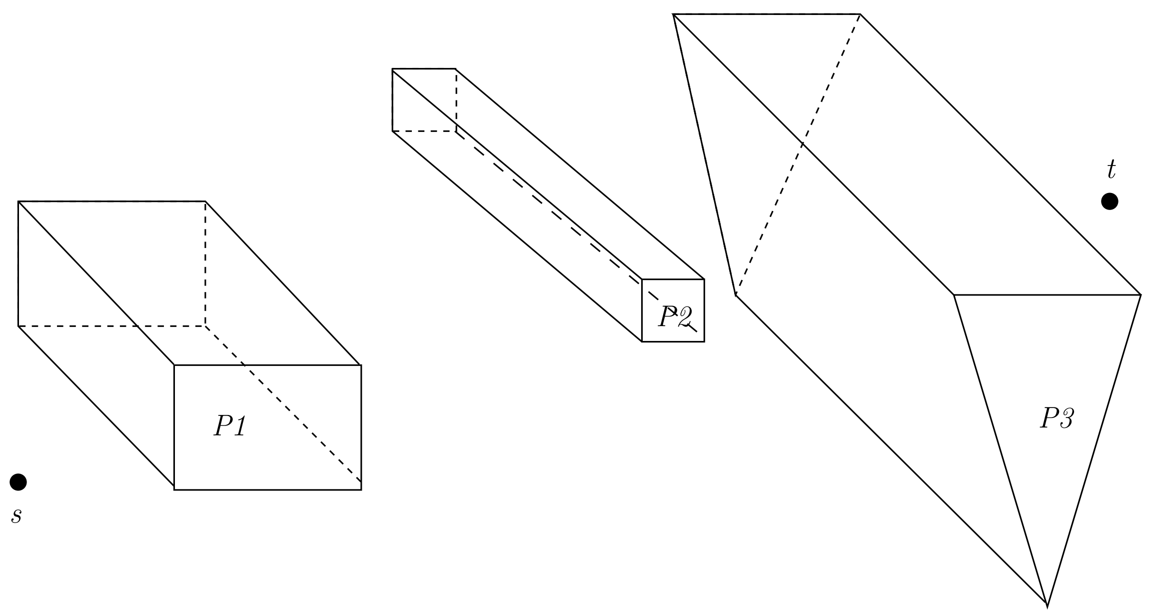
Polyeder-Szene in 3D

- Startpunkt s, Zielpunkt t
- Menge von Polyedern



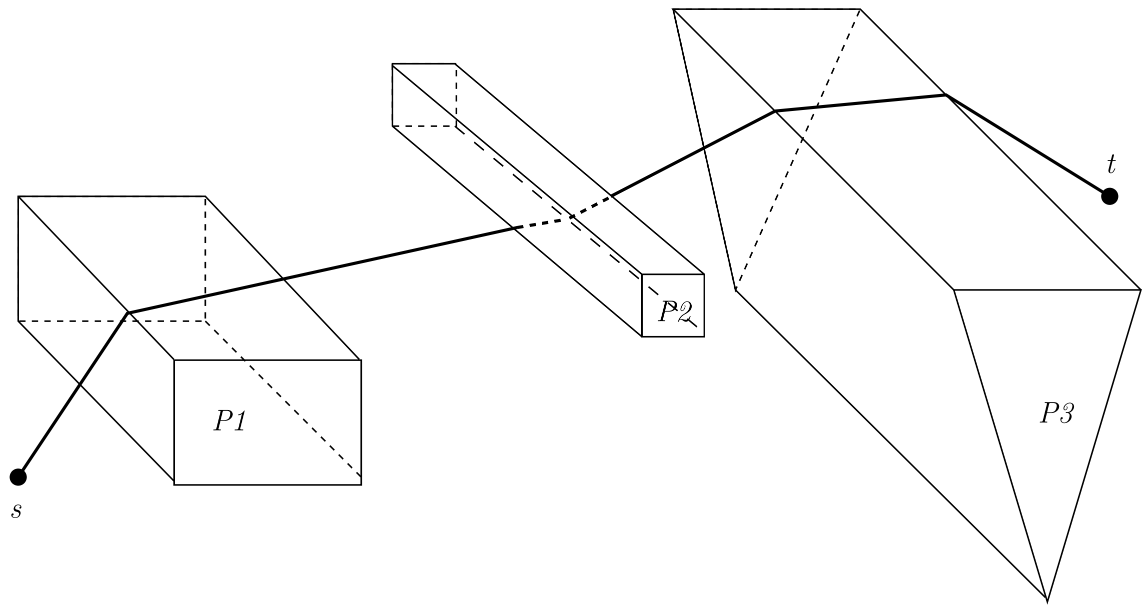
Polyeder-Szene in 3D

- Startpunkt s, Zielpunkt t
- Menge von Polyedern



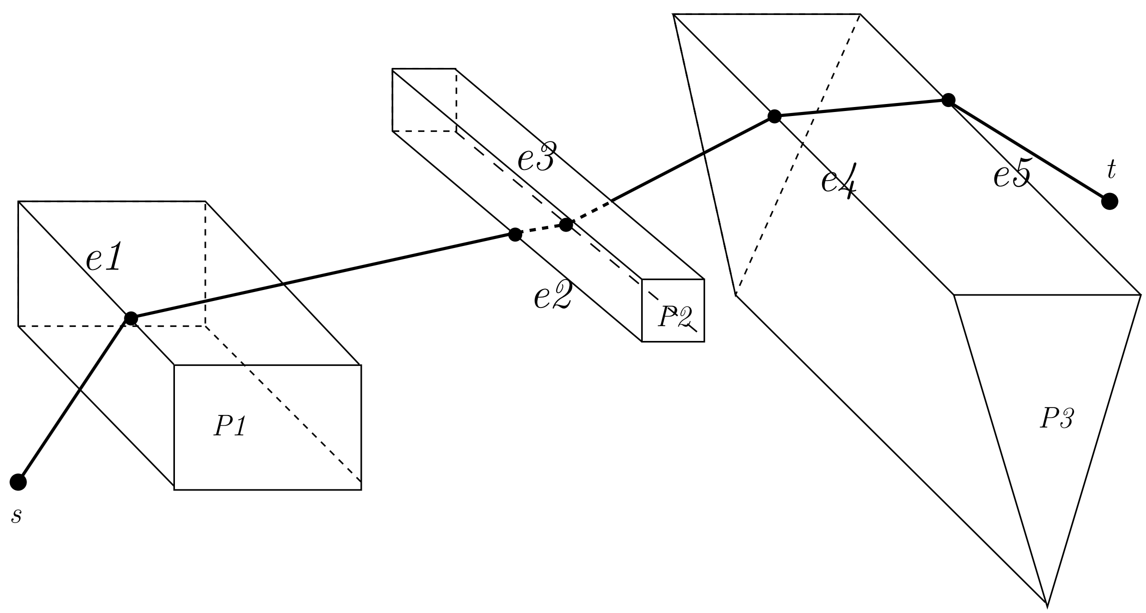
Polyeder-Szene in 3D

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



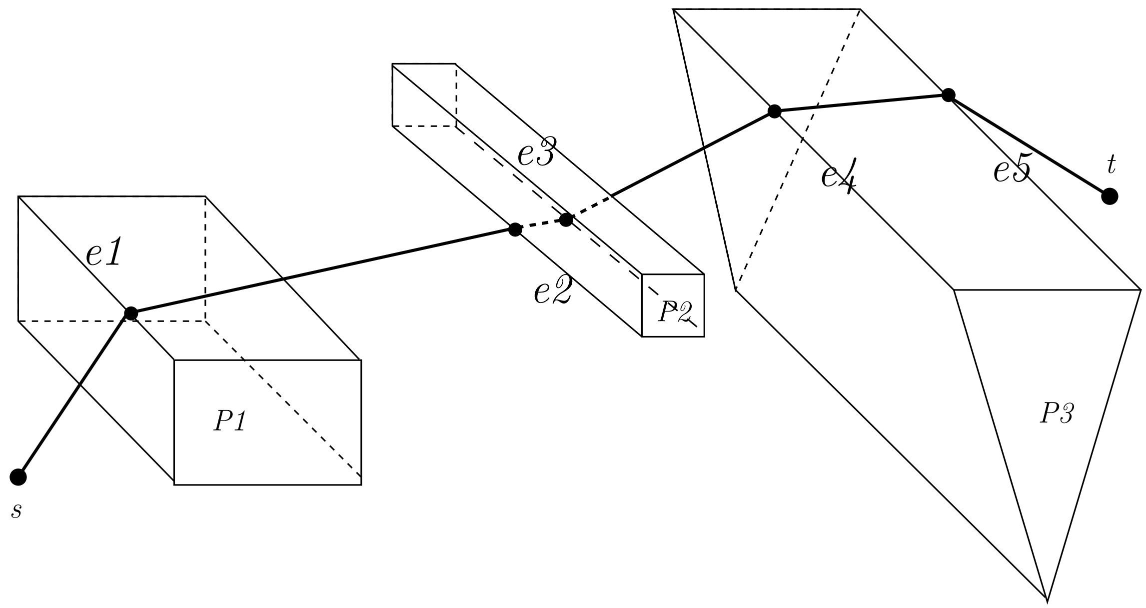
Polyeder-Szene in 3D

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

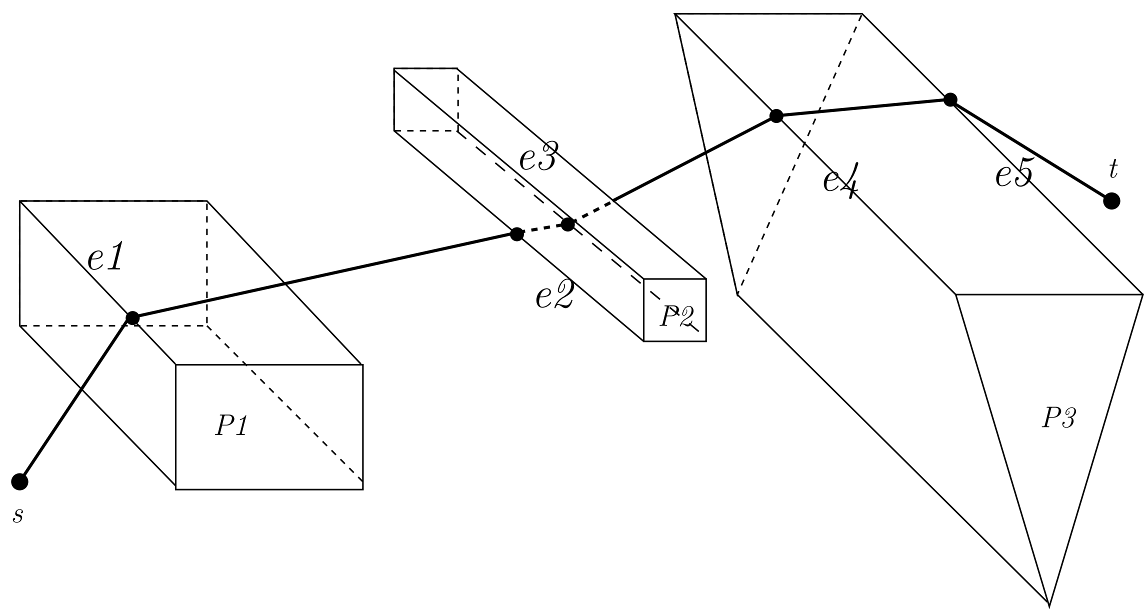


Polyeder-Szene in 3D

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

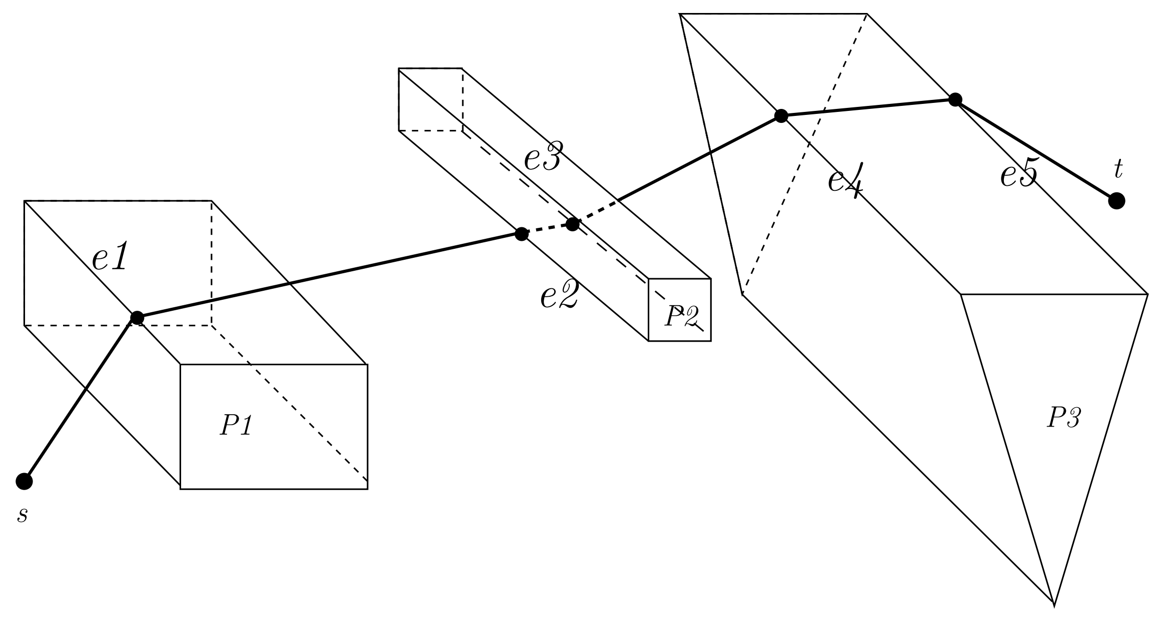


Polyeder-Szene in 3D



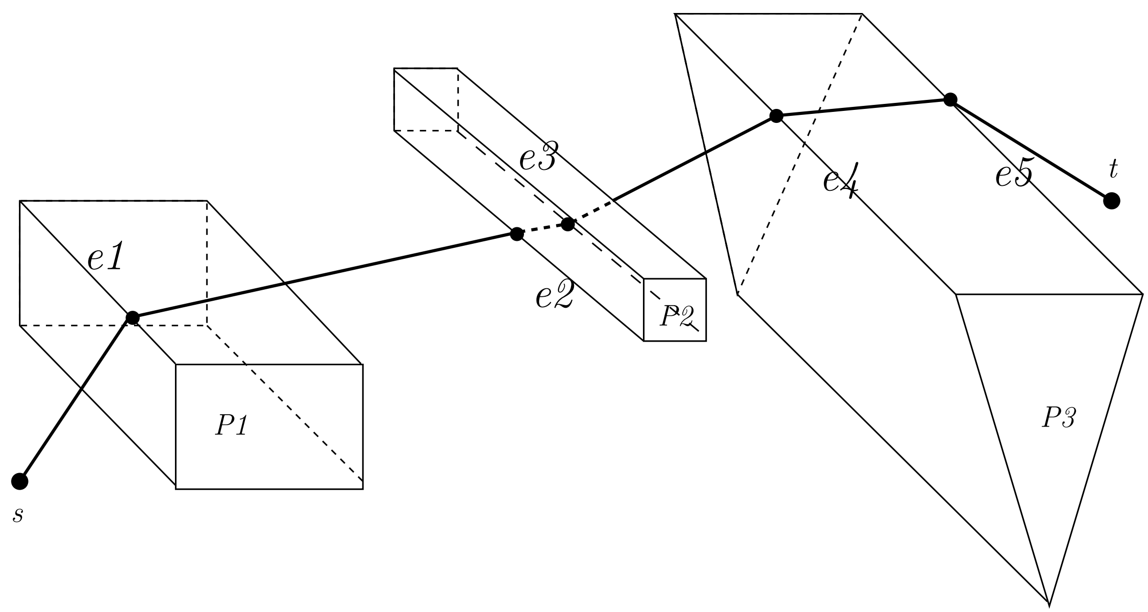
Polyeder-Szene in 3D

- Teilprobleme



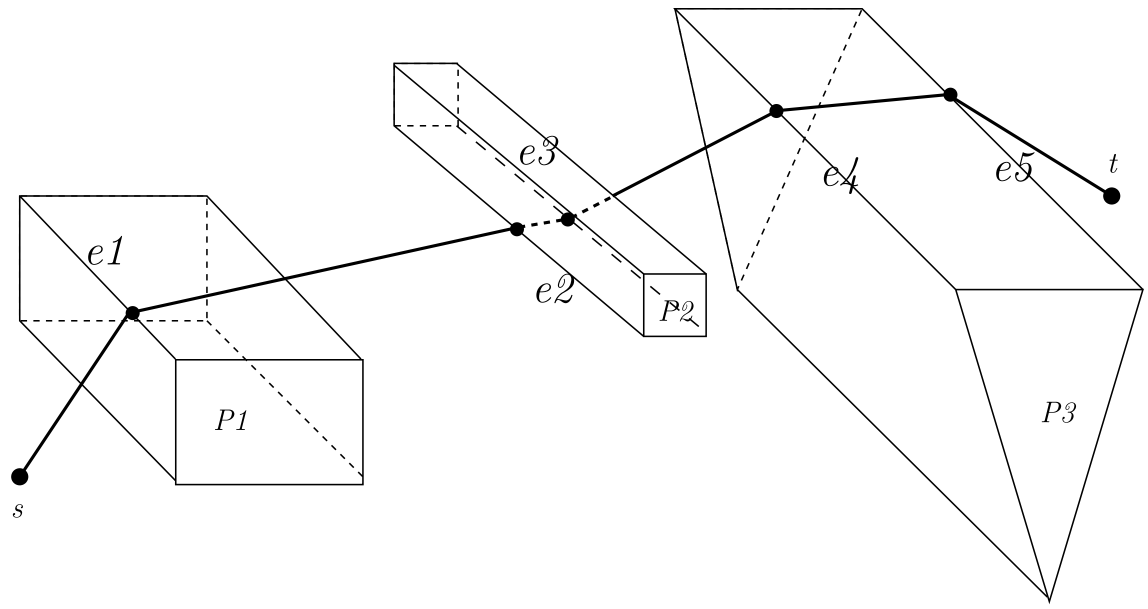
Polyeder-Szene in 3D

- Teilprobleme
- 1) Kantenreihenfolge



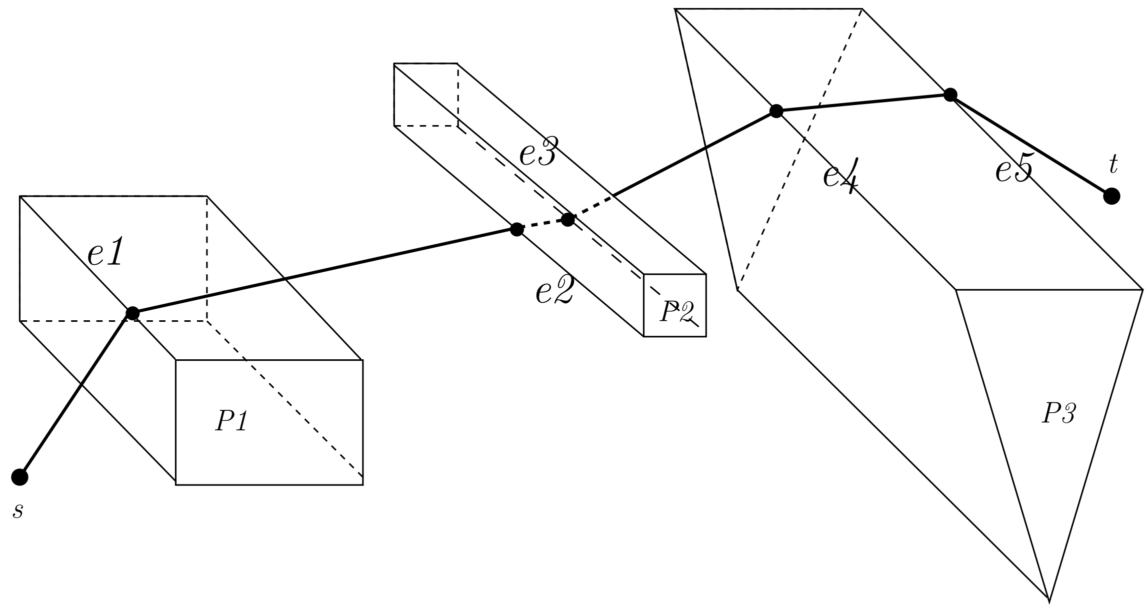
Polyeder-Szene in 3D

- Teilprobleme
- 1) Kantenreihenfolge
- 2) Verschiebung auf der Kante



Polyeder-Szene in 3D

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



Kantenreihenfolge: NP hard

Kantenreihenfolge: NP hard

- Reduktion $S' \subset \Omega'$ auf $S \subset \Omega$

Kantenreihenfolge: NP hard

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

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. $\forall x' \in \Omega'$: $f(x') \in S \Leftrightarrow x' \in S'$

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

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

3-SAT reduzieren auf Kantenreihenfolge

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

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:

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?

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:

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

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!

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

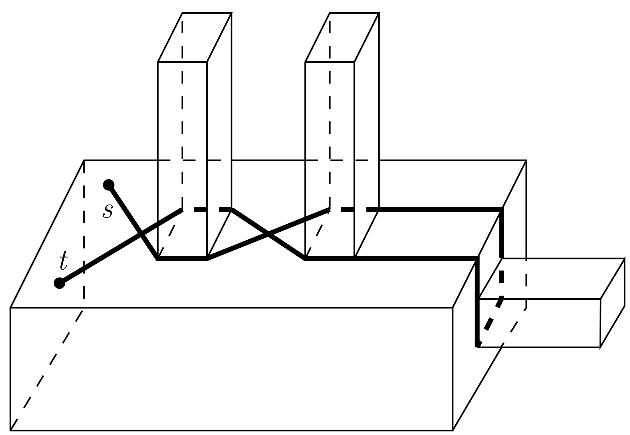
Parcours in $O(p(mn))$ erzeugen

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

- 2^n Belegungen der n Variablen

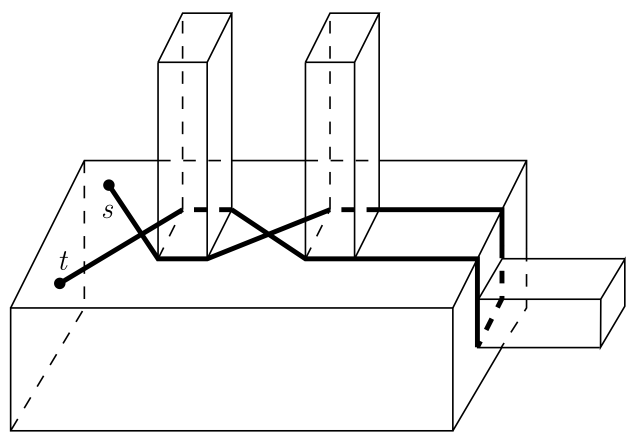
Parcours in $O(p(mn))$ erzeugen

- 2^n Belegungen der n Variablen
- 2^n *geodätisch* Kürzeste Wege



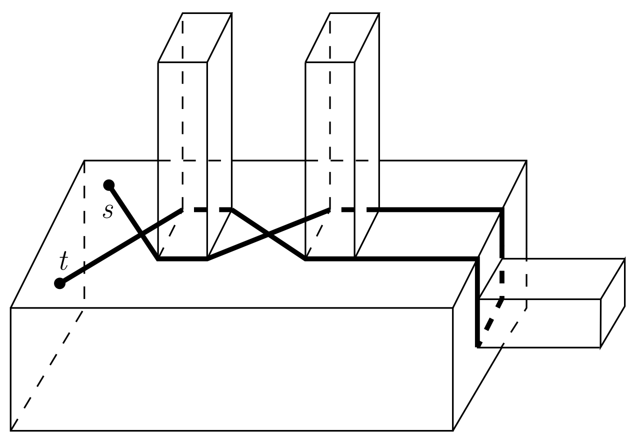
Parcours in $O(p(mn))$ erzeugen

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



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

Parcours erzeugen: Prinzip

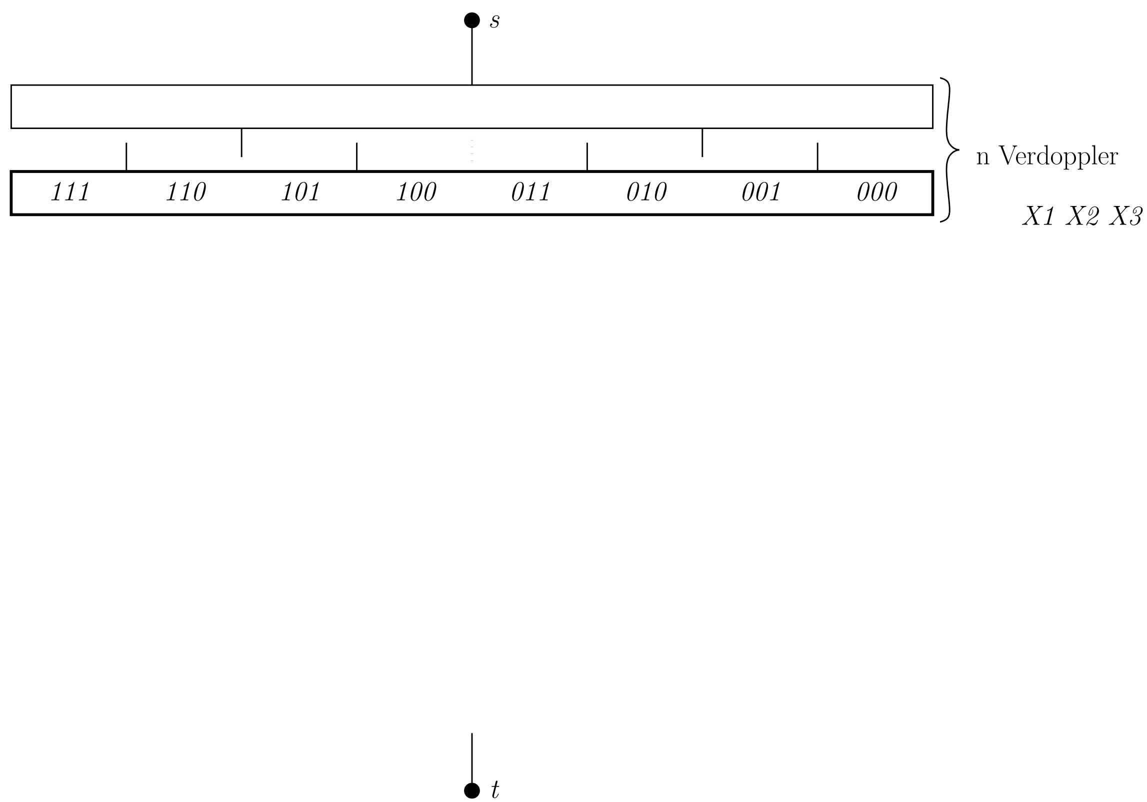
Beispiel: $(X1 \vee \neg X3) \wedge (\neg X2)$

Parcours erzeugen: Prinzip

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

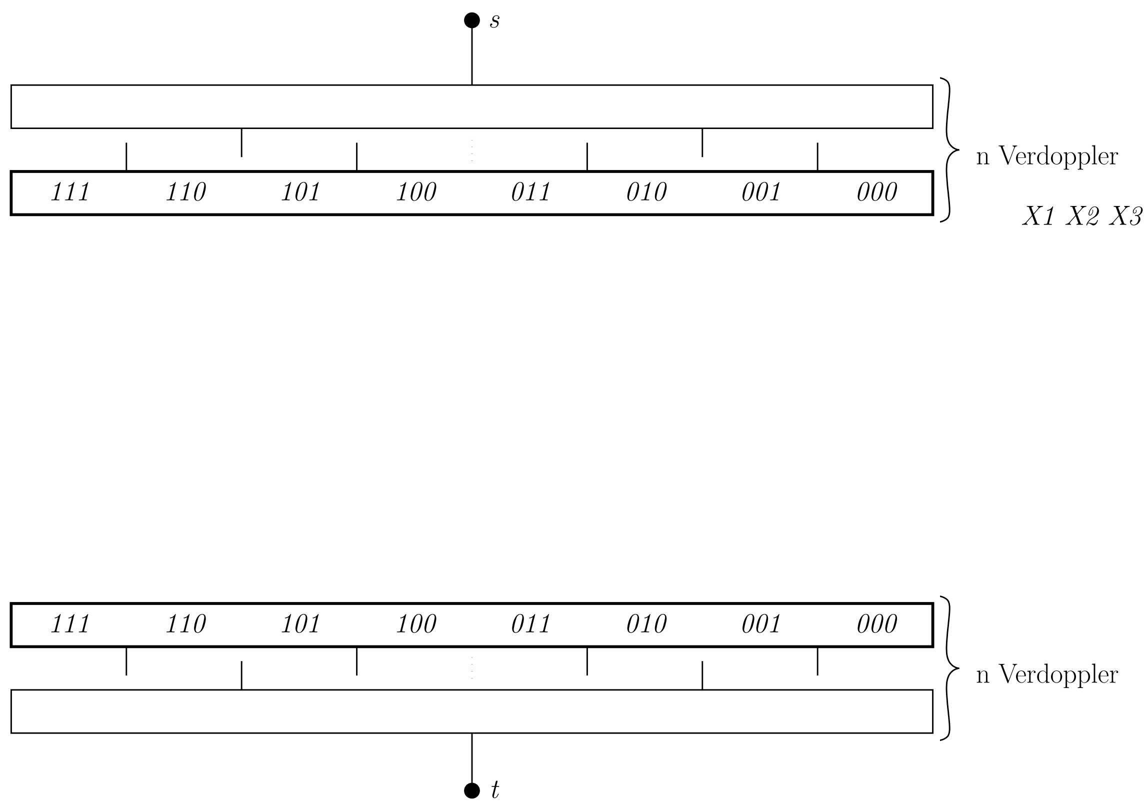
Parcours erzeugen: Prinzip

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



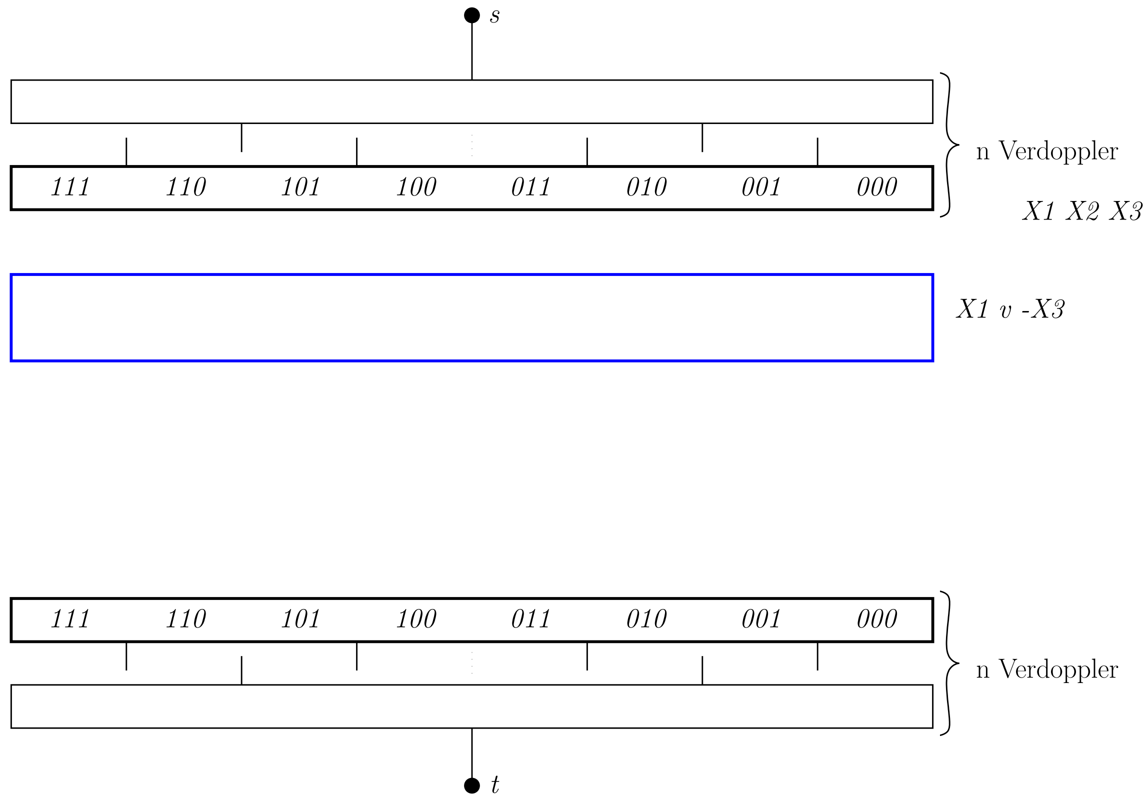
Parcours erzeugen: Prinzip

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



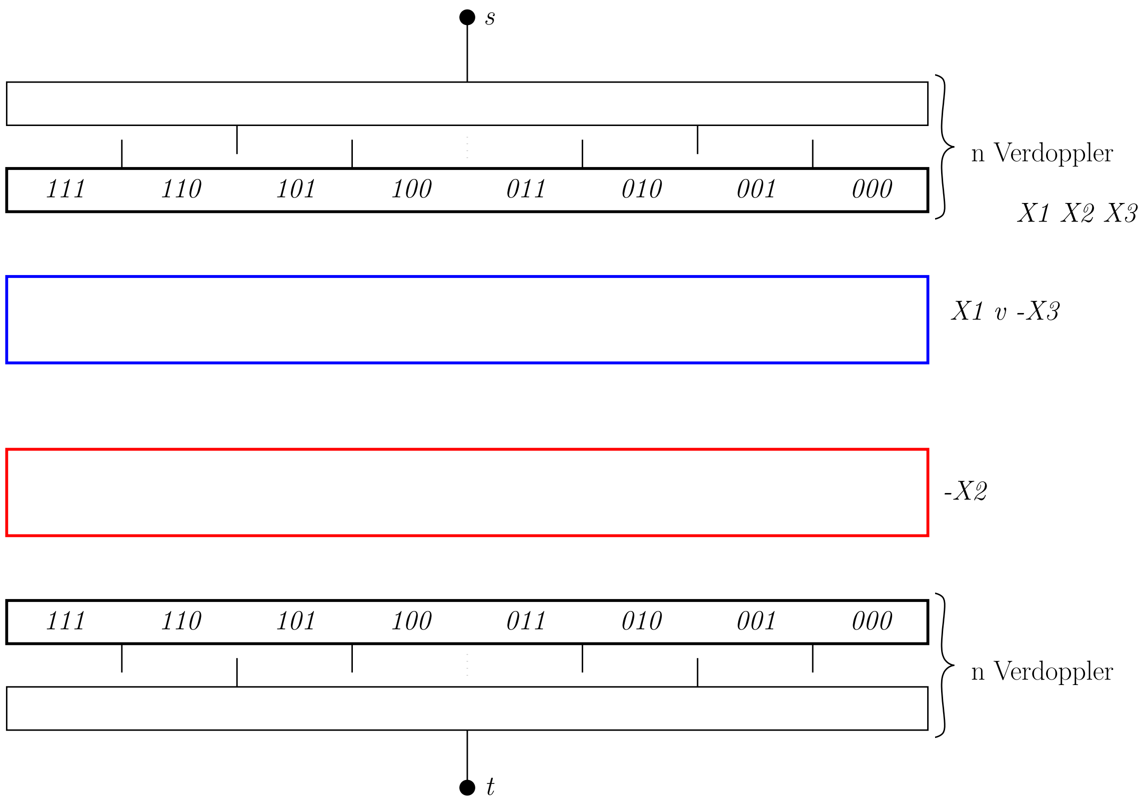
Parcours erzeugen: Prinzip

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



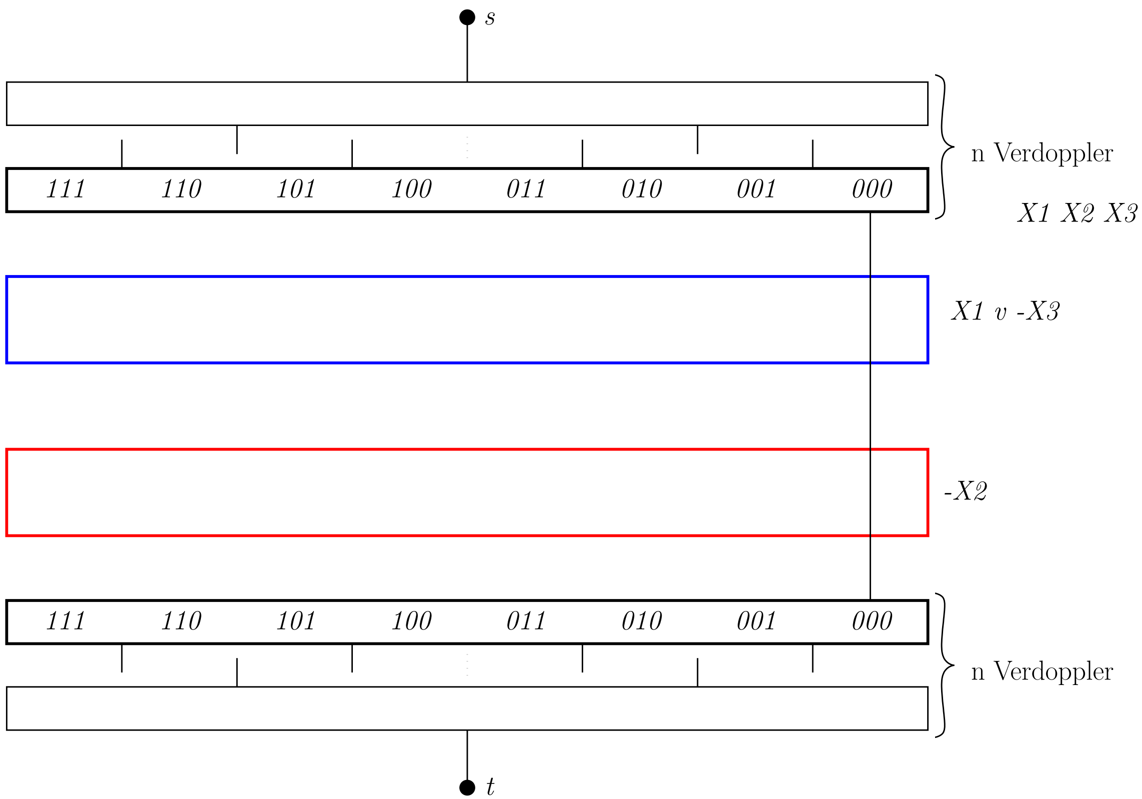
Parcours erzeugen: Prinzip

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



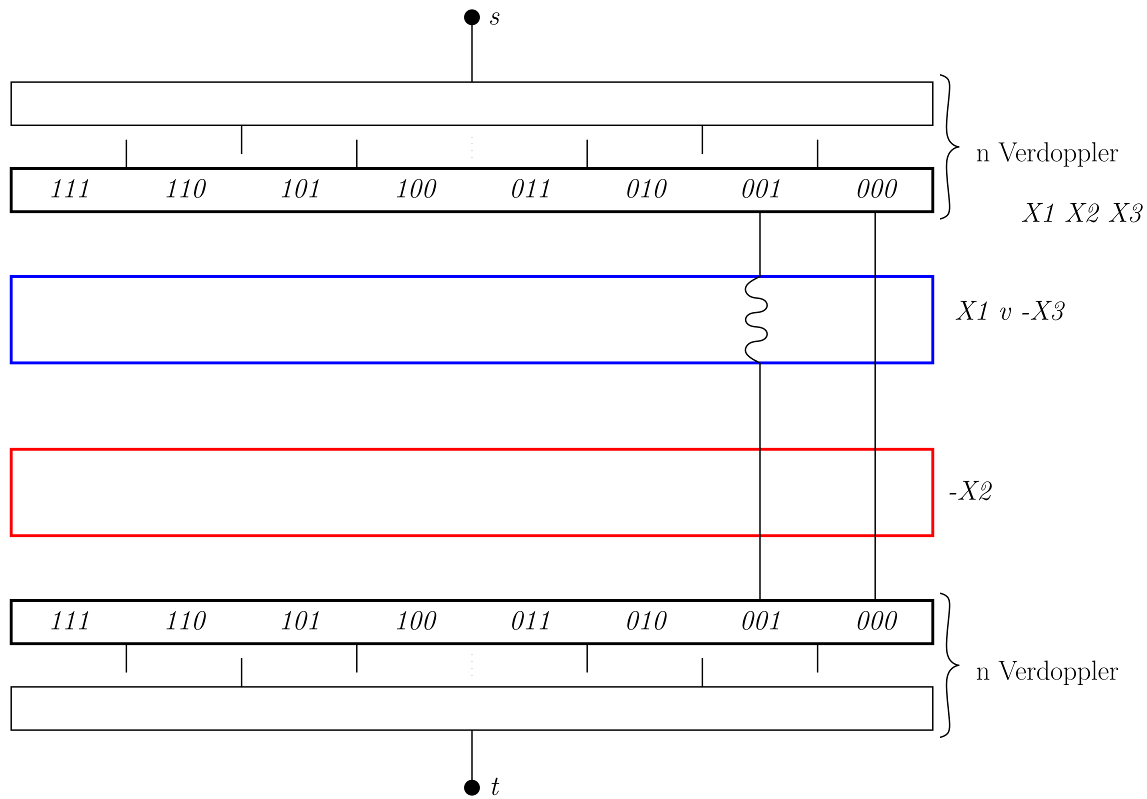
Parcours erzeugen: Prinzip

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



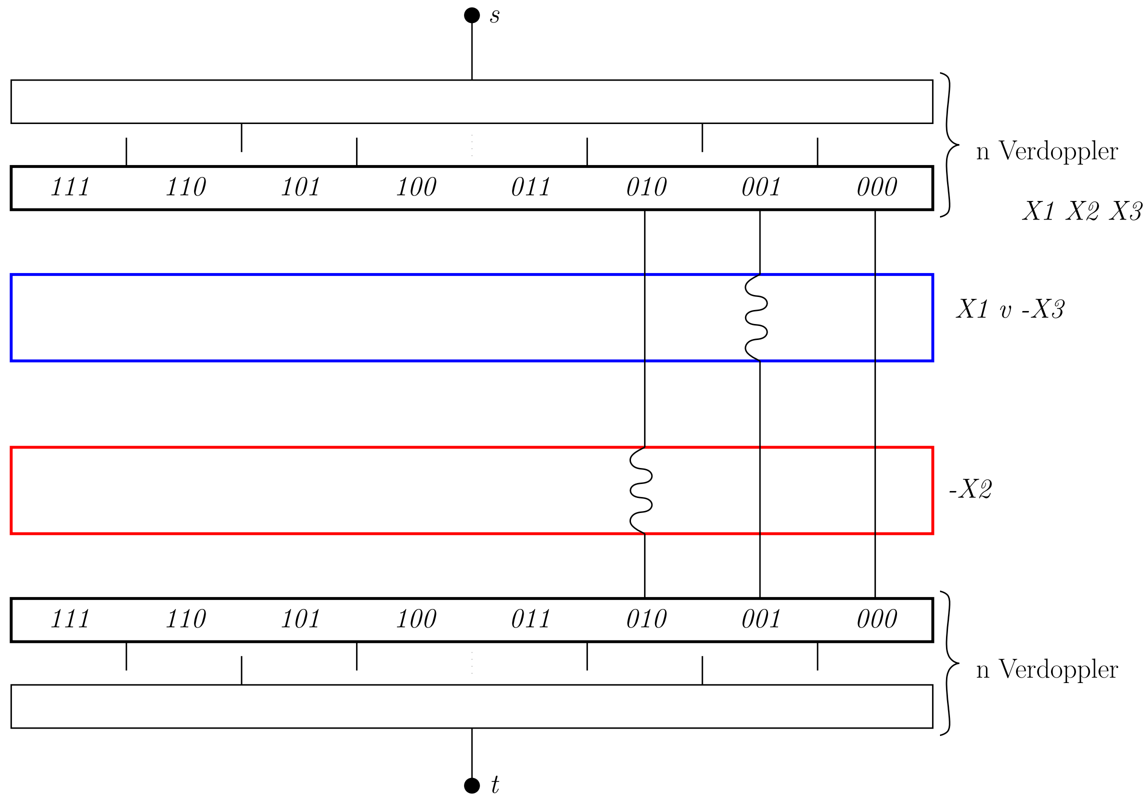
Parcours erzeugen: Prinzip

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



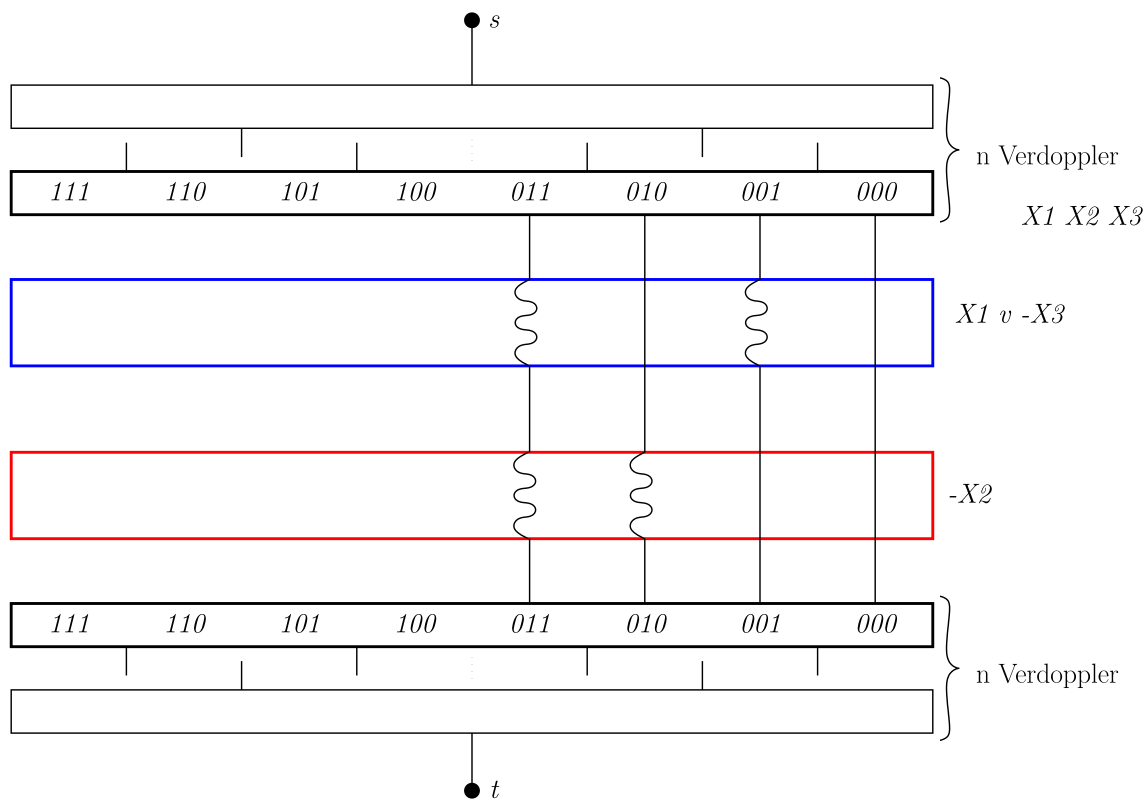
Parcours erzeugen: Prinzip

Beispiel: $(X1 \lor \neg X3) \land (\neg X2)$



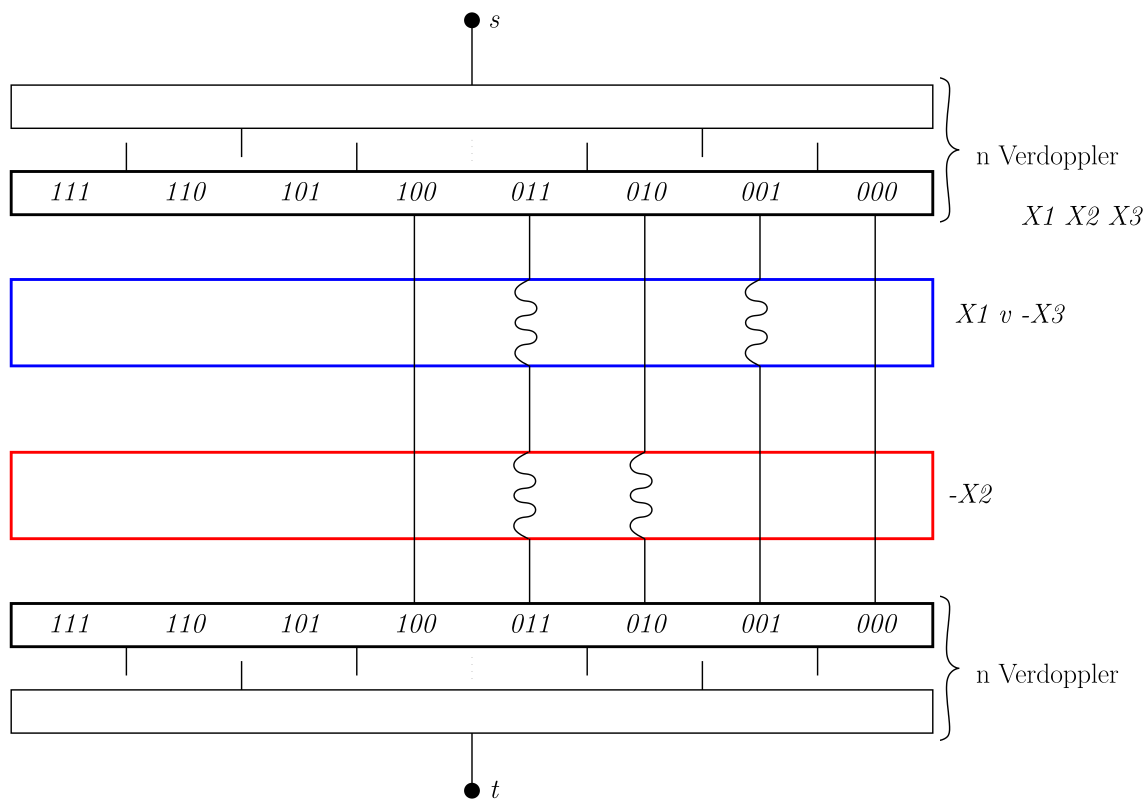
Parcours erzeugen: Prinzip

Beispiel: $(X1 \lor \neg X3) \land (\neg X2)$



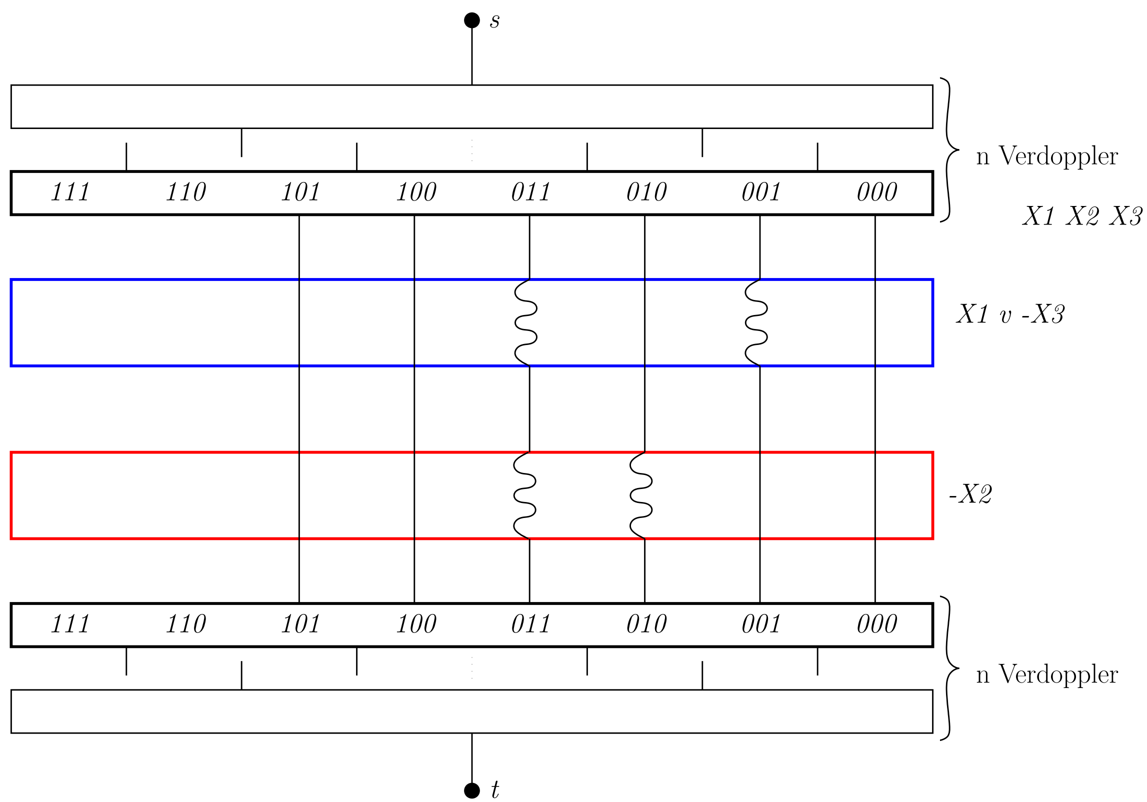
Parcours erzeugen: Prinzip

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



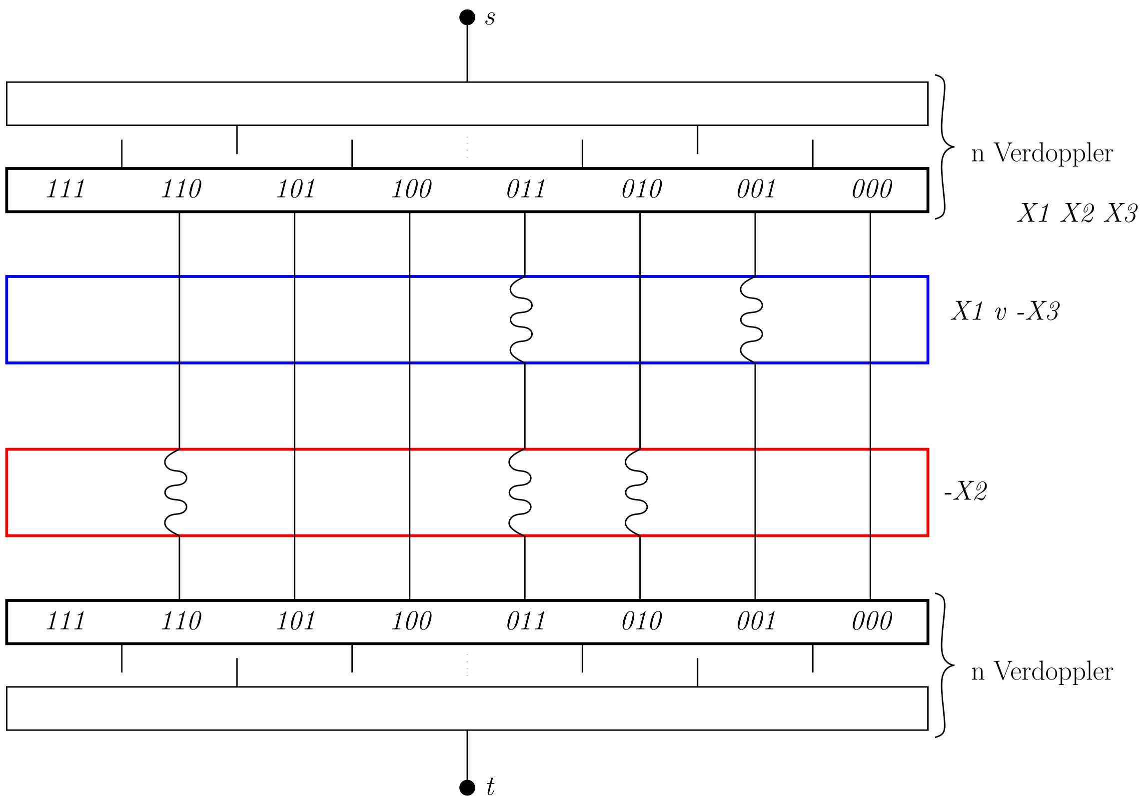
Parcours erzeugen: Prinzip

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



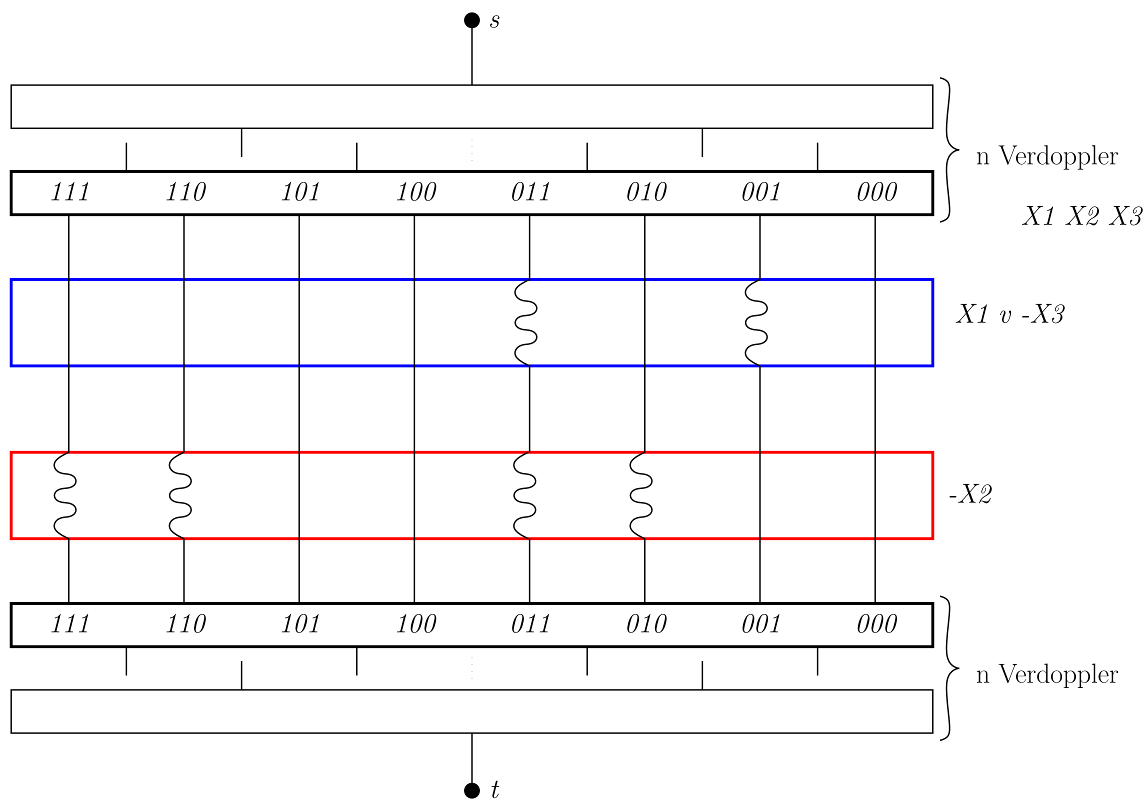
Parcours erzeugen: Prinzip

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



Parcours erzeugen: Prinzip

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



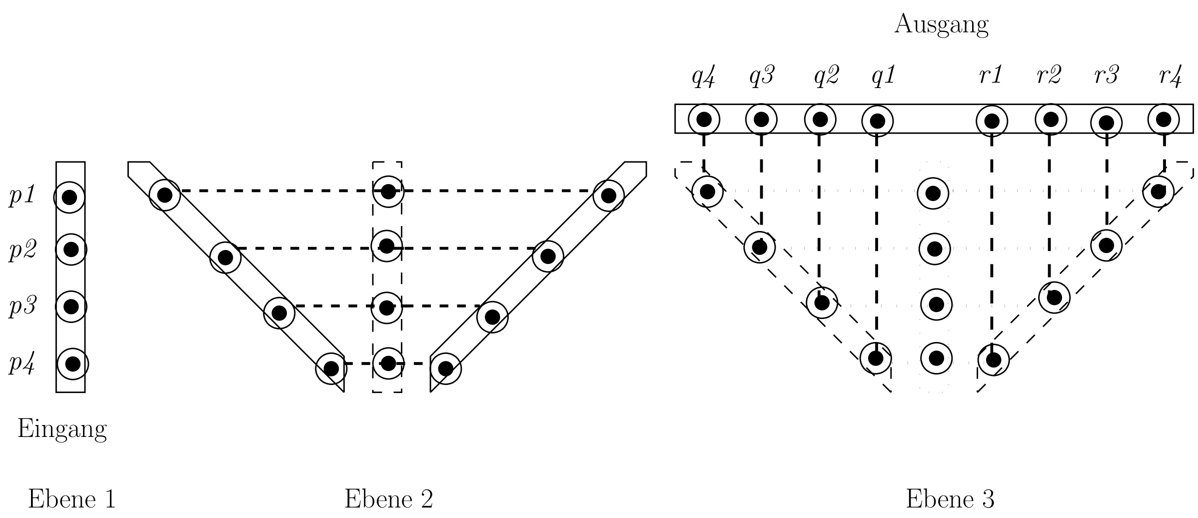
Komponenten: Verdoppler!

Komponenten: Verdoppler!

Dünne Platten mit Schlitzen eng hintereinander!

Komponenten: Verdoppler!

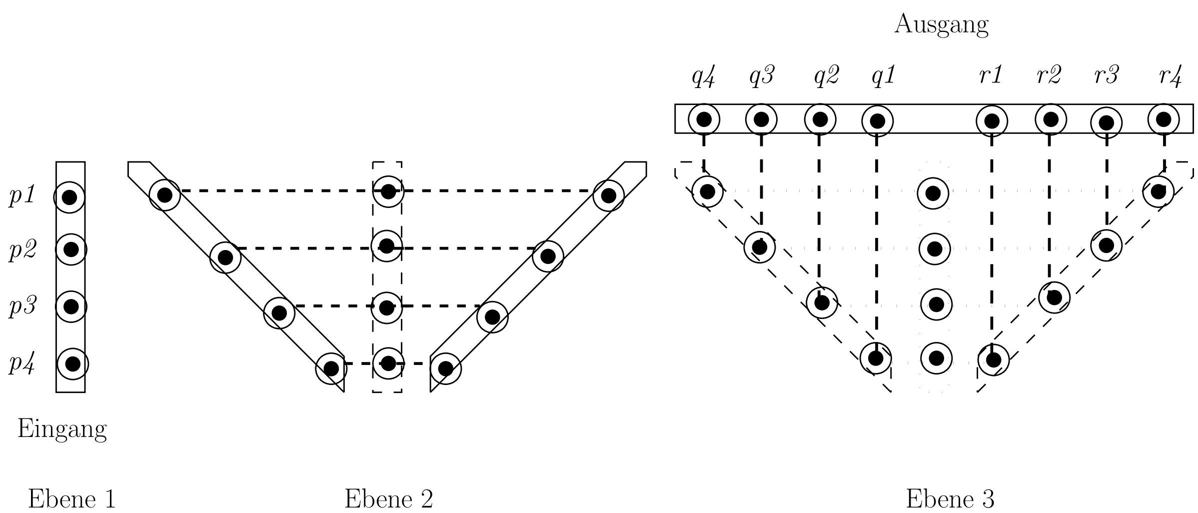
Dünne Platten mit Schlitzen eng hintereinander!



Sukzessive 2^n ungefähr gleichlange Wege erzeugen!

Komponenten: Verdoppler!

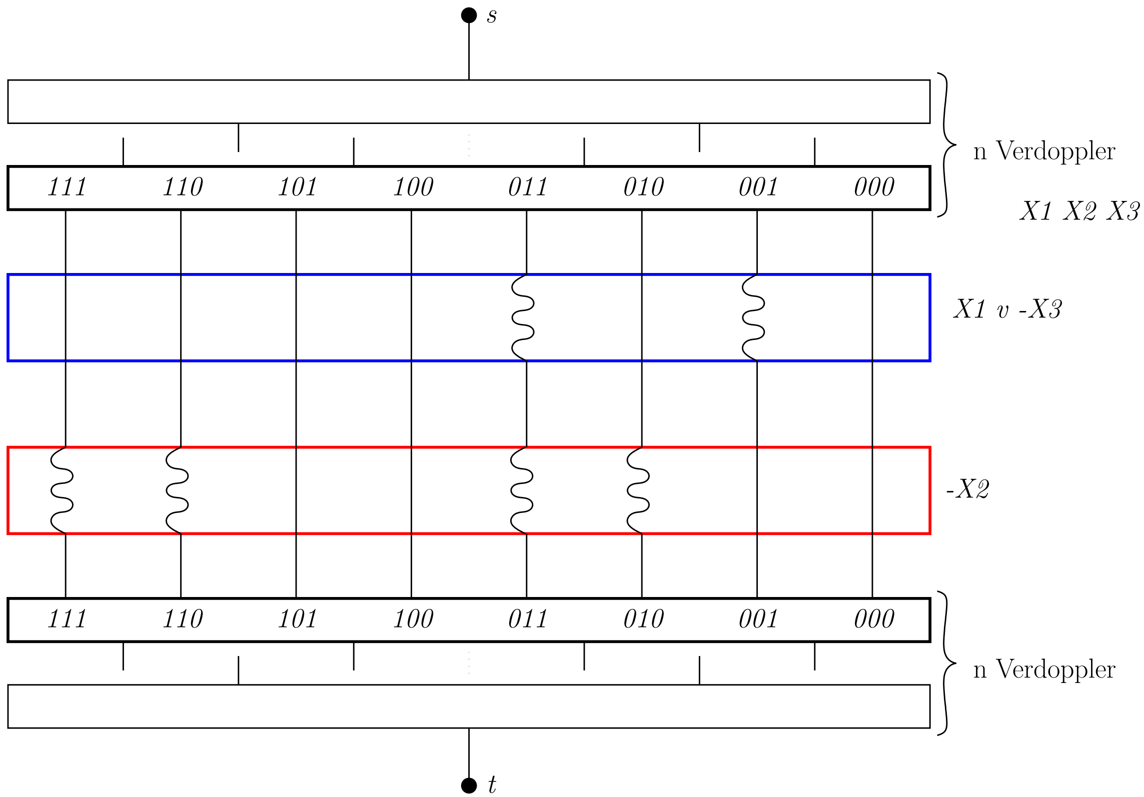
Dünne Platten mit Schlitzen eng hintereinander!



Sukzessive 2^n ungefähr gleichlange Wege erzeugen!

Kantenreihenfolge ist gleich!

Gesamtprinzip: Nacheinander Klauseln!



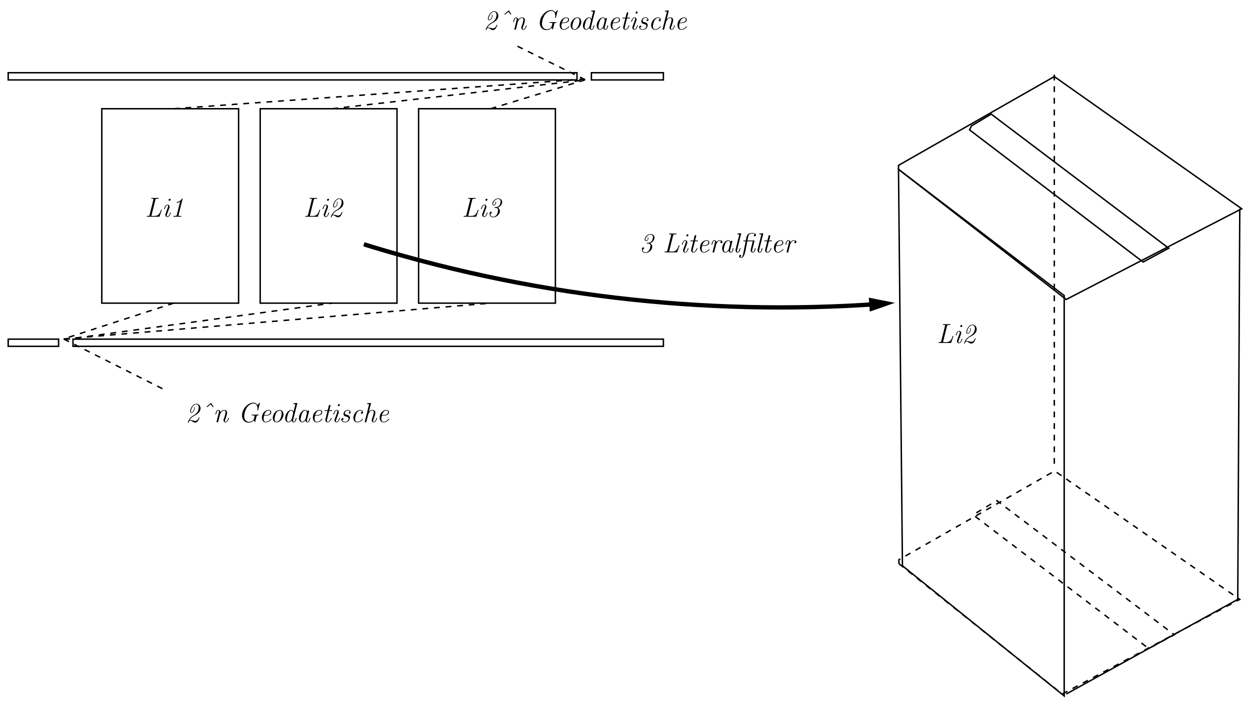
Komponenten: Klauselfilter

Komponenten: Klauselfilter

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

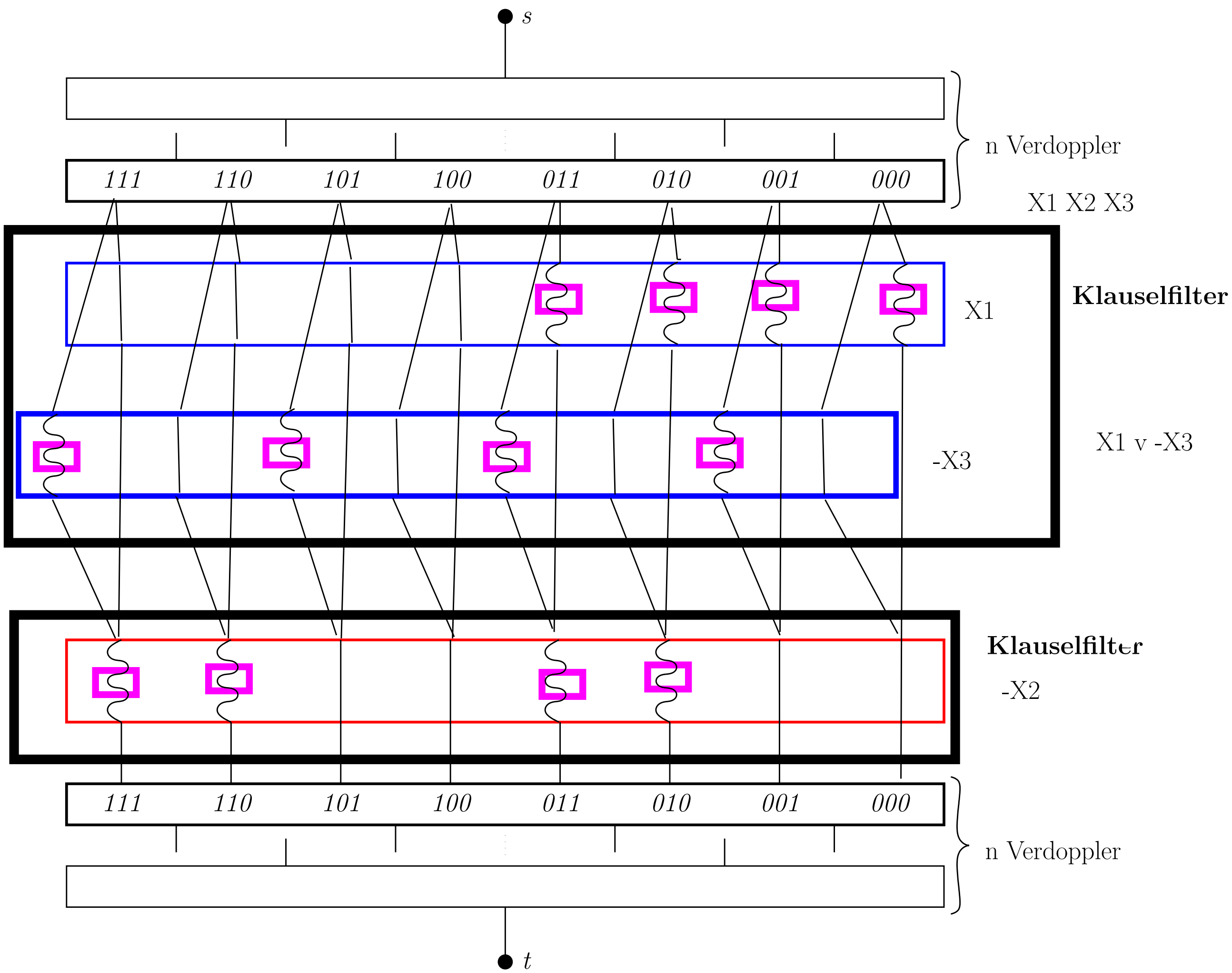
Komponenten: Klauselfilter

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



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

Gesamtprinzip: Einzelne Literale



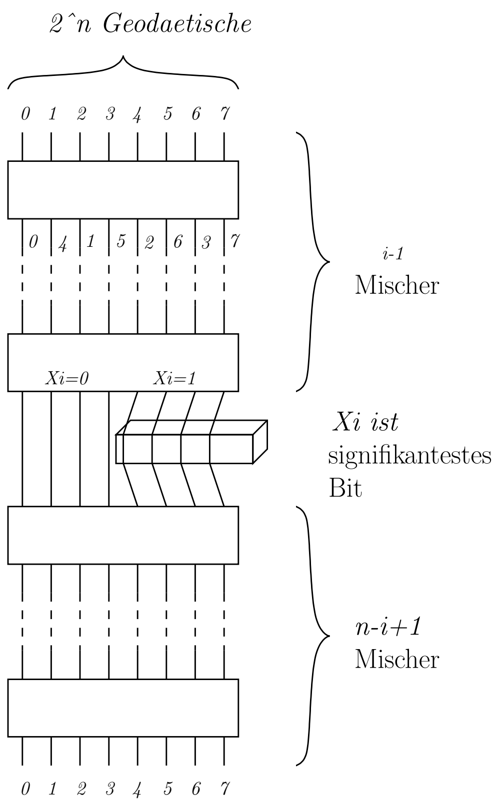
Komponenten: Literalfilter

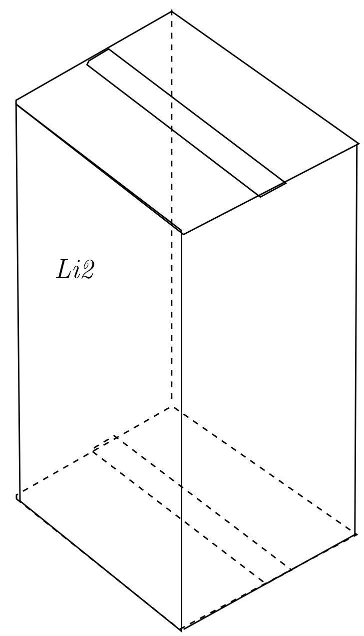
Komponenten: Literalfilter

Wege für signifikantes Bit sammeln!

Komponenten: Literalfilter

Wege für signifikantes Bit sammeln!





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

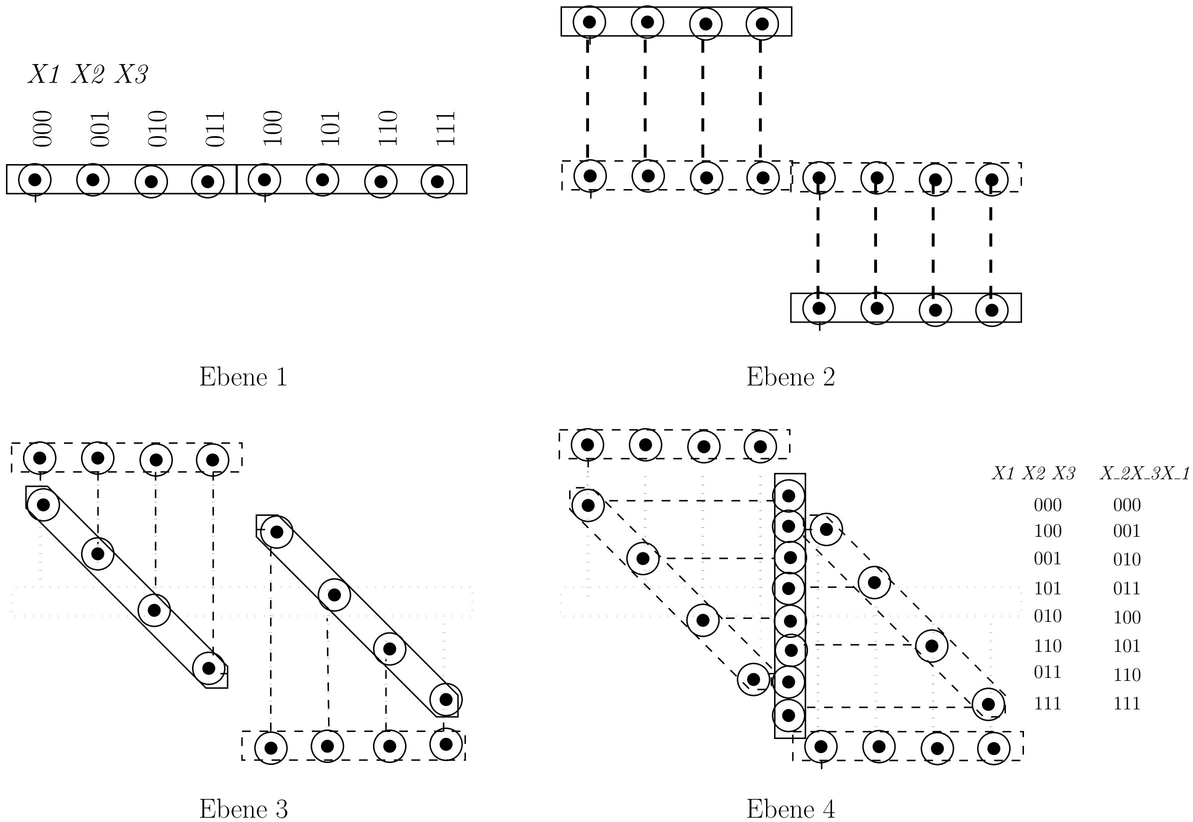
n Mischer pro Literalfilter!!

n Mischer pro Literalfilter!!

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

n **Mischer pro Literalfilter!!**

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



Kürzeste Wege Alg. für P_α

Kürzeste Wege Alg. für P_α

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

Kürzeste Wege Alg. für P_α

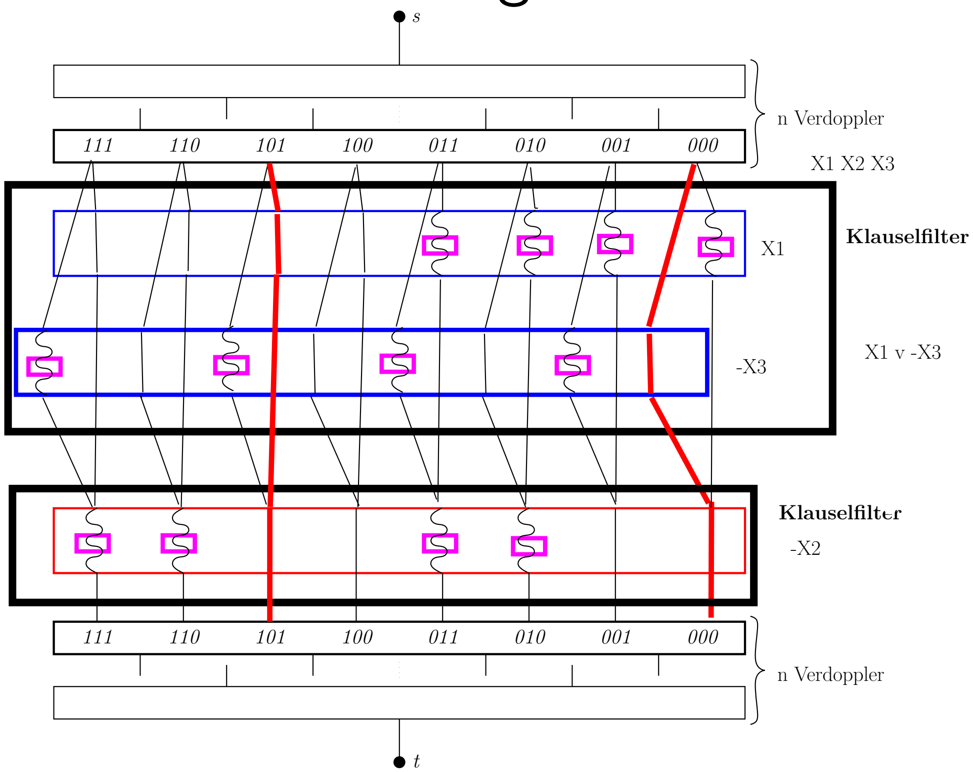
Ein Weg, der nicht verlängert wird entspricht genau einer Belegung, die die Formel erfüllt!!!

Das kann man der Kantenreihenfolge entnehmen!!

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!!

Konstruktion insgesamt!!

- $2n$ Verdoppler: $O(n)$ Kanten

Konstruktion insgesamt!!

- $2n$ Verdoppler: $O(n)$ Kanten
- m Klauselfilter: je Klauselfilter
  - 3 Literalfilter

Konstruktion insgesamt!!

- $2n$ Verdoppler: $O(n)$ Kanten
- m Klauselfilter: je Klauselfilter
  - 3 Literalfilter
  - n Mischer je Filter

Konstruktion insgesamt!!

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

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!!!

Ergebnis!!!

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

Ergebnis!!!

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

Beweis!!

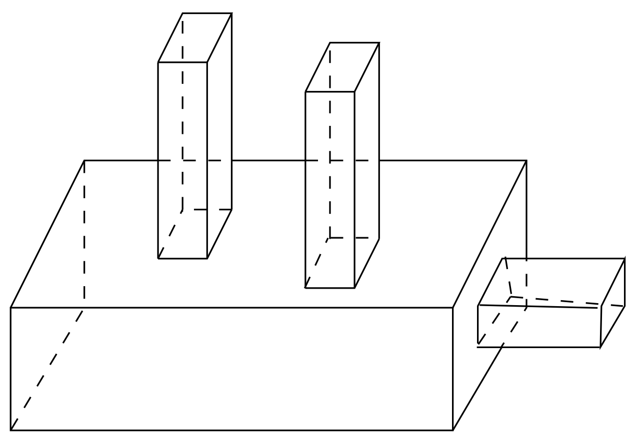
Downgrading: Oberfläche Polyeder!

Downgrading: Oberfläche Polyeder!

- Natürliche Erweiterung der Polygone auf 3D

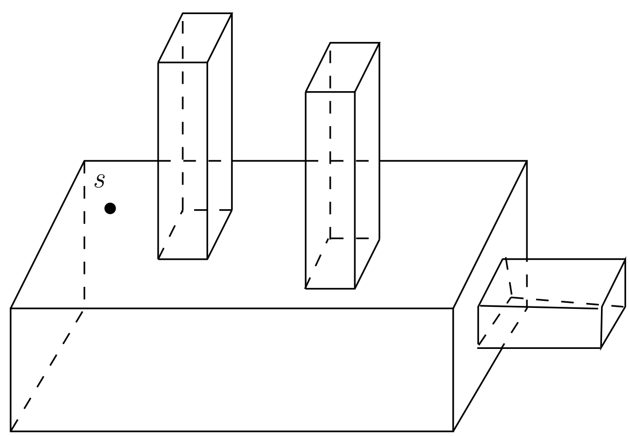
Downgrading: Oberfläche Polyeder!

- Natürliche Erweiterung der Polygone auf 3D



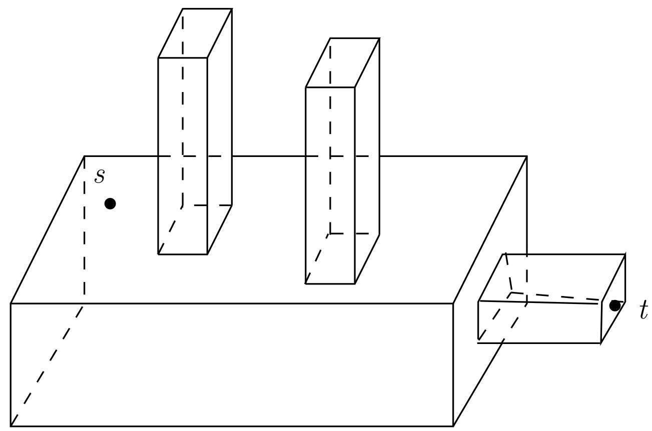
Downgrading: Oberfläche Polyeder!

- Natürliche Erweiterung der Polygone auf 3D



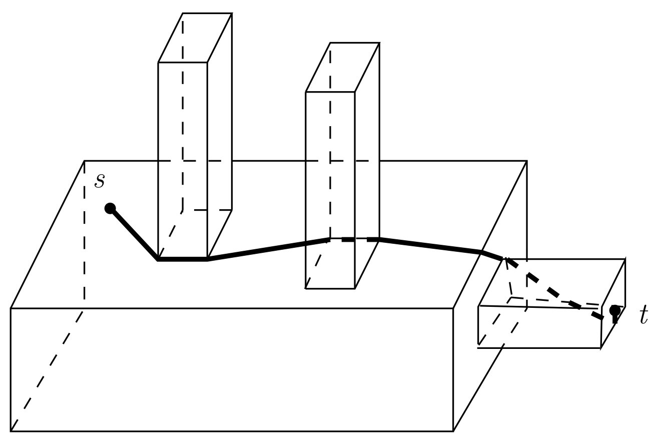
Downgrading: Oberfläche Polyeder!

- Natürliche Erweiterung der Polygone auf 3D



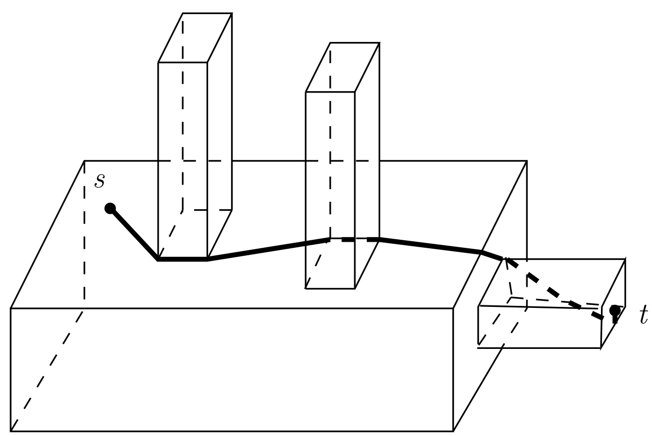
Downgrading: Oberfläche Polyeder!

- Natürliche Erweiterung der Polygone auf 3D



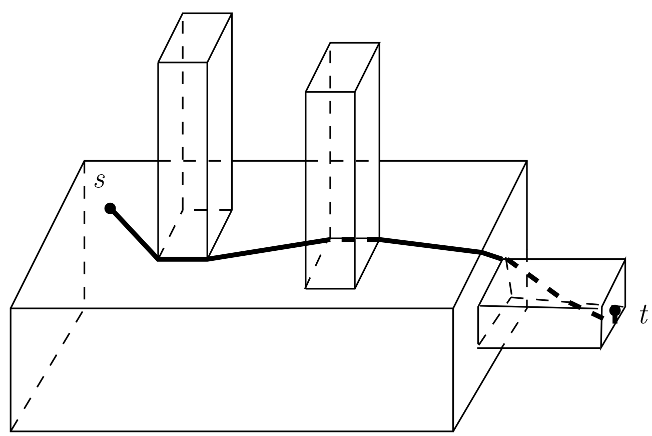
Downgrading: Oberfläche Polyeder!

- Natürliche Erweiterung der Polygone auf 3D
- Rand besteht aus Polygonen



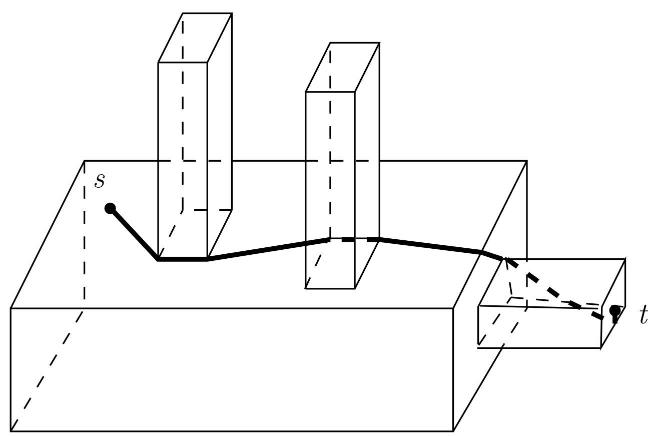
Downgrading: Oberfläche Polyeder!

- Natürliche Erweiterung der Polygone auf 3D
- Rand besteht aus Polygonen
- Keine dünnen Stellen:



Downgrading: Oberfläche Polyeder!

- Natürliche Erweiterung der Polygone auf 3D
- Rand besteht aus Polygonen
- Keine dünnen Stellen: ϵ-Kugeln



Downgrading: Oberfläche Polyeder!

- Natürliche Erweiterung der Polygone auf 3D
- Rand besteht aus Polygonen
- Keine dünnen Stellen: ϵ-Kugeln
- Datenstruktur QEDS: Triangulation Oberflächen, Navigation!