

Algorithmen zur Visualisierung von Graphen

Flussmethoden: orthogonales Graphenzeichnen

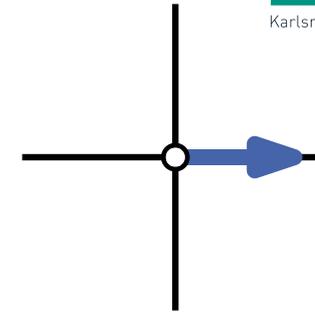
INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Tamara Mchedlidze · **Martin Nöllenburg**
12.01.2015

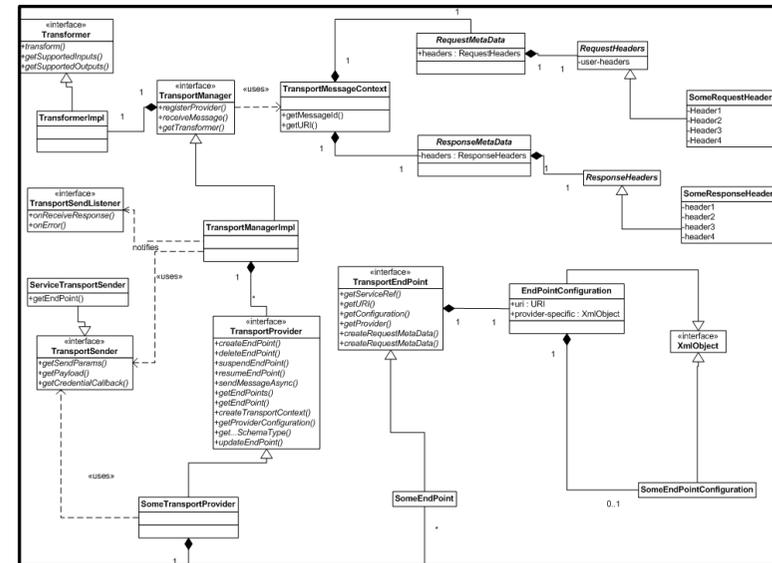
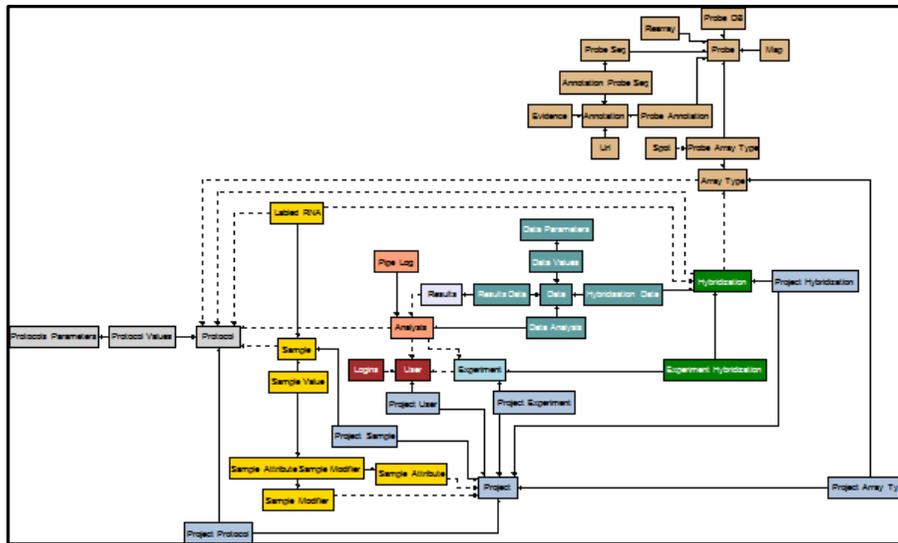


Orthogonales Graphenzeichnen

- Kantenrichtungen horizontal und vertikal
- gut untersuchte Zeichenkonvention
- in vielen Anwendungen relevant

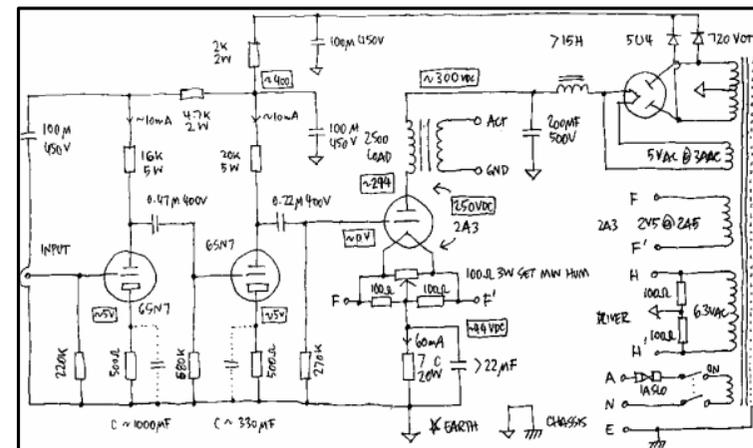
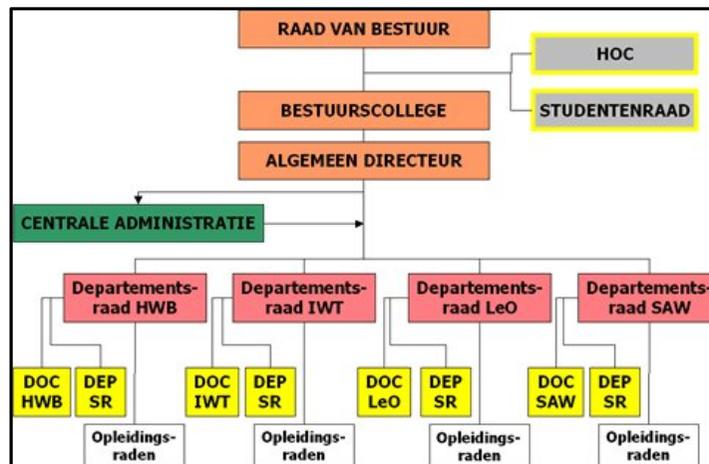


ER diagram in OGDf



UML diagram by Oracle

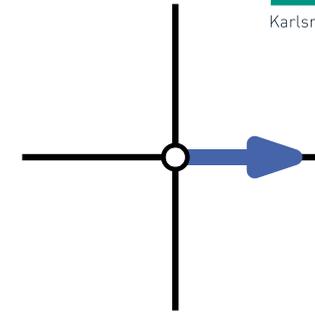
Organigram of HS Limburg



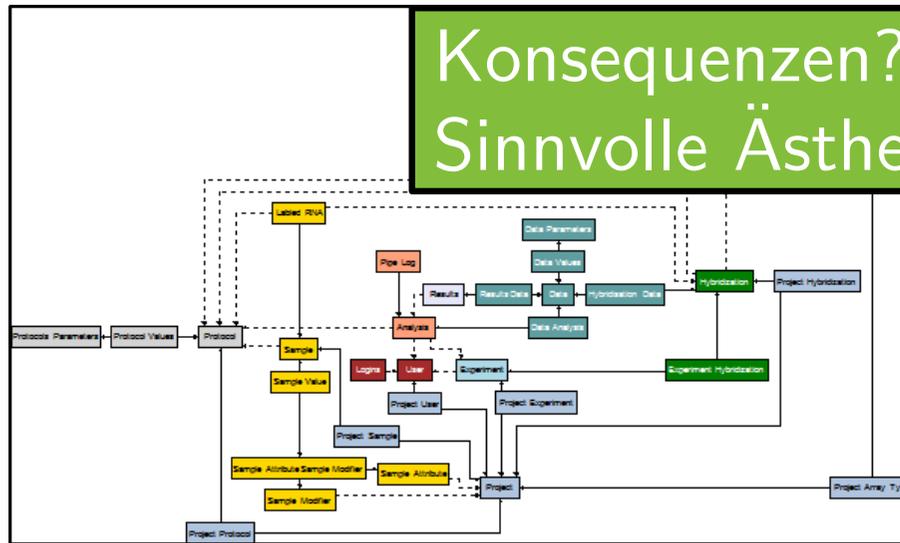
Circuit diagram by Jeff Atwood

Orthogonales Graphenzeichnen

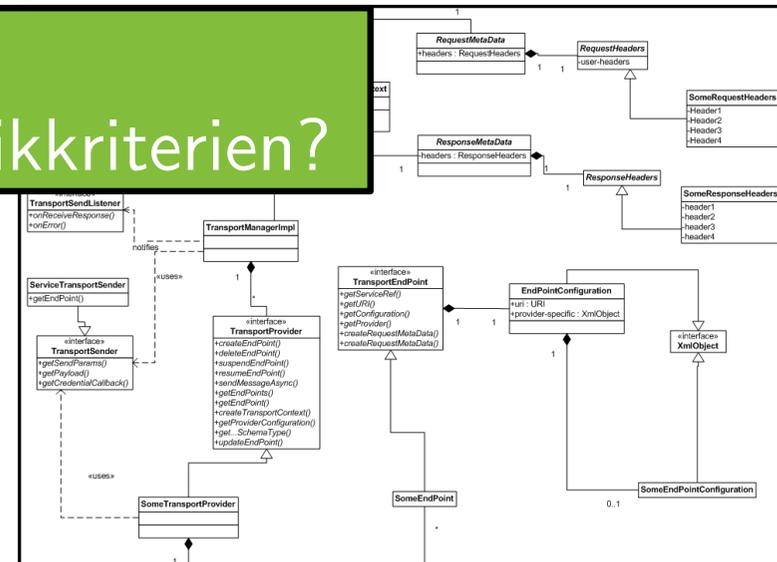
- Kantenrichtungen horizontal und vertikal
- gut untersuchte Zeichenkonvention
- in vielen Anwendungen relevant



ER diagram in OGDF

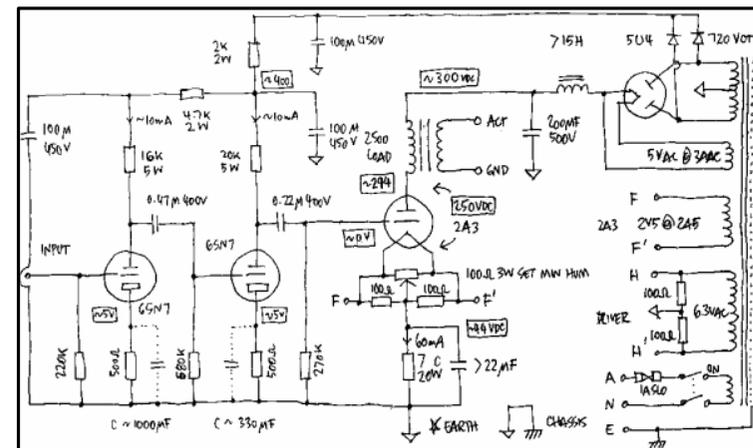
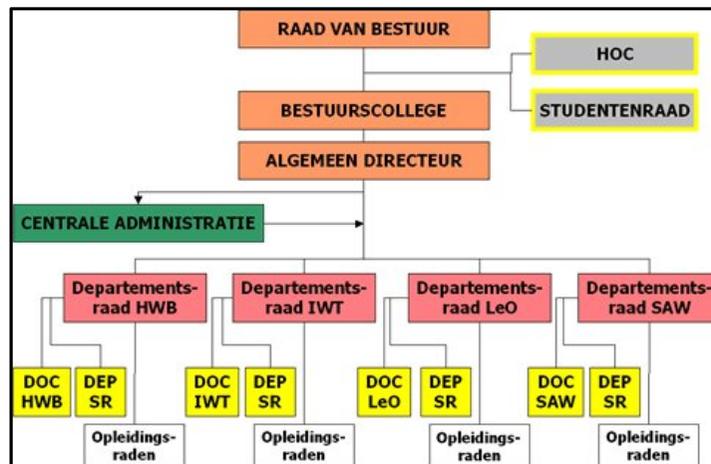


Konsequenzen?
Sinnvolle Ästhetikkriterien?



UML diagram by Oracle

Organigram of HS Limburg



Circuit diagram by Jeff Atwood

(Planare) Orthogonale Zeichnungen

Dreistufiger Ansatz: *Topology – Shape – Metrics*

[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

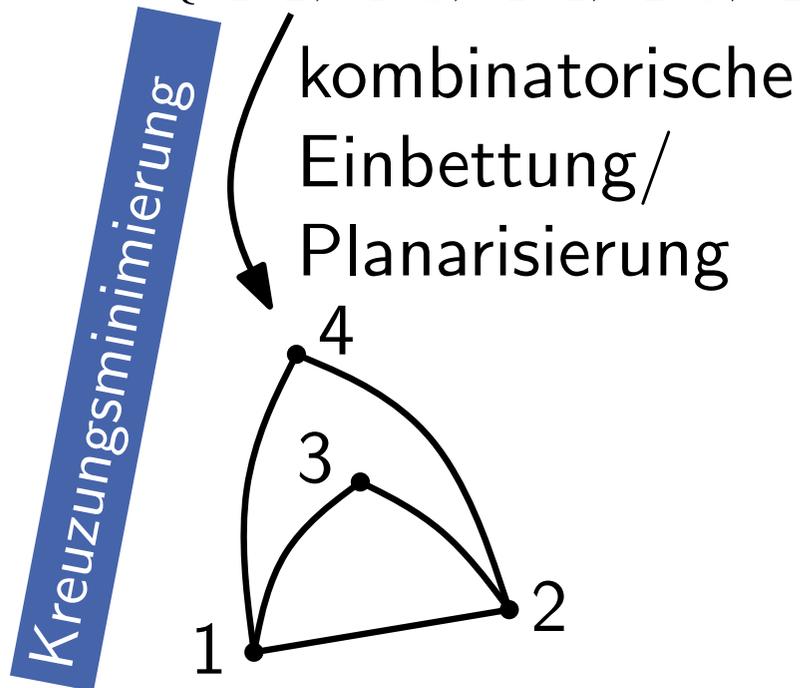
(Planare) Orthogonale Zeichnungen

Dreistufiger Ansatz: *Topology – Shape – Metrics*

[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$



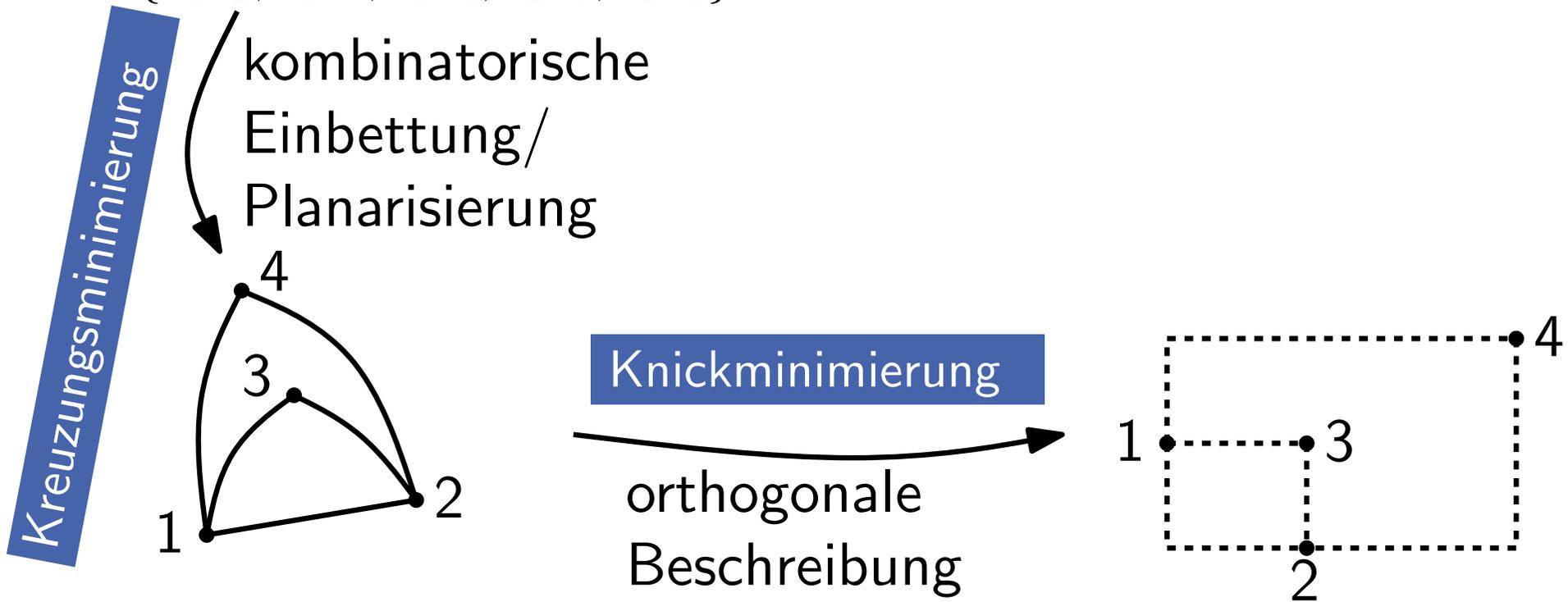
(Planare) Orthogonale Zeichnungen

Dreistufiger Ansatz: *Topology – Shape – Metrics*

[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$



(Planare) Orthogonale Zeichnungen

Dreistufiger Ansatz: *Topology – Shape – Metrics*

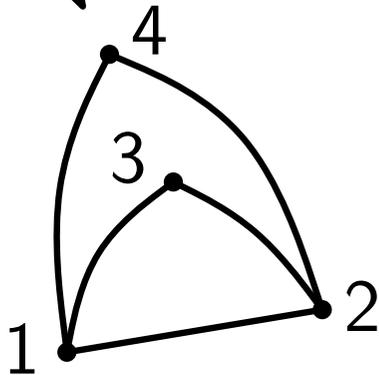
[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

Kreuzungsminimierung

kombinatorische
Einbettung/
Planarisierung

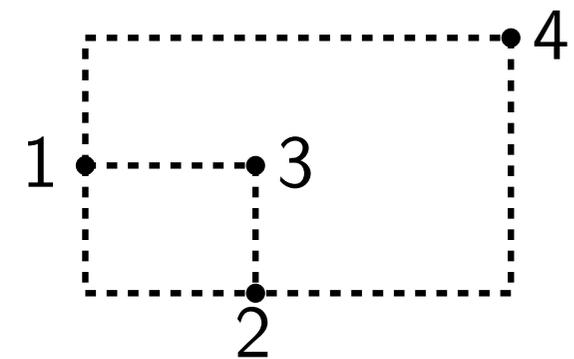
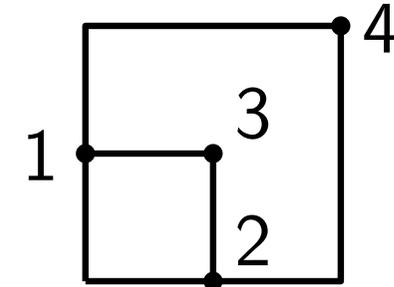


Knickminimierung

orthogonale
Beschreibung

planare
Einbettung

Flächen-
minimierung



(Planare) Orthogonale Zeichnungen

Dreistufiger Ansatz: *Topology – Shape – Metrics*

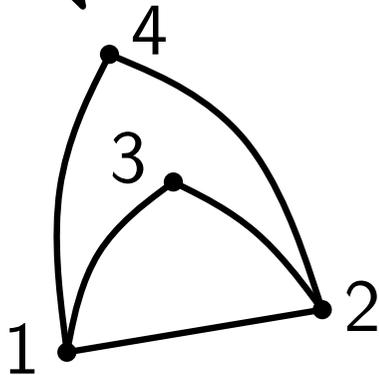
[Tamassia SIAM J. Comput. 1987]

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

Kreuzungsminimierung

kombinatorische
Einbettung/
Planarisierung

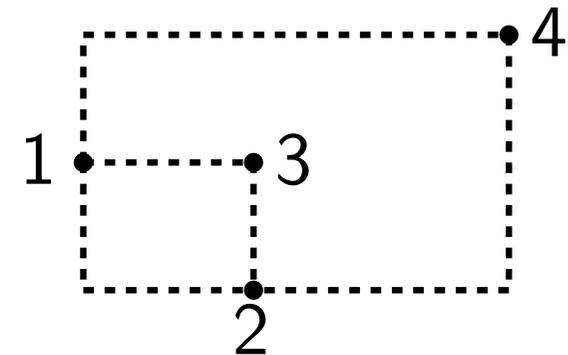
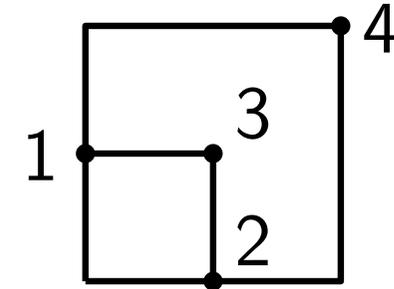


Knickminimierung

orthogonale
Beschreibung

planare
Einbettung

Flächen-
minimierung



Knickminimierung bei fester Einbettung

Problem geometrische Knickminimierung

Geg: ■ planarer Graph $G = (V, E)$ mit Maximalgrad 4
■ kombinatorische Einbettung \mathcal{F} und äußere Facette f_0

Ges: einbettungsäquivalente orthogonale Gitterzeichnung mit minimaler Knickanzahl

Knickminimierung bei fester Einbettung

Problem geometrische Knickminimierung

- Geg:
- planarer Graph $G = (V, E)$ mit Maximalgrad 4
 - kombinatorische Einbettung \mathcal{F} und äußere Facette f_0

Ges: einbettungsäquivalente orthogonale Gitterzeichnung mit minimaler Knickanzahl

bzw. zunächst folgende Variante

Problem kombinatorische Knickminimierung

- Geg:
- planarer Graph $G = (V, E)$ mit Maximalgrad 4
 - kombinatorische Einbettung \mathcal{F} und äußere Facette f_0

Ges: einbettungsäquivalente **orthogonale Beschreibung** $H(G)$ mit minimaler Knickanzahl

Orthogonale Beschreibung

Eingabe: planarer Graph $G = (V, E)$, Facettenmenge \mathcal{F} ,
äußere Facette f_0

Ausgabe: orthogonale Beschreibung $H(G) = \{H(f) \mid f \in \mathcal{F}\}$

Facettenbeschreibung $H(f)$: im UZS um f geordnete Folge
von Kantenbeschreibungen (e, δ, α) mit

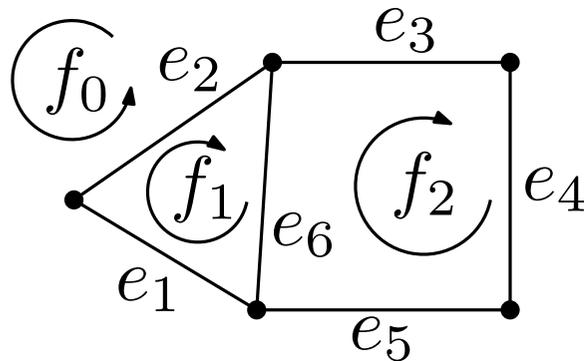
- e ist Randkante von f
- δ ist Folge in $\{0, 1\}^*$ ($0 =$ Rechtsknick, $1 =$ Linksknick)
- α ist Winkel $\in \{\frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi\}$ zwischen e und Nachfolger e'

Orthogonale Beschreibung: Beispiel

$$H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2}))$$

$$H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi))$$

$$H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2}))$$



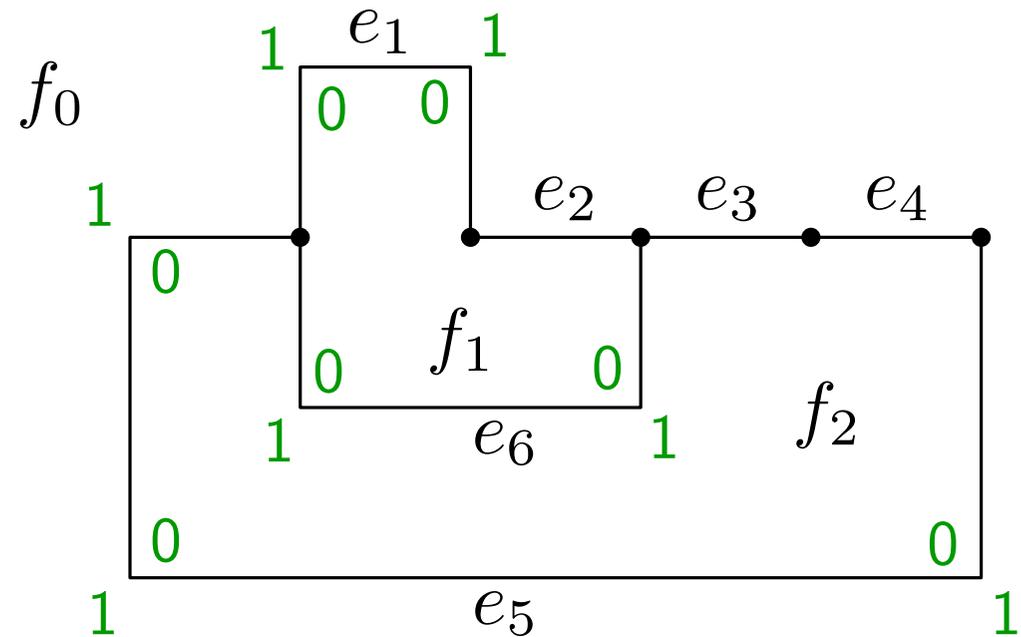
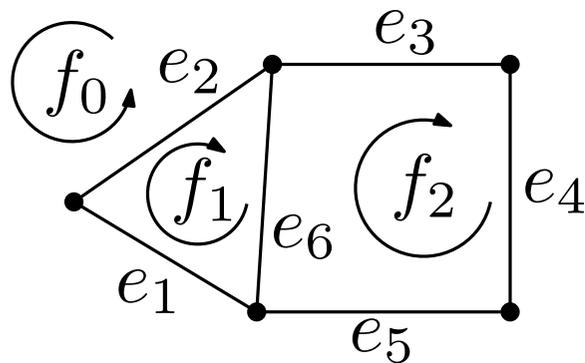
Wie sieht die kombinatorische
„Zeichnung“ zu $H(G)$ aus?

Orthogonale Beschreibung: Beispiel

$$H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2}))$$

$$H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi))$$

$$H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2}))$$



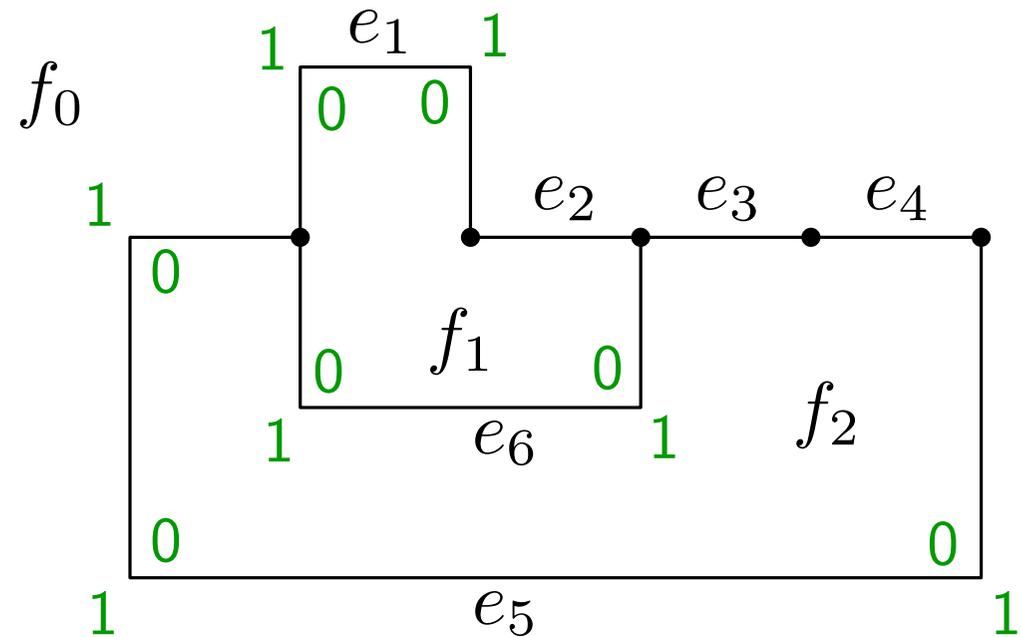
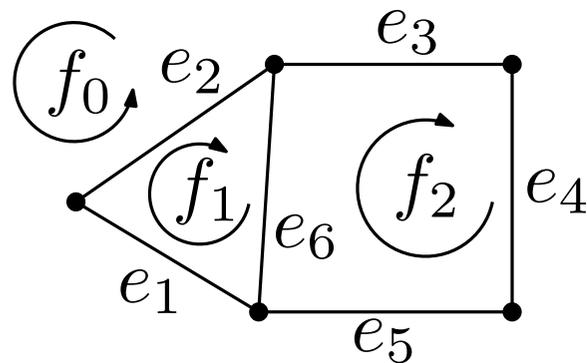
Orthogonale Beschreibung: Beispiel

$$H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2}))$$

$$H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi))$$

$$H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2}))$$

f₀ falsch rum!?

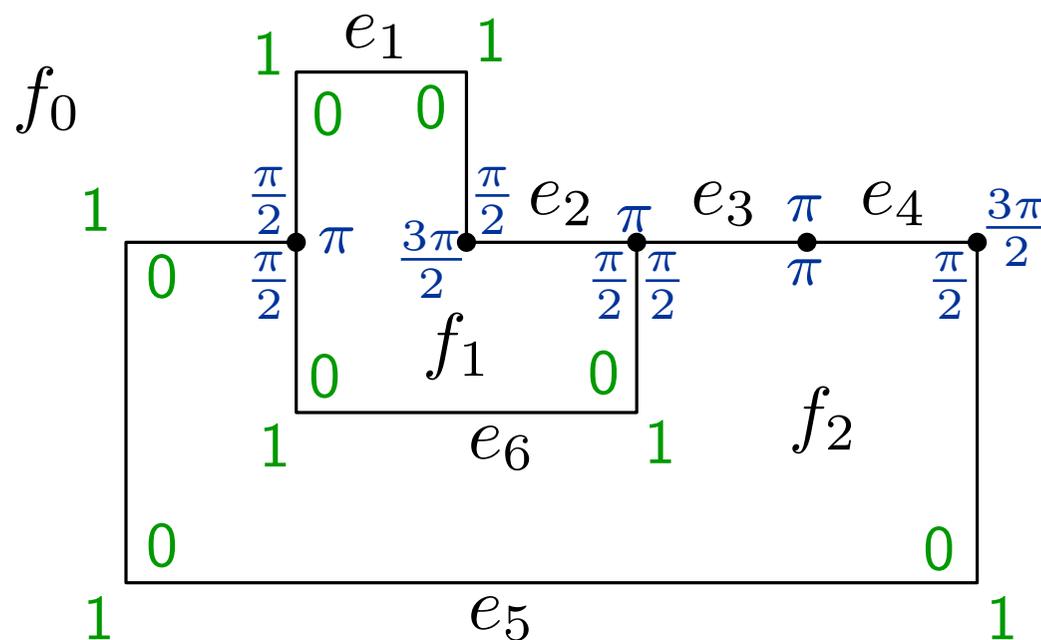
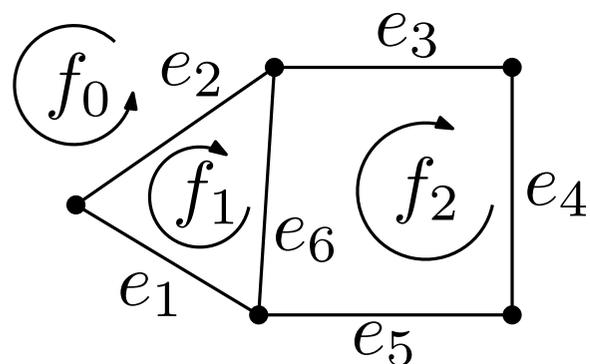


Orthogonale Beschreibung: Beispiel

$$H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2}))$$

$$H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi))$$

$$H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2}))$$

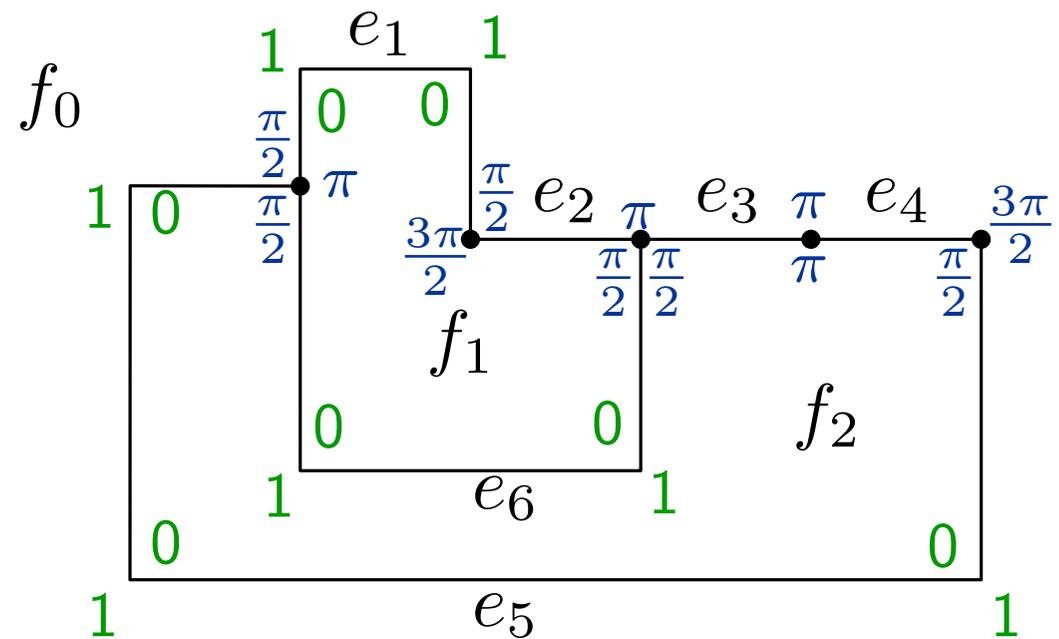
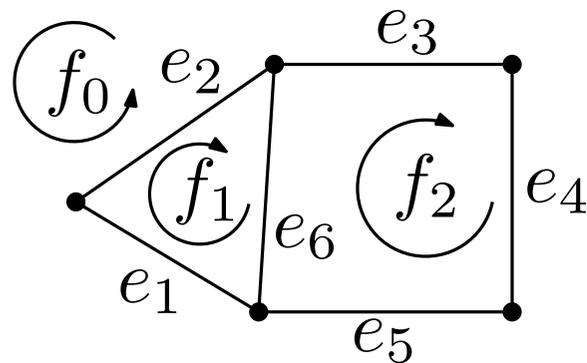


Orthogonale Beschreibung: Beispiel

$$H(f_0) = ((e_1, 11, \frac{\pi}{2}), (e_5, 111, \frac{3\pi}{2}), (e_4, \emptyset, \pi), (e_3, \emptyset, \pi), (e_2, \emptyset, \frac{\pi}{2}))$$

$$H(f_1) = ((e_1, 00, \frac{3\pi}{2}), (e_2, \emptyset, \frac{\pi}{2}), (e_6, 00, \pi))$$

$$H(f_2) = ((e_5, 000, \frac{\pi}{2}), (e_6, 11, \frac{\pi}{2}), (e_3, \emptyset, \pi), (e_4, \emptyset, \frac{\pi}{2}))$$



konkrete Geometrie noch nicht festgelegt!

Korrektheit einer orthogonalen Beschreibung

(H1) $H(G)$ entspricht \mathcal{F}, f_0

(H1) $H(G)$ entspricht \mathcal{F}, f_0

(H2) für gemeinsame Randkante $\{u, v\}$ zweier Facetten f und g mit $((u, v), \delta_1, \alpha_1) \in H(f)$ und $((v, u), \delta_2, \alpha_2) \in H(g)$ gilt δ_1 ist invertierte und umgedrehte Folge δ_2

(H1) $H(G)$ entspricht \mathcal{F}, f_0

(H2) für gemeinsame Randkante $\{u, v\}$ zweier Facetten f und g mit $((u, v), \delta_1, \alpha_1) \in H(f)$ und $((v, u), \delta_2, \alpha_2) \in H(g)$ gilt δ_1 ist invertierte und umgedrehte Folge δ_2

(H3) Sei $|\delta|_0$ (bzw. $|\delta|_1$) die Anzahl Nullen (bzw. Einsen) in δ und $r = (e, \delta, \alpha)$. Für $C(r) := |\delta|_0 - |\delta|_1 + 2 - 2\alpha/\pi$ gilt:
$$\sum_{r \in H(f)} C(r) = 4 \text{ für } f \neq f_0 \text{ und } \sum_{r \in H(f_0)} C(r) = -4$$

- (H1) $H(G)$ entspricht \mathcal{F}, f_0
- (H2) für gemeinsame Randkante $\{u, v\}$ zweier Facetten f und g mit $((u, v), \delta_1, \alpha_1) \in H(f)$ und $((v, u), \delta_2, \alpha_2) \in H(g)$ gilt δ_1 ist invertierte und umgedrehte Folge δ_2
- (H3) Sei $|\delta|_0$ (bzw. $|\delta|_1$) die Anzahl Nullen (bzw. Einsen) in δ und $r = (e, \delta, \alpha)$. Für $C(r) := |\delta|_0 - |\delta|_1 + 2 - 2\alpha/\pi$ gilt:
$$\sum_{r \in H(f)} C(r) = 4 \text{ für } f \neq f_0 \text{ und } \sum_{r \in H(f_0)} C(r) = -4$$
- (H4) Für jeden Knoten v ist die Summe der anliegenden Winkel gleich 2π

Problem kombinatorische Knickminimierung

Geg: ■ Graph $G = (V, E)$ mit Maximalgrad 4
■ kombinatorische Einbettung \mathcal{F} und äußere Facette f_0

Ges: einbettungsäquivalente **orthogonale Beschreibung** $H(G)$
mit minimaler Knickanzahl

Problem kombinatorische Knickminimierung

Geg: ■ Graph $G = (V, E)$ mit Maximalgrad 4
■ kombinatorische Einbettung \mathcal{F} und äußere Facette f_0

Ges: einbettungsäquivalente **orthogonale Beschreibung** $H(G)$
mit minimaler Knickanzahl

Idee: erstelle ein Flussnetzwerk mit

- eine Flusseinheit entspricht einem $\pi/2$ Winkel
- Fluss von Knoten zu Facetten entspricht Knotenwinkel
- Fluss zwischen Facetten entspricht Kantenknicken

Erinnerung: s - t Flussnetzwerk

Flussnetzwerk $(D = (V, A); s, t; c)$ mit

- gerichtetem Graph $D = (V, A)$
- Kantenkapazitäten $c : A \rightarrow \mathbb{R}_0^+$
- Quelle $s \in V$, Senke $t \in V$

Eine Abbildung $X : A \rightarrow \mathbb{R}_0^+$ heißt s - t -**Fluss**, falls gilt:

$$0 \leq X(i, j) \leq c(i, j) \quad \forall (i, j) \in A \quad (1)$$

$$\sum_{(i,j) \in A} X(i, j) - \sum_{(j,i) \in A} X(j, i) = 0 \quad \forall i \in V \setminus \{s, t\} \quad (2)$$

Erinnerung: Allgemeines Flussnetzwerk

Flussnetzwerk $(D = (V, A); \ell; u; b)$ mit

- gerichtetem Graph $D = (V, A)$
- untere Kantenkapazitäten $\ell : A \rightarrow \mathbb{R}_0^+$
- obere Kantenkapazitäten $u : A \rightarrow \mathbb{R}_0^+$
- Knotenbewertung $b : V \rightarrow \mathbb{R}$ mit $\sum_{i \in V} b(i) = 0$

Eine Abbildung $X : A \rightarrow \mathbb{R}_0^+$ heißt **Fluss**, falls gilt:

$$\ell(i, j) \leq X(i, j) \leq u(i, j) \quad \forall (i, j) \in A \quad (3)$$

$$\sum_{(i, j) \in A} X(i, j) - \sum_{(j, i) \in A} X(j, i) = b(i) \quad \forall i \in V \quad (4)$$

(A) Gültiger Fluss:

Finde einen gültigen Fluss $X : A \rightarrow \mathbb{R}_0^+$, d.h.

- Kapazitäten $l(e), u(e)$ werden respektiert
- Bedarf $b(v)$ wird genau erfüllt

(A) Gültiger Fluss:

Finde einen gültigen Fluss $X : A \rightarrow \mathbb{R}_0^+$, d.h.

- Kapazitäten $l(e), u(e)$ werden respektiert
- Bedarf $b(v)$ wird genau erfüllt

Zusätzlich gegeben: Kostenfunktion $\text{cost} : A \rightarrow \mathbb{R}_0^+$

Def: $\text{cost}(X) := \sum_{(i,j) \in A} \text{cost}(i, j) \cdot X(i, j)$

(A) Gültiger Fluss:

Finde einen gültigen Fluss $X : A \rightarrow \mathbb{R}_0^+$, d.h.

- Kapazitäten $l(e), u(e)$ werden respektiert
- Bedarf $b(v)$ wird genau erfüllt

Zusätzlich gegeben: Kostenfunktion $\text{cost} : A \rightarrow \mathbb{R}_0^+$

Def: $\text{cost}(X) := \sum_{(i,j) \in A} \text{cost}(i, j) \cdot X(i, j)$

(B) Minimalkostenfluss

Finde einen gültigen Fluss $X : A \rightarrow \mathbb{R}_0^+$, der $\text{cost}(X)$ minimiert (unter allen gültigen Flüssen)

Flussnetzwerk zur Knickminimierung

Flussnetzwerk $N(G) = ((V \cup \mathcal{F}, A); \ell; u; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ inzident zu } f\} \cup$
 $\{(f, g) \in \mathcal{F} \times \mathcal{F} \mid \forall e \in E \text{ mit } e \text{ trennt } f, g\}$

Flussnetzwerk zur Knickminimierung

Flussnetzwerk $N(G) = ((V \cup \mathcal{F}, A); \ell; u; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ inzident zu } f\} \cup$
 $\{(f, g) \in \mathcal{F} \times \mathcal{F} \mid \forall e \in E \text{ mit } e \text{ trennt } f, g\}$
- $b(v) = 4 \quad \forall v \in V$

Flussnetzwerk zur Knickminimierung

Flussnetzwerk $N(G) = ((V \cup \mathcal{F}, A); \ell; u; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ inzident zu } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid \forall e \in E \text{ mit } e \text{ trennt } f, g\}$
- $b(v) = 4 \quad \forall v \in V$
- $b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\}$
- $b(f_0) = -2(d_G(f_0) + 2)$

Flussnetzwerk zur Knickminimierung

Flussnetzwerk $N(G) = ((V \cup \mathcal{F}, A); \ell; u; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ inzident zu } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid \forall e \in E \text{ mit } e \text{ trennt } f, g\}$
 - $b(v) = 4 \quad \forall v \in V$
 - $b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\}$
 - $b(f_0) = -2(d_G(f_0) + 2)$
- } $\Rightarrow \sum_w b(w) \stackrel{?}{=} 0$

Flussnetzwerk zur Knickminimierung

Flussnetzwerk $N(G) = ((V \cup \mathcal{F}, A); \ell; u; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ inzident zu } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid \forall e \in E \text{ mit } e \text{ trennt } f, g\}$
 - $b(v) = 4 \quad \forall v \in V$
 - $b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\}$
 - $b(f_0) = -2(d_G(f_0) + 2)$
- } $\Rightarrow \sum_w b(w) = 0$
(Euler)

Flussnetzwerk zur Knickminimierung

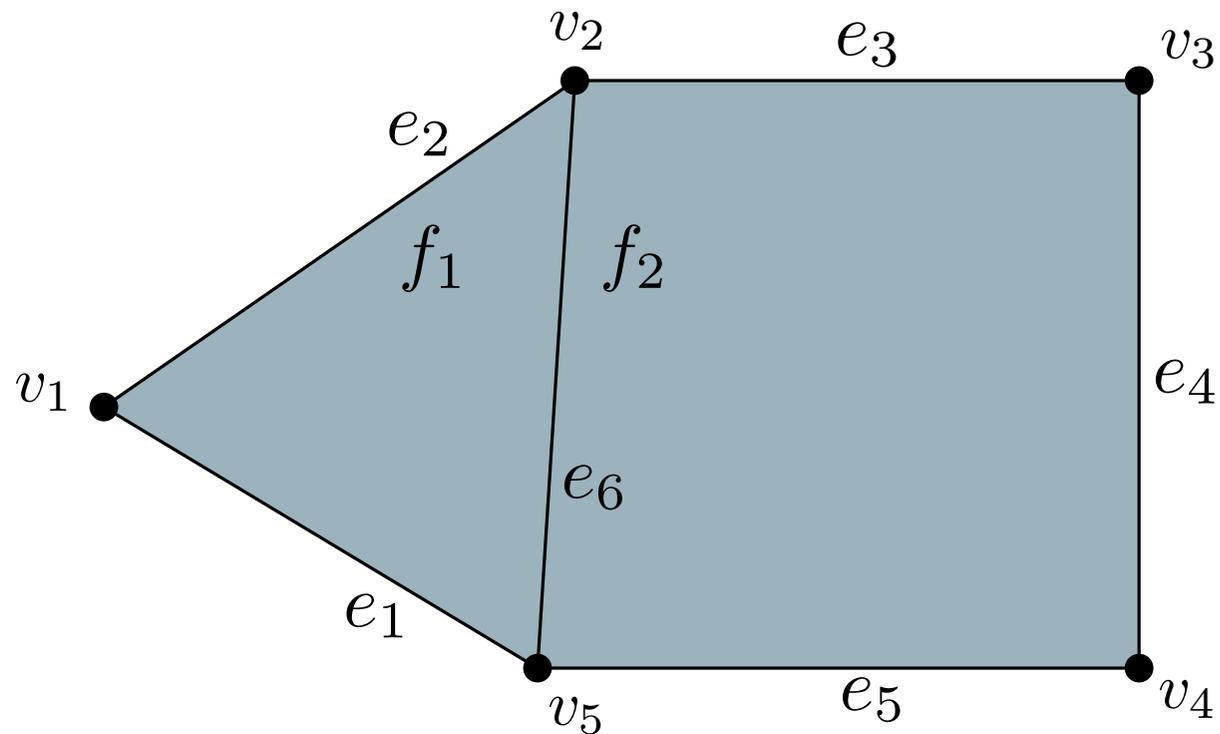
Flussnetzwerk $N(G) = ((V \cup \mathcal{F}, A); \ell; u; b; \text{cost})$

- $A = \{(v, f) \in V \times \mathcal{F} \mid v \text{ inzident zu } f\} \cup \{(f, g) \in \mathcal{F} \times \mathcal{F} \mid \forall e \in E \text{ mit } e \text{ trennt } f, g\}$
 - $b(v) = 4 \quad \forall v \in V$
 - $b(f) = -2(d_G(f) - 2) \quad \forall f \in \mathcal{F} \setminus \{f_0\}$
 - $b(f_0) = -2(d_G(f_0) + 2)$
- $\left. \vphantom{\begin{matrix} \bullet \\ \bullet \\ \bullet \end{matrix}} \right\} \Rightarrow \sum_w b(w) = 0$
(Euler)

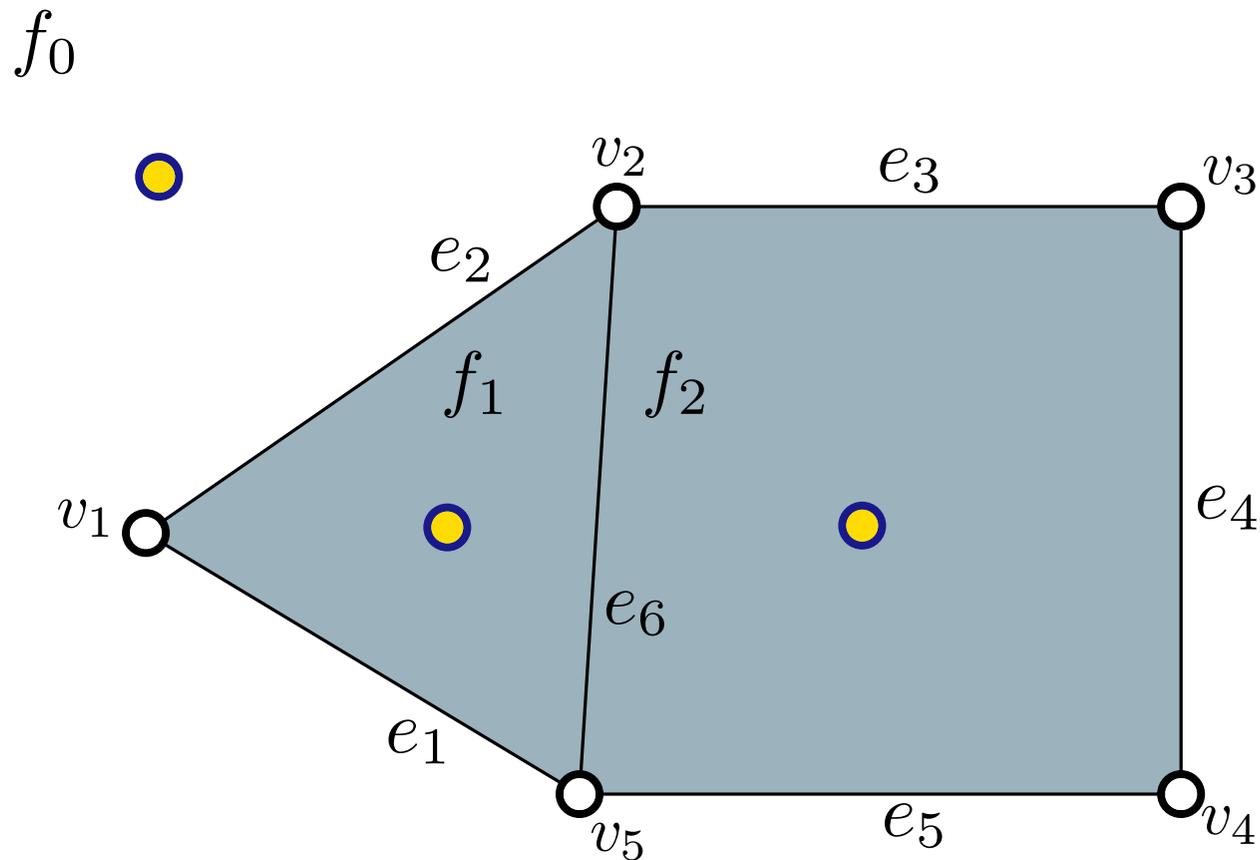
$$\begin{array}{ll} \forall (f, g) \in A, f, g \in \mathcal{F} & \ell(f, g) := 0 \leq X(f, g) \leq \infty =: u(f, g) \\ \forall (v, f) \in A, v \in V, f \in \mathcal{F} & \ell(v, f) := 1 \leq X(v, f) \leq 4 =: u(v, f) \\ \forall i \in V \cup \mathcal{F} & \sum_{(i, j) \in A} X(i, j) - \sum_{(j, i) \in A} X(j, i) = b(i) \end{array}$$

Beispiel Flussnetzwerk

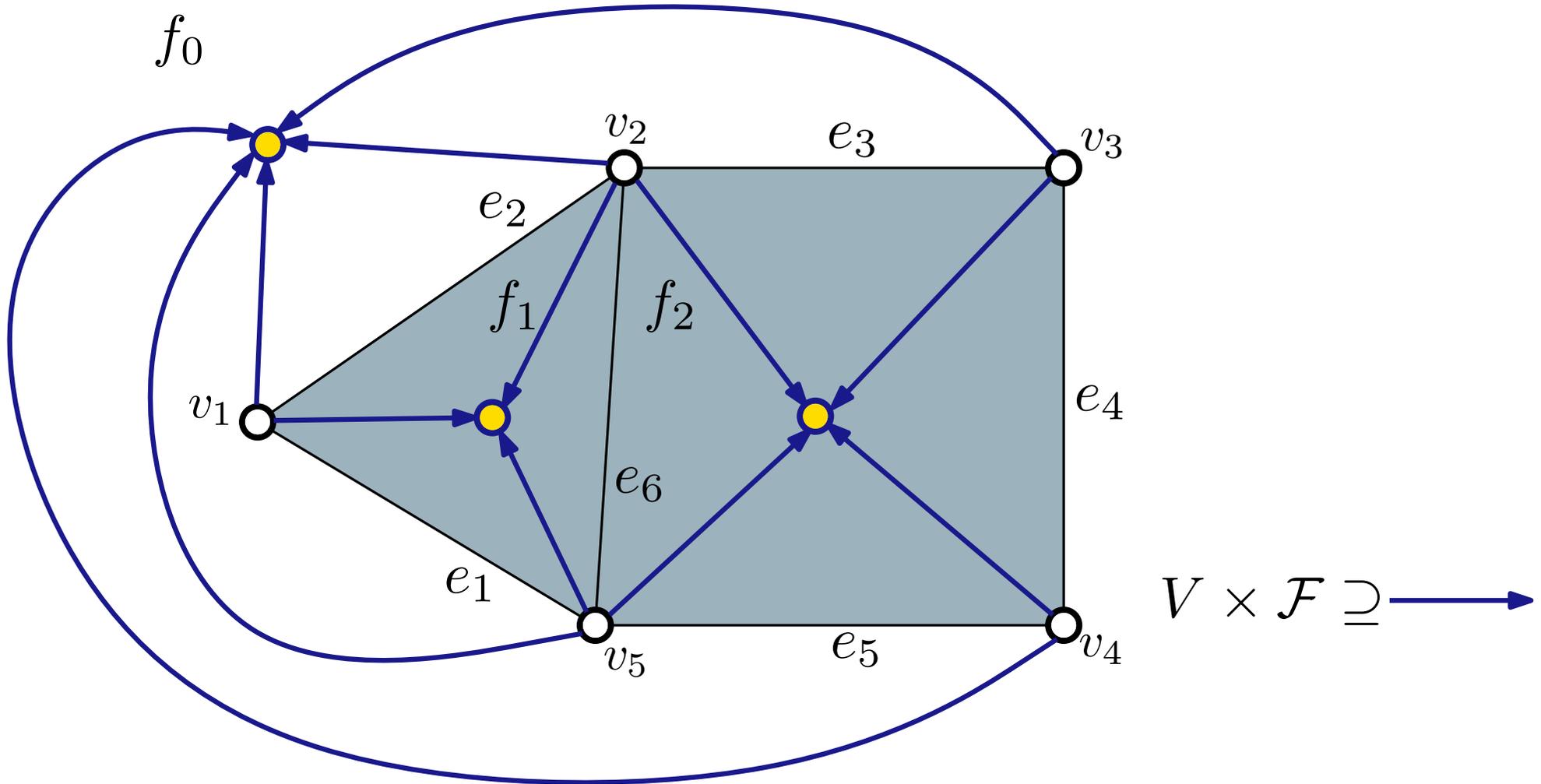
f_0



Beispiel Flussnetzwerk



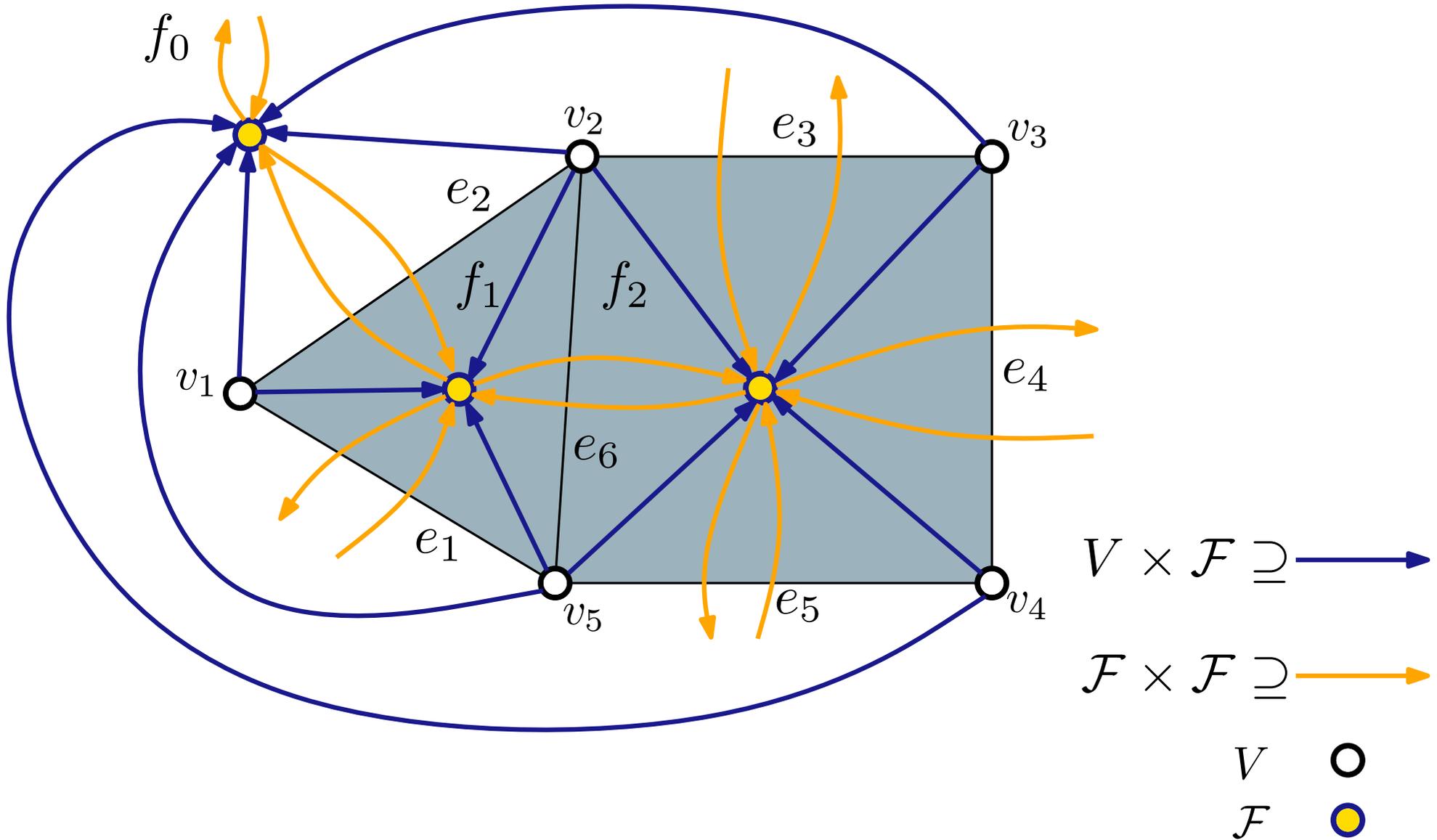
Beispiel Flussnetzwerk



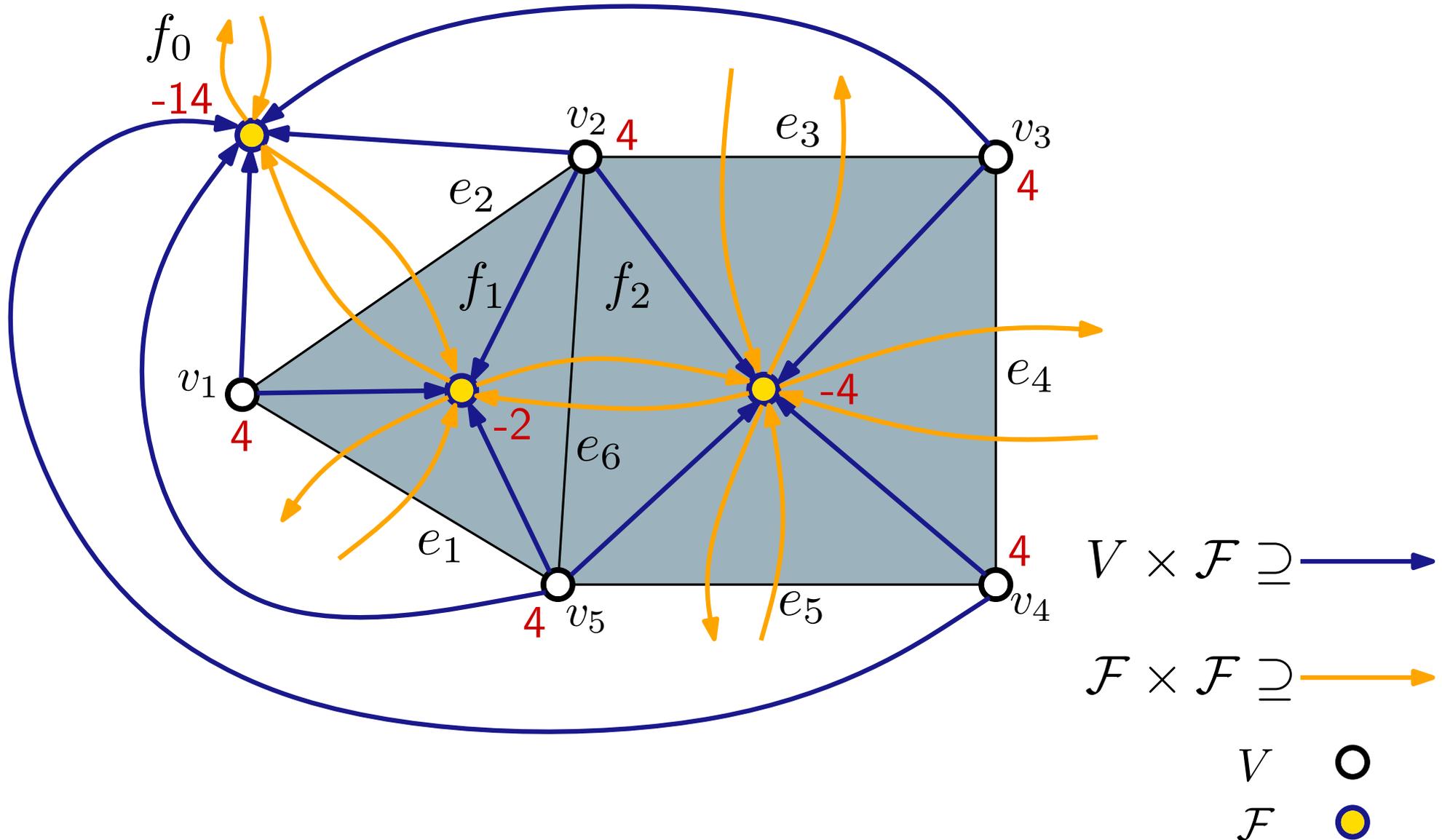
$V \times \mathcal{F} \supseteq \longrightarrow$

V ○
 \mathcal{F} ●

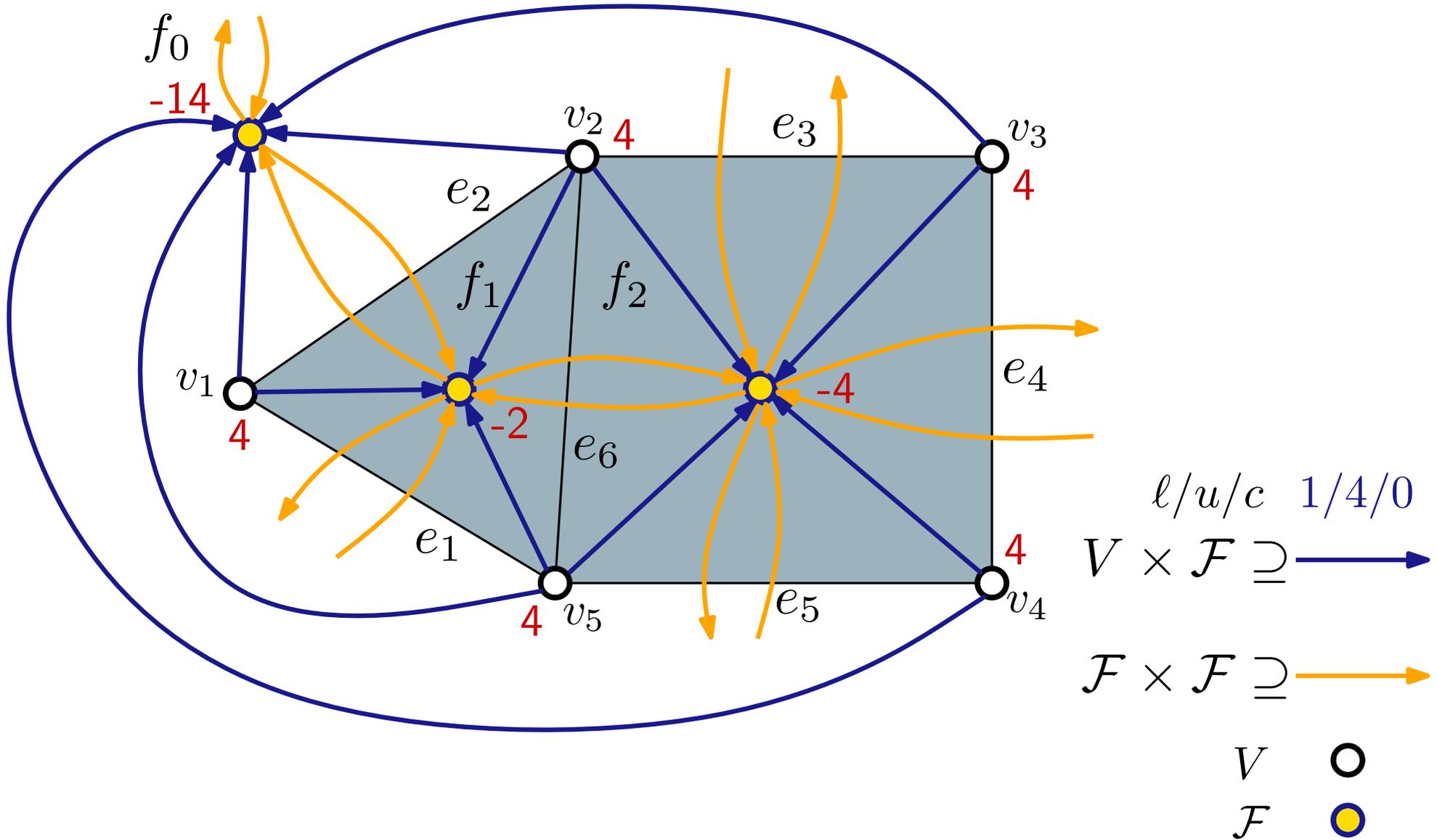
Beispiel Flussnetzwerk



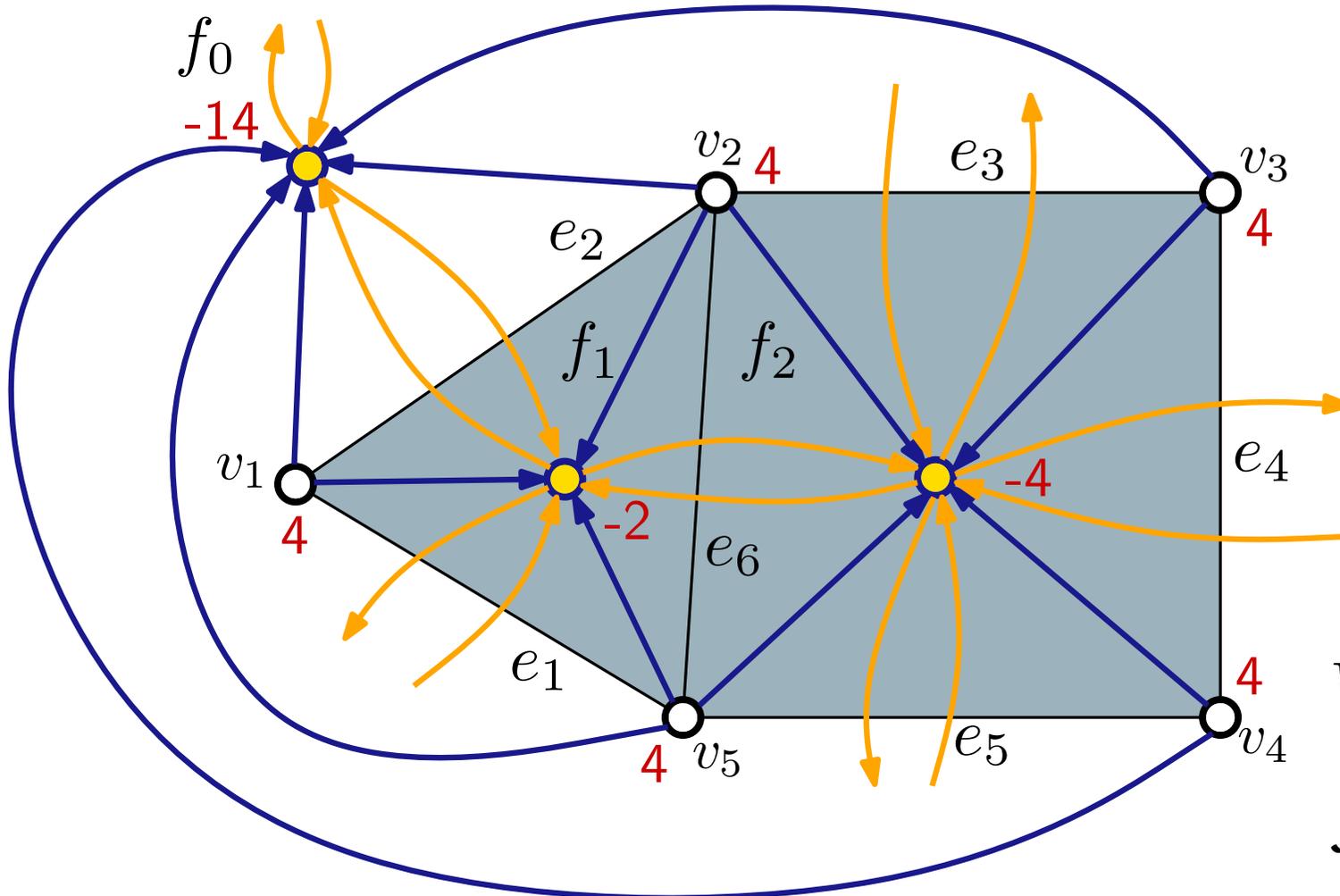
Beispiel Flussnetzwerk



Beispiel Flussnetzwerk



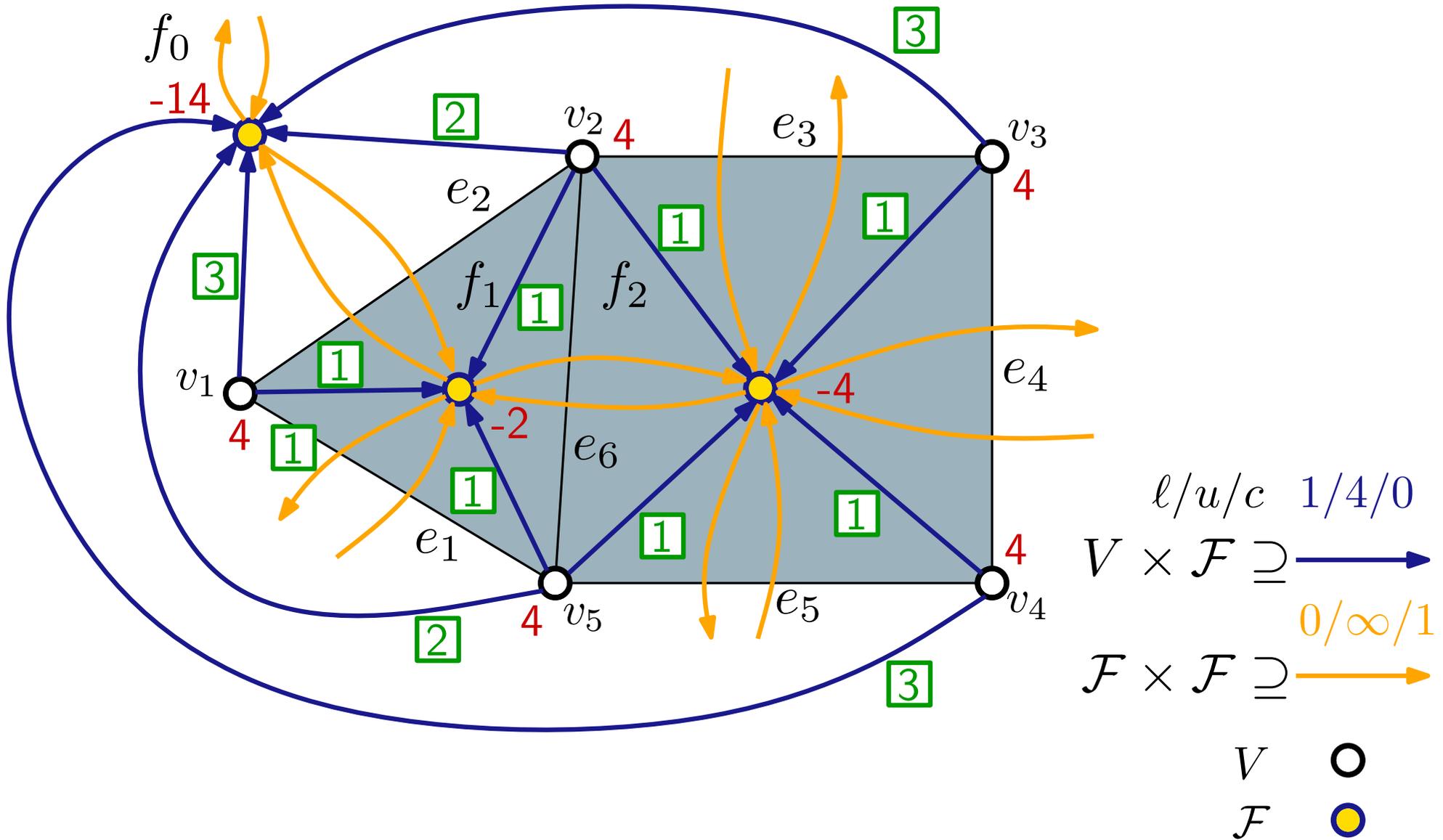
Beispiel Flussnetzwerk



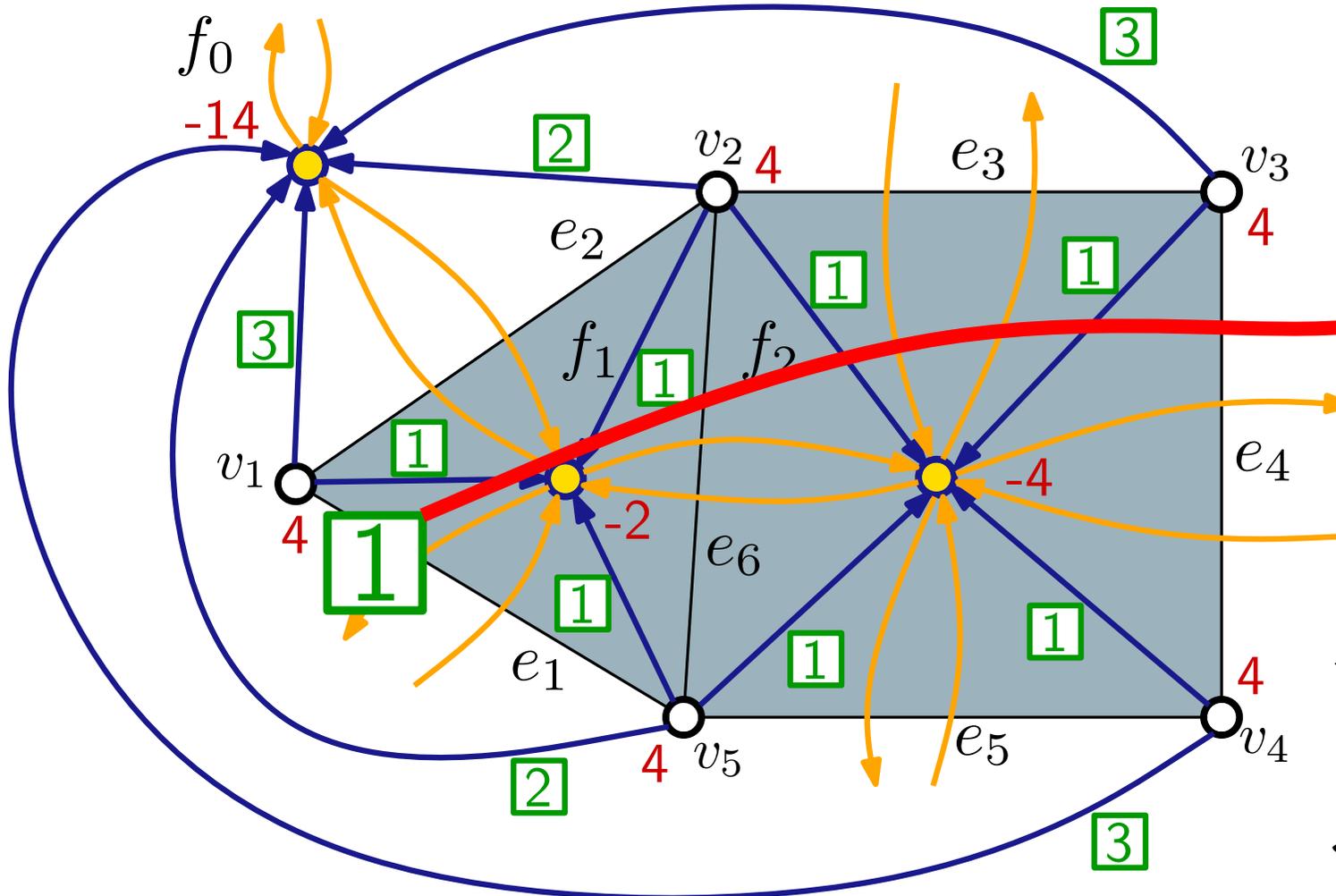
$l/u/c$ 1/4/0
 $V \times \mathcal{F} \supseteq$ 
 $\mathcal{F} \times \mathcal{F} \supseteq$  0/ ∞ /1

V 
 \mathcal{F} 

Beispiel Flussnetzwerk



Beispiel Flussnetzwerk



cost = 1
 Knick!
 (nach außen)

$l/u/c$ 1/4/0
 $V \times \mathcal{F} \supseteq$

$0/\infty/1$
 $\mathcal{F} \times \mathcal{F} \supseteq$

V

\mathcal{F}

Satz 1: Zu einem planar eingebetteten Graphen (G, \mathcal{F}, f_0) existiert genau dann eine zulässige orthogonale Beschreibung $H(G)$ mit k Knicken, wenn es im Flussnetzwerk $N(G)$ einen Fluss x mit Kosten k gibt.

Satz 1: Zu einem planar eingebetteten Graphen (G, \mathcal{F}, f_0) existiert genau dann eine zulässige orthogonale Beschreibung $H(G)$ mit k Knicken, wenn es im Flussnetzwerk $N(G)$ einen Fluss x mit Kosten k gibt.

Beweis:

\Leftarrow Geg: Fluss x in $N(G)$ mit Kosten k

Ges: orthogonale Beschreibung $H(G)$ mit k Knicken

Satz 1: Zu einem planar eingebetteten Graphen (G, \mathcal{F}, f_0) existiert genau dann eine zulässige orthogonale Beschreibung $H(G)$ mit k Knicken, wenn es im Flussnetzwerk $N(G)$ einen Fluss x mit Kosten k gibt.

Beweis:

\Leftarrow Geg: Fluss x in $N(G)$ mit Kosten k

Ges: orthogonale Beschreibung $H(G)$ mit k Knicken

- transformiere Fluss in orthogonale Beschreibung
- zeige Eigenschaften (H1)–(H4)

Satz 1: Zu einem planar eingebetteten Graphen (G, \mathcal{F}, f_0) existiert genau dann eine zulässige orthogonale Beschreibung $H(G)$ mit k Knicken, wenn es im Flussnetzwerk $N(G)$ einen Fluss x mit Kosten k gibt.

Beweis:

\Leftarrow Geg: Fluss x in $N(G)$ mit Kosten k

Ges: orthogonale Beschreibung $H(G)$ mit k Knicken

- transformiere Fluss in orthogonale Beschreibung
- zeige Eigenschaften (H1)–(H4)

\Rightarrow Geg: orthogonale Beschreibung $H(G)$ mit k Knicken

Ges: Fluss x in $N(G)$ mit Kosten k

Satz 1: Zu einem planar eingebetteten Graphen (G, \mathcal{F}, f_0) existiert genau dann eine zulässige orthogonale Beschreibung $H(G)$ mit k Knicken, wenn es im Flussnetzwerk $N(G)$ einen Fluss x mit Kosten k gibt.

Beweis:

\Leftarrow Geg: Fluss x in $N(G)$ mit Kosten k

Ges: orthogonale Beschreibung $H(G)$ mit k Knicken

- transformiere Fluss in orthogonale Beschreibung
- zeige Eigenschaften (H1)–(H4)

\Rightarrow Geg: orthogonale Beschreibung $H(G)$ mit k Knicken

Ges: Fluss x in $N(G)$ mit Kosten k

- definiere Abbildung $x : A \rightarrow \mathbb{R}_0^+$
- zeige x erfüllt Flussbedingungen und hat Kosten k

- Aus Satz 1 folgt, dass die orthogonale Knickminimierung für eingebettete planare Graphen durch Lösen eines Min-Cost-Flow Problems durchgeführt werden kann.

- Aus Satz 1 folgt, dass die orthogonale Knickminimierung für eingebettete planare Graphen durch Lösen eines Min-Cost-Flow Problems durchgeführt werden kann.
- Dieses spezielle Flussproblem lässt sich unter Ausnutzung der Planarität von $N(G)$ in $O(n^{3/2})$ Zeit lösen.

[Cornelsen, Karrenbauer GD 2011]

- Aus Satz 1 folgt, dass die orthogonale Knickminimierung für eingebettete planare Graphen durch Lösen eines Min-Cost-Flow Problems durchgeführt werden kann.
- Dieses spezielle Flussproblem lässt sich unter Ausnutzung der Planarität von $N(G)$ in $O(n^{3/2})$ Zeit lösen.

[Cornelsen, Karrenbauer GD 2011]

- Die Knickminimierung ohne gegebene kombinatorische Einbettung ist NP-schwer.

[Garg, Tamassia SIAM J. Comput. 2001]

- Aus Satz 1 folgt, dass die orthogonale Knickminimierung für eingebettete planare Graphen durch Lösen eines Min-Cost-Flow Problems durchgeführt werden kann.
- Dieses spezielle Flussproblem lässt sich unter Ausnutzung der Planarität von $N(G)$ in $O(n^{3/2})$ Zeit lösen.

[Cornelsen, Karrenbauer GD 2011]

- Die Knickminimierung ohne gegebene kombinatorische Einbettung ist NP-schwer. [Garg, Tamassia SIAM J. Comput. 2001]
- Das Entscheidungsproblem, ob ein Graph eine orthogonale Zeichnung mit höchstens $f(e)$ Knicken pro Kante e besitzt ist polynomiell lösbar, falls $f(e) \geq 1$ für alle $e \in E$.

[Bläsius et al. GD 2010]

(Planare) Orthogonale Zeichnungen

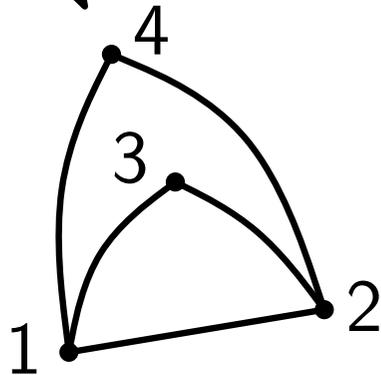
Dreistufiger Ansatz: *Topology – Shape – Metrics*

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

Kreuzungsminimierung

kombinatorische
Einbettung/
Planarisierung

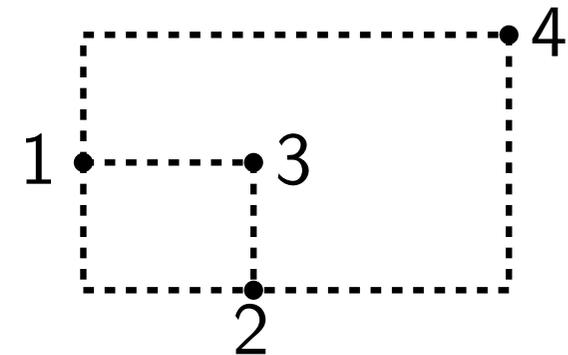
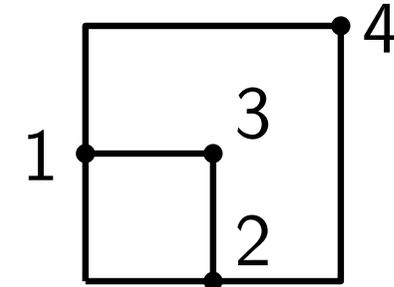


Knickminimierung

orthogonale
Beschreibung

planare
Einbettung

Flächen-
minimierung



(Planare) Orthogonale Zeichnungen

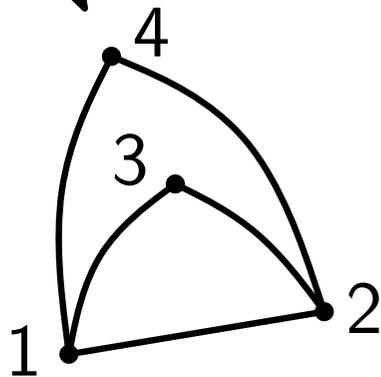
Dreistufiger Ansatz: *Topology – Shape – Metrics*

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

Kreuzungsminimierung

kombinatorische
Einbettung/
Planarisierung

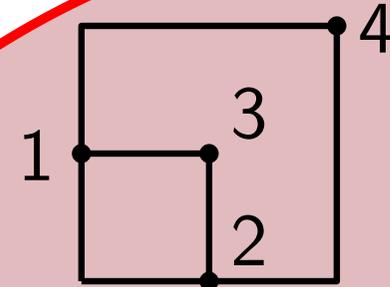
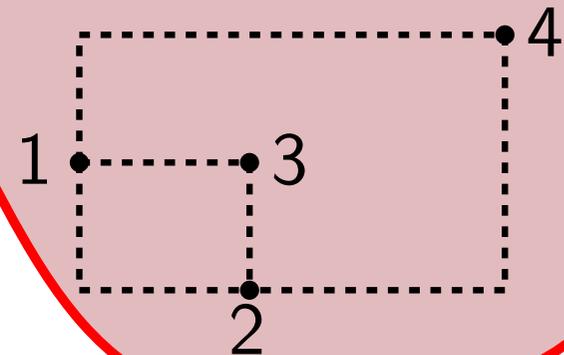


Knickminimierung

orthogonale
Beschreibung

planare
Einbettung

Flächen-
minimierung



Problem Kompaktierung

Geg: ■ planarer Graph $G = (V, E)$ mit Maximalgrad 4
■ orthogonale Beschreibung $H(G)$

Ges: kompaktes orthogonales Layout von G , das $H(G)$ realisiert

Problem Kompaktierung

Geg: ■ planarer Graph $G = (V, E)$ mit Maximalgrad 4
■ orthogonale Beschreibung $H(G)$

Ges: kompaktes orthogonales Layout von G , das $H(G)$ realisiert

Spezialfall: alle Facetten sind Rechtecke

→ Garantien möglich ■ minimale Gesamtkantenlänge
■ minimale Fläche

Problem Kompaktierung

Geg: ■ planarer Graph $G = (V, E)$ mit Maximalgrad 4
■ orthogonale Beschreibung $H(G)$

Ges: kompaktes orthogonales Layout von G , das $H(G)$ realisiert

Spezialfall: alle Facetten sind Rechtecke

→ Garantien möglich ■ minimale Gesamtkantenlänge
■ minimale Fläche

Eigenschaften:

- Knicke höchstens als Ecken der äußeren Facette
- gegenüberliegende Seiten einer Facette gleich lang

Problem Kompaktierung

- Geg:
- planarer Graph $G = (V, E)$ mit Maximalgrad 4
 - orthogonale Beschreibung $H(G)$

Ges: kompaktes orthogonales Layout von G , das $H(G)$ realisiert

Spezialfall: alle Facetten sind Rechtecke

- Garantien möglich
- minimale Gesamtkantenlänge
 - minimale Fläche

Eigenschaften:

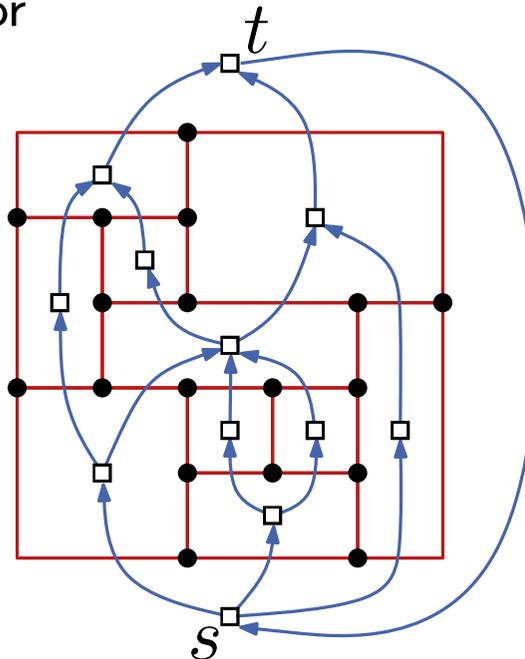
- Knicke höchstens als Ecken der äußeren Facette
- gegenüberliegende Seiten einer Facette gleich lang

Erstelle ein Flussnetzwerk zur
(horizontalen) Kompaktierung.

Flussnetzwerk Längenzuweisung

Def: Flussnetzwerk $N_{\text{hor}} = ((W_{\text{hor}}, A_{\text{hor}}); \ell; u; b; \text{cost})$

- $W_{\text{hor}} = \mathcal{F} \setminus \{f_0\} \cup \{s, t\}$
- $A_{\text{hor}} = \{(f, g) \mid f, g \text{ besitzen gemeinsames horizontales Kantensegment und } f \text{ liegt unterhalb von } g\} \cup \{(t, s)\}$
- $\ell(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $u(a) = \infty \quad \forall a \in A_{\text{hor}}$
- $\text{cost}(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $b(f) = 0 \quad \forall f \in W_{\text{hor}}$

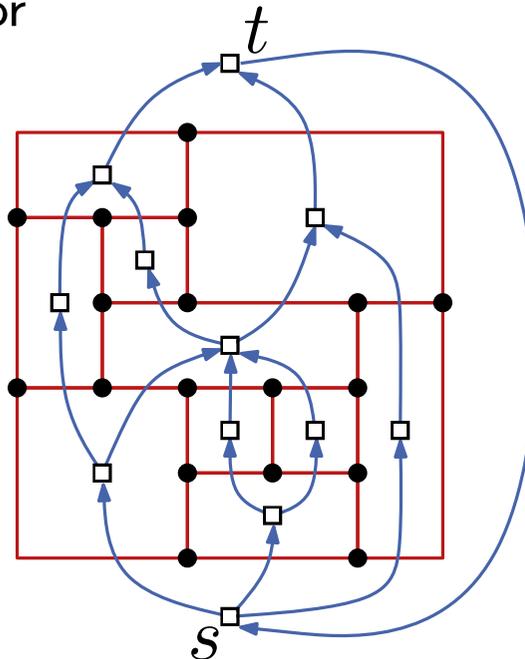


Flussnetzwerk Längenzuweisung

Def: Flussnetzwerk $N_{\text{hor}} = ((W_{\text{hor}}, A_{\text{hor}}); \ell; u; b; \text{cost})$

- $W_{\text{hor}} = \mathcal{F} \setminus \{f_0\} \cup \{s, t\}$
- $A_{\text{hor}} = \{(f, g) \mid f, g \text{ besitzen gemeinsames horizontales Kantensegment und } f \text{ liegt unterhalb von } g\} \cup \{(t, s)\}$
- $\ell(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $u(a) = \infty \quad \forall a \in A_{\text{hor}}$
- $\text{cost}(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $b(f) = 0 \quad \forall f \in W_{\text{hor}}$

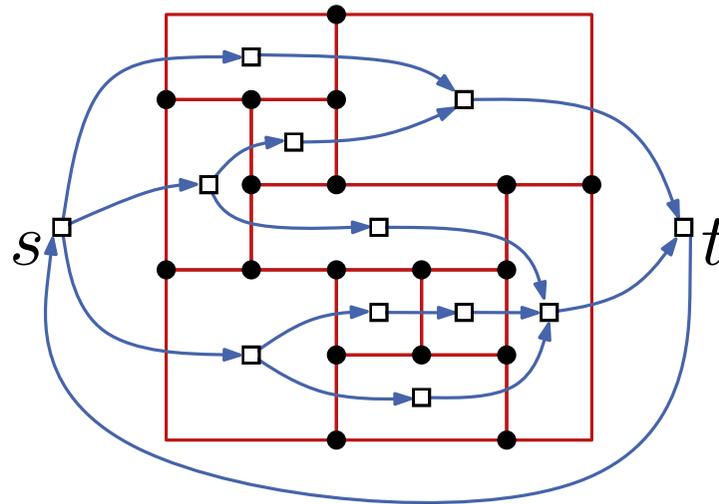
s und t repräsentieren untere und obere Hälfte von f_0

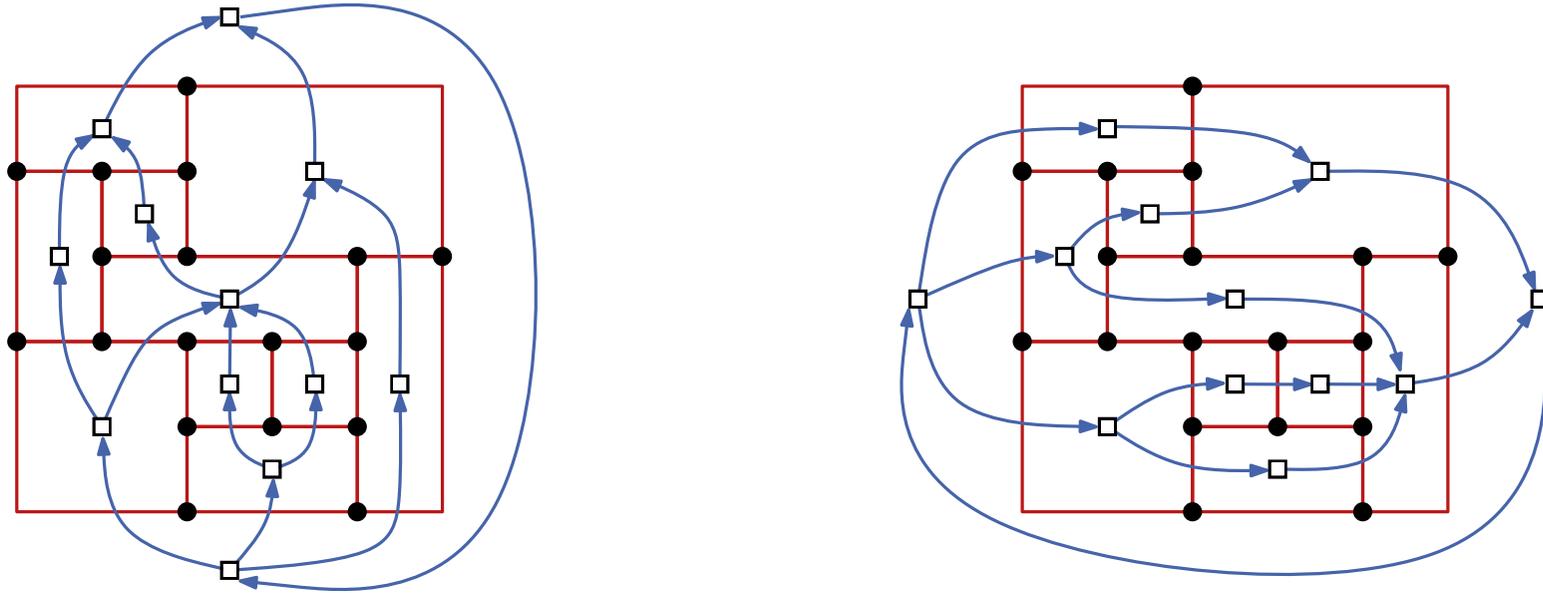


Flussnetzwerk Längenzuweisung

Def: Flussnetzwerk $N_{\text{ver}} = ((W_{\text{ver}}, A_{\text{ver}}); \ell; u; b; \text{cost})$

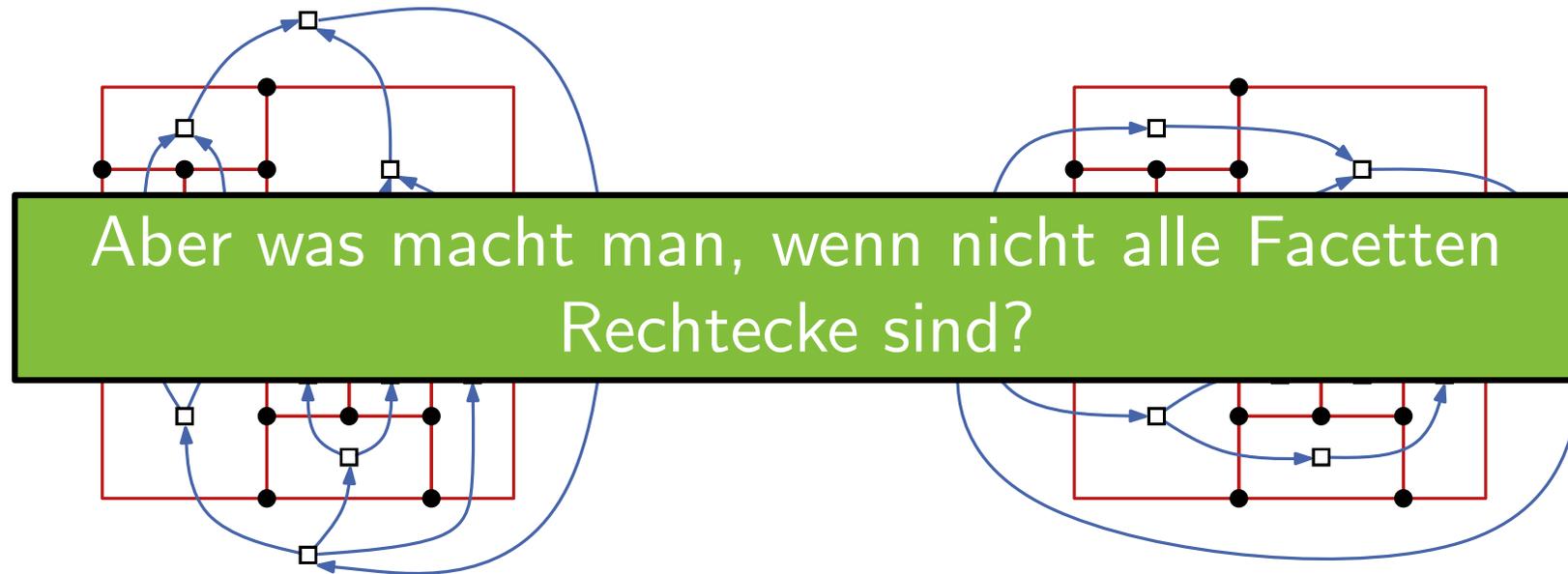
- $W_{\text{ver}} = \mathcal{F} \setminus \{f_0\} \cup \{s, t\}$
- $A_{\text{ver}} = \{(f, g) \mid f, g \text{ besitzen gemeinsames vertikales Kantensegment und } f \text{ liegt links von } g\} \cup \{(t, s)\}$
- $\ell(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $u(a) = \infty \quad \forall a \in A_{\text{hor}}$
- $\text{cost}(a) = 1 \quad \forall a \in A_{\text{hor}}$
- $b(f) = 0 \quad \forall f \in W_{\text{hor}}$





Satz 2: Ganzzahlige Flüsse x_{hor} und x_{ver} in N_{hor} und N_{ver} mit minimalen Kosten liefern:

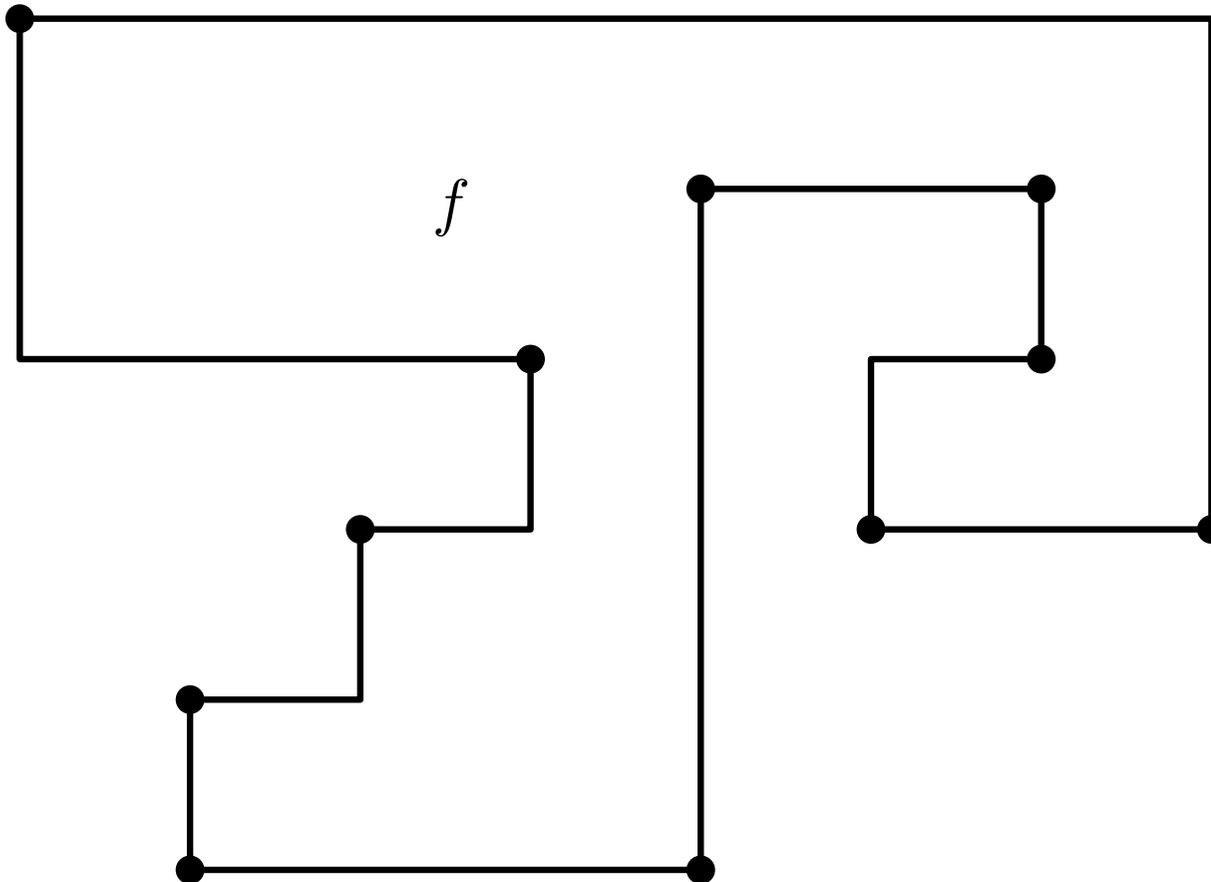
- zugeh. Seitenlängen induzieren gültiges Layout
- $|x_{\text{hor}}(t, s)|$ und $|x_{\text{ver}}(t, s)|$ entsprechen minimaler Breite und Höhe des Layouts
- $\sum_{a \in A_{\text{hor}}} x_{\text{hor}}(e) \cdot \text{cost}(a) + \sum_{e \in A_{\text{ver}}} x_{\text{ver}}(a) \cdot \text{cost}(a)$ entsprechen minimaler Gesamtkantenlänge



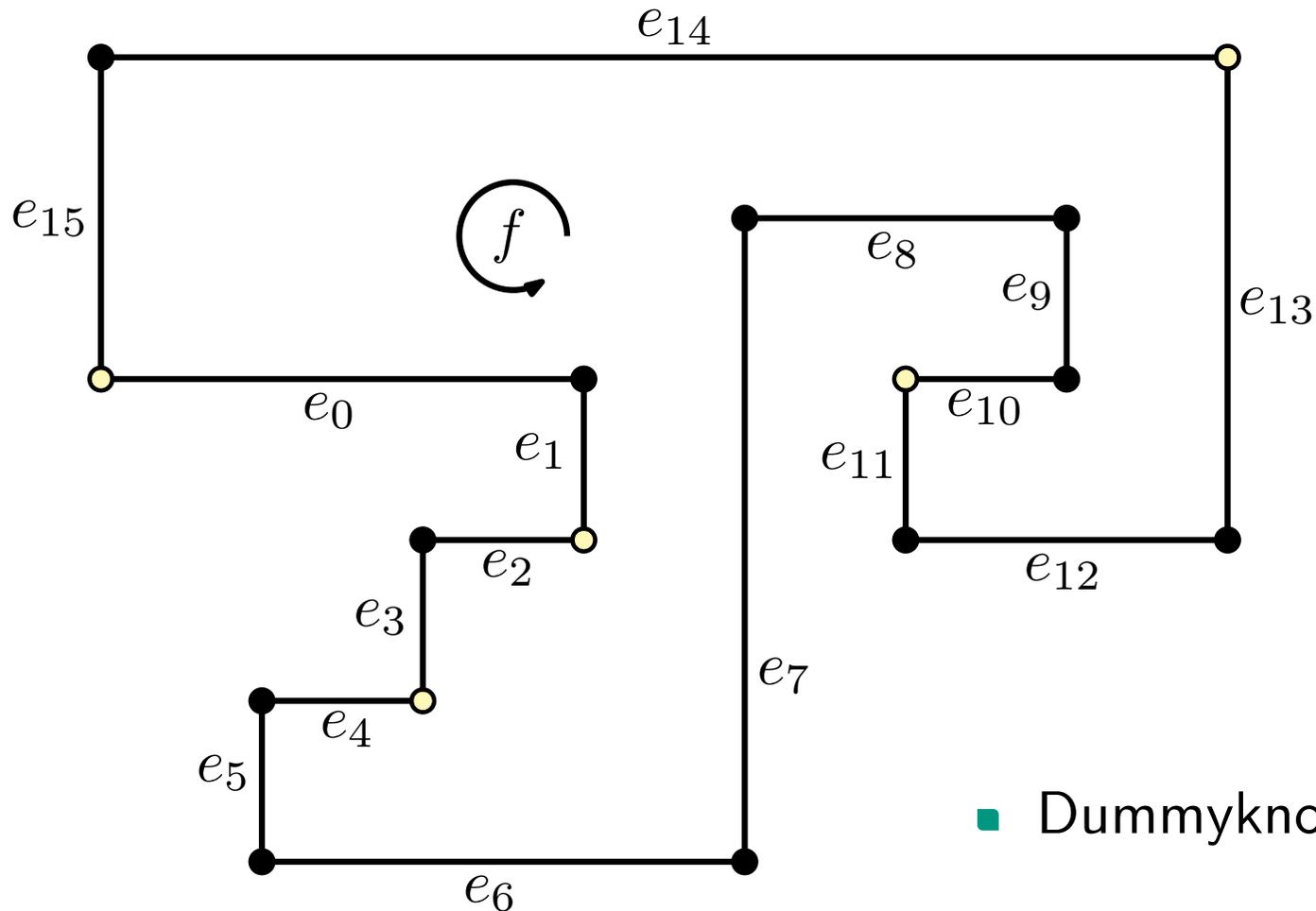
Satz 2: Ganzzahlige Flüsse x_{hor} und x_{ver} in N_{hor} und N_{ver} mit minimalen Kosten liefern:

- zugeh. Seitenlängen induzieren gültiges Layout
- $|x_{\text{hor}}(t, s)|$ und $|x_{\text{ver}}(t, s)|$ entsprechen minimaler Breite und Höhe des Layouts
- $\sum_{a \in A_{\text{hor}}} x_{\text{hor}}(e) \cdot \text{cost}(a) + \sum_{e \in A_{\text{ver}}} x_{\text{ver}}(a) \cdot \text{cost}(a)$ entsprechen minimaler Gesamtkantenlänge

Verfeinerung von (G, H) – innere Facette

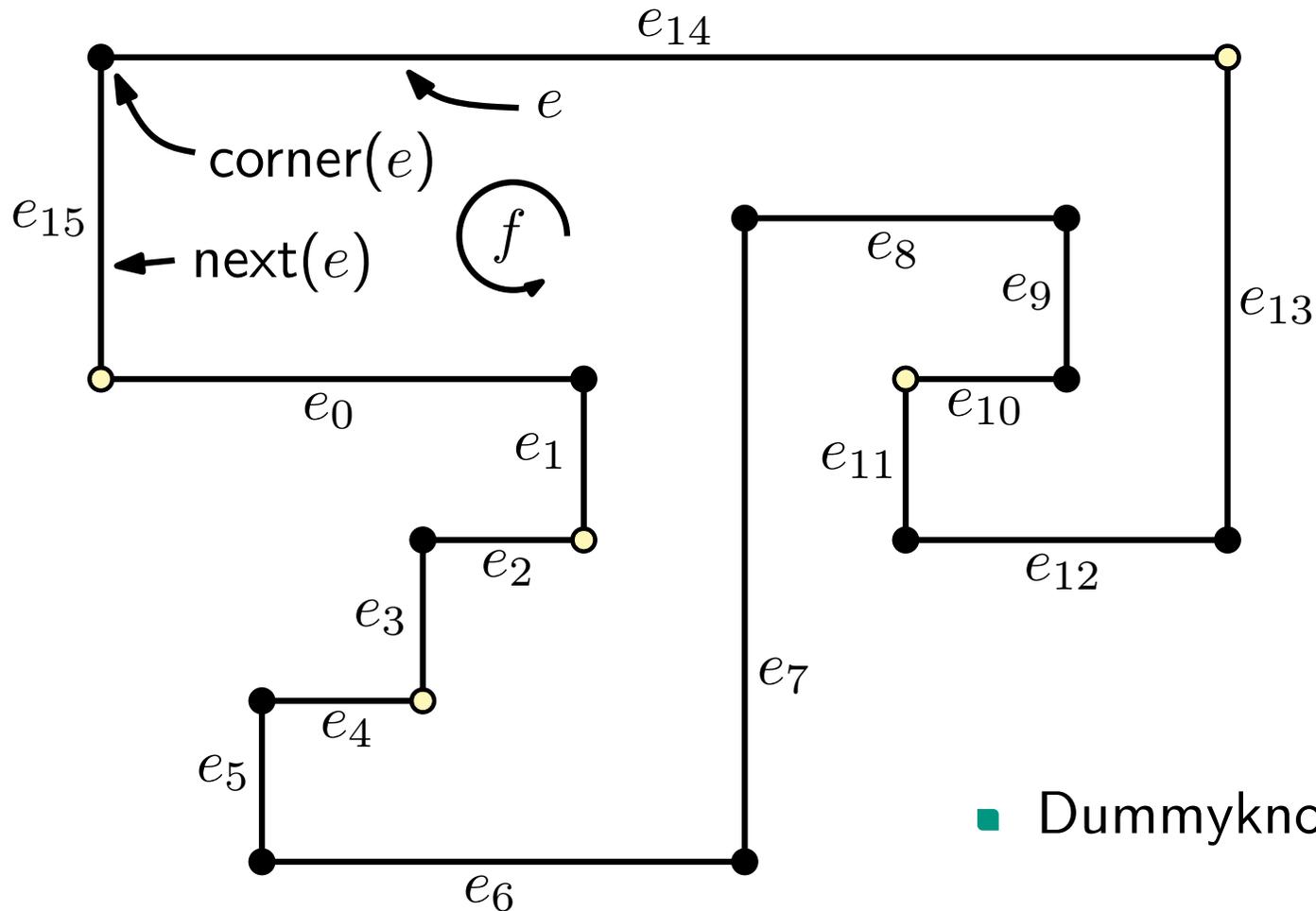


Verfeinerung von (G, H) – innere Facette



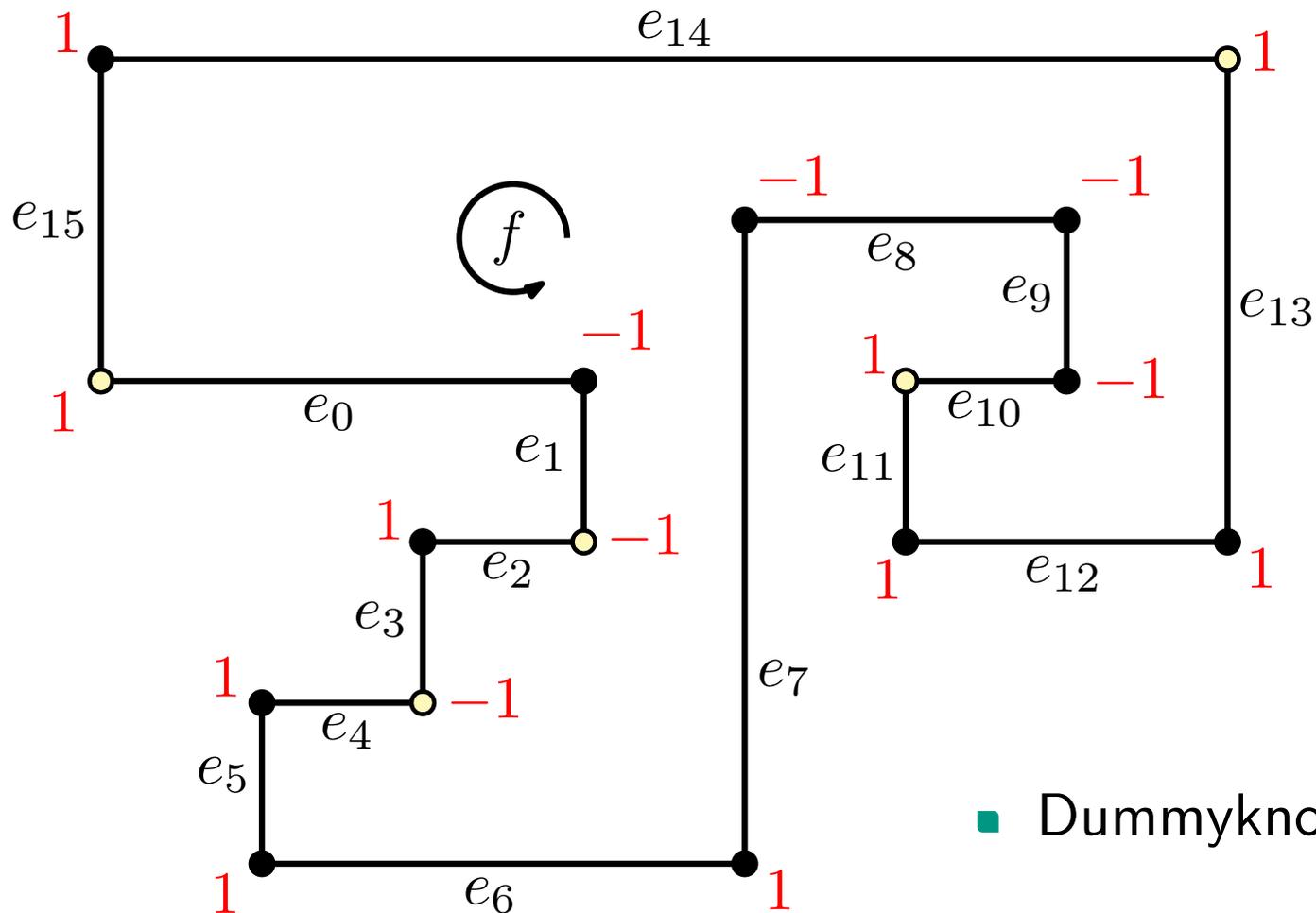
- Dummyknoten für Knicke

Verfeinerung von (G, H) – innere Facette



- Dummyknoten für Knicke

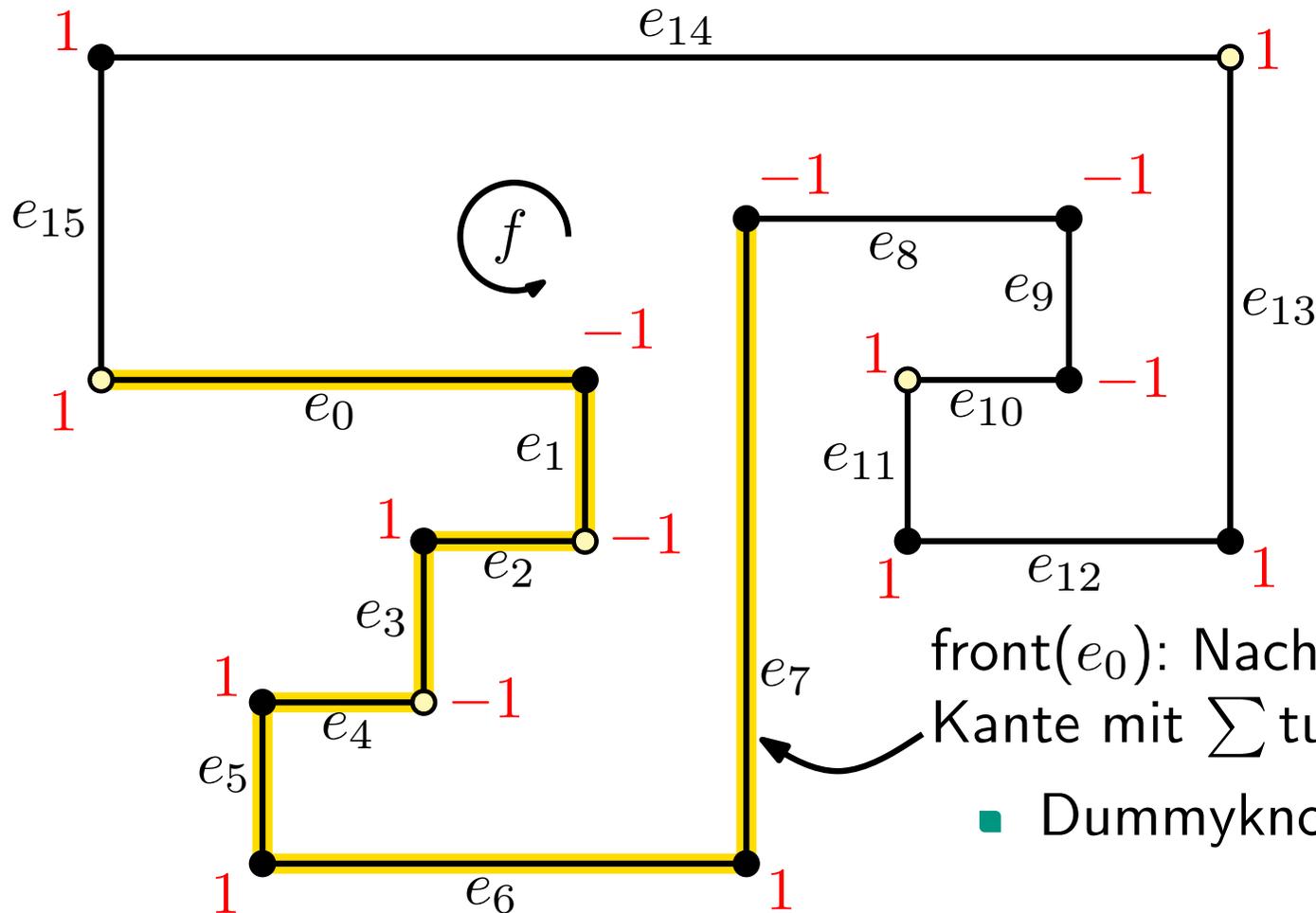
Verfeinerung von (G, H) – innere Facette



- Dummyknoten für Knicke

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

Verfeinerung von (G, H) – innere Facette

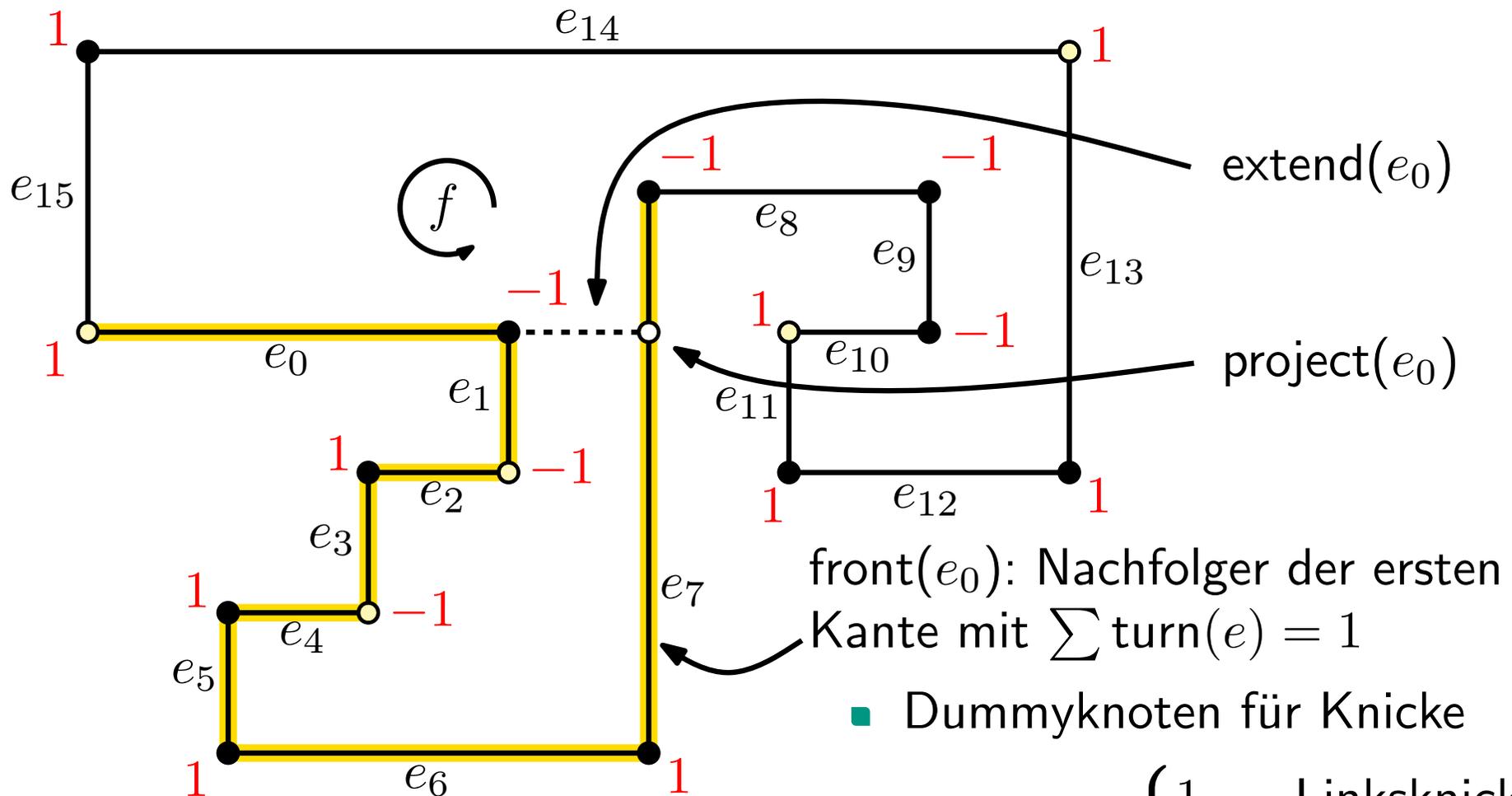


front(e_0): Nachfolger der ersten Kante mit $\sum \text{turn}(e) = 1$

- Dummyknoten für Knicke

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

Verfeinerung von (G, H) – innere Facette

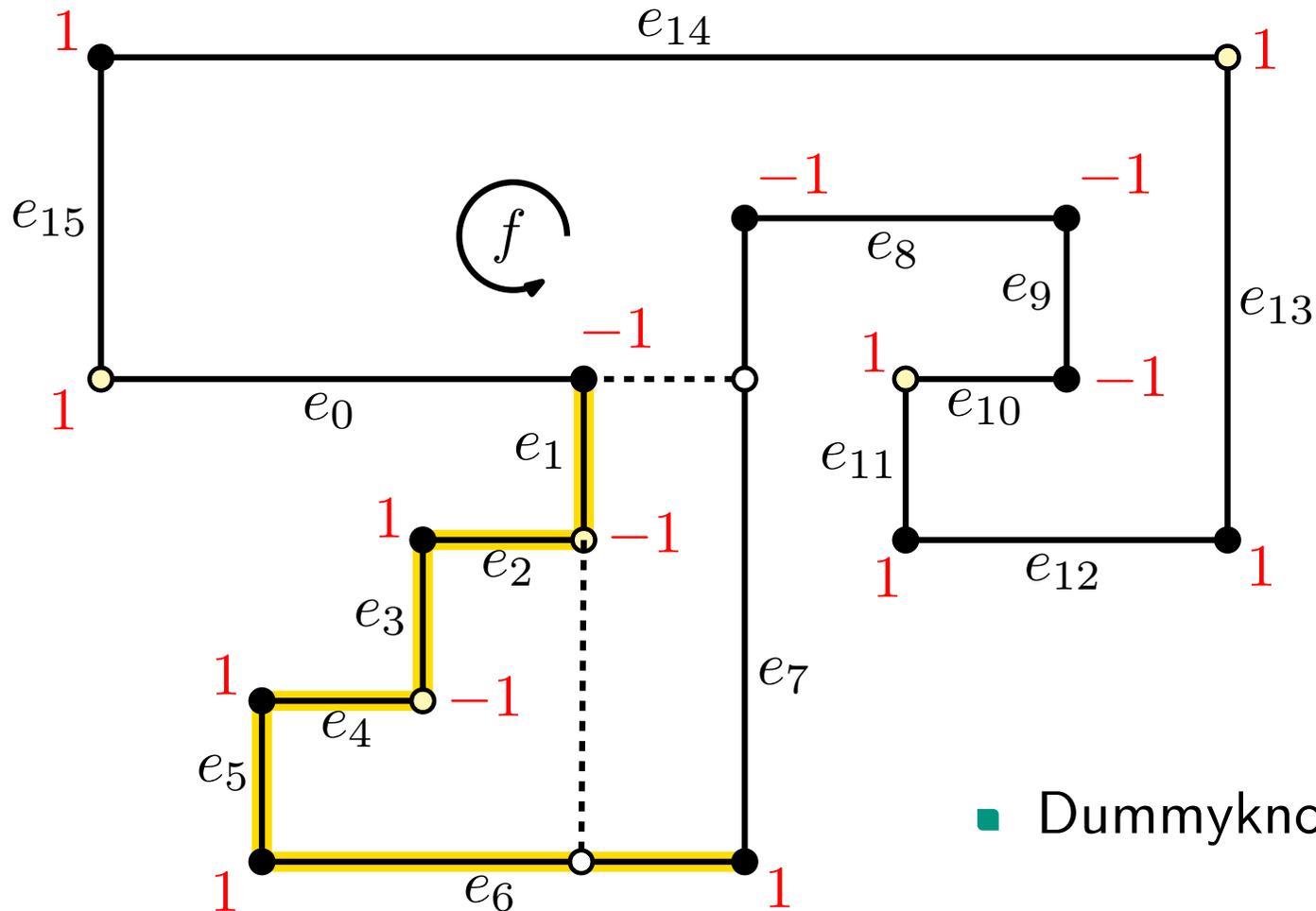


$\text{front}(e_0)$: Nachfolger der ersten Kante mit $\sum \text{turn}(e) = 1$

- Dummyknoten für Knicke

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

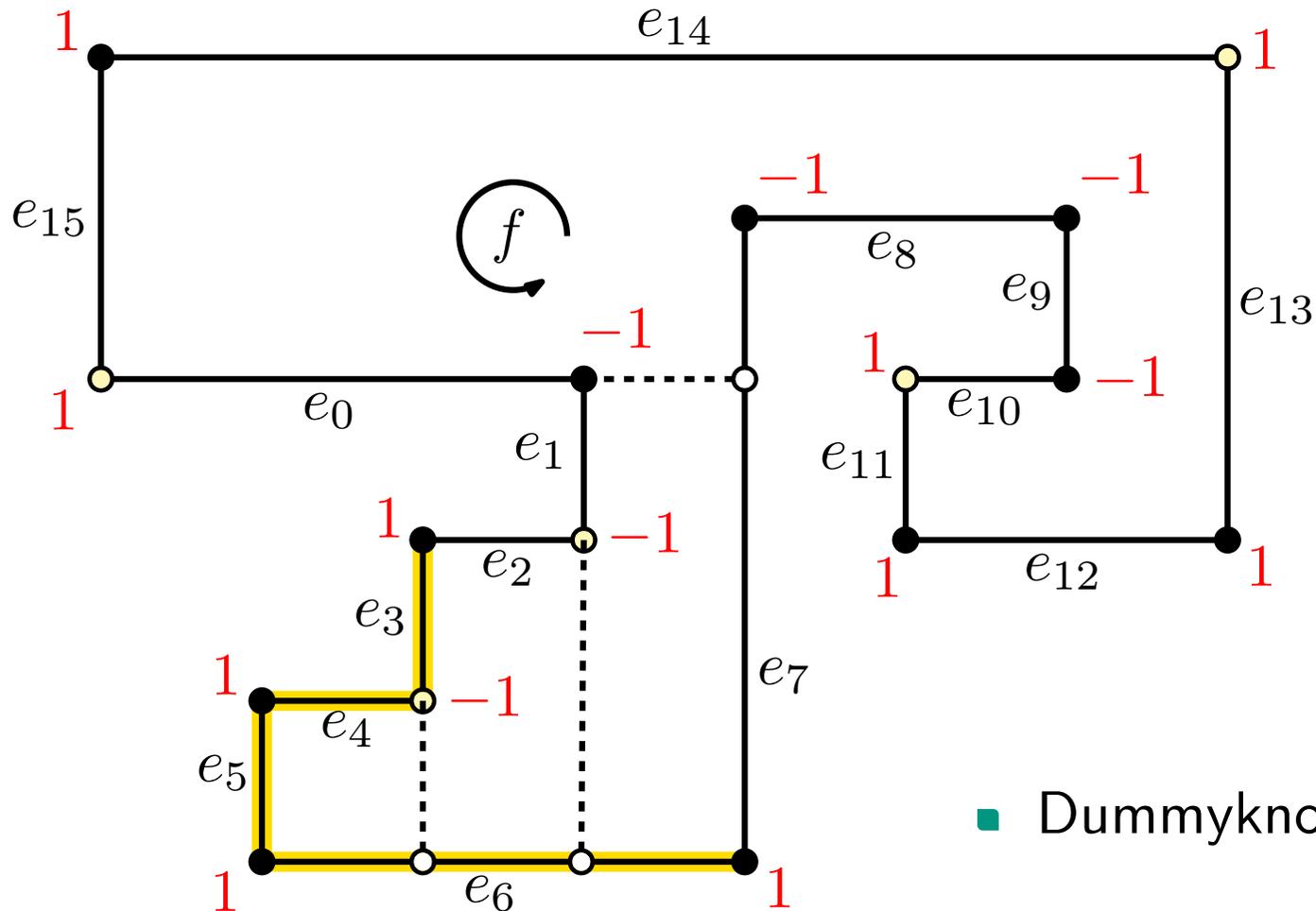
Verfeinerung von (G, H) – innere Facette



- Dummyknoten für Knicke

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

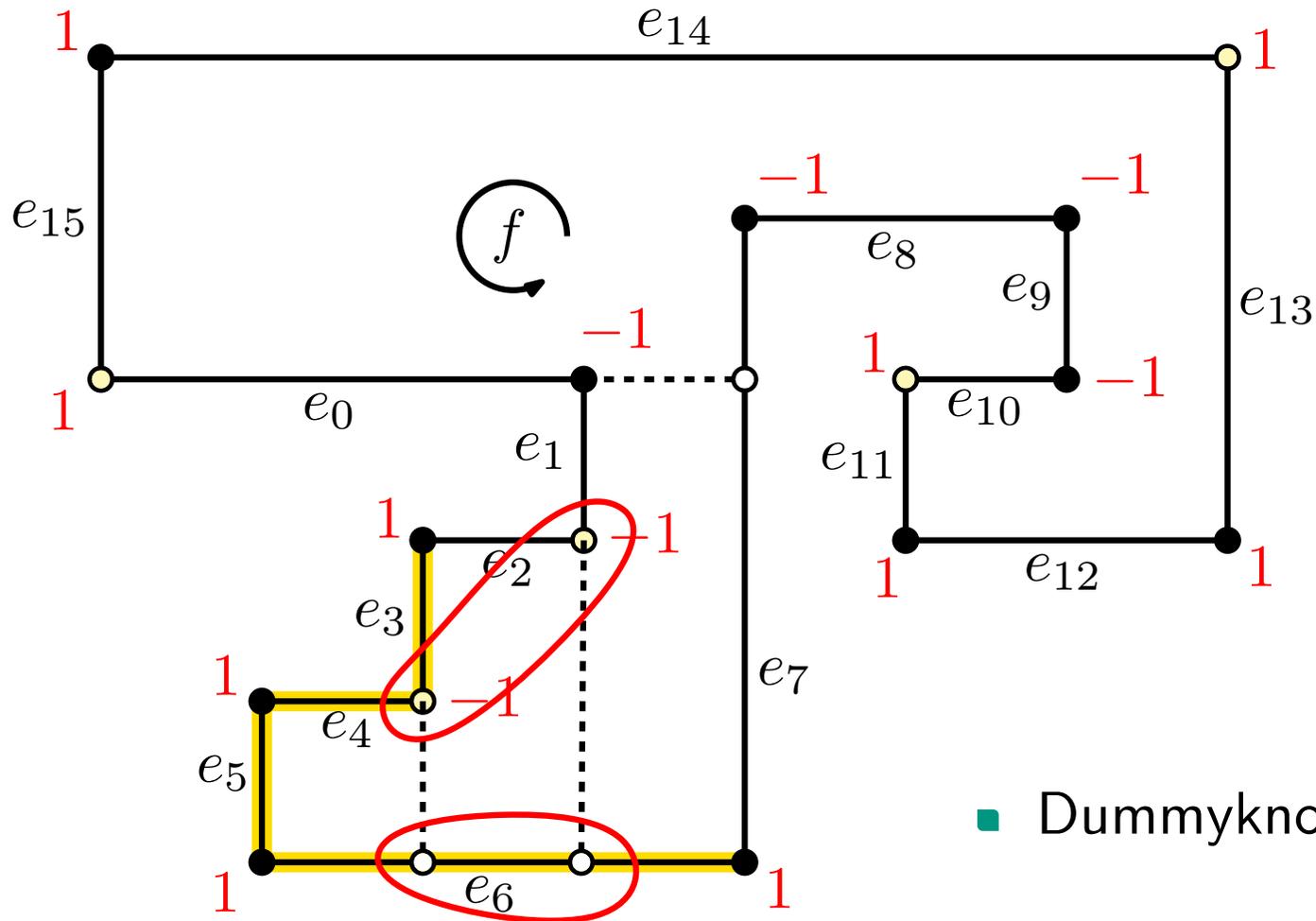
Verfeinerung von (G, H) – innere Facette



- Dummyknoten für Knicke

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

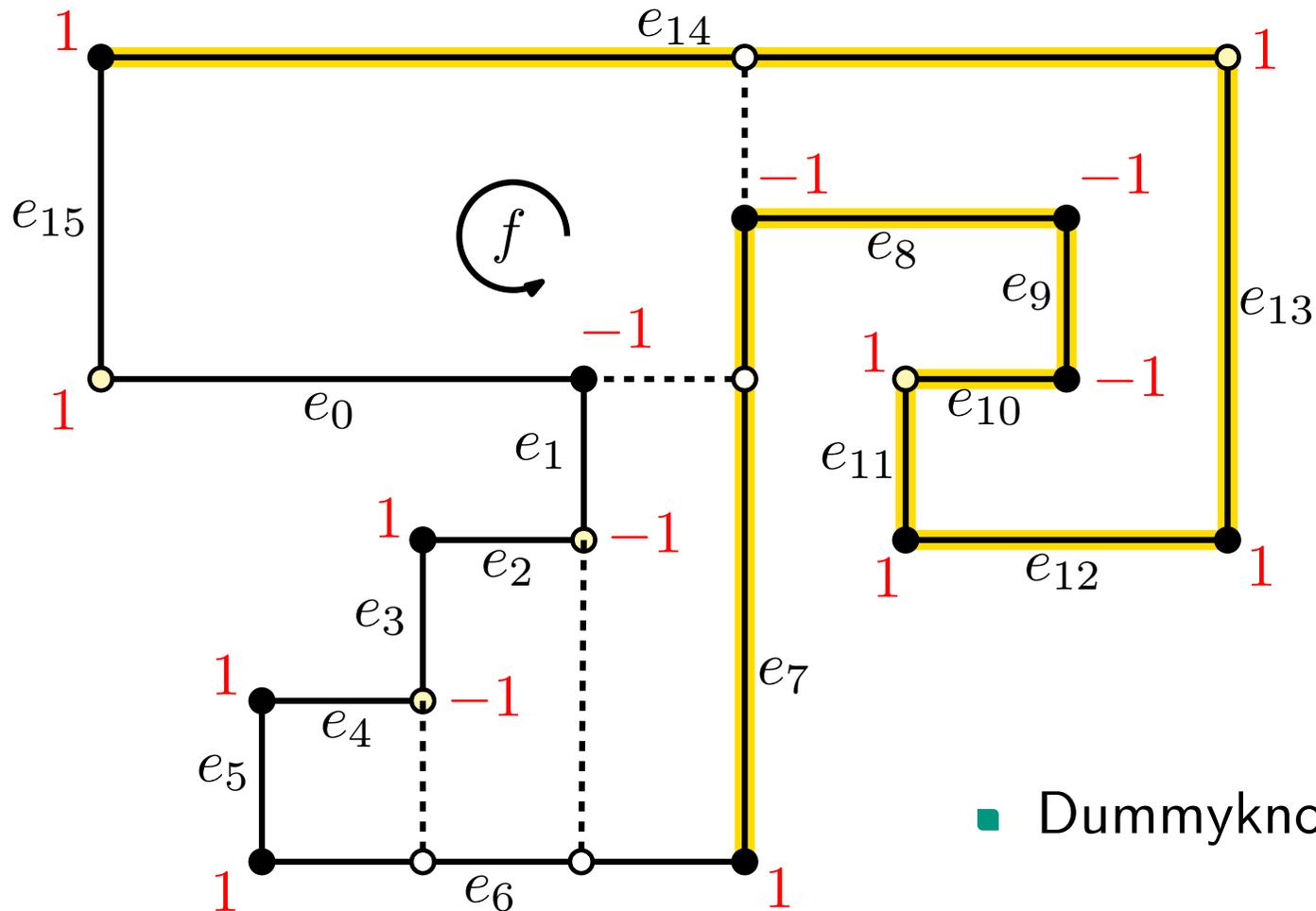
Verfeinerung von (G, H) – innere Facette



- Dummyknoten für Knicke

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

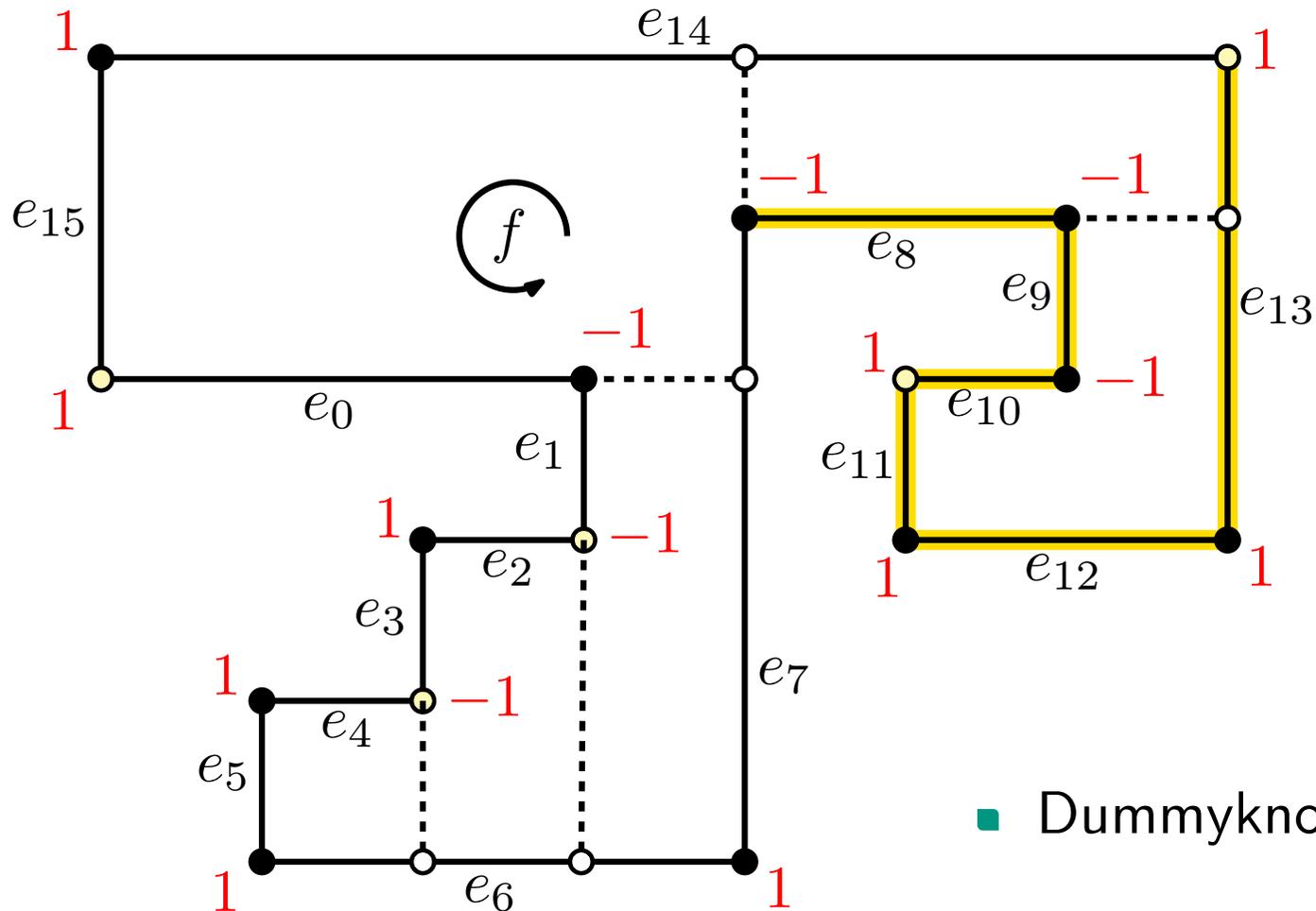
Verfeinerung von (G, H) – innere Facette



- Dummyknoten für Knicke

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

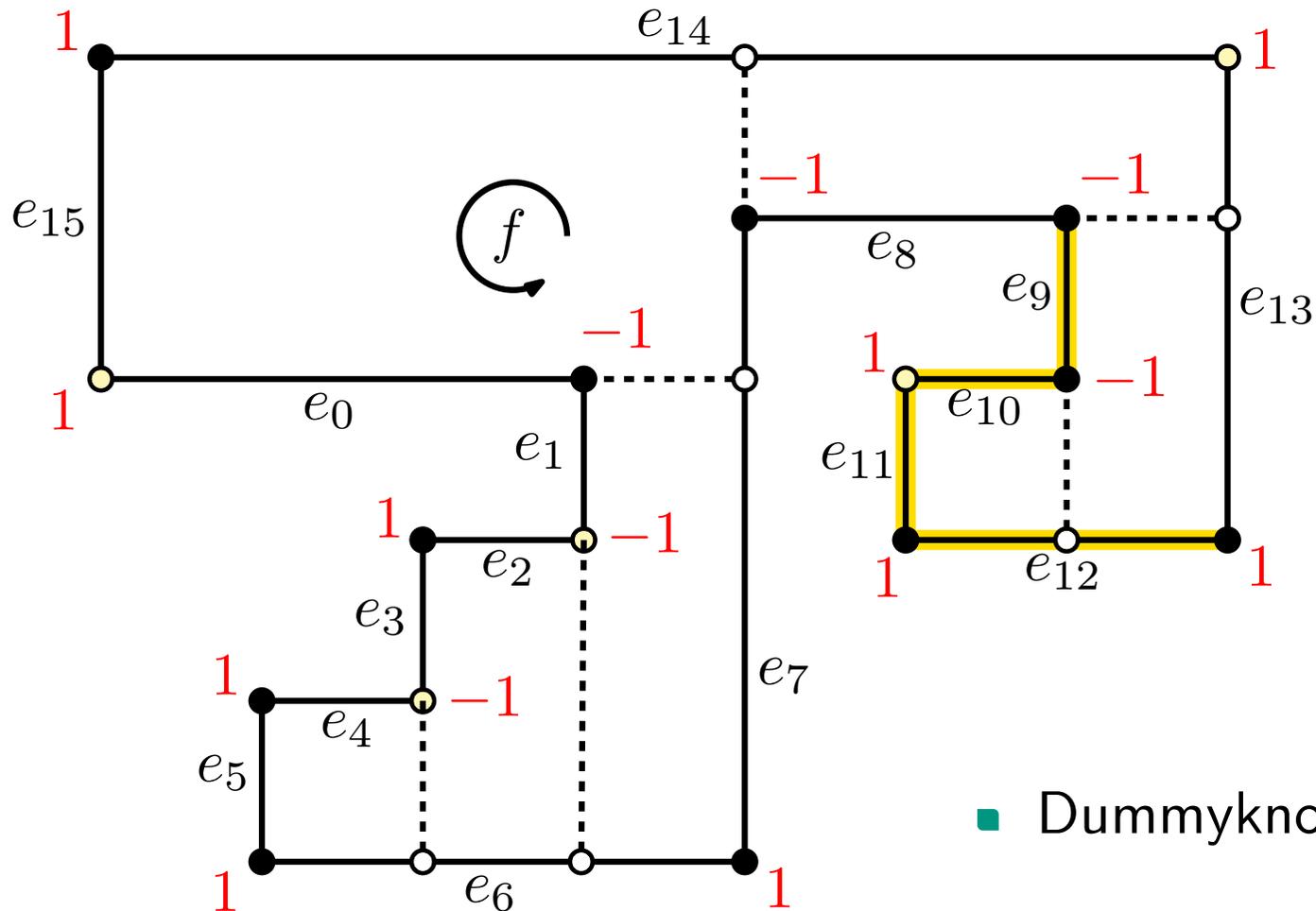
Verfeinerung von (G, H) – innere Facette



- Dummyknoten für Knicke

- $\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$

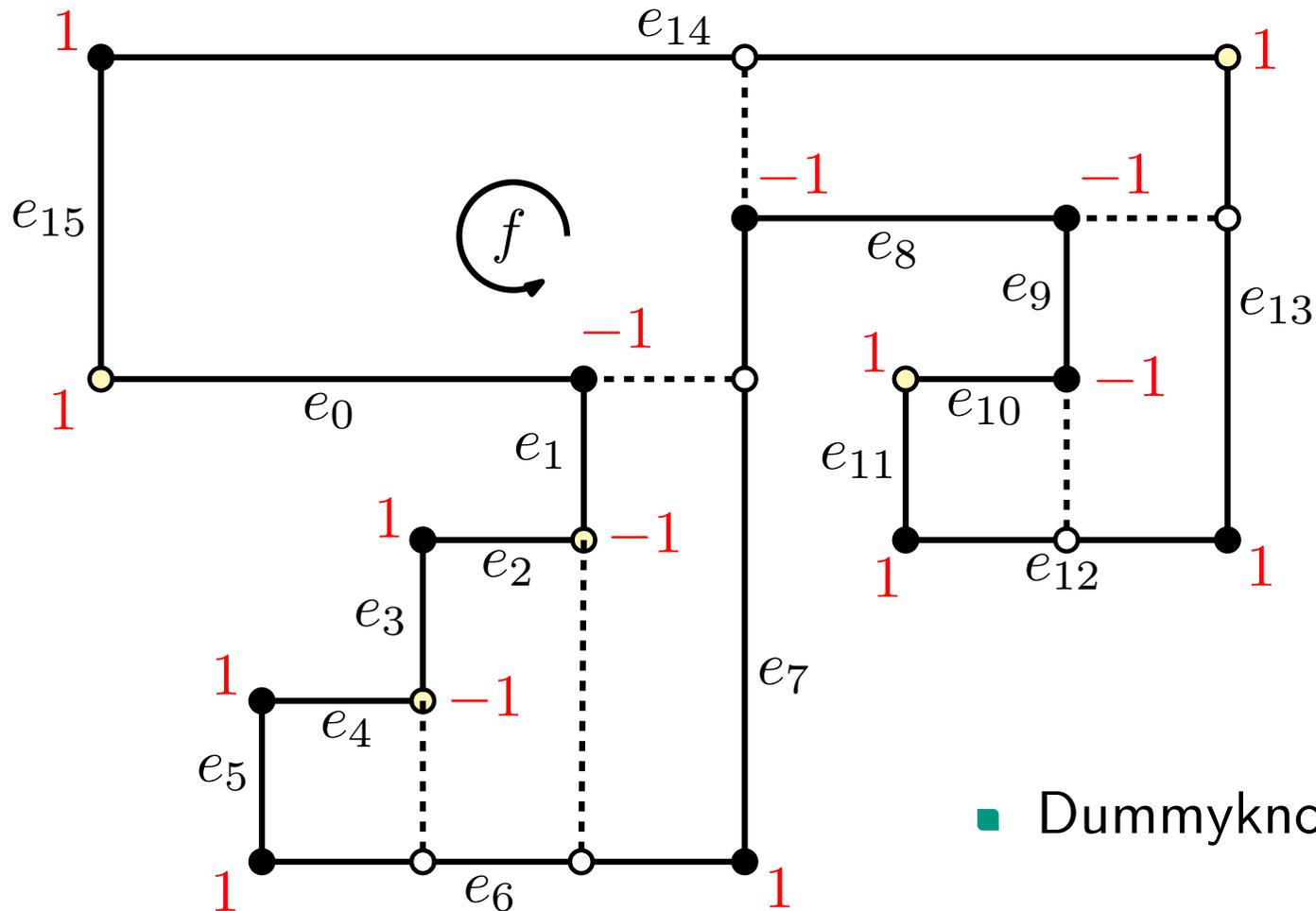
Verfeinerung von (G, H) – innere Facette



- Dummyknoten für Knicke

- $\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$

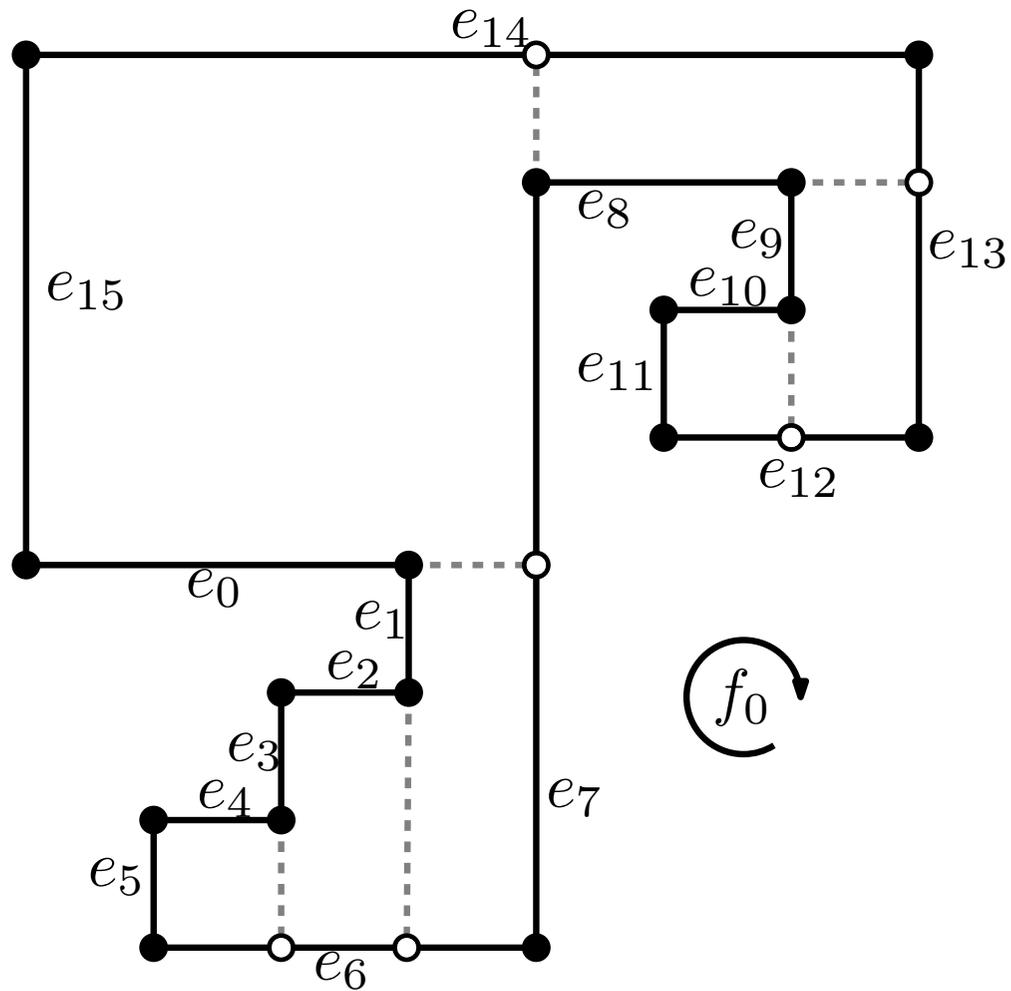
Verfeinerung von (G, H) – innere Facette



- Dummyknoten für Knicke

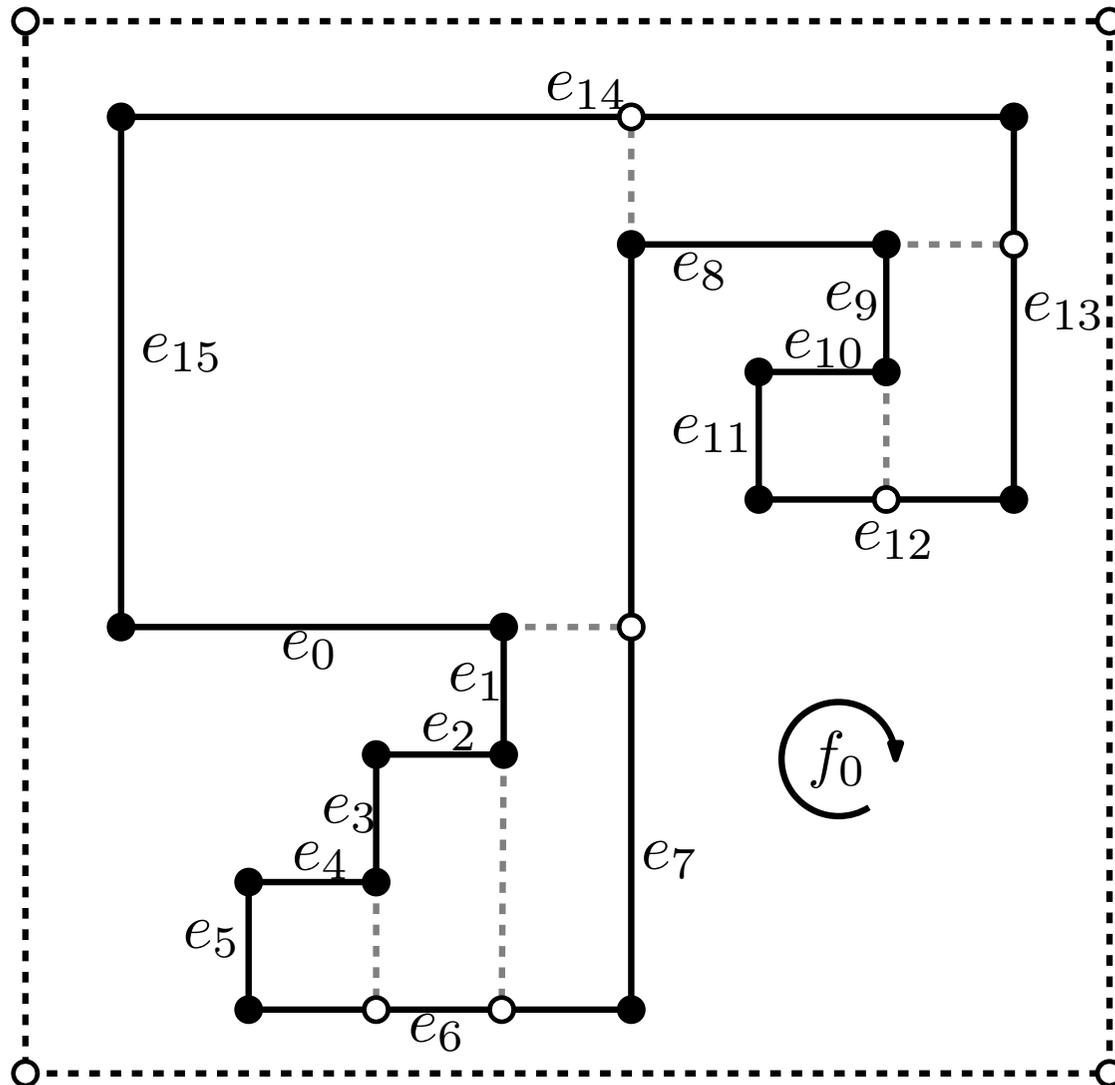
$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

Verfeinerung von (G, H) – äußere Facette



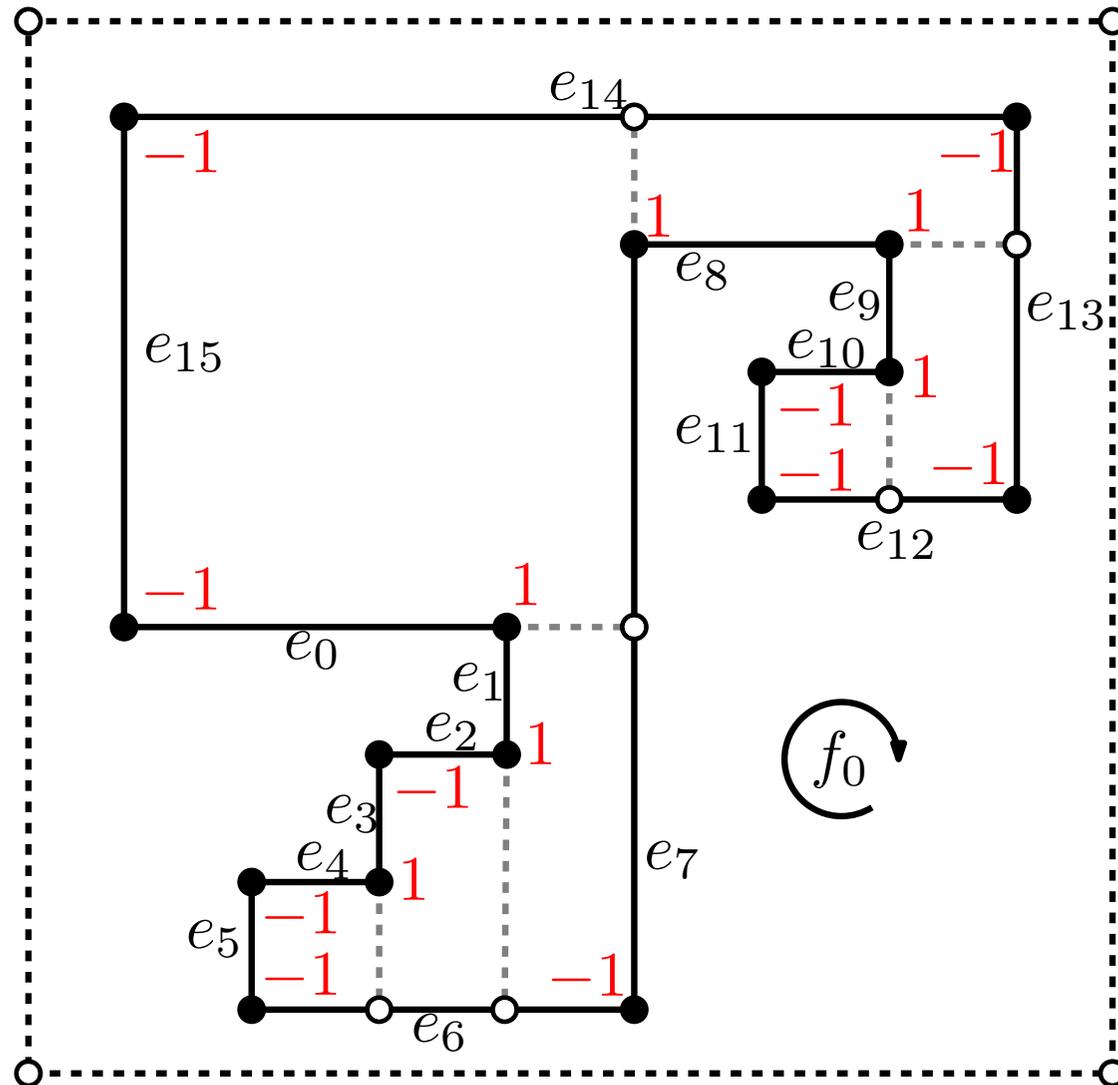
Verfeinerung von (G, H) – äußere Facette

- Rechteck R um die äußere Facette legen



Verfeinerung von (G, H) – äußere Facette

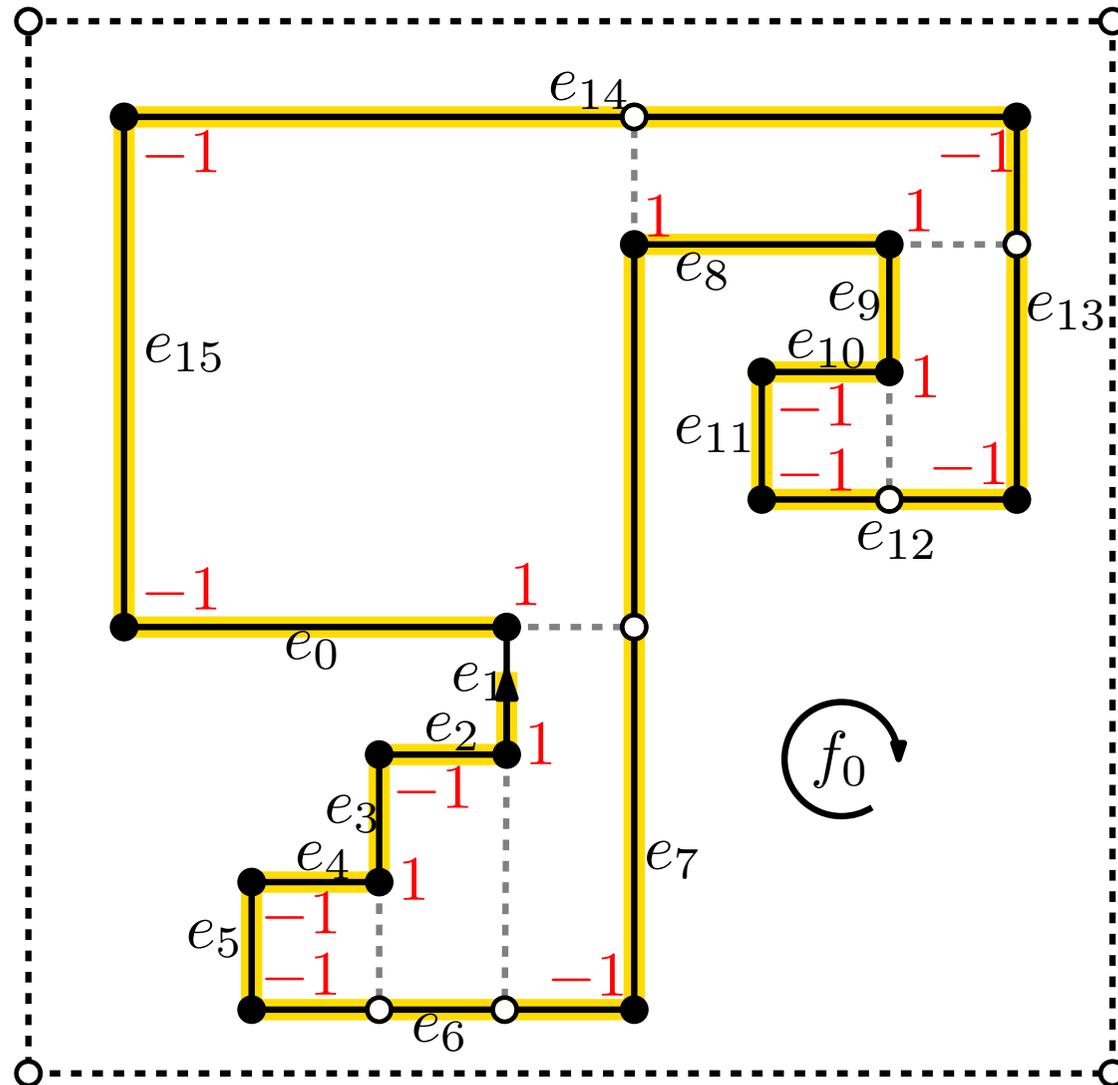
- Rechteck R um die äußere Facette legen



$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

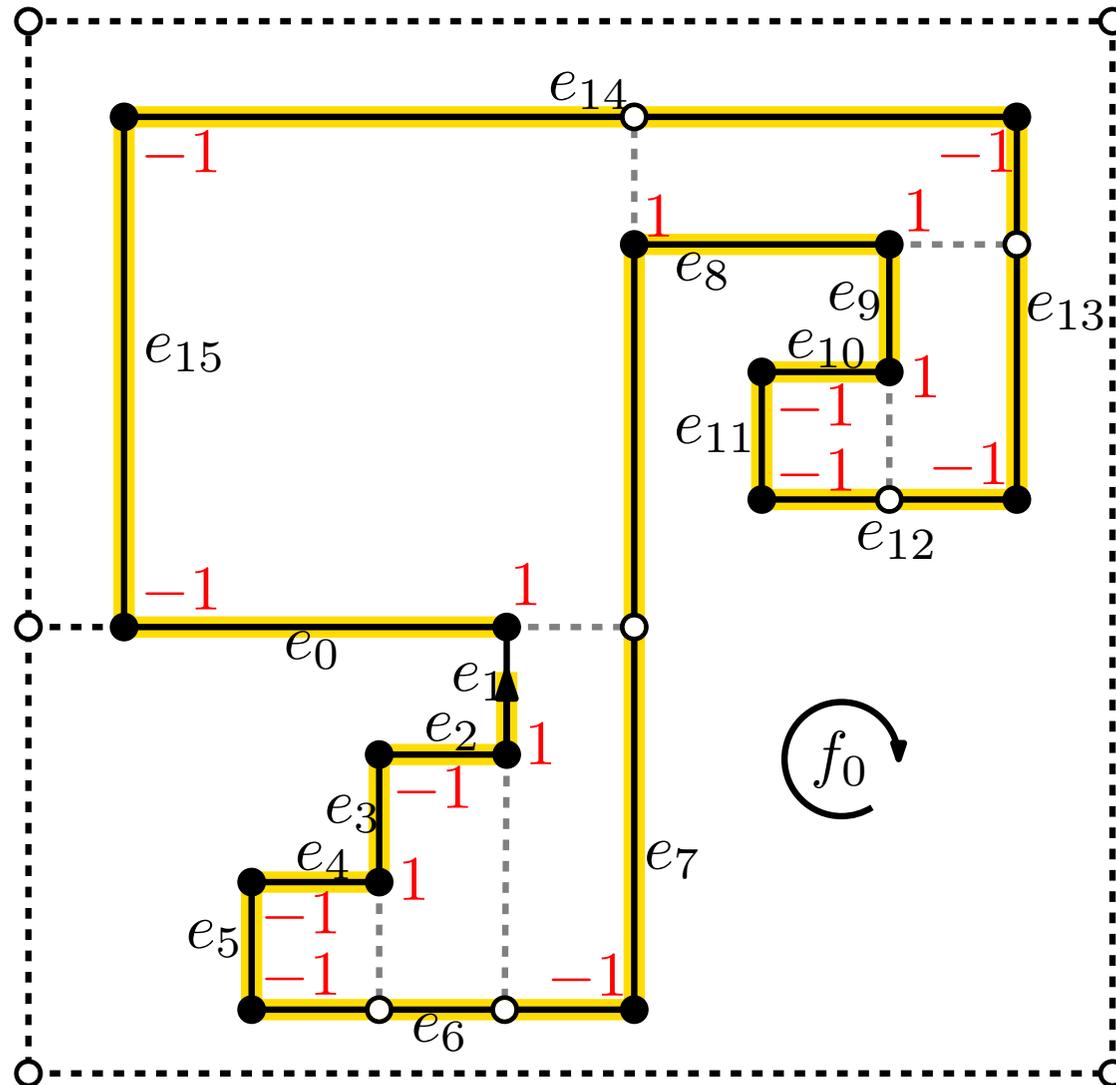
Verfeinerung von (G, H) – äußere Facette

- Rechteck R um die äußere Facette legen



$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

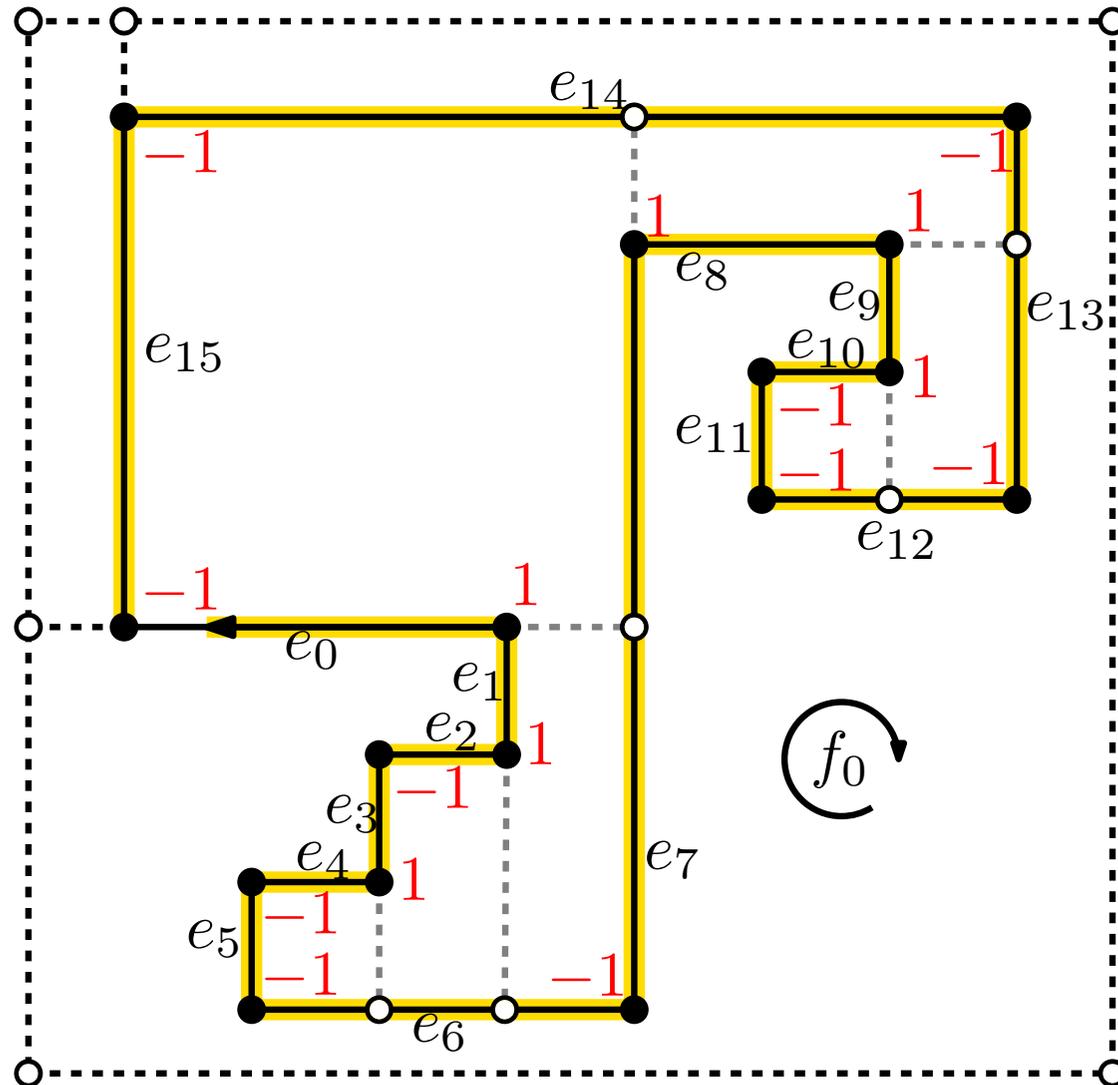
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

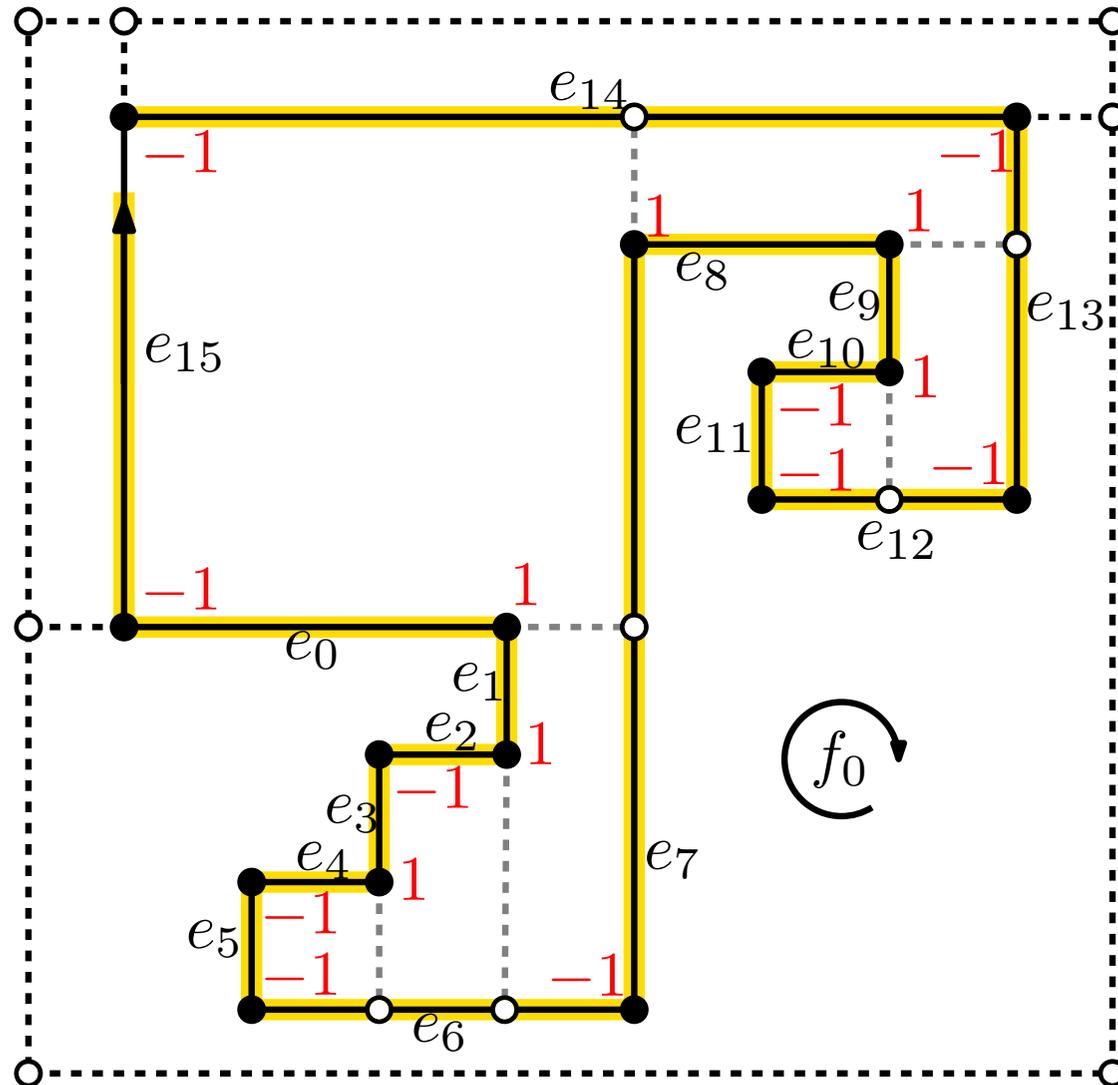
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

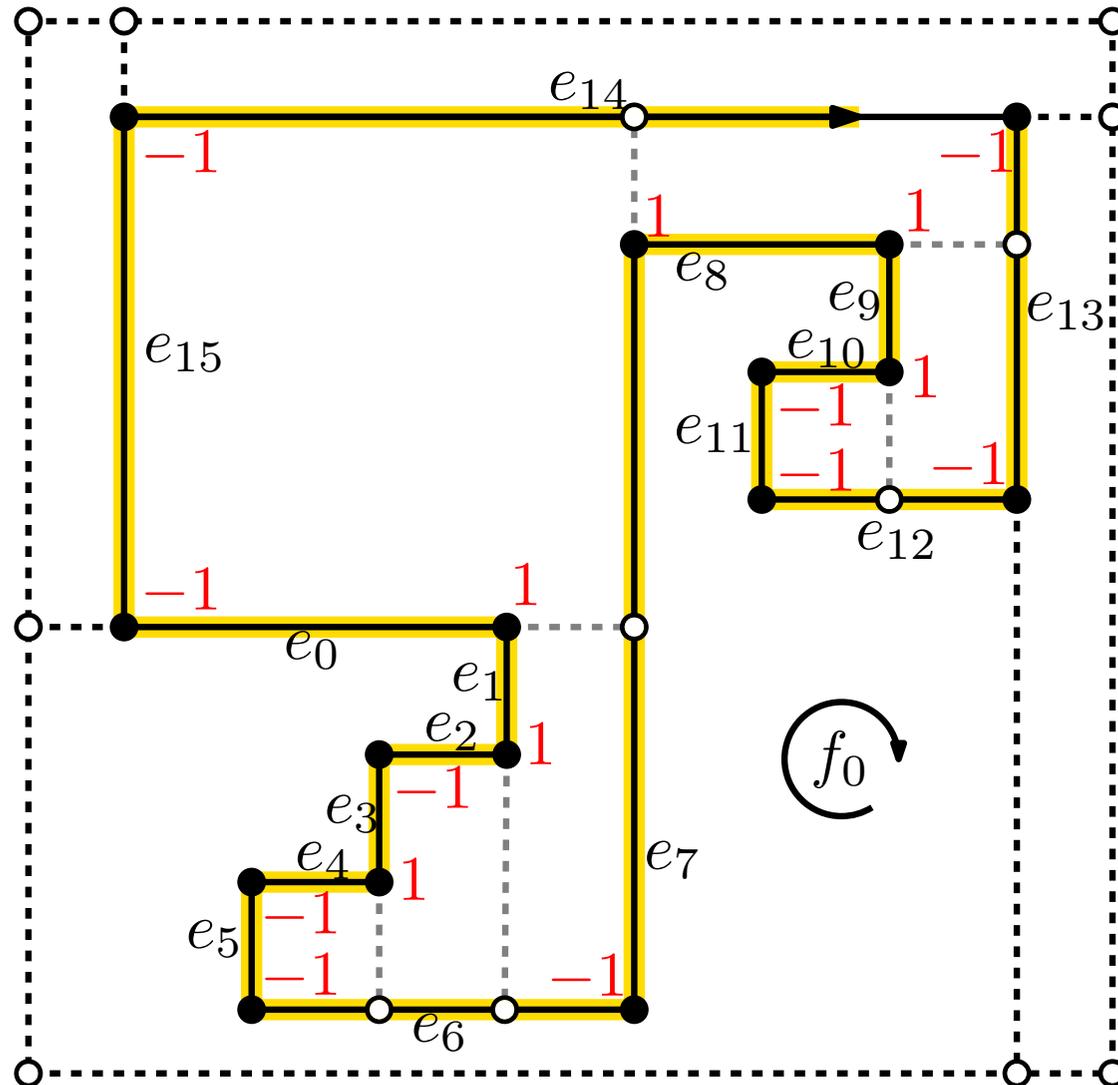
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

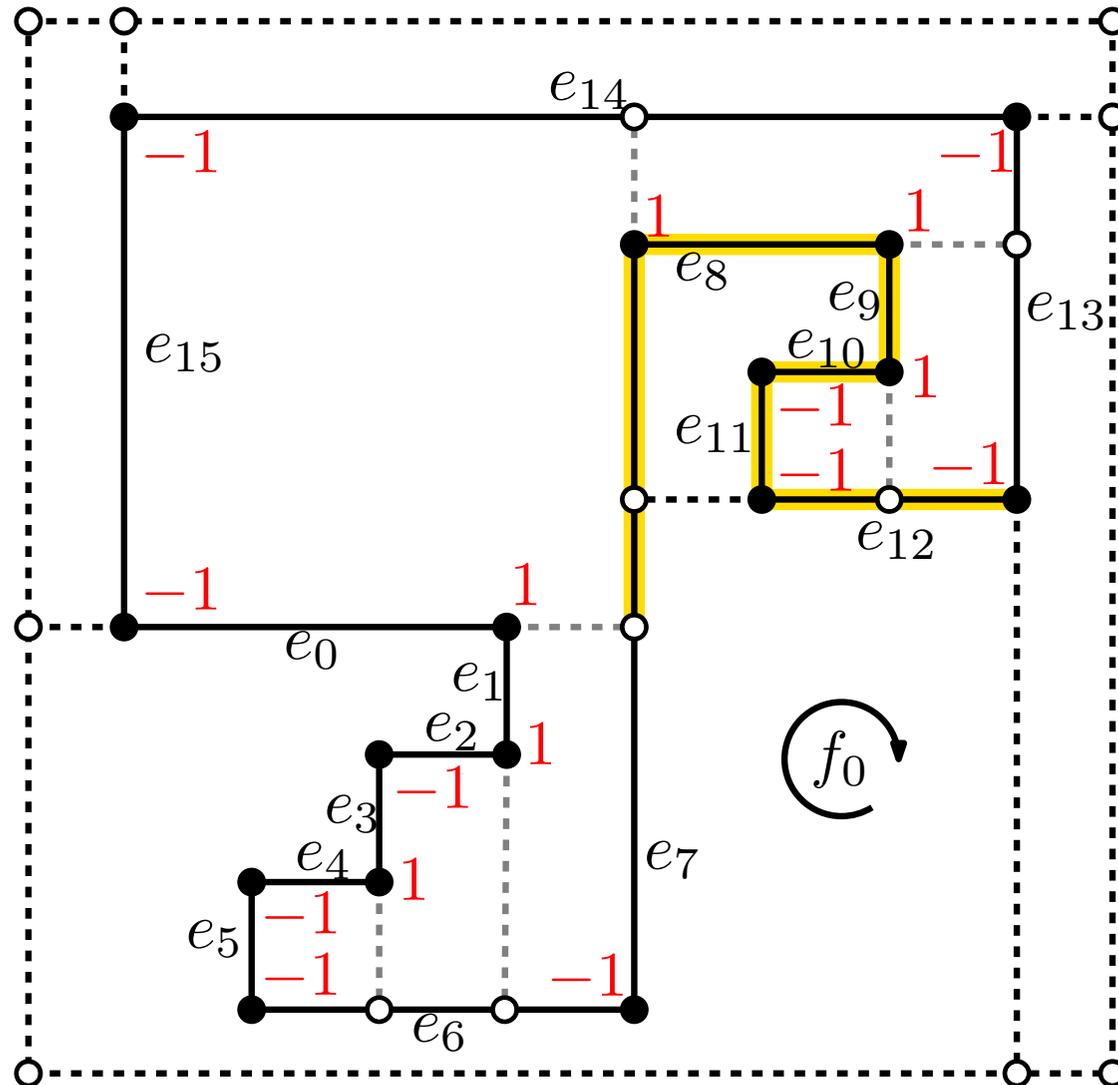
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

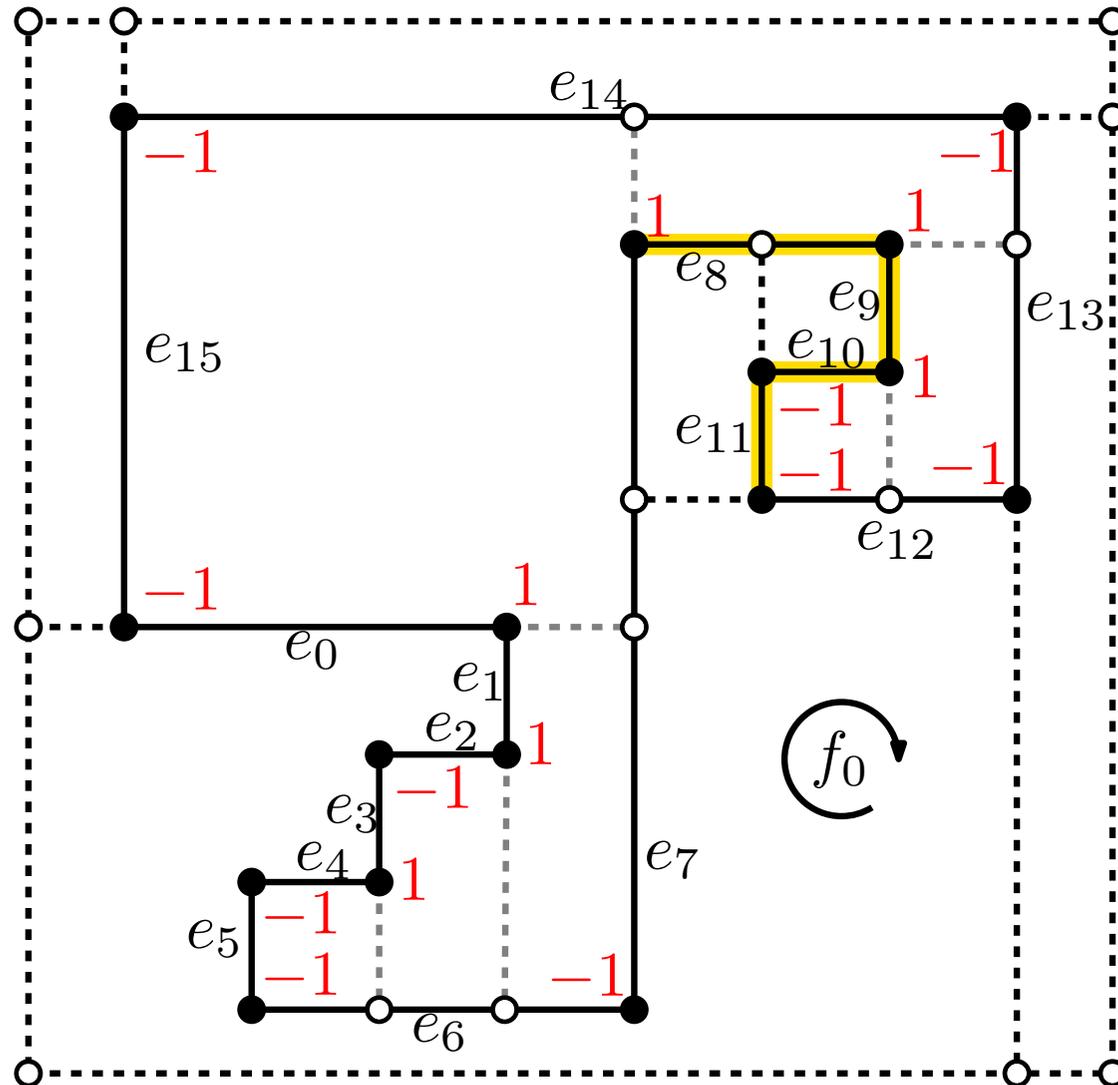
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

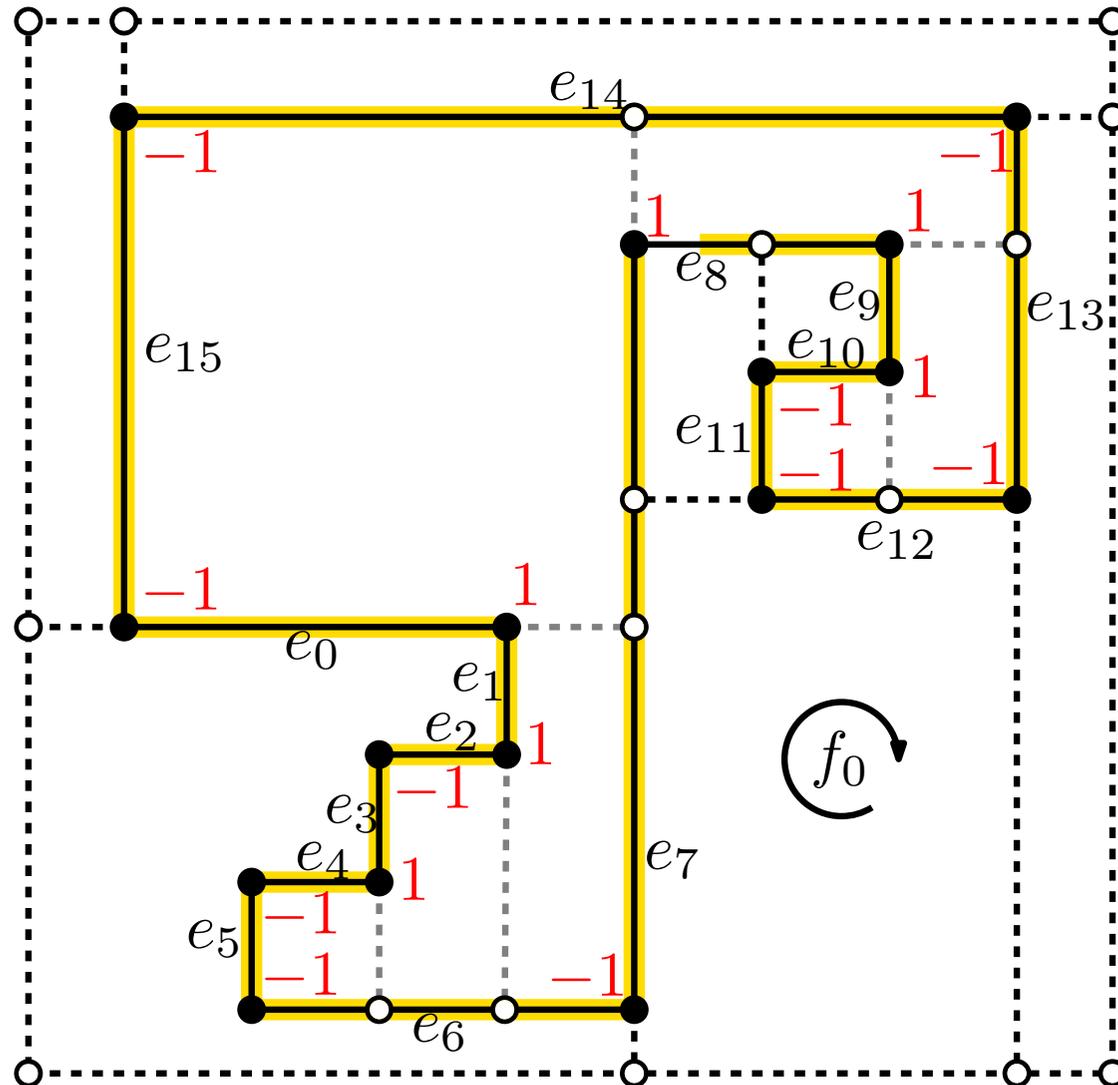
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

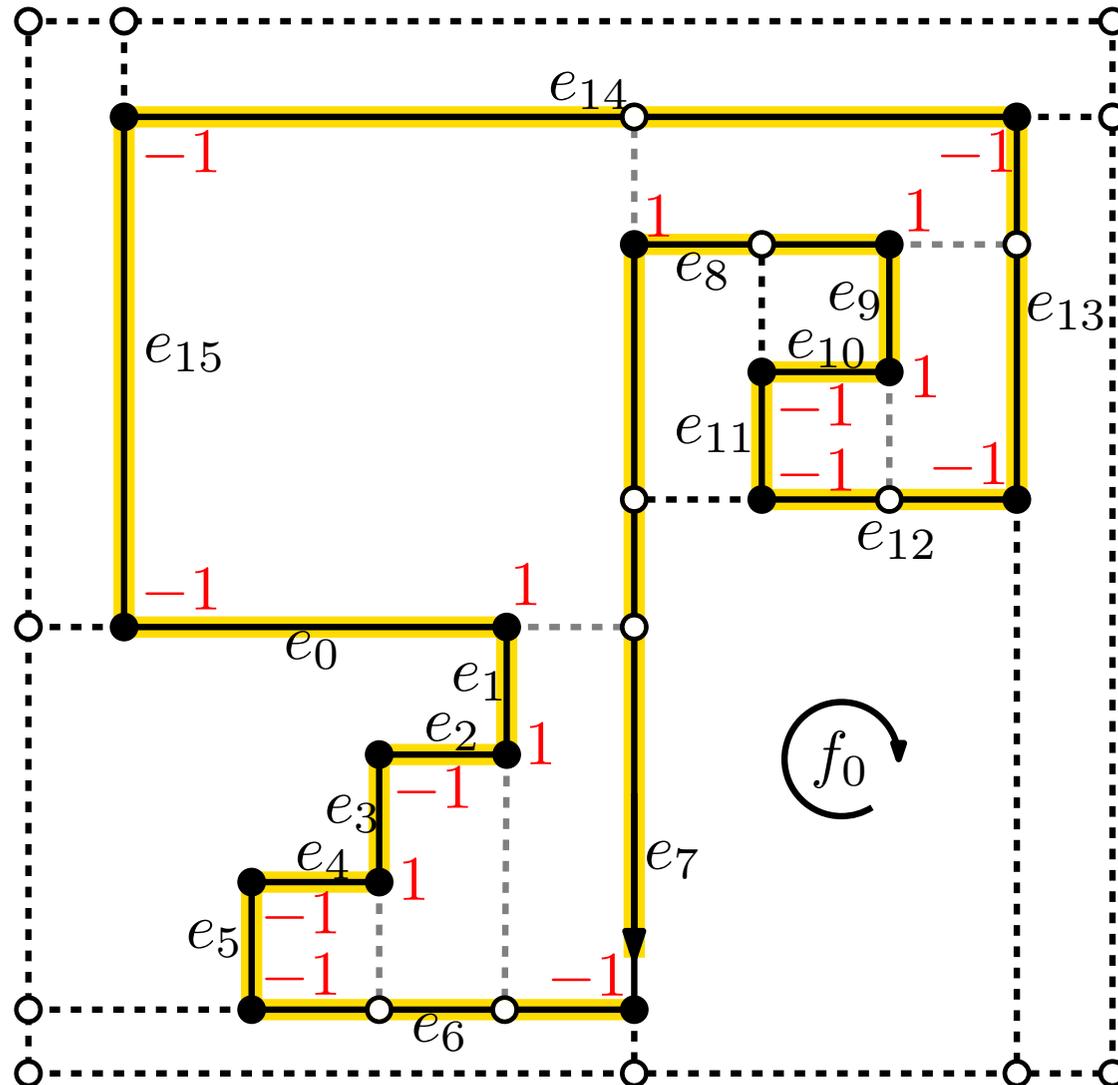
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

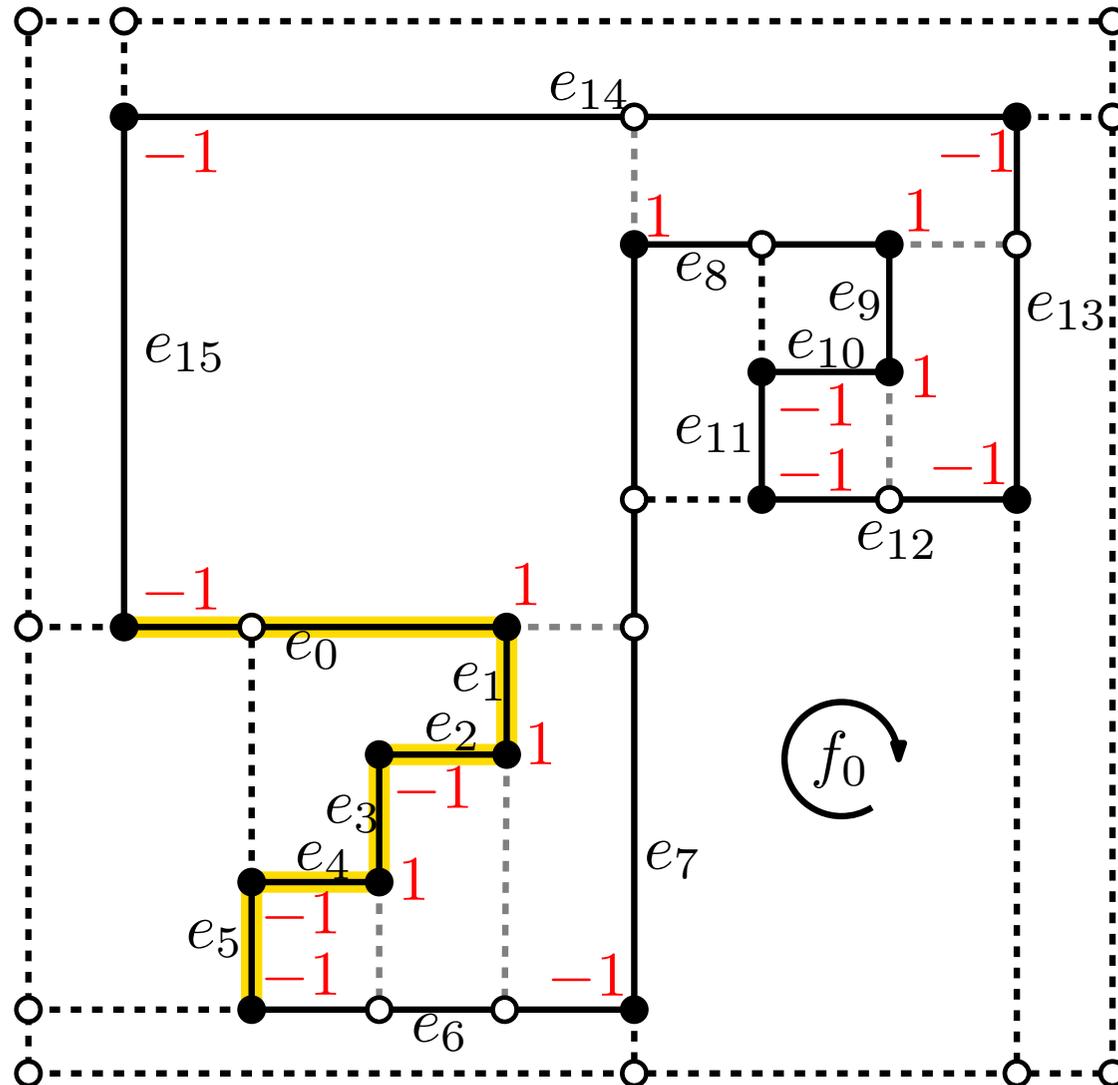
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

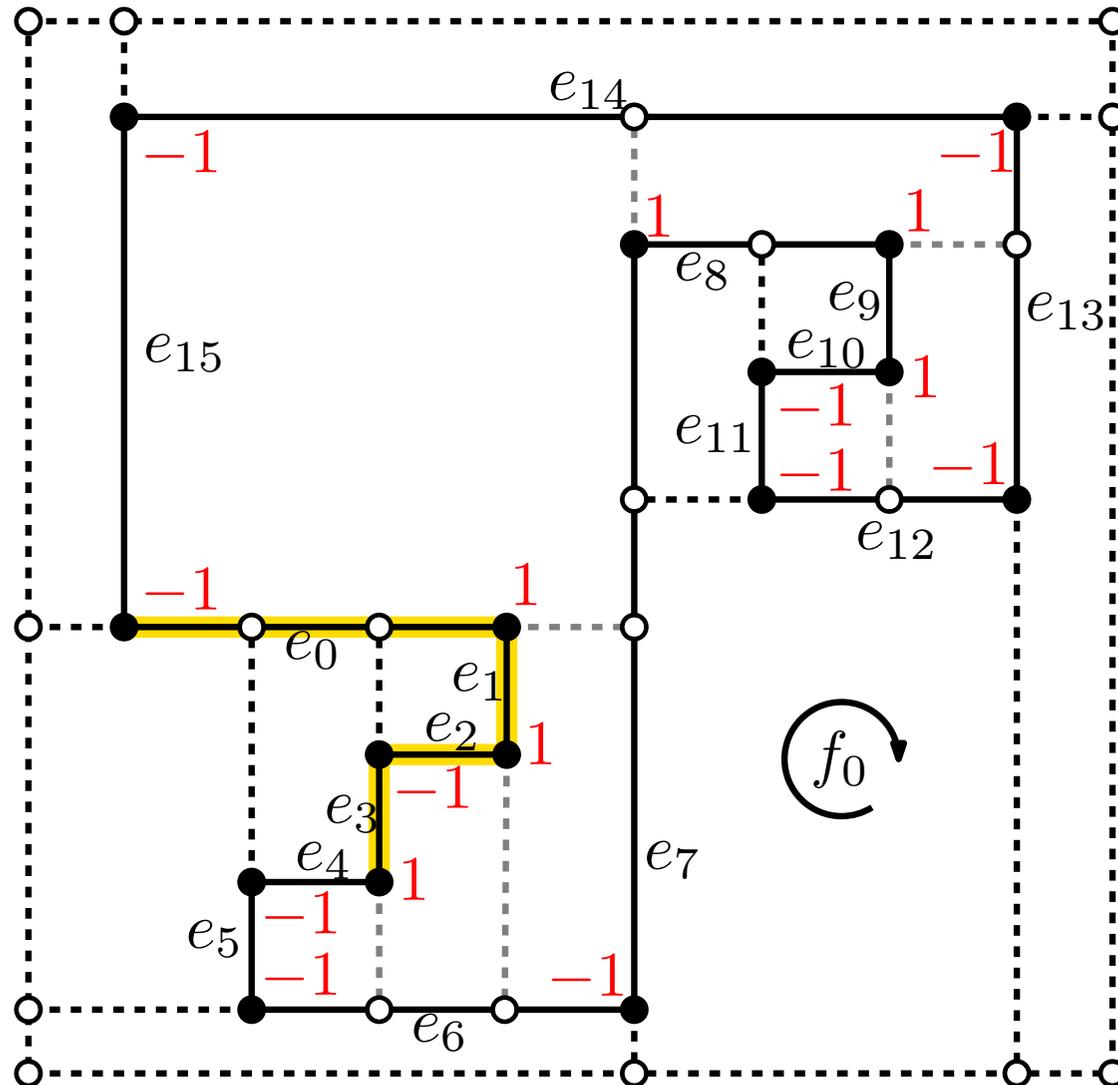
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

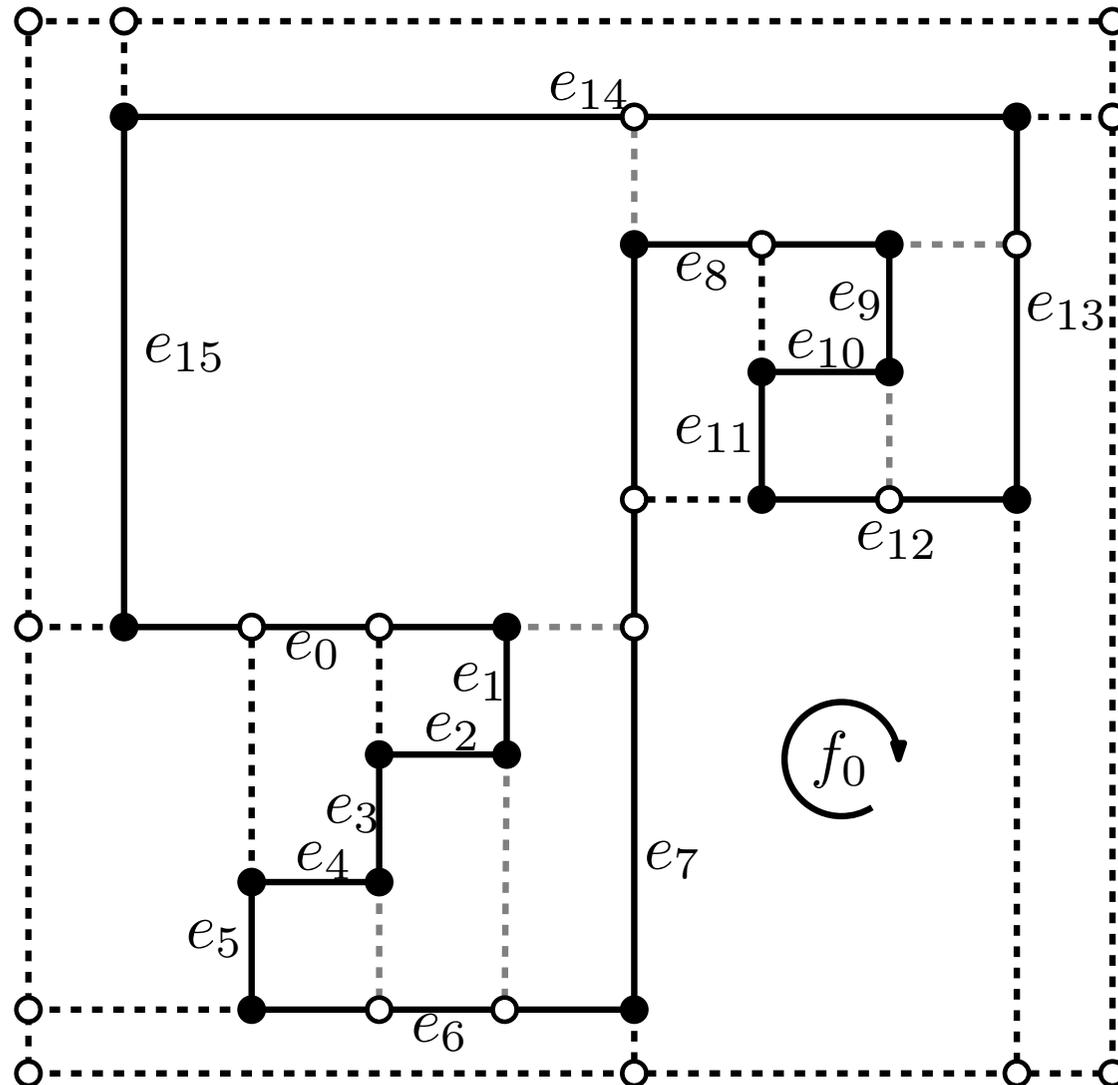
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

$$\text{turn}(e) = \begin{cases} 1 & \text{Linksknick} \\ 0 & \text{kein Knick} \\ -1 & \text{Rechtsknick} \end{cases}$$

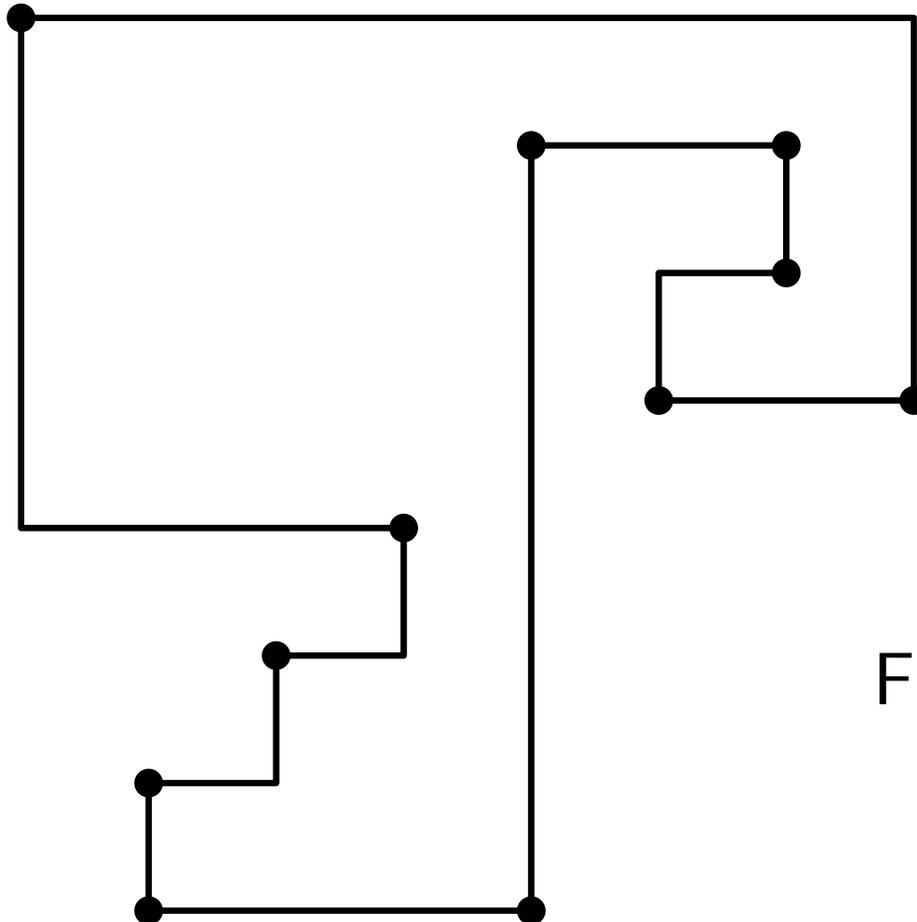
Verfeinerung von (G, H) – äußere Facette



- Rechteck R um die äußere Facette legen
- falls $\sum \text{turn}(e) < 1$ für komplette Umrundung von f_0 , projiziere auf R

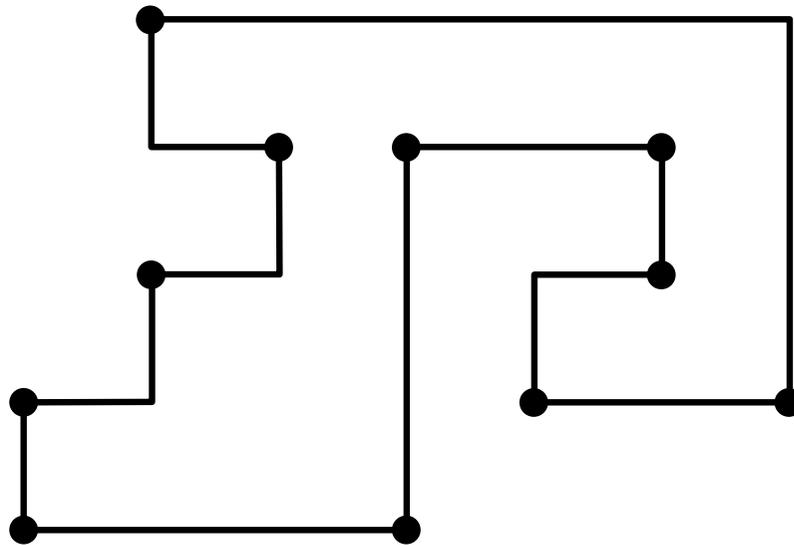
alle Facetten sind Rechtecke
→ wende Flussnetzwerke an

Verfeinerung von (G, H) – äußere Facette



Flächenminimal?

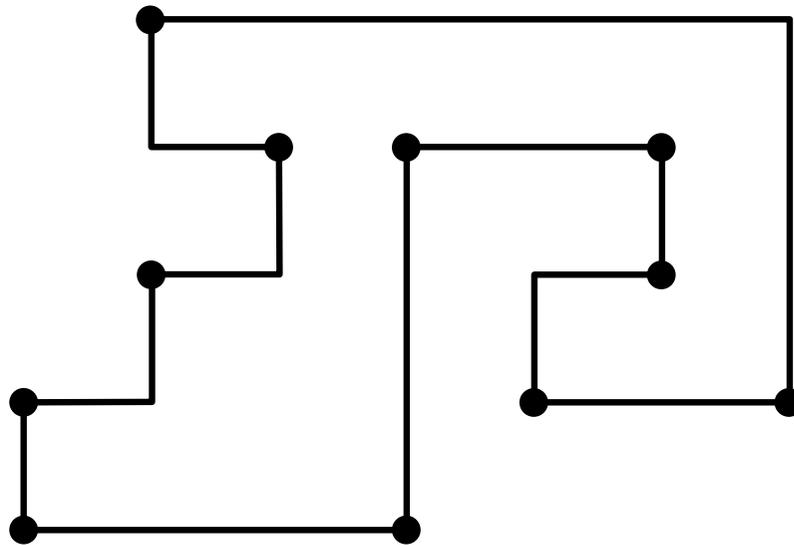
Verfeinerung von (G, H) – äußere Facette



Flächenminimal?

Nein!

Verfeinerung von (G, H) – äußere Facette



Flächenminimal?

Nein!

Die Flächenkompaktierung bei gegebener orthogonaler Beschreibung ist im Allgemeinen NP-schwer.

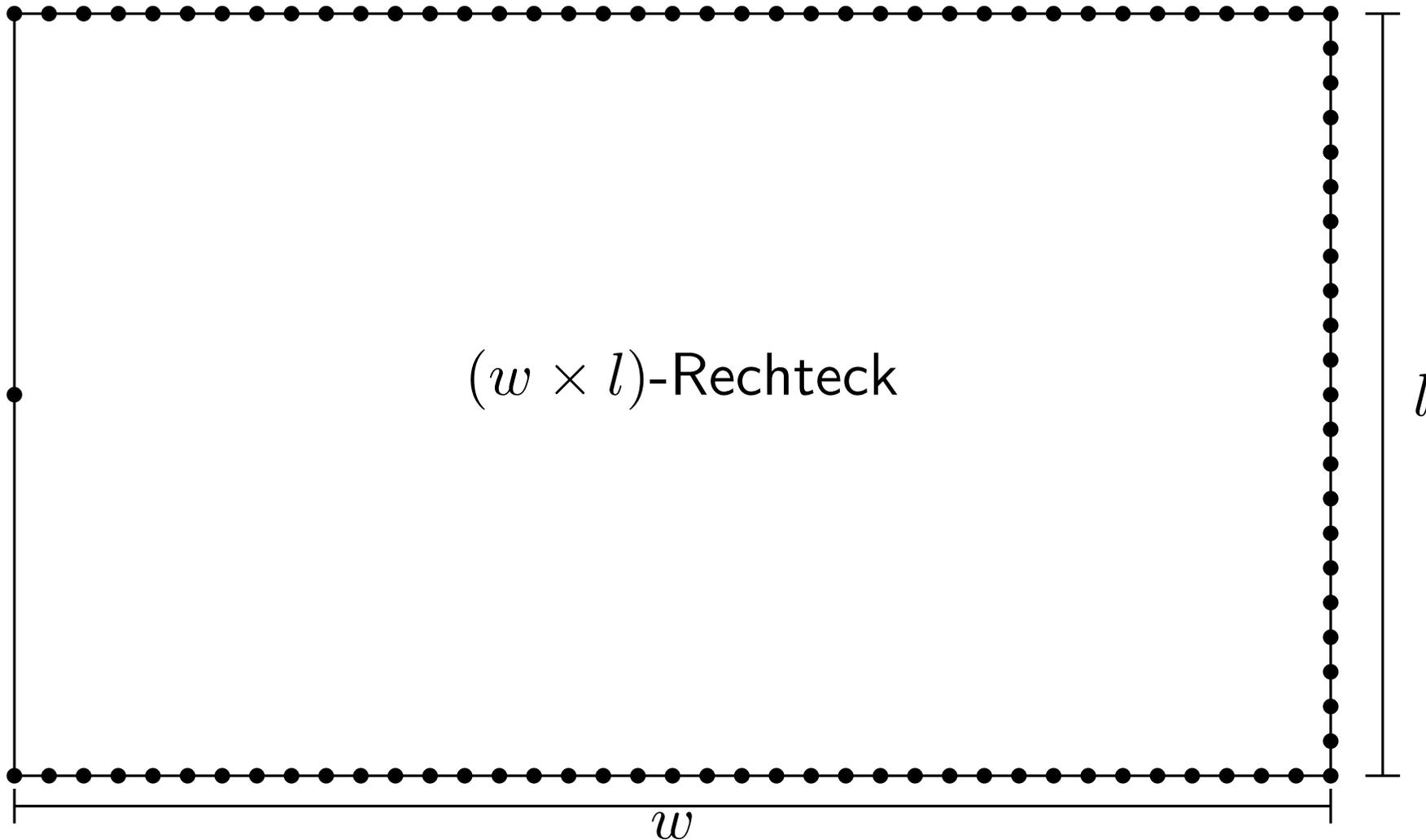
Satz 3: Für einen Graphen G mit gegebener orthogonaler Beschreibung $H(G)$ und ein $K \in \mathbb{N}$ ist es NP-schwer zu entscheiden, ob sich $(G, H(G))$ auf einem Gitter der Größe höchstens K zeichnen lässt.

Satz 3: Für einen Graphen G mit gegebener orthogonaler Beschreibung $H(G)$ und ein $K \in \mathbb{N}$ ist es NP-schwer zu entscheiden, ob sich $(G, H(G))$ auf einem Gitter der Größe höchstens K zeichnen lässt.

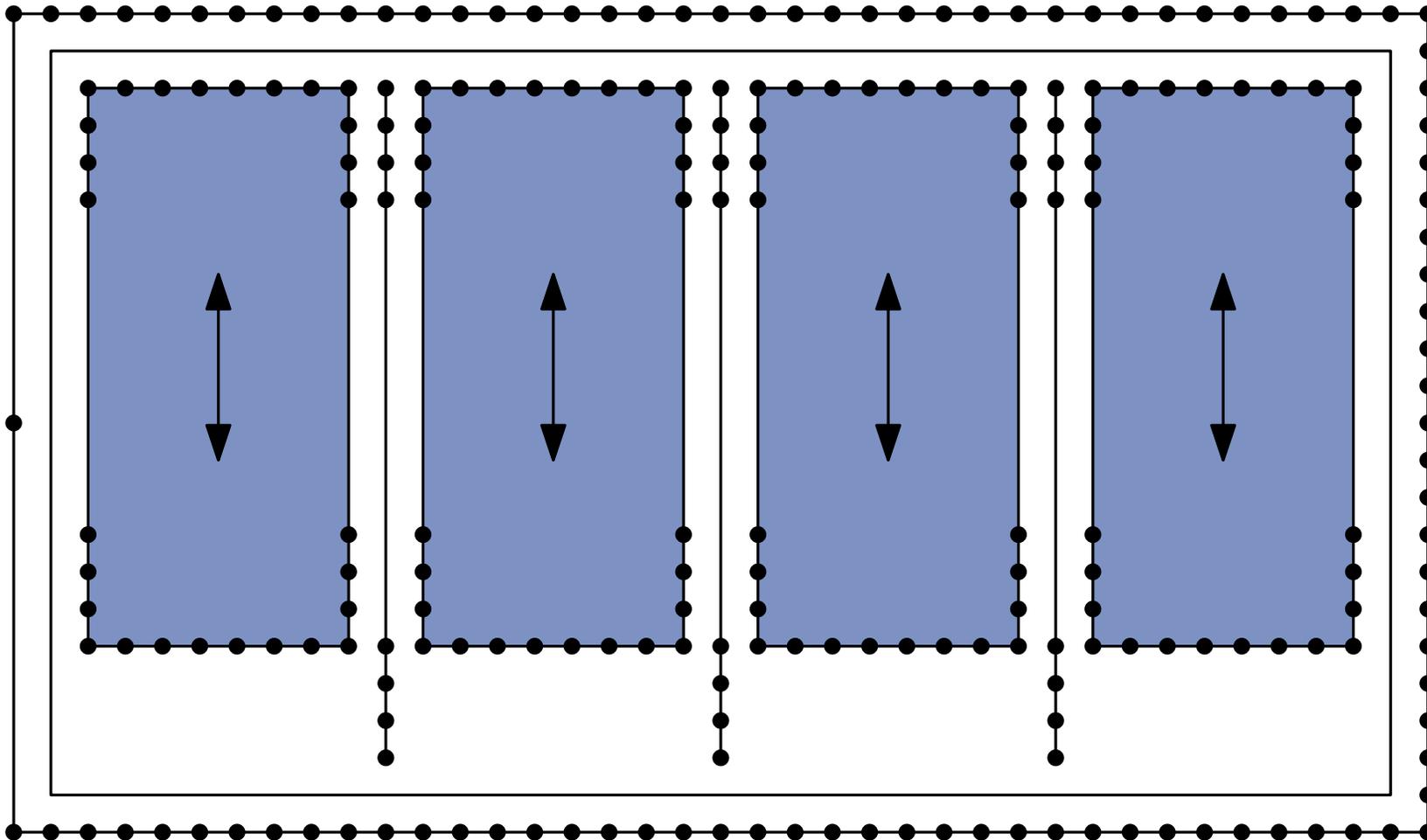
Beweisskizze:

- Reduktion von SAT, d.h. Erfüllbarkeitstest für Boolesche Formel $\Phi = c_1 \wedge \dots \wedge c_m$, wobei jedes c_i Klausel über Variablenmenge $\{x_1, \dots, x_n\}$
- geometrische Struktur der Reduktion
 - Klauselgadgets
 - Variabलगadgets
- Bestimme geeigneten Wert K , so dass sich (G, H) in Fläche K zeichnen lässt gdw. Φ erfüllbar

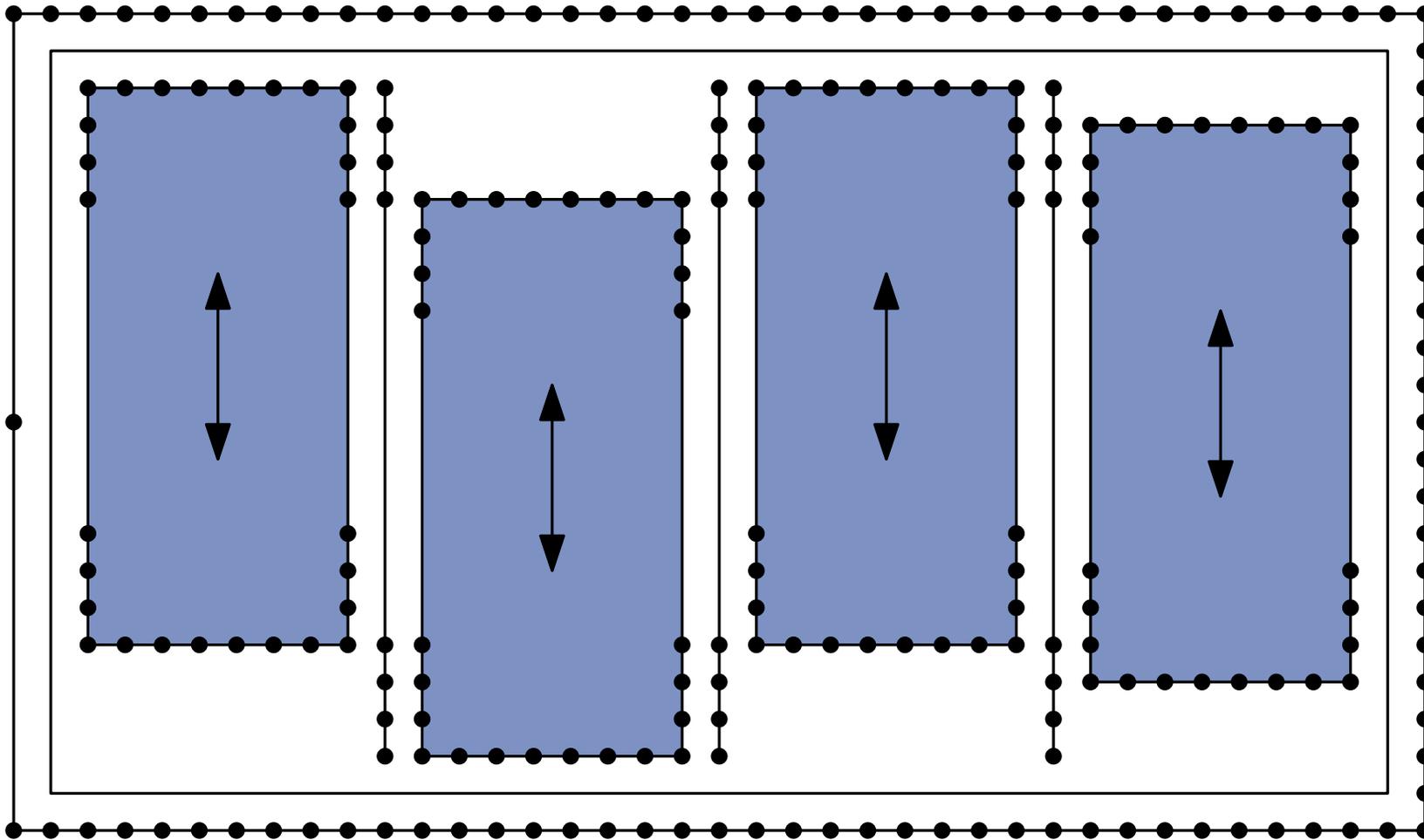
Variablengadget



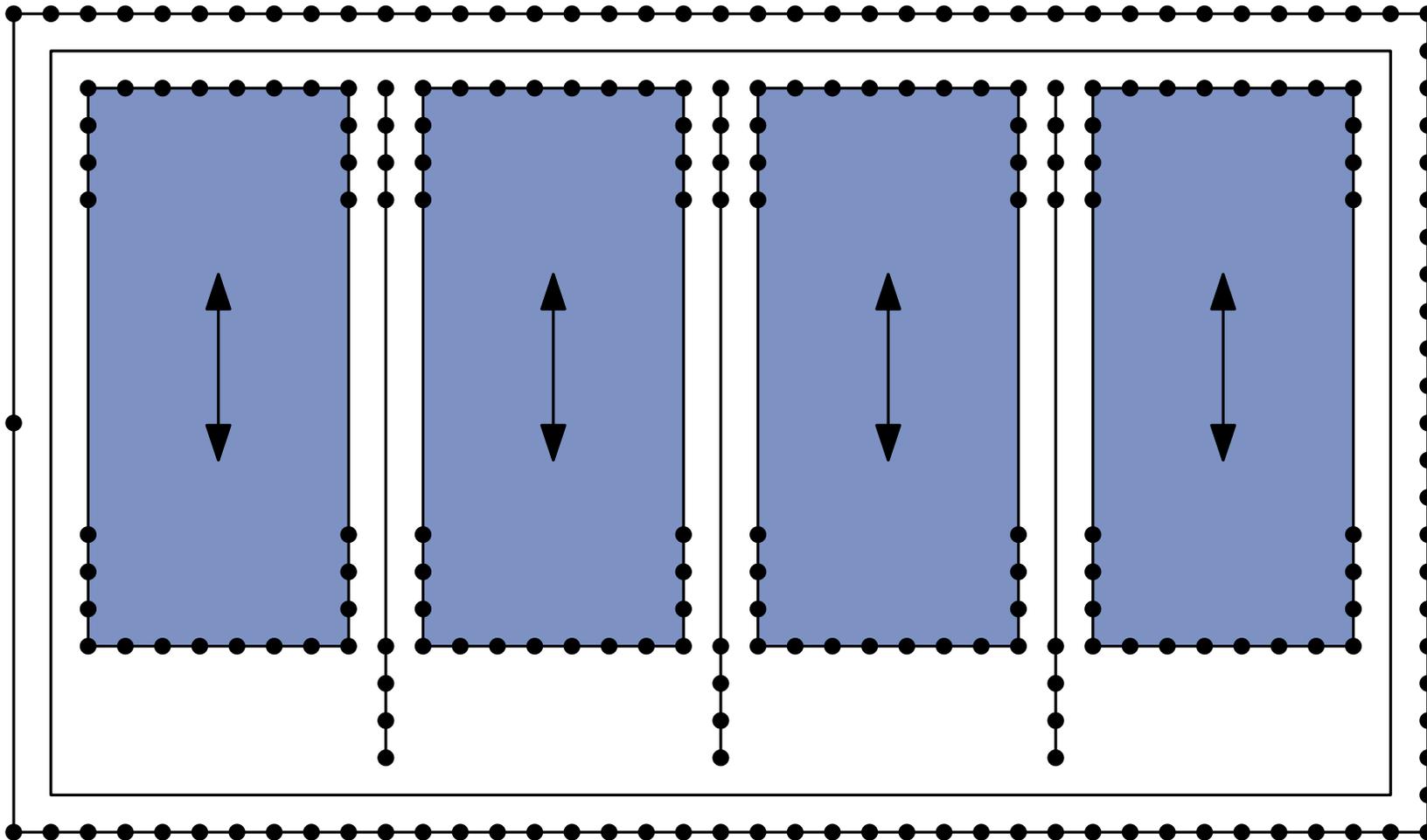
Variablengadget

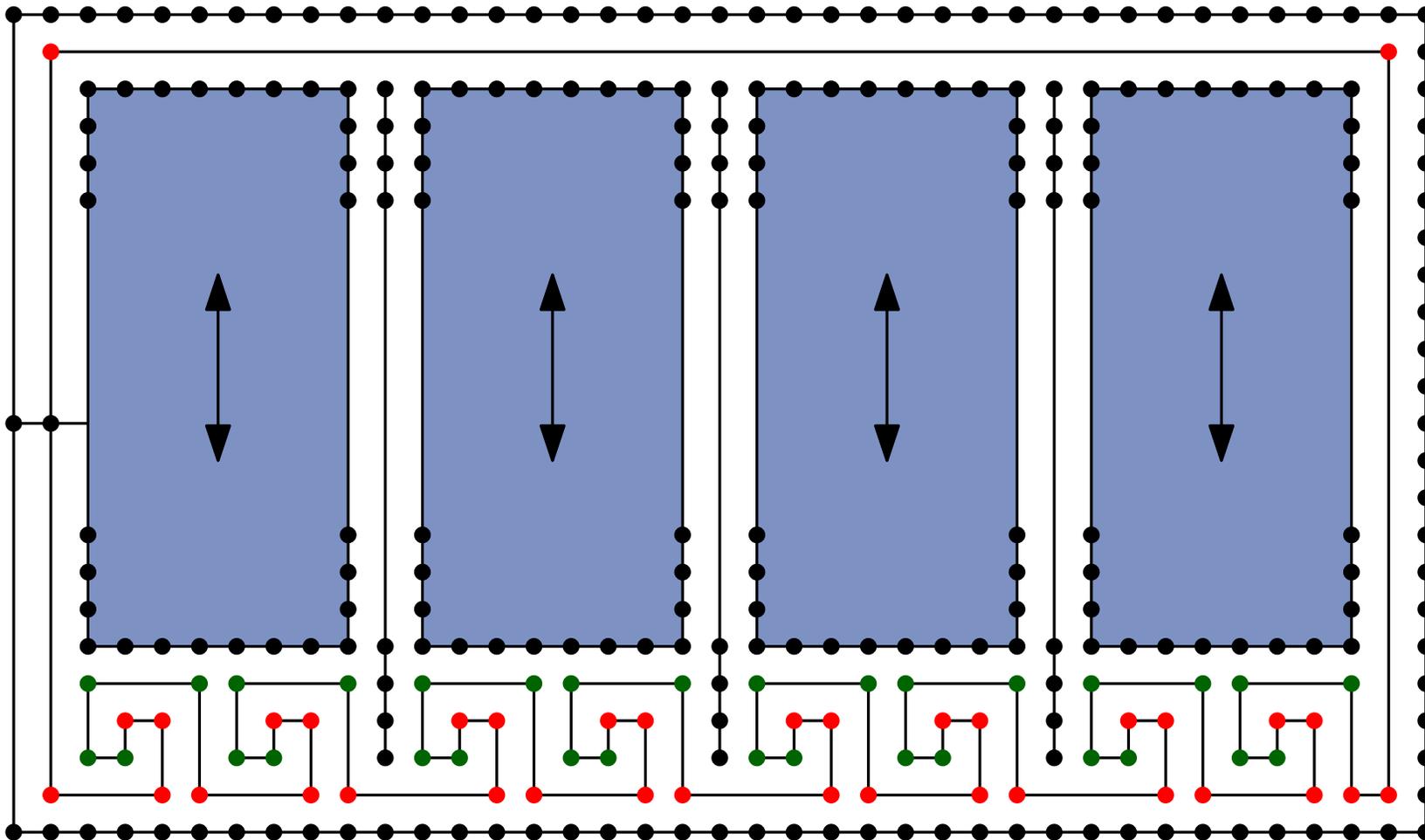


Variablengadget



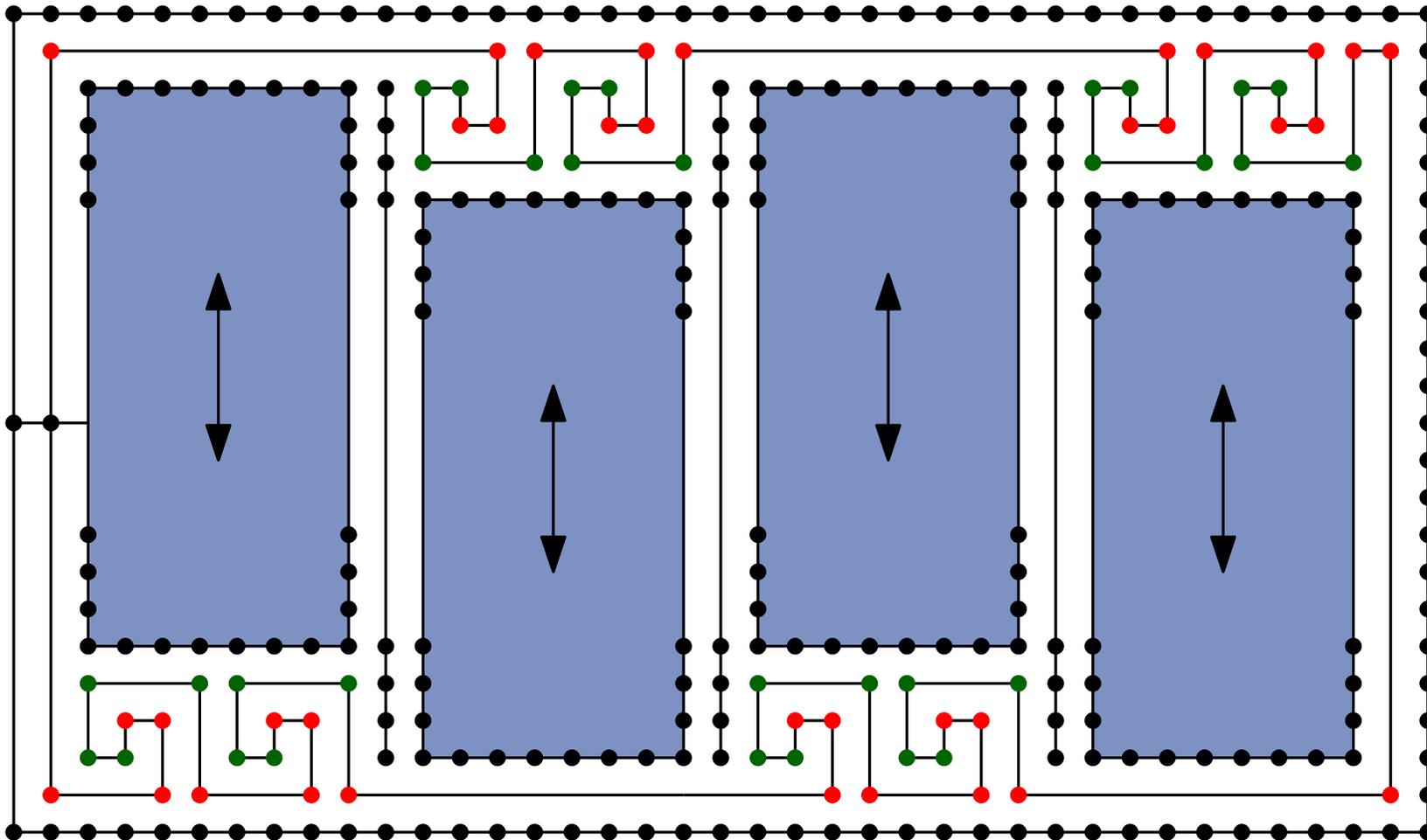
Variablengadget





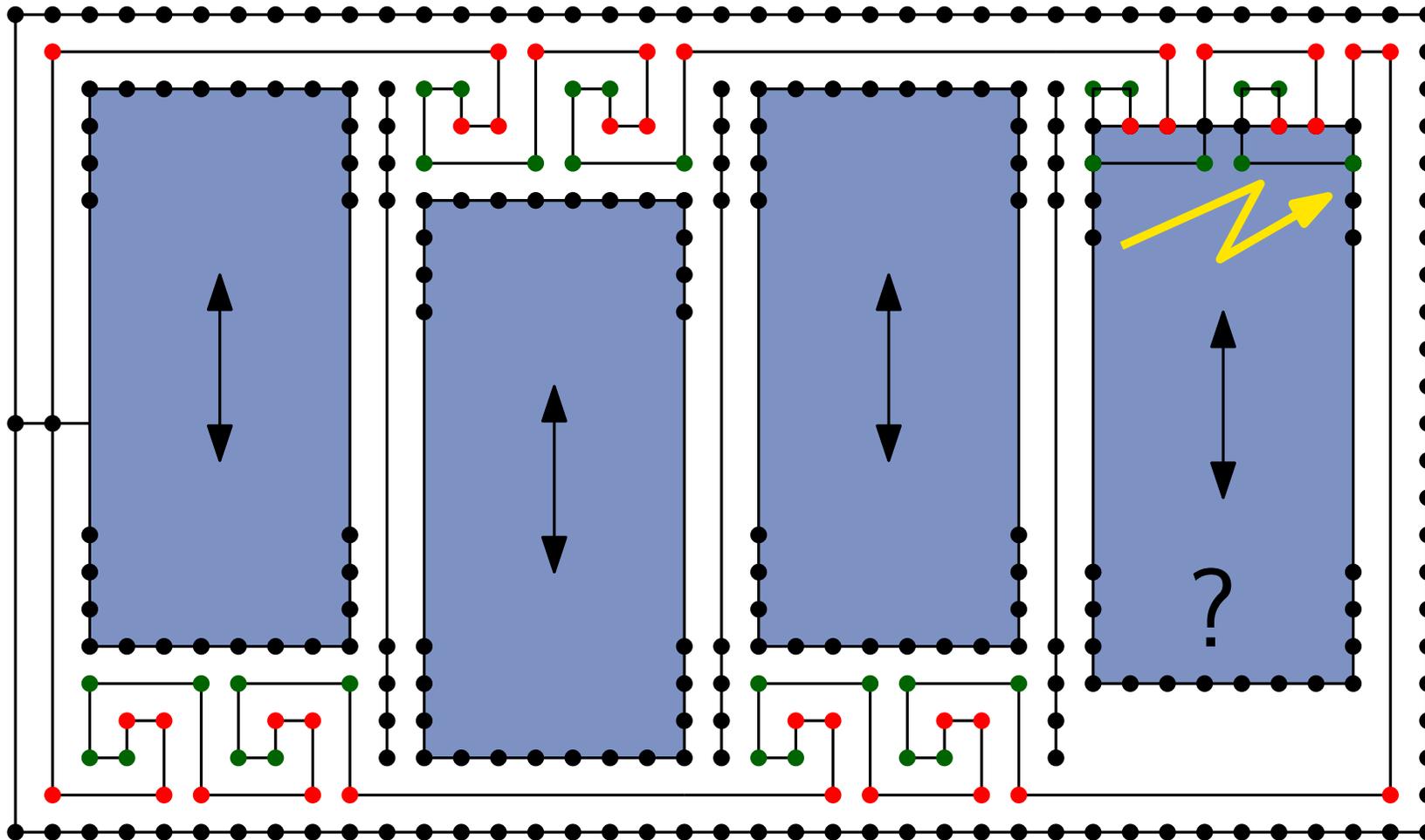
Gürtel ist eine Kante mit Beschreibung $(\underbrace{r^4}_{\text{red}} \underbrace{l^4}_{\text{green}})^{2n} \underbrace{r^4}_{\text{red}}$

- Gürtelzellen lassen sich innerhalb des Rahmens verschieben



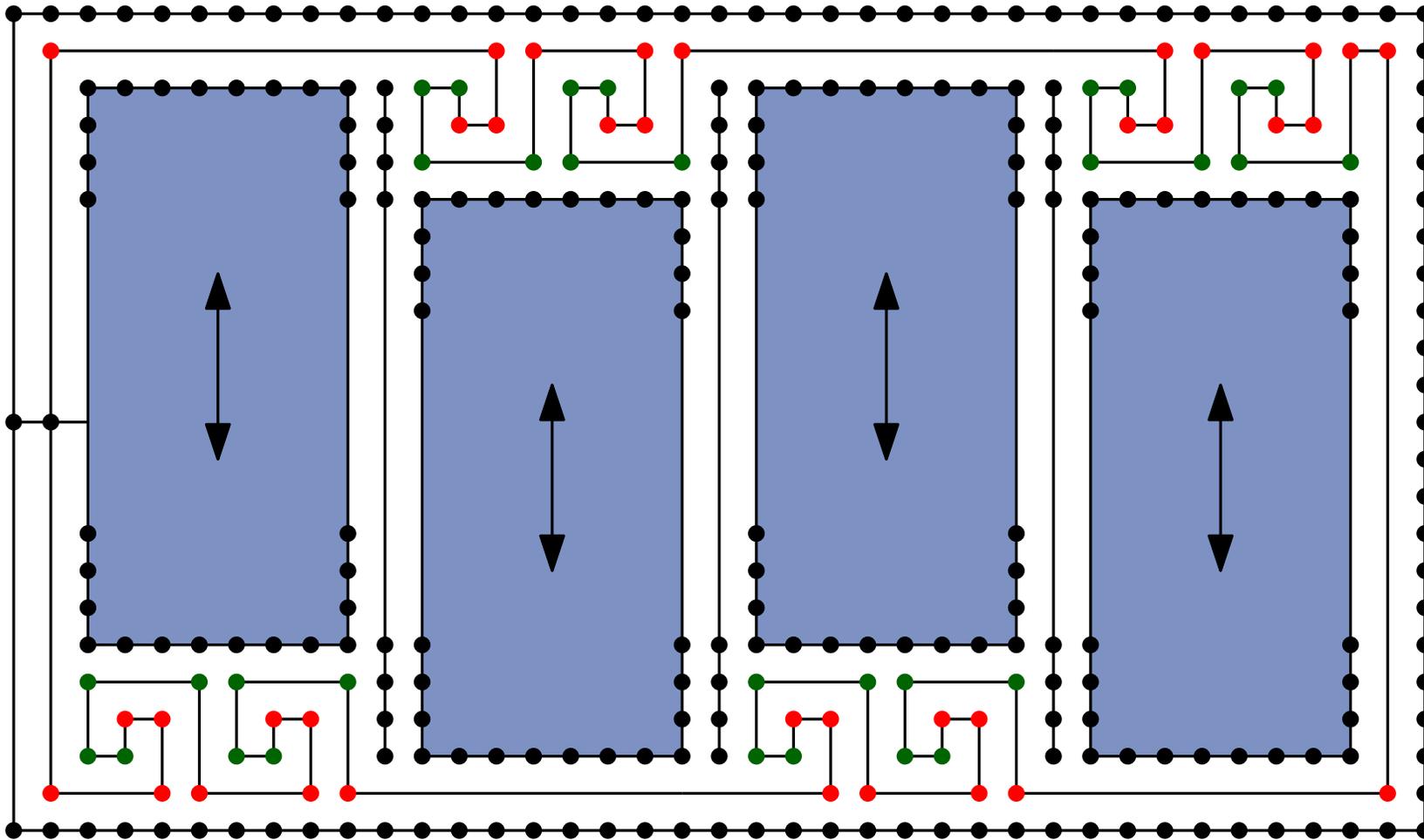
Gürtel ist eine Kante mit Beschreibung $(\overset{\text{red}}{\bullet\bullet\bullet}\overset{\text{green}}{\bullet\bullet\bullet})^{2n}\overset{\text{red}}{\bullet\bullet\bullet}r^4$

- Gürtelzellen lassen sich innerhalb des Rahmens verschieben
- jeder Kolben entspricht einer Variablen mit zwei Belegungen



Gürtel ist eine Kante mit Beschreibung $(\overset{\text{red}}{\bullet\bullet\bullet}\overset{\text{green}}{\bullet\bullet\bullet})^{2n}\overset{\text{red}}{\bullet\bullet\bullet}r^4$

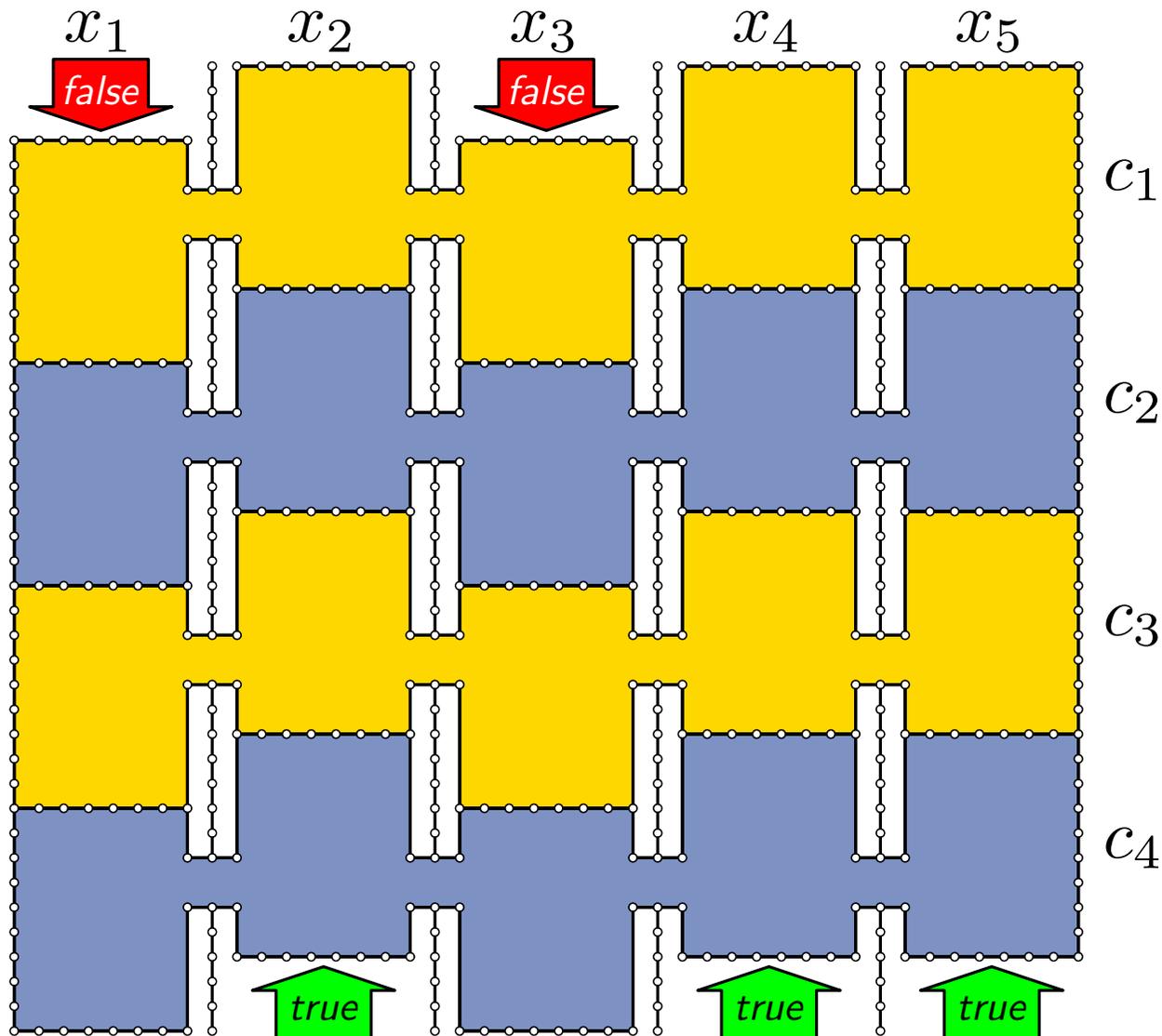
- Gürtelzellen lassen sich innerhalb des Rahmens verschieben
- jeder Kolben entspricht einer Variablen mit zwei Belegungen



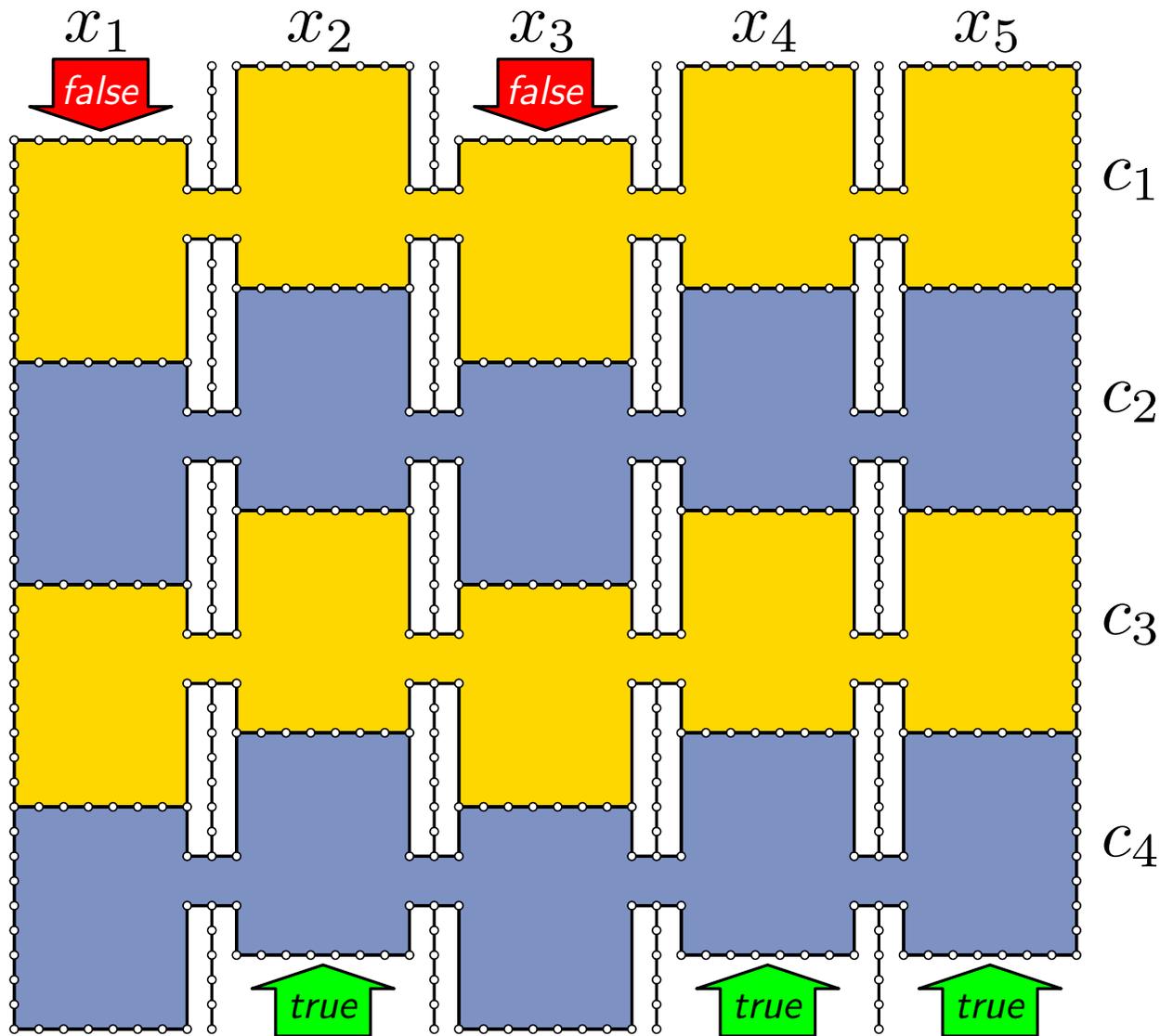
Gürtel ist eine Kante mit Beschreibung $(\overset{\text{red}}{\bullet\bullet\bullet}\overset{\text{green}}{\bullet\bullet\bullet})^{2n}\overset{\text{red}}{\bullet\bullet\bullet}r^4$

- Gürtelzellen lassen sich innerhalb des Rahmens verschieben
- jeder Kolben entspricht einer Variablen mit zwei Belegungen

Klauselgadgets



Klauselgadgets



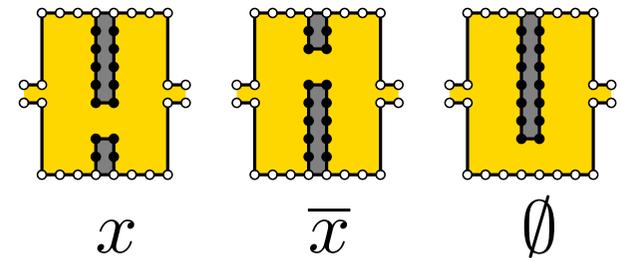
Beispiel:

$$c_1 = x_2 \vee \overline{x_4}$$

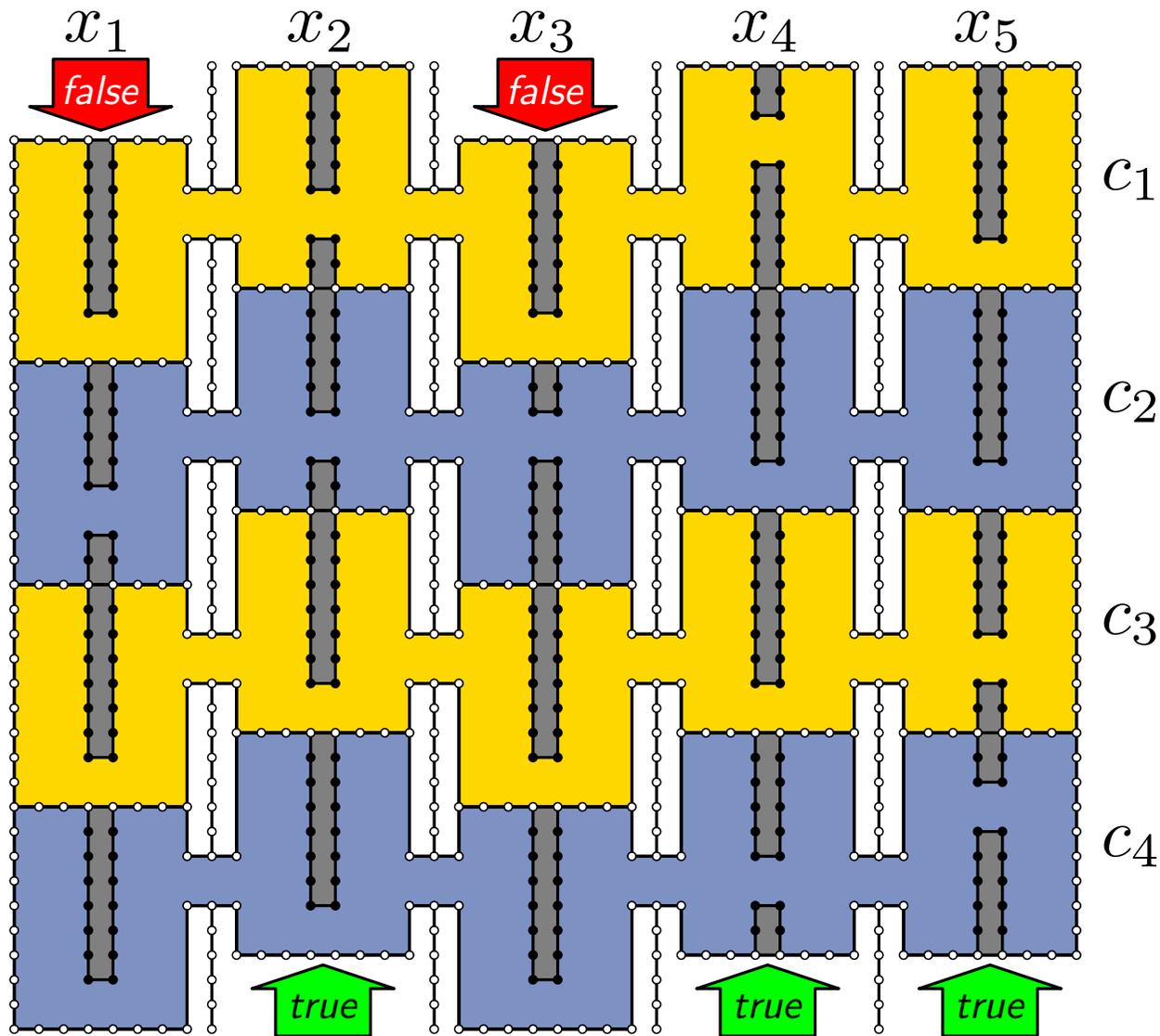
$$c_2 = x_1 \vee x_2 \vee \overline{x_3}$$

$$c_3 = x_5$$

$$c_4 = x_4 \vee \overline{x_5}$$



Klauselgadgets



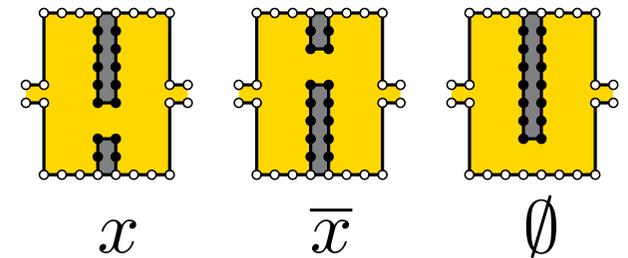
Beispiel:

$$c_1 = x_2 \vee \overline{x_4}$$

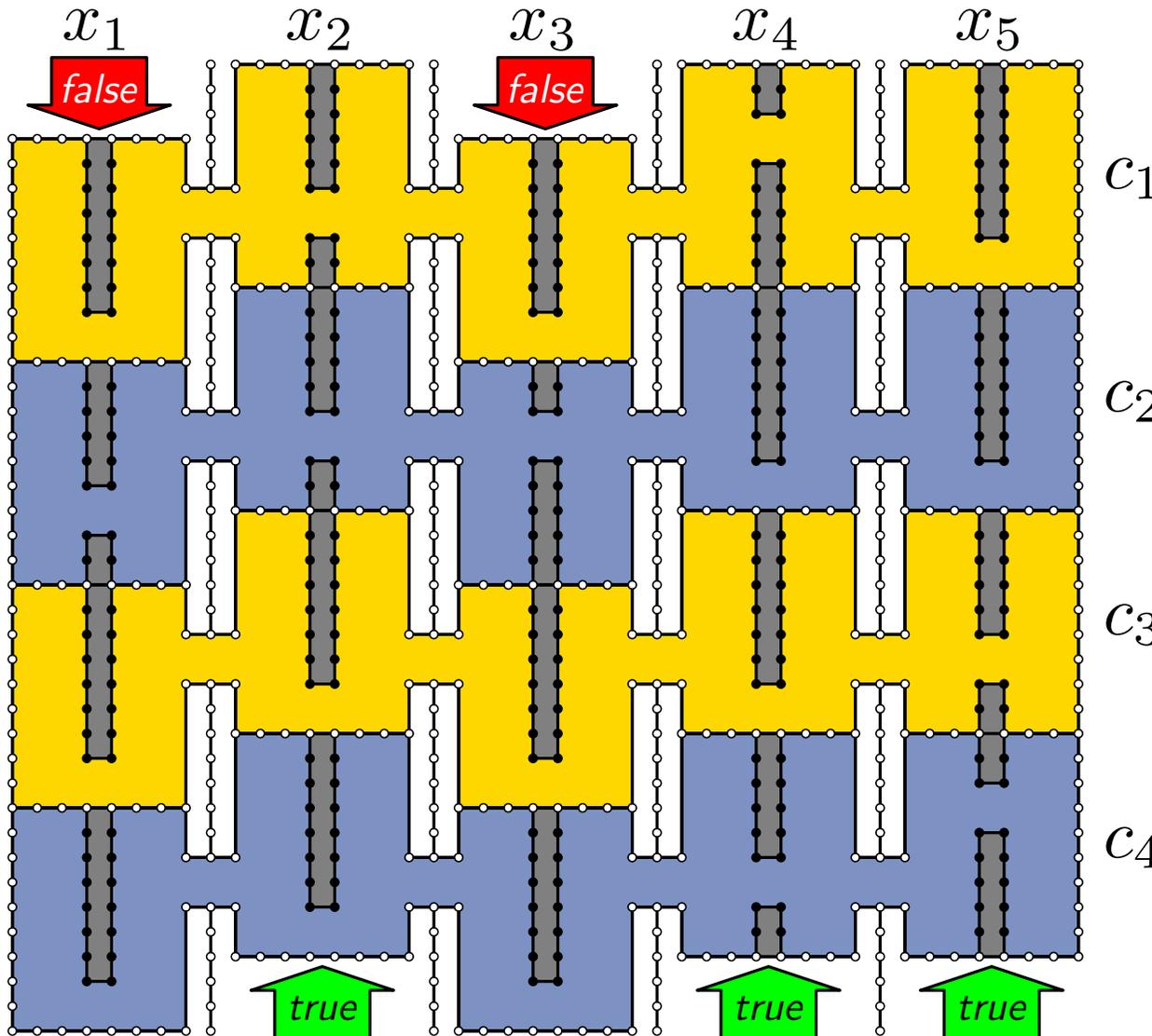
$$c_2 = x_1 \vee x_2 \vee \overline{x_3}$$

$$c_3 = x_5$$

$$c_4 = x_4 \vee \overline{x_5}$$



Klauselgadgets



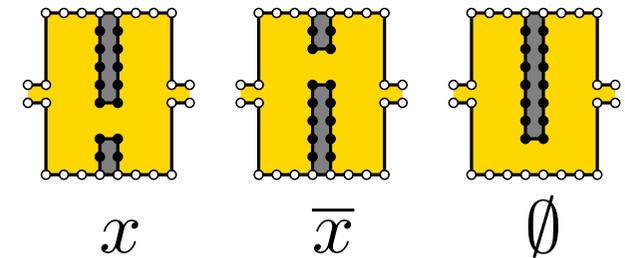
Beispiel:

$$c_1 = x_2 \vee \overline{x_4}$$

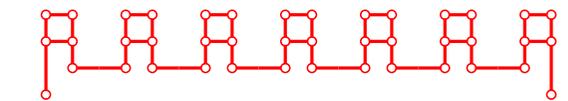
$$c_2 = x_1 \vee x_2 \vee \overline{x_3}$$

$$c_3 = x_5$$

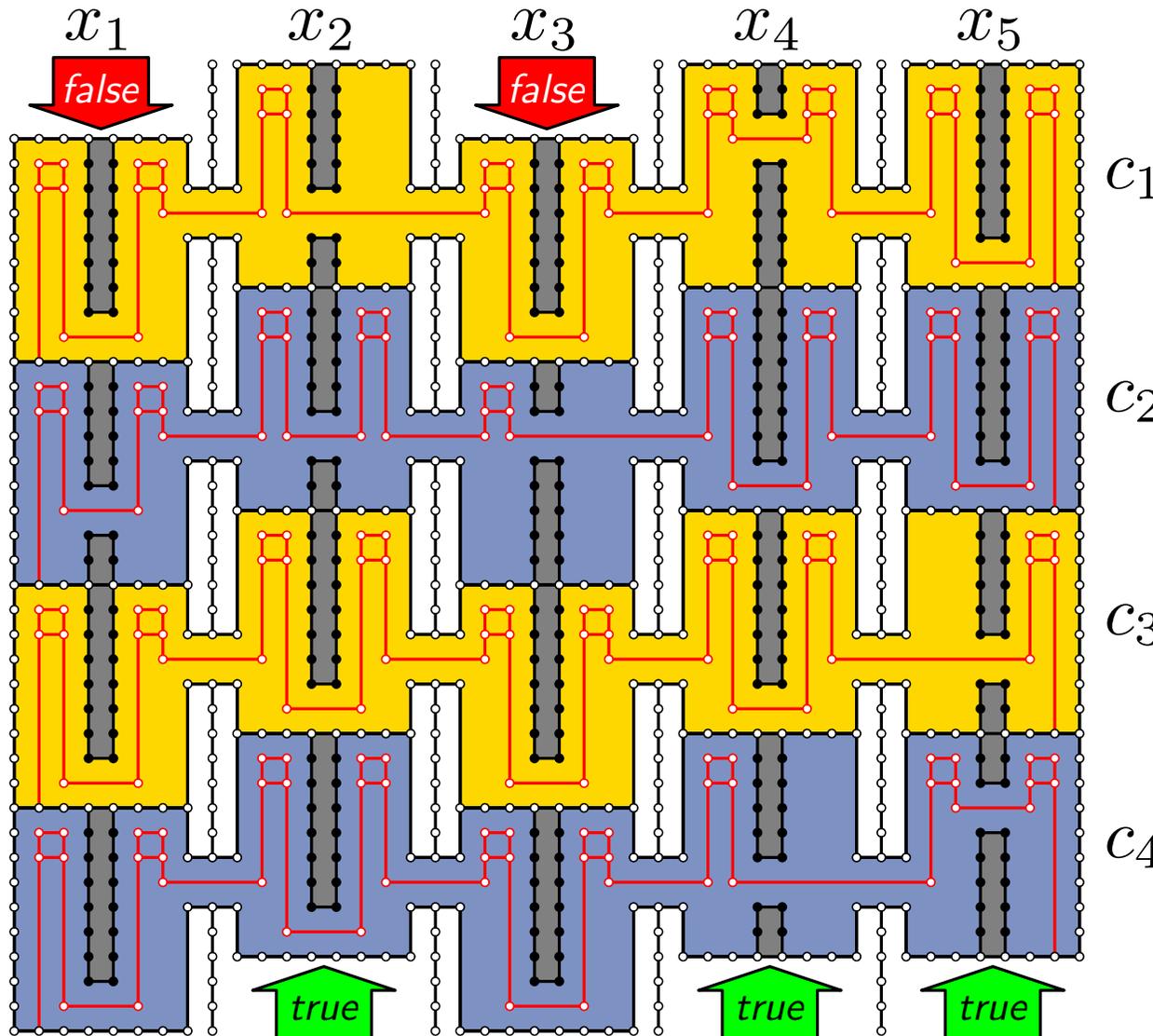
$$c_4 = x_4 \vee \overline{x_5}$$



lege A^{2n-1} -Kette durch jede Klausel



Klauselgadgets



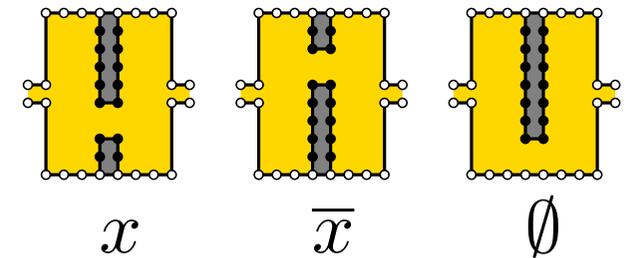
Beispiel:

$$c_1 = x_2 \vee \overline{x_4}$$

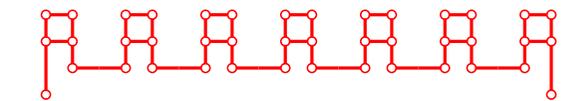
$$c_2 = x_1 \vee x_2 \vee \overline{x_3}$$

$$c_3 = x_5$$

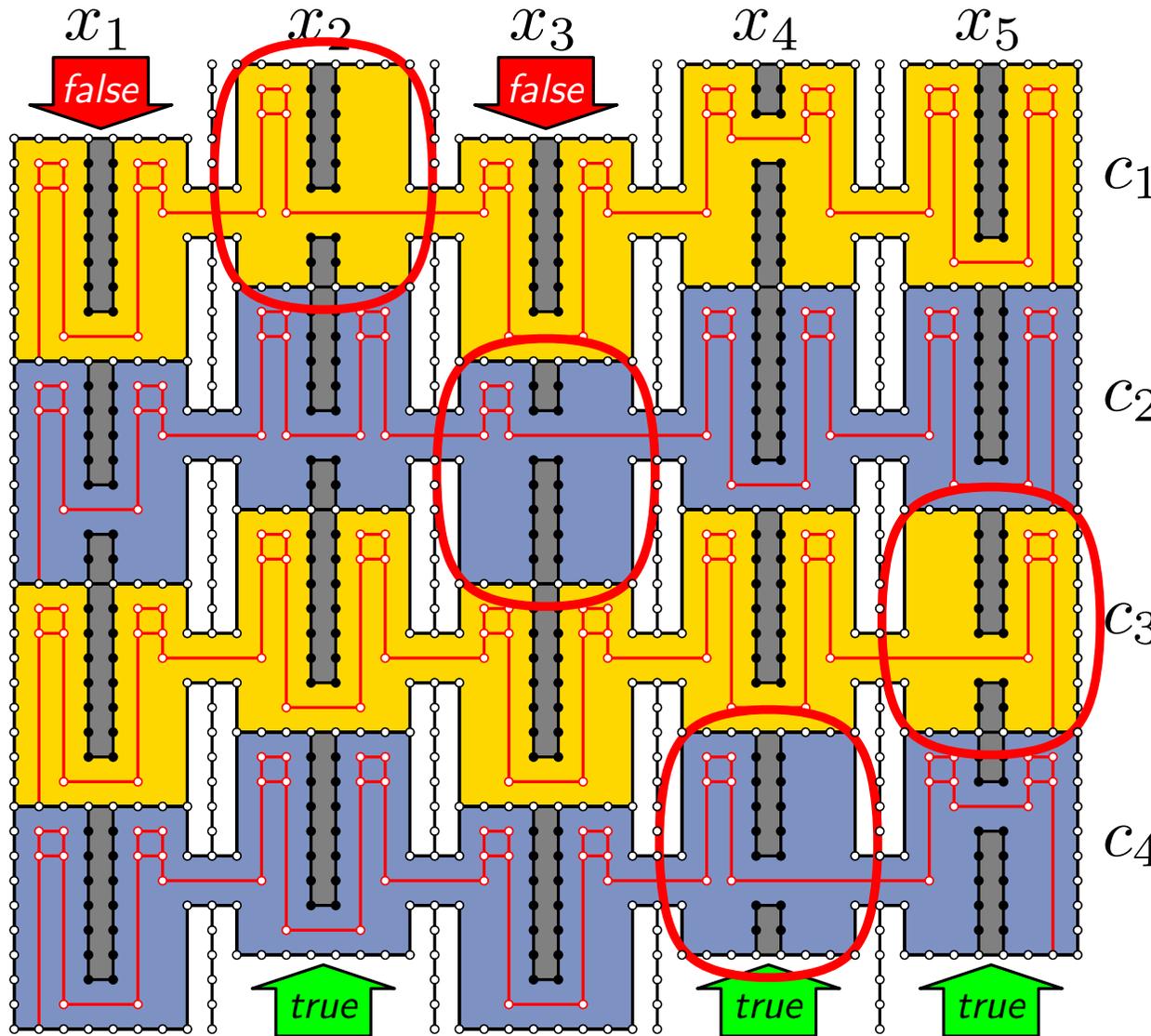
$$c_4 = x_4 \vee \overline{x_5}$$



lege A^{2n-1} -Kette durch jede Klausel



Klauselgadgets



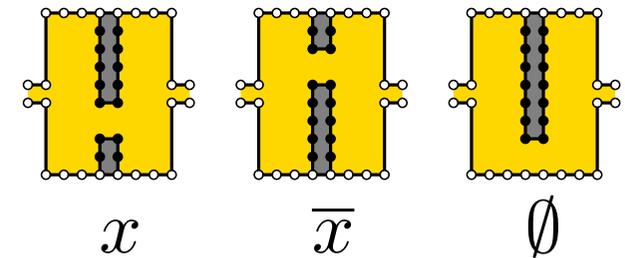
Beispiel:

$$c_1 = x_2 \vee \overline{x_4}$$

$$c_2 = x_1 \vee x_2 \vee \overline{x_3}$$

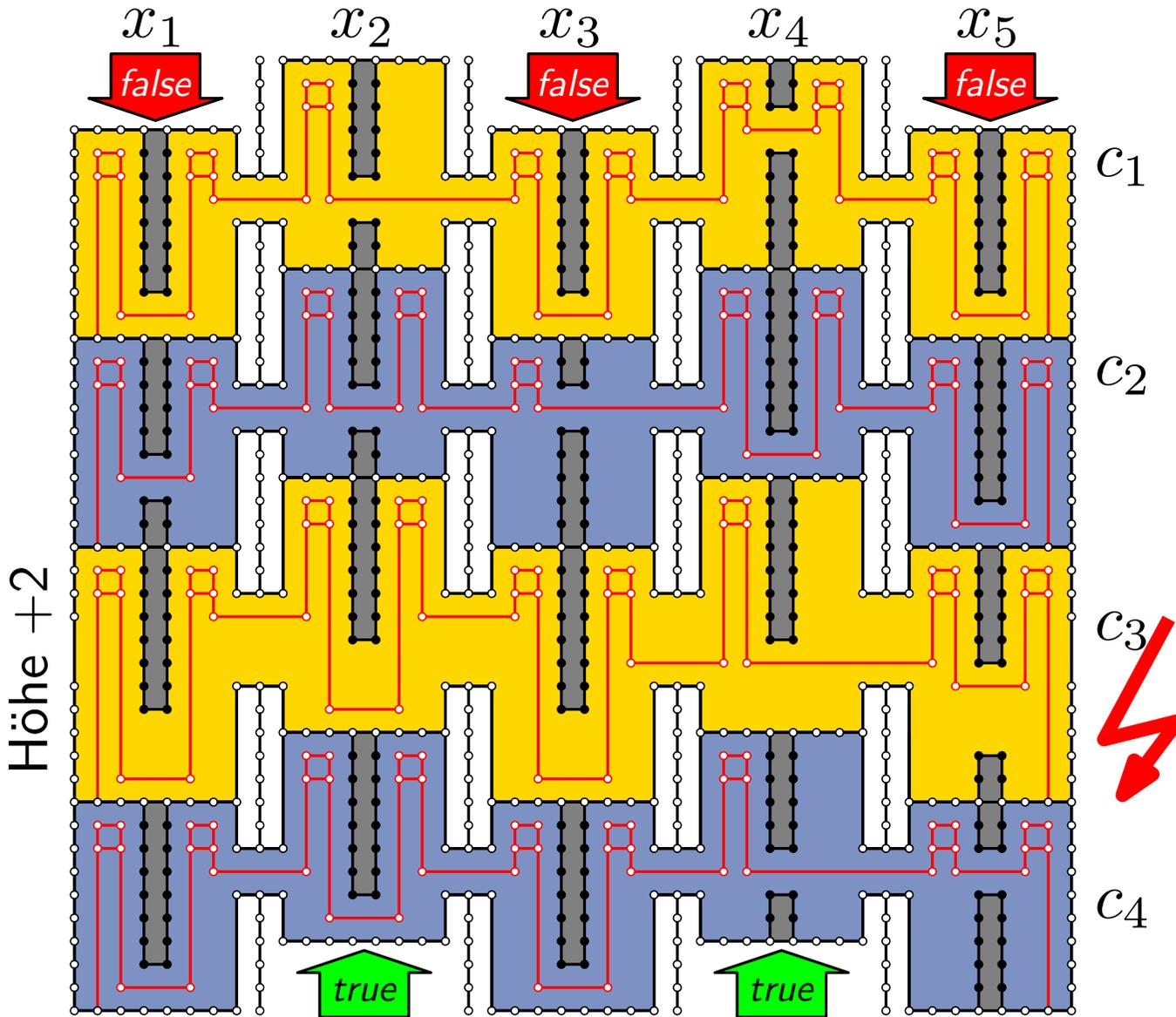
$$c_3 = x_5$$

$$c_4 = x_4 \vee \overline{x_5}$$



lege A^{2n-1} -Kette durch jede Klausel

Klauselgadgets



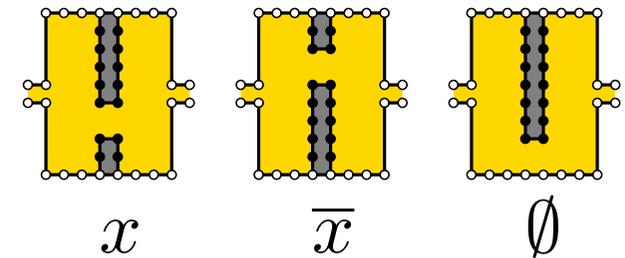
Beispiel:

$$c_1 = x_2 \vee \bar{x}_4$$

$$c_2 = x_1 \vee x_2 \vee \bar{x}_3$$

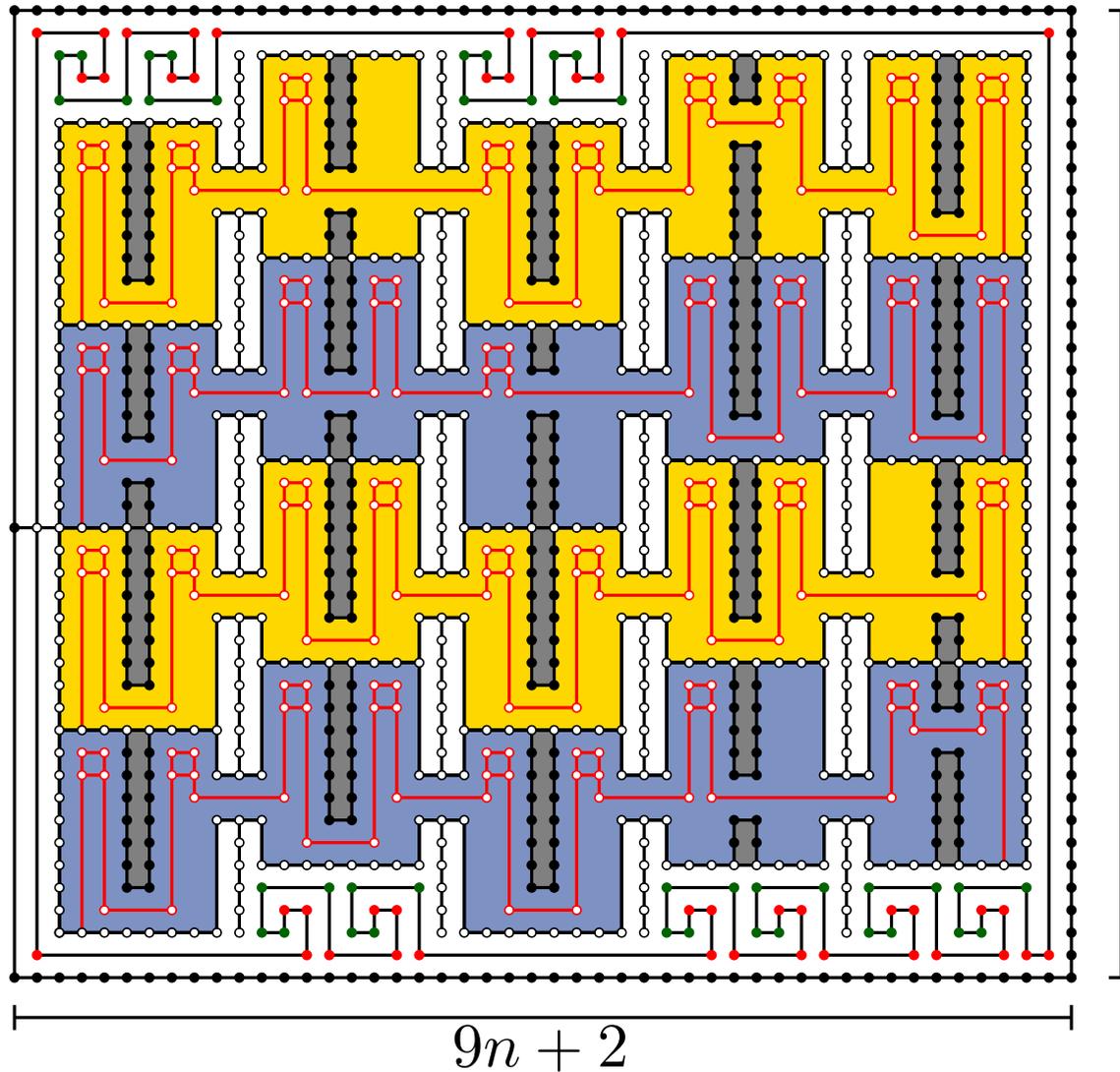
$$c_3 = x_5$$

$$c_4 = x_4 \vee \bar{x}_5$$



lege A^{2n-1} -Kette durch jede Klausel

Komplette Reduktion



Setze

$$K = (9n + 2) \cdot (9m + 7)$$

$$9m + 7$$

Es gilt:

(G, H) auf Fläche K
zeichenbar

\Leftrightarrow

Φ erfüllbar

□

- Flächenkompaktierung bei gegebener orthogonaler Beschreibung ist im Allgemeinen NP-schwer
- falls alle Facetten Rechtecke sind ist Kompaktierung durch Flussmodell effizient möglich
- heuristisch durch Verfeinerung nicht-rechteckiger Facetten zu Rechtecken lösbar

- Flächenkompaktierung bei gegebener orthogonaler Beschreibung ist im Allgemeinen NP-schwer
- falls alle Facetten Rechtecke sind ist Kompaktierung durch Flussmodell effizient möglich
- heuristisch durch Verfeinerung nicht-rechteckiger Facetten zu Rechtecken lösbar
- im allgemeinen Fall durch ganzzahlige lineare Programmierung (ILP) lösbar [Klau, Mutzel IPCO 1999]
- experimenteller Vergleich verschiedener heuristischer Methoden bzgl. Laufzeit und Qualität [Klau, Klein, Mutzel GD 2001]
- für nicht-planare Graphen sogar schwer zu approximieren [Bannister, Eppstein, Simons JGAA 2012]

(Planare) Orthogonale Zeichnungen

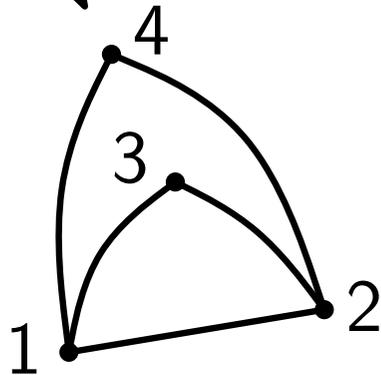
Dreistufiger Ansatz: *Topology – Shape – Metrics*

$$V = \{v_1, v_2, v_3, v_4\}$$

$$E = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_4\}$$

Kreuzungsminimierung

kombinatorische
Einbettung/
Planarisierung



Knickminimierung

orthogonale
Beschreibung

planare
Einbettung

Flächen-
minimierung

