Vorlesung Algorithmische Kartografie
Vereinfachung und Schematisierung von Polygonen

INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Benjamin Niedermann · Martin Nöllenburg

28.04.2015

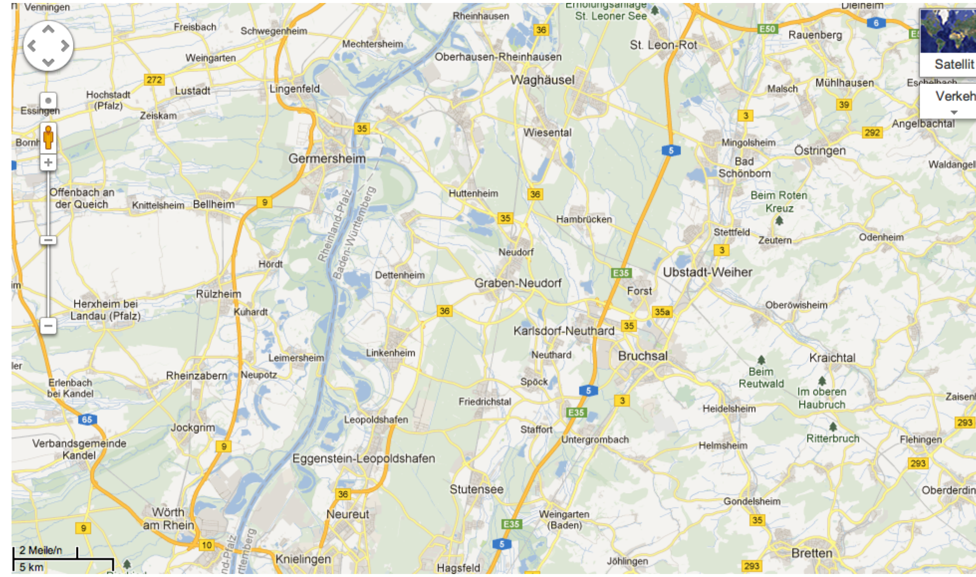
Übersicht

Ebene	Block	Termine	Themen
Karteninhalt	Linienzüge	3	Einführung
			Linienvereinfachung
			Konsistente Vereinfachung von Kantenzügen
	Flächen	5-6	Vereinfachung/Schematisierung von Polygonen
			Flächenaggregation
			Flächenkartogramme
Beschriftung	Statisch	2	Punktbeschriftung
			Straßenbeschriftung
	Dynamisch	6	Zoomen
			Rotieren
			Trajektorienbasierte Beschriftung
Geovisualisierung	Annotation	3	Beschriftung von bewegten Punkten
			Randbeschriftung
	Diagramme	1	Proportional Symbol Maps

Bisher: Linienvereinfachung

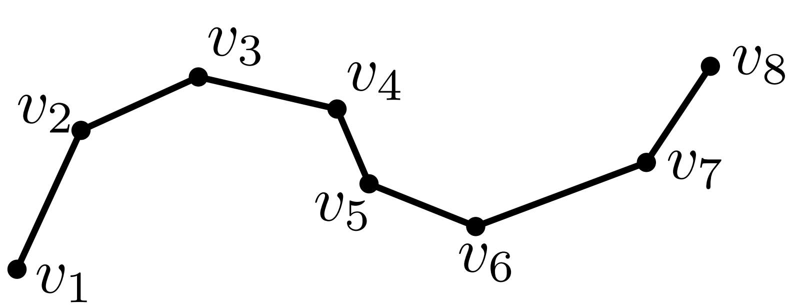
Karlsruhe Institute of Technology

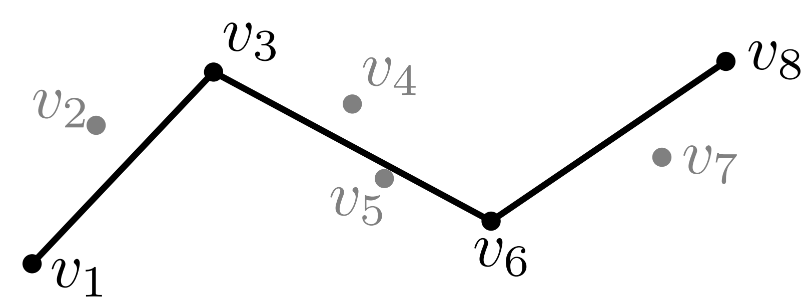
- Karten bestehen zum Großteil aus Linien-/ Polygonzügen
- Linienvereinfachung reduziert Komplexität: visuell und Speicherplatz



geg: Polygonzug $P = (v_1, v_2, \ldots, v_n)$

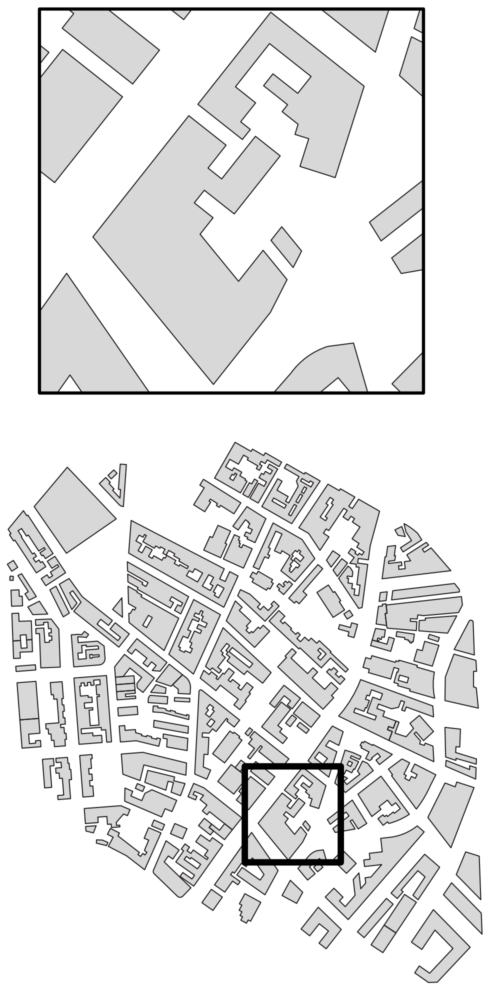
ges: Polygonzug $Q = (v_1 = v_{i_1}, v_{i_2}, v_{i_3}, \ldots, v_{i_k} = v_n)$ mit $i_1 < i_2 < \cdots < i_k$, so dass Q eine gute Approximation von P und k möglichst klein ist





Heute: Generalisierung von Polygonen

Verfahren: Vereinfachung.

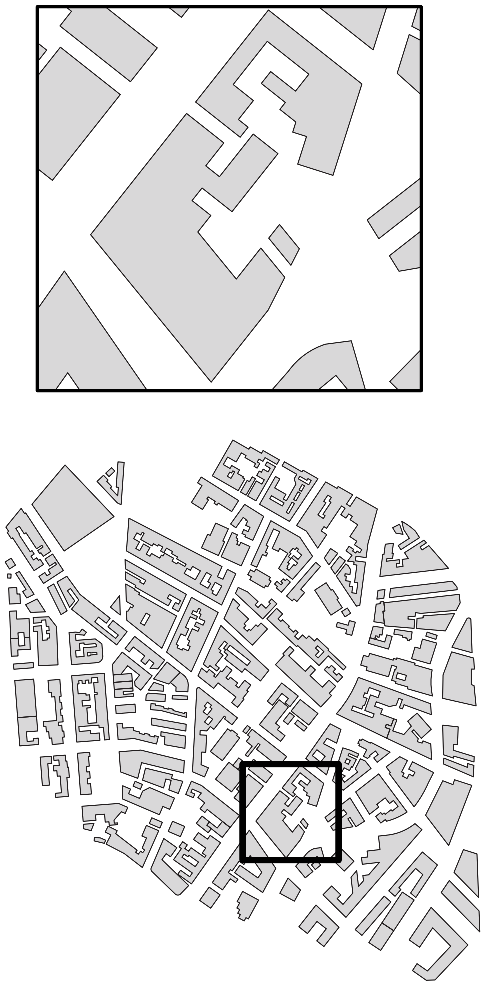


Original

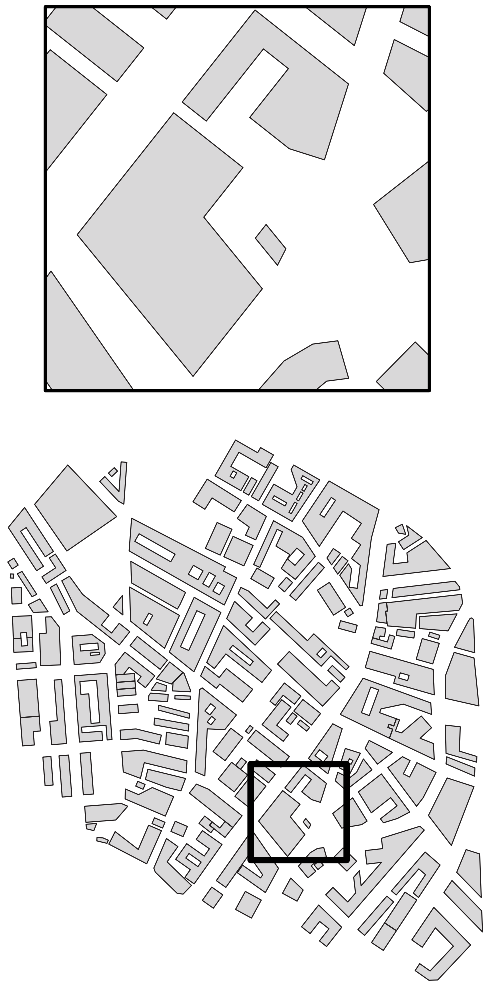
Heute: Generalisierung von Polygonen

Verfahren: Vereinfachung.

Anwendung: Vereinfachung von Gebäudeumrissen.



Original



Vereinfachung

Heute: Generalisierung von Polygonen

Verfahren: Vereinfachung.

Anwendung: Vereinfachung von Gebäudeumrissen.

Bisher:
Vereinfachung mit kleinem Fehler.

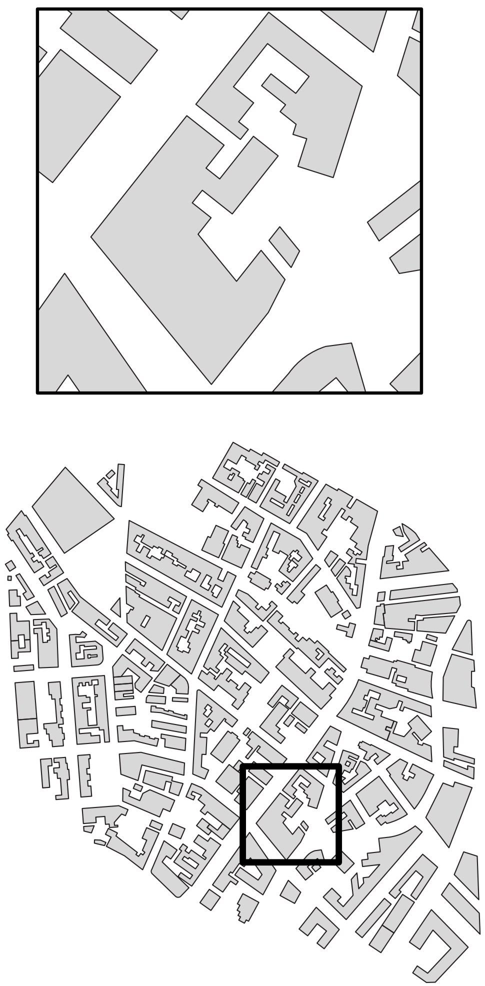
Original

Vereinfachung

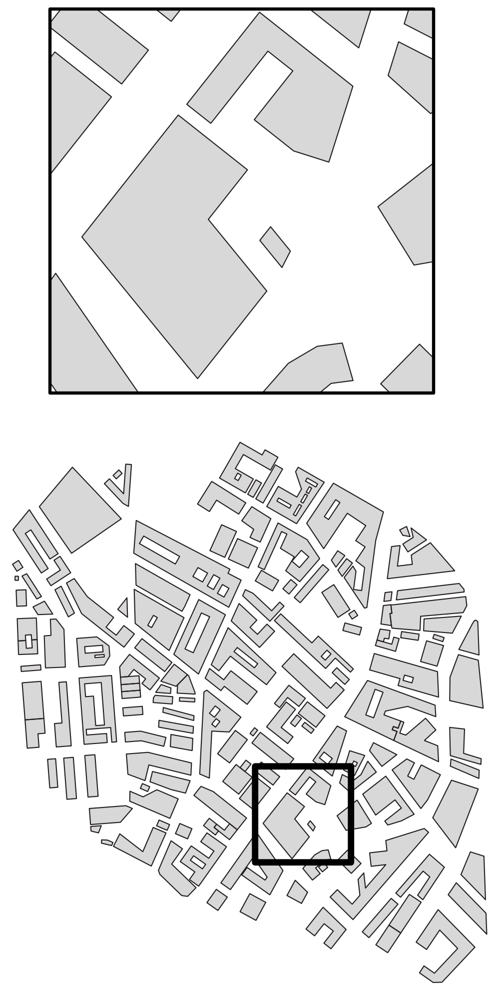
Heute: Generalisierung von Polygonen

Verfahren: Vereinfachung.

Anwendung: Vereinfachung von Gebäudeumrissen.



Original



Vereinfachung

Bisher:
Vereinfachung mit kleinem Fehler.

Jetzt:
Vereinfachung, sodass Flächeninhalt erhalten bleibt.

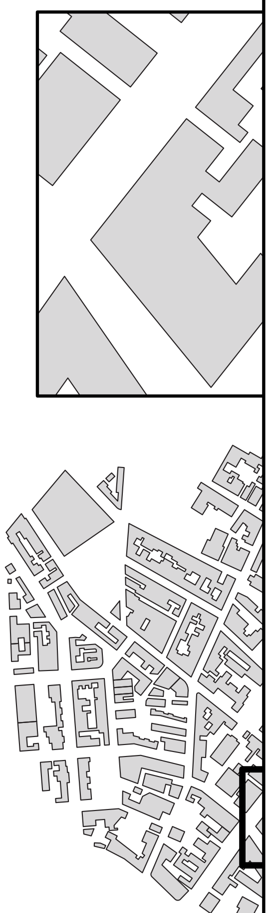
Heute: Generalisierung von Polygonen

Verfahren: Vereinfachung.

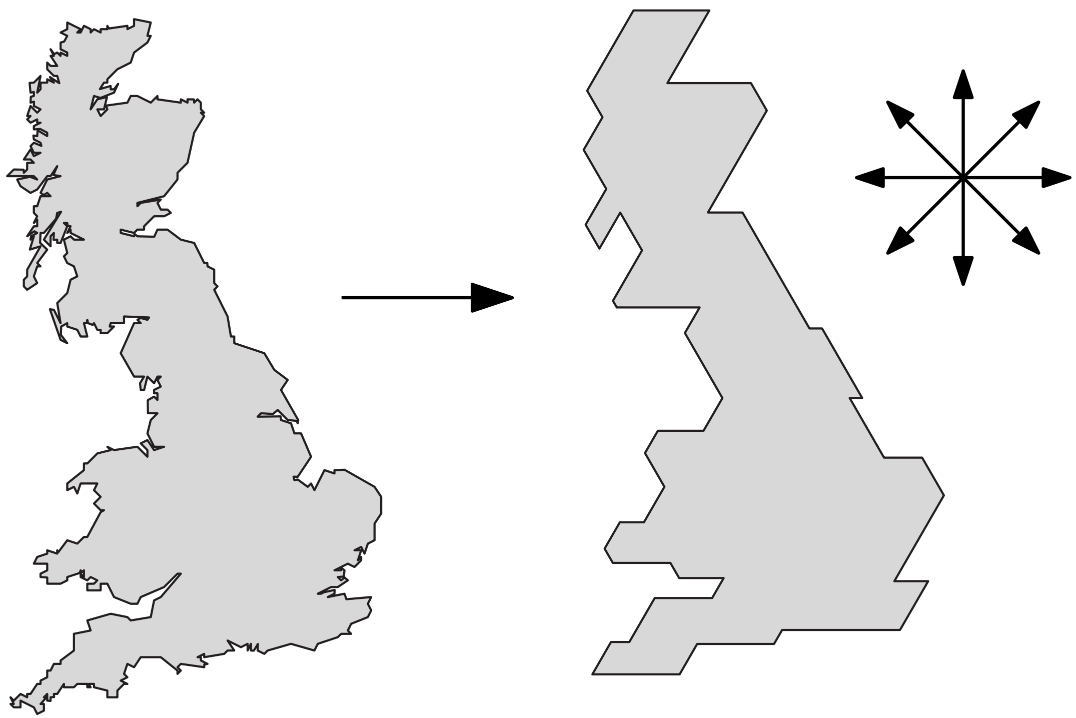
Anwendun

ung mit
hler.

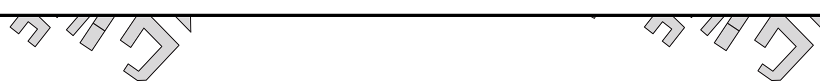
ung, sodass
alt erhalten



Im Anschluss: Schematisierung



Schematisierung: Einschränkung der möglichen Kantenrichtungen

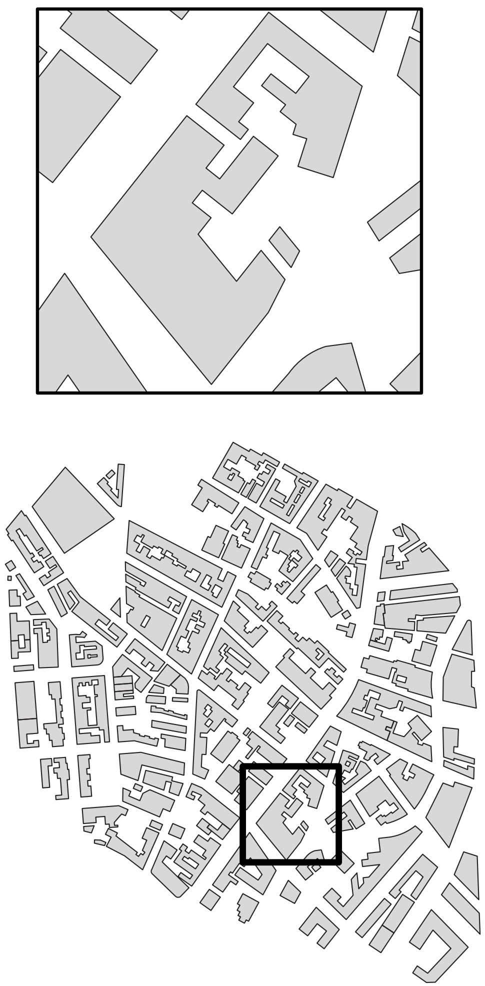


Original Vereinfachung

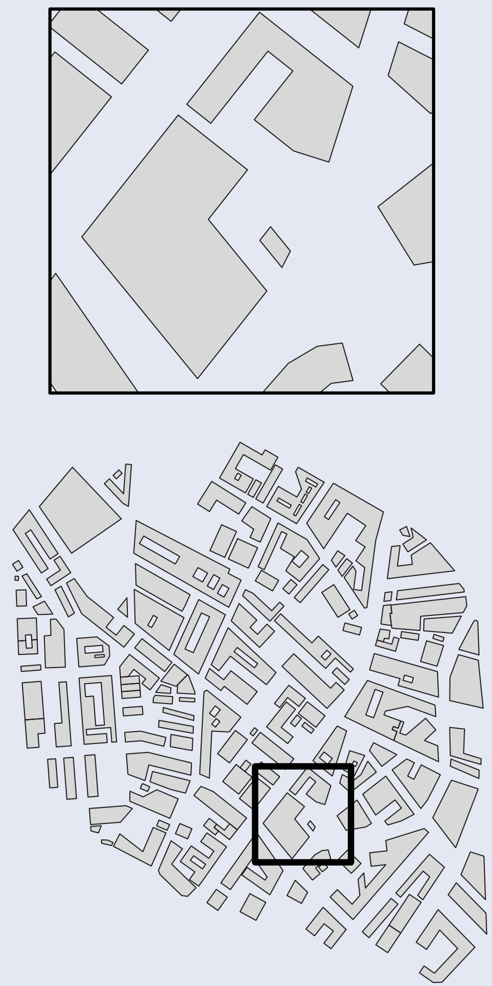
Heute: Generalisierung von Polygonen

Verfahren: Vereinfachung.

Anwendung: Vereinfachung von Gebäudeumrissen.



Original



Vereinfachung

Bisher:
Vereinfachung mit kleinem Fehler.

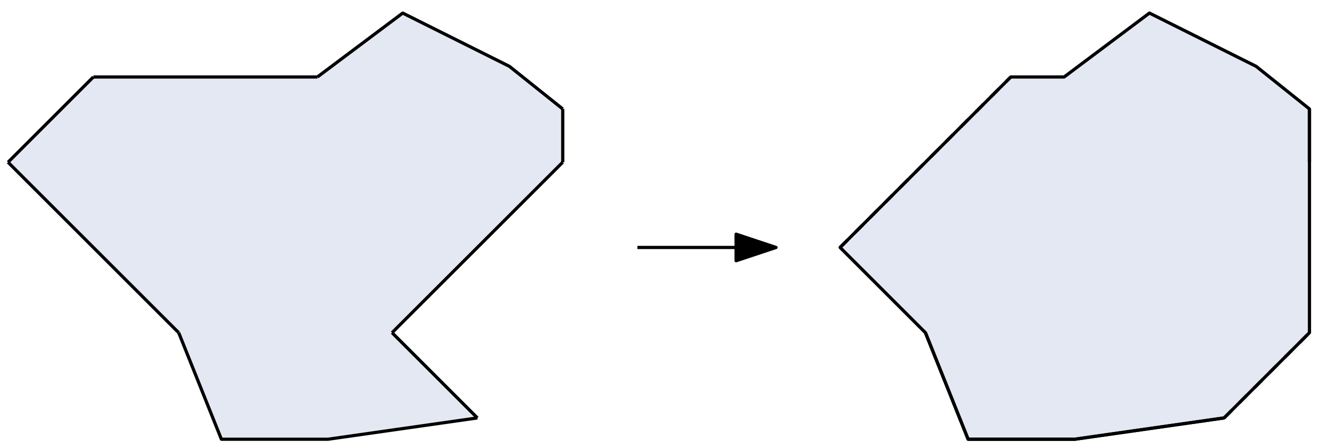
Jetzt:
Vereinfachung, sodass Flächeninhalt erhalten bleibt.

Formale Problemstellung

Problemstellung:

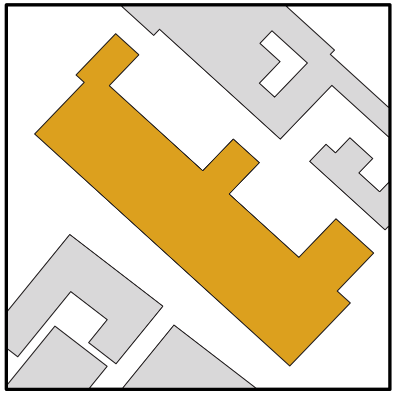
geg.: Einfaches Polygon P und Parameter $k \in \mathbb{N}$

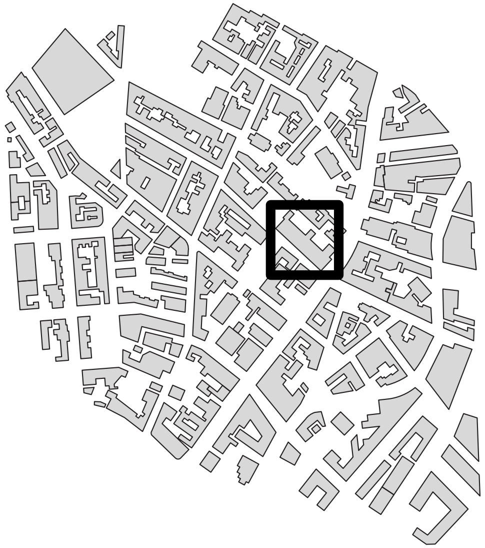
ges.: Einfaches Polygon P', sodass
- P und P' denselben Flächeninhalt besitzen, und
- die Kanten aus P' dieselben Orientierungen wie die in P verwenden,
- P' maximal k Kanten besitzt .



Vereinfachung des Problems

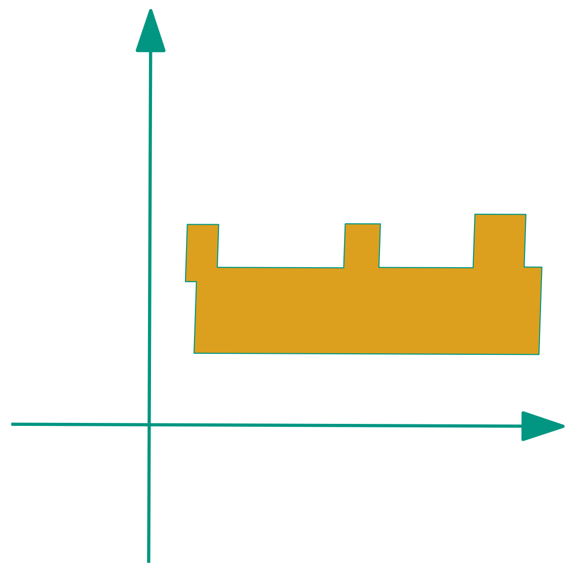
Beobachtung: Gebäudeumrisse häufig rechtwinklig





Vereinfachung des Problems

Beobachtung: Gebäudeumrisse häufig rechtwinklig



Vereinfachende Annahme:
Gegebenes Polygon ist rektilinear.
⟶ Kanten sind horizontal oder vertikal.

Vereinfachung von rektilinearen Polygonen

Karlsruhe Institute of Technology

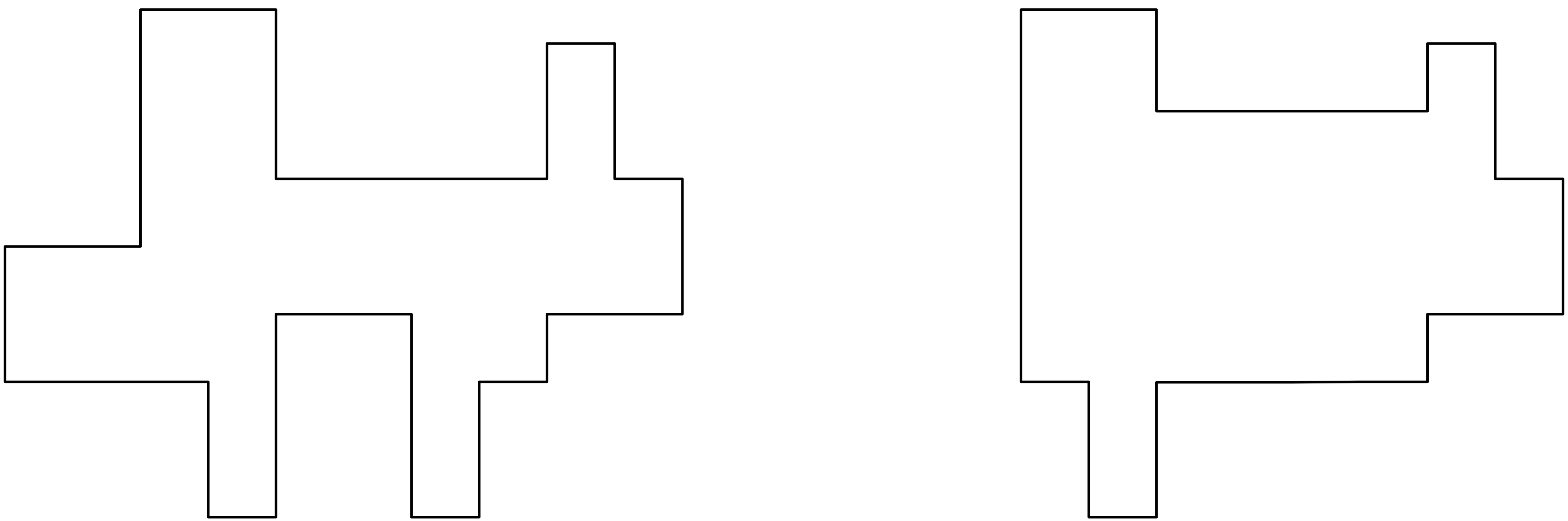
Problemstellung:

geg.: Einfaches rektilineares Polygon P und Parameter $k \in \mathbb{N}$

geg.: Einfaches rektilineares Polygon P', sodass

- P und P' denselben Flächeninhalt besitzen, und
- P' maximal k Kanten besitzt.

Beispiel:



Vereinfachung von rektilinearen Polygonen

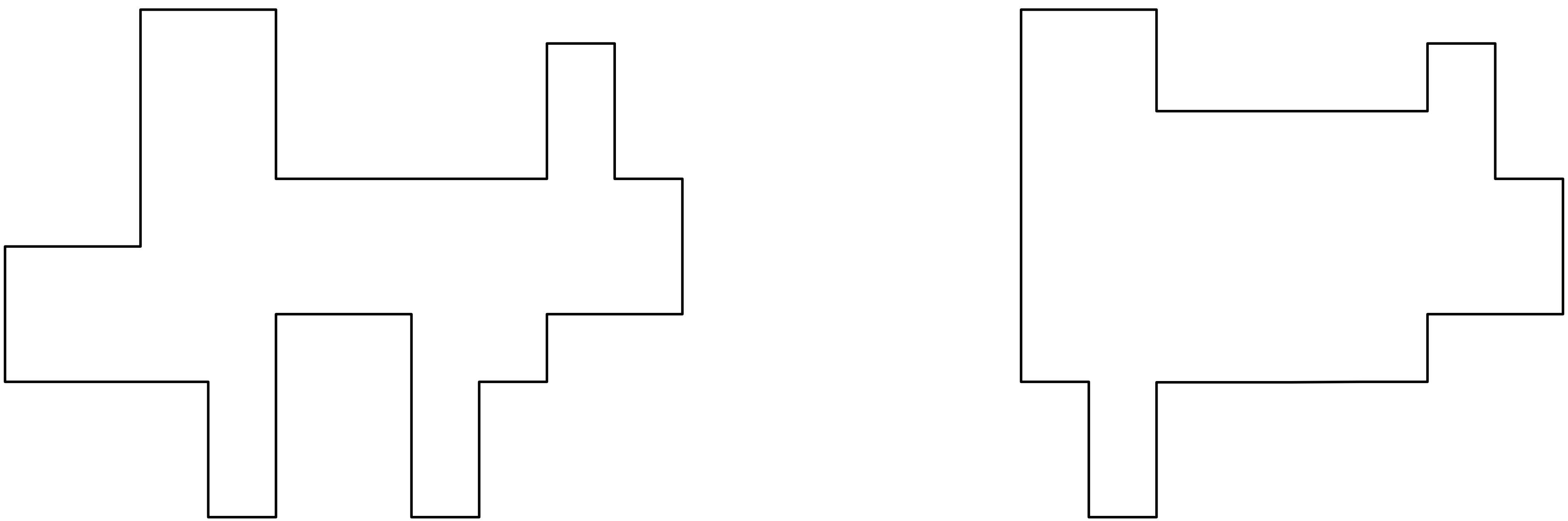
Problemstellung:

geg.: Einfaches rektilineares Polygon P und Parameter $k \in \mathbb{N}$

geg.: Einfaches rektilineares Polygon P', sodass

- P und P' denselben Flächeninhalt besitzen, und
- P' maximal k Kanten besitzt.

Beispiel:



Ideen für möglichen Algorithmus?

Vereinfachung von rektilinearen Polygonen

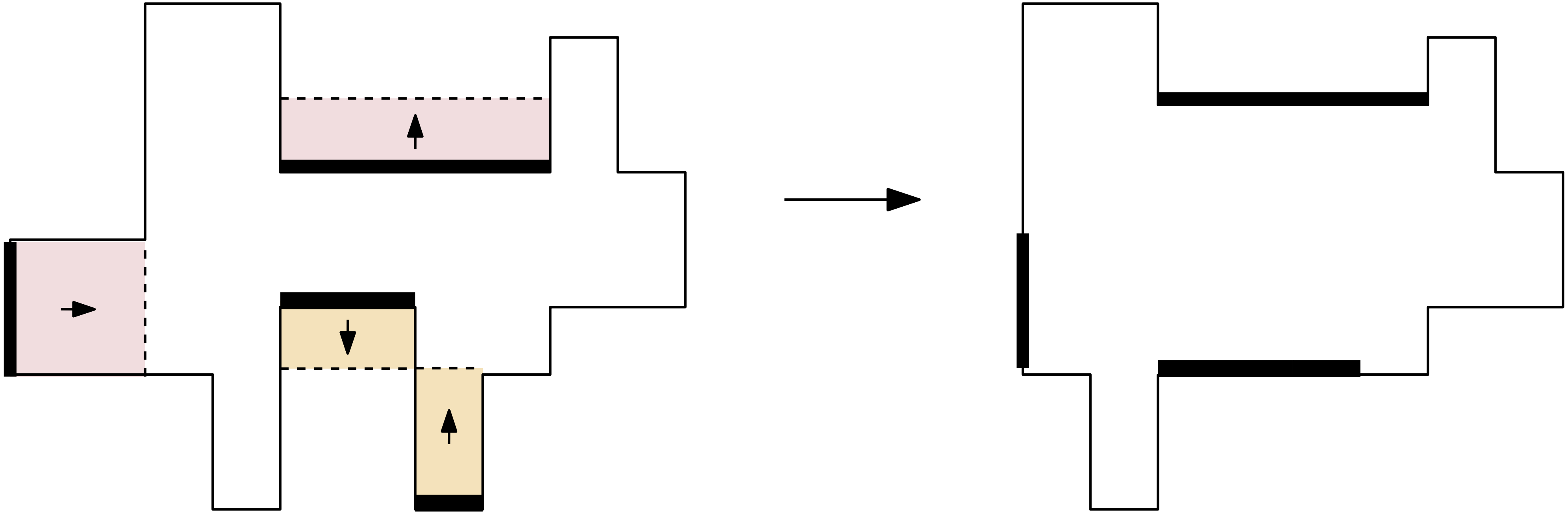
Problemstellung:

geg.: Einfaches rektilineares Polygon P und Parameter $k \in \mathbb{N}$

geg.: Einfaches rektilineares Polygon P', sodass

- P und P' denselben Flächeninhalt besitzen, und
- P' maximal k Kanten besitzt.

Beispiel:

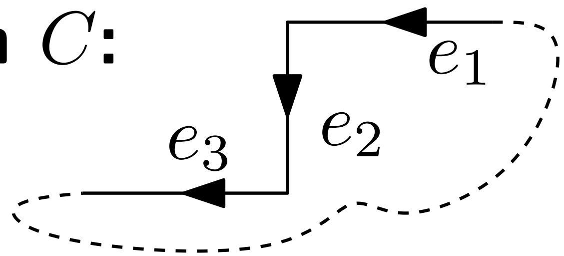


Idee: Verschiebe Kanten und erhalte dabei Flächeninhalt

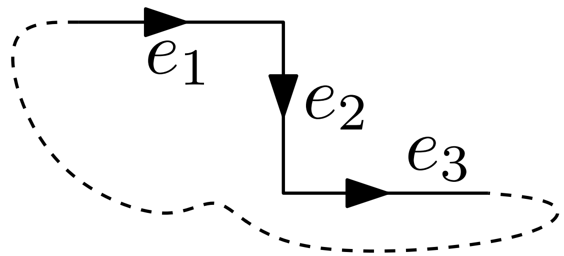
S-Konfiguration & S-Kontraktion

Seien e_1, e_2 und e_3 aufeinanderfolgende Kanten von P.

S-Konfiguration C:



oder



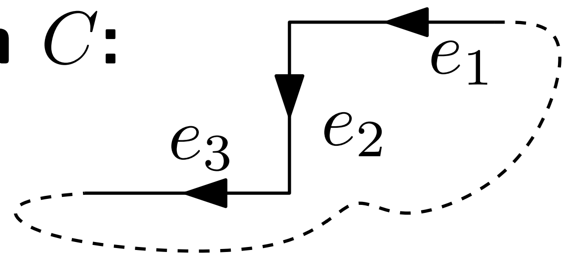
Knicke: links, rechts

Knicke: rechts, links

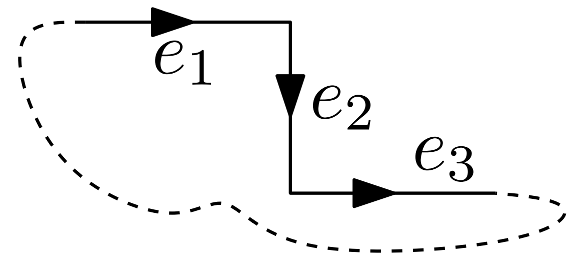
S-Konfiguration & S-Kontraktion

Seien e_1, e_2 und e_3 aufeinanderfolgende Kanten von P.

S-Konfiguration C**:**



oder

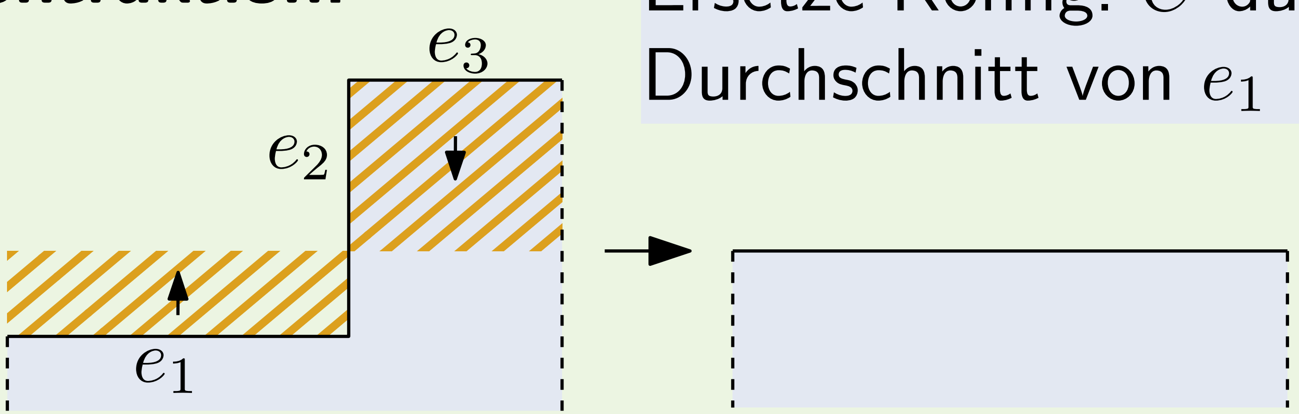


Knicke: links, rechts

Knicke: rechts, links

S-Kontraktion:

Ersetze Konfig. C durch gewichteten Durchschnitt von e_1 und e_3.



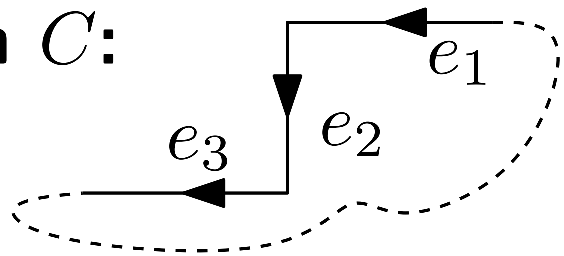
=Kontraktionsfläche K

C heißt *zulässig*, falls Inneres von K leer ist.

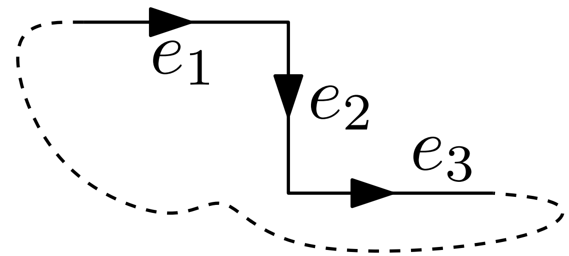
S-Konfiguration & S-Kontraktion

Seien e_1, e_2 und e_3 aufeinanderfolgende Kanten von P.

S-Konfiguration C**:**



oder

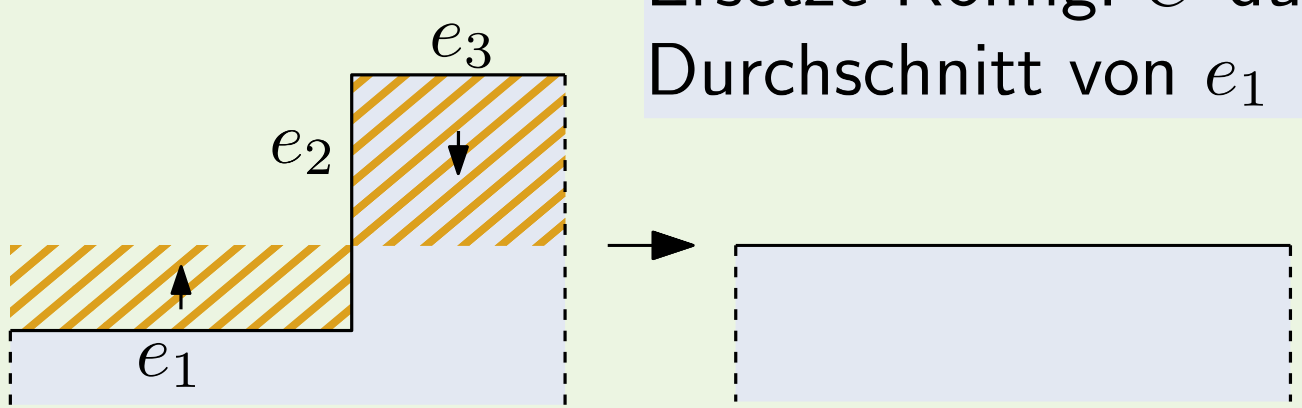


Knicke: links, rechts

Knicke: rechts, links

S-Kontraktion:

Ersetze Konfig. C durch gewichteten Durchschnitt von e_1 und e_3.



=Kontraktionsfläche K

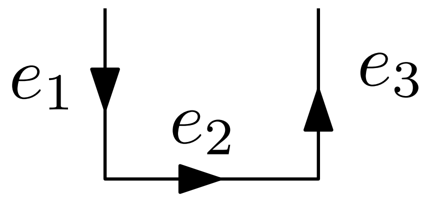
C heißt *zulässig*, falls Inneres von K leer ist.

Beob.: Nich jedes rektilineares Polygon hat zulässige S-Konfig.

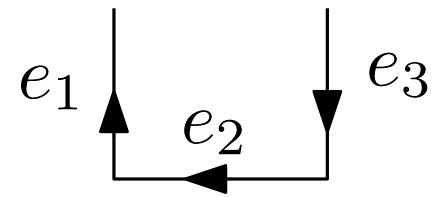
C-Konfiguration

Seien e_1, e_2 und e_3 aufeinanderfolgende Kanten von P.

C-Konfiguration C:



oder



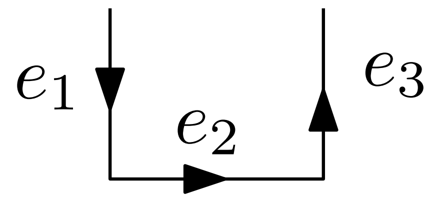
Knicke: links, links

Knicke: rechts, rechts

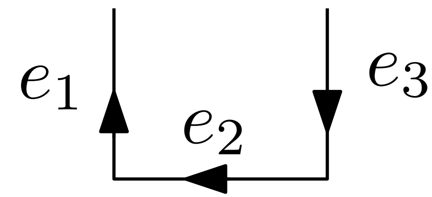
C-Konfiguration

Seien e_1, e_2 und e_3 aufeinanderfolgende Kanten von P.

C-Konfiguration C:



oder

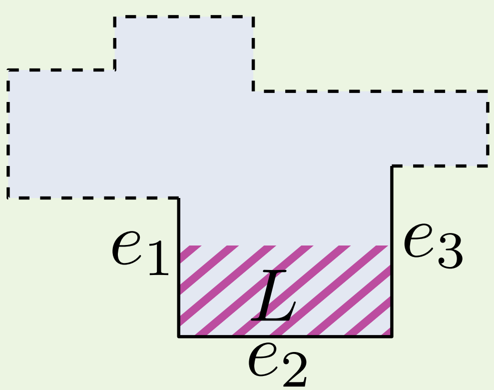


Knicke: links, links

Knicke: rechts, rechts

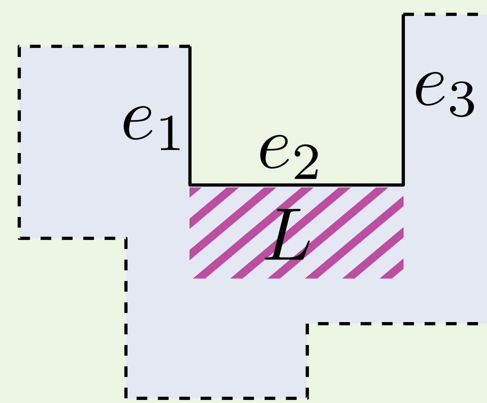
Sei L Rechteck in P, mit Seite e_2.

Innere C-Konfig.:



L liegt auf derselben Seite von e_2 wie e_1 und e_3

Äußere C-Konfig.:

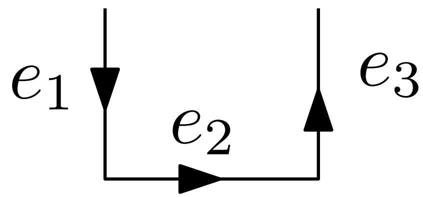


L liegt auf anderer Seite von e_2 wie e_1 und e_3

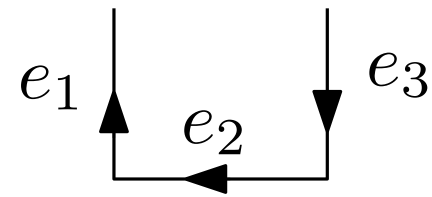
C-Konfiguration

Seien e_1, e_2 und e_3 aufeinanderfolgende Kanten von P.

C-Konfiguration C:



oder

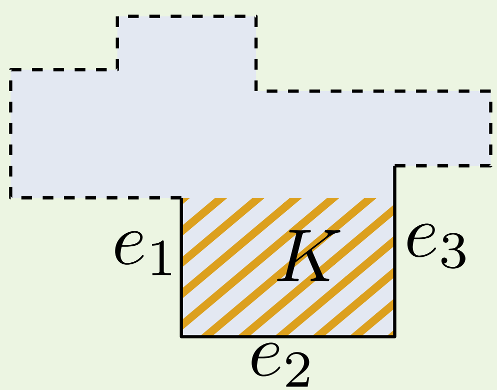


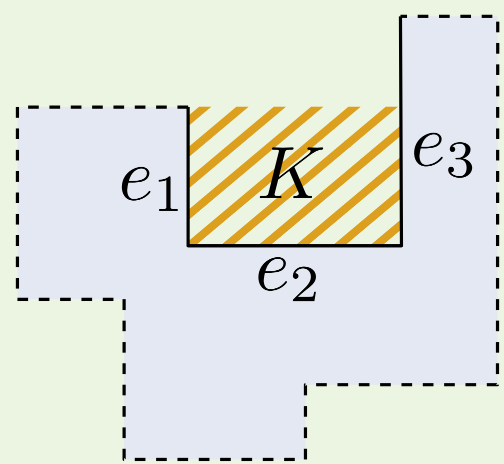
Knicke: links, links

Knicke: rechts, rechts

Rechteck K mit Seiten e_2 und $\min\{e_1, e_3\}$ heißt *Kontraktionsfläche* von C. ($\min\{e_1, e_3\}$ = kürzere der beiden Kanten e_1 und e_3)

Beispiele:





C heißt *zulässig*, falls Inneres von K leer ist.

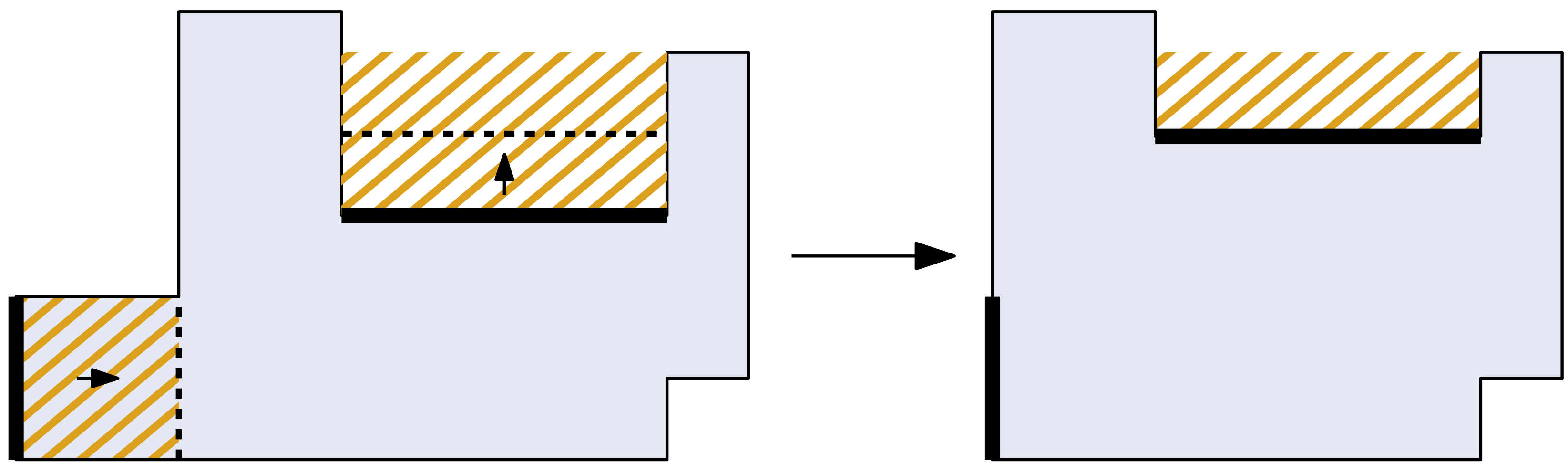
C-Kontraktion

C-**Kontraktion** arbeitet auf:

- Innerer Konfig. $C = (e_1, e_2, e_3)$.
- Äußerer Konfig. $C' = (e'_1, e'_2, e'_3)$.

Operation:

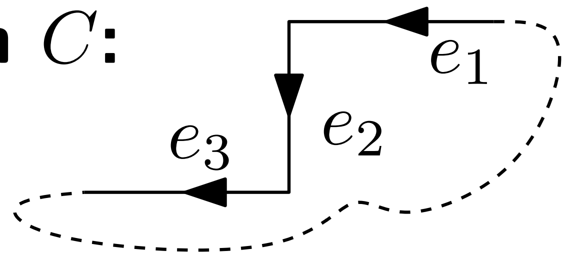
Bewege e_2 und e'_2, bis e_1, e_3, e'_1 oder e'_3 Länge 0 besitzt.



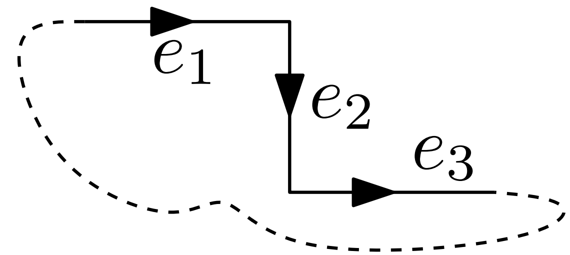
Wähle Geschwindigkeiten von e_2 und e'_2 so, dass Operation flächenerhaltend ist.

Überblick

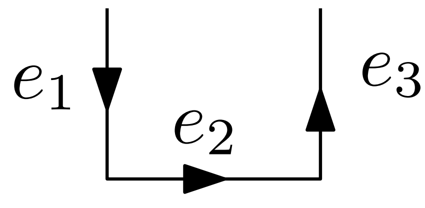
S-Konfiguration C:



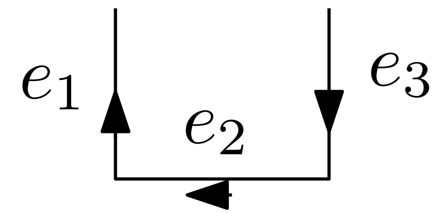
oder



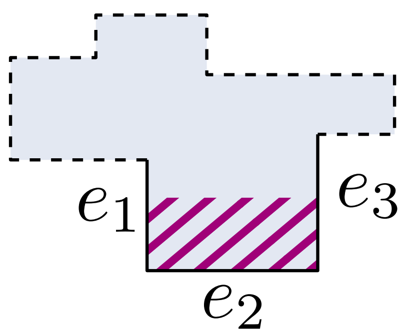
C-Konfiguration C:



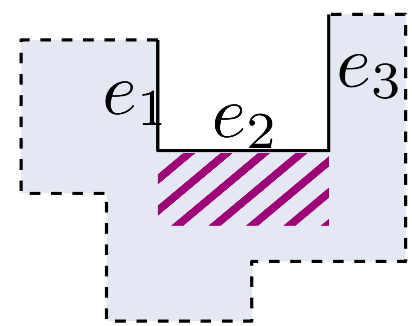
oder



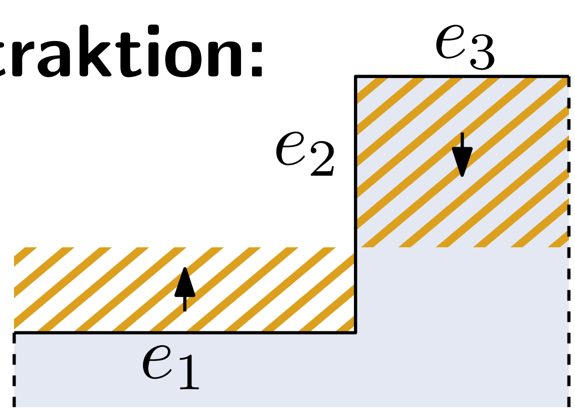
Innere:



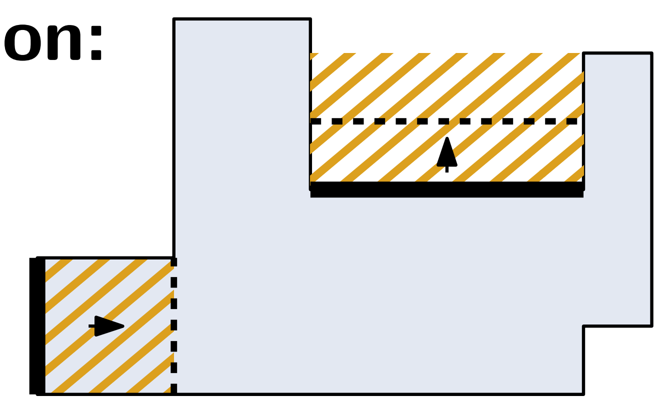
Äußere:



S-Kontraktion:

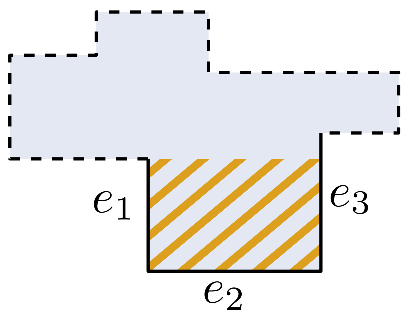


C-Kontraktion:



Vorgehen

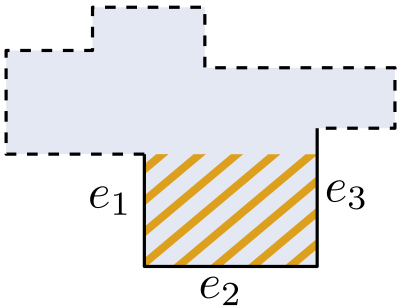
Behauptung:



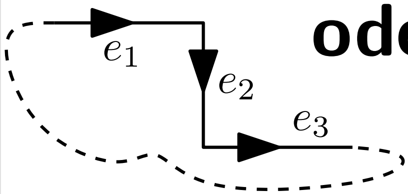
Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt eine zulässige innere C-Konfig.

Vorgehen

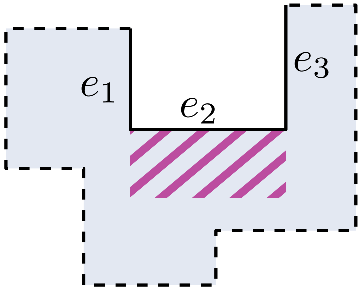
Behauptung:



Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt eine zulässige innere C-Konfig.



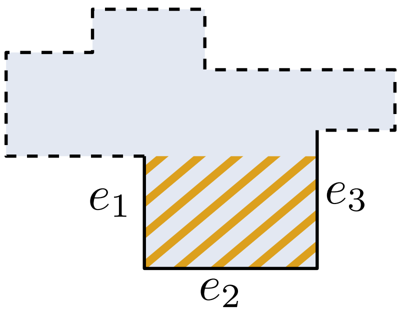
oder



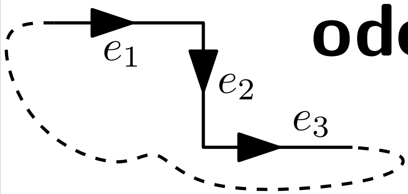
Jedes rektilin. Polygon P mit mind. 6 Kanten besitzt eine zulässige S-Konfig. oder eine äußere C-Konfiguration.

Vorgehen

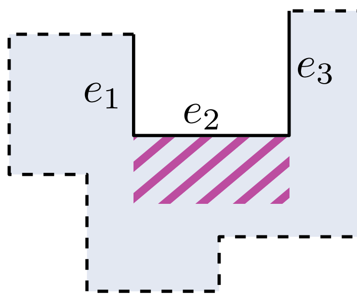
Behauptung:



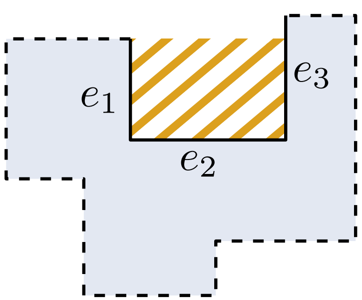
Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt eine zulässige innere C-Konfig.



oder



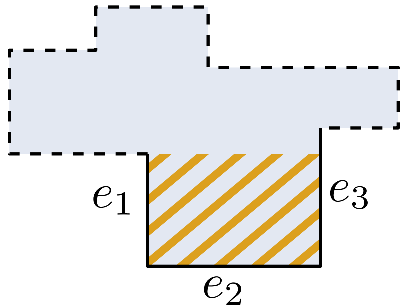
Jedes rektilin. Polygon P mit mind. 6 Kanten besitzt eine zulässige S-Konfig. oder eine äußere C-Konfiguration.



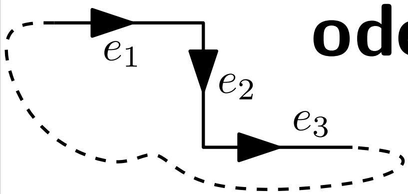
Falls ein rektilin. Polygon P eine äußere C-Konfig. besitzt, dann auch ein zulässige äußere C-Konfig.

Vorgehen

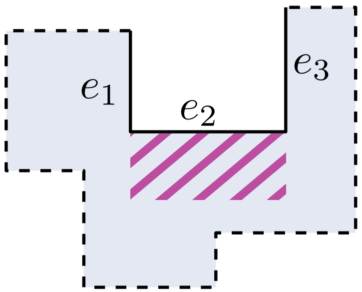
Behauptung:



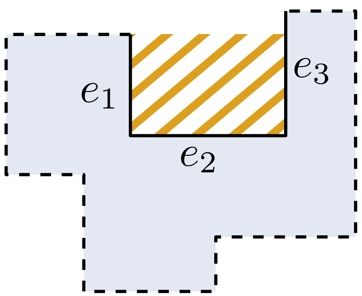
Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt eine zulässige innere C-Konfig.



oder



Jedes rektilin. Polygon P mit mind. 6 Kanten besitzt eine zulässige S-Konfig. oder eine äußere C-Konfiguration.



Falls ein rektilin. Polygon P eine äußere C-Konfig. besitzt, dann auch ein zulässige äußere C-Konfig.

→ Immer S- oder C-Kontraktion anwendbar.

Existenz zulässige innere C-Konfig.

Lemma 1: Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt eine zulässige innere C-Konfig.

Beweis: Siehe Tafel.

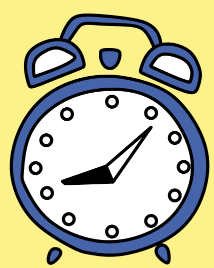
Existenz zulässige innere C-Konfig.

Lemma 1: Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt eine zulässige innere C-Konfig.

Beweis: Siehe Tafel.

Lemma 2: Falls ein rektilin. Polygon P eine äußere C-Konfig. besitzt, dann auch eine zulässige äußere C-Konfiguration.

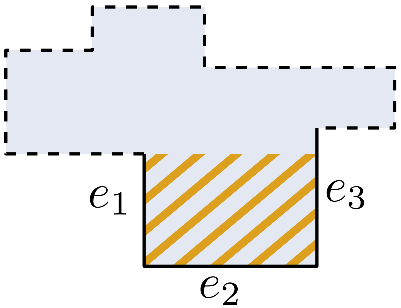
Beweis: Übungsaufgabe.



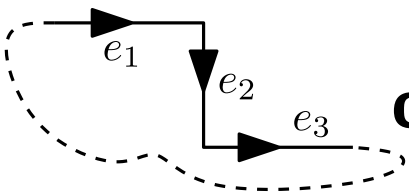
5 min Zeit

Vorgehen

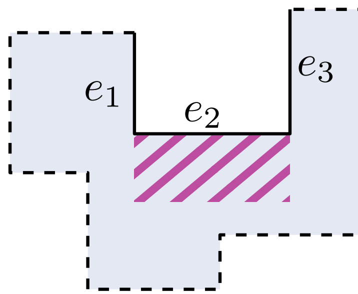
Behauptung:



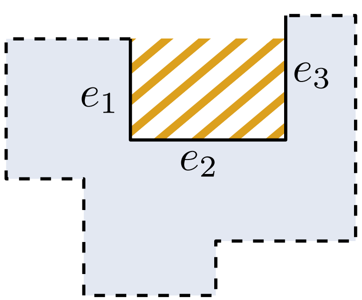
Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt eine zulässige innere C-Konfig.



oder



Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt entweder eine zulässige S-Konfig. oder eine äußere C-Konfig.



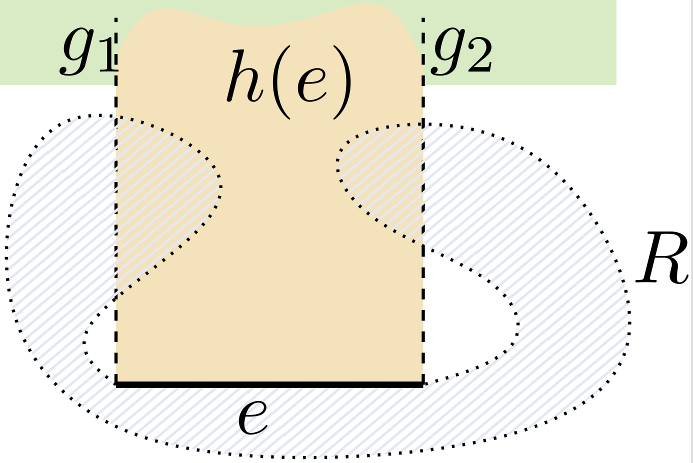
Falls ein rektilin. Polygon P eine äußere C-Konfig. besitzt, dann auch ein zulässige äußere C-Konfig.

$\longrightarrow$ Immer S- oder C-Kontraktion anwendbar.

Hof

Defintion: Der *Hof* $h(e)$ einer Kante $e \in P$ ist die Region, die durch e und zwei zu e orthogonale Halbgeraden g_1 und g_2 begrenzt wird, sodass

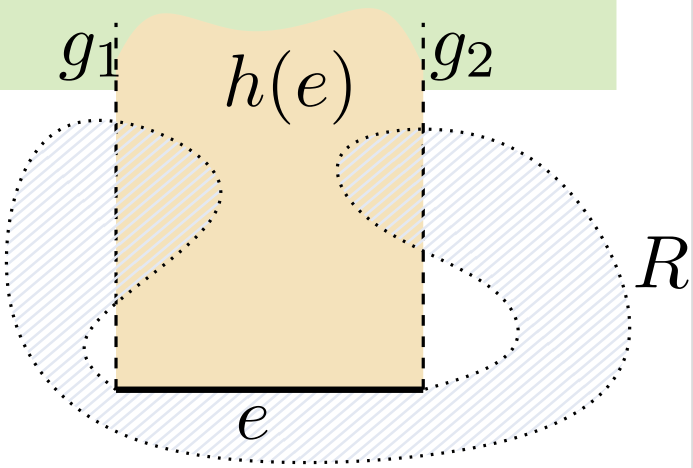
- g_1 und g_2 an den zwei Endpunkten von e starten, und
- $h(e)$ nicht das *Innere* von P in der nächsten Nachbarschaft von e schneidet.



Hof

Defintion: Der *Hof* $h(e)$ einer Kante $e \in P$ ist die Region, die durch e und zwei zu e orthogonale Halbgeraden g_1 und g_2 begrenzt wird, sodass

- g_1 und g_2 an den zwei Endpunkten von e starten, und
- $h(e)$ nicht das *Innere* von P in der nächsten Nachbarschaft von e schneidet.



Lemma 3: P hat eine äußere Konfig., wenn es eine Kante e und einen Randpunkt p von P gibt, sodass

- p in $h(e)$ liegt, und
- p nicht auf e oder einem ihrer Nachbarn liegt.

Beweis: Siehe Tafel.

Lemma 4

Lemma 3: P hat eine äußere Konfig., wenn es eine Kante e und einen Randpunkt p von P gibt, sodass

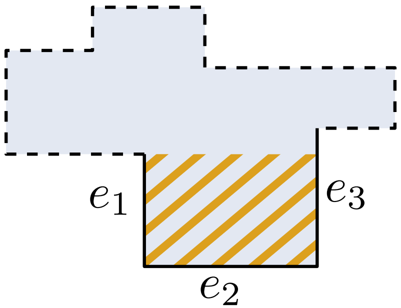
- p in $h(e)$ liegt, und
- p nicht auf e oder einem ihrer Nachbarn liegt.

Lemma 4: Jedes rektilin. Polygon P mit mindestens 6 Kanten besitzt eine zulässige S-Konfig. oder eine äußere C-Konfig.

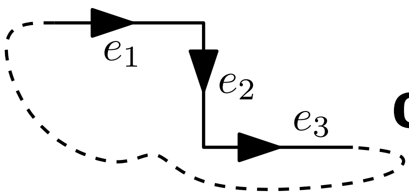
Beweis: Siehe Tafel.

Vorgehen

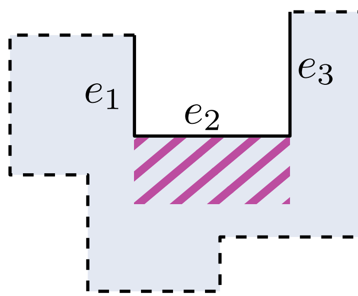
Behauptung:



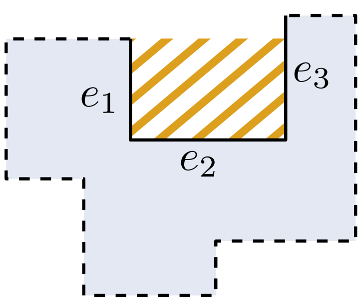
Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt eine zulässige innere C-Konfig.



oder



Jedes rektilineare Polygon P mit mindestens 6 Kanten besitzt entweder eine zulässige S-Konfig. oder eine äußere C-Konfig.



Falls ein rektilin. Polygon P eine äußere C-Konfig. besitzt, dann auch ein zulässige äußere C-Konfig.

→ Immer S- oder C-Kontraktion anwendbar.

Zeit für eine Diskussion

Wie sieht ein möglicher Algorithmus aus? Welche Laufzeit besitzt dieser?

Nach welchen Kriterien wählt man zulässige Konfigurationen?

Wie kann man die Anzahl zulässiger Kontraktionen erhöhen?
Hinweis: Passen Sie die Def. von *zulässige Kontraktion* an.

Wie kann man das Verfahren anpassen, sodass es auch für Unterteilungen der Ebene funktioniert?

5 min Zeit

Zeit für eine Diskussion

Wie sieht ein möglicher Algorithmus aus? Welche Laufzeit besitzt dieser?

Nach welchen Kriterien wählt man zulässige Konfigurationen?

Wie kann man die Anzahl zulässiger Kontraktionen erhöhen?
Hinweis: Passen Sie die Def. von *zulässige Kontraktion* an.

Wie kann man das Verfahren anpassen, sodass es auch für Unterteilungen der Ebene funktioniert?

5 min Zeit

Antworten

Wie sieht ein möglicher Algorithmus aus? Welche Laufzeit besitzt dieser?

Antworten

Wie sieht ein möglicher Algorithmus aus? Welche Laufzeit besitzt dieser?

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k

Ausgabe: P, sodass P max. k Kanten hat.

Solange $|P| > k$

1. Finde zulässige S-Kontraktion C.
2. **Falls** C vorhanden
   (a) Kontrahiere C
3. **sonst**
   (a) Finde zulässige äußere Konfig. C_1.
   (b) Finde zulässige innere Konfig. C_2.
   (c) Kontrahiere (C_1, C_2)

Antworten

Wie sieht ein möglicher Algorithmus aus? **Welche Laufzeit besitzt dieser?**

Antworten

Wie sieht ein möglicher Algorithmus aus? Welche Laufzeit besitzt dieser?

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P max. k Kanten hat.

Solange $|P| > k$

1. Finde zulässige S-Kontraktion C.
2. **Falls** C vorhanden
   (a) Kontrahiere C
3. **sonst**
   (a) Finde zulässige äußere Konfig. C_1.
   (b) Finde zulässige innere Konfig. C_2.
   (c) Kontrahiere C_1 und C_2

Antworten

Wie sieht ein möglicher Algorithmus aus? **Welche Laufzeit besitzt dieser?**

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P max. k Kanten hat.

Solange $|P| > k$

1. Finde zulässige S-Kontraktion C.
2. **Falls** C vorhanden
   (a) Kontrahiere C
3. **sonst**
   (a) Finde zulässige äußere Konfig. C_1.
   (b) Finde zulässige innere Konfig. C_2.
   (c) Kontrahiere C_1 und C_2

Zulässige Konfiguration finden: $O(n^2)$
Kontrahieren: $O(1)$

Schleife $O(n)$ $\longrightarrow$ $O(n^3)$

Antworten

Wie sieht ein möglicher Algorithmus aus? **Welche Laufzeit besitzt dieser?**

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P max. k Kanten hat.

Solange $|P| > k$

1. Finde zulässige S-Kontraktion C.
2. **Falls** C vorhanden
   (a) Kontrahiere C
3. **sonst**
   (a) Finde zulässige äußere Konfig. C_1.
   (b) Finde zulässige innere Konfig. C_2.
   (c) Kontrahiere C_1 und C_2

Zulässige Konfiguration finden: $O(n^2)$
Kontrahieren: $O(1)$

Schleife $O(n)$ $\longrightarrow$ $O(n^3)$

Beschleunigung auf $O(n^2)$:

Für jede Konfiguration C einen Zähler, der Anzahl Kanten, die C blockieren, speichert.

$\longrightarrow$ Zulässige Konfiguration finden: $O(n)$ Zeit.

$\longrightarrow$ Update der Zähler: $O(n)$ Zeit pro Schleifendurchlauf.

Antworten

Wie sieht ein möglicher Algorithmus aus? **Welche Laufzeit besitzt dieser?**

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P max. k Kanten hat.

Solange $|P| > k$

1. Finde zulässige S-Kontraktion C.
2. **Falls** C vorhanden

Theorem: Sei ein einfaches rektilineares Polygon P mit n Kanten und ein Parameter k mit $4 \leq k \leq n$ gegeben. Eine flächenerhaltende Vereinfachung von P mit maximal k Kanten kann in $O(n^2)$ Zeit brechnet werden.

Kontrahieren: $O(1)$

Beschleunigung auf $O(n^2)$:

Für jede Konfiguration C einen Zähler, der Anzahl Kanten, die C blockieren, speichert.

→ Zulässige Konfiguration finden: $O(n)$ Zeit.

→ Update der Zähler: $O(n)$ Zeit pro Schleifendurchlauf.

Mögliche Antworten

Nach welchen Kriterien wählt man zulässige Konfigurationen?

Wie kann man die Anzahl zulässiger Kontraktionen erhöhen?
Hinweis: Passen Sie die Def. von *zulässige Kontraktion* an.

Mögliche Antworten

Nach welchen Kriterien wählt man zulässige Konfigurationen?

[MvRS10]:

- Kontraktionen mit möglichst kleiner symmetrischer Differenz.
- C-Konfig., die *nahe* beieinander liegen.

Wie kann man die Anzahl zulässiger Kontraktionen erhöhen?
Hinweis: Passen Sie die Def. von *zulässige Kontraktion* an.

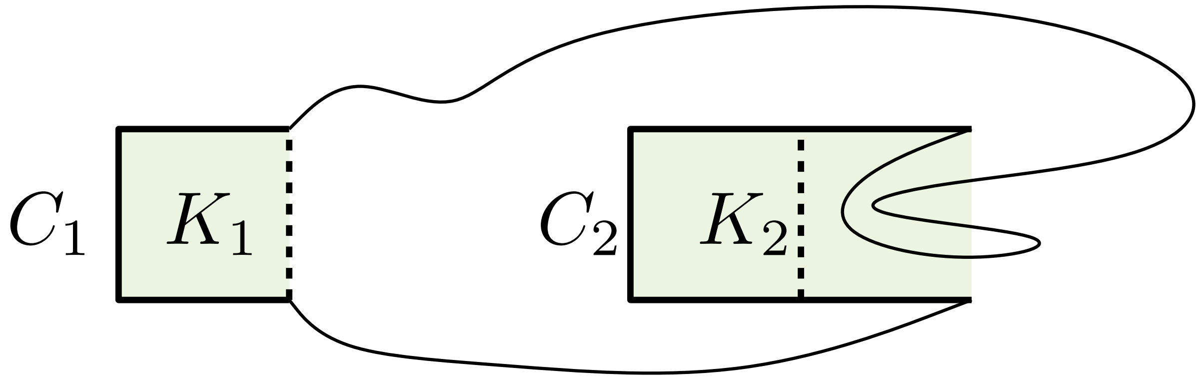
Mögliche Antworten

Nach welchen Kriterien wählt man zulässige Konfigurationen?

[MvRS10]:

- Kontraktionen mit möglichst kleiner symmetrischer Differenz.
- C-Konfig., die *nahe* beieinander liegen.

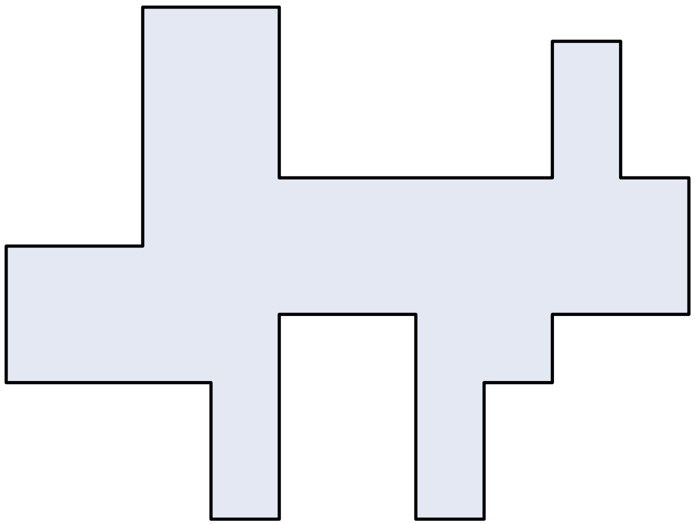
Wie kann man die Anzahl zulässiger Kontraktionen erhöhen?
Hinweis: Passen Sie die Def. von *zulässige Kontraktion* an.



Es reicht aus, wenn C_1 und C_2 Kontraktion von $\min\{|K_1|, |K_2|\}$ zulassen.

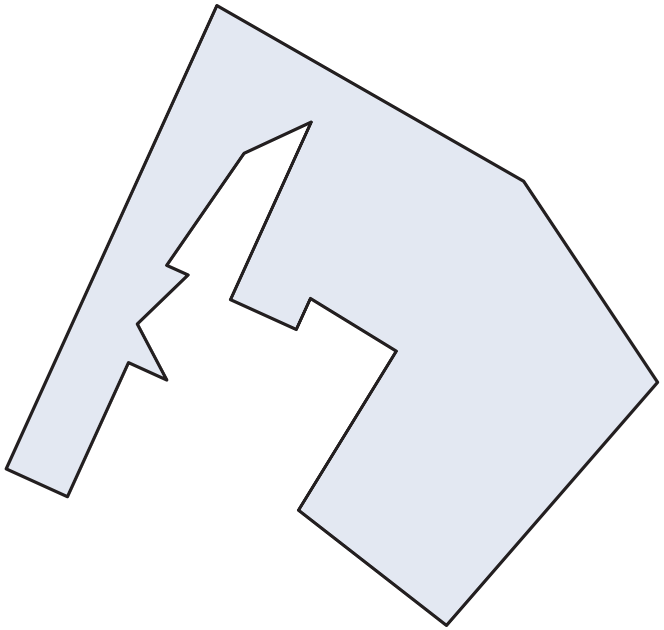
Überblick

Bisher:



Vereinfachung von einfachen rektilinearen Polygonen.

Jetzt:



Vereinfachung von allgemeinen einfachen Polygonen.

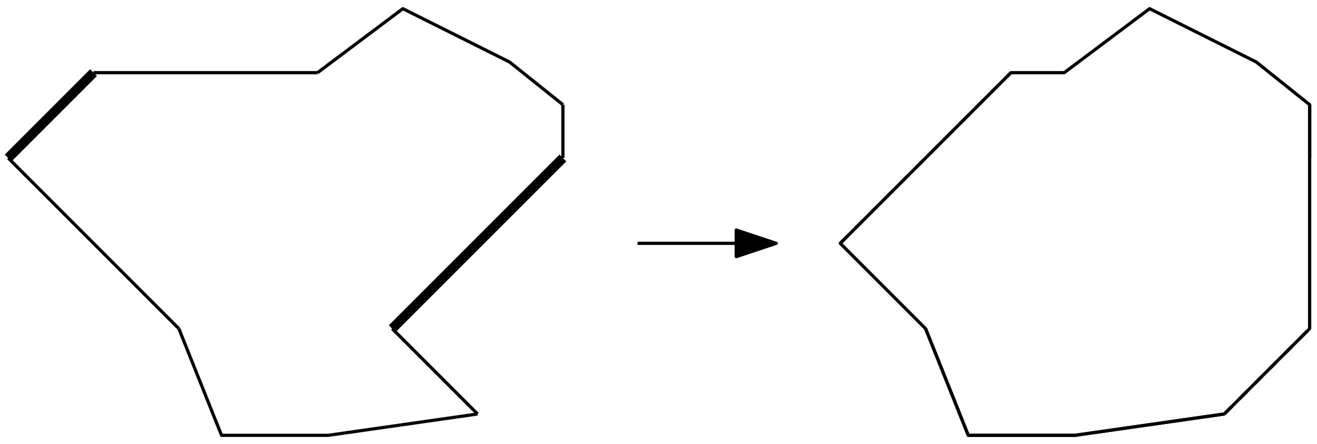
Formale Beschreibung

Problemstellung:

geg.: Einfaches Polygon P und Parameter $k \in \mathbb{N}$

ges.: Einfaches Polygon P', sodass

- P und P' denselben Flächeninhalt besitzen, und
- die Kanten aus P' dieselben Orientierungen wie die in P verwenden,
- P' maximal k Kanten besitzt .



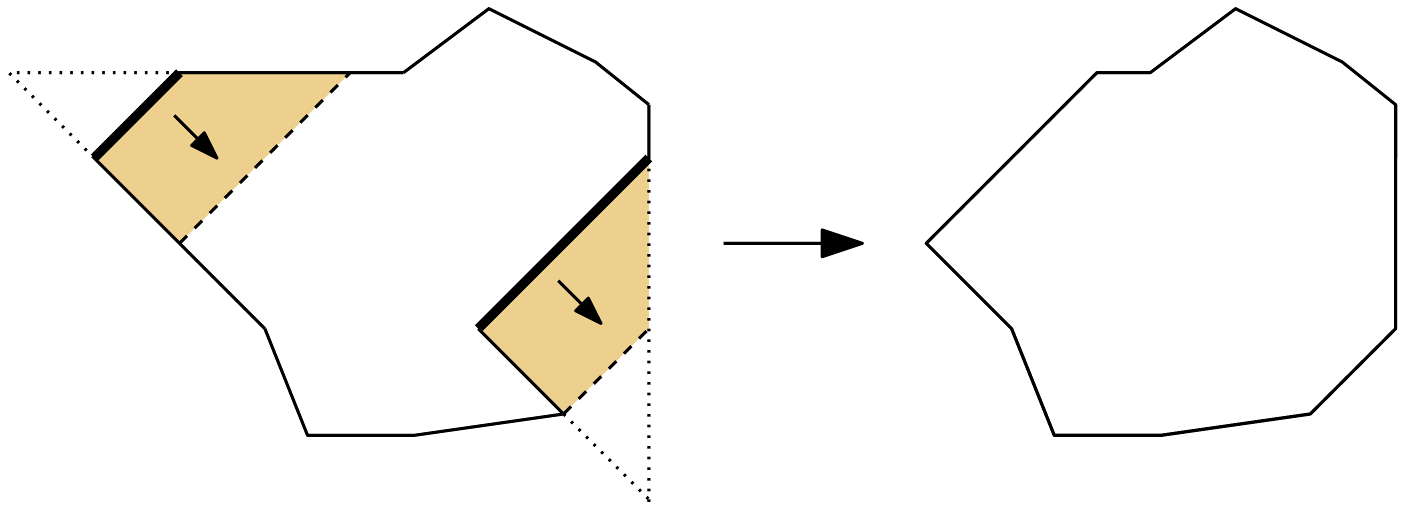
Formale Beschreibung

Problemstellung:

geg.: Einfaches Polygon P und Parameter $k \in \mathbb{N}$

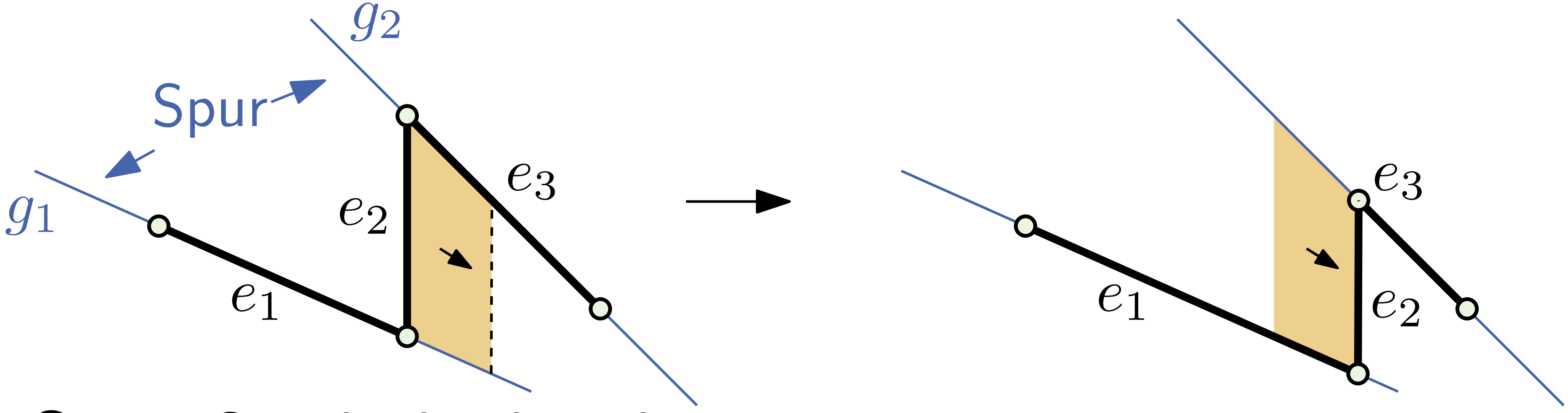
ges.: Einfaches Polygon P', sodass

- P und P' denselben Flächeninhalt besitzen, und
- die Kanten aus P' dieselben Orientierungen wie die in P verwenden,
- P' maximal k Kanten besitzt .



Operation Kantenverschiebung

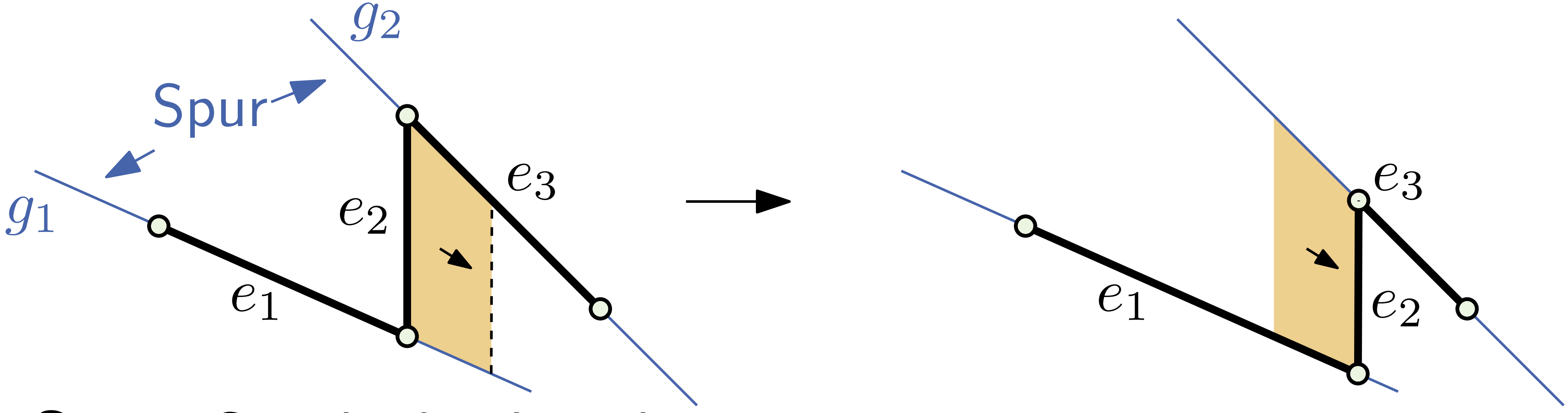
Konfiguration C**:** Drei aufeinanderfolgende Kanten e_1, e_2, e_3.



Spur: Gerade durch e_1 bzw. e_3

Operation Kantenverschiebung

Konfiguration C**:** Drei aufeinanderfolgende Kanten e_1, e_2, e_3.



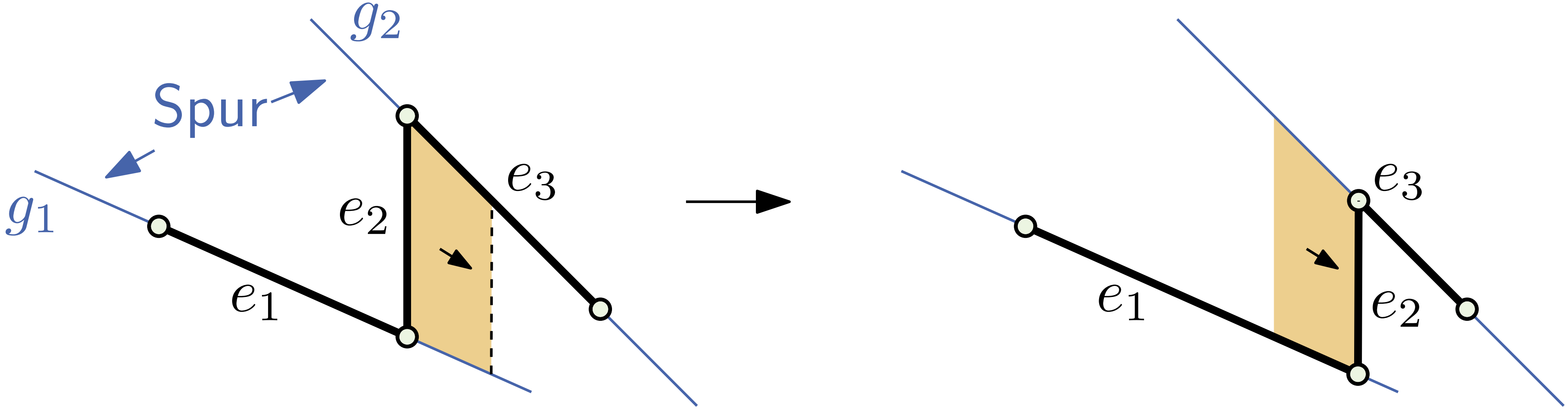
Spur: Gerade durch e_1 bzw. e_3

Operation: Kantenverschiebung auf C

- Bewegt e_2, sodass die Orientierung beibehalten wird.
- Die Knoten von e_2 verbleiben auf den Spuren.
  $\longrightarrow$ e_1 und e_3 werden verlängert/verkürzt.

Operation Kantenverschiebung

Konfiguration C: Drei aufeinanderfolgende Kanten e_1, e_2, e_3.



Spur: Gerade durch e_1 bzw. e_3

Operation: Kantenverschiebung auf C

- Bewegt e_2, sodass die Orientierung beibehalten wird.
- Die Knoten von e_2 verbleiben auf den Spuren.
  $\longrightarrow$ e_1 und e_3 werden verlängert/verkürzt.

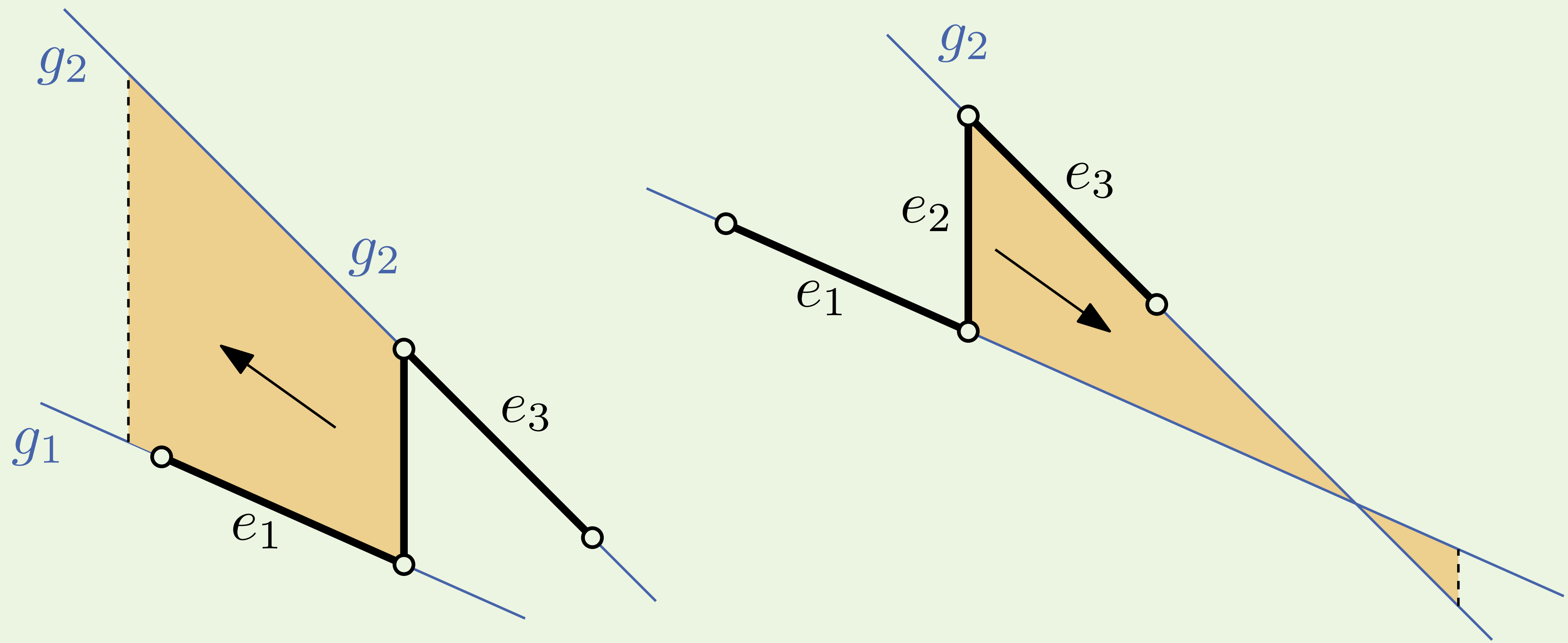
Kantenverschiebung ist **zulässig**, wenn

- mindestens ein Endpunkt von e_2 auf Orignal von e_1 oder e_3
- e_2 verbleibt auf derselben Seite von Schnitt von g_1 und g_2.

Operation Kantenverschiebung

Konfiguration C**:** Drei aufeinanderfolgende Kanten e_1, e_2, e_3.

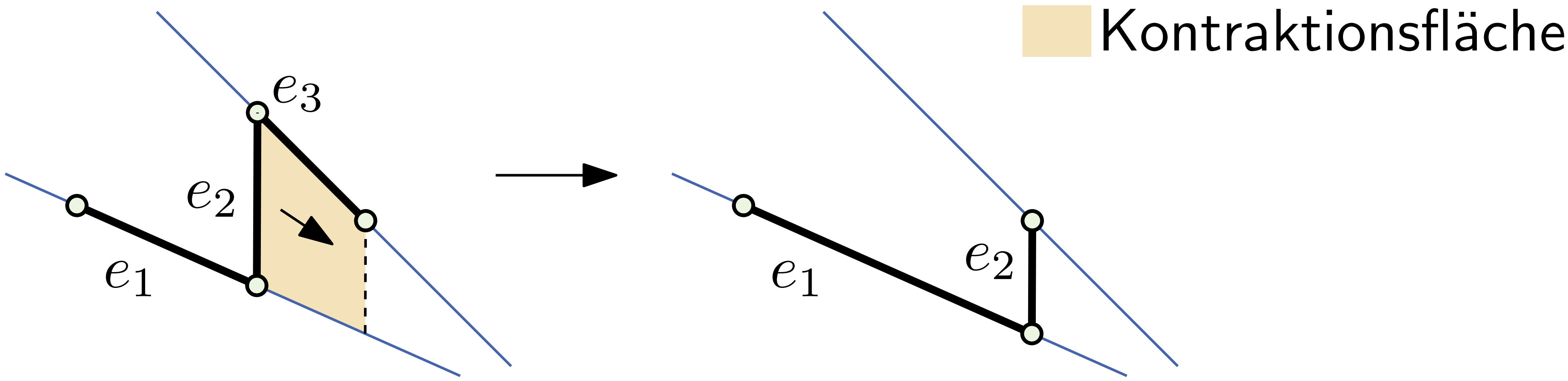
Beispiele für nicht zulässige Verschiebungen:



- mindestens ein Endpunkt von e_2 auf Orignal von e_1 oder e_3
- e_2 verbleibt auf derselben Seite von Schnitt von g_1 und g_2.

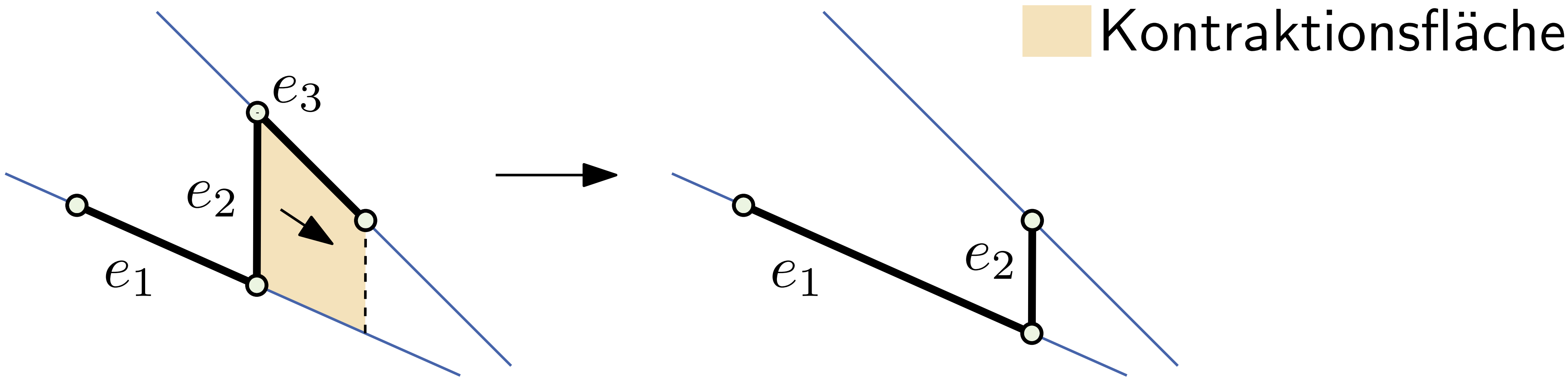
Kontraktion

Kantenverschiebung heißt **Kontraktion** von Konfig. C, falls eine Kante von C nach Verschiebung Länge 0 hat.



Kontraktion

Kantenverschiebung heißt **Kontraktion** von Konfig. C, falls eine Kante von C nach Verschiebung Länge 0 hat.

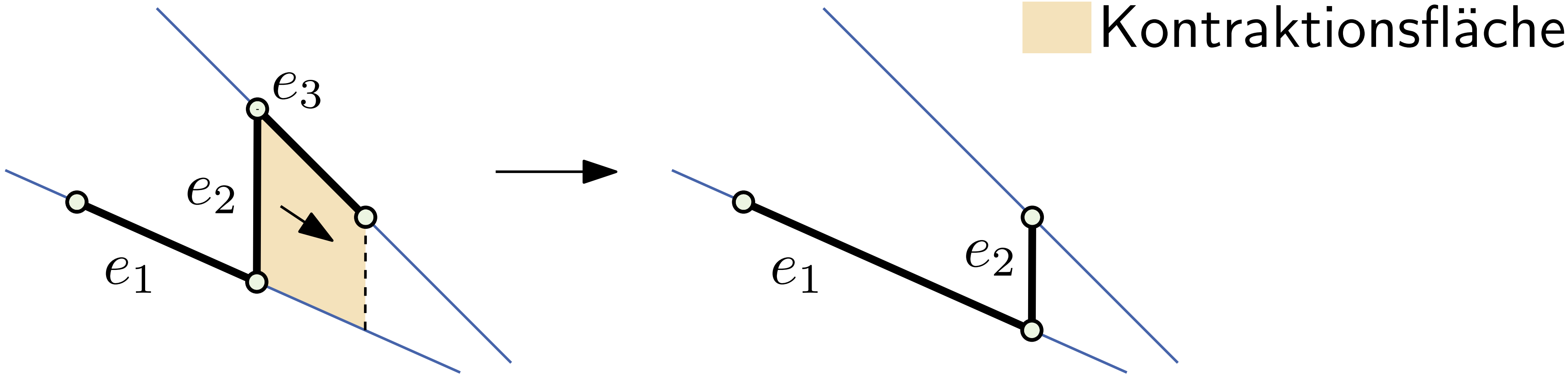


Kontraktion ist

- **zulässig**, falls Innerere der Kontrakt.fläche leer ist.
- **positiv** (negativ), falls sie P vergrößert (verkleinert).

Kontraktion

Kantenverschiebung heißt **Kontraktion** von Konfig. C, falls eine Kante von C nach Verschiebung Länge 0 hat.



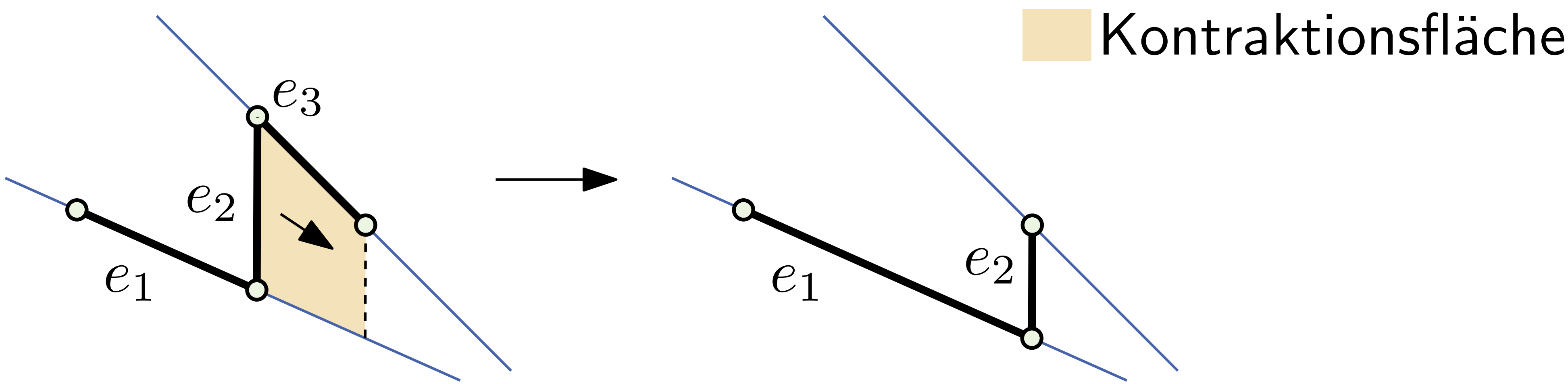
Kontraktion ist

- **zulässig**, falls Innerere der Kontrakt.fläche leer ist.
- **positiv** (negativ), falls sie P vergrößert (verkleinert).

Zwei zulässige Kontraktionen sind **komplementär**, falls die eine Kontraktion positiv und die andere negativ ist.

Kontraktion

Kantenverschiebung heißt **Kontraktion** von Konfig. C, falls eine Kante von C nach Verschiebung Länge 0 hat.



Kontraktion ist

- **zulässig**, falls Innerere der Kontrakt.fläche leer ist.
- **positiv** (negativ), falls sie P vergrößert (verkleinert).

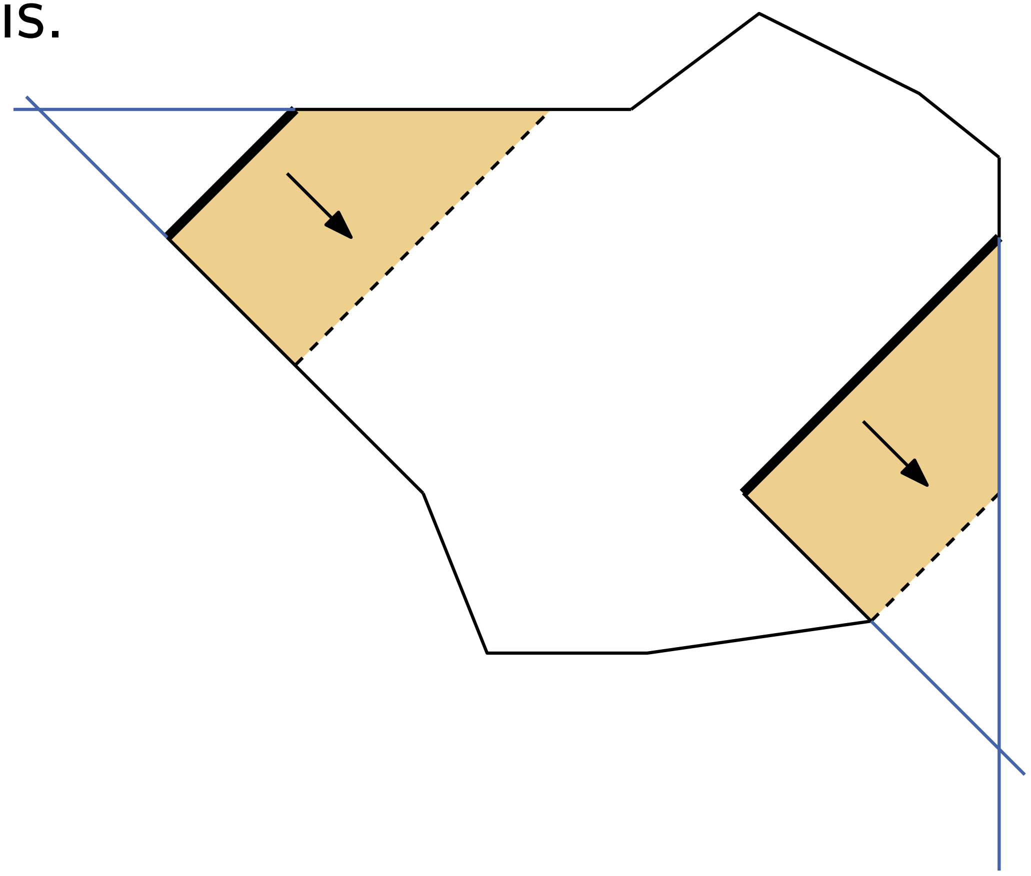
Zwei zulässige Kontraktionen sind **komplementär**, falls die eine Kontraktion positiv und die andere negativ ist.

Zwei Kontraktionen stehen im **Konflikt**, falls die entsprechenden Konfigurationen gemeinsame Kanten haben.

Kontraktion

Theorem: Jedes einfache nicht-konvexe Polygon P mit mind. 6 Kanten hat ein Paar komplementäres Kontraktionen, das nicht im Konflikt steht.

Ohne Beweis.



Vereinfachung von Polygonen

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k

Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere $(K_1, K_2)^\star$

*: Konfiguration mit kleinerer Fläche wird vollständig kontrahiert, während die andere die Flächenänderung kompensiert.

Vereinfachung von Polygonen

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k

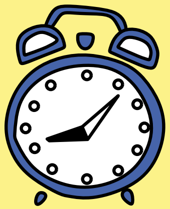
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere $(K_1, K_2)^\star$

Diskussion:

1. Weshalb ist der Algorithmus korrekt?
2. Wie wählt man die Kontraktionen in Schritt 1 und 2?
3. Was heißt *geeignet* in Schritt 3?
4. Welche Laufzeit hat der Algorithmus?



5 min Zeit

Mögliche Antworten

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

1. Weshalb ist der Algorithmus korrekt?

Mögliche Antworten

Algorithmus

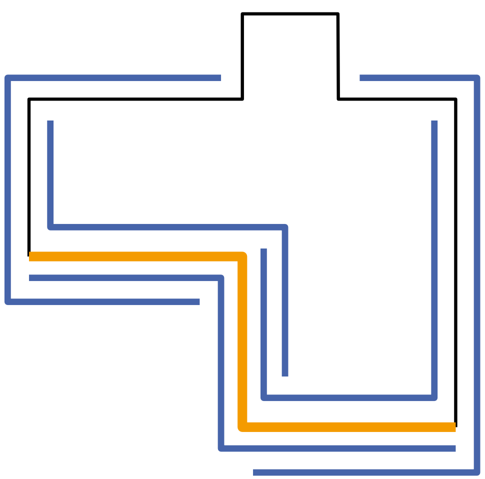
Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

1. Weshalb ist der Algorithmus korrekt?

positive (negative) Konfiguration hat Konflikt mit maximal 5 negativen (positiven) Konfigurationen.



$\longrightarrow$ Es gibt Kontraktionspaar (K_1, K_2)

Mögliche Antworten

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

Wie wählt man die Kontraktionen in Schritt 1 und 2?

Was heißt geeignet in Schritt 3?

Mögliche Antworten

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

Wie wählt man die Kontraktionen in Schritt 1 und 2?

[BMS:] Wähle Kontraktionen mit kleinsten Kontraktionsflächen

Was heißt geeignet in Schritt 3?

Mögliche Antworten

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

Wie wählt man die Kontraktionen in Schritt 1 und 2?

[BMS:] Wähle Kontraktionen mit kleinsten Kontraktionsflächen

Was heißt geeignet in Schritt 3?

[BMS:] Wähle Kontraktionspaar, das möglichs nahe beieinander liegt.

Mögliche Antworten

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k

Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

Welche Laufzeit besitzt der Algorithmus?

Mögliche Antworten

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

Welche Laufzeit besitzt der Algorithmus?

Schritt 1 und Schritt 2: $O(n^2)$

Schritt 3 und Schritt 4: $O(1)$

$\xrightarrow{\text{Schleife } O(n)} O(n^3)$

Mögliche Antworten

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

Welche Laufzeit besitzt der Algorithmus?

Schritt 1 und Schritt 2: $O(n^2)$
Schritt 3 und Schritt 4: $O(1)$

$\xrightarrow{\text{Schleife } O(n)} O(n^3)$

Beschleunigung auf $O(n^2)$:

Für jede Konfiguration C einen Zähler, der Anzahl Kanten, die C blockieren, speichert.

$\longrightarrow$ Schritt 1 und Schritt 2 in $O(1)$ Zeit.

$\longrightarrow$ Update der Zähler in $O(n)$ Zeit pro Schleifendurchlauf.

Ergebnis

Algorithmus

Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

Theorem: Sei ein einfaches Polygon P mit n Kanten und ein Parameter k mit $6 \leq k \leq n$ gegeben. Eine flächenerhaltende Vereinfachung von P mit maximal k Kanten kann in ?? Zeit brechnet werden.

Ergebnis

Algorithmus

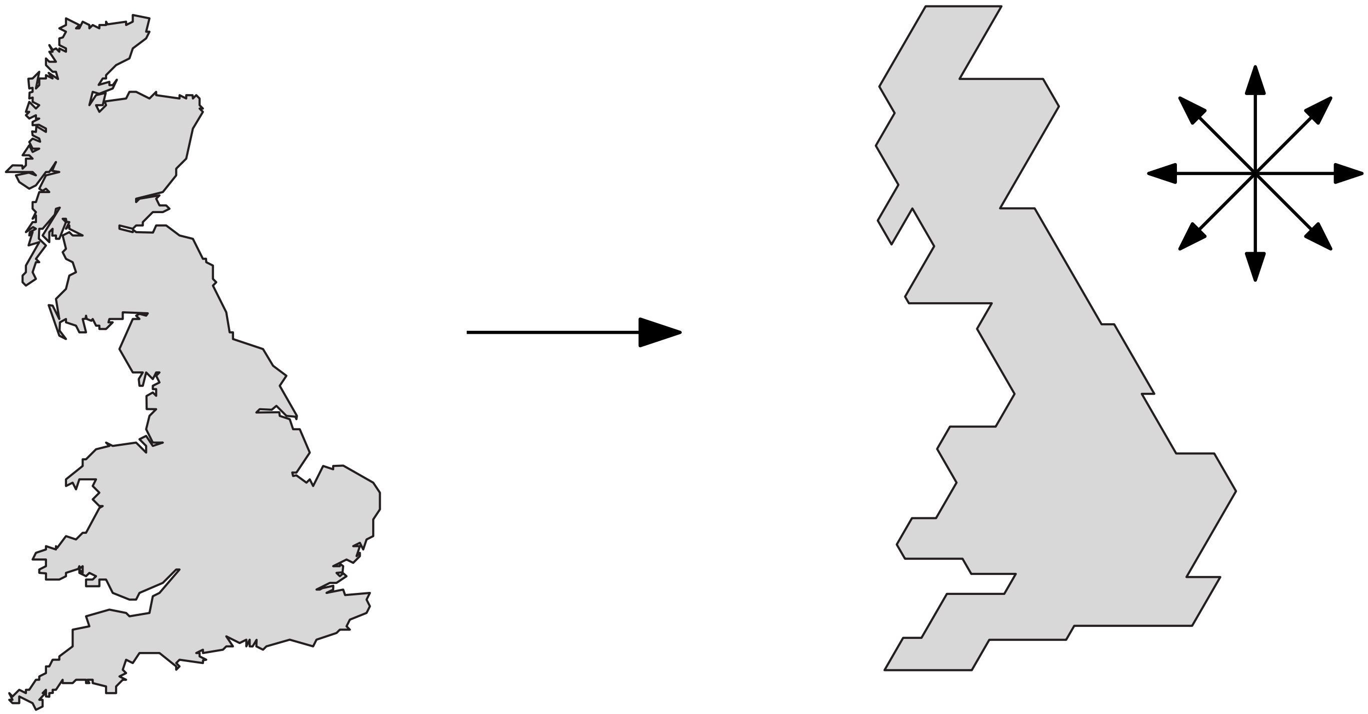
Eingabe: Einfaches Polygon P und Parameter k
Ausgabe: P, sodass P konvex ist oder max. k Kanten hat.

Solange $|P| > k$ und P nicht konvex ist.

1. Finde bis zu 6 positive zulässige Kontraktionen.
2. Finde bis zu 6 negative zulässige Kontraktionen.
3. Finde *geeignetes* Kontraktionspaar (K_1, K_2) unter diesen.
4. Kontrahiere (K_1, K_2)

Theorem: Sei ein einfaches Polygon P mit n Kanten und ein Parameter k mit $6 \leq k \leq n$ gegeben. Eine flächenerhaltende Vereinfachung von P mit maximal k Kanten kann in $O(n^2)$ Zeit brechnet werden.

Schematisierung



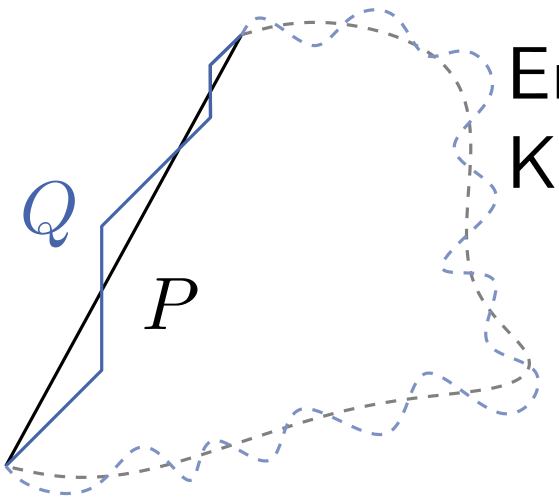
Geg.: Polygon P, Parameter $k \in \mathbb{N}$ und Richtungen $\mathcal{C} = \{\rightarrow \nearrow \uparrow \nwarrow \leftarrow \swarrow \downarrow \searrow\}$

Ges.: Einfaches Polygon P', sodass

- P' maximal k Kanten besitzt,
- Kanten von P' nur Richtungen aus $\mathcal{C}$ haben,
- P' gleichen Flächeninhalt wie P besitzt.

Verfahren

1. Schritt: Wandle P in Polygon Q um, sodass Q nur Kantenrichtungen aus $\mathcal{C}$ verwendet.



Ersetze jede Kante e von P durch einen Kantenzug, der nur Richtungen aus $\mathcal{C}$ verwendet.

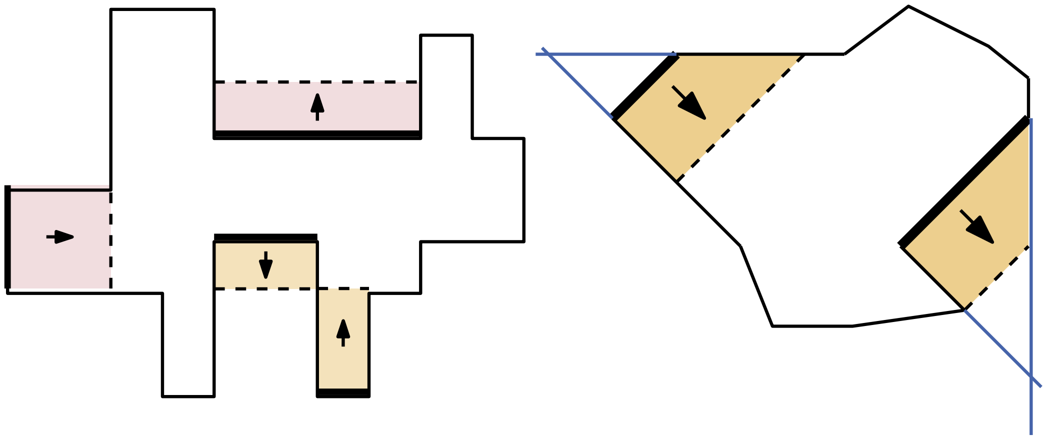
Unter der Bedingung:

Q und P haben gleichen Flächeninhalt.

2. Schritt: Vereinfache Polygon Q zu P' mithilfe von Kantenverschiebungen.

Zusammenfassung

Verfahren zur Vereinfachung von Polygonen



- Basiert auf Kontraktion von Konfigurationen.
- Erhält Flächeninhalt.

Grundlage für verschiedene Verfahren:

- Vereinfachung von Polygonen.
- Schematisierung
- Vereinfachung von Unterteilungen der Ebene.
- Aggregation von Flächen.
- Herrstellung von rechten Winkeln (*Squaring*).

Nicht behandelt.