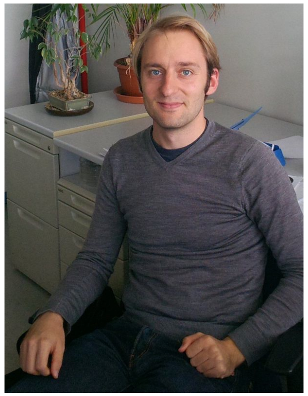
Karlsruher Institut für Technologie

Algorithmische Kartografie
Übung am 25.04.2013

INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

Übung für algorithmische Kartografie

Übungsleiter



- Benjamin Niedermann
- `niedermann@kit.edu`
- Raum 322
- Sprechzeiten: individuell per Mail vereinbaren

Termine

- Vorlesung: Di 9:45 – 11:15 Uhr, Raum 301
- Übung: Do 10:15 – 11:00 Uhr, Raum 301 (ab 25.04.)

Ablauf des Übungsbetrieb

Ausgabe Blatt 1

for *Woche* $i = 2 \ldots 14$ **do**

  if *Tag == Dienstag* **then**

    Ausgabe Blatt i

    Abgabe Blatt $(i-1)$

  if *Tag == Donnerstag & !isFeiertag(Tag)* **then**

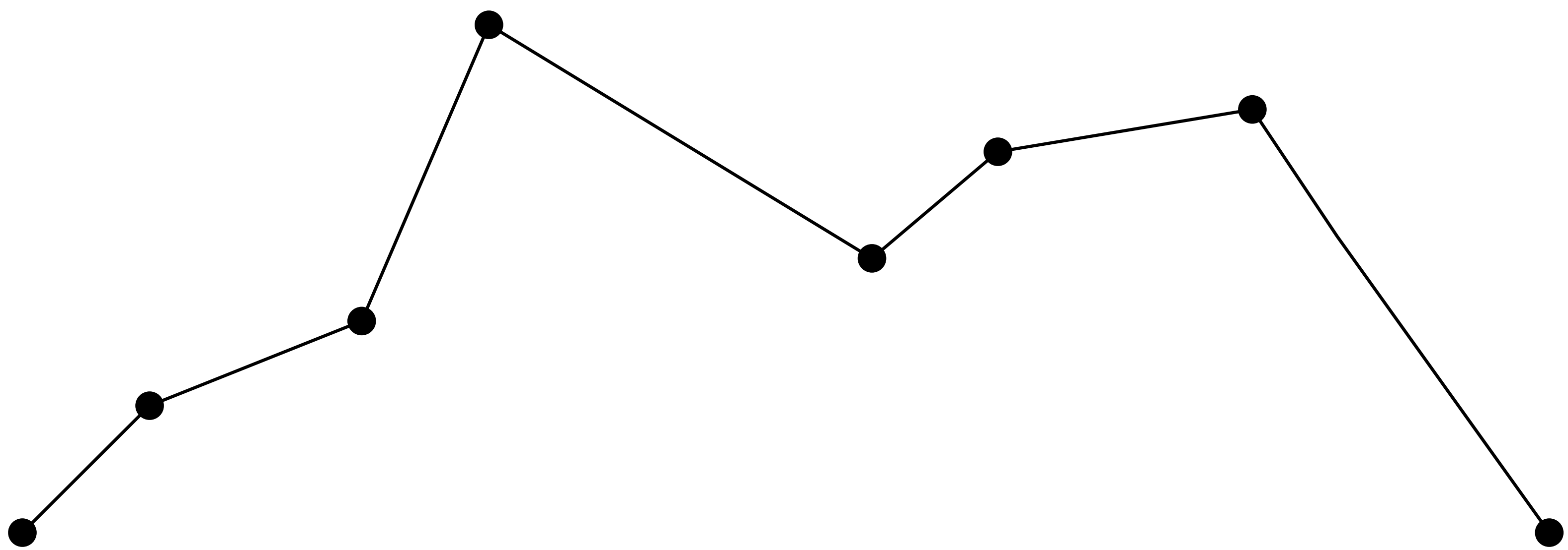
    Besprechung Blatt $(i-1)$

- Bearbeitung der Aufgaben und Abgabe der Lösungen in Zweiergruppen erwünscht
- Übung in der Regel wöchentlich ca. 45 Minuten
- abweichende Regelung an Feiertagen

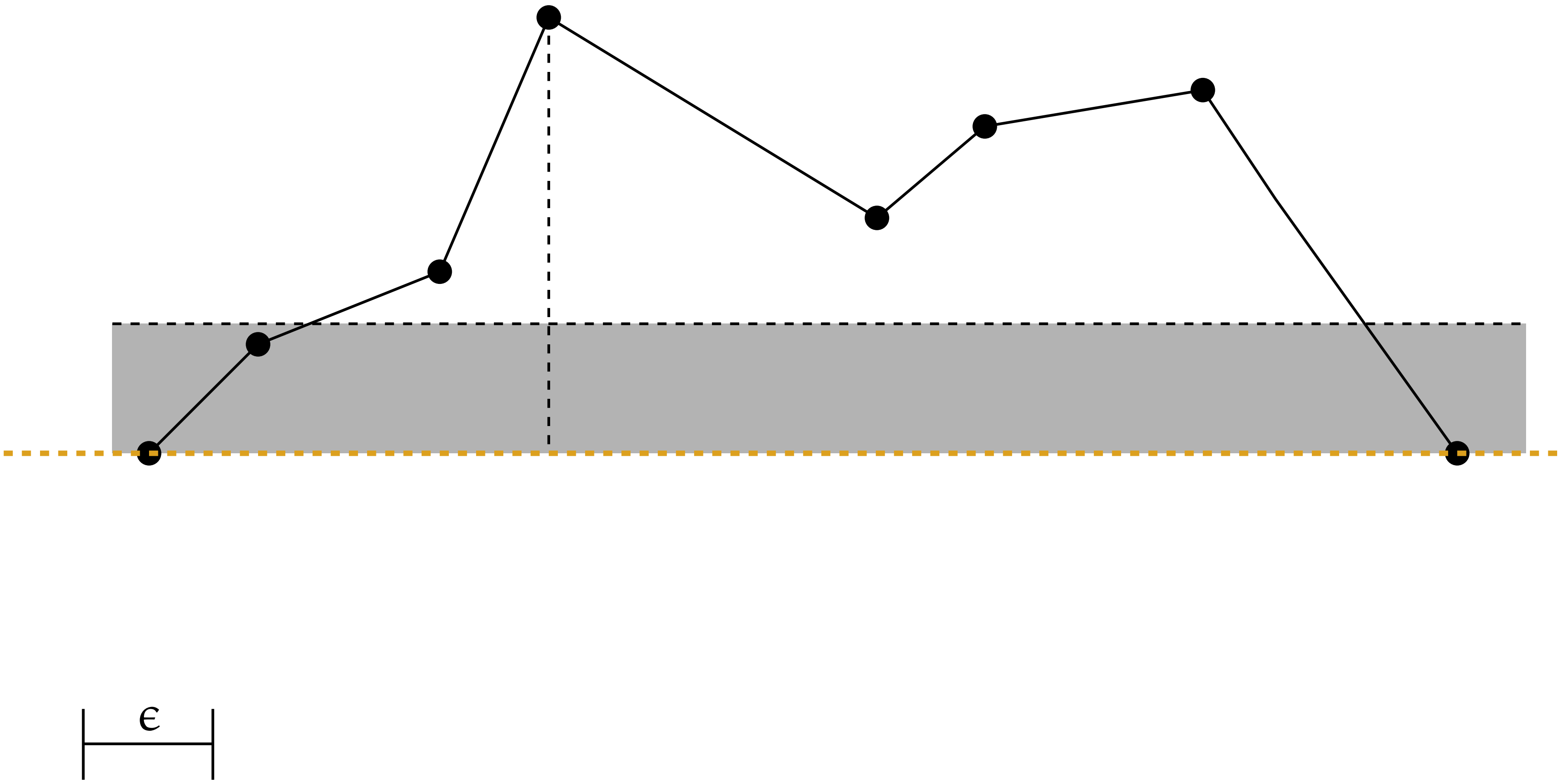
Douglas- und Peucker-Algorithmus

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

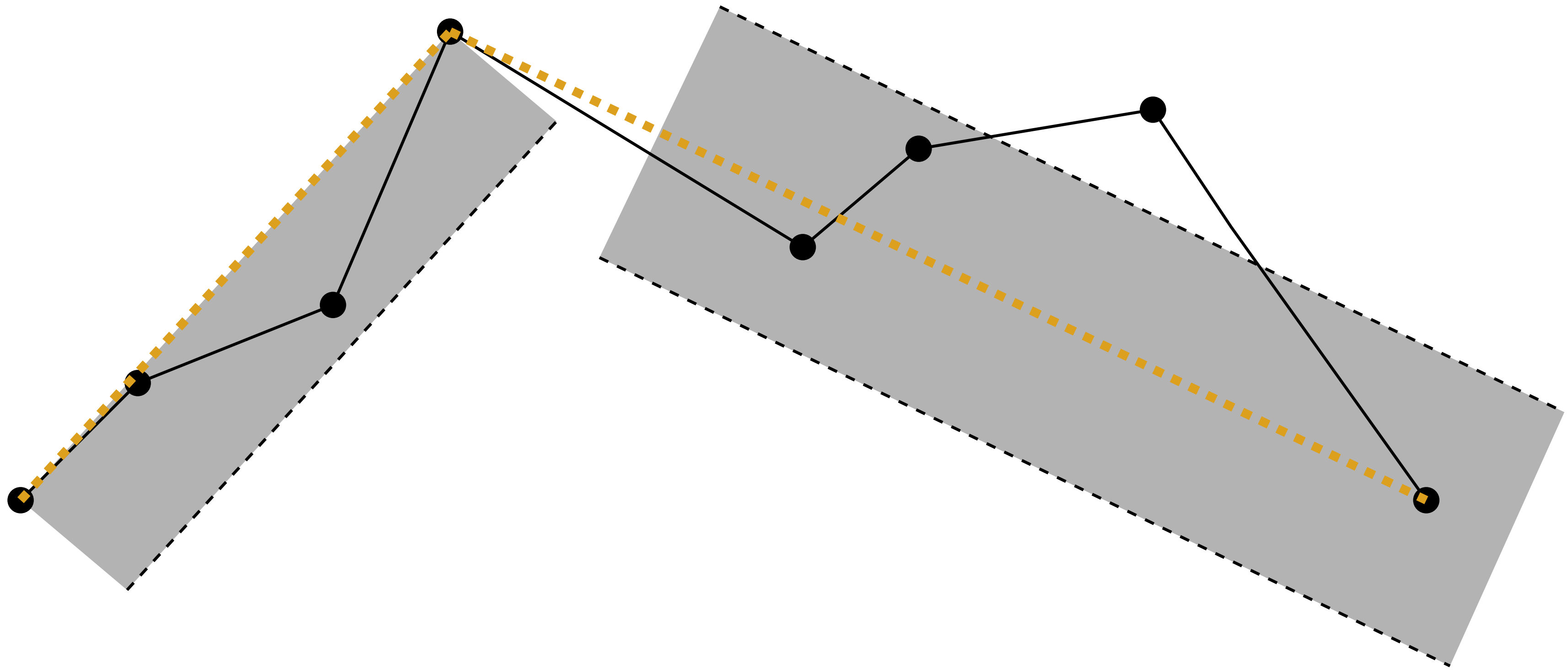
Douglas- und Peucker-Algorithmus



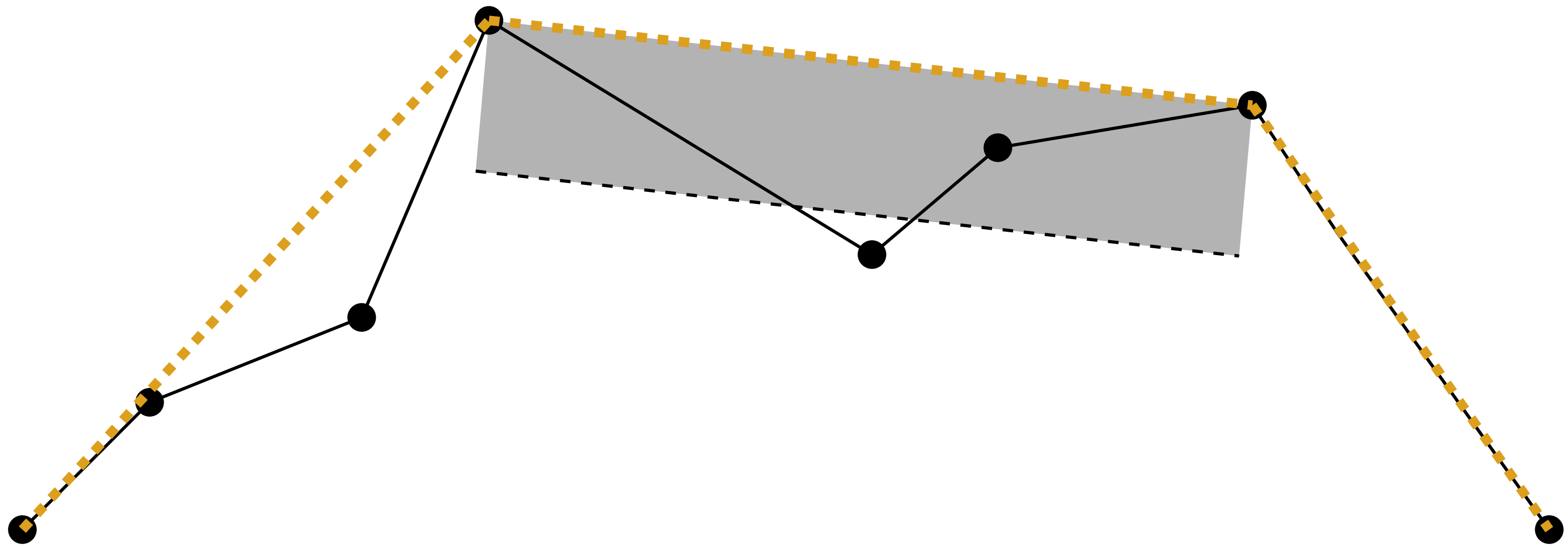
Douglas- und Peucker-Algorithmus



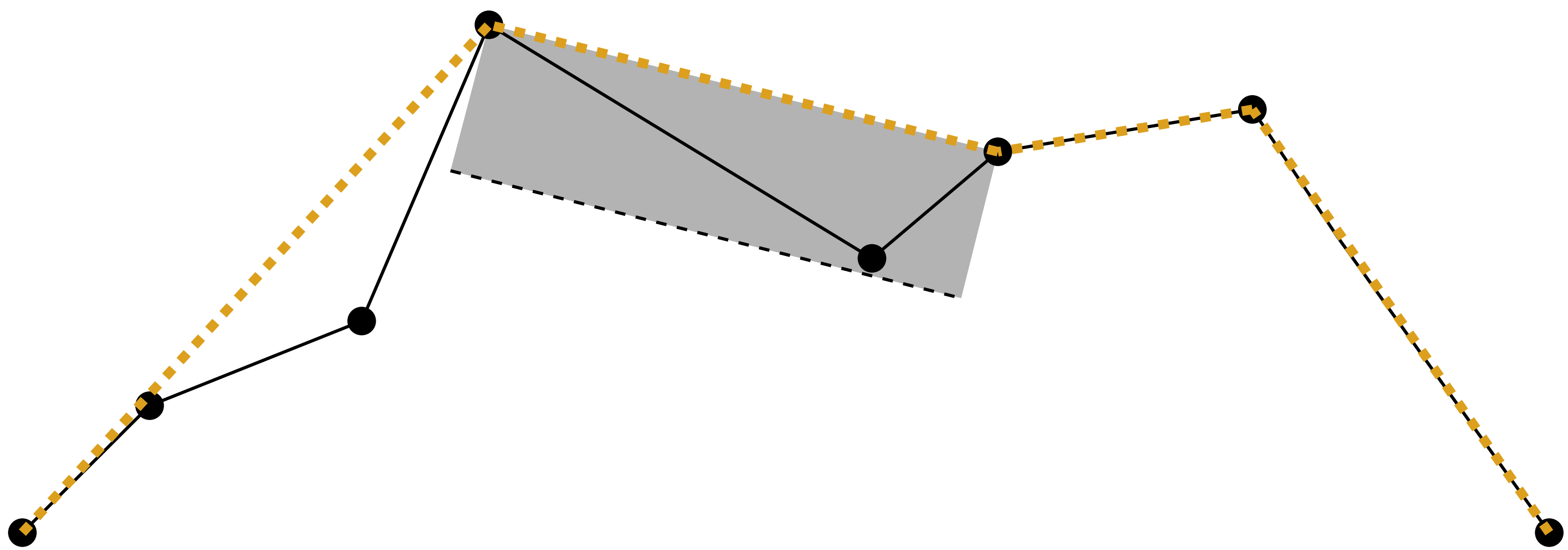
Douglas- und Peucker-Algorithmus



Douglas- und Peucker-Algorithmus



Douglas- und Peucker-Algorithmus

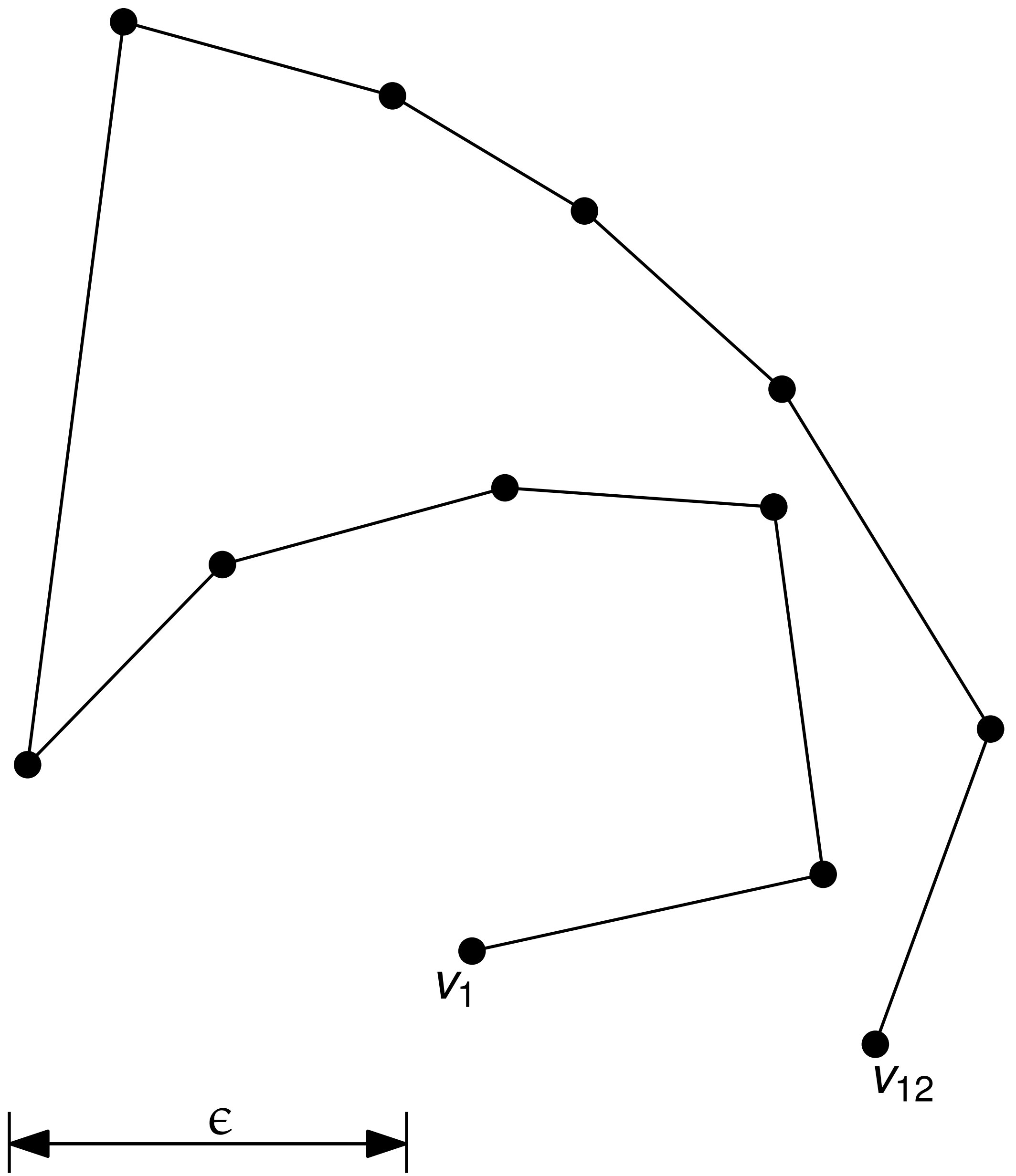


Aufgaben

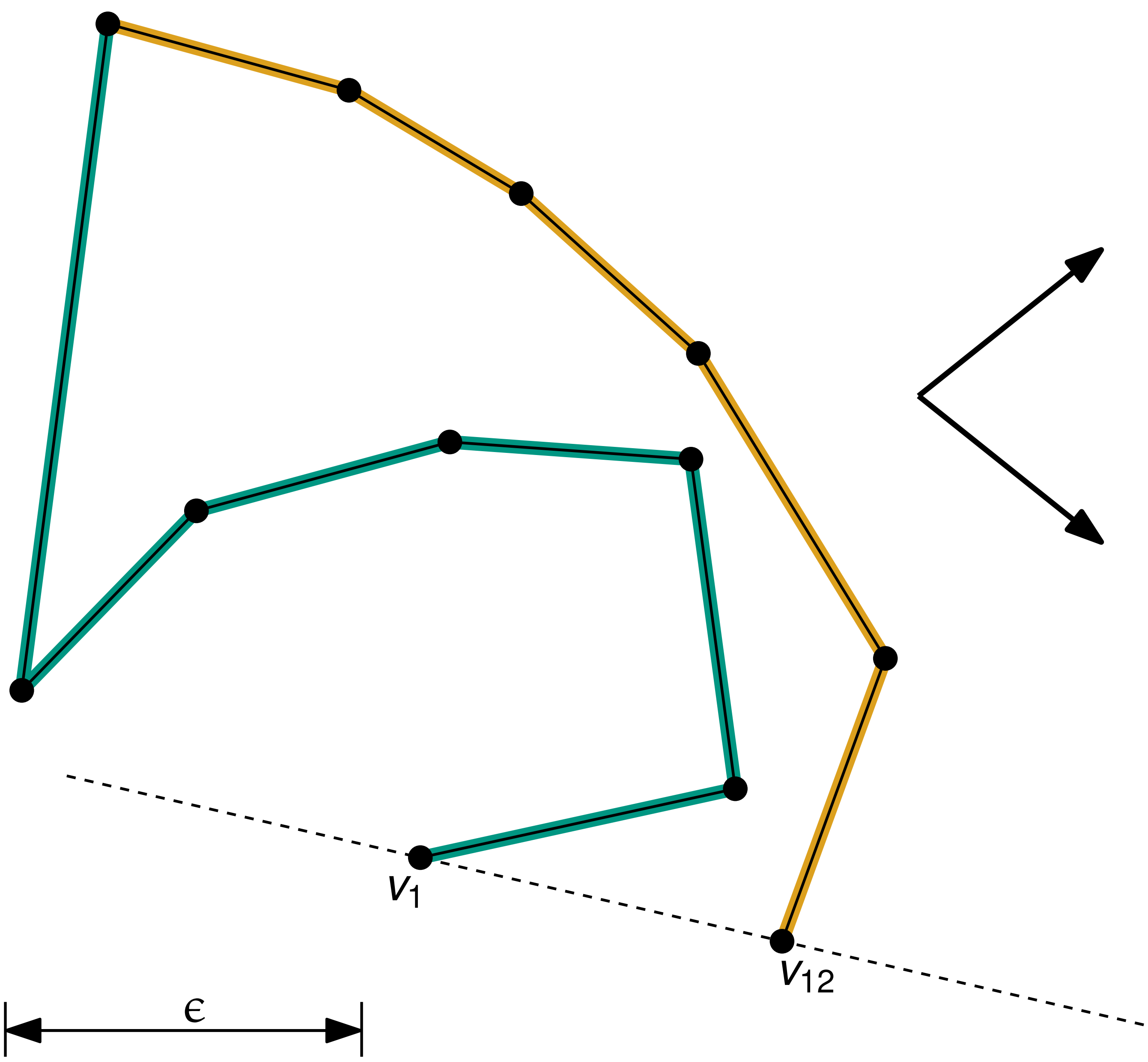
Karlsruher Institut für Technologie

Aufgabe:
Geben Sie einen einfachen Polygonzug an, der vom Douglas- und Peucker-Algorithmus zu einem nicht einfachen Polygonzug vereinfacht wird.

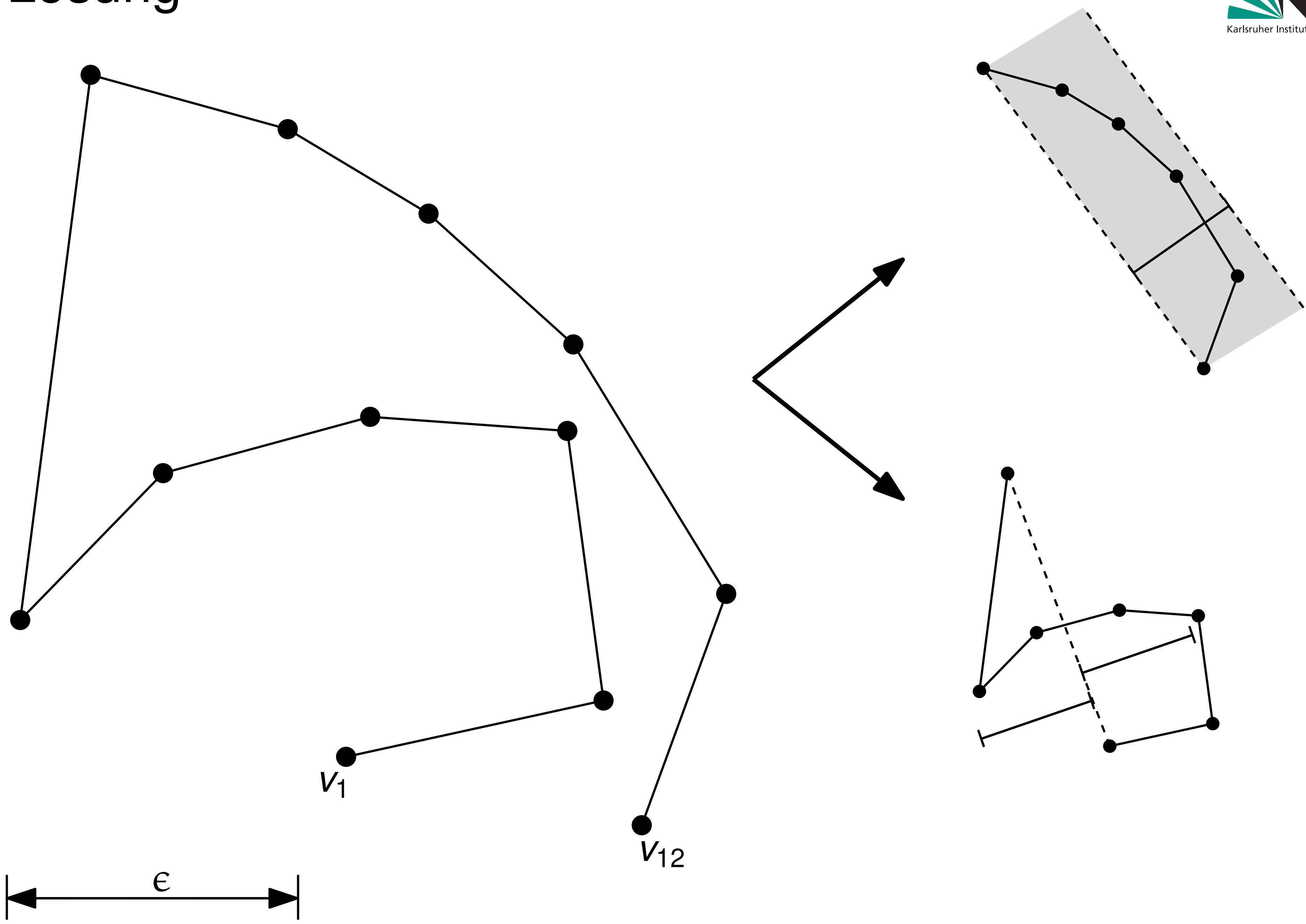
Lösung



Lösung

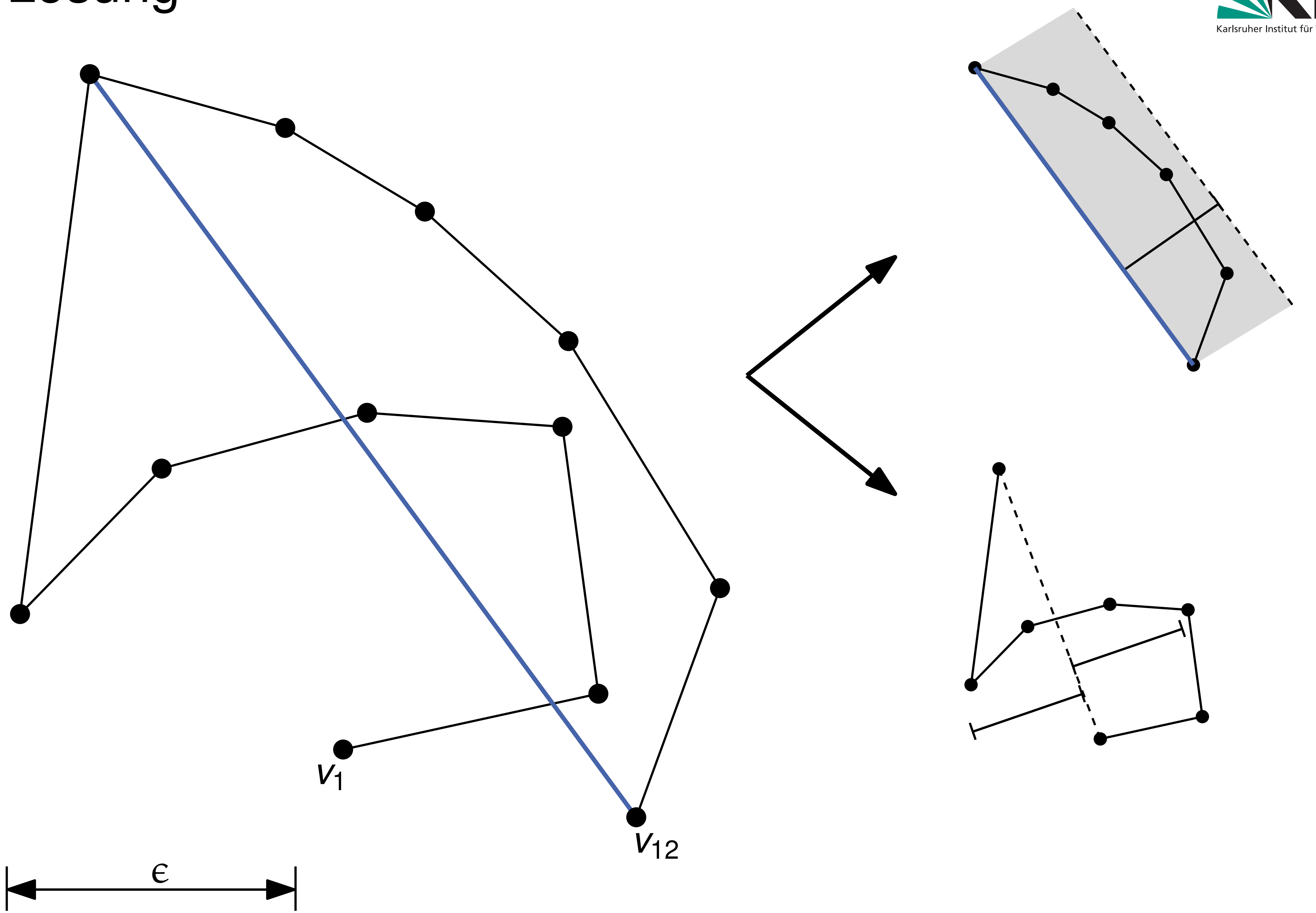


Lösung

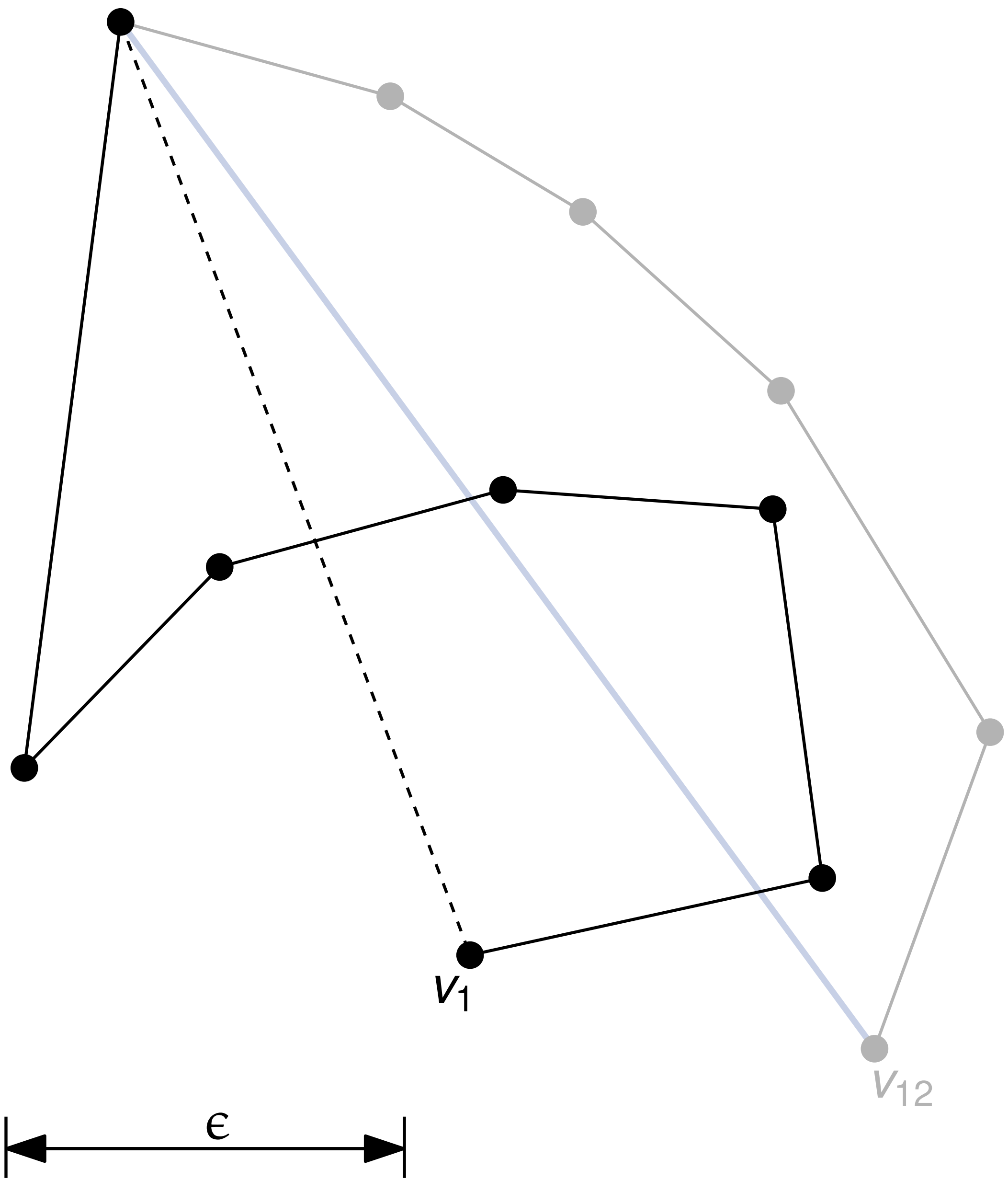


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

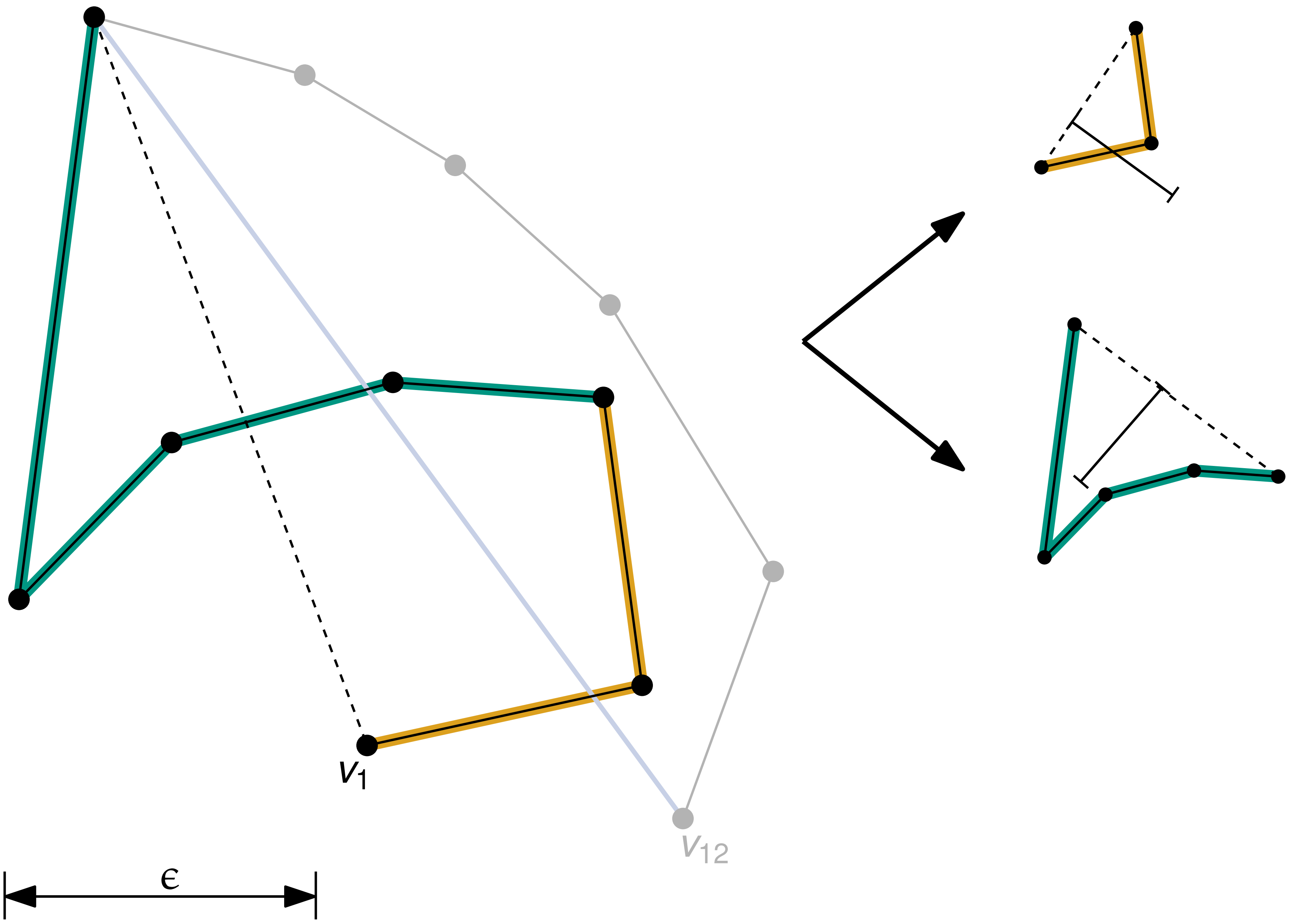
Lösung



Lösung

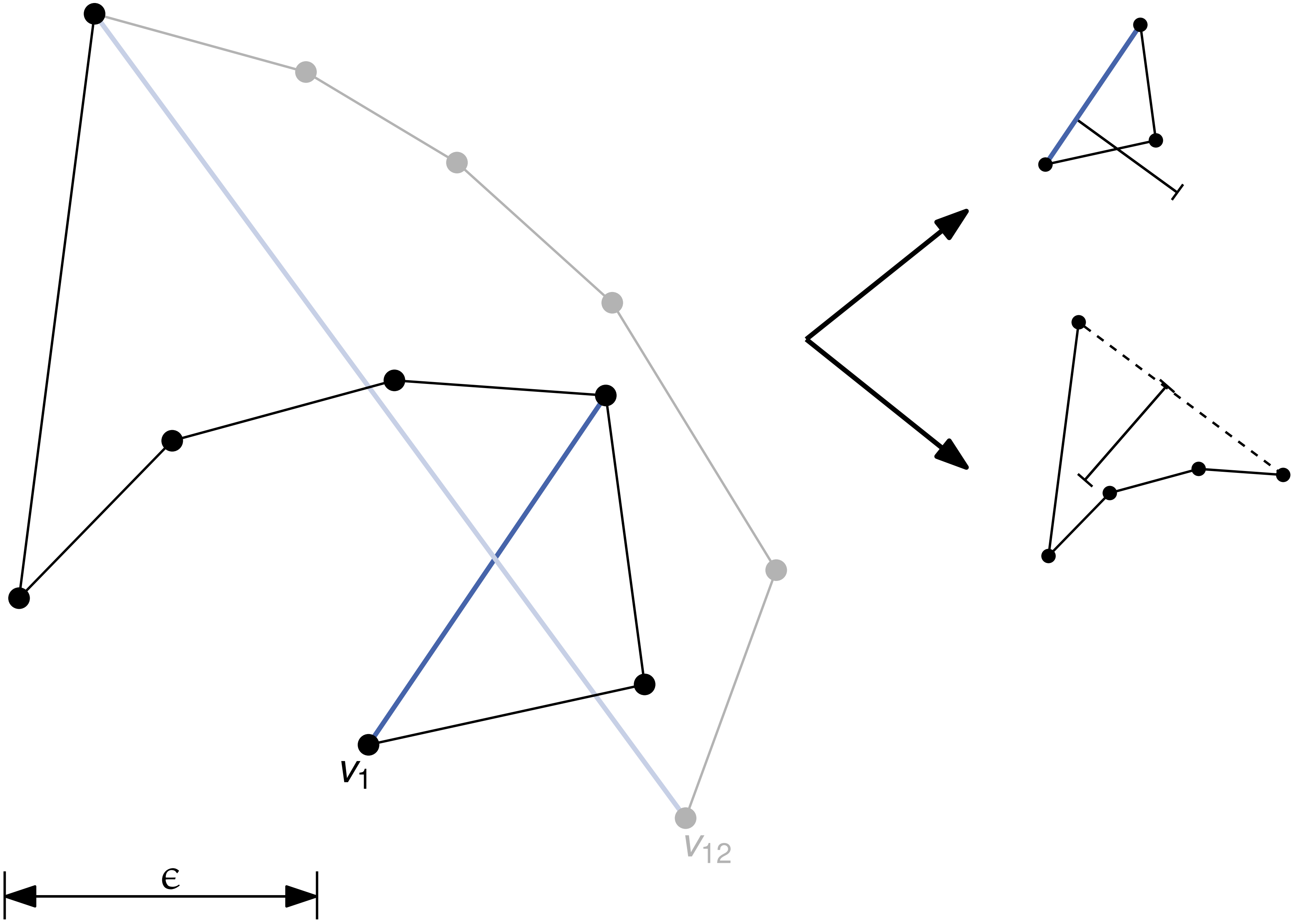


Lösung

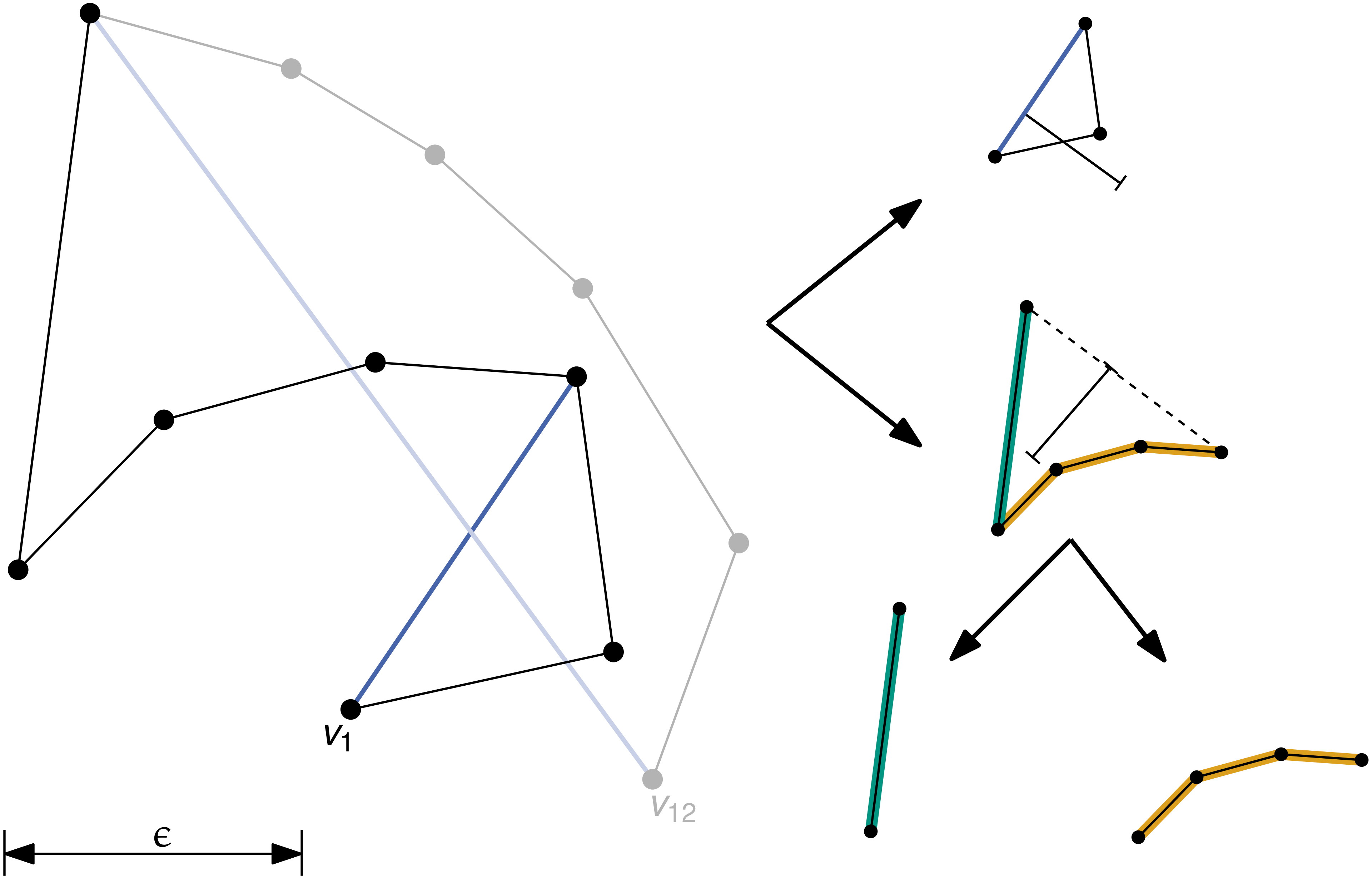


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

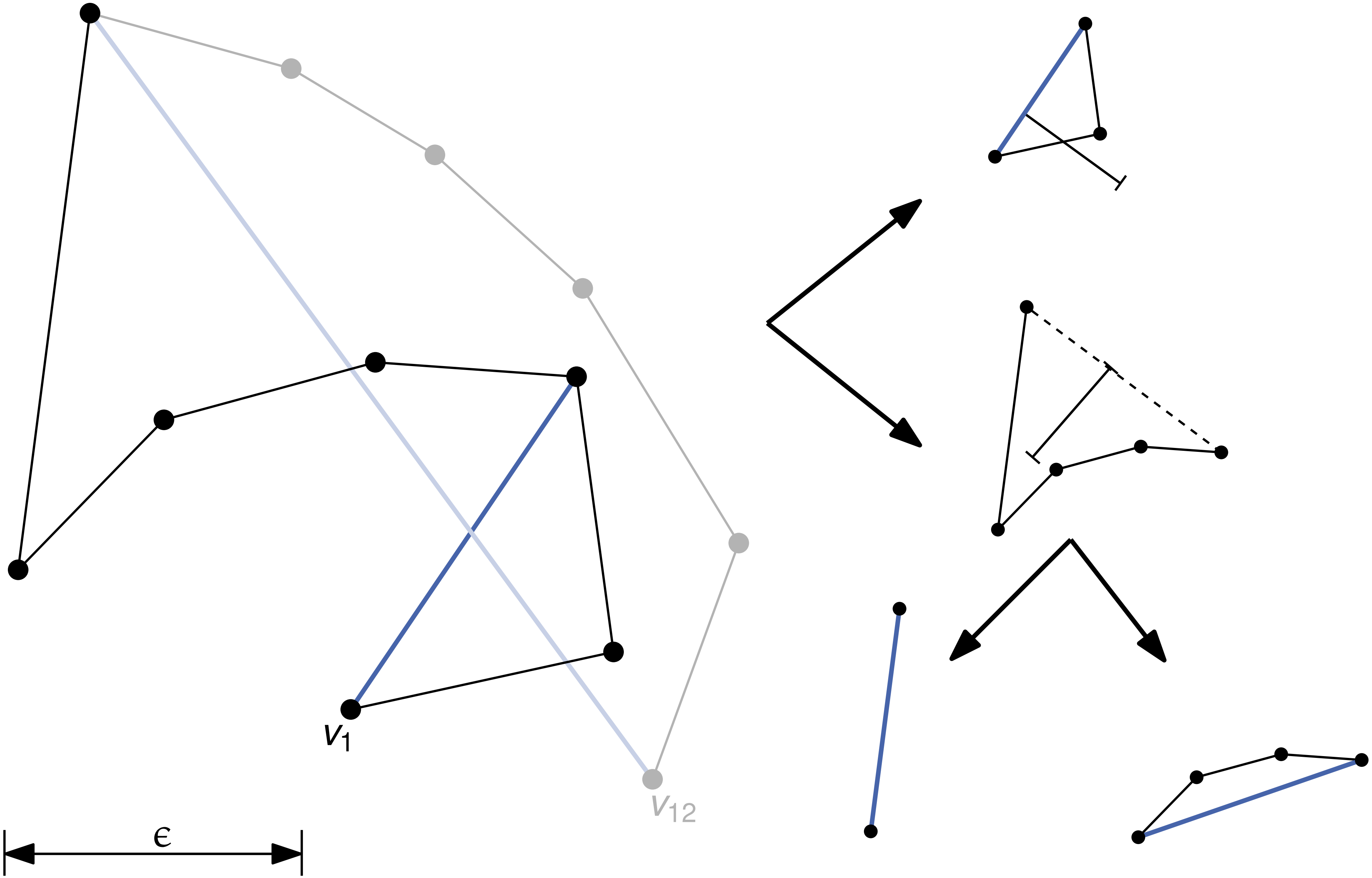
Lösung



Lösung

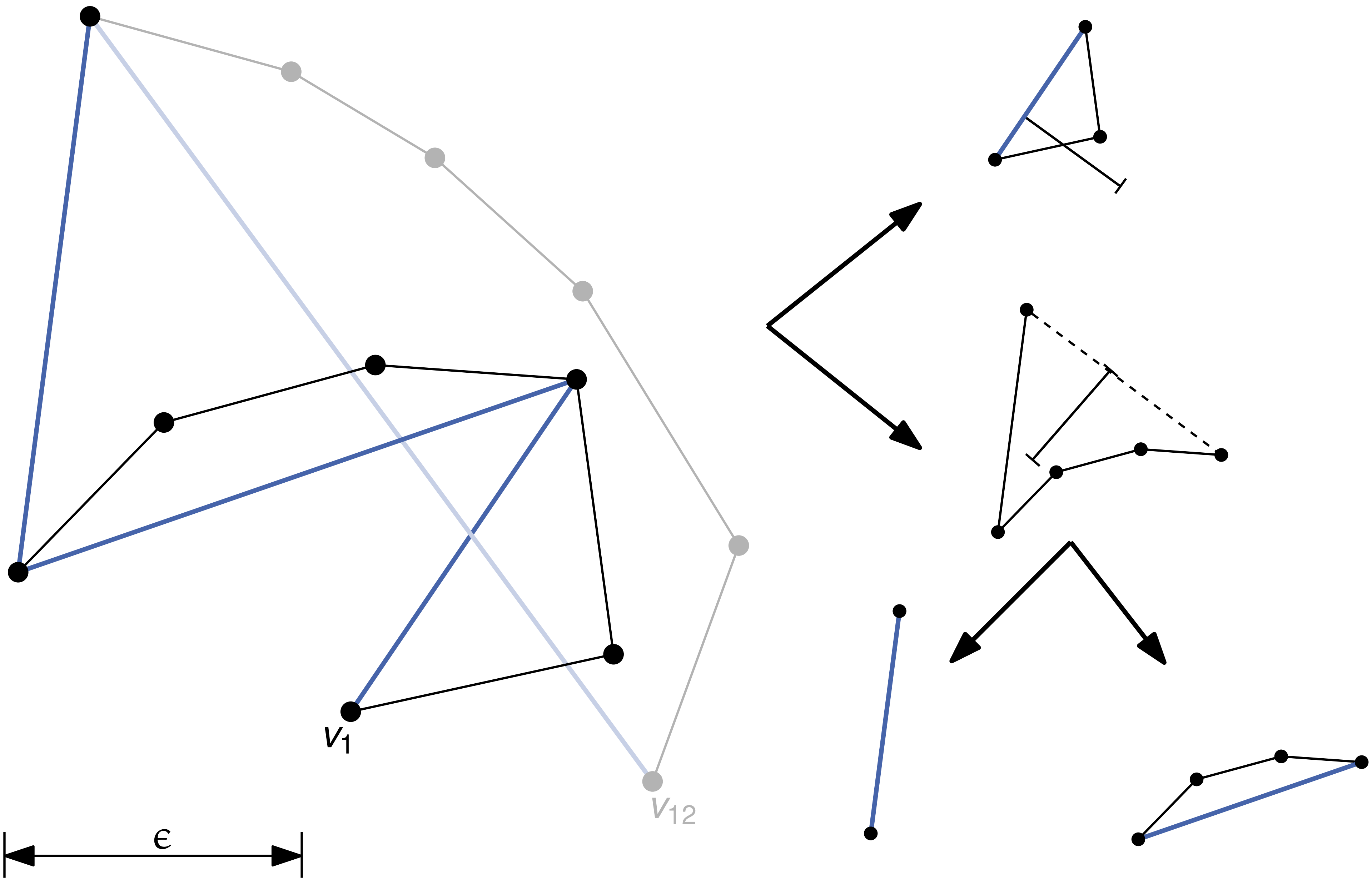


Lösung



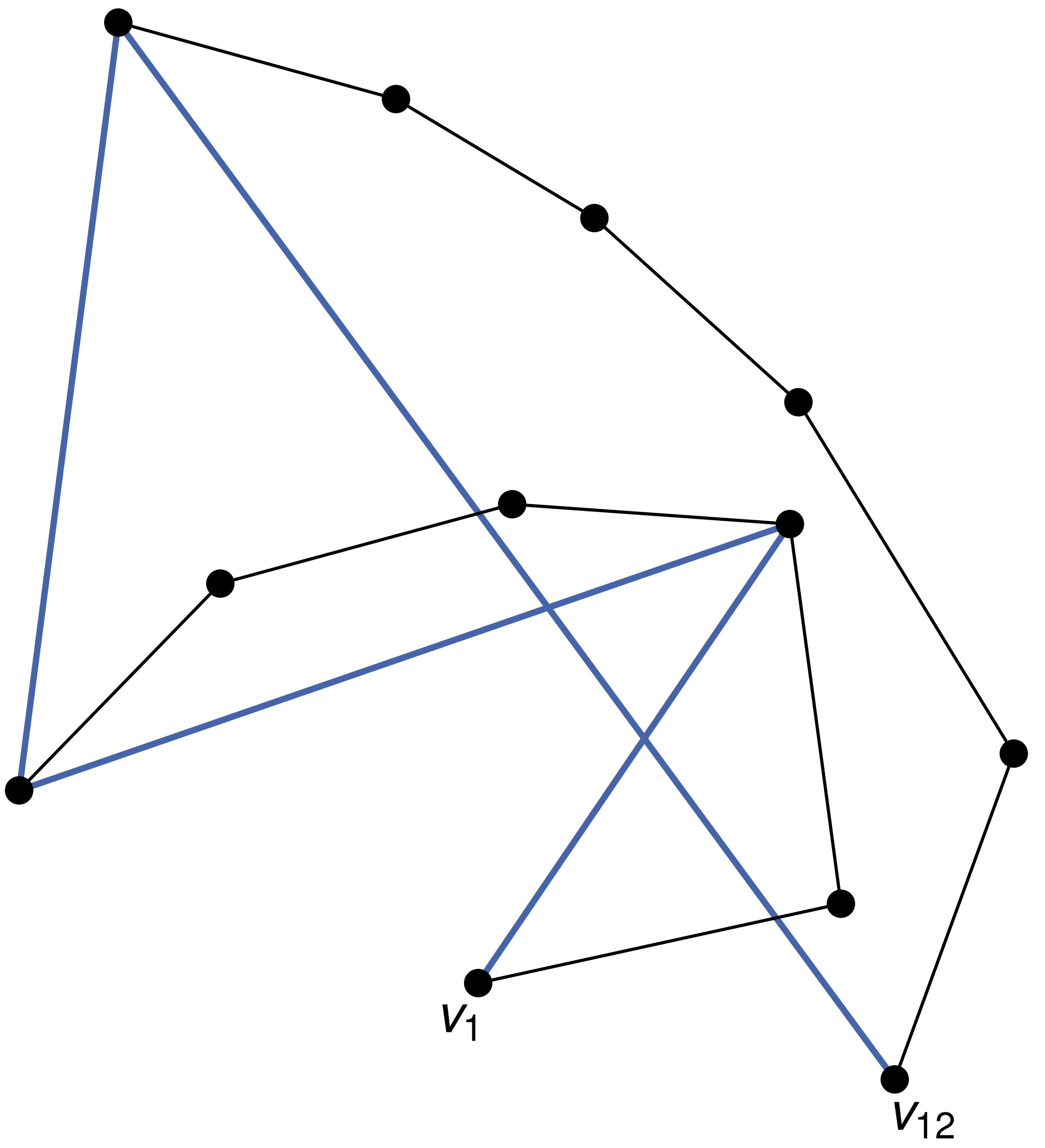
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Lösung



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Lösung

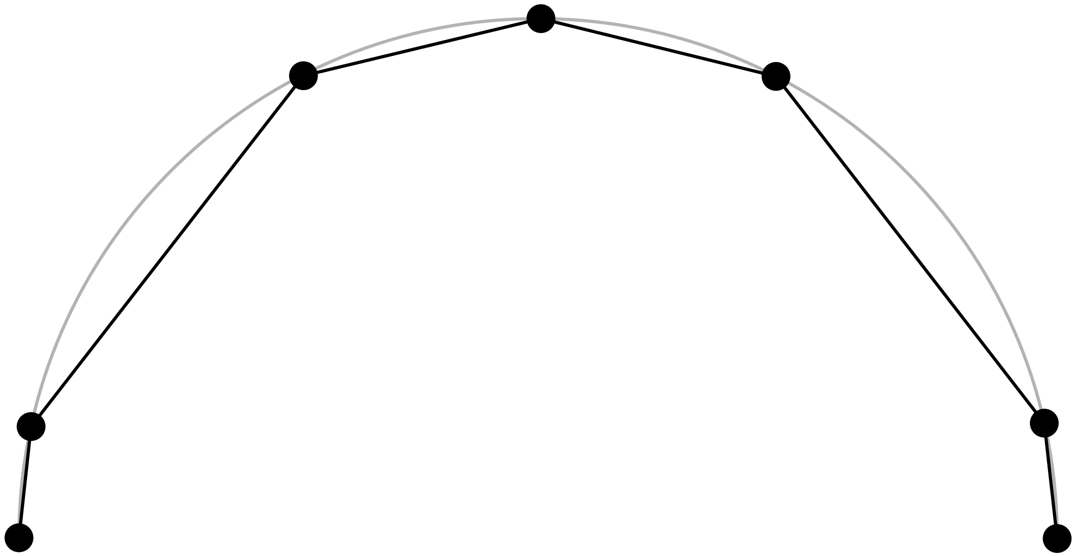


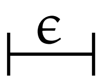
Aufgaben

Aufgabe:
Liefert der Algorithmus für jeden Polygonzug und jedes ϵ eine optimale Lösung, d.h. einen Polygonzug mit minimal vielen Kanten?

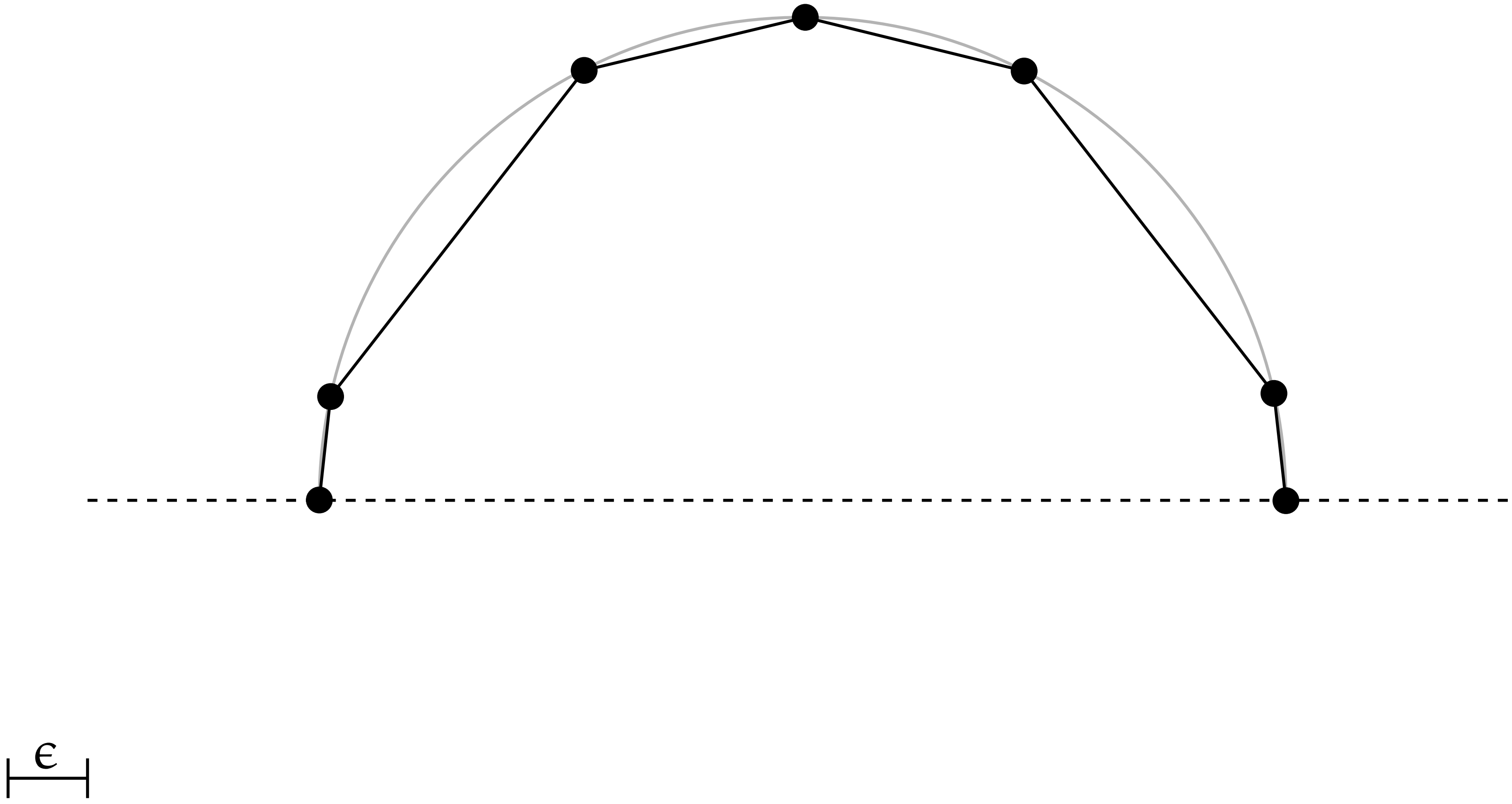
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Lösung

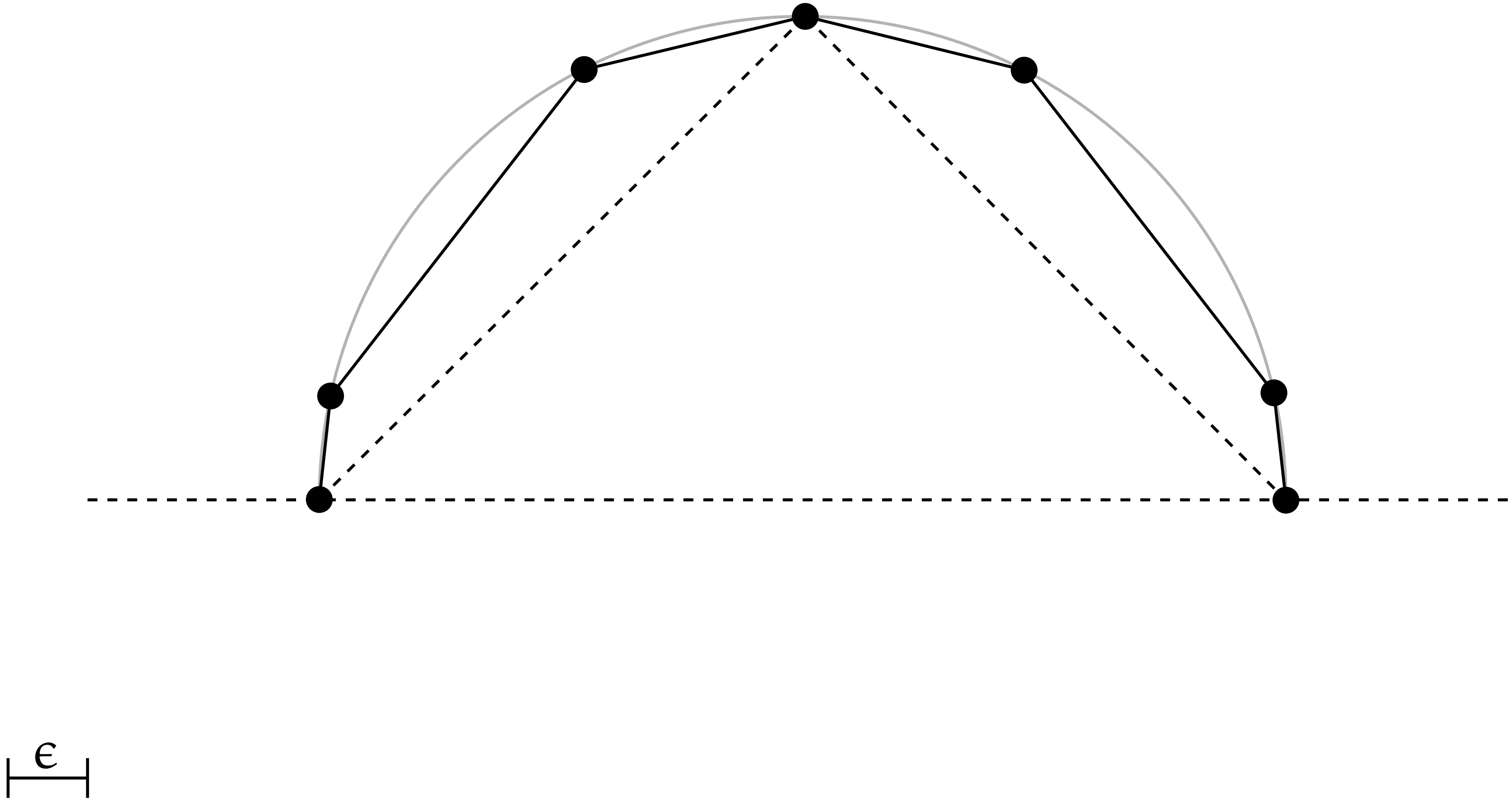




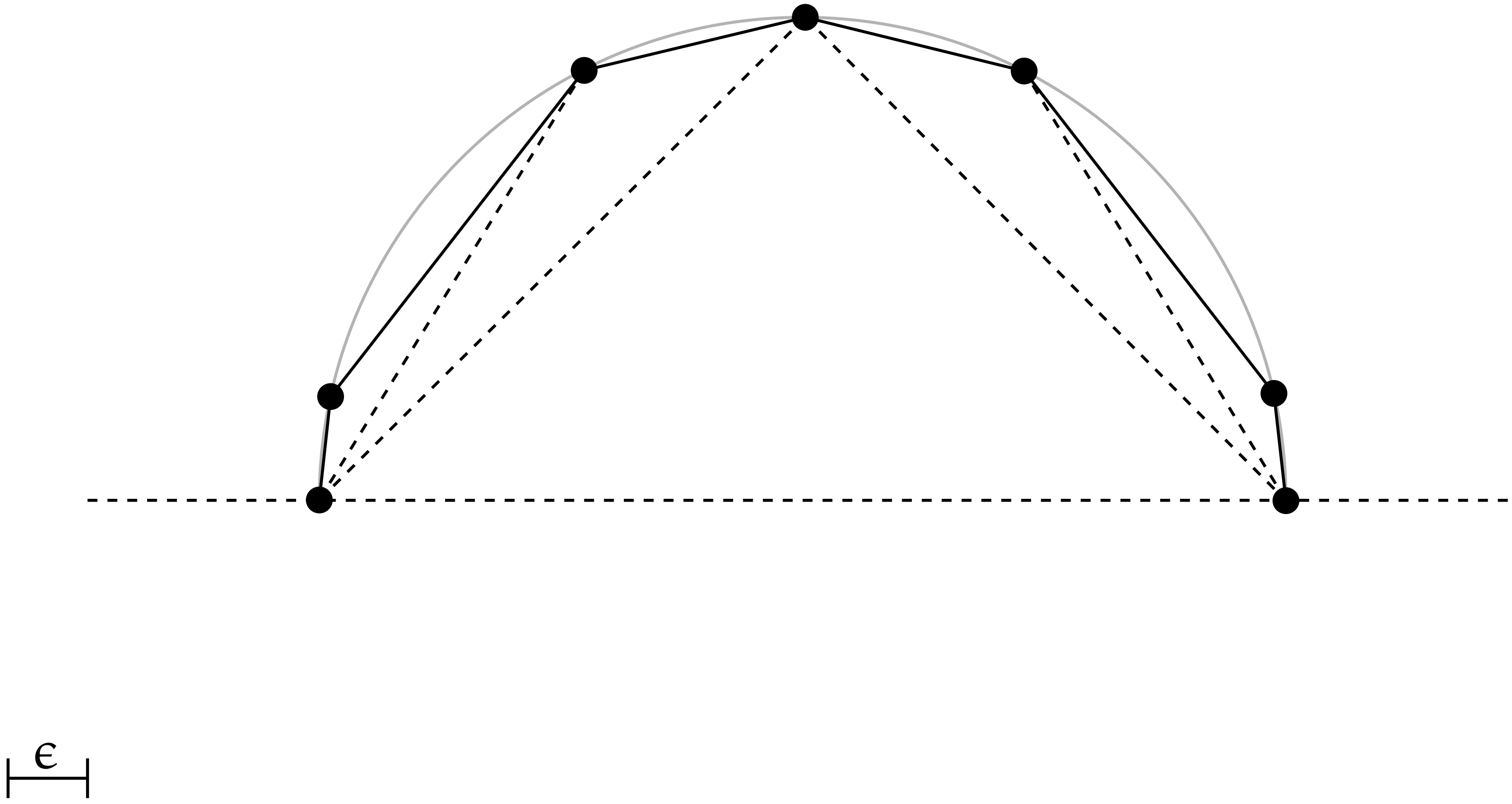
Lösung



Lösung

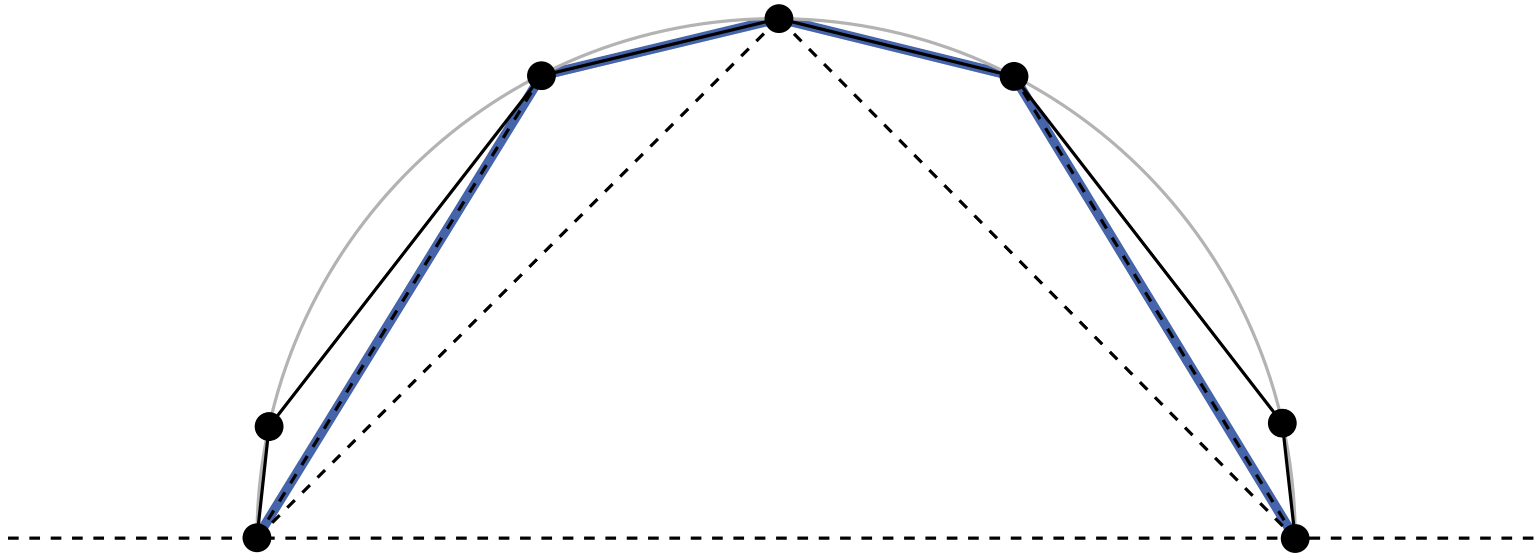


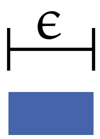
Lösung



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

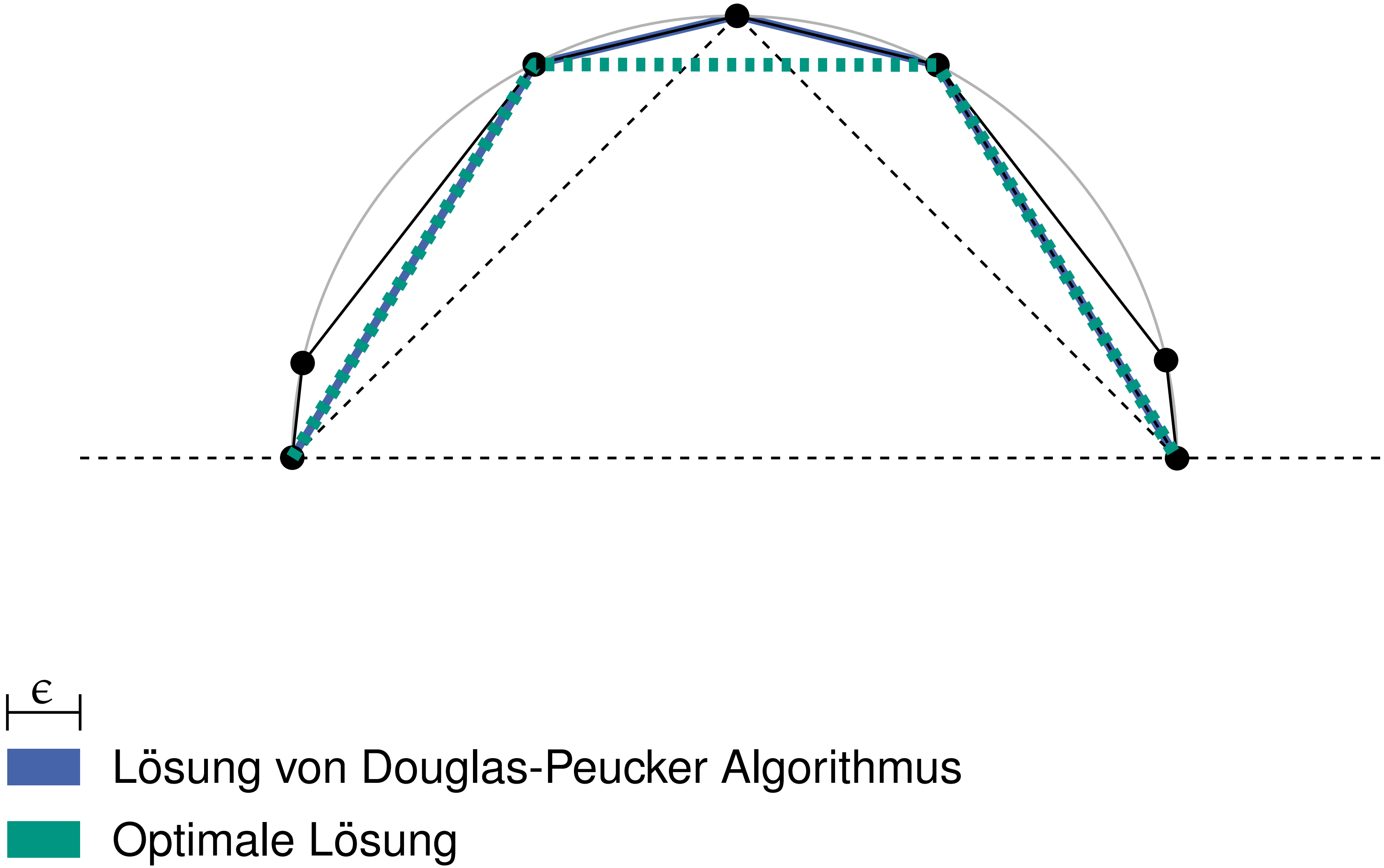
Lösung





Lösung von Douglas-Peucker Algorithmus

Lösung

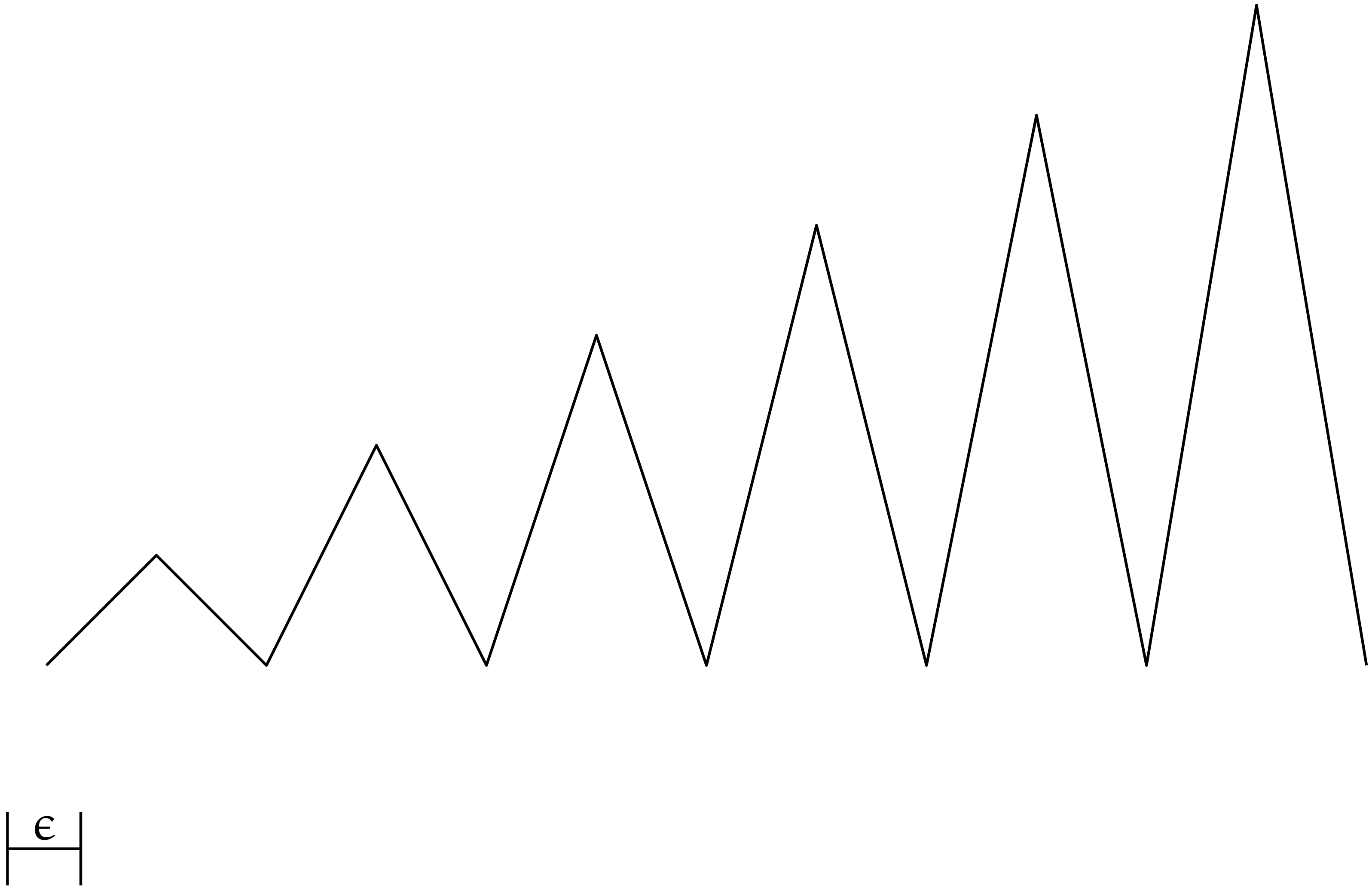


ϵ
Lösung von Douglas-Peucker Algorithmus
Optimale Lösung

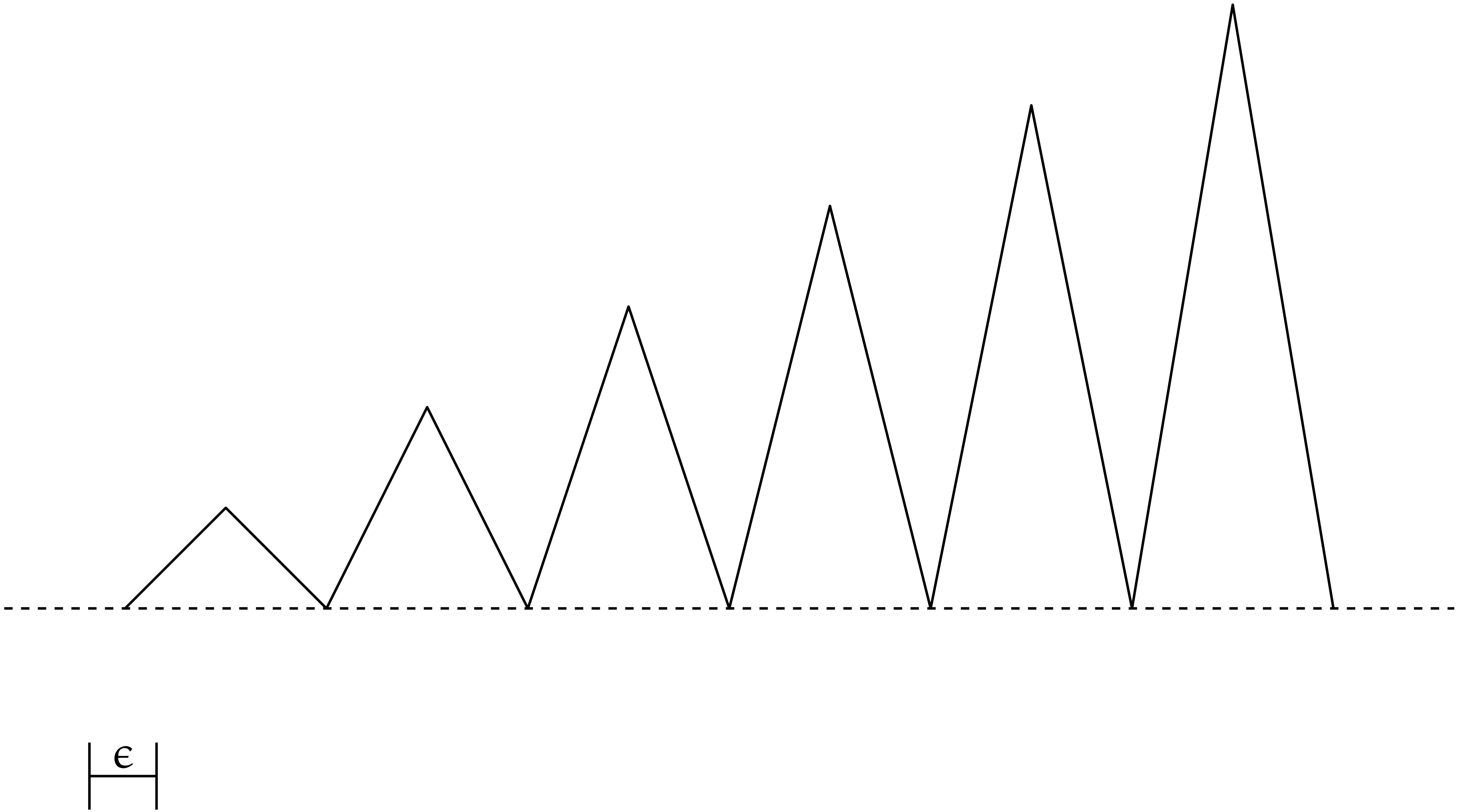
Aufgaben

Aufgabe:
Geben Sie einen Polygonzug an, für den der Algorithmus $\Omega(n^2)$ Zeit benötigt.

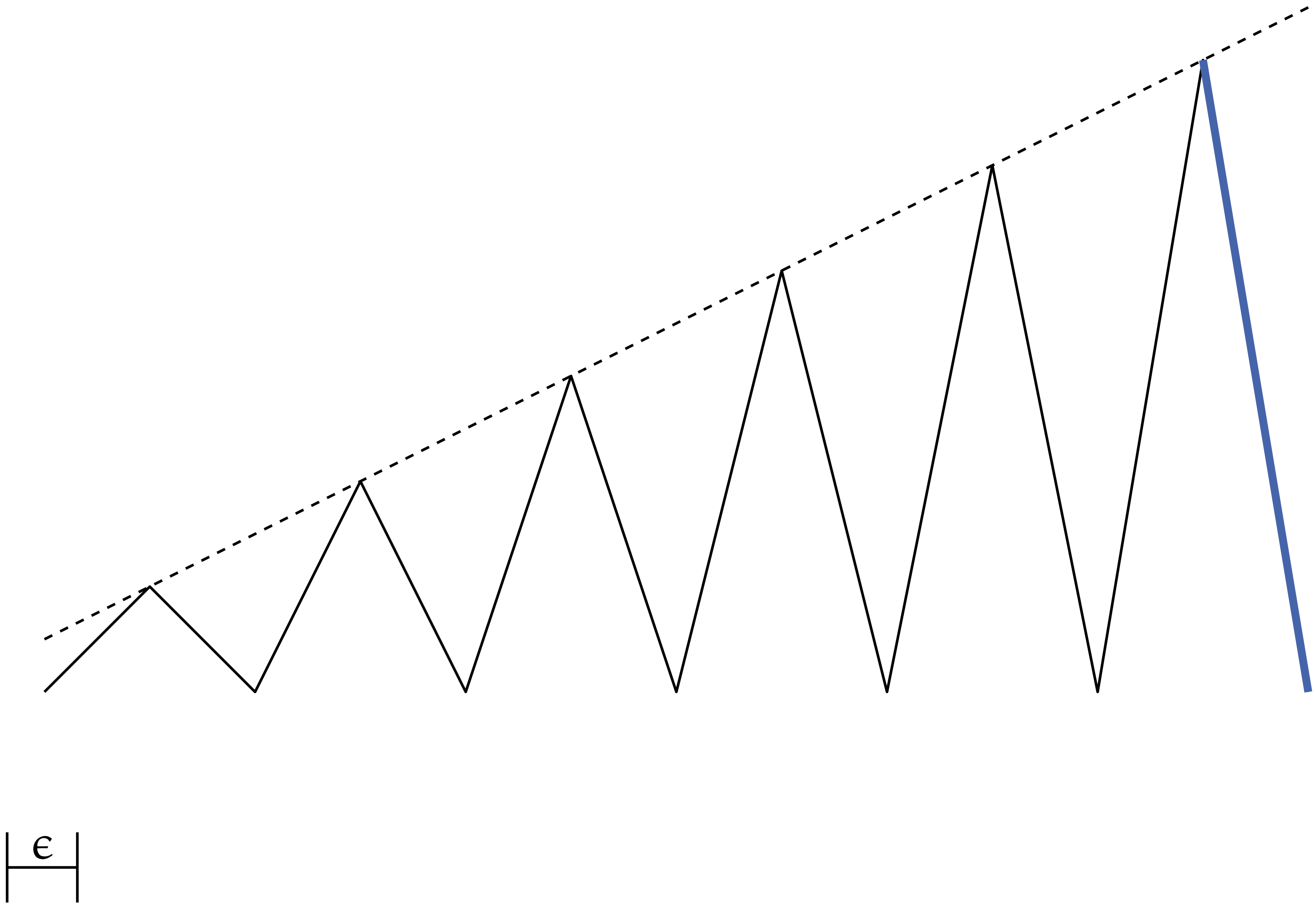
Lösung



Lösung

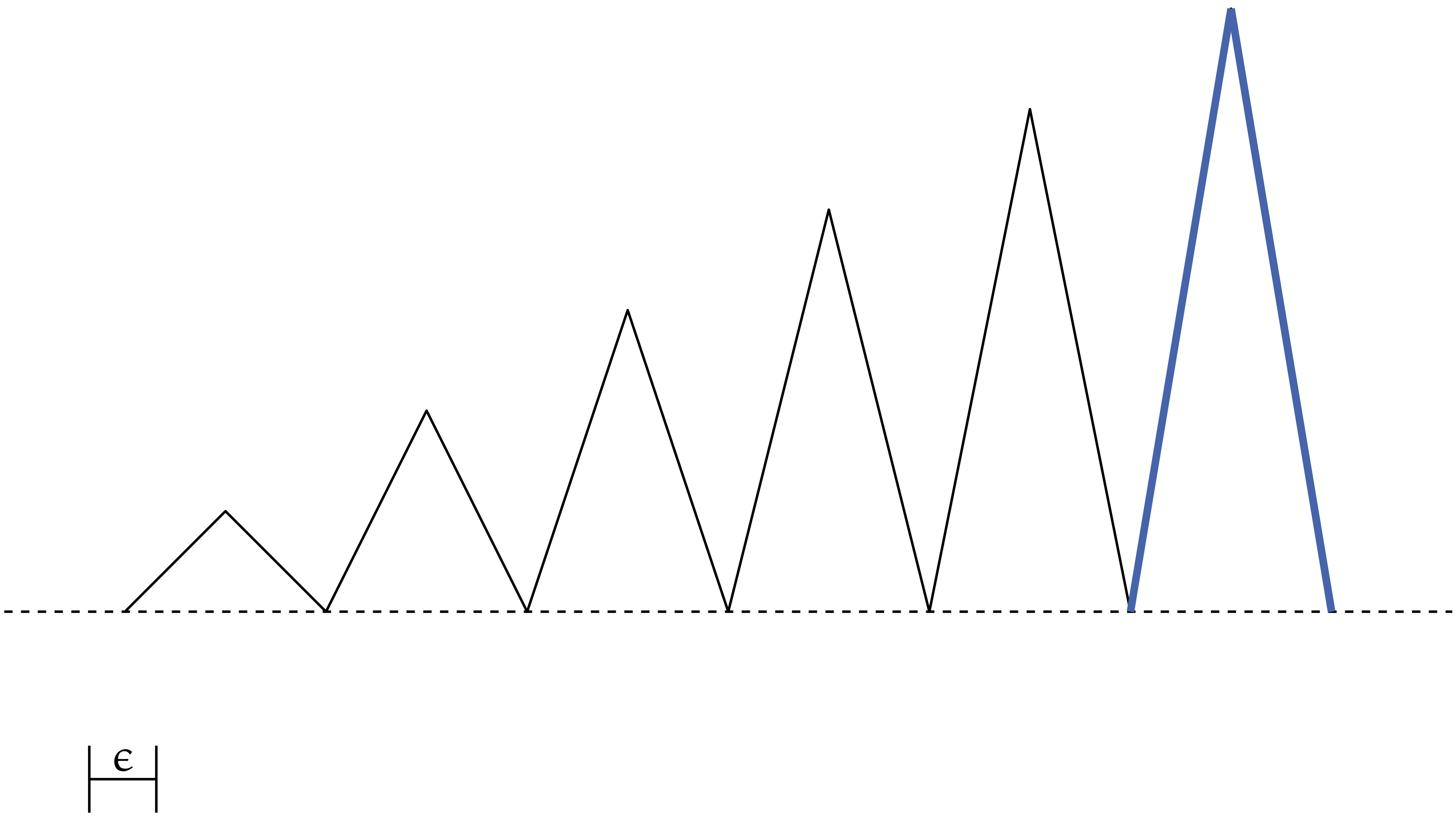


Lösung



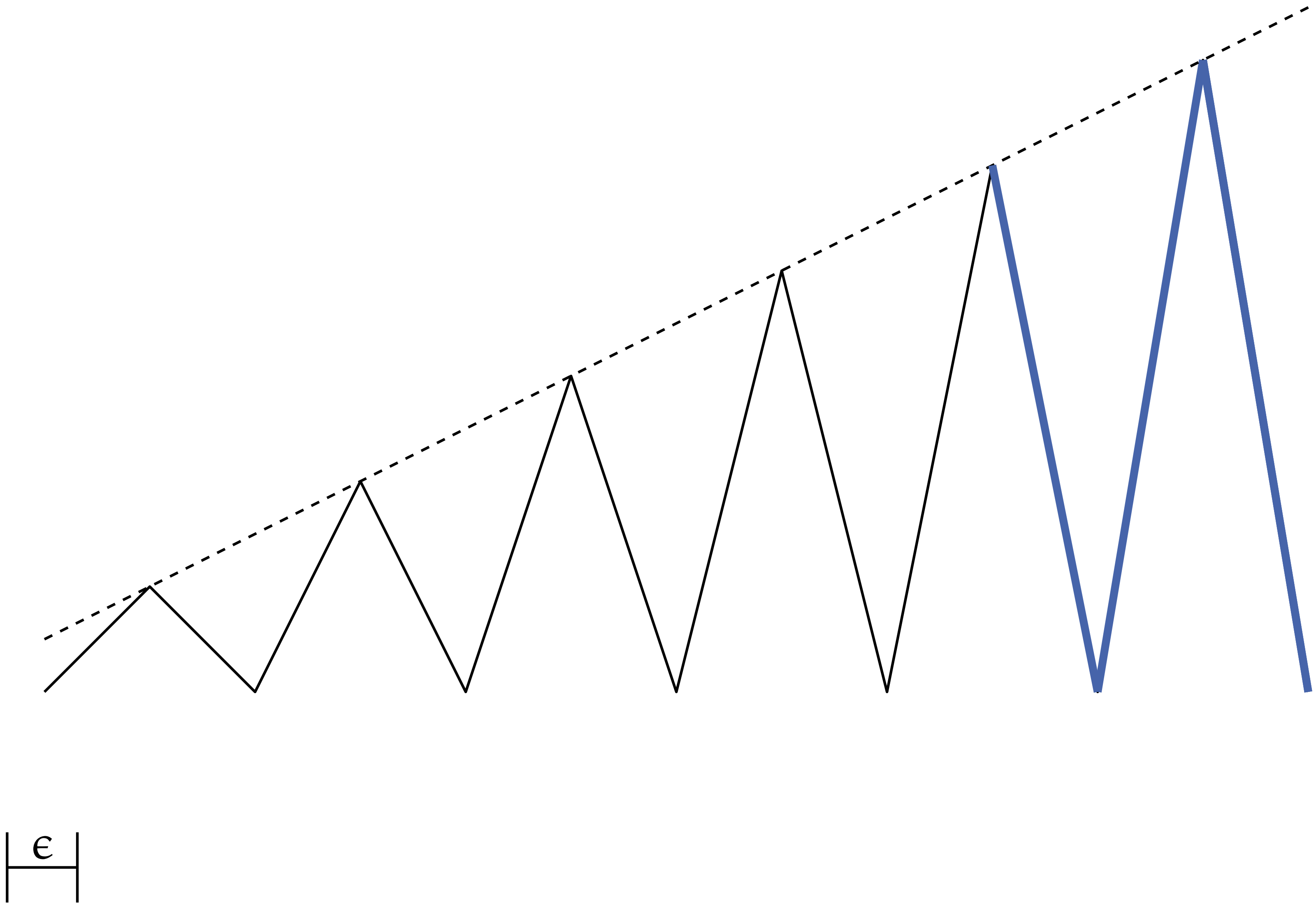
Lösung

Karlsruher Institut für Technologie

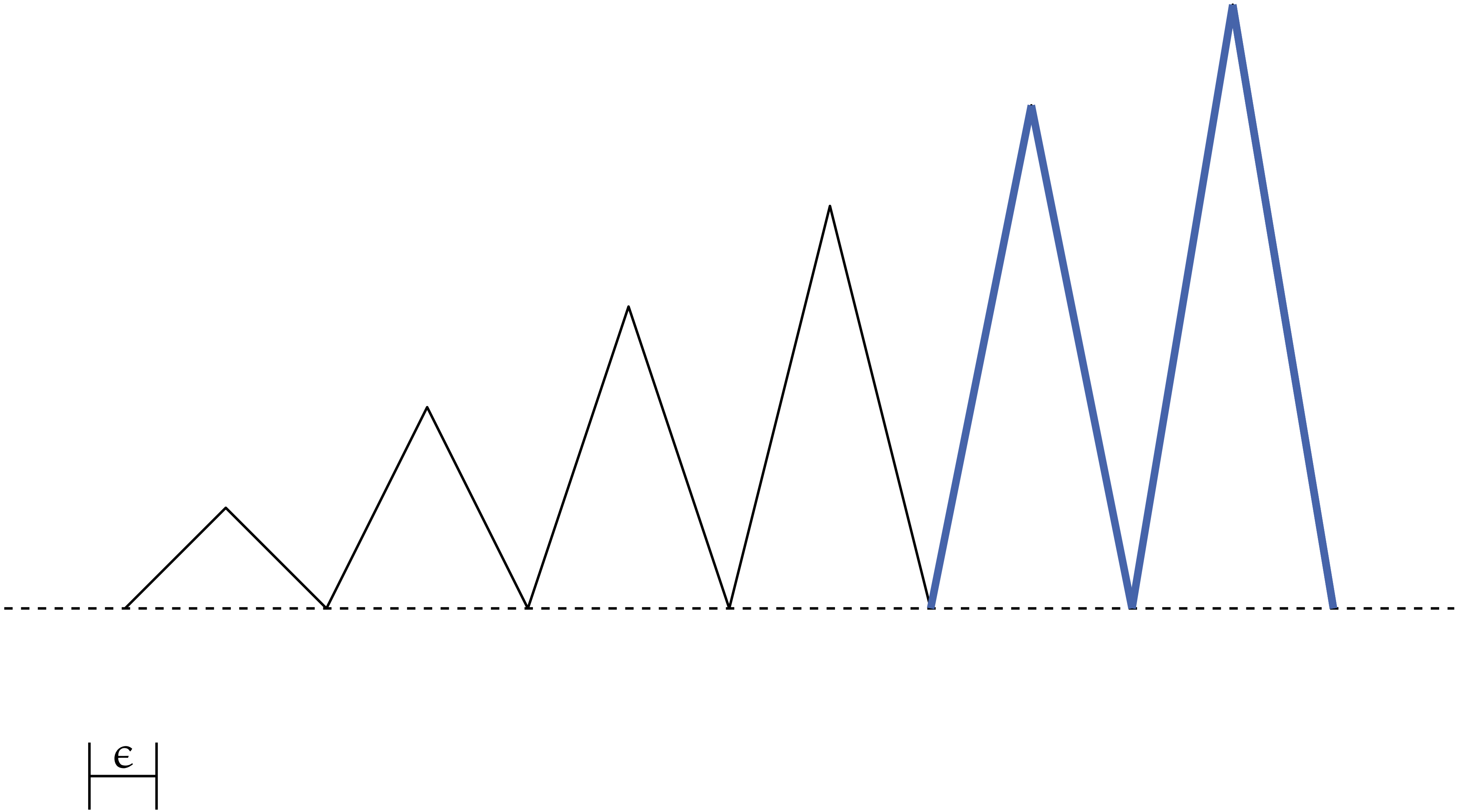


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Lösung



Lösung



Grundlegende Datenstrukturen und Techniken

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung der Ebene


Quelle: Google Earth

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung der Ebene

Quelle: Google Earth

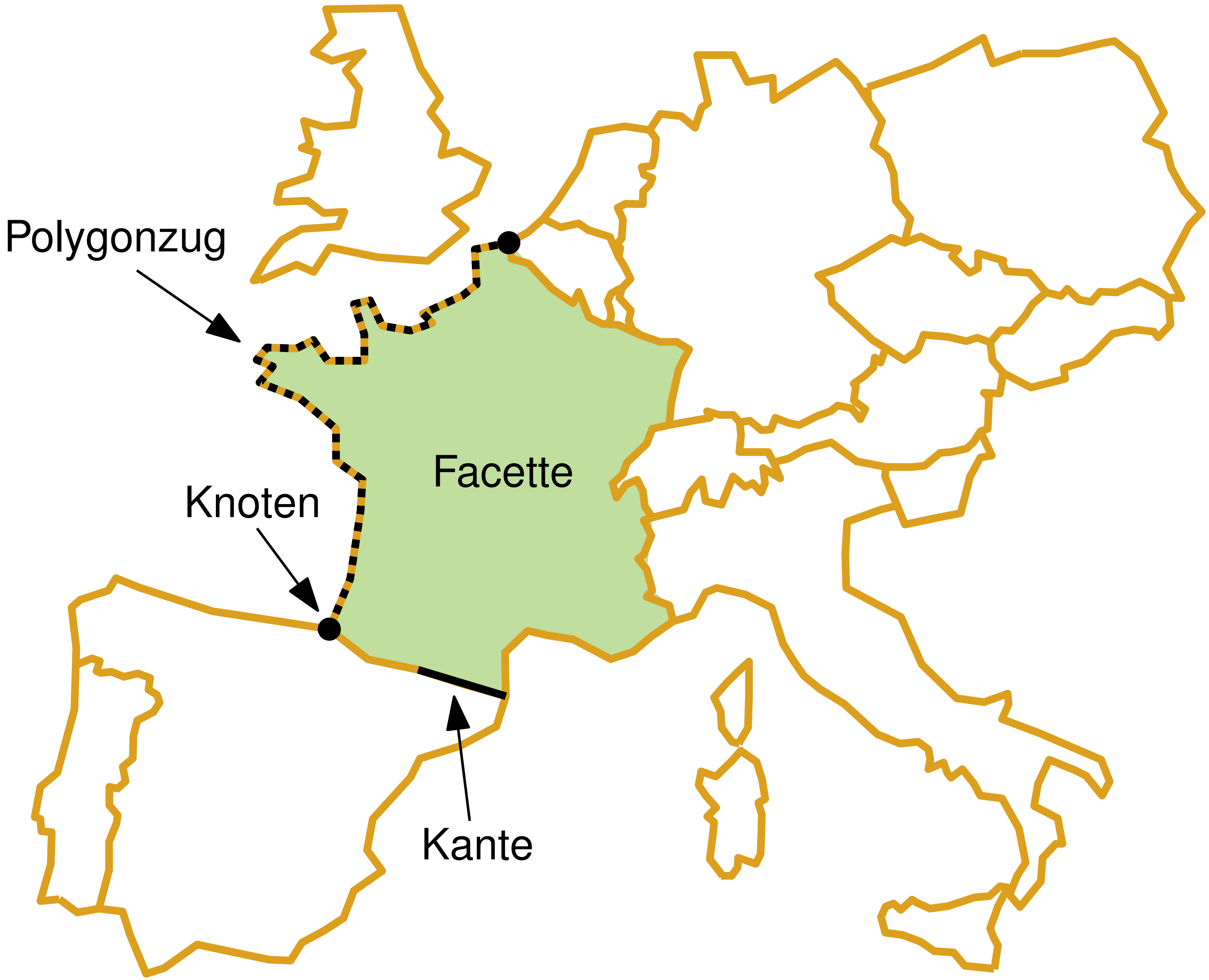
Unterteilung der Ebene

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung der Ebene

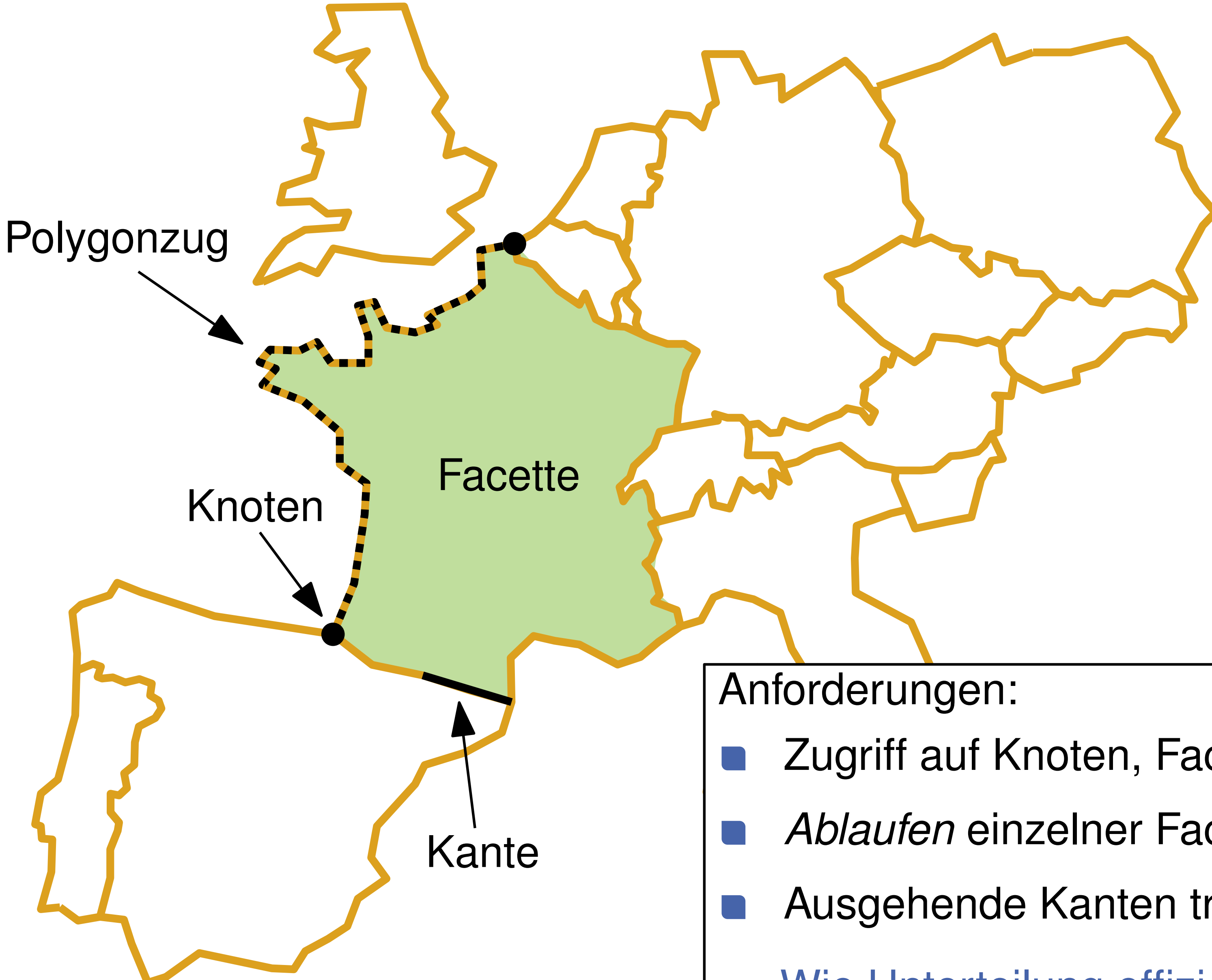
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung der Ebene



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung der Ebene



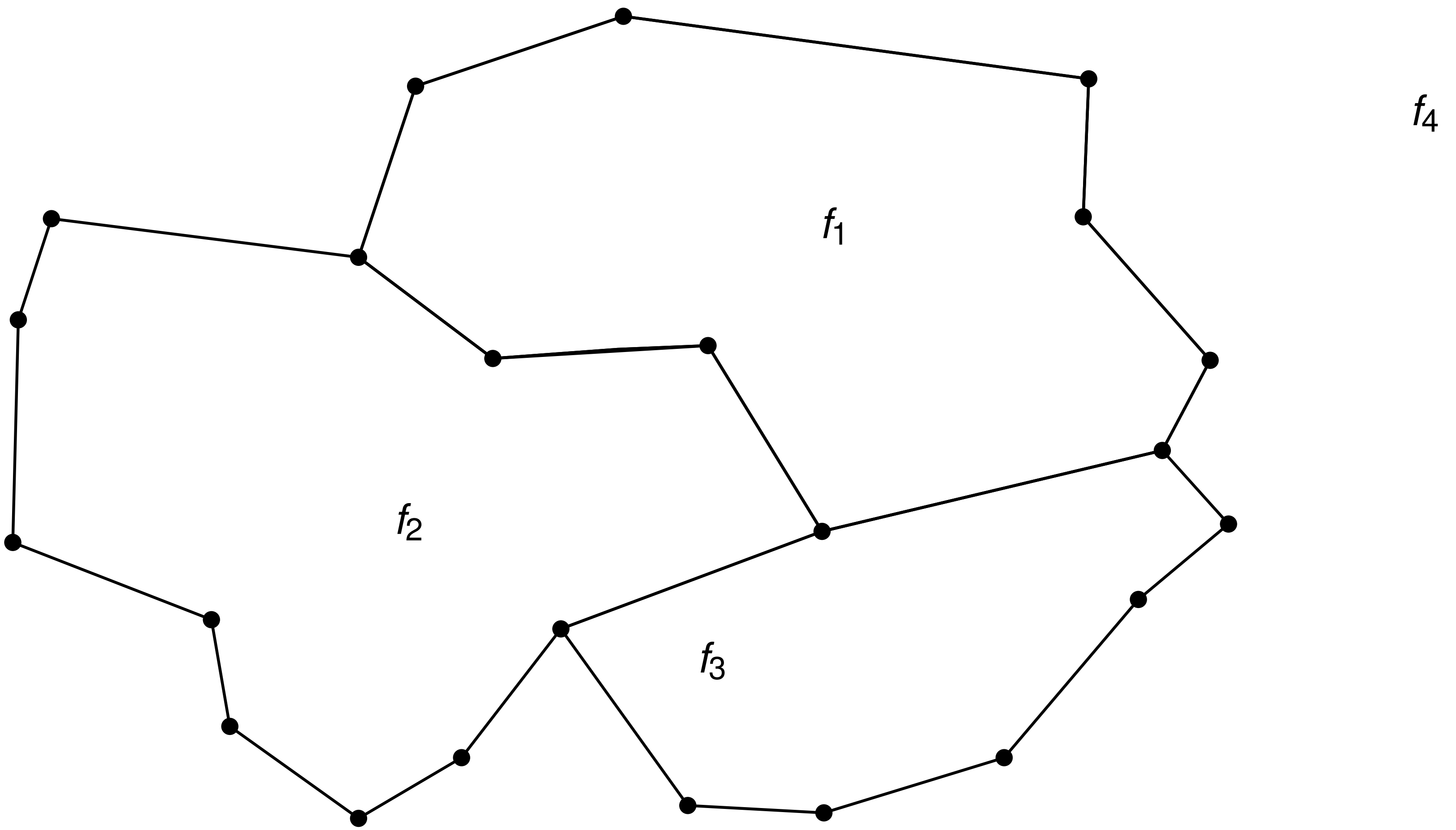
Anforderungen:

- Zugriff auf Knoten, Facetten und Kanten
- *Ablaufen* einzelner Facetten.
- Ausgehende Kanten traversieren.

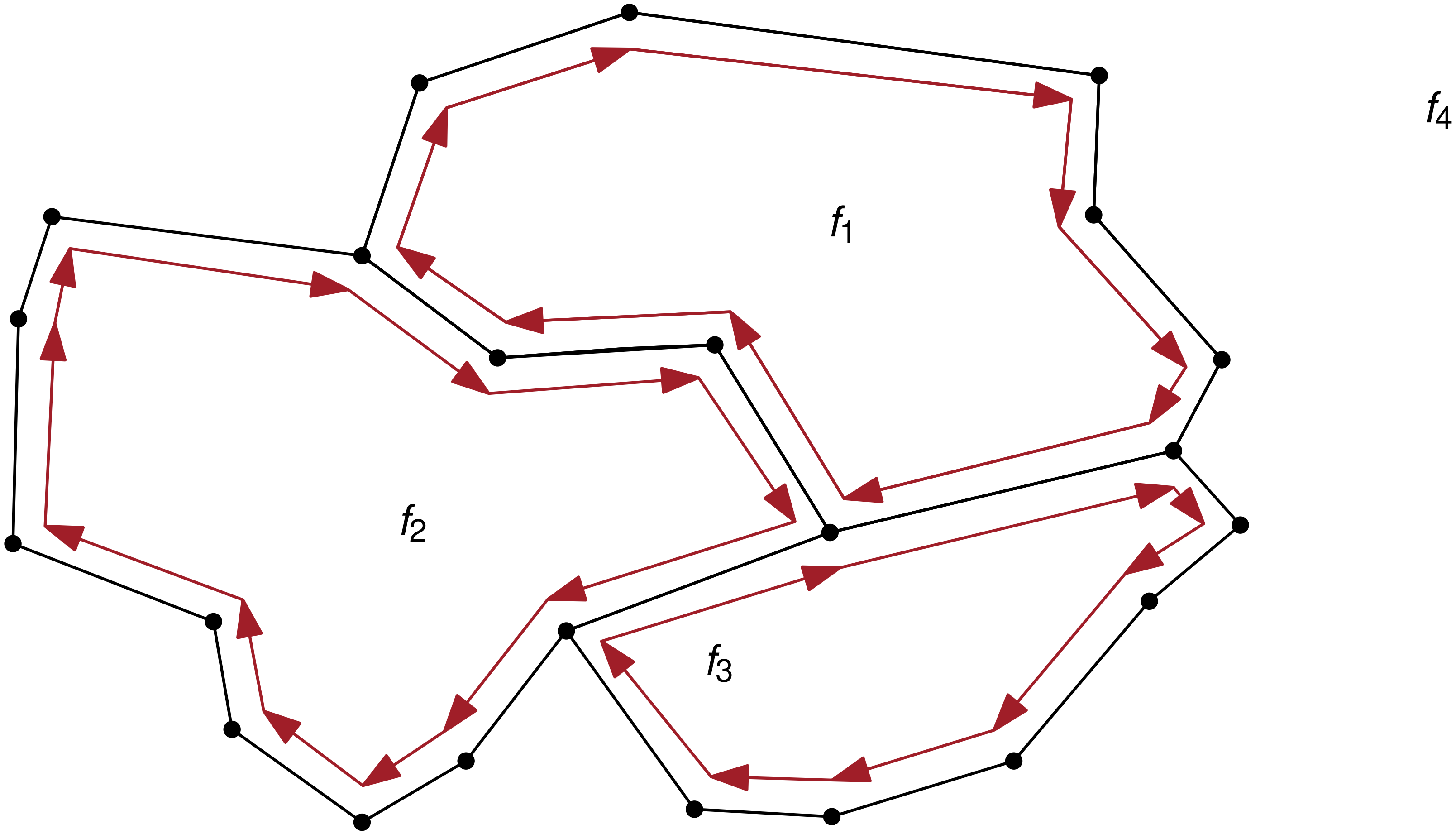
Wie Unterteilung effizient speichern?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung

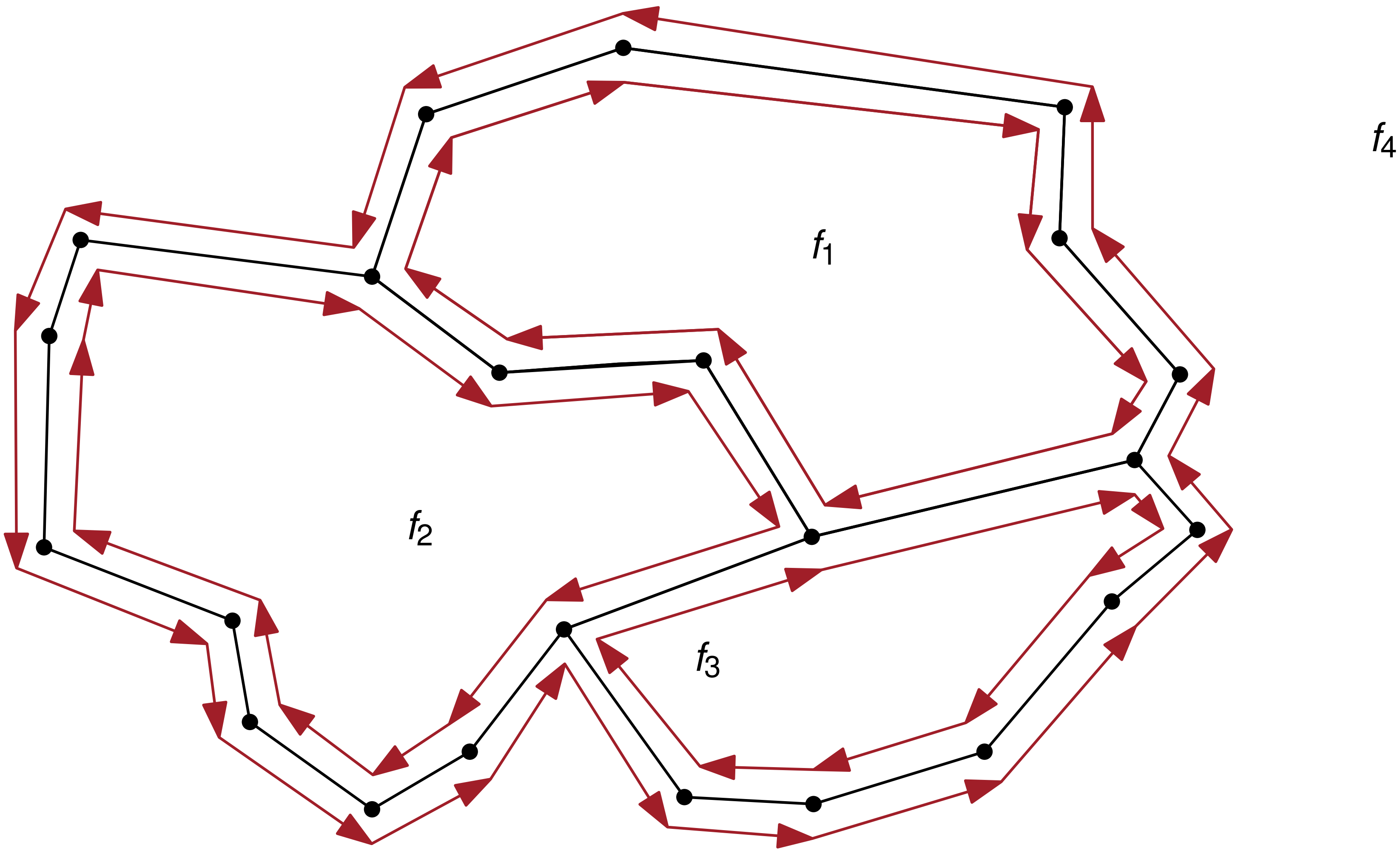


Unterteilung



Für jede Kante einer inneren Facette führe gerichtete Halbkante im Uhrzeigersinn ein.

Unterteilung

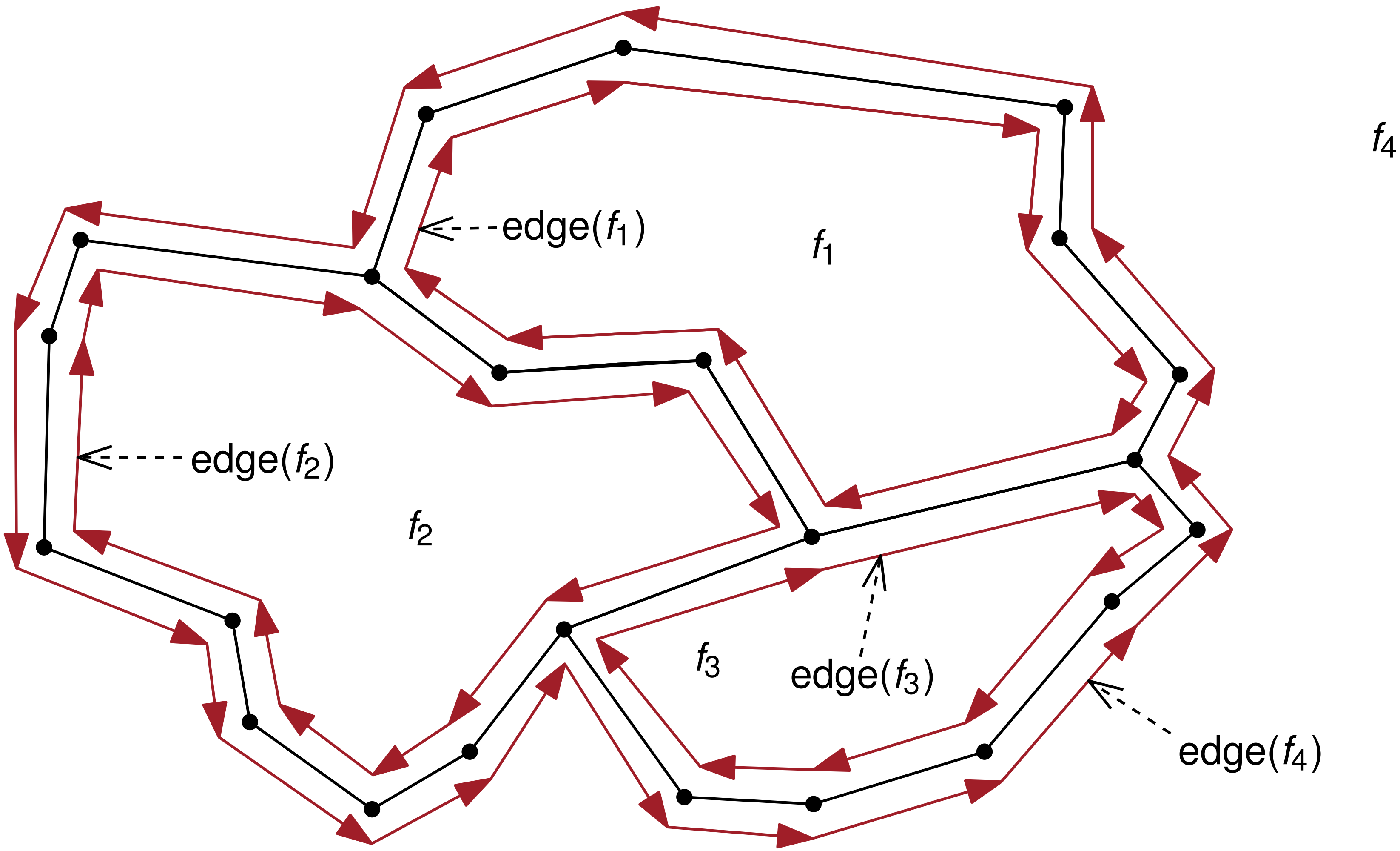


Für jede Kante einer inneren Facette führe gerichtete Halbkante im Uhrzeigersinn ein.

Für jede Kante der äußeren Facette führe gerichtete Halbkante gegen den Uhrzeigersinn ein.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

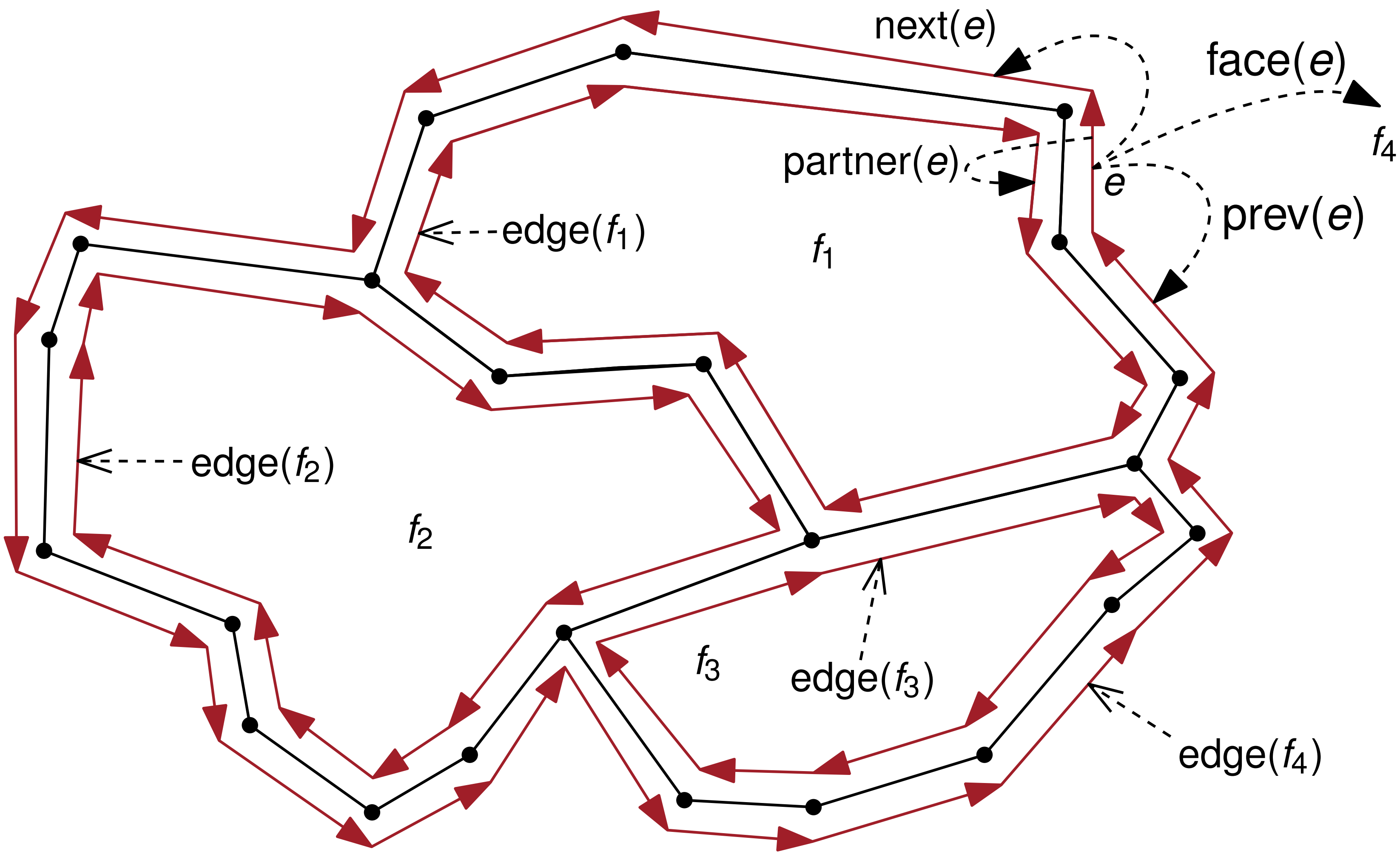
Unterteilung



Speichere für jede Facette eine beliebige angrenzende Halbkante.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung

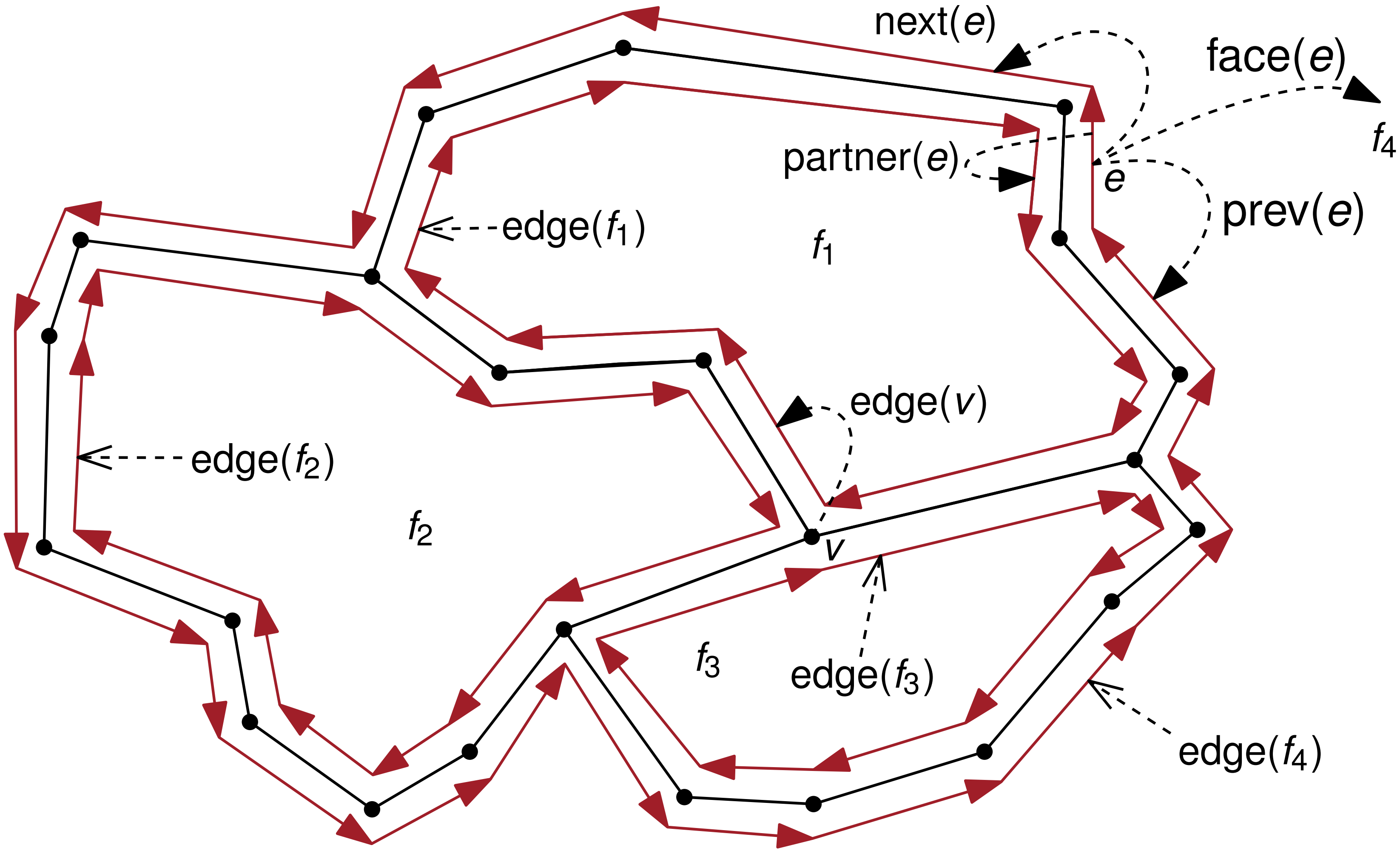


Speichere für jede Facette eine beliebige angrenzende Halbkante.

Speichere für jede Halbkante den Nachfolger/Vorgänger, die gegenüberliegende Halbkante und die angrenzende Facette.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung



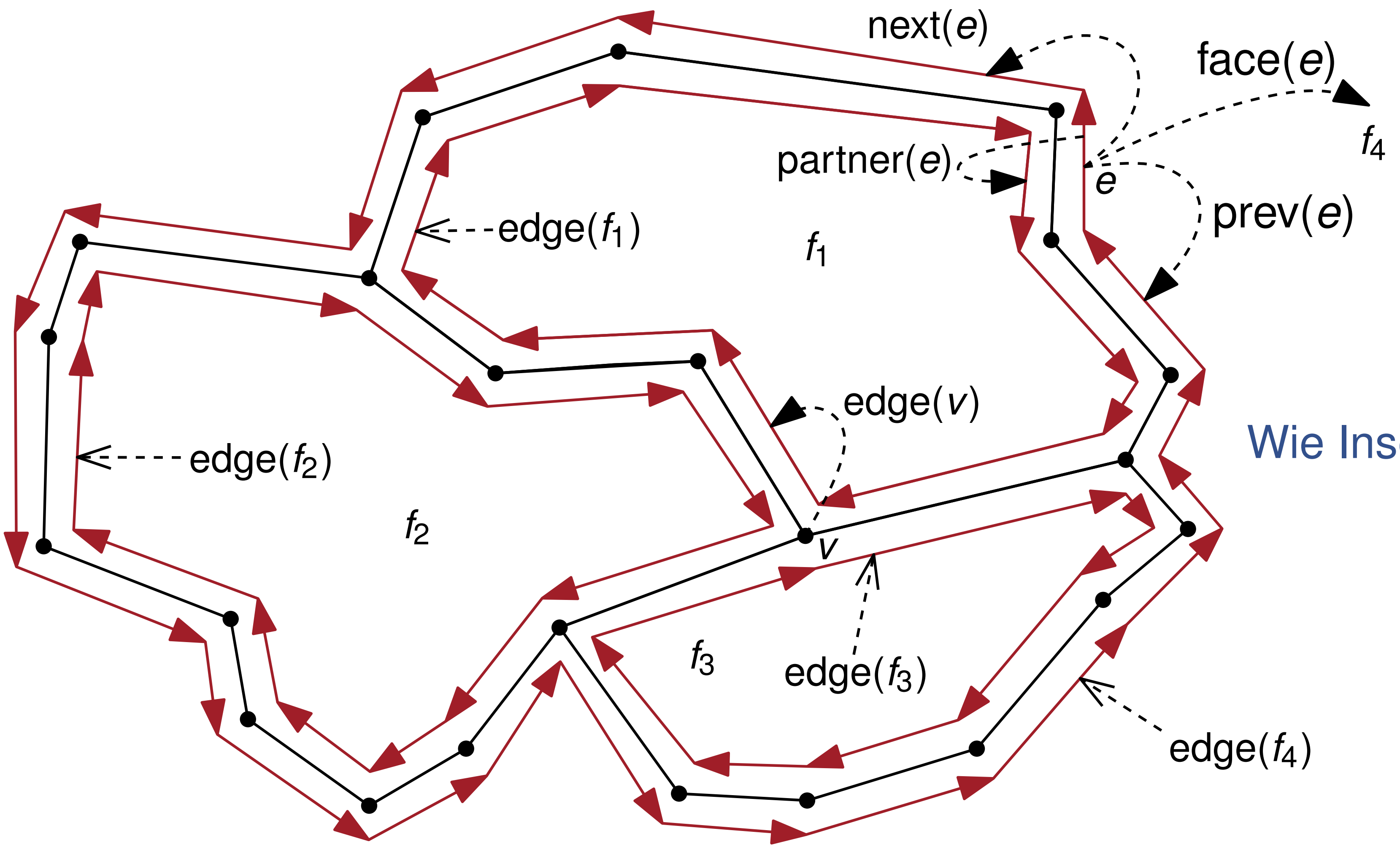
Speichere für jede Facette eine beliebige angrenzende Halbkante.

Speichere für jede Halbkante den Nachfolger/Vorgänger, die gegenüberliegende Halbkante und die angrenzende Facette.

Speichere für jeden Knoten eine beliebige inzidente ausgehende Halbkante.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung



Wie Inseln behandeln?

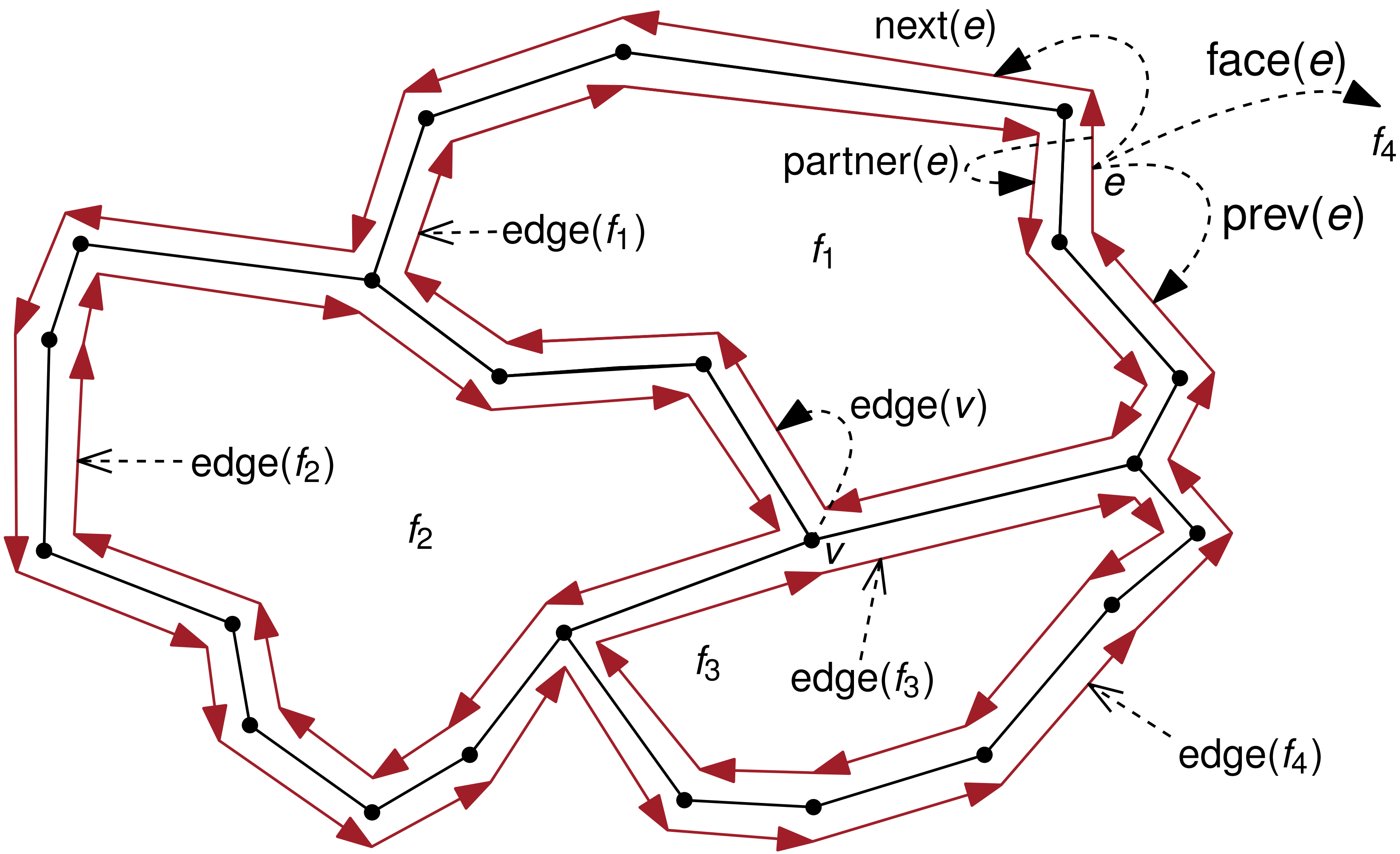
Speichere für jede Facette eine beliebige angrenzende Halbkante.

Speichere für jede Halbkante den Nachfolger/Vorgänger, die gegenüberliegende Halbkante und die angrenzende Facette.

Speichere für jeden Knoten eine beliebige inzidente ausgehende Halbkante.

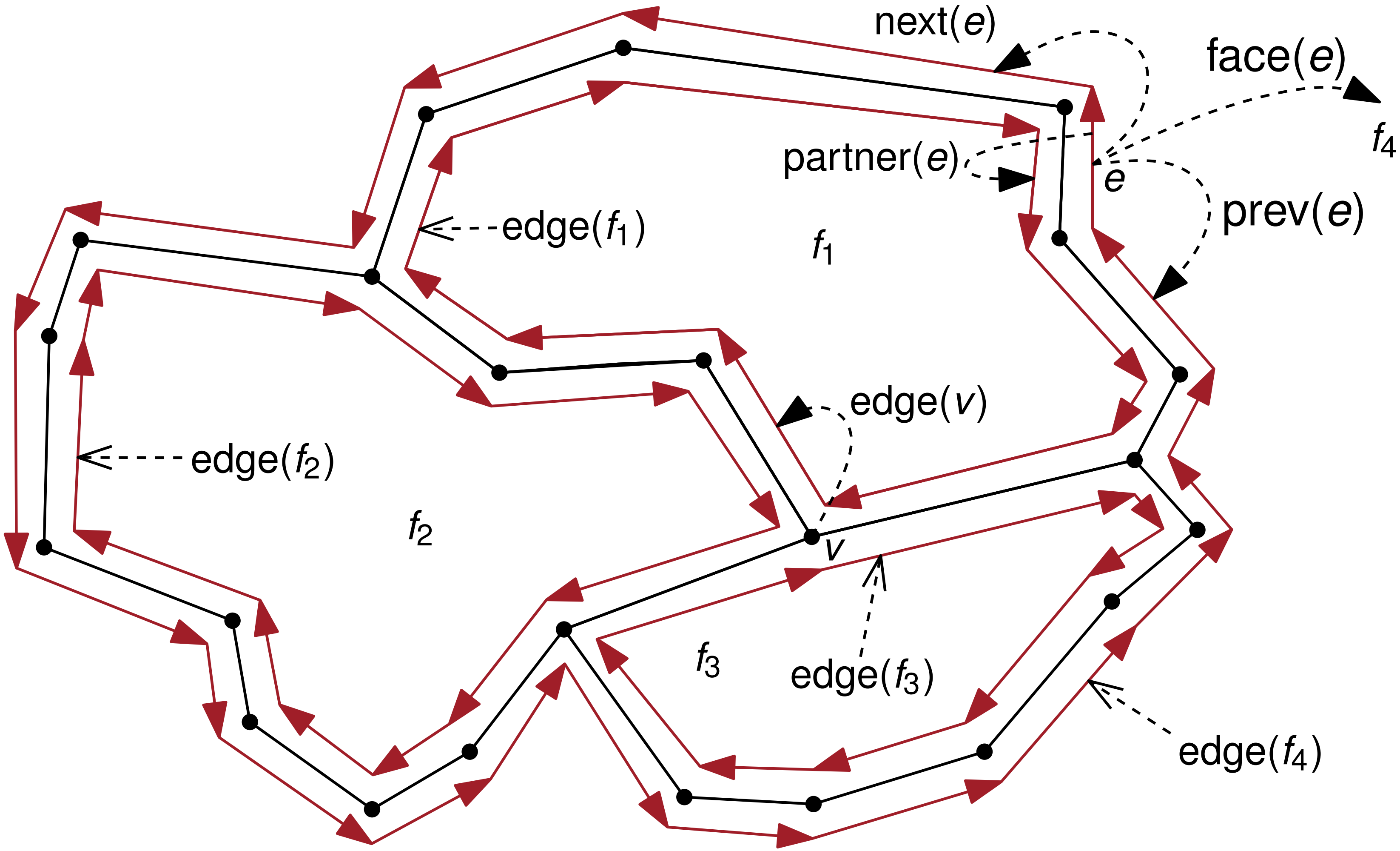
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Unterteilung



- Zugriff auf Knoten, Facetten und Kanten
- *Ablaufen* einzelner Facetten.
- Ausgehende Kanten eines Knoten traversieren.

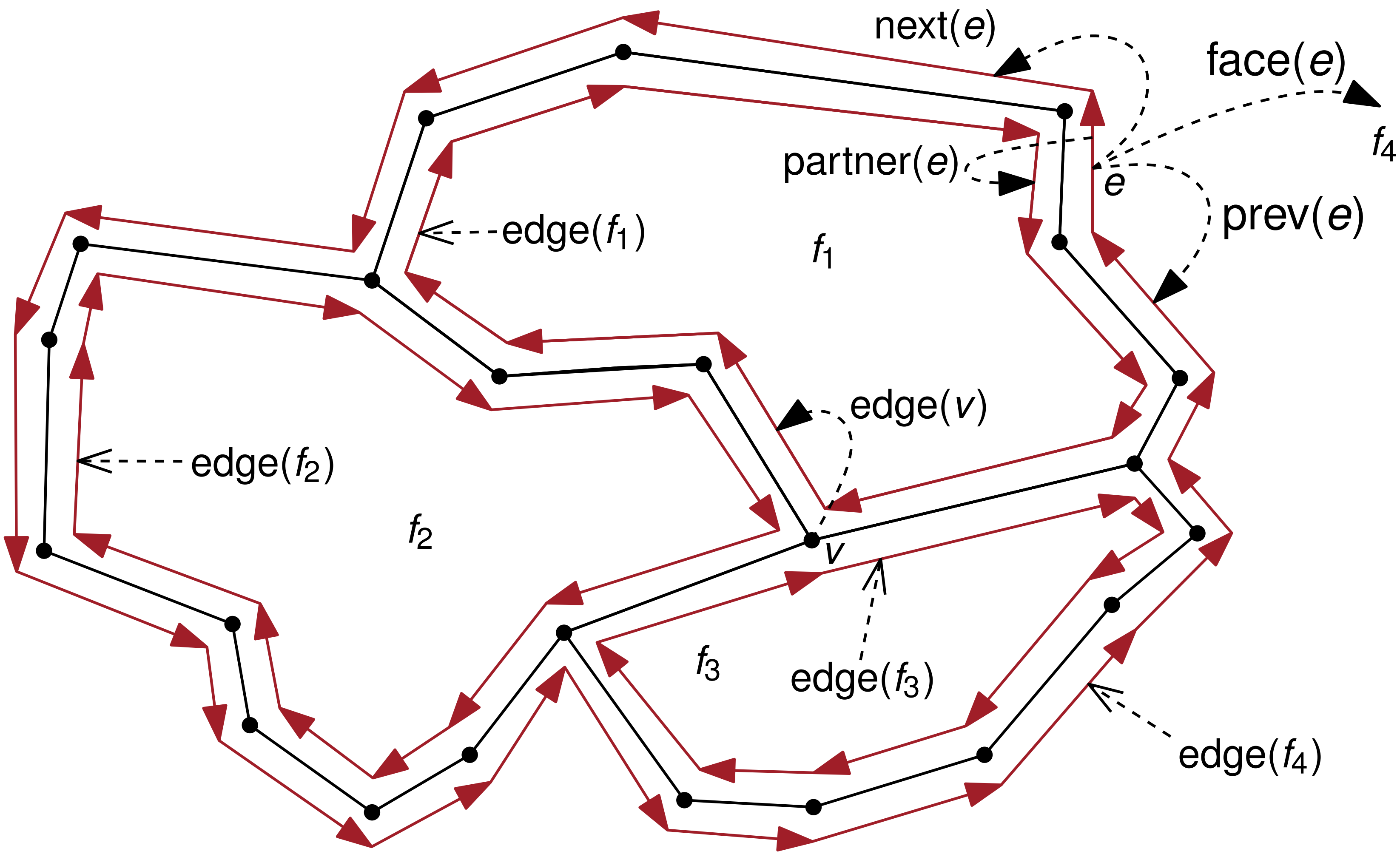
Unterteilung



- Zugriff auf Knoten, Facetten und Kanten ✓
- *Ablaufen* einzelner Facetten.
- Ausgehende Kanten eines Knoten traversieren.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

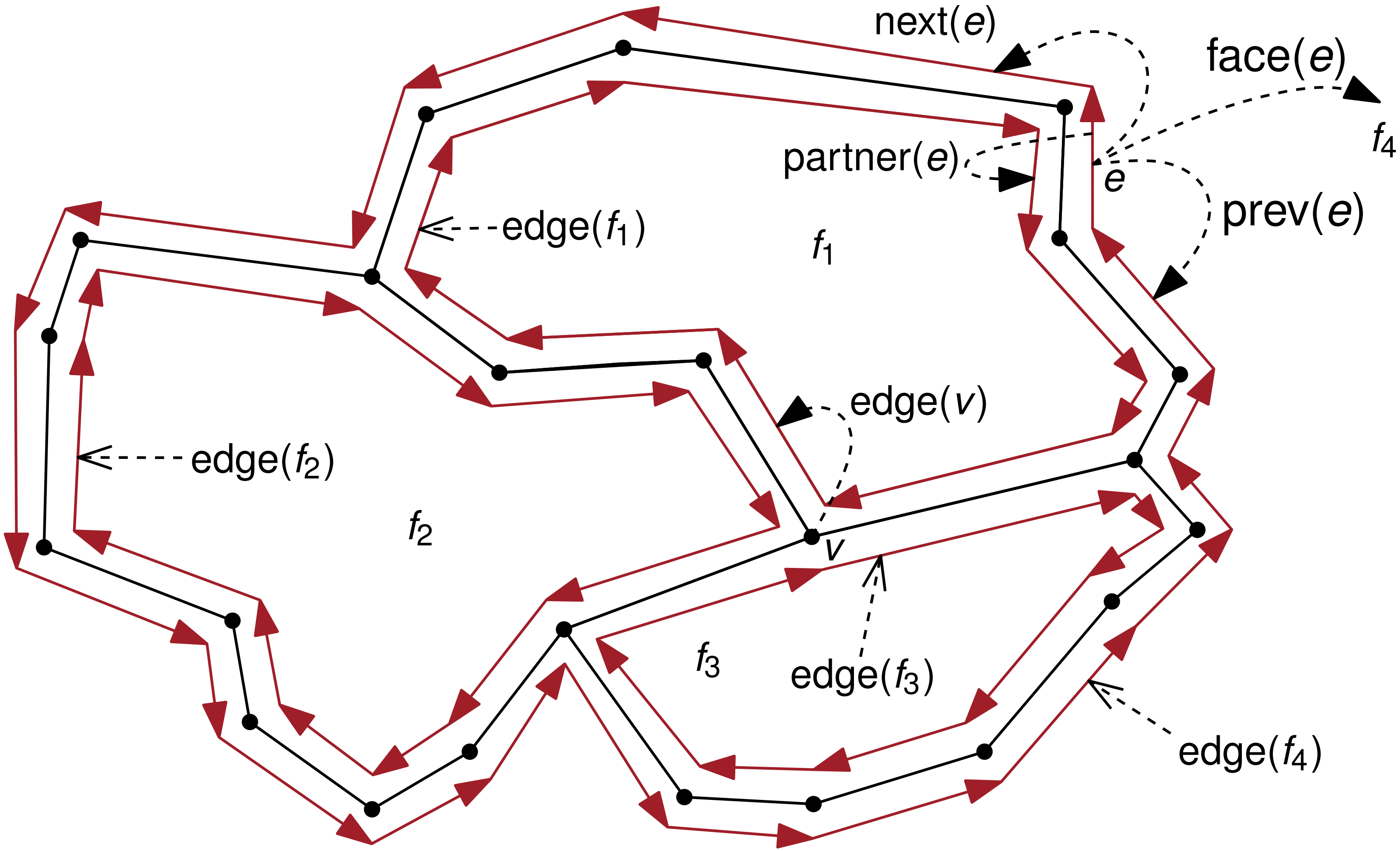
Unterteilung



- Zugriff auf Knoten, Facetten und Kanten ✓
- *Ablaufen* einzelner Facetten. ✓
- Ausgehende Kanten eines Knoten traversieren.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

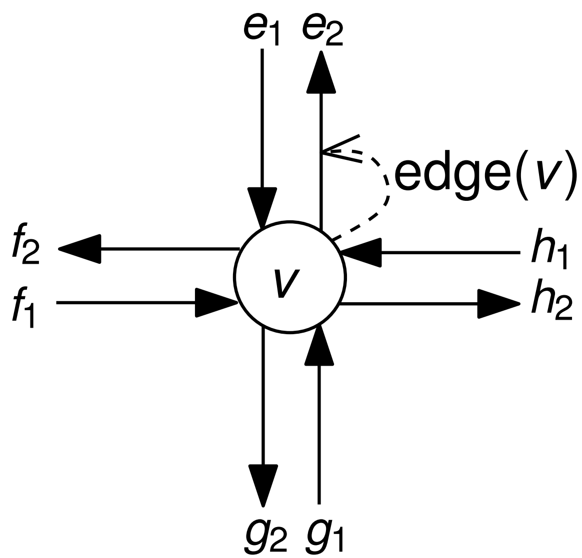
Unterteilung



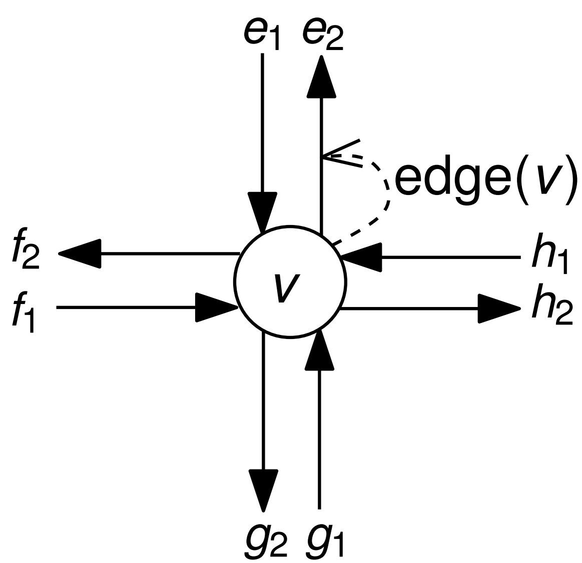
- Zugriff auf Knoten, Facetten und Kanten ✓
- *Ablaufen* einzelner Facetten. ✓
- Ausgehende Kanten eines Knoten traversieren. ?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Traversierung von inzidenten Kanten



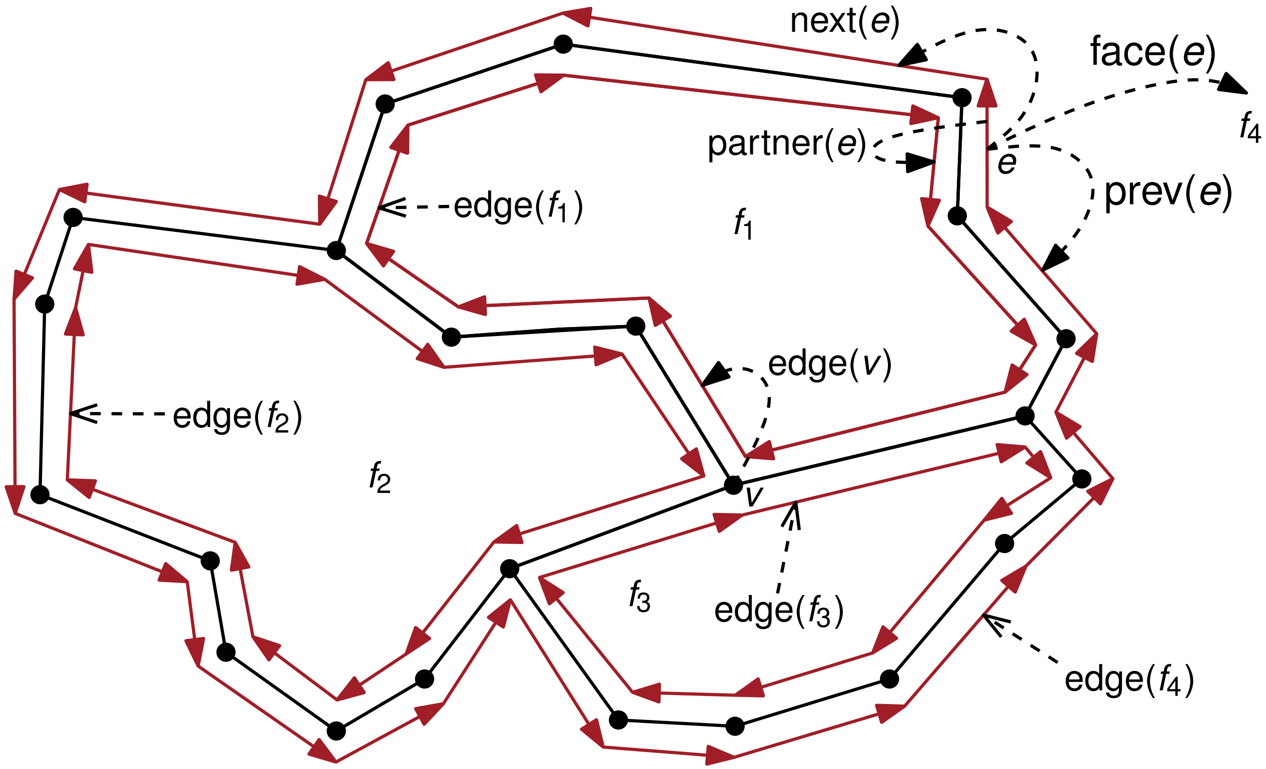
Traversierung von inzidenten Kanten



Traversierung im Uhrzeigersinn:

$$f_2 = \text{next}(\text{partner}(e_2))$$
$$g_2 = \text{next}(\text{partner}(f_2))$$
$$h_2 = \text{next}(\text{partner}(g_2))$$
$$e_2 = \text{next}(\text{partner}(h_2))$$

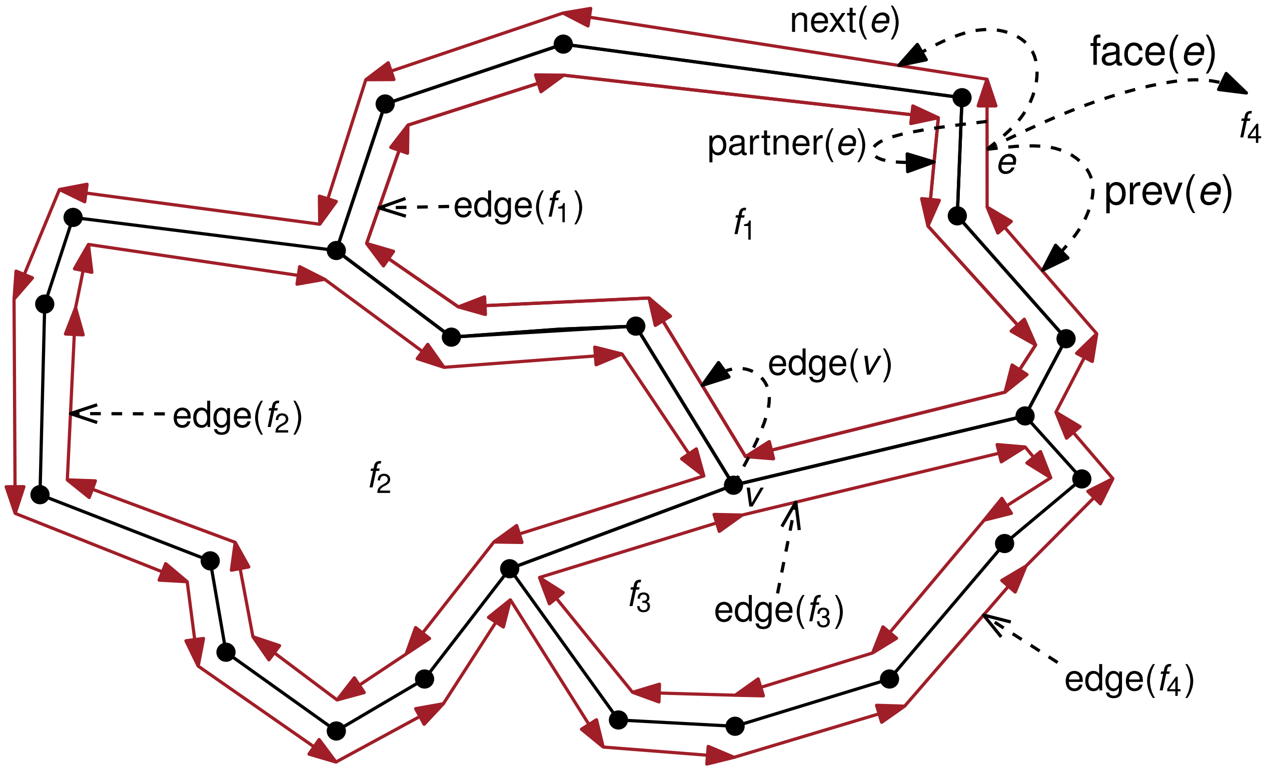
Welche Äquivalenzen sind immer wahr?



- partner(partner(e)) = e
- next(prev(e)) = e
- partner(prev(partner(e))) = next(e)
- face(e) = face(next(e))

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

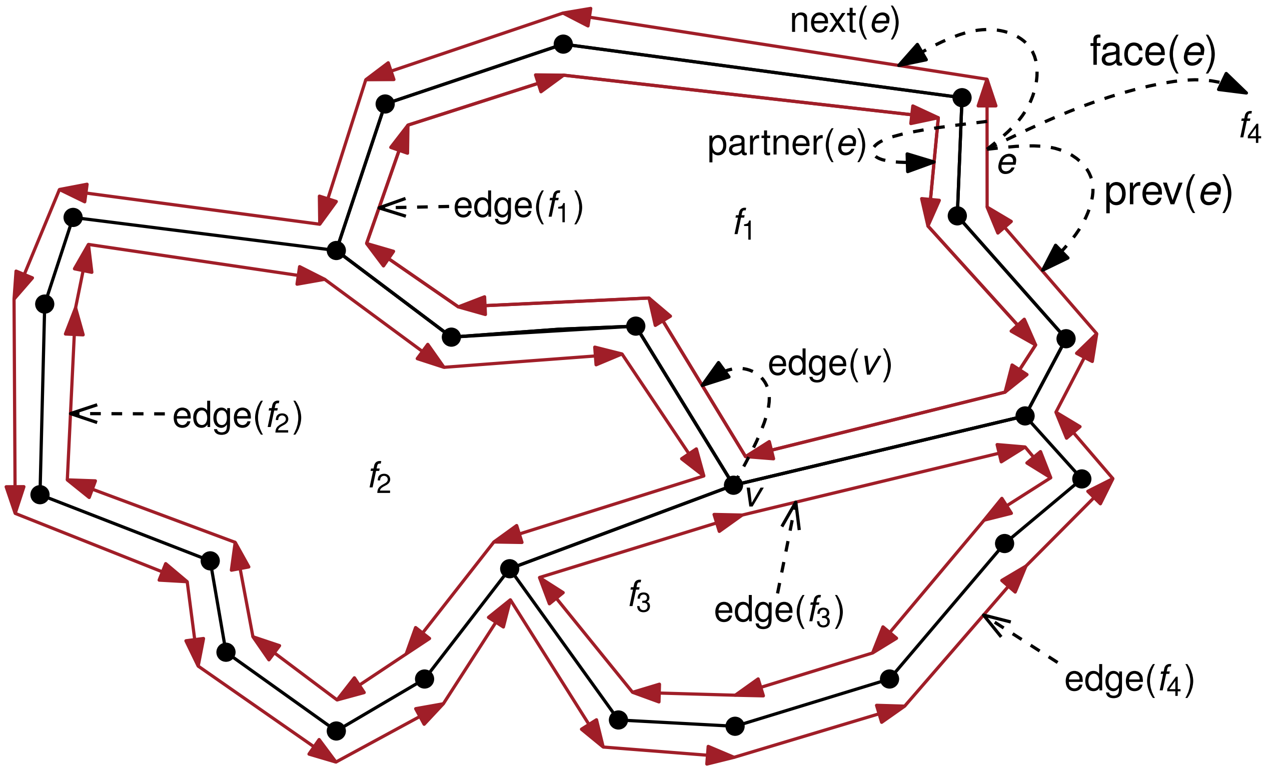
Welche Äquivalenzen sind immer wahr?



- partner(partner(e)) = e ✓
- next(prev(e)) = e
- partner(prev(partner(e))) = next(e)
- face(e) = face(next(e))

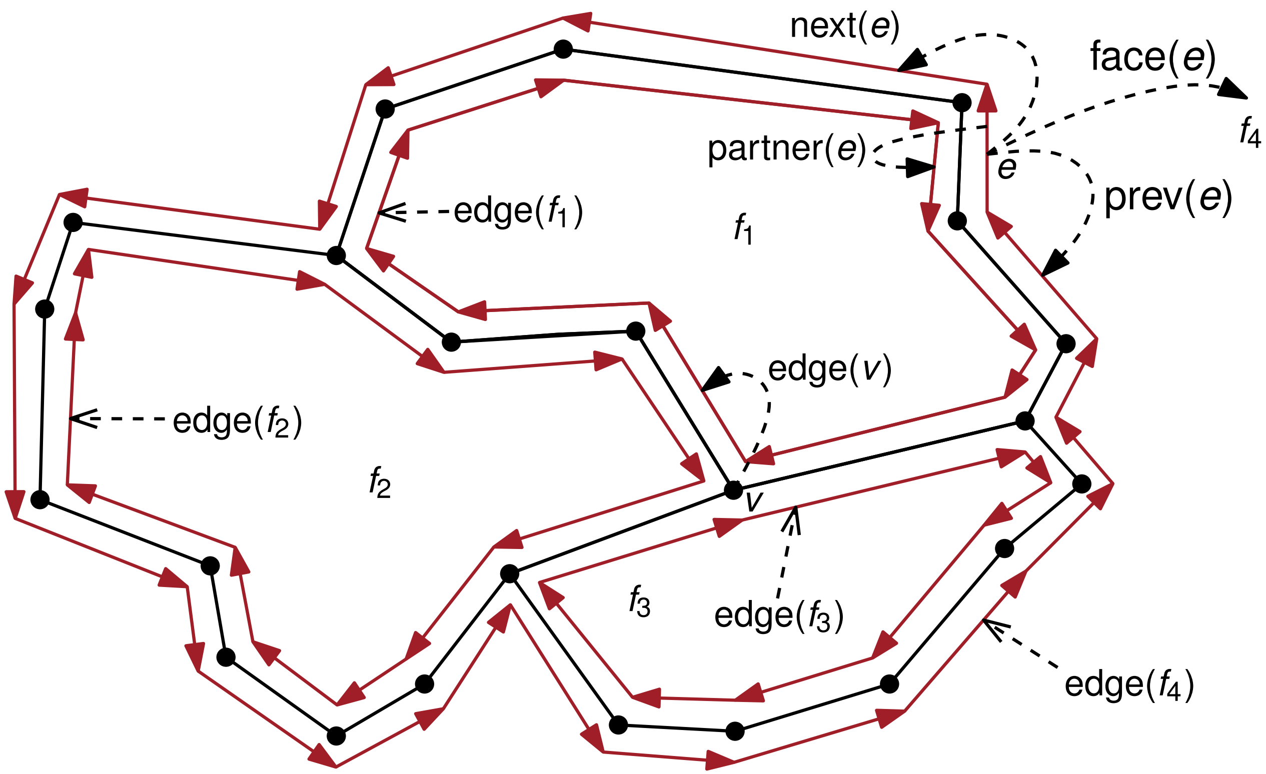
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Welche Äquivalenzen sind immer wahr?



- partner(partner(e)) = e ✓
- next(prev(e)) = e ✓
- partner(prev(partner(e))) = next(e)
- face(e) = face(next(e))

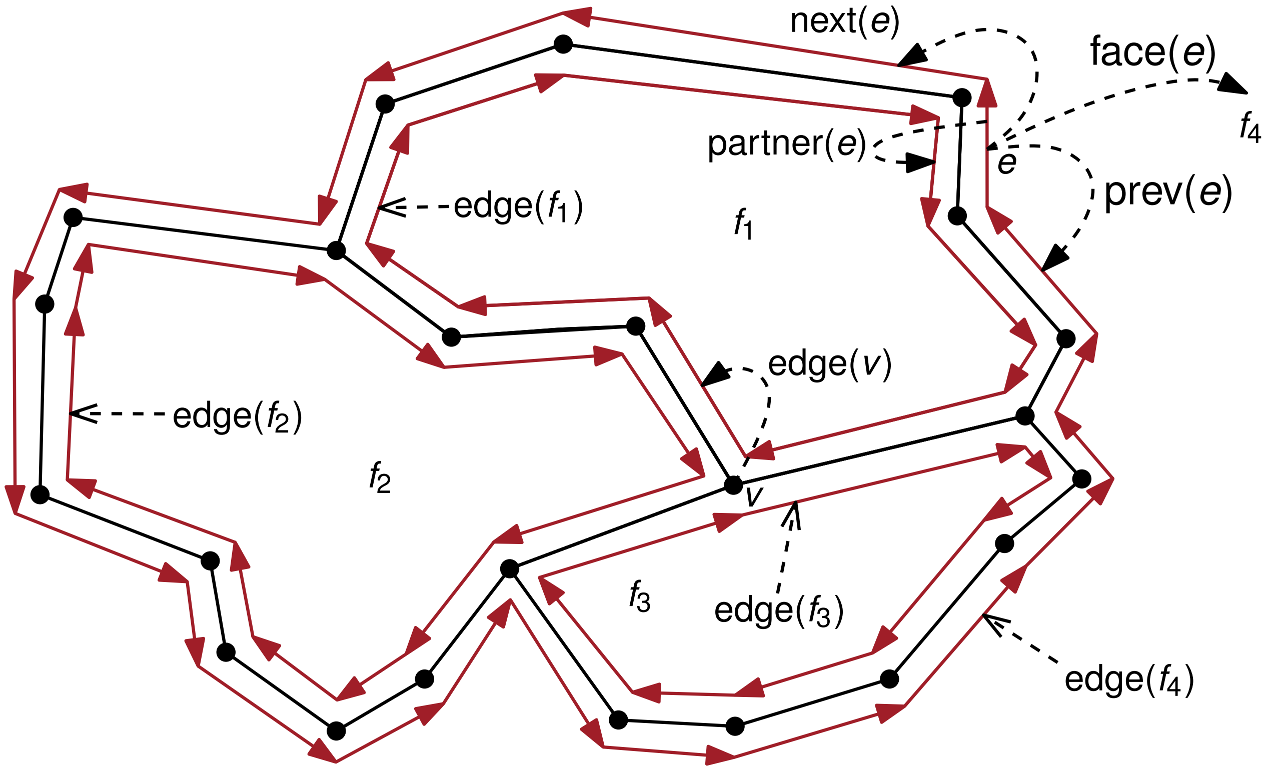
Welche Äquivalenzen sind immer wahr?



- partner(partner(e)) = e
- next(prev(e)) = e
- partner(prev(partner(e))) = next(e)
- face(e) = face(next(e))

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Welche Äquivalenzen sind immer wahr?



- partner(partner(e)) = e ✓
- next(prev(e)) = e ✓
- partner(prev(partner(e))) = next(e) ✗
- face(e) = face(next(e)) ✓

Komplexität

n: Anzahl Knoten
e: Anzahl Kanten
f: Anzahl Facetten

Welchen Platzbedarf besitzt eine zusammenhängende Unterteilung?

Komplexität

n: Anzahl Knoten
e: Anzahl Kanten
f: Anzahl Facetten

Welchen Platzbedarf besitzt eine zusammenhängende Unterteilung?

- Jeder Knoten besitzt einen Zeiger.
- Jede Kante wird in zwei Halbkanten aufgeteilt, die jeweils vier Zeiger besitzen.
- Jede Facette besitzt einen Zeiger.

Damit ergibt sich $O(n + e + f)$ Speicher.

Komplexität

n: Anzahl Knoten
e: Anzahl Kanten
f: Anzahl Facetten

Welchen Platzbedarf besitzt eine zusammenhängende Unterteilung?

- Jeder Knoten besitzt einen Zeiger.
- Jede Kante wird in zwei Halbkanten aufgeteilt, die jeweils vier Zeiger besitzen.
- Jede Facette besitzt einen Zeiger.

Damit ergibt sich $O(n + e + f)$ Speicher.

Satz von Euler besagt: $n - m + f = 2$

Komplexität

n: Anzahl Knoten
e: Anzahl Kanten
f: Anzahl Facetten

Welchen Platzbedarf besitzt eine zusammenhängende Unterteilung?

- Jeder Knoten besitzt einen Zeiger.
- Jede Kante wird in zwei Halbkanten aufgeteilt, die jeweils vier Zeiger besitzen.
- Jede Facette besitzt einen Zeiger.

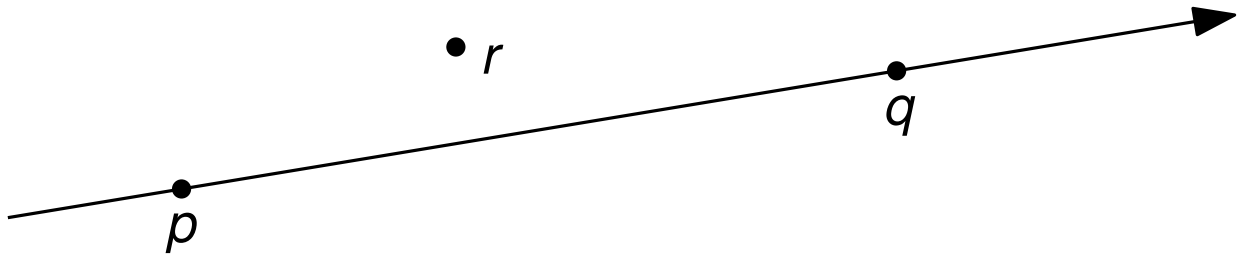
Damit ergibt sich $O(n + e + f)$ Speicher.

Satz von Euler besagt: $n - m + f = 2$

Daraus ergibt sich: Jeder planare Graph besitzt maximal $3n - 6$ Kanten und $2n - 4$ Facetten.

$O(n)$ Speicher für Unterteilung der Ebene

Position eines Punktes



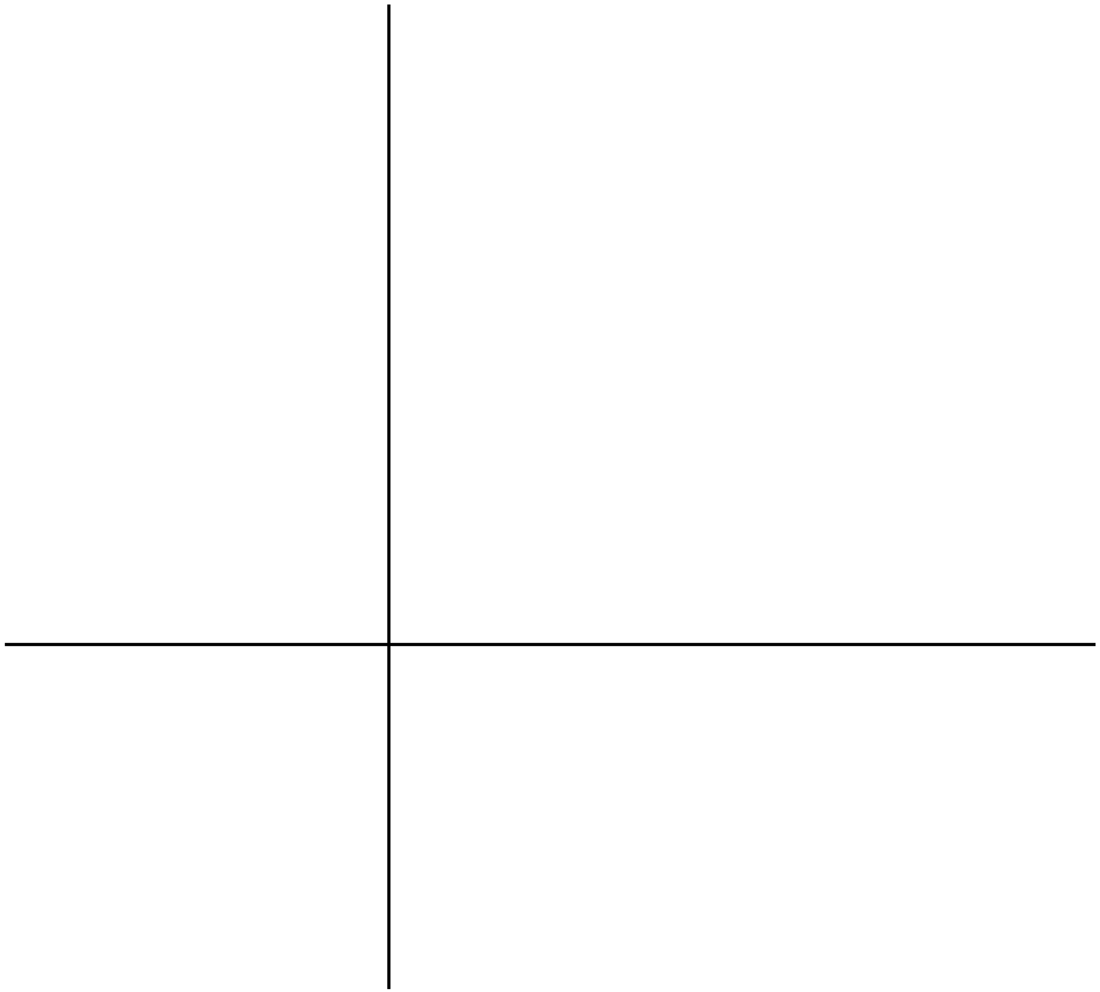
Gerichtete Gerade g durch $p = (p_x, p_y)$ und $q = (q_x, q_y)$

Punkt $r = (r_x, r_y)$

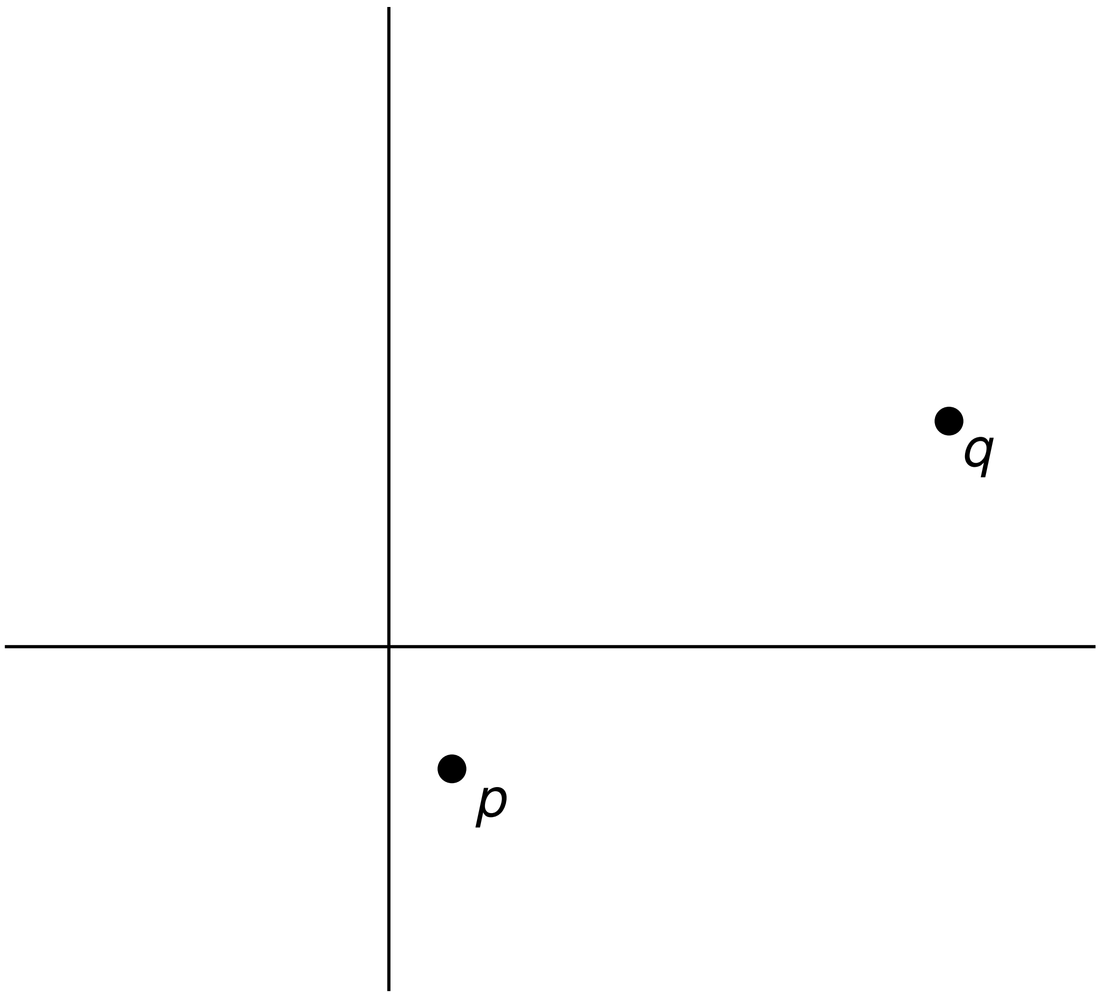
Liegt der Punkt r links oder rechts der Geraden g?

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Position eines Punktes

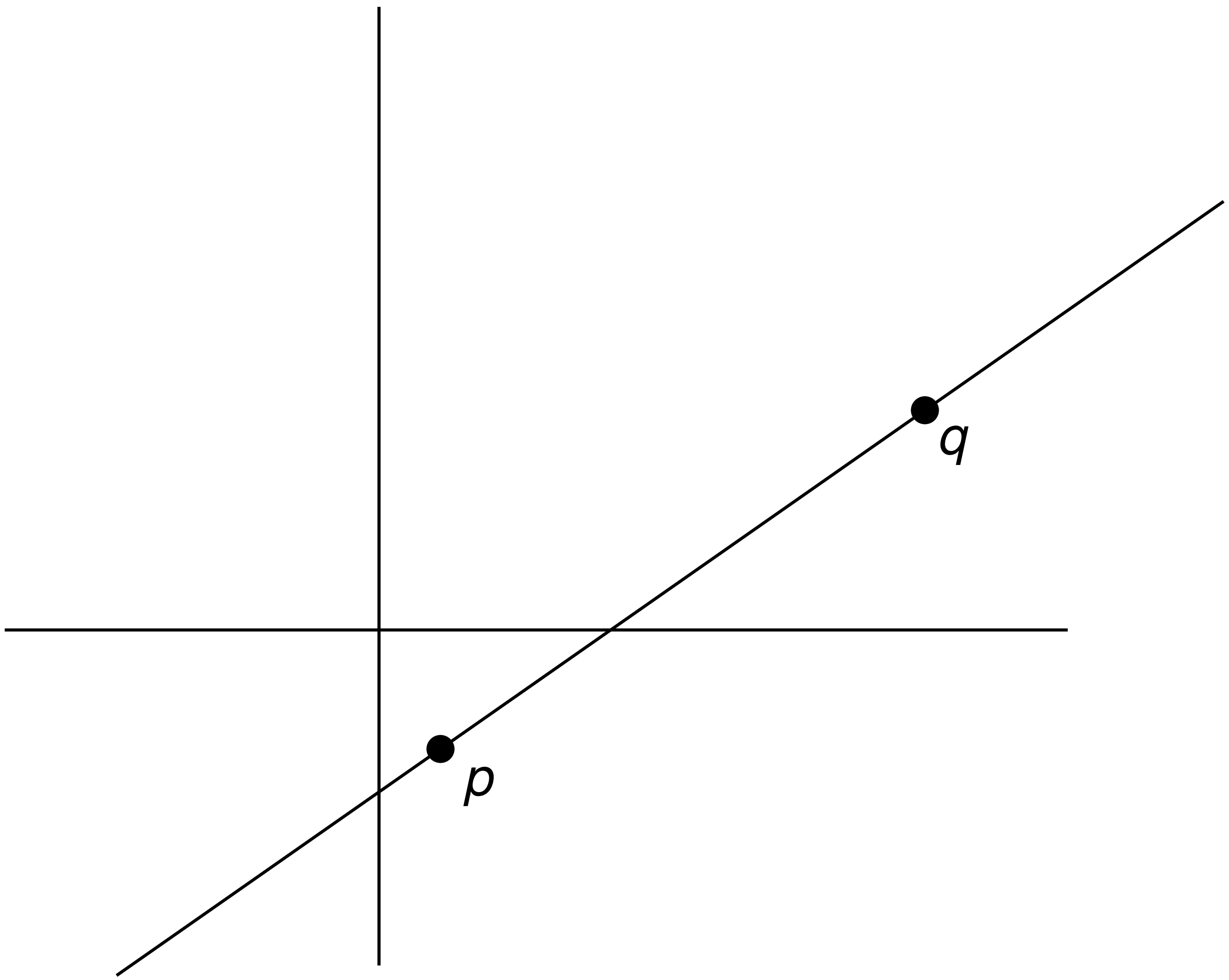


Position eines Punktes

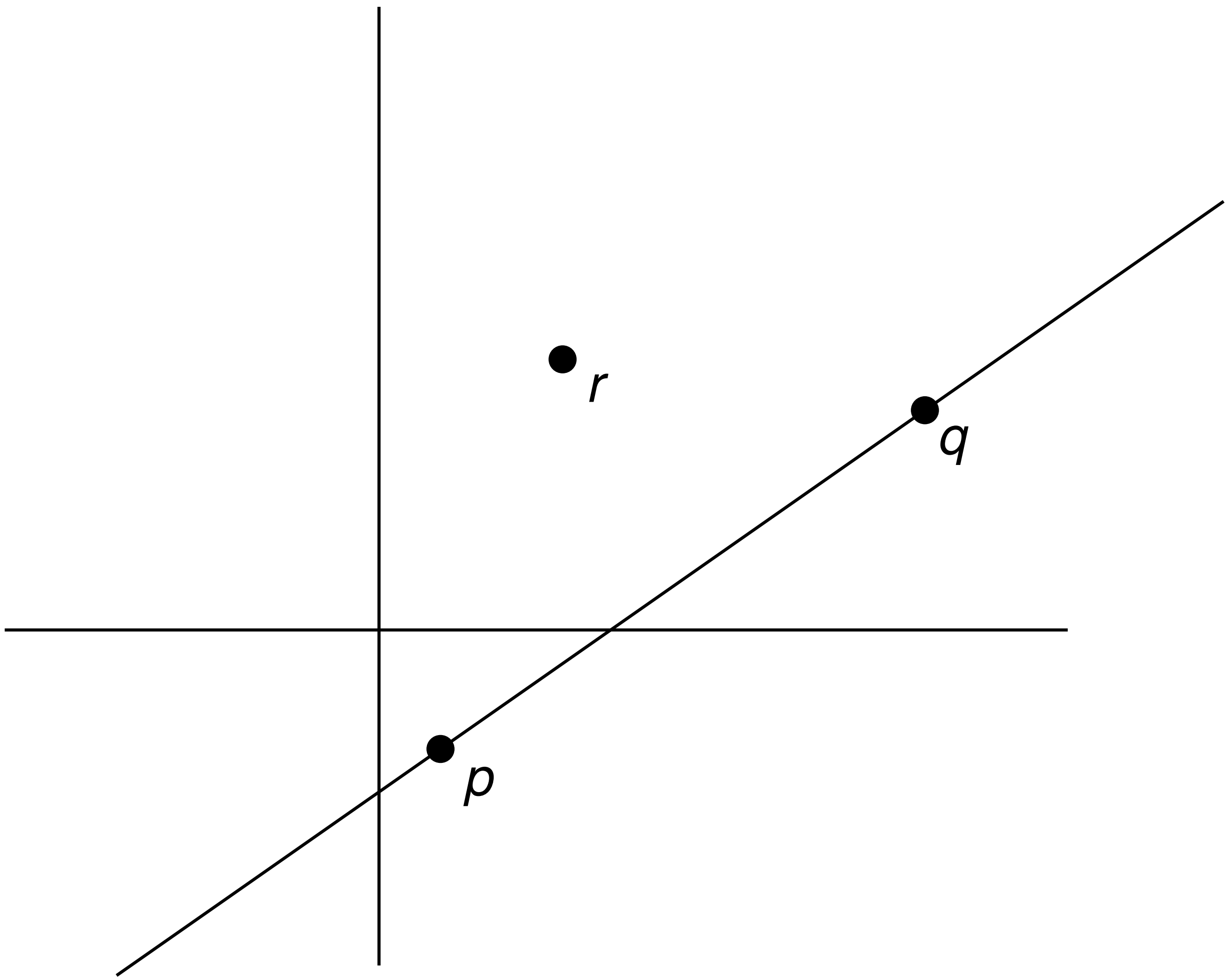


Position eines Punktes

Karlsruher Institut für Technologie

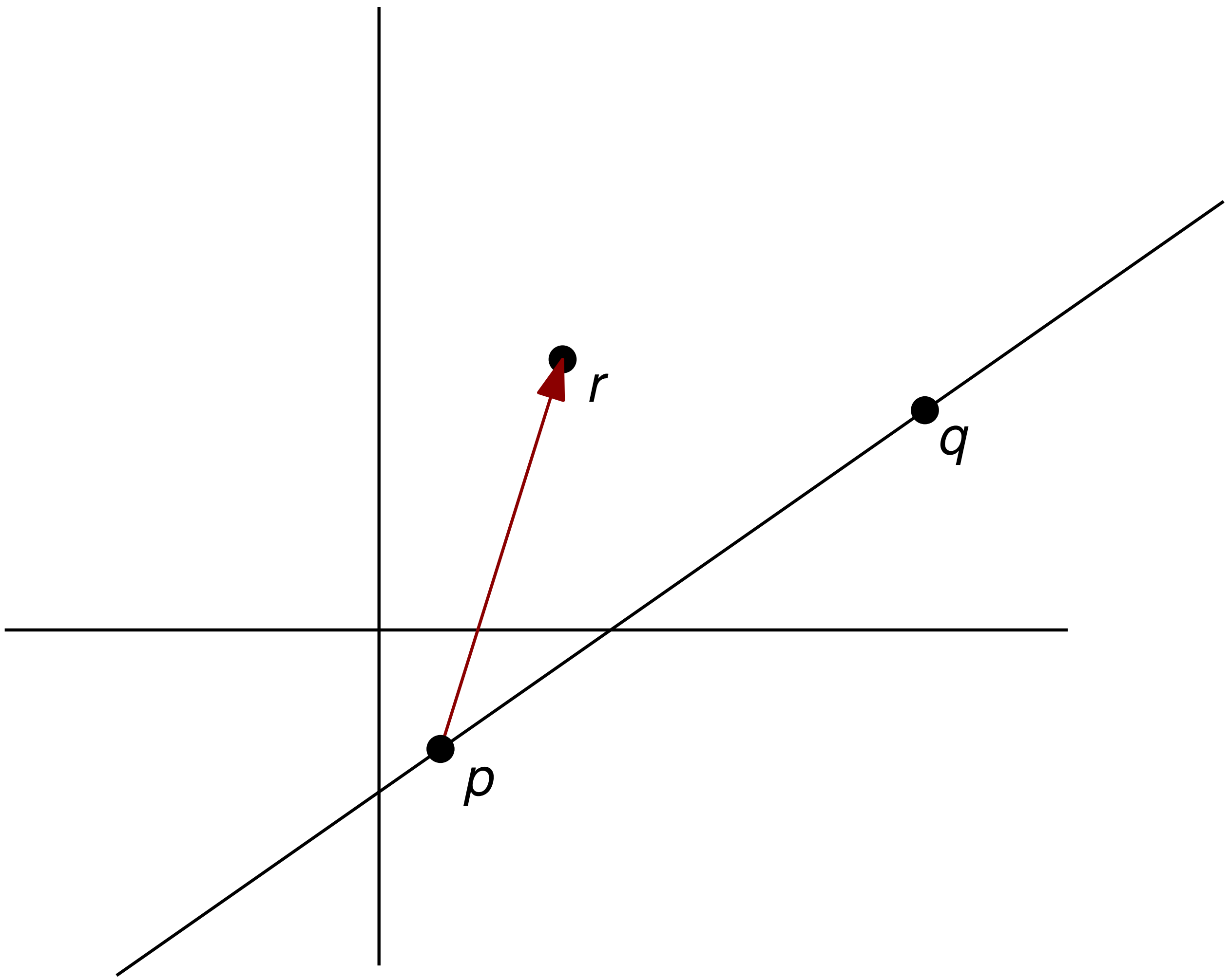


Position eines Punktes



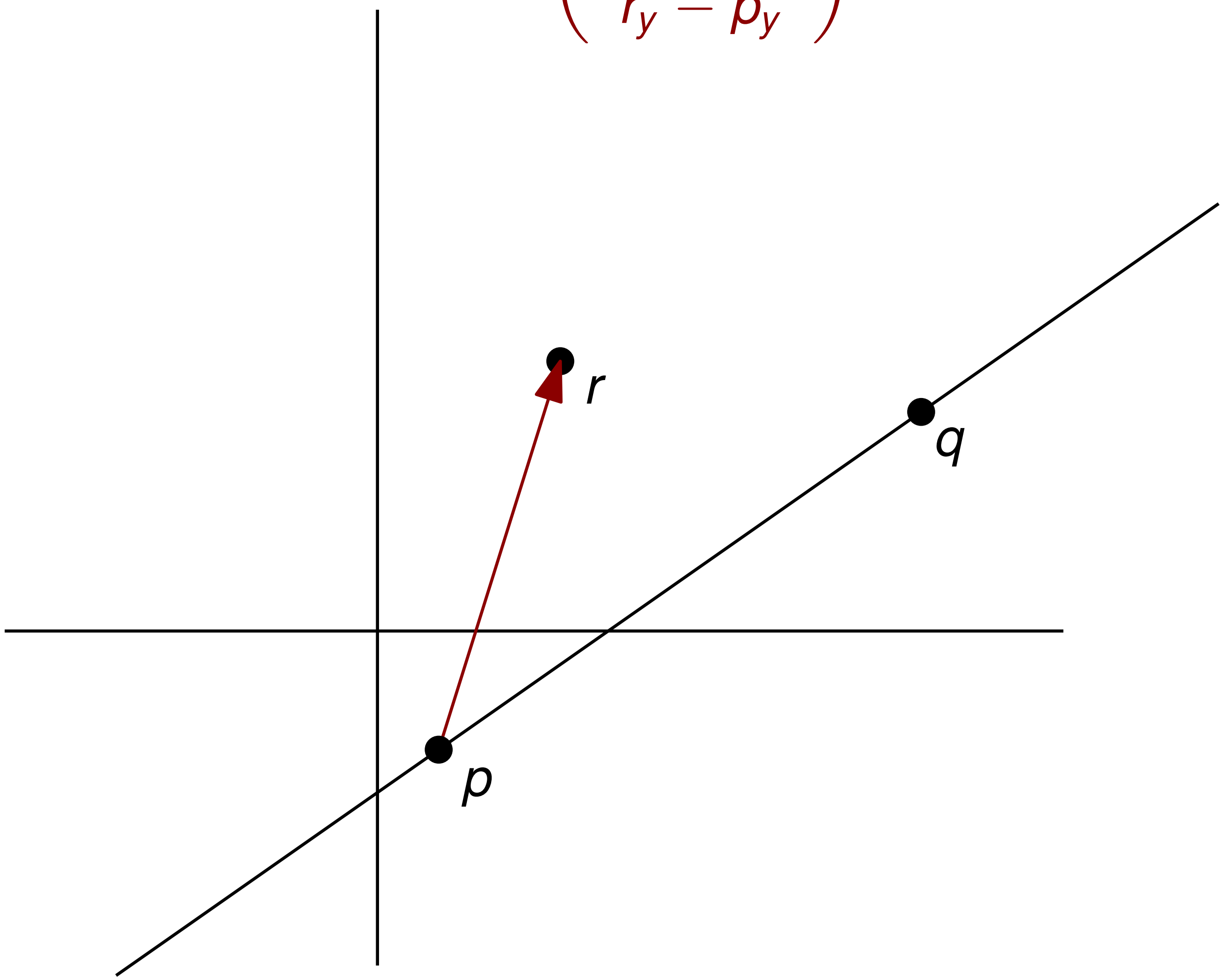
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Position eines Punktes



Position eines Punktes

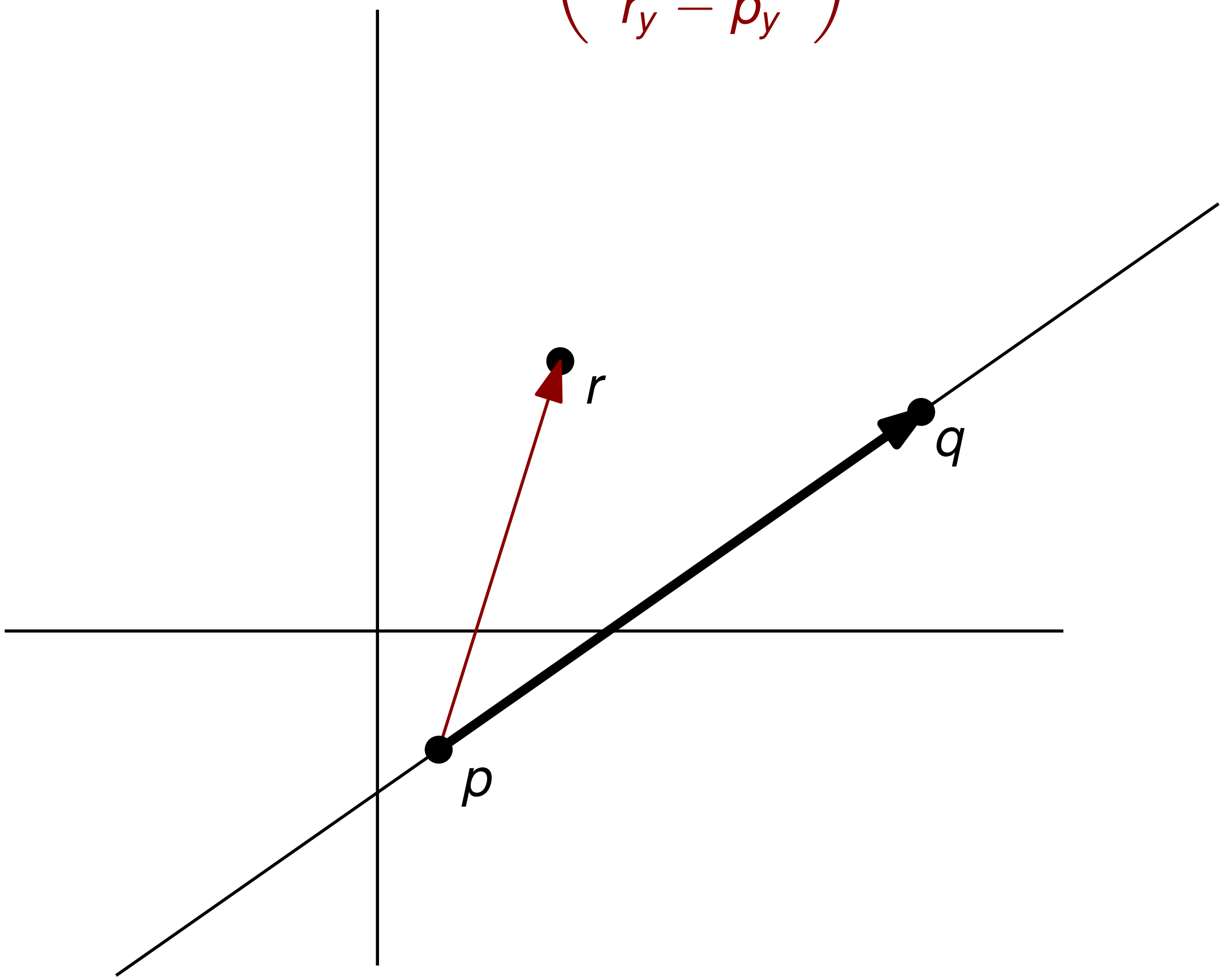
$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix}$$



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

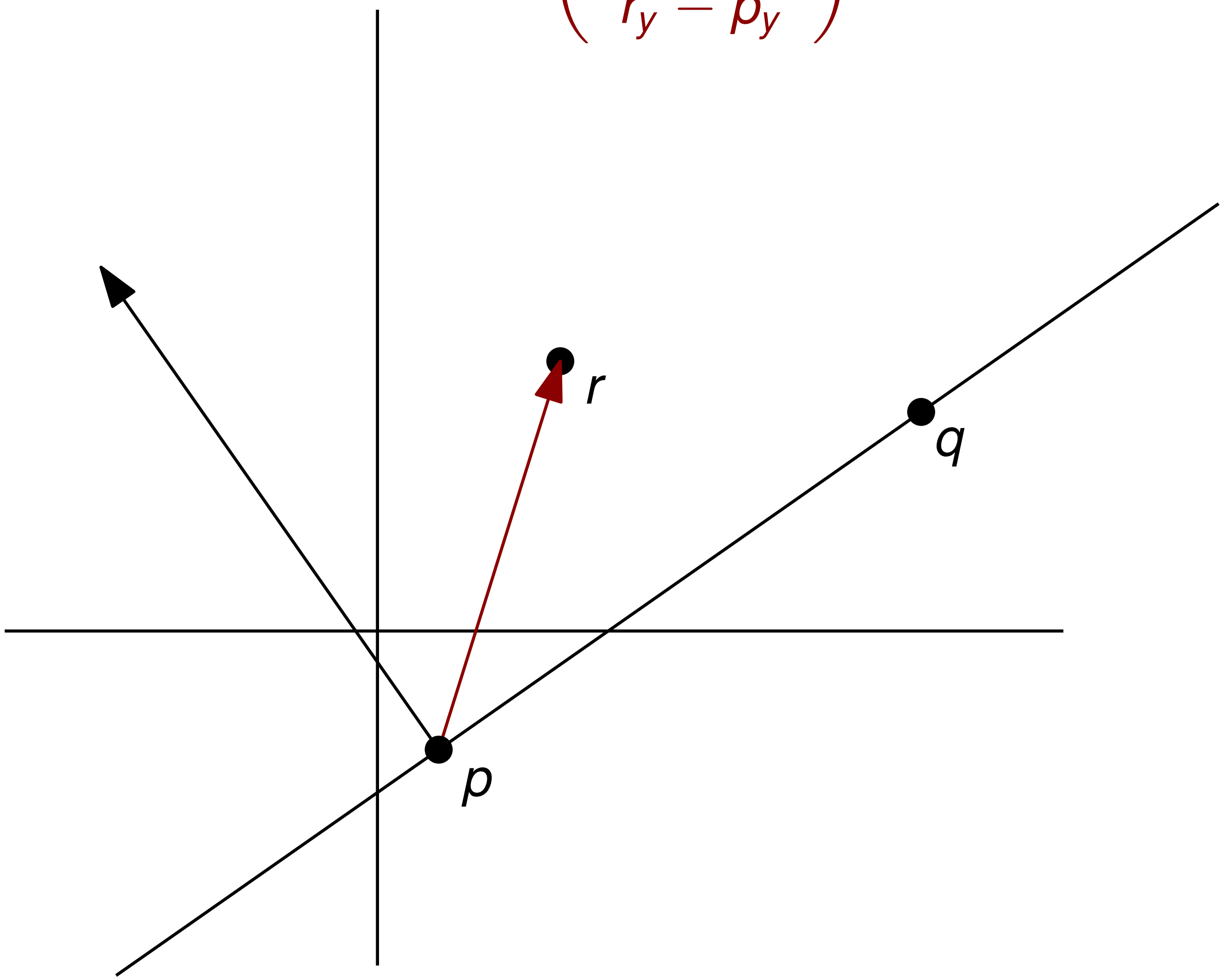
Position eines Punktes

$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix}$$



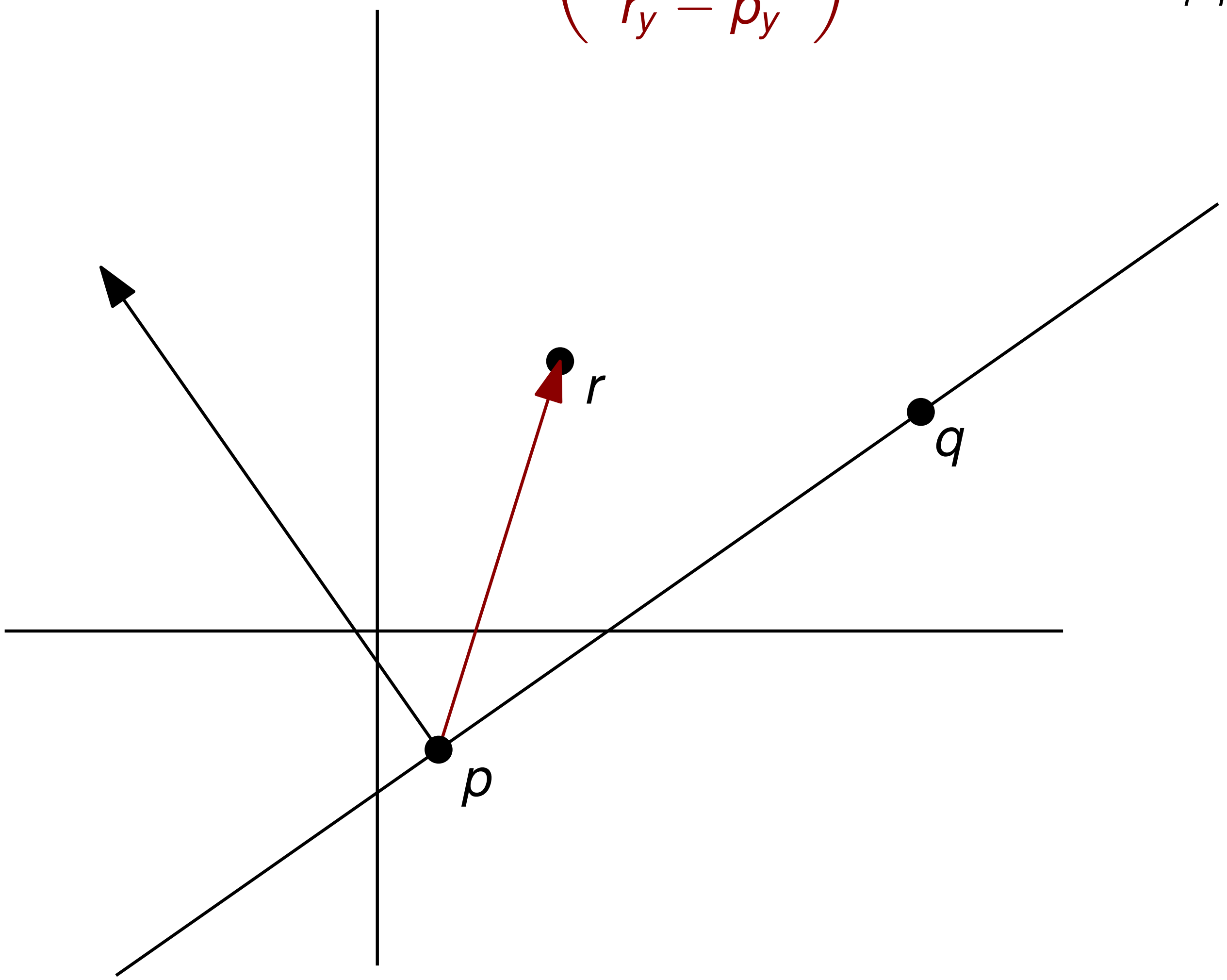
Position eines Punktes

$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix}$$



Position eines Punktes

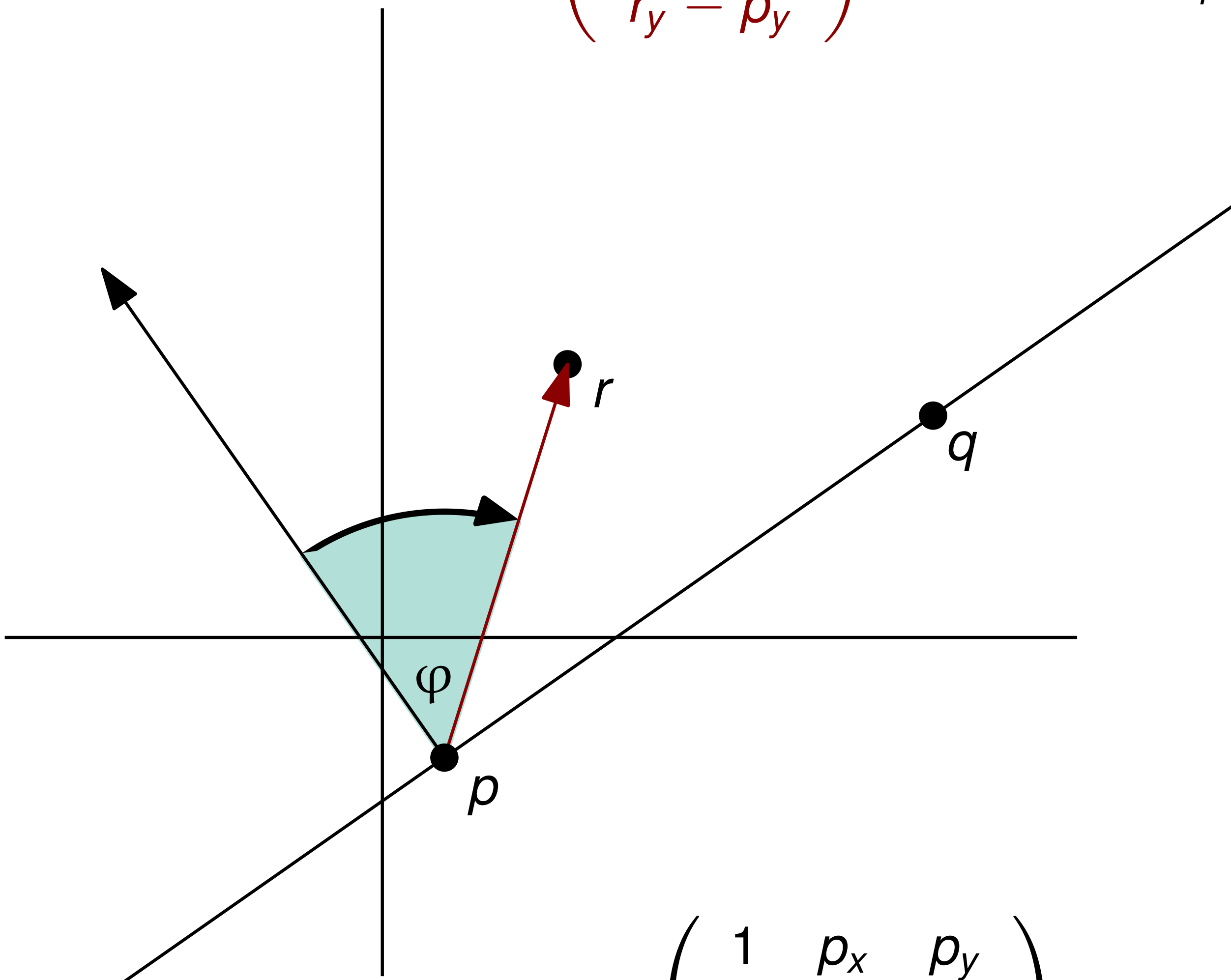
$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix} \qquad n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



Position eines Punktes

$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix}$$

$$n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



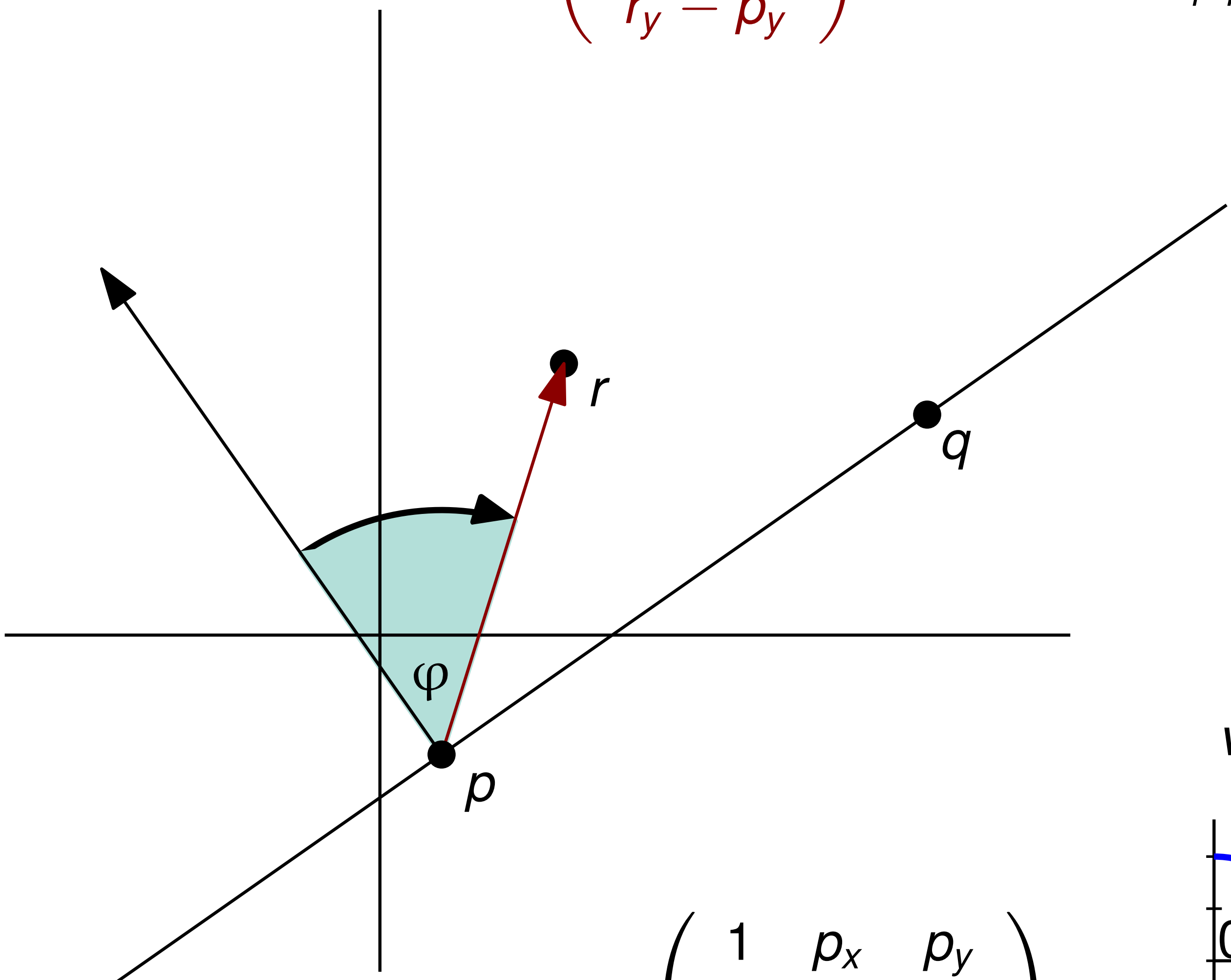
$$v^T \cdot n_{pq} = |v| \cdot |n_{pq}| \cos \varphi$$

$$A = \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}$$

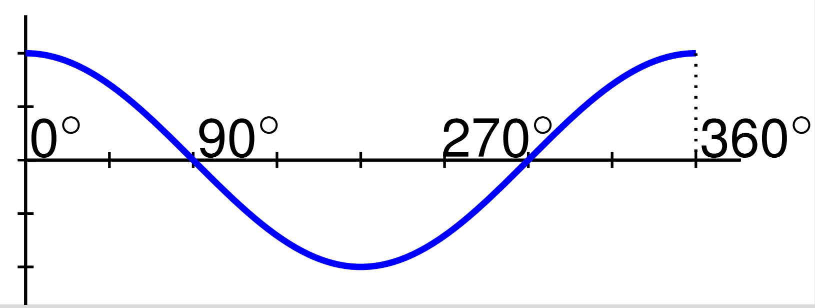
Position eines Punktes

$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix}$$

$$n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



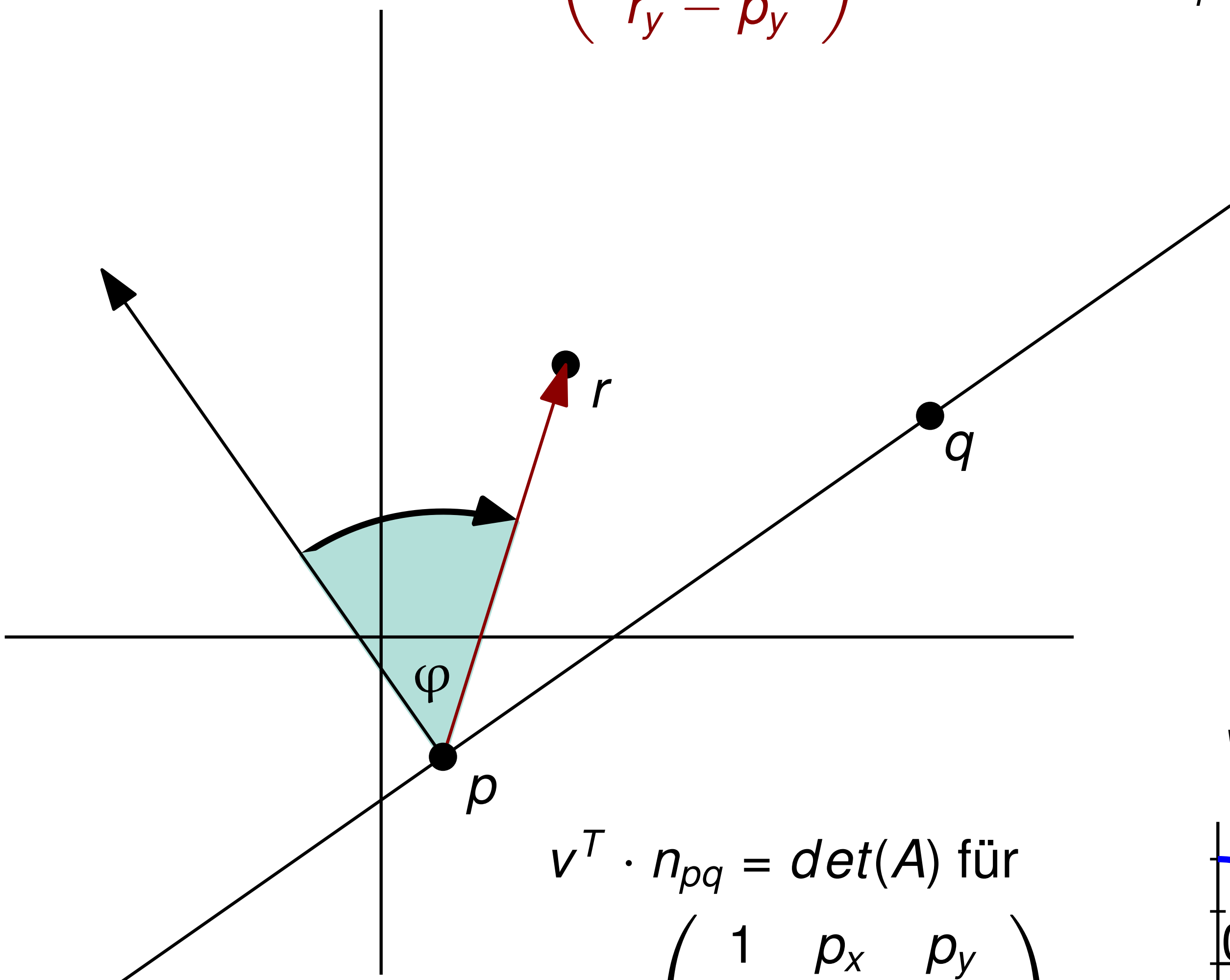
$$v^T \cdot n_{pq} = |v| \cdot |n_{pq}| \cos \varphi$$



$$A = \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}$$

Position eines Punktes

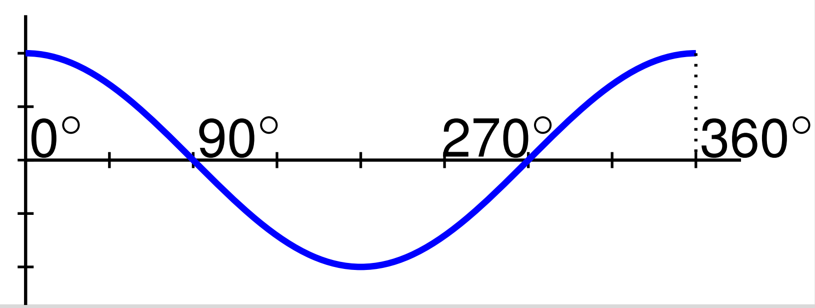
$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix} \qquad n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



$$v^T \cdot n_{pq} = det(A) \text{ für}$$

$$A = \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}$$

$$v^T \cdot n_{pq} = |v| \cdot |n_{pq}| \cos \varphi$$

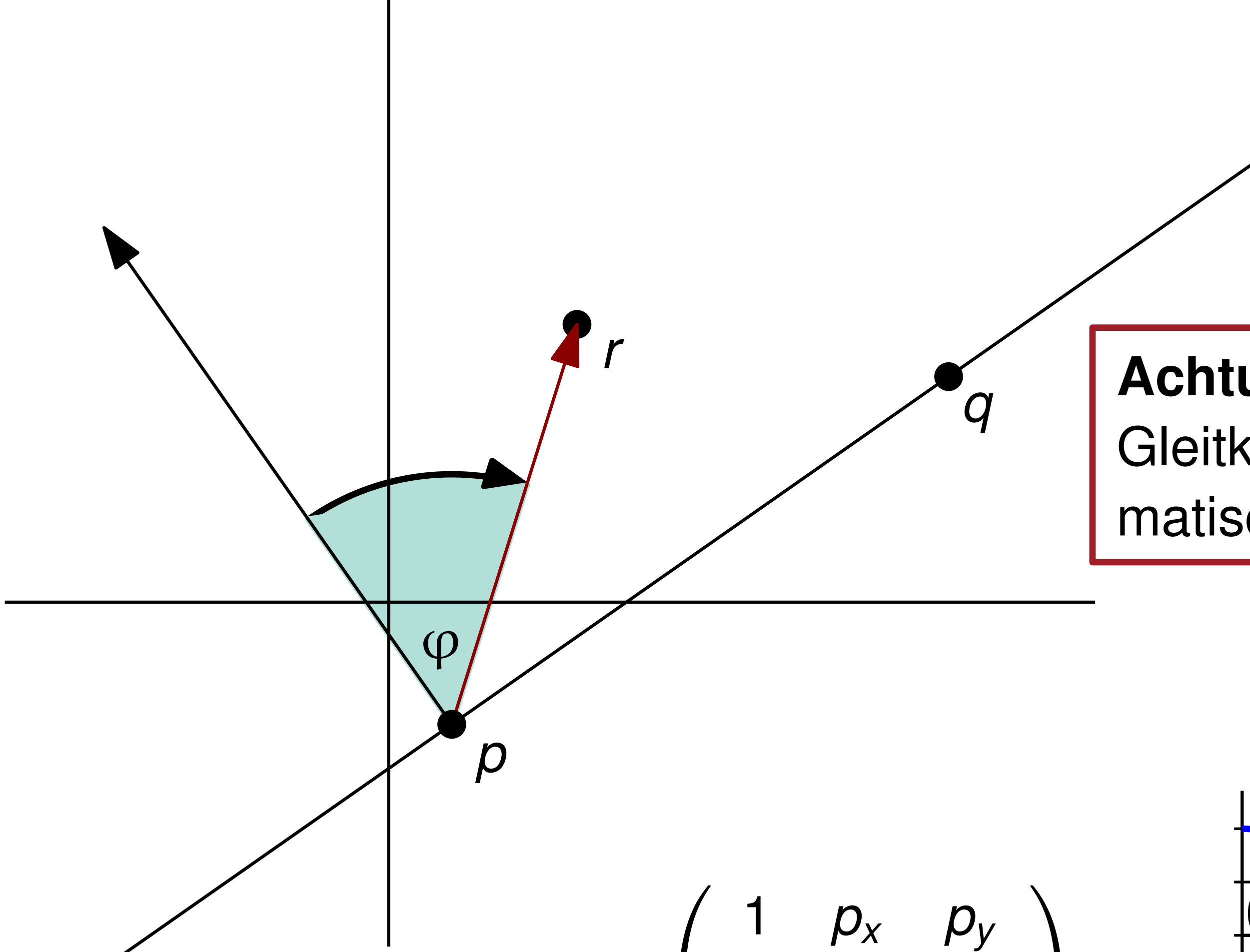


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Position eines Punktes

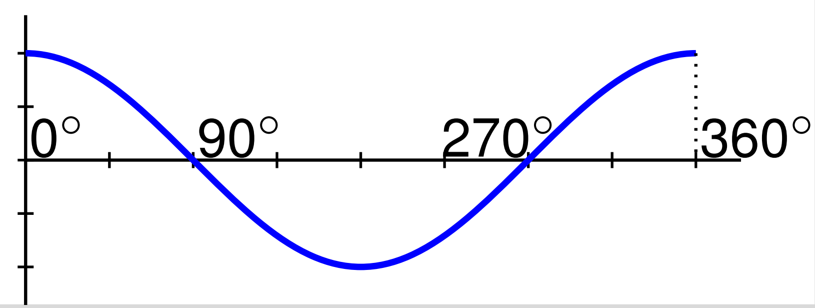
$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix}$$

$$n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



Achtung: In der Praxis wegen Gleitkomma-Zahlen nicht unproblematisch.

$$v^T \cdot n_{pq} = |v| \cdot |n_{pq}| \cos \varphi$$

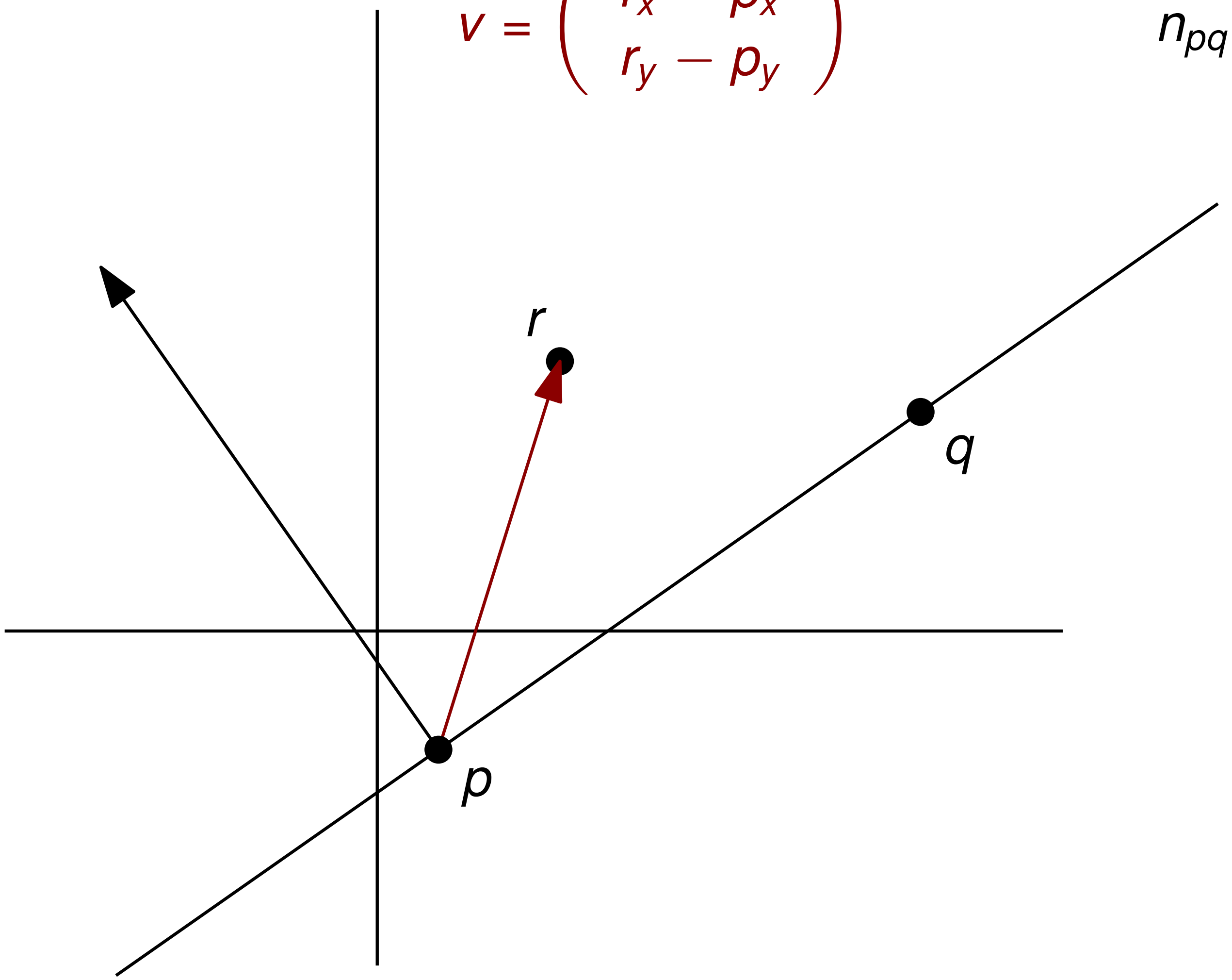


$$A = \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}$$

Determinante

Zeige:: $|det(A)|$ = doppelter Flaecheninhalt des Dreiecks pqr

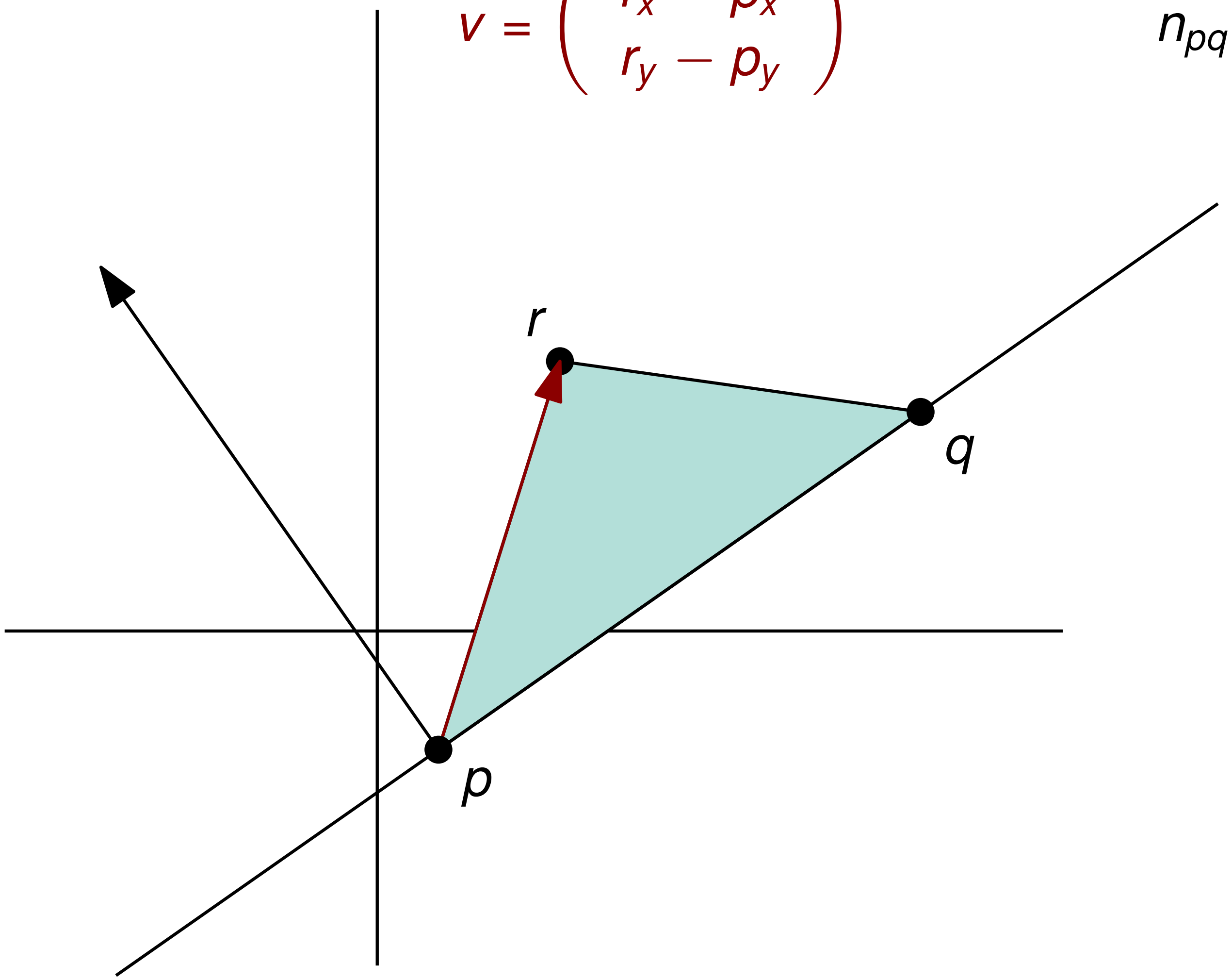
$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix} \qquad n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



Determinante

Zeige:: $|det(A)|$ = doppelter Flaecheninhalt des Dreiecks pqr

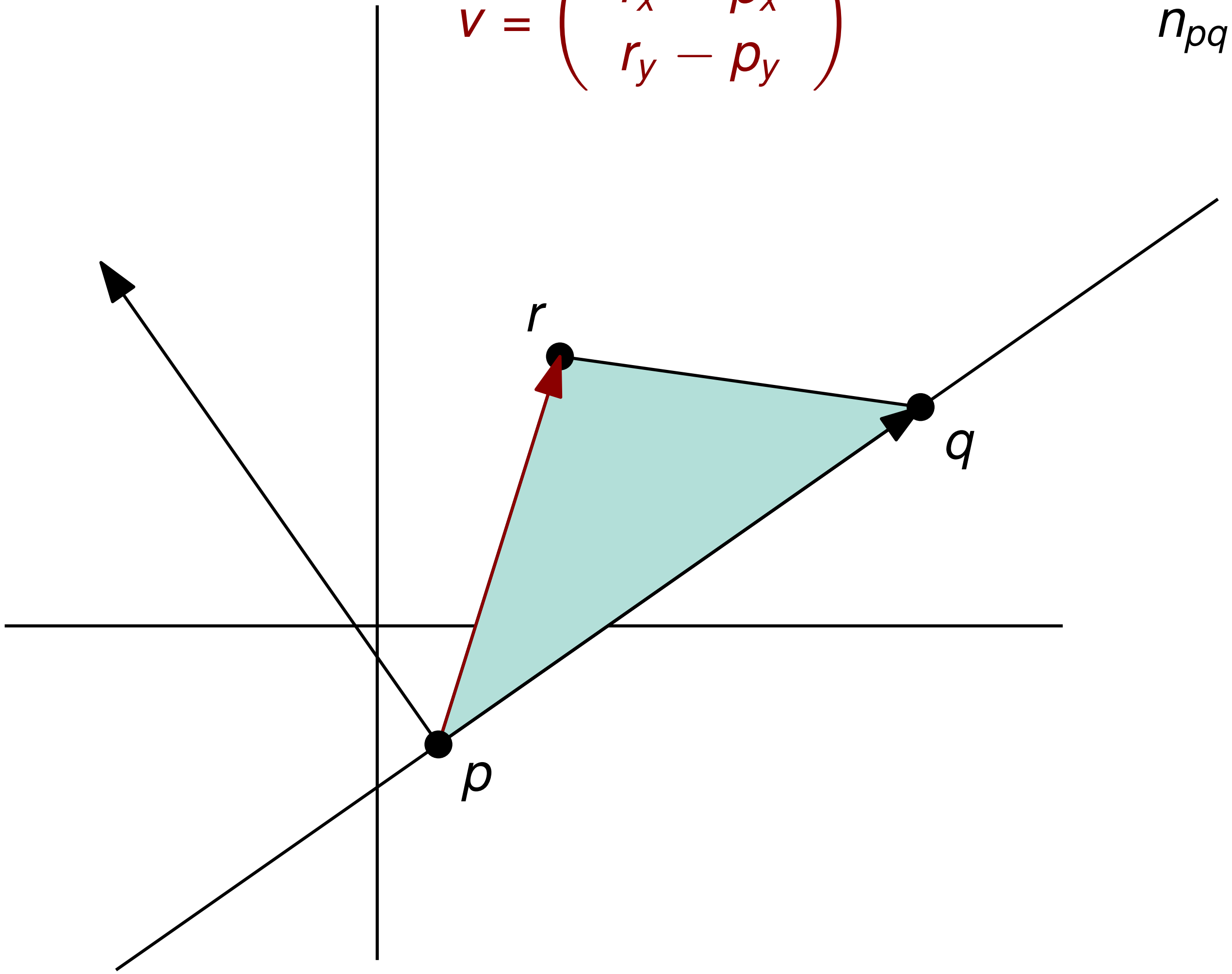
$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix} \qquad n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



Determinante

Zeige:: $|det(A)|$ = doppelter Flaecheninhalt des Dreiecks pqr

$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix} \qquad n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$

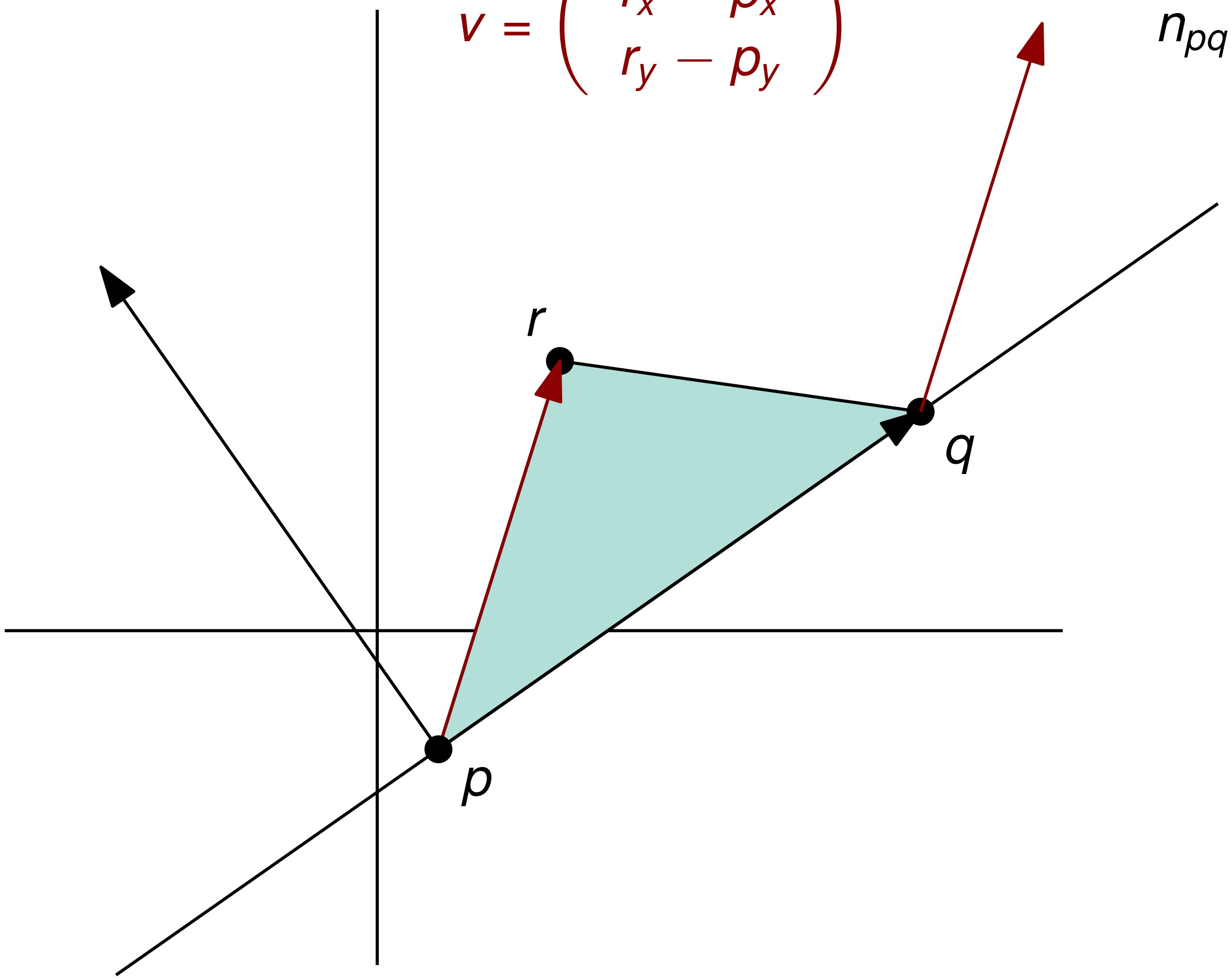


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Determinante

Zeige:: $|det(A)|$ = doppelter Flaecheninhalt des Dreiecks pqr

$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix} \qquad n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$

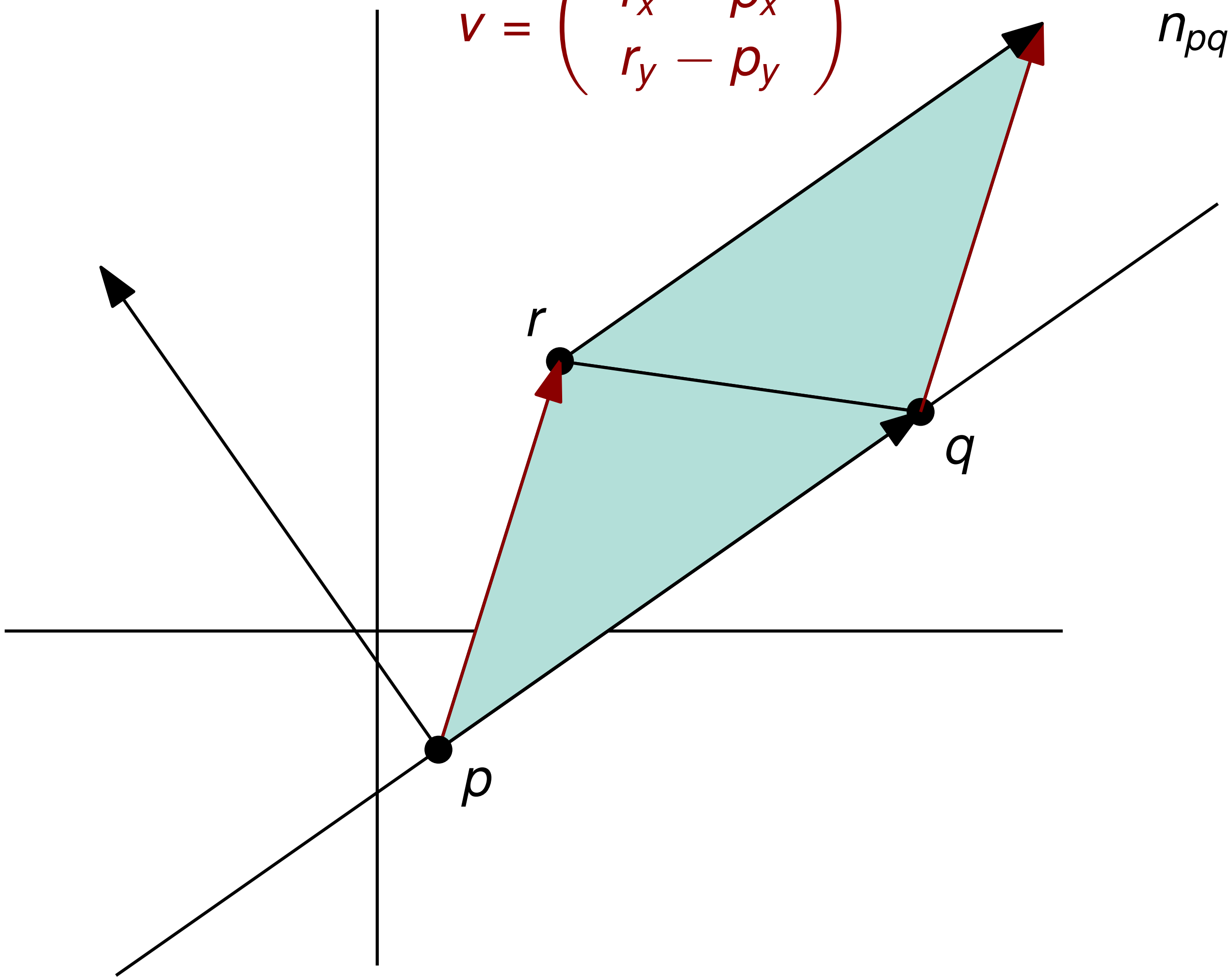


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Determinante

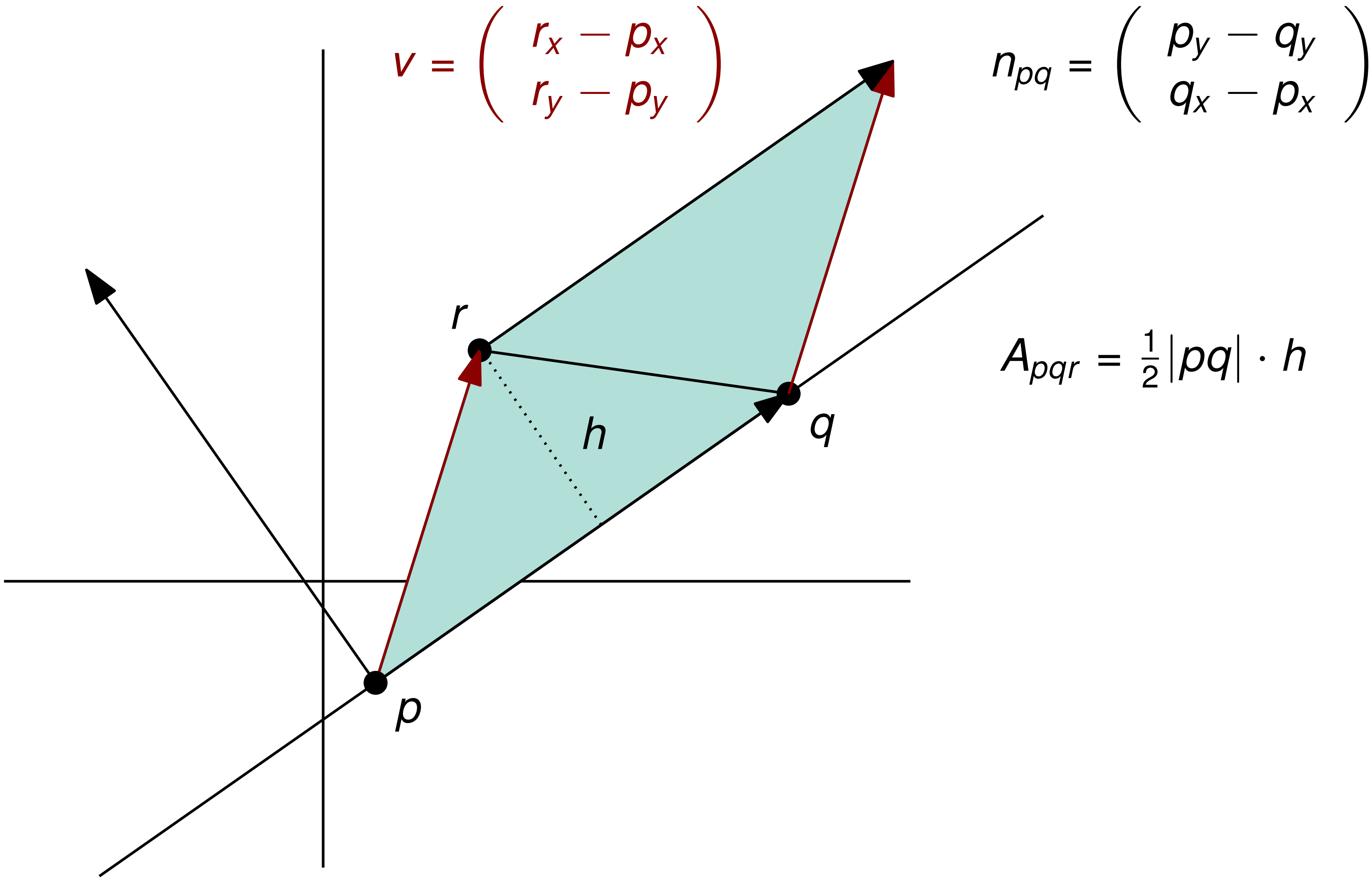
Zeige:: $|det(A)|$ = doppelter Flaecheninhalt des Dreiecks pqr

$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix} \qquad n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



Determinante

Zeige:: $|det(A)|$ = doppelter Flaecheninhalt des Dreiecks pqr

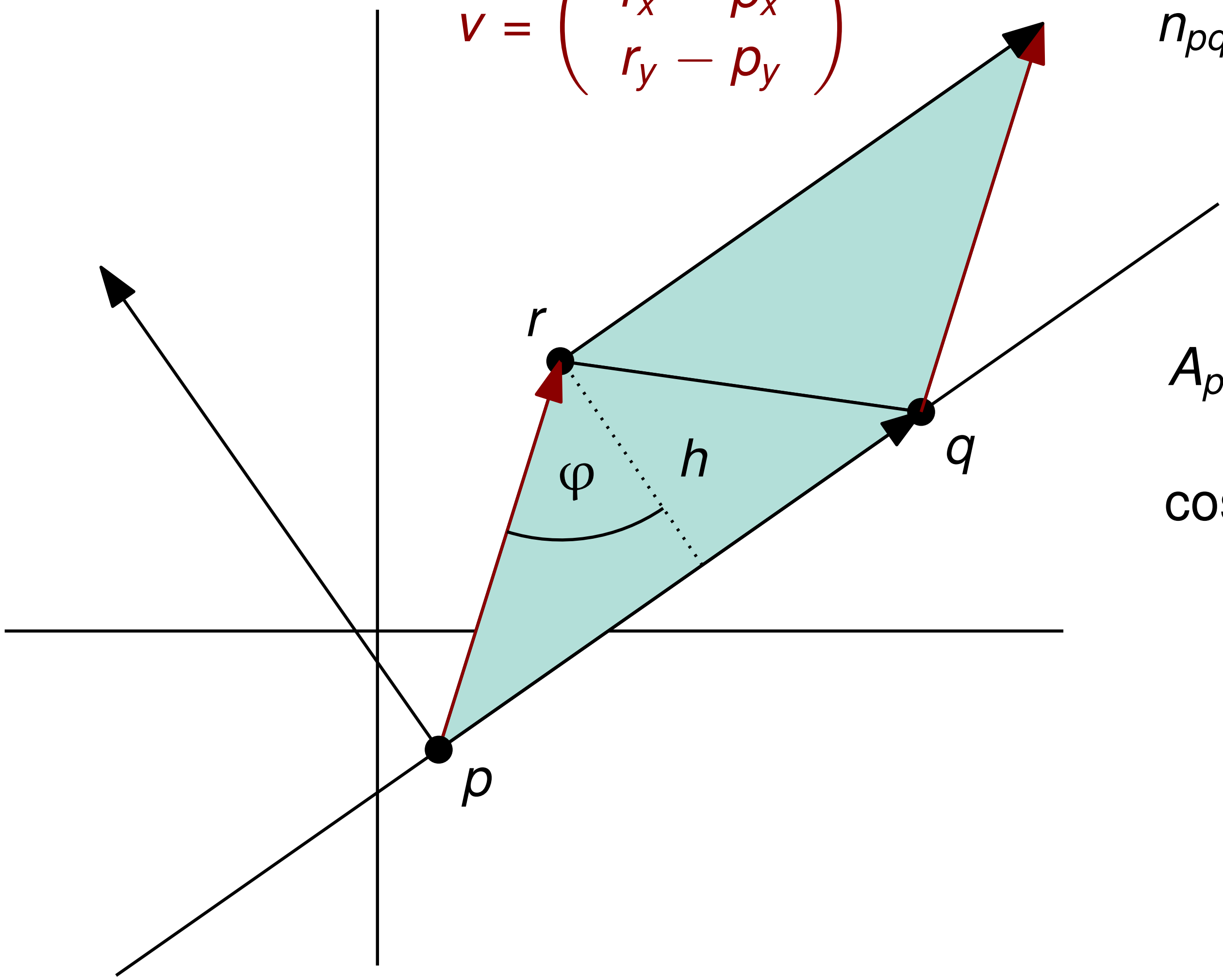


Determinante

Karlsruher Institut für Technologie

Zeige:: $|det(A)|$ = doppelter Flaecheninhalt des Dreiecks pqr

$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix} \qquad n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



$$A_{pqr} = \frac{1}{2}|pq| \cdot h$$

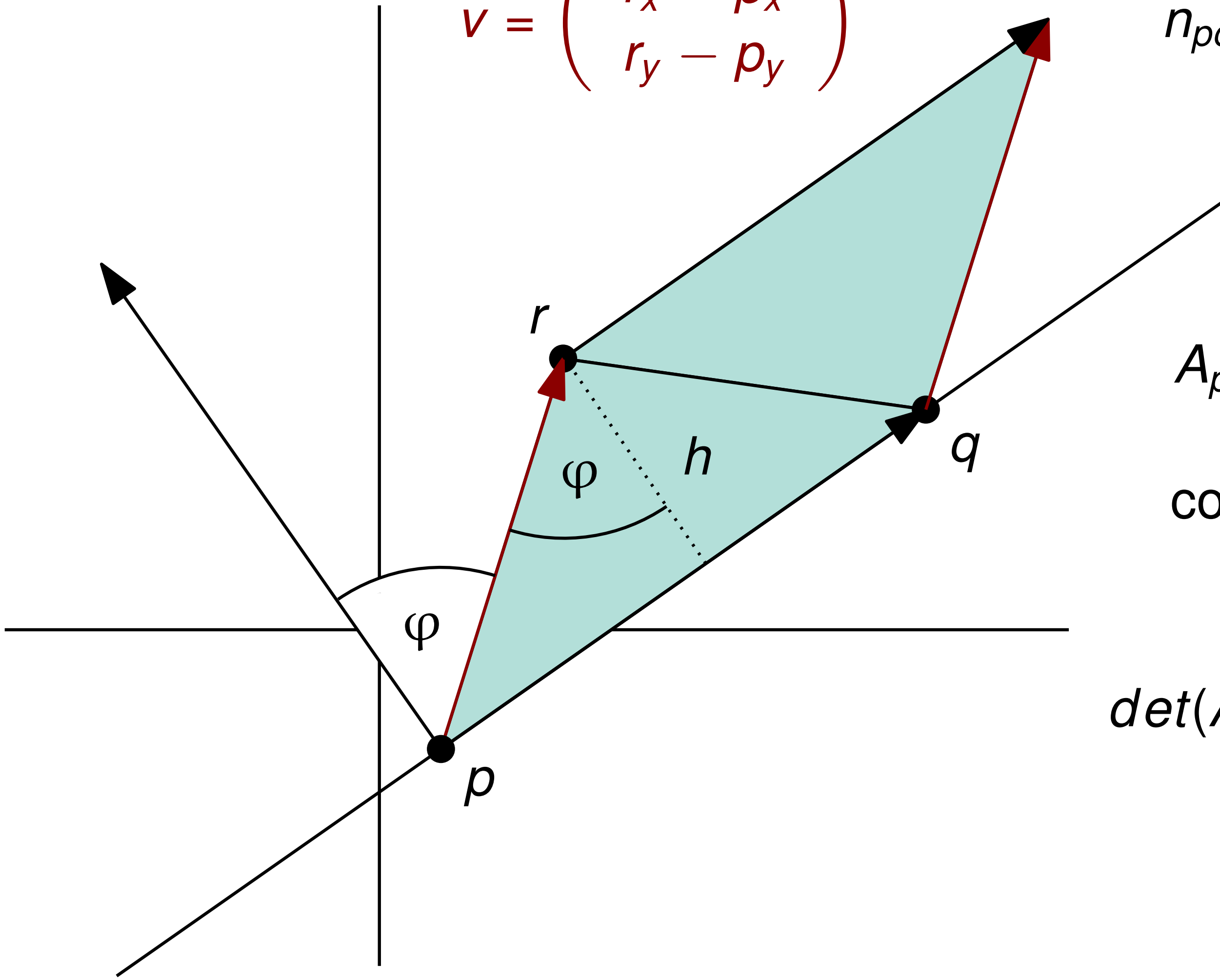
$$\cos\varphi = \frac{h}{|v|}$$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Determinante

Zeige:: $|det(A)|$ = doppelter Flaecheninhalt des Dreiecks *pqr*

$$v = \begin{pmatrix} r_x - p_x \\ r_y - p_y \end{pmatrix} \qquad n_{pq} = \begin{pmatrix} p_y - q_y \\ q_x - p_x \end{pmatrix}$$



$$A_{pqr} = \frac{1}{2}|pq| \cdot h$$

$$\cos\varphi = \frac{h}{|v|}$$

$$det(A) = v^T \cdot n_{pq} = |v| \cdot |n_{pq}| \cos\varphi$$