

Offline Bewegungsplanung: Kritische Orientierungen

Elmar Langetepe
University of Bonn

Weitere Vorgehensweise!

Weitere Vorgehensweise!

- Bis jetzt:

Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung

Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung
 - Zusammenhang in V^θ und ∂C_{frei}^θ sind identisch

Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung
 - Zusammenhang in V^θ und ∂C_{frei}^θ sind identisch
 - Nötige Änderungen:

Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung
 - Zusammenhang in V^θ und ∂C_{frei}^θ sind identisch
 - Nötige Änderungen: Anzahl $O(mn\lambda_6(mn))$ Orientierungen,

Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung
 - Zusammenhang in V^θ und ∂C_{frei}^θ sind identisch
 - Nötige Änderungen: Anzahl $O(mn\lambda_6(mn))$ Orientierungen, Aktualisierung für alle diese: $O(mn\lambda_6(mn) \log(mn))$

Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung
 - Zusammenhang in V^θ und ∂C_{frei}^θ sind identisch
 - Nötige Änderungen: Anzahl $O(mn\lambda_6(mn))$ Orientierungen, Aktualisierung für alle diese: $O(mn\lambda_6(mn) \log(mn))$
 - Leider nur Obermenge berechnet:

Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung
 - Zusammenhang in V^θ und ∂C_{frei}^θ sind identisch
 - Nötige Änderungen: Anzahl $O(mn\lambda_6(mn))$ Orientierungen, Aktualisierung für alle diese: $O(mn\lambda_6(mn) \log(mn))$
 - Leider nur Obermenge berechnet: Kandidaten (z.B. Krit. Platz.)

Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung
 - Zusammenhang in V^θ und ∂C_{frei}^θ sind identisch
 - Nötige Änderungen: Anzahl $O(mn\lambda_6(mn))$ Orientierungen, Aktualisierung für alle diese: $O(mn\lambda_6(mn) \log(mn))$
 - Leider nur Obermenge berechnet:
Kandidaten (z.B. Krit. Platz.)
 - Eventuell nicht frei!

Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung
 - Zusammenhang in V^θ und ∂C_{frei}^θ sind identisch
 - Nötige Änderungen: Anzahl $O(mn\lambda_6(mn))$ Orientierungen, Aktualisierung für alle diese: $O(mn\lambda_6(mn) \log(mn))$
 - Leider nur Obermenge berechnet:
Kandidaten (z.B. Krit. Platz.)
 - Eventuell nicht frei!
- Aufgabe:

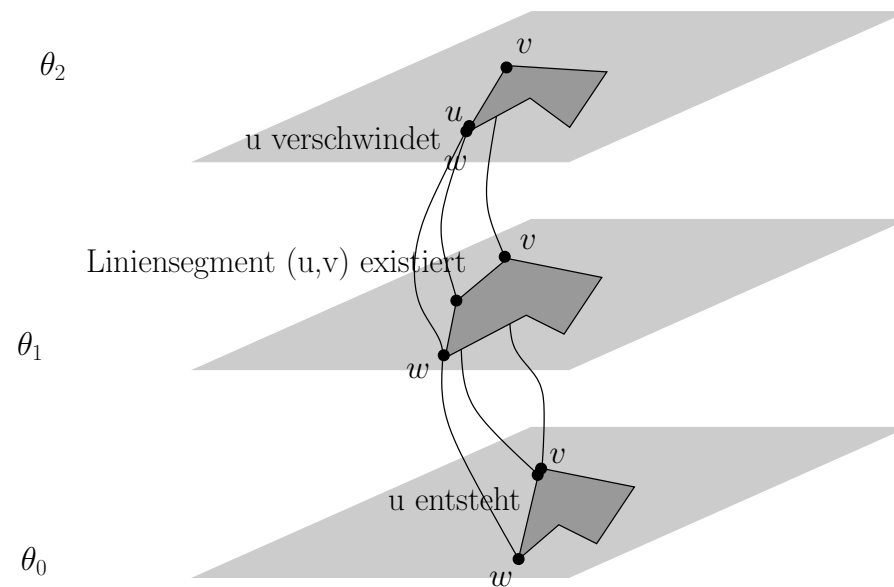
Weitere Vorgehensweise!

- Bis jetzt:
 - Graph V^θ für feste Orientierung
 - Zusammenhang in V^θ und ∂C_{frei}^θ sind identisch
 - Nötige Änderungen: Anzahl $O(mn\lambda_6(mn))$ Orientierungen, Aktualisierung für alle diese: $O(mn\lambda_6(mn) \log(mn))$
 - Leider nur Obermenge berechnet:
Kandidaten (z.B. Krit. Platz.)
 - Eventuell nicht frei!
- Aufgabe:
 - Alle diese Informationen in einer Struktur für die Bahnplanung abspeichern, zusammenfassen

– Dabei: Die freien herausfiltern

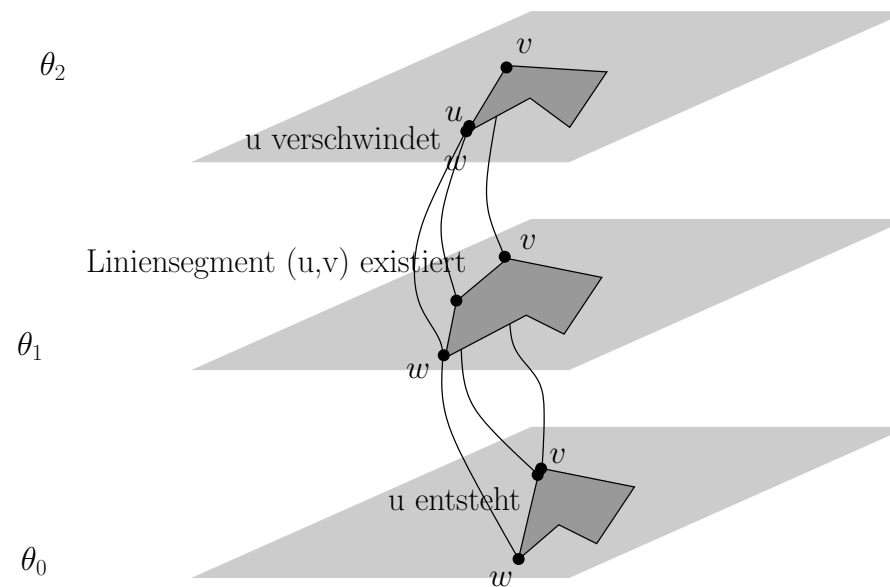
- Dabei: Die freien herausfiltern
- Bahnplanung mit Struktur

Struktur: Kantengraph **Def.: 2.43**



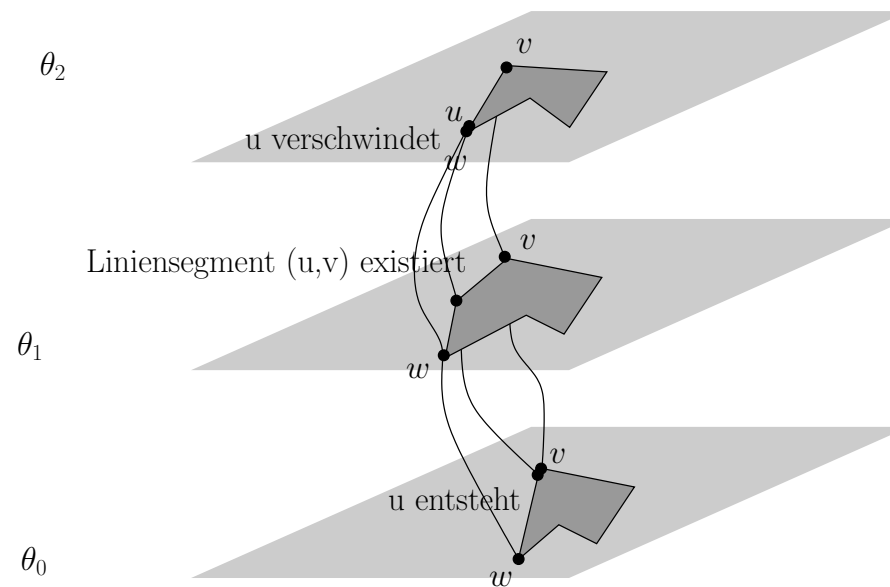
Struktur: Kantengraph **Def.: 2.43**

- Wie lange überlebt ein Knoten in C_{frei}^θ



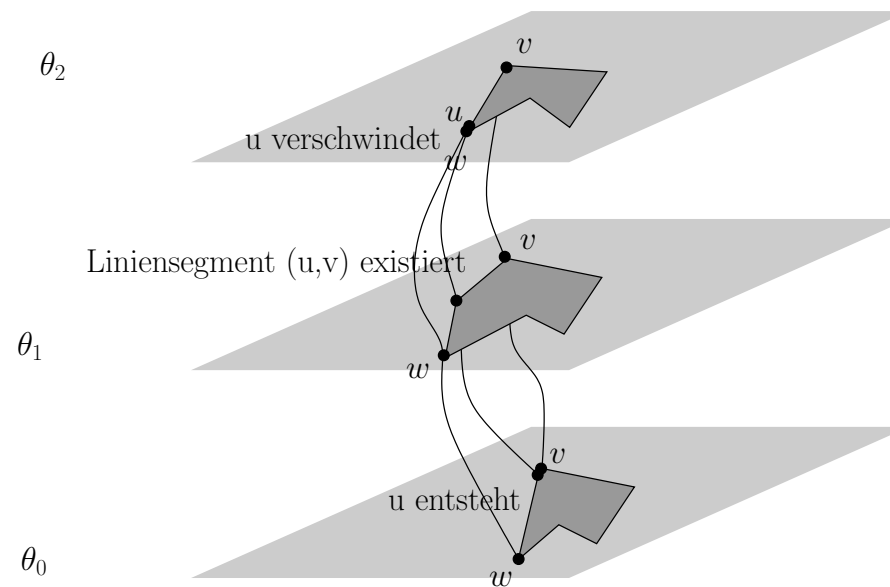
Struktur: Kantengraph **Def.: 2.43**

- Wie lange überlebt ein Knoten in C_{frei}^θ
- Wie lange überlebt eine Kante zwischen Knoten in C_{frei}^θ



Struktur: Kantengraph **Def.: 2.43**

- Wie lange überlebt ein Knoten in C_{frei}^θ
- Wie lange überlebt eine Kante zwischen Knoten in C_{frei}^θ
- Bilden eine gekrümmte Fläche in C_{frei} ?



Formal: Kantengraph Def.: 2.42

Formal: Kantengraph Def.: 2.42

- θ_0 nicht-kritische Orientierung, u Knoten in V^{θ_0} (Beschriftung)

Formal: Kantengraph Def.: 2.42

- θ_0 nicht-kritische Orientierung, u Knoten in V^{θ_0} (Beschriftung)
- Lebensdauer von u : max. Intervall $L(u) = (\theta_1, \theta_2) \ni \theta_0$, Knoten u kommt in jedem Knotengraph $V^\theta, \theta \in L(u)$, vor

Formal: Kantengraph Def.: 2.42

- θ_0 nicht-kritische Orientierung, u Knoten in V^{θ_0} (Beschriftung)
- Lebensdauer von u : max. Intervall $L(u) = (\theta_1, \theta_2) \ni \theta_0$, Knoten u kommt in jedem Knotengraph V^θ , $\theta \in L(u)$, vor
- Kantengraph E :

Formal: Kantengraph Def.: 2.42

- θ_0 nicht-kritische Orientierung, u Knoten in V^{θ_0} (Beschriftung)
- Lebensdauer von u : max. Intervall $L(u) = (\theta_1, \theta_2) \ni \theta_0$, Knoten u kommt in jedem Knotengraph V^θ , $\theta \in L(u)$, vor
- Kantengraph E :
 - erweiterte Knoten $(u, L(u))$

Formal: Kantengraph Def.: 2.42

- θ_0 nicht-kritische Orientierung, u Knoten in V^{θ_0} (Beschriftung)
- Lebensdauer von u : max. Intervall $L(u) = (\theta_1, \theta_2) \ni \theta_0$, Knoten u kommt in jedem Knotengraph V^θ , $\theta \in L(u)$, vor
- Kantengraph E :
 - erweiterte Knoten $(u, L(u))$
 - Knoten $(u, L(u))$ und $(v, L(v))$ bilden Kante e in E $:\Leftrightarrow$ es ex. $\theta \in L(u) \cap L(v)$ mit V^θ hat Kante (u, v)

Formal: Kantengraph Def.: 2.42

- θ_0 nicht-kritische Orientierung, u Knoten in V^{θ_0} (Beschriftung)
- Lebensdauer von u : max. Intervall $L(u) = (\theta_1, \theta_2) \ni \theta_0$, Knoten u kommt in jedem Knotengraph V^θ , $\theta \in L(u)$, vor
- Kantengraph E :
 - erweiterte Knoten $(u, L(u))$
 - Knoten $(u, L(u))$ und $(v, L(v))$ bilden Kante e in E $:\Leftrightarrow$ es ex. $\theta \in L(u) \cap L(v)$ mit V^θ hat Kante (u, v)
 - Lebensdauer Kante e : max. Intervall $L(e) = (\theta_1, \theta_2) \ni \theta_0$, Kante (u, v) kommt in jedem Knotengraph V^θ , $\theta \in L(e)$ vor

Formal: Kantengraph Def.: 2.42

- θ_0 nicht-kritische Orientierung, u Knoten in V^{θ_0} (Beschriftung)
- Lebensdauer von u : max. Intervall $L(u) = (\theta_1, \theta_2) \ni \theta_0$, Knoten u kommt in jedem Knotengraph V^θ , $\theta \in L(u)$, vor
- Kantengraph E :
 - erweiterte Knoten $(u, L(u))$
 - Knoten $(u, L(u))$ und $(v, L(v))$ bilden Kante e in E $:\Leftrightarrow$ es ex. $\theta \in L(u) \cap L(v)$ mit V^θ hat Kante (u, v)
 - Lebensdauer Kante e : max. Intervall $L(e) = (\theta_1, \theta_2) \ni \theta_0$, Kante (u, v) kommt in jedem Knotengraph V^θ , $\theta \in L(e)$ vor
 - $e = ((u, L(u)), (v, L(v)))$ und $L(e)$ ergibt $(e, L(e))$

Berechnung der Krit. Orient. 2.3.3

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* :

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

1. Eckpunkte der unteren Konturen $UK(A(O_i))$

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

1. Eckpunkte der unteren Konturen $UK(A(O_i))$
2. Zyklische Tripel (O_j, O_k, O_l) mit Krit. Platz.

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

1. Eckpunkte der unteren Konturen $UK(A(O_i))$
2. Zyklische Tripel (O_j, O_k, O_l) mit Krit. Platz.
3. Krit. Platzierung vom Typ: Ecke/Ecke plus Ecke/Kante

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

1. Eckpunkte der unteren Konturen $UK(A(O_i))$
2. Zyklische Tripel (O_j, O_k, O_l) mit Krit. Platz.
3. Krit.Platzierung vom Typ: Ecke/Ecke plus Ecke/Kante
4. Orient.: Kante des erweiterten Hindernisse ist waagerecht

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

1. Eckpunkte der unteren Konturen $UK(A(O_i))$
2. Zyklische Tripel (O_j, O_k, O_l) mit Krit. Platz.
3. Krit.Platzierung vom Typ: Ecke/Ecke plus Ecke/Kante
4. Orient.: Kante des erweiterten Hindernisse ist waagrecht
5. Orient.: Konv. Ecke von C_{frei}^θ liegt unterhalb and. Ecken

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

1. Eckpunkte der unteren Konturen $UK(A(O_i))$
 2. Zyklische Tripel (O_j, O_k, O_l) mit Krit. Platz.
 3. Krit.Platzierung vom Typ: Ecke/Ecke plus Ecke/Kante
 4. Orient.: Kante des erweiterten Hindernisse ist waagrecht
 5. Orient.: Konv. Ecke von C_{frei}^θ liegt unterhalb and. Ecken
- 1 und 2 zu T^+ : Obermenge von Krit. Platz. i)-ii) (Alg. 2.8)

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

1. Eckpunkte der unteren Konturen $UK(A(O_i))$
 2. Zyklische Tripel (O_j, O_k, O_l) mit Krit. Platz.
 3. Krit.Platzierung vom Typ: Ecke/Ecke plus Ecke/Kante
 4. Orient.: Kante des erweiterten Hindernisse ist waagerecht
 5. Orient.: Konv. Ecke von C_{frei}^θ liegt unterhalb and. Ecken
- 1 und 2 zu T^+ : Obermenge von Krit. Platz. i)-ii) (Alg. 2.8)
 - 3 war Übungsaufgabe (und Alg. 2.9): Krit. Platz. iii)-iv)

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

1. Eckpunkte der unteren Konturen $UK(A(O_i))$
 2. Zyklische Tripel (O_j, O_k, O_l) mit Krit. Platz.
 3. Krit. Platzierung vom Typ: Ecke/Ecke plus Ecke/Kante
 4. Orient.: Kante des erweiterten Hindernisse ist waagrecht
 5. Orient.: Konv. Ecke von C_{frei}^θ liegt unterhalb and. Ecken
- 1 und 2 zu T^+ : Obermenge von Krit. Platz. i)-ii) (Alg. 2.8)
 - 3 war Übungsaufgabe (und Alg. 2.9): Krit. Platz. iii)-iv)
 - 4 und 5: Obermenge von Krit. Orient. v) und vi), Alg. folgt aus Beschreibung vorab

Berechnung der Krit. Orient. 2.3.3

Def: Menge T^* : Berechnet in $O(mn\lambda_6(mn)\log(mn))$

1. Eckpunkte der unteren Konturen $UK(A(O_i))$
 2. Zyklische Tripel (O_j, O_k, O_l) mit Krit. Platz.
 3. Krit. Platzierung vom Typ: Ecke/Ecke plus Ecke/Kante
 4. Orient.: Kante des erweiterten Hindernisse ist waagrecht
 5. Orient.: Konv. Ecke von C_{frei}^θ liegt unterhalb and. Ecken
- 1 und 2 zu T^+ : Obermenge von Krit. Platz. i)-ii) (Alg. 2.8)
 - 3 war Übungsaufgabe (und Alg. 2.9): Krit. Platz. iii)-iv)
 - 4 und 5: Obermenge von Krit. Orient. v) und vi), Alg. folgt aus Beschreibung vorab

Problem: Orientierungen aus T^* sind eventuell nicht frei!!

T^* Obermenge der gültigen Krit. Orient.

T^* Obermenge der gültigen Krit. Orient.

Menge T^* : Bereits berechnet in $O(mn\lambda_6(mn) \log(mn))$

T^* Obermenge der gültigen Krit. Orient.

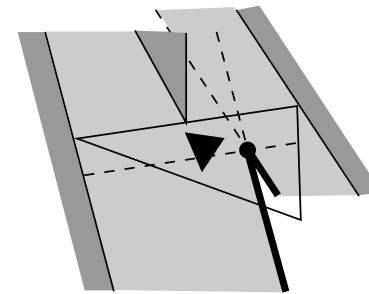
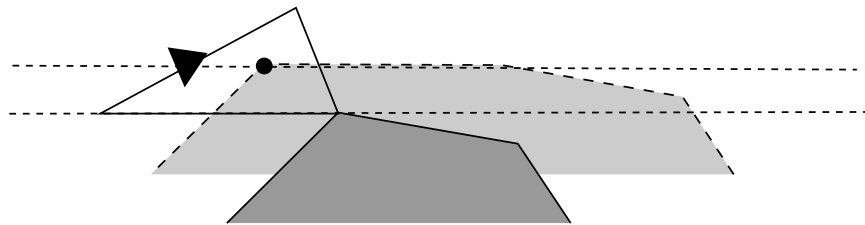
Menge T^* : Bereits berechnet in $O(mn\lambda_6(mn)\log(mn))$

Beispiele: 1. Krit. Platzierung! 4. Waagerechte Kante! Nicht frei!

T^* Obermenge der gültigen Krit. Orient.

Menge T^* : Bereits berechnet in $O(mn\lambda_6(mn) \log(mn))$

Beispiele: 1. Krit. Platzierung! 4. Waagerechte Kante! Nicht frei!



Idee des Filterns

Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?

Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$),

Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5) : (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2 , O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten,

Idee des Filterns

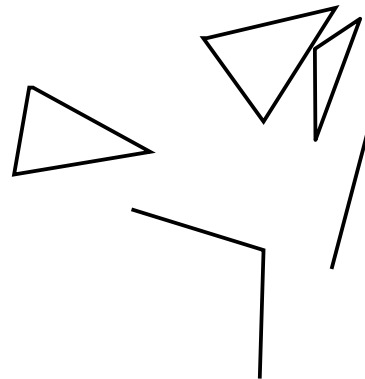
- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5) : (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2 , O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen

Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5) : (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2 , O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O :

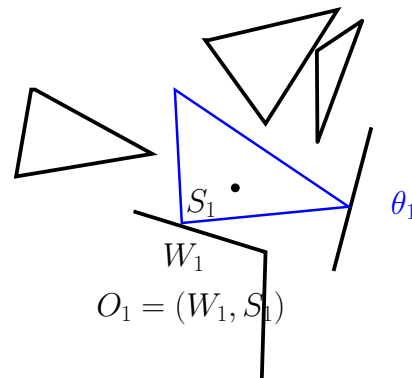
Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!



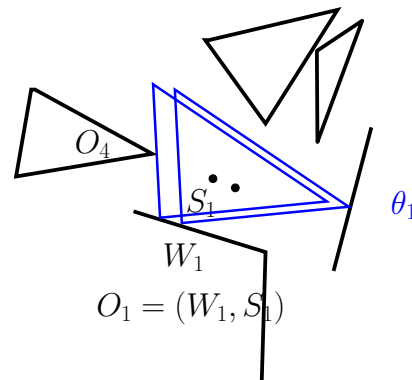
Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!
- Beispiel!!



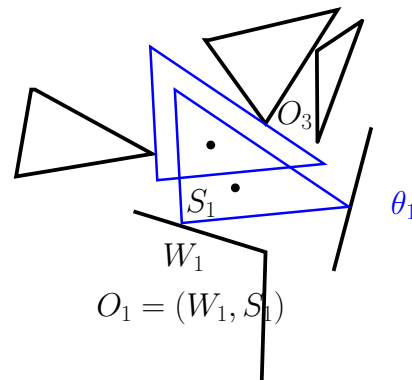
Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!
- Beispiel!!



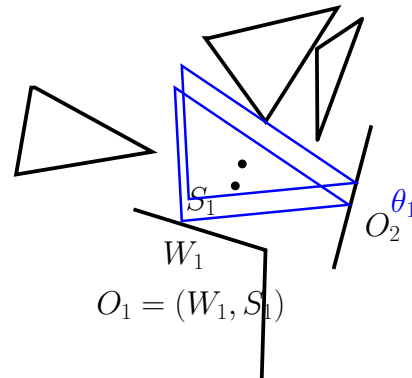
Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!
- Beispiel!!



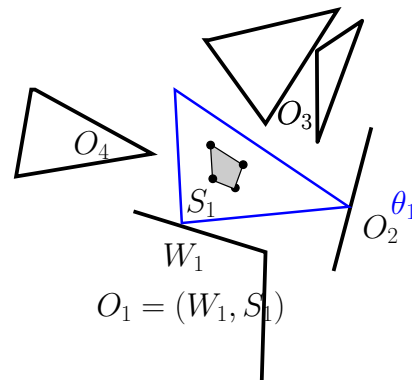
Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!
- Beispiel!!



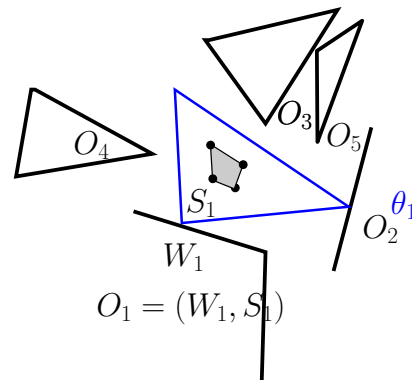
Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!
- Beispiel!!



Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!
- Beispiel!!



$$L(O_1)(\theta_1) = \{O_2, O_4\}$$

$$L(O_2)(\theta_1) = \{O_1, O_3\}$$

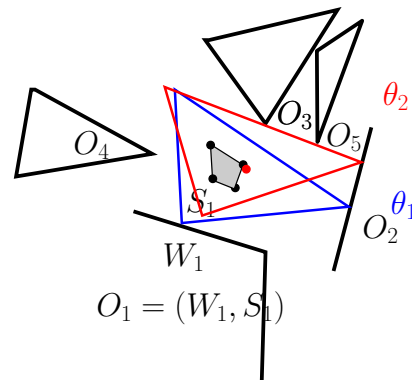
$$L(O_3)(\theta_1) = \{O_2, O_4\}$$

$$L(O_4)(\theta_1) = \{O_1, O_3\}$$

$$L(O_5)(\theta_1) = \{\}$$

Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!
- Beispiel!!



$$L(O_1)(\theta_1) = \{O_2, O_4\}$$

$$L(O_2)(\theta_1) = \{O_1, O_3\}$$

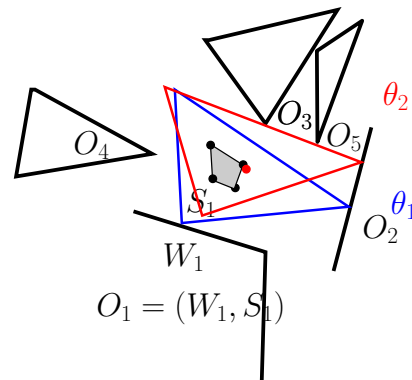
$$L(O_3)(\theta_1) = \{O_2, O_4\}$$

$$L(O_4)(\theta_1) = \{O_1, O_3\}$$

$$L(O_5)(\theta_1) = \{\}$$

Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!
- Beispiel!!



$$L(O_1)(\theta_1) = \{O_2, O_4\}$$

$$L(O_2)(\theta_1) = \{O_1, O_3\}$$

$$L(O_3)(\theta_1) = \{O_2, O_4\}$$

$$L(O_4)(\theta_1) = \{O_1, O_3\}$$

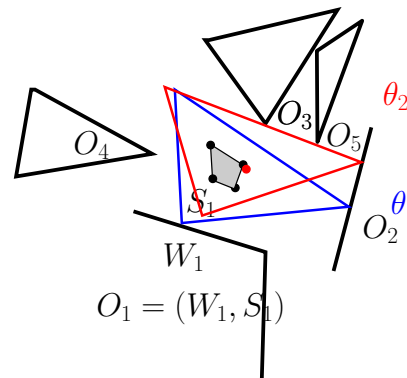
$$L(O_5)(\theta_1) = \{\}$$

(O_2, O_3, O_5)

Frei?

Idee des Filterns

- Krit. Platzierung θ_2 mit drei Kontakten (O_2, O_3, O_5): (Halb)Frei?
- **Bereits in Komponente**: Kurz vorher ($\theta_1 = \theta_2 - \epsilon$), O_2, O_3 und O_5 erzeugten mind. einen Knoten von $C_{frei}^{\theta_1}(V^{\theta_1})$, mutiert zu neuem Knoten, oder zwei Knoten verschmelzen
- Kontakt O : Listen L_O der Knotenerzeugung!
- Beispiel!!
- Test: Liste von L_{O_2} oder L_{O_3} oder L_{O_5} enthält andere



$$L(O_1)(\theta_1) = \{O_2, O_4\}$$

$$L(O_2)(\theta_1) = \{O_1, O_3\}$$

$$L(O_3)(\theta_1) = \{O_2, O_4\}$$

$$L(O_4)(\theta_1) = \{O_1, O_3\}$$

$$L(O_5)(\theta_1) = \{\}$$

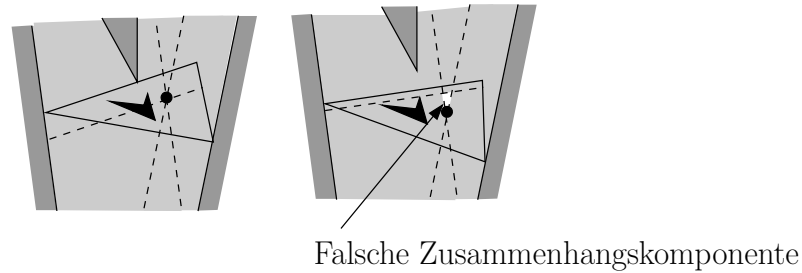
(O_2, O_3, O_5)

Frei?

Problem: Start Komponente

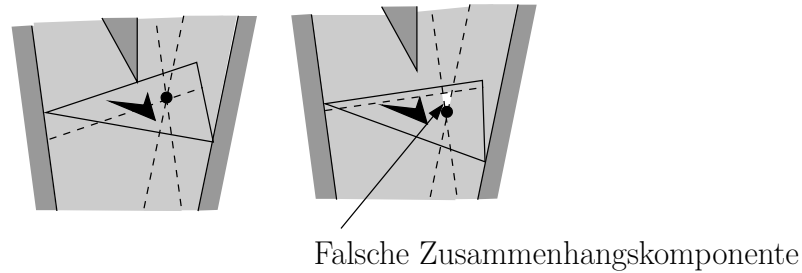
Problem: Start Komponente

- Beim Entstehen gibt es keine Listen und gar keinen Kontakt!



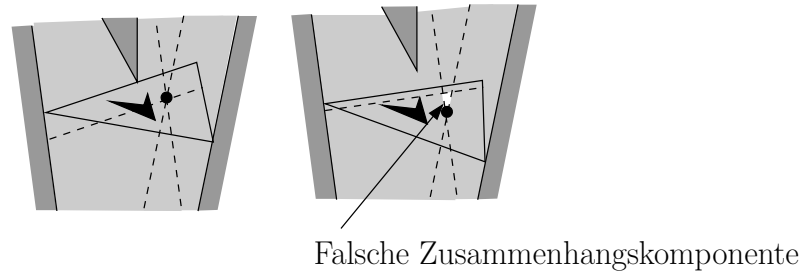
Problem: Start Komponente

- Beim Entstehen gibt es keine Listen und gar keinen Kontakt!
- Idee: Nur die Komponente, die Startpunkt s enthält: Z_s



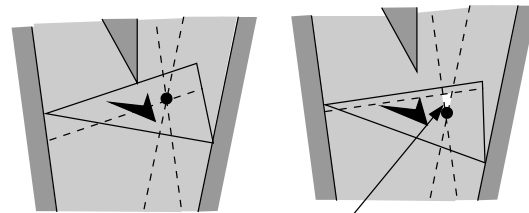
Problem: Start Komponente

- Beim Entstehen gibt es keine Listen und gar keinen Kontakt!
- Idee: Nur die Komponente, die Startpunkt s enthält: Z_s
- Zu Beginn: Zshg. Teil von C_{frei}^θ und V^θ



Problem: Start Komponente

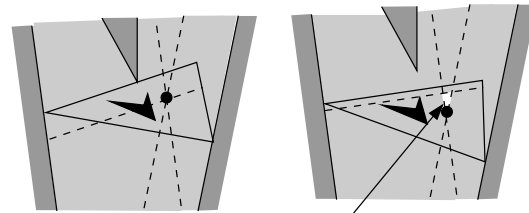
- Beim Entstehen gibt es keine Listen und gar keinen Kontakt!
- Idee: Nur die Komponente, die Startpunkt s enthält: Z_s
- Zu Beginn: Zshg. Teil von C_{frei}^θ und V^θ
- Wann können andere korr. Komponenten dazu kommen?
Vereinigung!



Falsche Zusammenhangskomponente

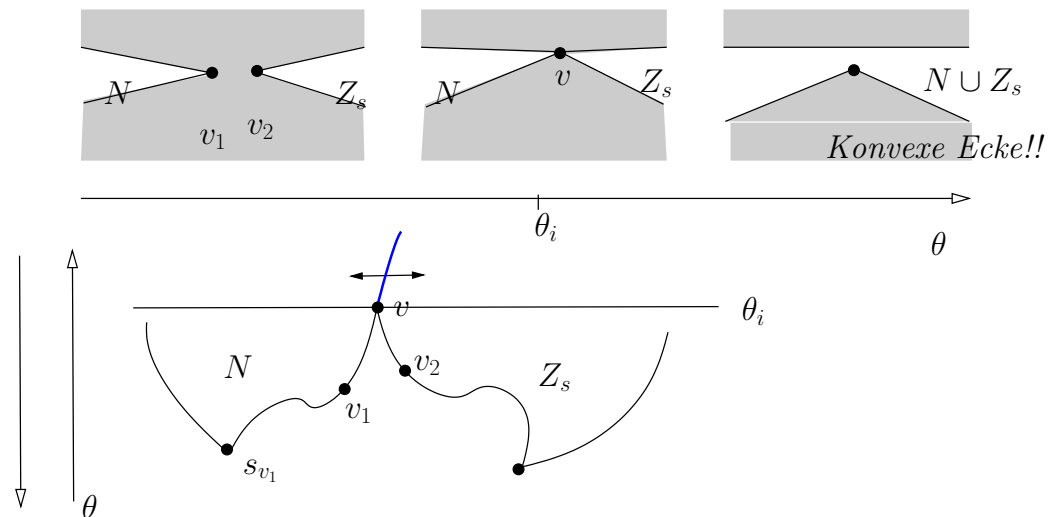
Problem: Start Komponente

- Beim Entstehen gibt es keine Listen und gar keinen Kontakt!
- Idee: Nur die Komponente, die Startpunkt s enthält: Z_s
- Zu Beginn: Zshg. Teil von C_{frei}^θ und V^θ
- Wann können andere korr. Komponenten dazu kommen?
Vereinigung!
- Krit. Orientierung Typ iv)!!



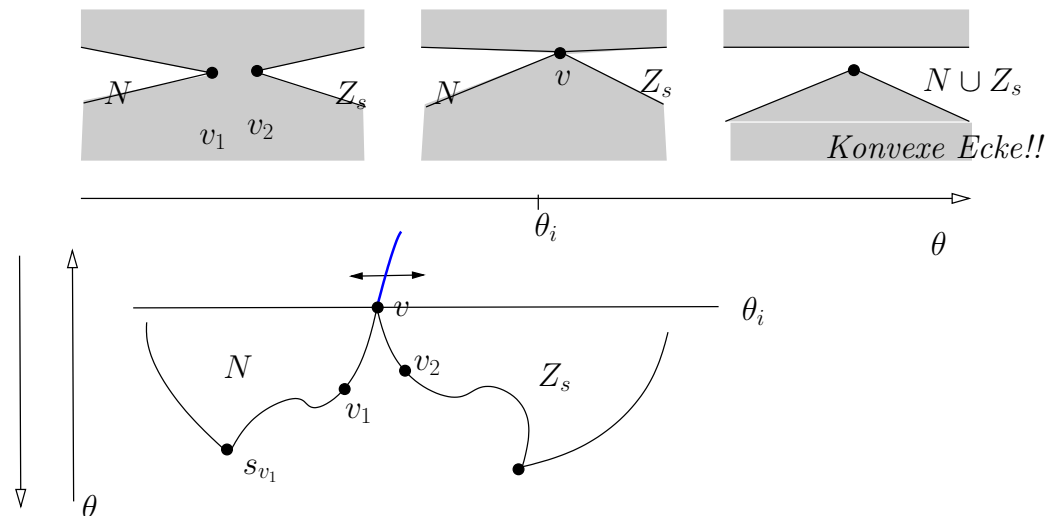
Falsche Zusammenhangskomponente

Vereinigung von Komponenten!



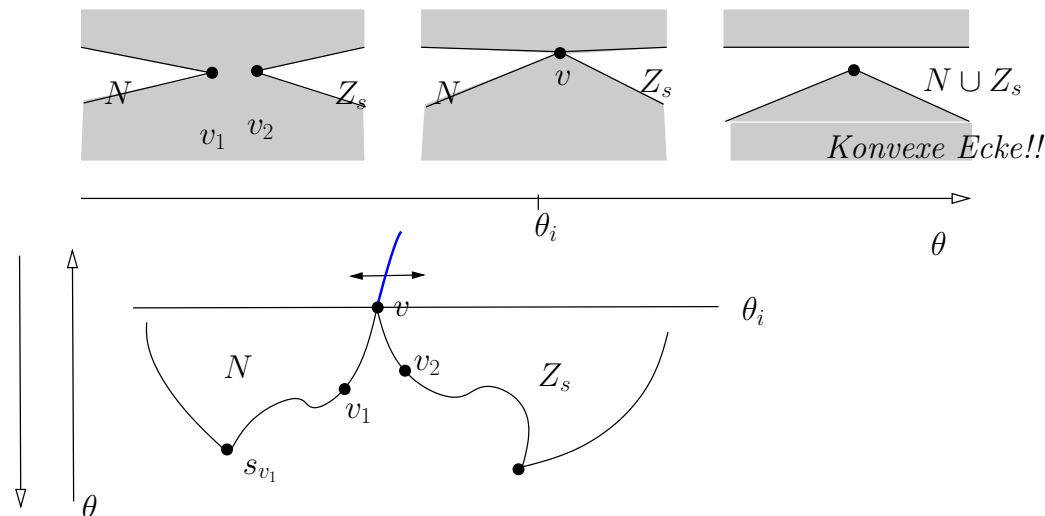
Vereinigung von Komponenten!

- Ecke/Ecke, Kante/Ecke: Konv. Ecke in C_{verb}^θ entsteht!



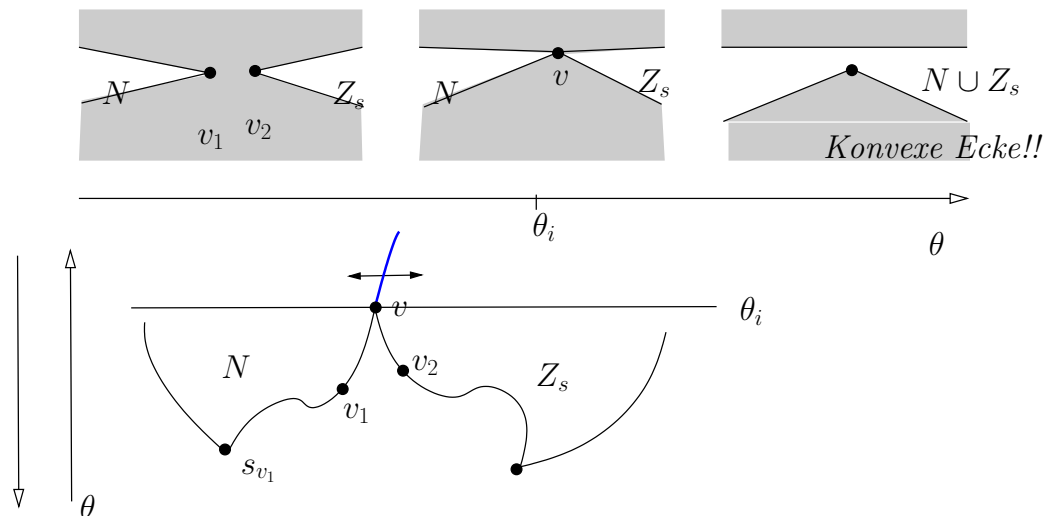
Vereinigung von Komponenten!

- Ecke/Ecke, Kante/Ecke: Konv. Ecke in C_{verb}^θ entsteht!
- Die freien Platz. dieser Art berechnen: Alg. 2.9



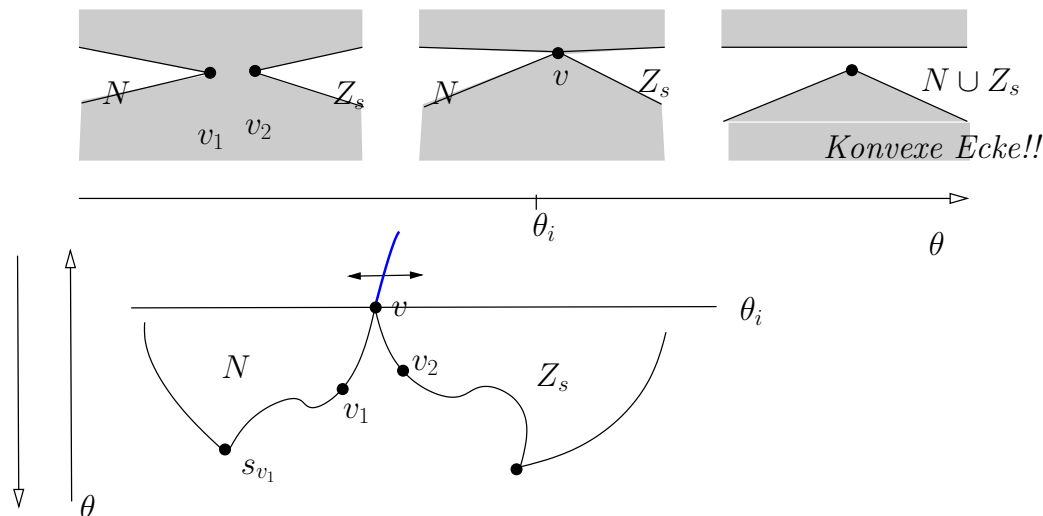
Vereinigung von Komponenten!

- Ecke/Ecke, Kante/Ecke: Konv. Ecke in C_{verb}^θ entsteht!
- Die freien Platz. dieser Art berechnen: Alg. 2.9
- $T' \subseteq T^*$: Freie Platzierung beginn. konvexer Ecke v



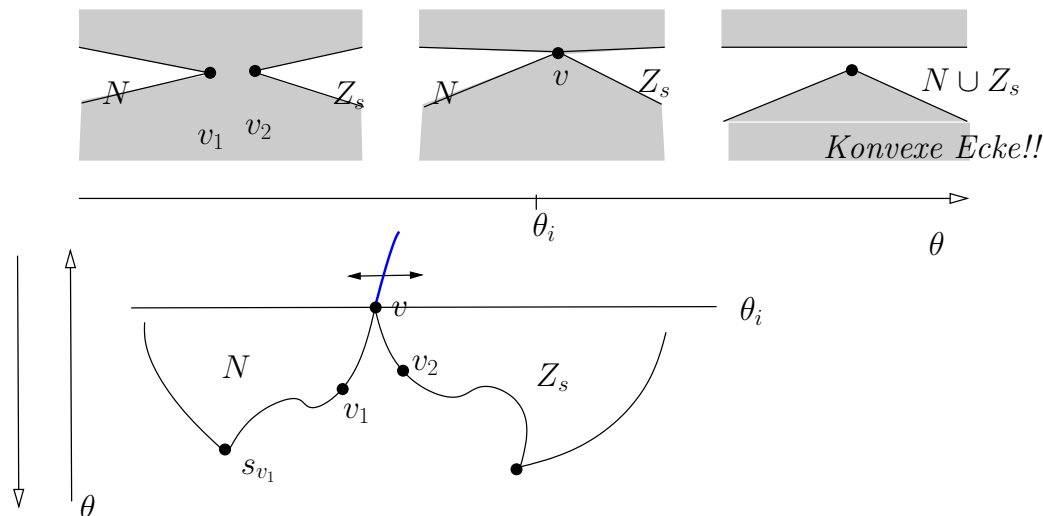
Vereinigung von Komponenten!

- Ecke/Ecke, Kante/Ecke: Konv. Ecke in C_{verb}^θ entsteht!
- Die freien Platz. dieser Art berechnen: Alg. 2.9
- $T' \subseteq T^*$: Freie Platzierung beginn. konvexer Ecke v
- Komponente, die irgendwann die konvexe Ecke enthält muss irgendwo entstanden sein



Vereinigung von Komponenten!

- Ecke/Ecke, Kante/Ecke: Konv. Ecke in C_{verb}^θ entsteht!
- Die freien Platz. dieser Art berechnen: Alg. 2.9
- $T' \subseteq T^*$: Freie Platzierung beginn. konvexer Ecke v
- Komponente, die irgendwann die konvexe Ecke enthält muss irgendwo entstanden sein
- Entstehungszeitpunkte s_v bestimmen: T''



Lösung

Lösung

Kritische Orientierungen $T^* O(mn \lambda_6(mn) \log(mn))$

Lösung

Kritische Orientierungen T^* $O(mn \lambda_6(mn) \log(mn))$

Aus T^* diejenigen herausfiltern, die tatsächlich frei sind:

Lösung

Kritische Orientierungen $T^* O(mn \lambda_6(mn) \log(mn))$

Aus T^* diejenigen herausfiltern, die tatsächlich frei sind:

1. Bei korrekter Startkomponente beginnen

Lösung

Kritische Orientierungen T^* $O(mn \lambda_6(mn) \log(mn))$

Aus T^* diejenigen herausfiltern, die tatsächlich frei sind:

1. Bei korrekter Startkomponente beginnen
2. Feststellen, wo sich Komponenten korrekt verbinden T'

Lösung

Kritische Orientierungen T^* $O(mn \lambda_6(mn) \log(mn))$

Aus T^* diejenigen herausfiltern, die tatsächlich frei sind:

1. Bei korrekter Startkomponente beginnen
2. Feststellen, wo sich Komponenten korrekt verbinden T'
3. Startpunkte korrekter Komponenten berücksichtigen T''

Lösung

Kritische Orientierungen T^* $O(mn \lambda_6(mn) \log(mn))$

Aus T^* diejenigen herausfiltern, die tatsächlich frei sind:

1. Bei korrekter Startkomponente beginnen
2. Feststellen, wo sich Komponenten korrekt verbinden T'
3. Startpunkte korrekter Komponenten berücksichtigen T''
4. Daraus T in $O(mn \lambda_6(mn) \log(mn))$ berechnen

Lösung

Kritische Orientierungen T^* $O(mn \lambda_6(mn) \log(mn))$

Aus T^* diejenigen herausfiltern, die tatsächlich frei sind:

1. Bei korrekter Startkomponente beginnen
2. Feststellen, wo sich Komponenten korrekt verbinden T'
3. Startpunkte korrekter Komponenten berücksichtigen T''
4. Daraus T in $O(mn \lambda_6(mn) \log(mn))$ berechnen
5. Orientierungen T enthält alles was wir brauchen! Sortiert!

Lösung

Kritische Orientierungen $T^* O(mn \lambda_6(mn) \log(mn))$

Aus T^* diejenigen herausfiltern, die tatsächlich frei sind:

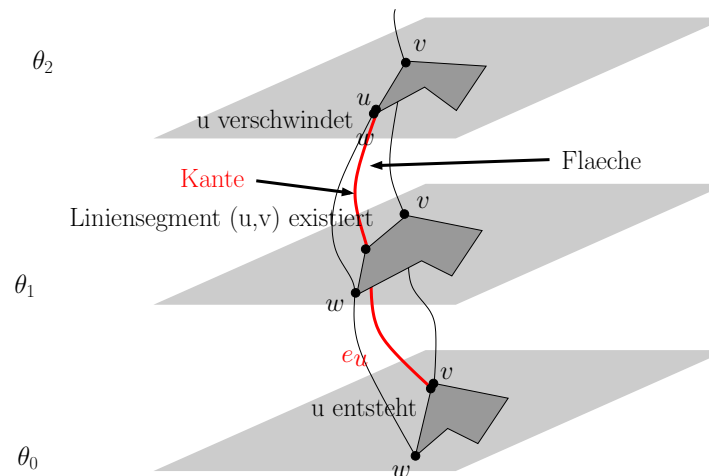
1. Bei korrekter Startkomponente beginnen
2. Feststellen, wo sich Komponenten korrekt verbinden T'
3. Startpunkte korrekter Komponenten berücksichtigen T''
4. Daraus T in $O(mn \lambda_6(mn) \log(mn))$ berechnen
5. Orientierungen T enthält alles was wir brauchen! Sortiert!

Danach: Mit T den notwendigen Kantengraph aufbauen!!

Aufbau des Kantengraphen: Alg. 2.12

Damit läßt sich sukzessive V^θ bestimmen, sukzessive der Kantengraph aufbauen Zeit $O(mn \lambda_6(mn) \log(mn))$

- erweiterte Knoten $(u, L(u))$: Kante in C_{frei}
- erweiterter Knoten $(v, L(v))$: Kante in C_{frei}
- erw. Kante $e = ((u, L(u)), (v, L(v)))$ mit $L(e)$: Fläche in C_{frei}



Aufbau des Kantengraphen: Alg. 2.12

Aufbau des Kantengraphen: Alg. 2.12

- Berechne Menge T aller kritischen Orientierungen, sortiert nach aufsteigendem θ !

Aufbau des Kantengraphen: Alg. 2.12

- Berechne Menge T aller kritischen Orientierungen, sortiert nach aufsteigendem θ ! Zeit: $O(nm\lambda_6(mn) \log mn)$,

Aufbau des Kantengraphen: Alg. 2.12

- Berechne Menge T aller kritischen Orientierungen, sortiert nach aufsteigendem θ ! Zeit: $O(nm\lambda_6(mn) \log mn)$,
 $|T| = O(nm\lambda_6(mn))$

Aufbau des Kantengraphen: Alg. 2.12

- Berechne Menge T aller kritischen Orientierungen, sortiert nach aufsteigendem θ ! Zeit: $O(nm\lambda_6(mn) \log mn)$,
 $|T| = O(nm\lambda_6(mn))$
- Wähle $\theta_0 \notin T$ und berechne V^{θ_0} .

Aufbau des Kantengraphen: Alg. 2.12

- Berechne Menge T aller kritischen Orientierungen, sortiert nach aufsteigendem θ ! Zeit: $O(nm\lambda_6(mn) \log mn)$,
 $|T| = O(nm\lambda_6(mn))$
- Wähle $\theta_0 \notin T$ und berechne V^{θ_0} .
- Initialisiere E mit den Kanten und Knoten aus V^{θ_0} , dabei bleiben die Lebensdauerintervalle $L(u)$ zunächst undefiniert.

Aufbau des Kantengraphen: Alg. 2.12

- Berechne Menge T aller kritischen Orientierungen, sortiert nach aufsteigendem θ ! Zeit: $O(nm\lambda_6(mn) \log mn)$,
 $|T| = O(nm\lambda_6(mn))$
- Wähle $\theta_0 \notin T$ und berechne V^{θ_0} .
- Initialisiere E mit den Kanten und Knoten aus V^{θ_0} , dabei bleiben die Lebensdauerintervalle $L(u)$ zunächst undefiniert. (Wir sind mitten in einem Intervall!!)

Aufbau des Kantengraphen: Alg. 2.12

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive:

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive: Für Orientierung θ_i in T :

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive: Für Orientierung θ_i in T :
 - Bestimme die Änderungen in V^θ : Übergang von $\theta = \theta_i - \varepsilon$ nach $\theta = \theta_i + \varepsilon$

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive: Für Orientierung θ_i in T :
 - Bestimme die Änderungen in V^θ : Übergang von $\theta = \theta_i - \varepsilon$ nach $\theta = \theta_i + \varepsilon$
 - Entferne alle verschwindenden Knoten und Kanten $v_1, \dots, v_r, e_1, \dots, e_s$ aus V^θ ,

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive: Für Orientierung θ_i in T :
 - Bestimme die Änderungen in V^θ : Übergang von $\theta = \theta_i - \varepsilon$ nach $\theta = \theta_i + \varepsilon$
 - Entferne alle verschwindenden Knoten und Kanten $v_1, \dots, v_r, e_1, \dots, e_s$ aus V^θ , setze θ_i als obere Grenze der Lebensdauer von $L(v_j)$ bzw. $L(e_k)$ ein

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive: Für Orientierung θ_i in T :
 - Bestimme die Änderungen in V^θ : Übergang von $\theta = \theta_i - \varepsilon$ nach $\theta = \theta_i + \varepsilon$
 - Entferne alle verschwindenden Knoten und Kanten $v_1, \dots, v_r, e_1, \dots, e_s$ aus V^θ , setze θ_i als obere Grenze der Lebensdauer von $L(v_j)$ bzw. $L(e_k)$ ein
 - Füge alle neu entstandenen Knoten und Kanten $v_1^*, \dots, v_{r^*}, e_1^*, \dots, e_{s^*}$ in V^θ und E ein,

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive: Für Orientierung θ_i in T :
 - Bestimme die Änderungen in V^θ : Übergang von $\theta = \theta_i - \varepsilon$ nach $\theta = \theta_i + \varepsilon$
 - Entferne alle verschwindenden Knoten und Kanten $v_1, \dots, v_r, e_1, \dots, e_s$ aus V^θ , setze θ_i als obere Grenze der Lebensdauer von $L(v_j)$ bzw. $L(e_k)$ ein
 - Füge alle neu entstandenen Knoten und Kanten $v_1^*, \dots, v_r^*, e_1^*, \dots, e_s^*$ in V^θ und E ein, setze θ_i als untere Grenze der Lebensdauer von $L(v_j^*)$ bzw. $L(e_k^*)$ ein

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive: Für Orientierung θ_i in T :
 - Bestimme die Änderungen in V^θ : Übergang von $\theta = \theta_i - \varepsilon$ nach $\theta = \theta_i + \varepsilon$
 - Entferne alle verschwindenden Knoten und Kanten $v_1, \dots, v_r, e_1, \dots, e_s$ aus V^θ , setze θ_i als obere Grenze der Lebensdauer von $L(v_j)$ bzw. $L(e_k)$ ein
 - Füge alle neu entstandenen Knoten und Kanten $v_1^*, \dots, v_r^*, e_1^*, \dots, e_s^*$ in V^θ und E ein, setze θ_i als untere Grenze der Lebensdauer von $L(v_j^*)$ bzw. $L(e_k^*)$ ein
- Nach 2π Durchlauf bei θ_0 :

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive: Für Orientierung θ_i in T :
 - Bestimme die Änderungen in V^θ : Übergang von $\theta = \theta_i - \varepsilon$ nach $\theta = \theta_i + \varepsilon$
 - Entferne alle verschwindenden Knoten und Kanten $v_1, \dots, v_r, e_1, \dots, e_s$ aus V^θ , setze θ_i als obere Grenze der Lebensdauer von $L(v_j)$ bzw. $L(e_k)$ ein
 - Füge alle neu entstandenen Knoten und Kanten $v_1^*, \dots, v_r^*, e_1^*, \dots, e_s^*$ in V^θ und E ein, setze θ_i als untere Grenze der Lebensdauer von $L(v_j^*)$ bzw. $L(e_k^*)$ ein
- Nach 2π Durchlauf bei θ_0 : Trage noch fehlenden Lebensdauern nach,

Aufbau des Kantengraphen: Alg. 2.12

- Sukzessive: Für Orientierung θ_i in T :
 - Bestimme die Änderungen in V^θ : Übergang von $\theta = \theta_i - \varepsilon$ nach $\theta = \theta_i + \varepsilon$
 - Entferne alle verschwindenden Knoten und Kanten $v_1, \dots, v_r, e_1, \dots, e_s$ aus V^θ , setze θ_i als obere Grenze der Lebensdauer von $L(v_j)$ bzw. $L(e_k)$ ein
 - Füge alle neu entstandenen Knoten und Kanten $v_1^*, \dots, v_r^*, e_1^*, \dots, e_s^*$ in V^θ und E ein, setze θ_i als untere Grenze der Lebensdauer von $L(v_j^*)$ bzw. $L(e_k^*)$ ein
- Nach 2π Durchlauf bei θ_0 : Trage noch fehlenden Lebensdauern nach, genauer: fasse zwei Knoten der Art $(u, (\theta_1, \text{undef.}))$ und $(u, (\text{undef.}, \theta_2))$ zu Knoten $(u, (\theta_1, \theta_2))$ zusammen

Aufbau des Kantengraphen: Alg. 2.12

Aufbau des Kantengraphen: Alg. 2.12

Analyse:

Aufbau des Kantengraphen: Alg. 2.12

Analyse:

Alle Änderungen in $O(mn\lambda_6(mn) \log mn)$,

Aufbau des Kantengraphen: Alg. 2.12

Analyse:

Alle Änderungen in $O(mn\lambda_6(mn) \log mn)$, bereits gezeigt

Aufbau des Kantengraphen: Alg. 2.12

Analyse:

Alle Änderungen in $O(mn\lambda_6(mn) \log mn)$, bereits gezeigt

Kantengraph hat Komplexität $O(mn\lambda_6(mn))$

Aufbau des Kantengraphen: Alg. 2.12

Analyse:

Alle Änderungen in $O(mn\lambda_6(mn) \log mn)$, bereits gezeigt

Kantengraph hat Komplexität $O(mn\lambda_6(mn))$

Anfangs: $O(mn)$,

Aufbau des Kantengraphen: Alg. 2.12

Analyse:

Alle Änderungen in $O(mn\lambda_6(mn) \log mn)$, bereits gezeigt

Kantengraph hat Komplexität $O(mn\lambda_6(mn))$

Anfangs: $O(mn)$, dann kommen noch $O(mn\lambda_6(mn))$ Änderungen dazu

Aufbau des Kantengraphen: Alg. 2.12

Analyse:

Alle Änderungen in $O(mn\lambda_6(mn) \log mn)$, bereits gezeigt

Kantengraph hat Komplexität $O(mn\lambda_6(mn))$

Anfangs: $O(mn)$, dann kommen noch $O(mn\lambda_6(mn))$ Änderungen dazu

Aufbau: $O(mn\lambda_6(mn) \log mn)$!

Bahnplanung mit Kantengraph

Bahnplanung mit Kantengraph

- Bewegung von Orientierung (x_1, y_1, θ_1) zu (x_2, y_2, θ_2) ?

Bahnplanung mit Kantengraph

- Bewegung von Orientierung (x_1, y_1, θ_1) zu (x_2, y_2, θ_2) ?
- Kantengraph repräsentiert den Rand von C_{frei}

Bahnplanung mit Kantengraph

- Bewegung von Orientierung (x_1, y_1, θ_1) zu (x_2, y_2, θ_2) ?
- Kantengraph repräsentiert den Rand von C_{frei}
- Erstmal auf den Rand!!

Bahnplanung mit Kantengraph

- Bewegung von Orientierung (x_1, y_1, θ_1) zu (x_2, y_2, θ_2) ?
- Kantengraph repräsentiert den Rand von C_{frei}
- Erstmal auf den Rand!!
- **Def. 2.44** Abbildung auf den Rand

Bahnplanung mit Kantengraph

- Bewegung von Orientierung (x_1, y_1, θ_1) zu (x_2, y_2, θ_2) ?
- Kantengraph repräsentiert den Rand von C_{frei}
- Erstmal auf den Rand!!
- **Def. 2.44** Abbildung auf den Rand
- $\Phi : C_{frei} \longrightarrow$ Knoten von E

Bahnplanung mit Kantengraph

- Bewegung von Orientierung (x_1, y_1, θ_1) zu (x_2, y_2, θ_2) ?
- Kantengraph repräsentiert den Rand von C_{frei}
- Erstmal auf den Rand!!
- **Def. 2.44** Abbildung auf den Rand
- $\Phi : C_{frei} \longrightarrow$ Knoten von E
- $(x, y, \theta) \longmapsto$ Knoten $(u, L(u))$

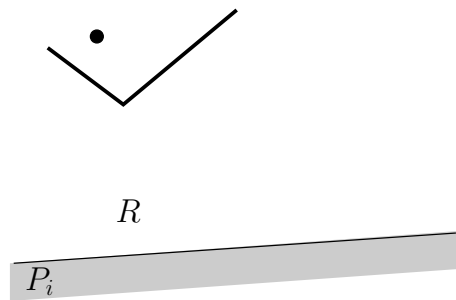
Bahnplanung mit Kantengraph

- Bewegung von Orientierung (x_1, y_1, θ_1) zu (x_2, y_2, θ_2) ?
- Kantengraph repräsentiert den Rand von C_{frei}
- Erstmal auf den Rand!!
- **Def. 2.44** Abbildung auf den Rand
- $\Phi : C_{frei} \longrightarrow$ Knoten von E
- $(x, y, \theta) \longmapsto$ Knoten $(u, L(u))$
- Entspricht Kante e_u von C_{frei} auf dem Rand der Zusammenhangskomponente Z von C_{frei} mit $(x, y, \theta) \in Z$

Bahnplanung mit Kantengraph

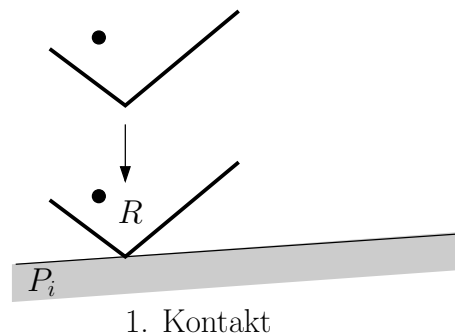
- Bewegung von Orientierung (x_1, y_1, θ_1) zu (x_2, y_2, θ_2) ?
- Kantengraph repräsentiert den Rand von C_{frei}
- Erstmal auf den Rand!!
- **Def. 2.44** Abbildung auf den Rand
- $\Phi : C_{frei} \longrightarrow$ Knoten von E
- $(x, y, \theta) \longmapsto$ Knoten $(u, L(u))$
- Entspricht Kante e_u von C_{frei} auf dem Rand der Zusammenhangskomponente Z von C_{frei} mit $(x, y, \theta) \in Z$
- Konstruktiv!!

Def. 2.44 Abbildung



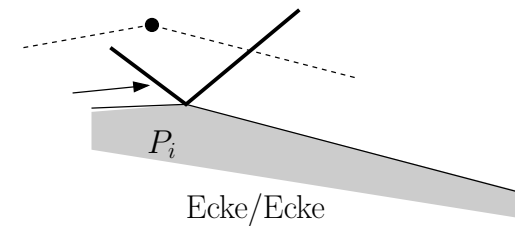
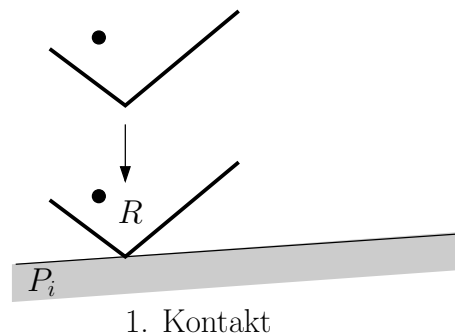
Def. 2.44 Abbildung

- Bewegung von R nach unten: Kontakt I



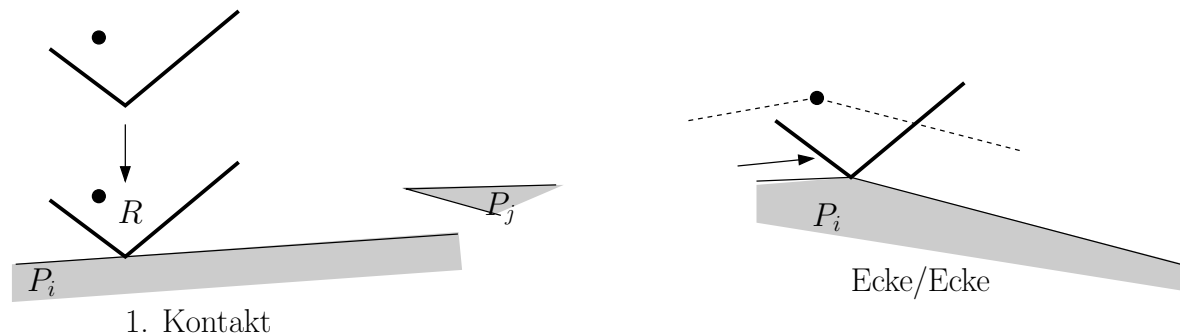
Def. 2.44 Abbildung

- Bewegung von R nach unten: Kontakt I
- Bereits Ecke/Ecke oder Mehrfach-Kontakt: Fertig



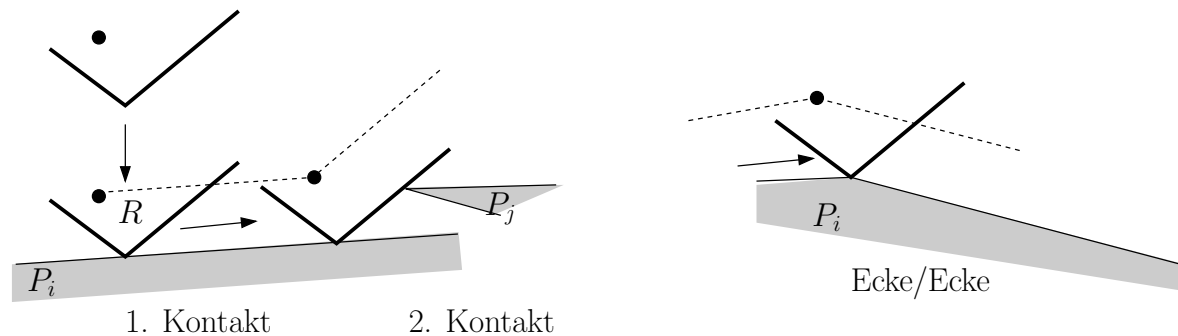
Def. 2.44 Abbildung

- Bewegung von R nach unten: Kontakt I
- Bereits Ecke/Ecke oder Mehrfach-Kontakt: Fertig
- Sonst Bewegung nach rechts, bis Knoten in V^θ



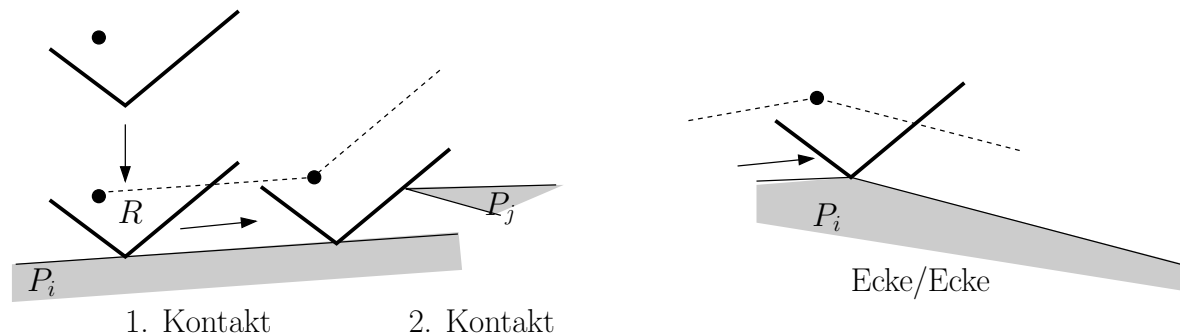
Def. 2.44 Abbildung

- Bewegung von R nach unten: Kontakt I
- Bereits Ecke/Ecke oder Mehrfach-Kontakt: Fertig
- Sonst Bewegung nach rechts, bis Knoten in V^θ



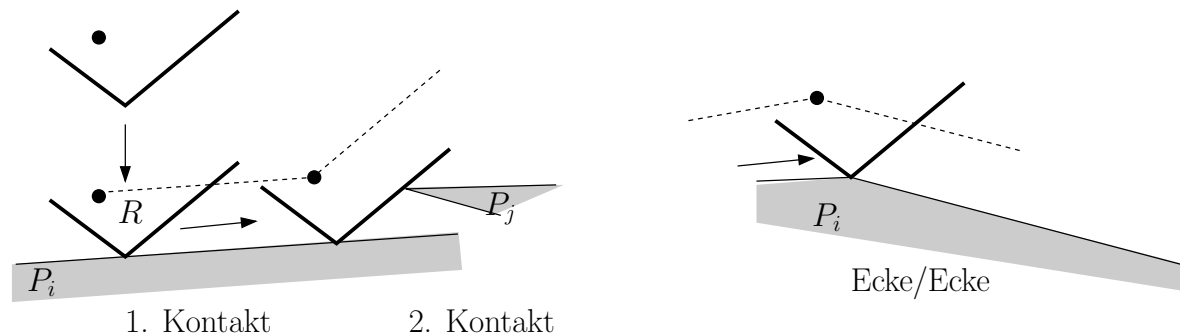
Def. 2.44 Abbildung

- Bewegung von R nach unten: Kontakt I
- Bereits Ecke/Ecke oder Mehrfach-Kontakt: Fertig
- Sonst Bewegung nach rechts, bis Knoten in V^θ
- Erweiterter Knoten $(u, L(u))$ in E



Def. 2.44 Abbildung

- Bewegung von R nach unten: Kontakt I
- Bereits Ecke/Ecke oder Mehrfach-Kontakt: Fertig
- Sonst Bewegung nach rechts, bis Knoten in V^θ
- Erweiterter Knoten $(u, L(u))$ in E
- In $O(nm)$ berechnen



Bahnplanung mit Kantengraph

Bahnplanung mit Kantengraph

- Bewegung von Orientierung $s = (x_s, y_s, \theta_s)$ zu $t = (x_t, y_t, \theta_t)$?

Bahnplanung mit Kantengraph

- Bewegung von Orientierung $s = (x_s, y_s, \theta_s)$ zu $t = (x_t, y_t, \theta_t)$?
- Kollisionsfrei?

Bahnplanung mit Kantengraph

- Bewegung von Orientierung $s = (x_s, y_s, \theta_s)$ zu $t = (x_t, y_t, \theta_t)$?
- Kollisionsfrei?
- Existiert Weg von $\Phi(s)$ zu $\Phi(t)$ in E ?

Bahnplanung mit Kantengraph

- Bewegung von Orientierung $s = (x_s, y_s, \theta_s)$ zu $t = (x_t, y_t, \theta_t)$?
- Kollisionsfrei?
- Existiert Weg von $\Phi(s)$ zu $\Phi(t)$ in E ?
- Korrektheit

Bahnplanung mit Kantengraph

- Bewegung von Orientierung $s = (x_s, y_s, \theta_s)$ zu $t = (x_t, y_t, \theta_t)$?
- Kollisionsfrei?
- Existiert Weg von $\Phi(s)$ zu $\Phi(t)$ in E ?
- Korrektheit

Theorem 2.45:

Bahnplanung mit Kantengraph

- Bewegung von Orientierung $s = (x_s, y_s, \theta_s)$ zu $t = (x_t, y_t, \theta_t)$?
- Kollisionsfrei?
- Existiert Weg von $\Phi(s)$ zu $\Phi(t)$ in E ?
- Korrektheit

Theorem 2.45: s, t zwei freie Plazierungen von R .

Bahnplanung mit Kantengraph

- Bewegung von Orientierung $s = (x_s, y_s, \theta_s)$ zu $t = (x_t, y_t, \theta_t)$?
- Kollisionsfrei?
- Existiert Weg von $\Phi(s)$ zu $\Phi(t)$ in E ?
- Korrektheit

Theorem 2.45: s, t zwei freie Plazierungen von R .

Es gibt eine kollisionsfreie Bewegung von s nach t genau dann, wenn $\Phi(s)$ und $\Phi(t)$ Knoten derselben Zusammenhangskomponente Z des Kantengraphen E sind.

Beweis: **Theorem 2.45**

Beweis: Theorem 2.45

Lemma 2.46:

Beweis: Theorem 2.45

Lemma 2.46:

Zur Zusammenhangskomponente W von C_{frei} gehört eindeutig eine Zusammenhangskomponente Z von E mit:

Beweis: Theorem 2.45

Lemma 2.46:

Zur Zusammenhangskomponente W von C_{frei} gehört eindeutig eine Zusammenhangskomponente Z von E mit:

Knoten von $Z \iff$ Kanten in ∂W .

Beweis: Theorem 2.45

Lemma 2.46:

Zur Zusammenhangskomponente W von C_{frei} gehört eindeutig eine Zusammenhangskomponente Z von E mit:

Knoten von $Z \iff$ Kanten in ∂W .

Zusammenhangskomponenten korrelieren

Beweis: **Theorem 2.45**

Beweis: Theorem 2.45

- Annahme: Lemma 2.46 gilt!

Beweis: Theorem 2.45

- Annahme: Lemma 2.46 gilt!
- $\Phi(s) = \Phi(x_s, y_s, \theta_s) = (u, L(u))$, $\Phi(t) = \Phi(x_t, y_t, \theta_t) = (v, L(v))$

Beweis: Theorem 2.45

- Annahme: Lemma 2.46 gilt!
- $\Phi(s) = \Phi(x_s, y_s, \theta_s) = (u, L(u))$, $\Phi(t) = \Phi(x_t, y_t, \theta_t) = (v, L(v))$
- e_u, e_v Kanten von C_{frei} , die zu Knoten $(u, L(u))$ und $(v, L(v))$ von E gehören

Beweis: Theorem 2.45

- Annahme: Lemma 2.46 gilt!
- $\Phi(s) = \Phi(x_s, y_s, \theta_s) = (u, L(u))$, $\Phi(t) = \Phi(x_t, y_t, \theta_t) = (v, L(v))$
- e_u, e_v Kanten von C_{frei} , die zu Knoten $(u, L(u))$ und $(v, L(v))$ von E gehören
- kollisionsfreie Translation von $s = (x_s, y_s, \theta_s)$ nach u (Knoten $C_{\text{frei}}^{\theta_s}$)

Beweis: Theorem 2.45

- Annahme: Lemma 2.46 gilt!
- $\Phi(s) = \Phi(x_s, y_s, \theta_s) = (u, L(u))$, $\Phi(t) = \Phi(x_t, y_t, \theta_t) = (v, L(v))$
- e_u, e_v Kanten von C_{frei} , die zu Knoten $(u, L(u))$ und $(v, L(v))$ von E gehören
- kollisionsfreie Translation von $s = (x_s, y_s, \theta_s)$ nach u (Knoten $C_{\text{frei}}^{\theta_s}$)
- kollisionsfreie Translationen von $t = (x_t, y_t, \theta_t)$ nach v (Knoten $C_{\text{frei}}^{\theta_t}$)

Beweis: **Theorem 2.45**

Beweis: Theorem 2.45

- \exists kollisionsfreie Bewegung von s nach t

Beweis: Theorem 2.45

- \exists kollisionsfreie Bewegung von s nach t
- $\Leftrightarrow \exists$ kollisionsfreie Bewegung von u nach v

Beweis: Theorem 2.45

- \exists kollisionsfreie Bewegung von s nach t
- $\Leftrightarrow \exists$ kollisionsfreie Bewegung von u nach v
- $\Leftrightarrow \exists$ kollisionsfreie Bewegung von jedem Punkt auf e_u zu jedem Punkt auf e_v

Beweis: Theorem 2.45

- \exists kollisionsfreie Bewegung von s nach t
- $\Leftrightarrow \exists$ kollisionsfreie Bewegung von u nach v
- $\Leftrightarrow \exists$ kollisionsfreie Bewegung von jedem Punkt auf e_u zu jedem Punkt auf e_v (Bewegung entlang der Kanten ist stets möglich)

Beweis: Theorem 2.45

- \exists kollisionsfreie Bewegung von s nach t
- $\Leftrightarrow \exists$ kollisionsfreie Bewegung von u nach v
- $\Leftrightarrow \exists$ kollisionsfreie Bewegung von jedem Punkt auf e_u zu jedem Punkt auf e_v (Bewegung entlang der Kanten ist stets möglich)
- \Leftrightarrow Kanten e_u und e_v auf Rand derselben Zusammenhangskomponente W von C_{frei}

Beweis: Theorem 2.45

- \exists kollisionsfreie Bewegung von s nach t
- $\Leftrightarrow \exists$ kollisionsfreie Bewegung von u nach v
- $\Leftrightarrow \exists$ kollisionsfreie Bewegung von jedem Punkt auf e_u zu jedem Punkt auf e_v (Bewegung entlang der Kanten ist stets möglich)
- \Leftrightarrow Kanten e_u und e_v auf Rand derselben Zusammenhangskomponente W von C_{frei}
- \Leftrightarrow (Lemma 2.46) Knoten $(u, L(u)) = \Phi(s)$ und $(v, L(v)) = \Phi(t)$ in derselben Zusammenhangskomponente Z von E

Bewegungen

Bewegungen

- Theorem 2.45:

Bewegungen

- **Theorem 2.45:** Weg in $E \Leftrightarrow$ kollisionsfreier Weg in C_{frei} existiert

Bewegungen

- **Theorem 2.45:** Weg in $E \Leftrightarrow$ kollisionsfreier Weg in C_{frei} existiert
- Jetzt auch berechnen!

Bewegungen

- **Theorem 2.45:** Weg in $E \Leftrightarrow$ kollisionsfreier Weg in C_{frei} existiert
- Jetzt auch berechnen!
- An den Rand schieben! e_u, e_v

Bewegungen

- **Theorem 2.45:** Weg in $E \Leftrightarrow$ kollisionsfreier Weg in C_{frei} existiert
- Jetzt auch berechnen!
- An den Rand schieben! e_u, e_v
- Breitensuche in E , Weg von $(u, L(u))$ nach $(v, L(v))$

Bewegungen

- **Theorem 2.45:** Weg in $E \Leftrightarrow$ kollisionsfreier Weg in C_{frei} existiert
- Jetzt auch berechnen!
- An den Rand schieben! e_u, e_v
- Breitensuche in E , Weg von $(u, L(u))$ nach $(v, L(v))$
- Existent?

Bewegungen

- **Theorem 2.45:** Weg in $E \Leftrightarrow$ kollisionsfreier Weg in C_{frei} existiert
- Jetzt auch berechnen!
- An den Rand schieben! e_u, e_v
- Breitensuche in E , Weg von $(u, L(u))$ nach $(v, L(v))$
- Existent? Falls nein: Es existiert kein Weg!

Bewegungen

- **Theorem 2.45:** Weg in $E \Leftrightarrow$ kollisionsfreier Weg in C_{frei} existiert
- Jetzt auch berechnen!
- An den Rand schieben! e_u, e_v
- Breitensuche in E , Weg von $(u, L(u))$ nach $(v, L(v))$
- Existent? Falls nein: Es existiert kein Weg!
- Falls ja: Diesen Weg Ablaufen!

Weg in E ablaufen

Weg in E ablaufen

- Weg in E ist Kantenfolge mit sukzessiven Knoten

Weg in E ablaufen

- Weg in E ist Kantenfolge mit sukzessiven Knoten
- $\pi = (\xi_1 = \Phi(s), \xi_2, \xi_3, \dots, \xi_\ell = \Phi(t))$

Weg in E ablaufen

- Weg in E ist Kantenfolge mit sukzessiven Knoten
- $\pi = (\xi_1 = \Phi(s), \xi_2, \xi_3, \dots, \xi_\ell = \Phi(t))$
- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$

Weg in E ablaufen

- Weg in E ist Kantenfolge mit sukzessiven Knoten
- $\pi = (\xi_1 = \Phi(s), \xi_2, \xi_3, \dots, \xi_\ell = \Phi(t))$
- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$
- θ_i in Lebensdauer Kante $[\xi_i, \xi_{i+1}]$

Weg in E ablaufen

- Weg in E ist Kantenfolge mit sukzessiven Knoten
- $\pi = (\xi_1 = \Phi(s), \xi_2, \xi_3, \dots, \xi_\ell = \Phi(t))$
- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$
- θ_i in Lebensdauer Kante $[\xi_i, \xi_{i+1}]$
- Start: An den Rand schieben! $(u_1, L(u_1))$ entspricht Kante e_{u_1} in C_{frei}

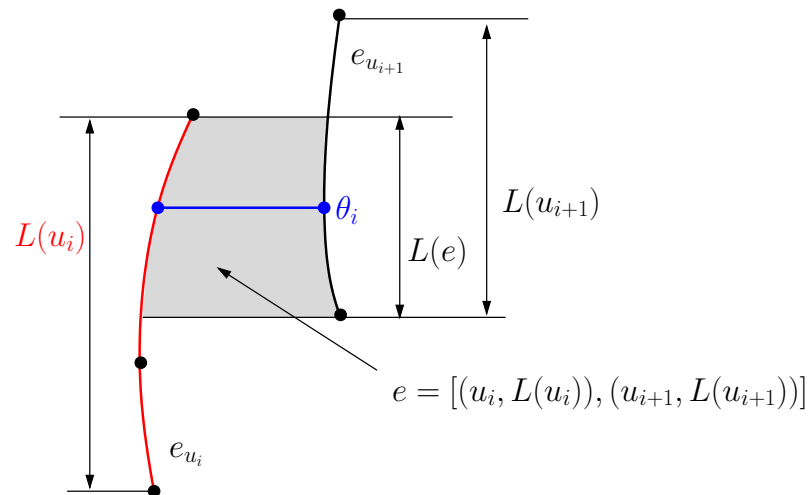
Weg in E ablaufen

- Weg in E ist Kantenfolge mit sukzessiven Knoten
- $\pi = (\xi_1 = \Phi(s), \xi_2, \xi_3, \dots, \xi_\ell = \Phi(t))$
- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$
- θ_i in Lebensdauer Kante $[\xi_i, \xi_{i+1}]$
- Start: An den Rand schieben! $(u_1, L(u_1))$ entspricht Kante e_{u_1} in C_{frei}
- Annahme: wir sind bereits auf Kante e_{u_i} in C_{frei}

Weg in E ablaufen

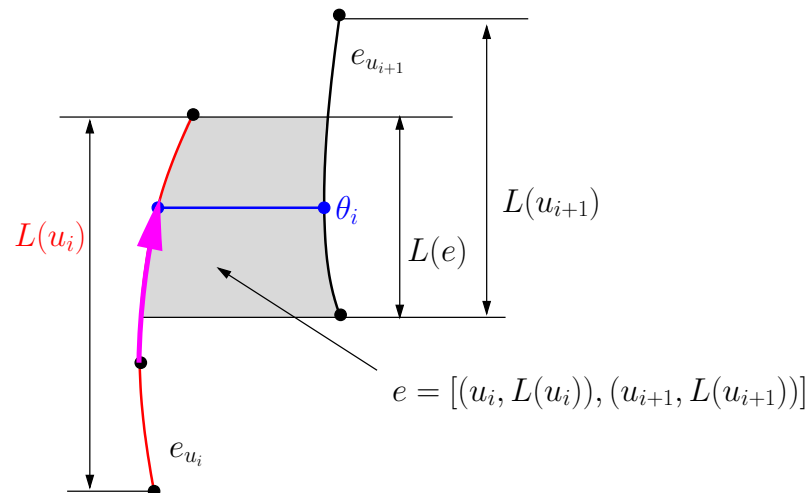
Weg in E ablaufen

- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$
- An Kante e_{u_i} in C_{frei} angekommen
- θ_i in Lebensdauer Kante $[\xi_i, \xi_{i+1}]$



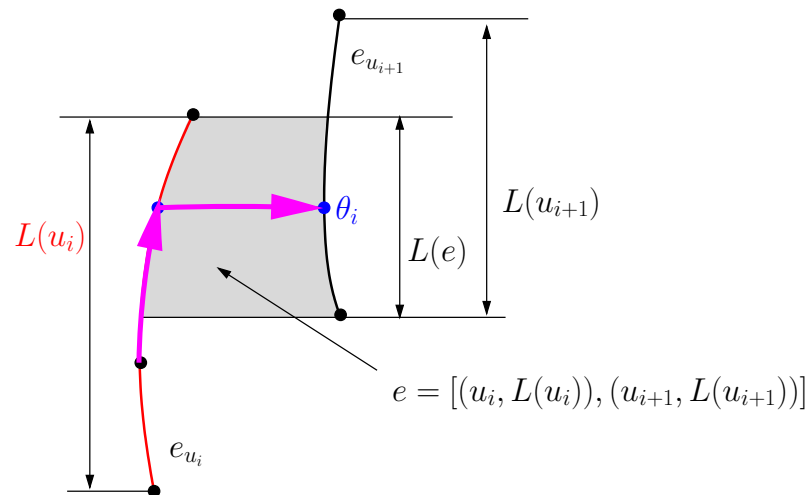
Weg in E ablaufen

- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$
- An Kante e_{u_i} in C_{frei} angekommen
- θ_i in Lebensdauer Kante $[\xi_i, \xi_{i+1}]$
- Rotation entlang e_{u_i} bis θ_i erreicht



Weg in E ablaufen

- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$
- An Kante e_{u_i} in C_{frei} angekommen
- θ_i in Lebensdauer Kante $[\xi_i, \xi_{i+1}]$
- Rotation entlang e_{u_i} bis θ_i erreicht
- Translation von e_{u_i} nach $e_{u_{i+1}}$



Algorithmus 2.13

Algorithmus 2.13

- Macht genau das!

Algorithmus 2.13

- Macht genau das!
- Start: An den Rand schieben! $(u_1, L(u_1))$ entspricht Kante e_{u_1} in C_{frei}

Algorithmus 2.13

- Macht genau das!
- Start: An den Rand schieben! $(u_1, L(u_1))$ entspricht Kante e_{u_1} in C_{frei}
- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$

Algorithmus 2.13

- Macht genau das!
- Start: An den Rand schieben! $(u_1, L(u_1))$ entspricht Kante e_{u_1} in C_{frei}
- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$
- Rotation entlang e_{u_i} bis θ_i erreicht

Algorithmus 2.13

- Macht genau das!
- Start: An den Rand schieben! $(u_1, L(u_1))$ entspricht Kante e_{u_1} in C_{frei}
- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$
- Rotation entlang e_{u_i} bis θ_i erreicht
- Translation von e_{u_i} nach $e_{u_{i+1}}$

Algorithmus 2.13

- Macht genau das!
- Start: An den Rand schieben! $(u_1, L(u_1))$ entspricht Kante e_{u_1} in C_{frei}
- Übergang von $\xi_i = (u_i, L(u_i))$ zu $\xi_{i+1} = (u_{i+1}, L(u_{i+1}))$
- Rotation entlang e_{u_i} bis θ_i erreicht
- Translation von e_{u_i} nach $e_{u_{i+1}}$
- Ende: Vom Rand wegschieben! $(u_l, L(u_l))$ entspricht Kante e_{u_l} in C_{frei}

Theorem 2.47

Theorem 2.47

- Konvexer Roboter R , polygonale Szene P

Theorem 2.47

- Konvexer Roboter R , polygonale Szene P
- Translation und Rotation

Theorem 2.47

- Konvexer Roboter R , polygonale Szene P
- Translation und Rotation
- Zeit: $O(mn \lambda_6(mn) \log(mn))$, Komplexität: $O(mn \lambda_6(mn))$

Theorem 2.47

- Konvexer Roboter R , polygonale Szene P
- Translation und Rotation
- Zeit: $O(mn \lambda_6(mn) \log(mn))$, Komplexität: $O(mn \lambda_6(mn))$
- Kollisionsfreie Bahn von s nach t ?

Theorem 2.47

- Konvexer Roboter R , polygonale Szene P
- Translation und Rotation
- Zeit: $O(mn \lambda_6(mn) \log(mn))$, Komplexität: $O(mn \lambda_6(mn))$
- Kollisionsfreie Bahn von s nach t ?
- Falls existent: Angeben

Theorem 2.47

- Konvexer Roboter R , polygonale Szene P
- Translation und Rotation
- Zeit: $O(mn \lambda_6(mn) \log(mn))$, Komplexität: $O(mn \lambda_6(mn))$
- Kollisionsfreie Bahn von s nach t ?
- Falls existent: Angeben
- Beweis: Gerade geführt!

Bemerkungen

Bemerkungen

- Untere Schranke Komp: $\Omega((mn)^2)$, $O(mn\lambda_6(mn))$; Laufzeit:
 $O(mn\lambda_6(mn) \log(mn))$

Bemerkungen

- Untere Schranke Komp: $\Omega((mn)^2)$, $O(mn\lambda_6(mn))$; Laufzeit:
 $O(mn\lambda_6(mn) \log(mn))$
- Spezialfall Leiter $O(n^2 \log n)$

Bemerkungen

- Untere Schranke Komp: $\Omega((mn)^2)$, $O(mn\lambda_6(mn))$; Laufzeit: $O(mn\lambda_6(mn) \log(mn))$
- Spezialfall Leiter $O(n^2 \log n)$
- Ganz C_{frei} auch in $O(mn\lambda_6(mn) \log(mn))$

Bemerkungen

- Untere Schranke Komp: $\Omega((mn)^2)$, $O(mn\lambda_6(mn))$; Laufzeit: $O(mn\lambda_6(mn) \log(mn))$
- Spezialfall Leiter $O(n^2 \log n)$
- Ganz C_{frei} auch in $O(mn\lambda_6(mn) \log(mn))$
- Nicht konvex: Komp: $\Omega((mn)^3)$; Laufzeit: $O((mn)^3 \log(mn))$

Bemerkungen

- Untere Schranke Komp: $\Omega((mn)^2)$, $O(mn\lambda_6(mn))$; Laufzeit: $O(mn\lambda_6(mn) \log(mn))$
- Spezialfall Leiter $O(n^2 \log n)$
- Ganz C_{frei} auch in $O(mn\lambda_6(mn) \log(mn))$
- Nicht konvex: Komp: $\Omega((mn)^3)$; Laufzeit: $O((mn)^3 \log(mn))$

