

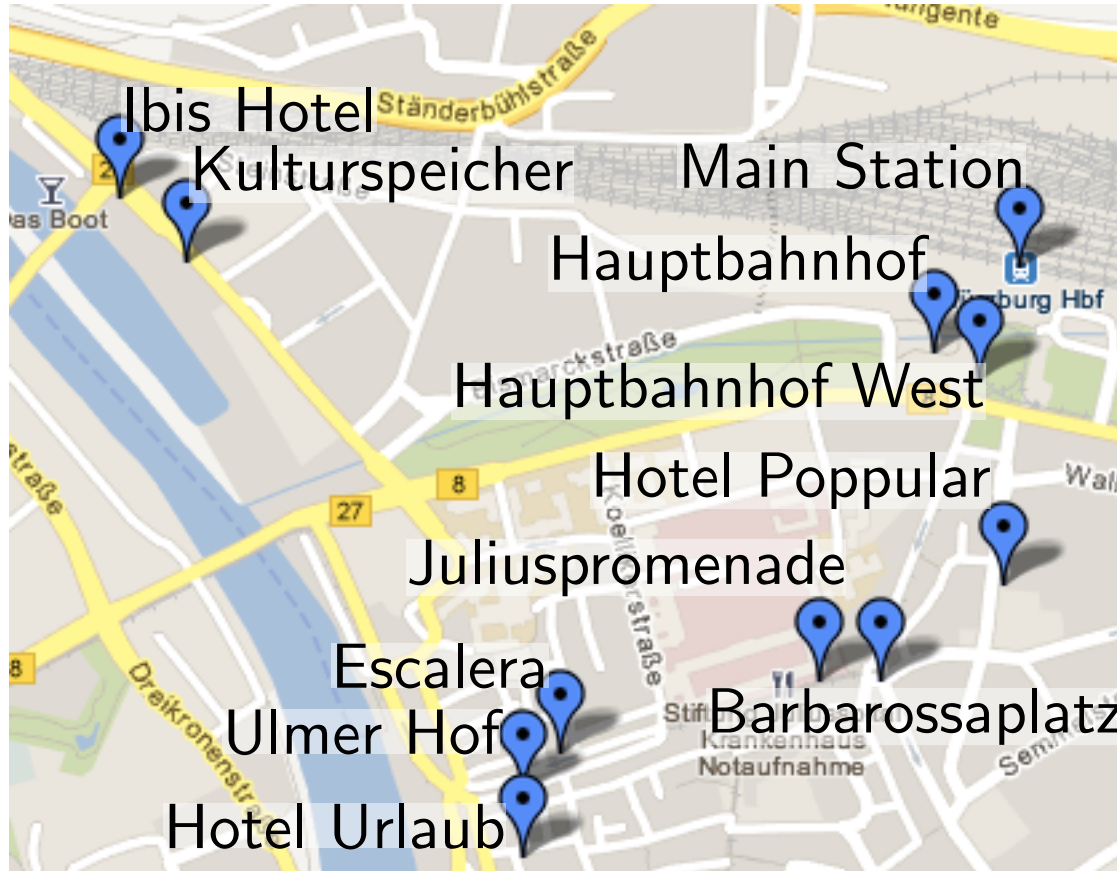
Vorlesung Algorithmische Kartografie

Randbeschriftung

LEHRSTUHL FÜR ALGORITHMIK I · INSTITUT FÜR THEORETISCHE INFORMATIK · FAKULTÄT FÜR INFORMATIK

Benjamin Niedermann/Martin Nöllenburg
16.06.2015

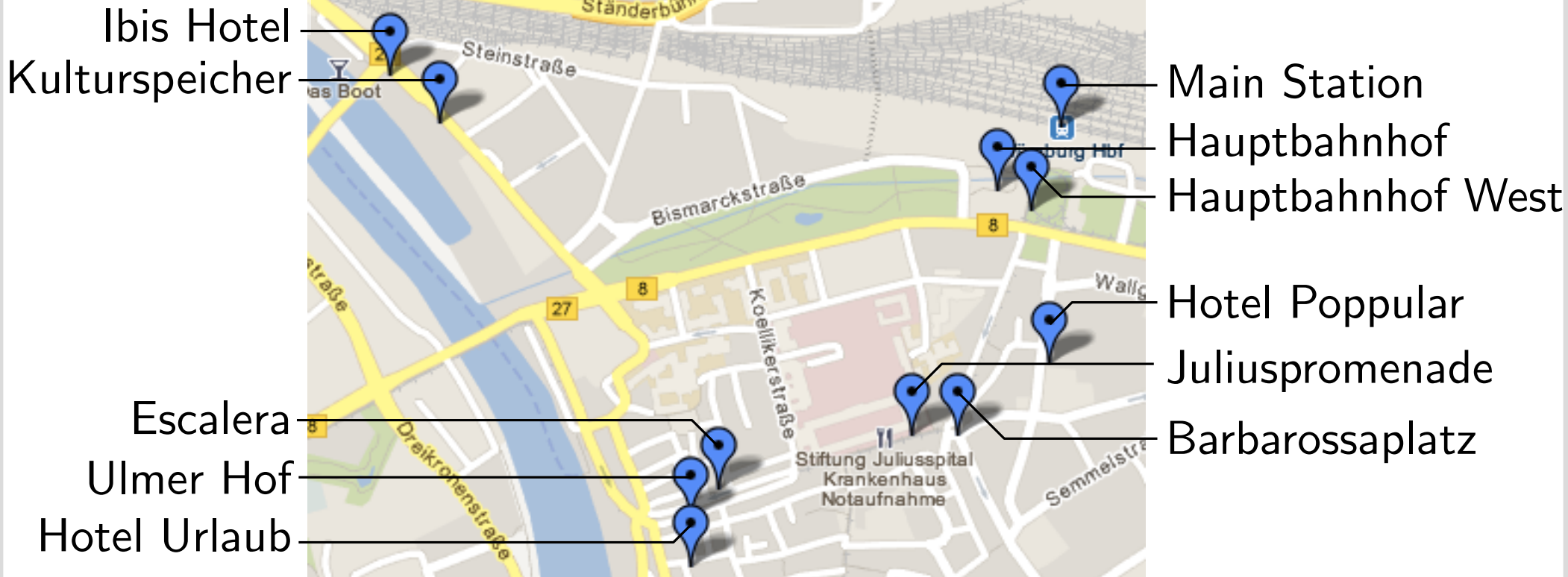




Nachteile von internen Beschriftungen

- funktioniert nur für „dünne“ Punktmengen gut
- Label verdecken unterliegende Topographie
- manche Label müssen ausgeblendet werden
- visuelles *cluttering*

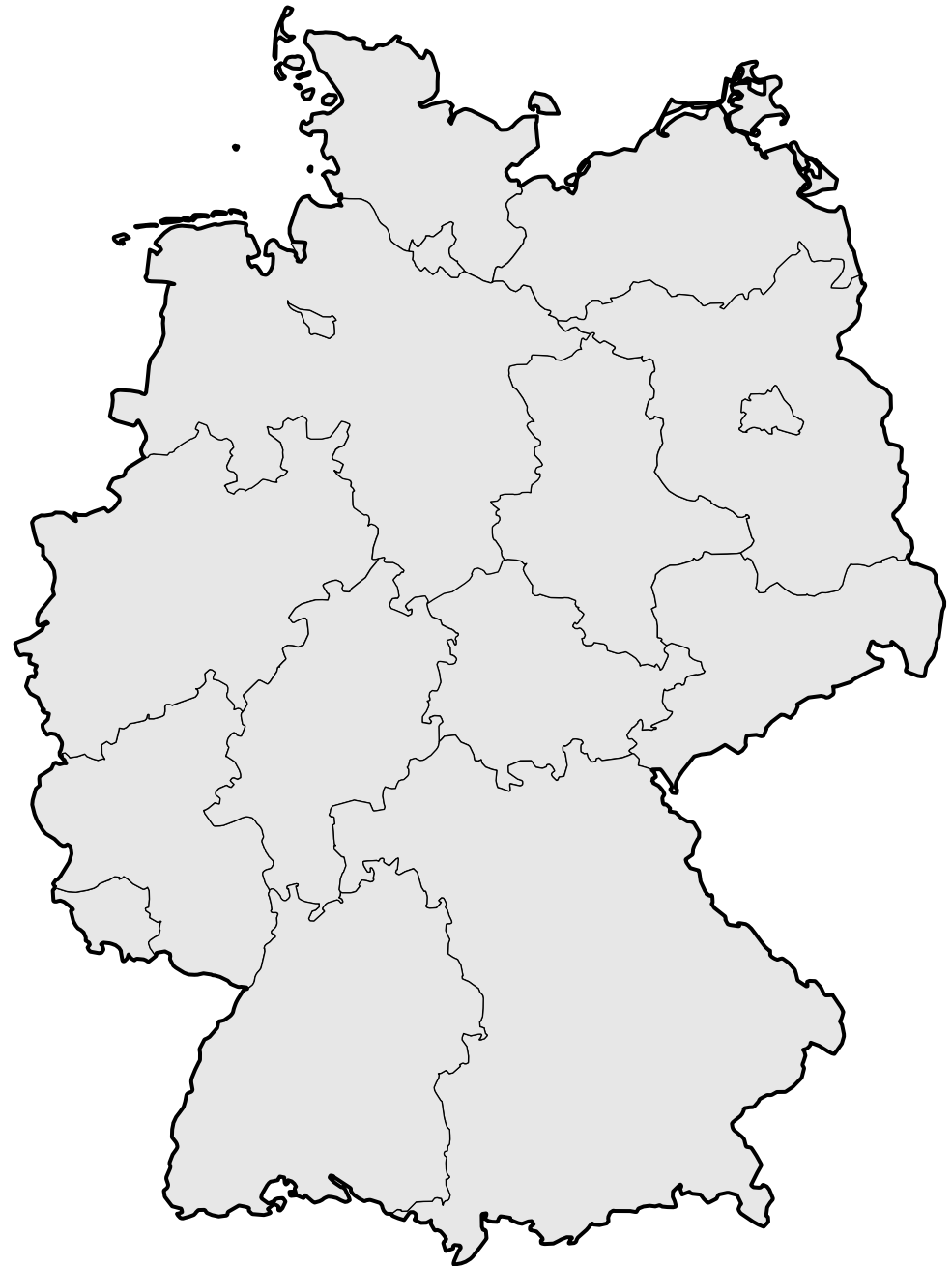
Randbeschriftungen



Vorteile von Randbeschriftungen

- dichte Punktmengen lassen sich beschriften
- weniger Überdeckungen der Karte
- schnelleres Finden Name → Ort
- besser geeignet für Label mit viel Text

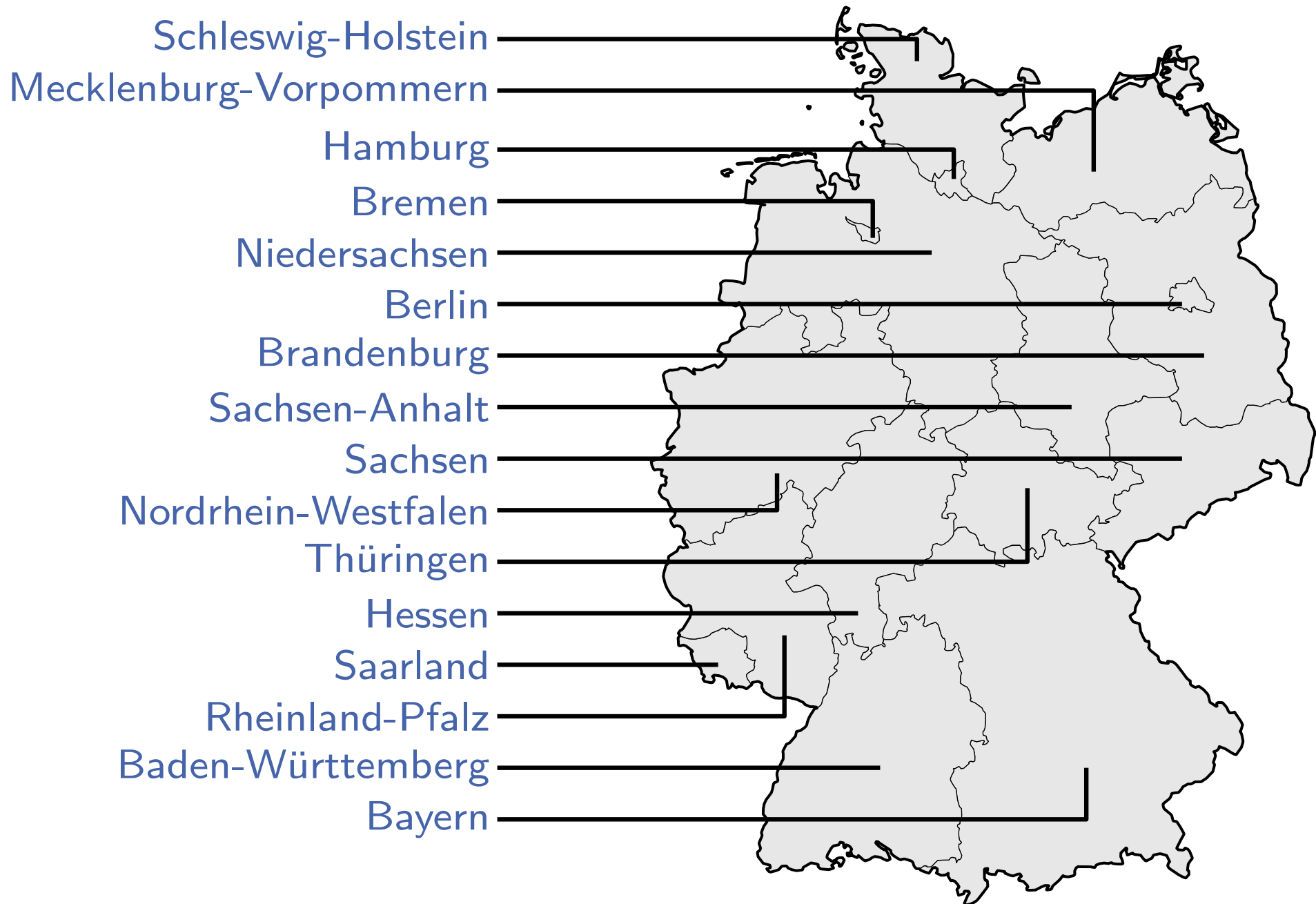
Randbeschriftungen



Randbeschriftungen



Randbeschriftungen



Randbeschriftungen

Schleswig-Holstein

Mecklenburg-

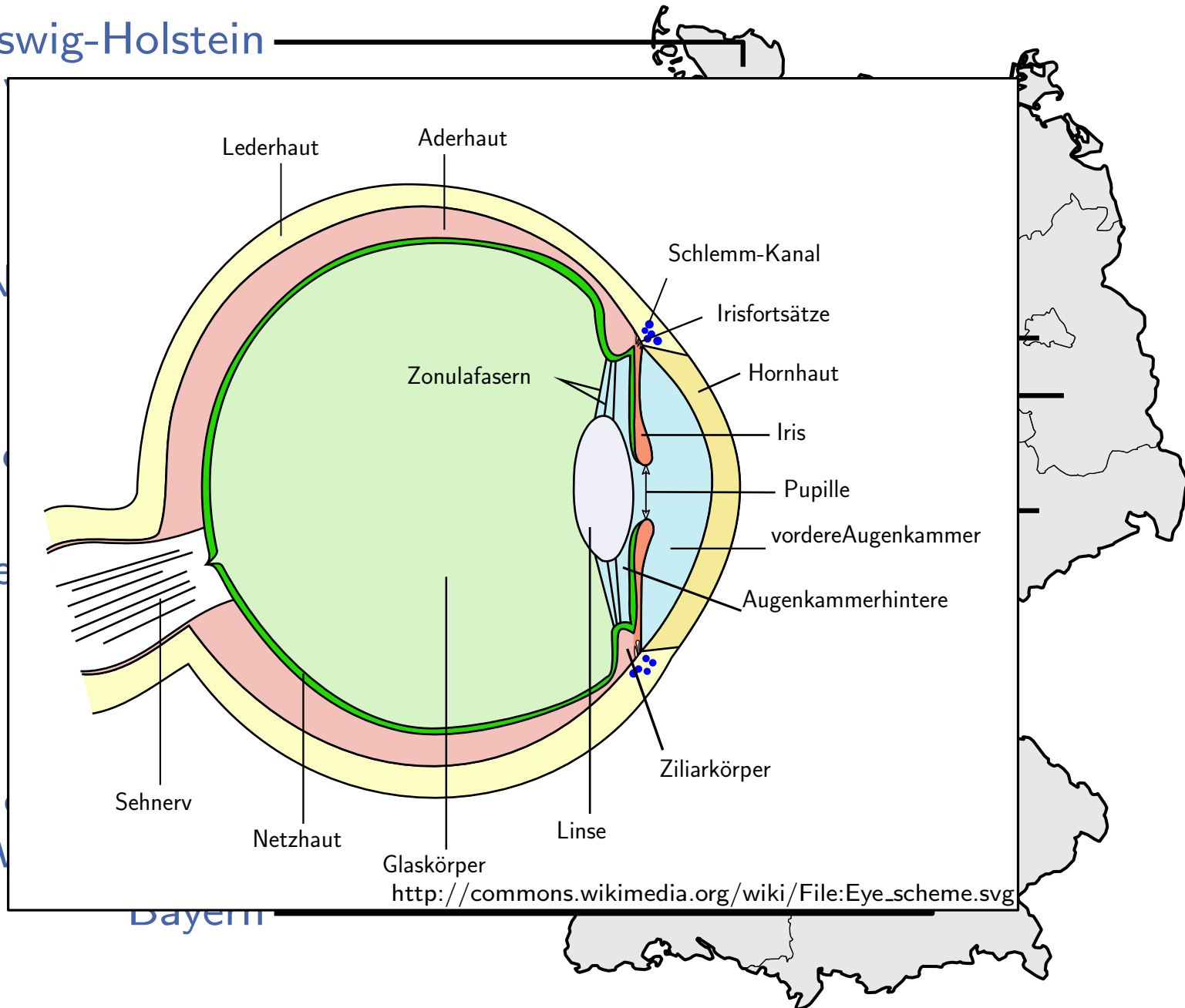
Niederrhein

Sachsen

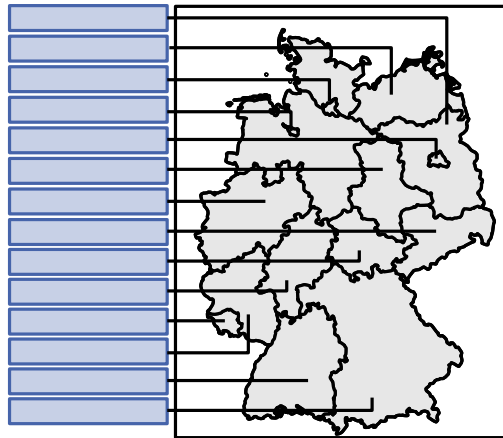
Nordrhein-

Rheinland-Pfalz
Baden-Württemberg

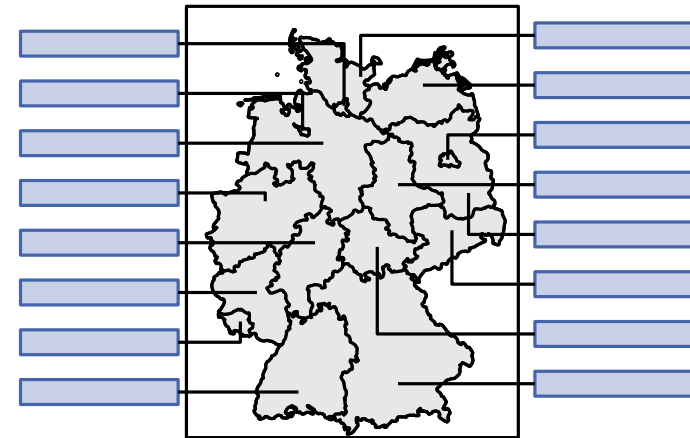
Bayern



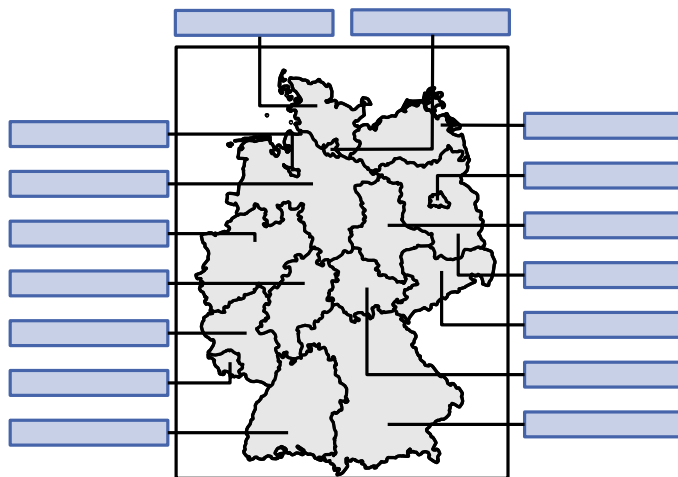
Arten von Randbeschriftungen



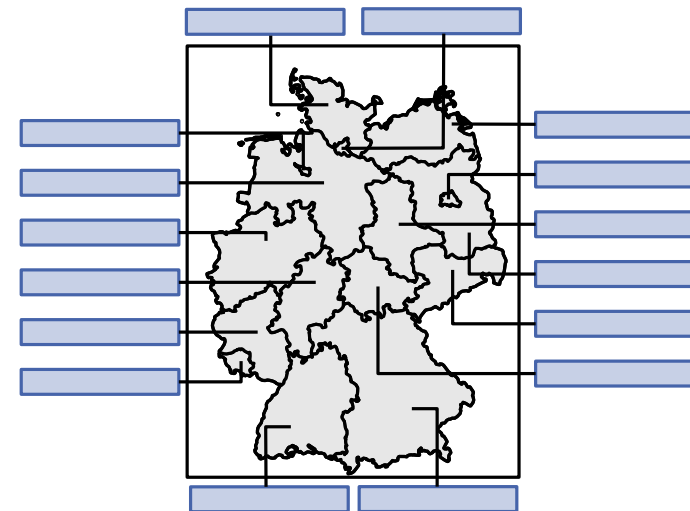
Einseitiger Fall



Zweiseitiger Fall

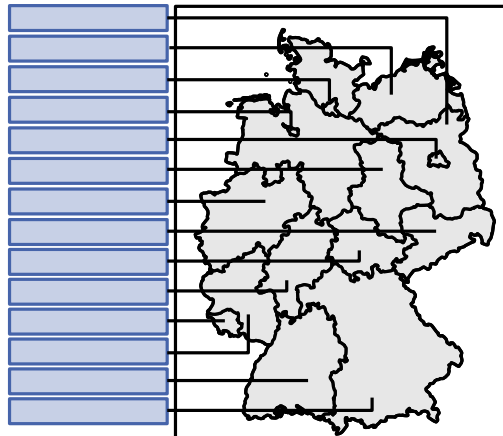


Dreiseitiger Fall

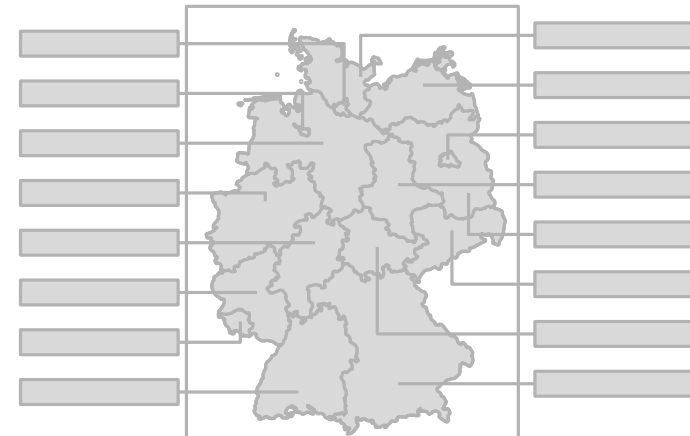


Vierseitiger Fall

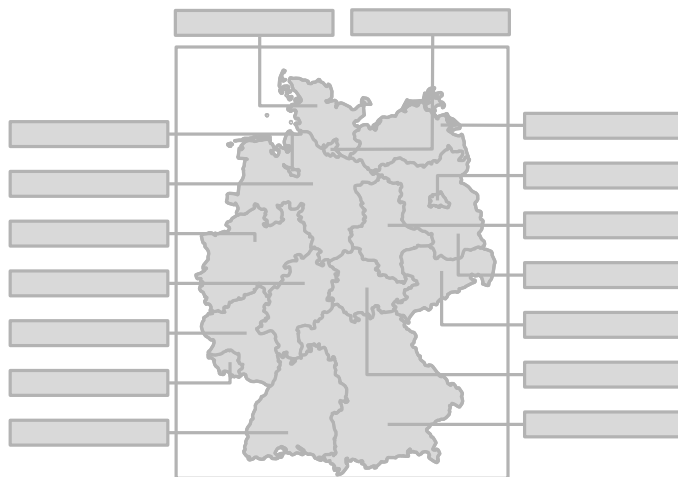
Arten von Randbeschriftungen



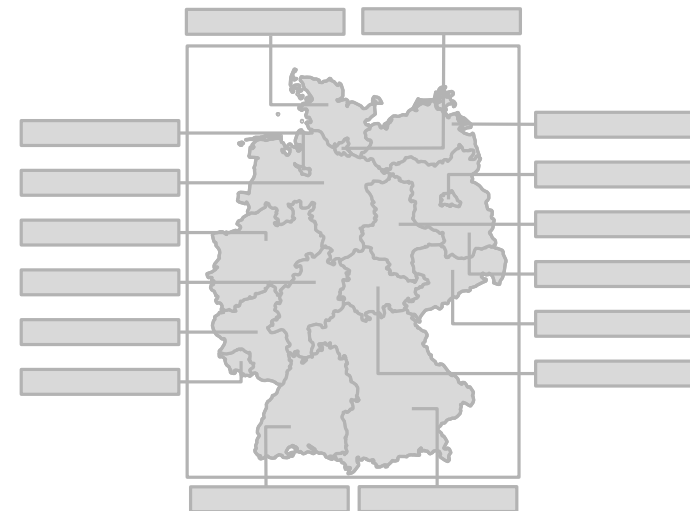
Einseitiger Fall



Zweiseitiger Fall



Dreiseitiger Fall

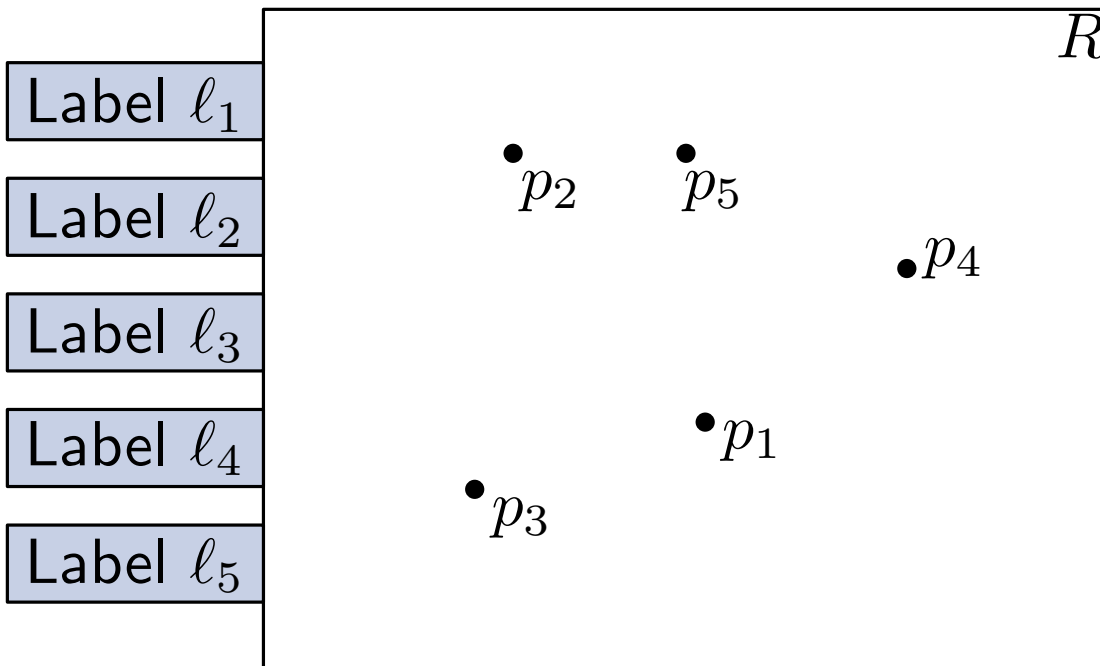


Vierseitiger Fall

Problemstellung

Gegeben:

- Punkte $P = \{p_1, \dots, p_n\}$ in einem Rechteck R .
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{l_1, \dots, l_n\}$ mit rechter Seite auf linker Seite von R . Rechtecke aus L heißen *Label*.



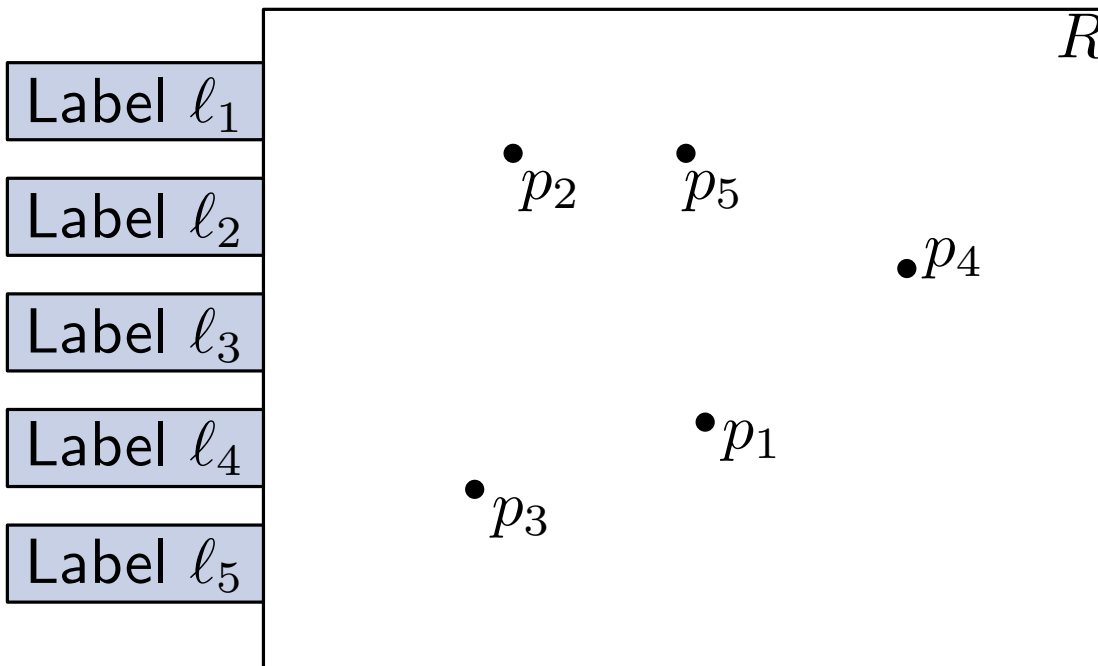
Problemstellung

Gegeben:

- Punkte $P = \{p_1, \dots, p_n\}$ in einem Rechteck R .
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \dots, \ell_n\}$ mit rechter Seite auf linker Seite von R . Rechtecke aus L heißen *Label*.

Annahme:

- Jedes Label ist groß genug, sodass die Beschreibung eines jeden Punkts hinein passt.
- Es sind genauso viele Label wie Punkte vorhanden.

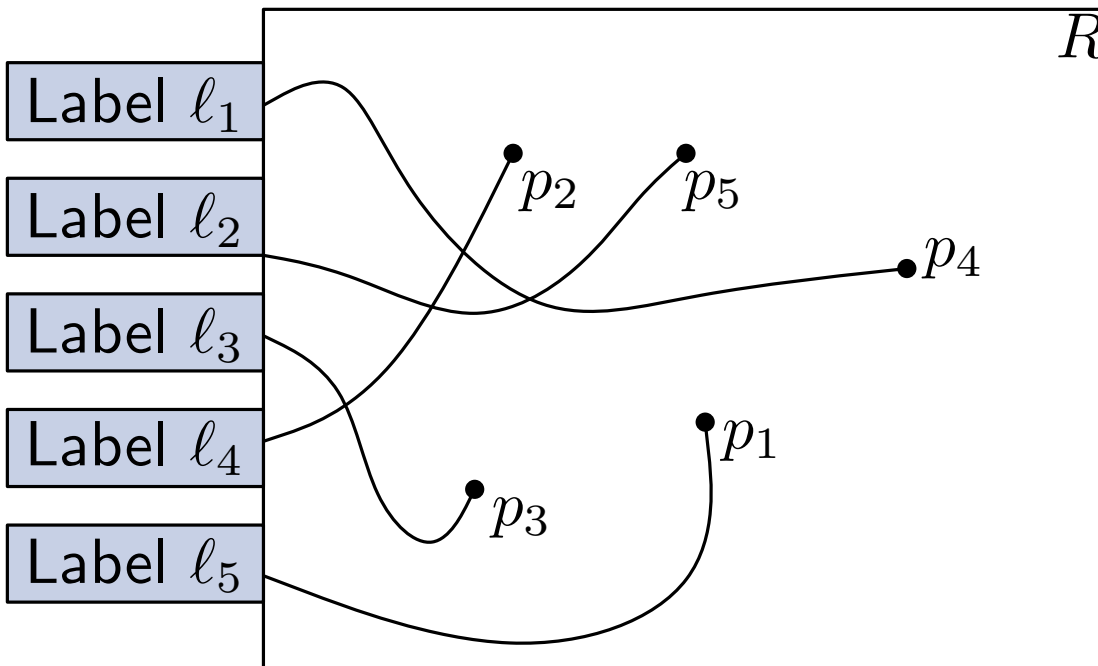


Gegeben:

- Punkte $P = \{p_1, \dots, p_n\}$ in einem Rechteck R .
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \dots, \ell_n\}$ mit rechter Seite auf linker Seite von R . Rechtecke aus L heißen *Label*.

Definition: Eine Menge $\mathcal{L} = \{\lambda_1, \dots, \lambda_n\}$ an Kurven in R heißt *Zuordnung* von L und P , wenn

- jede Kurve einen Punkt $p_i \in P$ mit einem Label $\ell_j \in L$ verbindet und
- jeder Punkt und jedes Label genau von einer Kurve angebunden wird.

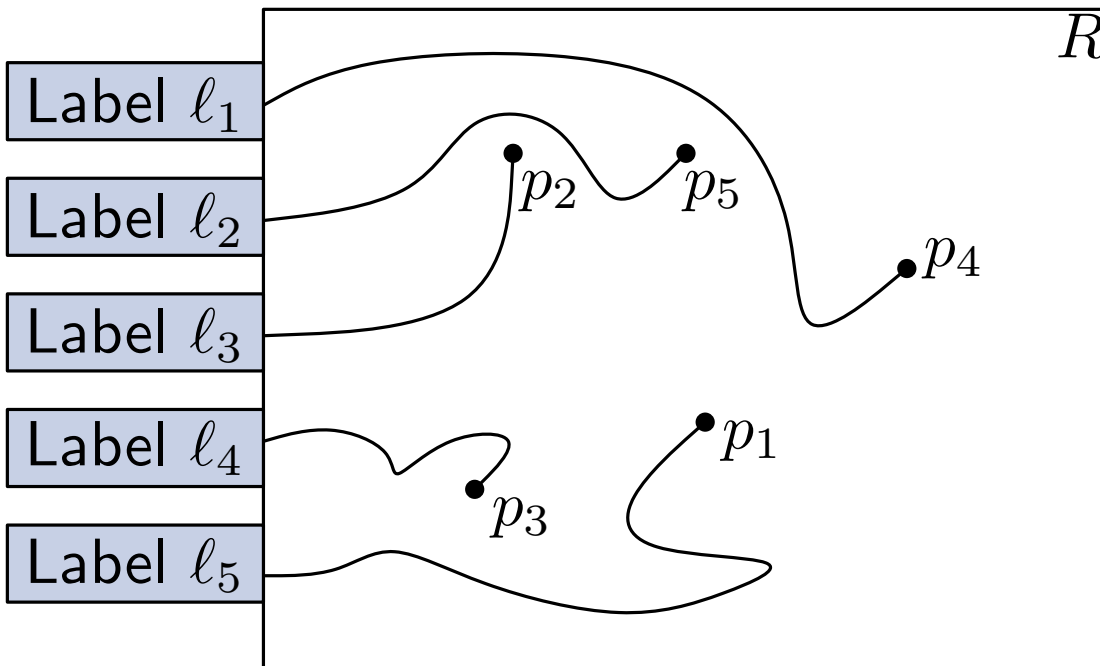


Problemstellung

Gegeben:

- Punkte $P = \{p_1, \dots, p_n\}$ in einem Rechteck R .
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \dots, \ell_n\}$ mit rechter Seite auf linker Seite von R . Rechtecke aus L heißen *Label*.

Gesucht: *Kreuzungsfreie* Zuordnung \mathcal{L} von L und P , d.h. die Kurven in \mathcal{L} schneiden sich nicht gegenseitig.

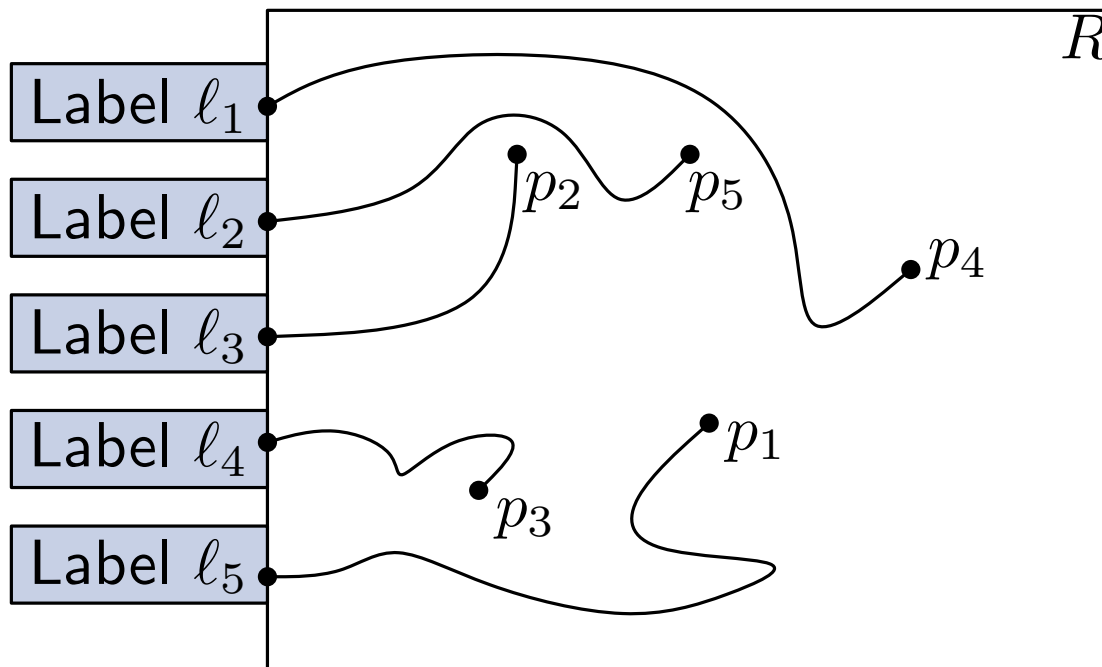


Problemstellung

Gegeben:

- Punkte $P = \{p_1, \dots, p_n\}$ in einem Rechteck R .
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \dots, \ell_n\}$ mit rechter Seite auf linker Seite von R . Rechtecke aus L heißen *Label*.

Gesucht: *Kreuzungsfreie* Zuordnung \mathcal{L} von L und P , d.h. die Kurven in \mathcal{L} schneiden sich nicht gegenseitig.



Begriffe:

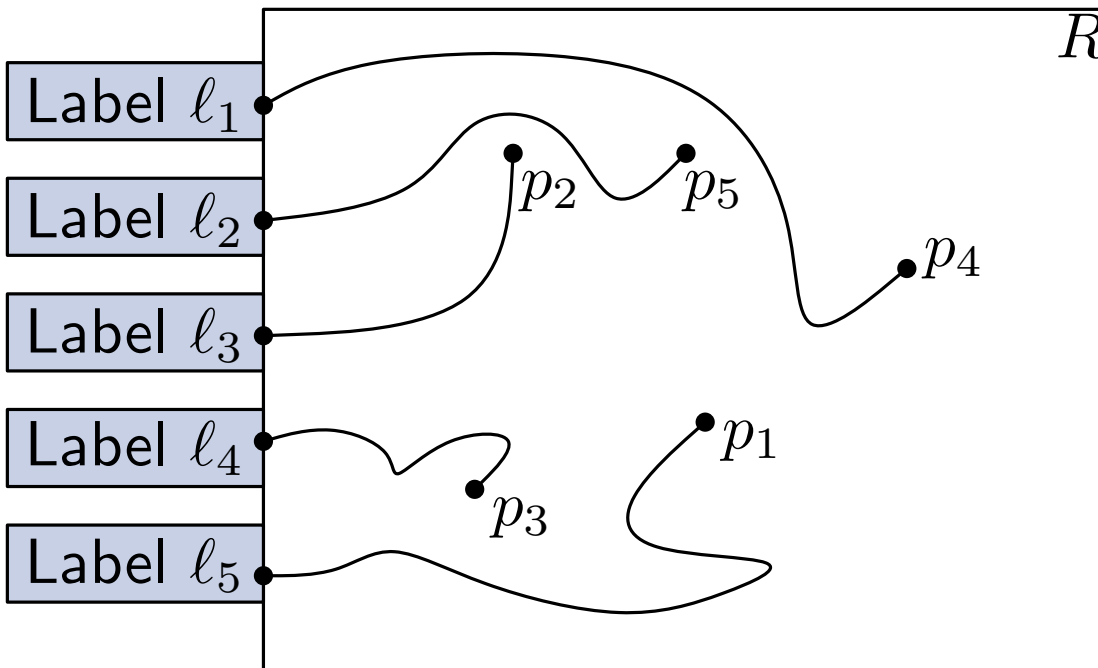
- Kurve λ_i heißt *Leader*.
- Verbindungspunkt von Leader und Label heißt *Port*.

Problemstellung

Gegeben:

- Punkte $P = \{p_1, \dots, p_n\}$ in einem Rechteck R .
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \dots, \ell_n\}$ mit rechter Seite auf linker Seite von R . Rechtecke aus L heißen *Label*.

Gesucht: *Kreuzungsfreie* Zuordnung \mathcal{L} von L und P , d.h. die Kurven in \mathcal{L} schneiden sich nicht gegenseitig.



Begriffe:

- Kurve λ_i heißt *Leader*.
- Verbindungspunkt von Leader und Label heißt *Port*.

Zwei Varianten:

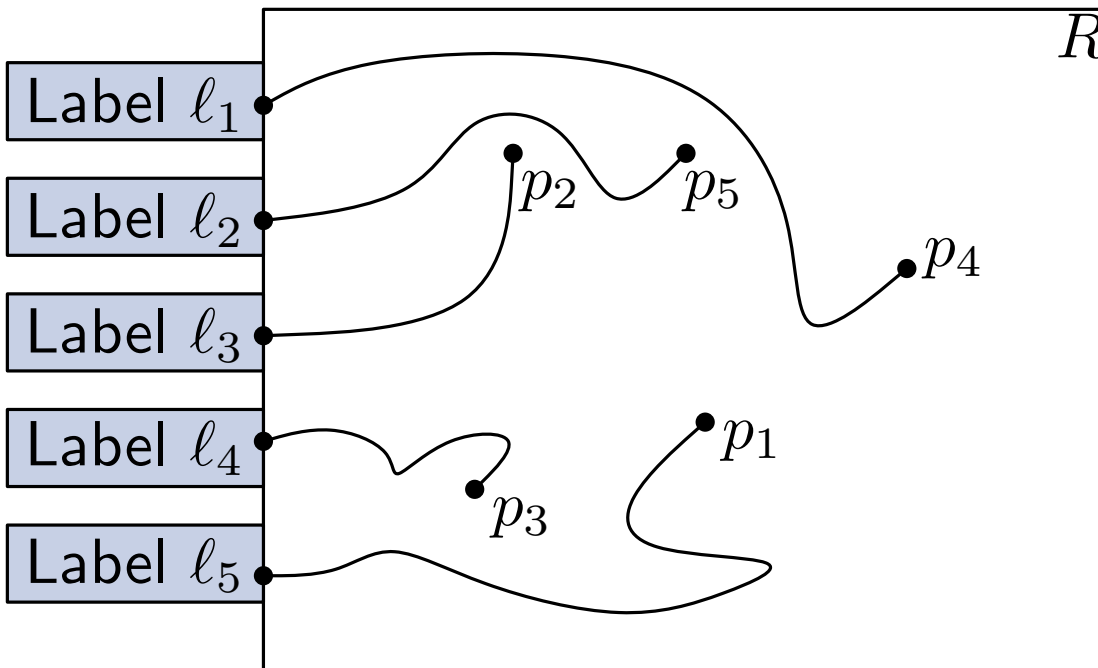
1. Ports fest vorgegeben.
2. Port auf rechter Seite von Label frei wählbar.

Problemstellung

Gegeben:

- Punkte $P = \{p_1, \dots, p_n\}$ in einem Rechteck R .
- Uniforme, achsenparallele, disjunkte Rechtecke $L = \{\ell_1, \dots, \ell_n\}$ mit rechter Seite auf linker Seite von R . Rechtecke aus L heißen *Label*.

Gesucht: *Kreuzungsfreie* Zuordnung \mathcal{L} von L und P , d.h. die Kurven in \mathcal{L} schneiden sich nicht gegenseitig.



Begriffe:

- Kurve λ_i heißt *Leader*.
- Verbindungspunkt von Leader und Label heißt *Port*.

Zwei Varianten:

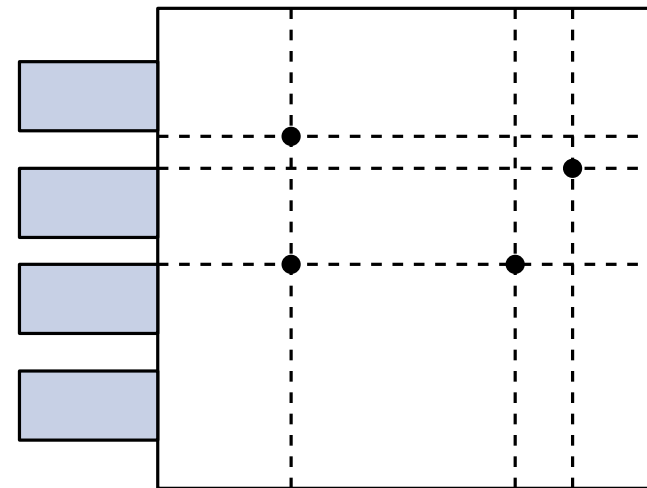
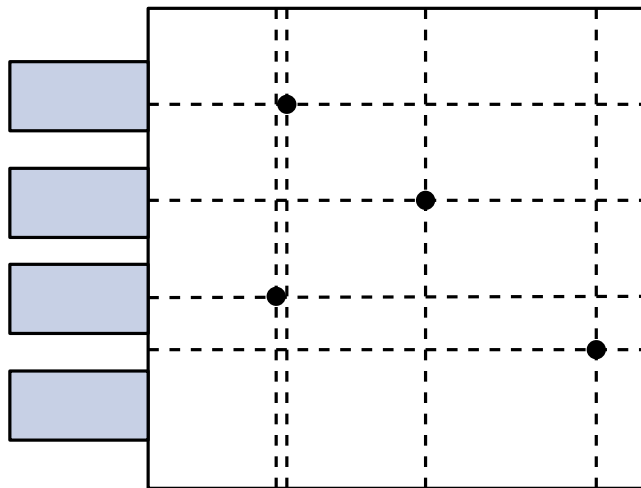
1. Ports fest vorgegeben.
2. Port auf rechter Seite von Label frei wählbar.

Gegeben:

- Punkte $P = \{p_1, \dots, p_n\}$ in einem Rechteck D

Annahme: Allgemeine Lage der Punkte in P und der Label in L .

- Keine zwei Punkte in P liegen auf einer gemeinsamen vertikalen oder horizontalen Geraden.
- Kein Punkt in P liegt auf einer horizontalen Geraden, welche die obere oder untere Seite eines Labels in L verlängert.



La

La

La

La

La

Port.

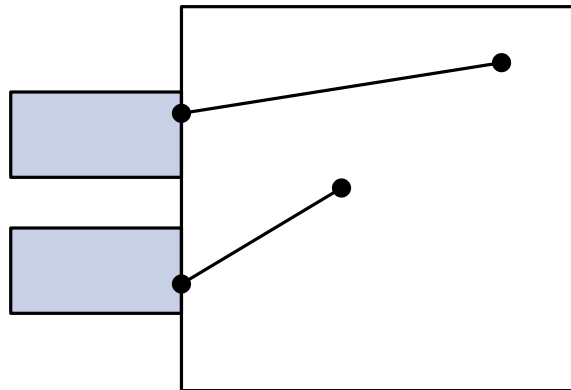
on

Label nicht wählbar.

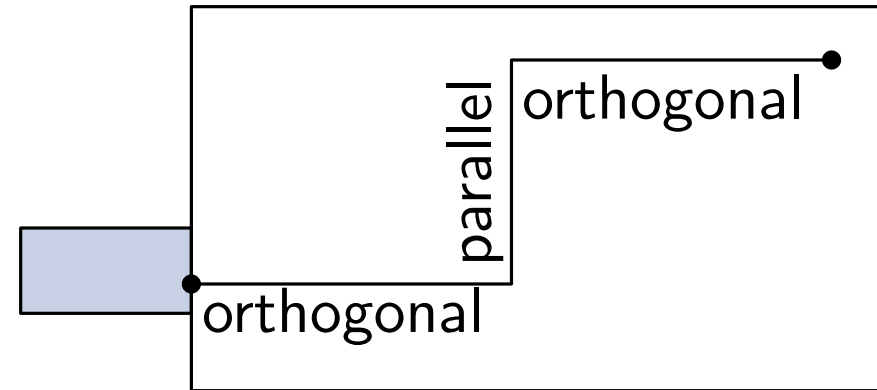
Typen von Leadern

Üblich: Leader werden aus Sequenz von Strecken zusammengesetzt.

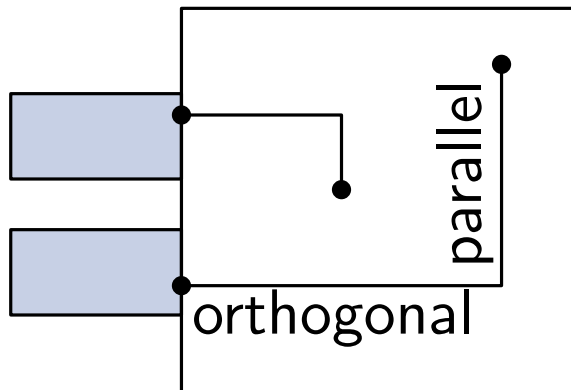
Typische Beispiele:



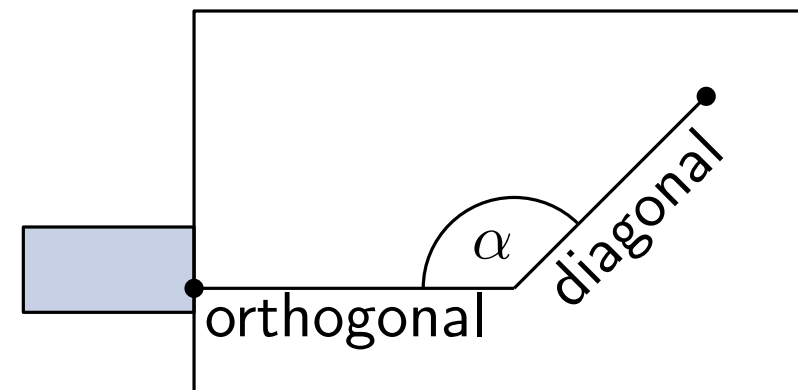
s-Leader



opo-Leader



po-Leader

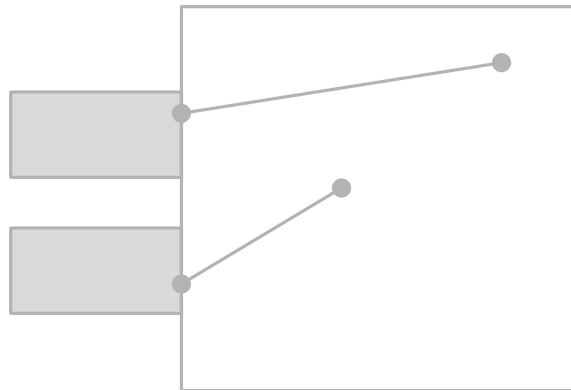


do-Leader

Typen von Leadern

Üblich: Leader werden aus Sequenz von Strecken zusammengesetzt.

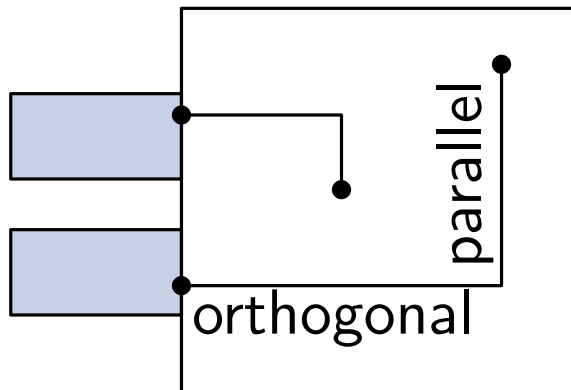
Typische Beispiele:



s-Leader



opo-Leader



po-Leader



do-Leader

Bewertungsfunktion:

$$\text{bad} : L \times P \rightarrow \mathbb{R}$$

Bewertet Zuordnung von Labeln L und Punkten P .

Optimierungsproblem:

Finde für gegebene Instanz (R, P, L) eine kreuzungsfreie Zuordnung \mathcal{L} , sodass

$$\sum_{\lambda \in \mathcal{L}} \text{bad}(\ell(\lambda), p(\lambda))$$

unter allen kreuzungsfreien Zuordnungen minimiert wird.

Notation:

- $\ell(\lambda)$: Label aus L , das von Leader λ angebunden wird.
- $p(\lambda)$: Punkt aus p , der von Leader λ angebunden wird.

Bewertungsfunktion:

$$\text{bad} : L \times P \rightarrow \mathbb{R}$$

Bewertet Zuordnung von Labeln L und Punkten P .

Optimierungsproblem:

Finde für gegebene Instanz (R, P, L) eine kreuzungsfreie Zuordnung \mathcal{L} , sodass

$$\sum_{\lambda \in \mathcal{L}} \text{bad}(\ell(\lambda), p(\lambda))$$

unter allen kreuzungsfreien Zuordnungen minimiert wird.

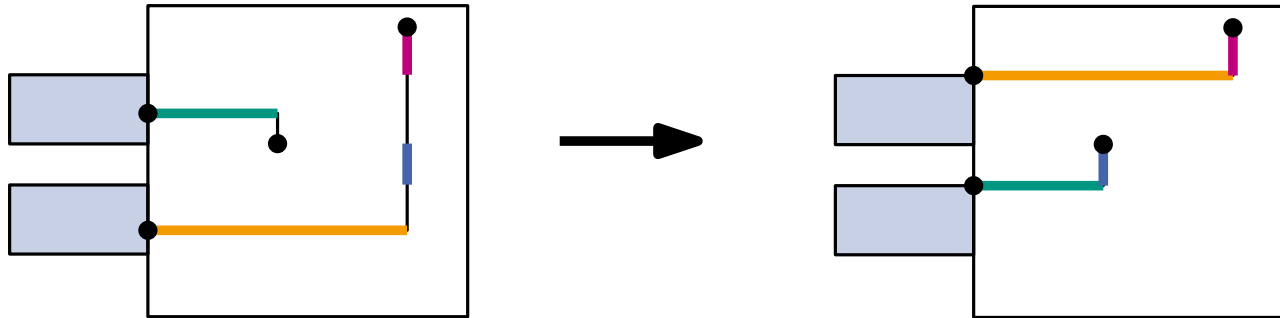
Notation:

- $\ell(\lambda)$: Label aus L , das von Leader λ angebunden wird.
- $p(\lambda)$: Punkt aus p , der von Leader λ angebunden wird.

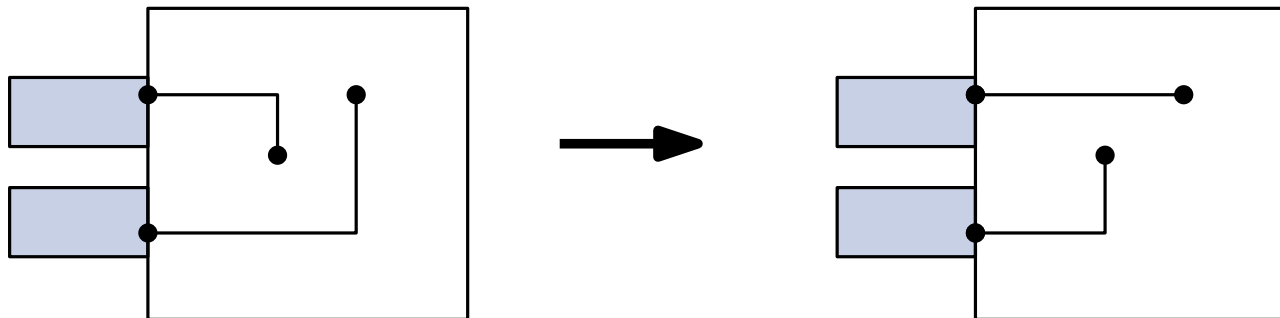
Beispiele für Bewertungsfunktionen?

Beispiele für Bewertungsfunktionen

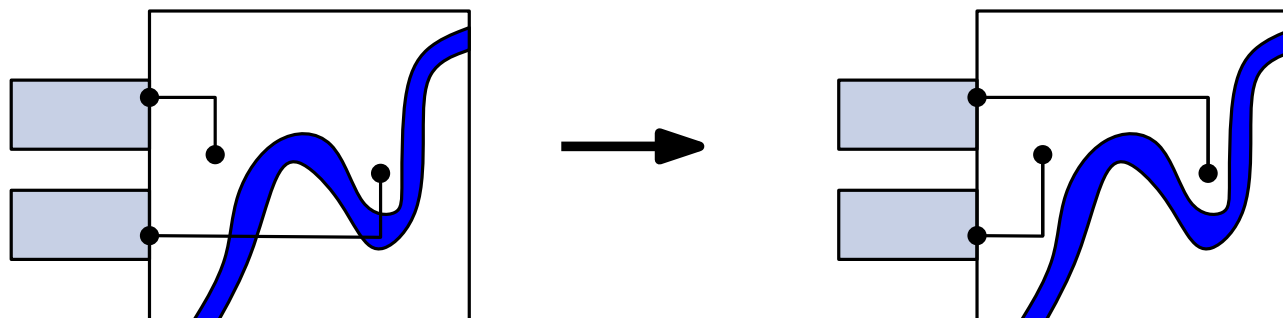
Längenminimierung:



Knickminimierung:

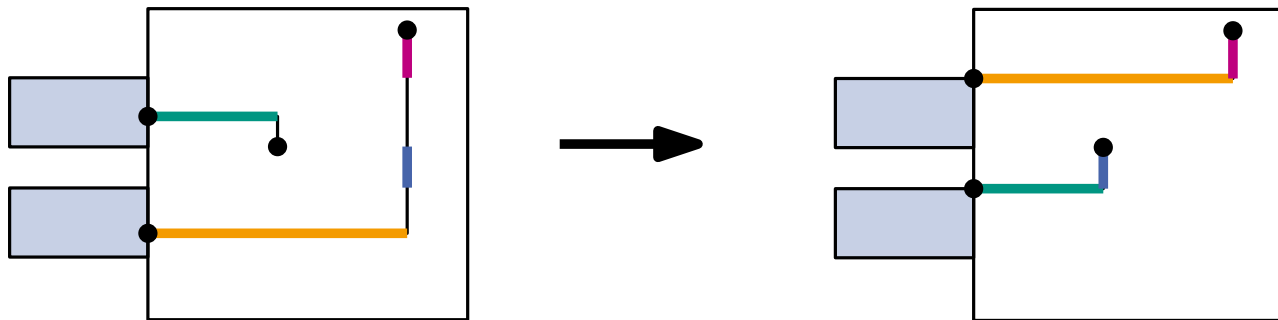


Interferenzminimierung mit Karte

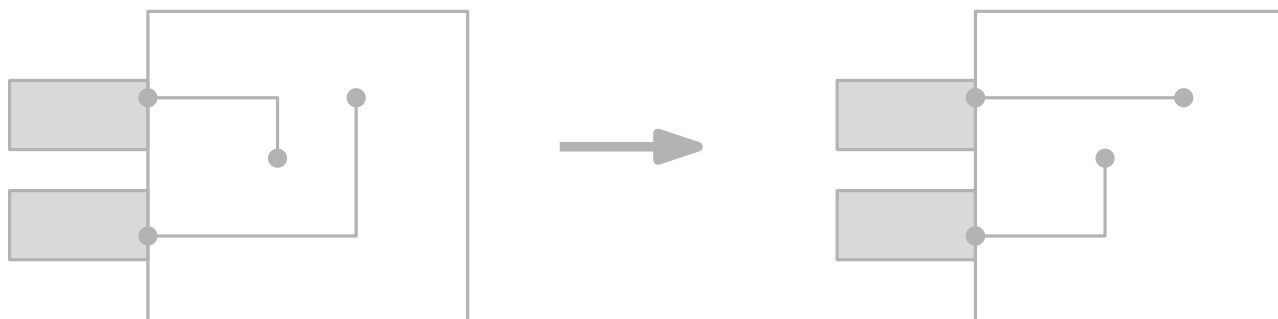


Beispiele für Bewertungsfunktionen

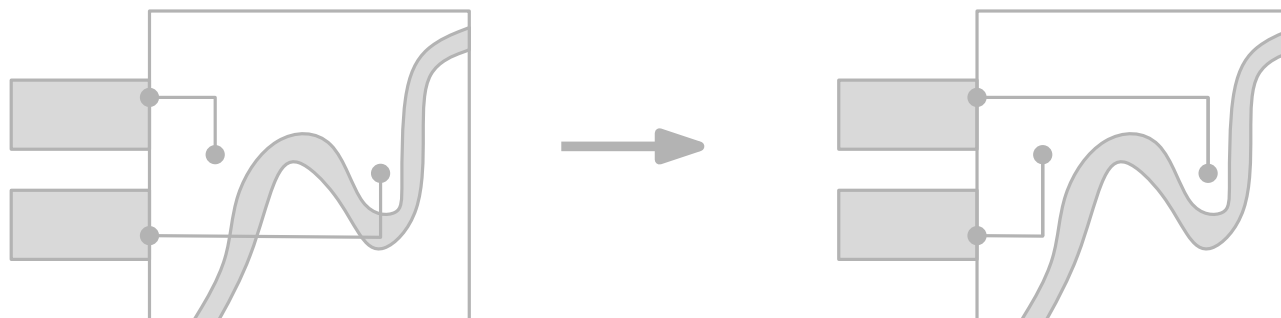
Längenminimierung:



Knickminimierung:

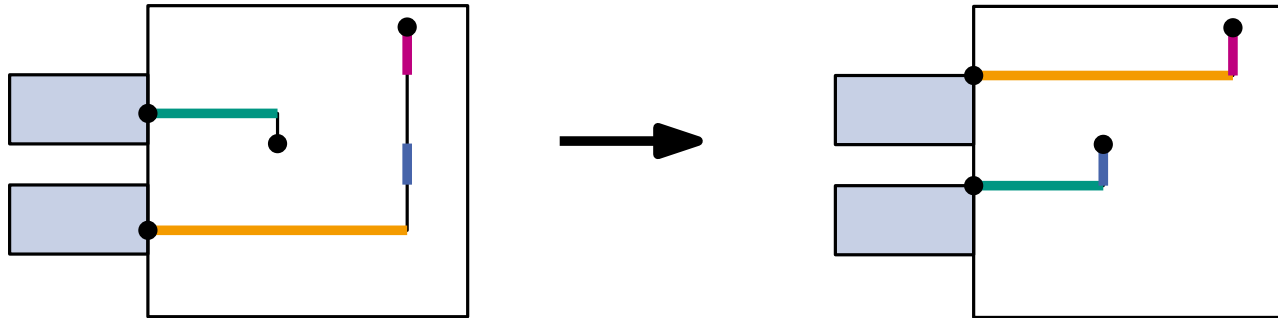


Interferenzminimierung mit Karte



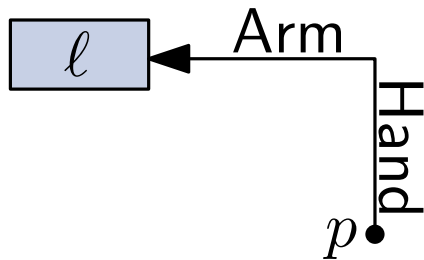
Beispiele für Bewertungsfunktionen

Längenminimierung:



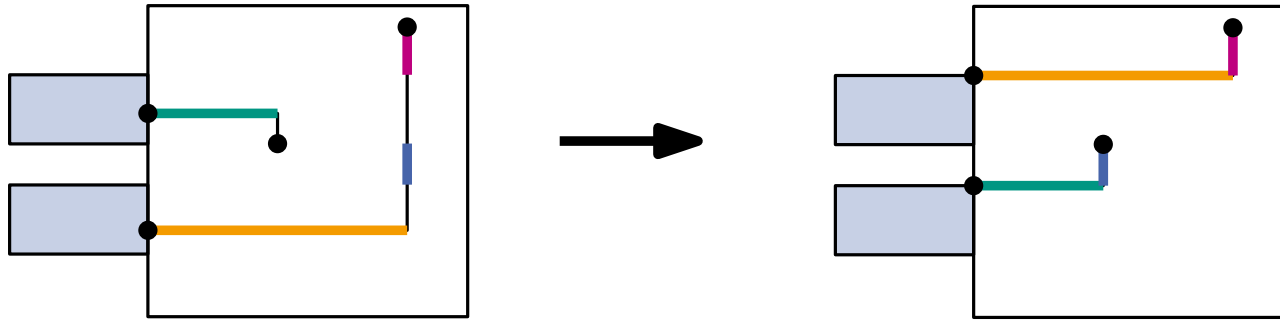
Begriffe:

- Länge einer Zuordnung \mathcal{L} : Die Summe der Leader-Längen.
- $\lambda(p, \ell)$ bezeichnet po-Leader der von p nach ℓ führt.



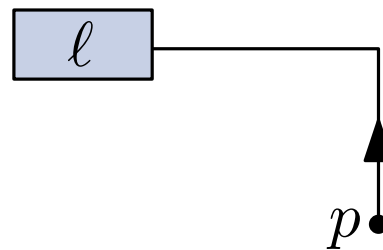
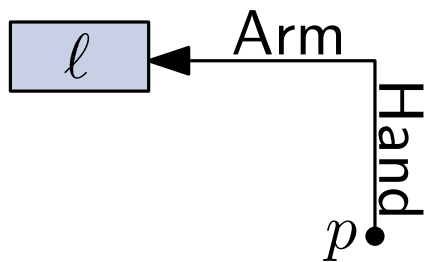
Beispiele für Bewertungsfunktionen

Längenminimierung:

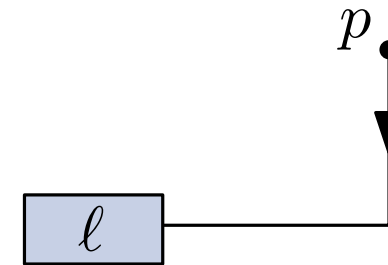


Begriffe:

- Länge einer Zuordnung \mathcal{L} : Die Summe der Leader-Längen.
- $\lambda(p, \ell)$ bezeichnet po-Leader der von p nach ℓ führt.



aufsteigender Leader



absteigender Leader



direkter Leader

Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in \mathcal{L}^* schneiden sich nicht.

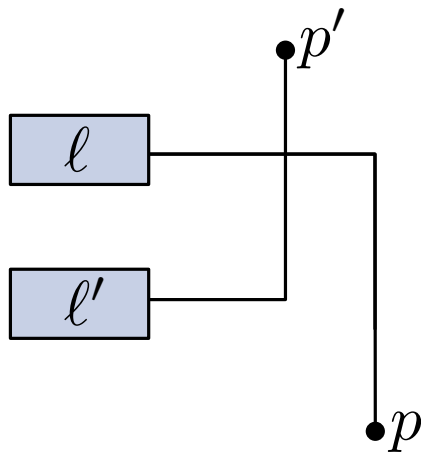
Neuverdrahtung

Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in \mathcal{L}^* schneiden sich nicht.

Annahme es gäbe solche Leader:



Neuverdrahtung

Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in \mathcal{L}^* schneiden sich nicht.

Annahme es gäbe solche Leader:



Kürzere Gesamtlänge der Leader nach Neuverdrahtung

⚡ Minimalität von \mathcal{L}^*

Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

- A) Absteigende und aufsteigende Leader in \mathcal{L}^* schneiden sich nicht.
- B) Kein direkter Leader in \mathcal{L}^* kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

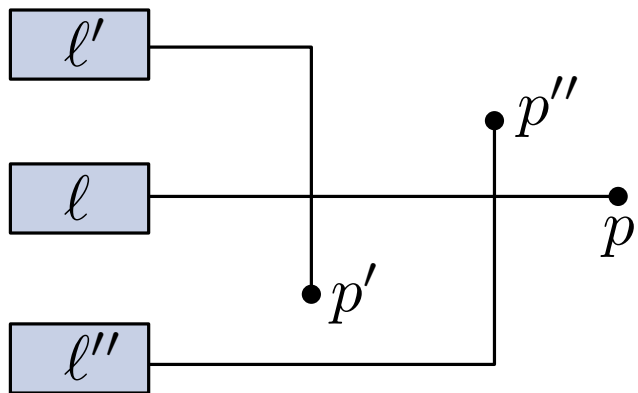
Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in \mathcal{L}^* schneiden sich nicht.

B) Kein direkter Leader in \mathcal{L}^* kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

Annahme es gäbe einen solchen Leader:



Einziger Fall, denn wegen A) können sich $\lambda(p', l')$ und $\lambda(p'', l'')$ nicht schneiden.

Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in \mathcal{L}^* schneiden sich nicht.

B) Kein direkter Leader in \mathcal{L}^* kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

Annahme es gäbe einen solchen Leader:



Kürzere Gesamtlänge der Leader nach Neuverdrahtung

 Minimalität von \mathcal{L}^*

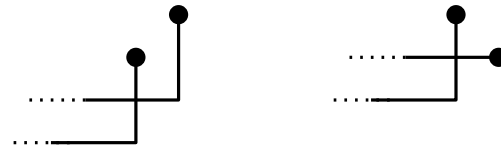
Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in \mathcal{L}^* schneiden sich nicht.

B) Kein direkter Leader in \mathcal{L}^* kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

Absteigende Kreuzung: Beteiligung eines absteigenden Leader.



Aufsteigende Kreuzung: Beteiligung eines aufsteigenden Leader.



Menge absteigender Kreuzungen und Menge aufsteigender Kreuzungen sind somit disjunkt.

Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

Beweis:

A) Absteigende und aufsteigende Leader in \mathcal{L}^* schneiden sich nicht.

B) Kein direkter Leader in \mathcal{L}^* kann sowohl einen absteigenden als auch einen aufsteigenden Leader schneiden.

Idee:

Entferne alle aufsteigenden Kreuzungen \mathcal{L}^* , sodass

- die Gesamtlänge der Leader unverändert bleibt und
- keine absteigenden Kreuzungen entstehen.

Selbes Vorgehen für absteigende Kreuzungen durch Spiegeln der Instanz.



Kreuzungsfreie Zuordnung \mathcal{L} mit gleicher Länge wie \mathcal{L}^* .

Fortsetzung Beweis:

Algorithmus zur Entfernung aller aufsteigenden Kreuzungen:

Sortiere Label von unten nach oben: l_1, \dots, l_n

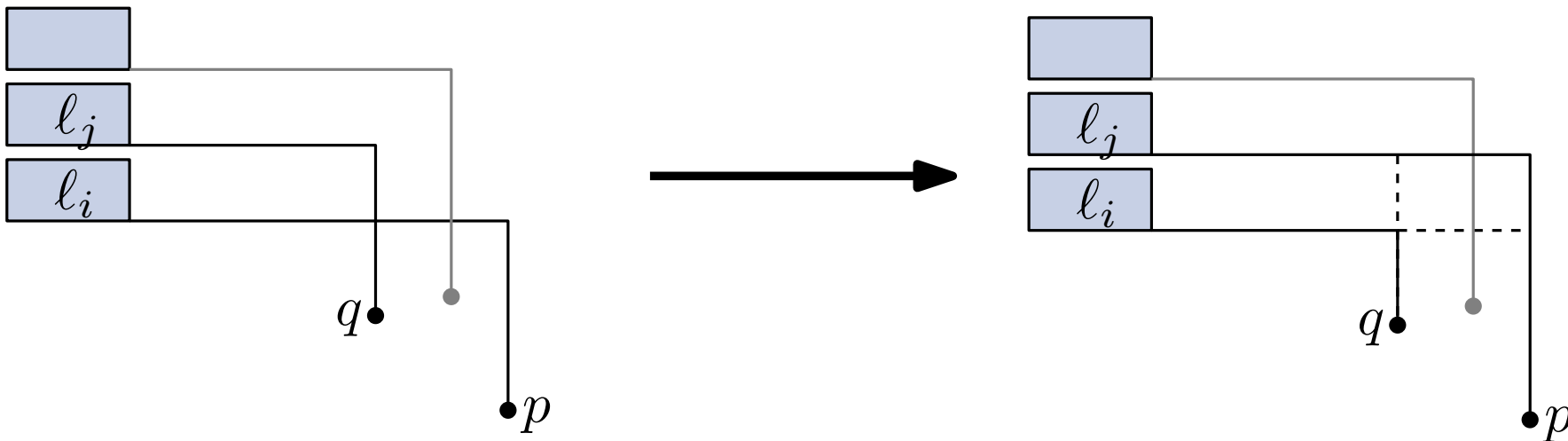
für $i = 1 \dots n$ **tue**

Sei $\lambda(p, l_i)$ Leader in \mathcal{L}^* .

wenn $\lambda(p, l_i)$ aufsteigender Leader ist **dann**

Bestimme leader $\lambda(q, l_j)$ mit linkester Kreuzung mit $\lambda(p, l_i)$.

Neuverdrahtung: $\lambda(q, l_i)$ und $\lambda(p, l_j)$.



Fortsetzung Beweis:

Algorithmus zur Entfernung aller aufsteigenden Kreuzungen:

Sortiere Label von unten nach oben: ℓ_1, \dots, ℓ_n

für $i = 1 \dots n$ **tue**

Sei $\lambda(p, \ell_i)$ Leader in \mathcal{L}^* .

wenn $\lambda(p, \ell_i)$ aufsteigender Leader ist **dann**

Bestimme leader $\lambda(q, \ell_j)$ mit linkerster Kreuzung mit $\lambda(p, \ell_i)$.

Neuverdrahtung: $\lambda(q, \ell_i)$ und $\lambda(p, \ell_j)$.

Zeige für Schritt i :

1. Gesamtlänge der Leader bleibt erhalten.
2. Aufsteigende Kreuzungen entstehen nur oberhalb von Arm von $\lambda(p, \ell_i)$.
3. Es entstehen keine absteigenden Kreuzungen.
4. Laufzeit beträgt $O(n)$ Zeit.

Beweis durch Induktion über i : siehe Tafel.

Satz 1: Für jede Zuordnung \mathcal{L}^* mit po-Leadern und minimaler Länge gibt es eine kreuzungsfreie Zuordnung \mathcal{L} mit po-Leadern selber Länge. \mathcal{L} kann mithilfe von \mathcal{L}^* in $O(n^2)$ Zeit konstruiert werden. (Bekos et al. '07 und Benkert et al. '09)

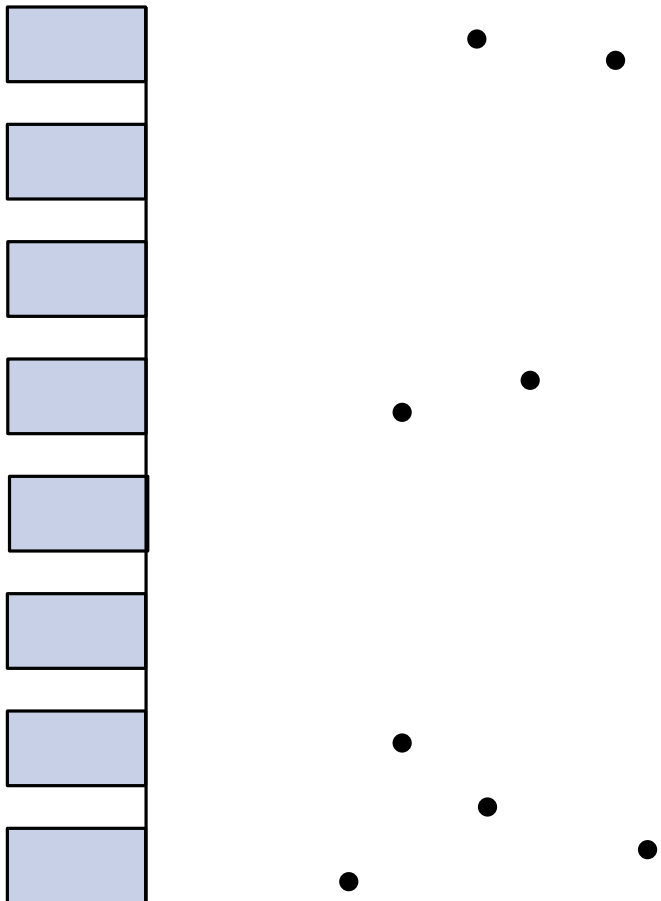
Satz kann auch auf do-Leader angepasst werden.

Sweep-Line-Verfahren

Jetzt: Verfahren für Minimierung der Leader-Längen in $O(n \log n)$ Zeit.

Vorgehen in zwei Phasen:

1. Teile Instanz in unabhängige Teilinstanzen auf.
2. Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

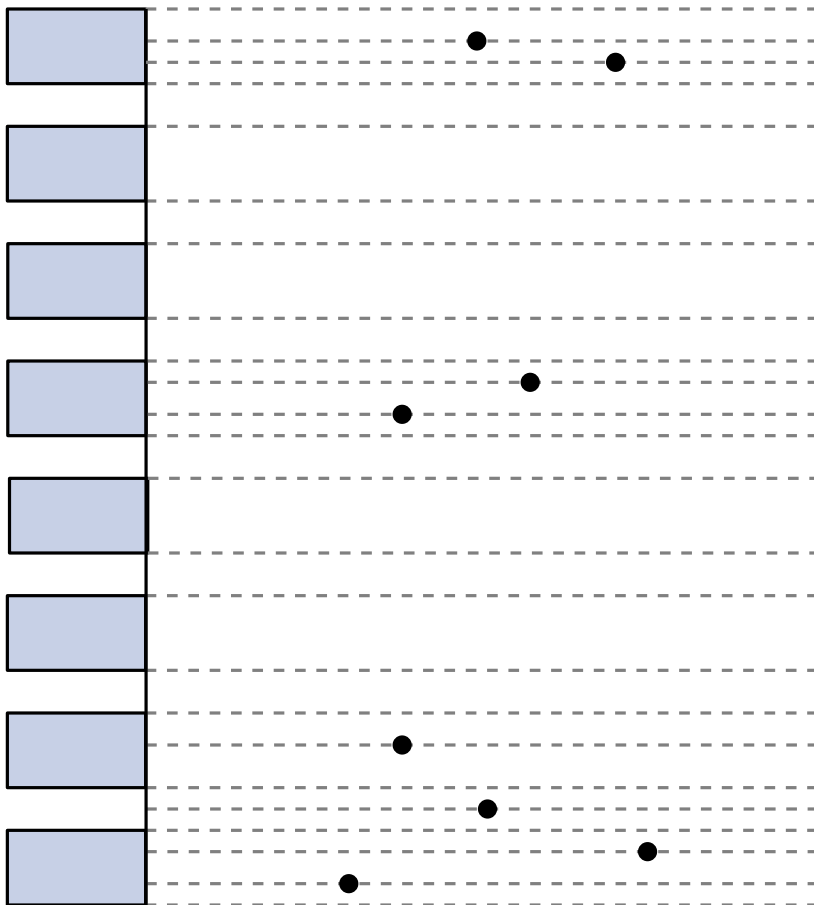


Sweep-Line-Verfahren

Jetzt: Verfahren für Minimierung der Leader-Längen in $O(n \log n)$ Zeit.

Vorgehen in zwei Phasen:

1. Teile Instanz in unabhängige Teilinstanzen auf.
2. Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

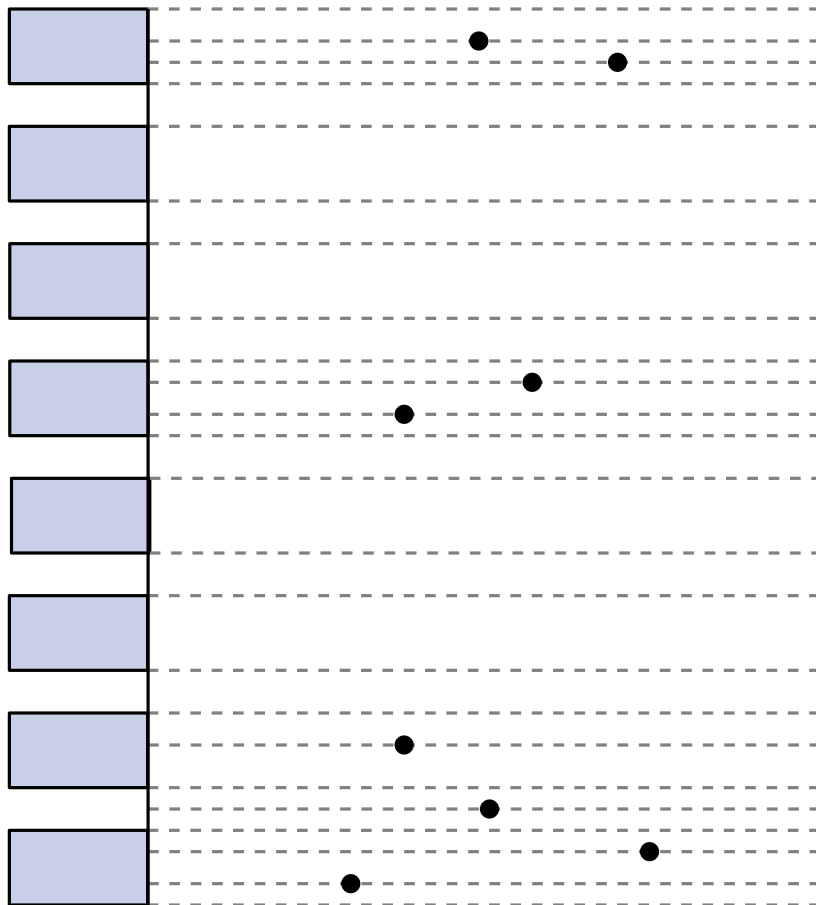


Unterteile hierzu Instanz in Streifen induziert durch Punkte in P und horizontale Seiten der Label.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

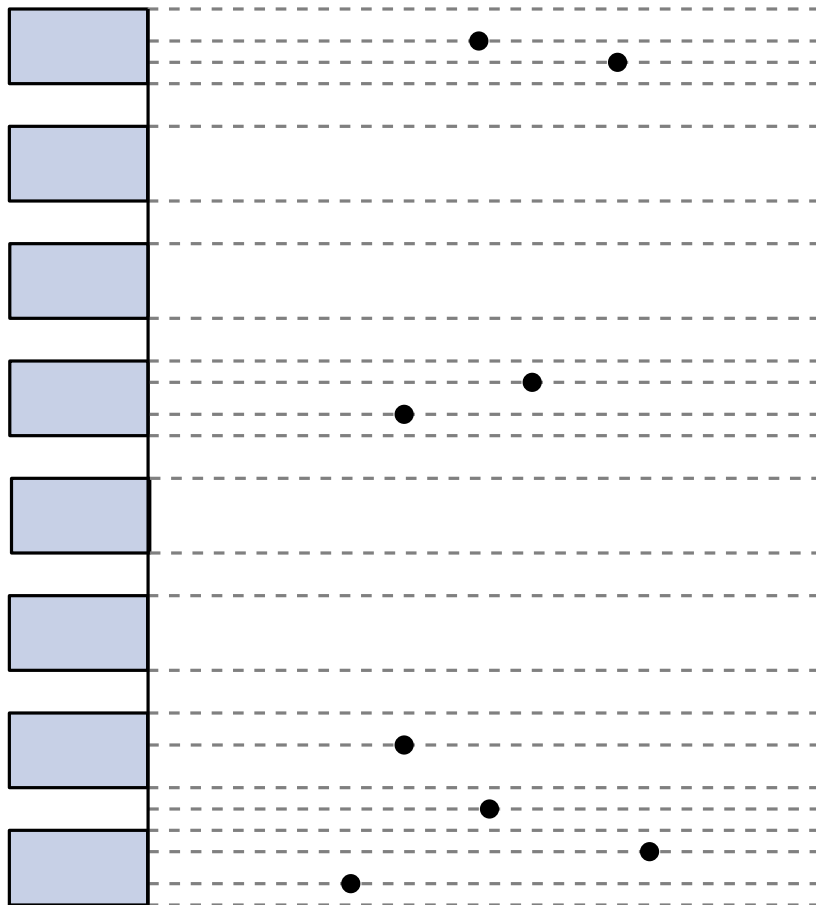
$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > la_\sigma$

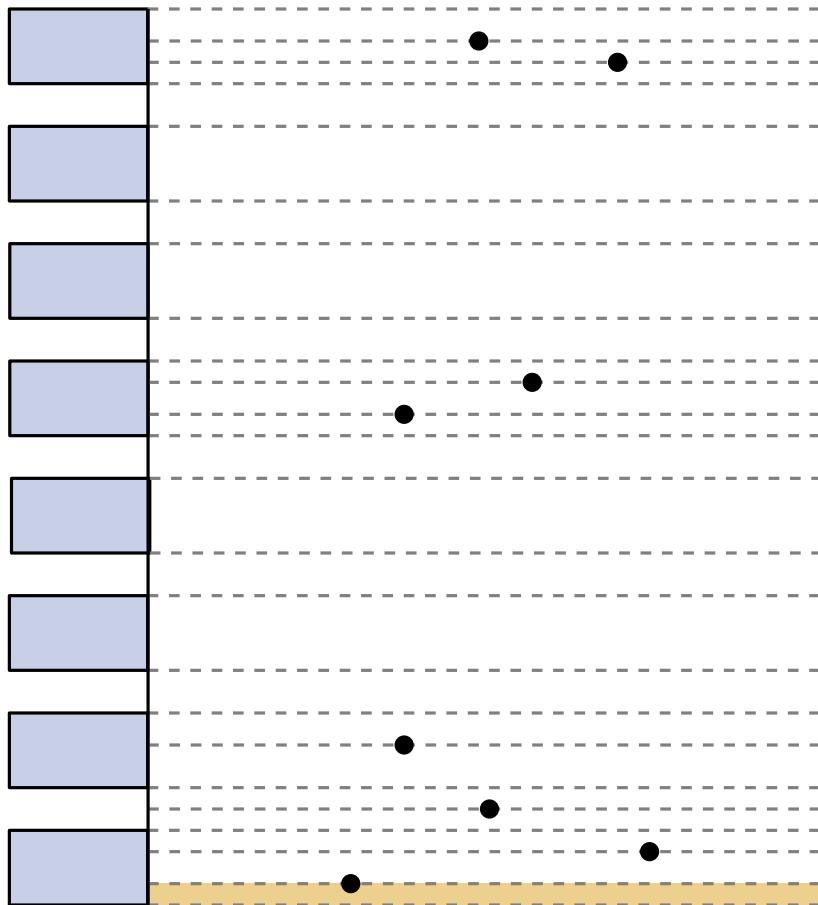
aufsteigend, falls $pb_\sigma > lb_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > la_\sigma$

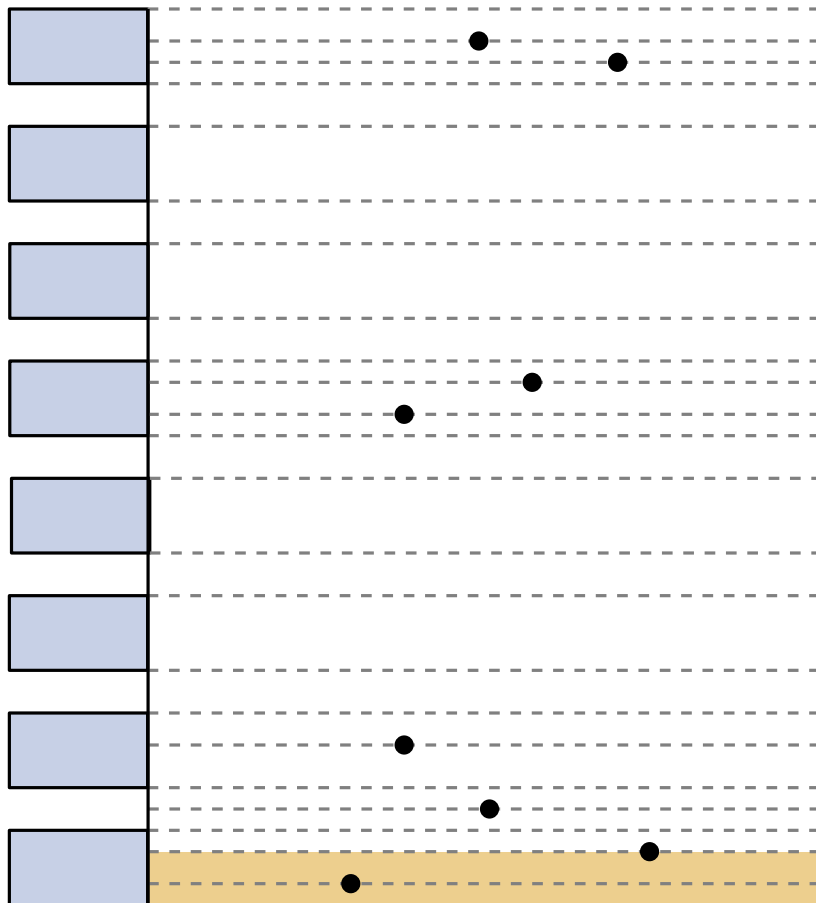
aufsteigend, falls $pb_\sigma > lb_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > la_\sigma$

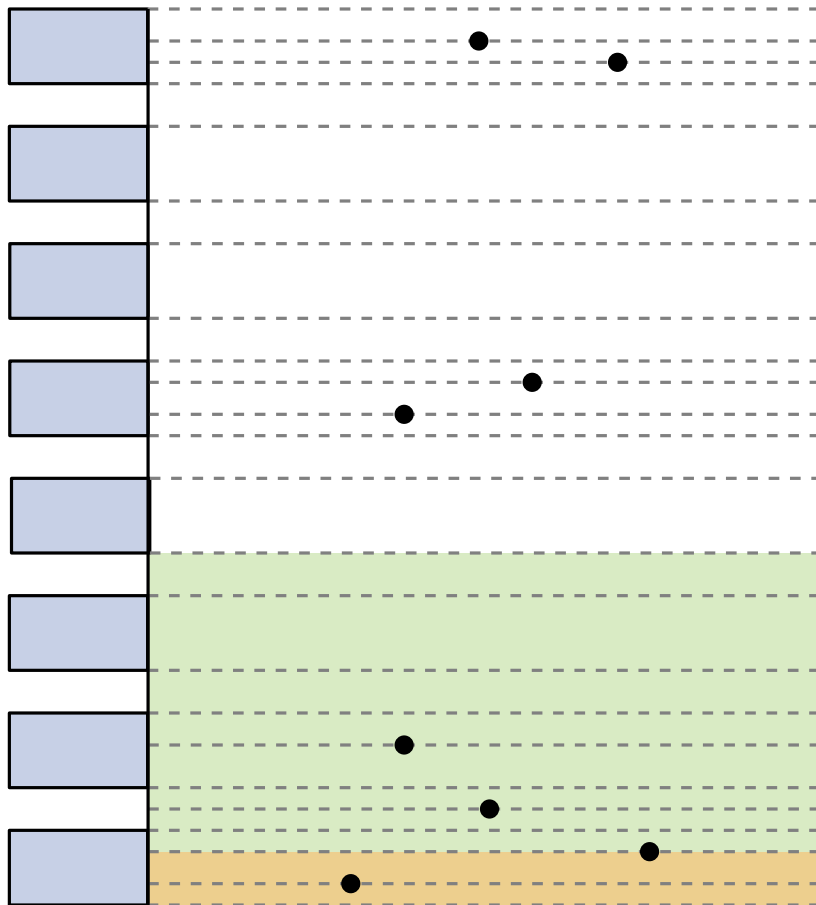
aufsteigend, falls $pb_\sigma > lb_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > la_\sigma$

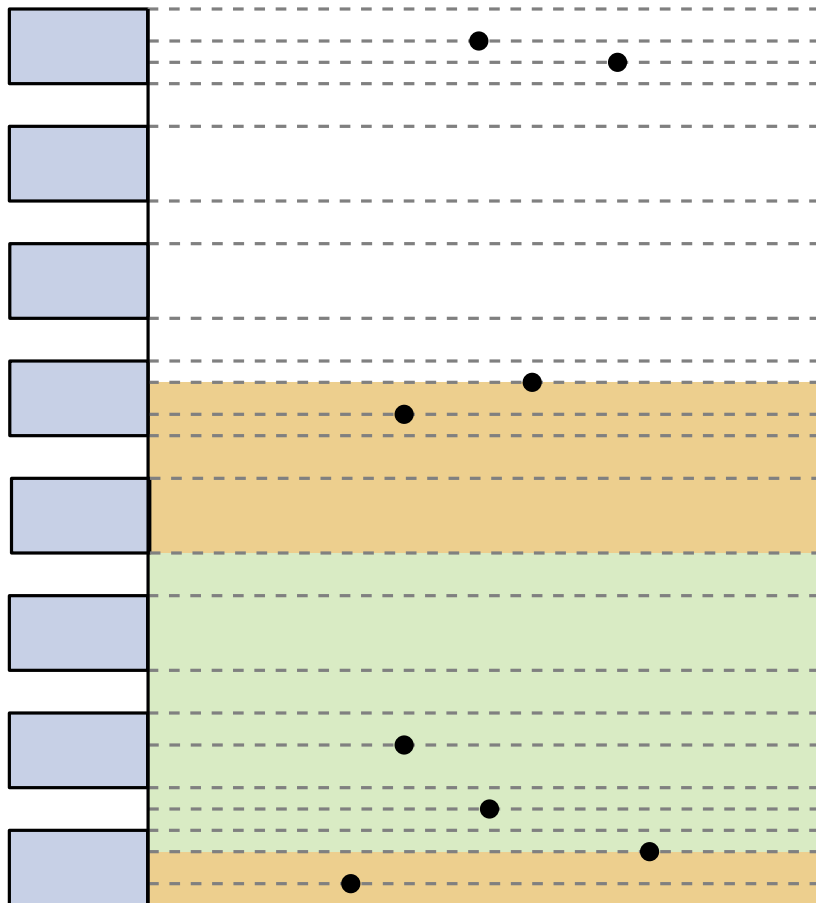
aufsteigend, falls $pb_\sigma > lb_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > la_\sigma$

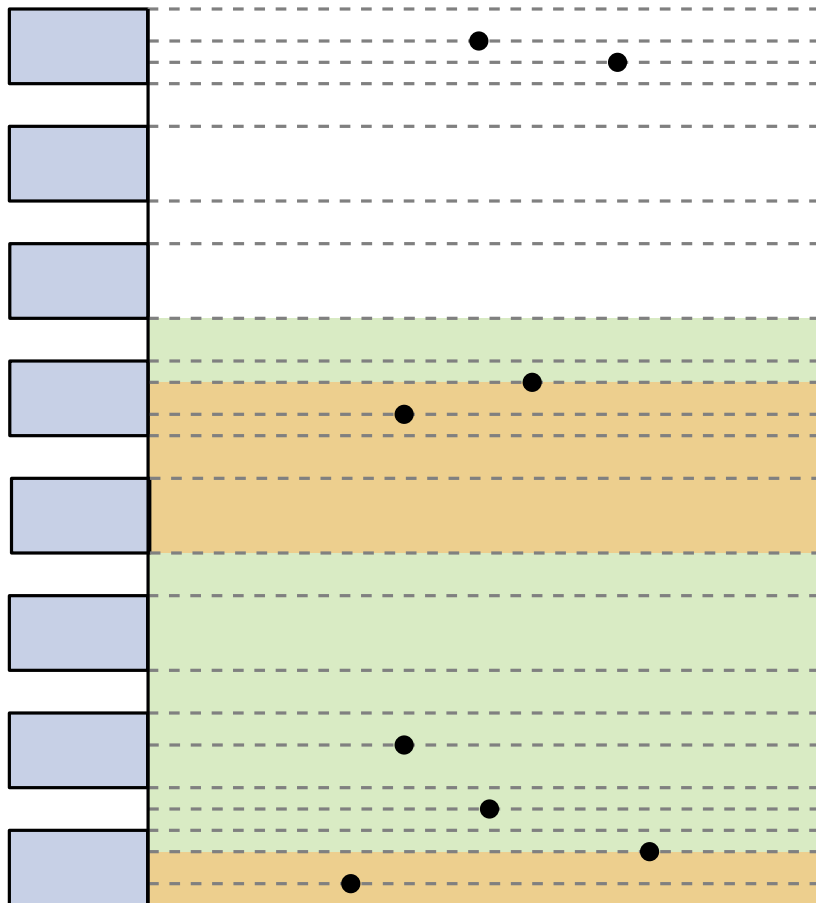
aufsteigend, falls $pb_\sigma > lb_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > la_\sigma$

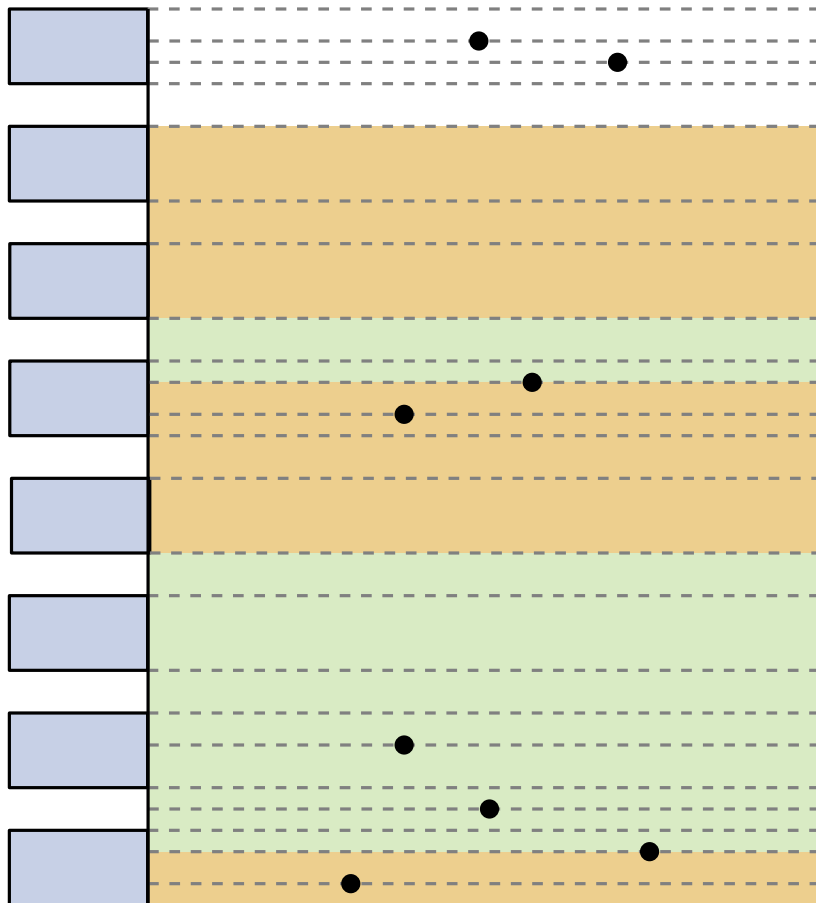
aufsteigend, falls $pb_\sigma > lb_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > la_\sigma$

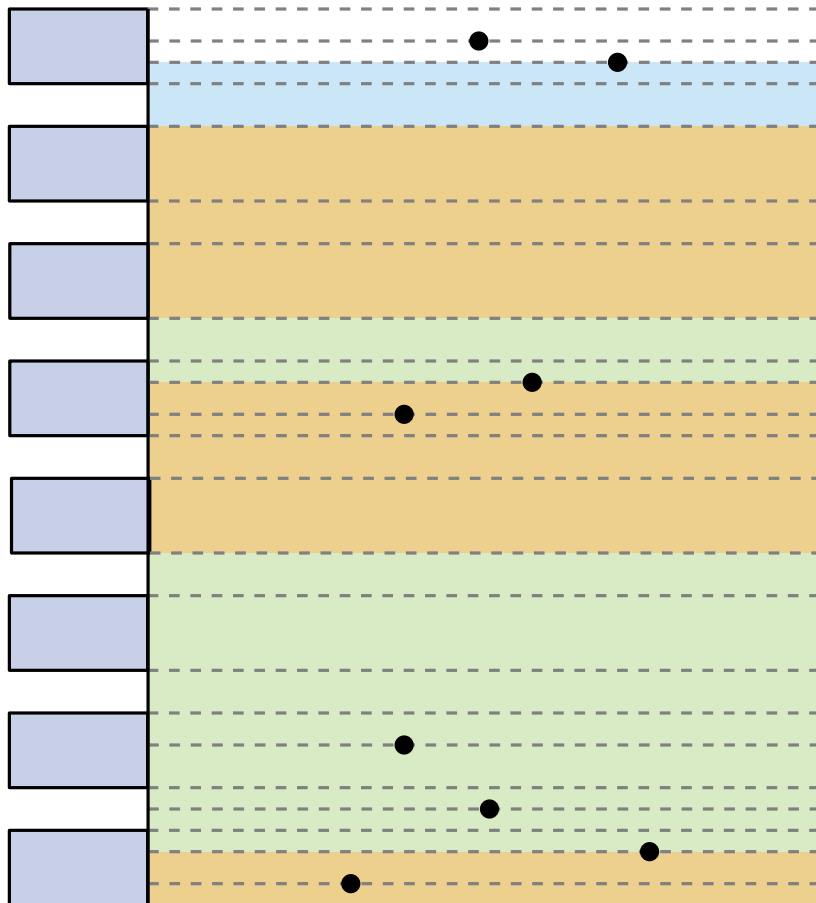
aufsteigend, falls $pb_\sigma > lb_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Ein Streifen σ heißt:

absteigend, falls $pa_\sigma > la_\sigma$

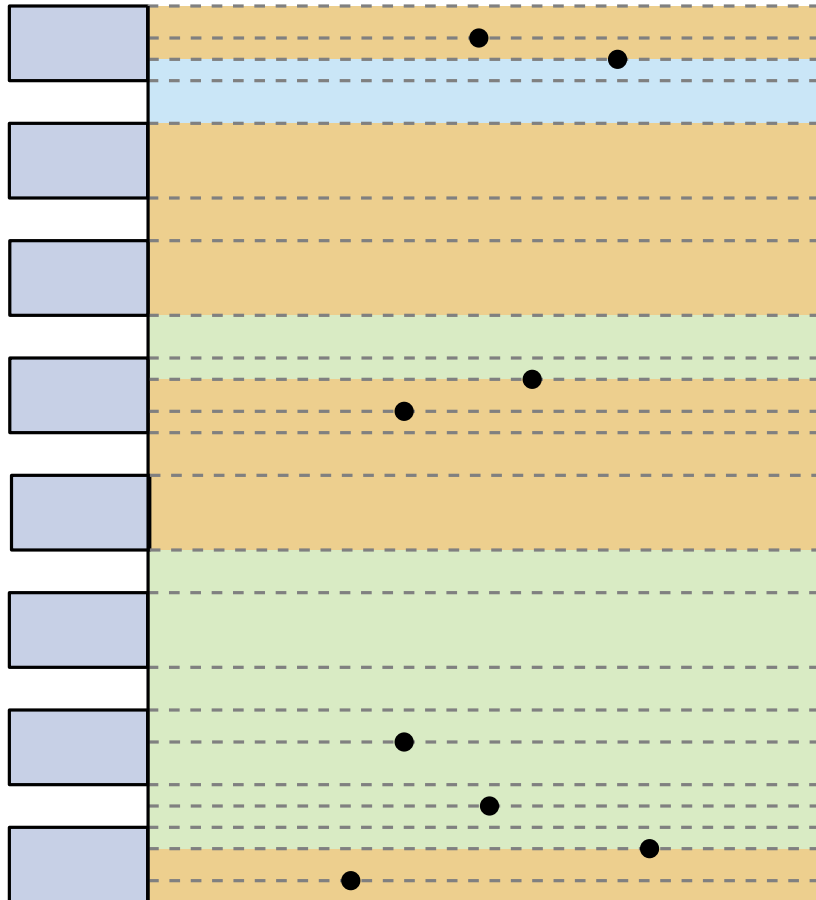
aufsteigend, falls $pb_\sigma > lb_\sigma$

neutral, in allen anderen Fällen.

Sweep-Line-Verfahren

1. Phase: Teile Instanz in unabhängige Teilinstanzen auf.

Traversiere Streifen von unten nach oben und bestimme folgende Anzahlen für jeden Streifen σ :



$pa_\sigma = \#$ Punkte oberhalb von σ inkl.
Punkte auf oberer Kante von σ .

$la_\sigma = \#$ Label oberhalb von σ inkl.
Label die σ schneiden.

$pb_\sigma = \#$ Punkte unterhalb von σ inkl.
Punkte auf unterer Kante von σ .

$lb_\sigma = \#$ Label unterhalb von σ inkl.
Label die σ schneiden.

Ein Streifen σ heißt:

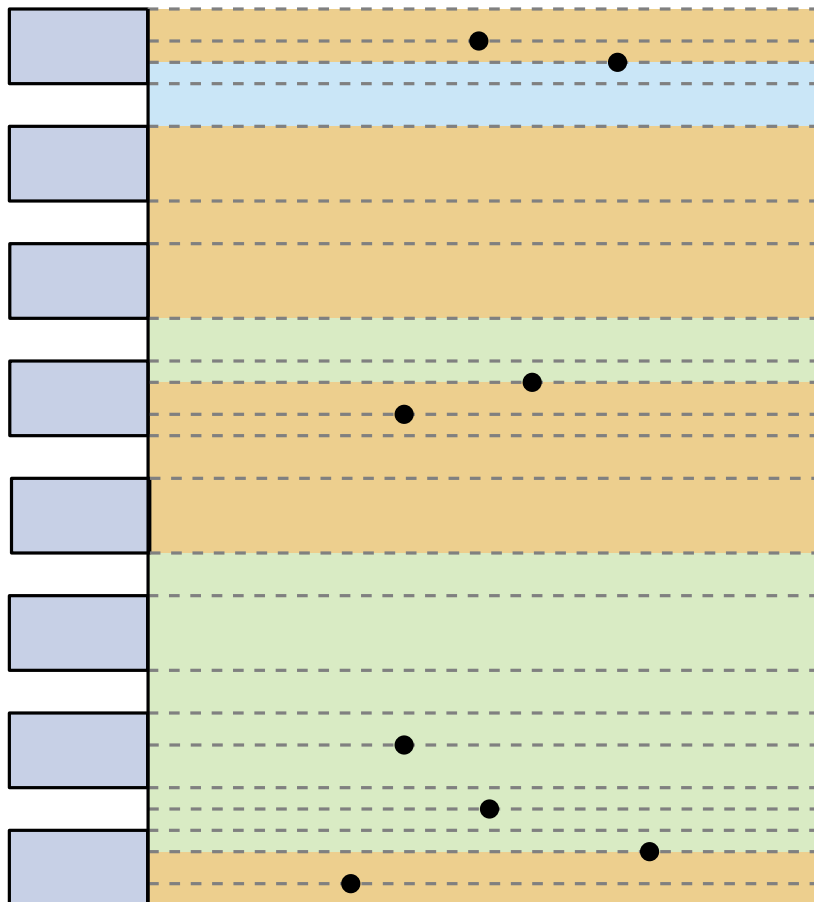
absteigend, falls $pa_\sigma > la_\sigma$

aufsteigend, falls $pb_\sigma > lb_\sigma$

neutral, in allen anderen Fällen.

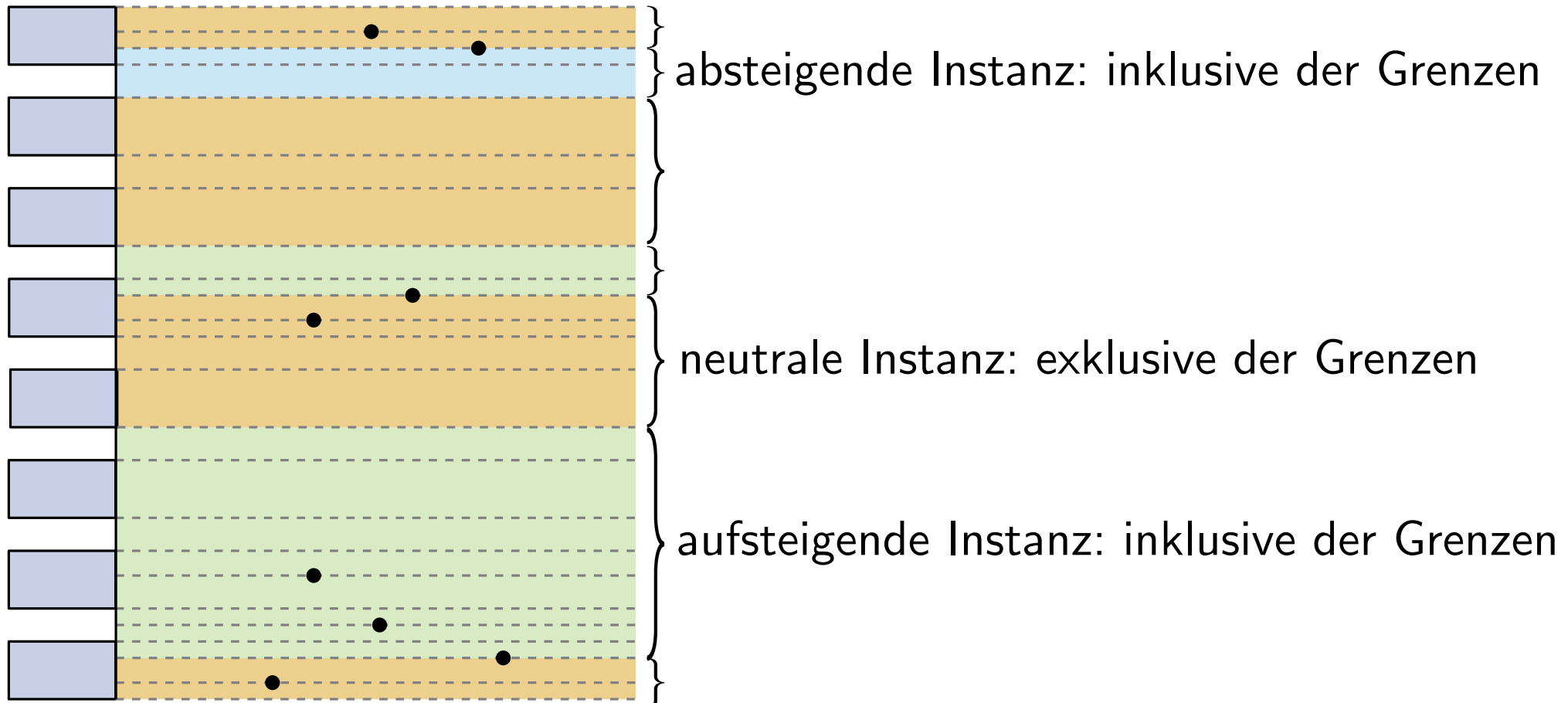
Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.



Sweep-Line-Verfahren

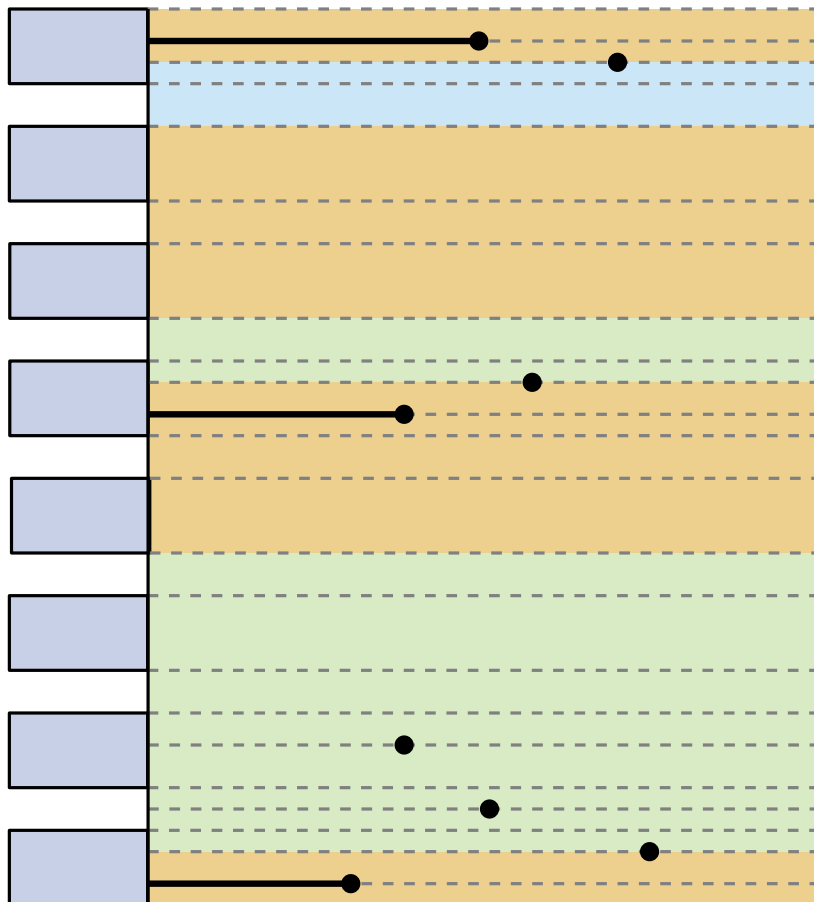
2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.



Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

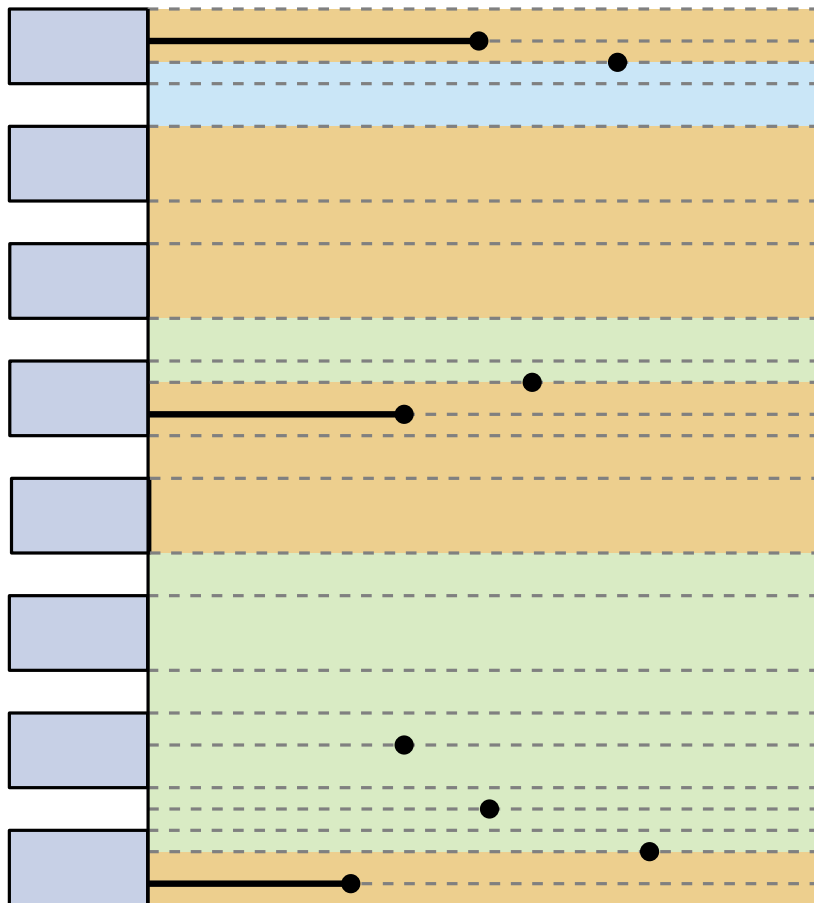


Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

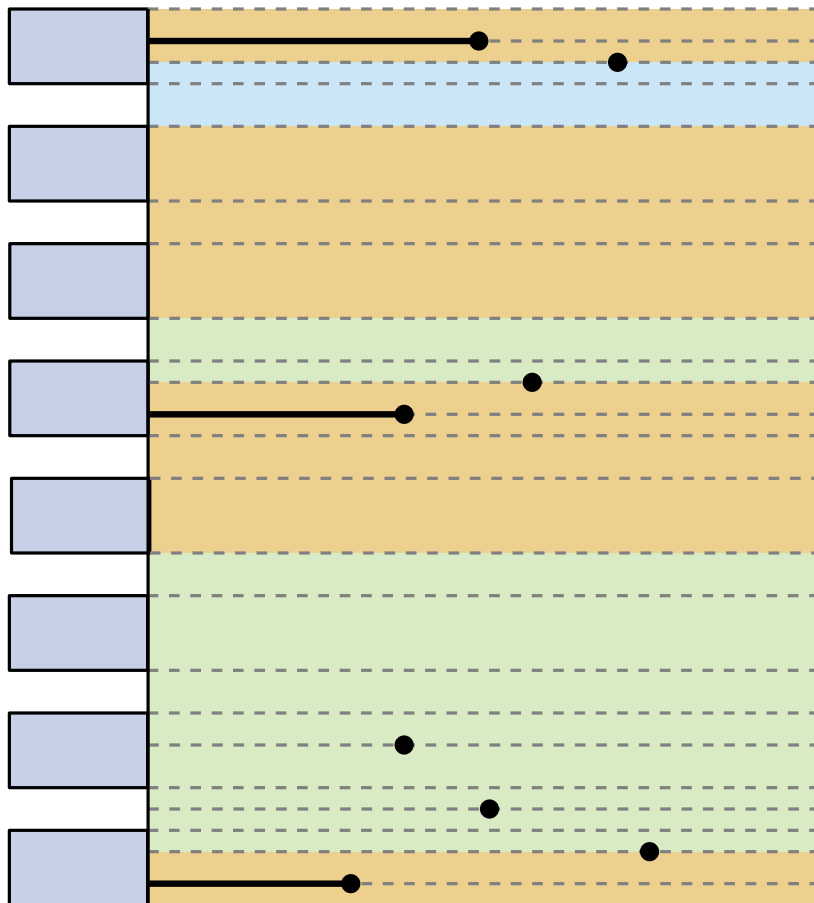
Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

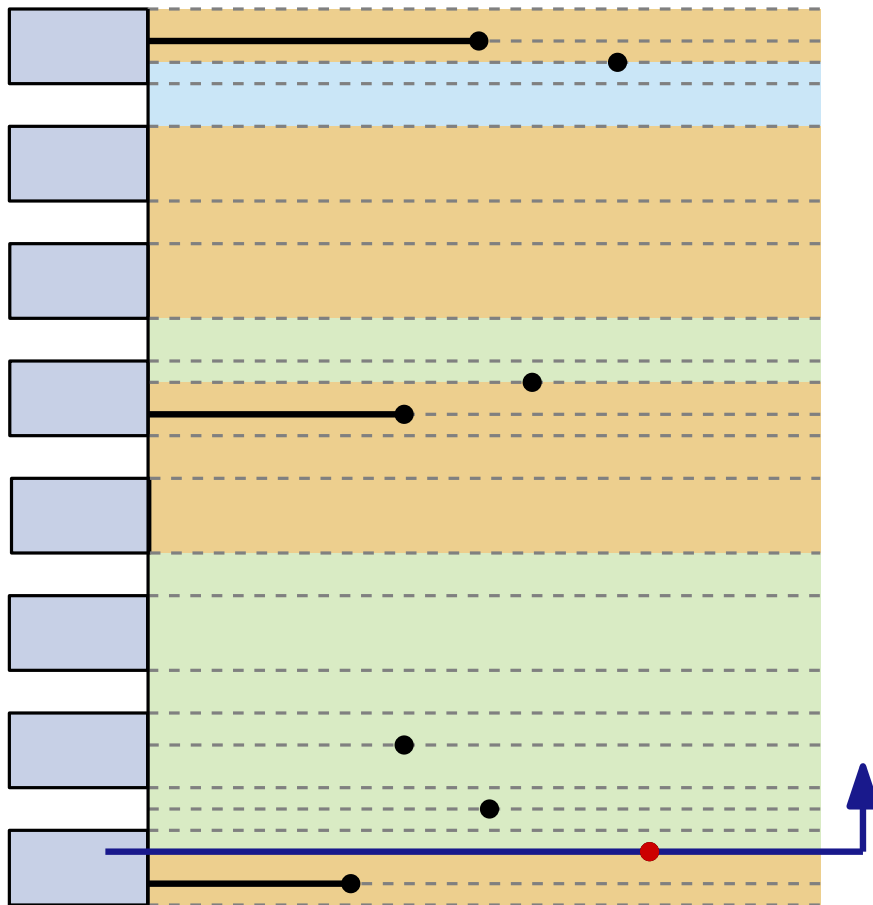
Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

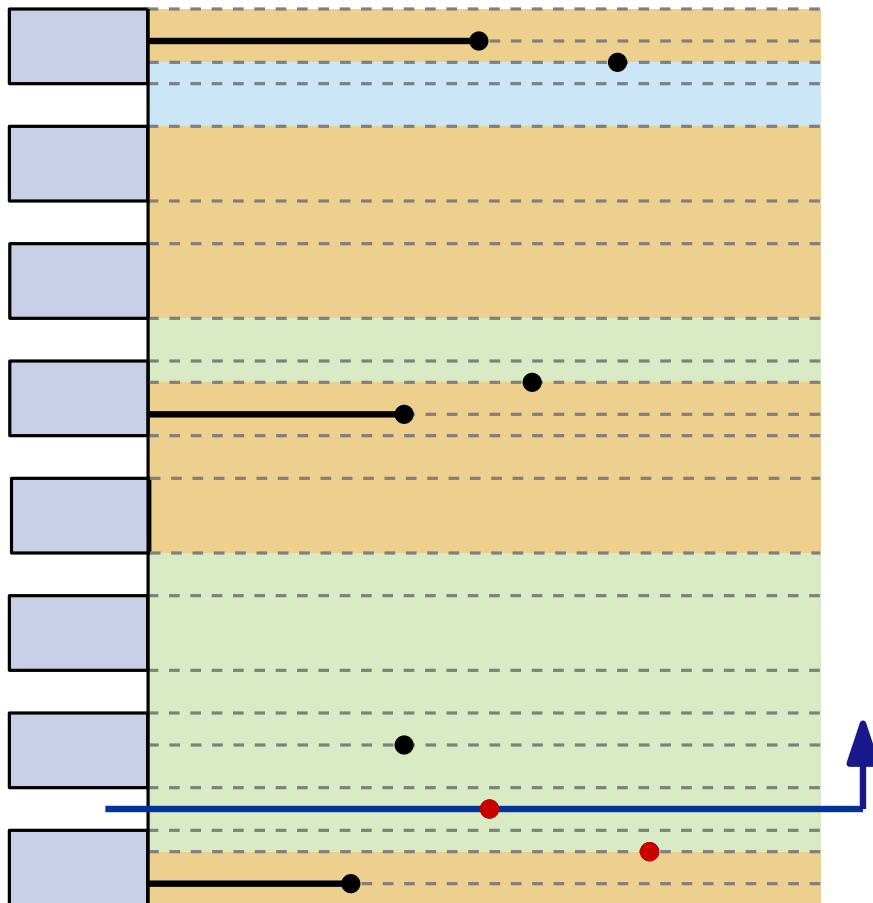
Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

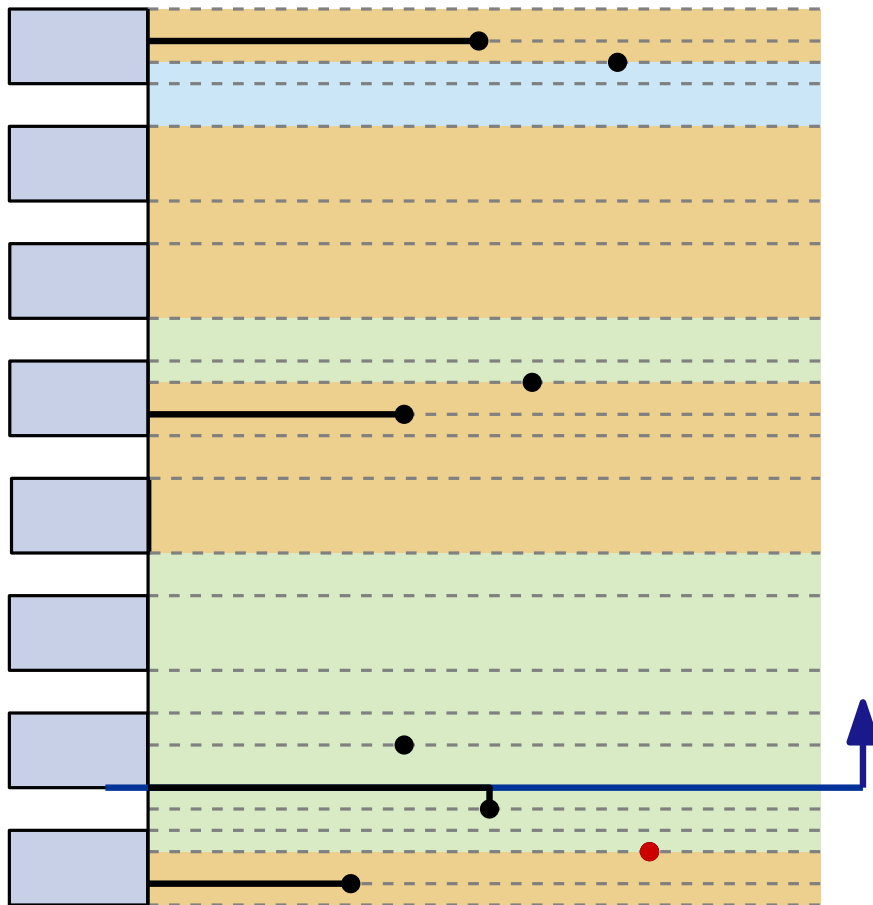
Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

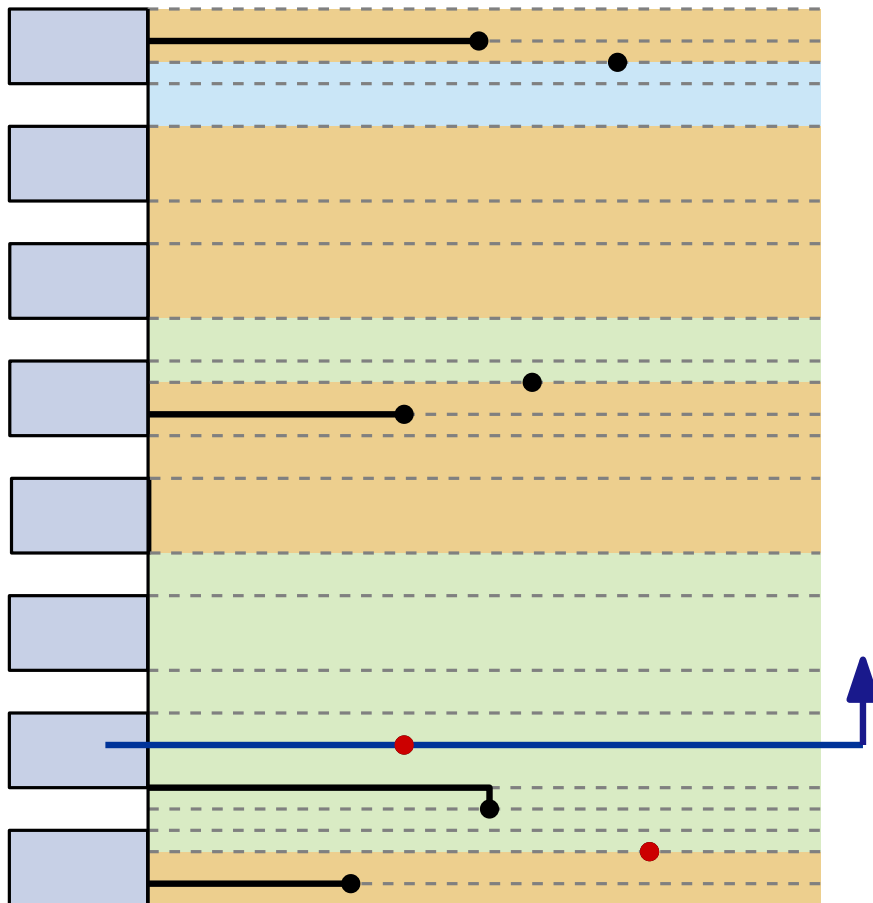
Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

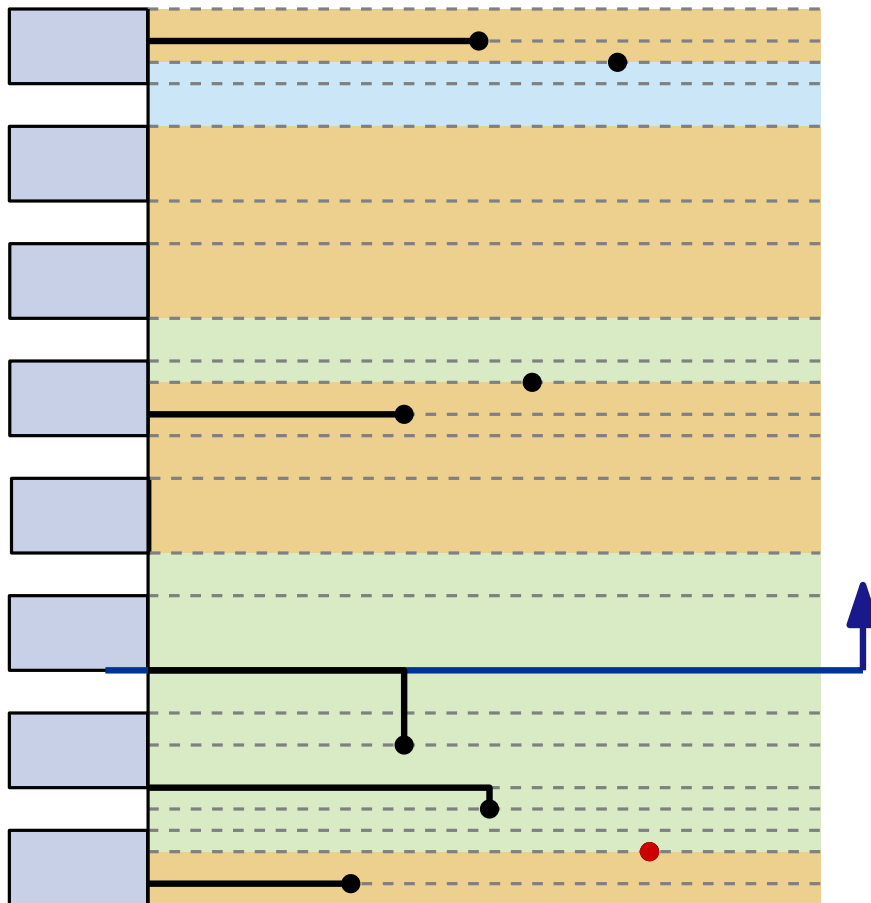
Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

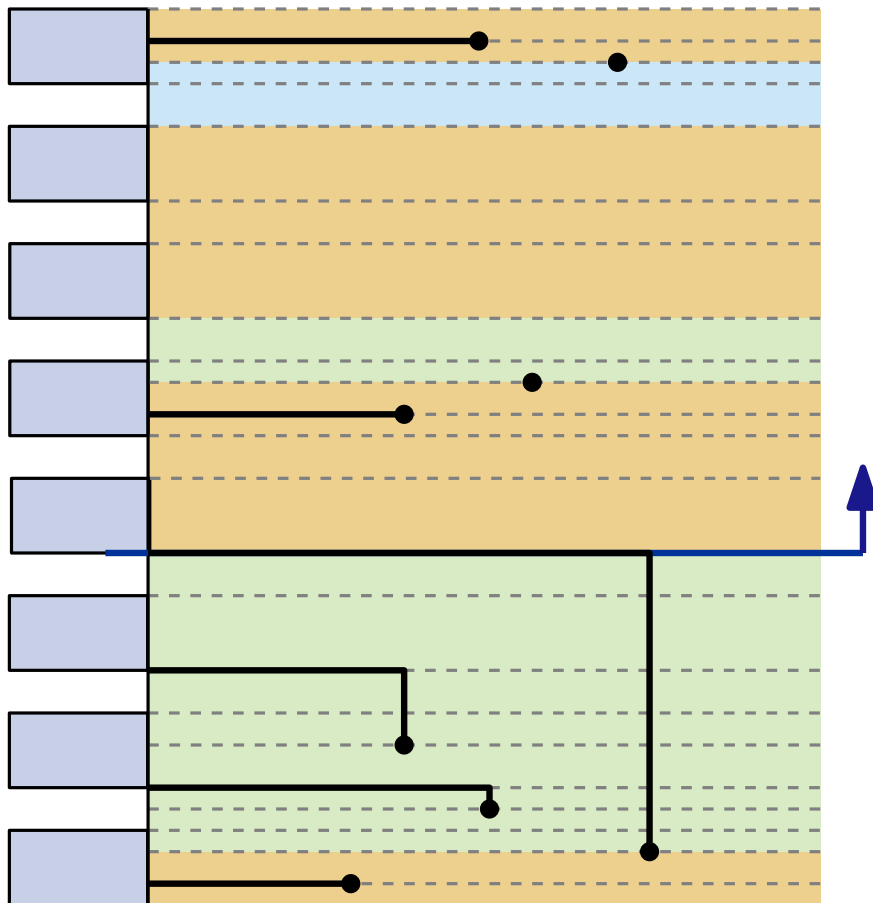
Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

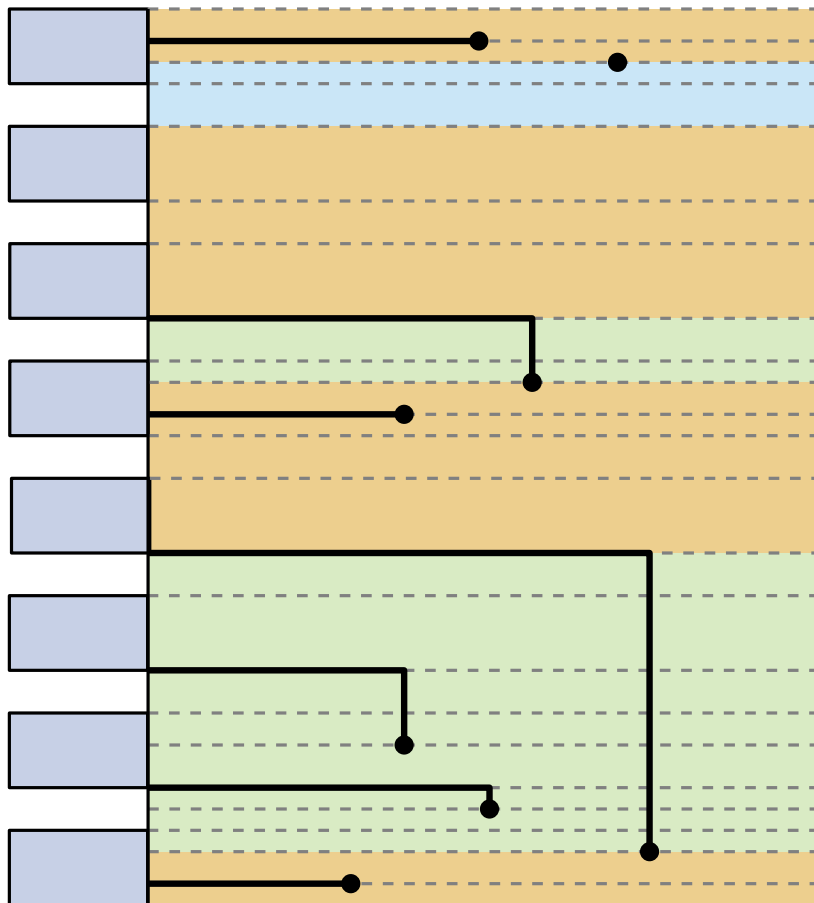
Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

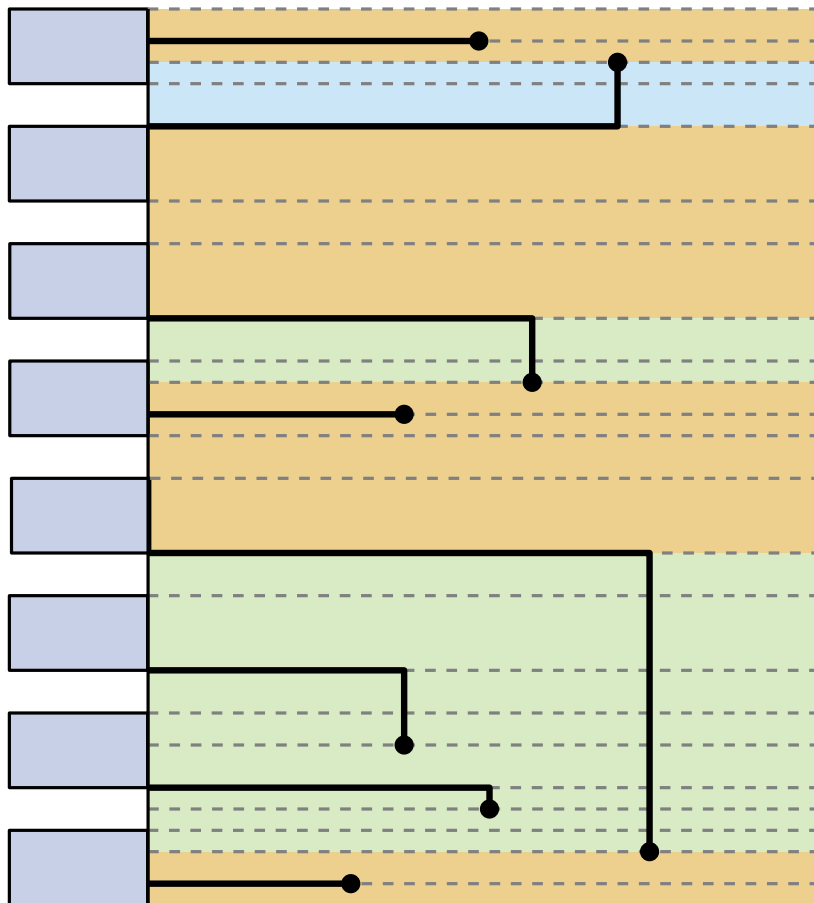
Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Sweep-Line-Verfahren

2. Phase: Löse Teilinstanzen mithilfe eines Sweep-Line-Verfahrens.

Neutrale Instanz: Verbinde Punkte und Label mit horizontaler Strecke.

Aufsteigende Instanz: Sweep-Line-Verfahren von unten nach oben.



Sweep-Line stoppt bei unterer Kante der Label und bei Punkten:

Punkt-Event p :

Füge p zu einer nach x -Koordinanten sortierten Warteliste W hinzu.

Label-Event ℓ :

Entferne Punkt p aus W , der am weitesten links liegt. Verbinde p mit ℓ mittels kürzesten Leader.

Analoges Vorgehen für absteigende Instanz.

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Beweis: Sei $pa_\sigma = \ell a_\sigma$. (Analog $pb_\sigma = \ell b_\sigma$)

Angenommen es gibt Leader λ , der neutralen Streifen σ kreuzt.

1. Fall: λ ist mit Label ℓ verbunden, das σ kreuzt.

2. Fall: λ verbindet Punkt p oberhalb von σ mit Label ℓ unterhalb von σ

3. Fall: λ verbindet Punkt p unterhalb von σ mit Label ℓ oberhalb von σ

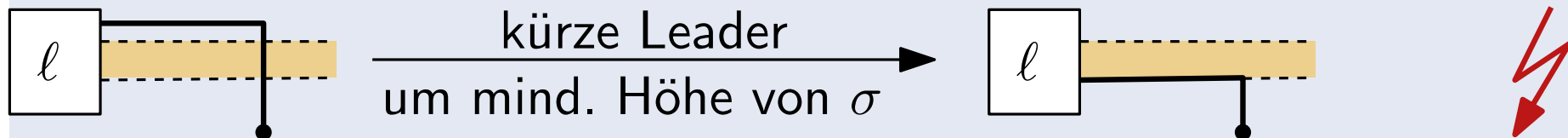
Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Beweis: Sei $pa_\sigma = \ell a_\sigma$. (Analog $pb_\sigma = \ell b_\sigma$)

Angenommen es gibt Leader λ , der neutralen Streifen σ kreuzt.

1. Fall: λ ist mit Label ℓ verbunden, das σ kreuzt.



2. Fall: λ verbindet Punkt p oberhalb von σ mit Label ℓ unterhalb von σ

3. Fall: λ verbindet Punkt p unterhalb von σ mit Label ℓ oberhalb von σ

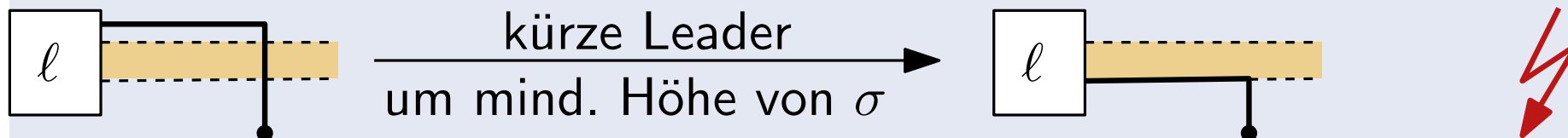
Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

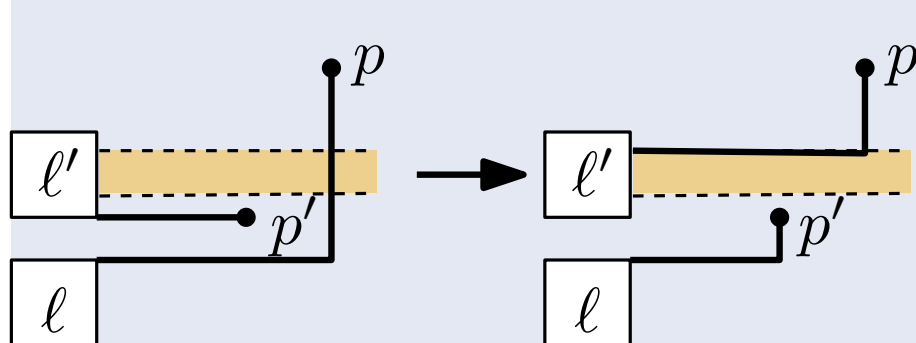
Beweis: Sei $pa_\sigma = la_\sigma$. (Analog $pb_\sigma = lb_\sigma$)

Angenommen es gibt Leader λ , der neutralen Streifen σ kreuzt.

1. Fall: λ ist mit Label ℓ verbunden, das σ kreuzt.



2. Fall: λ verbindet Punkt p oberhalb von σ mit Label ℓ unterhalb von σ



Wegen $pa_\sigma = la_\sigma$:

\exists Punkt p' unterhalb von σ , der mit Label ℓ' verbunden, sodass

- ℓ' schneidet σ , oder
- ℓ' liegt oberhalb von σ .

Vertausche Leader \rightarrow Kürzere Leader

3. Fall: λ verbindet Punkt p unterhalb von σ mit Label ℓ oberhalb von σ

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > la_\sigma$ $\rightarrow pa_\sigma - la_\sigma \geq 1$ und $pb_\sigma < lb_\sigma$

aufsteigend, falls $pb_\sigma > lb_\sigma$ $\rightarrow pb_\sigma - lb_\sigma \geq 1$ und $pa_\sigma < la_\sigma$

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > la_\sigma$ $\rightarrow pa_\sigma - la_\sigma \geq 1$ und $pb_\sigma < lb_\sigma$

aufsteigend, falls $pb_\sigma > lb_\sigma$ $\rightarrow pb_\sigma - lb_\sigma \geq 1$ und $pa_\sigma < la_\sigma$

Warum?

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > la_\sigma$ $\rightarrow pa_\sigma - la_\sigma \geq 1$ und $pb_\sigma < lb_\sigma$

aufsteigend, falls $pb_\sigma > lb_\sigma$ $\rightarrow pb_\sigma - lb_\sigma \geq 1$ und $pa_\sigma < la_\sigma$

$$pa_{\sigma \downarrow} - la_{\sigma \downarrow} \geq 1$$

$$pa_{\sigma \uparrow} - la_{\sigma \uparrow} < 0$$

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > la_\sigma$ $\rightarrow pa_\sigma - la_\sigma \geq 1$ und $pb_\sigma < lb_\sigma$

aufsteigend, falls $pb_\sigma > lb_\sigma$ $\rightarrow pb_\sigma - lb_\sigma \geq 1$ und $pa_\sigma < la_\sigma$

$$pa_{\sigma\downarrow} - la_{\sigma\downarrow} \geq 1$$

$$pa_{\sigma\uparrow} - la_{\sigma\uparrow} < 0$$

$$\rightarrow |(pa_{\sigma\downarrow} - la_{\sigma\downarrow}) - (pa_{\sigma\uparrow} - la_{\sigma\uparrow})| \geq 2$$

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > la_\sigma$ $\rightarrow pa_\sigma - la_\sigma \geq 1$ und $pb_\sigma < lb_\sigma$

aufsteigend, falls $pb_\sigma > lb_\sigma$ $\rightarrow pb_\sigma - lb_\sigma \geq 1$ und $pa_\sigma < la_\sigma$

$$pa_{\sigma\downarrow} - la_{\sigma\downarrow} \geq 1$$

$$pa_{\sigma\uparrow} - la_{\sigma\uparrow} < 0$$

$$\rightarrow |(pa_{\sigma\downarrow} - la_{\sigma\downarrow}) - (pa_{\sigma\uparrow} - la_{\sigma\uparrow})| \geq 2$$

Da Punkte und Label in allgemeiner Lage gilt aber für zwei benachbarte Streifen σ und σ' :

$$|(pa_\sigma - la_\sigma) - (pa_{\sigma'} - la_{\sigma'})| \leq 1$$

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

absteigend, falls $pa_\sigma > la_\sigma$ $\rightarrow pa_\sigma - la_\sigma \geq 1$ und $pb_\sigma < lb_\sigma$

aufsteigend, falls $pb_\sigma > lb_\sigma$ $\rightarrow pb_\sigma - lb_\sigma \geq 1$ und $pa_\sigma < la_\sigma$

$$pa_{\sigma\downarrow} - la_{\sigma\downarrow} \geq 1$$

$$pa_{\sigma\uparrow} - la_{\sigma\uparrow} < 0$$

$$\rightarrow |(pa_{\sigma\downarrow} - la_{\sigma\downarrow}) - (pa_{\sigma\uparrow} - la_{\sigma\uparrow})| \geq 2$$

Da Punkte und Label in allgemeiner Lage gilt aber für zwei benachbarte Streifen σ und σ' :

$$|(pa_\sigma - la_\sigma) - (pa_{\sigma'} - la_{\sigma'})| \leq 1$$

$\sigma \downarrow$ und $\sigma \uparrow$ sind nicht benachbart.

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

—→ Wegen Lemma 1: In jeder optimalen Lösung werden die Punkte einer aufsteigenden/absteigenden Instanz S von Leadern angebunden, die vollständig in S liegen.

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

—► Wegen Lemma 1: In jeder optimalen Lösung werden die Punkte einer aufsteigenden/absteigenden Instanz S von Leadern angebunden, die vollständig in S liegen.

2. Sei σ unterster Streifen einer aufsteigenden Instanz S . In jeder optimalen Lösung werden ausschließlich Label oberhalb von σ zur Beschriftung der Punkte in S verwendet (siehe Tafel). Analog: Absteigende Instanz.

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Beweis:

1. Zwischen absteigenden Streifen $\sigma \downarrow$ und aufsteigenden Streifen $\sigma \uparrow$ befindet sich immer ein neutraler Streifen.

—→ Wegen Lemma 1: In jeder optimalen Lösung werden die Punkte einer aufsteigenden/absteigenden Instanz S von Leadern angebunden, die vollständig in S liegen.

2. Sei σ unterster Streifen einer aufsteigenden Instanz S . In jeder optimalen Lösung werden ausschließlich Label oberhalb von σ zur Beschriftung der Punkte in S verwendet (siehe Tafel). Analog: Absteigende Instanz.

3. Der Algorithmus berechnet eine solche Zuordnung und diese ist optimal und kreuzungsfrei (siehe Tafel).

Sweep-Line-Verfahren

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von po -Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Lemma 1: Für jede kreuzungsfreie Zuordnung mit minimaler Länge gilt, dass kein Leader einen neutralen Streifen kreuzt (Übung).

Lemma 2: Für eine Instanz (P, L, R) kann eine kreuzungsfreie Zuordnung minimaler Länge mithilfe von *po*-Leadern in $O(n \log n)$ Zeit und $O(n)$ Speicher berechnet werden.

Satz 2: Die Berechnung einer kreuzungsfreien Zuordnung minimaler Länge einer Instanz (P, L, R) mit *po*-Leadern benötigt Laufzeit $O(n \log n)$ Zeit und $O(n)$ Speicher.