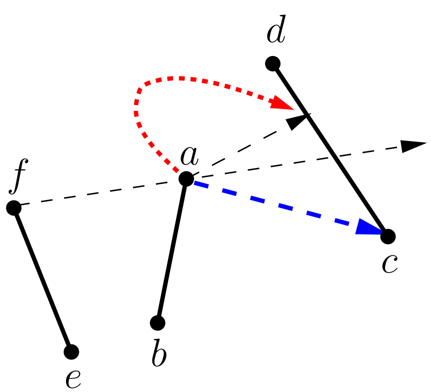
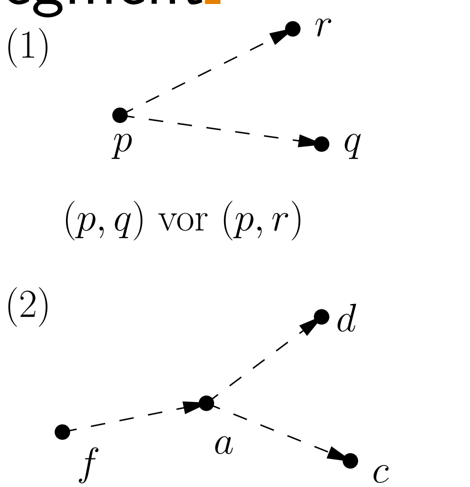
Offline Bewegungsplanung: Der simultane Sweep

Elmar Langetepe
University of Bonn

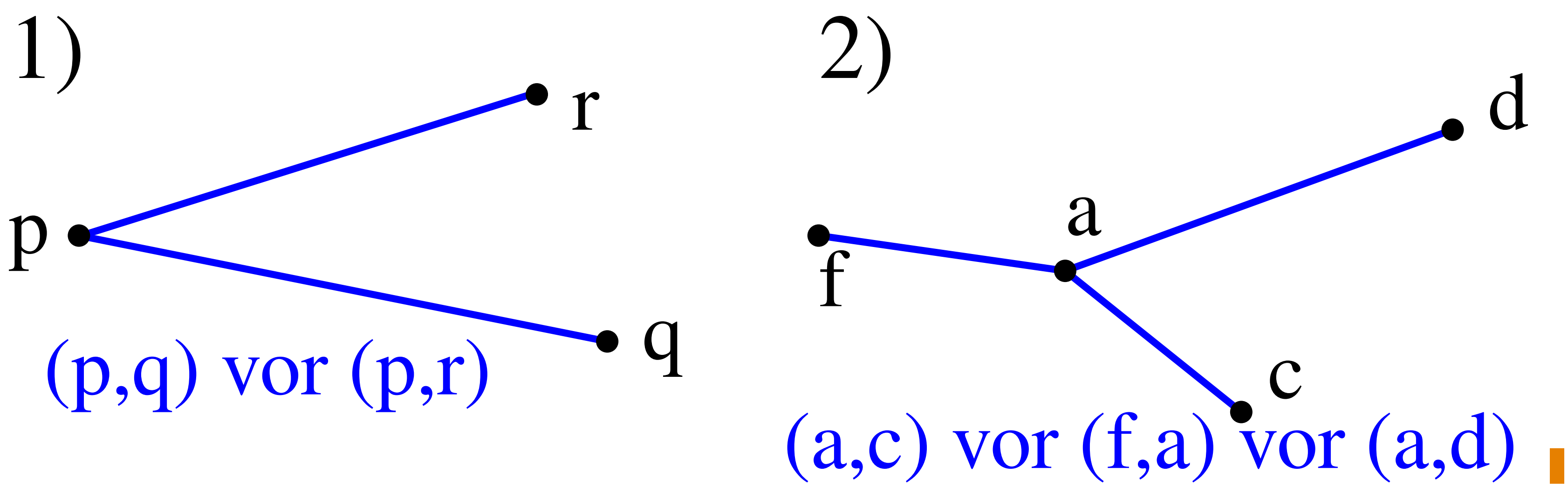
Geht das besser?

- Laufzeit $O(n^2 \log n)$, Komplexität $\Theta(n^2)$, Laufzeit $O(n^2)$?
- Simultaner Sweep, alle Segmente gleichzeitig drehen, Sicht-Informationen weiter geben
- Bearbeitungsreihenfolge: Paare von Endpunkten gemäß Steigung
- Zeiger auf momentan sichtbares Segment





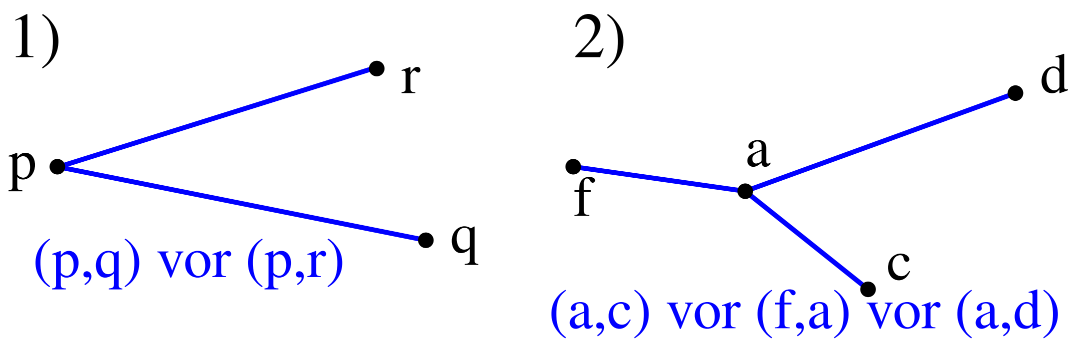
Bearbeitungsreihenfolge in $O(n^2)$

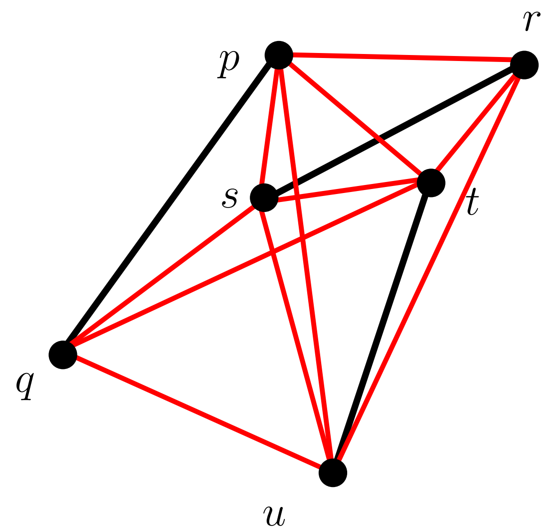


Liste von Punkten (p, q) mit $p_x \leq q_x$

Zeigen wir später! Annahme: Ist schon gegeben für die Punktpaare!

Beispiel Bearbeitungsreihenfolge





Volle Ordnung:

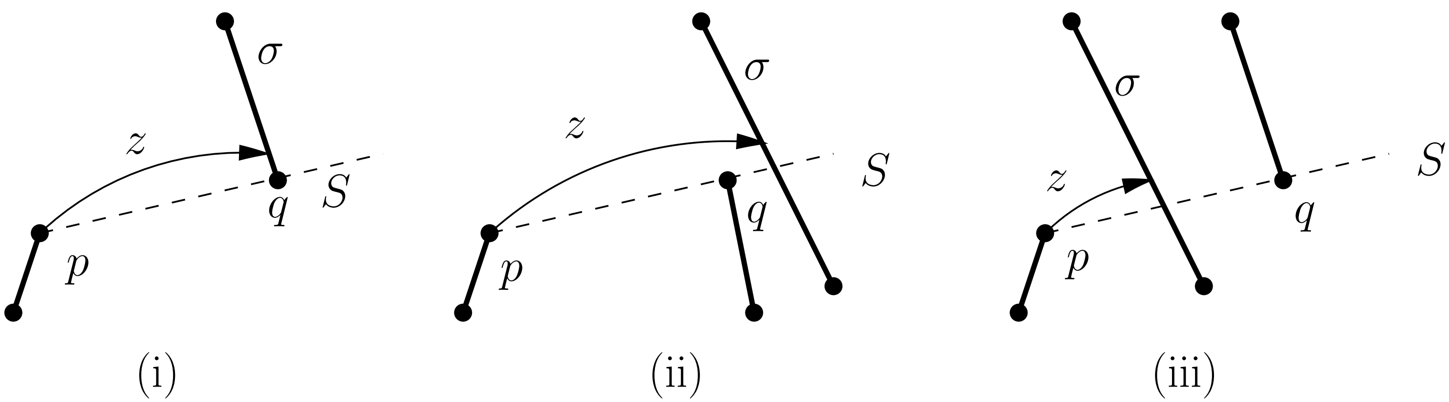
$(p,u),(s,u),(p,t),(q,u),(p,r),(s,t),(q,t),(q,s),(t,r),(u,r),(s,p)$

Partielle Ordung:

$(p,u),(p,t),(p,r),(q,u),(s,u),(s,t),(q,t),(q,s),\ldots$

Sweep mit Bearbeitungsreihenfolge

- ES gemäß Bearbeitungsreihenfolge: $\ldots, (p,q), \ldots, (p,r)$
- SSS: Für jedes p, (p,q) zuletzt dran:
  Zeiger z auf das kurz hinter q *sichtbare* Segment σ
- Drei Möglichkeiten: (i), (ii) und (iii)
- Ausgabe zwischendurch



Sweep mit Bearbeitungsreihenfolge

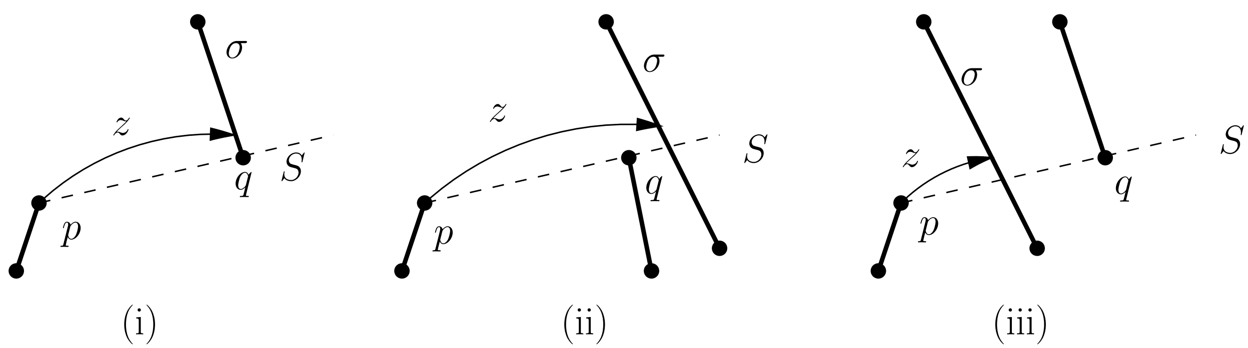
SSS: Für jedes p, (p, q) zuletzt dran:

Zeiger z auf kurz hinter q *sichtbare* Segment σ

(i) q ist von p aus sichtbar und Anfangspunkt von σ

(ii) q ist von p aus sichtbar und Endpunkt eines Segments. σ ist das auf dem Strahl S von p durch q nächste Segment.

(iii) q ist von p aus nicht sichtbar. σ ist das zu p nächste Segment auf dem Strahl S von p durch q.

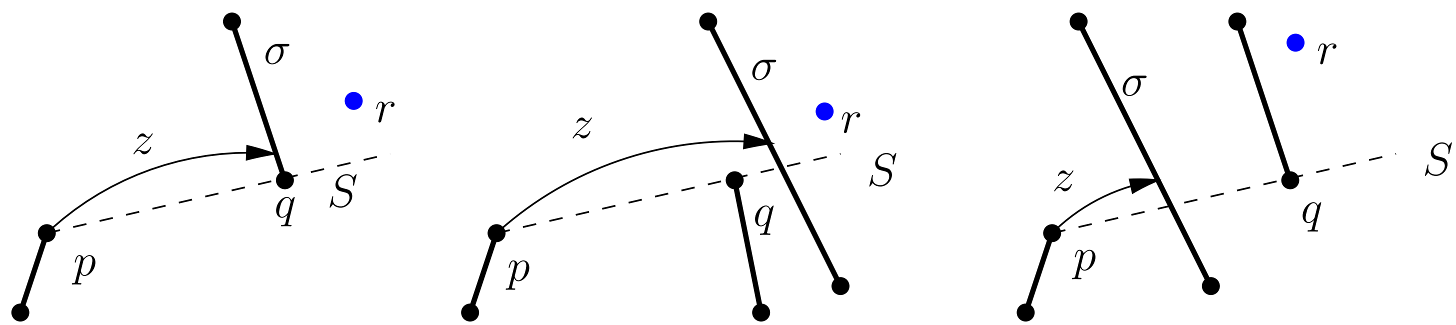


Sweep mit Bearbeitungsreihenfolge

Bearbeitung des nächsten Paares (p, r) nach (p, q)

F. 1) r liegt hinter σ

- Zeiger bleibt auf σ
- Invariante gilt nun für (p, r)



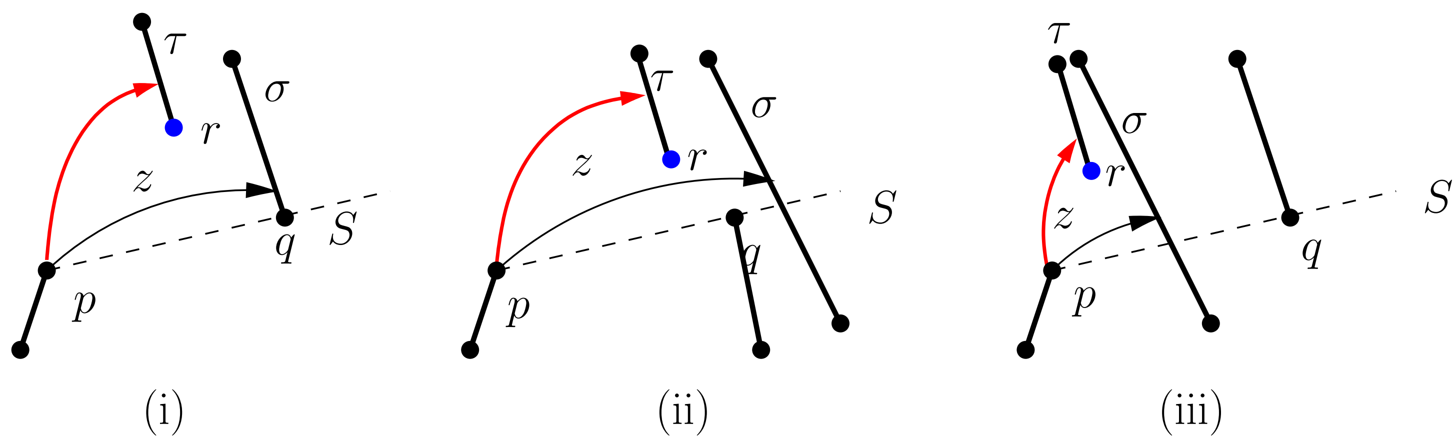
(i) (ii) (iii)

Sweep mit Bearbeitungsreihenfolge

Bearbeitung des nächsten Paares (p, r) nach (p, q)

F. 2) r liegt vor σ, Anfangspunkt von τ

- Zeiger auf τ setzen
- Ausgabe: (p, r)! Invariante gilt nun für (p, r)

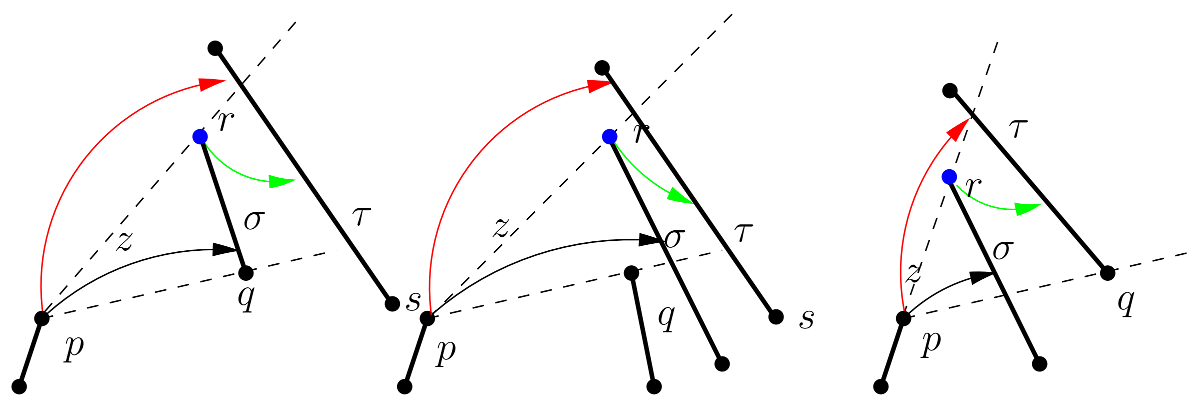


Sweep mit Bearbeitungsreihenfolge

Bearbeitung des nächsten Paares (p, r) nach (p, q)

F. 3) r ist Endpunkt von σ

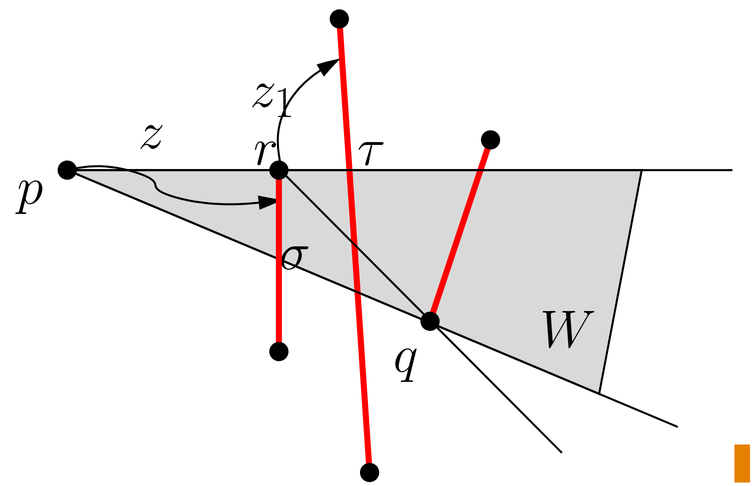
- Zeiger auf τ von r: (r, s) war schon dran Reihenfolge
- Winkelbereich ist leer!!
- Ausgabe: (p, r)! Invariante gilt nun für (p, r)



(i) (ii) (iii)

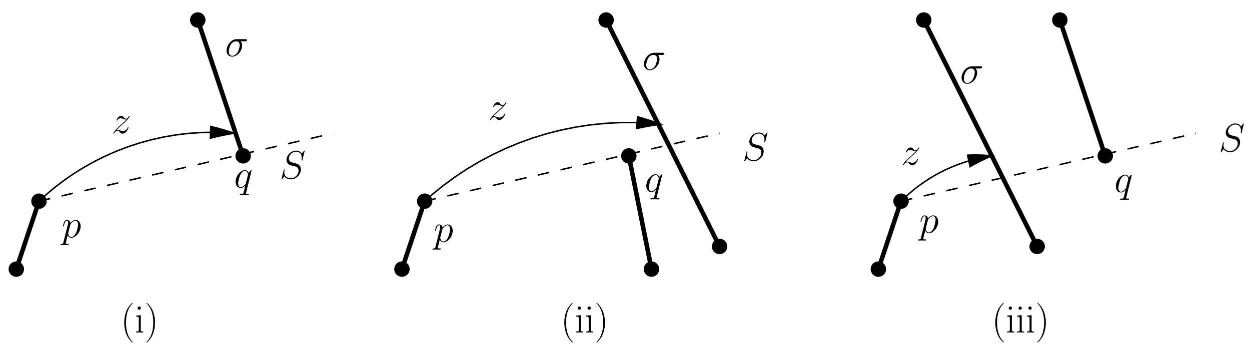
Simultaner Sweep: Ereignisverarbeitung Fall 3(iii)

- r Endpunkt von σ
- Winkelbereich W: Punktfrei
- (r, q) war dran: Umbiegen d. Zeigers/Ausgabe, $O(1)$

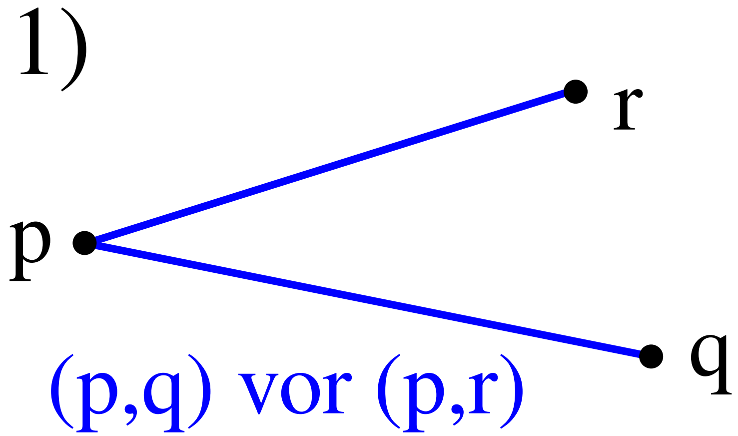


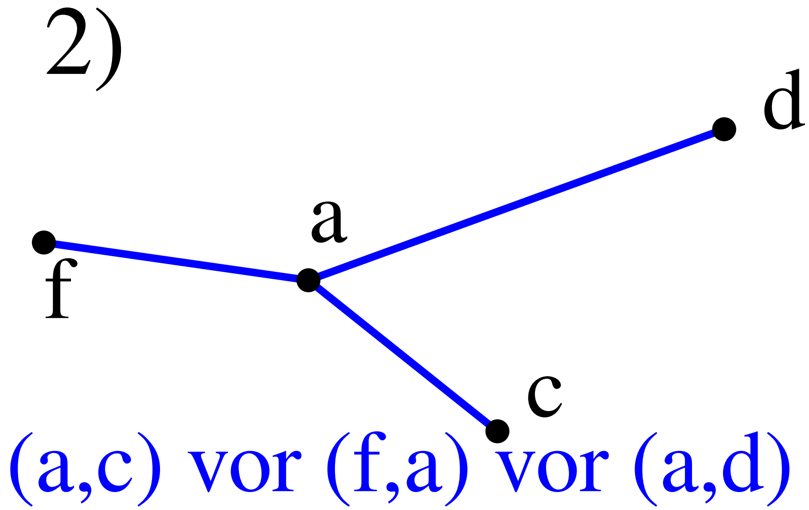
Simultaner Sweep: Laufzeit und Korrektheit

- Invariante ist stets erfüllt! Zeiger auf σ korrekt
- Bearbeitungsreihenfolge: Alle Paare (p, q) werden betrachtet!
- Immer wenn q auf σ liegt erfolgt Ausgabe
- Bearbeitungsreihenfolge: $O(n^2)$ Ereignisse, jeweils $O(1)$ Aufwand
- Korrekter Sweep in $O(n^2)$



Bearbeitungsreihenfolge in $O(n^2)$



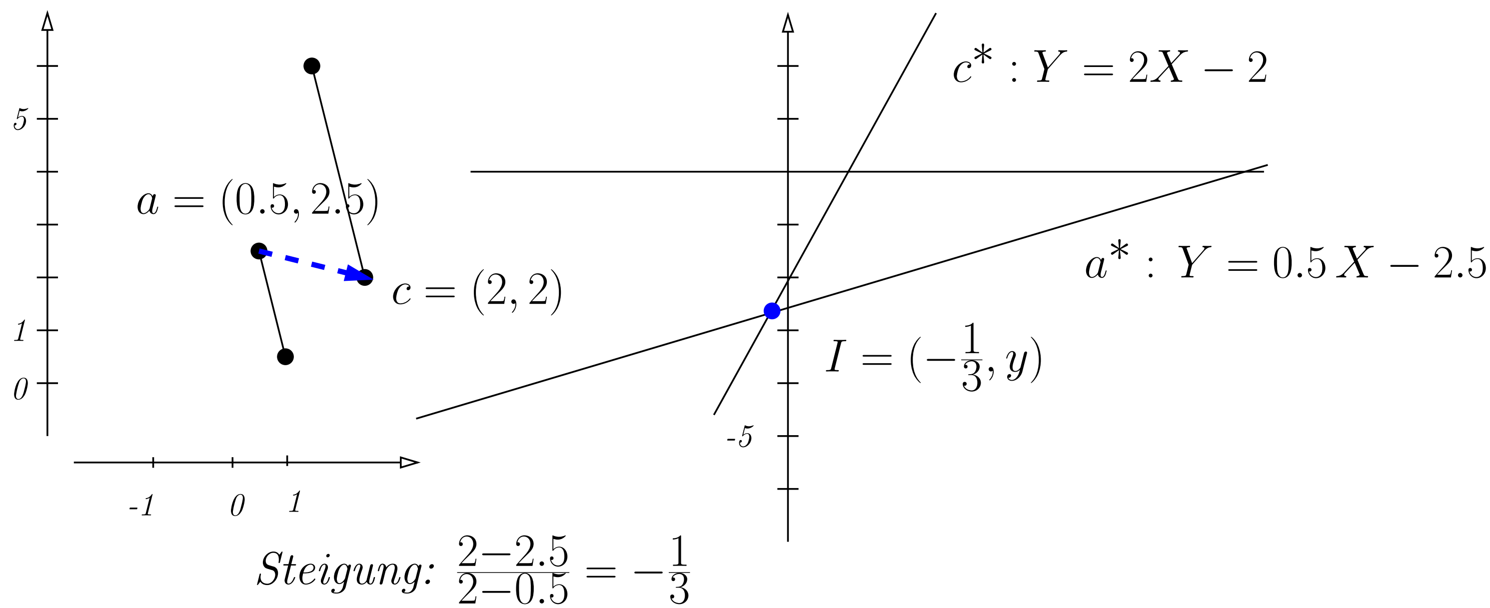


Offensichtlich wird nicht mehr gebraucht!

Diese muss nun berechnet werden!

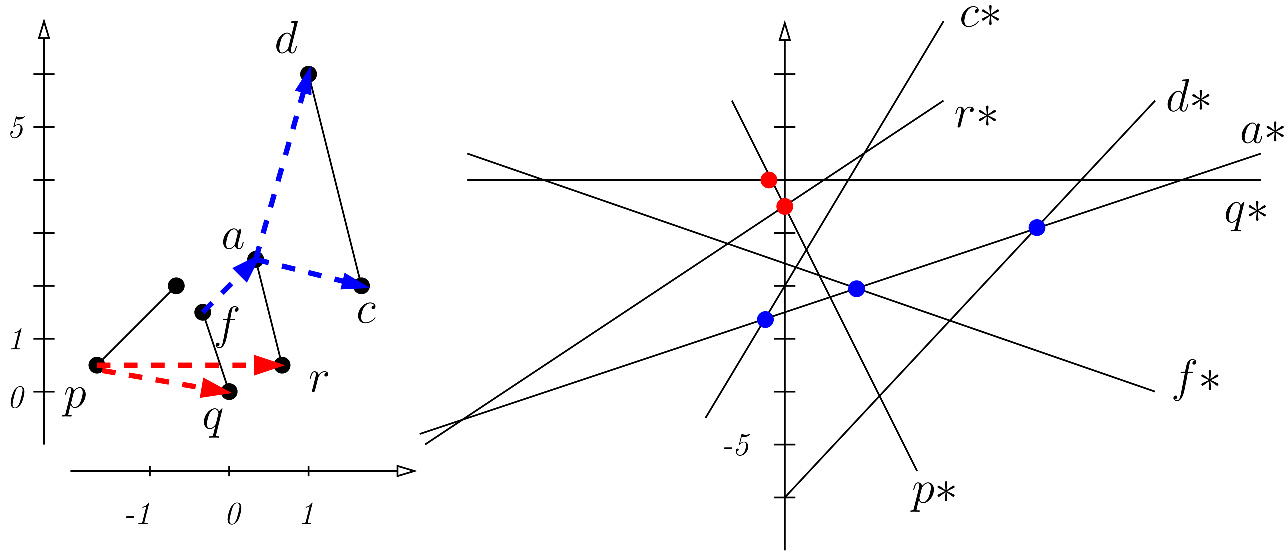
Dualitätsprinzip

1. Ordnungserhaltende Abbildung zwischen geometrischen Objekten
2. Transformiere Punkt $p = (p_x, p_y)$ zu $p^* : \{Y = p_x X - p_y\}$
3. Steigung zw. zwei Punkten $p = (p_x, p_y)$ und $q = (q_x, q_y)$, $p_x < q_x$
4. Ergebnis $\frac{q_y - p_y}{q_x - p_x}$! Entspricht X-Koordinate von $p^* \cap q^*$



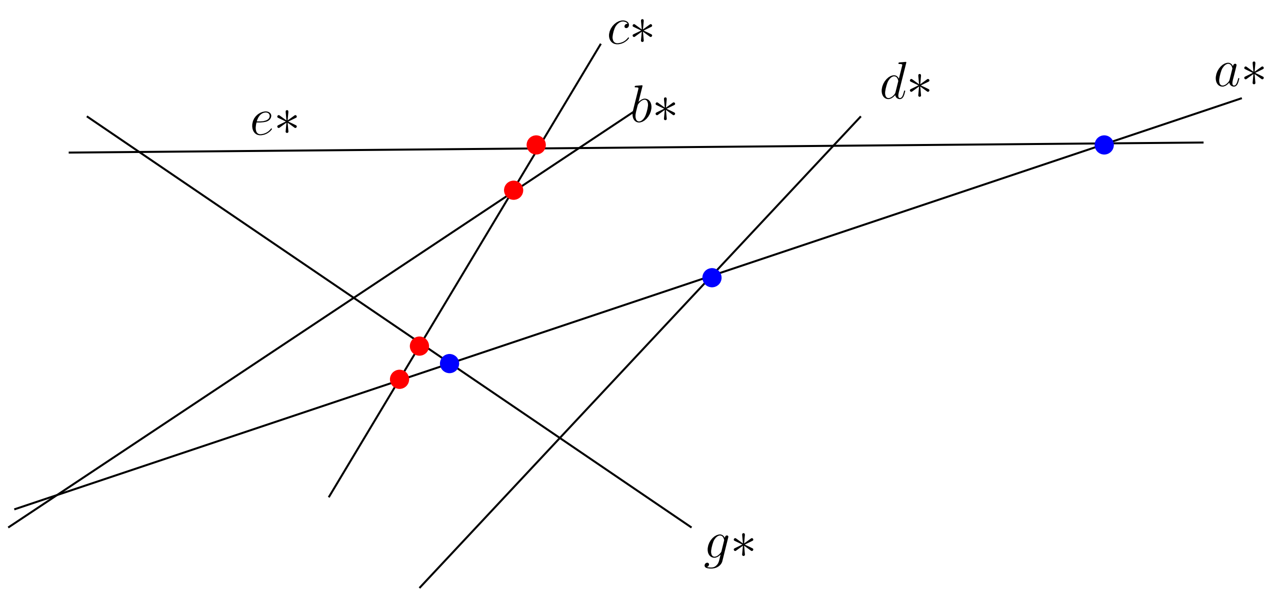
Dualitätsprinzip

1. Transformiere alle Punkte $p = (p_x, p_y)$ zu $p^* : \{Y = p_x X - p_y\}$
2. Bearbeitungsreihenfolge: 2 Bedingungen
3. Reihenfolge der Schnittpunkte entlang jeder Geraden einhalten



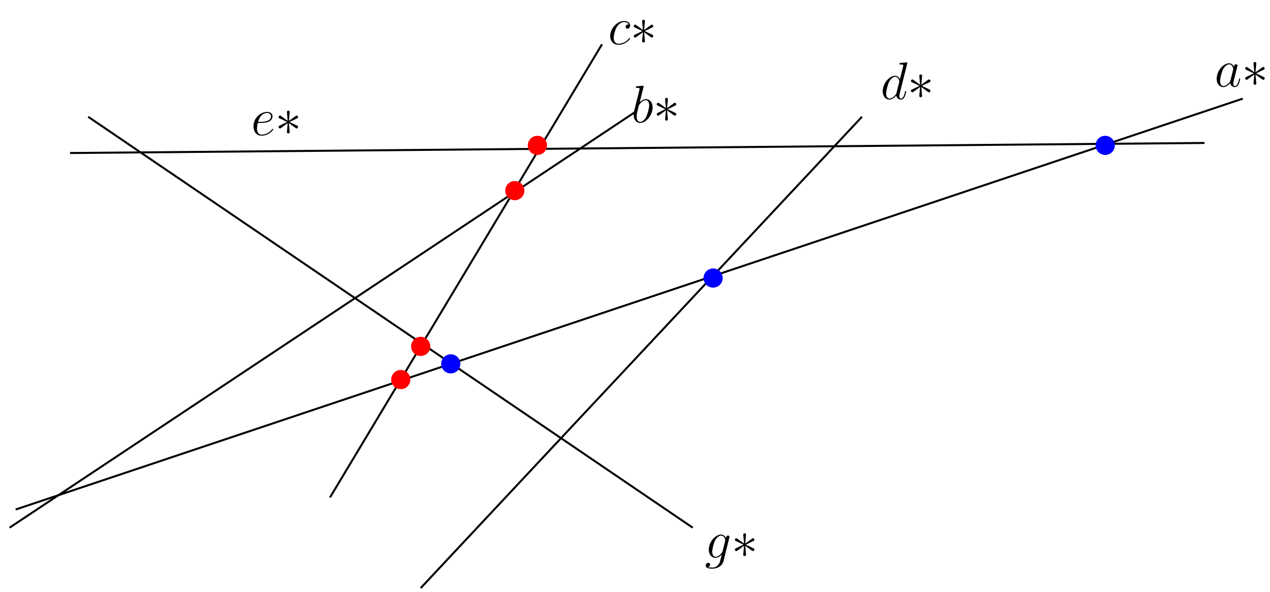
Arrangement von Geraden

- Berechne die Reihenfolge der Schnittpunkte entlang jeder Geraden
- Nicht eindeutig!!
- $a^* \cap c^*$, $g^* \cap c^*$, $g^* \cap a^*$, $c^* \cap b^*, a^* \cap d^*,$ $c^* \cap e^*$, $a^* \cap e^*$



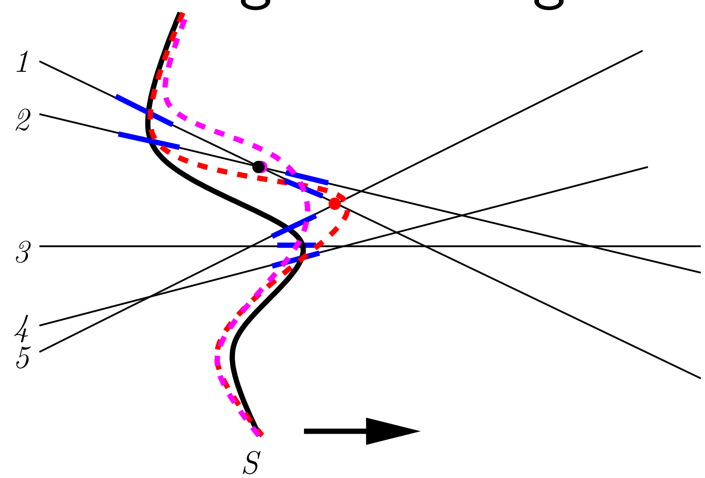
Neue Aufgabe für Arrangement von Geraden

- Berechne Schnittpunkte gemäß Ordnung entlang jeder Geraden
- Naiv! n^2 Schnittpunkte sortieren: $O(n^2 \log n)$
- Sweep mit Sweep-Gerade: $O((n+k)\log n)$, $k \in \Omega(n^2)$
- Idee: Keine volle Ordnung, Sweep mit Pseudogeraden



Sweep mit Pseudogeraden

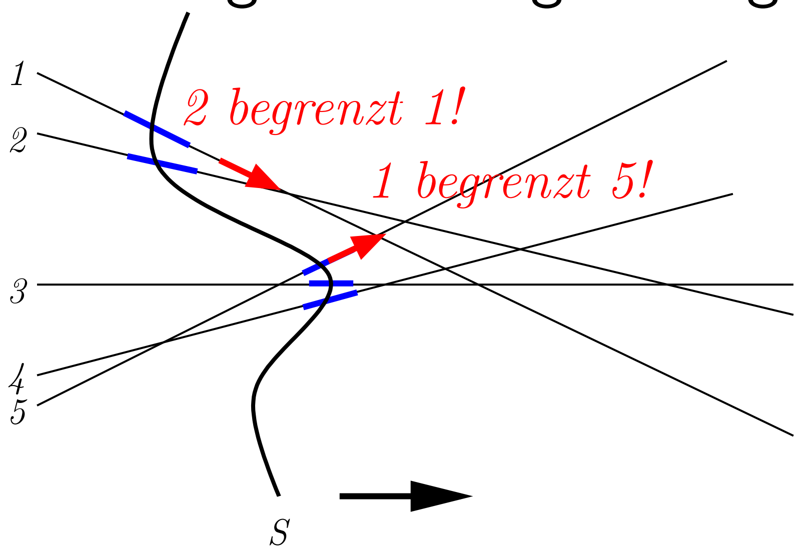
- **Def. 1.2:** Pseudogerade: Kurve schneidet jede Gerade genau einmal
- Schnittpunkte gemäß Ordnung entlang aller Geraden
- Hebe Pseudogerade über angrenzenden Schnittpunkt: Ausgabe!
- Welche Möglichkeiten?
- Pseudogerade Bed. $\Leftrightarrow$ Ausgabe richtige Reihenfolge



$L=(1,2,5,3,4)$ $L=(2,1,5,3,4)$

Sweep mit Pseudogeraden

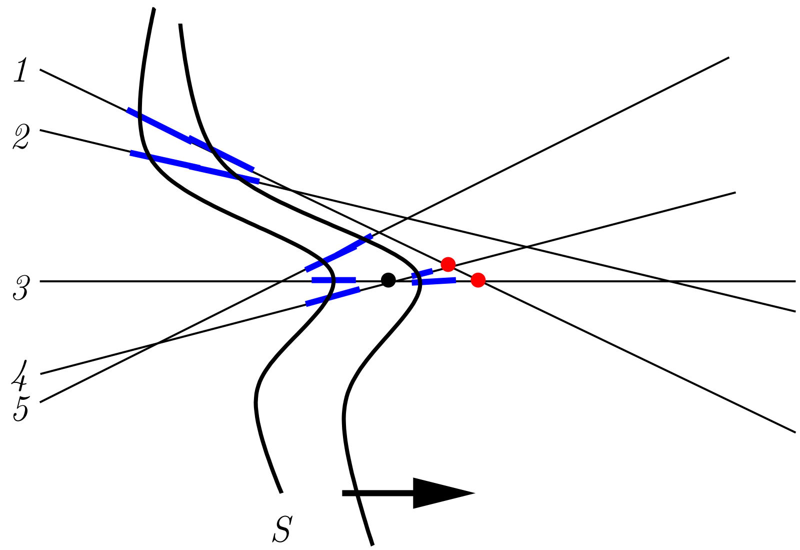
- Jede Gerade genau einmal schneiden, nächster SP
- Bedingung einhalten ⇔ Reihenfolge SP entlang jeder Geraden
- Auswahl? Begrenzungen ermitteln
- Wodurch wird Gerade nach rechts begrenzt? Liste N
- Kandidaten mit gleicher Begrenzung sind genau richtig!!



$L=(1,2,5,3,4)$ $N=((1,2),(2,1),(5,1),(3,4),(4,3))$

Neue Aufgabe: Begrenzungen ermitteln!

- Kandidatenliste N!
- Paare mit gleicher Beschränkung speziell markieren (Zeiger)
- Liste N aktualisieren bei nächstem Vorgang
- Neue Beschränkung zwei Geraden, mit spezieller Datenstruktur

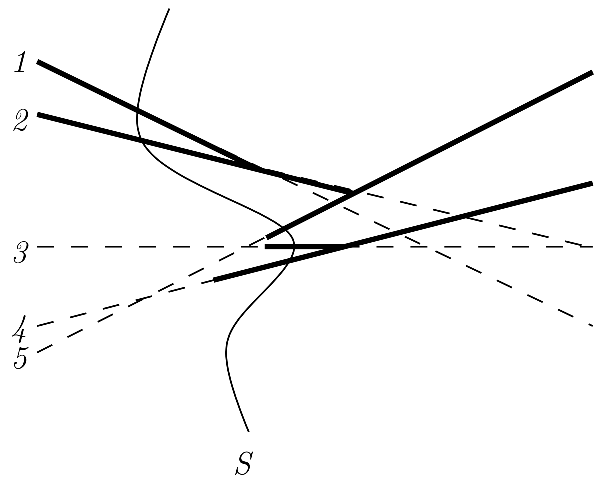


$N_{alt}=((1,2),(2,1),(5,1),(3,4),(4,3))$

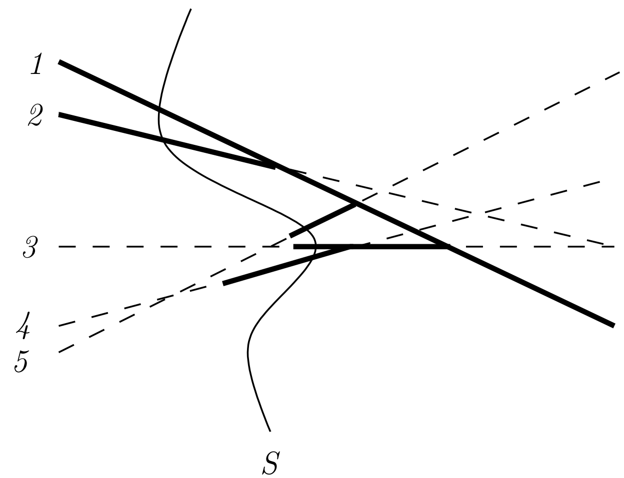
$N=((1,2),(2,1),(5,1),(4,1),(3,1))$

Datenstruktur Horizontbäume

- Arrangement und Pseudogerade mit Ordnung
- Verlängere die Geraden stets noch Oben/Unten
- Immer an nächstem Schnittpunkt
- Baumstruktur (kann zerfallen) Lineare Komplexität



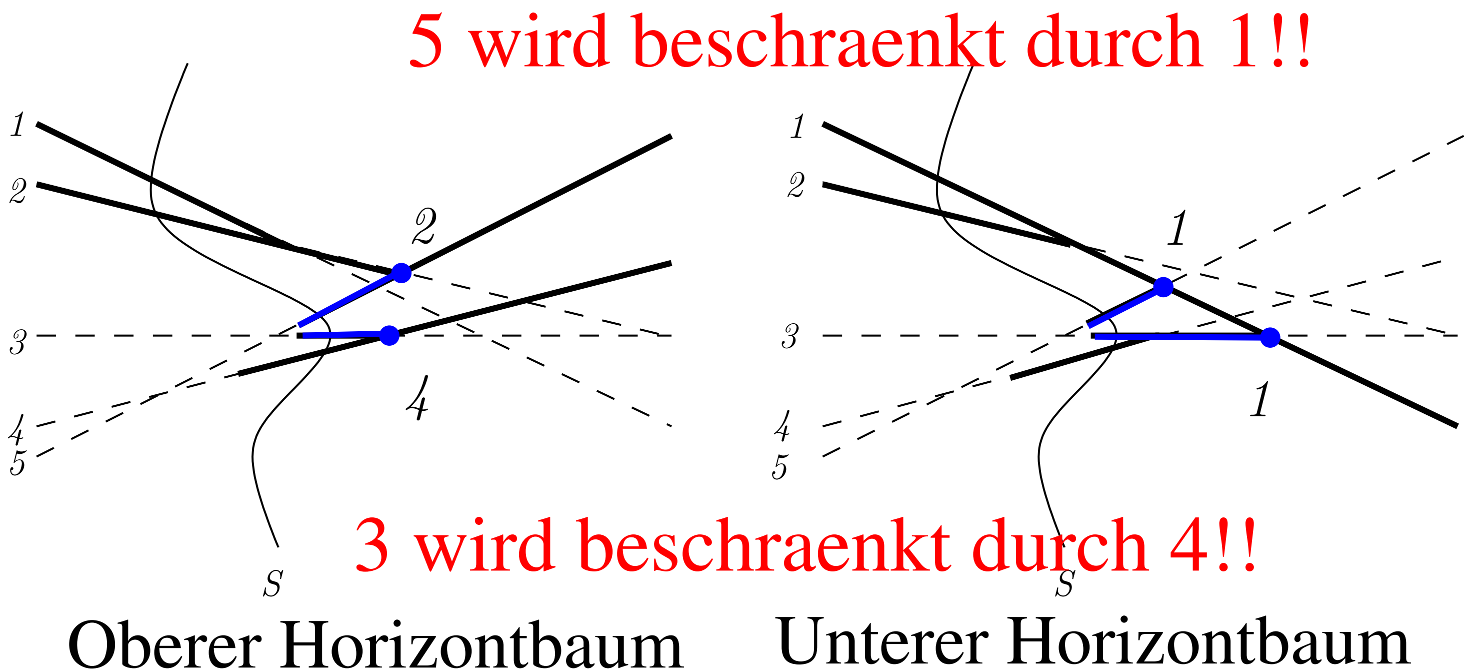
Oberer Horizontbaum



Unterer Horizontbaum

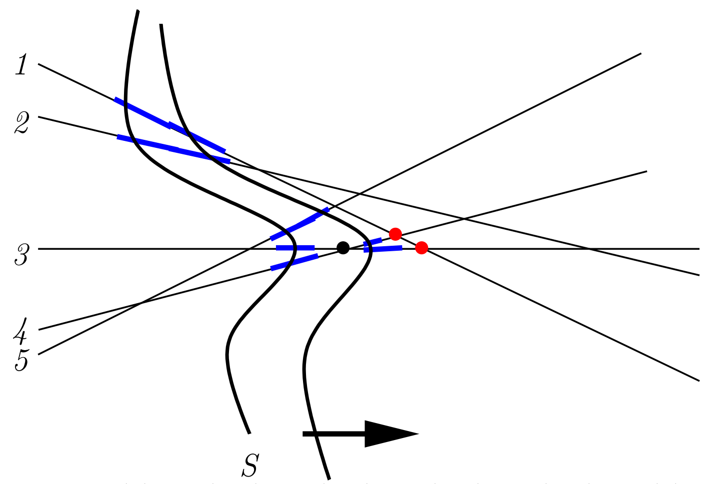
Neue Begrenzungen durch Horizontbäume ermitteln

- Oberer Baum/Unterer Baum gegeben
- *Minimum* ergibt begrenzende Gerade (Label im Baum)
- Bezüglich neuer Kanten den Bäumen folgen



Neue Aufgabe: Begrenzungen ermitteln!

- Kandidatenliste N!
- Paare mit gleicher Beschränkung speziell markieren (Zeiger)
- Liste N aktualisieren bei nächstem Vorgang
- Neue Beschränkung zwei Geraden, mit spezieller Datenstruktur
- Durch Aktualisierung von T^o und T_u

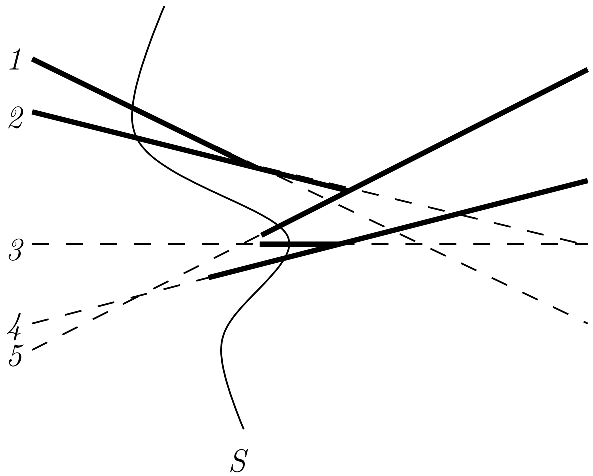


$N alt = ((1,2),(2,1),(5,1),(3,4),(4,3))$

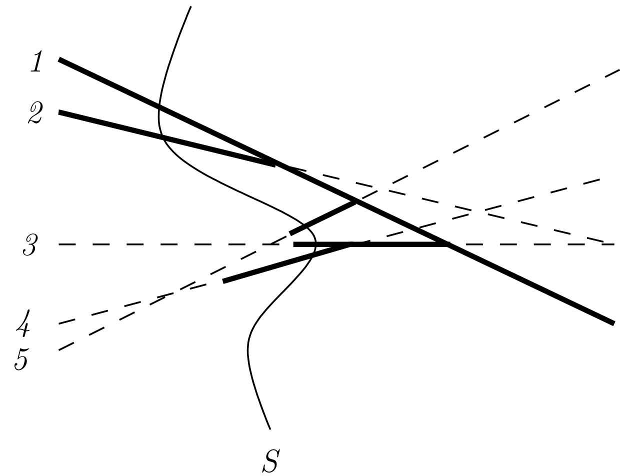
$N=((1,2),(2,1),(5,1),(4,1),(3,1))$

Datenstruktur Horizontbäume

- Für Pseudogerade mit Ordnung
- Oberer und Unterer Horizontbaum gegeben
- Hebe über nächsten erlaubten SP
- Aktualisiere die Bäume



Oberer Horizontbaum



Unterer Horizontbaum