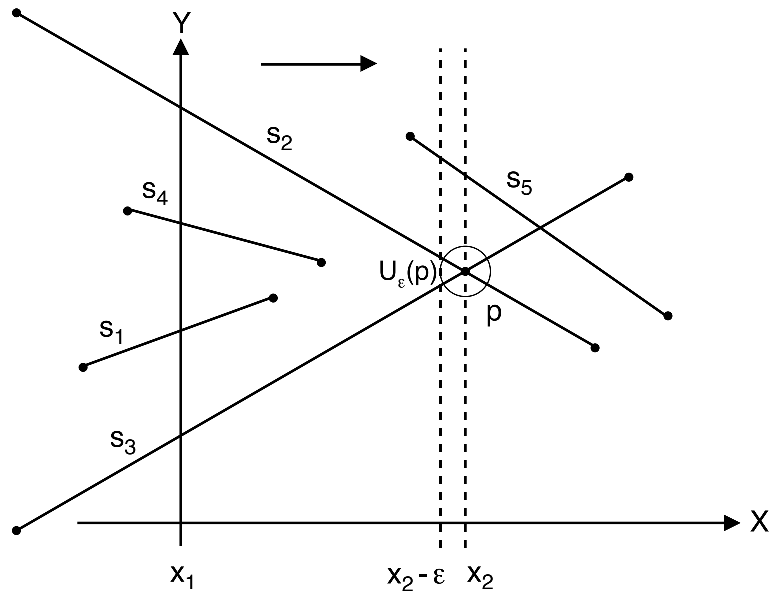
Sweep Verfahren Forsetzung

Elmar Langetepe
University of Bonn

SSS: Schnittpunkte von Liniensegmenten

SSS: Schnittpunkte von Liniensegmenten



SSS: Segmente gemäß Ordnung, Ereignisse:

1. Neuer linker Endpunkt
2. Neuer rechter Endpunkt
3. Schnittpunkt

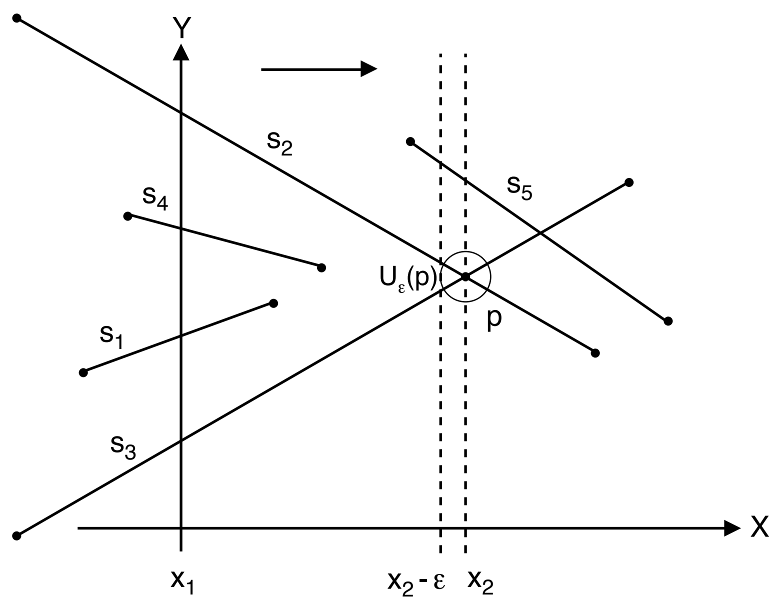
SSS: Strukturelle Eigenschaft

SSS: Strukturelle Eigenschaft

Lemma 2.7 Haben zwei Liniensegmente einen echten Schnittpunkt zum Zeitpunkt t, dann sind die Segmente unmittelbar vor t Nachbarn in der SSS gewesen.

SSS: Strukturelle Eigenschaft

Lemma 2.7 Haben zwei Liniensegmente einen echten Schnittpunkt zum Zeitpunkt t, dann sind die Segmente unmittelbar vor t Nachbarn in der SSS gewesen.



Lösung des Existenzproblems: Operationen

Lösung des Existenzproblems: Operationen

FügeEin(*SSS*,*Seg*,*x*): Fügt Segment *Seg* entsprechend der Ordnung zur Zeit x in die *SSS* ein.

Lösung des Existenzproblems: Operationen

FügeEin(*SSS*,*Seg*,*x*):	Fügt Segment *Seg* entsprechend der Ordnung zur Zeit x in die *SSS* ein.
Entferne(*SSS*,*Seg*,*x*):	Entfernt Segment *Seg* aus der *SSS* zum Zeitpunkt x.

Lösung des Existenzproblems: Operationen

FügeEin(*SSS*,*Seg*,*x*):	Fügt Segment *Seg* entsprechend der Ordnung zur Zeit x in die *SSS* ein.
Entferne(*SSS*,*Seg*,*x*):	Entfernt Segment *Seg* aus der *SSS* zum Zeitpunkt x.
Vorg(*SSS*,*Seg*,*x*):	Bestimmt den Vorgänger von *Seg* in der *SSS* zum Zeitpunkt x.

Lösung des Existenzproblems: Operationen

FügeEin(*SSS*,*Seg*,*x*):	Fügt Segment *Seg* entsprechend der Ordnung zur Zeit x in die *SSS* ein.
Entferne(*SSS*,*Seg*,*x*):	Entfernt Segment *Seg* aus der *SSS* zum Zeitpunkt x.
Vorg(*SSS*,*Seg*,*x*):	Bestimmt den Vorgänger von *Seg* in der *SSS* zum Zeitpunkt x.
Nachf(*SSS*,*Seg*,*x*):	Bestimmt den Nachfolger von *Seg* in der *SSS* zum Zeitpunkt x.

Lösung des Existenzproblems: Operationen

FügeEin(*SSS*,*Seg*,*x*):	Fügt Segment *Seg* entsprechend der Ordnung zur Zeit x in die *SSS* ein.
Entferne(*SSS*,*Seg*,*x*):	Entfernt Segment *Seg* aus der *SSS* zum Zeitpunkt x.
Vorg(*SSS*,*Seg*,*x*):	Bestimmt den Vorgänger von *Seg* in der *SSS* zum Zeitpunkt x.
Nachf(*SSS*,*Seg*,*x*):	Bestimmt den Nachfolger von *Seg* in der *SSS* zum Zeitpunkt x.
SchnittTest(*Seg1*,*Seg2*):	Teste auf Schnitt zw. Seg1 und Seg2.

Lösung des Existenzproblems: Operationen

FügeEin(*SSS*,*Seg*,*x*):	Fügt Segment *Seg* entsprechend der Ordnung zur Zeit x in die *SSS* ein.
Entferne(*SSS*,*Seg*,*x*):	Entfernt Segment *Seg* aus der *SSS* zum Zeitpunkt x.
Vorg(*SSS*,*Seg*,*x*):	Bestimmt den Vorgänger von *Seg* in der *SSS* zum Zeitpunkt x.
Nachf(*SSS*,*Seg*,*x*):	Bestimmt den Nachfolger von *Seg* in der *SSS* zum Zeitpunkt x.
SchnittTest(*Seg1*,*Seg2*):	Teste auf Schnitt zw. Seg1 und Seg2.

SSS, balancierter Binärbaum, jede Operation in $O(\log n)$

Lösung des Existenzproblems

Lösung des Existenzproblems

Theorem 2.8 Man kann in Zeit $O(n \log n)$ und mit Speicherplatz $O(n)$ ermitteln, ob von n Liniensegmenten in der Ebene mindestens zwei einen echten Schnittpunkt haben.

Lösung des Existenzproblems

Theorem 2.8 Man kann in Zeit $O(n \log n)$ und mit Speicherplatz $O(n)$ ermitteln, ob von n Liniensegmenten in der Ebene mindestens zwei einen echten Schnittpunkt haben.

- Segment Einfügen: $O(\log n)$
- Segment Entfernen: $O(\log n)$
- Schnitt Test: $O(1)$

Lösung des Existenzproblems

Theorem 2.8 Man kann in Zeit $O(n \log n)$ und mit Speicherplatz $O(n)$ ermitteln, ob von n Liniensegmenten in der Ebene mindestens zwei einen echten Schnittpunkt haben.

- Segment Einfügen: $O(\log n)$
- Segment Entfernen: $O(\log n)$
- Schnitt Test: $O(1)$

Nun Aufzählungsproblem behandeln! Wechsel der Ordung durch SP

Ereignisstruktur, Initialisierung

Ereignisstruktur, Initialisierung

- Priorityqueue

Ereignisstruktur, Initialisierung

- Priorityqueue
- Ereignisse: Schnittpunkt, Linker Endpunkt, Rechter Endpunkt

Ereignisstruktur, Initialisierung

- Priorityqueue
- Ereignisse: Schnittpunkt, Linker Endpunkt, Rechter Endpunkt
- Nächstes Ereignis

Ereignisstruktur, Initialisierung

- Priorityqueue
- Ereignisse: Schnittpunkt, Linker Endpunkt, Rechter Endpunkt
- Nächstes Ereignis
- Ereignis Einfügen

Ereignisstruktur, Initialisierung

- Priorityqueue
- Ereignisse: Schnittpunkt, Linker Endpunkt, Rechter Endpunkt
- Nächstes Ereignis
- Ereignis Einfügen

(* Initialisierung SSS und ES*)
Initialisiere die Strukturen *SSS* und *ES*;
sortiere die $2n$ Endpunkte nach aufsteigenden X-Koordinaten;
erzeuge daraus Ereignisse;
füge diese Ereignisse in die *ES* ein;

Sweep Liniensegmente

```
(* Sweep und Ausgabe *)
while ES ≠ ∅ do
    Ereignis := NächstesEreignis(ES);
    with Ereignis do
        case Typ of
            LinkerEndpunkt:
                FügeEin(SSS, Seg, Zeit);
                VSeg := Vorg(SSS, Seg, Zeit);
                TesteSchnittErzeugeEreignis(VSeg, Seg);
                NSeg := Nachf(SSS, Seg, Zeit);
                TesteSchnittErzeugeEreignis(Seg, NSeg);
```

RechterEndpunkt:

$VSeg := Vorg(SSS, Seg, Zeit)$;

$NSeg := Nachf(SSS, Seg, Zeit)$;

$Entferne(SSS, Seg, Zeit)$;

$TesteSchnittErzeugeEreignis(VSeg, NSeg)$;

Schnittpunkt:

Berichte ($USeg$, $OSeg$) als Paar mit Schnitt;

$Vertausche(SSS, USeg, OSeg, Zeit)$;

$VSeg := Vorg(SSS, OSeg, Zeit)$;

$TesteSchnittErzeugeEreignis(VSeg, OSeg)$;

$NSeg := Nachf(SSS, USeg, Zeit)$;

$TesteSchnittErzeugeEreignis(USeg, NSeg)$

Analyse Sweep

Analyse Sweep

Theorem 2.9 Das Sweep-Verfahren berechnet die k echten Schnittpunkte von n Liniensegmenten in Zeit $O((n+k)\log n)$ mit Platz $O(k+n)$.

Beweis!

Analyse Sweep

Theorem 2.9 Das Sweep-Verfahren berechnet die k echten Schnittpunkte von n Liniensegmenten in Zeit $O((n+k)\log n)$ mit Platz $O(k+n)$.

Beweis!

Theorem 2.10 Das Sweep-Verfahren berechnet die k echten Schnittpunkte von n Liniensegmenten in Zeit $O((n+k)\log n)$ mit Platz $O(n)$.

Analyse Sweep

Theorem 2.9 Das Sweep-Verfahren berechnet die k echten Schnittpunkte von n Liniensegmenten in Zeit $O((n+k)\log n)$ mit Platz $O(k+n)$.

Beweis!

Theorem 2.10 Das Sweep-Verfahren berechnet die k echten Schnittpunkte von n Liniensegmenten in Zeit $O((n+k)\log n)$ mit Platz $O(n)$.

Zusammenfassung: Auch degenerierte Fälle können behandelt werden