3.2.11. Geometrische Primitive (Prädikate Operatoren).

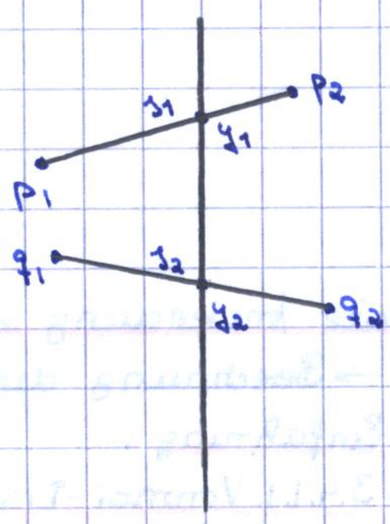
1. Schnitt von zwei Segmenten:
   - Test + Berechnung
2. Lineare Ordnung auf Y-Struktur.
   - abh. von xpos der SL
   - Datenstruktur (binärer blattorientierter Suchbaum)

     verwendet eine Fkt. compare zum Vergleich von zwei Segmenten.

     Suchbaum: lookup(s), compare(s, s')

     Mögl. Implementierung:

```
compare (s1, s2)   // berechne y-Koordinaten der Schnittpkte der Geradengleichungen v.
                   // s1 und s2 mit SL.
    y1 ← f1(xpos)
    y2 ← f2(xpos)
    wobei f1(x) = a1x + b1 ∧ f2(x) = a2x + b2  keine Vertikalen.
    return compare(y1, y2)
```



Übung: verwende nur orientation-Aufrufe.

Lsg: if (orientation (p1, p2, s2) > 0) ⇒ y1 > y2

if (orientation (q1, q2, s1) > 0) ⇒ y2 > y1

"=" gibt es nicht ...

3. Lineare Ordnung auf X-Struktur.

   = lexik. xy-Ordnung von Pkten.

   (Vergleich zweier Pkte p und q)

   → compare (p, q)

```
Δx ← (q.xcord() - p.xcord());
if (Δx ≠ 0) then
    return sign (Δx)
else
    return sign (q.ycord() - p.ycord).
```

4. Test, ob Pkt p rechts von SL

   trivial

   if (SL, p, p.xcord) < 0 ⇒ rechts von SL

   > 0 links

   = 0 auf.

3.2.12 Bem:

Bisher Segmente in allgemeiner Lage

a) Keine drei Segmente schneiden sich in einem Pkt

b) Alle Eventpkte haben verschiedene x-Koordinaten.

c) Keine vertikalen Segmente.

In der Praxis unrealistisch!

⇝ Log: Modifizierter Alg., der mit degenerierten Eingaben umgehen kann.

→ siehe Übung dazu.

3.2.13. Varianten des Problems:

⟶ 3.2.13.1. Red/Black-Intersection Problem:

Haben zwei Mengen von Segmenten S_1 und S_2

Aufgabe ist: Berechne Schnittpkte $\sin s_2 \; \forall s_1 \in S_1 \wedge \forall s_2 \in S_2$

⟶ 3.2.13.2. Kurvensegmente:

z.B. Kreisbögen

→ Übung

⟶ 3.2.13.3 Berechnung der planaren Unterteilung der Ebene

→ planarer Graph.

G = (V, E)

V = Endpkte und Schnittpkte

E = Intervalle durch Zerlegung der Segmente durch V