2.2. Anwendung der hierarchischen Darstellung:

→ verwenden immer dieselbe Strategie:

2.2.1. Strategie:

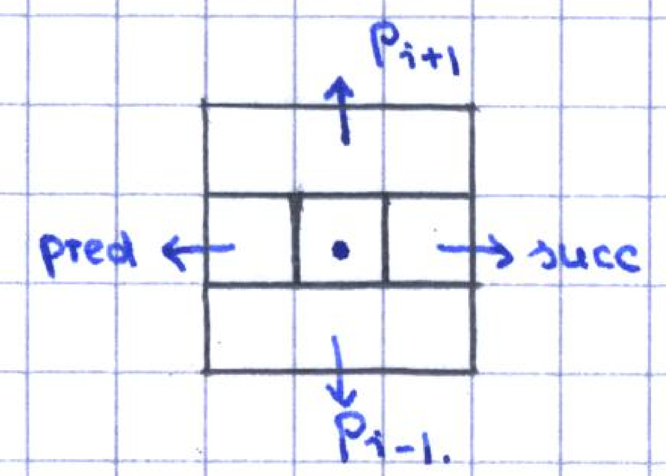
1). Wir berechnen eine Annäherung der Lösung für P_0.
Da P_0 maximal vier Ecken ⇒ in $O(1)$.

2). Dann durchlaufe die Folge $P_0, P_1 .. P_k$ und verfeinere die Lösung schrittweise $P_i \to P_{i+1}$.

3) Am Ende haben wir die Lsg für $P_k = P$.

2.2.2. Darstellung im Rechner:

- P_i, $0 \leq i \leq k$ als doppelt verkettete Liste der Ecken und jede Ecke von P_i zeigt zusätzlich auf ihre Kopie in P_{i+1} und umgekehrt.

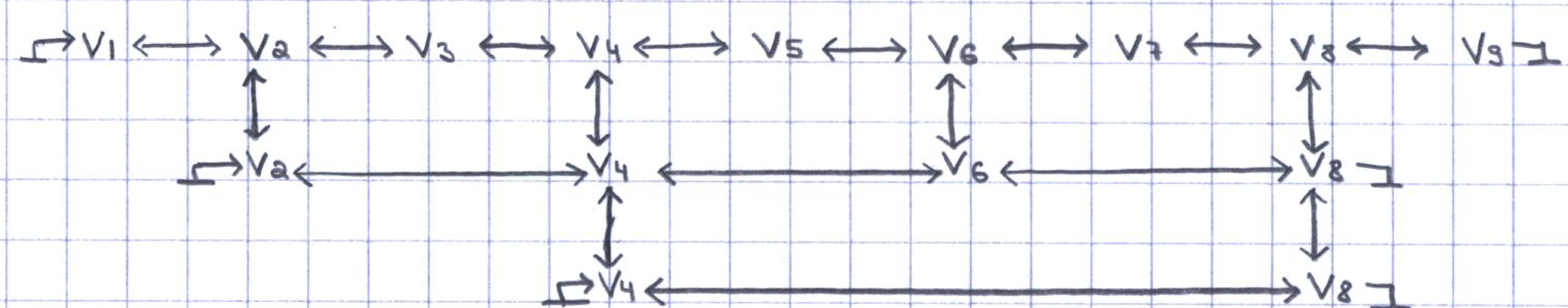


- explizite Speicherung des Kantenbaumes
→ Dynamisierung (Ecken einfügen / löschen)
(wie bei Suchbäumen).

2.2.3 Beispiel:

Sei P gegeben wie in 2.1.9.

⇒



2.2.4. Anwendung: Schnitt mit einer Geraden:

2.2.4.1. Ziel:

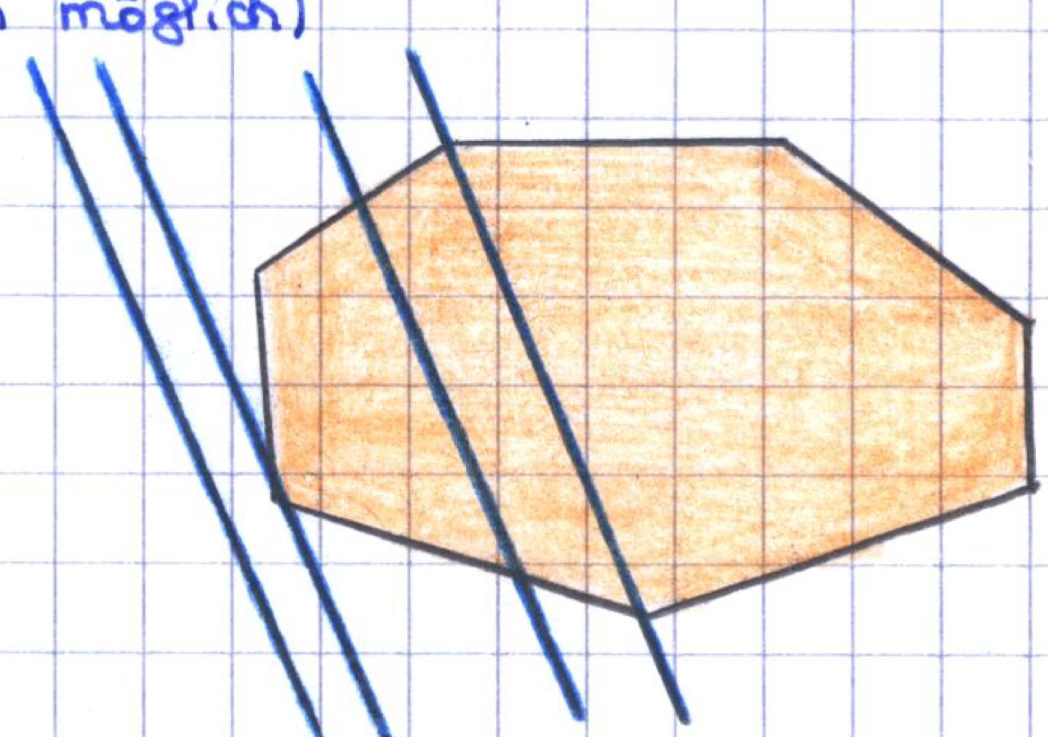
Geg: Hierarchische Darstellung $P_0 \ldots P_k$ eines konvexen Polygons P und eine Gerade g

Ausgabe: $P \cap g$

Konvexität ⇒ $P \cap g$ = Strecke (wenn nicht leer und Ecke)

⇒ Es genügt Schnittpkte a, b mit den Randsegmenten zu berechnen.

Fälle:
1) $a \neq b$ (allg. Fall, Ecken möglich)
2) $a = b$ (eine Ecke)
3) ex. nicht! $P \cap g = \emptyset$.



2.2.4.2. Idee für Algorithmus:

1). Schnitttest mit P_0:

2 Fälle:
a) $P_0 \cap g = \emptyset$
b) $P_0 \cap g \neq \emptyset$

Im Fall b) → STOP

Im Fall a):

sei $q_0 \in P_0$ mit minimalen Abstand zu g.