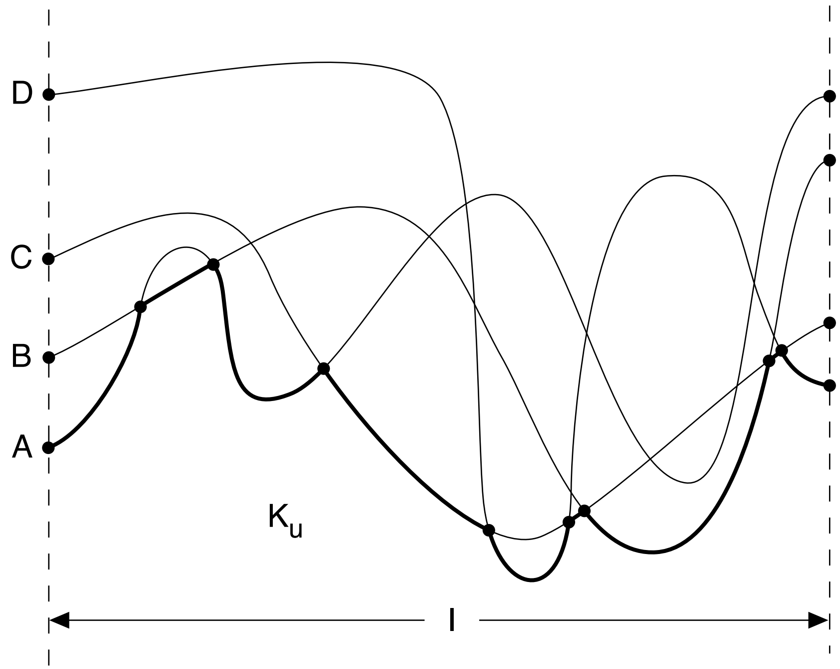
Zusammenfassung: Konturen, Datenstrukturen

Elmar Langetepe
University of Bonn

Kontur von Funktionen, Komplexität!



- Divide and Conquer mit Sweep im Merge-Schritt
- Sweep-Laufzeit entspricht Komplexität der Konturen K_u^1,K_u^2 und $(K^1 \cup K^2)_u$

Berechnung Kontur!

Theorem 2.13 Die untere Kontur von n verschiedenen X monotonen Wegen über einem gemeinsamen Intervall, von denen sich je zwei höchstens s-mal schneiden, kann in Zeit $O(\lambda_s(n) \log n)$ berechnet werden.

- Definition $\lambda_s(n)$: Maximale Komplexität der Kontur
- Lemma 2.12: Für $s, n \geq 1$ gilt $2\lambda_s(n) \leq \lambda_s(2n)$
- Abschätzen Alg.: $T(n) \in O(\lambda_s(n) \log n)$

$\lambda_s(n)$, kombinatorische Definition

Alphabet $\Sigma = \{A, B, C, \ldots\}$ über n Buchstaben.

Davenport–Schinzel–Sequenz der Ordnung s

- Wort w über Σ
- in w keine benachbarten Buchstaben gleich
- keine zwei verschiedenen Buchstaben wechseln mehr als s mal

Bsp.: ABRAKADABRA; max. 4 Wechsel (A und B)

Zu zeigen: $\lambda_s(n)$ ist maximale Länge eines solchen Wortes

Davenport-Schinzel-Sequenzen: Komplexität

Fast linear!! (Ohne Beweis!)

$$\begin{array}{rcl}
\lambda_1(n) & = & n \\
\lambda_2(n) & = & 2n-1 \\
\lambda_3(n) & \in & \Theta(n\,\alpha(n)) \\
\lambda_4(n) & \in & \Theta(n \cdot 2^{\alpha(n)}) \\
\lambda_s(n) & \in & O(n \log^*(n)) \in O(n^2)
\end{array}$$

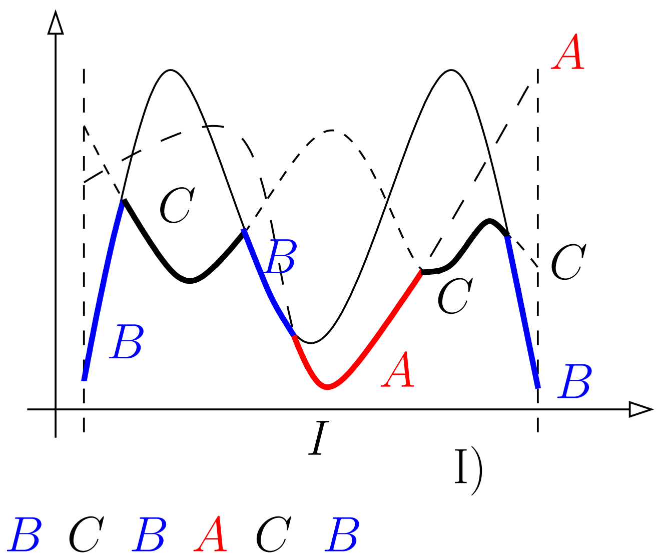
$\alpha(n)$ Inverse Ackermann Fkt.

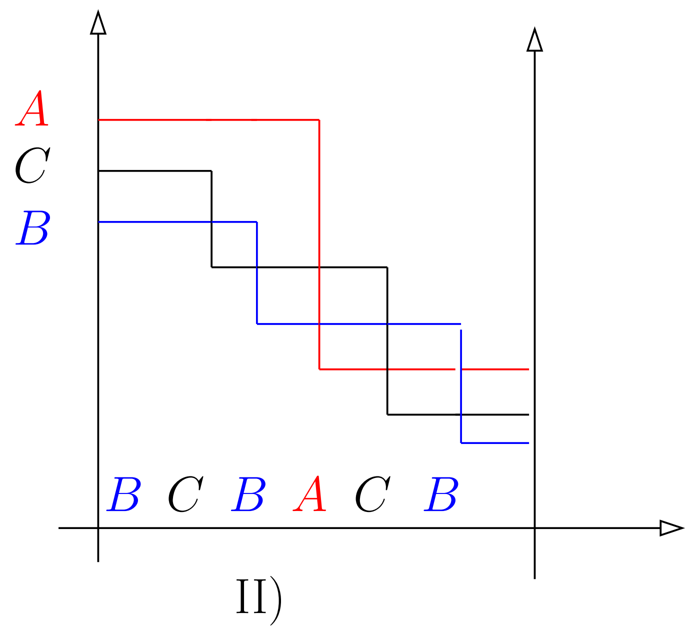
$\log^*(n)$

Beweis: Untere Kontur/DSS

Theorem 2.14 Die maximale Länge einer DSS der Ordnung s über n ist $\lambda_s(n)$.

I) Übersetzen in Buchst.-folge oder II) Buchst.-folge übersetzen





Ergebnisse

Korollar 2.15 Die untere Kontur von n Liniensegmenten über einem gemeinsamen Intervall kann in $O(n \log n)$ mit Platz $O(n)$ berechnet werden.

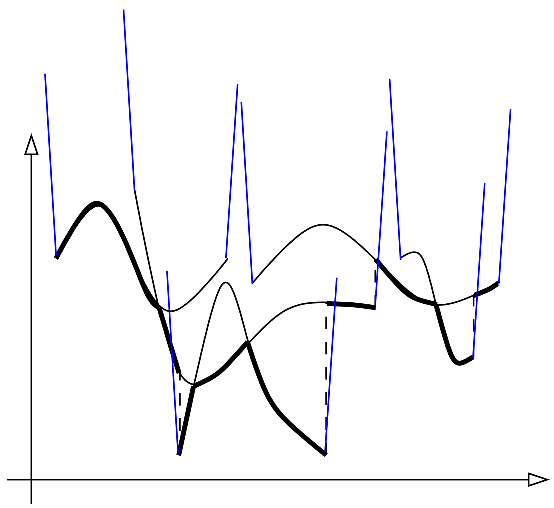
$\lambda_1(n) = n$

Funktionen über Teilintervalle

Aussage: Komplexität der Kontur von Funktionen die über gesamtem Intervall definiert sind 1) oder nur über ein Teilintervall 2).

1. $\lambda_s(n)$
2. $\lambda_{s+2}(n)$

2. Verlängern auf gesamtes Intervall, dann 1. verwenden, max. zwei zusätzliche Schnitte



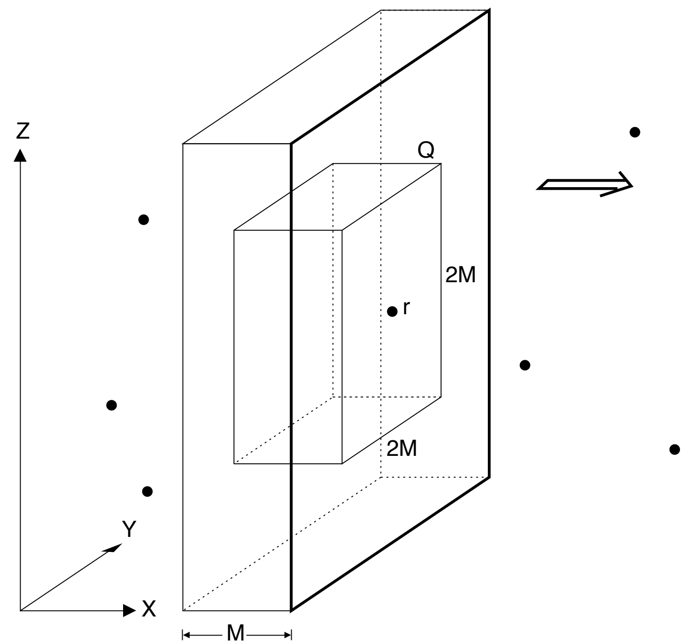
(2)

Ergebnisse

Korollar 2.16 Die untere Kontur von n Liniensegmenten beliebiger Länge enthält $\lambda_3(n)$ viele Segmente und kann in Zeit $O(\lambda_3(n) \log n)$ berechnet werden.

Weitere Sweepvarianten

- Sweep mit Halbgerader, Sweep mit Circle
- Sweep-Ebene, Sweep im Raum
- Closest-Pair: Laufzeit $O(n(e(n)+b(n)))$, Datenstrukturen

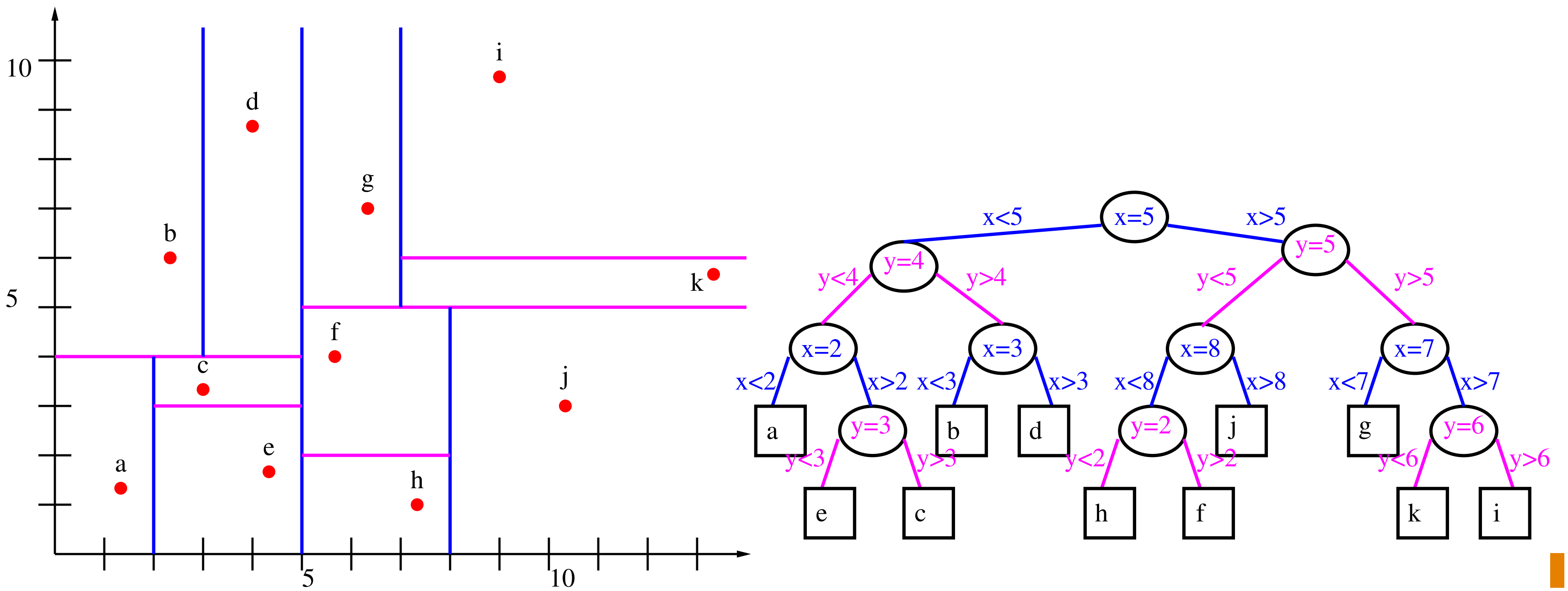


Geometrische Datenstruktur

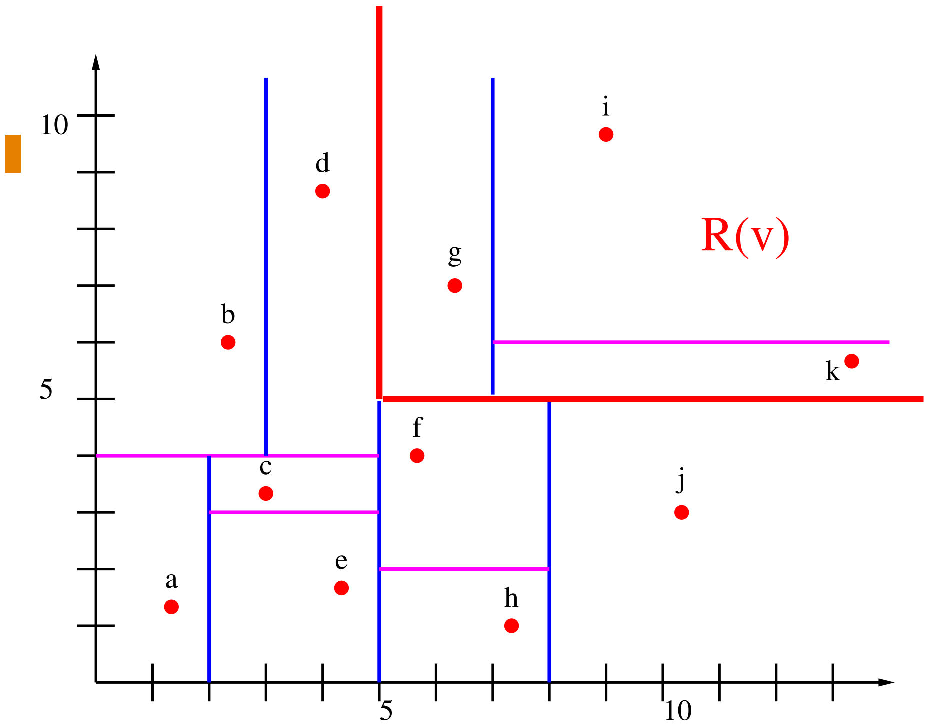
- Datenobjekte Punkte im $\mathbb{R}^d$, andere Objekte übertragen
- Bislang: Eindimensionaler balancierter Suchbaum
- Bereichsanfrage: Inklusionsanfrage, Schnittanfrage, Anfrageobj. q
- Interne, externe Datenstrukturen
- Statische, dynamische Datenstrukturen
- Rechteckige Bereichsanfrage im $\mathbb{R}^d$
- k-dimensionaler Suchbaum
- Achsenparalleles Rechteck q, Punkte im $\mathbb{R}^2$

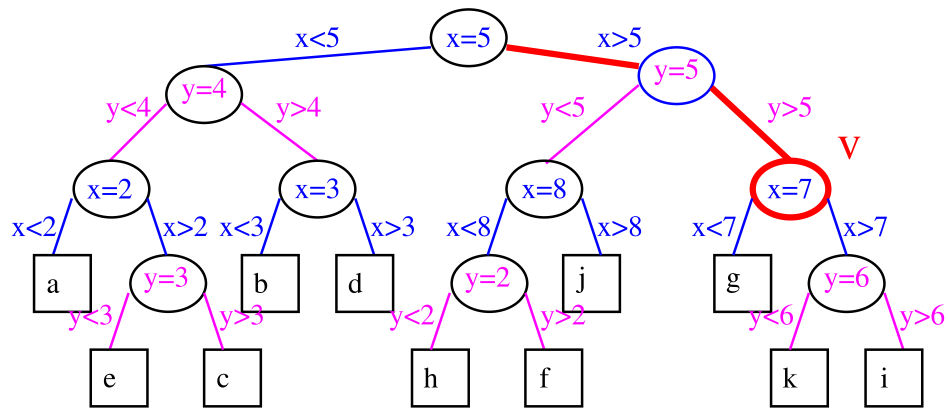
kd-Baum Aufbau

- Splitgeraden $X = s$, $Y = s$ im Wechsel!
- Binärer Baum mit entsprechenden Teilmengen



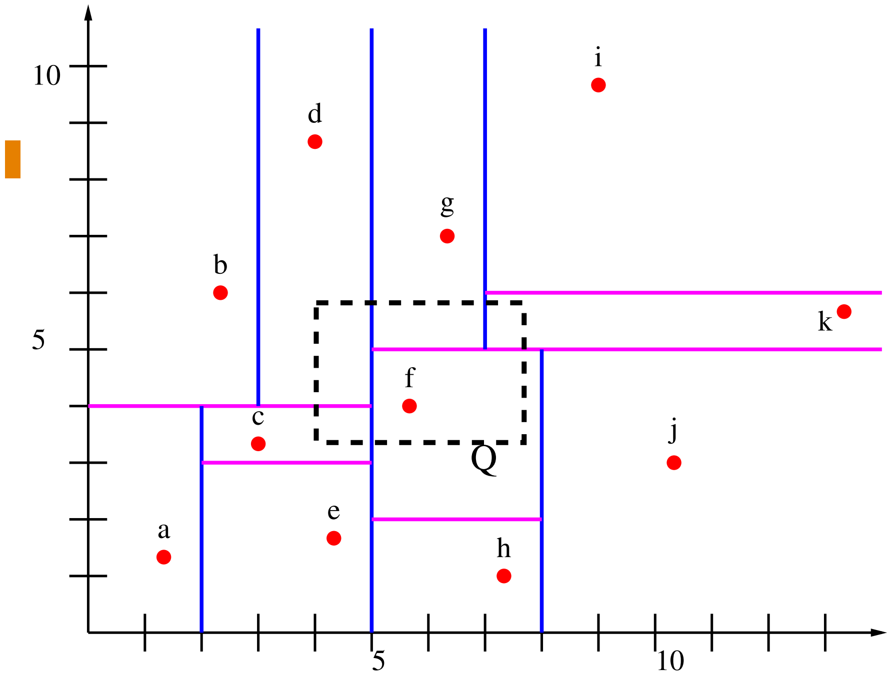
Knoten entspricht Rechteck $R(v)$

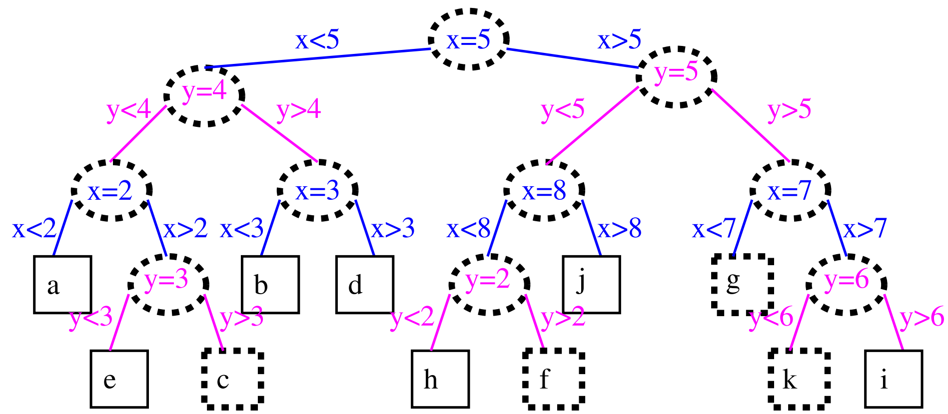




Jeder Knoten v entspricht einem Rechteck $R(v)$ als Schnitt von Halbebenen ($X = s$ und $Y = s$).

Anfrage-Query mit Rechteck q





- Bestimme alle Knoten v mit $R(v) \cap q \neq \emptyset$
- Falls v Blatt, teste $v \in q$

Buch Kapitel

Kapitel 2.3.3 Seite 82 oben – S. 88 unten

Kapitel 2.4 Seite 93 mitte – S. 95 mitte