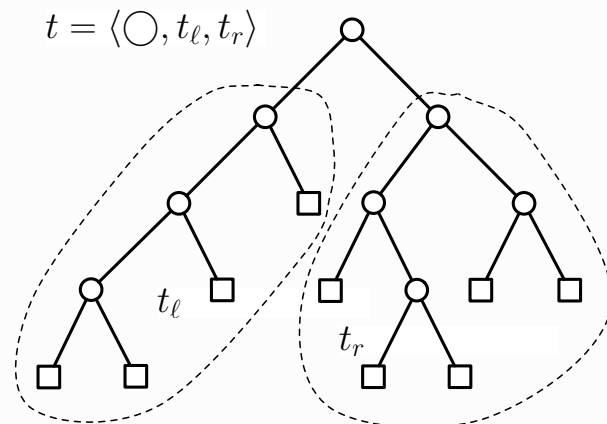


Binäre Bäume

\mathcal{B} : Menge der binären Bäume, rekursiv definiert durch die Regeln:

- ▶ \square ist ein binärer Baum
- ▶ sind t_ℓ, t_r binäre Bäume, so ist auch $t = \langle \bigcirc, t_\ell, t_r \rangle$ ein binärer Baum
- ▶ nur das, was durch die beiden vorigen Regeln erzeugt werden kann, ist ein binärer Baum

Beispiel



Übliche Darstellung von Binärbäumen:

Bäume mit Wurzel, wobei jeder Knoten zwei oder keinen Knoten als (geordnete) Nachfolger hat.

- ▶ zwei Nachfolger: mit \bigcirc bezeichnete inneren Knoten,
- ▶ kein Nachfolger: mit \square bezeichneten äusseren Knoten (Blätter).
- ▶ \square ist ein binärer Baum, der nur aus einem einzigen (äusseren) Knoten besteht;
- ▶ $t = \langle \bigcirc, t_\ell, t_r \rangle$ ist ein Baum mit Wurzel \bigcirc , an die zwei binäre Bäume t_ℓ und t_r als linker bzw. rechter Teilbaum angehängt sind.

Lineare Codierung von Binärbäumen
(Wörter über dem Alphabet $\{\bigcirc, \square\}$)

$$\text{code}(\square) = \square$$

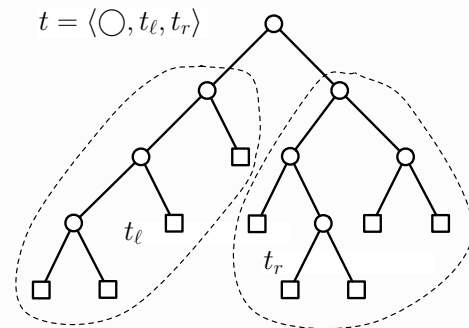
$$\text{code}(\langle \bigcirc, t_\ell, t_r \rangle) = \bigcirc \text{code}(t_\ell) \text{code}(t_r)$$

Entsprechend:

Generierung mittels einer kontextfreien Grammatik in BNF $G_{\mathcal{B}}$ mit:

- Variablensymbol B (auch Startsymbol)
- Terminalsymbolen \square und \bigcirc
- Produktionen

$$B \rightarrow \square \mid \bigcirc BB$$



$$\text{code}(t_\ell) = \bigcirc \bigcirc \bigcirc \square \square \square \square,$$

$$\text{code}(t_r) = \bigcirc \bigcirc \square \bigcirc \square \square \bigcirc \square \square$$

$$\text{code}(t) = \bigcirc \text{code}(t_\ell) \text{code}(t_r)$$

$$= \bigcirc \bigcirc \bigcirc \bigcirc \square \square \square \square \bigcirc \bigcirc \square \bigcirc \square \square \bigcirc \square \square$$

Bemerkungen:

- streicht man in jedem von $G_{\mathcal{B}}$ erzeugten Wort das letzte Symbol und identifiziert man

$\bigcirc \sim$ "linke Klammer" $\square \sim$ "rechte Klammer",

so entsprechen diese Wörter genau den wohlgeformten Klammerausdrücken entsprechender Länge.

- die von $G_{\mathcal{B}}$ erzeugte Sprache ist kontextfrei, aber nicht regulär

Notation

- $I(t)$: innere Knoten von $t \in \mathcal{B}$ (internal)
 - \square hat keine inneren Knoten
 - $\langle \bigcirc, t_\ell, t_r \rangle$ hat als innere Knoten die Wurzel \bigcirc , die inneren Knoten von t_ℓ und die inneren Knoten von t_r
- $E(t)$: äussere Knoten (Blätter) von $t \in \mathcal{B}$ (external)
 - \square hat \square als äusseren Knoten
 - $\langle \bigcirc, t_\ell, t_r \rangle$ hat als äussere Knoten die äusseren Knoten von t_ℓ und die äusseren Knoten von t_r
- \mathcal{B}_n : Menge der Binärbäume mit n inneren und $n + 1$ äusseren Knoten

Struktur und Anzahl

- Strukturelle Aussage

$$\mathcal{B}_0 = \{ \square \}$$

$$\mathcal{B}_{n+1} \leftrightarrow \mathcal{B}_0 \times \mathcal{B}_n \cup \mathcal{B}_1 \times \mathcal{B}_{n-1} \cup \mathcal{B}_2 \times \mathcal{B}_{n-2} \cup \dots \cup \mathcal{B}_n \times \mathcal{B}_0$$

$$= \sum_{0 \leq k \leq n} \mathcal{B}_k \times \mathcal{B}_{n-k}$$

▶ SEGNER'S Formel

$$c_0 = 1$$

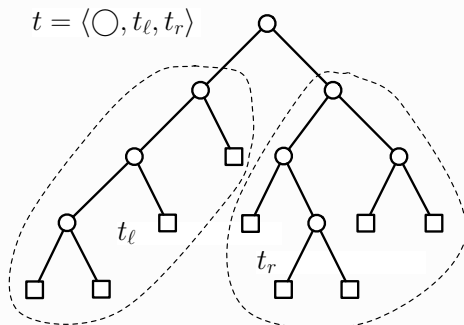
$$c_{n+1} = c_0 \cdot c_n + c_1 \cdot c_{n-1} + c_2 \cdot c_{n-2} + \dots + c_n \cdot c_0$$

▶ CATALANS Formel

$$c_n = \frac{1}{n+1} \binom{2n}{n} \in \Theta\left(\frac{4^n}{n^{3/2}}\right)$$

▶ $h(t)$ = Höhe ("height") von t :

$$h(t) = \begin{cases} 0 & \text{falls } t = \square \\ 1 + \max\{h(t_\ell), h(t_r)\} & \text{falls } t = \langle \bigcirc, t_\ell, t_r \rangle \end{cases}$$



$$i(t) = 8, e(t) = 9, s(t) = 17, h(t) = 4$$

Parameter für Binärbäume

▶ $i(t)$ = Anzahl der inneren ("internal") Knoten von t :

$$i(t) = \begin{cases} 0 & \text{falls } t = \square \\ 1 + i(t_\ell) + i(t_r) & \text{falls } t = \langle \bigcirc, t_\ell, t_r \rangle \end{cases}$$

▶ $e(t)$ = Anzahl der äusseren ("external") Knoten von t :

$$e(t) = \begin{cases} 1 & \text{falls } t = \square \\ e(t_\ell) + e(t_r) & \text{falls } t = \langle \bigcirc, t_\ell, t_r \rangle \end{cases}$$

▶ $s(t) = i(t) + e(t)$ = Grösse ("size") von t :

$$s(t) = \begin{cases} 1 & \text{falls } t = \square \\ 1 + s(t_\ell) + s(t_r) & \text{falls } t = \langle \bigcirc, t_\ell, t_r \rangle \end{cases}$$

Beziehungen zwischen den Parametern

Für alle binären Bäume t gelten folgende Aussagen:

▶ $e(t) = i(t) + 1$, und damit $s(t) = 2 \cdot i(t) + 1 = 2 \cdot e(t) - 1$

Beweis:

$$e(\square) - i(\square) = 1 - 0 = 1$$

$$\begin{aligned} e(\langle \bigcirc, t_\ell, t_r \rangle) - i(\langle \bigcirc, t_\ell, t_r \rangle) &= e(t_\ell) + e(t_r) - (i(t_\ell) + i(t_r) + 1) \\ &= e(t_\ell) - i(t_\ell) + e(t_r) - i(t_r) - 1 \\ &= 1 + 1 - 1 = 1 \end{aligned}$$

► $h(t) \leq i(t)$

Beweis:

$$\begin{aligned} h(\square) &= 0 = i(\square) \\ h(\langle \bigcirc, t_\ell, t_r \rangle) &= 1 + \max\{h(t_\ell), h(t_r)\} \\ &\leq 1 + \max\{i(t_\ell), i(t_r)\} \\ &\leq 1 + i(t_\ell) + i(t_r) \\ &= i(\langle \bigcirc, t_\ell, t_r \rangle) \end{aligned}$$

Höhe von Knoten

► Für $a \in I(t) \cup E(t)$:

$h(a, t)$ = Höhe von a in t = Abstand von a zur Wurzel von t

► rekursive Beschreibung:

$$\begin{aligned} h(\square, \square) &= 0 \\ h(a, \langle \bigcirc, t_\ell, t_r \rangle) &= \begin{cases} 0 & a = \bigcirc \\ h(a, t_\ell) + 1 & a \in E(t_\ell) \cup I(t_\ell) \\ h(a, t_r) + 1 & a \in E(t_r) \cup I(t_r) \end{cases} \end{aligned}$$

► $e(t) \leq 2^{h(t)}$, also $\log e(t) \leq h(t)$

Beweis:

$$\begin{aligned} e(\square) &= 1 = 2^0 = 2^{h(\square)} \\ e(\langle \bigcirc, t_\ell, t_r \rangle) &= e(t_\ell) + e(t_r) \\ &\leq 2^{h(t_\ell)} + 2^{h(t_r)} \\ &\leq 2 \cdot 2^{\max\{h(t_\ell), h(t_r)\}} \\ &= 2^{1 + \max\{h(t_\ell), h(t_r)\}} \\ &= 2^{h(\langle \bigcirc, t_\ell, t_r \rangle)} \end{aligned}$$

Innere und äussere Weglänge

► innere Weglänge von t :

$$w_i(t) = \sum_{a \in I(t)} h(a, t)$$

► äussere Weglänge von t :

$$w_e(t) = \sum_{a \in E(t)} h(a, t)$$

► mittlere innere Weglänge von t :

$$\bar{w}(t) = w_i(t)/i(t)$$

► mittlere äussere Weglänge = mittlere Höhe:

$$\bar{h}(t) = w_e(t)/e(t)$$

Rekursive Konstruktion von binären Suchbäumen

- Für eine Folge $\mathbf{a} = a_1 a_2 \dots a_n \in A^+$ sei

$\mathbf{a}_{<}$: die Teilfolge der $a_i < a_1$,

$\mathbf{a}_{>}$: die Teilfolge der $a_i > a_1$

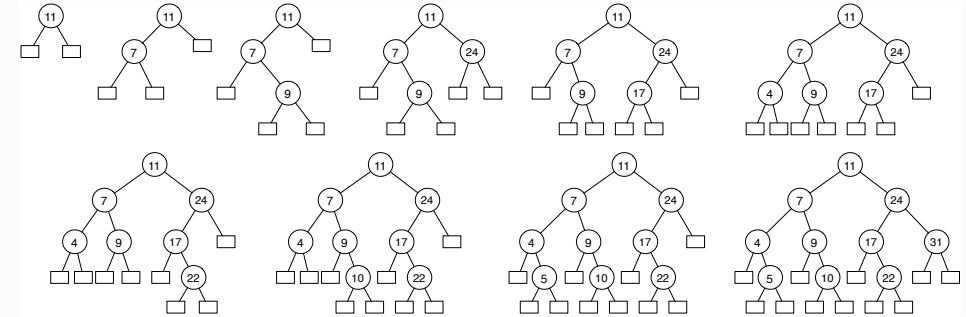
- Für die leere Folge λ sei $bs(\lambda) = (\square, \lambda)$
- Für $a \in A^+$ wird $bs(\mathbf{a}) = (t, v)$ definiert durch

$$v(\bigcirc) = a_1, \quad bs(\mathbf{a}_{<}) = (t_\ell, v_\ell), \quad bs(\mathbf{a}_{>}) = (t_r, v_r)$$

- Ein sequentieller Algorithmus zur Konstruktion verwendet sukzessives Einfügen bei erfolgloser Suche



Konstruktion des binären Suchbaumes zu
 $\mathbf{a} = (11, 7, 9, 24, 17, 4, 22, 10, 5, 31)$



Permutationsmodell

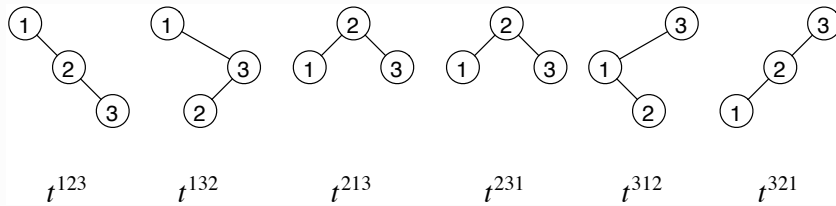
- Problem: wie gross ist der Aufwand für (erfolgreiche) Suche, wenn man aus Daten sequentiell einen binären Suchbaum konstruiert hat?
- Wird $a \in A$ in den A -Suchbaum (t, v) (erfolgreich) gesucht, so ist die Höhe $h(b, t)$ des Knotens $b \in I(t)$ mit $v(b) = a$ relevant.
- Damit ist klar: es kommt sowohl auf das gesuchte Datum an, als auch auf den Suchbaum (d.h. die Reihenfolge, in der die Daten zur Konstruktion des Suchbaumes verwendet wurden).



- Für n (verschiedene) Daten $\in A$ betrachte alle $n!$ Permutationen als gleichwahrscheinliche inputs für die Konstruktion eines bs .
- Man kann annehmen, dass $A = \{1, 2, \dots, n\}$ und alle $\sigma \in \mathcal{S}_n$ inputs sind. Für $\sigma \in \mathcal{S}_n$ sei $bs(\sigma) = (t^\sigma, v^\sigma)$, also $t \in \mathcal{B}_n$.
 - Vorsicht: verschiedene $\sigma \in \mathcal{S}_n$ können denselben Suchbaum liefern!
 - v^σ ist durch t^σ eindeutig bestimmt, also redundant!
- Es wird mit gleicher Wahrscheinlichkeit nach jedem der Daten $k \in \{1, 2, \dots, n\}$ gesucht
- Für einen festen A -Suchbaum (t, v) ist die (mittlere) innere Weglänge $w_i(t)$ bzw. $\bar{w}(t) = w_i(t)/n$ das Maß für den Suchaufwand.



Beispiel: die Suchbäume zu den Permutationen aus \mathcal{S}_3



Es gilt

$$w_i(t^{123}) = w_i(t^{132}) = w_i(t^{312}) = w_i(t^{321}) = 3, w_i(t^{213}) = w_i(t^{231}) = 2$$

also

$$w_3 = \frac{1}{3!} \sum_{\sigma \in \mathcal{S}_3} w_i(\sigma) = \frac{3 + 3 + 2 + 2 + 3 + 3}{6} = \frac{8}{3}, \bar{w}_3 = \frac{1}{3} w_3 = \frac{8}{9}$$

► Gesucht ist

$$w_n = \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} w_i(t^\sigma) \text{ bzw. } \bar{w}_n = \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} \bar{w}(t^\sigma) = \frac{1}{n} w_n$$

► Wir werden sehen: die w_n erfüllen die Quicksort-Rekursion!

$$w_{n+1} = \frac{2}{n+1} \sum_{k=0}^n w_k + n, \quad w_0 = 0$$

► Folgerung: der mittlere Suchaufwand bei binären Suchbäumen im Permutationsmodell ist

$$\bar{w}_n = \frac{1}{n} w_n = \left(2 + \frac{2}{n}\right) H_n - 4 \in \Theta(\log n)$$

Zum Beweis:

► Betrachte die Abbildung

$$\mathcal{S}_n \ni \sigma \mapsto (P_\sigma, \sigma_<, \sigma_>^{\ominus \sigma_1}) \in \binom{2..n}{\sigma_1-1} \times \mathcal{S}_{\sigma_1-1} \times \mathcal{S}_{n-\sigma_1}$$

Dabei ist (mit $\sigma_1 = k$):

- $P_\sigma = \{i_1, i_2, \dots, i_{k-1}\} \subseteq \binom{2..n}{k-1}$ mit $2 \leq i_1 < i_2 < \dots < i_{k-1} \leq n$ die Positionen $2 \leq i \leq n$ mit $\sigma_i < k$
- $\sigma_< = \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_{k-1}} \in \mathcal{S}_{k-1}$
- $\sigma_>^{\ominus k} = (\sigma_{j_1} - k)(\sigma_{j_2} - k) \dots (\sigma_{j_{n-k}} - k) \in \mathcal{S}_{n-k}$, wobei $j_1 < j_2 < \dots < j_{n-k} \leq n$ die Positionen $2 \leq j \neq n$ mit $\sigma_j > k$ sind

► Diese Abbildung ist bijektiv zwischen den besagten Mengen!

Beispiel für die Bijektion

$$\mathcal{S}_n \ni \sigma \mapsto (P_\sigma, \sigma_<, \sigma_>^{\ominus \sigma_1}) \in \binom{2..n}{\sigma_1-1} \times \mathcal{S}_{\sigma_1-1} \times \mathcal{S}_{n-\sigma_1}$$

mit $n = 9$

- $\sigma = (681379542)$, also $k = \sigma_1 = 6$
- $P_\sigma = (34789) \in \binom{2..9}{5}$
- $\sigma_< = (13542) \in \mathcal{S}_5$
- $\sigma_>^{\ominus 6} = (213) \in \mathcal{S}_3$

Mit $\sigma' = \sigma_{<}$ und $\sigma'' = \sigma_{>}^{\ominus \sigma_1}$ gilt

$$w_i(t) = w_i(t^{\sigma'}) + w_i(t^{\sigma''}) + i(t) - 1$$

und daher ist

$$\begin{aligned} w_n &= \frac{1}{n!} \sum_{\sigma \in \mathcal{S}_n} w_i(t^\sigma) \\ &= \frac{1}{n!} \sum_{k=1}^n \sum_{P \in \binom{[2..n]}{k-1}} \sum_{\sigma' \in \mathcal{S}_{k-1}} \sum_{\sigma'' \in \mathcal{S}_{n-k}} (w_i(t^{\sigma'}) + w_i(t^{\sigma''}) + n - 1) \\ &= \frac{1}{n!} \sum_{k=1}^n \binom{n-1}{k-1} \sum_{\sigma' \in \mathcal{S}_{k-1}} \sum_{\sigma'' \in \mathcal{S}_{n-k}} (w_i(t^{\sigma'}) + w_i(t^{\sigma''}) + n - 1) \\ &= \frac{1}{n} \sum_{k=1}^n \frac{1}{(k-1)!} \frac{1}{(n-k)!} \sum_{\substack{\sigma' \in \mathcal{S}_{k-1} \\ \sigma'' \in \mathcal{S}_{n-k}}} (w_i(t^{\sigma'}) + w_i(t^{\sigma''}) + (n-1)) \end{aligned}$$

Mit

$$\frac{1}{(k-1)!} \frac{1}{(n-k)!} \sum_{\substack{\sigma' \in \mathcal{S}_{k-1} \\ \sigma'' \in \mathcal{S}_{n-k}}} w_i(t^{\sigma'}) = \frac{1}{(k-1)!} \sum_{\sigma' \in \mathcal{S}_{k-1}} w_i(t^{\sigma'}) = w_{k-1}$$

und

$$\frac{1}{(k-1)!} \frac{1}{(n-k)!} \sum_{\substack{\sigma' \in \mathcal{S}_{k-1} \\ \sigma'' \in \mathcal{S}_{n-k}}} w_i(t^{\sigma''}) = \frac{1}{(n-k)!} \sum_{\sigma'' \in \mathcal{S}_{n-k}} w_i(t^{\sigma''}) = w_{n-k}$$

und

$$\frac{1}{(k-1)!} \frac{1}{(n-k)!} \sum_{\substack{\sigma' \in \mathcal{S}_{k-1} \\ \sigma'' \in \mathcal{S}_{n-k}}} (n-1) = n-1$$

folgt die Behauptung:

$$w_n = \frac{1}{n} \sum_{k=1}^n (w_{k-1} + w_{n-k} + n - 1) = \frac{2}{n} \sum_{k=0}^{n-1} w_{k-1} + n - 1$$