Ferienakademie im Sarntal 1999

Kurs 2: Bäume – Algorithmik und Kombinatorik

Bijektion und Codierung

Edgar Frank, FAU Erlangen-Nürnberg

Gliederung

- Die Formel von *Cayley*
- Darstellungsmöglichkeiten für Permutationen
- Bijektiver Beweis nach *Joyal*
- *Prüfer*-Code
- *Foata-Fuchs*-Codierung und Anwendungen

Die Formel von *Cayley*

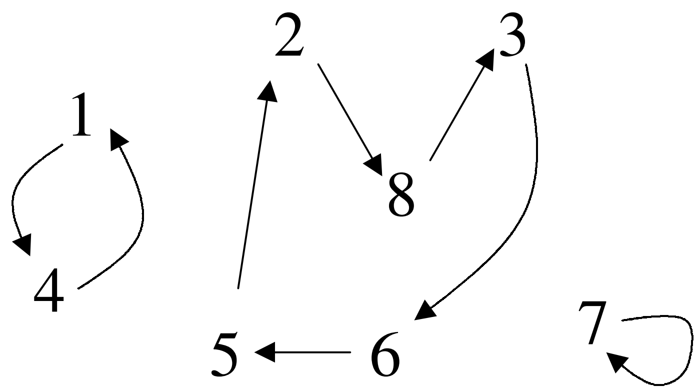
- Anzahl der Bäume auf einer Menge von n unterscheidbaren Elementen:

$$t_n = n^{n-2}$$

- Zählen von Bäumen $\leftrightarrow$ Bijektion auf Objekte, die man einfacher zählen kann

Darstellungsmöglichkeiten für Permutationen

- Graphisch



- Tabellarisch

i	1	2	3	4	5	6	7	8
$f(i)$	4	8	6	1	2	5	7	3

- Folge von Zyklen

4 1 7 8 2 5 6 3

Bijektiver Beweis nach *Joyal*

- Bijektion zwischen Wirbeltieren und Abbildungen einer Menge in sich
- M_n := Abbildungen von $\{2, ..., n-1\}$ nach $\{1, ..., n\}$
- $|M_n| = n^{n-2}$
- T_n := Bäume auf der Knotenmenge $\{1, ..., n\}$

- Codierung einer Abbildung $f \in M_n$ als Baum ($Encode$):
  - Gerichteter Graph D auf $V = \{1, ..., n\}$ mit $E = \{(i, f(i), i = 2, ..., n - 1\}$
  - Komponenten $C_0 \ldots C_{k+1}$: 2 Bäume (C_0 und C_{k+1}), k Zyklen mit angehefteten Bäumen
  - Für $i = 1, ..., k$:
    $r_i :=$ kleinstes Element des Zyklus von C_i
    $l_i := f(r_i)$
  - Ordne die Komponenten so, dass $r_1 < r_2 < \ldots < r_k$
  - Setze $r_0 := 1$
  - Für $i = 1, ..., k$:
    $E \leftarrow E \cup \{(r_{i-1}, l_i)\}$
    $E \leftarrow E \setminus \{(r_i, l_i)\}$
  - $E \leftarrow E \cup \{(r_k, n)\}$

- *Encode* erzeugt einen Baum:
  - Komponenten von D werden verbunden
  - Von jedem Zyklus wird eine Kante entfernt
  - Induktion über die Anzahl der Komponenten:
    - IA: $k = 0$ trivial
    - IS: Komponente C_1 wird mit C_0 verbunden, beide zusammen sind Baum
Anzahl k um 1 kleiner

- Decodieren eines Baums in eine Abbildung ($Decode$):
  - Setze $r_0 := 1$, $i = 0$
  - p_i : Pfad von r_i nach n
  - Solange $\mathrm{succ}(r_i) \neq n$:

$$\begin{array}{lcl} r_{i+1} & := & \text{kleinster Knoten} \in p_i \setminus r_i \\ l_{i+1} & := & \mathrm{succ}(r_i) \\ E & \leftarrow & E \setminus \{(r_i, l_{i+1})\} \\ E & \leftarrow & E \cup \{(r_{i+1}, l_{i+1})\} \\ i & \leftarrow & i + 1 \end{array}$$

  - $E \leftarrow E \setminus \{(r_i, n)\}$

- *Encode* und *Decode* sind Inverse und Bijektionen
  - Kanten, die von *Encode* gelöscht werden, werden bei *Decode* eingefügt und umgekehrt
  - *Encode* ist injektiv:
    - Annahme: $\exists f_1 \neq f_2$ mit $Encode(f_1) = Encode(f_2)$
    - $f_1 = Decode(Encode(f_1)) = Decode(Encode(f_2)) = f_2$
      $\Rightarrow$ Widerspruch
  - *Decode* ist ebenfalls injektiv
  - M_n und T_n endliche Mengen $\Rightarrow$ bijektiv

Prüfer-Code

- Direkte Bijektion zwischen Bäumen und Folgen
- Codierung $\gamma : T_n \to [n]^{n-2}$:
  - Für $t \in T_n$ definiere eine Folge $t_1, t_2, ..., t_{n-1}$ von Bäumen:
    - Setze $t_1 = t$
    - Für $i = 1, ..., n-2$:
      v_i := kleinster Blattknoten von t_i
      s_i := Nachbarknoten von v_i
      t_{i+1} := t_i ohne v_i und Kante (v_i, s_i)
    - $\gamma(t) = (s_1, s_2, \ldots, s_{n-2})$

- Decodierung:
  - Gegeben $\gamma(t) = (s_1, s_2, ..., s_{n-2})$
  - v_1 := kleinstes Element $\in \{1, ..., n\}$, das nicht in der Folge $\gamma(t)$ enthalten ist
  - Problem wird reduziert auf die Berechnung von t_2 (mit Knotenmenge $\{1, ..., n\} \setminus \{v_1\}$)
  - Aus Induktion über die Anzahl der Knoten n:
    Baum t kann eindeutig aus der Folge $\gamma(t)$ zurückgewonnen werden
    (IA: $n = 2$)

Foata-Fuchs-Codierung und Anwendungen

- Ausweitung des Prüfer-Codes auf die Codierung beliebiger Abbildungen einer Menge in sich
- Codierung einer Abbildung $f : [n] \to [n]$ durch ein Wort $c = c(f) \in \{x_1, x_2, ..., x_n\}^n$
- $x_i \leftrightarrow i$
- Gerichteter Graph D beschreibt f
- Komponenten: gerichtete Zyklen mit angehefteten Bäumen

- Codierung des zyklischen Anteils:
  - Jeder Zyklus liefert ein Wort z_j := größtes Element, danach restliche Elemente entgegen der Richtung der Pfeile
  - w_o := Konkatenation der z_j, aufsteigend nach ihren ersten Elementen geordnet
- Codierung des baumartigen Anteils:
  - Ordne die k Blätter von D nach aufsteigender Beschriftung: $l_1 < l_2 < ... < l_k$
  - Für $i = 1, ..., k$:
    w_i ist der Pfad, der l_i mit $w_0 w_1 ... w_{i-1}$ verbindet, ohne l_i, aber mit der Wurzel $r_i \in w_0 w_1 ... w_{i-1}$, aufgeschrieben in Richtung von r_i nach l_i
- Codierung $c(f) = w_0 w_1 ... w_k$

- Decodierung:
  - Für $w \in \{x_1, x_2, ..., x_n\}^n$ definiere eine Folge $w_0, w_1, ...$:
    - w_0 := Präfix von w bis zur ersten Wiederholung eines Zeichens
    - $w = w_0 v_0$
    - w_{i+1} := Präfix von v_i bis zur ersten Wiederholung eines Zeichens
    - $v_i = w_{i+1} v_{i+1}$
    - ... bis w vollständig abgearbeitet wurde
  - $w = w_0 w_1 ... w_k$
  - w_k beschreibt den Pfad von $w_0 w_1 ... w_{k-1}$ zum größten Blatt l_k
  - $l_k = \max\{x_j \in \{x_1, x_2, ..., x_n\} \mid x_j \notin w\}$
  - $w_k = y_1 y_2 ... y_l$: Problem reduziert sich Berechnung von f mit Knotenmenge $[n] \setminus \{l_k, y_1, y_2, ..., y_l\}$
  - Induktion über Anzahl der Knoten n:
    f kann eindeutig aus w zurückgewonnen werden

- Beispiele zur Codierung:
  - $f(i) = i \Rightarrow c(f) = x_1 x_2 \ldots x_n$
  - $f(i) = const = j \Rightarrow c(f) = x_j^n$
  - f ist zirkulär: $f(n) = 1$ und $f(i) = i + 1,\ i < n$
    $\Rightarrow c(f) = x_n x_{n\text{-}1} \ldots x_2 x_1$
  - $f(1) = 1$ und $f(i) = i - 1,\ i > 1 \Rightarrow c(f) = x_1{}^2 x_2 \ldots x_{n-1}$
  - $f(i) = f(m + i) = i,\ i \leq m, m = n/2 \Rightarrow c(f) = (x_1 x_2 \ldots x_m)^2$