Publ. I.R.M.A. Strasbourg, 1988, 348/S–17
Actes 17$^{e}$ Séminaire Lotharingien, p. 141–148

# EINIGE BEMERKUNGEN ZU GLEICHHEITSTESTS UND DEREN KOMPLEXITÄT

Herrn Prof. Dr. St. SCHOTTLAENDER
in Dankbarkeit und Respekt zu seinem 60. Geburtstag

VON

W. LEX

In der - sit venia verbo - Kunst des Programmierens wird man des öfteren mit der Aufgabe konfrontiert, festzustellen, ob in einer Liste Elemente übereinstimmen; etwa, wenn ein durch seine Cayleytafel gegebenes Magma auf Kürzbarkeit hin zu untersuchen ist oder wenn in einer Datenbank, z. B. einer Personalkartei, gleiche Attributwerte bestimmt werden sollen. Im wesentlichen läuft das darauf hinaus, daß eine endliche Folge natürlicher Zahlen auf gleiche Glieder hin getestet wird.

Genauer - $\mathbb{N}$ sei die Menge positiver ganzer Zahlen und $\mathbf{n} = \{x \in \mathbb{N} \mid x \leq n\}$ für $n \in \mathbb{N}$ - legen wir fest

**Definition 1:** $\alpha = (a_\varkappa)_{\varkappa \in \mathbf{k}}$ sei eine Folge auf $\mathbf{n}$ mit $k, n \in \mathbb{N}^{\neg 1}$; ein Algorithmus $\mathcal{y}$ heiße *Gleichheitstest* - für $\alpha$ -, falls $\mathcal{y}$ die Wahrheit bzw. Falschheit von

$$(1) \qquad \exists\, \varkappa, \lambda \in \mathbf{k}: \quad \varkappa < \lambda \;\wedge\; a_\varkappa = a_\lambda$$

feststellt.

Für $k > n$ liefert das Schubfachprinzip sofort die Gültigkeit von (1). - Wir beschränken uns hier auf $k = n$, da der allgemeine Fall später ausführlicher dargestellt werden soll.

Nach einer vollständigen Analyse des üblichen Vergleichstests in I wird in II eine Klasse von Algorithmen betrachtet, die die Folge $\alpha$ nur einmal linear durchlaufen und bei minimalem $\lambda$ in (1) abbrechen. - Gleichheitstests scheinen bislang in der Literatur nicht behandelt worden zu sein; für entsprechende Hinweise ist der Autor dankbar. [1]

## I

Der wohl nächstliegende Gleichheitstest für $\alpha$ mit $k = n$ besteht im sukzessiven Vergleich der einzelnen Folgenglieder mit den restlichen und Abbruch bei der ersten Gleichheit, genauer

**Definition 2:** Sei $\alpha$ wie in Definition 1 und $k = n$ ($\in \mathbb{N} \setminus 1$):

$\alpha$) $\mathfrak{w}$ bezeichne den durch folgende PASCAL-Anweisungen gegebenen Algorithmus, wobei $\alpha$ im Feld A stehe, G eine Boolesche und I, J Integer-Variable sowie N eine Konstante für n seien:

```
        G := false;
        for I := 1 to N-1 do
            for J := I + 1 to N do
                if A [I] = A [J] then ℊ;
    1:  if G then writeln ('Gleichheit')
             else writeln ('Alle verschieden');
```

mit $\mathfrak{g}$ als Abkürzung für

```
begin  G := true; goto 1  end
```

.

---

[1] Frl. cand. inf. M. Söding gilt mein aufrichtiger Dank für das Testen der Algorithmen und für kritische Durchsicht.

β) $V_n$ sei die *Durchschnittskomplexität* für $\mathfrak{W}$, Gleichverteilung vorausgesetzt, also die mittlere Anzahl der zur Feststellung von Gleichheit oder Verschiedenheit (zur Erreichung der Marke 1 im Teil α)) notwendigen Vergleiche (Abfrage in Zeile 4 des obigen Programmausschnittes), wenn jede Folge α gleichwahrscheinlich ist.

Optimale Komplexität, 1, und pessimale, $\sum_{\nu=1}^{n-1} (n-\nu) = \binom{n}{2}$, liefern eine erste grobe Abschätzung

$$1 \leq V_n \leq \binom{n}{2};$$

genauer erhält man

Satz 1: Für $n \in \mathbb{N}\setminus 1$ gilt

(2) $$V_n = n - R_n$$

mit

$$R_n \leftrightarrows \frac{n!}{n^n} \sum_{\nu=0}^{n-1} \frac{\nu^\nu}{\nu!}$$

und

(3) $$0{,}581\ 976\ldots = \frac{1}{e-1} < R_n \leq 1;$$

überdies erweist sich $\rho \leftrightarrows (R_n)_{n \in \mathbb{N}\setminus 1}$ als streng monoton fallende, gegen $(e-1)^{-1}$ konvergierende Folge.

*Beweis*: Wir betrachten alle $n^n$ n-tupel über n und bezeichnen mit $A_n$ die Gesamtzahl der notwendigen Vergleiche; also

(4) $$V_n = n^{-n} A_n.$$

Der Gleichverteilung wegen beschränken wir uns auf die etwa mit 1 beginnende Folgen, nennen die Gesamtzahl der notwendigen Vergleiche - bei $a_1 = 1$ - $E_n$ und haben

$$A_n = n E_n. \tag{5}$$

Ferner sei $B_n$ die Zahl der notwendigen Vergleiche bei

$$a_1 = 1 \wedge \exists \nu \in n \setminus 1 : a_\nu = 1;$$

damit ergibt sich

$$E_n - B_n = (n-1) \cdot (n-1)^{n-1} + A_{n-1}. \tag{6}$$

$b_\nu$ sei die Zahl der notwendigen Vergleiche im Falle $a_1 = 1 = a_{\nu+1}$ und $a_\mu \neq 1$ für $\mu \in \nu \setminus 1$ und $\nu \in n-1$; mithin gilt

$$b_\nu = \nu (n-1)^{\nu-1} \cdot n^{n-1-\nu},$$

unter Beachtung der sich durch Differentiation aus der Summe einer geometrischen Reihe ergebenden Summationsformel also

$$B_n = \sum_{\nu=1}^{n-1} b_\nu = n^n - (2n-1)(n-1)^{n-1}$$

und mit (4), (5) und (6) sowie $q_n \leq \frac{n-1}{n}$ für $n \in \mathbb{N}$ weiter

$$V_n = n + q_n^{n-1} (V_{n-1} - n),$$

was unschwer auf die Vermutung

$$V_n = n - \sum_{\mu=1}^{n} \prod_{\nu=\mu}^{n} q_\nu^{\nu-1}$$

führt, die sich mittels Induktion verifizieren läßt und nach einiger Umformung (2) impliziert.

Durch Induktion beweist man

$$\sum_{\nu=0}^{n-1} \frac{\nu^\nu}{\nu!} \leq \frac{n^n}{n!} \quad (n \in \mathbb{N}),$$

was die obere Schranke in (3) liefert, und wegen[2)]

$$R_{n+1} = (1+\tfrac{1}{n})^{-n} (R_n+1) \tag{7}$$

auch $R_{n+1} < R_n$, woraus die Konvergenz von $\rho$ folgt. Mit $\lim_{n\to\infty} (1+\frac{1}{n})^n = e$ und (7) ergibt sich daher der Grenzwert von $\rho$ zu $(e-1)^{-1}$.

## II

Wie erwähnt behandeln wir nun einige Tests, die das zu testende n-tupel nur linear durchwandern. Um deren Komplexität zu beschreiben, geben wir

**Definition 3**: Mit $\alpha$ wie in Definition 1 und $k = n$ ($\in \mathbb{N}\setminus 1$) sei

$$m(\alpha) \leftrightarrows \min \{\lambda \in \mathbf{n} \mid \exists\, \varkappa \in \mathbf{n}: \varkappa < \lambda \wedge a_\varkappa = a_\lambda\},$$

ferner

$$m_\alpha \leftrightarrows \begin{cases} m(\alpha) \\ n \end{cases}, \text{ falls } \begin{cases} (1) \\ \rightarrow (1) \end{cases}$$

und

$$M_n \leftrightarrows n^{-n} \sum_{\alpha \in \mathbf{n}^\mathbf{n}} m_\alpha$$

(wobei $A^B \leftrightarrows \{f \mid f : B \to A\}$).

Elementare Überlegungen und eine grobe Abschätzung liefern

---

[2)] Für den Hinweis, hier rekursiv vorzugehen, danke ich Herrn Dr. J. Schmalmack herzlich [2].

Satz 2: Für alle $n \in \mathbb{N} \setminus 1$ gilt

$$M_n = (n-1)!\left(\sum_{\nu=1}^{n-2} \frac{\nu(\nu+1)}{n^\nu (n-\nu)!} + n^{3-n}\right)$$

und

$$2 \leq M_n < \frac{n+3}{2} \tag{8}$$

(dabei sei $\sum_{\nu=1}^{0} \ldots \leftarrow 0$).

In der nächsten Definition werden drei Algorithmen der oben genannten Art festgelegt: $\mathcal{E}$ schöpft **n** aus, $\mathcal{P}$ nutzt - ähnlich wie bei Gödelisierungen - die Eindeutigkeit der Primfaktorzerlegung, und $\mathcal{Z}$[3] gebraucht - wie auch $\mathcal{P}$ - die Folgenglieder als Indizes.

**Definition 4:** $\alpha$ und $k = n$ seien wie in Definition 2, ferner werde der Programmteil von der Marke 1 an jeweils von dort übernommen ebenso wie $\mathcal{G}$ und die Bedeutung der einschlägigen Variablen und Konstanten, während der Typ der nicht eigens vereinbarten Variablen sinngemäß zu ergänzen ist. Durch die folgenden PASCAL-Anweisungen werden die erwähnten Verfahren beschrieben, und zwar

α) $\mathcal{E}$ durch

```
G := false; Z := [1..N]-[A[1]];
for I := 2 to N do
    if A[I] in Z then Z := Z-[A[I]] else g;
1: ...
```

β) $\mathcal{P}$ durch

---

3) Herrn Prof. Dr. J. Gillis danke ich aufrichtig für die freundliche Angabe eines Tests, der im wesentlichen mit $\mathcal{Z}$ übereinstimmt [1].

```
        G := false; Q := Q div P[A[1]];
        for I := 2 to N do
            begin D := P[A[I]];
                  if Q mod D = 0 then Q := Q div D else 𝒴 end;
    1: ... ,
```

wobei im Feld P die n ersten Primzahlen in natürlicher - oder auch beliebiger! - Reihenfolge und in Q deren Produkt stehen.

γ) $\mathcal{Z}$ durch

```
        G := false, W[A[1]] := false;
        for I := 2 to N do
            if W[A[I]] then W[A[I]] := false else 𝒴 ;
    1: ... ,
```

wobei jede Komponente des Feldes W zunächst auf "true" gesetzt ist.

Zum Abschluß vergleichen wir die eben definierten Tests mit dem in I behandelten Verfahren $\mathcal{W}$:

**Satz 3:** Bei einem Gleichheitstest $\mathcal{Y}$ für Folgen der Länge $k = n$ ($\in \mathbb{N} \setminus 1$) ergibt sich die mittlere Zeitkomplexität $K_n(\mathcal{Y})$ von $\mathcal{Y}$ unter Vernachlässigung der Aufbereitung der zusätzlich benutzten Felder und Mengen sowie eventueller additiver Konstanten zu

(9) $$K_n(\mathcal{W}) = V_n v, \qquad K_n(\mathcal{Y}) = M_n c_{\mathcal{Y}}$$

für $\mathcal{Y} \in \{\mathcal{E}, \mathcal{P}, \mathcal{Z}\}$ mit

$$v, c_{\mathcal{Y}} \in \mathbb{R}^+ \ (\leq \{x \in \mathbb{R} \mid x > 0\});$$

ferner gilt

$$2v > c_{\mathcal{y}} \Rightarrow \exists m \in \mathbb{N}\ \forall n \in \mathbb{N}\ (n > m \Rightarrow K_n(\mathcal{y}) < K_n(\mathcal{W}))$$

für $\mathcal{y} \in \{\mathcal{E}, \mathcal{P}, \mathcal{F}\}$.

*Beweis:* (9) folgt unmittelbar aus der Definition der betreffenden Algorithmen.

Es sei $2v > c \leftrightarrows c_{\mathcal{y}}$ und $m$ minimal aus $\mathbb{N}$ mit

$$m \geq \frac{3c + 2v}{2v - c},$$

was für alle $n \in \mathbb{N}$ mit $n > m$ unter Beachtung von (9), (8), (2) und (3) sofort

$$K_n(\mathcal{y}) = M_n c < \frac{n+3}{2} c < (n-1)v \leq V_n v = K_n(\mathcal{W}),$$

also die Behauptung liefert.

Numerisch empirische Bestätigung dieser Vergleiche sind geplant.

## Referenzen

1. J. Gillis: Mündliche Mitteilung Mai 1987; Eremo dei SS. Pietro e Paolo, Bienno.

2. J. Schmalmack: Mündliche Mitteilung Herbst 1985, Clausthal.

W. Lex
Institut für Informatik
der Technischen Universität Clausthal
Erzstr. 1
D-3392 Clausthal-Zellerfeld