Berichtigung zu: Tabulator-Turingmaschine und Komplexität

[Computing 7, 264—274 (1971)]

Von

R. Weicker, Berlin

(Eingegangen am 8. Februar 1972)

In Satz 4 des Aufsatzes sollte bewiesen werden, daß man durch die Definition einer Tabulator-Turingmaschine (TTM) zu anderen Komplexitätsklassen kommt als bei normalen Turingmaschinen (TM's). Herr Dr. B. MONIEN (Universität Hamburg) hat mich darauf aufmerksam gemacht, daß der angegebene Beweis das nicht leistet, da hier Aussagen über auf verschiedene Weise definierte On-Line-Turingmaschinen unzulässig miteinander in Beziehung gebracht wurden. Mit einer anderen Sprache L und für On-Line-Turingmaschinen im Sinne von HENNIE [2] (d. h. Maschinen mit einem Eingabeband, bei denen die Maschine das $(i+1)$ -te Eingabesymbol a^{i+1} erst dann lesen kann, wenn sie entschieden hat, ob die Kette $a^1 \dots a^i$ zur zu erkennenden Sprache gehört) läßt sich das gewünschte Ergebnis jedoch erzielen:

Satz: Es gibt eine Sprache L , für die gilt:

1. Ist L Bandmenge einer (ein- oder mehrbändigen) On-Line-TM der Komplexität $T(n)$, so gilt $T(n) \geq C n \sqrt{n}$ für ein $C > 0$.
2. L kann von einer einbändigen On-Line-TTM der Komplexität $T'(n) = C' n$ (für ein $C' > 0$) erkannt werden.

Beweis: L wird definiert als eine Menge von Wörtern w , die Folgen von mindestens $k+1$ 0–1-Blöcken der Länge k sind (k variabel, aber für jedes $w \in L$ fest), wobei der letzte Block gleich einem bestimmten Block in dem ersten, k Blöcke umfassenden Abschnitt von w ist.

Formal: $w \in L$ gilt genau dann, wenn für w gilt:

$$(i) \quad w = \underbrace{u_1 \ 2 \ u_2 \ 2 \ u_3 \ 2 \ \dots \ 2 \ u_k \ 2 \ 2}_{=: u} \ \underbrace{v_1 \ 2 \ v_2 \ 2 \ \dots \ 2 \ v_m \ 2}_{=: v}$$

mit $u_i, v_i \in \{0, 1\}^*$, $|u_i| = k$, $|v_i| = 2k$ ($i = 1, \dots, k$),

$$v_i = x_i \bar{j}_i \quad \text{mit} \quad x_i \in \{0, 1\}^*, |x_i| = k, \bar{j}_i = \underbrace{1 \ \dots \ 1}_{j_i} \ 0 \ \dots \ 0 \quad (i = 1, \dots, m)$$

$\underbrace{\hspace{1.5cm}}$ $\underbrace{\hspace{1.5cm}}_{k-j_i}$

(also $\bar{j}_i$ „Strichzahl-Verschlüsselung“ einer Zahl j_i , $1 \leq j_i \leq k$),

(ii) der letzte Block v_m von v erfüllt die Bedingung $x_m = u_{j_m}$.

Der Beweis von 1. verläuft analog dem entsprechenden Beweis in [2] und [1], hier sollen die einzelnen Schritte nur kurz angegeben werden:

a) L kann von einer mehrbändigen On-Line-TM in $C_1 k^3 = C_1 n \sqrt{n}$ Schritten erkannt werden.

b) Sei A_k für beliebiges, aber festes k die Menge aller $w \in \{0, 1, 2\}^*$, die (i) erfüllen und für die $l=k$ gilt, also $n_k := |w| = 3k^2 + 2k + 1$ für $w \in A_k$. Nach dem Einlesen zweier verschiedener Anfangsabschnitte u, u' von Wörtern $w, w' \in A_k$ muß sich eine TM M , die L erkennt, in verschiedenen Maschinenkonfigurationen (abgekürzt: MK) befinden. Denn sei etwa in u, u' $u_i \neq u'_i$. Dann läßt sich ein Block v_1 angeben (nämlich $v_1 := u_i \bar{i}$), für den M verschieden reagieren muß: Annahme für $u \geq v_1$, Ablehnung für $u' \geq v_1$. Die Anzahl der für die einzelnen Anfangsabschnitte u charakteristischen MK's beträgt also 2^k .

c) Es gibt für jedes k einen Anfangsabschnitt u eines Wortes $w \in A_k$ und einen Block v_1 , so daß M zur Entscheidung, ob $u \geq v_1 \in L$ gilt, mindestens

$$r_k := \frac{k^2 - \log q - t \log s - 1}{2 t \log s}$$

Schritte braucht; dabei ist q die Zahl der Zustände, t die der Bänder und s die der Bandsymbole von M . Denn die Zahl M_k der MK's, die sich im Zustand der TM oder in den Teilen der Bandinhalte voneinander unterscheiden, die die Leseköpfe jeweils in r_k Schritten erreichen können, ist

$$M_k = q s^{t(2r_k + 1)} = q s^{(k^2 - \log q - 1)/\log s} = 2^{k^2 - 1} < 2^{k^2}.$$

Wegen b) muß es also zwei verschiedene Anfangsabschnitte u, u' (es sei dabei etwa $u_i \neq u'_i$) geben, deren charakteristische MK's sich erst auf Feldern, die mehr als r_k Schritte von einem Lesekopf entfernt sind, voneinander unterscheiden. Dann muß aber M zur Entscheidung, ob $u \geq u_i \bar{i} \in L$ gilt, mindestens r_k Schritte machen.

d) Es gibt ein $w \in A_k$, bei dem diese Abschätzung auch für die anderen Blöcke des Abschnitts v gilt, bei dem also M zur Verarbeitung mindestens $k r_k = C_2 k^3 = C_2 n_k \sqrt{n_k}$ Schritte benötigt.

e) Wegen der Monotonie der Komplexitätsfunktionen läßt sich das Ergebnis auf allgemeine Eingabelängen n übertragen, eine L erkennende TM hat also mindestens die Komplexität $T(n) = C n \sqrt{n}$ für ein $C > 0$.

2. Eine mehrbändige TTM M' , die L erkennt, läßt sich leicht angeben; M' nimmt den Abschnitt u auf ein Arbeitsband und versieht die Trennzeichen 2 jeweils mit einer Marke ($k^2 + k$ Schritte). Nach dem Einlesen eines Blockes $v_i = x_i \bar{j}_i$ aus dem Endabschnitt v springt der Lesekopf vom Beginn von u aus $j_i - 1$ Marken weit nach vorne und ist damit an der richtigen Stelle für den Vergleich von u_{j_i} und x_i . Das Aufsuchen und Vergleichen erfordert höchstens $C_3 k$ Schritte, insgesamt benötigt M' also nur $C_4 k^2 = C_4 n$ Schritte, eine einbändige TTM also (mit einer anderen Konstanten C') $C'n$ Schritte.

Literatur

- [1] GALLAIRE, H.: Recognition of Context-Free Languages by On-Line Turing Machines. Inf. Contr. **15**, 288—295 (1969).
- [2] HENNIE, F. C.: On-Line Turing Machine Computations. IEEE Trans. Electron. Comp. **EC-15**, 35—44 (1966).

*Dipl.-Math. R. Weicker
Technische Universität Berlin
Informationsverarbeitung II
Ernst-Reuter-Platz 8
D-1 Berlin 10
Deutschland*