I. Zeitlich beschränkte Turingmaschinen und polynomiale Reduktion

von Walter Baur

Dieser Vortrag bildet vor allem eine Basis für die nächsten Vorträge. Vom Leser wird erwartet, dass er schon eine Ahnung hat, was Turingmaschinen sind und wie sie arbeiten: Es werden genaue Definitionen angegeben, doch auf Beispiele wird verzichtet und die Beweise werden nur skizziert. Beispiele können in (Hermes 1961) oder in einem etwas abweichenden Formalismus in (Hopcroft, Ullman 1969) gefunden werden.

Im ersten Paragraphen werden die Begriffe Turingmaschine, Turing-Berechnung und zeitlich beschränkte Turingmaschinen definiert. Polynomial entscheidbare Mengen werden im zweiten Paragraphen behandelt. Die dort angegebenen Sätze sollen darauf hinweisen, dass die gebildeten Begriffe im wesentlichen von dem benützten Maschinenmodell unabhängig sind. Im dritten Paragraphen wird der Transformierbarkeitsbegriff eingeführt, der für die meisten Vorträge grundlegend ist.
Schliesslich wird im letzten Paragraphen ein Hierarchiesatz für Klassen zeitlich beschränkt entscheidbarer Mengen bewiesen.

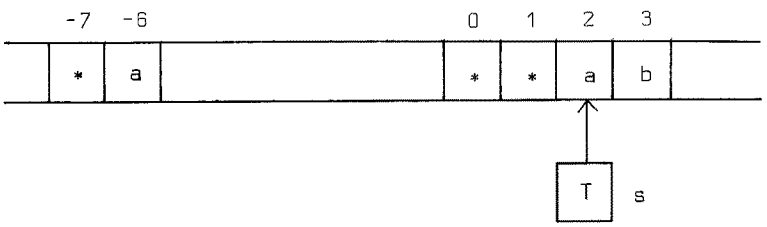
1. Turingmaschinen

Sei $\sum$ ein endliches Alphabet (= Menge) mit $\sum \supseteq \sum_0 = \{0,1,*\}$ und $L, R \notin \sum$. $\sum^*$ sei die Menge der Wörter (= endliche Folgen) über $\sum$ (das Zeichen $*$ in $\{0,1,*\}$ hat nichts mit dem Zeichen $*$ in $\sum^*$ zu tun). Für $X \in \sum^*$ bezeichnet $lh(X)$ die Länge von X. $\emptyset$ bezeichnet die leere Folge.

Definition 1 Eine Turingmaschine (TM oder T-Maschine) über $\sum$ ist ein 4-Tupel $T = \langle S, s_0, s_1, \Phi \rangle$, wobei

S endlich (Zustandsmenge),
$s_0 \in S$ (Anfangszustand),
$s_1 \in S$ (akzeptierender Zustand),
$\Phi: \sum \times (S - \{s_1\}) \longrightarrow (\sum \cup \{L,R\}) \times S$ eine partielle Funktion (Maschinentafel).

Anschauliche Beschreibung: Eine TM T besteht aus einem endlichen Automaten, einem Lese-Schreib-Lösch-Kopf und einem beidseitig unendlichen Band, das in Felder eingeteilt ist:



In jedem Feld des Bandes steht ein Zeichen aus Σ. Die Maschine handelt gemäss ihrer Tafel Φ: Sieht sie im Zustand s das Zeichen a im Feld n und ist $\Phi(a,s) = \langle c,s'\rangle$ mit $c \in \Sigma$, so ersetzt sie in diesem Feld a durch c und geht in den Zustand s'. Ist $\Phi(a,s) = \langle L,s'\rangle$ ($\langle R,s'\rangle$), so geht sie in den Zustand s' und bewegt sich ein Feld nach links (rechts) ohne etwas zu drucken. Ist $\Phi(a,s)$ nicht definiert, so stellt sie ab.

Definition 2 Eine (T-)Konfiguration ist ein Tripel $K = \langle (a_j)_{j\in\mathbb{Z}}, i, s\rangle$ mit $a_j \in \Sigma$ (für alle $j \in \mathbb{Z}$), $i \in \mathbb{Z}$, $s \in S$. ($(a_j)_{j\in\mathbb{Z}}$ ist die momentane Bandinschrift, s der Zustand von T, i die Nummer des Feldes, über dem sich der Kopf von T befindet).

Notation: Ist $K = \langle (a_j)_{j\in\mathbb{Z}}, i, s\rangle$ eine Konfiguration mit $a_j = *$ für $j<j_0$ und $j>j_1$ und mit $j_0 \leq i \leq j_1$, so schreiben wir für K auch

$$a_{j_0} \ldots . a_i \ldots . a_{j_1}$$
$$\uparrow$$
$$i\,|\,s$$

Wir definieren eine zweistellige Relation E_T auf der Menge aller T-Konfigurationen. Sei $K = \langle (a_j)_{j\in\mathbb{Z}}, i, s\rangle$, $K' = \langle (a'_j)_{j\in\mathbb{Z}}, i', s'\rangle$.

$E_T(K,K')$: $\Longleftrightarrow$

$\langle\langle a_i,s\rangle,\langle a'_i,s'\rangle\rangle \in \Phi$ & $i'=i$ & $(\forall j, j\neq i)$ $a'_j = a_j$

oder $\langle\langle a_i,s\rangle,\langle L,s'\rangle\rangle \in \Phi$ & $i'=i-1$ & $(\forall j)$ $a'_j = a_j$

oder $\langle\langle a_i,s\rangle,\langle R,s'\rangle\rangle \in \Phi$ & $i'=i+1$ & $(\forall j)$ $a'_j = a_j$.

Es ist klar, dass es zu jedem K höchstens ein K' gibt mit $E_T(K,K')$. Ein solches K' heisst Folgekonfiguration von K. Gibt es kein solches K', so heisst K Endkonfiguration.

Definition 3 Eine T-Berechnung ist eine endliche Folge $B = \langle K_0, \ldots, K_n\rangle$ von T-Konfigurationen, so dass für alle $i<n$ gilt $E_T(K_i, K_{i+1})$. n heisst die Länge von B, der Zustand von K_n der Endzustand von B. B heisst

vollständig, wenn K_n eine Endkonfiguration ist. Ist $X \in \{0,1\}^*$ und $K_0 = {}^{0}\!\uparrow\!{}^{*}X$, so heisst B eine Berechnung mit Input X.

Bemerkung: Seien B, B' Berechnungen mit demselben K_0. Da jede Konfiguration höchstens eine Folgekonfiguration hat, ist entweder B eine Fortsetzung von B' oder B' eine Fortsetzung von B.

Definition 4 Sei $X \in \{0,1\}^*$. T akzeptiert X, wenn es eine vollständige Berechnung mit Input X und Endzustand s_1 gibt. Die Menge $A(T) = \{X \in \{0,1\}^* \mid$ T akzeptiert X$\}$ heisst die von T akzeptierte Sprache. Sei $f: \{0,1\}^* \longrightarrow \{0,1\}^*$. T berechnet f, wenn es zu jedem $X \in \{0,1\}^*$ eine vollständige Berechnung $\langle K_0,\ldots,K_n\rangle$ gibt mit $K_0 = {}^{0}\!\uparrow\!{}^{*}X$, $K_n = {}^{0}\!\uparrow\!{}^{*}f(X)$.

Bemerkung: Allgemein nennt man eine Teilmenge von $\{0,1\}^*$ eine Sprache. Eine Sprache wird oft mit L bezeichnet. Solche L sind nicht mit dem L ("links") in der Definition der TM zu verwechseln.

Definition 5 Sei t eine Funktion von $\mathbb{N}$ nach $\mathbb{R}$. Eine TM T heisst t-beschränkt, wenn für jeden Input X jede Berechnung mit Input X Länge höchstens $t(lh(X))$ hat. T heisst schliesslich t-beschränkt, wenn es eine Konstante C gibt, so dass für jeden Input X mit $lh(X) \geq C$ jede Berechnung mit Input X Länge höchstens $t(lh(X))$ hat. T heisst polynomial beschränkt, wenn es ein Polynom t gibt so, dass T t-beschränkt ist.

Bemerkungen: 1. Jede schliesslich t-beschränkte TM ist $(t+D)$-beschränkt für eine geeignete Konstante D.
2. Im folgenden nehmen wir an, dass alle auftretenden Zeitfunktionen den Bedingungen $t(0)>0$ und $t(n)\geq n$ für alle n genügen.

2. Polynomiale Berechenbarkeit

(Cobham 1965) und (Edmonds 1965) setzten sich für die Identifikation der praktisch ausführbaren Algorithmen mit den polynomial-beschränkten Algorithmen ein. Cobham's Klasse $\mathcal{L}$ ist die unten definierte Klasse Π der polynomial-berechenbaren Funktionen:

$\Pi = \{f: \{0,1\}^* \longrightarrow \{0,1\}^* \mid$ es gibt eine polynomial beschränkte TM über einem Alphabet Σ, die f berechnet$\}$.

Weiter setzen wir

P = {L⊆{0,1}* | es gibt eine polynomial beschränkte TM T über einem Alphabeth Σ mit A(T) = L}.

Die Mengen aus P heissen polynomial entscheidbare Sprachen.

Bemerkung: Die charakteristische Funktion f einer Menge L⊆{0,1}* ist definiert durch f(X) = 0 falls X∈L und f(X) = Ø (leere Folge) falls X∉L. Man sieht leicht, dass P die Klasse der Mengen mit charakteristischer Funktion in Π ist.

Der folgende Satz zeigt, dass die Mengen Π und P sich nicht ändern, wenn man in ihrer Definition Σ durch Σ_0 ersetzt.

Satz 1 Sei T eine t-beschränkte TM. Dann gibt es eine Konstante C, eine $C(t(n)+n^2)$-beschränkte TM T_1 über Σ_0 und eine $C\cdot t^2$-beschränkte TM T_2 über Σ_0, so dass gilt

(i) $A(T_1) = A(T)$,

(ii) berechnet T eine Funktion, so berechnet T_2 dieselbe Funktion.

Beweisskizze. Sei Σ das Alphabet von T, $Kard\Sigma \leq 2^k$. Kodiere die Elemente von $\Sigma-\{*\}$ durch {0,1}-Folgen der Länge k. Ordne in offensichtlicher Weise jeder Σ-Bandinschrift W eine Σ_0-Bandinschrift W_0 zu. Man konstruiert leicht eine TM T' über Σ_0, so dass für eine geeignete Konstante C_0 gilt: Verändert T die Bandinschrift W in m Schritten zu W', so verändert T' die Bandinschrift W_0 in $C_0 m$ Schritten zu W_0'. Die gesuchte Maschine T_1 arbeitet wie T, ausser dass sie zuerst den Input X präpariert. Letzteres kann in $C_1(n^2+1)$ Schritten (n=lh(X)) gemacht werden. T_1 akzeptiert genau, wenn T akzeptiert. T_2 muss noch das Ergebnis rücktransformieren. Da dessen Länge höchstens $C_0 t(n)$ ist, werden dazu höchstens $C_2 t(n)^2$ Schritte benötigt. Daraus folgt die Behauptung.

Der Begriff der TM lässt sich auf verschiedene Arten verallgemeinern. Die angegebene Formulierung stimmt im Wesentlichen mit der von (Turing 1937) überein. Beispiele: 1. TM mit k Bändern und k Köpfen. 2. TM mit k Köpfen und k d-dimensionalen "Bändern". Dabei nehmen wir den Input eindimensional an.

Im ersten Fall ist Φ eine partielle Funktion von $\Sigma^k\times(S-\{s_1\})$ in $(\Sigma\cup\{L,R\})^k\times S$, im zweiten von $\Sigma^k\times(S-\{s_1\})$ in $(\Sigma\cup\{L_1,\ldots,L_d,R_1,\ldots,R_d\})^k\times S$.

Die nächsten Sätze zeigen, dass die Mengen P und Π nicht von dem in ihrer Definition verwendeten Maschinentyp abhängen. Dies ist ein Hinweis auf die mathematische Bedeutung dieser Mengen.

Satz 2 Sei T eine t-beschränkte k-bändige TM. Dann gibt es eine Konstante C und eine Ct^2-beschränkte (einbändige) TM T', welche dieselbe Sprache wie T akzeptiert.

Beweisskizze. T' arbeitet über dem Alphabet $\Sigma' = \Sigma^{2k}$, wobei Σ das Alphabet von T ist. Das Band von T' denken wir uns eingeteilt in 2k Spuren, in denen die verschiedenen Komponenten der Elemente von Σ' stehen. Die ersten k Spuren werden dazu benützt, den Inhalt der k Bänder von T darzustellen, und in den restlichen Spuren wird die Position der Köpfe von T festgehalten.
T' simuliert T schrittweise. Da sich nach s Schritten von T alle Köpfe von T höchstens s Felder vom Ursprung entfernt haben, braucht T' zur Simulation des (s+1)-ten Schrittes von T höchstens $C_0(s+1)$ Schritte, wobei C_0 eine von s unabhängige Konstante ist. Im ganzen werden also für ein geeignetes C höchstens $Ct(n)^2$ Schritte benötigt (n= Länge des Inputs).

Satz 3 Sei T eine t-beschränkte TM mit je einem Kopf auf k d-dimensionalen Bändern. Dann gibt es eine Konstante C und eine $C(t^2\log_2 t+1)$- beschränkte TM T', welche dieselbe Sprache wie T akzeptiert.

Beweisskizze für d=2, k=1. Wir geben eine Beschreibung einer T simulierenden einbändigen TM T', wobei wir Details wie das Präparieren des Inputs dem Leser überlassen. T' arbeitet über einem Alphabet, dessen Zeichen Sechstupel sind; ihre Komponenten stehen wie im Beweis von Satz 2 in verschiedenen Spuren des Bandes.
Der wesentliche Teil einer Berechnung der ebenen Maschine T ist der von ihrem Kopf durchlaufene Pfad, d.h. die Folge der Symbole, die T sieht. T' reproduziert im wesentlichen diesen Pfad nach links in der 1. Spur des Bandes und schreibt in die 2. Spur unter jedes Feld des Pfades eins der Zeichen N, S, E, W, welches angibt, ob T das entsprechende Feld der Ebene in Richtung Nord, Süd, Ost oder West verlassen hat. In den restlichen 4 Spuren (im folgenden N-, S-, E- und W-Zähler genannt) wird gezählt.
Vor dem i-ten Simulationsschritt steht der Kopf von T' über dem linken Endfeld der Bandinschrift. In der 1. Spur dieses Feldes steht dasjenige

Zeichen, welches T vor dem i-ten Schritt sieht; seine übrigen Spuren sind leer. Druckt T im Schritt i ein Zeichen, so druckt T' dasselbe Zeichen in die 1. Spur dieses Feldes und geht zum nächsten Simulationsschritt über. Bewegt sich T im Schritt i, so hält T' zunächst die Bewegungsrichtung in der 2. Spur fest, druckt eine 1 in den entsprechenden Zähler und bewegt sich anschliessend mit Blick auf die 2. Spur nach rechts, um die Inschrift des Feldes zu suchen, auf dem T jetzt steht. Vor jedem Schritt nach rechts stehen in den Spuren 3 bis 6 gewisse Zahlen in Dualdarstellung, beginnend im Feld unter dem Kopf, welche die Anzahl der bereits überstrichenen Felder, in deren 2. Spur N (bzw. S, E, W) steht, festhalten. Findet T' ein Feld, so dass gleichzeitig die Zahl im N-Zähler mit derjenigen im S-Zähler und die Zahl im E-Zähler mit derjenigen im W-Zähler übereinstimmt, so merkt sich T' das Zeichen a in der 1. Spur dieses Feldes, kehrt an den Anfang zurück und druckt a in die 1. Spur des Feldes j unmittelbar links des Ausgangsfeldes. Andernfalls kehrt T' nach Erreichen des rechten Endes der Bandinschrift um und geht zum selben Feld j, aber ohne etwas zu drucken. In beiden Fällen beginnt jetzt der nächste Simulationsschritt.
Man überlegt sich leicht, dass T' das Gewünschte leistet.

Bemerkung Martin Fürer [5] hat gezeigt, dass man in Satz 3 den Faktor $\log_2 t$ sogar weglassen kann.

3. Polynomiale Transformierbarkeit

Definition 6 Seien $L,M\subseteq\{0,1\}^*$. L heisst polynomial transformierbar in M ($L\leq_\pi M$), wenn es ein $f\in\Pi$ gibt mit $X\in L \iff f(X)\in M$ für alle $X\in\{0,1\}^*$.

Bemerkungen: 1. $\{0,1\}^*\leq_\pi M$ gilt genau, wenn $M\neq\emptyset$.
2. $\emptyset\leq_\pi M$ gilt genau, wenn $M\neq\{0,1\}^*$.

Trivialerweise gilt

(i) $L\leq_\pi L$,

(ii) $L\leq_\pi M \;\&\; M\leq_\pi N \Longrightarrow L\leq_\pi N$,

(iii) $L\leq_\pi M \;\&\; M\in P \Longrightarrow L\in P$.

Die durch

$$L\equiv_\pi M \iff L\leq_\pi M \;\&\; M\leq_\pi L$$

definierte Relation $\equiv_\pi$ ist eine Aequivalenzrelation (polynomiale

Aequivalenz). Die Menge aller polynomial entscheidbarer Sprachen ohne $\emptyset$, $\{0,1\}^*$ ist eine einzige Aequivalenzklasse. Die Aequivalenzklassen bilden bezüglich $\leq_\pi$ einen Halbverband: Sind $L,M\subseteq\{0,1\}^*$, nicht beide gleich $\{0,1\}^*$, so ist die Aequivalenzklasse von $\{0W \mid W\in L\}\cup\{1W \mid W\in M\}$ ein Supremum der Aequivalenzklassen von L und M. Dagegen haben $\emptyset$ und $\{0,1\}^*$ kein Infimum!

4. Ein Hierarchiesatz

Sei $n\in\mathbb{N}$. Mit $\overline{n}$ bezeichnen wir die Dualdarstellung von n, d.h. das Wort $a_0\ldots a_m$, wobei $n=\sum_{\mu=0}^{m} a_\mu 2^\mu$, $a_\mu\in\{0,1\}$ und $a_m=1$ falls $m>0$.

Satz 4 (vgl. (Hopcroft-Ullman 1969)) Sei $t:\mathbb{N}\to\mathbb{N}$ mit $t(n)\geq n^2$ so, dass es eine t-beschränkte TM über $\sum_0$ gibt, welche die Funktion $X\mapsto\overline{t(lh(X))}$ $(X\in\{0,1\}^*)$ berechnet. Dann gibt es eine Konstante C und eine $C(t\log_2 t+1)$-beschränkte TM T über $\sum_0$, so dass für jede schliesslich t-beschränkte TM T' über $\sum_0$ gilt $A(T')\neq A(T)$.

Bemerkungen: 1. Alle "genügend rasch wachsenden, genügend einfach zu berechnenden" Funktionen t erfüllen die Voraussetzung von Satz 3. Beispiele: $t(n)=n^2, n^3,\ldots,2^n$.
2. Satz 4 liefert Sprachen mit ziemlich genau bestimmtem, hohem Schwierigkeitsgrad.

Beweisskizze. Grobe Beschreibung von T: T fasst ein Anfangsstück des Inputs X (lh(X)=n) als Kodierung einer Maschine T' auf und simuliert t(n) Schritte von T' mit Input X. Falls T' in t(n) Schritten zu einem Ergebnis kommt, so akzeptiert T genau wenn T' verwirft. Da die Arbeitsweise von T' in T nicht verdrahtet ist, muss T vor jedem Simulationsschritt die Kodierung von T' studieren. Ist T' schliesslich t-beschränkt, so gilt nach Konstruktion $X\in A(T) \iff X\notin A(T')$, sofern X lang genug ist. Da unsere Kodierungen so sein werden, dass es zu jeder TM T' unendlich viele Inputs X gibt, deren Anfangsstücke T als Kodierung von T' auffasst, folgt daraus $A(T')\neq A(T)$ für alle schliesslich t-beschränkten Turingmaschinen T'.

Kodierung: Jedem Element von $\{L,R,*\}\cup\mathbb{N}$ ordnen wir eine Folge von 1'en zu: $L\to 1$, $R\to 11$, $*\to 111$, $0\to 1111$, $1\to 11111$, usw. Die Turingmaschinen $T'=\langle S,s_0,s_1,\Phi\rangle$ mit $S\subseteq\mathbb{N}$, $s_0=0$, $s_1=1$ kodieren wir durch endliche $\{0,1\}$-Folgen. Wir verzichten auf eine genaue Beschreibung dieser

Kodierung und geben stattdessen ein Beispiel:

T', gegeben durch die Maschinentafel

	0	1	*
$(s_0=)0$	<*,0>		<L,1>
$(s_1=)1$			
$(s_2=)2$		<1,0>	

wird zunächst beschrieben durch

$$(T') = \underbrace{11101111001001011111}_{\text{1. Block*}}\ \underbrace{0001001001}_{\text{2. Block}}\ \underbrace{000100111110111100010000}_{\text{3. Block}}$$

(* entspricht 1. Zeile der Maschinentafel)

Eine Kodierung von T' ist nun eine {0,1}-Folge der Gestalt (T')X, $X \in \{0,1\}^*$.

T arbeitet über einem Alphabet, dessen Zeichen 4-Tupel sind; ihre Komponenten schreiben wir wie früher in verschiedene Spuren des Bandes. (Mit Satz 1 können wir am Schluss T durch eine äquivalente Maschine über $\sum_0$ ersetzen). In der 1. Spur arbeitet T wie die zu simulierende Maschine T', deren Kodierung in der 3. Spur steht. In der 2. Spur (Zähler) werden die Schritte von T' gezählt, und in der 4. Spur steht eine Marke, welche den momentanen Zustand von T' bezeichnet. Die Kodierung von T' und der Zähler stehen vor jedem Simulationsschritt unmittelbar rechts des Kopfes von T. (Andernfalls würde nämlich T allzu lange brauchen, um die Kodierung von T' zu finden).

a) "Input präparieren".
T kopiert den Input X der Länge n in die 1. Spur und berechnet $\overline{t(n)}$ in der 2. Spur. Sei $k = \lfloor \sqrt{lh(\overline{t(n)})} \rfloor$. T kopiert die ersten min(k,n) Zeichen der 1. Spur in die 3. Spur und prüft nach, ob dies eine Kodierung einer Maschine T' ist. Ist dies der Fall, so setzt T eine 0 ins 1. Feld der 4. Spur. Andernfalls akzeptiert T.

b) Simulation des i-ten Schrittes von T'.
Sei K = <w,j,s> die (i-1)-te Konfiguration der Berechnung von T' mit Input X. T hat bereits i-1 Schritte von T' simuliert. Der Kopf von T steht auf dem Feld j. In der ersten Spur steht w, in der zweiten bzw. dritten $\overline{t(n)-(i-1)}$ bzw. die Kodierung von T', beginnend im Feld j+1. In der vierten Spur steht eine 0 unter der ersten 1 desjenigen Blocks der

Kodierung von T', der dem Zustand s (Zeile s der Maschinentafel) entspricht. Falls nun im Zähler $\overline{0}$ steht, so akzeptiert T. Andernfalls sucht T die Marke in der 4. Spur und findet so heraus, was T' als nächstes tut. Stellt T' ab, so stellt T ebenfalls ab und akzeptiert genau, wenn T' verwirft. Geht T' in den Zustand s', so verschiebt T die 0 der 4. Spur unter die erste 1 des s' entsprechenden Blocks der 3. Spur. Druckt T' ein Zeichen im Feld j, so druckt T dasselbe Zeichen in dieses Feld (in der 1. Spur). Geht T' nach links (rechts), so verschiebt T den Inhalt der Spuren 2, 3, 4 um ein Feld nach links (rechts). Schliesslich verkleinert T den Inhalt des Zählers um 1.

Anzahl Schritte von T (für Inputs von genügend grosser Länge n):
für a): $\leq C_1 t(n)$,
für b): $\leq C_2(k^2+\log_2 t(n)) \leq C_2' \log_2 t(n)$
(k^2 rührt vom Verschieben der Marke in Spur 4 bei Zustandsänderung von T' her)

Da T höchstens t(n) Schritte von T' simuliert, ist T $C(t\log_2+1)$-beschränkt für eine geeignete Konstante C.
Sei nun T' eine beliebige schliesslich t-beschränkte TM über Σ_0. Wähle $n \in \mathbb{N}$ so, dass T' t-beschränkt ist für alle Inputs der Länge $\geq n$, und dass $lh(T')) \leq \min(n, \lfloor\sqrt{lht(n)}\rfloor)$. Setze $X = \underbrace{(T')0\ldots0}_{\text{Länge } n}$. Da T' mit Input X in $\leq t(n)$ Schritten zu einem Ergebnis kommt, gilt nach Konstruktion von T $\in X$ $A(T) \Longleftrightarrow X \notin A(T')$, also $A(T) \neq A(T')$.

Bemerkung: Der Beweis (Wahl der Kodierung) zeigt, dass es für alle genügend grossen n ein X gibt mit $lh(X)=n$ und $X \in A(T) \Longleftrightarrow X \notin A(T')$.

Literatur

Cobham, A., The intrinsic computational difficulty of functions, Logic, Methodology and Philosophy of Science (Bar-Hillel, ed.), Amsterdam, 1965.

Edmonds, J., Paths, trees, and flowers, Canadian Journal of Math. 17 (1965) 449-467.

Hermes, H., Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit, Springer, 1961.

Hopcroft, J.E. and Ullman, J.D., Formal languages and their relation to automata, Addison-Wesley, Reading, Massachusetts, 1969.

Turing, A.M., On computable numbers with an application to the Entscheidungsproblem, Proc. London Math. Soc. 2/42 (1937) 230-265.