## VIII. Untere Schranken für die Komplexität log. Entscheidungsprobleme

von Joos Heintz

In diesem Vortrag werden die wichtigsten Ergebnisse und Beweise aus (Fischer-Rabin 1974) dargestellt.
Es wird unter anderem gezeigt, dass für gewisses c>o die Theorie der Presburgerarithmetik eine Entscheidungskomplexität > $2^{2^{cn}}$ hat. Das bedeutet, dass keine in der Zeit $2^{2^{cn}}$ arbeitende Turingmaschine existiert, die - bei geeigneter Kodierung der Formeln der Sprache der Presburgerarithmetik als Inputs von Turingmaschinen - die Presburgerarithmetik entscheidet.
Weil das Resultat auch für nichtdeterministische Turingmaschinen gilt, ergibt sich, dass man für die Sätze der Presburgerarithmetik i.a. keine kurzen Beweise finden kann. Genauer: Es gibt beliebig grosse n und Sätze der Länge n dazu, deren Beweise alle länger als $2^{2^{cn}}$ sind.

Die Methoden, die dabei verwendet werden, beruhen auf der Entdeckung von Meyer und Stockmeyer, dass sich die Gödelschen Unentscheidbarkeitsbeweise der mathematischen Logik zu Schwerentscheidbarkeitsbeweisen entscheidbarer Theorien umgestalten lassen.

Diese Methoden lassen sich auch auf gewisse andere Theorien, deren Modelle Gruppen sind, anwenden. Es stellt sich heraus, dass diese Theorien eine Entscheidungskomplexität > $2^{cn}$ haben.

Wo nichts anderes gesagt wird, soll unter "Turingmaschine" immer eine nichtdeterministische Turingmaschine verstanden werden, die über dem Alphabet {0,1,*} arbeitet und Inputs akzeptiert oder verwirft. "Input" bedeutet im folgenden immer ein Wort $X \in \{0,1\}^*$, wobei X = 0 ist oder X mit 1 beginnt. Seine Länge bezeichnen wir mit $|X|$. A(T) bezeichnet die Menge der von der Turingmaschine T akzeptierten Inputs. Die Maschine T stoppt, wenn sie im akzeptierenden Zustand ist, unabhängig davon, was auf dem Band steht.
Soweit wir über Turingmaschinen reden, schliessen wir uns der Terminologie und Bezeichnungsweise von I und II an mit folgender Abweichung: Unter $2^{2^{cn}}$- bzw. $2^{cn}$-beschränkten Turingmaschinen verstehen wir Turingmaschinen, die schliesslich $2^{2^{cn}}$- bzw. $2^{cn}$-beschränkt sind.

Unter polynomial berechenbaren Abbildungen verstehen wir stets Abbildungen der Menge aller Inputs in sich, die durch polynomial beschränkte deterministische Turingmaschinen berechenbar sind.

Sei S die Menge aller Symbole der Sprache $\mathcal{L}$ der Presburgerarithmetik. S besteht aus 0,1,+ und den üblichen prädikatenlogischen Zeichen. Sei $\varphi$: S $\longrightarrow$ Menge aller Inputs $\neq$ 0, injektiv und so, dass jedes $X \in \{0,1\}^*$ auf höchstens eine Weise als Konkatenation von Elementen aus dem Bild von $\varphi$ aufgefasst werden kann. $\varphi$ induziert einen Konkatenationshomomorphismus $\varphi^*$: $S^* \longrightarrow \{0,1\}^*$. Mit $Th_+$ bezeichnen wir die Menge aller Inputs der Form $\varphi^*(F)$, wo F eine Formel ohne freie Variable der Sprache $\mathcal{L}$ ist, welche in der Struktur $\langle \mathbb{N},0,1,+\rangle$ gilt. Sei $L_+ = \varphi^*(\mathcal{L})$.
Damit lautet der Satz, den wir beweisen wollen, folgendermassen:

Satz 1 Es gibt ein c>o, so dass keine $2^{2^{cn}}$-beschränkte Turingmaschine T existiert mit $Th_+ = A(T)$.

Satz 1 folgt aus

Satz 2 Zu jeder $2^{2^n}$-beschränkten Turingmaschine T gibt es eine Abbildung, die jedem Input X eine Formel $F_X \in L_+$ ohne freie Variable zuordnet mit folgenden Eigenschaften:

(i) $|F_X| = O(|X|)$

(ii) $X \longmapsto F_X$ ist polynomial berechenbar

(iii) $F_X \in Th_+ \Longleftrightarrow X \in A(T)$

Beweis von Satz 1 aus Satz 2: Zunächst betrachten wir den Fall, dass T deterministisch ist.
Nach I, Satz 4 gibt es zu beliebigen d>d'>o eine $2^{2^{dn}}$-beschränkte deterministische Turingmaschine T mit der Eigenschaft, dass keine $2^{2^{d'n}}$-beschränkte deterministische Turingmaschine T' mit $A(T) = A(T')$ existiert.
Wir wählen für d=1 und ein o<d'<1 eine solche Turingmaschine T und dazu nach Satz 2 eine Abbildung $X \longmapsto F_X$, die für ein geeignetes Polynom f durch eine f-beschränkte deterministische Turingmaschine B berechenbar sei und für die $|F_X| \leq L|X|$ für eine geeignete Konstante L gelte.

Wäre die Aussage des Satzes 1 für deterministische Turingmaschinen falsch, so gäbe es zu jedem c eine $2^{2^{cn}}$-beschränkte deterministische Turingmaschine, die $Th_+$ entscheidet.
Wähle c so, dass cL<d' ist. Dann gilt für grosse n:

$$2^{2^{cLn}} + f(n) \leq 2^{2^{d'n}}$$

Wähle eine $2^{2^{cn}}$-beschränkte, deterministische Turingmaschine, die $Th_+$

entscheidet. Sei T' die Turingmaschine, die wir erhalten, wenn wir auf einen Input X zuerst B und dann die Entscheidungsmaschine anwenden.

T' ist $(2^{2^{cLn}}+f(n))$-beschränkt, also auch $2^{2^{d'n}}$-beschränkt
und wegen

$$X \in A(T') \iff F_X \in Th_+ \iff X \in A(T)$$

gilt

$$A(T') = A(T).$$

Widerspruch.
Damit ist Satz 1 im deterministischen Fall bewiesen.

Sei nun T nichtdeterministisch:
Mit demselben Beweis wie in I, Satz 4 kann man zeigen:
Zu beliebigen $d>d'>0$ gibt es eine $2^{2^{dn}}$-beschränkte nichtdeterministische Turingmaschine T mit der Eigenschaft, dass keine $2^{2^{d'n}}$-beschränkte, nichtdeterministische Turingmaschine T' mit

$$A(T') = \text{Komplement von } A(T)$$

existiert.
Der Beweis von Satz 1 im nichtdeterministischen Fall verläuft nun völlig analog dem Beweis im deterministischen Fall bis auf folgende Modifikation:

Die Turingmaschine B berechne $X \mapsto \neg F$.
Daraus ergibt sich dann

$$A(T') = \text{Komplement von } A(T)$$

und damit ein Widerspruch.

Der Beweis von Satz 2 geht ungefähr so: Möglichst grosse Anfangsstücke der Multiplikationsrelation auf den natürlichen Zahlen werden mit möglichst kurzen Formeln der Presburgerarithmetik definiert. Mit Hilfe dieser Formeln lässt sich der Beweis des Satz 2 entsprechenden Satzes in der Theorie der natürlichen Zahlen mit Multiplikation auf die Presburgerarithmetik übertragen.

Wir gliedern den Beweis von Satz 2 in verschiedene Lemmas auf. Wir beginnen mit folgender
Bemerkung: Sei $a \in \mathbb{N}$. Jede natürliche Zahl $x<a^2$ besitzt eine Darstellung

$$x = x_1x_2 + x_3 + x_4$$

mit natürlichen Zahlen $x_1, x_2, x_3, x_4<a$.

Lemma 1 Zu jedem $n \in \mathbb{N}$ existiert eine Formel $M_n(x,y,z)$ und eine Zahl $p_n \geq 2^{2^{2^n}}$ mit folgenden Eigenschaften:

(i) $|M_n(x,y,z)| = O(n)$

(ii) $X \mapsto M_{|X|}(x,y,z)$ ist polynomial berechenbar

(iii) $M_n(x,y,z)$ bedeutet "$x,y,z \leq p_n$ und $xy = z$".

Ohne es jedesmal zu erwähnen, meinen wir mit Formeln, die wir mit einem Index n schreiben immer Formelfolgen, die (i) und (ii) erfüllen.

Beweis Wir konstruieren zuerst rekursiv Formeln $M_n^*(x,y,z)$, die "$x<2^{2^n}$ und $xy = z$" bedeuten.

Sei

$$M_0^*(x,y,z) \equiv (x=0 \wedge z=0) \vee (x=1 \wedge y=z).$$

Zur Konstruktion von $M_{n+1}^*$ aus $M_n^*$ denken wir uns natürliche Zahlen, die wir für den Moment mit x,y,z bezeichnen, gegeben. Nach der Bemerkung gilt $x<2^{2^{n+1}}$ und $xy = z$ genau dann, wenn es natürliche Zahlen $x_1, x_2, x_3, x_4 < 2^{2^n}$ gibt mit

$$x = x_1x_2 + x_3 + x_4 \quad \text{und} \quad z = x_2(x_1y) + x_3y + x_4y.$$

Da $M_n^*$ die Multiplikation von Zahlen (mit der Einschränkung, dass eine davon $< 2^{2^n}$ ist) beschreibt, hat folgende Formel $M_{n+1}^*$ jedenfalls die gewünschte Bedeutung:

$$M_{n+1}^*(x,y,z) \equiv (\exists x_1,x_2,x',x_3,x_4,z_1,z_2,z_3,z_4)((\forall u,v,w)(((x_1=u \wedge x_2=v \wedge x'=w) \vee$$
$$\vee (x_1=u \wedge y=v \wedge z_1=w) \vee (x_2=u \wedge z_1=v \wedge z_2=w) \vee (x_3=u \wedge y=v \wedge z_3=w) \vee$$
$$\vee (x_4=u \wedge y=v \wedge z_4=w)) \longrightarrow M_n^*(u,v,w)) \wedge x=x'+x_3+x_4 \wedge z=z_2+z_3+z_4).$$

Man überlegt sich leicht, dass man bei der Konstruktion der Folge $M_n(x,y,z)$ mit einer endlichen Menge E von Variablen auskommt, d.h. man kann die gebundenen Variablen $x_1,\ldots,w$ immer aus E wählen.
Weil zusätzlich $M_n^*$ in $M_{n+1}^*$ nur einmal vorkommt, folgt, dass (ii) erfüllt ist.
(iii) sei dem Leser überlassen.

Sei $p_n'$ das kleinste gemeinsame Vielfache aller Zahlen $< 2^{2^n}$.
$p_n'$ ist durch

$$P'_n(x) \equiv (\forall u)((M^*_n(u,0,0) \rightarrow (\exists v)M^*_n(u,v,x)) \wedge$$
$$\wedge (\forall w)(((\forall u)M^*_n(u,0,0) \rightarrow (\exists v)M^*_n(u,v,w)) \rightarrow w \geq x))$$

definierbar.
($w \geq x$ brauchen wir hier als Abkürzung für $(\exists z)x+z = w$).

Wir schätzen nun $p'_n$ ab:
Sei

$$p'_n = \prod_{\substack{q \text{ prim} \\ q^i < 2^{2^n} \leq q^{i+1}}} q^i$$

die Zerlegung von $p'_n$ in Primfaktoren.
Für die $q^i$ gilt $2^{2^{n-1}} \leq q^i$.
Nach Tschebyscheff's Satz (oder dem Primzahlsatz) (Chandrasekharan 1968) gibt es ein $F>0$ mit

$$F \frac{2^{2^n}}{2^n} \leq \#\{q \mid q \text{ prim}, q<2^{2^n}\}.$$

Daher ist $p'_n \geq 2^{(\frac{1}{2}F)2^{2^n}}$.

Als nächstes definieren wir zwei Prädikate, die wir zur Konstruktion von $M_n(x,y,z)$ benötigen.
Sei

$$Rem_n(x,y,z) \equiv (\exists u,v)(M^*_n(y,u,v) \wedge x=v+z \wedge z<y).$$

$Rem_n(x,y)$ bedeutet "$y<2^{2^n}$ und z ist Rest von x mod y".
Damit konstruieren wir die Formel

$$M'_n(x,y,z) \equiv ((\exists u)(P'_n(u) \wedge u>x \wedge u>y \wedge u>z) \wedge$$
$$\wedge (\forall v)(M^*_n(v,0,0) \rightarrow (\exists x_1,y_1,z_1)(Rem_n(x,v,x_1) \wedge$$
$$\wedge Rem_n(y,v,y_1) \wedge Rem_n(z,v,z_1) \wedge M^*_n(x_1,y_1,z_1)))),$$

die "$x,y,z<p'_n$ und $xy = z$" bedeutet.
Wir wählen nun $K \in \mathbb{N}$ so, dass $p'_{Kn} > 2^{2^{2^n}}$ für alle $n>0$ gilt und setzen

$$M_n(x,y,z) \equiv M'_{K(n+1)}(x,y,z)$$
$$p_n = p'_{K(n+1)}{}^{-1}$$

$M_n$ erfüllt alle Forderungen von Lemma 1.

Seien $a,b \in \mathbb{N}$ mit Binärdarstellungen $a_p \ldots a_0$ und $b_q \ldots b_0$ gegeben. $a \underset{\smile}{b}$ bezeichne die Zahl mit Binärdarstellung $a_p \ldots a_0\, b_q \ldots b_0$, also $a \underset{\smile}{b} = 2^{q+1}\, a+b$.

Lemma 2 Die Relation

$$"x,y,z \leq p_n \text{ und } x \underset{\smile}{y} = z"$$

lässt sich durch eine Formelfolge $Coc_n(x,y,z)$ beschreiben.

Beweis Sei

$$Pot_n(x,y) \equiv (x \leq y \wedge x+x>y \wedge M_n(y,0,0) \wedge$$
$$\wedge (\forall u)(((\exists v) M_n(u,v,x) \wedge u>1) \rightarrow (\exists w) w+w=u))$$

$Pot_n(x,y)$ bedeutet "$y \leq p_n$ und es gibt ein $q \in \mathbb{N}$ mit $x = 2^q \leq y < 2^{q+1}$".

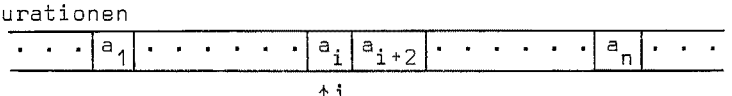
Damit konstruieren wir

$$Coc_n(x,y,z) \equiv ((\exists u,v)((Pot_n(u,y) \wedge M_n(u+u,x,v) \wedge z=v+y) \vee$$
$$\vee (y=0 \wedge x=z)) \wedge M_n(z,0,0))$$

Wir wählen eine injektive Abbildung $\sigma: A \rightarrow \mathbb{N}$, so dass sich die Binärdarstellung jeder Zahl auf höchstens eine Weise als Konkatenation der Binärdarstellungen von Elementen aus $\sigma(A)$ auffassen lässt.
$\sigma$ induziert einen injektiven Konkatenationshomomorphismus $A^* \rightarrow \mathbb{N}$, den wir ebenfalls mit $\sigma$ bezeichnen.
Dann gilt: es gibt ein $K > 0$, so dass $\sigma(W) \leq 2^{K|W|}$ für alle $W \in A^*$.
$\sigma(W)$ nennen wir Gödelnummer von $W \in A^*$.

Wir gehen nun zur Simulation von Turingmaschinen in der Presburgerarithmetik über.
Sei T eine Turingmaschine, die über dem Alphabet {0,1,*} arbeitet. Konfigurationen und Folgen von Konfigurationen von T beschreiben wir durch Wörter über dem Alphabet A = {0,1,*,s,|}. Die Zustände von T beschreiben wir durch positive Potenzen von s, den Anfangszustand durch s selbst, den akzeptierenden Zustand durch $s^2$.

Konfigurationen




mit $a_k \in \{0,1,*\}$ beschreiben wir durch $a_1 \ldots a_i s^j a_{i+1} \ldots a_n$, wobei wir durch eventuelles Anfügen von * dafür sorgen, dass $s^j$ nie am Rande steht.

Endliche Folgen von Konfigurationen, beschrieben durch

$$a_1^1 \ldots a_{i_1}^1 s^{j_1} \ldots a_{n_1}^1, \ldots, a_1^m \ldots a_{i_m}^m s^{j_m} \ldots a_{n_m}^m$$

kodieren wir als

$$|a_1^1 \ldots a_{i_2}^1 s^{j_1} \ldots a_{n_1}^1| \ldots\ldots |a_1^m \ldots a_{i_m}^m s^{j_m} \ldots a_{n_m}^m|.$$

Lemma 3 Die Relation

"$x,y,z \leq p_n$ und es gibt $W_1, W_2 \in A^*$ mit $x=\sigma(W_1)$, $y=\sigma(W_2)$, $z=\sigma(W_1W_2)$"

lässt sich durch eine Formelfolge $C_n(x,y,z)$ beschreiben.

Analog gibt es Formelfolgen $C_n(x,y,z,u)$ und $C_n(x,y,z,u,v)$ die Konkatenation von drei bzw. vier Wörtern beschreiben.

Beweis Sei

$$G_n(x) \equiv (((\exists r) \bigvee_{a \in A} Coc_n(\sigma(a),r,x) \wedge$$

$$\wedge (\forall u)(((\exists v) Coc_n(v,u,x) \wedge (\exists w)(\bigvee_{a \in A} Coc_n(\sigma(a),w,u) \wedge w>o)) \rightarrow$$

$$\rightarrow (\exists w') \bigvee_{a,a \in A} Coc_n(\sigma(aa'),w;u))) \vee x=0)$$

$G_n(x)$ bedeutet "$x<p_n$ und x liegt im Bild von $\sigma$".
Damit konstruieren wir

$$C_n(x,y,z) \equiv (G_n(x) \wedge G_n(y) \wedge Coc_n(x,y,z)).$$

Beweis von Satz 2 Sei die Turingmaschine T gegeben.
Wir konstruieren eine polynomial berechenbare Abbildung, die jedem Input X eine Formel $R_{\sigma(X)}(x)$ mit

$$|R_{\sigma(X)}(x)| = O(|X|)$$

zuordnet, welche $\sigma(X)$ definiert. Ferner konstruieren wir eine Formel $B_n(x,y)$ die, wenn x Gödelnummer eines Inputs X ist, bedeutet:

"$x,y \leq p_n$ und y ist Gödelnummer einer Berechnung von T, die X akzeptiert".

Für den Moment nehmen wir an, dass wir $R_{\sigma(X)}(x)$ und $B_n(x,y)$ bereits haben.
Sei X ein Input und $Y \in A^*$ beschreibe eine Berechnung der Länge $\leq 2^{2^{|X|}}$ von T, die X akzeptiert.
$\ell$ sei die maximale Länge derjenigen Wörter von $A^*$, die Zustände von T beschreiben. Dann haben die Wörter von $A^*$, die Konfigurationen der Berechnung beschreiben,

$$\text{Länge} \leq |X| + 2^{2^{|X|}} + \ell.$$

Y ist daher Konkatenation von $\leq 2^{2^{|X|}}$ Wörtern der

$$\text{Länge} \leq |X| + 2^{2^{|X|}} + \ell + 2,$$

also gilt

$$|Y| \leq 2^{2|X|}(|X|+2^{2|X|}+\ell+2) \leq 2^{2^{E|X|}}$$

für geeignetes E>o.

Daher gilt für hinreichend grosses M>o

$$\sigma(Y) \leq 2^{2^{2^{M|X|}}}$$

Aus den Eigenschaften von $B_n(x,y)$ und $R_{\sigma(X)}(x)$ ergibt sich, dass

$$F_X \equiv (\exists x)(R_{\sigma(X)}(x) \wedge (\exists y)B_{M|X|}(x,y))$$

die in Satz 2 verlangten Eigenschaften besitzt.

Konstruktion von $R_{\sigma(X)}(x)$: Wir definieren zunächst rekursiv zu jedem m,n mit $m<2^n$ eine Formel

$$Q_{m,n}(x,y) \text{ mit } |Q_{m,n}(x,y)| = O(n), \text{ die}$$

"$x = m$ und $y = 2^n$"

bedeutet.

Wir setzen

$$Q_{o,o}(x,y) \equiv (x=0 \wedge y=1).$$

Sei n fest und $Q_{m,n}(x,y)$ für alle $m<2^n$ erklärt. Wir definieren $Q_{m,n}(x,y)$ für alle $m<2^{n+1}$.

Falls $m<2^n$, setzen wir

$$Q_{m,n+1}(x,y) \equiv (\exists v)(Q_{m,n}(x,v) \wedge y=v+v),$$

falls $m\geq 2^n$ setzen wir

$$Q_{m,n+1} \equiv (\exists u,v)(Q_{m-2^n,n}(u,v) \wedge x=u+v \wedge y=v+v)$$

Bei der rekursiven Definition von $Q_{m,n}$ kommt man mit einem endlichen Vorrat an Variablen aus.

Daher gilt $|Q_{m,n}(x,y)| = O(n)$.

Sei n so, dass $2^{n-1} \leq \sigma(X)<2^n$ gilt.

$$R_{\sigma(X)}(x) \equiv (\exists y)Q_{\sigma(X),n}(x,y)$$

hat die geforderten Eigenschaften.

Der Konstruktion von $B_n(x,y)$ schicken wir zwei Hilfsformeln voraus:

Sei

$$AK_n(x,y) \equiv ((x=\sigma(0) \wedge C_n(x,\sigma(s*),y)) \vee (\exists u)(C_n(\sigma(1),u,x) \wedge$$
$$\wedge C_n(\sigma(1s),u,\sigma(*),y))).$$

$AK_n(x,y)$ bedeutet, falls x Gödelnummer eines Inputs X ist:
"$x,y \leq p_n$ und y ist Gödelnummer der zu X gehörigen Anfangskonfiguration".
Sei P die Uebergangsrelation von T. Mit den Bezeichnungen von II sei

$$FK_n(x,y) \equiv (\exists u,v)\Big( \bigvee_{((a,s^i),(a';s^j)) \in P} (C_n(u,\sigma(as^i),v,x) \wedge C_n(u,\sigma(a's^j),v,y))$$
$$\vee \bigvee_{((a,s^j),(L,s^j)) \in^n P} (C_n(u,\sigma(as^i),v,x) \wedge C_n(\sigma(*),u,\sigma(s^j a),v,y))$$
$$\vee \bigvee_{((a,s^i),(R,s^j)) \in^n P} (C_n(u,\sigma(s^i a),v,x) \wedge C_n(u,\sigma(as^j),v,\sigma(*),y)))$$

$FK_n(x,y)$ bedeutet, falls x Gödelnummer einer Konfiguration ist:

"$x,y \leq p_n$ und y ist Gödelnummer einer Nachfolgekonfiguration der x entsprechenden Konfiguration".

Schliesslich sei

$$B_n(x,y) \equiv ((\exists r,t)(AK_n(x,r) \wedge C_n(\sigma(|),r,\sigma(|),t,y)) \wedge$$
$$\wedge (\forall p;q') \neg C_n(p;\sigma(\|),q',y) \wedge (\forall u,p,q)((C_n(p,\sigma(|),u;\sigma(|),q,y) \wedge$$
$$\wedge (\forall w,w') \neg C_n(w,\sigma(|),w;u)) \rightarrow ((q>0 \rightarrow (\exists v)(FK_n(u,v) \wedge$$
$$\wedge (\exists q')C_n(v,\sigma(|),q;q))) \wedge (q=0 \rightarrow (\exists w,w')C_n(w,\sigma(s^2),w',u)))))$$

$B_n(x,y)$ hat die gewünschten Eigenschaften.

## Untere Schranken für die Entscheidungskomplexität von Theorien, deren Modelle Gruppen sind.

Bei der Konstruktion von $B_n(x,y)$ aus $M_n(x,y,z)$ haben wir nur durch $p_n$ beschränkte Quantoren verwendet. $B_n(x,y)$ behält daher seine Bedeutung in jedem Anfangsstück von $\mathbb{N}$, das $p_n$ enthält, bei. Die Beziehung $p_n \geq 2^{2^n}$ haben wir bei der Konstruktion von $B_n(x,y)$ nicht benutzt. Auf diese Weise können wir die Schwerentscheidbarkeit von Theorien mit Modellen, in denen sich beliebig grosse Anfangsstücke von $\mathbb{N}$ mit Addition und Multiplikation durch kurze Formeln darstellen lassen, nachweisen.

Sei L die Sprache der Gruppentheorie mit den nicht logischen Symbolen $\cdot$, 1, $^{-1}$, L werde aufgefasst als eine Menge von Inputs.

Sei Th eine widerspruchsfreie und unter Deduktion abgeschlossene Theorie, deren Modelle Gruppen sind.
Th habe ein Modell mit einem Element unendlicher Ordnung.

Satz 3 Zu Th existiert eine Konstante c>o, sodass es keine $2^{cn}$-beschränkte Turingmaschine T mit A(T) = Th gibt.

Beweis Satz 3 beweist man ähnlich wie Satz 1 unter Zuhilfenahme des Hierarchiesatzes von I für Turingmaschinen, die in exponentieller Zeit arbeiten, aus dem Analogon von Satz 2 für $2^n$-beschränkte Turingmaschinen.
Wir skizzieren daher nur, wie man zu einer $2^n$-beschränkten Turingmaschine T eine polynomial berechenbare Abbildung konstruiert, die jedem Input X eine Formel $F_X \in L$ ohne freie Variable zuordnet, so dass

$$|F_X| = O(|X|)$$

und

$$X \in A(T) \iff F_X \in Th$$

gilt.

Wir konstruieren zuerst rekursiv eine Formel $M_n(x,y,z,u)$, die

"es gibt ein $K<2^{2^n}$, sodass $x = y^K$ und $z = u^K$ gilt"

bedeutet.
Sei

$$M_o(x,y,z,u) = (x=1 \wedge z=1) \vee (x=y \wedge z=u).$$

Für die Konstruktion von $M_{n+1}$ aus $M_n$ nehmen wir an, wir hätten Gruppenelemente, die wir für den Moment mit x,y,z,u bezeichnen, und ein $K \in \mathbb{N}$ gegeben. Nach der Bemerkung vor Lemma 1 gilt $K<2^{2^{n+1}}$ und $x = y^K$ und $z = u^K$ genau, wenn es natürliche $K_1, K_2, K_3, K_4 < 2^{2^n}$ gibt mit $K = K_1 \cdot K_2 + K_3 + K_4$ und wenn gilt $x = (y^{K_1})^{K_2}\, y^{K_3}\, y^{K_4}$ und $z = (u^{K_1})^{K_2}\, u^{K_3}\, u^{K_4}$.
Aehnlich wie im Beweis von Lemma 1 setzen wir

$$M_{n+1}(x,y,z,u) = (\exists x_1,x_2,x_3,x_4,z_1,z_2,z_3,z_4)(((\forall w_1,w_2,w_3,w_4)$$
$$((w_1=x_1 \wedge w_2=y \wedge w_3=z_1 \wedge w_4=u) \vee (w_1=x_2 \wedge w_2=x_1 \wedge w_3=z_2 \wedge w_4=z_1) \vee$$
$$\vee (w_1=x_3 \wedge w_2=y \wedge w_3=z_3 \wedge w_4=u) \vee (w_1=x_4 \wedge w_2=y \wedge w_3=z_4 \wedge w_4=u)) \longrightarrow$$
$$\longrightarrow M_n(w_1,w_2,w_3,w_4)) \wedge x=x_2x_3x_4 \wedge z=z_2z_3z_4).$$

Sei

$$P_n(u) \equiv (\forall v)(M_n(1,1,v,u) \longrightarrow vu \neq 1).$$

$P_n(u)$ bedeutet "u hat Ordnung $> 2^{2^n}$".

Sei

$$Add_n(x,y,z,u) \equiv (M_n(1,1,x,u) \wedge M_n(1,1,y,u) \wedge M_n(1,1,z,u) \wedge P_n(u) \wedge z=xy).$$

$Add_n(x,y,z,u)$ bedeutet

"es gibt $K,K' \in \mathbb{N}$, sodass $x=u^K$, $y=u^{K'}$, $z=u^{K+K'}$, $K+K' < 2^{2^n}$ und u hat Ordnung $> 2^{2^n}$".

Sei

$$Mult_n(x,y,z,u) \equiv M_n(1,1,x,u) \wedge M_n(1,1,z,u) \wedge M_n(z,x,y,u) \wedge P_n(u)).$$

$Mult_n(x,y,z,u)$ bedeutet

"es gibt $K,K' \in \mathbb{N}$, so dass $x=u^K$, $y=u^{K'}$, $z=u^{K\cdot K'}$, $KK' < 2^{2^n}$ und u hat Ordnung $> 2^{2^n}$".

Ersetzt man im Beweis von Satz 2 bei der Konstruktion von $R_{\sigma(X)}(x)$ und $B_n(x,y)$ 0 durch die Eins der Gruppe, 1 durch u, + geeignet durch die Gruppenmultiplikation oder $Add_n$, und $M_n$ durch $Mult_n$, so erhält man eine Formel $R_{\sigma(X)}(x,u)$ mit $|R_{\sigma(X)}(x,u)| = O(|X|)$, die "$x=u^{\sigma(X)}$" bedeutet, und eine Formel $B_n(x,y,u)$, die, wenn $x=u^G$, $y=u^K$ gilt, wobei G Gödelnummer eines Inputs X sei, bedeutet:

"u hat Ordnung $> 2^{2^n}$, es ist $G,K < 2^{2^n}$ und K ist Gödelnummer einer Berechnung von T, die X akzeptiert".

$$F_X \equiv (\exists u,x)(R_{\sigma(X)}(x,u) \wedge (\exists y)B_{M|X|}(x,y,u))$$

hat für geeignet gewähltes M die gewünschten Eigenschaften.

## Literatur

Chandrasekharan, K., Analytic Number Theory. Springer Verlag, 1968.

Fischer, M.J. und Rabin, M.O., Super-Exponential Complexity of Presburger Arithmetic, MAC Technical Memorandum 43, M.I.T., 1974.

Meyer, A.R. und Stockmeyer, L.J., Inherent Computational Complexity of Decision Problems in Logic and Automata Theory, erscheint in Lecture Notes in Computer Science, Springer Verlag.