II. Polynomial beschränkte nichtdeterministische Turingmaschinen und die Vollständigkeit des aussagelogischen Erfüllungsproblems

von Alex Häussler

1. Nichtdeterministische Turingmaschinen

Ein mögliches Verfahren, um zu entscheiden, ob eine aussagelogische Formel in konjunktiver Normalform erfüllbar ist, besteht sicher darin, dass wir alle möglichen Wahrheitsbelegungen der in der Formel vorkommenden aussagelogischen Variablen systematisch aufschreiben und dazu den entsprechenden Wahrheitswert der Formel berechnen. Sie ist erfüllbar, genau wenn sie bei mindestens einer Belegung den Wahrheitswert 1 annimmt. Ein solches Verfahren kann mit einer exponentiell beschränkten T-Maschine präzisiert werden.

Analysieren wir die Schranke für die Berechnungslänge dieses Entscheidungsverfahrens, so stellen wir fest, dass der exponentielle Anteil durch die Anzahl der möglichen Belegungen entsteht, während das Berechnen des Wertes der Formel bei gegebener Belegung in polynomialer Zeit durchführbar ist. Um diese Differenzierung in der Rechenzeit mathematischen Betrachtungen zugänglich zu machen, definieren wir nichtdeterministische Turingmaschinen. Eine solche Maschine behandelt eine Formel wie folgt. Sie geht die aussagelogischen Variablen durch und gibt ihnen willkürlich den Wert 0 (falsch) oder 1 (wahr), um dann für diese nichtdeterministisch konstruierte Belegung den Wahrheitswert der ganzen Formel zu berechnen. Ist der Wahrheitswert 1, so hat die Maschine in polynomialer Zeit die Erfüllbarkeit der Formel festgestellt. Bei einer Berechnung aber, die den Wert 0 ergibt, kann die Maschine keine Entscheidung über die Erfüllbarkeit treffen.

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

S endlich (Zustandsmenge),
$s_0 \in S$ (Anfangszustand),
$s_1 \in S$ (akzeptierender Zustand),
$P \subset \sum \times (S-\{s_1\}) \times (\sum \cup \{L,R\}) \times S$ (Uebergangsrelation).

Die Begriffe Konfiguration, Folgekonfiguration, Berechnung und akzeptierte Sprache sind durch die Definitionen 2-4 in Kap. I, S. 2-3 auch

für nichtdeterministische Maschinen erklärt.

Bemerkung: Da für P auch Relationen ohne Funktionscharakter zugelassen sind, ist es möglich, dass es zu einer Konfiguration mehrere unmittelbare Folgekonfigurationen gibt. Die Arbeitsweise einer NT-Maschine unterscheidet sich also anschaulich von derjenigen einer T-Maschine dadurch, dass die NT-Maschine in gewissen Zuständen mehrere Möglichkeiten hat, auf das Eingelesene zu reagieren. Trotzdem wird von NT-Berechnung gesprochen; vielleicht wäre das Wort "NT-Ableitung" aus intuitiven Gründen besser geeignet. Während nämlich deterministische Turingmaschinen mathematische Präzisierungen mechanischer Prozeduren sind, können nichtdeterministische Turingmaschinen als mathematische Präzisierungen von mechanischen Beweisverfahren aufgefasst werden. Die Gültigkeit der Beweisschritte kann mechanisch geprüft werden, doch müssen sie zuerst erraten werden. Nur bei passender Wahl der Schritte gelangt man zu einem Ergebnis.

Trivialerweise ist jede T-Maschine auch eine NT-Maschine; somit wird jede durch eine T-Maschine akzeptierte Sprache auch durch eine NT-Maschine akzeptiert. Es gilt nun auch die Umkehrung:

Satz 1 Zu jeder NT-Maschine N gibt es eine T-Maschine T, welche dieselbe Sprache akzeptiert.

Beweisskizze: Eine Berechnung $B = \langle K_o, K_1, \ldots, K_n \rangle$ mit Input X der NT-Maschine N lässt sich durch eine endliche Folge $\langle c_1, c_2, \ldots, c_n \rangle$ mit c_i aus $(\sum \cup \{L,R\}) \times S$ charakterisieren (die Wahl unter den Folgekonfigurationen von K_{i-1} wird nämlich durch c_i vollständig festgelegt). Wir konstruieren eine deterministische Maschine T, die alle diese endlichen Folgen (beliebiger Länge) systematisch produziert und dazu jeweils die möglicherweise entsprechende Berechnung von N mit Input X probiert. Charakterisiert die vorgeschlagene Folge eine akzeptierende Berechnung von N, so soll T akzeptieren; sonst soll sie mit der Produktion der nächsten endlichen Folge weiterfahren.
Wird X von N nicht akzeptiert, so wird die Maschine T nie stoppen, und somit das X nicht akzeptieren. Wird dagegen X durch N akzeptiert, so gibt es akzeptierende Berechnungen von N und entsprechende charakterisierende endliche Folgen. Da T alle möglichen endlichen Folgen durchgeht, solange sie noch keine passende gefunden hat, wird sie eine solche Folge finden und X akzeptieren.

2. Polynomial beschränkte nichtdeterministische Turingmaschinen

Die Klasse NP

Definition 2 Sei t eine Funktion von $\mathbb{N}$ nach $\mathbb{R}$. Die NT-Maschine N ist t-beschränkt, wenn für jedes durch N akzeptierte Wort X mindestens eine seiner akzeptierenden Berechnungen eine Länge $\leq t(lh(X))$ hat.
NP ist die Menge der Sprachen, die von polynomial beschränkten NT-Maschinen akzeptiert werden.

Wie P ist auch NP vom benützten Maschinenmodell tatsächlich unabhängig. Trivialerweise ist P in NP enthalten; es wird vermutet, dass diese Inklusion echt ist (Cooksche Hypothese). Aus dem Beweis von Satz 1 kann kein Beweis von P = NP entnommen werden, weil die Schranke für die skizzierte T-Maschine exponentiell in der Schranke der gegebenen NT-Maschine ist.

Polynomiale Transformierbarkeit, vollständige Sprachen in NP:

Die Begriffe polynomiale Transformierbarkeit ($\leq_\pi$) und polynomiale Aequivalenz ($\equiv_\pi$) sind in Kap. I auf S. 6 definiert worden. Aus der trivialen Implikation $L \leq_\pi M \wedge M \in NP \Longrightarrow L \in NP$ folgt sofort, dass eine Klasse bezüglich $\equiv_\pi$ ganz in P oder ganz in NP-P oder ganz im Komplement von NP enthalten ist.

Definition 3 Eine Sprache $S \subset \{0,1\}^*$ ist vollständig in NP bezüglich $\leq_\pi$, falls
(1) $S \in NP$;
(2) für alle $L \in NP$ gilt $L \leq_\pi S$.

Wir werden zeigen, dass es vollständige Sprachen in NP gibt. Zusammen bilden sie eine Klasse, die in NP maximal ist bezüglich der von $\leq_\pi$ induzierten Ordnung. Damit ist allerdings die Cooksche Hypothese nicht bewiesen, denn auch in P gibt es eine maximale Klasse. Vollständige Sprachen sind aber gute Kandidaten für Sprachen aus NP, die vielleicht nicht in P liegen.

Das Erfüllungsproblem

Das Erfüllungsproblem besteht darin, zu entscheiden, ob eine vorgelegte aussagelogische Formel erfüllbar ist.

Es gibt bekanntlich eine "ganz einfache" Methode um zu erkennen, ob eine Formel in disjunktiver Normalform erfüllbar ist oder nicht. (Die Einfachheit lässt sich in diesem Fall als Zugehörigkeit zu P erklären). Diese Methode scheint auch für beliebige Formeln anwendbar zu sein, da jede Formel äquivalent zu einer Formel in disjunktiver Normalform ist. Leider ist die Transformation "zu kompliziert": die Länge der entsprechenden Formel in der disjunktiven Normalform hängt i.a. exponentiell von der Länge der ursprünglichen Formel ab.

Die Menge der Formeln in konjunktiver Normalform ist eine andere syntaktisch einfach erfassbare Menge von Formeln. Das Erfüllungsproblem dafür haben wir bereits am Anfang von II.1 betrachtet. Wir haben zwei Verfahren skizziert; ein Verfahren, das sich durch eine exponentiell beschränkte T-Maschine präzisieren lässt, ein zweites, von dem wir zeigen werden, dass es sich durch eine polynomial beschränkte NT-Maschine präzisieren lässt. Einerseits ist dieses Erfüllungsproblem schon recht kompliziert, andererseits doch noch in NP. Wir werden uns auf dieses Erfüllungsproblem beschränken.

Allerdings könnte man sich fragen, ob das allgemeine Erfüllungsproblem nicht noch komplizierter ist, da die Transformation einer Formel A in eine äquivalente Formel in konjunktiver Normalform ebenfalls die Länge exponentiell verändert. Tatsächlich ist es aber unnötig, eine äquivalente Formel zu betrachten; eine erfüllungsgleiche Formel A' genügt. Es gibt nun ein Verfahren, das einer Formel A eine Formel A' in konjunktiver Normalform zuordnet, und die folgenden Eigenschaften hat:

(1) Die Länge von A' ist linear in der Länge von A;
(2) alle Variablen in A kommen auch in A' vor;
(3) ist v' eine Belegung der Variablen in A' (mit Wahrheitswerten), die A' erfüllt, so erfüllt sie auch A;
(4) ist v eine Belegung der Variablen in A, die A erfüllt, so existiert eine Erweiterung v' von v - eine Belegung der Variablen in A' - die A' erfüllt.

Dieses Verfahren transformiert das allgemeine Erfüllungsproblem polynomial auf jenes der Formeln in konjunktiver Normalform. Es beruht auf dem folgenden Reduktionsschritt: Ist etwa die Formel $(B \vee (P_1 \wedge P_2))$ vorgelegt, so wird sie durch die Formel $(((B \vee P_3) \wedge (\neg P_3 \vee P_1)) \wedge (\neg P_3 \vee P_2))$ ersetzt, wobei P_3 eine neue Variable ist. Damit lassen sich Konjunktionen im Innern von Disjunktionen im Sinne der Behauptung eliminieren. Bei jedem solchen Schritt nimmt die Länge der Formel nur um eine uni-

verselle Konstante zu, und die Anzahl der Schritte ist linear in der Länge der Formel A.

Die erfüllbaren Formeln in konjunktiver Normalform (E)

Die Menge Fml der aussagelogischen Formeln wird rekursiv aufgebaut aus den aussagelogischen Variablen $P_1, P_2, P_3, \ldots$ mit den Verknüpfungszeichen $\wedge$, $\vee$, $\neg$ und den Klammern (,).
Beispiel: $((P_1 \vee \neg P_2) \wedge (P_3 \wedge P_2))$; $(A \rightarrow B)$ ist Abkürzung für $(\neg A \vee B)$.

Eine Belegung der Variablenmenge V mit Wahrheitswerten ist eine Abbildung $v: V \rightarrow \{0,1\}$, wo 0 falsch und 1 wahr bedeuten. Eine Belegung lässt sich mit den Wahrheitstafeln auf ganz Fml erweitern. Der Wert einer Formel A unter der Belegung v ist nur abhängig von der Belegung der in A vorkommenden Variablen.
Existiert eine Belegung v mit $v[A] = 1$, so ist A erfüllbar.

Die Menge G (die Grundformeln), D (die Disjunktionen) und KD (die Formeln in konjunktiver Normalform) werden durch folgende Klauseln definiert:

Eine Variable sowie die Negation einer Variablen ist eine Grundformel.
Ist A in G, so ist A in D; ist B in D und A in G, so ist $(B \vee A)$ in D.
Ist A in D, so ist A in KD; ist B in KD und A in D, so ist $(B \wedge A)$ in KD.

Definition 4 E ist die Menge der erfüllbaren Formeln in konjunktiver Normalform.
Das Erfüllungsproblem besteht darin, zu entscheiden, ob eine vorgelegte Formel in konjunktiver Normalform erfüllbar ist.

Die Sprache E^0

Die Menge E ist keine Teilmenge von $\{0,1\}^*$ - also in unserem Sinne keine Sprache. Wir definieren durch Kodierung der Formeln eine E entsprechende Sprache E^0. Die Kodierung in 0,1-Wörter ist nicht besonders sparsam, dafür aber sehr einfach. Im Uebrigen gelten die Resultate für alle einigermassen vernünftig gewählten Kodierungen.

Durch die folgenden Klauseln wird jeder Formel A in konjunktiver Normalform ein 0,1-Wort $[A]^0$ zugeordnet:

Kodierung von Formeln in G: $[P_i]^0 = 11\ldots11$ (2i mal)
$[\neg P_i]^0 = 11\ldots1$ (2i-1 mal)

Kodierung von Formeln in D: B in D und A in G, so
$[(B \vee A)]^0 = [B]^0 0 [A]^0$
Beispiel: $[((\neg P_2 \vee P_3) \vee \neg P_4)]^0 = 111011111101111111$

Kodierung von Formeln in KD: B in KD und A in D, so
$[(B \wedge A)]^0 = [B]^0 00 [A]^0$
Beispiel: $[(((\neg P_2 \vee P_3) \vee \neg P_4) \wedge P_3)]^0 = 11101111110111111100111111$

Satz 2 Die Sprache $E^0 = \{[A]^0 \mid A \in E\}$ liegt in NP.

Beweisskizze: Wir geben eine NT-Maschine an, die genau die Wörter aus E^0 akzeptiert. Gegeben ein Input $X \in \{0,1\}^*$. Ohne den Zustand s_1 zu benützen, kontrolliert die Maschine auf deterministische Weise mit einem Durchgang von links nach rechts, ob X die Kodierung einer Formel aus KD ist. (1-Folge, die höchstens durch ein oder zwei Nullen unterbrochen wird). Dann steht sie rechts von X. Falls X keine Kodierung einer Formel aus KD ist, so stoppt sie (und verwirft). Im anderen Fall geht sie wieder nach links und bewertet die einzelnen Grundformeln (genauer: kodierten Grundformel) auf nichtdeterministische Weise (zum Beispiel: Sie schreibt W oder F auf die erste 1 der kodierten Grundformel). Jedes Mal, wenn alle Grundformeln einer Disjunktion bewertet sind, geht sie nach rechts und kontrolliert erstens, ob in dieser Disjunktion mindestens eine Grundformel mit W bewertet worden ist und zweitens, ob die Bewertungen der neuen Grundformeln konsistent sind mit den schon bewerteten Grundformeln, die rechts davon stehen. Dazu muss sie natürlich wieder bis zum rechten Rand von X gehen. Falls beide Bedingungen erfüllt sind, geht sie nach links zur nächsten Disjunktion, sonst stellt sie ab. Falls sie alle Disjunktionen behandelt hat, steht sie links von X, geht in den akzeptierenden Zustand s_1 und stoppt.
Falls X in E^0 ist, so gibt es sicher eine solche Berechnung, die im Zustand s_1 endet, da ja X die Kodierung einer erfüllbaren Formel ist.

Falls umgekehrt die Maschine im Zustand s_1 nach einer Berechnung stoppt, so ist es ihr gelungen, die Grundformeln konsistent so zu bewerten, dass jede Disjunktion wahr ist und somit auch die ganze Konjunktion; also ist X in E^0.
Man überzeugt sich leicht, dass diese NT-Maschine polynomial beschränkt arbeitet.

3. Die Existenz vollständiger Sprachen in NP

Satz 3 E^0 ist eine vollständige Sprache in NP (Cook 1970).

Korollar $E^0 \in P \iff NP = P$

Beweisidee

Es wird die Vollständigkeit von E^0 in NP bezüglich $\leq_\pi$ behauptet. Da E^0 in NP ist (Satz 2), bleibt zu zeigen, dass jede Sprache L in NP polynomial auf E^0 transformierbar ist. Dazu transformieren wir L zuerst auf das unkodierte Erfüllungsproblem, indem wir jedem $X \in \{0,1\}^*$ eine Formel F(X) in konjunktiver Normalform zuordnen mit $X \in L \iff F(X) \in E$. Zudem soll die Formel F(X) aus X nach deterministischen Regeln in polynomialer Zeit hervorgehen. Die Funktion $f(X) = [F(X)]^0$ transformiert dann L polynomial auf E^0.

Die Transformationsidee: Ob eine zur Sprache L gehörige NT-Maschine M mit dem Beschränkungspolynom t den Input X akzeptiert, hängt nur von den Berechnungen der Länge $\leq N(X) = \overline{t(lh(X))}$ ab. Dank einer Modifikation der NT-Maschine M, genügt es, Berechnungen der genauen Länge N(X) zu betrachten. Die Konfigurationen einer solchen Berechnung sind in den Feldern ausserhalb -N(X) und +N(X) stets mit dem Zeichen * belegt und können deshalb dort abgeschnitten werden. Denken wir uns diese reduzierten Konfigurationen einer Berechnung mit Input X untereinander geschrieben, so entsteht eine endliche Tafel, deren Format nur von N(X) abhängt. Somit können wir uns für den Input X auf die Berechnungen innerhalb dieses endlichen Formats beschränken.
Eine Tafel lässt sich mit einer Wahrheitsbelegung gewisser aussagelogischer Variablen beschreiben. Für jedes Paar <Zeichen, Tafelfeld> führen wir eine aussagelogische Variable ein. Die Belegung einer solchen Variablen mit dem Wahrheitswert 1 soll besagen, dass das entsprechende Zeichen im entsprechenden Feld der Tafel steht. Jede Menge von Tafeln lässt sich durch die erfüllenden Belegungen einer geeigneten aussagelogischen Formel darstellen. Finden wir nun eine Formel, die genau die Tafeln der akzeptierenden Berechnungen mit Input X darstellt, so ist sie erfüllbar, genau wenn X durch eine Berechnung der Länge N(X) akzeptiert wird.
Die Formel ergibt sich, wenn wir die allgemeine Definition einer akzeptierenden Berechnung auf das endliche Format einschränken. Die vorkommenden Quantoren werden dadurch beschränkt und lassen sich durch

Konjunktionen ersetzen. Auch wird die Formel nicht zu lang, wenn wir den induktiven und lokalen Charakter der Definition ausnützen.

Vorbereitungen für den Beweis

Sei $M = \langle S, s_0, s_1, P\rangle$ im Weiteren eine NT-Maschine über $\sum$ mit:
$\sum = \{b_0, b_1, \ldots, b_p\}$ (wobei $b_0 = *$, $b_1 = 0$, $b_2 = 1$),
$S = \{s_0, s_1, \ldots, s_q\}$,
$P \subset \sum \times (S - \{s_1\}) \times (\sum \cup \{L, R\}) \times S$.

Die Modifikation

Setzen wir $P' = P \cup \{\langle b_j, s_k, b_j, s_k\rangle \mid 0 \leq j \leq p, 0 \leq k \leq q, U(b_j, s_k) = \emptyset\}$ mit $U(b_j, s_k) = \{\langle u, w\rangle \mid \langle b_j, s_k, u, w\rangle \in P\}$, so wird durch diese Modifikation eine NT-Maschine $M' = \langle S, s_0, s_1, P'\rangle$ definiert. Die zusätzlichen Uebergangsmöglichkeiten von M' bewirken, dass sich vollständige Berechnungen von M durch das Kopieren der Endkonfiguration fortsetzen lassen. Eine Berechnung von M' ist also entweder eine Berechnung von M allein oder aber eine durch Kopie der Endkonfiguration verlängerte vollständige Berechnung von M.
M' hat keine vollständigen Berechnungen; deshalb soll eine Berechnung B von M' schlechthin akzeptierend für den Input X heissen, falls sie die Verlängerung einer akzeptierenden Berechnung von M mit Input X ist. B ist genau dann akzeptierend, wenn die letzte Konfiguration im Zustand s_1 ist. (Hinweis: s_1 ist ein Endzustand von M.)

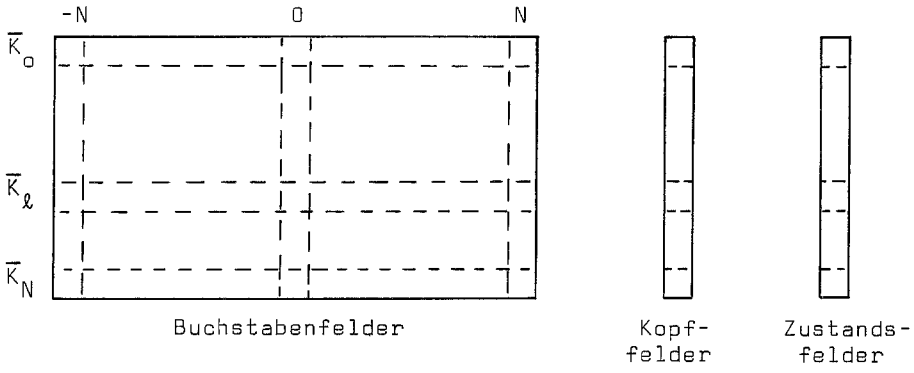
Die Tafel einer Berechnung

Sei N im Weiteren eine natürliche Zahl.

Definition Ist $B = \langle K_0, K_1, \ldots, K_N\rangle$ eine Berechnung der NT-Maschine M', so ist $\overline{B} = \langle \overline{K}_0, \overline{K}_1, \ldots, \overline{K}_N\rangle$ die Tafel von B; dabei ist
$\overline{K} = \langle (a_m)_{-N \leq m \leq N}, i, s\rangle$ die (N-) Reduktion der Konfiguration
$K = \langle (a_m)_{m \in \mathbb{Z}}, i, s\rangle$.

Bemerkung: Ist B eine Berechnung mit Input X und Länge N und ist $lh(X) < N$, so vernachlässigt diese Reduktion nur Bandfelder mit der Aufschrift b_0.

Schreiben wir die reduzierten Konfigurationen von $B = \langle \overline{K}_0, \overline{K}_1, \ldots, \overline{K}_N\rangle$ untereinander, so entsteht eine Tafel, deren Format nur von N abhängt.



Ist $\overline{K}_\ell = \langle (a_m)_{-N\leq m\leq N}, i, s_k \rangle$, so steht im Bu-Feld (ℓ,m) der Buchstabe a_m (die Bandaufschrift), im Ko-Feld ℓ die Zahl i mit $-\ell \leq i \leq \ell$ (d.h. der Kopf befindet sich über dem i-ten Feld), im Zu-Feld ℓ das Zeichen s_k (der Zustand der Maschine).

Die Darstellung von Tafeln vom Format N mit Hilfe von Belegungen gewisser aussagelogischer Variablen

Wir führen die folgenden aussagelogischen Variablen ein:
Für jedes ℓ,m und b_j die Variable $B_j(\ell,m)$;
für jedes ℓ,m die Variable $C(\ell,m)$;
für jedes ℓ und s_k die Variable $S_k(\ell)$;
(dabei ist $0\leq\ell\leq N, -N\leq m\leq N$, $b_j \in \sum$, $s_k \in S$).

Um eine Bijektion zwischen den Belegungen dieser Variablen und Tafeln zu erhalten, lassen wir auch Tafeln zu, in deren Feldern Teilmengen der entsprechenden Zeichen stehen - also in einem Bu-Feld eine Teilmenge von $\sum$, in einem Ko-Feld eine Teilmenge von $\{i \mid -N\leq i\leq N\}$, in einem Zu-Feld eine Teilmenge von S.

Die Belegung v der obigen Variablen, welche die Tafel T vom Format N beschreibt, ist gegeben durch:

$v[B_j(\ell,m)] = 1 \iff b_j$ steht im Bu-Feld (ℓ,m) von T
$v[C(\ell,m)] = 1 \iff m$ steht im Ko-Feld ℓ von T
$v[S_k(\ell)] = 1 \iff s_k$ steht im Zu-Feld ℓ von T

Notation: v_T ist die Belegung, die T beschreibt; T_v die Tafel, die zur Belegung v gehört.

Definition Eine Menge H von Tafeln wird durch eine aussagelogische Formel A dargestellt, wenn für jede Tafel T gilt:

$$T \in H \iff v_T[A] = 1$$

Bemerkung: (1) Eine einzige Tafel T wird durch die Formel A_T dargestellt, welche die Konjunktion aller Grundformeln ist, die unter der Belegung v_T den Wert 1 haben.
(2) Eine Menge H von Tafeln wird zum Beispiel durch die Formel $\bigvee_{T \in H} A_T$ dargestellt. Die Länge dieser Formel ist aber sicher grösser als $|H|$.

Die akzeptierenden Tafeln, die darstellende Formel

Seien M' die Modifikation der NT-Maschine M, X ein b_1,b_2-Wort und N eine natürliche Zahl mit $lh(X)<N$.

Definition $AT_N(X) = \{\overline{B} \mid$ B ist akzeptierende Berechnung von M' mit Input X und Länge N$\}$

Wir wollen $AT_N(X)$ darstellen. Verfahren wir gemäss (2) oben, so erhalten wir eine i.a. zu lange Formel. Benützen wir hingegen den induktiven und lokalen Charakter der Berechnungen, so lässt sich eine kürzere Formel angeben: $F_N(X)$.

$F_N(X)$ ist die Konjunktion einer Liste von Formeln in konjunktiver Normalform. Diese Liste ergibt sich, indem wir auf Grund gemeinsamer Eigenschaften der Tafeln $\overline{B}$ in $AT_N(X)$ Formeln aufschreiben, die unter jeder Belegung $v_{\overline{B}}$ den Wert 1 haben. Für die Konjunktion $F_N(X)$ gilt dann sicher $v_{\overline{B}}[F_N(X)] = 1$ für $\overline{B} \in AT_N(X)$. Es bleibt dann noch die Umkehrung zu beweisen - was gelingt, wenn die Liste genügend vollständig ist.

Bemerkung: Wir schreiben die Formeln nicht immer in konjunktiver Normalform auf, geben aber jeweils einen Hinweis, wie sie äquivalent umgeformt werden können. Zusätzlich machen wir Angaben über Grösse und Anzahl.

I In jedem Feld einer Tafel $\overline{B} \in AT_N(X)$ steht genau ein Zeichen:

für alle (ℓ,m): $\bigvee_{j=o}^{p} B_j(\ell,m)$; für alle (ℓ,m), $i \neq j$: $\neg B_i(\ell,m) \vee \neg B_j(\ell,m)$;

für alle ℓ: $\bigvee_{m=-N}^{N} C(\ell,m)$ (*); für alle $\ell, m \neq n$: $\neg C(\ell,m) \vee \neg C(\ell,n)$;

für alle ℓ: $\bigvee_{k=0}^{q} S_k(\ell)$; für alle $\ell,k\neq h$: $\neg S_k(\ell) \vee \neg S_h(\ell)$;

(dabei ist $0\leq\ell\leq N$; $-N\leq m,n\leq N$; $0\leq i,j\leq p$; $0\leq k,h\leq q$).

Anzahl: Grössenordnung N^3.

Länge : Nur (*) hat die Grössenordnung N, alle anderen Längen sind unabhängig von N.

II In der Zeile 0 einer Tafel $\overline{B}\in AT_N(X)$ steht der Input X.

$X = b_{i_1} b_{i_2} \ldots b_{i_n}$; da wir $lh(X)<N$ voraussetzen, ist $n<N$:

$B_o(0,m)$ für m mit $-N\leq m\leq 0$ oder $n<m\leq N$;

$B_{i_k}(0,k)$ (d.h. b_{i_k} im Feld $(0,k)$) für k mit $0<k\leq n$;

$C(0,0)$ (d.h. Kopf auf Feld 0);

$S_o(0)$ (d.h. Zustand s_o).

III Der Zeilenübergang von der Zeile ℓ $(0\leq\ell<N)$ zur Zeile $\ell+1$ einer Tafel $\overline{B}\in AT_N(X)$ wird durch die Uebergangsrelation P' bestimmt; aus lokalen Eigenschaften der Zeile ℓ folgen gemäss P' lokale Eigenschaften der Zeile $\ell+1$.

(1) Steht der Buchstabe b_j im Buchstabenfeld (ℓ,m) und befindet sich der Kopf nicht über diesem Feld, so wird der Buchstabe b_j in die Zeile $\ell+1$ kopiert: Für ℓ,m und j mit $0\leq\ell<N$, $-N\leq m\leq N$, $0\leq j\leq p$ nehmen wir die Formel $(B_j(\ell,m)\wedge\neg C(\ell,m)) \longrightarrow B_j(\ell+1,m)$ in die Liste auf.

Bemerkung: $(A\wedge\neg B) \longrightarrow C$ äq $((\neg A\vee B)\vee C)$; Anzahl: Grössenordnung N^2.

(2) Steht der Buchstabe b_j im Buchstabenfeld (ℓ,m), befindet sich über diesem Feld der Kopf und ist die Maschine im Zustand s_k, so ist der Uebergang von einem 4-Tupel $\langle b_j,s_k,u,w\rangle\in P'$, welches dem Berechnungsschritt zugrunde liegt, abhängig. Um Komplikationen am Tafelrand zu vermeiden, soll für jedes ℓ $(0\leq\ell<N)$ das m nur zwischen $-\ell$ und $+\ell$ variieren.

(a) Die folgende Formel $K(\ell,m,b_j,s_k,u,w)$ beschreibt die drei beteiligten Felder der Zeile $\ell+1$, falls der Uebergang von K_ℓ nach $K_{\ell+1}$ von B auf Grund von $\langle b_j,s_k,u.w\rangle\in \sum\times S\times(\sum\cup\{L,R\})\times S$ erfolgt ist:

$$K(\ell,m,b_j,s_k,u.w) \equiv \begin{cases} B_i(\ell+1,m) \wedge C(\ell+1,m) \wedge S_h(\ell+1), \text{ falls } u=b_i,\ w=s_h \\ \text{(d.h. } b_i \text{ wird überschrieben, Kopf bleibt, neuer} \\ \text{Zustand } s_h) \\ B_j(\ell+1,m) \wedge C(\ell+1,m+1) \wedge S_h(\ell+1), \text{ falls } u=R,\ w=s_h \\ \text{(d.h. } b_j \text{ wird kopiert, Kopf nach rechts, neuer} \\ \text{Zustand } s_h) \\ B_j(\ell+1,m) \wedge C(\ell+1,m-1) \wedge S_h(\ell+1), \text{ falls } u=L,\ w=s_h \\ \text{(d.h. } b_j \text{ wird kopiert, Kopf nach links, neuer} \\ \text{Zustand } s_h) \end{cases}$$

(b) Das Programm einer NT-Maschine lässt aber für den Uebergang mehrere Möglichkeiten zu; deshalb betrachten wir die - dank der Modifikation nichtleere Disjunktion dieser Möglichkeiten:

$$K_{p'}(\ell,m,b_j,s_k) \equiv \bigvee_{\substack{\text{alle } u,w \text{ mit} \\ <b_j,s_k,u,w> \in P'}} K(\ell,m,b_j,s_k,u,w)$$

(c) Da diese Formel alle Möglichkeiten einschliesst, ist sie unter einer Belegung $v_{\overline{B}}$ mit $\overline{B} \in AT_N(X)$ erfüllt, falls die Maschine in der Berechnung B im Feld (ℓ,m) den Buchstaben b_j einliest und sich im Zustand s_k befindet. Diese Voraussetzung wird durch die folgende Formel beschrieben: $H(\ell,m,b_j,s_k) \equiv B_j(\ell,m) \wedge C(\ell,m) \wedge S_k(\ell)$.
Für ℓ,m,j,k mit $0 \leq \ell < N$, $-\ell \leq m \leq \ell$, $0 \leq j \leq p$, $0 \leq k \leq q$ nehmen wir die Formel $H(\ell,m,b_j,s_k) \longrightarrow K_{p'}(\ell,m,b_j,s_k)$ in die Liste auf.

Bemerkung: Die Umformung in konjunktive Normalform ist unabhängig von N. Anzahl: Grössenordnung N^2.

IV Eine Tafel $\overline{B} \in AT_N(X)$ gehört zu einer akzeptierenden Berechnung von M'. Somit ist die letzte Konfiguration im Zustand s_1. Wir nehmen deshalb die Formel $S_1(N)$ in die Liste auf.

Definition $F_N(X)$ ist die Konjunktion der Formeln unter I-IV.$(lh(X)<N)$.

Sie ist in konjunktiver Normalform und ihre Länge ist polynomial in N. Auch geht die Formel $F_N(X)$ - bei gegebener NT-Maschine M - auf deterministische Weise in polynomialer Zeit aus X und N hervor.

Lemma Ist lh(X)<N, so stellt die Formel $F_N(X)$ die Menge $AT_N(X)$ dar.

Beweis: (1) Sei $T \in AT_N(X)$. Also ist $T = \overline{B}$ für eine akzeptierende Berechnung B von M' mit Input X. Die Formeln der obigen Liste haben wir so gewählt, dass sie mit einer Belegung $v_{\overline{B}}$ den Wahrheitswert 1 haben, also ist $v_T[F_N(X)] = 1$.

(2) Sei $v[F_N(X)] = 1$ für eine Belegung v. Behauptung: T_v ist die Tafel einer akzeptierenden Berechnung von M' mit Input X und Länge N.

Mit $v[F_N(X)] = 1$ haben alle Formeln der Liste auch den Wahrheitswert 1. Dank den Formeln in I steht in jedem Feld der Tafel T_v genau ein Zeichen. Durch Ergänzung der Zeilen mit b_o, entsteht aus T_v eine Folge von Konfigurationen: $\langle K_o, K_1, \ldots, K_N \rangle$. Mit Induktion nach ℓ ($0 \leq \ell < N$) zeigen wir, dass diese Folge eine Berechnung von M' mit Input X ist:

$\ell = 0$: Dank II und lh(X)<N ist K_o die Inputkonfiguration für X.
$\ell \Longrightarrow \ell+1$, $\ell < N$: Sei $\langle K_o, K_1, \ldots, K_\ell \rangle$ eine Berechnung von M' mit Input X. Sei $K_\ell = \langle (a_i)_{i \in \mathbb{Z}}, m, s_k \rangle$ und $K_{\ell+1} = \langle (a'_i)_{i \in \mathbb{Z}}, m', s' \rangle$. Wir zeigen: Der Uebergang von K_ℓ nach $K_{\ell+1}$ verläuft gemäss Programm P'.

- Kopie der Buchstaben in den Feldern n mit $n \neq m$:
Für $|n| > N$: $a'_n = b_o = a_n$
Für n mit $-N \leq n \leq N$: Sei $a_n = b_r$, also ist $v[B_r(\ell,n)] = 1$. Da $n \neq m$ ist, gilt $v[\neg C(\ell,n)] = 1$. Es gilt $v[(B_r(\ell,n) \wedge \neg C(\ell,n)) \rightarrow B_r(\ell+1,n)] = 1$ dank der Voraussetzung; somit ist $v[B_r(\ell+1,n)] = 1$ und $a'_n = b_r = a_n$.
- Uebergang im Feld m, Aenderung der Kopflage und des Zustands:
Wir betrachten das Feld m aus K_ℓ. Da $\langle K_o, K_1, \ldots, K_\ell \rangle$ eine Berechnung ist, gilt für m $-\ell \leq m \leq \ell$. Sei $a_m = b_j$, dann gilt: $v[B_j(\ell,m)] = 1$, $v[C(\ell,m)] = 1$ und $v[S_k(\ell)] = 1$, also ist $v[H(\ell,m,b_j,s_k)] = 1$. Mit der entsprechenden Formel der Liste haben wir auch $v[K_{p'}(\ell,m,b_j,s_k)] = 1$. $K_{p'}(\ell,m,b_j,s_k)$ ist eine Disjunktion - somit existieren u,w mit $\langle b_j, s_k, u, w \rangle \in P'$ und $v[K(\ell,m,b_j,s_k,u,w)] = 1$. Für $K_{\ell+1}$ gilt nun je nach dem, wie $\langle u,w \rangle$ aussieht, das folgende:
Falls $u = b_i$, $w = s_h$, so $a'_m = b_i$, $m' = m$, $s' = s_h$.
Falls $u = R$, $w = s_h$, so $a'_m = b_j = a_m$, $m' = m+1$, $s' = s_h$.
Falls $u = L$, $w = s_h$, so $a'_m = b_j = a_m$, $m' = m-1$, $s' = s_h$.
In allen Fällen ist der Uebergang gemäss P' und somit auch $\langle K_o, K_1, \ldots, K_\ell, K_{\ell+1} \rangle$ eine Berechnung von M'.

Der Zustand von K_N ist s_1, da $v[S_1(N)] = 1$. $\langle K_0, K_1, \ldots, K_N \rangle$ ist also eine akzeptierende Berechnung von M' mit Input X; also ist $T_v \in AT_N(X)$.

q.e.d.

Beweis von Satz 3

(1) $E^0 \in NP$ (Satz 2)

(2) Seien $L \in NP$, M eine L akzeptierende NT-Maschine über Σ, M' deren Modifikation und t ein Beschränkungspolynom mit $n < t(n)$ (alle n).

Seien $X \in \{b_1, b_2\}^*$, $N(X) = \overline{t(lh(X))}$:

$X \in L$

$\Longleftrightarrow$

M akzeptiert X

(M ist t-beschränkt) $\Longleftrightarrow$

Es existiert eine akzeptierende Berechnung von M mit Input X und Länge $\leq N(X)$.

(Modifikation) $\Longleftrightarrow$

Es existiert eine akzeptierende Berechnung von M' mit Input X und Länge $= N(X)$.

(Definition) $\Longleftrightarrow$

$AT_{N(X)}(X) \neq \emptyset$

(Lemma, $lh(X) < N(X)$) $\Longleftrightarrow$

$F_{N(X)}(X)$ ist erfüllbar

(Kodierung) $\Longleftrightarrow$

$f(X) = [F_{N(X)}(X)]^0 \in E^0$

L wird also mittels f auf E^0 transformiert.

Die Länge von $F_{N(X)}(X)$ ist polynomial in $N(X)$ und somit auch polynomial in $lh(X)$.
Sind M und t vorgelegt, so lässt sich eine T-Maschine angeben, die für den Input X die Zahl $N(X)$ berechnet und die Kodierung von $F_{N(X)}(X)$ aufschreibt; f ist in Π.

q.e.d.

4. Polynomial beschränkte $\exists$ -Quantifikation

Das Berechnen des Wahrheitswertes einer aussagelogischen Formel bei gegebener Belegung ist ein polynomialer Prozess. Dabei kann man sich den Input zweistellig denken: einerseits die Formel, andererseits die Belegung. Eine aussagelogische Formel ist genau dann erfüllbar, wenn es eine erfüllende Belegung ihrer Variablen gibt. Die Sprache E^0 der kodierten erfüllbaren Formeln in konjunktiver Normalform ist also die $\exists$ -Quantifikation einer zweistelligen polynomial entscheidbaren Relation; zudem ist diese $\exists$ -Quantifikation polynomial beschränkt (im Sinne der nachfolgenden Ausführungen).

Man überlegt sich leicht (Satz 4), dass eine polynomial beschränkte $\exists$ -Quantifikation einer zweistelligen polynomial entscheidbaren Relation eine Sprache aus NP liefert. Umgekehrt gibt es aber auch zu jeder Sprache aus NP eine zweistellige polynomial entscheidbare Relation, aus welcher sich die gegebene Sprache durch polynomial beschränkte $\exists$ -Quantifikation gewinnen lässt (Satz 5).

Dadurch erreicht man eine weitere Charakterisierung von NP.

Definition 5 Sei T eine T-Maschine über Σ. T akzeptiert genau dann das Wortpaar $\langle X,Y\rangle \in \{0,1\}^* \times \{0,1\}^*$, wenn es eine vollständige Berechnung mit Input X*Y und Endzustand s_1 gibt. Die von T akzeptierte zweistellige Relation wird mit $A^2(T)$ bezeichnet.

Der Begriff polynomial entscheidbar lässt sich so ohne weiteres auf Relationen übertragen. P^2 bezeichnet die Menge der zweistelligen polynomial entscheidbaren Relationen.

Bemerkung Es sei $Y(\emptyset) = \emptyset$, $Y(0) = 10$, $Y(1) = 11$ und $Y(VW) = Y(V)Y(W)$. Durch die Kodierung $Z: \{0,1\}^* \times \{0,1\}^* \longrightarrow \{0,1\}^*$ mit $Z(V,W) = Y(V)00Y(W)$ lassen sich die zweistelligen Relationen aus P^2 als Sprachen in P auffassen: Für $\overline{L} \subset \{0,1\}^* \times \{0,1\}^*$ gilt

$$\overline{L} \in P^2 \iff Z(\overline{L}) \in P$$

Satz 4 Sei p ein Polynom und $\overline{L} \in P^2$. Dann liegt
$L = \{X \mid (\exists Y)(lh(Y) \leq p(lh(X)) \wedge \langle X,Y\rangle \in \overline{L})\}$ in NP.

Beweisskizze: Sei T eine polynomial beschränkte (einbändige) T-Maschine mit $A^2(T) = \overline{L}$. Es ist leicht, eine polynomial beschränkte NT-Maschine

anzugeben, welche rechts vom Input X, abgetrennt durch *, eine beliebige 0,1-Folge der Länge $\leq p(lh(X))$ aufschriebt. Die Zusammensetzung mit T ergibt eine NT-Maschine, die polynomial beschränkt ist und genau L akzeptiert.

Satz 5 Zu jeder Sprache $L \in NP$ gibt es eine Relation $\overline{L} \in P^2$ und ein Polynom p derart, dass $L = \{X \mid (\exists Y)(lh(Y) \leq p(lh(X)) \wedge \langle X,Y\rangle \in \overline{L})\}$.

Beweisskizze: Sei N eine polynomial beschränkte NT-Maschine, die L akzeptiert und q ein Beschränkungspolynom. Wie im Beweis von Satz 1 lässt sich jede Berechnung B von N durch eine Folge $Y \in \{0,1\}^*$ der Länge $C \cdot lh(B)$ (für eine geeignete Konstante C) charakterisieren. Die zweibändige T-Maschine T (angewandt auf die beiden Argumente X und Y) soll die durch Y charakterisierte Berechnung von N mit Input X simulieren und $\langle X,Y\rangle$ genau dann akzeptieren, wenn die charakterisierte Berechnung von N eine akzeptierende Berechnung ist. Für die Simulierung jedes einzelnen Schrittes von N braucht T eine feste Anzahl Schritte. T ist also polynomial beschränkt und es ist
$L = \{X \mid (\exists Y)(lh(Y) \leq C \cdot q(lh(X)) \wedge \langle X,Y\rangle \in \overline{L}\}$.

Definition 6 Eine Sprache L ist die polynomial beschränkte $\exists$-Quantifikation einer Sprache L', wenn es ein Polynom p derart gibt, dass $L = \{X \mid (\exists Y)(lh(Y) \leq p(lh(X)) \wedge Z(X,Y) \in L'\}$, wobei Z die in der Bemerkung (zur Definition 5) definierte Zuordnung ist.

Mit Satz 4, Satz 5 und $P \subset NP$ ergibt sich das folgende Korollar:

Korollar Es gilt genau $P = NP$, wenn P gegenüber polynomial beschränkter $\exists$-Quantifikation abgeschlossen ist.

Literatur

Cook, S.A., The complexity of Theorem-Proving Procedures, Conf. Rec. of Third ACM Symp. on Th. of Computing (1970) 151-158.