Computing 5, 57—70 (1970)

Über einen Automaten mit Pufferspeicherung[1]

Von

R. Vollmar, Erlangen

Mit 2 Abbildungen

(Eingegangen am 24. März 1969)

Zusammenfassung — Summary

Über einen Automaten mit Pufferspeicherung. Es wird ein Automat mit Speicherband eingeführt und in seiner Eigenschaft als erkennender Automat untersucht. Er kann sein Arbeitsband nur als Puffer benutzen, d. h. was zuerst eingelesen wird, wird auch zuerst wieder ausgelesen. Es werden verschiedene Modifikationen des Grundmodells, insbesondere aber die deterministische, in Realzeit arbeitende Version betrachtet, sowohl hinsichtlich der Lage der durch den Automatentyp definierten Sprachklasse in der CHOMSKY-Hierarchie, als auch das Verhalten dieser Sprachen gegenüber gewissen Operationen. Einige Resultate über Entscheidbarkeitsfragen werden zitiert.

On an Automaton with Buffer-tape. A class of automata with tape has been defined and considered as recognition devices. Our automaton can use its working tape only as a buffer, i. e. in a "first-in-first-out" manner. Some modifications of the general model are considered, especially the deterministic, real-time version. The position in the CHOMSKY hierarchy of the class of languages accepted by such automata and some operations with these languages have been investigated. Some results on decidability questions are cited.

Einleitung

Die Untersuchung abstrakter Automaten mit Speicherbändern wurde angeregt von der Theorie der rekursiven Funktionen und der Mathematischen Linguistik.

Kellerautomaten, linear beschränkte Automaten und TURING-Maschinen sind typische Automaten dieser Art. Im Unterschied zu den endlichen, erkennenden Automaten verfügen die genannten Typen über einen wachsenden, potentiell unendlichen Speicher, von dessen Inhalt sie ihre Vorgehensweise abhängig machen können. Ihre unterschiedlichen Fähigkeiten beim Annehmen von Klassen formaler Sprachen resultieren aus den verschiedenen Einschränkungen, denen sie beim Auslesen der Speicherinhalte unterworfen sind.

Nachdem man längere Zeit in der Hauptsache solche Automaten betrachtete, die in enger, wechselseitiger Beziehung zu den Sprachklassen

[1] Diese Arbeit wurde von der Deutschen Forschungsgemeinschaft (Az.: Ha 417/7) gefördert.

der CHOMSKY-Hierarchie stehen, sind in den letzten Jahren verschiedene Typen von Automaten untersucht worden, die sich nicht mehr in natürlicher Weise in diese Hierarchie einordnen lassen. Dies gilt auch für den hier eingeführten Automaten, der Pufferautomat genannt werden soll, und der aus zweierlei Gründen interessant scheint:

1. In naheliegender Analogie zum Kellerautomaten, bei dem der Zugriff auf das Speicherband nach dem Prinzip „was zuletzt eingelesen wurde, wird zuerst ausgelesen" erfolgt, wird ein Automat definiert, bei dem nach dem Grundsatz „was zuerst eingelesen wurde, wird zuerst ausgelesen" verfahren wird.

2. Dieser Automat wird es ermöglichen, in „einfacher" Weise — wir wollen darunter eine Realzeit-Arbeitsweise verstehen — Ketten aus einer freien Halbgruppe über einem endlichen Alphabet, die aus konkatenierten homomorphen Bildern einer Kette bestehen, anzunehmen. Im Deutschen oder Englischen entspräche dies z. B. Sätzen, die mit „respektive" bzw. „respectively" gebildet werden: „Wenn eine Funktion rekursiv, A-rekursiv, partiell-rekursiv oder A-partiell-rekursiv ist, dann ist sie resp. berechenbar, A-berechenbar, partiell-berechenbar oder A-partiell-berechenbar" (übersetzt aus DAVIS [1]). Die Struktur solcher, im Prinzip beliebig langer Sätze läßt sich nicht durch kontextfreie Grammatiken erfassen. Ketten dieser Art, die in problemorientierten Sprachen, wie ALGOL, häufiger auftreten als in natürlichen Sprachen, sind deshalb auch nicht durch Kellerautomaten zu erkennen.

Ein Teil der Sätze (vor allem in den beiden ersten Abschnitten) wird ohne Beweis angegeben werden. Für Einzelheiten dazu muß auf VOLLMAR [2] verwiesen werden.

1. Der nichtdeterministische, unvollständige Pufferautomat

In den folgenden Definitionen werden wir vom Begriff der Korrespondenz Gebrauch machen, um nicht Abbildungen in Potenzmengen vornehmen zu müssen. Unter einer Korrespondenz δ wird dabei ein Tripel (G, A, B) von Mengen G, A, B verstanden, für die gilt: $G \subset A \times B$. Eine Korrespondenz heißt endlich, wenn G eine endliche Menge ist.

Als Grundmodell wird der nichtdeterministische, unvollständige Pufferautomat eingeführt:

Definition. *Ein nichtdeterministischer, unvollständiger Pufferautomat (PA)* $\mathfrak{B}'$ *ist ein Septupel* $(X, Z, P, B_1, B_2, \delta', z_0)$. *Dabei sind* X, Z, P, B_1 *und* B_2 *endliche, nichtleere Mengen,* $z_0 \in Z$ *und* δ' *ist eine endliche Korrespondenz von* $Z \times X \times P^*$ *in* $Z \times B_1 \times P^* \times B_2$.[2]

Folgende Sprechweise soll künftig verwendet werden:

X ist das Eingabealphabet, Z die Zustandsmenge, P das Pufferalphabet,

[2] Die freie Halbgruppe über einem endlichen Alphabet A mit der Konkatenation als Verknüpfung wird mit A^* bezeichnet. Elemente einer freien Halbgruppe werden Wörter oder Ketten genannt. Das neutrale Element der freien Halbgruppe wird mit ε bezeichnet. Die Menge aller endlich langen Wörter über einem endlichen Alphabet A werde ebenfalls mit A^* bezeichnet.

B_1 die Menge, die die Eingabebandbewegung beschreibt, B_2 die Menge, die die Bewegung des Pufferlesekopfes angibt und z_0 der Anfangszustand des Automaten. Der einfacheren Beschreibung wegen ist es wohl nützlich, von einer bildlichen Darstellung eines PA auszugehen (siehe Abb. 1).

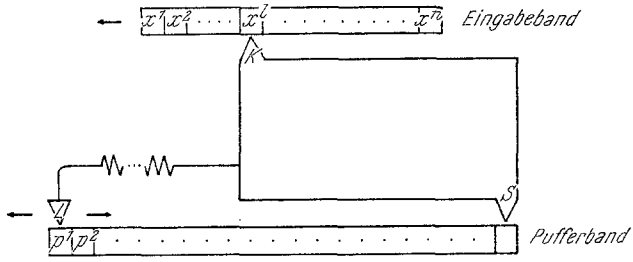

Abb. 1. Pufferautomat

Der Kern des PA ist ein endlicher Automat, der über einen Bandlesekopf K von einem Eingabeband Symbole entgegennimmt. Das Eingabeband ist in einzelne Felder unterteilt, und jedes dieser Felder kann genau ein Symbol aus X aufnehmen. Zum Automaten gehört weiterhin ein Arbeitsband, das als Pufferband oder einfach als Puffer bezeichnet wird und das ebenfalls in Felder unterteilt ist, die jeweils ein Symbol aus P enthalten können. Das Pufferband wird über den Pufferschreibkopf S beschrieben. Es ist in seiner Länge nicht begrenzt. Der Pufferlesekopf L kann jeweils eine beliebige, aber endliche Anzahl von Symbolen am Beginn des Pufferbandes betrachten. Da in diesem Abschnitt die Beweise nicht ausgeführt werden, soll auch die Arbeitsweise lediglich verbal beschrieben werden: Die Beziehung

$$\delta' [z_i, x_j, \pi_k] \supset \{(z_l, b_{1q}, \pi_r, b_{2s})\}$$

mit $z_i, z_l \in Z$, $x_j \in X$, π_k, $\pi_r \in P^*$, $b_{1q} \in B_1$, $b_{2s} \in B_2$ ist folgendermaßen zu interpretieren: Befindet sich der PA im Zustand z_i, steht das Symbol x_j auf dem Feld des Eingabebandes, auf das der Bandlesekopf zeigt, und enthalten die ersten Felder des Pufferbandes die Kette π_k, so geht der Automat in den Zustand z_l über, das Eingabeband wird um q ($q \in \{0,1\}$) Felder nach links bewegt und der Pufferlesekopf um s Felder des Pufferbandes (nach rechts) in Richtung zum Pufferschreibkopf, falls die Länge der im Puffer stehenden Kette $> s$ ist, und π_r wird in den Puffer geschrieben; waren vor dem Einschreiben von π_r dagegen nur s Felder oder weniger beschrieben, so erfolgt ein vollständiges Löschen des Puffers, d. h. der Pufferschreib- und der Pufferlesekopf stehen auf demselben Feld. Alle vom Pufferlesekopf überlesenen Symbole sind nicht mehr erreichbar.

Aus den sich bietenden Möglichkeiten, Automaten zu betrachten und zu vergleichen, sei die des Erkennens von bestimmten Mengen herausgegriffen: Eine Kette $w \in \overline{X}^*$ [3] heiße vom PA $\mathfrak{B}'$ angenommen, wenn

[3] Sei A^* eine freie Halbgruppe. Dann wird definiert: $\overline{A}^* := A^* \setminus \{\varepsilon\}$. Das Annehmen von Ketten durch PA wird hier jeweils so definiert, daß die leere Kette ε nicht zur Menge der angenommenen Ketten gehört.

eine endliche Folge von Zustandsübergängen existiert, die den Automaten aus dem Anfangszustand z_0 und der Kette w auf dem Eingabeband und leerem Puffer in eine Konfiguration überführt, in der bei abgearbeitetem Eingabeband der Puffer leer ist. Bezüglich des dabei erreichten Zustandes werden keine Einschränkungen gemacht, d. h. es wird keine Untermenge von Z ausgezeichnet, wie dies teilweise bei anderen Automatentypen der Fall ist (siehe z. B. GINSBURG [3]).

Für alle Automaten, für die das Annehmen von Ketten und damit auch von Mengen von Ketten definiert ist, wird der Begriff des Äquivalentseins erklärt: Zwei Automaten sind genau dann zueinander äquivalent, wenn sie die gleichen Mengen annehmen.

Als erstes Ergebnis erhalten wir, daß sogar die deterministische Variante des obigen PA, die dadurch charakterisiert ist, daß die Korrespondenz für jedes Argument höchstens einen Wert besitzt, im wesentlichen über die gleichen Fähigkeiten verfügt wie TURING-Maschinen (siehe auch SHEPHERDSON und STURGIS [4]):

Satz. *Zu jeder Turingmaschine $\mathfrak{T}$ über $X \setminus \{x_0\}$ — x_0 sei dabei das uneigentliche Symbol — gibt es einen deterministischen, unvollständigen Pufferautomaten $\mathfrak{B}$, so daß gilt: Wird $\mathfrak{T}$ auf das erste Symbol einer Kette $w \in (\overline{X \setminus \{x_0\}})^*$, die auf dem sonst leeren Band steht, angesetzt und gelangt $\mathfrak{T}$ durch eine Übergangsfolge endlicher Länge in eine Endkonfiguration, so wird $w\,\#$ mit $\# \notin X$ vom Pufferautomaten $\mathfrak{B}$ angenommen und umgekehrt.*

Der Beweis soll nur skizziert werden: Der zur TURING-Maschine $\mathfrak{T}$ konstruierte PA $\mathfrak{B}$ übernimmt das Teilwort w seiner Eingabekette — mit einem speziellen Symbol am Anfang versehen — in den Puffer. Der Pufferinhalt wird dann bei Feststehen des Eingabekopfes auf # in Abhängigkeit von den Änderungen auf dem Band der Turingmaschine und der Bewegung ihres Lese-Schreibkopfes modifiziert, wobei für $\mathfrak{B}$ keine Information verloren geht, da die vom Pufferlesekopf überlesenen Symbole wieder in den Puffer eingeschrieben werden. Der Prozeß läßt sich durch das Bild eines am Anfang und Ende verbundenen Bandes, das beliebig lang werden kann, veranschaulichen. Das anfangs eingefügte Symbol kennzeichnet jeweils den Beginn des Bandes der TURING-Maschine. Während das Überschreiben eines Symbols auf dem Band der TURING-Maschine und das Rechtsgehen in $\mathfrak{B}$ einfach simuliert werden können, ist das Nachbilden des Linksgehens nur durch das vorübergehende Löschen der ausgelesenen Symbole und durch insgesamt zweimaliges Rundschieben des gesamten Puffers zu erreichen. Stoppt die TURING-Maschine — nach der Definition von HERMES [5] durch besondere Stopanweisungen in der Maschinentafel bewirkt —, so wird der Puffer von $\mathfrak{B}$ gelöscht und das letzte Eingabesymbol # eingelesen.

Nach der Einteilung von AHO und ULLMAN [6] sind für jede Klasse von Automaten die folgenden vier Typen zu betrachten: 2-Weg-nichtdeterminiert, 2-Weg-determiniert, 1-Weg-nichtdeterminiert, 1-Weg-deter-

miniert. Da schon der letzte Typ des PA das obige Ergebnis lieferte, erübrigt sich die Untersuchung der drei anderen Typen.

Der Wunsch, Automaten ihrer Definition nach möglichst stark einzuschränken, läßt noch das folgende Ergebnis interessant erscheinen: Zu jedem wie oben definierten PA $\mathfrak{B}$ gibt es einen PA $\mathfrak{B}'$, der jeweils nur einzelne Symbole in den Puffer einschreiben und auch nur jeweils in Abhängigkeit einzelner Symbole am Beginn des Pufferbandes reagieren kann, so daß $\mathfrak{B}$ und $\mathfrak{B}'$ zueinander äquivalent sind.

Auf den Beweis wird hier verzichtet.

2. Der einfache Pufferautomat

Die enge Verwandtschaft von Pufferautomaten und TURING-Maschinen läßt eine weitere Beschäftigung mit dem betrachteten Typ des PA als überflüssig erscheinen. Dieser Zusammenhang ergab sich im wesentlichen aus der Fähigkeit des Automaten, sein Eingabeband beliebig lange „anhalten" zu können und in dieser Zeit eine beliebig große Zahl von Änderungen des Zustandes und des Puffers ausführen zu können. Die Allgemeinheit der Definition soll deshalb eingeschränkt werden: Bei jedem Zustandsübergang und jeder Pufferänderung muß das Eingabeband jetzt um ein Feld vorgerückt werden. Bei solchen Automaten, bei denen genau nach der Anzahl von Zustandsübergängen, die gleich der Länge der vorgelegten Kette ist, entschieden ist, ob die Kette vom Automaten angenommen wird, spricht man von in Realzeit arbeitenden Automaten. Zum anderen wird gefordert, daß der Pufferlesekopf jeweils um höchstens soviele Felder in Richtung auf den Pufferschreibkopf hin bewegt werden darf, wie die Länge der maximalen Kette, die bei einem Zustandsübergang gelesen wird, beträgt.

Definition. *Ein einfacher Pufferautomat* $\mathfrak{B}$ *ist ein Sextupel* $(X, Z, P, B_2, \delta, z_0)$. *Dabei sind* X, Z *und* P *endliche, nichtleere Mengen,* $B_2 := \{b_{20}, b_{21}, \ldots, b_{2m}\}$, *wobei* m *gleich der Länge der größten Kette ist, die als dritte Komponente eines Argumentes der endlichen Korrespondenz* δ *auftritt,* $z_0 \in Z$, *und* δ *ist eine endliche Korrespondenz von* $Z \times X \times P^*$ *in* $Z \times P^* \times B_2$.

Das Annehmen von Ketten ist wie früher definiert. Die Menge der von $\mathfrak{B}$ angenommenen Ketten werde mit $T(\mathfrak{B})$ bezeichnet und Puffersprache genannt. An den folgenden beiden Sätzen, die wiederum ohne die (umfangreichen) Beweise zitiert werden, wird sich zeigen, daß auch bei dieser Realzeitversion eines PA die Art des Einschreibens in den Puffer und das Auslesen aus dem Puffer eingeschränkt werden kann.

Satz. *Sei* $\mathfrak{B} = (X, Z, P, B_2, \delta, z_0)$ *ein einfacher PA mit der endlichen Korrespondenz* δ *von* $Z \times X \times P^*$ *in* $Z \times P^* \times B_2$, *wobei die dritte Komponente eines beliebigen Argumentes von* δ *höchstens Ketten der Länge* 2 *enthalten darf. Dann gibt es einen einfachen PA* $\mathfrak{B}' = (X', Z', P', B_2', \delta', z_0')$ *mit der endlichen Korrespondenz* δ' *von* $Z' \times X' \times (P' \cup \{\varepsilon\})$ *in* $Z' \times P'^* \times B_2'$, *so daß gilt:* $u \in T(\mathfrak{B}) \Leftrightarrow u\,\# \in T(\mathfrak{B}')$, *wobei* $\# \notin X$.

Zur Konstruktion von $\mathfrak{B}'$ aus $\mathfrak{B}$ muß eine entsprechende Codierung der auf das Pufferband von $\mathfrak{B}'$ einzuschreibenden Information vorgenommen werden.

Bezüglich des Annehmens von Mengen bedeutet es keine Einschränkung, wenn jeweils nur ein Symbol (eventuell auch ε) in den Puffer geschrieben werden darf:

Satz. *Sei* $\mathfrak{B}' = (X', Z', P', B_2', \delta', z_0')$ *ein einfacher PA mit der endlichen Korrespondenz* δ' *von* $Z' \times X' \times (P' \cup \{\varepsilon\})$ *in* $Z' \times P'^* \times B_2'$. *Dann gibt es einen einfachen PA* $\mathfrak{B}'' = (X', Z, P, B_2', \delta, z_0')$ *mit der endlichen Korrespondenz* δ *von* $Z \times X' \times (P \cup \{\varepsilon\})$ *in* $Z \times (P \cup \{\varepsilon\}) \times B_2$, *so daß gilt:* $\mathfrak{B}'$ *und* $\mathfrak{B}$ *sind zueinander äquivalent.*

Der Beweis wird konstruktiv mittels Erweiterung der Zustandsmenge und des Pufferalphabets von $\mathfrak{B}'$ geführt.

Im folgenden soll die deterministische Variante des einfachen PA $\mathfrak{B}''$ aus dem obigen Satz eingehender betrachtet werden.

3. Einordnung der deterministischen Puffersprachen in die Typenhierarchie für formale Sprachen

In diesem Abschnitt soll gezeigt werden, wie sich die noch zu definierenden deterministischen Puffersprachen in die Hierarchie der formalen Sprachen nach CHOMSKY [7] einfügen. Es ergibt sich, daß sie nicht mit einem bestimmten Typ von Sprachen identifizierbar sind.

Die oben definierten einfachen PA sind nichtdeterministisch und unvollständig, d. h. bei einer beliebigen Konfiguration kann es für den Automaten keine, eine oder mehrere Möglichkeiten geben, seine Arbeit fortzusetzen. Der einfache deterministische PA ist dadurch gekennzeichnet, daß es für ihn jeweils genau eine Möglichkeit gibt weiterzuarbeiten.

Definition. *Ein einfacher deterministischer Pufferautomat (DPA)* $\mathfrak{B}$ *ist ein Sextupel* (X, Z, P, B_2, d, z_0). *Dabei sind* X, Z *und* P *endliche, nichtleere Mengen,* $B_2 := \{b_{20}, b_{21}\}$, $z_0 \in Z$, *und* d *ist eine Abbildung von* $Z \times X \times (P \cup \{\varepsilon\})$ *in* $Z \times (P \cup \{\varepsilon\}) \times B_2$.

Um das Verhalten von DPA bequem beschreiben zu können, wollen wir noch einige Vereinbarungen treffen. Für DPA werde über $Z \times X^* \times P^*$ die Relation $\vdash_{\mathfrak{B}}^*$ definiert:

Definition. *Seien* $z, z' \in Z$, $x \in X$, $w \in X^*$, $p \in P$, $p' \in P \cup \{\varepsilon\}$, $\pi \in P^*$. *Dann gelte* $(z, xw, p\pi) \vdash_{\mathfrak{B}} (z', w, \pi p')$, *falls* $d(z, x, p) = (z', p', b_{21})$, $(z, xw, p\pi) \vdash_{\mathfrak{B}} (z', w, p\pi p')$, *falls* $d(z, x, p) = (z', p', b_{20})$, *und* $(z, xw, \varepsilon) \vdash_{\mathfrak{B}} (z', w, p')$, *falls* $d(z, x, \varepsilon) = (z', p', b_{20})$ *oder* $d(z, x, \varepsilon) = (z', p', b_{21})$. *Für* $x^i \in X$ $(1 \leqslant i \leqslant k)$ *gelte* $(z, x^1 x^2 \ldots x^k w, \pi) \vdash_{\mathfrak{B}}^* (z', w, \pi')$, *falls* $z = z^1, z^2, \ldots, z^{k+1} = z' \in Z$, $\pi = \pi^1, \ldots, \pi^{k+1} = \pi' \in P^*$ *existieren, so daß für* $1 \leqslant i \leqslant k - 1$ *gilt:* $(z^i, x^i \ldots x^k w, \pi^i) \vdash_{\mathfrak{B}} (z^{i+1}, x^{i+1} \ldots x^k w, \pi^{i+1})$ *und außerdem* $(z^k, x^k w, \pi^k) \vdash_{\mathfrak{B}} (z', w, \pi')$. *Gilt* $(z, w, \pi) \vdash_{\mathfrak{B}}^* (z', w', \pi')$, *so*

sagt man, daß es in $\mathfrak{B}$ eine Übergangsfolge gibt, die (z, w, π) in (z', w', π') überführt.

Definition. *Sei $\mathfrak{B}$ ein DPA. Eine Kette $w \in \overline{X}^*$ heißt von $\mathfrak{B}$ angenommen, wenn gilt: $(z_0, w, \varepsilon) \vdash_{\mathfrak{B}}^* (z, \varepsilon, \varepsilon)$ mit $z \in Z$. Die Menge der von $\mathfrak{B}$ angenommenen Wörter werde mit $T(\mathfrak{B})$ bezeichnet und deterministische Puffersprache (DPS) genannt.*

Während bei endlichen erkennenden Automaten und bei TURING-Maschinen die deterministischen Versionen gegenüber den nichtdeterministischen bezüglich des Annehmens von Mengen keine Einschränkung bewirken, gilt für die einfachen PA — analog wie für Kellerautomaten:

Satz 3.1

(a) Jede DPS ist eine Puffersprache.

(b) Das Umgekehrte gilt nicht.

Beweis

(a) Nach Definition der Puffersprache ist Teil (a) des Satzes klar.

(b) Am Beispiel der Sprache $L := \{ww/w \in \{a, b\}^*\}$ soll (b) bewiesen werden. (1) Es werde angenommen, $\mathfrak{B}$ sei ein DPA, für den gilt: $T(\mathfrak{B}) = L$. Da die Zustandsmenge Z von $\mathfrak{B}$ endlich ist, gibt es Ketten a^{2i} und a^{2j} mit $i, j \in N$, $i \neq j$, und einen Zustand $z \in Z$, so daß gilt: $(z_0, a^{2i}, \varepsilon) \vdash_{\mathfrak{B}}^* (z, \varepsilon, \varepsilon)$ und $(*)$ $(z_0, a^{2j}, \varepsilon) \vdash_{\mathfrak{B}}^* (z, \varepsilon, \varepsilon)$. — a^{2i} und a^{2j} sind ja Elemente von L. — $a^{2i} b^l a^{2i} b^l$ ist ebenfalls ein Element von L, d. h. es gilt $(z_0, a^{2i} b^l a^{2i} b^l, \varepsilon) \vdash_{\mathfrak{B}}^* (z, b^l a^{2i} b^l, \varepsilon) \vdash_{\mathfrak{B}}^* (z', \varepsilon, \varepsilon)$ mit $z' \in Z$. Wegen der Determiniertheit von $\mathfrak{B}$ gilt aber auch nach $(*)$ $(z_0, a^{2j} b^l a^{2i} b^l, \varepsilon) \vdash_{\mathfrak{B}}^* (z, b^l a^{2i} b^l, \varepsilon) \vdash_{\mathfrak{B}}^* (z', \varepsilon, \varepsilon)$, d. h. $a^{2j} b^l a^{2i} b^l$ ist ein Element aus $T(\mathfrak{B})$, aber wegen $i \neq j$ kein Element aus L. Damit ist die obige Annahme zum Widerspruch geführt. (2) Sei $\mathfrak{B}'$ der folgendermaßen gegebene einfache Pufferautomat: $\mathfrak{B}' = (\{a, b\}, \{z_0, z_1, z_2\}, \{u, v\}, \{b_{20}, b_{21}\}, \delta, z_0)$, und δ ist bestimmt durch: $\delta[z_0, a, \varepsilon] = \{(z_1, u, b_{20})\}$, $\delta[z_0, b, \varepsilon] = \{(z_1, v, b_{20})\}$, $\delta[z_1, a, u] = \{(z_1, u, b_{20}), (z_2, \varepsilon, b_{21})\}$, $\delta[z_1, a, v] = \{(z_1, u, b_{20})\}$, $\delta[z_1, b, u] = \{(z_1, v, b_{20})\}$, $\delta[z_1, b, v] = \{(z_1, v, b_{20}), (z_2, \varepsilon, b_{21})\}$, $\delta[z_2, a, u] = \{(z_2, \varepsilon, b_{21})\}$, $\delta[z_2, b, v] = \{(z_2, \varepsilon, b_{21})\}$. Es soll auf den Beweis, daß $T(\mathfrak{B}') = L$ ist, verzichtet werden — wie dies in der Mathematischen Linguistik üblich ist —, da man es sofort einsieht. Damit ist gezeigt, daß es eine Sprache gibt, die von einem einfachen PA, nicht aber von einem DPA angenommen wird.

Um die DPS gegenüber anderen formalen Sprachen abzugrenzen, ist es bequem, die „entsprechenden" Automatentypen zu betrachten. Es sei zunächst an den endlichen erkennenden Automaten erinnert (Näheres siehe z. B. HÄNDLER [8]).

Definition. *Ein endlicher erkennender Automat $\mathfrak{E}$ ist ein Quintupel (X, Z, d, z_0, F). Dabei sind X und Z endliche, nichtleere Mengen, $F \subset Z$, $z_0 \in Z$, und d ist eine Abbildung von $Z \times X$ in Z.*

Bekanntlich gibt es zu jeder regulären Sprache einen endlichen erkennenden Automaten, der sie annimmt. Eine Kette $w \in X^*$ heißt dabei von $\mathfrak{E}$ angenommen, wenn sich der Automat nach dem Einlesen von w in einem Zustand aus F befindet. Über das Verhältnis der ε-freien regulären Sprachen (das sind diejenigen regulären Sprachen, die nicht die leere Kette ε enthalten) zu den DPS gibt der folgende Satz Auskunft:

Satz 3.2

(a) Jede ε-freie reguläre Sprache ist eine DPS.

(b) Das Umgekehrte gilt nicht.

Beweis

(a) Der Beweis wird über die zugehörigen Automaten geführt. Sei L eine ε-freie reguläre Sprache und $\mathfrak{E} = (X, Z, d, z_0, F)$ mit $z_0 \notin F$ sei ein endlicher erkennender Automat, der sie annimmt. Dann sei $\mathfrak{B} = (X, Z, \{p\}, \{b_{20}, b_{21}\}, d', z_0)$ ein DPA mit der folgendermaßen definierten Abbildung: Für alle $(z, x) \in Z \times X$ werde gesetzt:

$$\left.\begin{aligned} d'(z, x, \varepsilon) &= (z', p, b_{20}) \\ d'(z, x, p) &= (z', \varepsilon, b_{20}) \end{aligned}\right\} \text{ falls } d(z, x) = z' \wedge z' \in Z \setminus F$$

$$\left.\begin{aligned} d'(z, x, \varepsilon) &= (z'', \varepsilon, b_{20}) \\ d'(z, x, p) &= (z'', \varepsilon, b_{21}) \end{aligned}\right\} \text{ falls } d(z, x) = z'' \wedge z'' \in F$$

$\mathfrak{B}$ weist dieselben Zustandsübergänge auf wie $\mathfrak{E}$, schreibt das Symbol p in den Puffer bzw. läßt es ungeändert stehen, falls $\mathfrak{E}$ in einen nichtmarkierten Zustand übergeht und löscht den Puffer bzw. läßt ihn leer, falls $\mathfrak{E}$ in einen markierten Zustand übergeht. Es ist unmittelbar ersichtlich, daß gilt: $T(\mathfrak{B}) = L$.

(b) Man macht sich leicht klar, daß der folgende DPA $\mathfrak{B}$ die kontextfreie, nichtreguläre Sprache $\{a^n b a^n / n \in N\}$ annimmt: $\mathfrak{B} = (\{a, b\}, \{z_0, z_1, z_2\}, \{p\}, \{b_{20}, b_{21}\}, d, z_0)$, wobei (in abgekürzter Schreibweise): $d(z_0, a, \varepsilon) = d(z_0, a, p) = (z_0, p, b_{20})$, $d(z_0, b, p) = (z_1, \varepsilon, b_{20})$, $d(z_0, b, \varepsilon) = (z_2, p, b_{20})$, $d(z_1, a, p) = (z_1, \varepsilon, b_{21})$, $d(z_1, a, \varepsilon) = d(z_1, b, \varepsilon) = d(z_1, b, p) = d(z_2, a, \varepsilon) = d(z_2, b, \varepsilon) = d(z_2, a, p) = d(z_2, b, p) = (z_2, p, b_{20})$.

Im Teil (a) des Beweises mußte der Puffer mitbenutzt werden, um $\mathfrak{E}$ zu simulieren, da es reguläre Sprachen gibt — z. B. $\{(ab)^n / n \in N\}$ —, die nicht von endlichen erkennenden Automaten angenommen werden, für die $F = Z$ gilt.

Eine Abgrenzung der DPS „nach oben" vermittelt der folgende Satz:

Satz 3.3. *Jede DPS ist eine kontext-sensitive Sprache.*

Beweis. Bekanntlich ist eine Menge genau dann eine kontext-sensitive Sprache, wenn es einen linear-beschränkten Automaten gibt, der sie annimmt (siehe z. B. GINSBURG und ROSE [9]). Der linear-beschränkte Automat ist dadurch charakterisiert, daß die Länge des benötigten Arbeitsbandes nur linear von der Länge des vorgelegten Wortes abhängen darf. Da jeder DPA, durch den eine DPS angenommen wird, in Realzeit arbeitet,

ist die Länge des benutzten Pufferbandes durch die Länge des gegebenen Wortes begrenzt. Daraus folgt, daß jede DPS von einem linear-beschränkten Automaten angenommen wird.

Für die Klasse der DPS soll nun noch gezeigt werden, daß sie weder mit der Klasse der kontextfreien noch mit der der kontext-sensitiven Sprachen zusammenfällt.

Der bequemeren Sprechweise halber wird der Begriff des Pufferbildes eingeführt:

Definition. *Gilt in einem DPA* $\mathfrak{B} = (X, Z, P, B_2, d, z_0)$ $(z_0, u_1 u u_2, \varepsilon) \vdash_{\mathfrak{B}}^{*} (z, u u_2, \pi_1 \pi_2) \vdash_{\mathfrak{B}}^{*} (z', u_2, \pi_2 \pi)$ *mit* $z, z' \in Z$, $u_1, u_2, u \in X^*$, $\pi_1, \pi_2, \pi \in P^*$, *so heißt* π *das Pufferbild von* u *(in* $\mathfrak{B}$*).*

Satz 3.4

(a) Es gibt kontext-sensitive, nicht-kontextfreie Sprachen, die DPS sind.

(b) Es gibt deterministische kontext-sensitive, nicht-kontextfreie Sprachen, die keine DPS sind.

Beweis

(a) Man überzeugt sich leicht, daß der folgende DPA $\mathfrak{B}$ die nicht-kontextfreie Sprache $\{wcw \mid w \in \{\overline{a, b}\}^*\}$ annimmt: $\mathfrak{B} = (\{a, b, c\}, \{z_0, z_1, z_2\}, \{u, v\}, \{b_{20}, b_{21}\}, d, z_0)$, wobei d wie folgt (in abgekürzter Schreibweise) definiert ist: $d(z_0, a, \varepsilon) = d(z_0, a, u) = d(z_0, a, v) = (z_0, u, b_{20})$, $d(z_0, b, \varepsilon) = d(z_0, b, u) = d(z_0, b, v) = (z_0, v, b_{20})$, $d(z_0, c, \varepsilon) = (z_2, u, b_{20})$, $d(z_0, c, u) = d(z_0, c, v) = (z_1, \varepsilon, b_{20})$, $d(z_1, a, u) = d(z_1, b, v) = (z_1, \varepsilon, b_{21})$, $d(z_1, a, \varepsilon) = d(z_1, b, \varepsilon) = d(z_1, c, \varepsilon) = d(z_1, b, u) = d(z_1, c, u) = d(z_1, a, v) = d(z_1, c, v) = (z_2, u, b_{20})$, $d(z_2, a, \varepsilon) = d(z_2, a, u) = d(z_2, a, v) = d(z_2, b, \varepsilon) = d(z_2, b, u) = d(z_2, b, v) = d(z_2, c, \varepsilon) = d(z_2, c, u) = d(z_2, c, v) = (z_2, u, b_{20})$.

(b) Der Beweis erfolgt am Beispiel der deterministischen kontext-sensitiven Sprache $L = \{wvcvw \mid v \in \{\overline{0,1}\}^* \wedge w \in \{\overline{a, b}\}^*\}$. I. G. z. B. wird angenommen, es gäbe einen DPA $\mathfrak{B} = (X, Z, P, B_2, d, z_0)$, so daß gilt: $T(\mathfrak{B}) = L$. Zunächst ergibt sich, daß sich $\mathfrak{B}$ nach Abarbeiten der Hälfte eines Wortes aus L in einer durch den Zustand und den Pufferinhalt durch wv umkehrbar eindeutig bestimmten Konfiguration befinden muß. Würde nämlich gelten $(z_0, wvcvw, \varepsilon) \vdash_{\mathfrak{B}}^{*} (z, cvw, \pi)$ und $(z_0, w'v'cv'w', \varepsilon) \vdash_{\mathfrak{B}}^{*} (z, cv'w', \pi)$ mit $z \in Z$, $\pi \in P^*$ und $w' \neq w \vee v' \neq v$, so würde wegen der Determiniertheit des Automaten mit $wvcvw$ auch $w'v'cvw$ angenommen, und diese Kette ist nicht in L. Die Anzahl der Elemente von Z sei α_1 und die von P sei α_2 $(\geqslant 2)$. Mit Ketten im Puffer der Länge $\leqslant m$ lassen sich — mit Hilfe der Zustände — $< \alpha_2^{m+1} \alpha_1$ verschiedene Konfigurationen bilden. Andererseits gibt es 2^{l+k} verschiedene Ketten wv mit $|w| = l$ [4] und $|v| = k$. Sollen Ketten dieser Länge umkehrbar eindeutig Konfigurationen im DPA zugeordnet werden, so muß für die Pufferlänge m gelten: $2^{l+k} < \alpha_2^{m+1} \alpha_1$. D. f. $(*)$ $lc_1 + kc_1 < (m+1)c_2 + c_3$ mit Konstanten $c_1, c_2 > 0$, $c_3 \geqslant 0$. Für ein Wort aus

[4] $|w|$ bezeichnet die Länge einer Kette $w \in A^*$; insbesondere gilt $|\varepsilon| = 0$.

L existiert eine Übergangsfolge $(z_0, wvcvw, \varepsilon) \vdash_{\mathfrak{B}}^{*} (z^1, vcvw, \pi_1) \vdash_{\mathfrak{B}}^{*} (z^2, cvw, \pi_2) \vdash_{\mathfrak{B}}^{*} (z^3, w, \pi_3)$ mit $z^1, z^2, z^3 \in Z$, $\pi_1, \pi_2, \pi_3 \in P^*$. Da die Pufferlänge linear mit der Länge von w wächst, gibt es in L beliebig viele Wörter für die gilt $|\pi_1| > |vcv|$, wobei $|v|$ — in Abhängigkeit von π_1 — beliebig groß werden kann. D. h. nach dem Einlesen von w enthält der Puffer so viele Symbole, daß sie im Verlauf des Einlesens von vcv nicht alle gelöscht oder überschrieben werden können. — Beim Einlesen eines Eingabesymbols kann ja höchstens ein Symbol aus dem Puffer ausgelesen werden. — Um nun die beiden Ketten v — sie seien als $v^{(1)}$ und $v^{(2)}$ bezeichnet — auf Übereinstimmung zu prüfen, ist es nötig, nach dem vollständigen Auslesen des Pufferbildes von w ihre Pufferbilder zu vergleichen. Da das Pufferbild der ersten Kette $v^{(1)}$ während des Einlesens der zweiten Kette $v^{(2)}$ für den Pufferlesekopf nicht erreichbar war, werden die Pufferbilder von $v^{(1)}$ und $v^{(2)}$ unabhängig voneinander in den Puffer geschrieben. Aus (*) folgt, daß es Wörter in L gibt, für die gilt: Zum Zeitpunkt des vollständigen Ausgelesenseins des Pufferbildes von w ist die Anzahl K der Symbole im Puffer, d. h. die Länge des Pufferbildes von vcv, eine lineare Funktion von $|v| = k$. Da der Pufferlesekopf lediglich Zugriff zu einem am linken Ende des Puffers befindlichen Symbol hat, müssen die Pufferbilder von $v^{(1)}$ und $v^{(2)}$ abwechselnd an das linke Ende geschoben werden. Wegen der endlichen Zustandszahl können aber während eines solchen Vorganges nur eine endliche Anzahl von Puffersymbolen verglichen und damit auch gelöscht werden. Der Vergleich beansprucht deshalb eine Anzahl von Zustandsübergängen — wobei jeweils auch ein Symbol vom Eingabeband eingelesen wird —, die proportional zu K^2 und damit auch zu k^2 ist. Da aber nach dem Einlesen der Teilkette $wvcv$ nur noch $|w| = l$ Zustandsübergänge möglich sind, gibt es nach (*) Wörter in L, für die der Vergleich nicht durchführbar ist, so daß auch Wörter der Form $wvcv'w$ mit $v \neq v'$ angenommen werden. Damit ist die Annahme, L sei DPS, zum Widerspruch geführt.

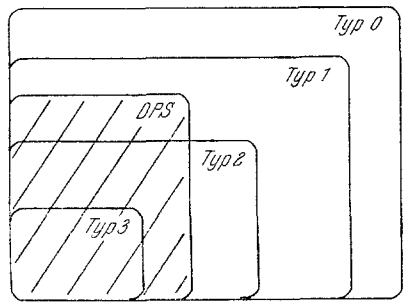

Abb. 2. Lage der DPS in der CHOMSKY-Hierarchie

Satz 3.5

(a) Es gibt kontextfreie, nicht-reguläre Sprachen, die DPS sind.

(b) Es gibt deterministische kontextfreie Sprachen, die keine DPS sind.

Beweis

(a) Siehe Beweis von Satz 3.2 (b).

(b) Der Beweis erfolgt mit denselben Worten wie der von Satz 3.4 (b) am Beispiel der deterministischen kontextfreien Sprache

$$\{wvcv^Rw^R \mid v \in \{\overline{0,1}\}^* \wedge w \in \{\overline{a,b}\}^*\}.^5$$

Damit ist die Stellung der DPS in der CHOMSKY-Hierarchie geklärt (siehe Abb. 2).

4. Operationen mit deterministischen Puffersprachen

In diesem Abschnitt soll das Verhalten der DPS gegenüber einigen charakteristischen Operationen untersucht werden. Es zeigt sich, daß die üblichen Operationen nicht klassenerhaltend sind. Dies ergibt sich teilweise durch die Determiniertheit der Sprachen und teilweise durch die Definition des Annehmens von Mengen (Leersein des Puffers). Erwähnt sei, daß es einige spezielle Operationen gibt, die nicht aus der Klasse hinausführen.

Satz 4.1. *Die Klasse der DPS ist bezüglich der*

(a) Durchschnittsbildung,

(b) Vereinigungsbildung und

(c) Spiegelung nicht abgeschlossen.

Beweis

(a) Der Beweis des Teils (a) ergibt sich sofort am Beispiel der Menge $\{wvcvw \mid v \in \{\overline{0,1}\}^* \wedge w \in \{\overline{a,b}\}^*\}$, von der im Beweis von Satz 3.4 (b) gezeigt wurde, daß sie keine DPS ist, und die als Durchschnitt der beiden DPS $L_1 = \{wv_1cv_2w \mid v_1, v_2 \in \{\overline{0,1}\}^* \wedge w \in \{\overline{a,b}\}^*\}$ und $L_2 = \{w_1vcvw_2 \mid v \in \{\overline{0,1}\}^* \wedge w_1, w_2 \in \{\overline{a,b}\}^*\}$ gebildet werden kann. Daß L_1 und L_2 DPS sind, wird lediglich am Konstruktionsprinzip entsprechender DPA plausibel gemacht: Um L_1 anzunehmen, wird zunächst das Pufferbild der ersten Teilkette w in den Puffer geschrieben, ohne Änderung des Puffers wird nachgeprüft, ob v_1 und v_2 Elemente aus $\{0,1\}^*$ sind, und anschließend wird die zweite Teilkette w mit dem Pufferinhalt verglichen. Ein Automat, der L_2 annimmt, schreibt zunächst ein Sondersymbol in den Puffer, überprüft, ob $w_1 \in \{a,b\}^*$, löscht beim Einlesen von v das Sondersymbol und schreibt gleichzeitig das Pufferbild von v ein, vergleicht es beim Einlesen der zweiten Teilkette v damit und schreibt beim Leerwerden des Puffers wiederum ein Sondersymbol ein, das beim Einlesen des ersten Symbols von w_2 gelöscht wird. Dann wird nur noch einmal etwas in den Puffer geschrieben, wenn $w_2 \notin \{a,b\}^*$.

(b) Es wird gezeigt, daß die als Vereinigung zweier DPS L_1 und L_2 gebildete Sprache (nach RABIN [10]) $L = \{uvdu\} \cup \{uvev\}$, wobei $u \in \{\overline{a,b}\}^*$, $v \in \{\overline{0,1}\}^*$ keine DPS darstellt. Auf den Nachweis, daß L_1

[5] Dabei gilt: $\varepsilon^R = \varepsilon$. Für $a_1 \ldots a_n \in A^*$: $(a_1 \ldots a_n)^R = a_n \ldots a_1$.

und L_2 DPS sind, wird hier verzichtet. Die Annahme, es gäbe einen DPA $\mathfrak{B}$ mit $T(\mathfrak{B}) = L$, soll zum Widerspruch geführt werden. Wie im Beweis von Satz 3.4 (b) zeigt man, daß sich $\mathfrak{B}$ nach Einlesen einer Teilkette uv in einer durch den Zustand und den Pufferinhalt umkehrbar eindeutig durch uv bestimmten Konfiguration befinden muß. Außerdem gilt wieder das dort über die Pufferlänge Gesagte. D. f. dann, daß es in L beliebig viele Wörter gibt, für die die Länge des Pufferbildes von u größer als die Länge von v ist, d. h. es gibt in $\mathfrak{B}$ eine Übergangsfolge $(z_0, uvev, \varepsilon) \vdash_{\mathfrak{B}}^{*} (z, ev, \pi)$ mit $|\pi| > |ev|$. Während des Abarbeitens von ev kann der Puffer nicht leer werden und damit das Wort $uvev$ nicht angenommen werden, obwohl es Element von L ist.

(c) Teil (c) des Satzes wird am Beispiel der obigen Sprache L gezeigt, die sich als Spiegelung der DPS $L' = \{udvu\} \cup \{vevu\}$ mit $u \in \{\overline{a, b}\}^*$ und $v \in \{\overline{0,1}\}^*$ darstellt. Daß L' eine DPS ist, überlegt man sich leicht.

Definition. *Seien X_1 und X_2 nichtleere Mengen. Ein Homomorphismus τ von X_1^* in X_2^* ist eine Abbildung von X_1^* in X_2^*, so daß gilt: $\tau(\varepsilon) = \varepsilon$ und $\tau(x^1 \ldots x^k) = \tau(x^1) \ldots \tau(x^k)$ für alle $k \geqslant 1$ und $x^i \in X_1$ $(1 \leqslant i \leqslant k)$. Gilt für alle $x \in X_1$ $\tau(x) \neq \varepsilon$, so heißt τ ε-freier Homomorphismus.*

Satz 4.2. *Die Klasse der DPS ist bezüglich der (a) Homomorphismenbildung und (b) ε-freien Homomorphismenbildung nicht abgeschlossen.*

Beweis

(a) Man überzeugt sich leicht, daß $L = \{fuvdu\} \cup \{guvev\}$ mit $u \in \{\overline{a, b}\}^*$ und $v \in \{\overline{0,1}\}^*$ eine DPS ist. Sei τ der folgendermaßen definierte Homomorphismus: $\tau(f) = \tau(g) = \varepsilon$; $\tau(a) = a$; $\tau(b) = b$; $\tau(d) = d$; $\tau(e) = e$; $\tau(0) = 0$; $\tau(1) = 1$. Dann ist $\tau(L)$ nach dem Beweis von Satz 4.1 (b) keine DPS.

(b) Nach dem Beweis von Satz 3.4 (a) ist $L' = \{wcw \mid w \in \{\overline{a, b}\}^*\}$ eine DPS. Sei τ' der folgende ε-freie Homomorphismus: $\tau'(a) = \tau'(c) = a$; $\tau'(b) = b$. Wie im Beweis von Satz 3.1 (b) ergibt sich, daß $\tau'(L')$ keine DPS ist.

Satz 4.3. *Die Klasse der DPS ist bezüglich der Differenzbildung nicht abgeschlossen.*

Beweis. Der Beweis erfolgt am Beispiel der Menge $L = \{b^{l+k} a^k \mid k, l \in N\}$, die sich als Differenz der beiden DPS $\{a^i b^j \mid i, j \in N\}$ und $\{a^i b^{i+m} \mid i \in N, m \in N \cup \{0\}\}$ ergibt und selbst keine DPS ist. Ein DPA, der L annehmen sollte, müßte in eine durch den Zustand und den Pufferinhalt gekennzeichnete, durch b^k umkehrbar eindeutig bestimmte Konfiguration gelangen. Wegen der Determiniertheit des Automaten wird aber, da l beliebig groß werden kann, der Pufferinhalt so groß, daß er während des Einlesens von a^k nicht mehr gelöscht werden kann.

Satz 4.4

(a) Es gibt DPS L und reguläre Sprachen R, so daß RL keine DPS sind.

(b) Es gibt DPS L' und reguläre Sprachen R', so daß $L'R'$ keine DPS sind.

Beweis

(a) Es wird die Konkatenation der regulären Sprache $R = \{b^l \mid l \in N\}$ und der DPS $L = \{b^k a^k \mid k \in N\}$ gebildet. $RL = \{b^{l+k} a^k \mid k, l \in N\}$ ist nach dem Beweis von Satz 4.3 keine DPS.

(b) Man überzeugt sich leicht, daß die Menge L', die aus solchen Wörtern aus $\{\overline{a, b}\}^*$ besteht, die die gleiche Anzahl von a's und b's enthalten, eine DPS ist. Die Konkatenation mit der regulären Sprache $R' = \{a^i \mid i \in N\}$ stellt keine DPS dar. Es sei nämlich angenommen, es gäbe einen DPA $\mathfrak{B} = (X, Z, P, B_2, d, z_0)$, der diese Menge annimmt. Wie oben sieht man, daß es dann Wörter mit genügend großem k gibt, so daß sich $\mathfrak{B}$ nach dem Einlesen von $b^i a^i a^k$ in einer durch Zustand und Pufferinhalt umkehrbar eindeutig durch a^k bestimmten Konfiguration befinden muß, d. h. es muß gelten $(z_0, b^i a^i a^k b^k, \varepsilon) \vdash_{\mathfrak{B}}^* (z, a^{k-1} b^k, \varepsilon)$, da $b^i a^{i+1}$ auch $\in L'R'$ und $(z, a^{k-1} b^k, \varepsilon) \vdash_{\mathfrak{B}}^* (z', b^k, \pi)$ mit $\pi \in \overline{P}^*$. Wegen der Determiniertheit von $\mathfrak{B}$ gilt dann auch $(z_0, b^i a^i a^k, \varepsilon) \vdash_{\mathfrak{B}}^* (z', \varepsilon, \pi)$. D. h. $b^i a^i a^k \notin T(\mathfrak{B})$, was wegen $b^i a^i a^k \in L'R'$ einen Widerspruch zur Annahme, daß $T(\mathfrak{B}) = L'R'$ sein soll, darstellt.

5. Entscheidbarkeitsfragen bei Puffersprachen

Zu den charakterisierenden Eigenschaften von Sprachen gehören neben ihrer Einordnung in die CHOMSKY-Hierarchie und ihrem Verhalten gegenüber gewissen Operationen auch die Möglichkeit bzw. Unmöglichkeit, bestimmte Fragen, wie z. B. die nach der Gleichheit zweier beliebiger Sprachen dieser Klasse, algorithmisch entscheiden zu können.

Zunächst sei ein entscheidbares Problem formuliert:

Satz 5.1. *Es ist entscheidbar, ob ein beliebiges Wort Element einer beliebig vorgegebenen Puffersprache ist.*

Dies folgt unmittelbar aus Satz 3.3.

Die Beweise der folgenden Unentscheidbarkeitsaussagen beruhen auf dem bekannten POSTschen Korrespondenzproblem (POST [11]) und verlaufen in Analogie zu entsprechenden Sätzen für kontextfreie Sprachen (siehe z. B. GINSBURG [3] und LANDWEBER [12]), die wiederum im wesentlichen auf einen Beweisgedanken von RABIN und SCOTT [13] zurückzuführen sind. Hier soll auf sie verzichtet werden.

Satz 5.2. *Für DPS L_1 und L_2 sind die folgenden Probleme rekursiv unentscheidbar:*

(a) Ist $L_1 \cap L_2 = \emptyset$?

(b) Ist $L_1 \cap L_2$ eine endliche Menge?

(c) Ist $L_1 \cap L_2$ eine reguläre Sprache?

(d) Ist $L_1 \cap L_2$ eine kontextfreie Sprache?

Satz 5.3. *Für Puffersprachen L_1 und L_2 sind die folgenden Probleme rekursiv unentscheidbar:*

(a) Gilt $L_1 \subseteq L_2$?

(b) Gilt $L_1 = L_2$?

Satz 5.4. *Für Puffersprachen L, kontextfreie Sprachen C und reguläre Sprachen R sind die folgenden Probleme rekursiv unentscheidbar:*

(a) Gilt $L = C$?

(b) Gilt $L = R$?

Literatur

[1] DAVIS, M.: Computability and Unsolvability. New York-London: McGraw-Hill. 1958.

[2] VOLLMAR, R.: Über einen Automaten mit Speicherband (Pufferautomat). Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung der Universität Erlangen-Nürnberg **2**, 1, (1969).

[3] GINSBURG, S.: The Mathematical Theory of Context-Free Languages. New York-London: McGraw-Hill. 1966.

[4] SHEPHERDSON, J. C., and H. E. STURGIS: Computability of Recursive Functions. J. ACM **10**, 217–255 (1963).

[5] HERMES, H.: Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit. Berlin-Göttingen-Heidelberg: Springer-Verlag. 1961.

[6] AHO, A. V., and J. D. ULLMAN: The Theory of Languages. Mathematical Systems Theory **2**, 97–125 (1968).

[7] CHOMSKY, N.: On Certain Formal Properties of Grammars. Information and Control **2**, 137–167 (1959).

[8] HÄNDLER, W.: Automatentheorie II (Vorlesungsausarbeitung). Hannover. 1966.

[9] GINSBURG, S., and G. F. ROSE: Preservation of Languages by Transducers. Information and Control **9**, 153–176 (1966).

[10] RABIN, M. O.: Real Time Computation. Israel Journal of Math. **1**, 203–211 (1963).

[11] POST, E. L.: A Variant of a Recursively Unsolvable Problem. Bull. Am. Math. Soc. **52**, 264–268 (1946).

[12] LANDWEBER, P. S.: Decision Problems of Phrase-Structure Grammars. IEEE Trans. on Electronic Computers **EC-13**, 354–362 (1964).

[13] RABIN, M. O., and D. SCOTT: Finite Automata and Their Decision Problems. IBM J. Res. Dev. **3** 114–125 (1959).

Dr.-Ing. Roland Vollmar
Institut für Mathematische Maschinen
und Datenverarbeitung
der Universität Erlangen-Nürnberg
Egerlandstraße 5, D-852 Erlangen
Bundesrepublik Deutschland