Über das Rechnen mit den Elementen abstrakt präsentierter Halbgruppen

Von H. JÜRGENSEN

Herrn Professor László Kalmár zum Gedächtnis

$X \neq \emptyset$ sei eine endliche Menge, X^+ die von X erzeugte freie Halbgruppe. R sei eine endliche Menge von Gleichungen (Relationen) über X^+. $\varrho_{(X,R)}$ sei die von R erzeugte Kongruenz auf X^+. Dann ist $S_{(X,R)} = X^+/\varrho_{(X,R)}$ die durch (X, R) präsentierte Halbgruppe. In der vorliegenden Arbeit geht es darum, eine Klasse $\mathfrak{N}$ von endlichen „normierten" Präsentationen anzugeben, für die gilt:

(1) $\mathfrak{N}$ wird „modulo Gruppen" formal beschrieben.

(2) Jede endliche Halbgruppe besitzt in $\mathfrak{N}$ eine Präsentation.

(3) Für $(X, R) \in \mathfrak{N}$ ist die durch (X, R) präsentierte Halbgruppe endlich.

(4) Aus der Beschreibung von $\mathfrak{N}$ läßt sich „modulo Gruppenelementen" eine formale Beschreibung „normierter Wörter" angeben, so daß für jede Präsentation $(X, R) \in \mathfrak{N}$ jede $\varrho_{(X,R)}$-Klasse ein normiertes Wort enthält.

(5) Zu jeder endlichen Halbgruppe gibt es eine Präsentation $(X, R) \in \mathfrak{N}$, so daß jede $\varrho_{(X,R)}$-Klasse genau ein normiertes Wort enthält.

(6) Aus der Beschreibung von $\mathfrak{N}$ und der normierten Wörter läßt sich ein „modulo Gruppenelementen" leicht programmierbarer, recht effizienter Algorithmus zum Normieren von Wörtern und damit zum Rechnen mit den normierten Wörtern angeben.

Wie schon in diesen Forderungen formulieren wir die meisten Aussagen und den Algorithmus zunächst nur „modulo Gruppen", d. h. unter Verwendung von Orakeln für das Rechnen mit Gruppen. Im Abschnitt 3 geben wir dann einige Hinweise auf Realisierungsmöglichkeiten unter der Voraussetzung spezieller Gruppenpräsentationen.

1. Normierte Präsentationen und normierte Wörter

1.1. Definition. (X, R) sei eine Halbgruppenpräsentation. (X, R) ist eine *normierte Präsentation*, wenn gilt:

(1) (X, R) ist endlich.

(2) X besitzt eine Partition in Mengen $X_0, X_1, \ldots, X_n$ mit $X_i = \{c^i\}$ oder $X_i = \{a_0^i, \ldots, a_{k_i}^i, b_0^i, \ldots, b_{m_i}^i\} \cup E_i$ für $i = 1(1)n$ und $X_0 = \{a_0^0, \ldots, a_{k_0}^0, b_0^0, \ldots, b_{m_0}^0\} \cup E_0$ mit $k_0, m_0, k_i, m_i \in \mathbf{N} \cup \{0\} = \mathbf{N}_0$ und $E_0, E_i \neq \emptyset$. Sei $X^i = \bigcup_{j=0}^{i} X_j$.

(3) R enthält genau die folgenden Relationen (jeweils für alle Indizes, für die die Symbole erklärt sind):

(a) Mengen S_i von Relationen über E_i^+, so daß (E_i, S_i) Halbgruppenpräsentation einer endlichen Gruppe G_i ist. $e^i \in E_i^+$ sei das Einselement von G_i.

(b) $a_0^i = e^i = b_0^i$.

(c) $a_j^i e^i = a_j^i$.

(d) $e^i b_j^i = b_j^i$.

(e) $b_j^i a_l^i = p_{jl}^i \in E_i^+ \cup (X^{i-1})^+$ für $(j, l) \neq (0, 0)$.

(f) $xy = q_{x,y} \in (X^{\min(i,j)})^+$ für $x \in X_i$, $y \in X_j$, $i \neq j$.

(g) $c^i c^i = r^i \in (X^{i-1})^+$.

Mit $\mathfrak{N}$ sei die Klasse aller normierten Präsentationen bezeichnet. Die offenbar überflüssigen a_0^i, b_0^i dienen nur zur Vermeidung von Fallunterscheidungen.

1.2. Lemma. Sei $(X, R) \in \mathfrak{N}$, $\varrho = \varrho_{(X,R)}$. Jede ϱ-Klasse von X^+ enthält ein Wort der Form c^i oder $a_j^i g^i b_k^i$ mit $g^i \in E_i^+$.

Beweis. $w = x_1 \dots x_v$ sei ein Wort mit $x_1, \dots, x_v \in X$. Sei zunächst $v = 1$. Falls $w \neq c^i$ ist, gilt

$$w = a_j^i \varrho a_j^i e^i b_0^i$$

oder

$$w = b_j^i \varrho a_0^i e^i b_j^i$$

oder

$$w = g^i \varrho e^i g^i e^i \varrho a_0^i g^i b_0^i \quad \text{mit} \quad g^i \in E_i.$$

Sei also jetzt $v > 1$. $\alpha(w)$ sei das Maximum der oberen Indizes der Symbole in w, $\beta(w)$ das Minimum. $\gamma(w)$ sei die Anzahl der Symbole in w mit oberem Index $\alpha(w)$. Wir unterscheiden zwei Fälle:

$\alpha(w) \neq \beta(w)$: Dann gibt es in w x_j, x_{j+1} mit $x_j \in X_{\alpha(w)}$, $x_{j+1} \notin X_{\alpha(w)}$ oder umgekehrt. Anwendung von (f), d. h. Ersetzen von $x_j x_{j+1}$ durch $q_{x_j, x_{j+1}}$ ergibt ein zu w ϱ-äquivalentes Wort w' mit $\alpha(w) = \alpha(w')$, $\gamma(w) > \gamma(w') \geqq 1$ oder $\alpha(w) > \alpha(w')$. Mit endlich vielen Anwendungen von (f) erhält man daher so ein zu w ϱ-äquivalentes Wort w'' mit $\alpha(w'') = \beta(w'') \leqq \beta(w)$.

$\alpha(w) = \beta(w)$: Falls $X_{\alpha(w)} = \{c^{\alpha(w)}\}$ ist, ist w wegen (g) zu

$$w' = \underbrace{r^{\alpha(w)} c^{\alpha(w)} \dots c^{\alpha(w)}}_{v-2 \text{ mal}}$$

ϱ-äquivalent; dabei ist für $v > 2$ $\alpha(w) = \alpha(w') \neq \beta(w')$ und für $v = 2$ $\alpha(w') < \alpha(w)$. Durch endlich viele Anwendungen von (f) und (g) erhält man also ein zu w ϱ-äquivalentes Wort w' mit $\alpha(w') = \beta(w')$ und $X_{\alpha(w')} \neq \{c^{\alpha(w')}\}$ oder $|w'| = 1$. Wir können dies also schon für w voraussetzen. Durch endlich viele Anwendungen von (b) bis (e), nämlich durch Ersetzen

von $b_j^i a_l^i$ für $(j, l) \neq (0, 0)$ durch p_{jl}^i,

von $b_0^i a_0^i$ durch e^i,

von $h^i a_j^i$ durch $h^i p_{0j}^i$, von $b_j^i h^i$ durch $p_{j0}^i h^i$ für $h^i \in E_i$, $j \neq 0$,

von $a_j^i a_l^i$ durch $a_j^i p_{0l}^i$ für $l \neq 0$ und durch a_j^i für $l = 0$,

von $b_j^i b_l^i$ durch $p_{j0}^i b_l^i$ für $j \neq 0$ und durch b_l^i für $j = 0$,

erhält man ein zu w ϱ-äquivalentes Wort w' mit $\beta(w')<\alpha(w')$ oder $w'\in E_i^+$ oder $w'\in a_j^i E_i^+$ oder $w'\in E_i^+ b_1^i$ oder $w'\in a_j^i E_i^+ b_1^i$. Im ersten Falle schließt man für w' statt w wie oben weiter. In den übrigen Fällen hat $a_0^i w' b_0^i$, $w' b_0^i$, $a_0^i w'$, bzw. w' die gewünschte Form und ist zu w ϱ-äquivalent wegen (b). □

1.3. Lemma. Für $(X, R)\in\mathfrak{R}$ hat $S_{(X,R)}$ höchstens die Ordnung

$$\delta(X, R) = c+\Sigma|G_i|(k_i+1)(m_i+1),$$

wobei die Summation über die i mit $|X_i|\neq 1$ durchgeführt wird und c die Anzahl der i mit $|X_i|=1$ ist.

Beweis. Es gibt höchstens c Wörter der Form c^i, und wegen 1.1. (3a) gibt es zu festen j, l höchstens $|G_i|$ paarweise nicht-äquivalente Wörter der Form $a_j^i g^i b_1^i$ mit $g^i\in E_i^+$. Damit gibt es höchstens $\delta(X, R)$ paarweise nicht-äquivalente Wörter dieser Formen, und aus 1.2 folgt die Behauptung. □

1.4. Definition. Sei $(X, R)\in\mathfrak{R}$. Zu jedem vorkommenden Paar (E_i, S_i) sei ein Repräsentantensystem $\{s^i\}$ der $\varrho_{(E_i,S_i)}$-Klassen beliebig, aber fest gewählt. Die Wörter der Form c^i und $a_j^i s^i b_l^i$ heißen *normiert*.

Zusammenfassend erhält man:

1.5. Satz. Die durch $(X, R)\in\mathfrak{R}$ präsentierte Halbgruppe ist endlich. Sie hat genau dann die Ordnung $\delta(X, R)$, wenn jede $\varrho_{(X,R)}$-Klasse genau ein normiertes Wort enthält.

Aus dem Beweis von 1.2 folgt weiter, indem man ein Orakel für das Rechnen mit Gruppenelementen voraussetzt:

1.6. Satz. Sei $(X, R)\in\mathfrak{R}$ und $\delta(X, R)=|S_{(X,R)}|$. Es gibt „modulo Gruppen" einen Algorithmus, der zu jedem Wort das $\varrho_{(X,R)}$-äquivalente normierte Wort berechnet.

Einen allgemeinen Beweis der Entscheidbarkeit des Wortproblems „modulo Gruppen" für beliebige normierte Halbgruppenpräsentationen (d. h. ohne die Forderung $\delta(X, R)=|S_{(X,R)}|$) erhält man wegen 1.5 aus [5], wo wir das Todd—Coxeter-Verfahren auf Halbgruppen übertragen haben. Der daraus resultierende Algorithmus für das Wortproblem ist jedoch sehr aufwendig.

Die Umkehrung von 1.5 erhält man mit Hilfe bekannter Strukturätze für endliche Halbgruppen [z. B. 1]:

1.7. Satz. Jede endliche Halbgruppe S besitzt eine normierte Präsentation (X, R) mit $\delta(X, R)=|S|$.

Beweis. S sei eine endliche Halbgruppe. S hat eine Kompositionsreihe

$$\Sigma_0 \subseteqq \Sigma_1 \subseteqq \ldots \subseteqq \Sigma_n = S,$$

wobei Σ_0 einfach und Σ_{i+1}/Σ_i 0-einfach oder 0 von der Ordnung 2 ist. Die Behauptung wird durch Induktion nach n bewiesen.

$n=0$: $S=\Sigma_0$ ist als endliche einfache Halbgruppe vollständig einfach und daher Rechteckhalbgruppe isomorpher Gruppen H_{jl} mit $j=0(1)k_0$, $l=0(1)m_0$. (E_0, S_0)

sei eine endliche Halbgruppenpräsentation von $H_{00}=G_0$, und $e^0\in E_0^+$ sei das Einselement von G_0. Sei $a_0^0=e^0=b_0^0$ und $a_j^0\in H_{j0}$, $b_l^0\in H_{0l}$ beliebig für $j, l\geqq 1$. Sei

$$X_0=\{a_0^0, a_1^0, \ldots, a_{k_0}^0\}\cup\{b_0^0, b_1^0, \ldots, b_{m_0}^0\}\cup E_0.$$

Die so gewählten Elemente erfüllen die Bedingungen aus 1.1 für $X=X_0$: (3a), (3b) gelten nach Wahl von X_0. Es ist $a_j^0H_{00}=H_{j0}$ und daher $a_j^0g=a_j^0$ für ein $g\in H_{00}$, also

$$a_j^0e^0=a_j^0ge^0=a_j^0g=a_j^0$$

und daher (3c). Analog folgt (d). Wegen $H_{0l}H_{j0}=H_{00}$ folgt (e), wenn man für p_{lj}^0 das Produkt $b_l^0a_j^0$ wählt. (f) und (g) sind leer. S ist daher homomorphes Bild der durch (X, R) mit

$$R=S_0\cup\left\{a_0^0=e^0=b_0^0,\ a_j^0e^0=a_j^0,\ e^0b_j^0=b_j^0,\ b_l^0a_j^0=p_{lj}^0\ \middle|\ \begin{matrix}j=0(1)k_0\\ l=0(1)m_0\\ (j,l)\neq(0,0)\end{matrix}\right\}$$

präsentierten Halbgruppe $S_{(X,R)}$. Wegen

$$|S|=|H_{00}|(k_0+1)(m_0+1)=\delta(X,R)$$

ist $S\cong S_{(X,R)}$.

Die Behauptung sei nun für alle Halbgruppen mit Kompositionsketten der Länge $\leqq n-1$ für $n\geqq 1$ bewiesen. S sei eine endliche Halbgruppe mit Kompositionskette der Länge

$n\geqq 1$: Für Σ_{n-1} sei eine normierte Präsentation (X', R') gegeben. Wir unterscheiden zwei Fälle:

Σ_n/Σ_{n-1} ist Nullhalbgruppe: $c^n\in\Sigma_n/\Sigma_{n-1}$ sei das von 0 verschiedene Element, $X_n=\{c^n\}$, $X=X'\cup X_n$,

$$R=R'\cup\{c^nc^n=r^n,\ xc^n=q_{x,c^n},\ c^nx=q_{c^n,x}|x\in X'\}.$$

Dabei sind $r^n, q_{x,c^n}, q_{c^n,x}$ Darstellungen der entsprechenden Produkte in S, die in $X'^+=(X^{n-1})^+$ gewählt werden können, weil sie in Σ_{n-1} liegen. Damit ist $(X, R)\in\mathfrak{N}$ mit $\delta(X, R)=|S|$. Weil R in S gilt, wird S durch (X, R) präsentiert.

Σ_n/Σ_{n-1} ist 0-einfach: Mit $j=0(1)k_n$, $l=0(1)m_n$ seien die $\mathscr{R}$-Klassen und $\mathscr{L}$-Klassen von Σ_n/Σ_{n-1} o. B. d. A. so indiziert, daß die $\mathscr{H}$-Klasse H_{00} eine Gruppe ist. (E_n, S_n) sei eine endliche Halbgruppenpräsentation der Gruppe $G_n=H_{00}$, und e^n sei eine Darstellung ihres Einselementes. Sei

$$X_n=\{a_0^n, a_1^n, \ldots, a_{k_n}^n\}\cup\{b_0^n, b_1^n, \ldots, b_{m_n}^n\}\cup E_n,$$

wo $a_j^n\in H_{j0}$, $b_l^n\in H_{0l}$ beliebig und $a_0^n=b_0^n=e^n$ gewählt werden. (3c) und (3d) gelten wie oben. Für $(l, j)\neq(0, 0)$ liegt das Produkt $b_l^na_j^n$ in H_{00} oder in Σ_{n-1}, kann also als

$$p_{lj}^n\in E_n^+\cup(X^{n-1})^+$$

dargestellt werden. Damit gilt (3e). Die $q_{x,y}$ mit $x \in X_n$ oder $y \in X_n$ können in $(X^{n-1})^+$ gewählt werden, weil die entsprechenden Produkte in Σ_{n-1} liegen. Also gilt auch (f). (g) ist leer. S ist daher homomorphes Bild der durch (X, R) mit $X = X' \cup X_n$ und

$$R = R' \cup \left\{ a_0^n = e^n = b_0^n,\ a_j^n e^n = a_j^n,\ e^n b_j^n = b_j^n,\ b_l^n a_j^n = p_{lj}^n \;\middle|\; \begin{array}{l} j = 0(1)k_0 \\ l = 0(1)m_0 \\ (j, l) \neq (0, 0) \end{array} \right\}$$

$$\cup \{xy = q_{x,y} | (x, y) \in (X \times X) \setminus (X_n \times X_n)\}$$

präsentierten Halbgruppe $S_{(X,R)}$. Wegen

$$\begin{aligned} |S| &= |\Sigma_{n-1}| + |\Sigma_n / \Sigma_{n-1}| - 1 \\ &= \delta(X', R') + |H_{00}|(k_n + 1)(m_n + 1) \\ &= \delta(X, R) \end{aligned}$$

gilt $S \cong S_{(X,R)}$. □

Die normierten Präsentationen bestimmen also genau die endlichen Halbgruppen, und jede endliche Halbgruppe besitzt sogar eine solche normierte Präsentation, bei der jedes Element durch genau ein normiertes Wort dargestellt wird. Dieser Fall ist unter algorithmischen Gesichtspunkten besonders interessant, weil sich dann sämtliche Rechenoperationen mit den Elementen der Halbgruppe auf das Bestimmen der zugehörigen normierten Wörter zurückführen lassen. Einer normierten Präsentation (X, R) kann man es jedoch im allgemeinen nicht ansehen, ob $S_{(X,R)}$ die Ordnung $\delta(X, R)$ hat. Einige Kriterien ergeben sich aus Sätzen über Idealerweiterungen von Halbgruppen [z. B. 9]. Algorithmisch läßt sich diese Frage „modulo Gruppen" z. B. mit dem Programm aus [5] lösen.

Es ist noch — insbesondere hinsichtlich der im weiteren zu behandelnden algorithmischen Fragen — zu bemerken, daß in 1.6 vorausgesetzt wird, daß von der Präsentation (X, R) nicht nur die Normiertheit, sondern auch die Partition in die X_i und E_i bekannt ist. Dies ist wegen des folgenden Satzes wichtig:

1.8. Satz. Es ist unentscheidbar, ob eine endliche Halbgruppenpräsentation normiert ist. Setzt man ein Orakel zur Entscheidung der Frage „definiert eine endliche Halbgruppenpräsentation eine endliche Gruppe?" voraus, so wird die Normiertheit für endliche Halbgruppenpräsentationen entscheidbar.

Beweis. Bekanntlich ist die Endlichkeit präsentierter Halbgruppen oder Gruppen unentscheidbar. Sei also (X', R') eine beliebige endliche Gruppenpräsentation. Durch Hinzunahme der Inversen erhält man in kanonischer Weise eine Halbgruppenpräsentation derselben Gruppe. Mit dem Beweis von 1.7 konstruiert man daraus eine Präsentation (X, R) dieser Gruppe, die genau dann normiert ist, wenn die Gruppe endlich ist. Damit ist die Normiertheit unentscheidbar. Setzt man jedoch ein Orakel für die genannte Frage voraus, so kann man die Normiertheit einer Präsentation folgendermaßen entscheiden: Für jede Partition von X gemäß 1.1 (2) prüfe man

(a) 1.1 (3b) bis (3g),
(b) ob die übrigen Relationen sich zu Mengen S_i über den E_i^+ aufteilen lassen,
(c) ob die (E_i, S_i) endliche Gruppen präsentieren,
(d) ob das durch (a) gegebene $e^i \in E_i^+$ Einselement der entsprechenden Gruppe ist.

Davon sind (a) und (b) formal zu entscheiden, (c) erhält man vom Orakel, (d) ist entscheidbar (z. B. mit [5]), wenn (c) bejaht wird. □

Selbstverständlich kann man das Orakel von 1.8 durch geeignete formale Voraussetzungen über die Präsentationen der Gruppen vermeiden. Auf diese Frage kommen wir im Abschnitt 3 zurück.

2. Der Multiplikationsalgorithmus

Sei $(X, R) \in \mathfrak{R}$ zusammen mit der Partition von X gemäß 1.1 gegeben, und sei $\delta(X, R) = |S_{(X,R)}|$. Wir formulieren „modulo Gruppen" einen Algorithmus, der zu zwei normierten Wörtern $u, v \in X^+$ das zu ihrem Produkt uv $\varrho_{(X,R)}$-äquivalente normierte Wort berechnet.

Wegen 1.2 kann man „modulo Gruppen" die zu den p_{ij}^i, $q_{x,y}$, r^i $\varrho_{(X,R)}$-äquivalenten normierten Wörter berechnen. Indem man in R die p_{ij}^i, $q_{x,y}$, r^i durch die entsprechenden normierten ersetzt, erhält man — „modulo Gruppen" effektiv — eine normierte Präsentation für $S_{(X,R)}$, in der die rechten Seiten zu 1.1 (3e—g) normiert sind. Mann kann also, und dies soll im folgenden geschehen, o. B. d. A. voraussetzen, daß R selbst schon diese Gestalt hat. Durch diese theoretisch irrelevante Forderung wird die Lösung der obigen Aufgabe sehr vereinfacht.

Zu $w \in X^+$ sei $\bar{w}$ das $\varrho_{(X,R)}$-äquivalente normierte Wort und $\tau(w)$ der Index i mit $\bar{w} \in X_i^+$. Für $|X_{\tau(w)}| = 1$ ist somit $\bar{w} = c^{\tau(w)}$; andernfalls hat $\bar{w}$ die Form

$$a_{\lambda(w)}^{\tau(w)} g_{w_1}^{\tau(w)} \dots g_{w_{\mu(w)}}^{\tau(w)} b_{\nu(w)}^{\tau(w)}$$

mit $g_{w_i}^{\tau(w)} \in E_{\tau(w)}$.

Die Beschreibung des Multiplikationsalgorithmus MULT erfolgt in einer leicht programmierbaren und (hoffentlich) aus sich verständlichen Weise. Er besteht neben wenigen elementaren Anweisungen aus zahlreichen Aufrufen von Unterprogrammen und Verteilersprüngen auf Marken, die in der Form

⟨Name⟩[⟨Parameterliste⟩] (⟨Argumentenliste⟩)

bzw.

⟨Name⟩[⟨Parameterliste⟩]

geschrieben werden. Die zunächst wohl künstlich anmutende Unterscheidung zwischen Parametern und Argumenten dient dazu, Programmverzweigungen, die nach Kenntnis der Präsentation unabhängig von den zu multiplizierenden Elementen feststehen, und solche, die von den jeweils zu multiplizierenden Elementen abhängen, zu trennen. Damit bereiten wir die spätere zweistufige Programmrealisierung von MULT vor, bei der ähnlich [2, 3, 6, 7, 8] aus (X, R) in einem Vorbereitungsschritt das eigentliche Multiplikationsprogramm erst berechnet wird. Aus diesem Grunde verzichten wir auch auf formale Vereinfachungen und Zusammenfassungen, die für diese Realisierung nur hinderlich wären. Es folgt die Definition des Algorithmus:

MULT (u, v): Voraussetzung: u, v normiert.
Wirkung: $\overline{uv}$ berechnen.
VERTEILERSPRUNG AUF $M[\tau(u), \tau(v)]$.

$M[i,j]$: Für $i,j=0(1)n$.
Fall 1: $|X_i|=1$, $i=j$.
RÜCKSPRUNG MIT r^i.
Fall 2: $|X_i|=|X_j|=1$, $i\neq j$.
RÜCKSPRUNG MIT q_{c^i,c^j}.
Fall 3: $|X_i|=1\neq|X_j|$.
$v:=C[i,j](v)$
RÜCKSPRUNG MIT v.
Fall 4: $|X_i|\neq 1$.
$v:=B[i,j](\nu(u), v)$
$v:=E[i](g^i_{u_{\mu(u)}}, v)$
$\vdots$
$v:=E[i](g^i_{u_1}, v)$
$v:=A[i](\lambda(u), v)$
RÜCKSPRUNG MIT v.

$B[i,j](l, v)$: Für $i,j=0(1)n$ mit $|X_i|\neq 1$.
Voraussetzung: v normiert, $v\in X_j^+$.
Wirkung: $\overline{b^i_l v}$ berechnen.
Fall 1: $|X_j|=1$.
RÜCKSPRUNG MIT $q_{b^i_l,c^j}$.
Fall 2: $|X_j|\neq 1$, $i=j$.
IST $l=\lambda(v)=0$?
WENN JA: RÜCKSPRUNG MIT v
SONST: VERTEILERSPRUNG AUF $BA[i,l,j,\lambda(v)]$.
Fall 3: $|X_j|\neq 1$, $i\neq j$.
VERTEILERSPRUNG AUF $BA[i,l,j,\lambda(v)]$.

$BA[i,l,j,k]$: Für $i,j=0(1)n$ mit $|X_i|\neq 1\neq|X_j|$ und für $l=0(1)m_i$, $k=0(1)k_j$ und $(l,k)\neq(0,0)$ bei $i=j$.
Fall 1: $i=j$, $x=p^i_{lk}\in a^i_0 E_i^+ b^i_0$.
$g:=GF[i, g^i_{x_{\mu(x)}}](g^i_{v_1}\cdots g^i_{v_{\mu(v)}})$
$g:=GF[i, g^i_{x_{\mu(x)-1}}](g)$
$\vdots$
$g:=GF[i, g^i_{x_1}](g)$
RÜCKSPRUNG MIT $a^i_0 g b^i_{\nu(v)}$.
Fall 2: $i=j$, $x=p^i_{lk}\notin a^i_0 E_i^+ b^i_0$ oder $i\neq j$, $x=q_{b^i_l,a^j_k}$.
Fall 2a: $|X_{\tau(x)}|=1$.
$v:=CS[\tau(x),j](g^j_{v_1}\cdots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$
RÜCKSPRUNG MIT v.
Fall 2b: $|X_{\tau(x)}|\neq 1$.
$v:=BS[\tau(x), \nu(x), j](g^j_{v_1}\cdots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$
$v:=EF[\tau(x), g^{\tau(x)}_{x_{\mu(x)}}](v)$
$\vdots$

$v := EF[\tau(x), g_{x_1}^{\tau(x)}](v)$
$v := AF[\tau(x), \lambda(x)](v)$
RÜCKSPRUNG MIT v.

$BS[i, l, j](v)$: Für $i, j = 0(1)n$, $|X_i| \neq 1 \neq |X_j|$, $l = 0(1)m_i$.
Voraussetzung: $v = g_{v_1}^j \dots g_{v_{\mu(v)}}^j b_{\nu(v)}^j$ mit $\mu(v) \geqq 1$ ist Postfix eines normierten Wortes.
Wirkung: $\overline{b_l^i v}$ berechnen.

Fall 1: $i = j$, $l = 0$.
RÜCKSPRUNG MIT $a_0^i v$.
Fall 2: $i = j$, $l \neq 0$, $x = p_{l0}^i \in a_0^i E_i^+ b_0^i$.
$g := GF[i, g_{x_{\mu(x)}}^i](g_{v_1}^i \dots g_{v_{\mu(v)}}^i)$
$g := GF[i, g_{x_{\mu(x)-1}}^i](g)$
$\vdots$
$g := GF[i, g_{x_1}^i](g)$
RÜCKSPRUNG MIT $a_0^i g b_{\nu(v)}^i$.
Fall 3: $i = j$, $l \neq 0$, $x = p_{l0}^i \notin a_0^i E_i^+ b_0^i$.
Fall 3a: $|X_{\tau(x)}| = 1$.
$v := CS[\tau(x), j](v)$
RÜCKSPRUNG MIT v.
Fall 3b: $|X_{\tau(x)}| \neq 1$.
$v := BS[\tau(x), \nu(x), j](v)$
$v := EF[\tau(x), g_{x_{\mu(x)}}^{\tau(x)}](v)$
$\vdots$
$v := EF[\tau(x), g_{x_1}^{\tau(x)}](v)$
$v := AF[\tau(x), \lambda(x)](v)$
RÜCKSPRUNG MIT v.
Fall 4: $i \neq j$
VERTEILERSPRUNG AUF $BSS[i, l, j, g_{v_1}^j]$.

$BSS[i, l, j, g]$: Für $i, j = 0(1)n$, $|X_i| \neq 1 \neq |X_j|$, $i \neq j$, $l = 0(1)m_i$, $g \in E_j$.
Sei $x = q_{b_l^i, g}$.
Fall 1: $|X_{\tau(x)}| = 1$.
IST $\mu(v) = 1$?
WENN JA: RÜCKSPRUNG MIT $q_{c^{\tau(x)}, b_{\nu(v)}^j}$
SONST: $v := CS[\tau(x), j](g_{v_2}^j \dots g_{v_{\mu(v)}}^j b_{\nu(v)}^j)$
RÜCKSPRUNG MIT v.
Fall 2: $|X_{\tau(x)}| \neq 1$.
IST $\mu(v) = 1$?
WENN JA: $v := BB[\tau(x), \nu(x), j](\nu(v))$; WEITER BEI $*$
SONST: $v := BS[\tau(x), \nu(x), j](g_{v_2}^j \dots g_{v_{\mu(v)}}^j b_{\nu(v)}^j)$
$*$ $v := EF[\tau(x), g_{x_{\mu(x)}}^{\tau(x)}](v)$
$\vdots$
$v := EF[\tau(x), g_{x_1}^{\tau(x)}](v)$
$v := AF[\tau(x), \lambda(x)](v)$
RÜCKSPRUNG MIT v.

$BB[i, l, j](k)$: Für $i, j=0(1)n$, $|X_i| \neq 1 \neq |X_j|$, $l=0(1)m_i$.
Wirkung: $\overline{b_l^i b_k^j}$ berechnen.
Fall 1: $i=j$, $x=p_{l0}^i$, $|X_{\tau(x)}|=1$.
RÜCKSPRUNG MIT $q_{c^{\tau(x)}, b_k^j}$.
Fall 2: $i=j$, $x=p_{l0}^i$, $|X_{\tau(x)}| \neq 1$.
$v:=BB[\tau(x), \nu(x), j](k)$
$v:=EF[\tau(x), g_{x_{\mu(x)}}^{\tau(x)}](v)$
$\vdots$
$v:=EF[\tau(x), g_{x_1}^{\tau(x)}](v)$
$v:=AF[\tau(x), \lambda(x)](v)$
RÜCKSPRUNG MIT v.
Fall 3: $i \neq j$.
RÜCKSPRUNG MIT $q_{b_l^i, b_k^j}$.

$E[i](g, v)$: Für $i=0(1)n$, $|X_i| \neq 1$.
Voraussetzung: $g \in E_i$, v normiert.
Wirkung: $\overline{gv}$ berechnen.
VERTEILERSPRUNG AUF $EFF[i, g]$.

$EF[i, g](v)$: Für $i=0(1)n$, $|X_i| \neq 1$, $g \in E_i$.
Voraussetzung und Wirkung wie $E[i](g, v)$.

$EFF[i, g]$:
IST $|X_{\tau(v)}|=1$?
WENN JA: RÜCKSPRUNG MIT $q_{g,v}$
SONST: VERTEILERSPRUNG AUF $EF1[i, g, \tau(v), \lambda(v)]$.

$EF1[i, g, j, l]$: Für $i, j=0(1)n$, $|X_i| \neq 1 \neq |X_j|$, $g \in E_i$, $l=0(1)k_j$.
Fall 1: $i=j$, $l=0$.
$h:=GF[i, g](g_{v_1}^i \cdots g_{v_{\mu(v)}}^i)$
RÜCKSPRUNG MIT $a_0^i h b_{\nu(v)}^i$.
Fall 2: $i=j$, $l \neq 0$, $x=p_{0l}^i \in a_0^i E_i^+ b_0^i$.
$h:=GF[i, g_{x_{\mu(x)}}^i](g_{v_1}^i \cdots g_{v_{\mu(v)}}^i)$
$h:=GF[i, g_{x_{\mu(x)-1}}^i](h)$
$\vdots$
$h:=GF[i, g_{x_1}^i](h)$
$h:=GF[i, g](h)$
RÜCKSPRUNG MIT $a_0^i h b_{\nu(v)}^i$.
Fall 3: $i=j$, $l \neq 0$, $x=p_{0l}^i \notin a_0^i E_i^+ b_0^i$, $|X_{\tau(x)}|=1$.
$v:=CS[\tau(x), j](g_{v_1}^j \cdots g_{v_{\mu(v)}}^j b_{\nu(v)}^j)$
$v:=EF[i, g](v)$
RÜCKSPRUNG MIT v.
Fall 4: $i=j$, $l \neq 0$, $x=p_{0l}^i \in a_0^i E_i^+ b_0^i$, $|X_{\tau(x)}| \neq 1$.
$v:=BS[\tau(x), \nu(x), j](g_{v_1}^j \cdots g_{v_{\mu(v)}}^j b_{\nu(v)}^j)$
$v:=EF[\tau(x), g_{x_{\mu(x)}}^{\tau(x)}](v)$
$\vdots$
$v:=EF[\tau(x), g_{x_1}^{\tau(x)}](v)$
$v:=AF[\tau(x), \lambda(x)](v)$
$v:=EF[i, g](v)$

RÜCKSPRUNG MIT v.

Fall 5: $i \neq j$, $x = q_{g, a_i^j}$, $|X_{\tau(x)}| = 1$.

$v := CS[\tau(x), j](g^j_{v_1} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

RÜCKSPRUNG MIT v.

Fall 6: $i \neq j$, $x = q_{g, a_i^j}$, $|X_{\tau(x)}| \neq 1$.

$v := BS[\tau(x), \nu(x), j](g^j_{v_1} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

$v := EF[\tau(x), g^{\nu(x)}_{x_{\mu(x)}}](v)$

$\vdots$

$v := EF[\tau(x), g^{\tau(x)}_{x_1}](v)$

$v := AF[\tau(x), \lambda(x)](v)$

RÜCKSPRUNG MIT v.

$A[i](k, v)$: Für $i = 0(1)n$, $|X_i| \neq 1$.
Voraussetzung: v normiert.
Wirkung: $\overline{a_k^i v}$ berechnen.

VERTEILERSPRUNG AUF $AFF[i, k]$.

$AF[i, k](v)$: Für $i = 0(1)n$, $|X_i| \neq 1$, $k = 0(1)k_i$.
Voraussetzung und Wirkung wie $A[i](k, v)$.

$AFF[i, k]$:

IST $|X_{\tau(v)}| = 1$?

WENN JA: RÜCKSPRUNG MIT $q_{a_k^i, v}$.

SONST: VERTEILERSPRUNG AUF $AF1[i, k, \tau(v), \lambda(v)]$.

$AF1[i, k, j, l]$: Für $i, j = 0(1)n$, $|X_i| \neq 1 \neq |X_j|$, $k = 0(1)k_i$, $l = 0(1)m_i$.

Fall 1: $i = j$, $l = 0$.

RÜCKSPRUNG MIT $a_k^i g^i_{v_1} \dots g^i_{v_{\mu(v)}} b^i_{\nu(v)}$.

Fall 2: $i = j$, $l \neq 0$, $x = p^i_{0l} \in a_0^i E_i^+ b_0^i$.

$g := GF[i, g^i_{x_{\mu(x)}}](g^i_{v_1} \dots g^i_{v_{\mu(v)}})$

$g := GF[i, g^i_{x_{\mu(x)-1}}](g)$

$\vdots$

$g := GF[i, g^i_{x_1}](g)$

RÜCKSPRUNG MIT $a_k^i g b^i_{\nu(v)}$.

Fall 3: $i = j$, $l \neq 0$, $x = p^i_{0l} \notin a_0^i E_i^+ b_0^i$, $|X_{\tau(x)}| = 1$.

$v := CS[\tau(x), j](g^j_{v_1} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

$v := AF[i, k](v)$

RÜCKSPRUNG MIT v.

Fall 4: $i = j$, $l \neq 0$, $x = p^i_{0l} \notin a_0^i E_i^+ b_0^i$, $|X_{\tau(x)}| \neq 1$.

$v := BS[\tau(x, \nu(x), j][g^j_{v_1} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

$v := EF[\tau(x), g^{\tau(x)}_{x_{\mu(x)}}](v)$

$\vdots$

$v := EF[\tau(x), g^{\tau(x)}_{x_1}](v)$

$v := AF[\tau(x), \lambda(x)](v)$

$v := AF[i, k](v)$

RÜCKSPRUNG MIT v.

Fall 5: $i \neq j$, $x = q_{a_k^i, a_i^j}$, $|X_{\tau(x)}| = 1$.

$v := CS[\tau(x), j](g^j_{v_1} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

RÜCKSPRUNG MIT v.

Fall 6: $i \neq j$, $x = q_{a_k^i, a_l^j}$, $|X_{\tau(x)}| \neq 1$.

$v := BS[\tau(x), \nu(x), j](g^j_{v_1} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

$v := EF[\tau(x), g^{\tau(x)}_{x_{\mu(x)}}](v)$

$\vdots$

$v := EF[\tau(x), g^{\tau(x)}_{x_1}](v)$

$v := AF[\tau(x), \lambda(x)](v)$

RÜCKSPRUNG MIT v.

$C[i, j](v)$: Für $i, j = 0(1)n$, $|X_i| = 1 \neq |X_j|$.
Voraussetzung: $v \in X_j^+$ normiert.
Wirkung: $\overline{c^i v}$ berechnen.

VERTEILERSPRUNG AUF $CA[i, j, \lambda(v)]$.

$CA[i, j, k]$: Für $i, j = 0(1)n$, $|X_i| = 1 \neq |X_j|$, $k = 0(1)k_j$.
Sei $x = q_{c^i, a_k^j}$.

Fall 1: $|X_{\tau(x)}| = 1$.

$v := CS[\tau(x), j](g^j_{v_1} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

RÜCKSPRUNG MIT v.

Fall 2: $|X_{\tau(x)}| \neq 1$.

$v := BS[\tau(x), \nu(x), j](g^j_{v_1} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

$v := EF[\tau(x), g^{\tau(x)}_{x_{\mu(x)}}](v)$

$\vdots$

$v := EF[\tau(x), g^{\tau(x)}_{x_1}](v)$

$v := AF[\tau(x), \lambda(x)](v)$

RÜCKSPRUNG MIT v.

$CS[i, j](v)$: Für $i, j = 0(1)n$, $|X_i| = 1 \neq |X_j|$.
Voraussetzung: $v = g^j_{v_1} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)} \in X_j^+$ mit $\mu(v) \geqq 1$ ist Postfix eines normierten Wortes.
Wirkung: $\overline{c^i v}$ berechnen.

VERTEILERSPRUNG AUF $CSS[i, j, g^j_{v_1}]$.

$CSS[i, j, g]$: Für $i, j = 0(1)n$, $|X_i| = 1 \neq |X_j|$, $g \in E_j$.
Sei $x = q_{c^i, g}$.

Fall 1: $|X_{\tau(x)}| = 1$.

IST $\mu(v) = 1$?

WENN JA: RÜCKSPRUNG MIT $q_{c^{\tau(x)}, b^j_{\nu(v)}}$.

SONST: $v := CS[\tau(x), j](g^j_{v_2} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

RÜCKSPRUNG MIT v.

Fall 2: $|X_{\tau(x)}| \neq 1$.

IST $\mu(v) = 1$?

WENN JA: $v := BB[\tau(x), \nu(x), j](\nu(v))$; WEITER BEI $*$

SONST: $v := BS[\tau(x), \nu(x), j](g^j_{v_2} \dots g^j_{v_{\mu(v)}} b^j_{\nu(v)})$

$*$ $v := EF[\tau(x), g^{\tau(x)}_{x_{\mu(x)}}](v)$

$\vdots$

$v := EF[\tau(x), g^{\tau(x)}_{x_1}](v)$

$v := AF[\tau(x), \lambda(x)](v)$

RÜCKSPRUNG MIT v.

$GF[i, g](h)$: Für $i=0(1)n$, $|X_i| \neq 1$, $g \in E_i$.
Voraussetzung: $h \in E_i^+$ in G_i normiert.
Wirkung: gh in G_i normiert berechnen.
ORAKEL (Zur Realisierung vgl. Abschnitt 3).

3. Programmierung und Einsatz von MULT

Für den beschriebenen Algorithmus MULT gilt in verstärktem Maße das für das Rechnen mit den Elementen abstrakt präsentierter Gruppen in [2, 3, 6, 7, 8] Gesagte: Durch die wiederholten Abfragen der für das Rechnen innerhalb einer Halbgruppe konstanten Relationen wird der Algorithmus extrem langsam. So bietet sich auch im vorliegenden Fall an, MULT zweistufig zu realisieren, indem alle von den jeweiligen zu multiplizierenden Elementen unabhängigen Entscheidungen in einer Vorbereitungsphase V getroffen werden. Entsprechende Programme V wurden für speziell geformte Präsentationen auflösbarer Gruppen in [2, 3, 6, 7, 8] dokumentiert. Unsere — zur Vereinfachung allerdings in LISP durchgeführte — Implementation von V für MULT basiert auf der wegen ihrer Portabilität besonders günstigen Version V_F aus [2]. Hier wie dort besteht das durch V generierte eigentliche Multiplikationsprogramm MULT$[X, R]$ aus einer Folge von Unterprogrammen, die jedes selbst wieder im wesentlichen nur aus vereinzelten Befehlen zum Holen von Konstanten usw. und aus zahlreichen Unterprogrammaufrufen bestehen — ihre jeweilige Gestalt ist für die entsprechenden Parameterwerte und Fälle durch den Algorithmus des vorigen Abschnitts vorgegeben. Die Beschleunigung von MULT$[X, R]$ gegen MULT ist im allgemeinen sehr groß, und die Erfahrungen lassen bei aller Problematik eines Vergleichs erkennen, daß MULT$[X, R]$ von der Geschwindigkeit her mit Multiplikationsprogrammen für andere Darstellungen von $S_{(X,R)}$ gut konkurrieren kann.

Der typische Einsatz von MULT ist analog den Gruppenprogrammen aus [2, 3, 6, 7, 8] folgendermaßen: Zur Lösung der Aufgabe, Eigenschaften der durch die normierte Präsentation (X, R) gegebenen Halbgruppe zu berechnen, wird zunächst mit dem Programm aus [5] nachgeprüft, ob $\delta(X, R) = |S_{(X,R)}|$ gilt. Falls nein, versucht man, die Präsentation geeignet zu modifizieren; häufig kann man dabei Zwischenergebnisse dieses Programmes ausnutzen. Falls die Bedingung erfüllt ist, wird mit V das Programm MULT$[X, R]$ generiert, welches dann vom eigentlichen Rechenprogramm (wie z. B. [4]) für die einzelnen Multiplikationen aufgerufen wird.

Die bisherigen Überlegungen erfolgten sämtlich unter der Voraussetzung geeigneter Orakel für das Rechnen mit den Elementen der durch die Halbgruppenpräsentationen (E_i, S_i) gegebenen Gruppen G_i. Die Realisierungsmöglichkeit und Realisierungsweise dieser Orakel $GF[i, g](h)$ hängt stark von der Form der Präsentationen (E_i, S_i) ab. So kann man als einen Extremfall etwa $E_i = G_i$ und S_i als die volle Cayleytafel von G_i wählen; dadurch erhält man mit den normierten Präsentationen sämtliche endlichen Halbgruppen, und die Orakel werden triviale Programme; der erforderliche Speicheraufwand wird jedoch schon für mäßig große Halbgruppen unerträglich groß. Günstiger wird es z. B., wenn man voraussetzt, daß die Präsentationen (E_i, S_i) als Halbgruppenpräsentationen von Gruppen durch die üblichen Umformungen aus „AG-Systemen" [3] (*pc*-presentation [2]) (E_i', S_i') hervorgehen. Als Orakel $GF[i, g](h)$ kann man dann die entsprechenden Teile des mit V_F aus (E_i', S_i') gewonnen eigentlichen Multiplikationsprogrammes für G_i verwenden. Die

Beschränkung auf die Klasse $\mathfrak{N}_{pc}$ der normierten Präsentationen, in denen die (E_i, S_i) kanonisch aus *pc*-Präsentationen gewonnen werden, hat noch weitere Vorteile: Ersetzt man 1.1 (3a) durch die *pc*-Forderungen aus [2] und die Umformungsregeln, so erhält man aus 1.1 für $\mathfrak{N}_{pc}$ eine formale, entscheidbare Charakterisierung. Unter der Voraussetzung $(X, R) \in \mathfrak{N}_{pc}$ kann man daher auf die Angabe der Partition von X verzichten, weil diese für $\delta(X, R) = |S_{(X,R)}|$ bis auf die Indizierung eindeutig mit 1.8 bestimmt werden kann. Algebraisch bedeutet die Beschränkung auf $\mathfrak{N}_{pc}$, daß nur und genau die endlichen Halbgruppen mit ausschließlich auflösbaren Untergruppen betrachtet werden, eine praktisch auch noch bei mäßig großen Halbgruppen fast unbedeutende Einschränkung.

INSTITUT FÜR THEORETISCHE INFORMATIK
TECHNISCHE HOCHSCHULE DARMSTADT
MAGDALENEN STR. 11
D—6100 DARMSTADT

Literatur

[1] CLIFFORD, A. H., und G. B. PRESTON, The algebraic theory of semigroups, I. (2. Aufl.) Providence, Rh. I., 1964.
[2] FELSCH, V., A machine independent implementation of a collection algorithm for the multiplication of group elements, *Proceedings SYMSAC 76*, New York, 1976.
[3] JÜRGENSEN, H., Calculation with the elements of a finite group given by generators and defining relations, In: *J. Leech* (Hrsg.), Computational problems in abstract algebra, *Proceedings of a conference* (Oxford, 1967) Oxford, 1970.
[4] JÜRGENSEN, H. und P. WICK, Bestimmung der Unterhalbgruppenverbände für zwei Klassen endlicher Halbgruppen, *Computing*, v. 11, 1973, pp. 337—351.
[5] JÜRGENSEN, H., Transformationendarstellungen endlicher abstrakt präsentierter Halbgruppen, *Bericht*, No. 7605, Inst. f. Informatik u. Prakt. Math., Universität Kiel, 1976.
[6] LINDENBERG, W,. Über die Darstellung von Gruppenelementen in digitalen Rechenautomaten, *Numer. Math.*, v. 4, 1962, pp. 151—153.
[7] LINDENBERG, W., Die Struktur eines Übersetzungsprogrammes zur Multiplikation von Gruppenelementen in digitalen Rechenautomaten, *Mitt. Rh.—Westf. Inst. für Instrum. Math.*, Bonn, v. 2, 1963, pp. 1—38.
[8] NEUBÜSER, J., Bestimmung der Untergruppenverbände endlicher *p*-Gruppen auf einer programmgesteuerten elektronischen Dualmaschine, *Numer. Math.*, v. 3, 1961, pp. 271—278.
[9] PETRICH, M., *Introduction to semigroups*, Columbus, Ohio, 1973.

(Eingegangen am 8. März 1977)