1 Variationen über Walther von Dyck und Dyck-Sprachen

Volker Diekert,
Universität Stuttgart, Institut für Formale Methoden der Informatik

Klaus-Jörn Lange
Universität Tübingen, Wilhelm-Schickard-Institut für Informatik

Zusammenfassung. Die Dyck-Sprachen sind ein Grundbegriff aus dem Bereich der formalen Sprachen. Ausgehend von der Person des Namensgebers werden ihre Geschichte und ihre Bedeutung in der theoretischen Informatik in diesem Überblick dargstellt.

Eine Arbeit zum Thema *Walther Ritter von Dyck* und die nach ihm benannten *Dyck-Srachen* in dieser Festschrift liegt in besonderer Weise nahe. Dies ergibt sich aufgrund von Gemeinsamkeiten zwischen Volker Claus und Walther von Dyck. Erwähnenswert ist zunächst die persönliche Ausstrahlung, die wir beim Kollegen Claus erleben und bei Dyck aus Erzählungen kennen. Wichtig ist auch ihr hochschulpolitisches Engagement für die Mathematik als wichtiger Bestandteil der universitären Ausbildung in den Ingenieurwissenschaften, und schließlich sind ihre Namen in der theoretischen Informatik präsent.

Von Dyck war der erste Rektor der heutigen Technischen Universität München, Mitbegründer des Deutschen Museums in München und ein geachteter und begeisterter Hochschullehrer. Dies zeigt sich schon in der beeindruckenden Lebensspanne, in der er aktiv wirkte. Bis fast zu seinem Lebensende hielt er Vorlesungen und erreichte fast die legendäre Zahl von 100 Semestern. Er beeinflusste in maßgeblicher Weise die mathematisch-naturwissenschaftliche Ausbildung an den technischen Hochschulen. Zu Ende des 19. und Anfang des 20. Jahrhunderts keimte eine Diskussion auf, dass die deutsche Ingenieurausbildung, im Vergleich etwa zur amerikanischen, zu grundlagenorientiert sei. Insbesondere die Mathematikausbildung galt als Ballast. Es gab Vorschläge, sie auf das Vermitteln von Kochrezepten zu reduzieren. Dadurch könne man in den Anwendungen schnelle Erfolge feiern.

Glücklicherweise erkannten maßgebliche Persönlichkeiten wie von Dyck den Irrtum und die Kurzsichtigkeit dieser Haltung und aufgrund ihrer Stellung gelang es ihnen, dieser Fehleinschätzung erfolgreich zu begegnen. So konnte eine grundlagenorientierte Ausbildung für Ingenieure an technischen Hochschulen nachhaltig etabliert werden, die bis heute zur Qualitätssicherung der deutschen Hochschulausbildung beiträgt. Hochschullehrer, wie Ritter von Dyck vor 100 Jahren und Herr Claus heute, haben sich stets für ein hohes wissenschaftliches an den Universitäten eingesetzt. In diesem Sinne möchten wir uns daher ab jetzt mit wissenschaftlichen Themen beschäftigen, die zum Begriff *Dyck-Sprache* einen direkten Bezug haben.

Dies führt uns auf einen Streifzug entlang mathematischer Begriffe, die in der theoretischen Informatik präsent sind. Der Ausgangspunkt ist die Dycksche Sichtweise einer freien Gruppe [4]. Sie wird zunächst im Sinne von Cayley abstrakt als eine Menge von Operatoren vorgestellt (sowie ihrer Inversen), ohne dass es weitere Relationen gibt. Ende des 19. Jahrhunderts war eine solche rein abstrakte Beschreibung durchaus ungewöhnlich. Auch für von Dyck erhielt die freie Gruppe erst ihre eigentliche Bedeutung, als er sie als Transformationsgruppe geometrisch realisieren konnte. Dies erläutert er in seiner Arbeit *Gruppentheoretische Studien* aus dem Jahr 1883, die in den Mathematischen Annalen erschien [5] und allein schon durch die ornamentartigen Bilder besticht. Interessant ist hier auch, wie er das Rechnen in einer freien Gruppe auf die Kombinatorik von Wörtern ohne negative Exponenten zurückführt. Für die Darstellung einer freien Gruppe mit 2 Erzeugenden verwendet er nicht die uns vertrauten vier Erzeugenden a, b, a^{-1}, b^{-1}, sondern nur drei Buchstaben a, b, c und betrachtet die Wörter hierüber, die kein Vorkommen von *abc*, *bca* oder *cab* haben. In der heutigen Sprechweise definiert dies drei Löschregeln, die *abc*, *bca* oder *cab* jeweils durch das leere Wort ersetzen. Es ergibt sich ein konvergentes Semi-Thue System, welches es erlaubt das Wortproblem mit Hilfe von deterministischen Kellerautomaten in linearer Zeit zu entscheiden. So führt uns Dycks Betrachtungsweise direkt zu den Kellerautomaten, wie sie heute in Vorlesungen vorkommen. Kellerautomaten charakterisieren genau die kontext-freien Sprachen. Das Kellerprinzip findet seinen Ursprung bei der Übersetzung von Formeln. 1955 erfanden Samelson und Bauer das „Kellerprinzip“, auf dem die effiziente Übersetzung von Formeln und anderen klammerartigen Ausdrücken beruht [13]. Sie entwarfen eine Maschine, die dieses Prinzip realisierte, für die sie 1957 ein deutsches und ein US-amerikanisches Patent erhielten. Eine Klammerstruktur findet sich auch in den freien Gruppen, die Dyck untersuchte. Die tiefere Bedeutung der freien Gruppen für die Theorie kontext-freier Sprachen wurde erstmals durch die Arbeiten von Chomsky und Schützenberger klar.

Der Begriff einer Dyck-Sprache scheint 1961/62 von Schützenberger geprägt worden zu sein. Diese Ansicht wird zumindest auch von Chomsky geteilt [1]. In der Arbeit [14] spricht Schützenberger von einem *D-event* und in [15] führt er explizit eine Menge D^* als Dyck Menge ein. Schließlich schreiben Chomsky und Schützenberger in ihrer 1963[1] erschienenen Arbeit [2]: „We define *the Dyck language* ...“ Ihre Definition liefert für $n = 2$ die Sprache D_2^* der Wörter w über dem Alphabet $\{a, \bar{a}, b, \bar{b}\}$, die interpretiert in der freien Gruppe über $\{a, b\}$ das neutrale Element darstellen. Analog zu D_2^* definiert man D_n^* für höhere n. Diese festen Sprachen D_n^* bilden sozusagen das Fundament, auf dem das Gebäude der kontextfreien Sprachen aufgebaut werden kann. In der oben erwähnten Arbeit findet sich der Satz von Chomsky und Schützenberger: *Jede kontext-freie Sprache ist homomorphes Bild des Durchschnitts einer regulären Sprache mit einer Dycksprache.*

Die Theorie kontext-freier Sprachen ist also eng mit der durch von Dyck mit initiierten Theorie freier Gruppen verbunden. Diese Verwandtschaft ist auch von der gruppentheoretischen Seite her untersucht worden. Wir können für jede Gruppe H mit einer endlichen erzeugenden Menge Σ die Menge der Wörter betrachten, die sich in der Gruppe zu Eins auswerten. Bilden diese eine kontext-freie Sprache, so hängt diese Eigenschaft nur von der Gruppe H und nicht von der Wahl von Σ ab. Es ist also sinnvoll, von *kontext-freien Gruppen* zu sprechen. Zu jeder solchen Gruppe finden wir eine reduzierte kontext-freie Grammatik $G = (V, \Sigma, P, S)$ und nach [9, 17] natürliche Isomorphismen zwischen der explizit endlich dargestellten Gruppe $F(V \cup \Sigma)/P$ und der Gruppe H. Die Resultate von Hotz und Valkema (oder ein direkter Beweis unter Verwendung des *uvwxy-Theorems*) zeigen, dass kontext-freie Gruppen endlich dargestellt sind. Hieraus folgt nach Arbeiten von Muller und Schupp [11] sowie Dunwoody [3], dass kontext-freie Gruppen stets eine freie Untergruppe von endlichem Index haben. Das ergibt das überraschende Resultat, dass die Klasse der kontext-freien Gruppen genau mit der Klasse der endlich erzeugten virtuell freien Gruppen übereinstimmt. Dies ist ein tiefsinniges mathematisches Ergebnis welches die freien Gruppen im Sinne von Dyck direkt mit moderner Theorie formaler Sprachen verbindet.

Wir wenden uns einigen Aspekten der Syntaxanalyse zu. Unter Linguisten ist es scheinbar beliebt, Menschen mit Fernrohren zu betrachten. Dies liegt an dem folgenden Satz, den wir zunächst in seinem englischen Original wiedergeben, da die Mehrdeutigkeit hier besser zur Geltung kommt.

I saw the man on the hill with the telescope.

In diesem Satz gibt es vier Akteure (Ich, Mann, Berg, Fernrohr) und fünf unterschiedliche Bedeutungen. Die Fragen, wo sich der Mann oder wo sich das Fernrohr

[1] Die Ergebnisse dieser Arbeit wurden schon 1961 vorgestellt.

befindet, lassen sich nicht aus dem Satz heraus klären. Die 5 Bedeutungen ergeben sich aus den 5 möglichen Klammerungen:

(((Ich sah den Mann) auf dem Berg) mit dem Fernrohr)

((Ich sah (den Mann auf dem Berg)) mit dem Fernrohr)

(((Ich sah den Mann) (auf dem Berg mit dem Fernrohr)

(Ich sah ((den Mann auf dem Berg) mit dem Fernrohr))

(Ich sah (den Mann (auf dem Berg mit dem Fernrohr)))

Damit sind wir bei der Frage nach der Anzahl möglicher Klammerungen und gelangen so zu den Catalanschen Zahlen, also einer exponentiellen Mehrdeutigkeit; und es ergibt sich die Notwendigkeit, Klammerstrukturen zu betrachten. Bezeichnen wir mit $a,b,c,\ldots$ eine Menge verschiedener öffnender Klammern und seien $\bar{a},\bar{b},\bar{c},\ldots$ die entsprechenden schließenden Klammern, so können wir die Sprache der wohlgeformten Klammerausdrücke bilden. So ist etwa $aab\bar{b}\bar{a}\bar{a}$ wohlgeformt, aber weder $\bar{a}ab\bar{b}\bar{a}a$ noch $aa\bar{b}b\bar{a}\bar{a}$ noch $aab\bar{a}\bar{b}\bar{a}$ sind wohlgeformt.

Die Menge der wohlgeformten Klammerausdrücke über k Klammerpaaren wird mit D_k bezeichnet und ebenfalls Dyck-Sprache genannt. Die Wörter aus D_k sind genau die Wörter aus der symmetrischen Dyck-Sprache D_k^*, in denen jeder Präfix mindestens soviel öffnende wie schließende Klammern enthält. Die Dyck-Sprachen D_k sind deterministisch kontext-frei, und Greibach hat 1973 gezeigt, dass mit Hilfe der Dyck-Sprache D_2 (also mit Hilfe von 2 Klammerpaaren) eine schwierigste kontext-freie Sprache definiert werden kann. Die folgende (ziemlich schwer zu lesende) Definition findet sich so im Original [7]:

$$L_0 = \left\{ x_1 c y_1 c z_1 d \cdots d x_n c y_n c z_n d \,\middle|\, \begin{array}{l} n \geq 1,\ y_1 \cdots y_n \in \not{c} D_2, \\ x_i, z_i \in \{a,\bar{a},b,\bar{b},c,\not{c}\}^* \end{array} \right\}$$

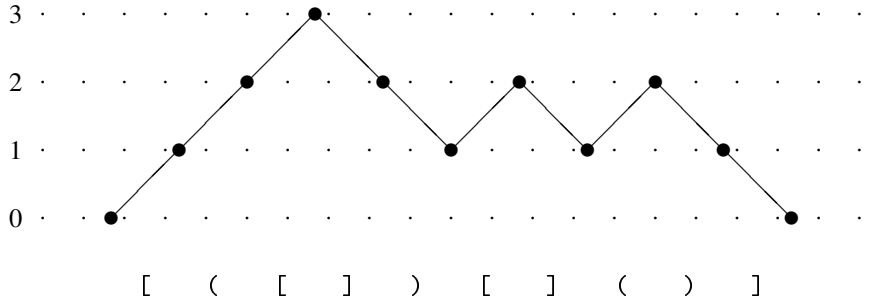
Greibach beweist, dass sich jede kontext-freie Sprache (ohne das leere Wort) als invers homomorphes Bild von L_0 darstellen lässt. Hinter dieser technischen Aussage verbirgt sich ein bemerkenswertes Resultat. Stellen wir uns vor, wir haben eine kontext-freie Sprache L definiert und suchen jetzt ein Verfahren, welches auf Eingabe eines Wortes w entscheidet, ob w zu L gehört. Nach Greibach finden wir für L einen Homomorphismus h mit $L = h^{-1}(L_0)$. Es reicht also, $h(w) \in \{a,\bar{a},b,\bar{b},c,d,\not{c}\}^*$ zu berechnen, was sehr einfach ist, und dann zu testen, ob $h(w) \in L_0$ gilt. Hierfür benötigen wir ein einziges festes Verfahren für L_0, welches wir zum fortwährenden Gebrauch in einer Programmbibliothek niederlegen. Das eine Verfahren kann also für alle Sprachen genutzt werden. Würde man etwa für

die eine Sprache L_0 ein quadratisches Worterkennungsverfahren finden, so würden sich alle kontext-freien Sprachen in quadratischer Zeit erkennen lassen. Die Sprache L_0 übernimmt in der Klasse der kontext-freien Sprachen damit eine analoge Rolle, wie sie etwa das Erfüllbarkeitsproblem Boolescher Formeln SAT für die prominente Klasse NP einnimmt[2]. Man beachte, dass Greibach das Konzept eines vollständigen Problems für eine Sprachklasse entwickelte, bevor die Theorie der NP-Vollständigkeit Verbreitung fand.

In einem letzten Abschnitt wollen wir erklären, dass ein besseres Verständnis der symmetrischen Dyck-Sprache D_2^* und der Klammersprache D_2 zu einem substantiellen Fortschritt in der Komplexitätstheorie führen könnte. Dies ist zwar nur eine Vermutung, aber in in der Komplexitätstheorie stehen allgemein den wenigen erzielten Ergebnissen ohnehin eine Unzahl von Vermutungen gegenüber. So vermutet man, dass NP-vollständige Probleme wie SAT nicht in P liegen, also nicht in polynomialer Zeit entscheidbar sind. Nach derzeitigem Kenntnisstand wäre es aber noch möglich, dass alle Probleme aus NP sehr einfach zu lösen sind. Vielleicht ist also das Erfüllbarkeitsproblem SAT beispielsweise in kubischer Zeit lösbar und damit nicht viel schwieriger als die Syntax-Analyse für die Sprache L_0, die durch die schnelle Matrixmultiplikation nach Strassen [16] in besserer als kubischer Zeit durchgeführt werden kann.

Einige der wenigen bekannten nicht-trivialen unteren Schranken für NP-vollständige Probleme folgt aus einem Resultat von Furst, Saxe, Sipser [6], nach dem nicht alle Probleme aus NP durch boolesche Schaltkreise konstanter Schaltungstiefe (und mit beliebigem großem Eingangsgrad der vorhandenen Gatter) erkannt werden können. Interessanterweise hat man diese Trennung nicht etwa mittels eines scheinbar schwierigen Problems wie etwa SAT gezeigt, sondern mit einem eigentlich ganz einfachen Problem. Man betrachtete die Sprache „Parity“, die aus den Binärwörtern besteht, die eine gerade Zahl von Einsen enthalten. Die Sprache Parity liegt in der Komplexitätsklasse TC^0 (sogar in ACC^0). Hinter der kryptischen Buchstabenkombination TC^0 steht, dass sich „Parity“ durch Schaltkreise konstanter Tiefe erkennen lässt, die zusätzlich „Threshold“-Gatter haben dürfen. Solche Gatter haben beliebigen Eingangsgrad und geben eine Eins aus, wenn an mehr als der Hälfte der Eingänge eine Eins anliegt. Diese Klasse scheint auf den ersten Blick wenig mächtig, aber es gibt bis dato keinen Beweis für $\mathrm{TC}^0 \neq \mathrm{NP}$. Die Klasse TC^0 kann immerhin die Dyck-Sprache D_2 erkennen: Die Grundidee ist, dass nach Rutishauser ein Dyck-Wort durch ein Klammergebirge dargestellt werden kann:

[2] NP steht für „Nicht-deterministisch Polynomial“.



Rutishausers Klammergebirge

Der TC^0-Algorithmus arbeitet (nach einer Beobachtung von Lynch [10]) wie folgt: Betrachte ein Wort w der Länge n. Jede Position i mit $1 \leq i \leq n$ entspricht einer Klammer. Für jede Position muss die Zahl der öffnenden Klammern bis zu dieser Position größer oder gleich der Zahl der schließenden Klammern sein. Dieser Test ist mit Threshold-Gattern leicht zu realisieren. Für je zwei Positionen i und j mit $1 \leq i < j \leq n$ verlangen wir: wenn die Zahl der öffnenden Klammern zwischen i und j gleich der Zahl der schließenden Klammern zwischen i und j ist, dann müssen i eine öffnende und j eine schließende Klammer vom selben Typ sein. Auch diese Forderung ist mit Threshold-Gattern leicht zu realisieren. Der so konstruierte Schaltkreis hat quadratische Größe in n und läßt sich in konstanter Schaltungstiefe realisieren.

Interessant ist, dass bisher kein TC^0-Algorithmus für die symmetrische Dyck-Sprache D_2^* bekannt ist. Auch diese Sprache kann durch Schaltkreise polynomieller Größe erkannt werden. In der Sprache der Komplexitätstheorie ist sie jedoch schwierig für *Nick's Class* der ersten Stufe NC^1 [12]. Diese Klasse beschreibt Probleme, die sich durch Schaltkreise logarithmischer Tiefe lösen lassen, wobei nur boolesche Gatter von beschränktem Fan-In erlaubt sind. Dies ist eine formale Fassung von Problemen, für die man auf eine effiziente Parallelisierung hoffen kann. Während viele Standardvermutungen die Verschiedenheit der natürlich definierten Komplexitätsklassen vorhersagen, ist das Bild für TC^0 und NC^1 weniger klar. Es gibt durchaus Gründe, die für die Gleichheit von TC^0 und NC^1 sprechen. So lassen sich die arithmetischen Operationen in TC^0 berechnen. Dieses ist nicht unmittelbar einsichtig, und erst vor Kurzem konnte mit recht aufwändigen Verfahren gezeigt werden, dass sogar die Division (uniform) in TC^0 durchführbar ist [8]. Würde es uns nun gelingen, das Wortproblem für freie Gruppen mit zwei Erzeugenden in TC^0 zu lösen, so wäre $TC^0 = NC^1$.

Dies Vorhaben muss scheitern, wenn $\mathrm{TC}^0 \neq \mathrm{NC}^1$ oder wenn $D_2^* \notin \mathrm{NC}^1$ gilt. Vielleicht sollten besser beide Möglichkeiten in Betracht gezogen werden. Dann verfügen wir jedoch mit D_2^* über einen möglicherweise geeigneten Kandidaten, für den – durch kombinatorische Methoden – die Trennung von TC^0 und NP gelingen könnte.

Wie auch immer die Antwort zu diesen Fragen aussehen mag, als von Dyck erstmals freie Gruppen untersucht hat, wird er kaum geahnt haben, dass Konzepte, die mit seinem Namen verbunden sind, heute noch in dieser Weise aktuell sind. Vielleicht ahnte er es doch, schließlich war ihm die Ausbildung von Ingenieuren über fast 100 Semester ein Anliegen.

Literaturverzeichnis

[1] N. Chomsky. Persönliche Mitteilung, 2004.

[2] N. Chomsky and M. P. Schützenberger. The algebraic theory of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*, pages 118–161. North-Holland, 1963.

[3] M. J. Dunwoody. The accessibility of finitely presented groups. *Inv. Math.*, 81:449–457, 1985.

[4] W. von Dyck. Ueber Aufstellung und Untersuchung von Gruppe und Irrationalität regulärer Riemann'scher Flächen. *Math. Annalen*, XVII:473–509, 1881.

[5] W. von Dyck. Gruppentheoretische Studien. *Math. Annalen*, XX:1–44, 1883.

[6] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Math. Systems Theory*, 17:13–27, 1984.

[7] S. Greibach. The hardest context-free language. *SIAM Journal on Computing*, 2:304–310, 1973.

[8] W. Hesse, E. Allender, and D. A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences*, 65:695–716, 2002.

[9] G. Hotz. Eine neue Invariante für kontextfreie Sprachen. *Theoretical Computer Science*, 11:107–116, 1980.

[10] N. Lynch. Log space recognition and translation of parentheses languages. *Journal of the Association for Computing Machinery*, 24:583–590, 1977.

[11] D. E. Muller and Schupp P. E. Groups, the theory of ends, and context-free languages. *Journal of Computer and System Sciences*, 26:295–310, 1983.

[12] D. Robinson. Parallel Algorithms for Group Word Problems. Dissertation, University of California, San Diego, 1993.

[13] K. Samelson and F. L. Bauer. Sequentielle Formelübersetzung. *Elektronische Rechenanlagen*, 1:176–182, 1959.

[14] M. P. Schützenberger. Certain elementary families of automata. In *Proc. Symp. on Math. Theory of Automata*, pages 139–153. Polytechnic Institute of Brooklyn, 1962a.

[15] M. P. Schützenberger. On context-free languages and pushdown automata. *Information and Control*, 6(3):246–264, 1963.

[16] V. Strassen. Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten. *Num. Math.*, 20:238–251, 1973.

[17] E. Valkema. On some relations between formal languages and groups. In *Proc. Categorical and Algebraic Methods in Computer Science and System Theory*, pages 116–123, 1978.