Merge und EFS

Urs Egli

1 Elementare Formale Systeme

Mit dem Ausdruck *merge* gebraucht Chomsky für das Zusammenfügen von Wörtern und Syntagmen zu Sätzen die Metapher des Verschmelzens. Damit reiht er sich ein in eine Jahrtausende alte Tradition der metaphorischen Erfassung grammatikalischer Begriffe.

Seit der griechischen Antike gibt es nämlich metaphorische Unschreibungen für das Zusammenfügen von Wörtern und Syntagmen zu Sätzen. Grammatikalische Begriffe mussten damals erst geschaffen werden und die griechischen Philosophen bedienten sich dabei ganz verschiedener Metaphern: Der berühmte Arzt und Sprachforscher Hippokrates nannte die Satzteile *arthra*, lat. *articula* ‚Glieder' und assoziierte so mit den Satzteilen das Zusammenspiel der Glieder des menschlichen Körpers. Der Stoiker Chrysipp nannte die Zusammenstellung der Satzteile *Syntax*, ein Wort, das auch die Aufstellung einer Schlachtordnung bedeutet. Lateinisch adaptiert hieß dies *constructio*, Konstruktion. Eine poetischere Metapher fand Platon: Er nannte die Verbindung der Satzteile *Symploke*, das ‚Zusammenflechten', was bei den Stoikern eher für einen Spezialfall, die Konjunktion, verwendet wurde, was wir hier *conjoin* nennen werden.

Symplekein ‚zusammenflechten', *syntattein* ‚zusammenstellen', oder im 20. Jahrhundert *concatenation* ‚Verkettung, Zusammenfügung' und *merge* ‚verschmelzen' sind also inhaltlich verwandte grammatikalische Begriffe.

Platon ist auch der Schöpfer der ersten und bekanntesten Syntaxregel des Abendlandes, die er im Dialog *Sophistes* wie folgt formulierte:

(1) Ein *logos* (ein Satz) ist geflochten aus *onoma* (einem nominalen Teil) und *rhema* (einem verbalenTeil).

Platon nannte das Beispiel *Theätet sitzt* aus *Theätet* und *sitzt*. Ein analoger englischer Satz wird uns als Beispiel dienen: *John walks* aus *John* und *walks*.

Unmetaphorisch kann man diese Regel durch einen Wenn-Dann-Satz formulieren:

(2) Wenn *John* ein nominaler Teil und *walks* ein verbaler Teil ist, dann ist die Konstruktion *John walks* ein Satz.

Eine Theorie solcher Wenn-Dann-Sätze ist in den Erörterungen von Raymond Smullyan zu den Elementaren Formalen Systemen von 1961 enthalten. Ein solches System ist eine Gruppe von Wenn-Dann-Sätzen, bei denen der Wenn-Teil auch fehlen kann. Diese Systeme nennen wir hier auch Grammatiken. Das Format dieser Grammatiken ist auch im Syntaxprogrammieren in der Programmiersprache Prolog verwendet. 1997 hat Annius Viktor Groenink eine praktisch identische Theorie unter dem Namen Literal Movement Grammar veröffentlicht, die leider viel zu wenig beachtet worden ist. Der Name spielt auf die Tatsache an, dass eine neue Theorie der Bewegungs-Transformationen geschaffen werden sollte, welche nicht Bäume bewegt, sondern Wortfolgen. Dieser Ursprung der Bezeichnung soll aber hier nicht verfolgt werden.

Die Regeln, die weitgehend Bestandteilen von Prolog-Programmen entsprechen, können auf folgende Art notiert werden:

(3) Satz(xy) <- Nominaler Teil(x) Verbaler Teil(y)

2 Minimalismus und EFS

Man kann aus der heutigen Diskussion zwei Versionen von Merge (nach Zwart, 2015) herausarbeiten, die beide in EFS wiedergegeben werden können.

(4) Conception of Merge (1)
   a. There is a Numeration N and a construction set $N*$
   b. There is an Operation (‚Merge‘) such that
      (i) Merge takes two elements x, y from $N*$ that belong to the simplex/complex sets A and B, and
      (ii) combines x and y to the complex xy as an element of the complex set C.

(5) Conception of Merge (2)
   a. There is a Numeration N and a construction set $N*$
   b. There is an Operation (‚Merge‘) such that
      (i) Merge takes two elements x, y from $N*$ such that the first belongs to the simplex/complex set A, and
      (ii) combines x and y to the complex xy as an element of the complex set C.

Man vergleiche die Formulierung von Merge in Trotzke & Zwart (2014), Zwart (2015), und die Besprechung von Merge in Sauerland (2015). Dabei ist der Versuch der Deutung, den ich erklären will, dadurch gekennzeichnet, dass nicht wie bei Zwart das sogenannte „Set Merge“ verwendet wird, auch nicht das „List Merge“ (vgl. Langendoen, 2003), sondern etwas, was ich „String Merge“ nennen möchte, bei dem es um die Konkatenation von Strings geht, die bestimmten Kategorien angehören und einer neuen Kategorie zugewiesen werden.

3 Arikawa et al., Miyano et al. und Groenink

Die Smullyanschen Systeme sind durch zwei Entwicklungen erweitert worden, die unabhängig sind, aber erstaunlich gut zusammenpassen: Arikawa et al. (1992) haben in einer Zusatzüberlegung gezeigt, wie die regulären, die kontextfreien und die kontextsensitiven Grammatiken auf eine interessante Weise in das Smullyansche Format integriert werden können. Groenink hat 1997 durch eine Neuerfindung und Erweiterung des Formats die Sprachen und

Relationen in seinem Format der SLMG, für welche gerade in polynomial vielen Schritten, $O(n^k)$ mit beliebig grossem k, für eine Eingabe der Grösse n entschieden werden kann, ob ein String zur Sprache oder Relation gehört oder nicht. Schon Miyano et al. (2000) und Ikeda & Arimura (1997) haben eine polynomiale Version der EFS definiert, die Hereditary Elementary Formal Systems.

Es gibt Mechanismen von Groenink und anderen, die zeigen, wie man die Adjunktion in TAG in Formalismen wie der SLMG darstellen kann (Kallmeyer, 2010). Diese nicht ganz trivialen Reduktionsalgorithmen kann man auch auf die Einordnung von Pair Merge als Erfassung der Adjunktion im Minimalismus anwenden, wenn man sie denn zulassen will. Außerdem kann man die Rekonstruktion des Minimalismus durch Stabler als lineare SLMG formulieren (vgl. Kallmeyer, 2010: 126, 164 und 169). Nebenbei wird bei diesen Reduktionen auch Tree Substitution und Tree Adjunction auf String-Manipulation umgestellt.

Um das Format von Arikawa et al. (1992) der length-bounded EFS gut mit dem Groeninkschen Format der SLMG vergleichen zu können, genügt ein Hinweis, dass es eine Form der SLMG gibt, die wir SLMG-λ nennen, das ohne Einsetzung von λ auskommt. Zur Herstellung einer solchen Normalform werden alle möglichen Kombinationen von Einsetzungen von λ in den Regeln für die Variablen vorgenommen und daraus eine Normalform hergestellt, die nur mit Einsetzungen für Variablen ohne λ auskommt, wobei S(λ) hinzugenommen wird, wenn das Leere Wort in der Sprache ist, was man effektiv testen kann mit der alten Grammatik. Außerdem kann man die Prädikationen mit λ loswerden, indem man mit dem Underlining Algorithm von Ebbinghaus & Flum (2001) die reinen Einsetzungen mit λ fortbringt, die Regeln, in denen man Bedingungen nicht fortbringt, weglässt und Argumente mit λ durch Einführung von neuen Prädikaten mit kleinerer Stellenzahl beseitigt.

Interessant ist wie erwähnt die Darstellung der CSG als EFS in Arikawa et al. (1992). Arikawa et al. (1992) geben eine Charakterisierung der CSG als eine sogenannte length-bounded EFS und der CFG als eine simple and regular EFS in ihrer eigenen Terminologie. Die Charakterisierung durch Groenink (1997) ist äquivalent und parallelisiert die Chomsky-Normalform. Die allgemeine Klasse der linearen SMLG ist eine Wiedergabe der mild kontext-sensitiven Grammatiken.

Es gibt eine constituency based bottom-up derivational grammar für die Kontextfreie Grammatik, die als EFS formuliert ist. Für sie gibt es auch eine Chomsky-NF, welche die Binarität von „Merge" herstellt. Das Rewriting-Format innerhalb der Chomsky-Hierarchie ist eine voll äquivalente Darstellung und die Formate der bottom-up derivation und der top-down derivation sind dabei äquivalent. Wenn man allerdings die Semantik mitberücksichtigen will, ist das Format bottom-up vorzuziehen. Ausgeführt ist die Semantik schon in Chiang (2012). Elementar sind die beiden Formate im Kapitel 3 von Kowalski (1979) auf den Seiten 49-74 erklärt.

4 Anhang über die Regeltypen

$R = A(t_1, \ldots, tp) \leftarrow B_1(s_1^1, \ldots, s_{p1}^1), \ldots, B_m(s_1^m, \ldots, s_{pm}^m)$

Conditions on Rules

Rules of LMG (according to Groenink, 1997)

- R is bottom-up linear iff no variable occurs more than once in the t_i.
- R is top-down linear iff no variable occurs more than once in the s_j^i.
- R is bottom-up non-erasing iff every variable occurring in one of the s_j^i occurs in one of the t_l
- R is top-down non-erasing iff every variable occurring in one of the t_l occurs in one of the s_j^i
- R is non-combinatorial iff every s_j^i is but a single variable
- R is simple (a simple literal movement grammar - SLMG - rule) iff it is bottom-up linear, bottom-up non-erasing and non-combinatorial.
- R is a 1-SLMG) rule iff it is a SLMG rule with one-Place predicates only.
- R is a linear SLMG rule iff it is bottom-up and top-down linear, bottom-up and top down non-erasing and non-combinatorial.
- R is a linear 1-SLMG rule iff it linear and simple with 1-place predicates.

Rules of EFS (according to Arikawa et al., 1992)

- R is variable-bounded (recursively enumerable EFS rule) iff the variables occurring in the predications of the right-hand side also occur in the predication of the left hand side.
- R is length-bounded (a CSG rule) iff
  1. the length of the predications of the right-hand side is less than or equal to the length of the left-hand side, and
  2. the number of occurrences of each variable in the predication on the right-hand side is equal to or less than the number of occurrences of this variable on the predications on the right hand side.
- R is simple iff it is length-bounded, contains unary predicates only, and all terms on the right-hand side are single variables, which are mutually distinct.
- R is regular (a CFG rule) iff it is simple and every variable in the head of R occurs at most once.

5 Anmerkung

Mit dem Aufsatz möchte ich Josef Bayer zu seinem 65. Geburtstag gratulieren. Ich danke ihm für die Einladung zu einem Vortrag über diese Themen und die Diskussion bei dieser Gelegenheit, auch mit Eleonore Brandner und Andreas Trotzke, die mir eine Verbesserung der Erfassung der Themen erlaubt hat. Ich danke wie immer meiner Frau Renata Egli-Gerber für viele Gespräche.

Literatur

Arikawa, S., T. Shinohara & A. Yamamoto. 1992. Learning elementary formal systems. *Theoretical Computer Science* 95. 97–113.

Chiang, D. 2012. *Grammars for language and genes: Theoretical and empirical investigations.* Berlin: Springer.

Ebbinghaus, H. & J. Flum. 2001. Mathematics of logic programming. In D. M. Gabbay & F. Guenthner (Hrsg.), *Handbook of philosophical logic*, 313–370. Dordrecht: Kluwer.

Groenink, A. V. 1997. *Surface without structure: Word order and tractability issues in natural language analysis.* University of Utrecht Diss.

Ikeda, D. & H. Arimura. 1997. The computational complexity of hereditary elementary formal systems. In *Proceedings of the 3rd International Conference on Developments in Language Theory. Thessaloniki, Greece, July 20-23, 1997*, 223–235.

Kallmeyer, L. 2010. *Parsing beyond context-free grammars.* Berlin: Springer.

Kowalski, R. 1979. *Logic for problem solving.* New York: Elsevier North-Holland.

Langendoen, D. T. 2003. Merge. In A. Carnie, M. Willie & H. Harley (Hrsg.), *Formal approaches to functional phenomena*, 307–318. Amsterdam: John Benjamins.

Miyano, S., A. Shinohara & T. Shinohara. 2000. Polynomial time learning of EFS. *New Generation Computing* 18. (Als technical report 1995 erschienen.), 217–242.

Sauerland, U. 2015. Against complexity parameters. In A. Trotzke & J. Bayer (Hrsg.), *Syntactic complexity across interfaces*, 9–24. Berlin: Mouton de Gruyter.

Smullyan, R. M. o.D. *Theory of formal systems.* Princeton: Princeton University Press.

Trotzke, A. & J.-W. Zwart. 2014. The complexity of narrow syntax: Minimalism, representational economy, and simplest Merge. In F. J. Newmeyer & L. B. Preston (Hrsg.), *Measuring grammatical complexity*, 128–147. Oxford: Oxford University Press.

Zwart, J.-W. 2015. Top-down derivation, recursion, and the model of grammar. In A. Trotzke & J. Bayer (Hrsg.), *Syntactic complexity across interfaces*, 25–42. Berlin: Mouton de Gruyter.