Paul Koop, M.A.
Josefstraße 152
0241/558369
52080 AACHEN

Über die Unentscheidbarkeit der GTG CHOMSKY's: Eine erste Zusammenfassung über die grundsätzlichen Fragen zur Entscheidbarkeit der GTG[1]

Zusammenfassung:
Der Text geht der Frage nach, ob eine Entscheidung über die Widerspruchsfreiheit und Vollständigkeit der GTG CHOMSKYs möglich ist. Es wird eine Zeichenkette beschrieben die einen Satz definiert, dessen Semantik über seine Transformation aus der Tiefenstruktur in die Oberflächenstruktur spricht. Es wird gezeigt, daß die Semantik dieses Satzes nicht zufällig mit seiner Syntax korreliert, weil er ohne Informationsverlust in eine zahlentheoretische Aussage transformiert werden kann. Diese zahlentheoretische Aussage ist äquivalent zu GÖDELs erstem Unvollständigkeitssatz. Dann wird gezeigt, daß der Satz endweder zu der Annahme führt, daß die GTG in sich widersprüchlich ist oder widerspruchsfrei aber unvollständig. Damit ist dann bewiesen, daß die GTG in dem einen, wie in dem anderen Fall Sprache nicht vollständig modellieren kann.

Problemstellung:

An anderer Stelle habe ich am Beispiel von Sprechhandeln gezeigt, daß es mehr Wortfunktionen gibt, als durch Algorithmen berechnet werden können (KOOP 1994, S.41f). Ich konnte weiter zeigen, daß es Sprechakte gibt, die nicht durch GTG-ableitbare Zeichenketten definiert werden können (a.a.O., S.45). Ich konnte zeigen, daß es sich dabei um Sprechakte handelt, die sich auf sich selbst beziehen.

---

[1] Die Frage der Entscheidbarkeit der GTG ist für den algorithmischen Deskriptionismus von entscheidender Bedeutung. Den algorithmischen Deskriptionismus habe ich an anderer Stelle beschrieben (**KOOP1994**). Der algorithmische Deskriptionismus geht davon aus, daß jede sozialwissenschaftliche Theorie letztlich auf ein Verständnis der zugrundeliegenden sozialen Strukturen angewiesen ist. Soziale Strukturen werden als regelgeleitet im Sinne von Algorithmen verstanden. Die Menge der möglichen Strukturen ist daher von abzählbarer Mächtigkeit. Die Menge der Sprechakte (**SEARLE, AUSTIN**) aber wird als eine überabzählbare Mannigfaltigkeit verstanden. In Anlehnung an **WEIZENBAUM** wird daher davon ausgegangen, daß Strukturhypothesen algorithmisch definiert werden müssen (vergleiche auch **KLÜVER**), während Strukturanalysen auf ein der objektiven Hermeneutik äquivalentes Verfahren angewiesen sind (vergleiche auch **REICHERTZ, SCHROER**). Der vorliegende Beitrag soll zeigen, daß eine GTG die Sprachkompetenz eines Sprechers nicht vollständig modellieren kann.

An dieser Stelle möchte ich zeigen, daß es möglich ist, einen Zeichenkette zu definieren, deren Oberflächenstruktur endweder mit einer in sich widersprüchlichen GTG aus ihrer Tiefenstruktur abgeleitet werden kann, oder aber eine in sich widerspruchsfreie GTG diese Zeichenkette nicht aus einer Tiefenstruktur ableiten kann.

Vorgehensweise:

Ich gehe intuitiv von der Idee aus, daß ein Satz konstruiert werden kann, der über seine eigene Ableitbarkeit aus einer Tiefenstruktur spricht und werde diese Aussage in eine Zeichenkette übersetzen, die von ihrem Transformationsmarker spricht, von dem ich dann zeigen werde, daß dieser Transformationsmarker entweder in sich widersprüchlich ist oder aber nicht existiert. Diese Intuition ist von den oben erwähnten selbstbezüglichen Sprechakten inspiriert. Zunächst muß das Problem gelöst werden, einen Satz über seine tiefenstruktur sprechen zu lassen. Dann muß ein Weg gefunden werden, eine Zeichenkette über ihren Transformationsmarker sprechen zu lassen. Zuletzt muß eine Zeichenkette beschrieben werden, die über ihren Transformationsmarker eine solche Aussage macht, die zu dem Widerspruch führt, daß sie keine GTG-Zeichenkette sein kann.

Problemdiskussion:

Ich werde die GTG an dieser Stelle nicht vorstellen. Das würde nicht nur den Rahmen des Beitrages sprengen, sondern auch den Blick für die wesentliche Idee verstellen. Ich möchte mich statt dessen auf zwei CHOMSKY-Zitate beschränken, die so treffend sind, daß sie nicht nur keiner Ergänzung bedürfen, sondern auch so allgemein sind, daß sie die wesentliche Grundidee einer GTG, in welcher aktuellen Spielart auch immer, treffen. Es handelt sich um zwei, zugegeben lange Zitate, die die Begriffe Phrasemarker (P-Marker) und Transformationsmarker (T-Marker) definieren[2]:

"Der natürliche Mechanismus zur Generierung von P-Markern [...] ist ein System von Ersetzungsregeln (rewriting rules). Eine Ersetzungsregel hat die Form: (4) A->Z/X_Y wobei X und Y (möglicherweise leere) Ketten von Symbolen sind, A ein einfaches Kategoriensymbol und Z eine nicht-leere Kette von Symbolen ist. Diese Regel wird so interpretiert: Die Kategorie A wird als die Kette Z realisiert, wenn sie in der Umgebung vorkommt, die zur Linken aus X und zur Rechten aus Y besteht. Die Anwendung der Ersetzungsregel (4) auf eine Kette ...XAY... überführt diese in die Kette ...XZY... Innerhalb einer gegebenen Grammatik definieren wir: Eine Folge von Ketten ist eine W-Derivation von V, falls W die erste V die Letzte Kette in dieser Folge ist und jede Kette der Folge aus der verhergehenden durch Anwendung einer Ersetzungsregel [...] abgeleitet ist. Ist V eine Kette von Formativen, so sagen wir, die betreffende W- Derivation von V ist abgeschlossen (terminated). Wir nennen V eine Endkette (terminal string), wenn es eine #S#-Derivation von #V# gibt, wobei S ein ausgezeichnetes Anfangssymbol der Grammatik ist [...] und # ein Grenzsymbol [...]. (CHOMSKY1983,S.91f)

"Streng formal wäre ein T-Marker zu repräsentieren als eine Menge von Ketten in einem Alphabet, das aus Basis-P-Markern und Transformationen als seinen Elementen besteht, ebenso, wie P-Marker formal zu repräsentieren sind als Menge von Ketten in einem Alphabet, das aus Endsymbolen, Kategoriensymbolen [...]besteht. (a.a.O.,S.168)

Die Semantik der dann so erzeugten Endketten ist eine Funktion ihrer Erzeugung mit Hilfe

---

[2] Ich habe es aufgegeben, mich mit der grundsätzlich ablehnenden Kritik an algorithmischen Sprachmodellen auseinanderzusetzen. Vorgetragen wird sie nämlich mit dem Anspruch auf "päpstliche Unfehlbarkeit" verbunden mit einer ebenso "päpstlichen Ahnungslosigkeit" über das, wovon man da eigentlich spricht. SCHEFFER zum Beispiel schreibt: "Ich bin nicht allzusehr überzeugt von den Glücksversprechungen der >>Künstlichen-Intelligenz<<, wonach alles a-logaritmisierbar sei bzw. wonach alles nichts sei, was nicht a-logarithmisierbar ist" (SCHEFFER1992, S.55). Das ist so überzeugend wie die Behauptung eines Fischverächters, Killersprotten seien bissig. Ein Trost aber bleibt. Denn wäre eine vollständig explizierbare GTG der deutschen Sprache möglich, sie könnte den zitierten Satz nicht als Zeichenkette erzeugen, weil ihr die Kreativität von Sprechakten fehlt. Ironischerweise ist das vielleicht genau das, was SCHEFFER sagen wollte. Genau das aber ist Thema dieses Aufsatzes.

der Regeln. So hat die in Jöhdel[3] geschriebene Zeichenkette

"Keine Katze mag Waldi"

die Semantik

Keine Katze mag Waldi.

Die Endkette läßt sich ohne Informationsverlust in die Prädikaten- und Quantorenlogische Formel

$$\neg \exists x \; (katze(x) \wedge mögen(x, waldi))$$

deren Semantik ebenfalls

Keine Katze mag Waldi

ist, transformieren. Die Semantik dieser Formel aber ist durch die Syntax vollständig definiert (vgl. GREWENDORF,u.a.1991,S.359) Gesucht ist ein Satz, desssen Semantik über sich selbst spricht. Zunächst scheint es unproblematisch zu sein, einen Satz zu finden, der über sich selbst spricht. Der Satz:

Dieser Satz besteht nicht aus sieben Worten.

etwa macht eine falsche Aussage über sich selbst. Diese Art von Selbstbezüglichkeit

---

[3] Zeichenketten wie **"Kurt Gödel"** gehören einer Objektsprache an, über die in der Metasprache Deutsch gesprochen wird. Die Objektsprache selbst ist natürlich nicht die Sprache Deutsch. Um es uns einfacher zu machen, benennen wir die Objektsprache Jöhdel (Im Aachener Platt wird "Gödel" als "Jöhdel" ausgesprochen). Alle Zeichenketten in diesem Beitrag sind in Jöhdel geschrieben, alle metasprachlichen Ausdrücke in Deutsch.
Beispiel:
Die Semantik der Jöhdel- Zeichenkette:

"Chomsky ist ein Linguist."

wird metasprachlich in Deutsch definiert mit dem Satz:

Chomsky ist ein Linguist.

$$\exists x{:}\,(chomsky(x) \wedge linguist(x))$$

Und mit der metasprachlichen deutschen Beschreibung:
"Der Unvollständigkeitssatz ist von Name des Mathematikers**"**
meine ich die Jöhdel- Zeichenkette
"Der Unvollständigkeitssatz ist von Kurt Gödel".

aber bringt uns nicht weiter. Die Selbstbezüglichkeit liegt in diesem Satz nämlich auf der semantischen Ebene. Wir müssen eine Zeichenkette definieren, die über GTG-ableitbare Sätze spricht, ohne ihre Semantik zu berücksichtigen. Eine solche Zeichenkette ist zum Beispiel:

> **"Der Satz "Der Satz besteht nicht aus sieben Worten." besteht nicht aus sieben Worten."**

Sie macht eine falsche Ausgage über die Zeichenkette des Satzes

> Der Satz besteht nicht aus sieben Worten,

den sie zitiert. Die semantische Ebene aber ist in der GTG vollständig durch die Syntax von Zeichenketten definiert. Wir benötigen also eine Zeichenkette, die über ihre GTG-Ableitbarkeit spricht. Ein solcher Satz ist die Zeichenkette:

> **"Der Satz "Der Satz kann nicht durch einen Phrasemarker definiert werden." kann nicht durch einen Phrasemarker definiert werden."**

Das bringt uns einen Schritt weiter, ist aber noch nicht genau. Phrasemarker sind letztlich Darstellungen von endlichen Ableitungsketten, die mit dem Startzeichen #S# beginnen und mit dem Satz enden. Solche Ableitungen sind endliche Zeichenketten über einem endlichen Alphabet von Nonterminal- und Terminalzeichen. Jede solche Zeichenkette aber läßt sich auch als Zahl, dargestellt in einem Stellwertsystem mit der Basis, die der Mächtigkeit des verwendeten Alphabetes entspricht, interpretieren. Man sieht sofort, daß diese Interpretation auch auf den Transformationsmarker zutrifft, der die Überführung eines Satzes aus seiner

Tiefenstruktur in seine Oberflächenstruktur beschreibt. Es existiert dann zu jeder Überführung eines Satzes aus seiner Tiefenstruktur in seine Oberflächenstruktur eine Codezahl, die für diese Transformation steht und aus der umgekehrt diese Transformation rekonstruiert werden kann. Eine solche Codezahl nenne ich ab sofort **Transformationsmarkerzahl**. Eine solche umkehrbar eindeutige Abbildung ist durch eine berechenbare Wortfunktion repräsentiert und damit in einen Algorithmus überführbar. Wenn aber die gesamte Transformation codierbar ist, dann ist es auch die Zeichenkette deren Semantik für den Satz selbst steht, weil die letzten Ziffern der Transformationsmarkerzahl die Zeichenkette definieren codieren. Die Codezahl des Satzes möcht ich ab sofort **Satzzahl** nennen. Verwende ich nun diese neuen Begriffe so wird aus der Zeichenkette:

> **"Der Satz "Dieser Satz besteht aus sechs Worten." kann nicht durch einen Phrasemarker definiert werden."**

der die Zeichenkette:

> **"Der Satz Sechsundfünfzigtausend-siebenhundertneunundachzig kann nicht durch einen Phrasemarker definiert werden."**

wenn die Zeichenkette **"Der Satz besteht aus sechs Worten"** die Satzzahl 56789 hat. Es ist dann der spezielle Fall denkbar, daß ein deutscher Satz seine eigene Codezahl erwähnt. In solchen Sätzen möchte ich diese Satzzahl dann Selbstcodierung nennen. Auch die Satzzahl ist durch eine berechenbare Wortfunktion definiert und damit algorithmisierbar[4]. Da die vorgenannten Codezahlen berechenbar sind, und es möglich ist, einen GTG-Parseralgorithmus zu definieren[5], ist es dann auch möglich, einen Algorithmus zu definieren, der eine Satzzahl und eine Transformationsmarkerzahl als Eingabe nimmt und überprüft, ob die Transformationsmarkerzahl einen Transformationsmarker repräsentiert, der eine Zeichenkette definiert deren Semantik den Satz definiert, der durch die Satzzahl definiert ist.

Bis zu diesem Punkt ist die Argumentation unproblematisch. Wir werden aber nun eine Jöhdel- Zeichenkette mit der Semantik definieren, daß keine Transformationsmarkerzahl eine Satzzahl definiert. Das ist natürlich eine falsche Aussage. Es wird sich aber zeigen, daß im speziellen Fall der Selbstcodierung diese Aussage einfach deshalb richtig ist, weil eine widerspruchsfreie GTG nicht dazu in der Lage ist, einen Transformationsmarker und eine Zeichenkette für einen Satz zu finden, der über seine Selbstcodierung sprich.
Nun können wir folgende Zeichenkette

[4] Ein Algorithums Satzzahl ist rekursiv zu definieren. Im Falle der Selbstcodierung nähmlich muß die Zeichenkette für die Satzzahl in den Satz für die Teilkette "Satzzahl" eingesetzt und dann die aktuelle Satzzahl berechnet werden.

```
UNTER_PROGRAMM satzzahl_code(satz,teilkette):satzzahl;
 ANFANG
  berechne das zahlwort für die satzzahl;
  WENN die teilkette nicht leer ist und im satz vorkommt
  DANN
   ANFANG
    ersetzte an jeder stelle des satzes die teilkette durch das
    zahlwort der satzzahl;
    berechne die satzzahl von satzzahl_code(satz,teilkette);
   ENDE
  gib die satzzahl aus;
 ENDE;
```

[5] vergleiche KOOP1993

definieren:[6]

> "Keine Transformationsmarkerzahl definiert die Satzzahl"

Weiter oben haben wir gesehen, daß es unproblematisch ist, zu jedem Satz eine Satzzahl zu berechnen. Die Zeichenkette des letzten Beispielsatzes besitzt also auf jedenfall eine Satzzahl. Nun Beschreiben wir folgende Jöhdel- Zeichenkette:

> "Keine Transformationsmarkerzahl definiert die Selbstcodierung"[7]

Die Semantik dieser Jöhdel-Zeichenkette ist:

> Keine Transformationsmarkerzahl
> definiert die Selbstcodierung.

Wenn der Satz wahr ist, dann existiert keine solche Transformationsmarkerzahl. Genau das aber behauptet der Satz.

Das aber ist äquivalent zu der Aussage, das diese Zeichenkette nicht durch eine GTG

---

[6] Im folgenden benutze ich folgende GTG:

GTG := (*Nonterminale*, *Terminale*, *Regeln*, "☺")

Nonterminale :=
{☺, ☻, ♥, ♦, ♣, ♠}

Terminale:=
{*Transformationsmarkerzahl*, *Selbstcodierung*, *die*, *definiert*, *Keine*}

Regeln :=
{☺ → ☻♥,
☻ → ♦♠,
♥ → ♦☻,
♦ → *Keine*/*die*,
♣ → *definiert*,
♠ → *Transformationsmarkerzahl*/ *Selbstcodierung*/ {*Menge aller Zahlworte*}}

[7] Der Leser beachte bitte, daß in dieser Beschreibung einer Zeichenkette nicht die Teilkette "**Selbstcodierung**" steht sondern gesagt wird, daß an einer bestimmten Stelle der beschriebenen Zeichenkette die Teilkette für die Selbstcodierung einzusetzen ist (vgl. Fußnote 3).

repräsentiert ist[8]. Wenn der Satz falsch ist, dann existiert eine solche Transformationsmarkerzahl. Das aber ist äquivalent zu der Aussage, das der auf der GTG beruhende Parser und der auf der GTG beruhende Codierungsalgorithmus miteinander in Widerspruch stehen. Was nichts anderes bedeutet, als das die GTG dann mit sich selbst in Widerspruch steht. Dieses Ergebnis ist genau dann auch keine blose semantische Spielerei ohne Bezug zur Syntax, die ja durch die Transformationsmarkerzahl repräsentiert ist. Der Oberflächenstruktur des Satzes, für den diese Zeichenkette steht, läßt sich zwar ein Phrasemarker zuordnen, die Zeichenkette aber, die für den Satz steht, auf den der Satz dieser Zeichenkette verweist, läßt sich nicht durch einen Transformationsmarker aus der Tiefenstruktur erzeugen, weil dieser Transformationsmarker eine unendlich lange

---

[8] Ein Beispiel mag das veranschaulichen. Gegeben sei eine wie in Fußnote 5 definierte widerspruchsfreie GTG, erweitert um Transformationsregeln. Ein Parser, der überprüfen könnte, ob eine Transformationsmarkerzahl eine Satzzahl definiert hätte etwa folgenden Algorithmus:

```
DEFINITION_PROGRAMM Parser (eingabe,ausgabe);
 DEFINITION_UNTERPROGRAMM transformation(t_markerzahl):ausgabe;
 ANFANG
  WENN  die t_markerzahl keinen t_marker bestimmt gib die t_markerzahl zurück
  SONST gib den t_marker zurück;
 ENDE;
 DEFINITION_UNTERPROGRAMM satz_bestimmung (satzzahl):ausgabe;
 ANFANG
  WENN der satzzahl kein Phrasemarker zugeordnet werden kann
  DANN gib die satzzahl zurück
  SONST
   ANFANG
    bestimme den durch die satzzahl gegebenen satz;
    WENN der satz eine weitere_satzzahl zitiert(*<<<<<<<<<*)
    DANN ersetze die weitere_satzzahl durch      (*SCHLEIFE bei Selbstcodierung*)
         satz_bestimmung (weitere_satzzahl);(*<<<<<<<<<<<<*)
    gib den satz aus;
  ENDE;
 ENDE;
ANFANG
 LESE (eingabe,transformationsmarkerzahl);
 WENN die transformation(t_markerzahl) keinen transformationsmarker bestimmt
 DANN gib die transformationsmarkerzahl und die Zeichenkette
       "Keine Transformationsmarkerzahl" zurück
 SONST
  ANFANG
   LESE (eingabe, satzzahl);
   WENN die satz_bestimmung(satzzahl) keinen satz bestimmt
   DANN gib die satzzahl und die zeichenkette "Keine Satzzahl" zurück
   SONST
   ANFANG
    WENN der transformationsmarker den satz definiert
    DANN gib die zeichenkette "ja" zurück
    SONST gib die zeichenkette "nein" zurück;
   ENDE;
  ENDE;
ENDE.
```

Ein "Bleistifttest", zeigt daß das Programm bei Eingabe der Selbstcodierung im Unterprogramm satz_bestimmung in eine Endlosschleife käme und keine Ausgabe erzeugen würde.

Ableitung besitzt und deshalb keine Transformationsmarkerzahl hat. Das sieht man sofort, wenn man erkennt, das die Zeichenkette des Beispielsatzes ohne Informationsverlust in die PEANO-Arithmetik übersetzt werden kann. Das ist deshalb möglich, weil die Zeichenkette Aussagen über zwei natürliche Zahlen macht, nähmlich über die Transformationsmarkerzahl und über die Selbstcodierung. In der PEANO-Notation lauter der Satz:[9]

$r \leftrightarrow \neg \exists t{:}\exists s{:}[P(t,s) \wedge C(E(r),s)]$

*

dabei bedeutet im einzelnen:

*

P := *Parserfunktion*
C := *Satzzahlcodierer*
E := *Funktion, die allgemeine Satzvariabl*
durchdenSatz selbst ersetzt
t := *transformationsmarkerkettenzahl*
s := *selbstbezüglicheSatzzahl*
r := *derSatzselbst*

Es ist kein Zufall, daß dieser Satz äquivalent auf den ersten Unvollständigkeitssatz GÖDELs abgebildet werden kann, denn GÖDELs Unvollständigkeitssatz macht ja eine äquivalente Aussage über formale Systeme, die der PEANO-Arithmetik äquivalent sind. Aber Vorsicht: Während GÖDELs Unvollständigkeitssatz zeigt, das die PEANO-Arithmetik widerspruchsfrei aber unvollständig ist, zeigt die Beschreibung der Zeichenkette:

"Keine Transformationsmarkerzahl definiert die Selbstcodierung"

[9] vergleiche HOFSTADTER1987,S.470ff, GREWENDORF, u.a.1991,S.430ff

das die GTG möglicherweise entweder widersprüchlich ist und deshalb alles Mögliche definieren kann oder aber widerspruchsfrei ist, aber nicht die Tiefenstruktur der Zeichenkette jeden Satzes definieren kann. In beiden Fällen aber wäre die GTG eine unvollkommene Beschreibung von Sprechhandeln.

Literatur:

CHOMSKY,N.1983: Aspekte der Syntax-Theorie, Frankfurt 1983(3)

GREWENDORF,G., HAMM,F., STERNEFELD,W. 1991: Sprachliches Wissen, Frankfurt 1991(5)

GÖDEL,K.1931: Über formal unendscheidbare Sätze der Principia Mathematica und verwandter Systeme I, in: Monatshefte für Mathematik und Physik,38(1931), S.173-198

HOFSTADTER,D.R.1987: Gödel, Escher, Bach: Ein endloses geflochtenes Band, Stuttgart 1987(10)

KOOP,P.1993: "Syntaktische Analyse natürlicher Sprache unter Pascal, unveröffentlichtes Manuskript, Aachen 1993

KOOP,P.1994: "Eine Hand wäscht die andere":Bedingungen der Möglichkeit eines algorithmischen Deskriptionismus: Eine Voruntersuchung am Beispiel der Sequenzanalyse mikroökonomischer Interakte, unveröffentlichtes Manuskript, Universität Essen 1994

SCHEFFER,N.1992: Interpretation und Lebensroman, Frankfurt 1992