

Quenya Syntax 1: Verbal-Phrasen

Informationslinguistik, SS 2004

Claudia Henghuber <henghuber@gmx.de>

Hubert Feyrer <hubert@feyrer.de>

Inhalt

- Satzaufbau und Wortstellung
- Beispielsätze
- Quenya-Grammatik
- Parser – Theorie und Umsetzung in Prolog

Satzaufbau und Wortstellung

Inhalt Satzaufbau

- Valenzgrammatik
- Phrasenstrukturgrammatik
- Generative Transformationsgrammatik
- Satzstellung

Valenzgrammatik: Begründer

Begründer der Valenzgrammatik (VG):
Lucien Tesnière

1959: "Elements de syntaxe structurale"

1953: "Esquisse de syntaxe structurale"

VG: Allgemeines #1

- Verb bildet das *strukturelle Zentrum* des Satzes
 - Wenn eine syntaktische Struktur gleichbleibend in einer Reihe von Sätzen wiederkehrt, spricht man von einem *Satzbauplan*.
- ⇒ Satzbauplan wird durch das Prädikat begründet
- ⇒ Subjekt wird auf gleiche Stufe wie die vorhandenen Objekte gestellt

VG: Allgemeines #2

- Subjekt und Objekte bezeichnet Tesnière als "*actants*" (Aktanten, Ergänzungen)
- Darstellung des Satzes in einem *Stemma*

VG: Stemma

Beispiel für ein Stemma:

“Hans gibt Maria Bücher.”

Verbalknoten:

gibt

Ergaenzungen:

Hans

Maria

Buecher

VG: Valenz

- Ein Stemma ist ein Graph, dessen Knoten hier durch ein Wort belegt ist, und dessen nach unten gerichtete Kanten als Abhängigkeitsrelationen, sogenannte *Konnexionen*, aufgefasst werden.
- Tesniere vergleicht die Fähigkeit der Verben eine Anzahl und Art von Ergänzungen zu sich zu nehmen, mit der Wertigkeit eines Atoms und nennt sie *Valenz*.

VG: Dependenzgrammatik

- Die Grammatik, in der Abhängigkeitsbeziehungen dargestellt werden, nennt man *Dependenzgrammatik*. Die Untersuchung nach Valenz ist ein Teilbereich der Dependenzgrammatik.

VG: Ergänzungen

Vom Verb geforderte Ergänzungen im Deutschen:

- Nominativ-Ergänzung
- Genitiv-Ergänzung
- Dativ-Ergänzung
- Akkusativ-Ergänzung
- Präpositional-Ergänzung
- Prädikatsnomen-Ergänzung

VG: Ergänzungen & Angaben

- Obligatorische und Fakultative Ergänzungen
- *Obligatorische Ergänzungen* sind alle Satzglieder, die für einen syntaktischen Sinn notwendig sind.
- Im Gegensatz dazu stehen die *fakultativen Ergänzungen*, die weglassbar sind. Fakultative Ergänzungen nennt man auch *Angaben*.

VG: Verbarnten

- Wenn die Anzahl der Ergänzungen ermittelt worden ist, kann man die Wertigkeit der Vollverben festlegen.
- Vollverben sind Verben wie arbeiten, tanzen, lachen, singen usw.
- Dem gegenüber stehen Hilfsverben (z.B. werden) und Modalverben (wie sollen).

Phrasenstrukturgrammatik (PSG)

- Als Begründer der Grammatik gilt Leonard Bloomfield mit seinem Hauptwerk "Language" 1933.
- Ursache für die Entwicklung dieser Grammatik war das Interesse an den aussterbenden Indianersprachen.
- Bei der Phrasenstrukturgrammatik wird der Satz in Nominalphrasen und Verbalphrasen segmentiert.

PSG: Beispiel

- Die Einzelelemente werden nicht durch das bestimmt, was sie sind, sondern durch ihre Umgebung und Beziehung zueinander (paradigmatische und syntaktische Beziehung).

Beispiel: “Schöne blonde Mädchen haben blaue Augen.”

- “Schöne blonde Mädchen” = Nominalphrase
- “haben blaue Augen” = Verbalphrase

Generative Transformationsgrammatik

- Generative Transformationsgrammatik (=GTC)
- Der Begründer der GTC ist Noam Chomsky mit den beiden Hauptwerken “Syntactic Structures” 1957 und “Aspects of the Theory of Syntax” 1965

GTC #2: Schreibweise

- Bedient sich der Schreibweise der Mathematik und der Logik:
 - : Ersetze das links vom Pfeil Stehende durch das rechts vom Pfeil Stehende, also $x \rightarrow y + z$: Ersetze x durch y gefolgt von z

GTC #3: Symbole

Die oberste Einheit der Grammatik ist der Satz

Symbole:

S = Satz

NP = Nominalphrase

VP = Verbalphrase

N = Nomen

V = Verben

T = Terminanten (Artikel im weiteren Sinn)

GTC #4: Funktionsregeln

Die Grammatik besteht aus Formationsregeln in Form von "Ersetzungsregeln"

$$S \rightarrow NP + VP$$
$$NP \rightarrow T + N$$
$$VP \rightarrow V + NP$$

In einem zweiten Schritt werden mit Hilfe der Lexikonregeln den grammatischen Kategorien Klassen von Wörtern zugeordnet.

GTC #5: Lexikonregeln

T → the, a

N → woman, man

V → shoots

GTC #6: Ersetzungsregeln

S

NP + VP

T + N + VP

T + N + V + NP

a + N + V + NP

a + woman + V + NP

a + woman + shoots + NP

a + woman + shoots + T + N

a + woman + shoots + a + N

a + woman + shoots + a + man

Die Reihenfolge der Ersetzung ist dabei frei!

Die Satzgliedstellung im Quenya

- Im Quenya ist die Satzstellung im Allgemeinen sehr frei
- Durch die Casusendungen ist klar, welchen grammatikalischen Zusammenhang ein Satz hat, so dass einzelne Elemente im Satz herumgeschoben werden können.
- Adjektive können vor dem Substantiv stehen das sie beschreiben, oder dahinter:

“tara oron” oder “oron tara” (ein hoher Berg)

Satzgliedstellung #2

- Genitive können vor oder nach dem Substantiv stehen, auf das sie sich beziehen:
“elenion ancalima” (hellster unter den Sternen)
“quenta Silmarillion” (die Erzählung von den Silmaril)

Satzgliedstellung #3

- Das Verb na kann in der Mitte des Satzes stehen, am Ende, oder ganz entfallen:

“elda na tulca” (ein Elb ist stark)

“elda tulca na” (ein Elb ist stark)

“elda tulca” (ein Elb ist stark)

Satzgliedstellung #4

- Üblicherweise entspricht der Satzbau dem Englischen, also:
Subjekt — Prädikat — Objekt
Subjekt und Akkusativobjekt besitzen die gleiche Endung.
- In poetischer oder besonders betonter Wortstellung kann sich der Satzbau ändern:
Prädikat — Subjekt — Objekt
Das Subjekt wird aber in der Regel vor dem Objekt genannt.

Satzgliedstellung #5

Das dem Sprecher Wichtige wird nach vorne geschoben:

Et Earello Endoreнна utulien.

(Von großen Meer her bin ich nach Mittelerde gekommen.)

Endoreнна utulien et Earello.

(Nach Mittelerde bin ich vom großen Meer her gekommen.)

Utulien et Earello Endoreнна.

(Gekommen bin ich vom großen Meer her nach Mittelerde.)

Beispielsätze

Beispiel-Schema

ᖃᖅ

Quenya

Deutsch

Fauskanger-Referenz

(Kommentar)

⇒ Grammatik

Beispiel 1: Einfacher Satz

i íġ wîp

i elda MÁTA

Der Elf ISST

less-a/59

(a-Verb)

⇒ NP, Verb

Beispiel 2: Satz mit Objekt

i íġ êîĥ êĥ

i elda MÁTA massa

Der Elf ISST Brot

less-a/59

(mit Objekt)

⇒ NP, Verb, NP

Beispiel 3: Andere Wortstellung

îîî i îî

MÁTA i elda

less-a/62

ES ISST der Elf

⇒ Verb, NP

Beispiel 4: Adjektiv (Positiv)

iḥz mî êzî

isil NÁ calima

Der Mond IST hell

less-a/63f

(Copula+Adj.)

⇒ NP, Copula, Adjektiv(Positiv)

Beispiel 5: Komparativ

iŋz mî êzîzî žî ízî

isil ná calima LÁ eleni

less-a/63f

Der Mond ist heller ALS die Sterne

⇒ NP, Copula, Adjektiv(Komparativ), NP

Beispiel 6: Superlativ

iŋz mî jcerêzîz

isil ná ANcalima

less-a/63f

Der Mond ist AM hellSTEN

⇒ NP, Copula, Adjektiv(Superlativ)

Beispiel 7: Wörtliche Rede

íḡ jḡḡḡ: íḡ íḡḡ íḡḡḡ íḡḡḡ:

EQUË Elendil: ...

less-c/52

Elendil SPRACH: ...

(zeitlos)

⇒ “eque”, NP, “:”, E

Beispiel 8: Relativsatz #1

i λnâz mâi îzâ

i harma ná alta
der Schatz ist groß

less-c/76

λnâz

hirnelyë
Du hast ihn gefunden

Beispiel 8: Relativsatz #2

i Ñwê ã Ñwóç mî îçp

i harma YA hirnyë ná alta

der Schatz, DEN du gefunden hast, ist groß

⇒ NP, “ya”, E, VP

Beispiel 9: Imperativ

Xĉ

heca!

less-d/10

Geh zur Seite!

⇒ Verb(imperativ)

Beispiel 10: Verb+Subjekt (1.P)

ᓂᓂᓂ

melON

ICH liebe

less-c/76

(-n/-nye)

⇒ Verb(“-nye”)

Beispiel 11: Verb+Subjekt und Objekt

ḿḿḿ ḿḿ

melON massa

ICH liebe Brot

less-c/76

(-n/-nye)

⇒ Verb(“-nye”), NP

Beispiel 12: Verb+Subjekt (2.P)

λϑêζ

hiruvaLYË

DU wirst finden

less-b/33

(-lye)

⇒ Verb(“-lye”)

Beispiel 13: Verb+Subjekt+Objekt

ḿḥḿḥ

meIONLYË

ICH liebe DICH

less-b/33

-SE-OE

⇒ Verb(“-nye”, “-lye”)

Merke:

Subjekt-Endung vor Objekt-Endung! (less-b/36)

Quenya-Grammatik

Zusammenfassung der Regeln

1. $S \rightarrow NP, \text{ Verb}$
2. $S \rightarrow NP, \text{ Verb, NP}$
3. $S \rightarrow \text{Verb, NP}$
4. $S \rightarrow NP, \text{ Copula, Adjektiv(Positiv)}$
5. $S \rightarrow NP, \text{ Copula, Adjektiv(Komparativ), NP}$
6. $S \rightarrow NP, \text{ Copula Adjektiv(Superlativ)}$
7. $S \rightarrow \text{"eque", NP, ":", E}$
8. $S \rightarrow NP, \text{"ya", E, VP}$
9. $S \rightarrow \text{Verb(imperativ)}$
10. $S \rightarrow \text{Verb("-nye")}$
11. $S \rightarrow \text{Verb("-nye"), NP}$
12. $S \rightarrow \text{Verb("-lye")}$
13. $S \rightarrow \text{Verb("-nye", "-lye")}$

Zusammenfassung der Regeln

1. $S \rightarrow NP, \text{ Verb}$
2. $S \rightarrow NP, \text{ Verb, NP}$
3. $S \rightarrow \text{Verb, NP}$
4. $S \rightarrow NP, \text{ Copula, Adjektiv(Positiv)}$
5. $S \rightarrow NP, \text{ Copula, Adjektiv(Komparativ), NP}$
6. $S \rightarrow NP, \text{ Copula Adjektiv(Superlativ)}$
7. $S \rightarrow \text{"eque"}, NP, \text{":"}, E$
8. $S \rightarrow NP, \text{"ya"}, E, \text{VP}$
9. $S \rightarrow \text{Verb(imperativ)}$
10. $S \rightarrow \text{Verb("-nye")}$
11. $S \rightarrow \text{Verb("-nye"), NP}$
12. $S \rightarrow \text{Verb("-lye")}$
13. $S \rightarrow \text{Verb("-nye", "-lye")}$

Zusammenfassung der Regeln

1. $S \rightarrow NP, \text{ Verb}$
2. $S \rightarrow NP, \text{ Verb, NP}$
3. $S \rightarrow \text{ Verb, NP}$
4. $S \rightarrow NP, \text{ Copula, Adjektiv(Positiv)}$
5. $S \rightarrow NP, \text{ Copula, Adjektiv(Komparativ), NP}$
6. $S \rightarrow NP, \text{ Copula Adjektiv(Superlativ)}$
7. $S \rightarrow \text{ "eque", NP, ":", E}$
8. $S \rightarrow NP, \text{ "ya", E, VP}$
9. $S \rightarrow \text{ Verb(imperativ)}$
10. $S \rightarrow \text{ Verb("-nye")}$
11. $S \rightarrow \text{ Verb("-nye"), NP}$
12. $S \rightarrow \text{ Verb("-lye")}$
13. $S \rightarrow \text{ Verb("-nye", "-lye")}$

Quenya-Grammatik: Satz

1. $S \rightarrow NP, VP$
2. $S \rightarrow \text{Verb}, NP$
3. $S \rightarrow \text{"eque"}, NP, \text{":"}, E$
4. $S \rightarrow NP, \text{"ya"}, E, VP$
5. $S \rightarrow \text{Verb}(\text{imperativ})$
6. $S \rightarrow \text{Verb}(\text{"-nye"})$
7. $S \rightarrow \text{Verb}(\text{"-nye"}), NP$
8. $S \rightarrow \text{Verb}(\text{"-lye"})$
9. $S \rightarrow \text{Verb}(\text{"-nye"}, \text{"-lye"})$

Quenya-Grammatik: Verb-Phrasen

1. VP → Verb
2. VP → Verb, NP
3. VP → Copula, Adjektiv(Positiv)
4. VP → Copula, Adjektiv(Komparativ), NP
5. VP → Copula Adjektiv(Superlativ)

Parser – Theorie und Umsetzung in Prolog

Überblick zur Theorie

- Kontextfreie Grammatiken (CFG)
- CFG-Implementierung mit `append()`
- CFG-Implementierung mit Differenzlisten
- Definite Clause Grammatiken (DCG)
- Erweiterungen für DCGs

Kontextfreie Grammatiken (1/2)

- CFG: Sammlung von Regeln, die besagen welche Sätze grammatikalisch (syntaktisch) richtig sind, und wie sie aufgebaut werden.
- Besteht aus terminalen Symbolen (Alphabet) und non-terminalen Symbolen (Regeln)
- Anhand der Regeln kann die Struktur eines Satzes z.B. in Form eines Syntaxbaumes aufgestellt und die Richtigkeit eines Satzes verifiziert werden

Kontextfreie Grammatiken (2/2)

- Ein kontextfreier Parser ist ein Programm das entscheidet, ob eine Zeichenkette (Wort oder Satz) zu der Sprache gehört, die durch die Kontextfreie Grammatik definiert wird, UND wie die zugehörige Struktur aussieht
- Eine kontextfreie Sprache umfasst alle Sätze, die mit Hilfe einer kontextfreien Grammatik erschaffen werden können.

CFG Beispiel

“a woman shoots a man”:

s -> np vp

np -> det n

vp -> v np

vp -> v

det -> a

det -> the

n -> woman

n -> man

v -> shoots

sentence

nominal-phrase

verbal-phrase

verbal-phrase

artikel

artikel

nomen

nomen

verb

CFG-Implementierung mit append() #1

Satz: [a,woman,shoots,a,man]

s(Z) :- np(X), vp(Y), append(X, Y, Z).

np(Z) :- det(X), n(Y), append(X, Y, Z).

vp(Z) :- v(X), np(Y), append(X, Y, Z).

vp(Z) :- v(Z).

det([the]).

det([a]).

n([woman]).

n([man]).

v([shoots]).

CFGs mit append()

- Problem: Es werden alle möglichen Sätze der Sprache generiert, und gegen den eingegebenen Satz verglichen.
- Abhilfe: Eingegebenen Satz zerlegen:

$s(Z) :- \text{append}(X, Y, Z), \text{np}(X), \text{vp}(Y).$

$\text{np}(Z) :- \text{append}(X, Y, Z), \text{det}(X), \text{n}(Y).$

$\text{vp}(Z) :- \text{append}(X, Y, Z), \text{v}(X), \text{np}(Y).$

CFGs mit append() #2

Ablauf:

- der Eingabesatz wird in die verschiedenen Möglichkeiten zerlegt

$X = [a], Y = [women, shoots, a, man],$

$X = [a, woman], Y = [shoots, a, man],$

...

- die verschiedenen Möglichkeiten werden gegen die Regeln verglichen

CFGs mit `append()` #3

Problem: bei kurzen Eingaben effizient, bei langen Sätzen jedoch auch zuviel herumprobieren.

Alle Ansätze mit `append()` sind zeitaufwendig!

CFGs mit Differenzlisten #1

Idee:

- nicht alle/verschiedene mögliche Sätze generieren und gegen die Eingabe vergleichen
- Eingabe von links nach rechts durchlesen, und das jeweils ganz links stehende Eingabe-Token klassifizieren, um zu prüfen ob die Komponenten innerhalb der Grammatik erlaubt sind.

CFGs mit Differenzlisten #2

Beispiel: $s([a, \text{woman}, \text{shoots}, a, \text{man}], [])$.

```
s(X, Z) :- np(X, Y), vp(Y, Z).  
np(X, Z) :- det(X, Y), n(Y, Z).  
vp(X, Z) :- v(X, Y), np(Y, Z).  
vp(X, Z) :- v(X, Z).  
det([the | W], W).  
det([a | W], W).  
n([woman | W], W).  
n([man | W], W).  
v([shoots | W], W).
```

CFGs: append() vs. Differenzlisten

Vergleich der beiden Ansätze mit append() und Differenzlisten:

- append(): lesbar aber ineffizient
- Differenzlisten: effizient, aber weniger lesbar

Wunsch: Ansatz mit Differenzlisten, der so lesbar wie die Implementierung mit append() ist!

⇒ Definite Clause Grammar

Definite Clause Grammatiken #1

DCG: Schreibweise, die die unterliegenden Differenzlisten versteckt.

Abfrage wie bei Differenzlisten:

```
s([a,woman,shoots,a,man],[ ]).
```

Definite Clause Grammatiken #2

Regeln in Prolog:

s --> np, vp.

np --> det, n.

vp --> v, np.

vp --> v.

det --> [the].

det --> [a].

n --> [woman].

n --> [man].

v --> [shoots].

Definite Clause Grammatiken #3

- Die Schreibweise mit “-->” ist nur syntaktischer Zucker, die die komplexe Schreibweise der Differenzlisten versteckt, intern jedoch effizient durch diese umgesetzt wird
- DCGs schützen nicht vor Fehlern:

s --> s , conj , s .

conj --> [and] .

conj --> [or] .

conj --> [but] .

Erweiterungen von DCGs

- Erweiterung mit zusätzlichen Attributen
- Erweiterung durch zusätzliche Prädikate

DCG-Erweiterung mit Attributen #1

Beispiel: Es sollen Pronomen "he", "she", "him", "her" eingeführt werden. Die Nomen-Regel wird zusätzlich entsprechend erweitert:

```
np--> pro.  
pro --> [he].  
pro --> [she].  
pro --> [him].  
pro --> [her].
```

DCG-Erweiterung mit Attributen #2

Problem:

she shoot him
he shoot her

werden erkannt, aber auch:

him shoot she
her shoot him

Lösung: Unterschiedliches Pronomen für Subjekt
und Objekt

DCG-Erweiterung mit Attributen #3

Verschlimmbesserung der Grammatik:

```
s --> np_subject, vp.  
np_subject --> det, n.  
np_object --> det, n.  
np_subject --> pro_subject.  
np_object --> pro_object.  
vp --> v, np_object.  
vp --> v.  
pro_subject --> [he].  
pro_subject --> [she].  
pro_object --> [him].  
pro_object --> [her].
```

...

DCG-Erweiterung mit Attributen #4

Einfacher geht's mit zusätzlichen Attributen:

```
s --> np(subject),vp.  
np(_) --> det,n.  
np(X) --> pro(X).  
vp --> v,np(object).  
vp --> v.  
pro(subject) --> [he].  
pro(subject) --> [she].  
pro(object) --> [him].  
pro(object) --> [her].  
...
```

DCG-Erweiterung mit Prädikaten #1

Hinzufügen beliebiger Prolog-Klauseln auf der rechten Seite, die “wahr” sein müssen:

```
lex(the,det).  
lex(a,det).  
lex(woman,n).  
lex(man,n).  
lex(shoots,v).
```

```
det --> [Word], {lex(Word,det)}.  
n    --> [Word], {lex(Word,n)}.  
v    --> [Word], {lex(Word,v)}.
```

DCG-Erweiterung mit Prädikaten #2

Anstatt über Prädikate mit fester Ordinalität können die Merkmale auch in Listen von [Merkmalsname, Merkmalsausprägung] gehalten werden:

```
ne([[cat, noun], [stem, andond],  
    [num, singular],  
    [casus, alt([nominative,  
                possessive, allative,  
                instrumental])],  
    [germ, [tor]]]) --> "andon".
```

Anwendung der DCG

- Kongruenzbeziehungen: Attribute von Verbal- und Nominalphrasen mit Hilfe zusätzlicher Attribute und Prädikate überprüfen
- Abgrenzung zur Semantik: anhand von Syntaxbäumen, die mit Hilfe der DCG erstellt wurden

Literatur #1

- Rosemarie Lühr: Neuhochdeutsch – Eine Einführung in die Sprachwissenschaft. UTB Verlag, Stuttgart, 1996.
- Thorsten Renk: Quetin i lambe eldaiva – Ein Quenya-Sprachkurs. Version 0.5. URL: <http://www.lambion-ardava.de/Downloads/queThkom.pdf> (Stand: 9.12.2003)

Literatur #2

- Helge Kåre Fauskanger: Quenya Course. URL: <http://www.uib.no/People/hnohf/qcourse.htm> (Stand: 17.5.2004)
- Patrick Blackburn, Johan Bos und Kristina Striegnitz: Learn Prolog Now! URL: <http://www.coli.uni-sb.de/~kris/learn-prolog-now/> (Stand: 17.5.2004)