
Vorlesung “Grammatikformalismen”

5. Übung (28.06.2019)

Sommersemester 2019 – Prof. Dr. Alexander Koller

Willkommen in der letzten Übung! Diesmal entwickeln Sie eine Grammatik für Englisch mit einer vereinfachten Variante von HPSG. Dazu verwenden Sie das LKB-System, das Sie auf Ihrem Computer installieren können (siehe Webseite).

Da eine LKB-Grammatik aus mehreren Dateien besteht, reichen Sie diesmal bitte nur eine einzige Grammatik (als ZIP-Datei gepackt) ein. Bitte dokumentieren Sie dafür um so gründlicher, wie Sie die einzelnen Teilaufgaben gelöst haben. Als Ausgangspunkt verwenden Sie die einfache HPSG-Grammatik für Englisch, die wir auf der Webseite bereitgestellt haben. Ihre Grammatik soll dann alle angegebenen grammatischen Sätze parsen können und alle angegebenen ungrammatischen Sätze ablehnen. Wenn Sie möchten, können Sie eine feinkörnigere Typhierarchie einführen, um das Lexikon so aufgeräumt und einfach wie möglich zu halten.

1 Ditransitive Verben

Die Grammatik enthält Regeln, die größere Phrasen mit ditransitiven Verben bilden können (`head-complement-rule2` kombiniert ein Verb mit zwei NP-Komplementen). Es gibt allerdings noch keine Lexikoneinträge für ditransitive Verben.

Fügen Sie einen geeigneten Lexikoneintrag für das Verb *showed* zur Grammatik hinzu. Ihre Grammatik soll die folgenden Vorhersagen machen:

- (1) The dog showed the cat the cow.
- (2) *The dog showed.
- (3) *The dog showed the cat the cow the dog.

Die Werte der Features `SPR` und `COMPS` sind Listen mit Einträgen vom Typ *cat-struct*, auch wenn man das nicht direkt in der Typdeklaration sieht. Es ist wichtig, dass die Einträge nicht Typ *word* oder *phrase* haben. Warum?

2 Präpositionalphrasen

Das Verb *show* kann auch für NP+PP subkategorisieren.

- a) Fügen Sie zunächst PPs zur Grammatik hinzu. Sie brauchen dafür einen neuen Subtyp von *pos* mit der Definition `prep := pos & [PFORM string]`. Das bedeutet: *prep* ist ein Subtyp von *pos*, der zusätzlich ein PFORM-Feature mit Werten vom Typ *string* hat.
- b) Definieren Sie Lexikoneinträge für die Präpositionen *from* und *to*. Nehmen Sie an, dass eine Präposition für ein NP-Komplement subkategorisiert. Stellen Sie sicher, dass das PFORM-Feature passende Werte erhält. Sie sollten jetzt einen String wie *to the cow* als PP analysieren können. (Man kann das ausprobieren, indem man vorübergehend den Wert von `*start-symbol*` in der Datei `globals.lsp` von `root` auf `*top*` abändert.)
- c) Fügen Sie nun einen zweiten Lexikoneintrag für *showed* zum Lexikon hinzu (er muss anders heißen als der andere, z.B. `showed_nppp`). Stellen Sie sicher, dass *showed* nur PP-Komplemente mit Kopf *to* zulässt:
 - (4) The dog showed the cow to the cat.
 - (5) * The dog showed the cow from the cat.

3 Weg mit head-complement-rule-0

Die Datei `rules.tdl` enthält die Schemata, z.B. die `head-complement`-Regeln (zum Kombinieren von Kopf und Komplement) und die `head-specifier`-Regel (zum Kombinieren von Kopf und Specifier). Man kann sich in LKB die Parse-Chart mit den verwendeten Regeln anzeigen lassen.

Die einzige Funktion der Regel `head-complement-rule-0` ist es, Wörter mit `COMPS < >` in Phrasen mit `COMPS < >` zu verwandeln. Entfernen Sie diese Regel aus der Grammatik, ohne dabei die Sprache und die Anzahl der Analysen pro Satz zu verändern. Tipp: Bedenken Sie die Typconstraints, die andere Regeln (z.B. `head-specifier-rule`) für ihre Töchter annehmen.

4 Kopfmerkmals-Prinzip

In der Vorlesung haben wir das Kopfmerkmals-Prinzip besprochen. In der Grammatik in der Übung ist dieses Prinzip hart in die `Head-Specifier`- und die `Head-Complement`-Regeln codiert.

Da alle unsere Phrasen Köpfe haben, können wir das Kopfmerkmals-Prinzip (anders als in der Vorlesung, wo es sich auf *headed-phrase* bezieht) als Constraint auf dem Typ *phrase* implementieren. Dazu modifiziert man am einfachsten die Definition von *phrase* in `types.tdl`. Machen Sie das, und vereinfachen Sie dann die Definitionen der Schemata in `rules.tdl`.

Da in der `ARGS`-Liste die Kopftochter nicht immer an der gleichen Stelle steht, werden Sie einen Weg finden müssen, sie separat zu kennzeichnen.

5 Eine binär verzweigende Head-Complement-Regel

Die Grammatik enthält jetzt noch zwei Varianten der Head-Complement-Regel; sie unterscheiden sich in der Anzahl der Komplemente, die sie in einem Schritt konsumieren. Ersetzen Sie diese beiden Regeln durch eine einzige, die bei jeder Anwendung genau ein Komplement konsumiert. Sie können dabei die COMPS-Liste entweder mit der abgekürzten TDL-Listensyntax auseinandernehmen, oder die Implementierung von Listen als Featurestrukturen der Form [FIRST *top*, REST *list*] ausnutzen. Stellen Sie sicher, dass die veränderte Grammatik den ditransitiven Sätzen nicht mehr als eine Analyse zuweist.

6 Modifikation

Implementieren Sie ein Head-Adjunct-Schema wie in der Vorlesung (oder auch in Müller, *Grammatiktheorie*, Kapitel 8.1.7). Sie müssen daran denken, dass die Grammatik in der Übung einfachere Featurestrukturen verwendet. Achten Sie außerdem darauf, dass die SPR- und COMPS-Features der Head-Adjunct-Phrase angemessene Werte bekommen. (Diese werden bei Müller vom allgemeineren “Subcategorization Principle” behandelt.)

Probieren Sie Ihr neues Schema mit einem neuen Lexikoneintrag für *fat* aus. Dafür müssen Sie zunächst einen neuen Subtyp von *pos* für Adjektive mit einem zusätzlichen MOD-Feature einführen. Ihre Grammatik sollte jetzt die folgenden Vorhersagen machen:

- (6) The fat cat barks.
- (7) * Fat the cat barks.

7 Kongruenz

Ändern Sie die Lexikoneinträge so ab, dass Kongruenz zwischen Verb und Subjekt sowie zwischen Substantiv und Artikel sichergestellt wird. Dazu brauchen Sie neue Typen. Sie müssen sich nur um Numerus-Kongruenz kümmern.

- (8) The dogs bark.
- (9) * The dog bark.
- (10) * Those dog barks.