

Chapter 4: Features and Augmented Grammar

4.1 Feature Systems and Augmented Grammars

- Context Free Grammars provides the basis for most the computational parsing mechanisms, but they would be very inconvenient for capturing natural languages, because they can not capture meaning and categories of words.
- The chapter describes a extension to the basic context free mechanism, that defines the constituents by a set of features.

•number-agreement

- + in NP: between **art** and **noun** ; example: *a men*
- + subject-verb agreement; example: *the man cry*
- + gender agreement for pronouns; example restriction between the head of a phrase and the form of its complement.

Chapter 4: Features and Augmented Grammar

4.1 Feature Systems and Augmented Grammars

-Allow constituents to have features.

-Feature NUMBER may take a value of either s (singular) or p (plural):

Example: NP – sing □ ART – sing N – sing

 NP – plural □ ART – plural N – plural

-Constituent is defined as a feature structure:

Example 1: ART1 (CAT ART

 ROOT: a

 NUMBER s)

 or ART1 (ART ROOT a NUMBER s)

Chapter 4: Features and Augmented Grammar

4.1 Feature Systems and Augmented Grammars

Example 2:

NP1: (NP NUMBER s

1 (ART ROOT a NUMBER s)

2 (N ROOT fish NUMBER s))

-Variable are allowed as feature values so that the rules can apply to wide range of situations.

Example 3: (NP NUMBER ?n) →

(ART NUMBER ?n) (N NUMBER ?n)

- **Chapter 4: Features and Augmented Grammar**

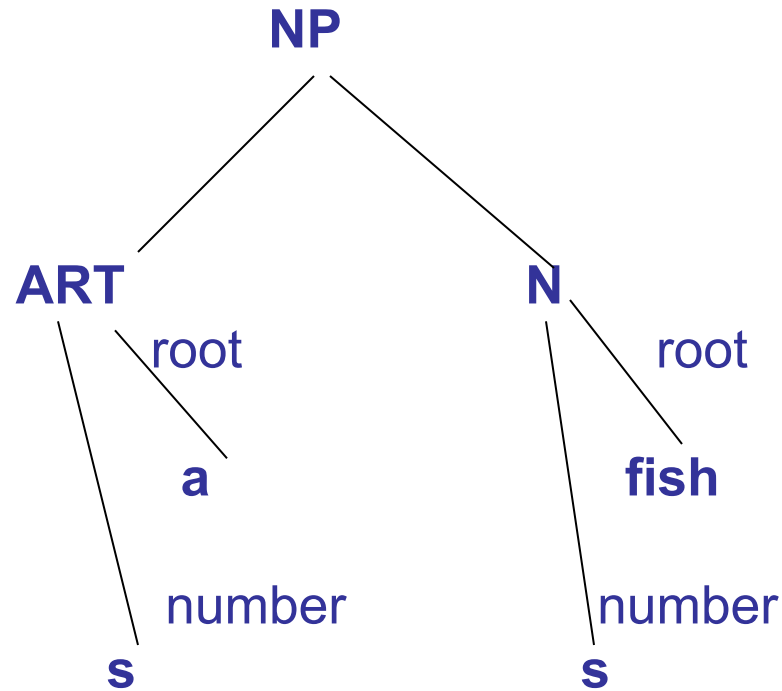


Figure 4.1: A feature structure as an extended parse tree

Chapter 4: Features and Augmented Grammar

4.2. Basic feature *systems* for English

- Person and number features: 1s, 2s, 3s, 1p, 2p, 3p
- Verb Form and verb subcategorization: VFORM, SUBCAT
- Prepositional feature: PFORM
- Binary feature
- Default value of feature
- ***Person and number features:***
 - Number system in English: word may be described as a single object or multiple object.
 - Number agreement restrictions occur in subject-verb agreement. But subjects and verbs must also agree on another dimension, namely with respect to ***person:***

Chapter 4: Features and Augmented Grammar

4.2. Basic feature *systems* for English

- + First Person (1): speaker or a group of people including the speaker:
I, we, you and I.
- + Second Person (2): referring to listener, or a group of listener but no speaker: you, all of you.
- + Third Person (3): referring to one or more objects, not including the speaker or hearer.
- Since number and person features always co-occur, so to combine the two into a single feature ***AGR***, it has six possible values: 1s, 2s, 3s, 1p, 2p, 3p.

Chapter 4: Features and Augmented Grammar

4.2. Basic feature systems for English

- *Verb form Features* - VFORM

Form	Explanation	Examples
base	base	go, be, say, write
pres	simple present	go, goes, is, says, writes
past	simple past	went, was, said, wrote
fin	equivalent pres, past	
ing	Continuous participle	going, being, saying, writing
pastprt	past participle	Gone, been, said, written
inf	infinitive with <i>to</i>	go, be, say, write

Chapter 4: Features and Augmented Grammar

4.2. Basic feature systems for English

▪ *Verb Subcategorization* - SUBCAT

Feature value	verb	examples
-none	laugh	Jack laughed
-np	take	Jack takes a bus
-np-np	give	Jack gave Mary the book
-vp: inf	want	Jack wants to run
-np-vp:inf	tell	Jack told the man to go
-vp:ing	keep	I keep hoping for the best
-np-vp:base	catch	He caught the bus moving on the street
-np-vp:base	watch	Jack watched Sue look at her dress

Chapter 4: Features and Augmented Grammar

4.2. Basic feature systems for English

- *Feature PFORM*

Feature value	preposition	examples
to	to	John gave the money to the bank
Loc	to, on, by, inside, on top, of	I put it on the desk
Mot	to, from, along	We walked to the beach

- *Binary features*

Binary feature is a part of syntactic structure , in which a constituent either has or doesn't have feature. A binary feature has a value to be either + or - .

+ INV feature is a binary feature that indicates whether or not S structure has inverted subject (as in yes/no question).

Example: Jack laughed (-INV); Did Jack laugh ? (+INV)

Chapter 4: Features and Augmented Grammar

4.2. Basic feature systems for English

- *The default value for features*

It will be useful to allow a default value of features. Any time a constituent is constructed that could have a feature, but a value is not specified, the feature takes a default value -. This is especially useful for binary features, but is used for non-binary features as well. The default value is inserted when the constituent is first constructed

Chapter 4: Features and Augmented Grammar

4.3 Morphological Analysis and the lexicon

Morphological Analysis

The lexicon

Often a word will have multiple interpretations that use different entries and different lexical rules.

Example word *saw*, that has three entries in the lexicon

saw: (CAT N
ROOT SAW1
AGR 3s)

saw: (CAT V
ROOT SAW2
VFORM base
SUBCAT _np)

saw: (CAT V
ROOT SEE1
VFORM past
SUBCAT _np)

Chapter 4: Features and Augmented Grammar

4.3 Morphological Analysis and the lexicon

a	(CAT ART ROOT A1 AGR 3s)	saw	(CAT N ROOT SAW1 AGR 3s)	is	(CAT V ROOT BE1 VFORM pres SUBCAT{ _adjp _np } AGR 3s)
be	(CAT V ROOT BE1 VFORM base IRREG-PRES + IRREG-PAST + SUBCAT(_adjp _np))	saw	(CAT V ROOT SAW2 VFORM base SUBCAT _np)	jack	(CAT NAME AGR 3s)
cry	(CAT V ROOT CRY1 VFORM base SUBCAT none)	saw	(CAT V ROOT SEE1 VFORM past SUBCAT _np)	man	(CAT N1 ROOT MAN1 AGR 3s)
dog	(CAT N ROOT DOG1 AGR 3s)	see	(CAT V ROOT SEE1 VFORM base SUBCAT _np IRREG-PAST + EN-PASTPRT +)	men	(CAT N ROOT MAN1 AGR 3p)
fish	(CAT N TOOT FISH1 AGR (3s, 3p) IRREG-PL+)	seed	(CAT N ROOT SEED1 AGR 3s)	want	(CAT V ROOT WANT1 VFORM base SUBCAT{ _np _ vp:inf }
happy	(CAT ADJ SUBCAT _vp:inf)	to	(CAT TO)	was	(CAT V ROOT BE1 VFORM past AGR {1s, 3s} SUBCAT { _adjp _np })
he	(CAT PRO ROOT HE1 AGR 3s)	the	(CAT ART ROOT THE1 AGR {3s, 3p})	were	(CAT V ROOT BE1 VFORM past AGR {2s, 1p, 2p, 3p})

Figure 4.2: A Lexicon

Chapter 4: Features and Augmented Grammar

4.4. A simple Grammar using Features

Using a lexicon and features to define grammars with features (Augmented Grammars)

1. $S[-inv] \rightarrow (NP \ AGR \ ?a) (VP [\{pres \ past\}] \ AGR \ ?a)$
2. $NP \rightarrow (ART \ AGR \ ?a) (N \ AGR \ ?a)$
3. $NP \rightarrow PRO$
4. $VP \rightarrow V[_{none}]$
5. $VP \rightarrow V[_{np}] NP$
6. $VP \rightarrow V[_{vp:inf}] VP[inf]$
7. $VP \rightarrow V[_{np_vp:inf}] NP \ VP[inf]$
8. $VP \rightarrow V[_{adjp}] ADJP$
9. $VP[inf] \rightarrow TO \ VP[base]$
10. $ADJP \rightarrow ADJ$
11. $ADJP \rightarrow ADJ[_{vp:inf}] VP[inf]$

Head features for S, VP: VFORM, AGR

Head features for NP: AGR

Figure 4.3: A simple grammar in abbreviated form

Chapter 4: Features and Augmented Grammar

4.4. A simple Grammar using Features

1. $(S \text{ INV} - \text{VFORM } ?v[\text{pres past}] \text{ AGR } ?a) \rightarrow$
 $(NP \text{ AGR } ?a) (VP \text{ VFORM } ?v[\text{pres past}] \text{ AGR } ?a)$
2. $(NP \text{ AGR } ?a) \rightarrow (\text{ART AGR } ?a) (N \text{ AGR } ?a)$
3. $(NP \text{ AGR } ?a) \rightarrow (\text{PRO AGR } ?a)$
4. $(VP \text{ AGR } ?a \text{ VFORM } ?v) \rightarrow (V \text{ SUBCAT_none AGR } ?a \text{ VFORM } ?v)$
5. $(VP \text{ AGR } ?a \text{ VFORM } ?v) \rightarrow (V \text{ SUBCAT_np AGR } ?a \text{ VFORM } ?v) NP$
6. $(VP \text{ AGR } ?a \text{ VFORM } ?v) \rightarrow$
 $(V \text{ SUBCAT_vp_inf AGR } ?a \text{ VFORM } ?v) (VP \text{ VFORM inf})$
7. $(VP \text{ AGR } ?a \text{ VFORM } ?v) \rightarrow$
 $(V \text{ SUBCAT_np_vp_inf AGR } ?a \text{ VFORM } ?v) NP (VP \text{ VFORM inf})$
8. $(VP \text{ AGR } ?a \text{ VFORM } ?v) \rightarrow$
 $(V \text{ SUBCAT_adjp AGR } ?a \text{ VFORM } ?v) \text{ADJP}$
9. $(VP \text{ SUBCAT inf AGR } ?a \text{ VFORM inf}) \rightarrow$
 $(\text{TO AGR } ?a \text{ VFORM inf}) (VP \text{ VFORM base})$
10. $\text{ADJP} \rightarrow \text{ADJ}$
11. $\text{ADJP} \rightarrow \text{ADJ} (\text{SUBCAT_inf}) (VP \text{ VFORM inf})$

Figure 4.4: The expanded grammar showing all features

Chapter 4: Features and Augmented Grammar

4.4. A simple Grammar using Features

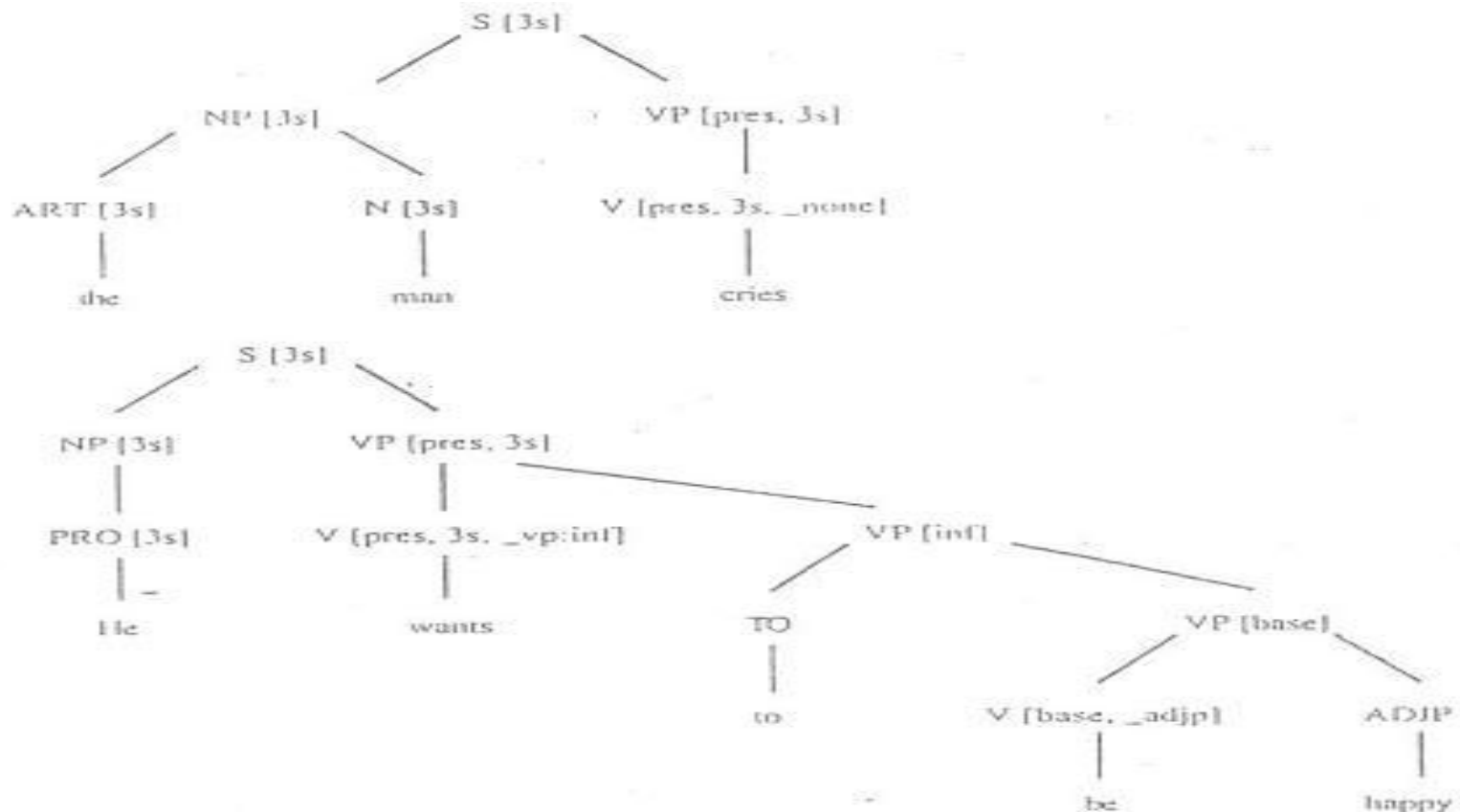


Figure 4.5: Two simple parse trees with feature values

Chapter 4: Features and Augmented Grammar

4.4. A simple Grammar using Features

- Many feature values are unique to a feature.

Example: feature value *inf* can only appear in the VFORM, *_np_vp:inf* can only appear in SUBCAT.

- Unique feature values will be listed in square parentheses “[]”, example (VP SUBCAT inf) will be abbreviated as VP [inf].
- The feature value on the mother must be identical to the value on its head constituent, there are called head features.

Example: (VP VFORM ?v AGR ?a) □

(V VFORM ?v AGR ?a SUBCAT _np_vp:inf)

(NP)

(VP VFORM inf)

(Look at the grammar on the figure 4.4)

Chapter 4: Features and Augmented Grammar

4.4. A simple Grammar using Features

The head features may be declared separately from the rules.

With VFORM and AGR declared as head features, the VP rule can be abbreviated as:

VP \rightarrow (V SUBCAT _np_vp:inf) NP (VP VFORM inf)

The rule could be further simplified to:

VP \rightarrow V [_np_vp:inf] NP VP [inf]

(Look at the grammar on the figure 4.3)

Chapter 4: Features and Augmented Grammar

4.5 Parsing with Features

The parsing algorithms developed in Chapter 3 for CFG can be extended to handle augmented CFG.

Chart parsing algorithms developed in Chapter 3 all used an extending active arcs with new constituent.

$C \sqsubset C_1 \dots C_i \cdot X \dots C_n$
to produce a new arc of the form:

$C \sqsubset C_1 \dots C_i X \cdot \dots C_n$

Each constituent has feature values

Example: Parse the sentence “*a dog*”

1. $(NP \text{ AGR ? a }) \sqsubset \cdot (ART \text{ AGR ? a }) (N \text{ AGR ? a })$

$(NP \text{ AGR ? a }) \sqsubset (ART \text{ AGR ? a }) \cdot (N \text{ AGR ? a })$

2. Take information of ART from lexicon:

$(ART \text{ root a AGR 3s })$

To make arc 1 applicable, the variable ?a must be 3s, producing

Chapter 4: Features and Augmented Grammar

4.5 Parsing with Features

3. (NP AGR 3s) \square · (ART AGR 3s) (N AGR 3s)

This arc can now be extended, because every feature in the rule in constituent 2.

4. (NP AGR 3s) \square (ART AGR 3s) · (N AGR 3s)

Consider extending this arc with constituent (N AGR 3s) for the word *dog*

5. (N root DOG1 AGR 3s)

Then can done because the AGR features agree. This completes the arc:

6. (NP AGR 3s) \square (ART AGR 3s) (N AGR 3s) ■

That means the parser found a constituent with then form
(NP AGR 3s)

Chapter 4: Features and Augmented Grammar

4.5 Parsing with Features

- *The algorithm can be specified more precisely:*

- Given an arc A, the constituent following the dot is called NEXT and a new constituent X, which is being used to extend the arc.

- Find an *instantiation* of the variables such as that features specified in NEXT, are found in X.
- Create a new arc A' is a copy of A, except for the *instantiations* of the variables determined in step a.
- Update A' as usual in a chart parser.

Figure 4.10 describes the process of chart parsing for the sentence *He wants to cry*.

Chapter 4: Features and Augmented Grammar

4.5 Parsing with Features

S1 CAT S AGR 3s VFORM pres INV- 1 NP1 2 VP3			
		VP3 CAT VP VFORM pres AGR 3s 1 V1 2 VP2	
		VP2 CAT VP VFORM inf 1 TO1 2 VP1	
NP1 CAT NP AGR 3s 1 PRO1			VP1 CAT VP VFORM base 1 V2
PRO1 CAT PRO AGR 3s	V1 CAT V ROOT want VFORM pres AGR 3s SUBCAT [_np,_vp:inf, _np_vp:inf]	TO1 CAT TO	V2 CAT V ROOT cry VFORM base SUBCAT _none
He	wants	to	cry

Figure 4.6: The chart parsing of sentence “He wants to cry”

EXERCISES FOR CHAPTER 4

1. Extend the lexicon in Figure 4.2, slide 12, and the grammar in figure 4.3, slides 13. So that the following two sentences are accepted:

He was sad to see the dog cry.

He saw the man saw the wood with the saw.

Parse two above sentences by Top-Down Chart Parsing.

2. Specify an augmented context free grammar and lexicon for simple subject-verb-object sentences. The grammar only allows appropriate pronouns in subject and object positions and does number agreement between the subject và verb.

Tus it should accept “*I hit him*” but ont “*me love you*”