

# Computational Linguistics

CSC 485  
Summer 2020

5A

Assignment Project Exam Help

## 5a. Extending grammars with features

<https://powcoder.com>

Add WeChat powcoder

Gerald Penn

Department of Computer Science, University of Toronto

Reading: Jurafsky & Martin: 12.3.4–6, 15.0–3;  
[Allen: 4.1–5]; Bird et al: 9.

Copyright © 2017 Suzanne  
Stevenson, Graeme Hirst  
and Gerald Penn. All rights  
reserved.

# Agreement and inflection

- Problem: **Agreement** phenomena.

*Nadia {washes/\***wash**} the dog.*

*The boys {\***washes**/wash} the dog.*

*You {\***washes**/wash} the dog.*

Assignment Project Exam Help

<https://powcoder.com>

- **Morphological inflection** of verb must match subject noun in person and number.

Add WeChat powcoder

# Subject—verb agreement 1

## Present tense

	Singular	Plural
1	<i>I wash</i>	<i>we wash</i>
2	<i>you wash</i>	<i>you wash</i>
3	<i>he/she/it washes</i>	<i>they wash</i>
1	<i>I am</i>	<i>we are</i>
2	<i>you are</i>	<i>you are</i>
3	<i>he, she, it is</i>	<i>they are</i>

# Subject–verb agreement 2

## Past tense

	Singular	Plural
1	<i>I washed</i>	<i>we washed</i>
2	<i>you washed</i>	<i>you washed</i>
3	<i>he, she, it washed</i>	<i>they washed</i>
1	<i>I was</i>	<i>we were</i>
2	<i>you were</i>	<i>you were</i>
3	<i>he, she, it was</i>	<i>they were</i>

# Agreement features 1

- English agreement rules are fairly simple.
  - Subject : verb w.r.t. person and number.
  - No agreement required between verb and object.
- Many languages have other agreements.
  - *E.g.*, German: Article and adjective ending depends on noun gender and case:

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Agreement features 2

## Nominative Case (Subject Case)

Masculine	Feminine	Neuter	Plural
<b>der</b>	<b>die</b>	<b>das</b>	<b>die</b>

der neu**e** Wagen

the new car

die schön**e** Stadt

the beautiful city

das alt**e** Auto

the old car

die neu**en** Bücher

the new books

Masculine	Feminine	Neuter	Plural
<b>ein</b>	<b>eine</b>	<b>ein</b>	<b>keine</b>

ein neu**er** Wagen

a new car

eine schön**e** Stadt

a beautiful city

ein alt**es** Auto

an old car

keine neu**en** Bücher

no new books

Ask about.com: German language: Adjective endings I and II.  
<http://german.about.com/library/weekly/aa030298.htm> and  
[aa033098.htm](http://german.about.com/library/weekly/aa033098.htm)

# Agreement features 2

## Accusative Case (Direct Object)

Masculine	Feminine	Neuter	Plural
<b>den</b>	<b>die</b>	<b>das</b>	<b>die</b>

den neu**en** Wagen

the new car

die schön**e** Stadt

the beautiful city

das alt**e** Auto

the old car

die neu**en** Bücher

the new books

Masculine	Feminine	Neuter	Plural
<b>einen</b>	<b>eine</b>	<b>ein</b>	<b>keine</b>

einen neu**en** Wagen

a new car

eine schön**e** Stadt

a beautiful city

ein alt**es** Auto

an old car

keine neu**en** Bücher

no new books

Ask about.com: German language: Adjective endings I and II.  
<http://german.about.com/library/weekly/aa030298.htm> and  
[aa033098.htm](http://german.about.com/library/weekly/aa033098.htm)

# Agreement features 3

*E.g.*, Chinese: Numeral classifiers, often based on shape, aggregation, ....:

两条鱼	<i>liang tiao yu</i> ‘two CLASSIF-LONG-ROPELIKE fish’
两条河	<i>liang tiao he</i> ‘two CLASSIF-LONG-ROPELIKE rivers’
两条腿	<i>liang tiao tui</i> ‘two CLASSIF-LONG-ROPELIKE legs’
两条裤子	<i>liang tiao kuzi</i> ‘two CLASSIF-LONG-ROPELIKE pants’
两只胳膊	<i>liang zhi gebo</i> ‘two CLASSIF-GENERAL arms’
两件上衣	<i>liang jian shangyi</i> ‘two CLASSIF-CLOTHES-ABOVE-WAIST tops’
两套西装	<i>liang tao xizhuang</i> ‘two CLASSIF-SET suits’

Zhang, Hong (2007). Numeral classifiers in Mandarin Chinese. *Journal of East Asian Linguistics*, 16(1), 43–59. Thanks also to Tong Wang, Vanessa Wei Feng, and Helena Hong Gao.



# Agreement features 1

- English agreement rules are fairly simple.
- Many languages have other agreements.
- Some languages have multiple grammatical genders.  
*Assignment Project Exam Help*
- E.g. Chichewa has genders for men, women, bridges, houses, diminutives, men inside houses, etc. Between 12-18 in total.  
*<https://powcoder.com>  
Add WeChat powcoder*
- Some languages overtly realize many of these distinctions.
- E.g. some Hungarian verbs have as many as 4096 inflected forms.

# Inflectional morphology

- Word may be inflected ...
  - ... to indicate paradigmatic properties, e.g. singular / plural, past / present, ...
  - ... to indicate some (other) semantic properties
  - ... to agree with inflection of other words.
- Each (open-class) word-type has a **base form / stem / lemma**.
- Each occurrence of a word includes inflection by a (possibly null) morphological change.

# Rule proliferation 1

- **Problem:** How to account for this in grammar.
- **Possible solution:** Replace all NPs, Vs, and VPs throughout the grammar.

Assignment Project Exam Help

$S \rightarrow NP VP$

$VP \rightarrow V NP$

$NP \rightarrow you, dog, dogs, bear, bears,$

$V \rightarrow washes, wash, washed, is,$

...

<https://powcoder.com>  
Add WeChat powcoder



$S \rightarrow NP_{3s} VP_{3s}$

$NP_2 \rightarrow you$

$V_{1s} \rightarrow am, was, wash,$

$S \rightarrow NP_{3p} VP_{3p}$

$\vdots$

$washed, \dots$

$S \rightarrow NP_2 VP_2$

$VP_{3s} \rightarrow V_{3s} NP$

$\vdots$

$S \rightarrow NP_{1s} VP_{1s}$

$\vdots$

$S \rightarrow NP_{1p} VP_{1p}$

$V_{3s} \rightarrow is, was,$   
 $washes, washed, \dots$

$NP_{3s} \rightarrow dog, bear, \dots$

$V_{3p} \rightarrow are, were,$

$NP_{3p} \rightarrow dogs, bears$

$wash, washed, \dots$

# Rule proliferation 2

- *Drawback 1:* the result is big ... really big.
- *Drawback 2:* Losing the generalization:
  - All these Ss, NPs, VPs have the same structure.
  - Doesn't depend on particular verb, noun, and number.  
<https://powcoder.com>
- CF rules collapse together structural and featural information.  
[Add WeChat powcoder](#)
- All information must be completely and directly specified.
  - *E.g.*, can't just say that values must be equal for some feature without saying exactly what values.

# Feature structures 1

- *Solution:* Separate feature information from syntactic, structural, and lexical information.
- A **feature structure** is a list of pairs:  
[*feature-name* *feature-value*]
- Feature-values may be atoms or feature structures.
- Can consider syntactic category or word to be bundle of features too.
- Can represent syntactic structure.

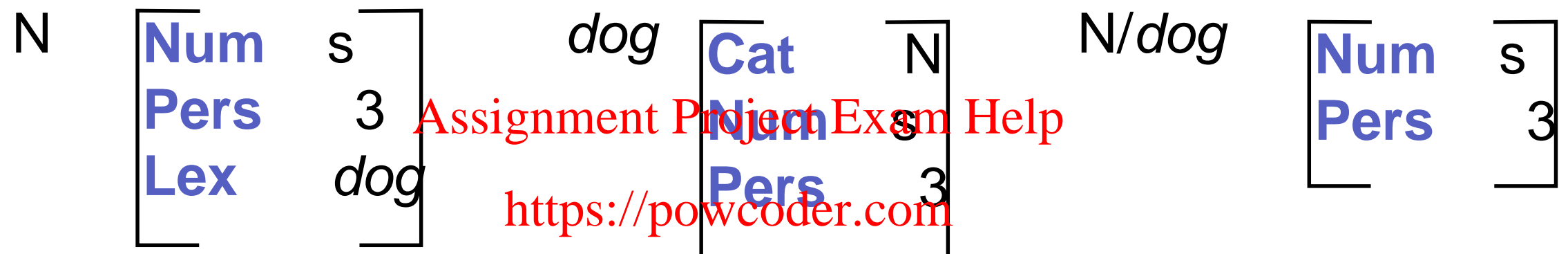
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Feature structures 2

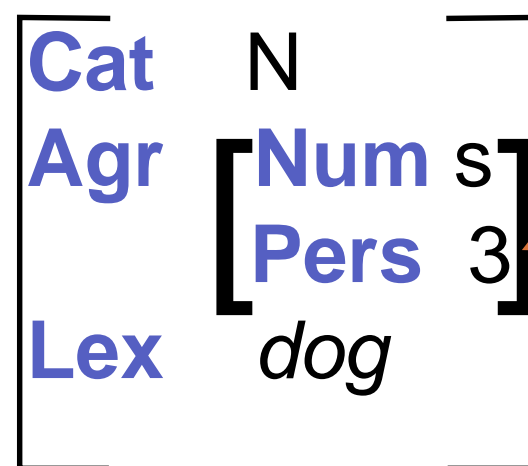
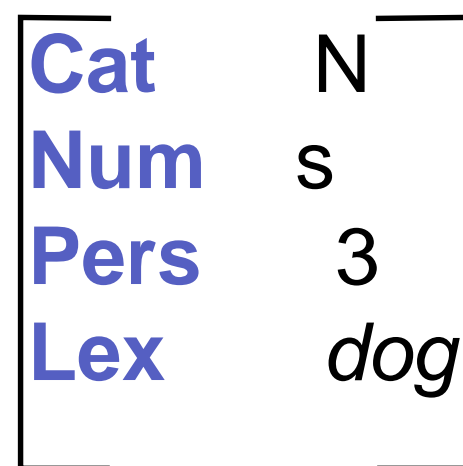
- *Drawback*: many equivalent notations.



Assignment Project Exam Help

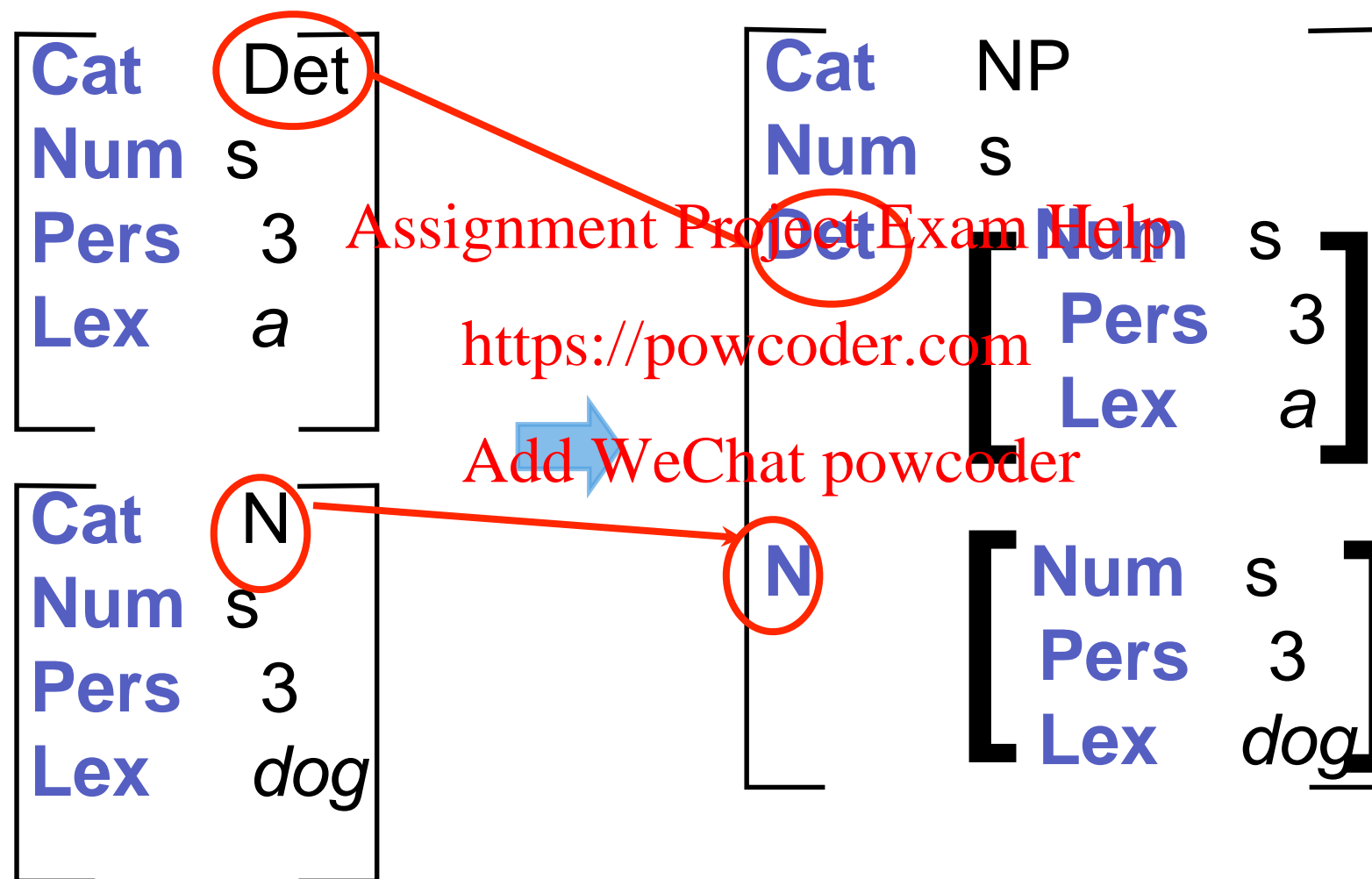
<https://powcoder.com>

Add WeChat powcoder



*Feature paths:*  
features of  
features; e.g.,  
(Agr Pers 3)

# Feature structures 3



NP formed from Det and N.  
Feature values in components become  
feature names in new constituent.

# Components of feature use

- 1. **Lexical specification:**

Description of *properties* of a word:  
morphological, syntactic, semantic, ...

Assignment Project Exam Help

*dog:*  $\begin{bmatrix} \text{Cat} & N \\ \text{Agr} & 3s \end{bmatrix}$  *sleeps:*  $\begin{bmatrix} \text{Cat} & V \\ \text{Agr} & 3s \end{bmatrix}$

*dogs:*  $\begin{bmatrix} \text{Cat} & N \\ \text{Agr} & 3p \end{bmatrix}$  *sleep:*  $\begin{bmatrix} \text{Cat} & V \\ \text{Agr} & \{1s, 2s, 1p, 2p, 3p\} \end{bmatrix}$

Or:  $\wedge 3s$

Or:  $N \rightarrow \text{dog}$   
 $(N \text{ Agr}) = 3s$

$N \rightarrow \text{dogs}$   
 $(N \text{ Agr}) = 3p$

$V \rightarrow \text{sleeps}$   
 $(V \text{ Agr}) = 3s$

$V \rightarrow \text{sleep}$   
 $(V \text{ Agr}) = \{1s, 2s, 1p, 2p, 3p\}$



# Components of feature use

- **2. Agreement:**
  - **Constraints** on co-occurrence in a rule — within or across phrases.
  - Typically are equational constraints.

<https://powcoder.com>

NP → Det N  
(Det **Num**) = (N **Num**)

S → NP VP  
(NP **Agr**) = (VP **Agr**)

# Components of feature use

- **3. Projection:**

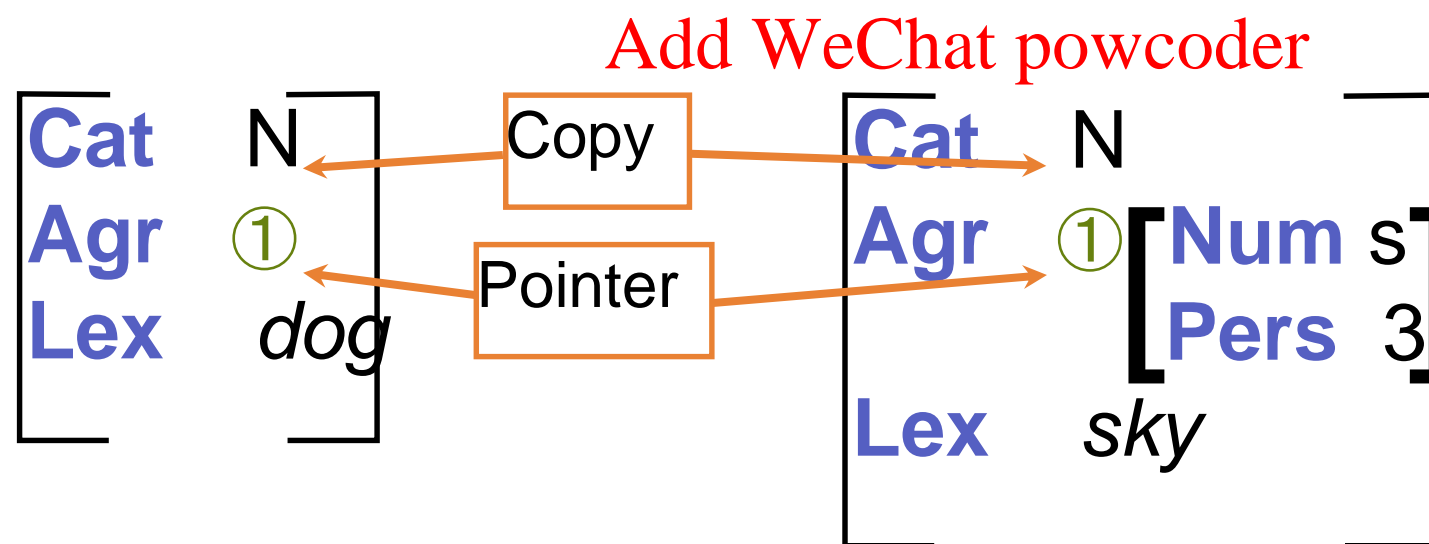
- ***Sharing of features*** between the head of a phrase and the phrase itself.

VP  $\rightarrow$  V ...  
(VP Agr) = (V Agr)

- Head features:
  - **Agr** is typical, but so is the head-word itself as a feature.  
(Common enough that there's usually a mechanism for "declaring" head features and omitting them from rules.)

# Constraints on feature values 1

- What does it mean for two features to be “equal”?
- A *copy* of the value or feature structure, or a *pointer* to the same value or feature structure (re-entrancy, shared feature paths).



# Constraints on feature values 2

- But: It may be sufficient that two features are not equal, just *compatible* — that they can be *unified*.

- E.g., 

Cat	N
Pers	3
Num	s

 and 

Cat	N
Pers	3
Grnd	F

Assignment Project Exam Help

<https://powcoder.com>

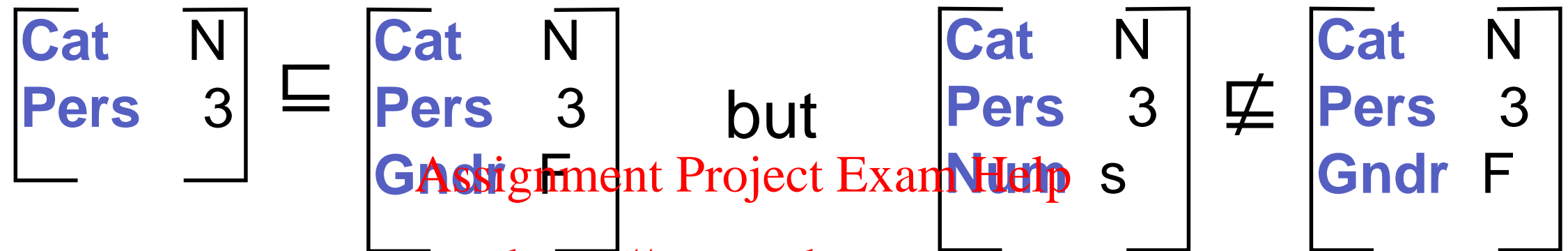
Add WeChat powcoder

# Subsumption of feature structures 1

- Feature structure  $X$  *subsumes* feature structure  $Y$  if  $Y$  is consistent with, and at least as specific as  $X$ .
- Also say that  $Y$  *extends*  $X$ .  
*Assignment Project Exam Help*  
*<https://powcoder.com>*  
*Add WeChat powcoder*  
 $Y$  can add (non-contradictory) features to those in  $X$ .
- **Definition:**  $X$  *subsumes*  $Y$  ( $X \sqsubseteq Y$ ) iff there is a *simulation* of  $X$  inside  $Y$ , i.e., a function s.t.:
  - $\text{sim}(X) = Y$
  - If  $X$  is atomic, so is  $Y$  and  $X = Y$
  - Otherwise, for all feature values  $X.f$ :  $Y.f$  is defined, and  $\text{sim}$  simulates  $X.f$  inside  $Y.f$ .

# Subsumption of feature structures 2

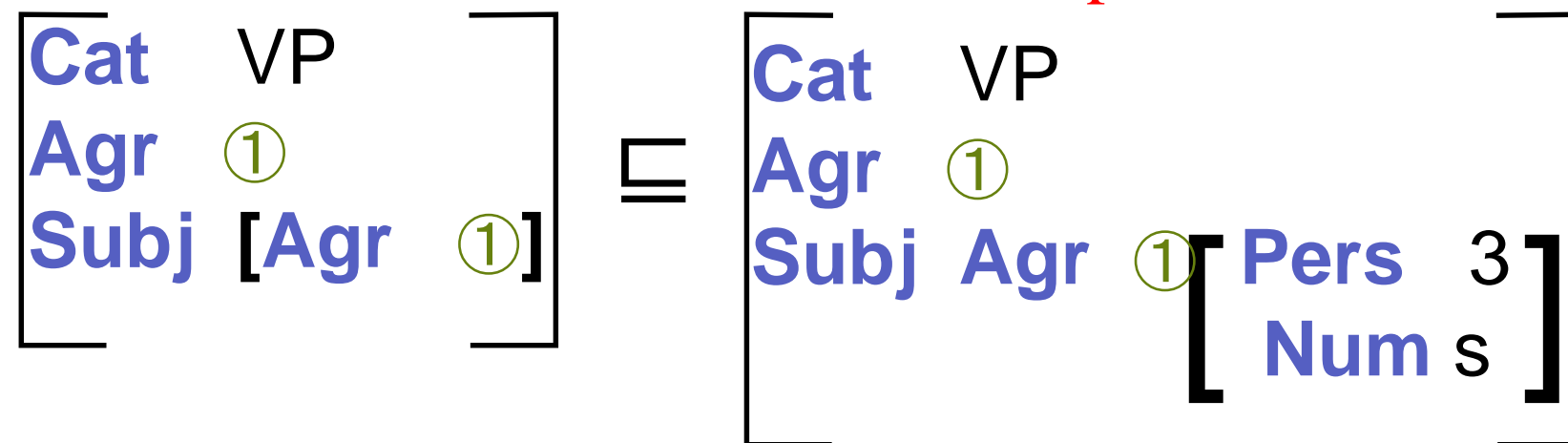
- Examples:



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



# Unification 1

- The *unification* of  $X$  and  $Y$  ( $X \sqcup Y$ ) is the most general feature structure  $Z$  that is subsumed by both  $X$  and  $Y$ .
- $Z$  is the smallest feature structure that extends both  $X$  and  $Y$ . <https://powcoder.com>
- Unification is a constructive operation.
  - If any feature values in  $X$  and  $Y$  are incompatible, it fails.
  - Else it produces a feature structure that includes all the features in  $X$  and all the features in  $Y$ .

# Unification 2

Assignment Project Exam Help  
<https://powcoder.com>  
Add WeChat powcoder

$$\begin{bmatrix} \text{Cat} & N \\ \text{Pers} & 3 \\ \text{Num} & s \end{bmatrix} \cup \begin{bmatrix} \text{Cat} & N \\ \text{Pers} & 3 \\ \text{Gndr} & F \end{bmatrix} = \begin{bmatrix} \text{Cat} & N \\ \text{Pers} & 3 \\ \text{Num} & s \\ \text{Gndr} & F \end{bmatrix}$$



# Features in chart parsing

- Each constituent has an associated feature structure.
  - Constituents with children have a feature structure for each child.
- Arc addition: <https://powcoder.com>
  - The feature structure of the new arc is initialized with all known constraints.
- Arc extension:
  - The feature structure of the predicted constituent must unify with that of the completed constituent extending the arc.

# Sample grammar fragment

$S \rightarrow NP VP$

$(NP \text{ Agr}) = (VP \text{ Agr})$

$NP \rightarrow Det N$

$(NP \text{ Agr}) = (N \text{ Agr})$

$(Det \text{ Agr}) = (N \text{ Agr})$

$VP \rightarrow V$

$(VP \text{ Agr}) = (V \text{ Agr})$

$Det \rightarrow a$

$Det \rightarrow all$

$Det \rightarrow the$

$[Agr \ 3s]$

$[Agr \ 3p]$

$[Agr \ \{3s, 3p\}]$

$N \rightarrow dog$

$N \rightarrow dogs$

$[Agr \ 3s]$

$[Agr \ 3p]$

$V \rightarrow sleep$

$V \rightarrow sleeps$

$[Agr \ ^3s]$

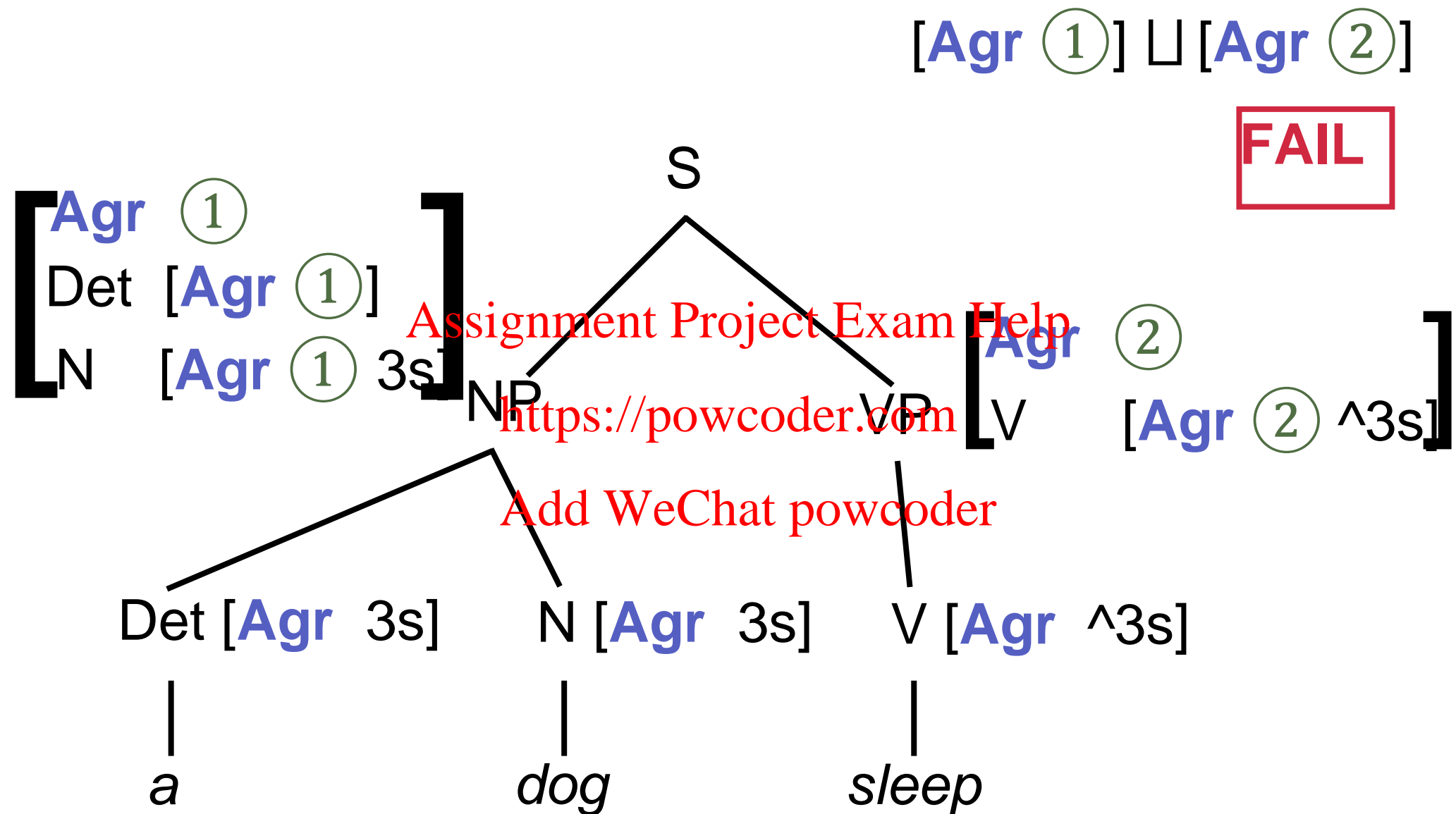
$[Agr \ 3s]$

Assignment Project Exam Help

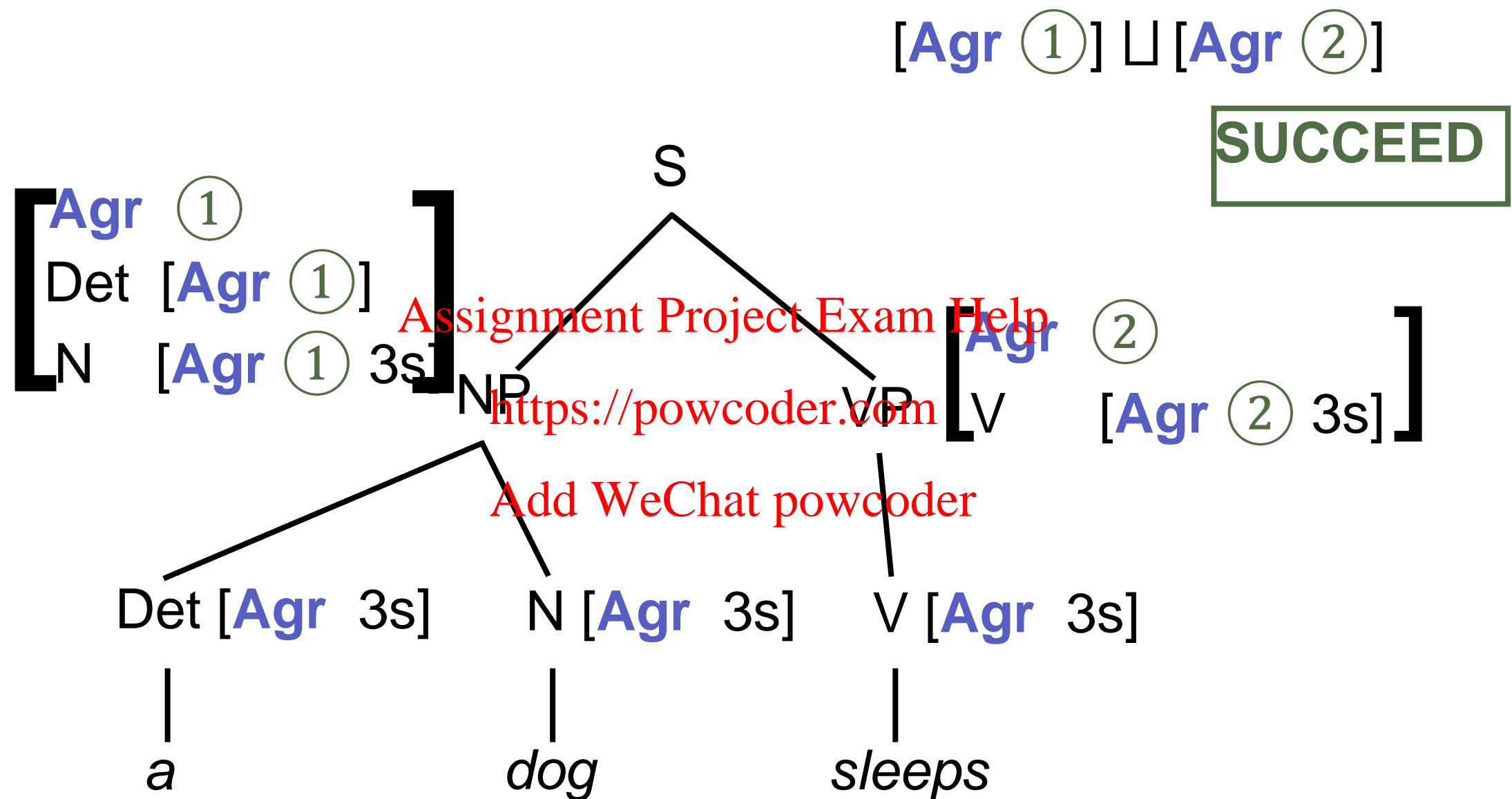
<https://powcoder.com>

Add WeChat powcoder

# Mismatched features fail



# Unifiable features succeed



# Advantages of this approach

- Distinguishes structure from "functional" info.
- Allows for economy of specification:
  - Equations in rules:  
$$S \rightarrow NP VP$$
$$(NP \text{ Agr}) = (VP \text{ Agr})$$

Assignment Project Exam Help  
<https://powcoder.com>  
Must unify with
  - Sets of values in lexicon:  
$$N \rightarrow \textit{fish}$$
$$(N \text{ Agr} \{3s, 3p\})$$

Add WeChat powcoder
- Allows for indirect specification and transfer of information, e.g., head features.

# Features and the lexicon

- Lexicon may contain each inflected form.
  - Feature values and base form listed.
- Lexicon may contain only base forms.
  - Process of *morphological analysis* maps inflected form to base form plus feature values.
  - Time–space trade-off, varies by language.
- Lexicon may contain *semantics* for each form.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Morphological analysis

- Morphological analysis is simple in English.
- Reverse the rules for inflections, including spelling changes.

*dogs* → *dog* [**Agr** 3p]      *eats* → *eat* [**Agr** 3s, **Tns** pres]  
*dog* → *dog* [**Agr** 3s]      *ripped* → *rip* [**Tns** past]  
*berries* → *berry* [**Agr** 3p]      *tarried* → *tarry* [**Tns** past]  
*buses* → *bus* [**Agr** 3p]      *running* → *run* [**Tns** pp]

- Irregular forms will always have to be explicitly listed in lexicon.

*children* → *child* [**Agr** 3p]      *sang* → *sing* [**Tns** past]

# Morphology in other languages

- Rules may be more complex in other (even European) languages.
- Languages with compounding (e.g., German) or agglutination (e.g., Finnish) require more-sophisticated methods.  
<https://powcoder.com>
- E.g., *Verdauungsspaziergang*, a stroll that one takes after a meal to assist in digestion.  
[Add WeChat powcoder](#)



# Semantics as a lexical feature

- Add a **Sem** feature:

<b>Cat</b>	N	
<b>Num</b>	s	
<b>Pers</b>	3	
<b>Lex</b>	<i>dog</i>	
<b>Sem</b>	<i>dog</i>	

Assignment Project Exam Help  
Typewriter font  
for semantic objects  
<https://powcoder.com>  
Add WeChat powcoder

- The meaning of *dog* is *dog*.  
The meaning of *chien* and *Hund* are both *dog*.  
The meaning of *dog* is G52790.

# Verb subcategorization 1

- **Problem:** Constraints on verbs and their complements.

*Nadia told / instructed / \*said / \*informed Ross to sit down.*

*Nadia \*told / \*instructed / said / \*informed to sit down.*

*Nadia told / \*instructed / \*said / informed Ross of the requirement to sit down.*

*Nadia gave / donated her painting to the museum.*

*Nadia gave / \*donated the museum her painting.*

*Nadia put / ate the cake in the kitchen.*

*Nadia \*put / ate the cake.*

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Verb subcategorization 2

- VPs are much more complex than just V with optional NP and/or PP.
- Can include more than one NP.
- Can include clauses of various types:  
*that Ross fed the marmoset  
to pay him the money*
- **Subcat**: A feature on a verb indicating the kinds of verb phrase it allows:  
\_np, \_np\_np, \_inf, \_np\_inf, ...

Write this way to distinguish from constituents.

# Verb tense and aspect 1

- **Tense and aspect** markings on verb:
  - Locate the event in time (relative to another time).
  - Mark the event as complete/finished or in progress.

*Nadia rides the horse.* — In progress now.

*Nadia rode the horse.* — Completed before now.

*Nadia had ridden the horse.* — Completed before before now.

*Nadia was riding the horse.* — In progress before now.

⋮

# Verb tense and aspect 2

- Tense: past or present
- Aspect: simple, progressive, or perfect

<i>Nadia ...</i>		Assignment Project Exam Help <a href="https://powcoder.com">https://powcoder.com</a> Add WeChat powcoder		
		In progress		
		Complete		
		Auxiliary verb		
		Simple	Progressive	Perfect
Present	<i>rides</i>	<i>is riding</i>	<i>has ridden</i>	
Past	<i>rode</i>	<i>was riding</i>	<i>had ridden</i>	
		Present participle	Past participle	<i>... the horse</i>

# Verb tense and aspect 3

- Tense: past or present
- Aspect: simple, progressive, or perfect

<i>Nadia ...</i>		<div>Auxiliary verbs</div>	
	<b>Simple</b>	<b>Perfect progressive (continuous)</b>	
<b>Present</b>	<i>rides</i>	<i>has been riding</i>	
<b>Past</b>	<i>rode</i>	<i>had been riding</i>	
<i>... the horse</i>			

# Modal verbs

- **Modal verbs:** Auxiliary verbs that express degrees of certainty, obligation, possibility, prediction, etc.

*Nadia*

Assignment Project Exam Help

{*could, should, must, ought to, might, will, ...*}

{*ride, be riding, have ridden, have been riding*}  
*the horse.*

# English auxiliary system

- Structure (so far):  
[MODAL] [HAVE] [BE] MAIN-VERB

- General pattern:

VP → AUX VP

AUX → MODAL | HAVE | BE

- Use features to capture necessary agreements.



# Voice 1

*The goalie kicked the ball.*

**ACTIVE**

Assignment Project Exam Help

Event: *kicked*

<https://powcoder.com>

Add WeChat powcoder

Role: Agent  
(doer)

Role: Theme  
(thing affected)

Thing: *the goalie*    Thing: *the ball*

kick (agent=goalie, theme=ball)

# Voice 2

*The ball was kicked.*

**PASSIVE**

Assignment Project Exam Help

Event: *kicked*

<https://powcoder.com>

Add WeChat powcoder

Role: Theme  
(thing affected)

Thing: *the ball*

kick (agent=?, theme=ball)

# Voice 3

*The ball was kicked by the goalie.* **PASSIVE**

Assignment Project Exam Help

Event: *kicked*

<https://powcoder.com>

Add WeChat powcoder

Role: Theme  
(thing affected)

Role: Agent  
(doer)

Thing: *the ball*    Thing: *the goalie*

kick (agent=goalie, theme=ball)

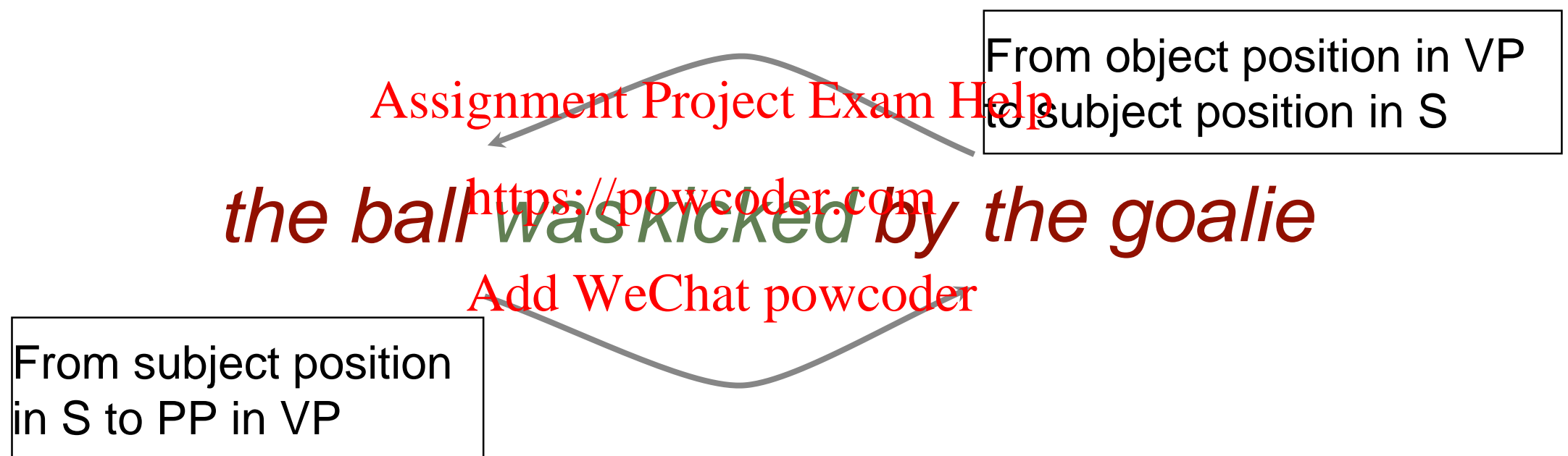
# Passive as *Diathetic alternation*

Assignment Project Exam Help

*the goalie* <https://powcoder.com> *kicked* *the ball*

Add WeChat powcoder

# Passive as *Diathetic alternation*



But the semantic representation doesn't change

# Voice 4

- **Voice:** System of assigning thematic roles to syntactic positions.
- English has **active** and **passive** voices.

Assignment Project Exam Help

- Passive expressed with **be+past participle**.

<https://powcoder.com>

Other auxiliaries may also apply, including progressive *be*.

Add WeChat powcoder

- *Nadia was kissed.*      *Nadia was being kissed.*  
*Nadia had been kissed.*      *Nadia had been being kissed.*  
*Nadia could be kissed.*      *Nadia could have been being kissed.*

- Structure:  
[MODAL] [HAVE] [BE<sub>1</sub>] [BE<sub>2</sub>] MAIN-VERB

# Some useful features

- **VForm**: The tense/aspect form of a verb:  
passive, pastprt, ...
- **CompForm**: The tense/aspect form of the  
complement of an auxiliary.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Augmenting rules for passive voice

- For all rules of the form:

$VP \rightarrow V \ NP \ X$

$(V \text{ Subcat}) = \_y$

ADD



$VP \rightarrow V \ X$

$(V \text{ Subcat}) = \_y$

$(V \text{ VForm}) = \text{passive}$

$(VP \text{ VForm}) = \text{passive}$

Metarule to ease grammar coding

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

- Augment Aux+VP rules:

$VP \rightarrow AUX \ VP$

$(AUX \text{ Root}) = \text{Be2}$

$(AUX \text{ CompForm}) = (VP_2 \text{ VForm})$

$(VP_2 \text{ VForm}) = \text{passive}$



# The GAP feature for passive voice

S → NP VP

- <sup>1</sup> (NP **Agr**) = (VP **Agr**)
- <sup>2</sup> (VP **VForm**) = passive
- <sup>3</sup> (VP **Gap Cat**) = NP
- <sup>4</sup> (VP **Gap Agr**) = (NP **Agr**)
- <sup>5</sup> (VP **Gap Sem**) = (NP **Sem**)

VP → AUX VP

- <sup>1</sup> (VP<sub>1</sub> **Agr**) = (AUX **Agr**)
- <sup>2</sup> (VP<sub>1</sub> **VForm**) = (VP<sub>2</sub> **VForm**)
- <sup>3</sup> (VP<sub>1</sub> **Gap**) = (VP<sub>2</sub> **Gap**)
- <sup>4</sup> (AUX **Lex**) = be2
- <sup>5</sup> (VP<sub>2</sub> **VForm**) = passive

V → *kicked*

- <sup>1</sup> (V **VForm**) = {pastprt, passive}
- <sup>2</sup> (V **Subcat**) = \_np
- <sup>3</sup> (V **Lex**) = *kick*
- <sup>4</sup> (V **Sem**) = *kick*

VP → V NP

- <sup>1</sup> (VP **VForm**) = (V **VForm**)
- <sup>2</sup> (VP **Gap**) = (NP **Gap**)
- <sup>3</sup> (V **Subcat**) = \_np

NP → ε

- <sup>1</sup> (NP **Gap Cat**) = NP
- <sup>2</sup> (NP **Gap Agr**) = (NP **Agr**)
- <sup>3</sup> (NP **Gap Sem**) = (NP **Sem**)

NP → *cans*

- <sup>1</sup> (NP **Agr**) = 3p
- <sup>2</sup> (NP **Lex**) = *can*
- <sup>3</sup> (NP **Sem**) = *cans*

AUX → *were*

- <sup>1</sup> (AUX **Agr**) = 3p
- <sup>2</sup> (AUX **Lex**) = be2

Assignment Project Exam Help

Empty string

<https://powcoder.com>

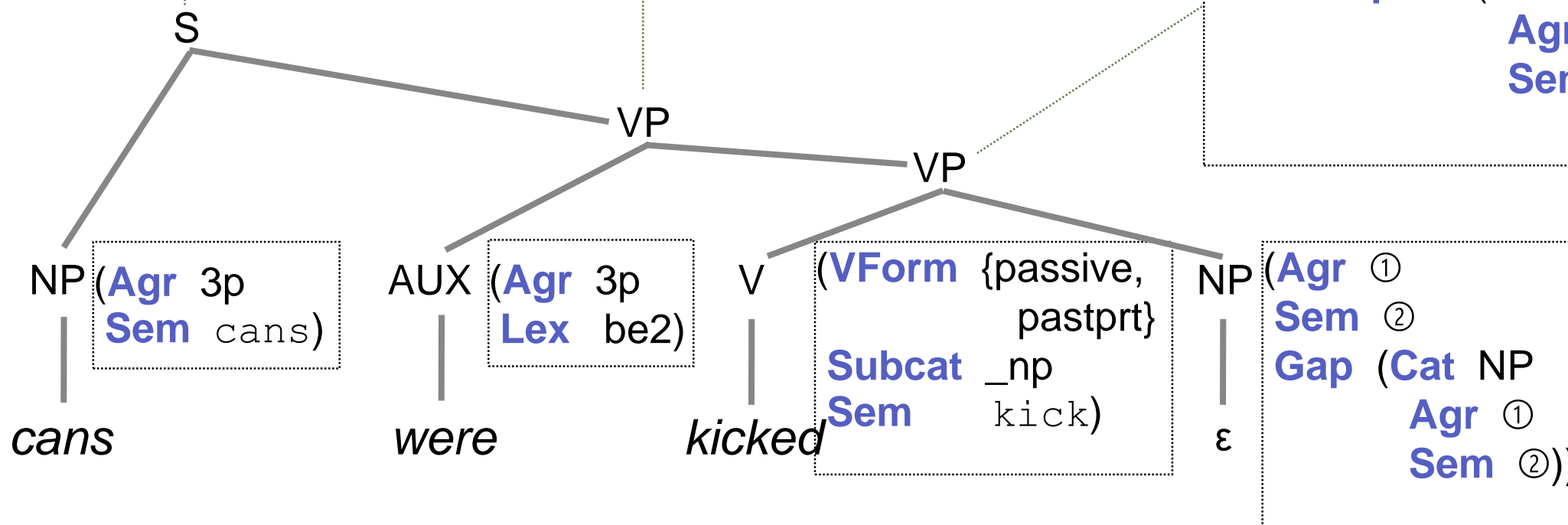
Add WeChat powcoder

(NP (Agr ① 3p  
Sem ② cans )  
VP (Agr ①  
VForm ③  
Gap ④  
AUX (Agr ① 3p  
Lex be2)  
VP (VForm ③  
Gap ④  
V (VForm ③ passive  
Subcat \_np  
Sem kick)  
NP (Agr ①  
Sem ②  
Gap ④ (Cat NP  
Agr ①  
Sem ②))))

(Agr ⑤  
VForm ③  
Gap ④  
AUX (Agr ⑤ 3p  
Lex be2)  
VP (VForm ③  
Gap ④  
V (VForm ③ passive  
Subcat \_np  
Sem kick)  
NP (Agr ①  
Sem ②  
Gap ④ (Cat NP  
Agr ①  
Sem ②))))

Note: The green ①'s of the S were ⑤'s until the 4th constraint of the rule  $S \rightarrow NP VP$ . The 5th constraint fills in the Sem of the Gap ②.

(VForm ③  
Gap ④  
V (VForm ③ {passive,  
pastprt}  
Subcat \_np  
Sem kick)  
NP (Agr ①  
Sem ②  
Gap ④ (Cat NP  
Agr ①  
Sem ②))))



# Other cases of *gap percolation*

- Other constructions involve NPs in syntactic configurations where they would not get the right thematic roles using linear order alone.

*Nadia seems to like Ross.*

*Nadia seems to be liked.*

*Nadia is easy to like.*

*Who did Nadia like?*

*I fed the dog that Nadia likes to walk.*

- Can use grammar rules with gap features to ensure correct structure/interpretation of these as well.

# Summary

- Features help capture syntactic constructions in a general and elegant grammar.
- Features can encode the compositional semantics of a sentence as you parse it.
- Features can accomplish mapping functions between syntax and semantics that simplify the interpretation process.