

# Derivation

Prof. Joseph Pentangelo

LING 350: The Structure of Words

7 March 2024

# Research Volunteers Needed!

The Speech Laboratory at CSI is in need of monolingual native English speakers for a research study

Are you a native English speaker between the ages of 18 and 65? Are you free of any history of speech, language or hearing disorder? If so, you may be eligible to participate in a language research study in the Linguistics Program at the College of Staten Island!

---

Participation takes place in the Speech Laboratory (Building 2S) and requires approximately 60 to 90 minutes to complete. The study involves listening to recordings of spoken language, as well as completing questionnaires.

---

Participants who complete the study receive \$40 for their time

**If you are interested, please contact Professor Jason Bishop at [jason.bishop@csi.cuny.edu](mailto:jason.bishop@csi.cuny.edu) to get more information, or to schedule a time to participate**



[lɪŋˈɡwɪstɪks]

## Linguistics Tutoring

### On-campus Tutoring

Monday: 3 PM – 9 PM

Tuesday: 9:30 AM – 12:30 PM

Location: Bldg. 1L (Library), Rm. 117

### Online Tutoring

Tuesday: 10 AM – 12 PM

Saturday: 10 AM – 2 PM

To access our tutoring schedule online, please visit  
[csi.cuny.edu/CASATutoring](http://csi.cuny.edu/CASATutoring)

Linguistics tutoring offered by the  
Office of Academic Support  
718.982.4221

# What we'll cover:

- What derivation is and how it works
- Writing derivational templates
- The Right-hand Head Rule
- Next week, we'll go over Quiz #2 and WA#1.

Quiz #2 is due right now!!!

If you forgot about it, do it by midnight tonight.

# Derivation

- What is the basic function of derivation?

“The basic function of derivational processes is to enable the language user to make new lexemes.” (p.51)

- What are lexical categories?
- What’s an open class?
- What’s a closed class?
- What are categorial shifts?

Is this class open or closed:

Adjective

Verb

Preposition

Adverb

Noun

Conjunction

Determiner

Pronoun

# Inputs and outputs

- We can think of derivation as a factory.
- There are different factories for different morphological operations.
- Each different operation – each suffix and each prefix – has its own factory.
- Each factory has rules about what kind of lexemes they take as **inputs**, and specifications about what kind of lexemes they deliver as **outputs**.

This is the *-able* ‘able to be Xed’ factory.

It can take “believe” as an input, and spit out “believable” as an output.

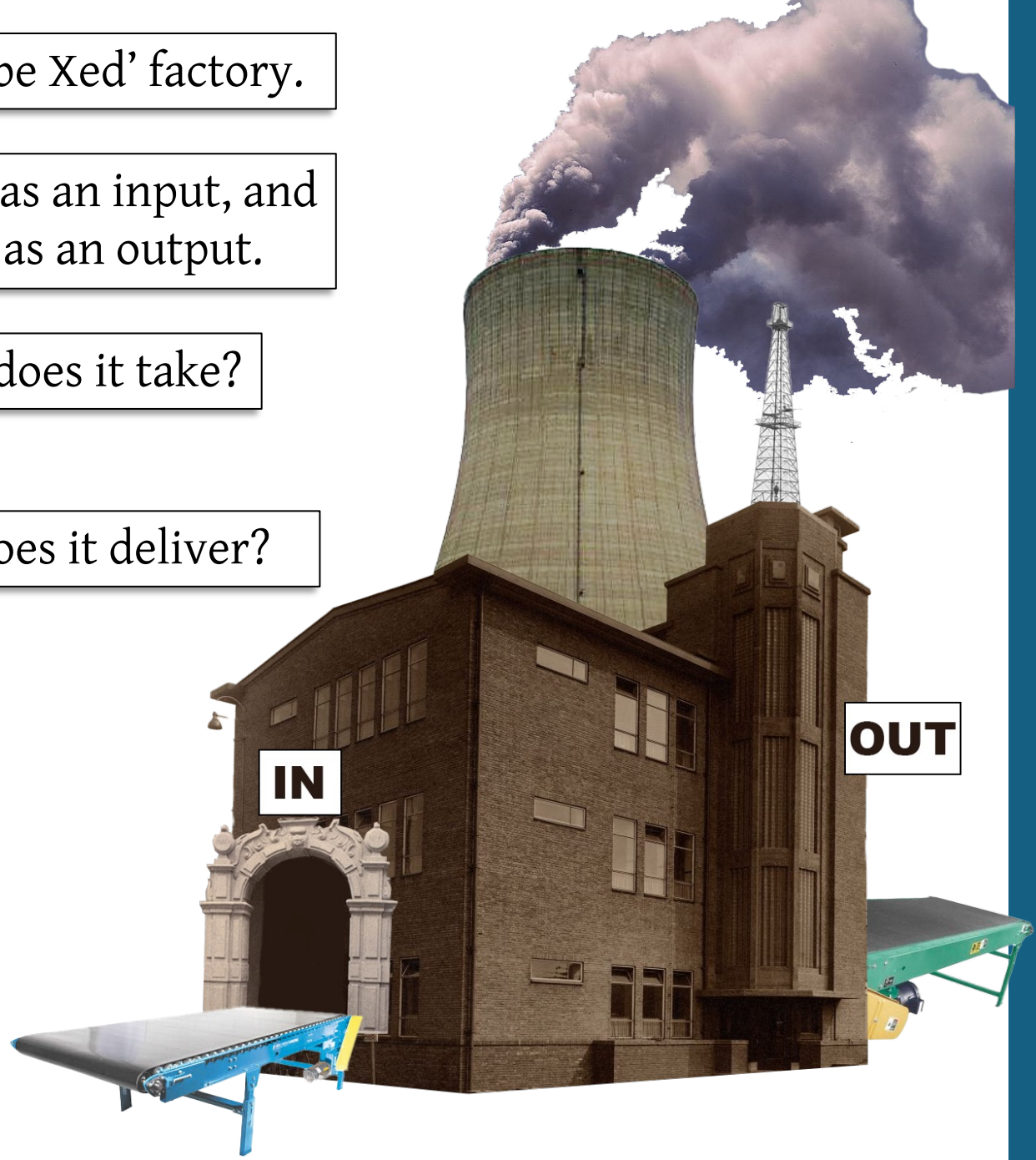
What kind of inputs does it take?

verbs

What kind of outputs does it deliver?

adjectives

believe



[[believe]<sub>V</sub> able]<sub>A</sub>

Note that this is a **paradigmatic** approach. It is oriented around processes.



This is the *-er* 'one who Xes' factory.

It can take "drive" as an input, and spit out "driver" as an output.

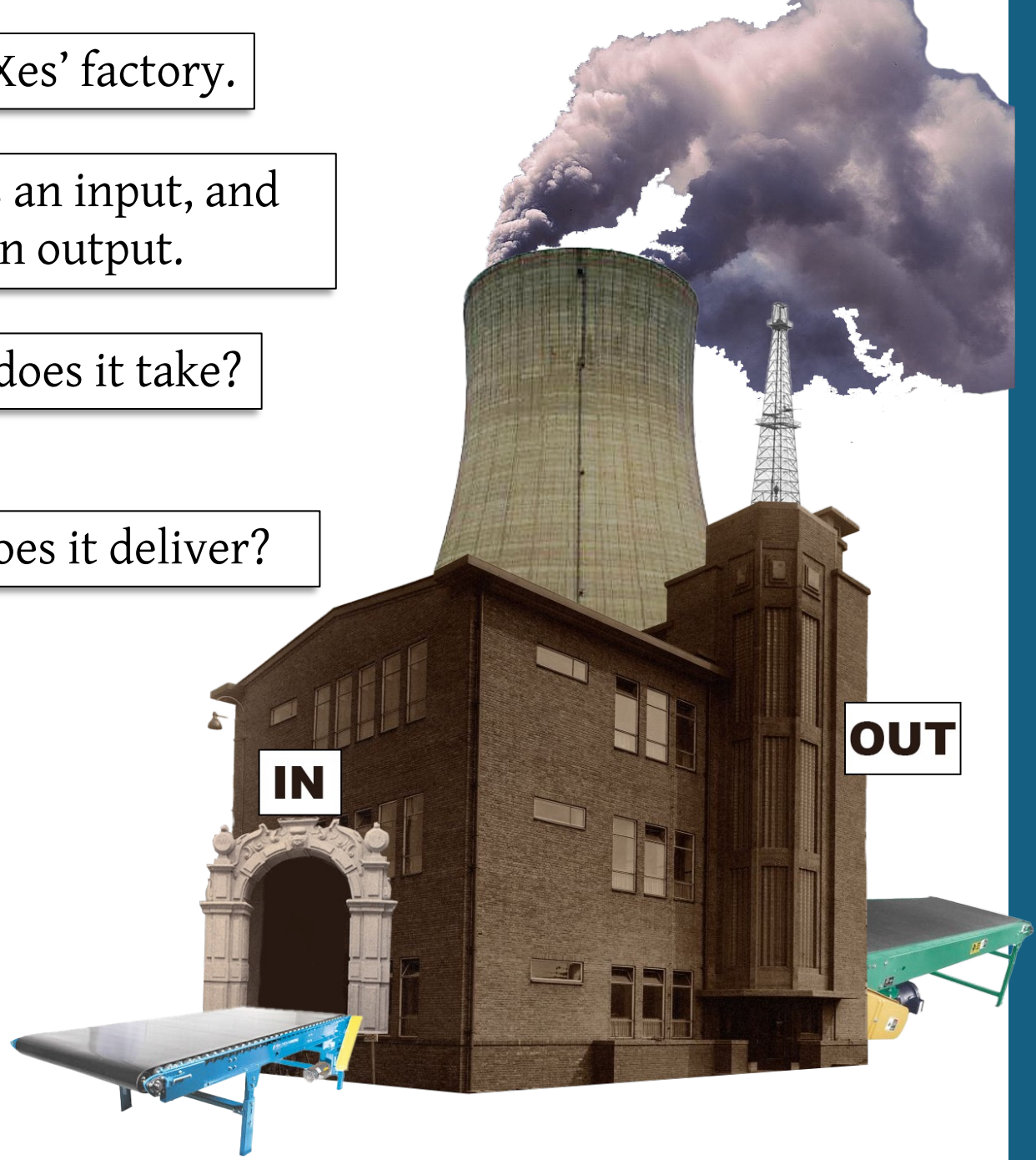
What kind of inputs does it take?

verbs

What kind of outputs does it deliver?

nouns

drive



[[drive]<sub>V</sub> er]<sub>N</sub>



# Templates

- We can write templates to illustrate derivational processes.
- Where the affix does determine the lexical category, like the English nominalizer *-ity*, we can write...

$[[X]_A \text{ ity}]_N$  ‘the state, condition, or quality of being X’

e.g.  $\text{stupid}_A \rightarrow \text{stupidity}$  ‘the state of being stupid’

$\text{rapid}_A \rightarrow \text{rapidity}_N$  ‘the quality of being rapid’

$\text{absurd}_A \rightarrow \text{absurdity}_N$  ‘the state of being absurd’

What’s the lexical category of this process’s input?

What’s the lexical category of this process’s output?

# Templates

- Some affixes allow for multiple kinds of inputs.
- One example is the Dutch diminutive *-tje*.
- For this, we can write...

[[X]<sub>Y</sub> tje]<sub>N</sub> ‘small entity with ~~X~~-related property’ where Y = N, A, V

e.g. *blond*<sub>A</sub> ‘blond’ → *blondje*<sub>N</sub> ‘blond girl’  
*speel*<sub>V</sub> ‘play’ → *speeltje*<sub>N</sub> ‘toy’

What’s the lexical category of this process’s input?

What’s the lexical category of this process’s output?

# Derivation: Lexeme formation

- Where the affix does not determine the lexical category, like the Dutch negator *on-*, we can write...

$[on[X]_Y]_Y$  'not-Y', where  $Y = N, A$

e.g.  $gewoon_A$  'common'  $\rightarrow$   $ongewoon_A$  'uncommon'

$zin_N$  'sense'  $\rightarrow$   $onzin_N$  'nonsense'

What's the lexical category of this process's input?

What's the lexical category of this process's output?

What does *prison* mean?

What does *jail* mean?

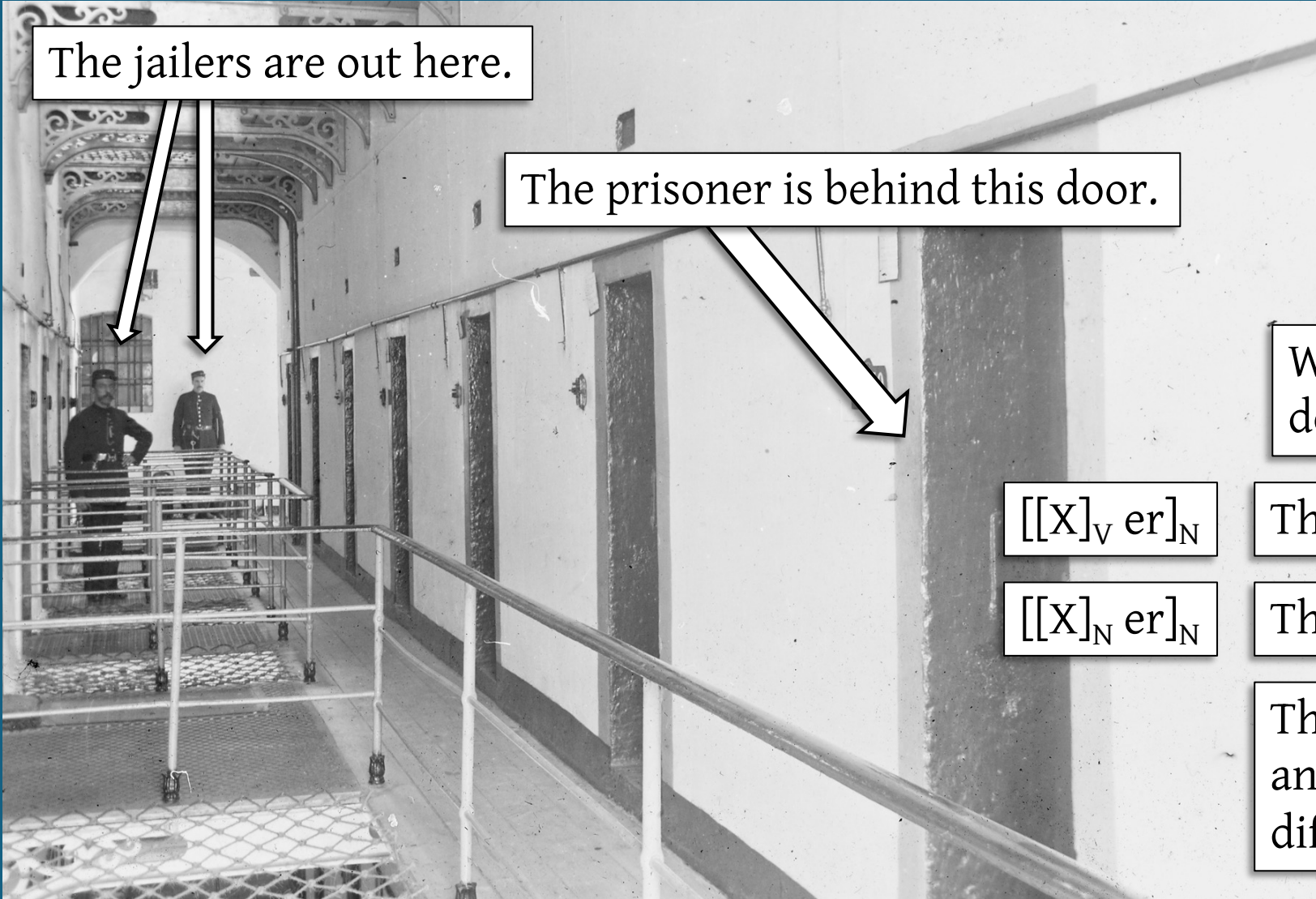
What does *prisoner* mean?

What does *jailer* mean?

Why do *prisoner* and *jailer* mean opposite things, even though *jail* and *prison* are so similar?

The jailers are out here.

The prisoner is behind this door.



Consider these two paradigms:

1. jailer  
baker  
dancer

2. prisoner  
banker  
high-schooler

What's different about the derivational process in (1) vs in (2)?

$[[X]_V \text{er}]_N$

The *-er* in (1) takes verbs as its input.

$[[X]_N \text{er}]_N$

The *-er* in (2) takes nouns as its input.

They are two different processes and the resulting outputs have different meanings.

---

### Derivation of nouns

---

|       |             |                    |                           |
|-------|-------------|--------------------|---------------------------|
| A → N | suffixation | schoon “beautiful” | schoon-heid “beauty”      |
| V → N | suffixation | spreek “to speak”  | sprek-er “speaker”        |
|       | prefixation | praat “to talk”    | ge-praat “talking”        |
| N → N | suffixation | moeder “mother”    | moeder-schap “motherhood” |
|       | prefixation | zin “sense”        | on-zin “nonsense”         |

---

### Derivation of adjectives

---

|       |             |                  |                         |
|-------|-------------|------------------|-------------------------|
| N → A | suffixation | meester “master” | meester-lijk “masterly” |
| V → A | suffixation | lees “to read”   | lees-baar “readable”    |
| A → A | suffixation | blauw “blue”     | blauw-ig “blueish”      |
|       | prefixation | gewoon “common”  | on-gewoon “uncommon”    |

---

### Derivation of verbs

---

|       |             |                          |                                      |
|-------|-------------|--------------------------|--------------------------------------|
| N → V | suffixation | analyse “analysis”       | analys-eer “to analyse”              |
|       | prefixation | slaaf “slave”            | ver-slaaf “to enslave”               |
| A → V | suffixation | kalm “calm”              | kalm-eer “to calm down”              |
|       | prefixation | bleek “pale”             | ver-bleek “to turn pale”             |
| V → V | suffixation | <u>krab</u> “to scratch” | <u>krabb-el</u> “to scratch lightly” |
|       | prefixation | rijd “to ride”           | be-rijd “to ride on”                 |

---

Let's make a template for *-el*

What's the input?

V only

What's the output?

V only

$[[X]_V \text{el}]_V$





---

### *Derivation of nouns*

---

|       |             |                    |                           |
|-------|-------------|--------------------|---------------------------|
| A → N | suffixation | schoon “beautiful” | schoon-heid “beauty”      |
| V → N | suffixation | spreek “to speak”  | sprek-er “speaker”        |
|       | prefixation | praat “to talk”    | ge-praat “talking”        |
| N → N | suffixation | moeder “mother”    | moeder-schap “motherhood” |
|       | prefixation | zin “sense”        | on-zin “nonsense”         |

---

### *Derivation of adjectives*

---

|       |             |                  |                         |
|-------|-------------|------------------|-------------------------|
| N → A | suffixation | meester “master” | meester-lijk “masterly” |
| V → A | suffixation | lees “to read”   | lees-baar “readable”    |
| A → A | suffixation | blauw “blue”     | blauw-ig “blueish”      |
|       | prefixation | gewoon “common”  | on-gewoon “uncommon”    |

---

### *Derivation of verbs*

---

|       |             |                    |                               |
|-------|-------------|--------------------|-------------------------------|
| N → V | suffixation | analyse “analysis” | analys-eer “to analyse”       |
|       | prefixation | slaaf “slave”      | ver-slaaf “to enslave”        |
| A → V | suffixation | kalm “calm”        | kalm-eer “to calm down”       |
|       | prefixation | bleek “pale”       | ver-bleek “to turn pale”      |
| V → V | suffixation | krab “to scratch”  | krabb-el “to scratch lightly” |
|       | prefixation | rijd “to ride”     | be-rijd “to ride on”          |

---

Let's make the template for *on-*

What's the input?

N and A

What's the output?

N and A

Whatever the input is, the output matches its lexical category.

$[on[X]_Y]_Y$

Where Y is N or A.

---

### *Derivation of nouns*

---

|       |             |                    |                           |
|-------|-------------|--------------------|---------------------------|
| A → N | suffixation | schoon “beautiful” | schoon-heid “beauty”      |
| V → N | suffixation | spreek “to speak”  | sprek-er “speaker”        |
|       | prefixation | praat “to talk”    | ge-praat “talking”        |
| N → N | suffixation | moeder “mother”    | moeder-schap “motherhood” |
|       | prefixation | zin “sense”        | on-zin “nonsense”         |

---

### *Derivation of adjectives*

---

|       |             |                  |                         |
|-------|-------------|------------------|-------------------------|
| N → A | suffixation | meester “master” | meester-lijk “masterly” |
| V → A | suffixation | lees “to read”   | lees-baar “readable”    |
| A → A | suffixation | blauw “blue”     | blauw-ig “blueish”      |
|       | prefixation | gewoon “common”  | on-gewoon “uncommon”    |

---

### *Derivation of verbs*

---

|       |             |                    |                               |
|-------|-------------|--------------------|-------------------------------|
| N → V | suffixation | analyse “analysis” | analys-eer “to analyse”       |
|       | prefixation | slaaf “slave”      | ver-slaaf “to enslave”        |
| A → V | suffixation | kalm “calm”        | kalm-eer “to calm down”       |
|       | prefixation | bleek “pale”       | ver-bleek “to turn pale”      |
| V → V | suffixation | krab “to scratch”  | krabb-el “to scratch lightly” |
|       | prefixation | rijd “to ride”     | be-rijd “to ride on”          |

---

# Practice

Group 1: Write templates for *-heid*, *-er*, and *-lijk*

Group 2: Write templates for *-schap*, *-baar*, and *ver-*

Group 3: Write templates for *-ig*, *-eer*, and *be-*



-heid

$[X]_A \text{heid}]_N$

-er

$[X]_V \text{er}]_N$

-lijk

$[X]_N \text{lijk}]_A$

-schap  
~~-shap~~

$[X]_N \text{ schap}]_N$

-baar

$[X]_V \text{ baar}]_A$

ver-

$FN, A [ver [X]_V]_V$

-ig

[~~ix~~]A ɪŋA

-eer

[~~ix~~]y eer) vɪx= 2, A

be-

[be [~~x~~]<sub>v</sub>]<sub>v</sub>



Now let's do the same with some English affixes.

group one:  
-ness  
wickedness

-ful  
beautiful

-ship  
mentorship

group two:  
-ish  
childish

pre-  
premature

-ion  
action

group two:  
un-  
unhappy

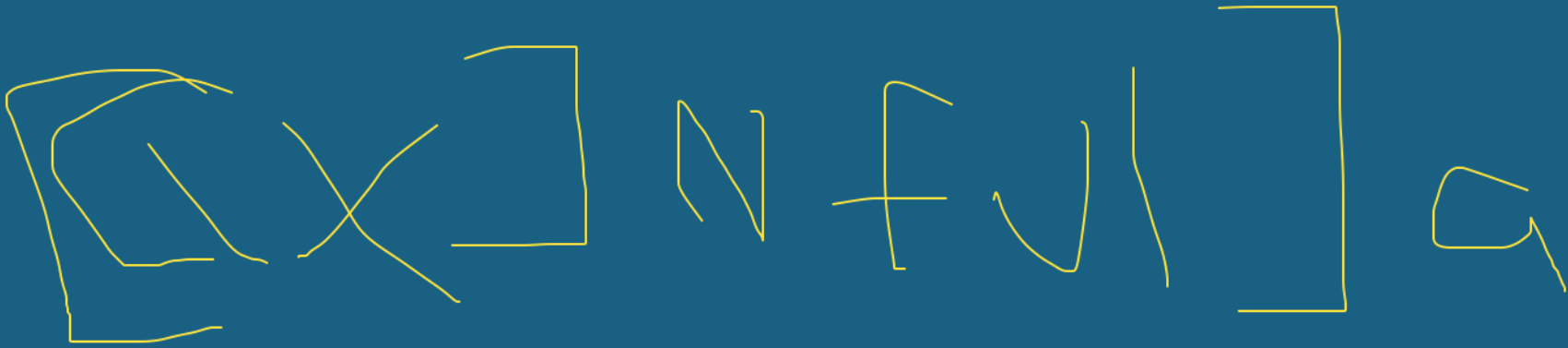
un-  
unzip

-en  
darken

-ness



-ful



-ship



-ish

$[[X]_N \text{ ish}]_A$

pre-

$[\text{pre } [X]_A]_A$

-ion

$[[X]_V \text{ ion}]_N$

un-



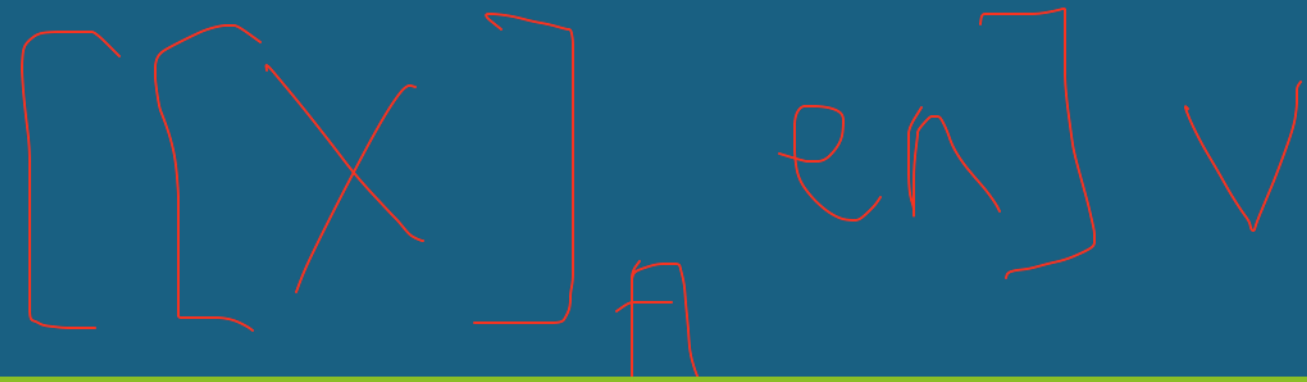
unhappy

un-



unzip

-en



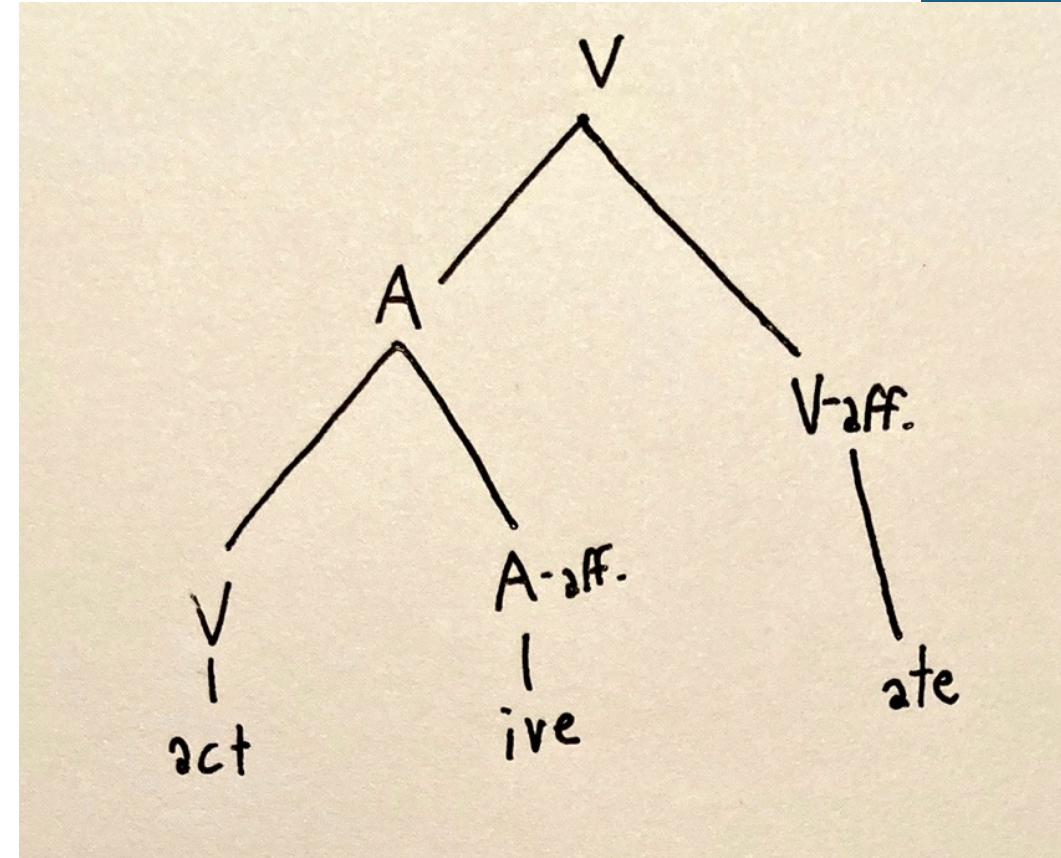
darken





# Right-hand Head Rule

- What is the Right-hand Head Rule?
- A head “determines the syntactic category” of the word.
- The RHR posits that in any given word, the rightmost morpheme is the head.
- The features of the head “percolate” up to apply to the whole word.
- It implies that there are no category-changing prefixes.



# Right-hand Head Rule

- In English, it works for compounds as well as derived words.
- *swim trunks, greenhouse, sea green, etc.*
- The RHR was proposed as a linguistic universal. It doesn't hold up in all cases.
- There are, in fact, category-changing prefixes, including in English.
- An example is  $[en [X]_Y]_V$  where  $Y = N, A$ .
- With noun inputs: *enmesh, enrobe, enthrone, enshrine, encourage...*
- With adj inputs: *enable, ensure, enrich, enfeeble...*

# For next week...

- Read the rest of this chapter (pp.61–73).