Assignment 3 Tutorial 1

Jinyue Feng

Overview

- **TRALE Basics**
- **Grammatical Gender Agreement**
- Subcategorization
- Passive Voice (Gap Constuct)
- Semantic Head
- Beta Reduction
- Quantifier Storage

Don't forget to check out the tutorial online!

Types + Lexicon + Rule = Your First Grammar

- Types
  - Define linguistic categories
  - Hierarchical: types → subtypes
- Lexicon
  - Define vocabulary
  - Each word is defined by its type (and feature)
- Rules
  - Define grammar
  - How do the elements of your grammar combine?

Types - Syntax

The bottom supertype

Defines subtypes

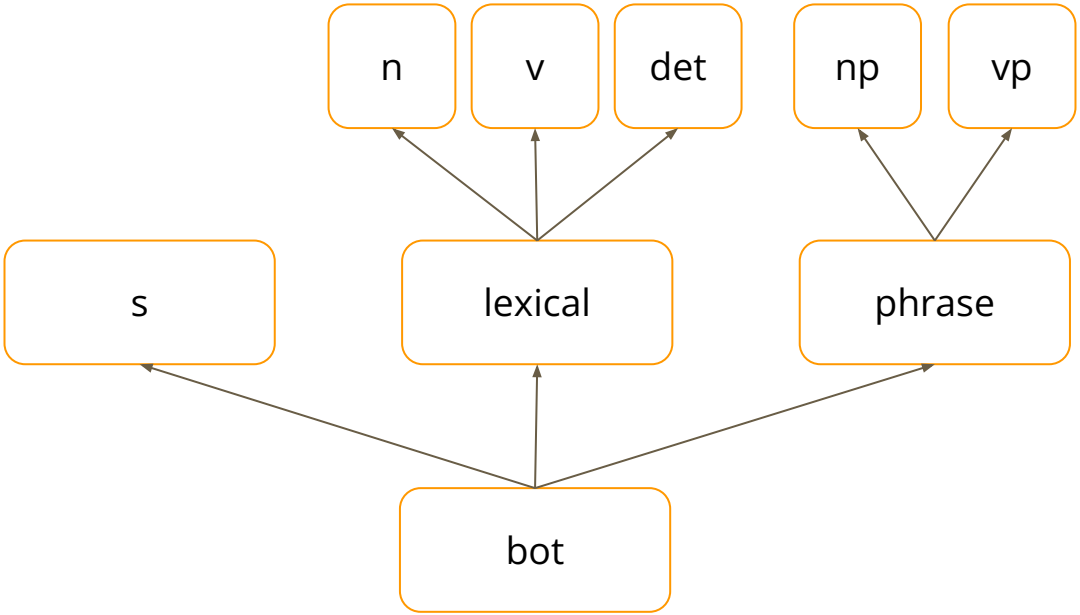
```
bot sub [s, lexical, phrase].

phrase sub [np, vp].

lexical sub  [n, v, det].
```

Don’t forget the period that ends a statement.

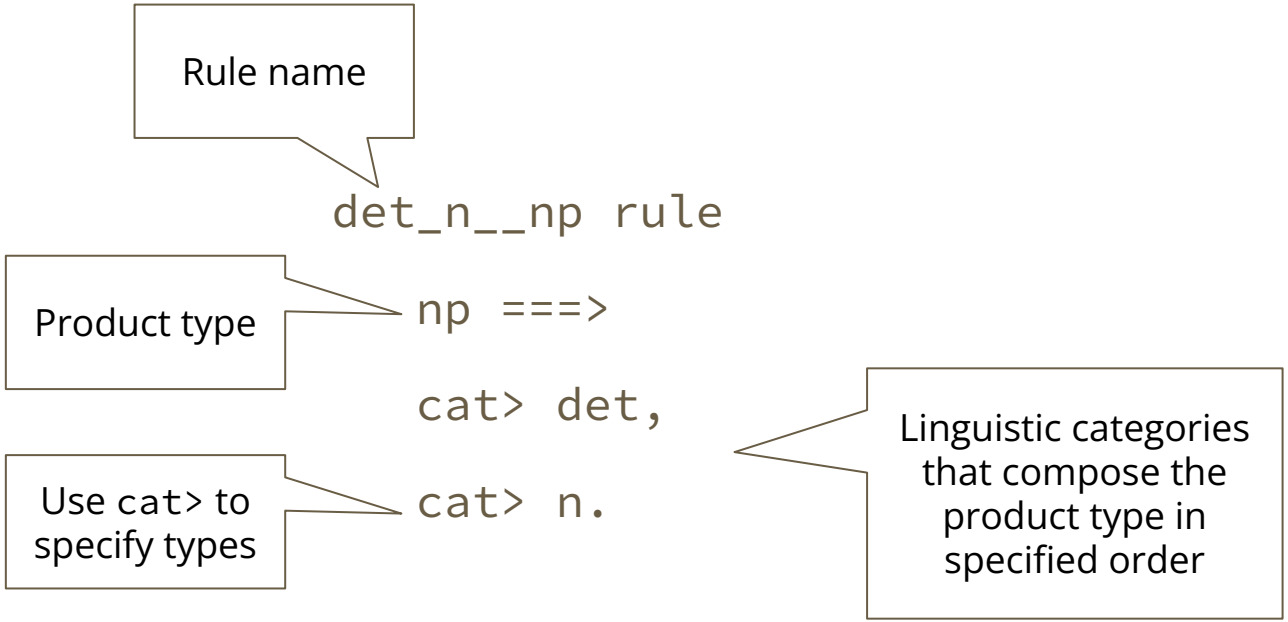
Types - Hierarchical Type Tree



Lexicon

```
dog ---> n.
cat ---> n.
the ---> det.
likes ---> v.
record ---> n.
record ---> v.
```

Rules



The rule “det_n__np” specifies that a noun phrase (np) can be composed by a determiner (det) and a noun (n).

Example: g1.pl

Tips

- Create an alias for easier usage
- Use the -fsug options to correctly start the system with graphical interface
- Use -c to compile your grammar file
- For easier testing:
  - Define a function in your grammar file and directly run your function

```
t :- rec[the,dog,likes,the,cat].
```

Introducing Features

How do we encode more complex linguistic phenomena, such as gender agreement?

- She is a waitress
- She is a waiter

Define features for your types!

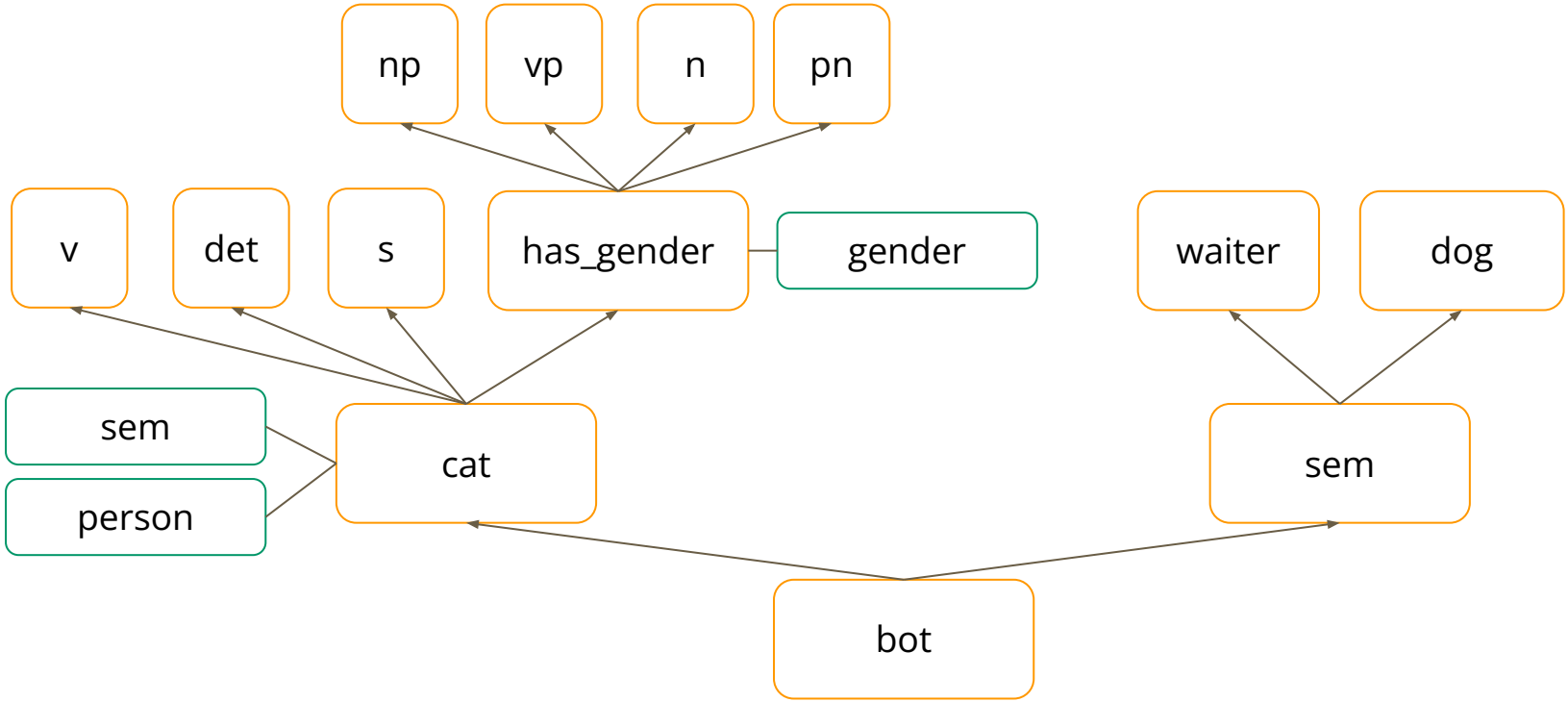
Type Feature Structure

Use the `intro` keyword to define features

The `sem` feature takes values from the `sem` type

```
bot sub [cat, sem].
    cat sub [v, det, s, has_gender] intro [sem:sem,person:per_type].
        has_gender sub [np, vp, n, pn] intro [gender:gen_type].
    gen_type sub [m, f, neu].
    per_type sub [first, second, third].
    sem sub [waiter, dog].
```

Types - Hierarchical Type Tree with features

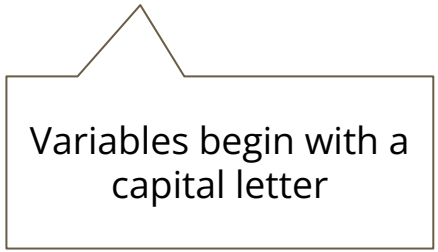


Lexicon with features

```
she ---> (pn, gender:f,person:third).
he ---> (pn, gender:m,person:third).
i ---> (pn, person:first).
is ---> (v,person:third).
the ---> (det).
waiter ---> (n, gender:m, sem:waiter, person:third).
waitress ---> (n, gender:f, sem:waiter, person:third).
dog ---> (n, gender:neu, sem:dog, person:third).
```

Rules with Features

```
det_n__np rule
   (np, gender:Gender, person:Person, sem:Sem) ===>
   cat> det,
   cat> (n, gender:Gender, person:Person, sem:Sem).
```



Features for Agreement in a Rule

Example g2.pl

Let’s fix the grammar to enforce correct gender agreement.

Questions?