

# Tree Adjoining Grammars: XMG (Session 2)

Laura Kallmeyer & Simon Petitjean

HHU Düsseldorf

WS 2015

20.11.2015

# Plan

- 1 Recap
- 2 Homework
- 3 Example: copy language
- 4 Example: English grammar

# Recap

- eXtensible MetaGrammar: metagrammar  $\rightarrow$  grammar
- MetaGrammar: abstract and compact description of the grammar
- Different dimensions for different levels of linguistic description
- **<syn>** dimension: describe trees using dominance and precedence constraints
- **<syn>** solver: find all minimal models for the description

## Useful commands

- To compile the metagrammar in file `mg_file.mg`:

```
xmg compile synsem mg_file.mg --force
```

The resulting grammar is produced in file `mg_file.xml`

- To load the tree viewer:

```
xmg gui tag
```

File → Load an XML file. Select the generated grammar file.

## Some other useful comments

- To get the latest version of XMG-NG: `bzr pull`

List of properties for nodes (TAG specific):

- `mark=subst`: substitution node (↓)
- `mark=foot`: foot node (\*)
- `mark=nadj`: forbidden adjunction
- `mark=anchor`: lexical insertion site (◇)
- `mark=flex`: lexical item (without anchoring)

Documentation: <http://dokufarm.phil.hhu.de/xmg>

## Example: copy language

- We described everything by hand, could we generalize anything?

## Example: English grammar

Let us now generate some trees for natural language (based on the XTAG grammar<sup>1</sup>).

- Create a class `alphanx0V` for intransitive verbs
- Create classes for `alphanx0Vnx1` (transitive verbs) and `alphaW1nx0Vnx1` (extracted objects)

---

<sup>1</sup>The trees you need to describe can be found in the XTAG manual:

<ftp://ftp.cis.upenn.edu/pub/xtag/release-2.24.2001/tech-report.ps>