

Using GF from Python

Jordi Saludes, UPC
MOLTO 1st meeting, Varna

Goal

Goal

- Part of WVP2

Goal

- Part of WVP2
- To use gf from the NLTK

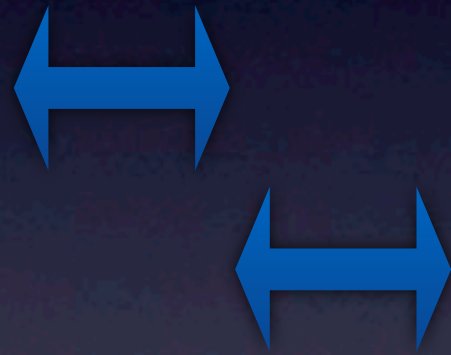
Goal

- Part of WVP2
- To use gf from the NLTK
- ...which is python-based

Goal

- Part of WVP2
- To use gf from the NLTK
- ...which is python-based
- Mainly STM



Plugin



Plugin



Plugin

- python  C
- FFI layer: C  Haskell

Plugin

- python  C
- FFI layer: C  Haskell
- Inspired by Kevin Kofler on **c-bindings**

Features

Features

- Reading pgf modules

Features

- Reading pgf modules
- Parsing/linearization

Features

- Reading pgf modules
- Parsing/linearization
- Breaking down the structure of a gf expression.

The C side

- GFpgf, GFLang, GFType, GFTree
- Each struct stores a **opaque reference** to a haskell structure.

The Haskell side

The Haskell side

- Storable class

The Haskell side

- Storable class
- [therning.org/ magnus](http://therning.org/magnus) » hsc2hs

The Haskell side

- Storable class
- therning.org/ magnus » hsc2hs
- A Haskell class where you define:

The Haskell side

- Storable class
- therning.org/ magnus » hsc2hs
- A Haskell class where you define:
 - peek

The Haskell side

- Storable class
- [therning.org/ magnus](http://therning.org/magnus) » hsc2hs
- A Haskell class where you define:
 - peek
 - poke

Demo

Haskell: f a l a2 ...

Haskell: f a l a2 ...



Haskell: `f a1 a2 ...`



Python: `[f, a1, a2, ...]`

Haskell: `f a1 a2 ...`



Python: `[f, a1, a2, ...]`

`CId`

`Expr`

`Expr`



- Features

- Features
- parsing/linearizing

- Features
 - parsing/linearizing
 - Expression structure

- Features
 - parsing/linearizing
 - Expression structure
- Unicode

- Features
 - parsing/linearizing
 - Expression structure
- Unicode
- Only for Linux

- Features
 - parsing/linearizing
 - Expression structure
- Unicode
- Only for Linux
 - depends on ghc

- Features
 - parsing/linearizing
 - Expression structure
- Unicode
- Only for Linux
 - depends on ghc
- Integration into nltk **core**?

- Features
 - parsing/linearizing
 - Expression structure
- Unicode
- Only for Linux
 - depends on ghc
- Integration into nltk **core**?