

# Progress with GF Portuguese Resource Grammar



August 2009

Rami Galil

# Overview

- Progress on Portuguese
- Code Extracts
- Demos and Results

# Progress on Portuguese

- completed morphology for nouns, verbs and adjectives (improved smart paradigms)
- adjectivation is the most complex syntactic rule so far
- extended lexicon
- determination not completely finished

## Some numbers

- 70 nouns in LexiconPor
- 60 of them are built given one form of the noun
- 20 verbs
- 16 adjectives

# Nouns

- two smart paradigms for nouns
- first one accounts for the typical cases
- second does not try to decide on gender
- the order of patterns influences performance (maybe not so important)

smartNoun : Str -> Noun

notSoSmartNoun : Str -> Gender -> Noun

## Nouns (cont.)

- alemão (Sg) => alemães (Pl)
- coração (Sg) => corações (Pl)
- smart paradigm can only account for one case
- what to do if the user wants to extend the lexicon?

# Repeated code?

```
regVerbConj1 : Str -> Verb = \amar ->
let am = Predef.tk 2 amar
in mkVerb amar (am + "o") (am + "as") (am + "a") (am + "amos") (am + "ais") (am + "am")
(am + "e") (am + "es") (am + "e") (am + "emos") (am + "eis") (am + "em")
(am + "a") (am + "ai")
(am + "ado") (am + "ados") (am + "ada") (am + "adas")
(am + "ando") ;
```

```
regVerbConj1G : Str -> Verb = \jogar ->
let jog = Predef.tk 2 jogar
in mkVerb jogar (jog + "o") (jog + "as") (jog + "a") (jog + "amos") (jog + "ais") (jog + "am")
(jog + "ue") (jog + "ues") (jog + "ue") (jog + "uemos") (jog + "ueis") (jog + "uem")
(jog + "a") (jog + "ai")
(jog + "ado") (jog + "ados") (jog + "ada") (jog + "adas")
(jog + "ando") ;
```

# Romance Functor

- o policía || a policía => Gender parameter has to account for this case `CN = {s : Number => Str ; g : Gender} ;`
- difficult to integrate (a lot of linguistics concepts I don't know)
- *common* library seems to be useful, allows the reuse of code

# Future Work

- complete morphology by the end of the year
- study some linguistics in order to implement more complex syntactic rules
- make use of GF in a relatively big case-study (proven to be very useful when we are dealing with specific domains)
- *translating* between program specification languages (seen as software contracts)

Thank you All!

