

# The GF Mathematics Library

Jordi Saludes & Sebastian Xambó

THedu'11, 31 July 2011, Wrocław

# What is it?

- ▶ A *GF* grammar library for *simple* mathematical exercises.
- ▶ It started as part of the WebALT project (EDC-22253).
- ▶ Repository of exercises
  - ▶ Linear algebra
  - ▶ Calculus
  - ▶ Excluding *word problems*

## From EBNF to GF

proposition = number "<" number

number = "x"

| "y"

| "|" number "|"

| number "+" number

- ▶ Split into *abstract* and *concrete*
- ▶ Naming the rules
- ▶ non-terminals become *categories*

## The abstract part

```
cat
  number ; proposition ;
fun
  lt : number -> number -> proposition ;
  x : number ;
  y : number ;
  abs : number -> number ;
  plus : number -> number -> number ;
```

- ▶ Not only string concatenation
  - ▶ strings
  - ▶ parameters,
  - ▶ finite maps,
  - ▶ tuples of all of these
- ▶ non-terminals have a *linearizing category* here

## the concrete part

lin

```
x = "x" ;  
y = "y" ;  
abs e = "|" ++ e ++ "|"  
plus a b = a ++ "+" ++ b  
lt a b = a ++ "<" ++ b
```

- ▶ More than *Context-free* grammars
- ▶ Discontinuous constituents
- ▶ Bidirectional (*parsing* and *linearizing*)
- ▶ More than one *concrete*



## For natural language

lincat

number = NP ; — Noun Phrase

proposition = S ; — Sentence

lin

x = symb {s= "x"} ;

y = symb {s = "y"} ;

abs e = DefGenCN absolute\_CN e ;

plus a b = DefGenCN sum\_CN (both\_and a b) ;

lt a b = PositiveCl

(mkCl a (comparative small\_A b)) ;

- ▶ The Molto math grammar library (mgl) follows the lines of the webALT library. It will follow the OpenMath directives:



*OpenMath is an extensible standard for  
representing the semantics of mathematical objects*

- ▶ Organized into **Content Dictionaries**
- ▶ Abstract part is taken from OpenMath *Small Type System* (STS)

For  $X$  in Bulgarian, German, Catalan, Italian, English, Romanian, Finnish, Spanish, French, Swedish:

**Basic** Logic1X, Quant1X, Relation1X, Set1X,  
SetName1X;

**Aritmetics** Arith1X, Arith2X, Integer1X, Integer2X

**Calculus** Calculus1X, Complex1X, Fns1X, Interval1X,  
Limit1X, MinMax1X, Nums1X, Rounding1X,  
Transc1X;

**Linear Algebra** LinAlg1X, LinAlg2X, PlanGeo1X,  
VecCalc1X;

**Other** SData1X.

# Layers

The library is organized in 3 layers of increasing complexity:

**Ground:** For literal integers and variables

**OpenMath:** For *OpenMath* objects

**Operations:** Problems, verbalizations, ...

- ▶ For each language  $X$  we have a *concrete* and a *resource* module.
- ▶ `LexiconX`
  - ▶ `GroundX` and `VariablesX`
  - ▶ `OpenMathX`
  - ▶ `OperationsX`
- ▶ Common material is reused.

## Some numbers

Total lines	12,544	Bulgarian	568	German	588
mathres	1,162	Catalan	595	Italian	628
MathForms	665	English	614	Romanian	735
gf files	425	Finnish	693	Spanish	624
Abs lines	275	French	642	Swedish	559
languages	10 + 1				

- ▶ Plus Polish and Urdu
- ▶ **Notice:** all languages have a similar complexity

# Mathbar

## Next steps

- ▶ Commanding a CAS by natural language
- ▶ Dialog system for assisting in solving *word problems*



## Example

*A farm has ducks and rabbits. There are  $N$  animals and they have  $M$  legs. How many rabbits and ducks are in the farm?*

## Example

*David and John are father and son. David is now three times much older than John but in 10 years, he will be only twice older. How old are David and John now?*

- ▶ Connecting GF with *ontologies*
- ▶ Part of the MOLTO project
- ▶ Adding factual information

$$\forall d (\text{isa } d \text{ duck}) \implies (\text{has-legs } d \text{ } 2)$$

- ▶ To be rendered as: “a duck has 2 legs” or “ducks have 2 legs”
- ▶ instead of: “for all  $d$  such that  $d$  is a duck,  $d$  has 2 legs”
- ▶ Parsing ?

# Interacting with a grammar

- ▶ The portable grammar format (pgf) allows us to interface it using Haskell, C or Python
- ▶ A grammar can be compiled into JavaScript (js) and into speech recognition systems (gsl)

# How to make a new language

1. Fill the lexicon
2. Compile
3. Get a native to correct the *treebank*
4. Fix the differences

# Conclusion

- ▶ For now, GF can provide linearizations in multiple languages
- ▶ Robust parsing in the future