

WP1

Grammarian's Tools

Aarne Ranta (UGOT)

Summary

IDE's: Eclipse (John Camilleri) and cloud-based (Thomas Hallgren)

Support for on-the-fly extension

Resource grammars: Hindi, Latvian, Nepali, Persian, Punjabi, Sindhi, Thai (Shafqat Virk & al., Normunds Gruzitis)

Issues

Move WP end to Month 27.

Eclipse IDE

For power users

Support for

- large-scale project management (hundreds of GF modules)
- browsing libraries, continuous compilation
- treebank-based testing

Project Explorer

- Demo
 - AbsCat.gf
 - HelloAbs.gf
 - HelloEng.gf
 - ResEng.gf

```
*AbsCat.gf  
abstract AbsCat = {  
  cat greet ; Recipient ;  
};
```

```
ResEng.gf  
--# -path=.:.../abstract:.../common:.../.../prelude  
resource ResEng = {  
  param  
    Gender = Masc | Fem ;  
};
```

Outline

- HelloAbs
 - Extends
 - AbsCat
 - Flags
 - startcat = Greeting
 - cat
 - Farewell
 - fun
 - Hello
 - Goodbye
 - World
 - Parent
 - Friends

```
*HelloAbs.gf  
abstract HelloAbs = AbsCat ** {  
  flags startcat = Greeting ;  
  cat Farewell ;  
  fun  
    Hello : Recipient -> Greeting ;  
    Goodbye : Recipient -> Farewell ;  
    World, Parent, Friends : Recipient ;  
}
```

```
HelloEng.gf  
concrete HelloEng of HelloAbs = open ResEng in {  
  lincat  
    greeting, Farewell = {s : Str} ;  
    Recipient = {s : Gender => Str} ;  
  lin  
    Hello recip = {s = "hello" ++ recip.s | ResEng.Masc} ;  
    Goodbye recip = {s = "goodbye" ++ recip.s | Fem} ;  
    World = {s = \"\\\" => "world"} ;  
    Parent = {s = table {  
      Masc => "dad" ; Fem => "mum"  
    }} ;  
    Friends = superate "friends" ;  
  oper  
    superate : Str -> Recipient = \s ->  
      lin Recipient { s = \"\\\" => "super" ++ s } ;  
}
```

Console
GF Compiler

Cloud-based IDE

For beginners and occasional users: zero-click installation

Potential for collaborative use: grammar cloud

Integration with translation tools for on-the-fly extensions

Support for

- using libraries and GF server on compiler
- creating translation systems and quizzes on the cloud

GF online editor for simple multilingual grammars

Hello%

Abstract **x Finnish** **x Romanian** **x Swedish** **x English** **Show plain** **Upload** **X** **+**

concrete HelloFin of Hello =

open

+

lincat
Greeting = Str%
Friend = Str%

lin
Hello friend = ("terve" | "hei" | "moro") ++ friend%

World = "maailma" %

param

+

oper

+

Enable editing on touch devices. **+**=Add an item, **x**=Delete item, **%**=Edit item.

GF Resource Grammar Library

Start of MOLTO: 15 languages, now 24

Efficient grammar-writing by domain experts

Common API for syntax constructs

New languages: Hindi, Latvian, Nepali, Persian, Punjabi, Sindhi, Thai
(Shafqat Virk & al., Normunds Gruzitis)

mkC1	<u>NP</u> -> <u>V2</u> -> <u>NP</u> -> <u>Cl</u>	she loves him
mkC1	<u>NP</u> -> <u>V3</u> -> <u>NP</u> -> <u>NP</u> -> <u>Cl</u>	she sends it to him
mkC1	<u>NP</u> -> <u>VV</u> -> <u>VP</u> -> <u>Cl</u>	she wants to sleep
mkC1	<u>NP</u> -> <u>VS</u> -> <u>S</u> -> <u>Cl</u>	she say
mkC1	<u>NP</u> -> <u>VO</u> -> <u>QS</u> -> <u>Cl</u>	she work
mkC1	<u>NP</u> -> <u>VA</u> -> <u>A</u> -> <u>Cl</u>	she become
mkC1	<u>NP</u> -> <u>VA</u> -> <u>AP</u> -> <u>Cl</u>	she become
mkC1	<u>NP</u> -> <u>V2A</u> -> <u>NP</u> -> <u>A</u> -> <u>Cl</u>	she paid
mkC1	<u>NP</u> -> <u>V2A</u> -> <u>NP</u> -> <u>AP</u> -> <u>Cl</u>	she paid
mkC1	<u>NP</u> -> <u>V2S</u> -> <u>NP</u> -> <u>S</u> -> <u>Cl</u>	she answer
mkC1	<u>NP</u> -> <u>V2Q</u> -> <u>NP</u> -> <u>QS</u> -> <u>Cl</u>	she ask
mkC1	<u>NP</u> -> <u>V2V</u> -> <u>NP</u> -> <u>VP</u> -> <u>Cl</u>	she beg
mkC1	<u>NP</u> -> <u>A</u> -> <u>Cl</u>	she is
mkC1	<u>NP</u> -> <u>A</u> -> <u>NP</u> -> <u>Cl</u>	she is
mkC1	<u>NP</u> -> <u>A2</u> -> <u>NP</u> -> <u>Cl</u>	she is
mkC1	<u>NP</u> -> <u>AP</u> -> <u>Cl</u>	she is
mkC1	<u>NP</u> -> <u>NP</u> -> <u>Cl</u>	she is
mkC1	<u>NP</u> -> <u>N</u> -> <u>Cl</u>	she is
mkC1	<u>NP</u> -> <u>CN</u> -> <u>Cl</u>	she is
mkC1	<u>NP</u> -> <u>Adv</u> -> <u>Cl</u>	she is
mkC1	<u>NP</u> -> <u>VP</u> -> <u>Cl</u>	she always

- APl: mkC1 she NP want_VV (mkVP sleep_V)
- Afr: sy wil te slaap
- Bul: ma ucka òa emu
- Cat: ella vol dormir
- Dan: hun vil sove
- Dut: ze wil slapen
- Eng: she wants to sleep
- Fin: hän tahtoo nukkuua
- Fre: elle veut dormir
- Ger: sie will schlafen
- Hin: वह सोना चाहती है
- Ita: lei vuole dormire
- Lav: viņa grib gulēt
- Nep: उनी सुत्न चाहन्छिन्
- Nor: hun vil sove
- Pes: او می خواهد بخوابد
- Pnb: او سوتا چاندی اے
- Pol: ona chce spać
- Ron: ea vrea să doarmă
- Rus: она хочет спать
- Snd: ميري چاهي چاهي
- Spa: ella quiere dormir
- Swe: hon vill sova
- Tha: หลอนอยากนอนหลับ
- Urd: وہ سوينا چاهي, مير

Smart Paradigms

Resource Grammar Library support for lexicon extension

Infer complete inflection table from one or a few forms:

```
mkV "kill" --> kill, kills, killed, killed, killing
mkV "cry" --> cry, cries, cried, cried, crying
mkV "stop" --> stop, stops, stopped, stopped, stopping
mkV "put" "put" "put" --> put, puts, put, put, putting
```

Concepts and evaluation

G. D etrez and A. Ranta. Smart Paradigms and the Predictability and Complexity of Inflectional Morphology. *EACL (European Association for Computational Linguistics)*, Avignon, April 2012.

Lexicon	Forms	Entries	Cost	$m = 1$	$m \leq 2$
Eng N	2	15,029	1.05	95%	100%
Eng V	5	5,692	1.21	84%	95%
Swe N	9	59,225	1.70	46%	92%
Swe V	20	4,789	1.13	97%	97%
Fre N	3	42,390	1.25	76%	99%
Fre V	51	6,851	1.27	92%	94%
Fin N	34	25,365	1.26	87%	97%
Fin V	102	10,355	1.09	96%	99%

Last deliverable D2.3: User Manual and Best Practices

How to use GF tools: a short hands-on introduction

How we have done in MOLTTO - and how we should have done

When to use GF (as opposed to Prolog, Yacc,...)

When to use a grammar (as opposed to SMT,...)

Language-specific issues

Skills: resource/application grammarians

Case studies

Results and statistics

Issue: extending WP2 duration

From March to June 2012

Reason: we want to include the experience from Be Informed

(Recall: the start of MOLTO extension was delayed)