Multilingual Resource Grammar Library

Aarne Ranta

Demo, MOLTO Open Day, Gothenburg, 9 March 2011

The GF Resource Grammar Library

Goal: encapsulate linguistic knowledge

- morphology
- syntax

This knowledge is created by **linguists**

This knowledge is used by **application programmers**, e.g. **domain experts**

Morphological knowledge

English ”regular” verbs, under the hood:

```
mkV : Str -> V = \v -> case v of {
  _ + ("s"|"z"|"x"|"ch")      => s_regVerb v ;   -- munch, munches
  _ + "ie"                    => ie_regVerb v ;  -- die, dying
  _ + "ee"                    => ee_regVerb v ;  -- free, freed, freeing
  _ + "e"                     => e_regVerb v ;   -- use, used, using
  _ + ("a"|"e"|"o"|"u") + "y" => regVerb v ;     -- play, played
  _ + "y"                     => y_regVerb v ;   -- cry, cried
  _ + #vowel + #consonant     => dupRegVerb ;    -- stop, stopped
  _                           => regVerb v
  } ;
```

Syntactic knowledge

English clause formation: negation, tenses, inversion:

```
  mkCl : NP -> VP -> Cl = \np,vp -> {
      s = \\t,a,b,o =>
        let
          np    = np.s ! Nom ;
          agr   = np.a ;
          verb  = vp.s ! t ! a ! b ! o ! agr ;
          compl = vp.s2 ! agr
        in
        case o of {
          ODir   => subj ++ verb.aux ++ verb.adv ++ vp.ad ++ verb.fin ++ verb.inf ++ compl ;
          OQuest => verb.aux ++ subj ++ verb.adv ++ vp.ad ++ verb.fin ++ verb.inf ++ compl
          }
    } ;

predV : V -> VP = \verb -> {
    s = \\t,ant,b,ord,agr =>
      let
        inf  = verb.s ! VInf ;
        fin  = presVerb verb agr ;
        part = verb.s ! VPPart ;
      in
      case <t,ant,b,ord> of {
        <Pres,Simul,CPos,ODir>   => vff            fin [] ;
        <Pres,Simul,CPos,OQuest> => vf (does agr)   inf ;
    <Pres,Anter,CPos,_>      => vf (have agr)   part ;
    <Pres,Anter,CNeg c,_>    => vfn c (have agr) (havent agr) part ;
    <Past,Simul,CPos,ODir>   => vff (verb.s ! VPast) [] ;
    <Past,Simul,CPos,OQuest> => vf "did"        inf ;
    <Past,Simul,CNeg c,_>    => vfn c "did" "didn’t"     inf ;
    <Past,Anter,CPos,_>      => vf "had"        part ;
    <Past,Anter,CNeg c,_>    => vfn c "had" "hadn’t"     part ;
    <Fut, Simul,CPos,_>      => vf "will"       inf ;
    <Fut, Simul,CNeg c,_>    => vfn c "will" "won’t"      inf ;
    <Fut, Anter,CPos,_>      => vf "will"       ("have" ++ part) ;
    <Fut, Anter,CNeg c,_>    => vfn c "will" "won’t"("have" ++ part) ;
    <Cond,Simul,CPos,_>      => vf "would"      inf ;
    <Cond,Simul,CNeg c,_>    => vfn c "would" "wouldn’t"   inf ;
    <Cond,Anter,CPos,_>      => vf "would"      ("have" ++ part) ;
    <Cond,Anter,CNeg c,_> => vfn c "would" "wouldn’t" ("have" ++ part) ;
    <Pres,Simul,CNeg c,_>    => vfn c (does agr) (doesnt agr) inf
    } ;
prp  = verb.s ! VPresPart ;
inf  = verb.s ! VInf ;
ad   = [] ;
s2 = \\a => if_then_Str verb.isRefl (reflPron ! a) []
} ;
```

Linguistic knowledge via the API

```
mkV : (munch : Str) -> V
mkV : (drink, drank, drunk : Str) -> V

mkCl : NP -> V         -> Cl  -- she sleeps
mkCl : NP -> V2 -> NP -> Cl  -- she sees us
mkCl : NP -> A         -> Cl  -- she is old
mkCl : NP -> N         -> Cl  -- she is a student
...
```

Shared syntax, specific morphologies

Syntax*L*, common syntax API for all 18+ languages

Paradigms*L*, inflection paradigms for each language

Browsing and testing the API

For browsing:

- http://www.grammaticalframework.org/lib/doc/synopsis.html

For testing:

- download and install the GF shell and the libraries
- now also in the cloud!