The Latin Language Ressource Grammar

GF Summer School, Frauenchiemsee 2013

Herbert Lange
Centrum für Informations- und Sprachverarbeitung
München

29. August 2013

Current Status

```
 lib/src/latin/AdjectiveLat.gf   |   14 +-
 lib/src/latin/AdverbLat.gf      |    4 +-
 lib/src/latin/AllLat.gf         |    4 +-
 lib/src/latin/AllLatAbs.gf      |   10 +-
 lib/src/latin/CatLat.gf         |   69 ++---
 lib/src/latin/ConjunctionLat.gf |   37 ++-
 lib/src/latin/ExtraLat.gf       |    8 +
 lib/src/latin/ExtraLatAbs.gf    |    5 +
 lib/src/latin/GrammarLat.gf     |    5 +-
 lib/src/latin/IrregLat.gf       |  631 +++++++++++++++++++++++++++++++-------------------
 lib/src/latin/IrregLatAbs.gf    |   11 +
 lib/src/latin/LangLat.gf        |    2 +
 lib/src/latin/LexiconLat.gf     |  752 ++++++++++++++++++++++++++--------------------------
 lib/src/latin/MorphoLat.gf      |  800 ++++++++++++++++++++++++++++++++++++++++++-------------------
 lib/src/latin/NounLat.gf        |   92 ++++---
 lib/src/latin/ParadigmsLat.gf   |   63 ++++-
 lib/src/latin/PhraseLat.gf      |   24 +-
 lib/src/latin/ResLat.gf         | 1223 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----------------------------
 lib/src/latin/SentenceLat.gf    |   22 +-
 lib/src/latin/StructuralLat.gf  |  226 ++++++++--------
 lib/src/latin/VerbLat.gf        |   13 +-
 21 files changed, 2610 insertions(+), 1405 deletions(-)
```

Lexicon

- All strings replaced by (more or less) appropriate Latin translations
- Some words only translatable by phrases (e.g. camera_N, travel_V) ⇒ Create phrases and wrap them up as a noun or verb ⇒ Works fine for CN but not so well for VPs
- Discovered all kinds of irregular word forms in the lexicon (e.g. deponent verbs, defective verbs, plural only nouns) ⇒ Motivation to implement as much of the morphology as possible
- Challenge to implement modern words (airplane_N, train_N, ...) ⇒ Wikipedia as a useful source for translations

Examples from the Lexicon

Example

Excerpt from the Lexicon[1]

```
lin
  [...]
  camera_N =
    ResLat.useCNasN (AdjCN (PositA (mkA "photographicus") )
                           (UseN (mkN "machina" ) )
                    ) ; -- (http://la.wikipedia.org/wiki/Machina_photographica / Pons)
  [...]
  train_N = mkN "hamaxostichus" ; -- -i m. (http://la.wikipedia.org/wiki/Hamaxostichus)
  travel_V =
    ResLat.useVPasV ( ComplSlash ( SlashV2a ( mkV2 "facere" ) )
                          ( DetCN ( DetQuant IndefArt NumSg )
                                  ( UseN ( mkN "iter" "itineris" Neutr ) )
                          )
             ) ; -- facio, feci, factum 3
  [...]
  science_N = pluralN (mkN "litera" ) ; -- only pl. (Langenscheidts)
```

[1] LexiconLat.gf

Morphology

- Trying not to use sound laws
- Morphology for Nouns, Adjectives, Verbs and Personal/Possesive Pronouns
- Smart Paradigms as smart as possible

Noun Morphology

- Six Cases (but mostly Nominative and Vocative have the same form)
- Two Number categories
  $\Rightarrow 6x2 = 12$ Forms
- Five Declension classes

Example

Noun type and parameters[2]

```
param
  Case = Nom | Acc | Gen | Dat | Abl | Voc ;
  Gender = Masc | Fem | Neutr ;
oper
  Noun : Type = {s : Number => Case => Str ; g : Gender} ;
```

[2] ResLat.gf

Smart paradigm

Example

Smart paradigm[3]

```
noun : Str -> Noun = \verbum ->
  case verbum of {
    _ + "a"  => noun1 verbum ;
    _ + "us" => noun2us verbum ;
    _ + "um" => noun2um verbum ;
    _ + ( "er" | "ir" ) =>
      noun2er verbum ( (Predef.tk 2 verbum) + "ri" ) ;
    _ + "u"  => noun4u verbum ;
    _ + "es" => noun5 verbum ;
    _  =>
      Predef.error
        ("3rd declinsion cannot be applied to just
          one noun form " ++ verbum)
    } ;
```

[3] MorphoLat.gf

Smart paradigm

Example

Smart paradigm[4]

```
noun_ngg : Str -> Str -> Gender -> Noun = \verbum,verbi,g ->
  let s : Noun = case <verbum,verbi> of {
    <_ + "a",  _ + "ae"> => noun1 verbum ;
    <_ + "us", _ + "i">  => noun2us verbum ;
    <_ + "um", _ + "i">  => noun2um verbum ;
    <_ + ( "er" | "ir" ) , _ + "i">  => noun2er verbum verbi ;

    <_ + "us", _ + "us"> => noun4us verbum ;
    <_ + "u",  _ + "us"> => noun4u verbum ;
    <_ + "es", _ + "ei"> => noun5 verbum ;
    _  => noun3 verbum verbi g
    }
  in
  nounWithGen g s ;
```

[4] MorphoLat.gf

Example

Paradigm of friend_N

```
Lang> l -table friend_N
s Sg Nom : amicus                      s Sg Nom : amica
s Sg Acc : amicum                      s Sg Acc : amicam
s Sg Gen : amici                       s Sg Gen : amicae
s Sg Dat : amico                       s Sg Dat : amicae
s Sg Abl : amico                       s Sg Abl : amica
s Sg Voc : amice                       s Sg Voc : amica
s Pl Nom : amici                       s Pl Nom : amicae
s Pl Acc : amicos                      s Pl Acc : amicas
s Pl Gen : amicorum                    s Pl Gen : amicarum
s Pl Dat : amicis                      s Pl Dat : amicis
s Pl Abl : amicis                      s Pl Abl : amicis
s Pl Voc : amici                       s Pl Voc : amicae
```

Adjective Morphology

- Three Gender categories
- Two Number categories
- Six Cases
- Three degrees of comparation
  $\Rightarrow 3x2x6x3 = 108$ Forms
- Three Declination Classes

Example

Adjective type[5]

```
param
  Agr = Ag Gender Number Case ; -- Agreement for NP et al.
oper
  Adjective : Type = { s : Degree => Agr => Str } ;
```

[5] ResLat.gf

- More complex than noun declension
- Some hard-coded exception handling (Maybe find a better solution later)

Example

Exceptions in adjective declension[6]

```
adj12 : Str -> Adjective = \bonus ->
  let
    bon : Str = case bonus of {
      -- Exceptions Bayer-Lindauer 41 3.2
      ("asper" | "liber" | "miser" | "tener" | "frugifer") => bonus ;
      -- Usual cases
      pulch + "er" => pulch + "r" ;
      bon + "us" => bon ;
      _ => Predef.error ("adj12 does not apply to" ++ bonus)
      } ;
      [...]
    in [...]
```

[6] MorphoLat.gf

Verb conjugation

- Lots of Forms: Active 60 forms, passive 30 forms, participle 108, gerund 4 forms, gerundive 36 forms, infinitive 12 forms, imperative 8 forms, supine 2 forms ⇒ Total 260 Forms

Example

Verb parameters[7]

```
param
  VActForm  = VAct VAnter VTense Number Person ;
  VPassForm = -- No anteriority because perfect forms are built using participle
    VPass VTense Number Person ;
  VInfForm  = VInfActPres | VInfActPerf Gender | VInfActFut Gender |
    VInfPassPres | VInfPassPerf Gender | VinfPassFut ;
  VImpForm  = VImp1 Number | VImp2 Number Person ;
  VGerund   = VGenAcc | VGenGen |VGenDat | VGenAbl ;
  VSupine   = VSupAcc | VSupAbl ;
  VPartForm = VActPres | VActFut | VPassPerf ;

  VAnter = VAnt | VSim ;
  VTense = VPres VMood | VImpf VMood | VFut ;
  VMood  = VInd | VConj ;
```

[7] ResLat.gf

Example

Verb type[8]

```
oper
  Verb : Type = {
    act   : VActForm => Str ;
    pass  : VPassForm => Str ;
    inf   : VInfForm => Str ;
    imp   : VImpForm => Str ;
    ger   : VGerund => Str ;
    geriv : Agr => Str ;
    sup   : VSupine => Str ;
    part  : VPartForm =>Agr => Str ;
    } ;
```

[8] ResLat.gf

A (nearly) complete verb lemma

Example

```
act (VAct VAnt (VPres VInd) Sg P1) : legi
act (VAct VAnt (VPres VInd) Sg P2) : legisti
act (VAct VAnt (VPres VInd) Sg P3) : legit
act (VAct VAnt (VPres VInd) Pl P1) : legimus
act (VAct VAnt (VPres VInd) Pl P2) : legistis
act (VAct VAnt (VPres VInd) Pl P3) : legerunt
act (VAct VAnt (VPres VConj) Sg P1) : legerim
act (VAct VAnt (VPres VConj) Sg P2) : legeris
act (VAct VAnt (VPres VConj) Sg P3) : legerit
act (VAct VAnt (VPres VConj) Pl P1) : legerimus
act (VAct VAnt (VPres VConj) Pl P2) : legeritis
act (VAct VAnt (VPres VConj) Pl P3) : legerint
act (VAct VAnt (VImpf VInd) Sg P1) : legeram
act (VAct VAnt (VImpf VInd) Sg P2) : legeras
act (VAct VAnt (VImpf VInd) Sg P3) : legerat
act (VAct VAnt (VImpf VInd) Pl P1) : legeramus
act (VAct VAnt (VImpf VInd) Pl P2) : legeratis
act (VAct VAnt (VImpf VInd) Pl P3) : legerant
act (VAct VAnt (VImpf VConj) Sg P1) : legissem
act (VAct VAnt (VImpf VConj) Sg P2) : legisses
act (VAct VAnt (VImpf VConj) Sg P3) : legisset
act (VAct VAnt (VImpf VConj) Pl P1) : legissemus
act (VAct VAnt (VImpf VConj) Pl P2) : legissetis
act (VAct VAnt (VImpf VConj) Pl P3) : legissent
act (VAct VAnt VFut Sg P1) : legero
act (VAct VAnt VFut Sg P2) : legeris
act (VAct VAnt VFut Sg P3) : legerit
act (VAct VAnt VFut Pl P1) : legerimus
act (VAct VAnt VFut Pl P2) : legeritis
act (VAct VAnt VFut Pl P3) : legerint
act (VAct VSim (VPres VInd) Sg P1) : lego
act (VAct VSim (VPres VInd) Sg P2) : legis
act (VAct VSim (VPres VInd) Sg P3) : legit
act (VAct VSim (VPres VInd) Pl P1) : legimus
act (VAct VSim (VPres VInd) Pl P2) : legitis
act (VAct VSim (VPres VInd) Pl P3) : legunt
act (VAct VSim (VPres VConj) Sg P1) : legam
act (VAct VSim (VPres VConj) Sg P2) : legas
act (VAct VSim (VPres VConj) Sg P3) : legat
act (VAct VSim (VPres VConj) Pl P1) : legamus
act (VAct VSim (VPres VConj) Pl P2) : legatis
act (VAct VSim (VPres VConj) Pl P3) : legant
act (VAct VSim (VImpf VInd) Sg P1) : legebam
act (VAct VSim (VImpf VInd) Sg P2) : legebas
act (VAct VSim (VImpf VInd) Sg P3) : legebat
act (VAct VSim (VImpf VInd) Pl P1) : legebamus
act (VAct VSim (VImpf VInd) Pl P2) : legebatis
act (VAct VSim (VImpf VInd) Pl P3) : legebant
act (VAct VSim (VImpf VConj) Sg P1) : legerem
act (VAct VSim (VImpf VConj) Sg P2) : legeres
act (VAct VSim (VImpf VConj) Sg P3) : legeret
act (VAct VSim (VImpf VConj) Pl P1) : legeremus
act (VAct VSim (VImpf VConj) Pl P2) : legeretis
act (VAct VSim (VImpf VConj) Pl P3) : legerent
act (VAct VSim VFut Sg P1) : legam
act (VAct VSim VFut Sg P2) : leges
act (VAct VSim VFut Sg P3) : leget
act (VAct VSim VFut Pl P1) : legemus
act (VAct VSim VFut Pl P2) : legetis
act (VAct VSim VFut Pl P3) : legent
ger VGenAcc : legendum
ger VGenGen : legendi
ger VGenDat : legendo
ger VGenAbl : legendo
geriv (Ag Masc Sg Nom) : legendus
geriv (Ag Masc Sg Acc) : legendum
geriv (Ag Masc Sg Gen) : legendi
geriv (Ag Masc Sg Dat) : legendo
geriv (Ag Masc Sg Abl) : legendo
geriv (Ag Masc Sg Voc) : legende
geriv (Ag Masc Pl Nom) : legendi
geriv (Ag Masc Pl Acc) : legendos
geriv (Ag Masc Pl Gen) : legendorum
geriv (Ag Masc Pl Dat) : legendis
geriv (Ag Masc Pl Abl) : legendis
geriv (Ag Masc Pl Voc) : legendi
geriv (Ag Fem Sg Nom) : legenda
geriv (Ag Fem Sg Acc) : legendam
geriv (Ag Fem Sg Gen) : legendae
geriv (Ag Fem Sg Dat) : legendae
geriv (Ag Fem Sg Abl) : legenda
geriv (Ag Fem Sg Voc) : legenda
geriv (Ag Fem Pl Nom) : legendae
geriv (Ag Fem Pl Acc) : legendas
geriv (Ag Fem Pl Gen) : legendarum
geriv (Ag Fem Pl Dat) : legendis
geriv (Ag Fem Pl Abl) : legendis
geriv (Ag Fem Pl Voc) : legendae
geriv (Ag Neutr Sg Nom) : legendum
geriv (Ag Neutr Sg Acc) : legendum
geriv (Ag Neutr Sg Gen) : legendi
geriv (Ag Neutr Sg Dat) : legendo
geriv (Ag Neutr Sg Abl) : legendo
geriv (Ag Neutr Sg Voc) : legendum
geriv (Ag Neutr Pl Nom) : legenda
geriv (Ag Neutr Pl Acc) : legenda
geriv (Ag Neutr Pl Gen) : legendorum
geriv (Ag Neutr Pl Dat) : legendis
geriv (Ag Neutr Pl Abl) : legendis
geriv (Ag Neutr Pl Voc) : legenda
imp (VImp1 Sg) : lege
imp (VImp1 Pl) : legite
imp (VImp2 Sg P1) : ######
imp (VImp2 Sg P2) : legito
imp (VImp2 Sg P3) : legito
imp (VImp2 Pl P1) : ######
imp (VImp2 Pl P2) : legitote
imp (VImp2 Pl P3) : legunto
inf VInfActPres : legere
inf (VInfActPerf Masc) : legisse
inf (VInfActPerf Fem) : legisse
inf (VInfActPerf Neutr) : legisse
inf (VInfActFut Masc) : lecturum
inf (VInfActFut Fem) : lecturam
inf (VInfActFut Neutr) : lecturum
inf VInfPassPres : legeri
inf (VInfPassPerf Masc) : lectum
inf (VInfPassPerf Fem) : lectam
inf (VInfPassPerf Neutr) : lectum
inf VinfPassFut : lectum
part VActPres (Ag Masc Sg Nom) : legens
part VActPres (Ag Masc Sg Acc) : legentem
part VActPres (Ag Masc Sg Gen) : legentis
part VActPres (Ag Masc Sg Dat) : legenti
part VActPres (Ag Masc Sg Abl) : legente
part VActPres (Ag Masc Sg Voc) : legens
part VActPres (Ag Masc Pl Nom) : legentes
part VActPres (Ag Masc Pl Acc) : legentes
part VActPres (Ag Masc Pl Gen) : legentium
part VActPres (Ag Masc Pl Dat) : legentibus
part VActPres (Ag Masc Pl Abl) : legentibus
part VActPres (Ag Masc Pl Voc) : legentes
part VActPres (Ag Fem Sg Nom) : legens
part VActPres (Ag Fem Sg Acc) : legentem
part VActPres (Ag Fem Sg Gen) : legentis
part VActPres (Ag Fem Sg Dat) : legenti
part VActPres (Ag Fem Sg Abl) : legente
part VActPres (Ag Fem Sg Voc) : legens
part VActPres (Ag Fem Pl Nom) : legentes
part VActPres (Ag Fem Pl Acc) : legentes
part VActPres (Ag Fem Pl Gen) : legentium
part VActPres (Ag Fem Pl Dat) : legentibus
part VActPres (Ag Fem Pl Abl) : legentibus
part VActPres (Ag Fem Pl Voc) : legentes
part VActPres (Ag Neutr Sg Nom) : legens
part VActPres (Ag Neutr Sg Acc) : legens
part VActPres (Ag Neutr Sg Gen) : legentis
part VActPres (Ag Neutr Sg Dat) : legenti
part VActPres (Ag Neutr Sg Abl) : legente
part VActPres (Ag Neutr Sg Voc) : legens
part VActPres (Ag Neutr Pl Nom) : legentia
part VActPres (Ag Neutr Pl Acc) : legentia
part VActPres (Ag Neutr Pl Gen) : legentium
part VActPres (Ag Neutr Pl Dat) : legentibus
part VActPres (Ag Neutr Pl Abl) : legentibus
part VActPres (Ag Neutr Pl Voc) : legentia
part VActFut (Ag Masc Sg Nom) : lecturus
part VActFut (Ag Masc Sg Acc) : lecturum
part VActFut (Ag Masc Sg Gen) : lecturi
part VActFut (Ag Masc Sg Dat) : lecturo
part VActFut (Ag Masc Sg Abl) : lecturo
part VActFut (Ag Masc Sg Voc) : lecture
part VActFut (Ag Masc Pl Nom) : lecturi
part VActFut (Ag Masc Pl Acc) : lecturos
part VActFut (Ag Masc Pl Gen) : lecturorum
part VActFut (Ag Masc Pl Dat) : lecturis
part VActFut (Ag Masc Pl Abl) : lecturis
part VActFut (Ag Masc Pl Voc) : lecturi
part VActFut (Ag Fem Sg Nom) : lectura
part VActFut (Ag Fem Sg Acc) : lecturam
part VActFut (Ag Fem Sg Gen) : lecturae
part VActFut (Ag Fem Sg Dat) : lecturae
part VActFut (Ag Fem Sg Abl) : lectura
part VActFut (Ag Fem Sg Voc) : lectura
part VActFut (Ag Fem Pl Nom) : lecturae
part VActFut (Ag Fem Pl Acc) : lecturas
part VActFut (Ag Fem Pl Gen) : lecturarum
part VActFut (Ag Fem Pl Dat) : lecturis
part VActFut (Ag Fem Pl Abl) : lecturis
part VActFut (Ag Fem Pl Voc) : lecturae
part VActFut (Ag Neutr Sg Nom) : lecturum
part VActFut (Ag Neutr Sg Acc) : lecturum
part VActFut (Ag Neutr Sg Gen) : lecturi
part VActFut (Ag Neutr Sg Dat) : lecturo
part VActFut (Ag Neutr Sg Abl) : lecturo
part VActFut (Ag Neutr Sg Voc) : lecturum
part VActFut (Ag Neutr Pl Nom) : lectura
part VActFut (Ag Neutr Pl Acc) : lectura
part VActFut (Ag Neutr Pl Gen) : lecturorum
part VActFut (Ag Neutr Pl Dat) : lecturis
part VActFut (Ag Neutr Pl Abl) : lecturis
part VActFut (Ag Neutr Pl Voc) : lectura
part VPassPerf (Ag Masc Sg Nom) : lectus
part VPassPerf (Ag Masc Sg Acc) : lectum
part VPassPerf (Ag Masc Sg Gen) : lecti
part VPassPerf (Ag Masc Sg Dat) : lecto
part VPassPerf (Ag Masc Sg Abl) : lecto
part VPassPerf (Ag Masc Sg Voc) : lecte
part VPassPerf (Ag Masc Pl Nom) : lecti
part VPassPerf (Ag Masc Pl Acc) : lectos
part VPassPerf (Ag Masc Pl Gen) : lectorum
part VPassPerf (Ag Masc Pl Dat) : lectis
part VPassPerf (Ag Masc Pl Abl) : lectis
part VPassPerf (Ag Masc Pl Voc) : lecti
part VPassPerf (Ag Fem Sg Nom) : lecta
part VPassPerf (Ag Fem Sg Acc) : lectam
part VPassPerf (Ag Fem Sg Gen) : lectae
part VPassPerf (Ag Fem Sg Dat) : lectae
part VPassPerf (Ag Fem Sg Abl) : lecta
part VPassPerf (Ag Fem Sg Voc) : lecta
part VPassPerf (Ag Fem Pl Nom) : lectae
part VPassPerf (Ag Fem Pl Acc) : lectas
part VPassPerf (Ag Fem Pl Gen) : lectarum
part VPassPerf (Ag Fem Pl Dat) : lectis
part VPassPerf (Ag Fem Pl Abl) : lectis
part VPassPerf (Ag Fem Pl Voc) : lectae
part VPassPerf (Ag Neutr Sg Nom) : lectum
part VPassPerf (Ag Neutr Sg Acc) : lectum
part VPassPerf (Ag Neutr Sg Gen) : lecti
part VPassPerf (Ag Neutr Sg Dat) : lecto
part VPassPerf (Ag Neutr Sg Abl) : lecto
part VPassPerf (Ag Neutr Sg Voc) : lectum
part VPassPerf (Ag Neutr Pl Nom) : lecta
part VPassPerf (Ag Neutr Pl Acc) : lecta
part VPassPerf (Ag Neutr Pl Gen) : lectorum
part VPassPerf (Ag Neutr Pl Dat) : lectis
part VPassPerf (Ag Neutr Pl Abl) : lectis
part VPassPerf (Ag Neutr Pl Voc) : lecta
pass (VPass (VPres VInd) Sg P1) : legor
pass (VPass (VPres VInd) Sg P2) : legeris
pass (VPass (VPres VInd) Sg P3) : legitur
pass (VPass (VPres VInd) Pl P1) : legimur
pass (VPass (VPres VInd) Pl P2) : legimini
pass (VPass (VPres VInd) Pl P3) : leguntur
pass (VPass (VPres VConj) Sg P1) : legar
pass (VPass (VPres VConj) Sg P2) : legaris
pass (VPass (VPres VConj) Sg P3) : legatur
pass (VPass (VPres VConj) Pl P1) : legamur
pass (VPass (VPres VConj) Pl P2) : legamini
pass (VPass (VPres VConj) Pl P3) : legantur
pass (VPass (VImpf VInd) Sg P1) : legebar
pass (VPass (VImpf VInd) Sg P2) : legebaris
pass (VPass (VImpf VInd) Sg P3) : legebatur
pass (VPass (VImpf VInd) Pl P1) : legebamur
pass (VPass (VImpf VInd) Pl P2) : legebamini
pass (VPass (VImpf VInd) Pl P3) : legebantur
pass (VPass (VImpf VConj) Sg P1) : legerer
pass (VPass (VImpf VConj) Sg P2) : legereris
pass (VPass (VImpf VConj) Sg P3) : legeretur
pass (VPass (VImpf VConj) Pl P1) : legeremur
pass (VPass (VImpf VConj) Pl P2) : legeremini
pass (VPass (VImpf VConj) Pl P3) : legerentur
pass (VPass VFut Sg P1) : legar
pass (VPass VFut Sg P2) : legeris
pass (VPass VFut Sg P3) : legetur
pass (VPass VFut Pl P1) : legemur
pass (VPass VFut Pl P2) : legemini
pass (VPass VFut Pl P3) : legentur
sup VSupAcc : lectum
sup VSupAbl : lectu
```

Problems with verb morphology

Problems:

- Hard to overlook
- Rarely all fields filled in (Deponent verbs $\Rightarrow$ Passive forms with active usage, defective Verbs $\Rightarrow$ Perfect forms with preset usage, ...)
- Right place for derivative morphology?

Pronouns

Only handling

Example

Pronoun type and parameters[9]

```
param
  PronReflForm = -- refelxive usage of pronoun like 'I see myself'
    PronRefl | PronNonRefl ;
  PronDropForm = PronDrop | PronNonDrop;
oper
  Pronoun : Type = {
    pers : PronDropForm => PronReflForm => Case => Str ;
    poss : PronReflForm => Agr => Str ;
    g : Gender ;
    n : Number ;
    p : Person ;
  } ;
```

[9] ResLat.gf

Syntax

At the moment: just the basic syntax constructions to create VPs and NPs and to form sentences from them

Different word orders on sentence level possible

Example

Possible word orders[10]

```
param
  Order = SVO | VSO | VOS | OSV | OVS | SOV ;
```

[10] ResLat.gf

Syntax

Attributes can appear in front of or after nouns (only implemented for APs)

Example

Handling APs in different possitions[11]

```
CompoundNoun : Type =
{
  s : Number => Case => Str ;
  g : Gender ;
  preap : {s : Agr => Str } ;
  postap : {s : Agr => Str } ;
} ;
```

[11] ResLat.gf

Syntax

Trying to handle Pro-Drop

Example

Default UsePron[12]

```
UsePron p = -- Pron -> Np
  {
    g = p.g ;
    n = p.n ;
    p = p.p ;
    s = p.pers ! PronDrop ! PronRefl ;
  } ;
```

Only works correctly in the subject possition

[12] NounLat.gf

Beginning of Summer school: Plans

- Handling of modifiers for NPs (Adjectives, APs, ...)
- Rules to create S and Utt from Cl
- Evaluation of the Lexicon
- Further testing of the morphology

Handling of modifiers for NPs (Adjectives, APs, ...)

Example

Better handling of APs and adjectives (Variable order before or after noun)

```
param
  AdjPos = Pre | Post ;
lin
  AdjCN ap cn =  -- AP -> CN -> CN
    let pos = variants { Post ; Pre }
    in
    {
      s = cn.s ;
      postap =
        case pos of {
          Pre => cn.postap ;
          Post => { s = \\a => ap.s ! a ++ cn.postap.s ! a }
        } ;
      preap =
        case pos of {
          Post => cn.preap ;
          Pre => { s = \\a => ap.s ! a ++ cn.preap.s ! a }
        } ;
      g = cn.g
    } ;
```

Rules to create S and Utt from Cl

Example

Sentence rules[13]

```
PredVP np vp = -- NP -> VP -> Cl
  {
    s = \\tense,anter,pol,order =>
      case order of {
        [...]
        OSV -- Object - Subject - Verb
          => vp.obj ++ np.s ! Nom ++ negation pol ++
          vp.fin ! VAct ( anteriorityToVAnter anter ) ( tenseToVTense tense ) np.n np.p ;
        [...]
        SOV -- Subject - Objecy - Verb
          => np.s ! Nom ++ vp.obj ++ negation pol ++
          vp.fin ! VAct ( anteriorityToVAnter anter ) ( tenseToVTense tense ) np.n np.p
      }
    } ;

[...]

UseCl  t p cl = -- Temp -> Pol-> Cl -> S
  {
    s = t.s ++ p.s ++ cl.s ! t.t ! t.a ! p.p ! SOV
  } ;
```

[13]SentenceLat.gf

Testing and evaluation

Still to be done
$\Rightarrow$ Different ressources (Latin treebank, compilation of a test corpus, ...)

Any Questions?