Plan

• Morphology
• The Lexicon
• Syntax
• Future Plan
Words categories

• So far I have tried with following word categories.
  – Nouns
  – Verbs
  – Adjectives
  – Pronoun
  – Proper Names
  – Closed Classes
Nouns

- Urdu Noun Inflects in **Number** (*Singular, Plural*) and **Case** (*Direct, Oblique, Vocative*)
- Inherent parameter: Gender (*Masculine, Feminine*)
- Nouns are divided into 15 groups based on their inflection
A Running Example

• A group as running example:
• Singular masculine nouns ending with ( ,a) ,( ,h) and ( ,e )
• Making:
• If a word ends with letter ( ,a) or ( ,h) then:
  – Plural nominative, singular oblique, Singular Vocative: last letter is replaced by ( ,E)
  – Plural oblique: the last letter is replaced by ( ,w)
  – Plural vocative: last letter is replaced by ( ,wN)
• If a word ends with ( ,e): above mentioned letters will be added at the end without replacing any existing letter
Noun Example

• boy_N = mkN "IRka" ;
• This will call mkN in paradigmsUrd.gf

mkN = overload {
  mkN : Str -> N
  = \s -> regNoun s ** {lock_N = <>} ;
  mkN : Str -> Gender -> N
  = \s,g -> reggNoun s g ** {lock_N = <>} ;
  mkN : (x1,_,_,_,_,x6 : Str) -> Gender -> N
  = \sd,so,sv,pd,po,pv,g -> mkNoun sd so sv pd po pv g
  ** {lock_N = <>} ;
};
Noun Example

regNoun : Str -> Noun ;
regNoun s = case s of {
    _ + "ya" => mkN05 (s);
    _ + ("a" | "e" | "h") => mkN01 (s);
    _ + "y" => mkN03 (s);
    _ + ("aN" | "wN") => mkN04 (s);
    _ + "w^" => mkN12 (s);
    _ => regNoun2 (s)
};
Noun Example

-- Masculine nouns end with alif, choTi_hay, ain Transliteration: (a, h, e)

mkN01 : Str -> Noun ;
mkN01 lRka = let end = last (lRka) ;
lRk = if_then_else Str (eq end "e") lRka (tk 1 lRka)
in mkNoun (lRka) (lRk+"E") (lRk+"E")
(lRk+"E") (lRk+"wN") (lRk+"w")
Masc ;
Noun Example

oper

Noun = \{s : Number => Case => Str ; g : Gender\};

mkNoun : (x1,_,_,_,_,x6 : Str) -> Gender -> Noun = \sd,so,sv,pd,po,pv,g -> {
  s = table {
    Sg => table {
      Dir => sd ;
      Obl => so ;
      Voc => sv
    } ;
    Pl => table {
      Dir => pd ;
      Obl => po ;
      Voc => pv
    }
  } ;
  g = g
};
Example Noun: (I\text{R}ka, \text{,boy})

<table>
<thead>
<tr>
<th></th>
<th>Nominative</th>
<th>Oblique</th>
<th>Vocative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singular</td>
<td>I\text{R}ka</td>
<td>I\text{R}kE</td>
<td>I\text{R}kE</td>
</tr>
<tr>
<td>Plural</td>
<td>I\text{R}kE</td>
<td>I\text{R}kw</td>
<td>I\text{R}kwN</td>
</tr>
</tbody>
</table>
Urdu Lexicon

- A Test Lexicon of almost 300 words of different forms (Nouns, Verbs, Pronouns, Adjectives) and closed class words has been built.
Syntax

- So far following functions have been tried

  - DetCN : Det -> CN -> NP ;
  - PPartNP : NP -> V2 -> NP ;
  - AdvNP : NP -> Adv -> NP ;
  - UseN : N -> CN ;
  - MassNP : CN -> NP ;
  - UsePN : PN -> NP ;
  - UsePron : Pron -> NP ;
  - AdjCN : AP -> CN -> CN ;
  - PredetNP : Predet -> NP -> NP ;
  - DetQuantOrd : Quant -> Num -> Ord -> Det ;
  - DetQuant : Quant -> Num -> Det ;
  - DetNP : Det -> NP ;
  - AdvCN : CN -> Adv -> CN ;
  - ComplN2 : N2 -> NP -> CN ;
  - UseN2 : N2 -> CN ;
  - UseV : V -> VP ;
  - ComplV2 : V2 -> NP -> VP ;
  - PredVP : NP -> VP -> Cl ;
Example

• PredVP : NP -> VP -> Cl ;
Example

- PredVP np vp = mkClause np vp ;
Example

```plaintext
mkClause : NP -> VPH -> Clause = \np,vp -> {
  s = \\vt,b =>
  let
    subjagr : NPCase * Agr = case vt of {
      VPImpPast => case vp.subj of {
        VTrans => <NPErg, vp.obj.a> ;
        VTransPost => <NPErg, defaultAgr> ;
        _ => <NPC Dir, np.a>
      };
      _ => <NPC Dir, np.a>
    };
    subj = subjagr.p1 ;
    agr = subjagr.p2 ;
    vps = vp.s ! b ! VPTense vt agr ;
in
    np.s ! subj ++ vps.inf ++ vps.neg ++ vps.inf2 ++ vps.fin

  }
};
```
Demos
Futur Work

• Will continue working on Syntax
• Will try to complete as much as I can before December 2009