A miniresource grammar for Tswana (and Zulu)

Laurette Marais and Laurette Pretorius

CSIR

our future through science

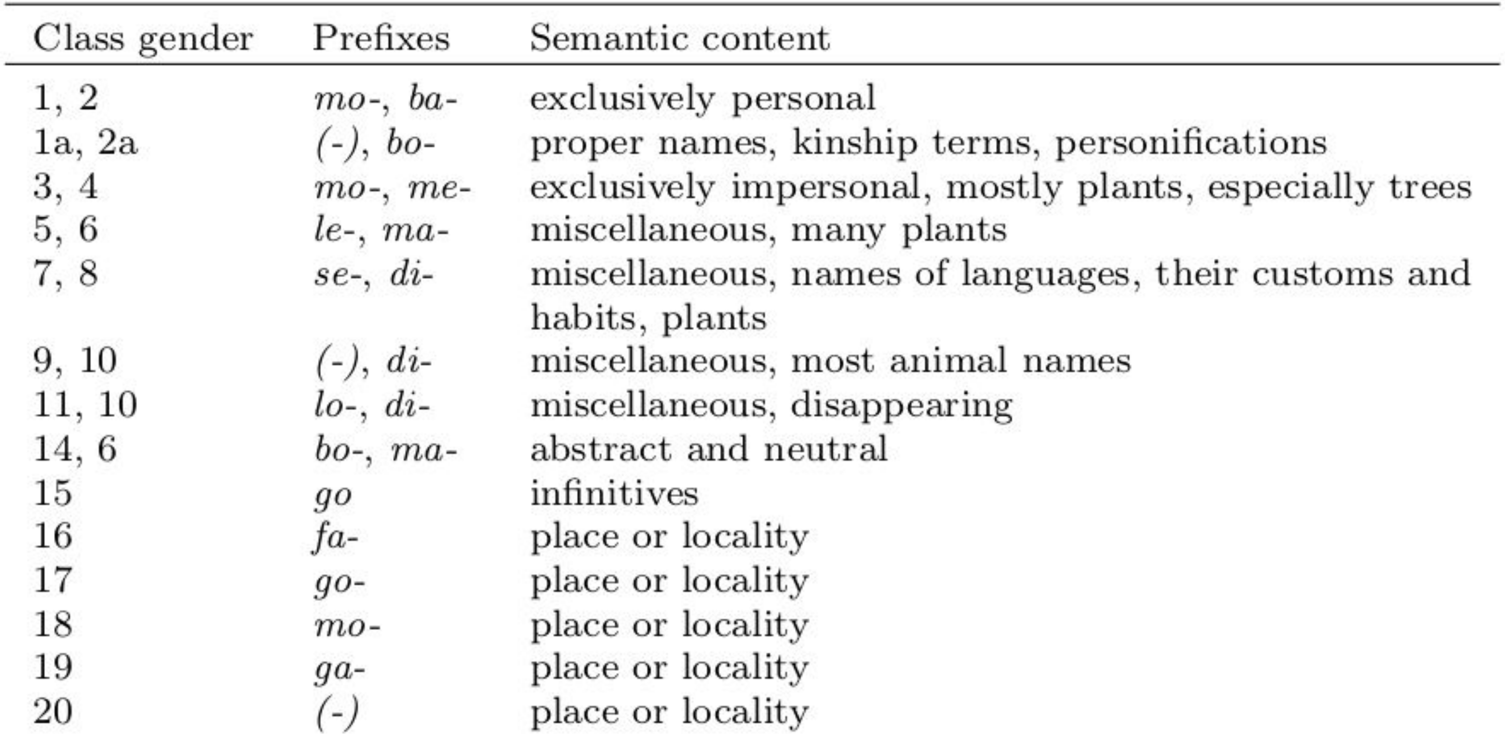
UNISA university of south africa

Academy of African Languages and Science (AALS)

Tswana

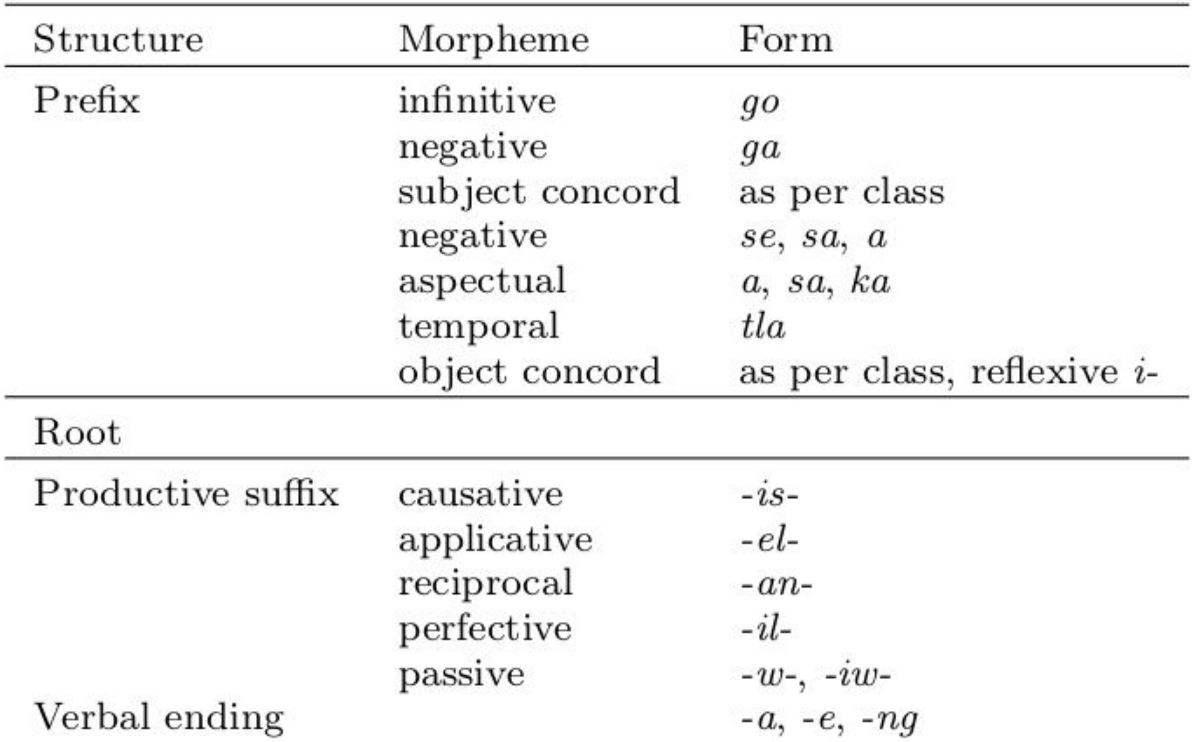
- Southern Bantu language, Sotho-Tswana-group
- SVO
- Agglutinative
- Verb prefixes written disjunctively
- Verb suffixes written conjunctively
- Morphotactics (morpheme sequences)
- Alternation rules (sound changes)

Tswana - noun



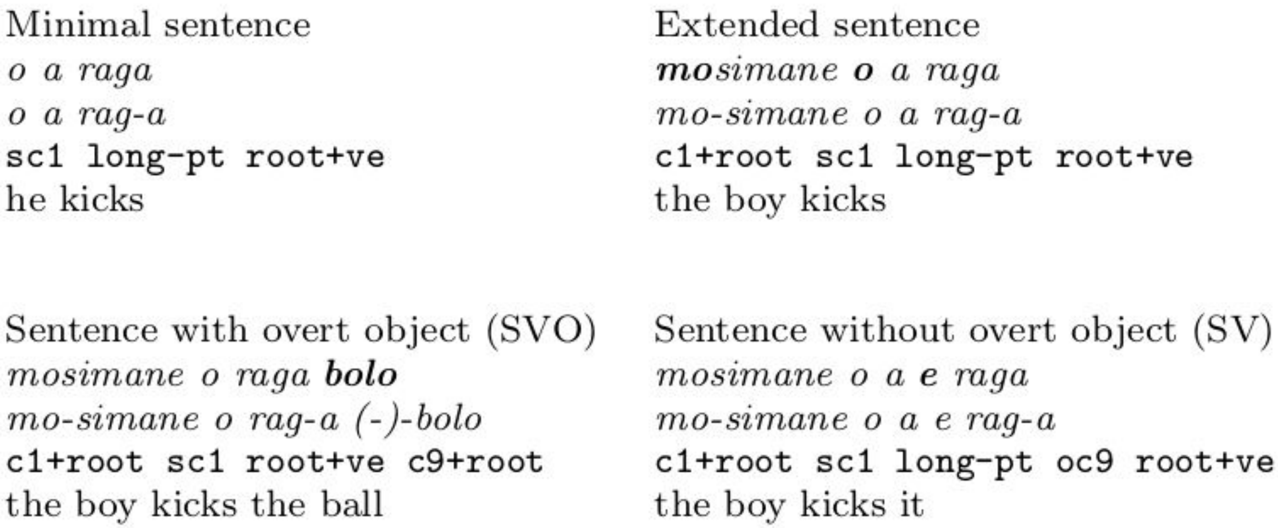
Class gender	Prefixes	Semantic content
1, 2	*mo-*, *ba-*	exclusively personal
1a, 2a	*(-)*, *bo-*	proper names, kinship terms, personifications
3, 4	*mo-*, *me-*	exclusively impersonal, mostly plants, especially trees
5, 6	*le-*, *ma-*	miscellaneous, many plants
7, 8	*se-*, *di-*	miscellaneous, names of languages, their customs and habits, plants
9, 10	*(-)*, *di-*	miscellaneous, most animal names
11, 10	*lo-*, *di-*	miscellaneous, disappearing
14, 6	*bo-*, *ma-*	abstract and neutral
15	*go*	infinitives
16	*fa-*	place or locality
17	*go-*	place or locality
18	*mo-*	place or locality
19	*ga-*	place or locality
20	*(-)*	place or locality

Tswana - verb



Structure	Morpheme	Form
Prefix	infinitive	*go*
	negative	*ga*
	subject concord	as per class
	negative	*se*, *sa*, *a*
	aspectual	*a*, *sa*, *ka*
	temporal	*tla*
	object concord	as per class, reflexive *i*-
Root		
Productive suffix	causative	-*is*-
	applicative	-*el*-
	reciprocal	-*an*-
	perfective	-*il*-
	passive	-*w*-, -*iw*-
Verbal ending		-*a*, -*e*, -*ng*

Tswana - concordial agreement



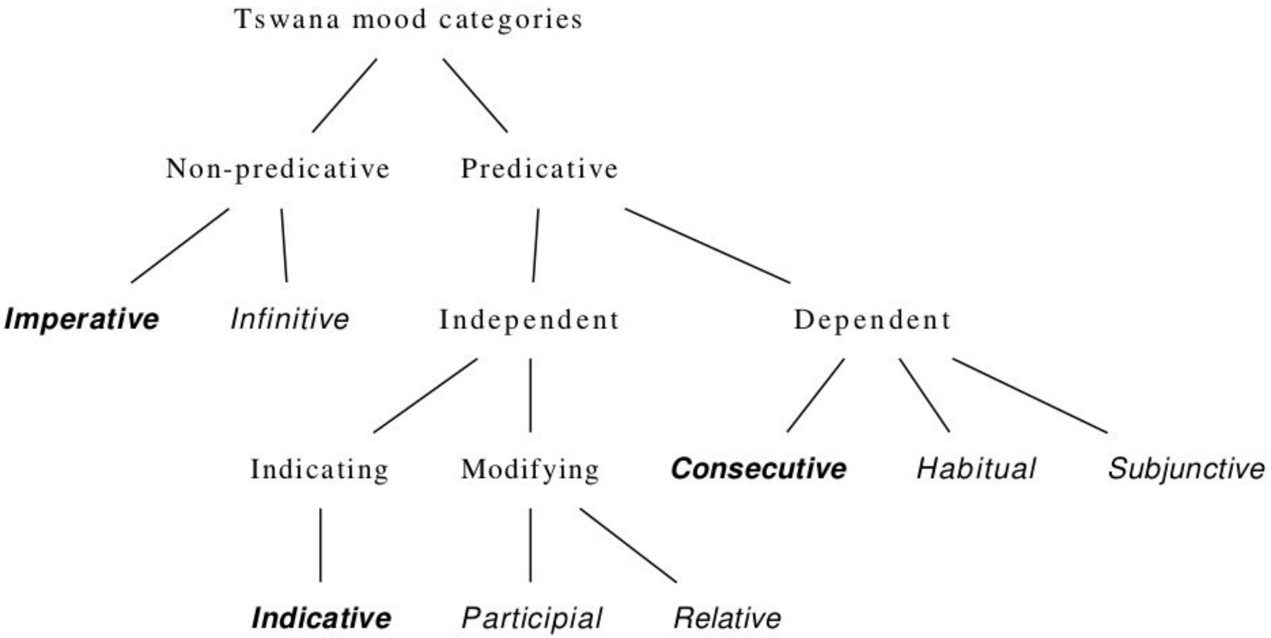
Minimal sentence
o a raga
o a rag-a
`sc1 long-pt root+ve`
he kicks

Extended sentence
***mo**simane **o** a raga*
mo-simane o a rag-a
`c1+root sc1 long-pt root+ve`
the boy kicks

Sentence with overt object (SVO)
*mosimane o raga **bolo***
mo-simane o rag-a (-)-bolo
`c1+root sc1 root+ve c9+root`
the boy kicks the ball

Sentence without overt object (SV)
*mosimane o a **e** raga*
mo-simane o a e rag-a
`c1+root sc1 long-pt oc9 root+ve`
the boy kicks it

Tswana - mood



Tswana mood categories

Non-predicative

Predicative

Imperative

Infinitive

Independent

Dependent

Indicating

Modifying

Consecutive

Habitual

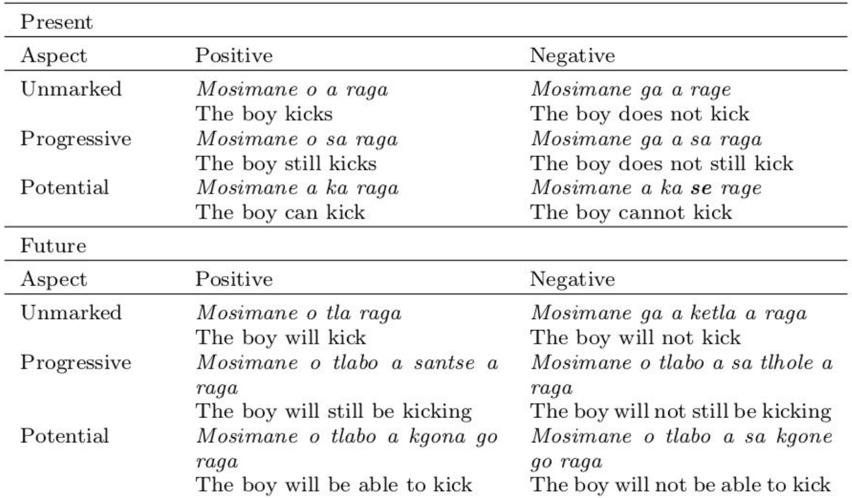
Subjunctive

Indicative

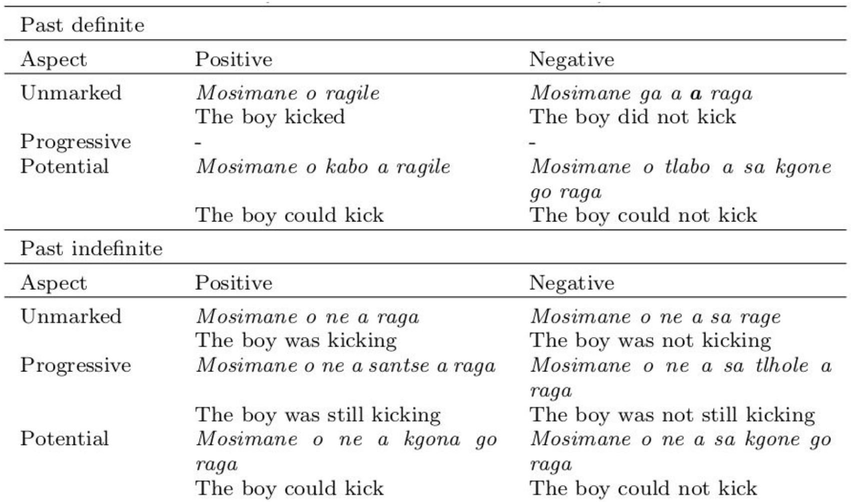
Participial

Relative

Tswana - tense, polarity, aspect

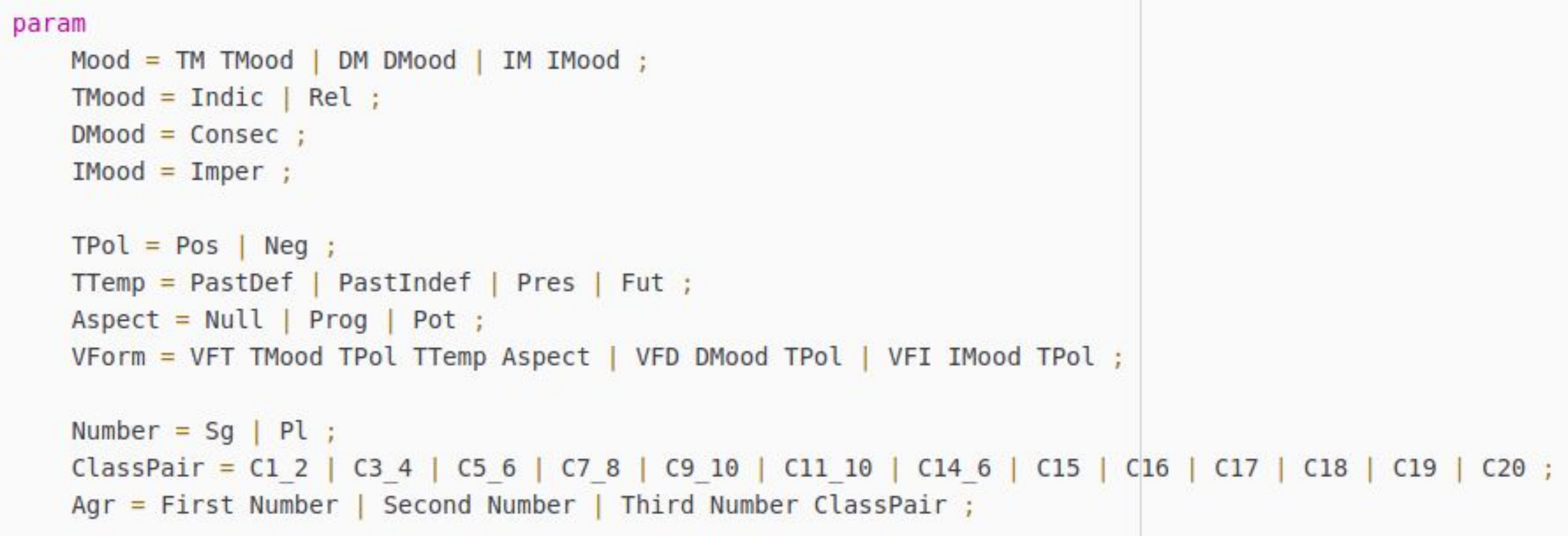


Present		
Aspect	**Positive**	**Negative**
Unmarked	*Mosimane o a raga* The boy kicks	*Mosimane ga a rage* The boy does not kick
Progressive	*Mosimane o sa raga* The boy still kicks	*Mosimane ga a sa raga* The boy does not still kick
Potential	*Mosimane a ka raga* The boy can kick	*Mosimane a ka* ***se*** *rage* The boy cannot kick
Future		
Aspect	**Positive**	**Negative**
Unmarked	*Mosimane o tla raga* The boy will kick	*Mosimane ga a ketla a raga* The boy will not kick
Progressive	*Mosimane o tlabo a santse a raga* The boy will still be kicking	*Mosimane o tlabo a sa tlhole a raga* The boy will not still be kicking
Potential	*Mosimane o tlabo a kgona go raga* The boy will be able to kick	*Mosimane o tlabo a sa kgone go raga* The boy will not be able to kick



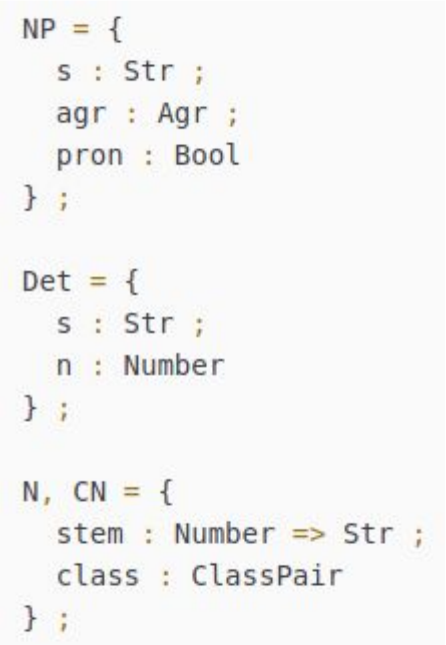
Past definite		
Aspect	**Positive**	**Negative**
Unmarked	*Mosimane o ragile* The boy kicked	*Mosimane ga a* ***a*** *raga* The boy did not kick
Progressive	-	-
Potential	*Mosimane o kabo a ragile* The boy could kick	*Mosimane o tlabo a sa kgone go raga* The boy could not kick
Past indefinite		
Aspect	**Positive**	**Negative**
Unmarked	*Mosimane o ne a raga* The boy was kicking	*Mosimane o ne a sa rage* The boy was not kicking
Progressive	*Mosimane o ne a santse a raga* The boy was still kicking	*Mosimane o ne a sa tlhole a raga* The boy was not still kicking
Potential	*Mosimane o ne a kgona go raga* The boy could kick	*Mosimane o ne a sa kgone go raga* The boy could not kick

Tswana - GF parameters

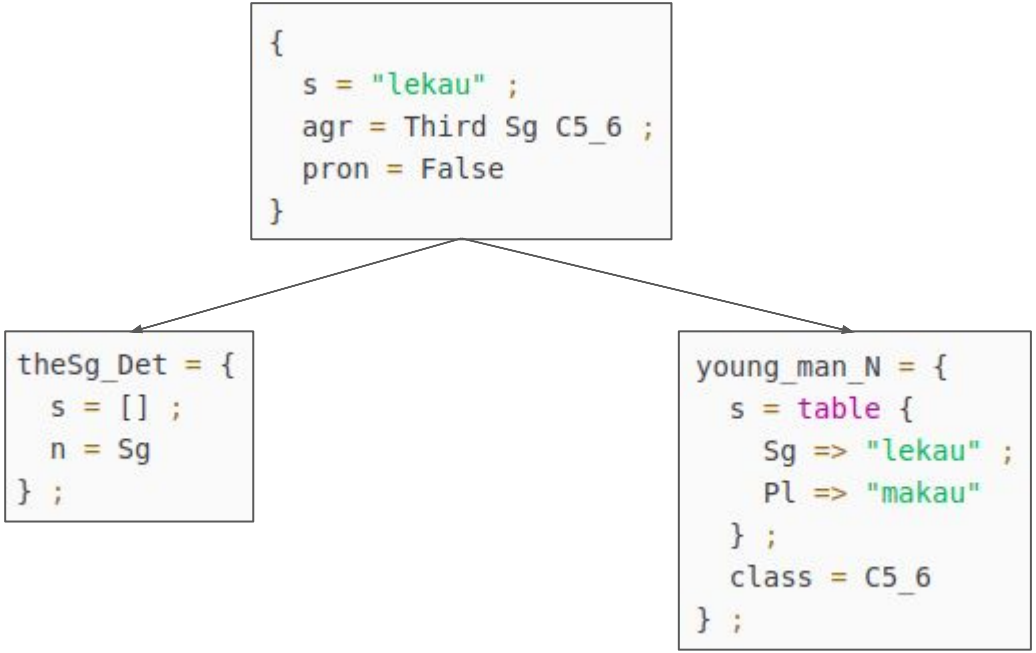


```
param
    Mood = TM TMood | DM DMood | IM IMood ;
    TMood = Indic | Rel ;
    DMood = Consec ;
    IMood = Imper ;

    TPol = Pos | Neg ;
    TTemp = PastDef | PastIndef | Pres | Fut ;
    Aspect = Null | Prog | Pot ;
    VForm = VFT TMood TPol TTemp Aspect | VFD DMood TPol | VFI IMood TPol ;

    Number = Sg | Pl ;
    ClassPair = C1_2 | C3_4 | C5_6 | C7_8 | C9_10 | C11_10 | C14_6 | C15 | C16 | C17 | C18 | C19 | C20 ;
    Agr = First Number | Second Number | Third Number ClassPair ;
```

Tswana - GF noun phrase



```
NP = {
  s : Str ;
  agr : Agr ;
  pron : Bool
} ;

Det = {
  s : Str ;
  n : Number
} ;

N, CN = {
  stem : Number => Str ;
  class : ClassPair
} ;
```

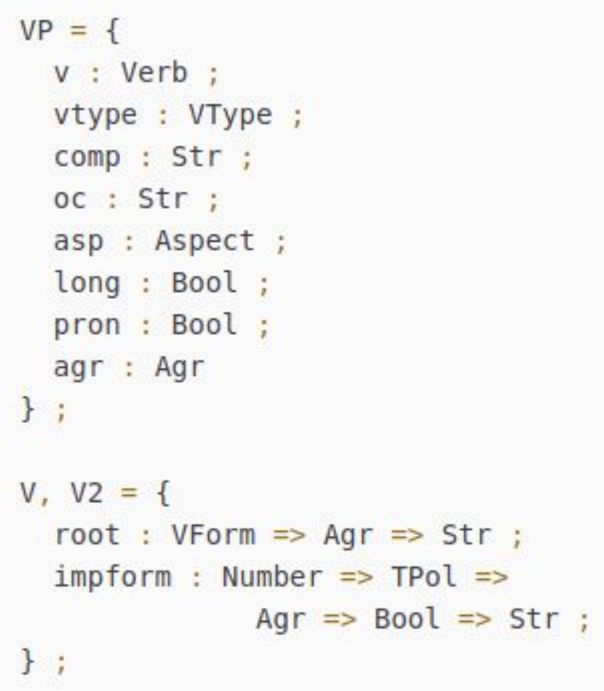


```
{
  s = "lekau" ;
  agr = Third Sg C5_6 ;
  pron = False
}
```

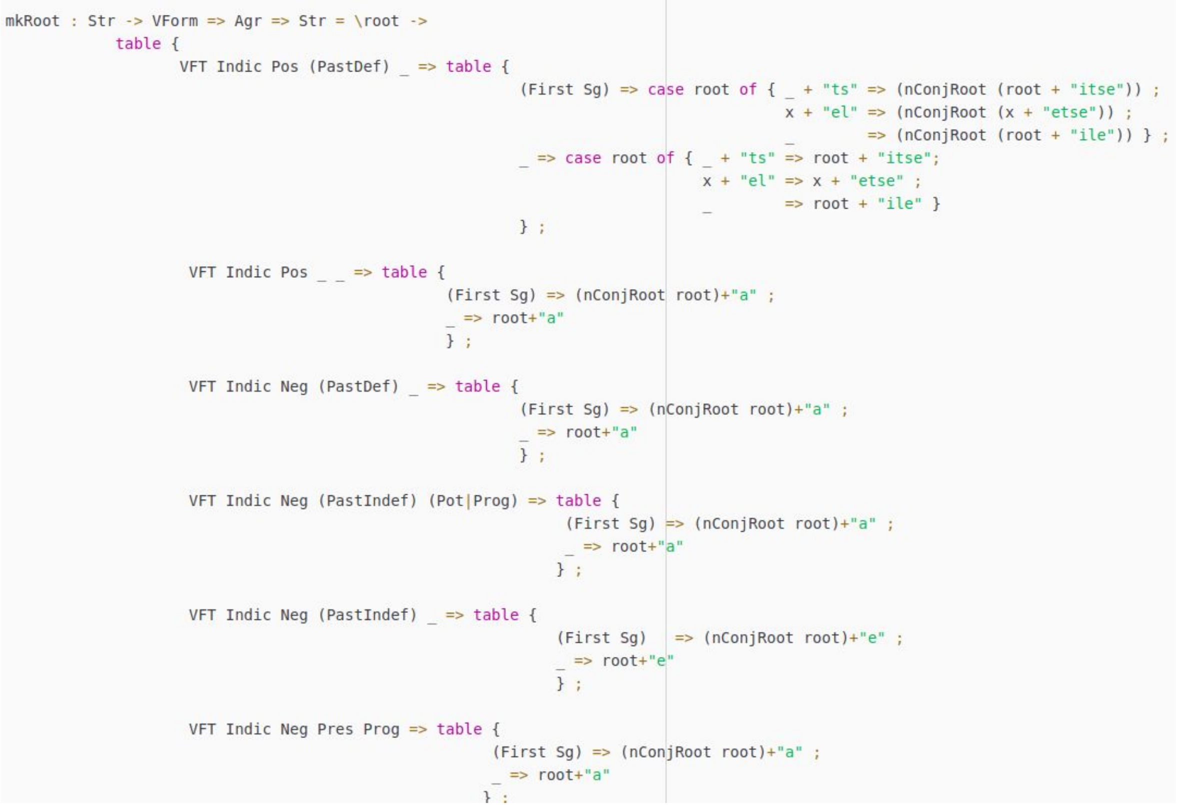
```
theSg_Det = {
  s = [] ;
  n = Sg
} ;
```

```
young_man_N = {
  s = table {
    Sg => "lekau" ;
    Pl => "makau"
  } ;
  class = C5_6
} ;
```

Tswana - GF verb phrase

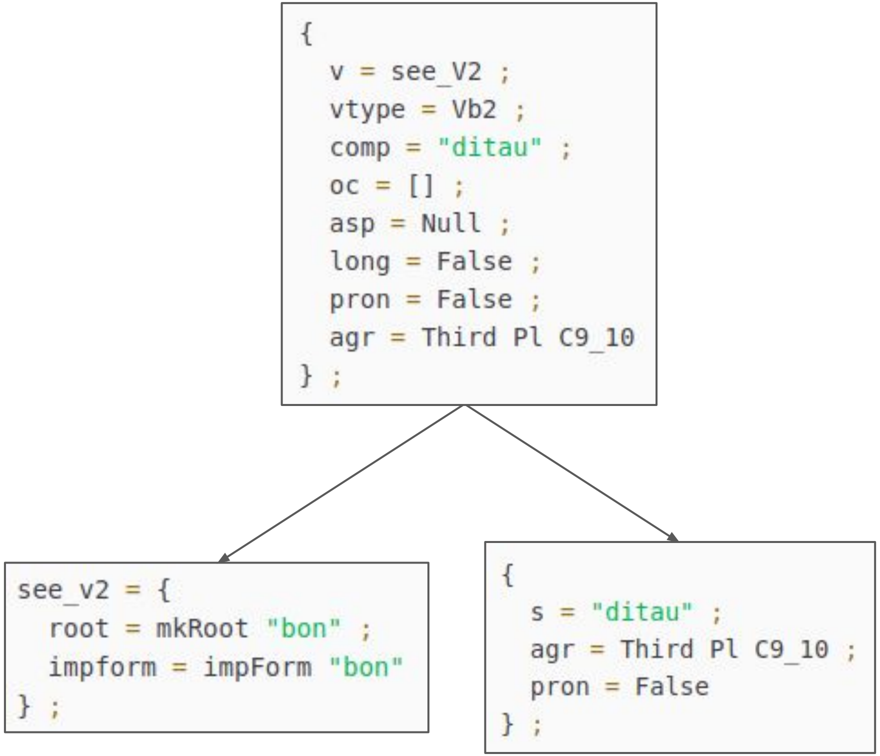


```
VP = {
  v : Verb ;
  vtype : VType ;
  comp : Str ;
  oc : Str ;
  asp : Aspect ;
  long : Bool ;
  pron : Bool ;
  agr : Agr
} ;

V, V2 = {
  root : VForm => Agr => Str ;
  impform : Number => TPol =>
                Agr => Bool => Str ;
} ;
```



```
mkRoot : Str -> VForm => Agr => Str = \root ->
            table {
                   VFT Indic Pos (PastDef) _ => table {
                                                        (First Sg) => case root of { _ + "ts" => (nConjRoot (root + "itse")) ;
                                                                                     x + "el" => (nConjRoot (x + "etse")) ;
                                                                                     _        => (nConjRoot (root + "ile")) } ;
                                                        _ => case root of { _ + "ts" => root + "itse";
                                                                            x + "el" => x + "etse" ;
                                                                            _        => root + "ile" }
                                                        } ;

                    VFT Indic Pos _ _ => table {
                                                (First Sg) => (nConjRoot root)+"a" ;
                                                _ => root+"a"
                                                } ;

                    VFT Indic Neg (PastDef) _ => table {
                                                        (First Sg) => (nConjRoot root)+"a" ;
                                                        _ => root+"a"
                                                        } ;

                    VFT Indic Neg (PastIndef) (Pot|Prog) => table {
                                                             (First Sg) => (nConjRoot root)+"a" ;
                                                             _ => root+"a"
                                                            } ;

                    VFT Indic Neg (PastIndef) _ => table {
                                                            (First Sg)   => (nConjRoot root)+"e" ;
                                                            _ => root+"e"
                                                            } ;

                    VFT Indic Neg Pres Prog => table {
                                                     (First Sg) => (nConjRoot root)+"a" ;
                                                     _ => root+"a"
                                                    } ;
```

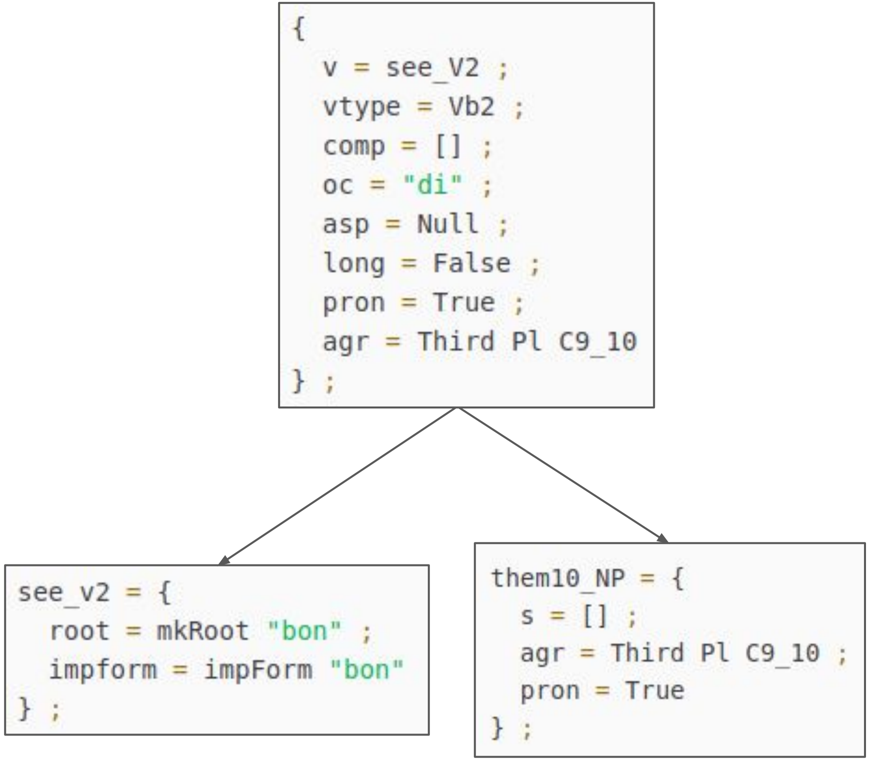
Tswana - GF verb phrase



```
{
  v = see_V2 ;
  vtype = Vb2 ;
  comp = "ditau" ;
  oc = [] ;
  asp = Null ;
  long = False ;
  pron = False ;
  agr = Third Pl C9_10
} ;
```

```
see_v2 = {
  root = mkRoot "bon" ;
  impform = impForm "bon"
} ;
```

```
{
  s = "ditau" ;
  agr = Third Pl C9_10 ;
  pron = False
} ;
```

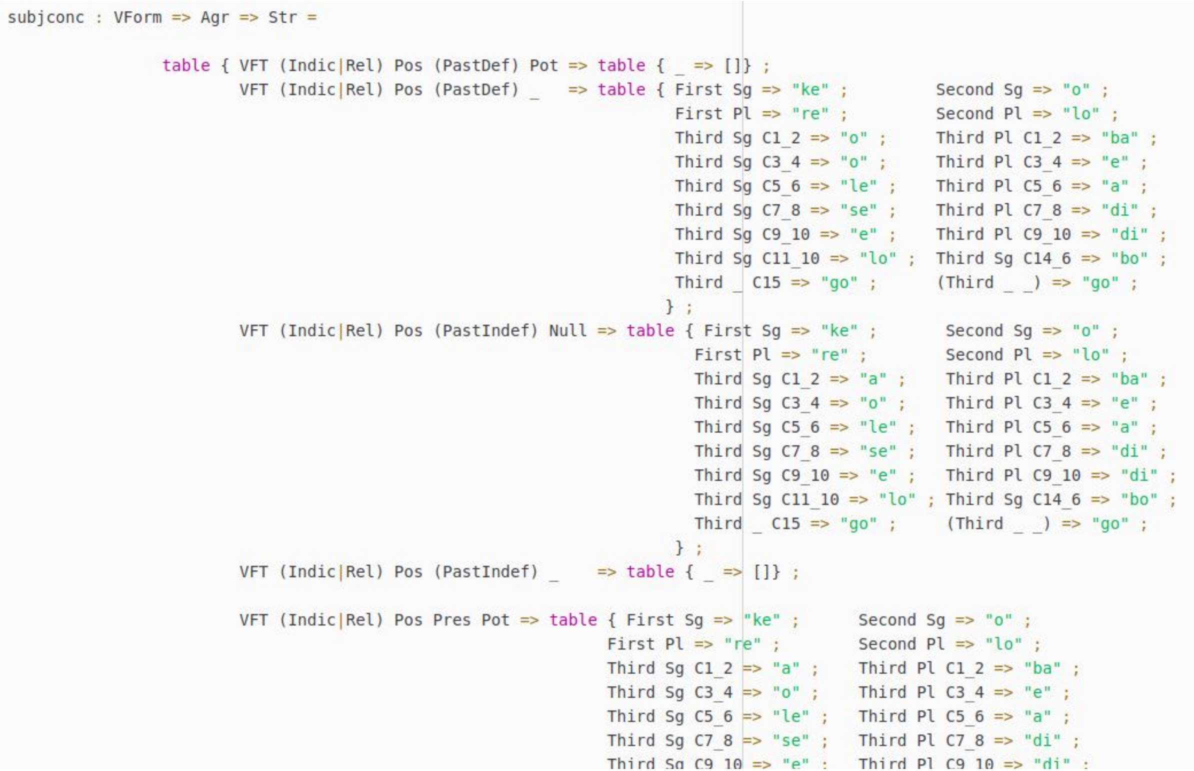


```
{
  v = see_V2 ;
  vtype = Vb2 ;
  comp = [] ;
  oc = "di" ;
  asp = Null ;
  long = False ;
  pron = True ;
  agr = Third Pl C9_10
} ;
```

```
see_v2 = {
  root = mkRoot "bon" ;
  impform = impForm "bon"
} ;
```

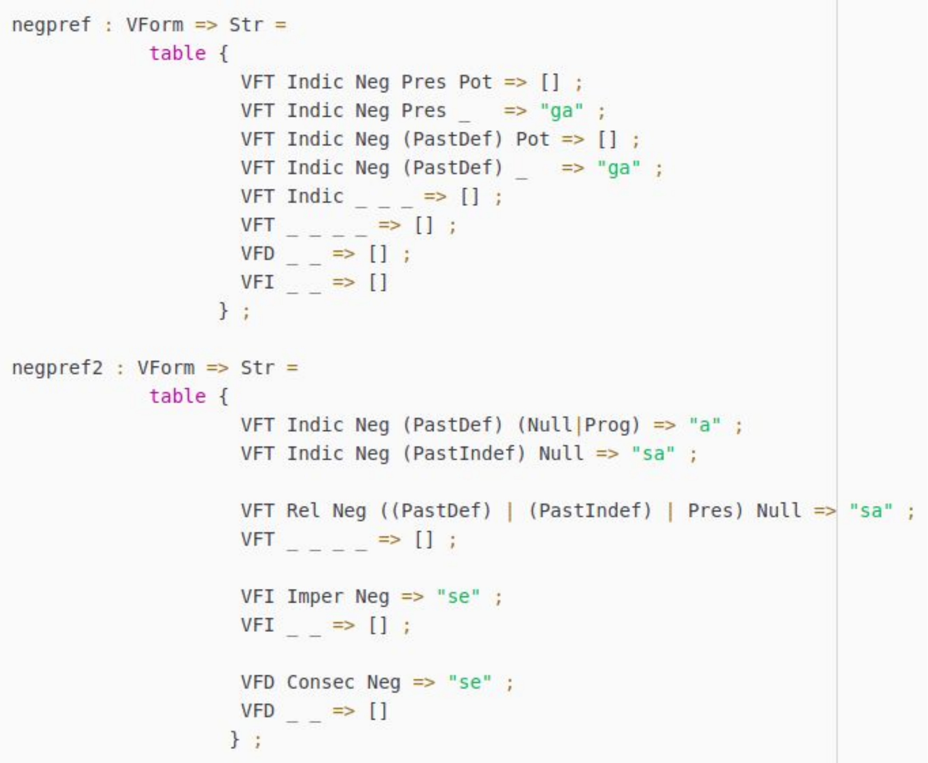
```
them10_NP = {
  s = [] ;
  agr = Third Pl C9_10 ;
  pron = True
} ;
```

Tswana - GF clause



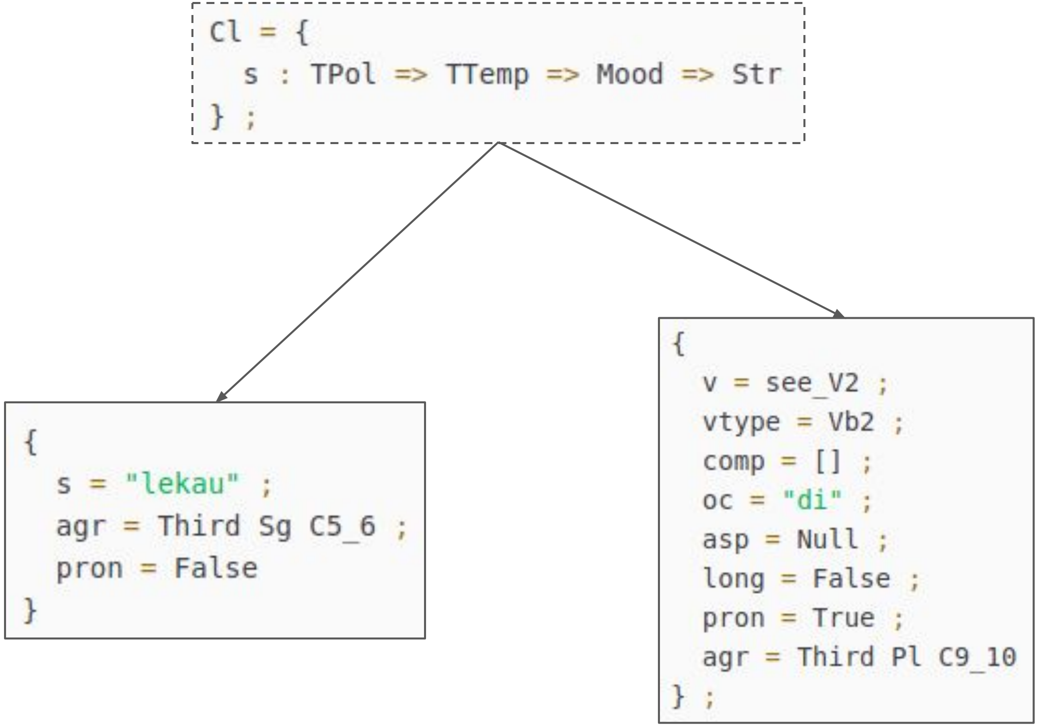
```
subjconc : VForm => Agr => Str =

                table { VFT (Indic|Rel) Pos (PastDef) Pot => table { _ => []} ;
                        VFT (Indic|Rel) Pos (PastDef) _   => table { First Sg => "ke" ;         Second Sg => "o" ;
                                                                     First Pl => "re" ;         Second Pl => "lo" ;
                                                                     Third Sg C1_2 => "o" ;      Third Pl C1_2 => "ba" ;
                                                                     Third Sg C3_4 => "o" ;      Third Pl C3_4 => "e" ;
                                                                     Third Sg C5_6 => "le" ;     Third Pl C5_6 => "a" ;
                                                                     Third Sg C7_8 => "se" ;     Third Pl C7_8 => "di" ;
                                                                     Third Sg C9_10 => "e" ;     Third Pl C9_10 => "di" ;
                                                                     Third Sg C11_10 => "lo" ;  Third Sg C14_6 => "bo" ;
                                                                     Third _ C15 => "go" ;      (Third _ _) => "go" ;
                                                                    } ;
                        VFT (Indic|Rel) Pos (PastIndef) Null => table { First Sg => "ke" ;       Second Sg => "o" ;
                                                                       First Pl => "re" ;        Second Pl => "lo" ;
                                                                       Third Sg C1_2 => "a" ;     Third Pl C1_2 => "ba" ;
                                                                       Third Sg C3_4 => "o" ;     Third Pl C3_4 => "e" ;
                                                                       Third Sg C5_6 => "le" ;    Third Pl C5_6 => "a" ;
                                                                       Third Sg C7_8 => "se" ;    Third Pl C7_8 => "di" ;
                                                                       Third Sg C9_10 => "e" ;    Third Pl C9_10 => "di" ;
                                                                       Third Sg C11_10 => "lo" ; Third Sg C14_6 => "bo" ;
                                                                       Third _ C15 => "go" ;     (Third _ _) => "go" ;
                                                                     } ;
                        VFT (Indic|Rel) Pos (PastIndef) _    => table { _ => []} ;

                        VFT (Indic|Rel) Pos Pres Pot => table { First Sg => "ke" ;      Second Sg => "o" ;
                                                              First Pl => "re" ;        Second Pl => "lo" ;
                                                              Third Sg C1_2 => "a" ;     Third Pl C1_2 => "ba" ;
                                                              Third Sg C3_4 => "o" ;     Third Pl C3_4 => "e" ;
                                                              Third Sg C5_6 => "le" ;    Third Pl C5_6 => "a" ;
                                                              Third Sg C7_8 => "se" ;    Third Pl C7_8 => "di" ;
                                                              Third Sg C9_10 => "e" ;    Third Pl C9_10 => "di" ;
```

Tswana - GF clause



```
negpref : VForm => Str =
            table {
                    VFT Indic Neg Pres Pot => [] ;
                    VFT Indic Neg Pres _    => "ga" ;
                    VFT Indic Neg (PastDef) Pot => [] ;
                    VFT Indic Neg (PastDef) _    => "ga" ;
                    VFT Indic _ _ _ => [] ;
                    VFT _ _ _ _ => [] ;
                    VFD _ _ => [] ;
                    VFI _ _ => []
                  } ;

negpref2 : VForm => Str =
            table {
                    VFT Indic Neg (PastDef) (Null|Prog) => "a" ;
                    VFT Indic Neg (PastIndef) Null => "sa" ;

                    VFT Rel Neg ((PastDef) | (PastIndef) | Pres) Null => "sa" ;
                    VFT _ _ _ _ => [] ;

                    VFI Imper Neg => "se" ;
                    VFI _ _ => [] ;

                    VFD Consec Neg => "se" ;
                    VFD _ _ => []
                   } ;
```

Tswana - GF clause

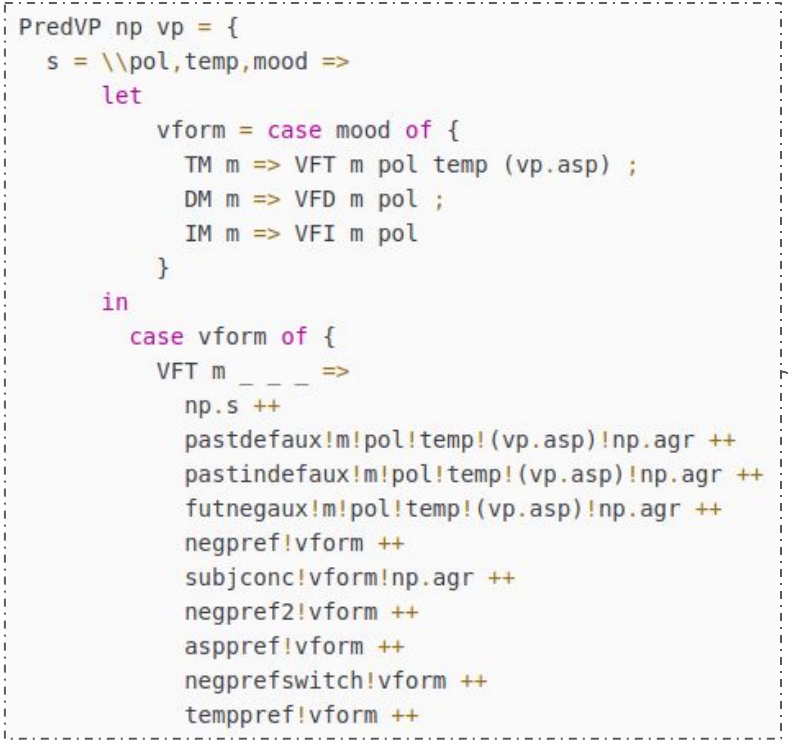


```
Cl = {
  s : TPol => TTemp => Mood => Str
} ;
```

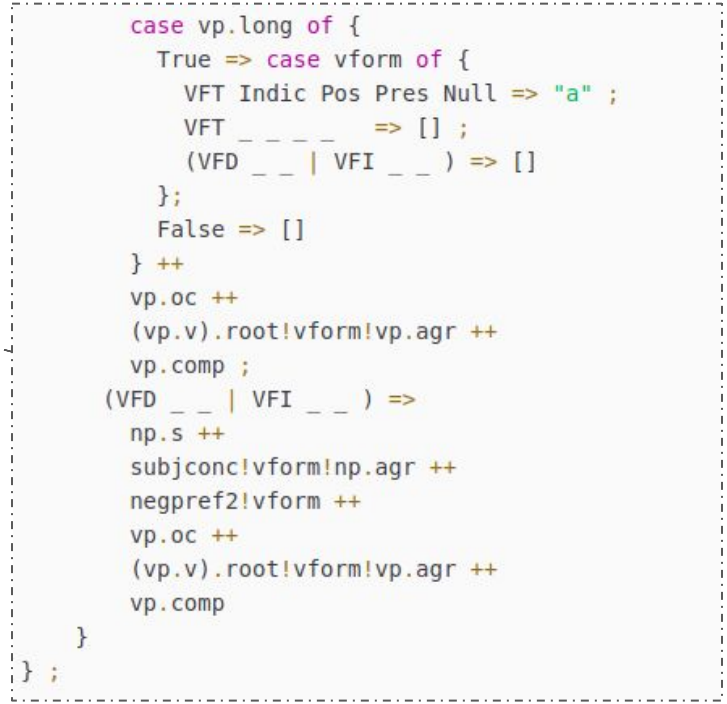
```
{
  s = "lekau" ;
  agr = Third Sg C5_6 ;
  pron = False
}
```

```
{
  v = see_V2 ;
  vtype = Vb2 ;
  comp = [] ;
  oc = "di" ;
  asp = Null ;
  long = False ;
  pron = True ;
  agr = Third Pl C9_10
} ;
```

Tswana - GF clause



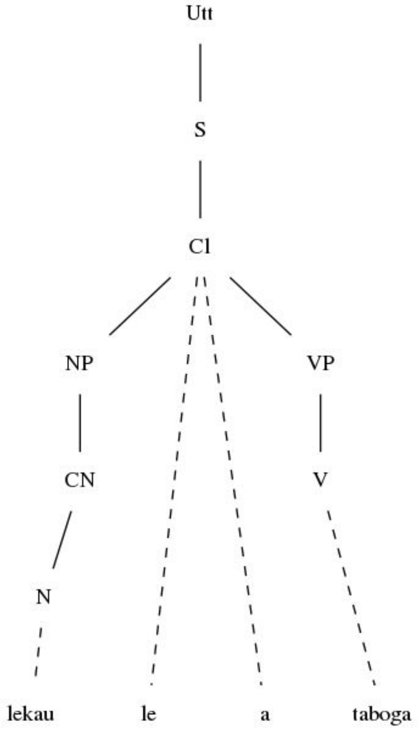
```
PredVP np vp = {
  s = \\pol,temp,mood =>
      let
          vform = case mood of {
            TM m => VFT m pol temp (vp.asp) ;
            DM m => VFD m pol ;
            IM m => VFI m pol
          }
      in
        case vform of {
          VFT m _ _ _ =>
            np.s ++
            pastdefaux!m!pol!temp!(vp.asp)!np.agr ++
            pastindefaux!m!pol!temp!(vp.asp)!np.agr ++
            futnegaux!m!pol!temp!(vp.asp)!np.agr ++
            negpref!vform ++
            subjconc!vform!np.agr ++
            negpref2!vform ++
            asppref!vform ++
            negprefswitch!vform ++
            temppref!vform ++
```



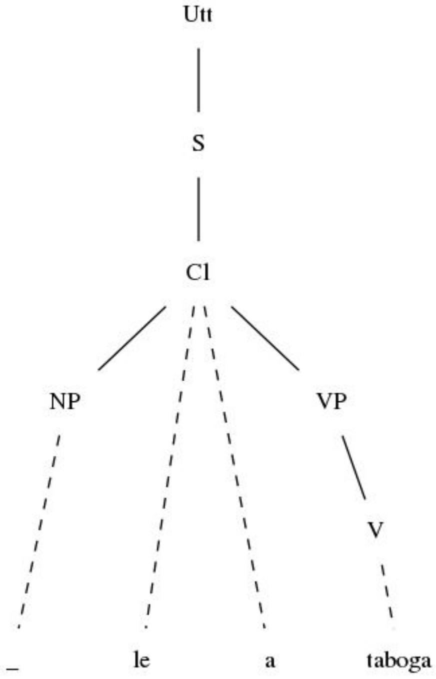
```
        case vp.long of {
          True => case vform of {
            VFT Indic Pos Pres Null => "a" ;
            VFT _ _ _ _    => [] ;
            (VFD _ _ | VFI _ _ ) => []
          };
          False => []
        } ++
        vp.oc ++
        (vp.v).root!vform!vp.agr ++
        vp.comp ;
      (VFD _ _ | VFI _ _ ) =>
        np.s ++
        subjconc!vform!np.agr ++
        negpref2!vform ++
        vp.oc ++
        (vp.v).root!vform!vp.agr ++
        vp.comp
    }
} ;
```

Tswana - examples

the young man runs

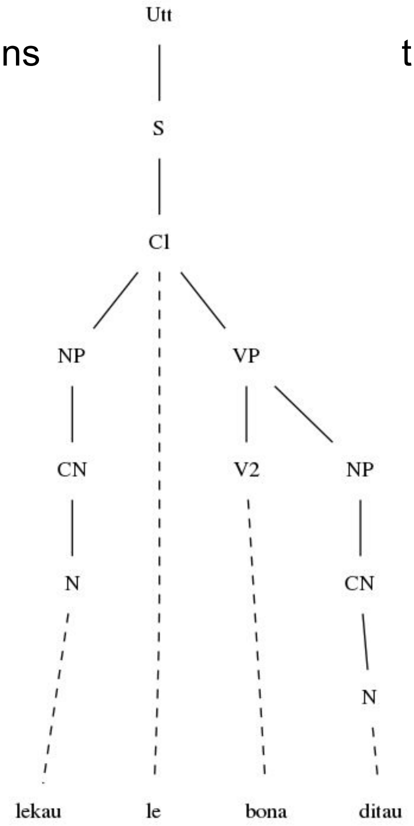


he runs

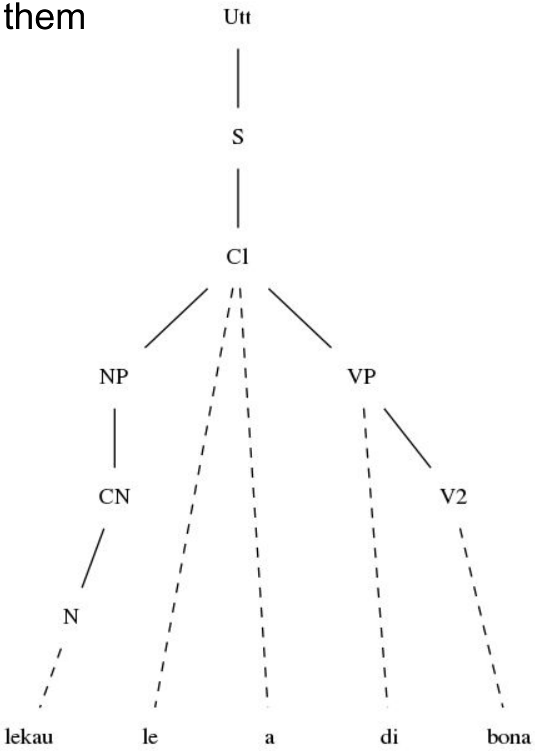


Tswana - examples

the young man sees the lions

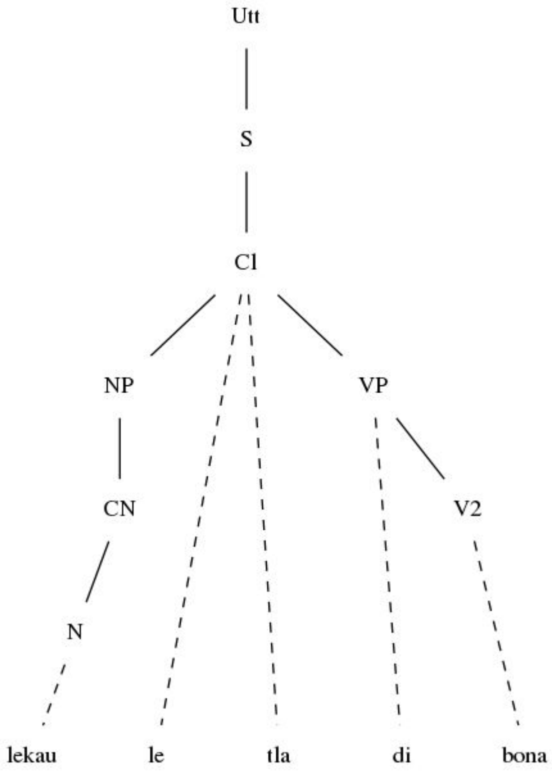


the young man sees them

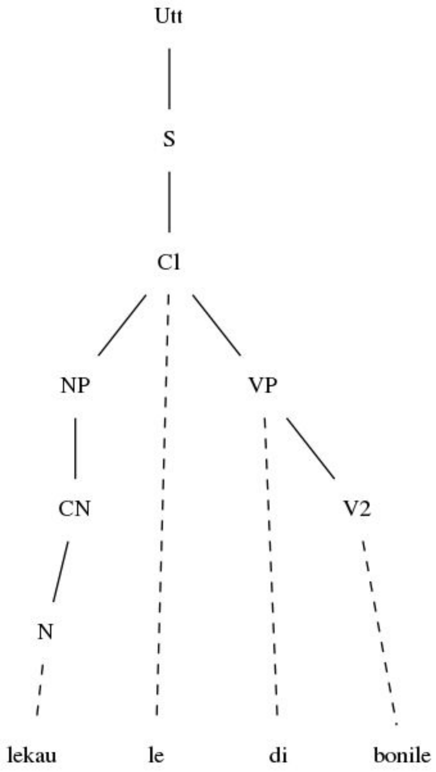


Tswana - examples

the young man will see them

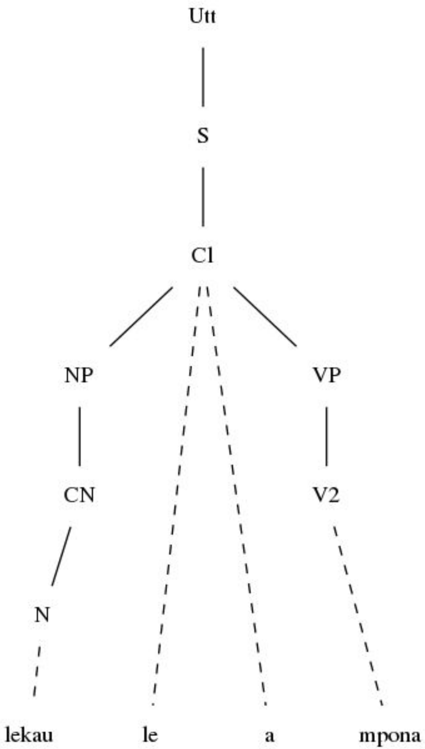


the young man saw them

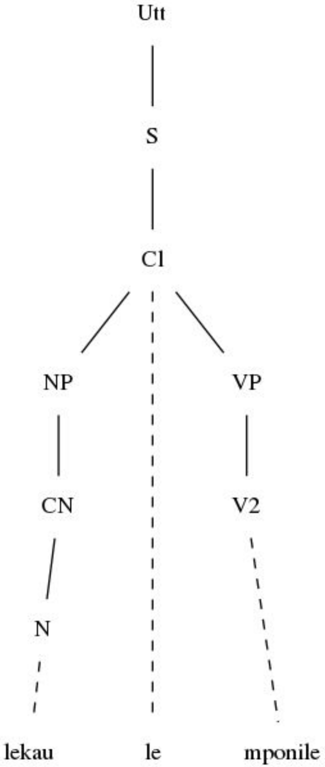


Tswana - examples

the young man sees me



the young man saw me



Zulu - code demo