Verb morphology of Hebrew and Maltese — Towards an open source type theoretical resource grammar in GF

Dana Dannélls
Department of Swedish Language, University of Gothenburg
dana.dannells@svenska.gu.se

John J. Camilleri
Department of Intelligent Computer Systems, University of Malta
jcam0003@um.edu.mt

Our work

Aim to increase the coverage of Semitic languages in the Grammatical Framework (GF) resource grammar library. In this paper we describe and compare the implementations of Hebrew and Maltese verb morphologies. We found that although Hebrew and Maltese share some common characteristics in their morphology, it seems difficult to generalize morphosyntactic rules across Semitic verbs when the focus is towards computational linguistics motivated lexicons.

The Grammatical Framework

The Grammatical Framework is a grammar formalism for multilingual grammars and their applications (Ranta,2004). It has a Resource Grammar Library (Ranta, 2009) that is designed to gather and encapsulate morphological and syntactic rules of languages, which normally require expert knowledge, and make them available for non-expert application programmers by defining a complete set of morphological paradigms and a syntax for each language. Currently, the GF's library contains grammar rules for seventeen languages.

The grammar design

Modern Hebrew

Parameters:

```
  VPerNumGen = Vp1Sg | Vp1Pl | Vp2Sg Gender |
  Vp2Pl Gender | Vp3Sg Gender | Vp3Pl Gender ;
```

Operations:

```
      Pattern : Type={C1, C1C2, C2C3, C3 : Str};
      Root : Type={C1,C2,C3 : Str};
      Root4 : Type=Root ** {C4 : Str};
      Verb : Type={s : Tense => VPerNumGen ) Str };
     mkVPaal : Str ! Verb = \v ! let root = getRoot v in {s = table {
        Perf => table {
                 Vp1Sg => appPattern root C1aC2aC3ty ;
                 Vp1Pl => appPattern root C1aC2aC3nw ;
                 Vp2Sg Masc => appPattern root C1aC2aC3th ;
                 Vp2Sg Fem => appPattern root C1aC2aC3t ;
                 Vp2Pl Masc => appPattern root C1aC2aC3tM ; …}
        Imperf => table { …} } };
     getRoot : Str ! Root = \s -> case s of {
      C1@? + C2@? + C3 => {C1 = C1 ; C2 = C2 ; C3 = C3} };
     appPattern : Root ! Pattern ! Str = \r,p ->
      p.C1 + r.C1 + p.C1C2 + r.C2 + p.C2C3 + r.C3 + p.C3 ;
```

Patterns:

```
      C1aC2aC3ty = {C1=""; C1C2=""; C2C3=""; C3="ty"};
      C1aC2aC3nw = {C1=""; C1C2=""; C2C3=""; C3="nw"};
      C1aC2aC3th = {C1=""; C1C2=""; C2C3=""; C3="th"};
```

Lexicon:

```
      write_V2 = mkVPaal "ktb" ;
      pray_V = mkVHitpael "pll" ;
      sleep_V = mkVPaalGroup3_py "ysn";
```

Common parameters:

```
  Number = Sg | Pl ;
  Gender = Masc | Fem ;
  Case = Nom | Acc | Gen ;
  Person = P1 | P2 | P3 ;
  Voice = Active | Passive ;
  Tense = Perf | Part | Imperf ;
```

Hebrew	**Maltese**
mkVPaal "ktb"	mkVerb "kiteb"
Perfect	
Vp1Sg ⇒ "ktbty"	(Per1 Sg) ⇒ "ktibt"
Vp1Pl ⇒ "ktbnw"	(Per1 Pl) ⇒"ktibna"
Vp2SgMasc ⇒ "ktbt"	(Per2 Sg) ⇒"ktibt"
Vp2SgFem ⇒ "ktbt"	
Vp2PlMasc ⇒ "ktbtM"	(Per2 Pl) ⇒"ktibtu"
Vp2PlFem ⇒ "ktbtN"	
Vp3SgMasc ⇒ "ktb"	(Per3Sg Masc) ⇒ "kiteb"
Vp3SgFem ⇒ "ktbh"	(Per3Sg Fem) ⇒ "kitbet"
Vp3PlMasc ⇒ "ktbw"	Per3Pl ⇒ "kitbu"
Vp3PlFem ⇒ "ktbw"	
Imperfect	
Vp1Sg ⇒ "Aktwb"	(Per1 Sg) ⇒ "nikteb"
Vp1Pl ⇒ "nktwb"	(Per1 Pl) ⇒ "niktbu"
Vp2SgMasc ⇒ "tktwb"	(Per2 Sg) ⇒ "tikteb"
Vp2SgFem ⇒ "tktby"	
Vp2PlMasc ⇒ "tktbw"	(Per2 Pl) ⇒ "tiktbu"
Vp2PlFem ⇒ "tktbw"	
Vp3SgMasc ⇒ "yktwb"	(Per3Sg Masc) ⇒ "jikteb"
Vp3SgFem ⇒ "tktwb"	(Per3Sg Fem) ⇒ "tikteb"
Vp3PlMasc ⇒ "yktbw"	Per3Pl ⇒ "jiktbu"
Vp3PlFem ⇒ "yktbw"	

Maltese

Parameters:

```
   VType = Strong | Defective | Weak | Hollow |Double ;
   VOrigin = Semitic | Romance ;
   VForm = VPerf PerGenNum | VImpf PerGenNum | VImp Number ;
```

Operations:

```
      Pattern : Type = {v1, v2 : Str} ;
      Root : Type = {K, T, B, L : Str} ;
      Verb : Type = {s : VForm ) Str ; t : VType ; o : VOrigin} ;
    mkVerb : Str ! Verb = \mamma -> let class = classifyVerb mamma in
     case class.t of {
          Strong => mkStrong class.r class.p ;
          Defective => mkDefective class.r class.p ;
          Quad => mkQuad class.r class.p ; …} ;
    classifyVerb : Str -> { t:VType ; r:Root ; p:Pattern } = \mamma ->
       case mamma of {
            K@#Consonant + v1@#Vowel + T@#Consonant +
   B@#Consonant +
            v2@#Vowel + L@#Consonant =>
            { t=Quad ; r={ K=K ; T=T ; B=B ; L=L } ; p={ v1=v1 ; v2=v2 } } ; }
```

Lexicon:

```
     write_V2 = mkVerb "kiteb" ;
     pray_V = mkVerb "talab" "itlob" "itolbu";
```

Coverage

The core syntax covers simple syntactic constructions including predication rules which are built from noun and verb phrases (around 13 categories and 22 construction functions). The lexicons cover around 20 verbs and 10 nouns in each language. The Maltese verb morphology covers the root groups: strong, defective and quadriliteral. In Hebrew, the strong verb paradigms and five weak verb paradigms in binyan pa'al are covered.

Advantages

➢ Strings are formed by records, and not through concatenation.
➢ Once the core lexicon is determined, it is possible to automatically acquire lexical entries from exiting lexical resources.

Future work

➢ Employ semantic markings in the lexicons of the Semitic languages.
➢ Focus on semantic aspects of morphological processing.

Conclusions

In this work, the design decisions taken by the programmers are based on different points of arguments concerning the division of labour between a linguistically trained grammarian and a lexicographer. One fundamental question that rises from our implementations is to what extent we can generalize the concrete syntaxes of Semitic languages.

References

•R. Frost, A. Deutsch, and K. I. Forster. 2000. Decomposing morphologically complex words in a nonlinear morphology. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 26(3):751–765.
•Gideon Goldberg. 1994. Principles of Semitic word structure. In G. Goldberg and S. Raz, editors, *In Semantic and ushitic studies*, pages 29–64. Wiesbaden: Harrassowitz.
•Robert D. Hoberman and M. Aronoff, 2003. *The verbal morphology of Maltese: From Semitic to Romance*, chapter 3, pages 61–78. Amsterdam: John Benjamins.
•Alon Itai and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98, March.
•Aarne Ranta. 2004. Grammatical framework, a type theoretical grammar formalism. *Journal of Functional Programming*, 14(2):145–189.
•Aarne Ranta. 2009. The GF resource grammar library. *The on-line journal Linguistics in Language Technology*, 2(2).
•M. Rosner, J. Caruana, and R. Fabri. 1999. *Linguistic and computational aspects of maltilex*. In Arabic Translation and Localisation Symposium: Proceedings of the Workshop, pages 2—-10, Tunis.
•Shlomo Yona and ShulyWintner. 2008. A finite-state morphological grammar of Hebrew. *Natural Language Engineering*,14(2):173–190.

LREC workshop on Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages, 17 May 2010