Ohad Stossel

Finite State Morphology – Final project

The Hebrew verb system in xfst

Introduction

For my final project in the course I decided to base my project on the Hebrew verb system. The verb system in Hebrew is Semitic: Hence, roots made of 3 consonants are being put into one of 7 binyanim (templates of vowels and additional consonents) in order to create a final word. Every binyan in the verb system can be "bent" according to specific rules in order to create agreement of tense, number, person and gender.

For this project I built a system that accepts verbs in Binyan Pa'al (also known as Binyan Kal) that are made of non-exceptional roots, and a few kinds of exceptional roots. The system is built as expandable, so there is no problem in adding other kinds of roots and more binyanim.

Alphabet

Since Hebrew has a different, non-ASCII writing system, I matched ASCII symbols to it. For the consonants, I matched it letter-to-letter, with no regards of the original sound:

***Letter:***

***Symbol:***

***sound(s):***

***Letter:***

***Symbol:***

***sound(s):***

Since Hebrew writing has no clear vowel system (the letters א,ה,ו,י are used to convey vowels in certain occasions but not in others), I also added the vowels a,e,i,o,u, in order to represent the vowels in the program.

The Hebrew verb system

The Hebrew verb system is Semitic. That is, verbs are created by combining roots – series of 3/4 consonants – and binyanim – templates for these roots, that add vowels and additional consonants between them. (for example, the root LBJ can be "labaj" (he wore [one piece of clothing]), "hilbij" (he put clothes on someone else) or "hitlabej" (he got dressed)).

Modern Hebrew contains 7 binyanim. For this exercise I used binyan Pa'al (also known as "Kal"). This binyan contains most of the basic words of the language.

Binyan Pa'al and its derivations

The binyans in Hebrew work in accordance to tense, person, gender and number. The derivations of binyan Pa'al can be seen by the following tables:

|  |  |  |  |
| --- | --- | --- | --- |
| Fut | Pres | Past | Sg. Masc. |
| eXXaX | XoXeX | XaXaXti | 1P |
| tiXXaX | XoXeX | XaXaXta | 2P |
| yiXXaX | XoXeX | XaXaX | 3P |

|  |  |  |  |
| --- | --- | --- | --- |
| Fut | Pres | Past | Sg. Fem. |
| eXXaX | XoXeXet | XaXaXti | 1P |
| tiXXeXi | XoXeXet | XaXaXt | 2P |
| tiXXaX | XoXeXet | XaXXa | 3P |

|  |  |  |  |
| --- | --- | --- | --- |
| Fut | Pres | Past | Pl. Fem. |
| niXXaX | XoXeXot | XaXaXnu | 1P |
| tiXXeXu | XoXeXot | XaXaXten | 2P |
| tiXXaXna | XoXeXot | XaXXu | 3P |

|  |  |  |  |
| --- | --- | --- | --- |
| Fut | Pres | Past | Pl.Masc |
| niXXaX | XoXeXim | XaXaXnu | 1P |
| tiXXeXu | XoXeXim | XaXaXtem | 2P |
| yiXXeXu | XoXeXim | XaXXu | 3P |

Note that there are many duplications: for example, present tense does not change according to person, and 1st person does not change according to gender. However, I treat them all as separated words for the sake of simplicity.

The Program

The script: Heb1.xfst

I use the "define" function to list all vowels. This is very important for the rule file later.

I add "list C" and "list V" to use the xfst methods for Semitic languages.

Other than that, the script is basic and speaks for itself.

The lexc file: heb-verbs.lexc

The process of the lexc file is according to the following:

* First, a binyan is picked (only Pa'al is available for now). A marking of it is added both to lower and to upper side for the rule file. All lexicons from now on will be marked with 1 – the binyan number – because the process is unique for this binyan.

Each binyan leaves a space for the 3 consonants by 3 Cs. Each C has one the letters K, T, L before it, to continue marking the first, second and third consonant, even after the substitution will be made. That in order to cover exceptions by the rule file.

* The finite state mode is opened by .%m.
* An appropriate root is picked. (Not all roots fit to all binyans). The flag diacritic the roots have will be explained later.
* A tense is picked. A marking of it is added both to lower and to upper side (by a number) for the rule file.
* All other features (person, gender and number) are picked. The 3 letters of the root will be entered into the Cs, with K, T, L marking the consonants from the root for the rule file.
* Some forms of the future tense contain a V space and continue to Mleft to fill it according to the flag diacritic. (see later)
* The finite state part is closed and so is the word.

The high irregularity of the future tense

The future tense causes some problems. As shown in the tables, the second vowel is often 'a', but very often it can be 'o', like in the root {pgj} (therefore: I will wear = elb**a**j, but I will meet = efg**o**j). This is completely random, depending on the root itself and not on phonological change based on consonants like in most other exceptions in Hebrew.

The solution is achieved by adding flag diacritics to the roots, declaring the vowel it needs to pick in future tense. The problematic forms in future tense leave the vowel undetermined and moved to Mleft, where the appropriate vowel is pushed in.

Managing phonological exceptions by the rule file

Most of the exceptions in Hebrew derive from certain consonants appearing in certain places in the roots. Each case would affect different forms. For example, every root that has 'y' as first consonant will change some forms of the binyan in the same way. Every root that has 'y' as a second consonant will change these forms in a different way.

I wrote the rule file to cover two of these instances: roots with 'n' as the first consonant and roots with 'A' as the first consonant. (There is also an unfinished coverage of roots with 'y'/'w' as the second consonants. I will not describe it here).

These are the cases it covers:

When encountering a root with 'n' as a first consonant, the 'n' shall disappear when it has no vowel following. For example: The root n-p-l in 1st person sg., future tense should be "enpol", but in Hebrew it's pronounced "epol".

When encountering a root with 'A' as a first consonant, the 'A' shall disappear when it has a vowel preceding it, and the preceding vowel will change to 'o'. For example: The root A-m-r in 1st person sg., future tense should be "eAmar", but in Hebrew it's pronounced "omar".

Lastly, when a series of Consonant-Vowel-Consonant-Vowel occurs (except in Present tense-Feminine) the mid vowel will be dropped. For example: The root n-p-l in 2st person pl., future tense would go (according to the rule described above) under the process "ti**n**pelu" 🡪 "tipelu". However, because of this last rule the 'e' drops and it becomes "tiplu". That is also covered, after all other changes.

I hope I managed to make this file clear enough. Please ask if something is not clear.