LING 472 Spring 2020
Assignment 2

Adapted from Mans Hulden's assignment

A grammar of Kaixo verbs

Kaixo is a fictitious language that contains each and every phenomenon and grammatical category known to present in some language of the world. In this exercise, you will write a limited core verb grammar for *Kaixo*. Because the language is so rich, you will have an opportunity to tackle many different morphological and phonological phenomena. This experience will come in handy when modeling other languages.

How to read this document

The document below is a linguistic description of the morphophonological phenomena in Kaixo. Since your task is to build finite state automata for this language, you could start with drawing some preliminary FSA on paper, as you read. For example, the fact that the document says the negation marker always comes first is your clue as to which state is the start state. After you drew the FSA on paper, you can start encoding it in foma formalism in kaxio.lexc. You need to think additionally about which affixes in the language are optional. Can the person marker be optional, would it make sense? What about the aspect markers, do you always need them?

The writing system

The writing system of the language is phonemic and entirely IPA-based. You can therefore make assumptions such as **k,g** being velar stops, etc.

Basic verb structure

In the assignment, we will focus on modeling the basic verb structure of *Kaixo*. This will involve writing a lexc-file `kaixo.lexc` which models the morphotactics, and a *foma* file `kaixo.foma` which handles morphophonology and additional phenomena such as reduplication.

The basic verb structure of the language is quite simple. Verbs inflect for person, number, and tense by prefixes or suffixes, with a scheme that looks like this:

- The infinitive forms of verbs (which we can consider stems here) are always maximally two syllables long, e.g. **kata** (to lie).
- The first person marker is the suffix **-ti**.
- The second person marker is the suffix **-ke**.
- The third person marker is the prefix **s-**.
- Tense is marked by either no suffix (present), or by **-n**, immediately following the stem (past). There are only two tenses.
- Number is marked by a suffix that follows the person and tense suffixes, **-s** for plural, and $\emptyset$ for singular (i.e., it's unmarked).

This is illustrated by the basic table below. Note that you'll see a phonological rule in effect here where nasals assimilate in place with a following velar (such as **k**), surfacing as **ŋ**.

		verb: kata			
	Pres			Past	
	Sg	Pl		Sg	Pl
1P	katati	katatis	1P	katanti	katantis
2P	katake	katakes	2P	kataŋke	kataŋkes
3P	skata	skatas	3P	skatan	skatans

Some phonology

Before we tackle the more complex verb forms involving aspect and evidentials, a word about phonology. The language only has five vowels: **a,e,i,o,u**. However, the front vowels **e,i** never co-occur with a back vowel **o,u** in the same word. This has a bearing on suffixes such as **-ke** and **ti**, which will surface as **ko** and **tu** if the stem contains a back vowel, such as **lupo** 'to drink'. In your implementation, it's probably best to capture this by *underspecifying* these suffixes as **-kE** and **-tI**, and then using a morphophonological assimilation rule causing them to surface as **-ke/ko** or **ti/tu**, depending on the vowels in the stem. The vowel **a** is neutral; combining with a stem with only neutral vowels, such as **kata**, the default realization is front, i.e. **-ti/-ke**.

Additionally, two adjacent fricatives are never tolerated in the language and these are broken up by the vowel **a**. For example: **s+sape+s**, where the first **s** is the 3P marker, and the **sape** is the stem, and the second **s** is the plural marker, surfaces as: **sasapes**.

As mentioned above, there is a process of nasal place assimilation before velars: all nasals become **ŋ** before any velar consonant.

Lexicon

You will work with only four verb stems:

Stem	**Gloss**
kata	to lie
lupo	to drink
pele	to play soccer
sape	to run

Evidentials, Aspect, and Negation

The previous covers the basic verb grammar and it is recommended that you implement this first to perfection before moving on to this section. This implies that you would generate 4 × 12 = 48 forms correctly.

Durative

The language has a durative prefix marker **maa-** which comes right before the stem and the third person marker, if present. Thus, **skata** translates as 'he/she lies', while **maaskata** means 'he she lies over an extended period of time'. This prefix, having only the vowel **a** has no allomorphs pertaining to vowel harmony.

Advanced, optional: Hearsay

The language has a somewhat pejorative evidential circumfix marker for *hearsay*: **bla- ... -bla**. For example, **skatans** means ‘they lied’, while **blaskatansbla** means ‘they supposedly lied (a conjecture based on hearsay)’.

Advanced, optional: Iterative

The iterative aspect is marked by reduplication of the first onset and nucleus *after* all the other affixes mentioned previously are in place. For example, **skatans** means ‘they lied’ while **skaskatans** means ‘they repeatedly lied’. This interacts with the other affixes. For example, to apply the iterative in addition to hearsay, the hearsay prefix is the one that gets reduplicated **blablaskatansbla**: ‘they supposedly repeatedly lied’.

Negation (not advanced and not optional!)

The last affix is the negation-prefix: **na-**. This always attaches as the *very first element*. That means that even reduplication occurs before this affix is attached. For example: **naskatans** means ‘they didn’t lie’, while **nablablaskatansbla** means ‘they did not supposedely repeatedly lie.’

A note on flag diacritics

If you use flags to implement hearsay and iterative or to make sure you can only have one person marker in a word, be sure to eliminate the flags and calculate an equivalent flagless transducer immediately after compiling the lexicon. That is, when `kaixo.foma` file compiles the lexicon, be sure to include this:

```
read lexc kaixo.lexc
eliminate flags
def Lex;
```

Remember that if you use flag diacritics, you must declare each flag as a multicharacter symbol in the lexc-file. (Hint: you only need unification flags for this grammar).

Tags

Use the tags provided in the lexc-file `kaixo.lexc`. These are:

```
[NEG] [1P] [2P] [3P] [SG] [PL] [DURATIVE] [PAST] [PRES]
```

(Add [ITERATIVE] and [HEARSAY] if doing the bonus option.)

Getting started

A verb lexicon is included in `kaixo.lexc`. Make sure to include the four stems in your grammar, and that you can generate all the correct forms (test little by little). Work incrementally: include only **kata** first, and only the basic tenses and person/number inflection before moving to the other verb stems, then include evidentials and aspect markers with only **kata**, then include all stems with all affixes.