LDK 2023

# Beyond Concatenative Morphology: Applying OntoLex-Morph to Maltese

**Maxim Ionov**
University of Cologne, Germany
mionov@uni-koeln.de

**Michael Rosner**
University of Malta
mike.rosner@um.edu.mt

## Abstract

OntoLex-Morph is an extension of OntoLex-lemon, (a *de facto* standard vocabulary for publishing lexical data) that is designed to accommodate the description of morphological phenomena into lexical datasets. It is intended to be universally applicable, but so far its application has been focused on the more familiar European languages. This article attempts to show that the morphology extension to OntoLex-lemon can also be applied to Maltese, and by extension, to other Semitic languages. We present our modelling, show how generation rules can be used, and offer some recommendations for changes to the module which would considerably improve the transparency of descriptions that make use of it. Finally, we conclude that if such recommendations are accepted, future discussion should attempt to better delimit the scope of the module to avoid incorporation of information that rightly belongs elsewhere.

## 1 Introduction

OntoLex is a formal model for representing lexical resources, such as dictionaries and thesauri, in a machine-readable format.[1] It was developed to provide a standardised framework for representing lexical entities and relationships between them, with the aim of improving interoperability and reusability of lexical data across different applications and domains.

OntoLex is an RDF model built on top of existing semantic web standards. This allows for the interoperability and integration of lexical resources with other semantic web resources, and for the querying and analysis of lexical data using RDF-based tools and applications.

The model was designed to be modular and extensible, with different modules representing different aspects of lexical information, such as lexical senses, syntactic frames, and semantic relations. This allows for the representation of complex lexical information in a structured and flexible way, and for the customisation of the model to suit different linguistic and domain-specific needs.

One of the modules that is currently being developed is *OntoLex-Morph*, a module that allows representing rich morphological information that is often provided in lexicographic resources. In addition to representing static data such as morphemes and their grammatical information, the module provides the means to model information on how to generate wordforms given lexical entries and finite state-like rules. Despite being developed with a goal to support a wide variety of languages and language phenomena, to the best of our knowledge, it has not yet been applied to languages with nonconcatenative morphology.[2] Semitic languages, having a system of consonantal roots with a complex system of inflection and derivation, belong to this category. In this paper we show how OntoLex-Morph can be applied to model lexical data from one such language, Maltese. Although various computational approaches to Maltese lexical and morphological data have been proposed (e.g. Borg and Gatt (2017); Ravishankar et al. (2017); Sagot and Walther (2013)), this is a first time a linked-data approach has been investigated. We present a small subset of a Maltese dictionary together with a discussion of issues encountered along the way. Additionally, we provide a reference implementation for form generation, bringing the model one step closer to completion.

The rest of the paper is structured as follows: Section 2 provides an overview of the Maltese language and describes the phenomena we chose for this paper. Section 3 gives an overview of OntoLex and OntoLex-Morph vocabularies. In Section 4 we talk about modelling decisions for both static

[1] https://www.w3.org/2016/05/ontolex/.

[2] At least to languages where it is the primary way of inflection and derivation.

data and generation rules and present our reference implementation for form generation. Finally, we discuss what we found along the way, whether the model as it is right now is suitable for such data (spoiler: we think so), and suggest some additions that could help the model transparency.

## 2 The Maltese Language

Maltese is a mixed language made up of Semitic and romance substrates, which respectively share many important characteristics of other languages in those classes. In this article we focus on the Semitic substrate which manifests itself both lexically and morpho-syntactically with respect to different syntactic categories. Thus, the Maltese words *kelb* (Eng. "dog") and *kiteb* (Eng. "write") not only resemble their counterparts in e.g. Arabic and Hebrew from a lexical perspective, but are susceptible to morphological processes for generating nominal and verbal paradigms similar to those operating in such languages. These processes are a superset of the affixation phenomena that characterise most European languages, primarily because word formation in Semitic languages is based on roots and templates. The formation of a word is effected in part by *interdigitation* whereby a pair of vowels called a *vocalism* is inserted into a sequence of consonants. To give a simple example, the word *kiteb* is formed by interdigitating `i-e` with `k-t-b`.

The result of such interdigitation may be a word in its own right or may, as in the case of verbs, be subjected to further processes to yield a complete conjugation paradigm. These vary greatly in complexity, from simple affixation to subtle vowel changes depending on considerations of syllabic structure and vowel harmony. Maltese has no infinitive form, so for citing lexical entries for verbs, the de facto convention in is to use the third person singular masculine (3SG.M) perfective form since many other verbal forms can be derived from it relatively easily. We refer the reader to Rosner and Borg (2022) for further details on the Maltese language.

In this article we focus on the extent to which it is possible to generate complete paradigms using the morphological rules proposed by OntoLex-Morph. We are primarily concerned with the generative capacity of such rules. Subsequently we will turn to some considerations of their descriptive efficiency. We start with the easiest case of well-behaved Maltese strong verbs (such as *kiteb*).

| person/gender | perfective | imperfective |
|---|---|---|
| 1SG | ***k*t*i*b*t*** | ***ni*k*t*e*b*** |
| 2SG | ***k*t*i*b*t*** | ***ti*k*t*e*b*** |
| 3SG.M | ***k*i*t*e*b*** | ***ji*k*t*e*b*** |
| 3SG.F | ***k*i*t*b*e*t*** | ***ti*k*t*e*b*** |
| 1PL | ***k*t*i*b*na*** | ***ni*k*t*b*u*** |
| 2PL | ***k*t*i*b*tu*** | ***ti*k*t*b*u*** |
| 3PL | ***k*i*t*b*u*** | ***ji*k*t*b*u*** |

Figure 1: Conjugation of *kiteb*

### 2.1 Maltese Strong Verbs

Many verbs within the Semitic substrate of Maltese are triliteral i.e. built from a skeleton of three consonants. There are two aspects: perfective and imperfective.

In *perfective* aspect, suffixes *-t, -et, -na, -tu, -u* mark person, first vowel is deleted with consonant-initial suffixes; second vowel *e* before consonant initial suffix (i.e. when stressed), becomes *i*. In *imperfective* aspect, prefixes mark person, suffix *-u* marks plural. A completely regular example is *kiteb* (Eng. 'he wrote') which conjugates as shown in Figure 1.

There are many ways these conjugations can vary when one of the root consonants (radicals) falls into a certain category. Thus, when the *first* radical is *silent għ* or *h*, the first vowel is retained when there is a consonant-initial suffix. So for the verb *għamel* (Eng. 'he made') we have *għamilt* instead of **għmilt* as shown in Figure 6 in the Appendix.

On the other hand, when the *second* radical is a *liquid* consonant, i.e. an *l, m, n, r*, an issue arises in terms of pronunciation of plural imperfective forms, so a helping (euphonic/epenthetic) vowel is required and placed between the first and second root consonants. So for the verb *telaq* (Eng. 'he left') we have e.g. *nitilqu* instead of **nitlqu* as shown in Figure 7 in the Appendix.

These examples are by no means exhaustive, but they clearly illustrate the need to discriminate behaviour on the basis of consonant classes.

### 2.2 Maltese Alphabet

The Maltese alphabet is based on the Latin one and comprises 6 vowels — *a e i o u ie* — and 24 consonants — *b ċ d f ġ g għ h ħ j k l m n p q r s t v w x ż z*. It poses two challenges when formulating

| Class | Characters |
|---|---|
| silent | *għ, h* |
| liquid | *l, m, n, r* |
| normal | *b ċ d f ġ g ħ j k p q s t v w x ż z* |

Figure 2: Character classes

replacement rules regardless of a formalism. First, it contains digraphs, *ie* and *għ*. So if there are rules that operate with concepts like a "letter", a vowel, or a consonant it cannot be just assumed that one letter is one character. When working with regular expressions, for example, it would be incorrect to simply use `.` or `\w` to represent any letter of the alphabet. Furthermore, if we aim to minimise the number of rules and create them as universal as possible, we need means to refer to certain character classes. Based on the examples above, in order to discriminate amongst the classes of verb to which the above cases belong, we need to distinguish at least between silent, liquid and normal consonants as listed in Figure 2.

## 3 OntoLex and OntoLex-morph

OntoLex-lemon (McCrae et al., 2017) is the *de facto* standard for publishing lexical resources in RDF, compliant with established web standards. The central class in the core model, depicted in Figure 3 is `LexicalEntry` — a lexeme or a dictionary entry. It must have at least one (word)form (`canonicalForm`) and can have a number of other forms, a number of senses, which can then be then linked to either lexical concepts or entities in an ontology. Basic morphological information like a part of speech and grammatical categories can be provided for lexical entries and forms using elements of any suitable vocabulary, such as LexInfo.[3]

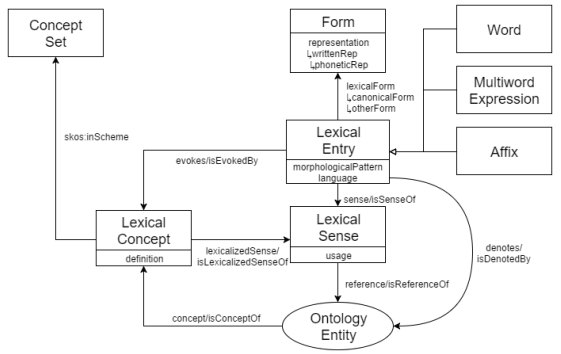


Figure 3: OntoLex-Lemon core model

[3] https://lexinfo.net/.

One thing to note is that a single lexical entry cannot have more than one part of speech, which is an important factor for our design decisions described below.

Although there is a place for including basic morphological information in the core model, it does not allow the representation of paradigmatic relationships between lexical entries and forms (inflectional morphology) or derivational relationships between lexical entries. In order to close this gap, an extension to the core module, OntoLex-Morph is being developed.[4] The model, depicted in Figure 4 consists of three main parts: derivation (left), inflection (right), and rules for generating new forms, both for inflection and derivation (top). The central part of the module is the class `Morph`, which corresponds to a morph — a specific realisation of a morpheme. It is a subclass of `LexicalEntry`, which might be a bit counterintuitive at first, but this allows for resources where morphs are dictionary entries of their own.

Another part of OntoLex-Morph important for us is a representation of rules that can be used to generate forms from lexical entries (or, more specifically, from their forms). The mechanism behind this is the following:[5] (i) A lexical entry can be a part of an inflectional paradigm. (ii) For each paradigm, there can be a number of rules, each of them having information on how to produce a form and grammatical meaning that should be assigned to this form; (iii) The formalism to encode a rule is a (POSIX-compatible) regular expression.

For example, a rule for forming a standard English plural form can look as following:

```
<rule_plural>
   a morph:InflectionRule ;
   morph:replacement [
        a morph:Replacement ;
        morph:source "$" ;
        morph:target "s"@en ; ] ;
   morph:involves [
        a ontolex:Affix ;
        rdfs:label "-s"@en ;
        morph:grammaticalMeaning [
             a morph:MorphologicalMeaning ;
             lexinfo:number lexinfo:plural ] ].
```

It is, of course, possible to use instances of the `morph:Morph` class (and its subclasses) instead of blank nodes, and in most situations this will be the case. However, this will depend on the

[4] https://www.w3.org/community/ontolex/wiki/Morphology.

[5] Here, we focus on the rules for generating inflected forms. For the more complete description of the model refer to Chiarcos et al. (2022).

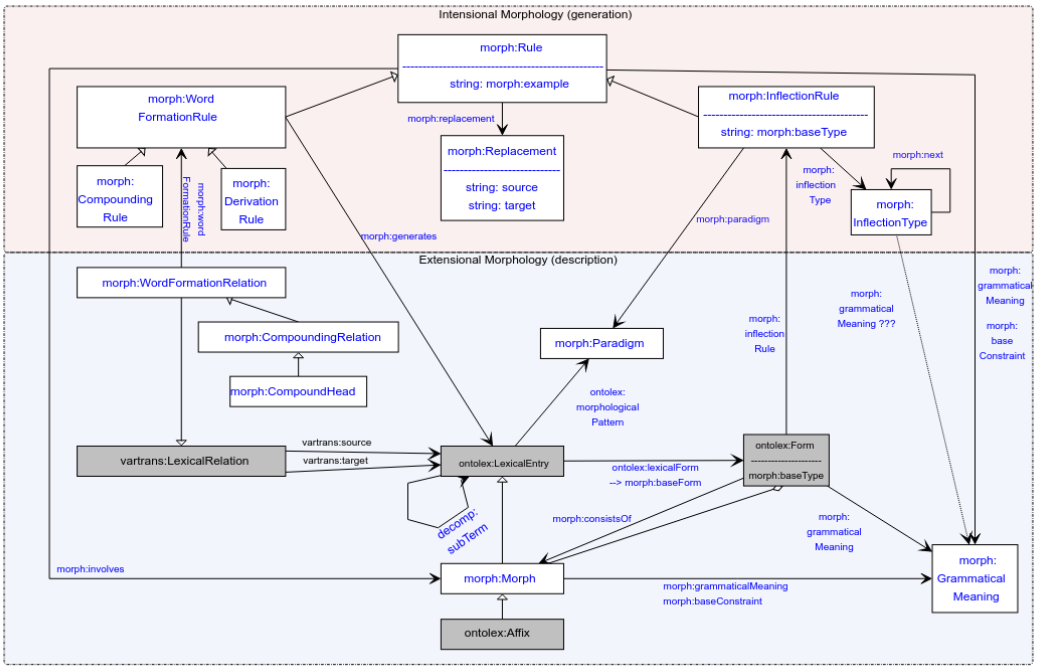


Figure 4: OntoLex-Morph draft model

dataset, and in principle it is possible to connect `GrammaticalMeaning` directly to a rule.

The process of generating forms using these rules is decoupled from the rules and the module in general, so it is up to the users of the model to choose how this is done. We describe our implementation below, in Section 4.2.

## 4 Modelling data with OntoLex-Morph

So far, OntoLex-Morph have been primarily applied to fusional languages with concatenative morphology, such as Greek, Latin and German (Chiarcos et al., 2022). There are proof-of-concept examples of modelling of typologically diverse language data, such as snippets for agglutinative Finnish and Turkish or polysynthetic Inuktitut.[6] But there was very limited exploration of the model applicability to nonconcatenative morphology, and Semitic languages have not been modelled so far. This paper and, more specifically, this section is set to close this gap by showing an application of the model to represent a part of Maltese morphology.

Gold-standard data concerning the conjugation of Maltese verbs appears in a number of grammar books (e.g. Henry (1980)). There are also some online resources such as Cooljugator[7] and Ġabra (Camilleri, 2013), which can be accessed via the Maltese Language Resource Server (MLRS).[8] Ġabra is a free, open lexicon for Maltese, built by collecting various different lexical resources into one common database containing 19,918 entries and 4,514,682 inflectional word forms, many linked by root, translations in English, and marked for various morphological features. We have made use of the latter in this study. For this paper, we only model the verbs described in Section 2.1.

### 4.1 Modelling

In English and many other languages, verbal forms are structured around the infinitive form. Typically, the infinitive is taken as a basis from which all the other forms can be generated, largely by affixation. Maltese has no infinitive, so the third person singular masculine (3SG.M) perfective form is used instead. At the same time, certain semitic lexemes that are semantically related can often be grouped according to their underlying root consonants, often transcending syntactic categories. Thus *kiteb (Eng. 'wrote'), ktieb (Eng. 'book'), kittieb (Eng. 'writer')* all share the root `k-t-b`. Roots have an important role to play in linking semantically re-

[6] See `https://github.com/ontolex/morph/tree/master/data` for some example data.

[7] https://cooljugator.com

[8] https://mlrs.research.um.edu.mt/

lated words, and should therefore be explicitly reflected in the modelling.

This creates a choice: to model a root as a lexical entry and all the forms derived from it as forms, or to represent each lexeme as a lexical entry, with verbs having their 3SG.M form as their canonical form, additionally connecting each lexical entry to its root. There are good reasons to prefer the latter. First of all, the principle of separating lexemes into different lexical entries while preserving root information is shared by printed dictionaries of Maltese, e.g. Aquilina (1987) and other Semitic languages. The resource we are modeling, Ġabra, also shares this design. Second, lexical entries in OntoLex cannot have more than one part of speech, which makes using roots as lexical entries problematic, if not impossible. Additionally, this fits into the model's dichotomy of inflection vs. derivation, where semantically related entries (e.g. *'to write'* vs. *'writer'*) could be distinct lexical entries connected by a derivational relationship instead of two forms, members of the same inflectional paradigm.

We therefore represent root consonants as a `lexinfo:RootMorph`, a subclass of `morph:Morph`, and each form that stems from that root cluster is connected to that morph with the property `morph:consistsOf`.

This way, for each verb we are modelling, there is a single lexical entry and a canonical form that corresponds to a 3SG.M perfective form. That form is connected to the corresponding root morph. Also, this form is connected to the lexical entry as a `morph:baseForm`, which means that its written representation will be used as a base for form generation. Furthermore, the lexical entry links to a corresponding `morph:Paradigm` to specify an inflectional paradigm for that word:

```
roots:k-t-b    a lexinfo:RootMorph ;
               rdfs:label "k-t-b" .

:1    a ontolex:Word ;
      lexinfo:partOfSpeech lexinfo:verb ;
      morph:morphologicalPattern
                    <kiteb_paradigm> ;
      ontolex:canonicalForm <1_form> ;
      morph:baseForm <1_form> .

<1_form>   a ontolex:Form ;
           morph:consistsOf roots:k-t-b ;
           ontolex:writtenRep "kiteb"@mlt.
```

Instead of explicitly providing the forms, we provide rules for how the forms should be generated for each of the verbs as described in Section 2.1. As described above, the core of each rule is a mapping as specified by a pair of regular expressions: a source and a replacement. Unlike the example for English plural above, we need to match the whole form and replace it with a new one. Since we know the number of characters in the base form, we can simply match each of them to a capturing group. To illustrate this with respect to the perfective 3SG.M → 1SG mapping of *kiteb* we can use the following:

```
source:      (.)(.)(.)(.)(.)
replacement: \1\3i\5t
```

The input specifies a sequence of 5 segments. The dot is an unrestricted wildcard matching any character. Thus the input matches any sequence of 5 characters, which become bound, in order, to numerical variables 1–5. Thus after matching *kiteb*, 1=k, 3=t, 5=b, and the output, \1\3i\5t = *ktibt*

The problem with this approach comes from the fact that it assumes that each letter corresponds to one character, which is not true for Maltese alphabet. Instead, we need to provide a list of possible options for each of the positions:

```
(b|ċ|d|f|ġ|g|għ|h|ħ|j|k|l|m|n|p|q|r|s|t
|v|w|x|ż|z)(a|ie|e|i|o|u)
(b|ċ|d|f|ġ|g|għ|h|ħ|j|k|l|m|n|p|q|r|s|t
|v|w|x|ż|z)(a|ie|e|i|o|u)
(b|ċ|d|f|ġ|g|għ|h|ħ|j|k|l|m|n|p|q|r|s|t
|v|w|x|ż|z)
```

This can be slightly simplified by tailoring each group to symbols that can appear in a given paradigm, but even in this case, rules produced this way are clearly unwieldy. A simple yet elegant approach would be to use character classes like:

```
source:   (C)(V)(C)(V)(C)
```

where C and V respectively stand for the sets of consonants and vowels. Using this logic, it is possible to use more specific character classes, e.g. liquid consonants, to reduce the number of paradigms by creating more universal rules. However, this would again make the rules more complex and less readable. In our dataset we tried to keep the balance, creating three paradigms (and three sets of rules) for each of the cases described in Section 2.1.

### 4.2 Character classes and generation

An important question with regards to character classes is where and how to model them. We see three distinct possibilities: (i) externally, using a preprocessor to generate rules without character classes or generate forms directly; (ii) with a dataset-specific property; (iii) with a property specified in OntoLex-Morph. While the first two options

are less invasive and prevent the module from growing in complexity, it is worth noting that only the last option allows interoperability and reusability, not only for rules themselves, but also for any software that will use these rules. In our modelling, we propose a class `CharacterClass` that can be used in the following way:

```
gabra:V a gabra:CharacterClass ;
     rdfs:label "V" ;
     rdfs:member "a", "e", "i", "o", "u" .
```

## 5 Conclusion

We have verified the hypothesis that Morph can be applied to some key non-concatenative morphological phenomena in Maltese. The implication is that this generalises to other Semitic languages. We have also illustrated the need to provide facilities for incorporating definitions of character classes. The dataset, our implementation of form generation, and additional information can be found on GitHub.[9]

The main discussion point to emerge is whether such definitions should be external or internal to Ontolex-Morph. The pros of keeping character classes external is that the module remains lightweight. However there is a price to be paid. At some point, externally defined character classes will have to be replaced in each rule with lists of characters that will become exceedingly verbose and illegible. Conversely, character class definitions could become an integral part of the module. We favour the latter approach on the grounds that the benefit of legibility for producers and consumers of morphological information far outweighs the cost of slightly increased complexity in the formalism.

Of course there are limits to this line of argumentation. It would be theoretically possible to absorb morphological processing of arbitrary complexity (e.g. to include the article used with nouns, clitic pronouns, etc. all of which end up as one word on the page). However, the inclusion of this level of expressivity would contradict the intention to keep the module reasonably simple and transparent. The module aims to represent elements involved in both the decomposition and formation of lexical entries/word forms (Klimek et al., 2019, p. 579), but fine-grained description of phonological processes involved in stem or word formation on the phoneme level is excluded.

The line between justifed and unjustified refinements to OntoLex-Morph is delicate, but somewhere in between the two is an as yet unidentified cutoff point whose placement would be an apt task for imminent future discussion.

[9] https://github.com/max-ionov/maltese-morph.

## Acknowledgements

The research described in this paper was conducted in the context of the COST Action NexusLinguarum – European network for Web-centered linguistic data science (CA18209).

## References

J. Aquilina. 1987. *Maltese-English Dictionary*. Midsea Books, Valletta.

C. Borg and A. Gatt. 2017. Morphological analysis for the Maltese language. In *Proc. 3rd Arabic NLP Workshop*, pages 25–34, Valencia, Spain. Ass. Comp. Ling.

J. Camilleri. 2013. *A Computational Grammar and Lexicon for Maltese*. MSc Thesis, Chalmers University of Technology, Gothenburg, Sweden.

C. Chiarcos, K. Gkirtzou, F. Khan, P. Labropoulou, M. Passarotti, and M. Pellegrini. 2022. Computational morphology with OntoLex-Morph. In *Proc. 8th Workshop on Linked Data in Linguistics*, pages 78–86.

Brother FSC Henry. 1980. *Grammatika Maltija*. De La Salle Brother Publications, Malta.

B. Klimek, J. McCrae, J. Bosque-Gil, M. Ionov, J. Tauber, and C. Chiarcos. 2019. Challenges for the representation of morphology in ontology lexicons. *Proceedings of eLex 2019*, pages 570–591.

John P McCrae, J. Bosque-Gil, J. Gracia, P. Buitelaar, and P. Cimiano. 2017. The OntoLex-Lemon Model: Development and Applications. In *Proceedings of eLex-2017*, pages 19–21.

V. Ravishankar, F. Tyers, and A. Gatt. 2017. A morphological analyser for Maltese. In K. Shaalan and S. El-Beltagy, editors, *Procedia Computer Science vol 117*, pages 175–182. Elsevier.

M. Rosner and C. Borg. 2022. Report on the Maltese Language. *Deliverable D1.25, European Language Equality Project*.

B. Sagot and G. Walther. 2013. Implementing a formal model of inflectional morphology. In *3rd International W/S on Systems and Frameworks for Computational Morphology*, volume 380 of *Communications in Computer and Information Science*, pages 115–134, Berlin. Humboldt-Universität, Springer.

# A Appendix

| person/gender | perfective | imperfective |
|---|---|---|
| 1SG | ***k*t*ib*t** | *ni**kteb*** |
| 2SG | ***k*t*ib*t** | *ti**kteb*** |
| 3SG.M | ***kiteb*** | *ji**kteb*** |
| 3SG.F | ***kitb*et** | *ti**kteb*** |
| 1PL | ***k*t*ib*na** | *ni**ktb*u** |
| 2PL | ***k*t*ib*tu** | *ti**ktb*u** |
| 3PL | ***kitb*u** | *ji**ktb*u** |

Figure 5: Conjugation of *kiteb*

| person/gender | perfective | imperfective |
|---|---|---|
| 1SG | ***għamil*t** | *n**għamel*** |
| 2SG | ***għamil*t** | *t**għamel*** |
| 3SG.M | ***għamel*** | *j**għamel*** |
| 3SG.F | ***għaml*et** | *t**għamel*** |
| 1PL | ***għamil*na** | *n**għaml*u** |
| 2PL | ***għamil*tu** | *t**għaml*u** |
| 3PL | ***għaml*u** | *j**għaml*u** |

Figure 6: Conjugation of *għamel*

| person/gender | perfective | imperfective |
|---|---|---|
| 1SG | ***tlaq*t** | *ni**tlaq*** |
| 2SG | ***tlaq*t** | *ti**tlaq*** |
| 3SG.M | ***telaq*** | *ji**tlaq*** |
| 3SG.F | ***telq*et** | *ti**tlaq*** |
| 1PL | ***tlaq*na** | *ni**tilq*u** |
| 2PL | ***tlaq*tu** | *ti**tilq*u** |
| 3PL | ***telq*u** | *ji**tilq*u** |

Figure 7: Conjugation of *telaq*