A Formal Grammar for Toki Pona

Zach Tomaszewski

ICS661

11 Dec 2012

1) Introduction

Toki pona is a simple constructed language. Although it is an artificial language with a very limited and closed vocabulary, toki pona still exhibits many of the features of a natural human language. In this project, I developed a machine-readable formal grammar for toki pona and then used a CKY parser to recognize valid and invalid toki pona sentences.

Toki Pona. Toki pona is a constructed language--or "conlang"--invented by Sonja Elen Kisa. It is inspired by Taoism and the Sapir-Whorf hypothesis. Specifically, Kisa proposes that toki pona encourages its speakers to think simply and to focus on basic reality rather than abstract or euphemistic concepts [1].

Toki pona has been fairly successful for a conlang, gaining interested speakers outside of the normal conlang community. Kisa has largely abandoned the project. This has left the main tokipona.org website in a state of disrepair. However, a scattered community continues to play with the language elsewhere. This community presence is mostly scattered over various blogs and personal sites, community groups and forums, wikis, and a few YouTube videos. The best learning resource is a tutorial [2] by *jan Pije* (Bryant Knight), an early fluent toki pona speaker. Although a fair amount of language-tinkering has been proposed, most of the community adheres to the original words and rules laid out by Kisa.

Toki pona has a 14-letter alphabet. Letters are always lowercase except for the first letter of a proper name. Toki pona contains about 120 words, depending on how you count them. A small number of words were dropped during the development of the language. One word has two accepted spellings (*ale* and *ali*). Five words were added near the end of Kisa's involvement, and they have not been widely adopted by the community. One of those five words, *pu*, has no known definition.

Sentences are given in subject-verb-object order. A special marker word, *li*, marks the separation between the subject and verb, though *li* is dropped when the subject is simply *mi* ("I") or *sina* ("you"). Another separator, *e*, marks the transition between verb and object. Toki pona has no tense, gender, or number, though each of these can be explicitly specified with an appropriate adjective or conditional preface to the sentence if necessary. Modifiers, whether adjectives or adverbs, come after the words they modify.

Most words have a broad conceptual range. For example, as an adjective, *suli* can mean "big", "fat", "tall", or "important". Similarly, *pona* means "good", "simple", or "pure" as an adjective or "fix", "improve", or "simplify" as a verb. Not all words are this general, however. *oko* ("eye") is used only as a noun. "Looking" as a verb and "visual" as a modifier is covered by a different word, *lukin*.

Most, but not all, of the words can be used as either noun, verb, or modifier depending on their placement in the sentence. For example, *moku* can mean "food" as a noun, "edible" as a modifier, or "eat" as a verb. Occasionally, it can be difficult to tell which role a word is filling. For example, in the sentence

mi moku.

moku is most likely a verb, which gives this sentence the meaning "I eat". However, if we read *moku* as a predicate adjective or predicate nominative, this sentence could also be parsed as "I am edible" or "I am food". This combination of grammatical vagueness with the wide conceptual range of most words can make toki pona highly ambiguous at times. While the greater context often gives clues to help disambiguate, it can often be harder to read or understand toki pona than it is to write or speak it.

Because of the limited vocabulary, descriptive phrases are very common. Many of these have become fairly standardized. Some examples include:

- *jan pona* = person + good = friend
- *jan ike* = person + bad/evil = enemy
- *jan utala* = person + fighting = soldier
- *tomo tawa* = room/structure + moving = vehicle
- *ma tomo* = land/area + (of) room/structures = city

I have personally been dabbling with toki pona on and off for a couple years. At this point, I am basically conversant but not fluent.

Formal Grammars. A formal grammar is a precise description of all possible strings (or sentences) of a particular language. Formal grammars can be specified in machine-readable form in order to construct parsers and generators. Parsers recognize whether a string of symbols is a valid instance of the language, and generators produce valid strings that are in the language.

One example parsing algorithm is the Cocke-Younger-Kasami (CYK or CKY) algorithm. The CKY algorithm starts with the tokens of the input sentence. Using an efficient dynamic programming approach, the parser works bottom-up through the rules of the grammar to see if it can reach the highest-level start symbol in the grammar. If this start symbol is reached, then the input sentence is a valid string in the language. The CYK algorithm only works with a particular class of grammars--context-free grammars (CFGs)--and the rules of the grammar used must be in Chomsky Normal Formal (CNF). In CNF, each production rule in the grammar must produce either two non-terminals or a single terminal. Conveniently, any CFG can be converted into an equivalent Chomsky Normal Form.

Project Goal. The goal of this project was to develop a formal context-free grammar that describes all valid toki pona sentences. A CYK parser is then used to recognize whether a given string is a valid toki pona sentence. The parse produced--and there may be more than one possible parse or reading of a valid sentence--also shows the internal grammatical structure of the sentence.

This parser could provide useful feedback for toki pona learners to check their sentence productions. It could also aid reading by explicitly showing the different possible structures of a valid sentence, thus making any ambiguity explicitly clear. It is also an important first step--syntactic parsing--that could be used as a foundation for more advanced semantic processing, such as machine translation.

2) Description

Previous Work. This is not the first project to specify a formal grammar for toki pona. The Wikipedia article for toki pona has gone through at least 2 major iterations trying to concisely describe the language rules. The first attempt [3] was so simple that it lacked even some of the basic rules such as dropping *li* when the subject is only *mi* or *sina*. The current form [4] is longer, though it is not in a precise formal grammar format.

jan Kipo, a significant member of the toki pona community, sketched out a more formal grammar [5]. Matthew Martin then converted this grammar to a machine-readible form for use with the AGFL parser [6].

For the parser used in this project, I had previously developed two relevant programs as part of earlier assignment work. The first program converts any CFG into CNF. The second is a CKY parser that shows all possible parses of a given sentence based on a given CNF grammar. Both programs are written in Python 3. The source code is available online, as described in Appendix C.

Methodology. I first collected a corpus of 100 valid toki pona sentences. For this, I used the toki-pona-to-English problems given in j*an Pije*'s tutorial. I also added a poem from the official toki pona website to bring the number of sentences up to 100. This corpus is provided in Appendix B. I also wrote approximately 20 invalid sentences that mirrored common mistakes made by toki pona novices.

For ease of parsing, each sentence was placed on its own line and all punctuation except commas was removed. A space was added before every comma in order to make it its own token. All proper names--easily recognized by their initial capital letter--where replaced with a single 'Name' token.

I then developed my own toki pona grammar. For the lexicon, I assigned words to noun, verb, modifier, or preposition according Kisa's descriptions. I largely worked independently on the higher levels of the grammar, although I did refer occasionally to the current Wikipedia descriptions.

I ran the corpus of valid sentences through the parser, examined the parses, and tweaked the grammar accordingly. This required a few hours of work. The resulting grammar is given in Appendix A.

Converting this context-free grammar to Chomsky Normal Form produced 3907 rules. This high number is partly due to a small bug in the CNF program that occasionally produces duplicate production rules for the form *ZZ1* → *A B* and *ZZ2* → *A B*. This does not affect the correctness of the resulting grammar or parses; it is simply somewhat inefficient. This was not fixed due to time constraints.

An example parse of the sentence *jan utala li seli ala seli e tomo* ("Did the soldier(s) burn the building?") is:

```
S: [S [ZZ122 [NP_NoMiSina [N jan] [N utala]] [ZZ121 li]]
      [Pred [Verb [ZZ106 [V seli] [ZZ105 ala]] [V seli]] [DO [ZZ36 e] [NP tomo]]]]
```

3) Analysis

Results. I compared my grammar's results to the earlier Martin grammar for the AFGL parser using the same 100-sentence corpus. (Since Martin's grammar does not handle proper names, all 'Name' tokens were converted to *pona* in order to produce valid sentences for those tests.) I achieved the following results:

	My Grammar	**Martin's Grammar**
Failed parses	1	27
Median parses generated per sentence	2	4 (7, if disregarding 0 counts)
Sentences with >= 20 parses	9%	23%
Greatest number of parses for a single sentence	54	382
Average parses per sentence (after dropping max outlier)	6.8	14.5

Table 1: Performance comparison of my grammar to Martin's grammar

My grammar failed on a single sentence: *tawa pona*. This is an unusual interjection phrase meaning "good journey", "bon voyage", or "goodbye". It could easily be handled by the addition of a specific rule for it. The general form of a modified noun by itself is not normally a valid sentence structure, however.

In addition to recognizing more of the valid sentences, my grammar also produced fewer duplicate or alternate parses overall, even given the CNF-converter's inefficiency mentioned above.

Dicussion. Developing and evaluating this grammar revealed a number of interesting lessons.

First of all, humans apparently do not think in formal grammars. Despite recently learning the rules of toki pona grammar--often phrased casually in the form of "do this, except in this case"--it proved rather challenging to convert this knowledge into a CFG format.

Regarding toki pona, a number of words that are classified as nouns and not modifier are still frequently used in a modifier-like way. For example, *ilo* means "tool" and *suno* means "sun" or "light". Although *suno* is not technically not a modifier, the construction *ilo suno* ("light tool", meaning flashlight or lamp) is still a valid construction.

To handle this, I added a rule that any noun can be modified by another noun. This produces many duplicate parses, since for any word that can be either a noun or modifier produces both possible interpretations. A future extension to this project could use probability to favor any noun-modifier constructions over noun-noun constructions.

As the grammar stands now, not all modifying phrases are correctly placed. For example, the imperative sentence *o pana e moku tawa mi* means "(you) give food to me". The phrase

tawa mi ("to me") most accurately modifies *pana* ("give") as an adverb, not *moku* ("food") as an adjective. The grammar currently only produces the second, less-correct parse.

The grammar does not encode full recursion in all cases. For example, it handles only up to two direct objects, as in *o pana e moku e telo,* meaning "give me food and water". It only allows for a single modal--such as *ken* or *wile*--to modify a verb. For example, it will accept *mi wile tawa* ("I want to go") and *mi ken wile* ("I am able to go"), but it will not accept the valid *mi wile ken tawa* ("I want to be able to go"). Sentences with more than a single modal are rare though, and more than two would produce a sentence of questionable validity.

Toki pona breaks certain complex or compound relationships into multiple related simple sentences. For example, "I want you to give me food" would be translated as *mi wile e ni: sina pana e moku tawa mi* ("I want this: you give food to me"). Although tightly bound semantically, these two sentences are handled separately by the current grammar.

Toki pona also has an odd idiom for asking yes-or-no questions. For example, *sina moku ala moku* = "you eat not eat" = "Are you eating?" The correct response would then be either *moku* ("yes") or *moku ala* ("no"). The grammar is currently too permissive with this question form, accepting any `V ala V` form, even if these the two `V`'s are different verb tokens. This could easily be corrected with an extra rule for very possible `V ala V` form, though this would noticeable increase the size of the grammar. Also the answer format of only a single verb (possibly modified by *ala*) would normally be an invalid sentence except in this context. The grammar allows this form regardless of context.

These last few deficiencies highlight the main limitation of this project: It handles only the syntactic rules, without considering either semantics (meaning) or pragmatics (context). I particularly noticed this when parsing "invalid" sentences that were actually accepted as valid. For example, a common beginner mistake is to forget to include the *e* marker between the verb and direct object. Rather than *mi wile e telo* ("I want water"), a beginner may say *mi wile telo*. Most listeners would recognize the beginner's intention and correct their syntax. However, without considering semantics, this *mi wile telo* formation is technically correct, meaning either "I want/desire wetly" or "I am wet desire".

4) Conclusion

The essential goals of this project were achieved by constructing a context-free formal grammar for toki pona. In an empirical evaluation using a CKY parser, this grammar was shown to be an improvement over existing tools.

However, future work remains to be done. The grammar has a few known minor deficiencies. Further evaluation should be done on both valid and invalid sentences to flush out any other limitations or errors that have gone unnoticed. Also, given that so many of toki pona's words can serve as any part of speech, using a top-down, rather than bottom-up, parser would likely be more efficient.

Once these syntactic processing tools are stable and well-tested, they could then be used as the foundation for more interesting translation work at the semantic and pragmatic levels.

Works Cited

1. "Toki Pona." http://en.tokipona.org/. Last accessed: 06 Dec 2012
2. Knight, Bryant. "o kama sona e toki pona!" http://bknight0.myweb.uga.edu/toki/lesson/lesson0.html. Last accessed: 06 Dec 2012
3. "Toki Pona." *Wikipedia.* http://en.wikipedia.org/w/index.php?title=Toki_Pona&oldid=70379072 18 Aug 2006.
4. "Toki Pona." *Wikipedia.* http://en.wikipedia.org/w/index.php?title=Toki_Pona&oldid=518453679 18 Oct 2012.
5. jan Kipo. "Grammar." *nimi pi toki pona.* Blog. http://tpnimi.blogspot.com/2010/09/grammar.html. 05 Sept 2010.
6. Martin, Matthew. "A Machine Parseable Context Free Grammar for Toki Pona." *My Suburban Destiny*. Blog. http://www.suburbandestiny.com/?p=805. 10 Sept 2010.

Appendix A: Formal context-free grammar for toki pona

```
S -> Interjection | VocativeS | Sentence | YNAnswer

VocativeS -> NP o | NP o , Sentence | NP o Pred | o Pred | Conditional o Pred
YNAnswer -> V | V ala

Sentence -> SubjPred | Conditional SubjPred | taso SubjPred
Conditional -> SubjPred la | Context la | NP la

SubjPred -> mi Pred | sina Pred | NP_NoMiSina li Pred | CompoundSubj li Pred
CompoundSubj -> NP en CompoundSubj

NP_NoMiSina -> N_NoMiSina | CompNP
NP -> N | CompNP
CompNP -> NMod | NPpi | NP anu NP
NPpi -> NP pi N Modifier | NP pi Name
NMod -> N Modifier | N N | N N Modifier

Modifier -> Mod | Mod Modifier
Pred -> VP | VP li Pred

VP -> IntransVP | TransVP | VP PrepPh
IntransVP -> Verb | lon NP | tawa NP | Modal lon NP | Modal tawa NP | Modifier |
                  NP TransVP -> Verb DO | Modal Verb DO
DO -> e NP | e NP DO
Verb -> V | Modal V | V Mod | YnV | Modal YnV | YnV Mod
YnV -> V ala V

PrepPh -> Prep NP | Prep NP PrepPh

Context -> ante | ken

Modal -> PosModal | PosModal ala | YnModal
PosModal -> kama | ken | wile
YnModal -> kama ala kama | ken ala ken | wile ala wile

V -> anpa | ante | awen | ijo | ike | jaki | jan | jo | kalama | kama | ken
| kepeken | kule | lape | lawa | lete | lili | lon | lukin | moku | moli | musi
| mute | nasa |olin | open | pakala | pali | pana | pilin | pimeja | pini | poka
| pona | seli | sin | sitelen | sona | suli | suwi | tawa | telo | toki | tomo | tu
| unpa | utala | wan | wawa | weka | wile

N -> mi | sina | N_NoMiSina
N_NoMiSina -> Name | akesi | ala | ale | ali | ante | ijo | ike | ilo | insa
| jaki | jan | jo | kala | kalama | kama | kasi | ken | kili | kiwen | kule
| kute | kulupu | lawa | len | lete | lili | linja | lipu | luka | lupa | ma
| mama | mani | meli | mije | moku | moli | monsi | mun | musi | mute | nanpa
| nasin | nena | ni | nimi | noka | oko | olin | ona | pakala | pali | palisa
| pana | pilin | pimeja | pini | pipi | poki | poka | pona | seli | selo | seme
| sewi | sijelo | sike | sinpin | sitelen | sona | soweli | suli | suno | supa
| suwi | tan | tawa | telo | tenpo | toki | tomo | tu | unpa | uta | utala | walo
| wan | waso | wawa | weka | wile

Mod -> ala | ale | ali | ante | awen | ijo | ike | insa | jaki | jan | jelo
| kalama | kama | kin | kiwen | kule | kute | kulupu | laso | lape | lawa | lete
| lili | loje | lukin | mama | meli | mi | mije | moku | moli | monsi | mun | musi
```

```
| mute | nasa | ni | olin | ona | pakala | pali | pimeja | pini | poka | pona
| sama | seli | seme | sewi | sike | sin | sina | suli | suwi | taso | tawa | telo
| toki | tomo | tu | unpa | uta | walo | wan | wawa | weka | wike

Prep -> kepeken | lon | poka | sama | tan | tawa

Interjection -> a | a a | a a a | ala | ike | jaki | mu | o | pakala | pona | toki

Name -> jan 'Name' | ma 'Name' | ma tomo 'Name' | toki 'Name' | soweli 'Name'

# New words not included here: alasa, esun, pan, kipisi, pu
```

Appendix B: Corpus of 100 toki pona sentences

```
# Sentences taken from http://bknight0.myweb.uga.edu/toki/lesson/lesson0.html

#lesson 3
suno li suli
mi suli
jan li moku

#lesson 4
mi lukin e ni
mi wile unpa e ona
jan li wile jo e ma
mi jan li suli

#lesson 5
mi lukin sewi e tomo suli
seli suno li seli e tomo mi
jan lili li wile e telo kili
ona mute li nasa e jan suli

#lesson 6
sina wile kama tawa tomo toki
jan li toki kepeken toki pona lon tomo toki
mi tawa tomo toki
ona li pona tawa mi
sina kama jo e jan pona lon ni

#lesson 7
poka mi li pakala
mi kepeken e poki e ilo moku
jan li lon insa tomo

#lesson 8
sina wile ala wile pali
wile ala
jan utala li seli ala seli e tomo
jan lili li ken ala moku e telo nasa
sina kepeken ala kepeken e ni

#lesson 9
mu
mi wile kama sona e toki 'Name'
jan 'Name' o pana e moku tawa mi
o tawa musi poka mi
jan 'Name' o lawa e mi mute tawa ma pona
tawa pona

#lesson 10
jan 'Name' o , mi olin e sina
ni li jan seme
sina lon seme
mi lon tan seme
jan seme li meli sina
sina tawa ma tomo tan seme
sina wile tawa ma seme

#lesson 11
kili pi jan 'Name' li ike
```

len pi jan 'Name' li jaki
mi sona ala e nimi pi ona mute
mi wile ala toki pi kalama musi
mi wile toki meli
sina pakala e ilo kepeken nasin seme
jan 'Name' li jan lawa pona pi ma 'Name'
wile pi jan ike li pakala e ijo

#lesson 12
mi olin kin e sina
mi pilin e ni
ona li jo ala e mani
mi wile lukin e ma ante
mi wile ala e ijo
mi lukin taso
sina wile toki tawa mije anu meli

#lesson 13
suno li jelo
telo suli li laso
mi wile moku e kili loje
ona li kule e tomo tawa

#lesson 14
mama ona li kepeken e kasi nasa
akesi li pana e telo moli
pipi li moku e kasi
soweli mi li kama moli
jan 'Name' o , mi wile ala moli
mi lon ma kasi

#lesson 15
a
telo sijelo loje li kama tan nena kute mi
selo mi li wile e ni
mi pilin e ona
o pilin e nena
o moli e pipi kepeken palisa
luka mi li jaki
mi wile telo e ona
o pana e sike tawa mi
mi pilin e seli sijelo sina

#lesson 16
mi weka e ijo tu ni
o tu
mi lukin e soweli luka
mi weka

#lesson 17
ken la jan lili li wile moku e telo
tenpo ali la o kama sona
sina sona e toki ni la sina sona e toki pona

#lesson 18
sina sona e toki pona
toki pona li pona ala pona tawa sina
mi wile e ni
ona li pona tawa sina

sina wile pali e seme kepeken sona sina
lipu mi li jo e toki awen pona mute
sina ken lukin e toki awen ni
sina ken kin toki tawa jan ante pi toki pona
o kama tawa tomo toki pi toki pona

sina wile pona mute la o pana e sona pi toki pona tawa jan ante
o pana e sona tawa jan pona sina
o sitelen e toki awen
o toki kepeken toki pona
toki pona li toki pona
mi wile e ni
jan mute li sona e ona
sina ken kama e ni

#pilin ike (from: http://en.tokipona.org/wiki/Dark_teenage_poetry)
mi lon pimeja
waso ike li tawa sike lon lawa mi
pipi jaki li moku lili e noka mi
mi wile e pini

Appendix C: Program source code

The following files:

- cnf.py -- a CFG-to-CNF grammar converter
- cky.py -- a CKY parser
- TPgrammar.txt -- a plain-text copy of the CFG for toki pona
- data/output.txt – a copy of all parses generated for the 100 sentence corpus

are available online at:

- http://www2.hawaii.edu/~ztomasze/ics661/source/