A Hebrew Tree Bank
Based on Cantillation Marks

Andi Wu

GrapeCity Inc.
andi.wu@grapecity.com

Kirk Lowery

Westminster Hebrew Institute
klowery@whi.wts.edu

Abstract

In the Masoretic text of the Hebrew Bible (HB), the cantillation marks function like a punctuation system that shows the division and subdivision of each verse, forming a tree structure which is similar to the prosodic tree in modern linguistics. However, in the Masoretic text, the structure is hidden in a complicated set of diacritic symbols and the rich information is accessible only to a few trained scholars. In order to make the structural information available to the general public and to automatic processing by the computer, we built a tree bank where the hierarchical structure of each HB verse is explicitly represented in XML format. We coded the punctuation system in a context-tree grammar which was then used by a CYK parser to automatically generate trees for the whole HB. The results show that (1) the CFG correctly encoded the annotation rules and (2) the annotation done by the Masoretes is highly consistent.

1. Introduction

Linguistic annotation of written texts has been one of the main efforts of language resource development in recent years, due to the availability of electronic texts in large quantities and the advancement of NLP technologies. However, text annotation is not a modern invention. It dates back at least to the 9th century when a group of Jewish scholars, called the Masoretes, annotated the text of the Hebrew Bible (HB). One of the annotation tasks they undertook was to punctuate the text in a systematic way so that the verses may be read or chanted with the correct intonational oral units. In the eyes of a modern linguist, the punctuation represents a hierarchical bracketing of each verse, marking a structure that resembles the prosodic structure (Selkirk 1984) of the verse.

The punctuation system consists of a set of cantillation marks[1]. There are three uses of the cantillation: syntax (text segmentation), phonology (where the accent occurs), and music (melodies). Unlike modern punctuation marks that are independent characters, the cantillation marks are small diacritics added to the consonant characters of the text, which makes them hard to identify. Besides, the marking system is so complicated with such a variety of symbols that only a few trained scholars know how to use it. As a result, the rich structural information has remained hidden for most of the contemporary readers.

In order to make the information accessible to common readers for better understanding of HB and to the computer for automatic processing, we researched this punctuation system to work out the rules underlying the annotation, coded the rules in a context-free grammar, and parsed the whole HB with this CFG. The result is a prosodic tree bank of HB where the structure of each verse is represented in a format that is familiar to modern linguists. The tree bank can be used to view the structures of the verses or as a basis for developing a syntactic tree bank of BH.

2. The Cantillation System

2.1. Types of cantillation marks

There are two types of cantillation marks: conjunctive marks and disjunctive marks. They serve different functions but are both structurally significant.

Conjunctive marks group two or more words/morphemes into a single unit. When a word[2] bears a conjunctive mark, it is supposed to be pronounced "together with" the following word, with no break between the two. There are several different diacritic symbols of conjunctive marks, but their functions are similar.

In the text we use, which is from Groves & Lowery (2006), some words do not carry any cantillation marks. They were not independent words in the original Maroretic text and became words only after further segmentation in more recent analysis. These words are always grouped with the word that immediately follows it, as if it carried a conjunctive mark. We will consider those words as having a zero (invisible) conjunctive mark. This way every word in the verse will have a cantillation mark.

The disjunctive marks, on the other hand, are much more complicated. They divide and sub-divide a verse into successively smaller units until a single word or a unit

[1] The cantillation marks show how a text is to be sung. See http://en.wikipedia.org/wiki/Cantillation. They are also called "accents". For consistency of terminology, we will always call them "cantillation marks" or just "marks".

[2] The distinction between words and morphemes is fuzzy in Hebrew. Many segments are words syntactically but are traditionally treated as morphemes simply because they are not independent phonologically. In this paper, we will use "word" to refer to any syntactically independent unit regardless of its phonological status.

formed by conjunctive marks is reached. The structure formed by this division is hierarchical and more much interesting syntactically.

2.2. Hierarchy of disjunctive marks

The disjunctive marks can be classified into different ranks according to their dividing power. The mark of the highest rank (Rank-1) is *Soph Pasuq* which divides between verses and can be considered the root node of a verse. The Rank-2 marks (such as *Athnach*) divide a verse into two halves, with the major break of the verse occurring right after the word bearing the mark. Each of the two parts can then be sub-divided by Rank-3 marks. There are also Rank-4 and Rank-5 disjunctive marks that further divide the segments resulting from the division of a higher-rank mark.

Take Genesis 1:3 as an example. If we use English words instead of Hebrew words, use hyphen to stand for conjunctive marks, and use diacritic numbers to stand for disjunctive marks of different ranks, the verse will look like this[3]:

and- said- God^{3} be- light^{2} and- was- light$^{2\ 1}$

We can see that the last word has two marks, the Rank-1 mark (*Soph Pasuq*) indicating the end of a verse and a Rank-2 mark indicating the end of the second half of the verse. We can also see the primary break of this verse is after the first "light". The secondary break is after "God". The units formed by the conjunctive marks are "and said God", "let be light" and "and was light". The equivalent bracketing of this verse is:

[[[and said God] [be light]] [and was light]]

As we can see, the structure represented by the cantillation marks provides valuable information for the correct reading of this verse. Without the cantillation marks, there is nothing there to prevent us from getting the following wrong analysis where "let there be light" and "there was light" are conjoined to serve as the object of "God said":

[[[and said God] [[be light] [and was light]]]

Although Hebrew reads right-to-left, the computer will read left-to-right. Besides, the left-to-right order is also more natural for most readers of this paper. Therefore we use this order throughout the whole paper.

2.3. Two different systems

Two different cantillation systems are used in the Masoretic text: the *poetic* system that is used in the books of Psalms, Proverbs and Job and the *prosaic* system that is used in all the other books. The symbols used in the two systems overlap a great deal, but they are used in quite different ways in two different rule systems. However, the annotation principles behind the rule systems are the same. Both have conjunctive and disjunctive marks and both represent hierarchical structures.

3. The Cantillation Grammars

3.1. Existing syntactic analyses

Attempts have been made since the 17th century to figure out the syntax of the cantillation system. Among them are Wickes (1881), Price (1990), Richter (1999), Jacobson (2002) and (BFBS 2002). Each of them can be coded in a context-free grammar and used by a parser to generate tree structures for each verse. The geometry of the trees will of course vary depending on the grammar to be adopted. The major difference between those analyses is whether the rules are binary or not.

The binary analysis (e.g. Wickes 1970; BFBS 2002) views the structure as a continuous binary division of a verse: a Rank *n* disjunctive mark divides a segment into two parts, each of which is then further divided into two parts by the Rank *n+1* marks, if any. In the trees produced by this analysis, every non-terminal node is binary-branching.

The non-binary analysis (e.g. Price 1990) adopts a flatter structure in places where a disjunctive mark of a given rank appears more than once in a segment. Consider the following sequence where both Seg-A and Seg-B end in a word that bears a Rank-4 disjunctive mark:

Seg-A^{4} Seg-B^{4} Seg-C^{3}

The non-binary analysis will produce a flat structure:

[Seg-A Seg-B Seg-C]

The binary analysis, on the other hand, will group those units iteratively in a binary fashion and produce the following structure:

[Seg-A [Seg-B Seg-C]]

3.2. Our analysis

3.2.1. General Design

We built a context-free grammar of our own to encode all the annotation rules. In this grammar, we let each word or terminal node have a "part-of-speech" which is the name of the cantillation mark it carries. So there are as many parts of speech as the different types of cantillation marks, either conjunctive, zero-conjunctive (for words that do not have a mark), or disjunctive.

All the branching rules in this grammar have the form *A* → *w A* where *A* is the POS (the cantillation mark) of a node and *w* is the sequence of nodes that precede *A*. This says that, given a string of nodes on the RHS of the rule, the POS of the LHS node is always is POS of the last node on the RHS. In other words, the last RHS node is the "head of a phrase" and its POS projects to its parent. This is so because the cantillation mark on each word always indicates the amount of break/pause on the right-hand side of the word. Here is a sample rule:

Athnach → Tiphcha Athnach

This says that combining a segment ending in Tiphcha and a segment ending in Athnach produces a larger segment that also ends in Athnach.

[3] Due to space limitation, we are not able to show structures of greater depth involving Rank-3, Rank-4 and Rank-5 disjunctive marks.

3.2.2. Rules for zero-conjunctive marks

Words that have zero-conjunctive marks need to be grouped into a unit that has a non-zero cantillation mark, either conjunctive or disjunctive. This unit can then have a POS and participate in the application of other rules. The rules that group those words have the form $A \rightarrow w\ A$ where A is a word that bears a non-zero mark and w is a sequence of one or more words bearing zero marks.

The trees formed by these rules have a flat structure and act like terminal nodes. We prefer a flat tree here because the structure we are dealing with here is usually morphological rather than syntactic in nature.

3.2.3. Rules for non-zero conjunctive marks

Rules for conjunctive marks build the basic building blocks for higher-level structures which are formed by disjunctive marks. Words with non-zero conjunctive marks need to be grouped into a unit that has a disjunctive mark, which will enable the unit to participate in the application of disjunctive rules. Rules of this type all have the form $B \rightarrow A\ B$ where A is a word with a conjunctive mark and B a word with a disjunctive mark.

When there is more than one conjunctive word before a disjunctive word, the rules can be applied iteratively to group all segments of the same type into a single unit. Given a string of *a a a b* where *a* is a conjunctive word and *b* a disjunctive word, for example, the structure resulting from such rule application will be [*a* [*a* [*a* *b*]]].

3.2.4. Rules for disjunctive marks

After the conjunctive rules are applied, the disjunctive rules will group the basic segments into increasingly large units until the whole verse is reduced to a single node. These rules all have the form $B \rightarrow A\ B$ where A is a word with a Rank *n* disjunctive mark and B a word with a Rank *n+1* mark. When a Rank *n* mark is preceded by more than one Rank *n+1* mark, the rules will apply iteratively, as in the case of non-zero conjunctive rules.

3.2.5. Why go binary?

As we can see, the rules in 3.2.3 and 3.2.4. are all binary rules. They produce a multi-layered right-branching structure instead of a flat structure in cases where there is a sequence of equal-rank units. We adopted this binary analysis because we believe the system is binary in nature. Besides, there are good linguistic and computational reasons for the binary approach.

Linguistically, the layered structure usually corresponds better to the syntactic structure that modern linguists expect. Given a sequence such as “the big oak tree”, the binary rules will produce [the [big [oak tree]] instead of [the big oak tree], for instance. Obviously, the binary analysis is able to provide a structure that brings out more of the phrase’s syntactic structure. Because Hebrew is largely right-branching, the binary structures produced by our rules correspond fairly well to the syntactic structures of Biblical Hebrew.

Computationally, the binary analysis reduces the number of CFG rules we have to write. For flat structures, we have to write a separate rule for each case with a different number of RHS nodes. To cover the following sequences, for example,

A B
A A B
A A A B

we will need 3 different rules:

$B \rightarrow A\ B$
$B \rightarrow A\ A\ B$
$B \rightarrow A\ A\ A\ B$

However, the binary analysis only needs one rule:

$B \rightarrow A\ B$

In addition, the binary rules are more robust as they capture generalizations instead of listing all the possible cases. We do not have to know in advance how many units of equal rank can appear in succession, as we do in the flat analysis.

4. A Cantillation Tree Bank

In order to make the rich structural information encoded in the cantillation marks accessible to the general public and readable by the computer for further processing, we decided to build a tree bank where the structure of every verse is explicitly represented as a tree in XML format. We also built a tree viewer where the trees can be viewed graphically in the way that most linguists are familiar with.

4.1. Creation of the tree bank

The tree bank is created automatically by a CYK parser that uses the CFG grammar described in the previous section. Two grammars were used: a poetry grammar for parsing Psalms, Proverbs and Job and a prose grammar for the rest of HB. Some verses in HB (mostly in the books of Ezra and Daniel) were written in Aramaic instead of Hebrew, but they were marked with the same cantillation system. So in terms of the cantillation grammar, there is no distinction between Hebrew and Aramaic and the same grammar can be used for parsing both.

Since the annotation is supposed to unambiguously mark the structure of every verse, we expect to parse every verse successfully with exactly one tree assigned to it, given that (1) the annotation is perfectly correct and (2) the CFG grammars correctly encoded the annotation rules. The actual results we got were not far from our expectation: all the 23213 verses were successfully parsed, of which 23099 received exactly one complete parse tree. The success rate is 99.5 percent. The 174 verses that received multiple parse trees all have words that carry more than one cantillation mark.[4] Just like having multiple parts of speech on a word, this can create syntactic ambiguity and result in multiple parse trees.

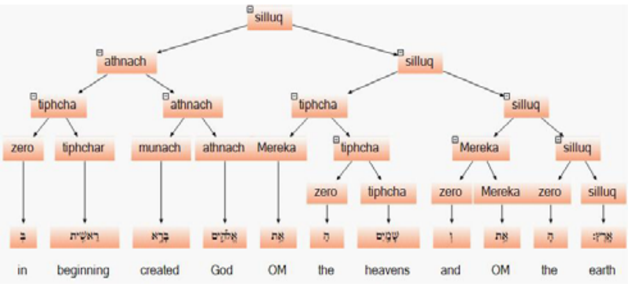
We have good reasons to believe that the grammars we used are correct. We would have failed to parse some verses if the grammars had been incomplete and we would have gotten multiple trees for a much greater number of verses if the grammars had been ambiguous.

[4] This does not include the last word of every verse which always carries two cantillation marks: a disjunctive mark like *Silluq* plus a *Soph Pasuq* which is a verse divider.

We can also see that the Masoretes did an excellent job at the annotation. Considering the fact that the annotation was done completely by hand more than 1000 years ago, without the assistance of computers, the error rate is extremely low even for modern standards.

4.2. Viewing the trees

We created a tree viewer that can read the XML files produced by the parser and display the structures in a linguist-friendly form. Here is the tree for Genesis 1:1 (with English gloss), displayed left-to-right to make it easier to view for non-Hebrew readers.[5]

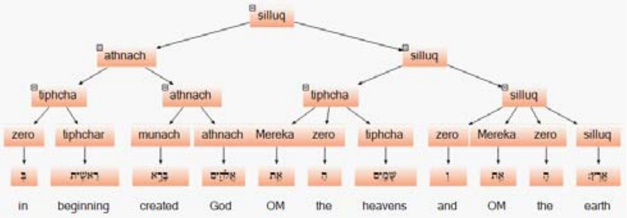


The tree actually presents the derivational history of rule application. The names of the cantillation marks serve as the category of each word: *silluq*, *athnach* and *tiphcha* are disjunctive marks; *munach* and *Mereka* are conjunctive marks; *zero* is a zero-conjunctive mark. We can also see the projection of the head categories. "OM" in the gloss stands for "object marker" which appears before each definite object NP.

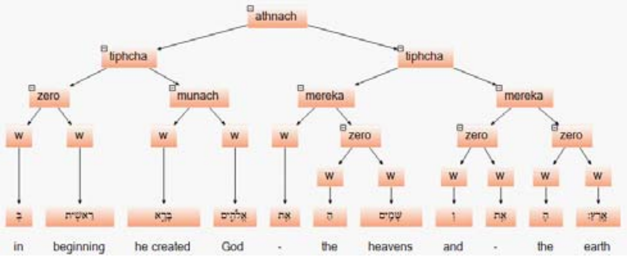
The resemblance of this tree to a syntactic tree is obvious. The nodes covering "in beginning", "the heavens", "the earth" and "the heavens and the earth" are all syntactic units. Also noticeable is the treatment of "and" which is not syntactically correct but fairly reasonable for a prosodic structure.

4.3. Transforming the trees

Because the tree bank is in XML, it is easy to transform the trees to meet the requirements of different standards. To get the "flatter trees" of Price (1990), we can flatten certain sub-trees by putting nodes with marks of equivalent ranks on the same level. After this transformation, the tree of Genesis 1:1 will look like this, where some sub-trees produced by binary conjunctive rules have been flattened:



Some analyses, such as BFBS (2002), prefer to treat the cantillation mark on each word as a punctuation mark occurring in the space between this word and the following word. This way the mark can be viewed as a node that joins the preceding segment and following segment. In BFBS (2002), when two nodes are combined to form a new node, the label of the new node will be the name of the cantillation mark that joins the two nodes. Given a rule of *A* → *B A* in our CFG, where *B* carries the cantillation mark *X*, the BFBS rule will be *X* → *B A* where *B* and *A* will not have their cantillation marks as their labels. This rendering of the tree does look more intuitive to some people and we have a process that can transform our trees to this view. Genesis 1:1, for example, will have the following tree after the transformation:



4.4. Building syntactic trees

We are currently building a syntactic tree bank of HB. A careful examination of the cantillation tree bank shows that many of the nodes in its prosodic structures correspond or can be adjusted to correspond to syntactic units. The brackets around those units can provide valuable information for syntactic parsing. Therefore, we extracted the bracketing information from the cantillation trees and used it to guide our syntactic parser. This information has greatly reduced both the complexity of the parsing process and the amount of manual work needed in building a tree bank. We will describe this project in a different paper.

5. References

BFBS (2002). The Masoretes and Punctuation of Bibilcal Hebrew. British & Foreign Bible Society, Machine Assisted Translation Team.

Groves, A & K. Lowery, eds. (2006). *The Westminster Hebrew Bible Morphology Database*. Philadelphia: Westminster Hebrew Institute.

Jacobson, J.R. (2002). *Chanting the Hebrew Bible*. Philadelphia: The Jewish Publication Society.

Price, J. (1990). *The Syntax of Masoretic Accents in the Hebrew Bible.* Lewiston/Queenston/Lampeter: The Edwin Mellen Press.

Richter, H. (2004). Hebrew Cantillation Marks and Their Encoding. http://www.lrz-muenchen.de/~hr/teami.

Selkirk, E. (1984). *Phonology and Syntax: The Relation between Sound and Structure*. Cambridge, MA: The MIT Press.

Wickes, W. (1881). *Two Treatises on the Accentuation of the Old Testament.* Rev. Reprint by KTAV, New York, 1970.

[5] The tree viewer has the option of displaying a tree either left-to-right or right-to-left.