

A Brief User's Guide
for the
Accordance Hebrew Syntax Database

Robert D. Holmstedt, John A. Cook, Martin G. Abegg, and Roy B.
Brown

October 2016

TABLE OF CONTENTS

1. History of the Accordance Hebrew Syntax Project.....	1
The Purpose of this work.....	3
2. Terms, Abbreviations, and Basic Syntax Searching.....	4
2.1. Brief Definitions of Terminology.....	4
2.2. Special Syntax Features.....	5
2.3. Viewing Syntax.....	7
2.3.1. Instant Details Box.....	7
2.3.2. Syntax Diagram Pane.....	7
4. Syntax Abbreviations.....	10
5. Simple Syntax Searching.....	11
6. Construct Tab Syntax Searching.....	12
2.6.1. Clauses and Phrases.....	14
2.6.2. Clause Searching.....	16
2.6.3. Phrase Searching.....	18
2.6.4. Null Searching.....	20
2.6.5. Compounds.....	20
2.6.6. A note on the Specifier (Definite Article).....	21
2.6.7. Rare non-alignment of morphology and syntax.....	22
3. Linguistic Background.....	24
3.1. Hierarchical, Non-Binary Phrase Structure.....	25

3.1.1. Binary versus Non-Binary.....	31
3.1.2. ‘Constituents’.....	34
3.1.3. “Where’s the Direct Object?”.....	35
3.2. Non-Movement Approach to Constituent Discontinuity....	36
3.3. Inclusion of Null Constituents.....	38
3.4. The Narrow Syntactic Focus of the Database.....	39
4. Appendix: Cross-Referencing in the Syntax Database.....	42
4.1. Dislocations.....	42
4.2. Relative Clause Resumption.....	43
4.3. Ellipsis (‘Gapping’).....	45
5. Sample Searches.....	50
5.1. Null Predicate (=“Verbless”) Clause.....	50
5.2. Null Predicate Without Participial Complement.....	51
5.3. One-Part Null Subject, Null Predicate Clause.....	52
5.4. Null Predicate Clause with a Nominal Complement.....	54
5.5. Pronominal Copula.....	54
5.6. הָיָה and הָיָה as Copula.....	55
5.7. Monovalent Verbs.....	56
5.8. Trivalent Verb הָיָה with NP Comp and PP Comp.....	56
5.9. Infinitive “Absolute” used Verbally.....	57
5.10. Subject-Participle Clause.....	58

5.11. Discontinuous Predicate.....	58
5.12. Discontinuous Subject.....	59
5.13. Compound Subjects.....	60
5.14. Subject-Verb Clause.....	62
5.15. Verb-Subject Clause.....	64
5.16. Adjuncts.....	64
5.17. Apposition.....	66
5.18. Subordinate Clauses.....	67
5.19. Vocatives.....	68
5.20. Conjunctions versus Focus Words.....	69
5.21. Dislocation.....	71
5.22. Ellipsis/Gapping.....	71
REFERENCES.....	73

1. History of the Accordance Hebrew Syntax Project

In 2008 Martin G. Abegg Jr. (Trinity Western University) and I began collaborating, with significant input from John A. Cook (Asbury Theological Seminary) and Roy B. Brown (Oaktree Software), on the development of a syntactic database for all ancient Hebrew texts. The Accordance Syntax Database and now this Manual are the first-fruits of that rich and ongoing collaboration.

Morphologically-tagged databases of the Hebrew Bible have existed in some form for well over two decades. Within the last decade databases for the epigraphic Hebrew texts, Qumran texts, Ben Sira, texts from the Judean Desert, and the Mishna have been released. In contrast, syntactically-tagged databases, while they have been in production for decades, have only been made publicly available within the last ten years: the *WIVU Emdros* database of the Werkgroep Informatica of the Vrije Universiteit in Amsterdam (wivu.dans.knaw.nl/, now called the ETCBC database and available at <https://shebanq.ancient-data.org>) and the *Andersen-Forbes Analyzed Text of the Hebrew Bible* (www.andersen-forbes.org).¹ While both existing databases were produced by noted Hebraists and are ground-breaking in distinct ways (and so immensely valuable), we saw a need for a third database: one that was focused more narrowly on syntax and covered both biblical and extra-biblical texts. As the project design matured, it was clear that four features make our

1. Yet another syntactic database remains in production: the *Westminster Hebrew Syntax* database (www.grovescenter.org/GC/projects/westminster-hebrew-syntax; see also Lowery 2009). Its current status is unclear.

database unique.

First, our project covers all ancient Hebrew in the first millennium B.C.E. This will not only provide access to the non-biblical texts, it will also facilitate comparative and historical syntactic analyses (e.g., comparing the syntactic features of ‘late’ biblical books to select Qumran texts).

Second, our project has not been designed as a stand-alone database, but is native to the Accordance Bible database software. Although the data files are simple enough so that they could be easily incorporated into any existing database software, the advantages of working with an existing software package have been manifestly clear: access to programming expertise at every step of development and the luxury of not needing to use any existing mark-up language (e.g., *HTML*, see Kroeze 2002; *XML*; see, e.g., Kroeze 2006, 2008) or database engine such as Emdros (see, e.g., Petersen 2004).

Third, our database is focused very tightly on clause syntax: we build on existing morphological databases (which also facilitates our schedule) and do not address semantic or discourse-pragmatic features of the Hebrew texts. In contrast, the Andersen-Forbes database, for example, includes such non-syntactic issues as semantic categories (e.g., as ‘purpose’, ‘result’, even ‘undesired outcome’) and additional issues of less grammatical import such as the time, region, dialect, register, and/or source of the biblical texts (Andersen and Forbes 2003:44).

Fourth, our approach to the parsing and analysis of the syntax is

unique in its Chomskyan generative theoretical orientation.² Thus, our database will also serve as a necessary counterpart to the other databases, which represent different theoretical approaches to the nature of syntax and syntactic analysis.

The purpose of this work

This short manual is intended for Accordance users who have the Hebrew syntax database and desire both to understand more about the construction of the database and to know better how to perform simple and complex syntactic searches.³

In the following sections, we provide information on the terms and abbreviations as well as basic instructions for syntax searching (§2). In the subsequent section, we describe the linguistic background of the project and database (§3), after which we briefly describe the use of cross-referencing (§4). The final section details a few dozen syntax searches in order to illustrate the process of building a search using the Hebrew construct window. (§5).

-
2. The Andersen-Forbes database is strongly influenced by varieties of non-Chomskyan generative phrase structure grammar, along with a mixture of corpus linguistics, computational linguistics, and some functionalism. Even so, they explicitly identify themselves as “generativists” in Andersen and Forbes 2012, which is a companion to their database.
 3. We are immensely grateful for the feedback that numerous Accordance Forum users provided in a draft of this work.

2. Terms, Abbreviations, and Basic Syntax Searching

2.1. Brief Definitions of Terminology

The following is a list of the Syntax database terminology currently used in Accordance followed by a short definition.

- Constituent—The syntactic unit that combines with other units to form larger constituents (phrases and clauses).
- Phrase—A constituent consisting of a single word to a large and complex group of words relating to each other syntactically, functioning as a constituent of a larger phrase, and lacking its own predication.
- Clause—The most complex type of constituent, consisting of a subject and predicate.
 - Independent (Main) Clause—A clause that can stand on its own as a sentence and conveys a statement, question, exclamation, or command.
 - Dependent (Subordinate) Clause—A clause that cannot stand on its own as a sentence, but is a constituent within a larger clause (the superordinate clause, often but not always a main clause) and conveys causal, temporal, result, purpose, etc., information relating to the main clause.
- Parenthesis—A parenthesis is a special clause category that interrupts the syntactic flow, whether within a clause or within a string of coordinated clauses.
- Subject—The “doer” (agent) or “experiencer” (patient) of the predicate.
- Predicate—The verb and any modifiers; also known as the Verb Phrase (VP).
- Complement—The constituents required by either a verb or a preposition in order to “complete” the (valency) semantics of each.
- Adjunct—The constituents not required by any syntactic or (valency)

semantic requirements, but add additional information about a verb or noun.

- Specifier — The definite article.
- Appositive—A clause, phrase, or word that elaborates on a preceding clause, phrase, or word of the same type.
- Vocative—A word or phrase of direct address that is referentially related to a (mostly animate) syntactic participant (e.g., the subject) but plays no grammatical role within the clause.
- Exclamation and Interjection—A word or phrase that interrupts the normal syntax to orient the attention of the addressee, often with an emotional content, but which has no grammatical role in the clause.
- Casus pendens (dislocation)—A noun or pronoun placed outside of a following clause and resumed within the clause by a resumptive pronoun.
- Null—A syntactically real, but phonologically empty constituent, such as an “implied” subject or an “understood” verb.
- Antecedent—A word, such as a noun (e.g., David), to which another word, such as an anaphoric pronoun (e.g., he) points back to.

Although the above definitions have been tailored to the presentation of the Accordance Hebrew syntax database, an accessible entrance to many these concepts in general linguistics is David Crystal’s *A Dictionary of Linguistics and Phonetics* (Crystal 2008).

2.2. Special Syntax Features

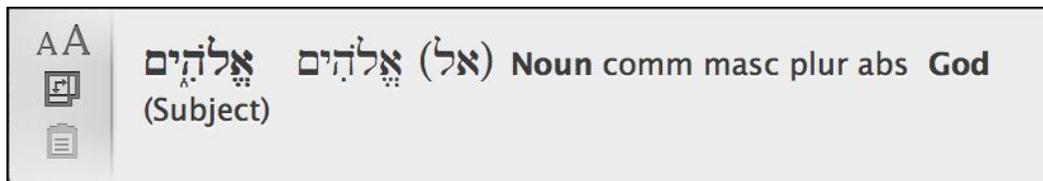
In addition to the standard syntax terminology described above, the Accordance Syntax database has the following special features:

1. Speech—If an independent clause is in the context of direct speech (i.e., is a quote of some speaker), it is tagged as speech to allow searching for speech versus non-speech. Speech occupies an existence between independent clauses and dependent clauses, since speech has its own deictic center that may differ from the superordinate clause but is also the complement of a verb of speaking. Within the Accordance Syntax database, the first clause of direct speech is kept within the domain of the higher clause containing the verb of speaking, but subsequent speech clauses are separated out as independent clauses. For this reason, Accordance contains two methods for searching for speech (or excluding it), one by the *Begin Speech* tag and the other within the *Clause Type Pop-up window* under *Select Clause Speech*.
2. Nulls—Though by their very nature null constituents are not written in the Hebrew text, their syntactic presence is noted in the Accordance Syntax database to show where an understood subject, predicate, or complement would be placed within the clause.
3. Antecedents—To assist in showing syntactical relationships, antecedent identifier labels (e.g., 1, 2, 3) are sometimes added to show relationships between a word and its antecedent. A null identifier may also be combined with an antecedent identifier.
4. Discontinuous constituents—see below §3.2 for an explanation of constituent discontinuity. The search interface accounts for this (see below §2.6.3).

2.3. Viewing Syntax

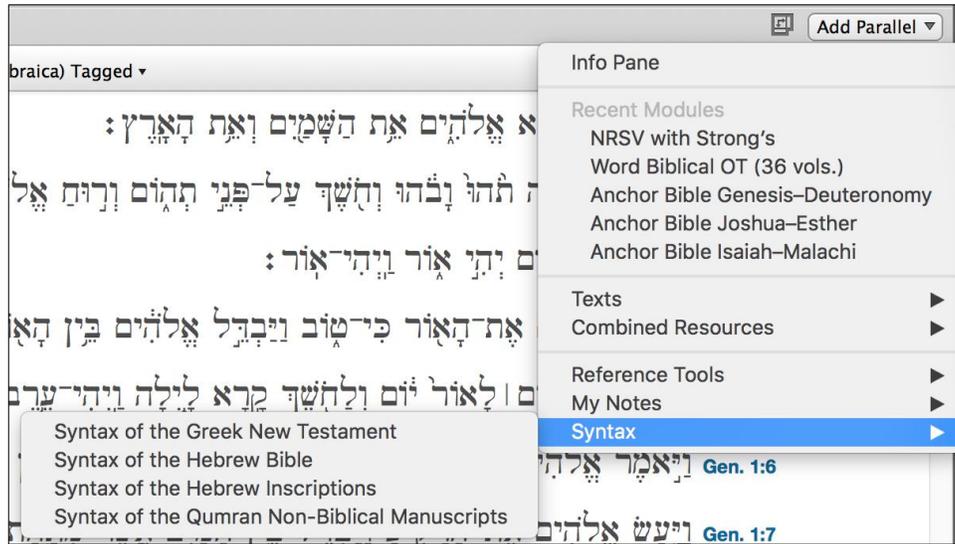
2.3.1. Instant Details Box

There are two ways to view the syntax of a word, phrase, or clause in Accordance. The first way is to place the cursor over the individual word to view its syntactic label in the Instant Details box. This gives a quick view of the morphology and syntax of the word, but does not show the syntactical relationship of the word to other words in the same clause. Below is an example for a Subject:



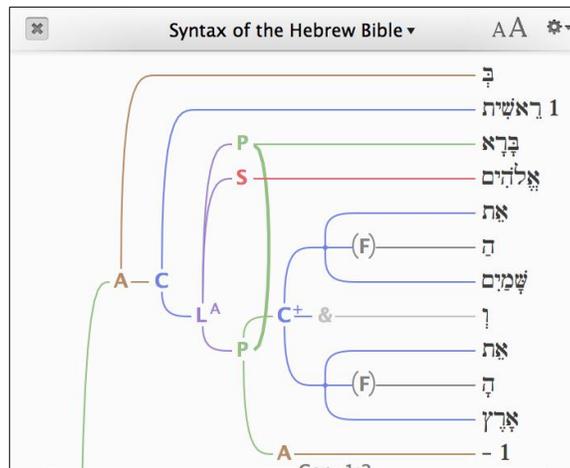
2.3.2. Syntax Diagram Pane

To view the syntactic relationships beyond a single word one must open the *Syntax diagram* pane so that the syntax can be viewed in parallel with the text. The Syntax pane is opened by clicking on an item in the “Syntax” submenu under the “Add Parallel” pull-down menu as shown below. (If there are no syntax modules installed, this submenu will not have any items.)



The Syntax pane will show the syntactical relationship between words of a parallel text in the form of a tree. The tree will be color coded according to the syntax of the tree “branch.” The actual words of the text appear at the ends of the tree, as shown below.

The tree can be made larger or smaller with the “A” buttons on the top bar. The tree can be viewed either vertically as shown above, or horizontally by clicking on the Move Down item in the “Gear” menu.



If the Syntax pane is selected (by clicking anywhere on the pane), the scroll bar will control the scrolling of the pane, with other parallel

2.4. Syntax Abbreviations

The following abbreviations are used for each syntax term on the tree:

- N—Independent (main) clause. An N with an under-arc indicates that the clause is direct speech, with the number of arcs representing the level of speech (the more arcs, the more nested the direct speech). A superscripted C or A after the N (e.g., N^C) indicates that the direct speech clause is the complement of a higher verb (i.e., a verb of speaking).
- L—Dependent (=subordinate/embedded) clause. A superscripted S, C, or A after the L (e.g., L^S) indicates the syntactic role the dependent clause plays within the higher clause.
- T—Parenthesis
- S—Subject
- P—Predicate (as a word = verb; as a phrase = Verb Phrase)
- C—Complement
- A—Adjunct
- F—Specifier (=the definite article)
- X—Appositive
- V—Vocative
- E—Exclamation or interjection
- D—Caus pendens (=left and right dislocation)
- “_” —Null constituent
- 1-9—Antecedent identifier (i.e., cross-referencing)
- -1 - -9—Null antecedent identifier (i.e., cross-referencing)
- “:”—Begin speech in the following independent clause
- U—Unknown

2.5. Simple Syntax Searching

Simple syntax structures can be found by entering the syntax term directly in the Search tab Entry box for a specific text. Note that syntax terms exist for both “words” (e.g., a noun in its role as a subject), “phrases” (e.g., a noun and all its modifiers as a subject), or “clause” (a subject and predicate).

Entering syntax items may be done with or without combining other search elements, such as morphological terms, or lexical items. The syntax terms can generally be combined in a search argument the same way as morphological terms, using the standard Accordance commands.

Syntax terms are usually entered automatically by using the appropriate item in the submenu of the *Enter Tag* item in the *Search* or *right-click* menu as shown. All of the basic syntax tags are listed below the morphological tags in the menu list.

For example, finding a subject within three words of a verb would be entered as (note that a search like this produces hits with the subject in front of the verb and vice versa):

[SUBJECT] <WITHIN 3 Words> [VERB]

Adjective...	
Noun...	
Paragraph	
Particle...	
Pronoun...	
Suffix...	
Verb...	
ANY Tag	⌘⌘Y
Tag Entry	⌘⌘G
Holmstedt Syntax	
Subject	
Predicate	
Complement	
Adjunct	
Appositive	
Vocative	
CasusPend.	
Specifier	
Exclamation	
Unknown	
Null	
Antecedent	
BeginSpeech	

To find where the word אֱלֹהִים is used as a subject in the HMT-W4 module, the entry would be:

אֱלֹהִים@[SUBJECT]

2.6. Construct Tab Syntax Searching

More complex syntax searching can be performed by use of the Accordance Construct tab. The Greek or Hebrew Construct tabs are opened from the *New Construct* submenu in the *New Tab* submenu in the *File* menu, or with the respective Command keys. If the Syntax module is installed, the Construct tab palette will show all of the syntax items below the morphological items, as shown to the right. As with many of the morphological items, the syntax items are dragged into the desired columns to form a search expression.

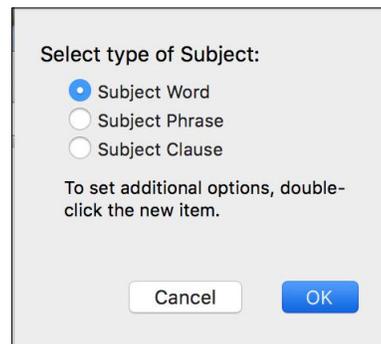
Most of the syntax items work the same way as the morphological items in the Construct tab, and can be easily mixed together with the morphological items. For example, to find the same search (the word אֱלֹהִים used as a subject), the respective items would be dragged into the right-most column as shown to the left.



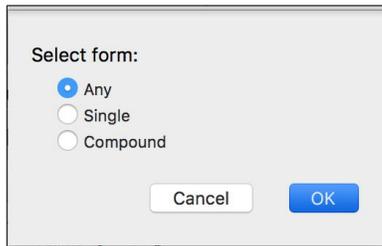
The most basic syntax terms, namely Subject, Predicate, Complement, and Adjunct can be expressed as either a simple word, a type of Phrase, or a type of Clause. To avoid confusion between the various ways to express Syntax with these basic terms, the dialog box to

the right will appear after dragging a simple Subject, Predicate, Complement, or Adjunct.

This dialog is intended to help the first-time user distinguish between the various uses of a specific Syntax term.



If a simple word is selected after it is dragged into a column, then

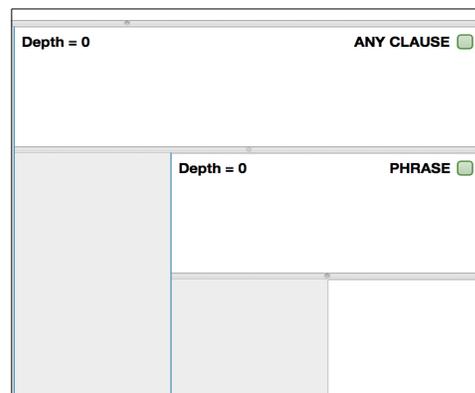


when it is double-clicked, a different dialog will appear, as shown to the left. This dialog box will allow the user to select a single or compound form of the term, such as a compound Subject, as shown to the left. For

most searches, the dialog can remain as *Any*.

2.6.1. Clauses and Phrases

The Clause and Phrase Syntax items can enclose groups of other terms and can enclose other Clause or Phrase items to allow nested searching to any level, as shown to the right.



It is useful to think of the Clause or Phrase item as a “Construct within a Construct.” Each Clause or Phrase contains its own columns, with space above the columns for

connecting items (by, for instance, using the AGREE or WITHIN functions), all of which belong only within the enclosing Clause or Phrase being searched.

Note that the basic structure of every clause is for the subject phrase and predicate phrase to be the first level within the clause, followed by any complement or adjunct phrases within the predicate or adjunct phrases within the subject. This basic structure is illustrated in the search template below.

Depth = 0		ANY CLAUSE <input type="checkbox"/>	
Depth = 0		PREDICATE PHRASE <input type="checkbox"/>	Depth = 0
		SUBJECT PHRASE <input type="checkbox"/>	
Depth = 0	ADJUNCT PHRASE <input type="checkbox"/>	Depth = 0	COMPLEMENT PHRASE <input type="checkbox"/>
		<input type="checkbox"/>	PREDICATE
		Depth = 0	ADJUNCT PHRASE <input type="checkbox"/>
			<input type="checkbox"/>
			SUBJECT

It is also very important to try to build a search for a desired syntactic construction so that it mirrors the nested structure of the construction as closely as possible. To help with this, it is very helpful to consult the syntax diagrams to see how nesting works. And while it is typically advisable to search with the *Depth* set to 0 (see below on the Depth feature), increasing the Depth to 1 or 2 can be a means to finding similar but not identical constructions. Increasing the Depth setting allows the searched-for items to be found in lower levels of the

hierarchy.

After dragging a Clause or Phrase item into a column, two new columns within the Clause or Phrase replace the original column. The number of columns within a Clause or Phrase can be adjusted by dragging the vertical blue dividing line. As with the main Construct, connecting items (e.g., using WITHIN) can be placed above the columns within a Clause or Phrase.

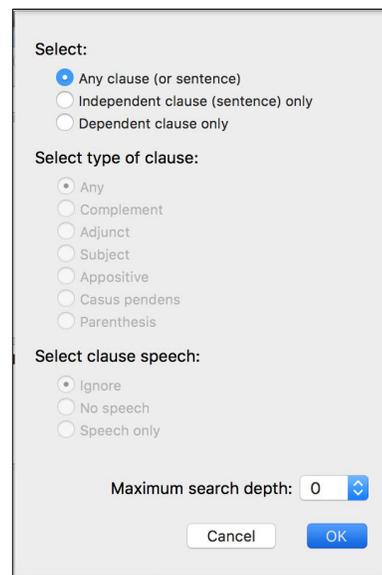
Note also that the Search tab pop-up menu labeled “Scope” (see below) must be set to Chapter or Book when a Clause or Phrase item is used, since clauses and phrases may cross verse boundaries.



2.6.2. Clause Searching

After dragging a Clause item into a search construct, the dialog box shown to the right will appear. The following options are available to specify details of the Clause:

1. **Type of Clause**—The default option is “any type of clause,” though the search can be limited to independent or dependent clauses.
2. **Type of Dependent Clause**—Once either



an independent or a dependent clause is chosen in the first set of options, the next set of clause options will become available and allow the choice of any, complement, adjunct, subject, appositive, casus pendens, or parenthesis. These options are straightforward for dependent clause types, but this set of choices also becomes available for independent clauses. Due to the complexities of direct speech, which straddles the independent/dependent divide, an independent clause that is also, say, a complement is necessarily a direct speech clause.

3. **Clause Speech**—Another way to limit the search to direct speech (or exclude it) is to use the third set of options, which become available when an independent clause is selected. This third set allows one to choose whether or not direct speech is included in the results.
4. **Maximum Search Depth**—The Depth feature allows for “squishy” searching. That is, if the precise syntax, with its phrasal nesting, is unclear or some variation in the results is desired, the Depth can be increased from 0 (= no squish) to 1, 2, etc., to allow the search to capture hierarchical variations. The number in the pull-down Depth menu indicates the maximum number of extra hierarchical terms between the Clause and the terms inside the Clause. That is, setting the depth to zero (0) means that the item must appear immediately below the Clause with no intervening terms in the hierarchy. The default depth when a Clause or Phrase is first added from the palette may be set in the preferences (Preferences>Greek & Hebrew>Default clause/phrase depth).

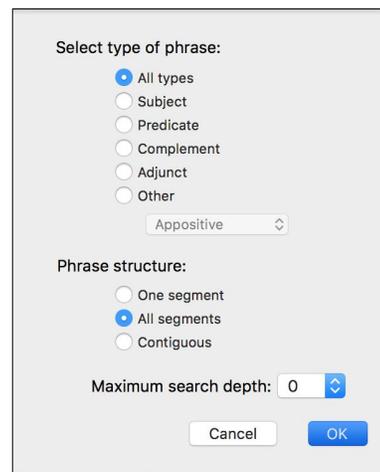
When searching within a Clause the following should be kept in mind:

1. If no terms (lexical items or morphological tags) are added inside the Clause Construct item, the search will still be performed, and all matching syntactic structures will be found, as indicated by a thin solid line drawn through each constituent. The solid lines drawn through the search results indicate level of hierarchy, with one line indicating the highest level (main clause) and multiple lines indicating lower levels of hierarchy (i.e., dependent clauses or even recursive dependency). See an example of this below in [c.Phrase Searching](#).
2. Connecting items, such as AGREE or WITHIN, can be added between the columns inside the Clause. If a WITHIN item connects to an item within the Clause that is another Clause or Phrase, the results are understood as the proximity from the last word of one phrase to the first word of the following phrase.

2.6.3. Phrase Searching

After dragging a Phrase item into a search construct, the dialog box shown to the right will appear. The following options are available to specify details of the Phrase:

1. **Type of Phrase**—The type of Phrase can be selected, whether Subject, Predicate, Complement, or Adjunct.
2. **Phrase Structure**—Due to the discontinuity



Select type of phrase:

- All types
- Subject
- Predicate
- Complement
- Adjunct
- Other

Appositive

Phrase structure:

- One segment
- All segments
- Contiguous

Maximum search depth: 0

Cancel OK

of many Hebrew constituents (see below, §3.2), it is possible to specify if a

search should find all elements of constituents, regardless whether they are split or not (“All segment”) or force the search to look for discontinuous constituents (“One segment”—finding split phrases) or non-discontinuous constituents (“Contiguous”—finding phrases without any splits).

3. **Maximum Search Depth**—As with Clauses, the Depth feature allows for “squishy” searching. That is, if the precise syntax, with its phrasal nesting, is unclear or some variation in the results is desired, the Depth can be increased from 0 (= no squish) to 1, 2, etc., to allow the search to capture hierarchical variations. The default depth when a Clause or Phrase is first added from the palette may be set in the preferences (Preferences>Greek & Hebrew>Default clause/phrase depth).

When searching within a phrase the following should be kept in mind:

1. If no terms (lexical items or morphological tags) are added inside the Phrase Construct item, the search will still be performed, and all matching syntactic structures will be found, as indicated by a thin solid line drawn through each constituent. The solid lines drawn through the search results indicate level of hierarchy, with one line indicating the highest phrasal level and multiple lines indicating lower levels of hierarchy or nesting, as shown in the example below. Note how the first verse of Genesis has two lines, indicating that it is somehow nested, and that the first non-nested material is **וַיְהִי-אֵוֶר**, indicated by the single line.



2. Connecting items, such as AGREE or WITHIN, can be added between the columns inside a Phrase. If a WITHIN item connects to an item within the Phrase that is another Clause or Phrase, the results are understood as the proximity from the last word of one constituent to the first word of the following constituent.

2.6.4. Null Searching

If the Null item is dragged to a column, then only Null forms will be found in that column. The Null item can be combined with other simple Syntax items, such as Subject or Predicate to find Null Subjects or Predicates respectively. The results of a Null search will be indicated by a “flag” or “pointer” icon in the position where the Null appears between two words. If a Null item is used in combination with other simple Syntax items, the Null item should be the top item in the construct column.

2.6.5. Compounds

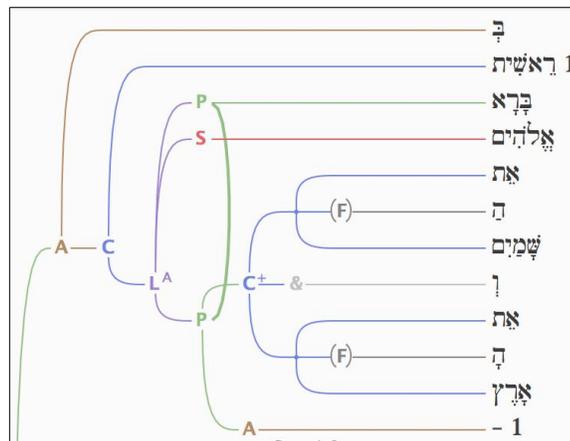
Compounds add an extra level of hierarchy to the normal Syntax

structure, since a compound constituent includes all of its coordinated parts. In order to facilitate searching of compound phrases, the extra hierarchy level of a compound term is ignored, so that the search results will be consistent in hierarchy between compound and non-compound phrases.

That is, a search for a Subject Noun will return both compound and non-compound hits if the *Any* option is chosen, otherwise the *Single* or *Compound* options will dictate the type of search results.

The treatment of Compound Syntax terms is also shown in the Syntax pane. The example below for Gen 1:1 illustrates a compound Complement:

There are three special things in the Syntax Diagram structure that indicate the Compound Complement form in this example:



1. A superscript “+” sign is added to the Complement term “C” to indicate that this is a Compound Complement.
2. The Complement terms enclosed by the Compound Complement do not have a “C”, but are simply joined together to the right of the Compound Complement to indicate that this extra level of hierarchy is ignored during searching.
3. The connecting words (such as וְ in this example) are strictly speaking not

part of the Compound Complement, but simply serve to connect the individual terms together. To illustrate more clearly the connecting words, a light gray ampersand (&) character is included between the Compound Complement, with a light gray line following the ampersand.

2.6.6. A note on the Specifier (Definite Article)

The definite article הַ, which is a specifier in syntactic terminology, can be searched for using the Specifier label, the Particle>Article morphology label, or the lexical item הַ. In terms of the phrase structure approach to the article, we included it within the phrase structure of the noun it precedes and is attached to. However, inserting the Specifier label into a construct immediately before a noun in, say, a subject phrase does not require an increase of the depth feature to 1. The searching treats the specifier on the same level as the main syntactic item it precedes.

2.6.7. Rare non-alignment of morphology and syntax

The morphology of Hebrew generally allows for a clear mapping of the syntactic relationships onto the words in a clause. For example, even in a fusional form like עָשִׂיתִּיהָ 'I did it', the 3ms clitic pronoun can be separately tagged for its syntactic role apart from the predicate syntax label given to the verbal part of the word.

However, one area of complexity in this morphology-to-syntax mapping concerns complex prepositions. For examples, in a word such

3. Linguistic Background

There are numerous complexities involved with an ancient language syntactic database project. They range from issues of *audience* to *theory* to *programming*. Among other questions, we asked ourselves (repeatedly, in many cases):

- *who will use this database and what will they expect to see?*
- *how much can we draw upon linguistic theory—and which one?—while still making the modules usable for the broadest audience?*
- *how much theory-internal structure can we set aside yet not produce a scientifically naive and theoretically flawed database?*

Our primary goal for the creation of the database is to produce a usable research tool for the academic community. Determining syntactic relationships, though, not only requires judgment, which is necessarily subjective, but also depends on one's theory of grammar. To think that such a project can be accomplished *without* a theory would be like saying that exegesis can be performed without an explicit methodology or that interpretation can exist in a vacuum, without a hermeneutical theory. It is simply not scientific reality—even if an exegete or interpreter is unaware or ignorant of theories and methodologies, there is always a framework in which analysis occurs (however coherent that framework may or may not be).

And yet, although some of us have situated our research on

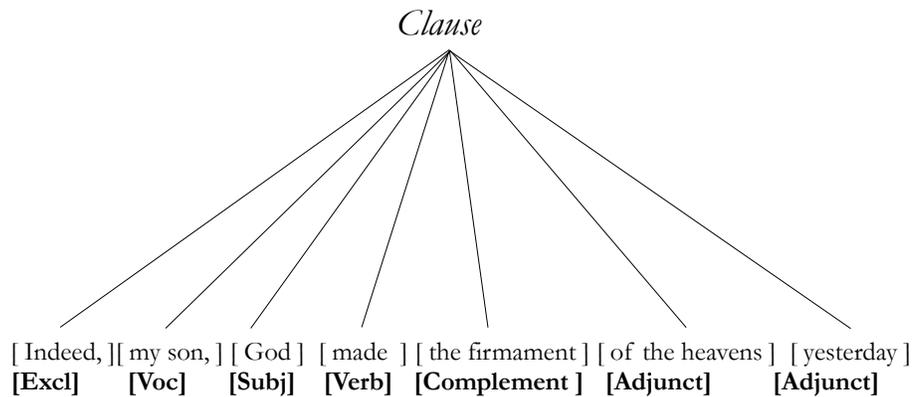
Hebrew syntax within the linguistic approach of generative grammar, specifically as it is articulated within the program of Chomskyan minimalism (Chomsky 1995; Radford 1997; Boeckx 2006, 2008), we knew that to base the database and its underlying tagging scheme on a fully articulated minimalist framework would be wildly inappropriate. Not only would its usability be severely limited, since it is unlikely that most users of the database will subscribe to Chomskyan linguistics, but given the ever-changing nature of linguistic theory, the database would become obsolete before it was finished.

To keep our balance on a very narrow beam, we sought to develop a tagging scheme that reflected what became our motto: “**data primary, theory wise.**” That is, while the project team has read broadly in linguistics, from various types of functionalism and typology to generative grammar, it was important for the project that the usability and accessibility dictated our use of linguistic theory. Three decisions will illustrate our balance beam act.

3.1. Hierarchical, Non-Binary Phrase Structure

There are two basic options for clause structure: a flat clause structure and a hierarchical clause structure. The flat clause structure is based on a finite state model, the ‘Markov Model’ (Malmkjaer 2002:138-39) in which it is argued that a clause is constructed word-by-word in a linear fashion; clauses in this model are also called ‘word chains’. In this model, which is often associated with computational linguistics, it is proposed

that the speaker has a simple mental system that allows him to make a decision about the appropriateness of each word as it is added to the clause-in-making and, when all the given words are added, the product is either accepted or rejected based on a final analysis. An example of a flat-structure clause is given here:



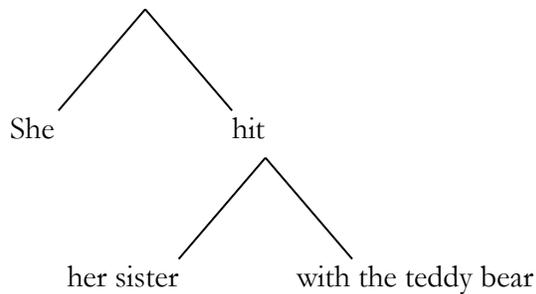
The central problem with this flat structure model of the clause is the inability to account for long-distance syntactic relationships, in which two syntactic elements that somehow depend on each other are separated by an arbitrary number of words. For example, in the first two examples below, the subject and verb are adjacent and so the subject-verb agreement is immediate, or ‘local’; in the third example, though, the agreement is non-local or long distant.

- (1) The [baby_{SG}] [cries_{SG}].
 The [babies_{PL}] [cry_{PL}].
 The [babies_{PL}] in the nursery [cry_{PL}].
-

In contrast to the flat structure, the hierarchical approach to clause

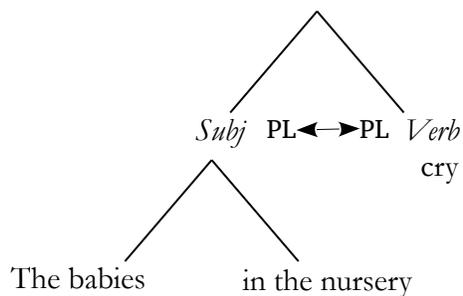
structure is not primarily linear but, as its name signals, hierarchical. The syntactic elements relate to each other in terms of how they ‘cluster’ together. For example, in the clause “*she hit her sister with the teddy bear*,” we might suggest that ‘she’ and ‘hit’ relate to each other non-hierarchically, as the two basic halves of the clause. But we would not put the rest of the clause on the same level: the words ‘her sister’, which seem to belong together, and the words ‘with the teddy bear’, which also seem to form a group, both seem to form a group with the verb ‘hit’. These hierarchical relationships are typically represented by brackets or trees:

(2) [She] [hit [her sister] [with the teddy bear]].



This hierarchical clause structure can also account for how long-distance dependencies exist, illustrated below:

(3) The [babies_{PL}] in the nursery [cry_{PL}].



In this example, the element ‘in the nursery’ is hierarchically dominated by ‘the babies’. This allows the plural ‘the babies’ to be hierarchically adjacent to the plural verb ‘cry’, thus providing an explanation for how the subject and verb may agree even though they are separated by other words.

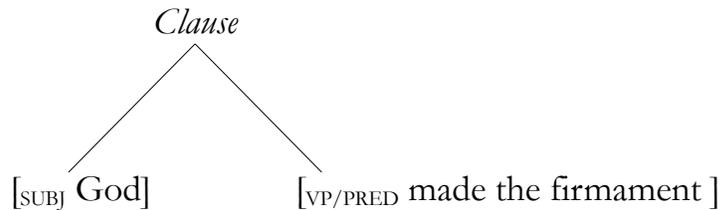
The process of formation is from the bottom-up, that is, as each lexical item is introduced into the ‘clause-in-the-making’ (called a ‘derivation’), the lexical items merge with each other and project a larger structure, a phrase. The lexical item that gives the phrase its syntactic identity is the phrasal head. Thus, a prepositional phrase is the projection of the hierarchy around a preposition, a noun phrase is the projection of a noun, a verb phrase the projection of a verb, etc.

The highest level constituent is a clause. A clause is a single constituent consisting of a subject phrase and a verb phrase. Main clauses (or ‘independent’) are self-contained and thus do not function within a larger *syntactic* hierarchy, while subordinate (or ‘dependent’) clauses are contained within a phrase, typically a verb phrase in a higher clause.

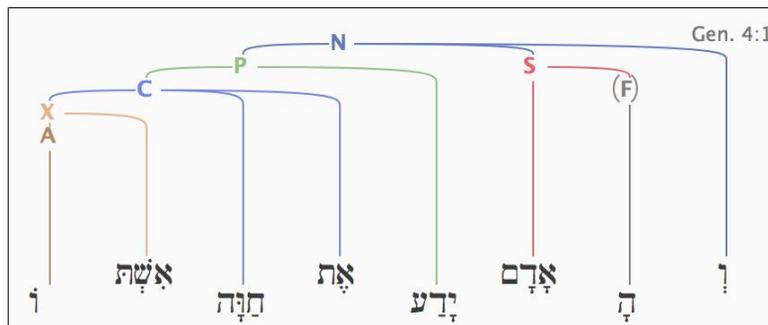
The point of this discussion of hierarchical clause structure has been to establish that we designed our database on a well-known linguistic theory of phrase structure, in which it is argued that constituents *are contained within* larger constituents, all the way up to the clause level. For each word, we and our tagging team have had to make a

decision regarding the word’s location in the syntactic hierarchy—within what other constituent does it reside? And for that resulting complex constituent, the same question must be answered, until there are no more constituents and one is left with a clause.

The clause itself seems to consist of **two basic parts**: a **subject phrase** (no matter how simple or complex) and a **verb phrase** (no matter how simple or complex). Note that we labeled the verb phrase as the **P(redicate) phrase** in the database. Thus, at a basic level the hierarchy that we have followed is binary in nature.



The syntactic “tree” diagrams in the Accordance syntax database account for both hierarchical levels in Hebrew syntax as well as the basic division of each clause into subject and predicate, as the diagram from the first clause of Gen 4:1 illustrates.



Notice that under the N node (which represents independent clauses),

there are two halves of the clause, the subject (S) and the predicate (P). The subject has a specifier (F), i.e., the article, and within the predicate there is the verb and its complement (C). The complement is a noun phrase that is itself modified by an apposition (X) phrase.

Searches within the Accordance syntax database should match the desired structure. For example, to construct a search for similar subject–verb clauses as in Gen 4:1, one could begin simply with a clause in which a subject and predicate (= verb) are inserted, as so:

The screenshot shows a search interface with three main sections. The top section is a single box labeled "ANY CLAUSE" with a green checkmark and "Depth = 0". The middle section is split into two boxes: "PREDICATE PHRASE" with a green checkmark and "Depth = 0", and "SUBJECT PHRASE" with a green checkmark and "Depth = 0". The bottom section is split into two columns. The left column contains "PREDICATE" (checked) and "VERB" (unchecked). The right column contains "NOUN" (unchecked).

Note that the Depth feature in the clause must be set to 0 and that we have included all the levels of the hierarchy we want to find—clause, phrase, and then word. The depth feature allows for the construction of tight searches (i.e., looking for syntactic structures just as they are represented in the database, as in the illustration above) or squishy searches (i.e., increasing the depth allows for broader searches that may catch relevant hits that are more syntactically complex, even as the search may catch irrelevant hits).

To further specify the search—for example, to find a subject-verb clause in which the verb has a noun phrase (NP) complement (as in Gen 4:1)—would look like so:

The screenshot shows a search interface with a hierarchical tree structure. At the top level, there is a search box labeled "Depth = 0" with a dropdown menu set to "ANY CLAUSE". Below this, the tree branches into two main categories: "Depth = 0 PREDICATE PHRASE" and "Depth = 0 SUBJECT PHRASE". The "PREDICATE PHRASE" branch further divides into "Depth = 0 COMPLEMENT PHRASE" and "Depth = 0 PREDICATE". The "COMPLEMENT PHRASE" branch is expanded to show "Depth = 0 COMPLEMENT" and "Depth = 0 NOUN". The "PREDICATE" branch is expanded to show "Depth = 0 VERB". The "SUBJECT PHRASE" branch is currently collapsed.

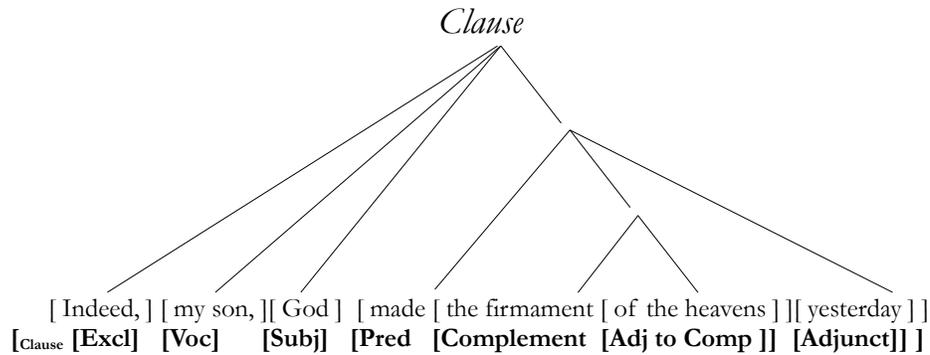
The desire to search for an NP complement requires the insertion of a complement phrase and then then complement+noun word level items (again, the structure must be strict if the depth level is set at 0).

3.1.1. Binary versus Non-Binary

Earlier forms of Chomskyan generative syntax allowed for “*n*-ary” branching (i.e., whatever number of branches appear to be required) and some non-Chomskyan generative frameworks maintain an *n*-ary principle of phrase structure. However, since the mid-1980s, the Government-and-Binding model, followed by minimalist syntax, adopted a strictly binary approach to constituent structure.⁴ But the

4. Though minimalist phrase structure is explicitly hierarchical and binary, and

addition of clause-edge constituents, such as dislocations (*casus pendens*), vocatives, and exclamatives results in a tree that is not easy to fit into a binary structure and to do so requires a good deal of theory-internal arguments.



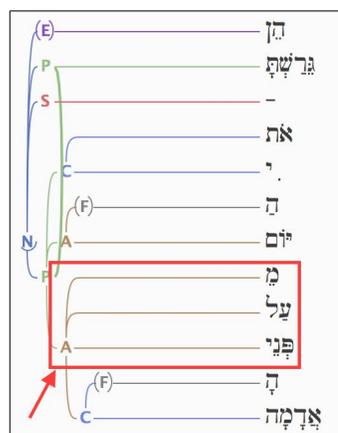
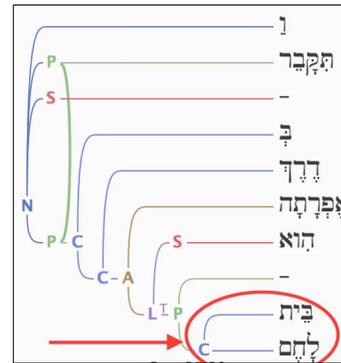
Thus, we made the decision to depart from a basic principle of this particular theory in favor of presenting hierarchical data in a manner that is not so theory dependent, even at the risk of analytical error. Here, data-presentation outweighed analytical preference. The non-binary structure of our database is apparent in clauses in which the verb has multiple complements, a complement and adjunct(s), or when “edge” constituents such as dislocations are present. Two such examples are provided below.

supports this position with data and an appeal to theoretical economy, there is certainly no consensus among linguists in general on whether the syntactic structure of human language reflects a flat or hierarchical structure, and if the latter, whether the hierarchy is fundamentally binary or “*n*-ary” (Culicover and Jackendoff 2005:112-16; Berg 2009:33-56, 325-8; Guevara 2007). Of course, empirical support and theoretical advantage are claimed by everyone who bothers to make an explicit defense of their phrase structure(s).

non-binary structure.

3.1.2. 'Constituents'

The syntactic elements at each stage of derivation are referred to as constituents. A constituent is a single syntactic unit that has a place within the hierarchy of a larger syntactic unit. It is important to recognize that morphological words and constituents *may* overlap but are not always identical. That is, a single word may represent more than one syntactic constituent, such as English *teacher's*, in which the constituent *teacher* has a syntactic role that is distinct from the syntactic role of the possessive *'s*. This is the case in Hebrew, too. Moreover, the converse is also true: occasionally multiple words form one syntactic constituent, as with many proper nouns, such as **בֵּית לֶחֶם**



Bethlehem 'House of Bread':

We also analyze complex prepositions similarly. Consider **מֵעַל פְּנֵי**, which is decomposable morphologically as 'from.upon the.face.of'. Syntactically we analyze this string of words as a single syntactic constituent, a preposition with the dominant meaning 'from'.

Constituents within a hierarchical clause structure approach stand in some tension to an analysis based on parts

of speech. Parts of speech are inadequate for syntactic analysis. Using the parts of speech labels typically used for Hebrew, some may suffice for syntactic description, so that *verb* and *adjective*, for example, may also describe the syntactic roles those words play; however, the other parts of speech labels, *noun*, *pronoun*, *preposition*, and the umbrella label *particle*, are wholly opaque concerning the syntactic relationships between these words and any others in a given clause. Therefore, syntacticians often use a different set of labels for the various constituents in a clause. The core labels are *subject*, *predicate* (or *verb*), *complement*, and *adjunct*, with the non-core constituents (in our database) *vocative*, *exclamative/interjection*, *parenthesis*, *appositive*, and *casus pendens*.

3.1.3. “Where’s the Direct Object?”

No doubt some will look through the short list of syntactic roles above and ask, “Where is the direct object? And what about the indirect object?” The answer is that they are not syntactic relationships that are explicitly tagged in our database. Why? The answer to that is more complex, but here is the beginning of an explanation.

The **complement** essentially corresponds to ‘object’, of which there are a number of sub-types. The direct object is the *accusative* (to borrow a case term), or a nominal (non-prepositional) constituent that is the person or thing undergoing the (active, transitive) verbal action or process, i.e., the ‘patient’. In contrast, the indirect object is limited to a small set of verbs that require a ‘recipient’ (or ‘beneficiary’) of the verbal

action or process to be specified.

There are two basic problems with encoding the concepts of direct and indirect object in a syntactic database, especially one for Hebrew. First, these concepts are not exclusively syntactic in nature; one must necessarily interact with argument structure (or thematic role) information concerning the predication, information that is explicitly outside the scope of our *syntactic* database (more on this below in §D). Second, whereas direct objects in English are always in the accusative case (i.e., non-prepositional), verbs in Hebrew (and Greek) are varied in their selection of a syntactic constituent as their object: some select a non-prepositional constituent, while others select some type of prepositional constituent. In sum, using complement allows us to capture a greater generalization: regardless of the type of constituent—non-prepositional, prepositional, or even clausal. In contrast to the complements, which are required to “complete” a verb’s syntactic-semantic features, **adjuncts** are those constituents which are optional.

3.2. Non-Movement Approach to Constituent Discontinuity

Constituent movement is a hallmark of transformational generative grammar (Brown 2010), although it has been dismissed by much non-Chomskyan generative theory (i.e., ‘monostratal’ theories). The basic idea is that the linear order of constituents in many actual clauses cannot reflect the ‘original’ order of those constituents. Neither defending nor criticizing this proposal, we determined that *representing* it in our database

was not desirable or necessary. Yet, we were forced to deal with the results of constituent movement, that is, discontinuous constituents—constituents that are divided into parts separated by other, unrelated constituents. This happens less in English than in Hebrew, although it does occur with some English relative clauses, as below in (4):

(4) [A new king] arose over Egypt, [who had not known Joseph]



The relative clause, ‘who had not known Joseph’, clearly modifies the NP ‘a new king’, and yet it is separated from this NP by the VP ‘arose over Egypt’.

In Hebrew, discontinuity is extremely common, since many narrative clauses begin with the *wayyiqtol* narrative verb, switch to a subject, and then continue with the rest of the predicate, as in (5).

(5) וַיִּרְא אֱלֹהִים אֶת־הָאֹר
and-saw God DOM⁵-the-light



‘and God saw the light’ (Gen 1:4)

The challenge of constituent discontinuity is that, based on the hierarchy and the projection principle that a phrase contains all its complements and/or adjuncts, a verb and its modifiers together make up a *single*

5. We take אֶת to be a “differential object marker”; see Bekins 2014.

constituent. But how, then, can this be represented when they are broken by non-related intervening constituents, such as a subject?

To account for discontinuous constituents we employ a system of cross-referencing, which allows us both to include discontinuous constituents in syntactic searches and display the connection in the tree display. We have used this cross-referencing system to allow us to represent more accurately three additional phenomena: dislocation (*casus pendens*), resumption in relative clauses, and ellipsis (or ‘gapping’). For a description of the cross-referencing, with examples, see below, §4. For how to deal with discontinuity in searching, see above §2.6.

3.3. Inclusion of Null Constituents

The third illustrative interaction with linguistic theory in our database production is the recognition of null constituents.⁶ On the principle that every phrase has a ‘head’, whether a ‘verb’ for a Predicate or a noun or similar nominal(ized) constituent for a Subject, we have inserted a null marker (0) in every phrase that lacks an overt head.

The use of null constituents is most common in the Subject position, since Hebrew allows an overt subject to be omitted, as in example (6), and nearly as common in Hebrew is the use of a null copula in the Predicate position, the so-called verbless clause, as in (7):

וַיֵּשְׁבֶתּוּ _____ בַּיּוֹם הַשְּׁבִיעִי מִכָּל-מְלֶאכֶתוֹ (6)

6. On null constituents in Hebrew, see Creason 1991; Naudé 1991, 2001; Holmstedt 2013.

‘and **(he)** rested on the seventh day from all his work’
(Gen 2:2)

(7) עַל־פְּנֵי תְהוֹם — וַחֹשֶׁךְ
‘and darkness **(was)** upon the face of the deep’ (Gen
1:2)

In addition to null subjects and predicates, Hebrew also allows null complements and null relative clause heads. All of these null items have been included and tagged appropriately in our databases. For more detail on our use of null constituents, see below §II.

3.4. The Narrow Syntactic Focus of the Database

A final defining principle of the Accordance syntax database that we will mention here is a *narrow focus* on syntax. That is, the tagging scheme provides phrasal, clausal, and inter-clausal information to the exclusion of semantic judgments, discourse relationships, and implicational pragmatics. For example, when the particle **כִּי** is a subordinator, we make no distinction between its use as a temporal (‘when’) subordinator or a clausal (‘because’) subordinator. Those distinctions are left to the user to determine. What we provide is the distinction between **כִּי** as an adjunct subordinator (temporal or causal), a complement subordinator (‘that’), a conjunction (‘but’), and an exclamative (‘indeed!’).

What we do include is verbal valency information, which we associate with the lexical entry of a verb (see Cook 2014 and

forthcoming). The term valency derives from chemistry and has been employed in linguistics for about a half-century. Verbal valency, in particular, refers to the property of a verb that determines the syntactic environments in which it may appear. For example, the English verb *snored* requires a subject, *help* requires both a subject and an NP complement and *returned* requires a subject and prepositional (locative) complement:

- She *snored*.
- He *helped* the boy.
- They *returned* to the house.

For the database project, it was necessary that we use valency information to determine whether the non-subject constituents associated with a given verb were complements or adjuncts (for more on complements and adjuncts, see below §IV). And yet, we do not identify these complements or adjuncts by any semantic categories, such as locative, temporal, means, manner, etc. Moreover, we do not include any discourse-pragmatic judgments, such as whether a complement preceding a verb has a Topic or Focus function.⁷

7. A theoretical issue that has absolutely nothing to do with the narrow focus of our project is the “autonomy of syntax” debate (Cheng 2007). From the project’s perspective, we take an agnostic stance with regard to this debate. Whether or not semantic and pragmatic information is allowed to directly affect syntax or whether they are formulated as functional features and categories that operate within syntax seems to be an irrelevant theoretical argument when it comes to the goals of our project (however interesting it may be in general).

This decision on the narrow focus of our database was made for two practical reasons:

1. every additional layer adds an increasing amount of subjectivity, and we want this research tool to be as broadly usable as possible.
2. the additional semantic and pragmatic layers would add a disproportionate number of years to the project. Whereas we have completed all our ancient Hebrew texts in eight years, it would likely take twice that time (if not much more) to produce a multi-layered database.

4. Appendix: Cross-Referencing

Cross-referencing is used in only three kinds of constructions in the Accordance Hebrew syntax database.

4.1. Dislocations (*‘Casus Pendens’*)

‘Casus pendens’, Latin for ‘hanging case’, (also sometimes ‘nominative absolute’) refers to the construction in many European morphological case languages in which a constituent stands at the front edge of a clause and is ‘resumed’ within the clause by a different constituent in a different case. Since ancient Hebrew does not exhibit morphological case, this term is descriptively inaccurate. The term current in linguistics is *dislocation*, which aptly describes the relationship between the initial noun and the clause juxtaposed to it. Dislocation of a constituent at the front of the clause (“left dislocation”) is illustrated in (8).⁸

- (8) מְקַנְהֵם וְקַנְיָנֵם וְכָל־בְּהֵמָתָם הֲלוֹא לָנוּ הֵם
‘[their cattle and their property and all their beasts]₁ —
are they₁ not ours?’ (Gen 34:23)

In (8) the initial compound NP מְקַנְהֵם וְקַנְיָנֵם וְכָל־בְּהֵמָתָם cannot be a

8. “Left” and “right” refer to spatial orientation from the perspective of a left-to-right writing system. Though languages use a variety of directional writing systems (left-to-right, right-to-left, boustrophedon, top-to-bottom, etc.), it is critical to have a single accepted convention for cross-linguistic analysis and description, and since modern linguistics has its origin in Western European language traditions, the currently accepted terminological conventions reflect a left (front of the clause) – to – right (end of the clause) orientation.

formal syntactic part of the null copula clause, which already has a subject, **הוּא**, and a complement of the null copula, **לִּנְי**. Moreover, the initial NP is separated from the clause proper by the interrogative **הֲ**, which has a fixed position at the normal front edge of a clause. The pronoun **הוּא** within the clause refers back (i.e., it is “anaphoric”) to the syntactically ‘hanging’ or ‘dislocated’ NP at the front edge of the clause. Constituents can also be dislocated at the end of the clause; this is referred to as “right dislocation.” Because the anaphoric pronoun resuming (or anticipating, in the case of right dislocation) a dislocated constituent is the only formal connection between the core clause and the dislocated constituent, the two are numerically cross-referenced in the Accordance Hebrew syntax database.

4.2. Relative Clause Resumption

Cross-referencing is also used in the Accordance Hebrew syntax database to connect the antecedent/‘head’ of a relative clause with the empty position (9) or resumptive constituent (10) within the relative that marks the head’s role within the relative clause.

- (9) **וְשָׁתִית מֵאֲשֶׁר יִשְׂאֲבוּן הַנְּעָרִים**
‘and (you) shall drink \emptyset (some)₁ of (the water) that the
lads draw ____₁’ (Ruth 2:9)

- (10) **וַתֵּצֵא מִן־הַמָּקוֹם אֲשֶׁר הֵיטָהָהּ שָׁמָּה**
‘and she went out from the place₁ that (she) had lived
there₁’ (Ruth 1:7)

Notice that in both (9) and (10) the ‘gap’ or resumptive constituent was in a required complement position within the relative clause. This is the most common syntax for אֲשֶׁר and שֶׁ relative clauses, but it is not uncommon for the head to be cross-referenced with the subject position within the relative (11).

- (11) הָבִי הַמְטַפַּחַת אֲשֶׁר-עָלֶיךָ
‘take the cloak₁ that ____₁ is on you’ (Ruth 3:15)

In fact, this is the only cross-referencing allowed in ה-relatives, which are constrained to relativizing from the subject position (12).

- (12) אֶת־יְהוָה אֱלֹהֶיךָ הַמוֹצִיאֲךָ מֵאֶרֶץ מִצְרַיִם מִבֵּית עֲבָדִים
‘...Yhwh₁, your God, who ____₁ brought you out from
the land of Egypt, from slavery’ (Deut 8:14)

Another example of a ה-relative is instructive: this example contains a transitive verb, making the relative analysis unambiguous (13).

- (13) וַתָּשָׁב נָעֲמִי וְרוּת הַמוֹאֲבִיָּה כִלְתָּהּ עִמָּה הַשָּׁבָה מִשְׂדֵי מוֹאָב
‘and No‘omi₁ returned (and Ruth, the Moabitess, her
daughter-in-law was with her), who ____₁ returned
from Moab’ (Ruth 1:22)

It is worth noting that in (13) both the predicate and the subject נָעֲמִי are discontinuous. In particular, note that the subject is interrupted by the adjunct (comitative, null predicate) clause וְרוּת הַמוֹאֲבִיָּה כִלְתָּהּ עִמָּה. The constituent נָעֲמִי is resumed with the ה-relative, which explains the

subject discontinuity/resumption.

Finally, there are relatives in which the head has come out of an adjunct position within the relative clause. Note that these constructions are the only ones in which a null adjunct is entered into the tagged text (14).

- (14) __ עֵינֶיךָ בְּשָׂדֵה אֲשֶׁר־יִקְצְרוּן
'(let) your eyes be on the field₁ that they harvest
__(there)₁' (Ruth 2:9)

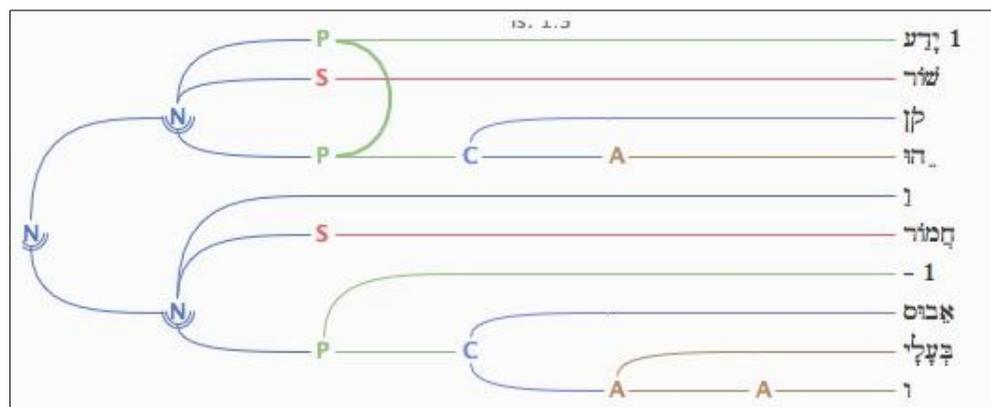
The justification for antecedent-anaphora cross-referencing with relative clauses is simple: without it there would be no way to search for relative clauses that, for example, have overt resumptive constituents versus gaps.

4.3. Ellipsis ('Gapping')

The third and final phenomenon for which cross-referencing is included in the database is ellipsis, also known as gapping. This occurs most commonly in poetry, where, say, a poetic line-pair interact in such a way that one constituent is stated in one line and assumed in the other. This is a very common feature of poetic parallelism. However, ellipsis is also used to a lesser extent in prosaic texts, as in (15), including narrative (16)-(17).

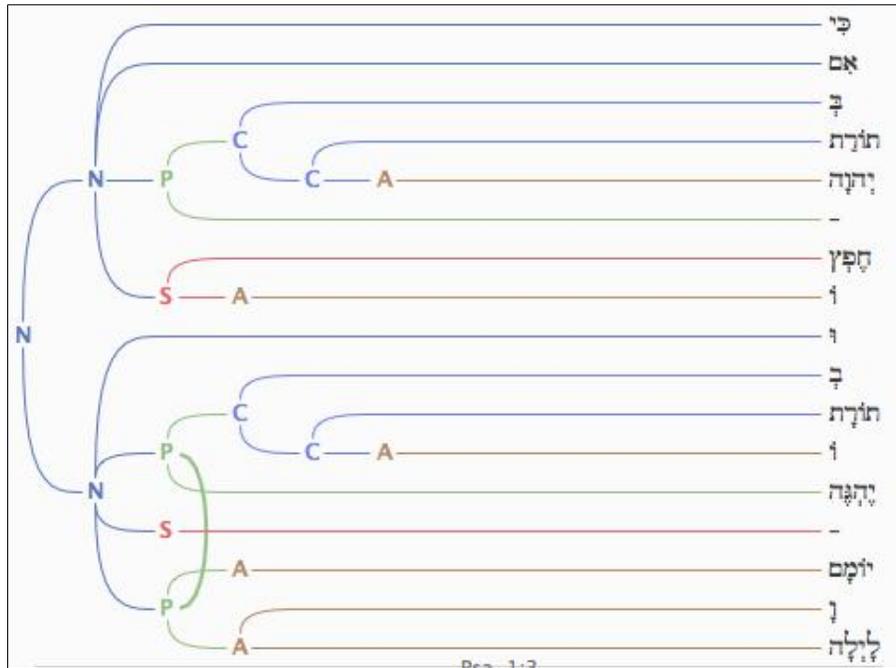
- (15) יִדַע שׁוֹר קִנְהוֹ וְחִמּוֹר __ אָבוֹס בְּעֵלְיוֹ

‘the ox knows₁ its owner and the ass ___(knows)₁ its master’s trough’ (Isa 1:3)



In (15) the first line of the pair has the verb **יָדַע**, which is then assumed in the second line. In the second line, the null copy of **יָדַע** is cross-referenced with the numeral 1, in order to tie it to **יָדַע** and thus distinguish it from typical null verb clauses, which are copular (i.e., “be” clauses).

Note that in the Accordance syntax database, we connect poetic bicola and tricola by containing them within one overarching clause while at the same time providing each colon with its own clausal structure. This can be seen in the syntax diagrams by the N (clauses) nested within the larger N, as in the diagram of Ps 1:2.



Both (16) and (17) present complex examples of verbal ellipsis.

(16) שְׁבְנָה אִשָּׁה ____ לְבֵית אִמָּהּ

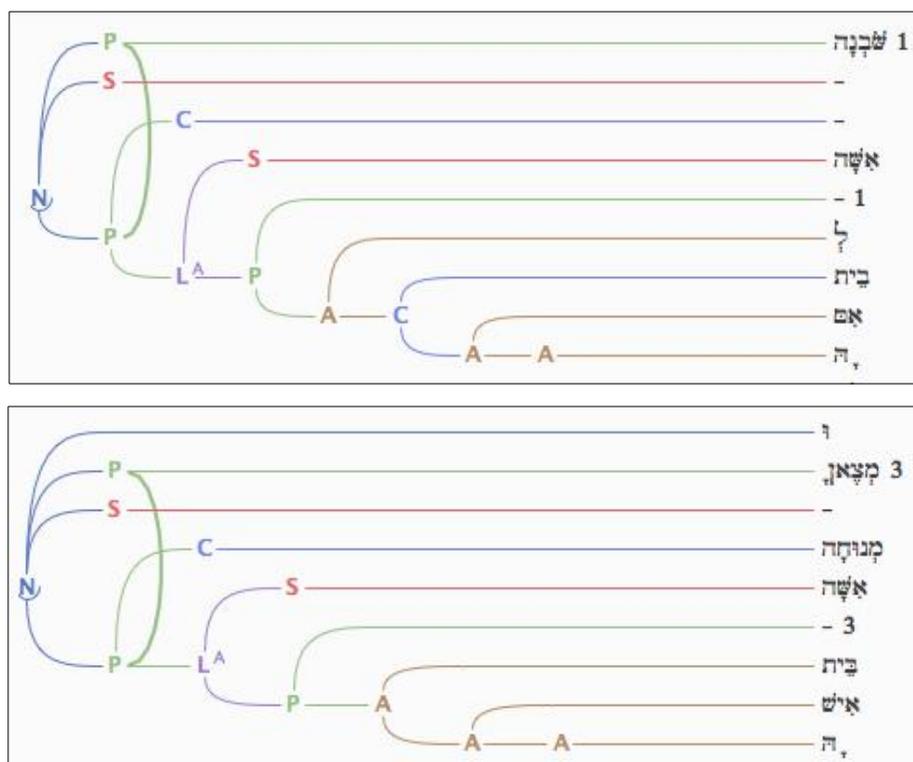
‘return₁, each ____ (returning)₁ to the house of her mother’ (Ruth 1:8)

(17) וּמְצָאֵן מְנוּחָה אִשָּׁה ____ בֵּית אִישָׁהּ

‘and find₃ rest, each ____ (finding)₃ the house of her [next] husband’ (Ruth 1:9)

In most narrative cases, a null verb will represent the copular verb, a null version of הָיָה. However, in some cases (most in poetry) a null verb will represent a copy of a verb in the previous line, i.e., it has been gapped.

That is the case in (16) and (17), in which the absent finite verbs represent elided copies of **שָׁבַח** and **מָצְאוֹ**, respectively, and so have a cross-reference to indicate that the null non-finite verb (represented by the absence of any element) is a partial semantic copy of the higher verb.

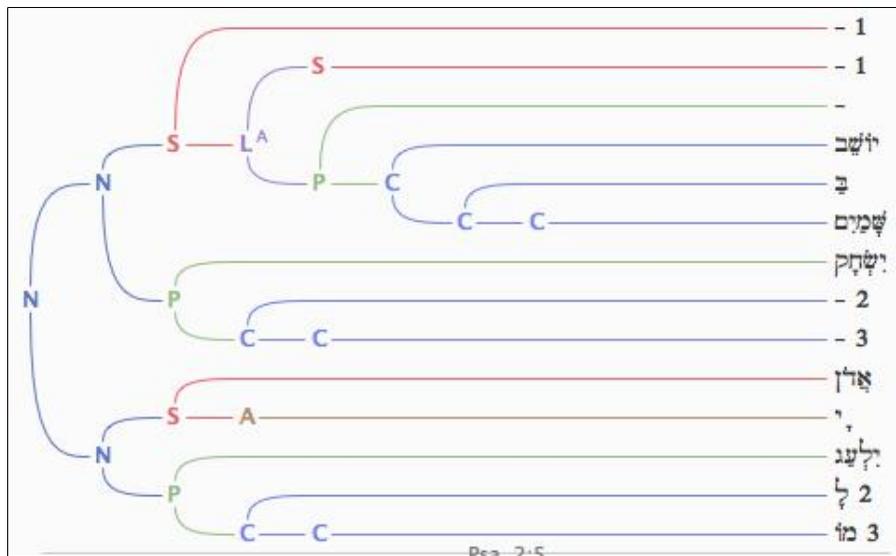


Finally, note the multi-item ellipsis in (18):

(18) **יָוֹשֵׁב בַּשָּׁמַיִם יִשְׁחַק אֶדְנִי יִלְעַגְלָמוֹ**

‘(The one₁) (who) __ (he)₁ is dwelling in the heavens
 laughs __ (at)₂ __ (them)₃; the Lord jeers _{2at} _{3them}’
 (Ps 2:4)

In (18) we see backwards ellipsis of the preposition and its pronominal complement of the parallel verbs שחק/לעג (so Miller 2007). In the Accordance syntax database, this is signaled by cross-referencing each element individually: the preposition in (18) uses [2]] and the pronoun uses [3]. Note the representation in the syntax diagram:



5. Sample Searches

The sample searches below are intended to represent the variety of syntax searching that the Accordance Hebrew syntax database is capable of. The sample search workspaces are numbered according to the list of searches below. These searches are provided separately as saved workspaces that can be downloaded and opened directly by Accordance by downloading them from the site below:

<https://www.accordancebible.com/Documentation>

5.1. Null Predicate (=“Verbless”) Clause

This search is the simple way to search for “verbless clauses,” or what we refer to as Null Copula clauses.

The screenshot displays the Accordance Hebrew Bible software interface. The main window shows a search for "Null Predicate (Null Copula = "Verbless" Clause)". The search results are displayed in a list format, with each entry showing the Hebrew text, the corresponding English translation, and a syntax diagram. The syntax diagrams use various colors and letters (N, P, C, S, A, F) to represent different syntactic elements. The sidebar on the right contains a list of search filters and options, including "Suff.", "Pron.", "ANY", "Verb", "LEX", "INFLECT", "ROOT", "CHAR.", "HITS", "COUNT", "PLACE", "Negating Item", "NOT", "Syntax", "CLAUSE", "PHRASE", "Subject", "Predicate", "Complement", "Adjunct", "Appositive", "Vocative", "CaususPend.", "Specifier", "Exclamation", "Unknown", "Null", "Antecedent", and "BeginSpeech".

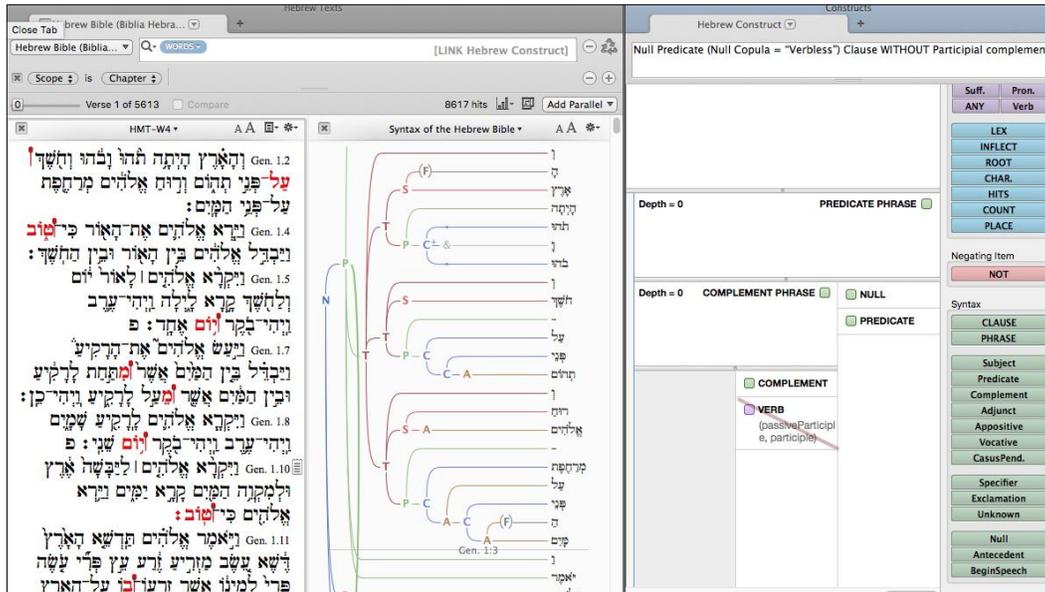
Note that the null item is indicated in the Hebrew text by the red “flag” or “pointer.”

This search returns what for most users will be both expected and unexpected results. The expected results conform to typical definitions of “verbless” clauses, i.e., a clause with a Subject and Predicate Complement (also known in classical grammar as a “predicate nominative”) but no overt predicate/verb.

The unexpected results concern **Hebrew participial clauses**. The Hebrew participle is notorious for looking like a noun (it is inflected like an adjective, with masculine-feminine and singular-plural morphological marking) and yet acting like a verb (participles often take Complements like verbs do, and even have Adjuncts). Recent research into the nature of the Hebrew participle strongly suggests that *they are best understood as adjectives*, regardless of the other features (see Cook 2008). Thus, in this database, participles are consistently tagged as the complements of a copular predicate, which is mostly null although there are cases with the overt copular Hebrew verb **הָיָה**.

5.2. Null Predicate Without Participial Complement

To find all the null copula (=“verbless”) clauses without the participial complements, that is, all the expected null copula clauses, one must use a phrase structure search in which a Null Predicate is followed by a Complement that *excludes* Participles (active and passive).



This search accurately finds all null copula clauses that do not have a Participial Complement. Note that the position of the “missing” copula is marked by the red flag and the complement is indicated by red text (a user-definable preference). It may help to understand the structure of Null copula clauses: they consist of a Subject, a Null copula, and a Complement. The Subject may be Null like the copula, but there is always an overt Complement. The result in such cases is a copular clause with only one overt constituent (a “one-part verbless clauses”), which are illustrated in the next search.

5.3. One-Part Null Subject, Null Predicate Clause

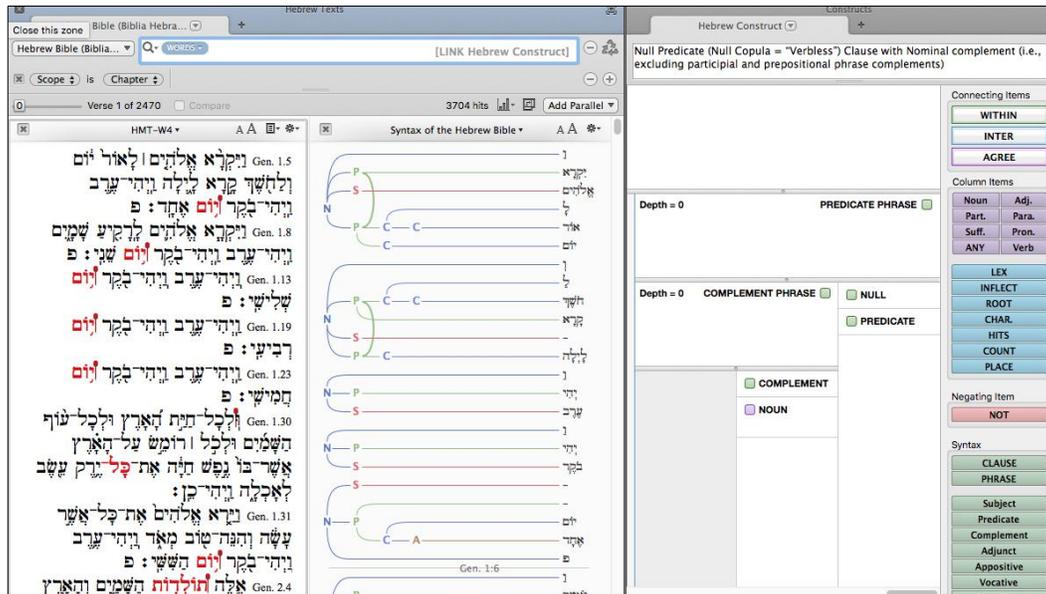
While most Null copula clauses have an overt Subject and Complement,

there are a number that have a Null Subject, thus leaving only one constituent, the Complement, to represent the clause.

The screenshot displays the Accordance Hebrew Syntax Database interface. On the left, a list of Hebrew Bible verses is shown with syntax diagrams. The diagrams use color-coded lines to connect words to their syntactic functions: N (Null Subject), P (Predicate), S (Subject), C (Complement), and L (Link). A red flag is placed over the missing copula in the diagrams. The right pane shows a detailed view of a 'One-Part Null Copula (= "Verbless") Clause, i.e., the subject is null and only the complement is overt'. The diagram shows a 'Depth = 1 ANY CLAUSE' containing a 'Depth = 0 PREDICATE PHRASE' which includes a 'Depth = 0 COMPLEMENT PHRASE' containing a 'COMPLEMENT' which is a 'VERB (passiveParticiple, participle)'. The right sidebar contains various filters and options for the search results.

In the search above, Participial Complements are excluded simply to highlight the prototypical one-part null copula clause, as in Gen 1:4, 10. Again, note that the position of the “missing” copula is marked by the red flag and the complement is indicated by red text (a user-definable preference). These examples illustrate that the great majority of these null Subject, null Predicate clauses are inside dependent clauses, such as **כִּי** or **אֲשֶׁר** clauses. Cases of independent one-part null copula clauses are in poetry or in lists or similar constructions and are often translated as sentence fragments, such as “(they) (are) 14 cities and their villages” (Josh 15:36) or “(it) (is) an announcement of Yhwh” (Joel 2:12).

5.4. Null Predicate Clause with a Nominal Complement



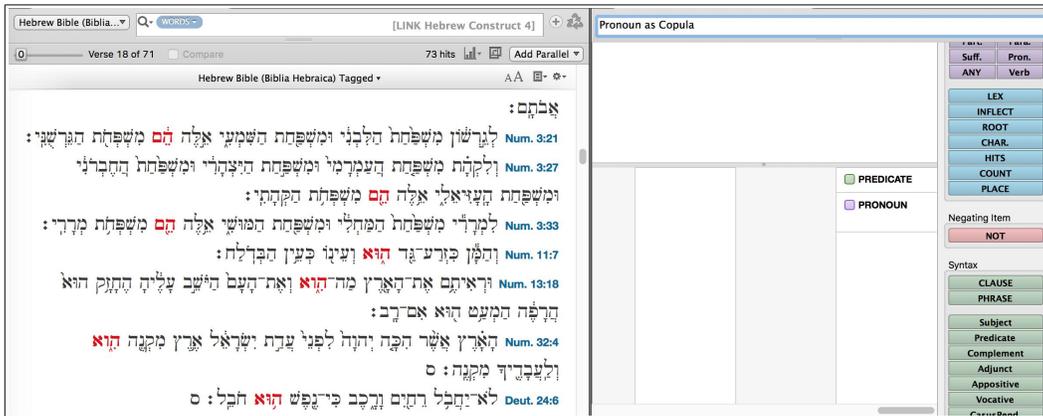
To find null copula clauses that have only nouns as their Complements (not prepositional phrases or participial phrases), simply specify the Complement as a Noun.

Similar searches such as these last four could be composed for null subjects and null adjuncts, and even null complements, by themselves or combined with a variety of other items or restrictions.

5.5. Pronominal Copula

Finding all cases of the 3rd person pronoun used as a copula in the syntax database is simple and straightforward. One need only insert the “Pronoun” morphological label and combine it with the “Predicate”

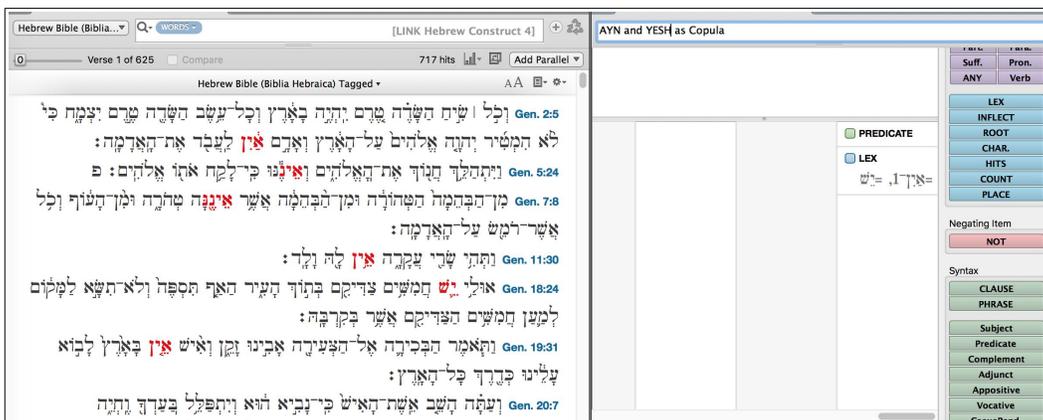
label, as below.



For examples of the demonstratives used as copular items, simply insert Adjective>Demonstrative into the column instead of the Pronoun item.

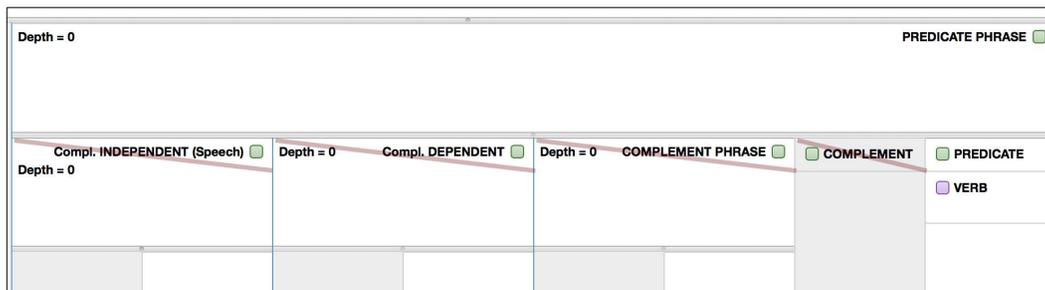
5.6. Copular *אין* and *יש*

Similarly, finding all cases of *אין* or *יש* that are judged to be copular is simple — insert the Predicate label along with the lexical items *אין* and *יש*, as so:



5.7. Monovalent Verbs

To find monovalent verbs, or even the rare aivalent verb (that is, a verb that has no subject and no complements), it is necessary to use the NOT function and negate the presence of any complements (or, in the case of the aivalent verb, any complement or subject). Below is a search for monovalent verbs.



Note that since a complement can have a number of different syntactic manifestations (all of which are treated differently in Accordance's search construct), to find a verb that has no complements requires negating four different types of complements in a row—a word, phrase, and clausal complement, as well as direct speech complements.

5.8. Trivalent Verb **נָתַן** with NP Complement and PP Complement

When looking for types of verbs, it is possible to combine any feature of verbal morphology with the syntactic features. For instance, to find cases of the trivalent verb **נָתַן** followed first by an NP complement and then by a PP complement, the appropriate search would be created as below:

The screenshot displays the Accordance Hebrew Syntax Database interface. The left pane shows a list of Hebrew Bible verses with their corresponding syntax tree diagrams. The right pane shows the search results for the query 'GIVE + NP Comp + PP Comp'. The search results are organized into a table with columns for 'Depth = 0', 'COMPLEMENT PHRASE', and 'PREDICATE PHRASE'. The search results show various Hebrew words and their corresponding syntax tree diagrams. The interface includes a search bar, a list of search results, and a detailed view of the search results.

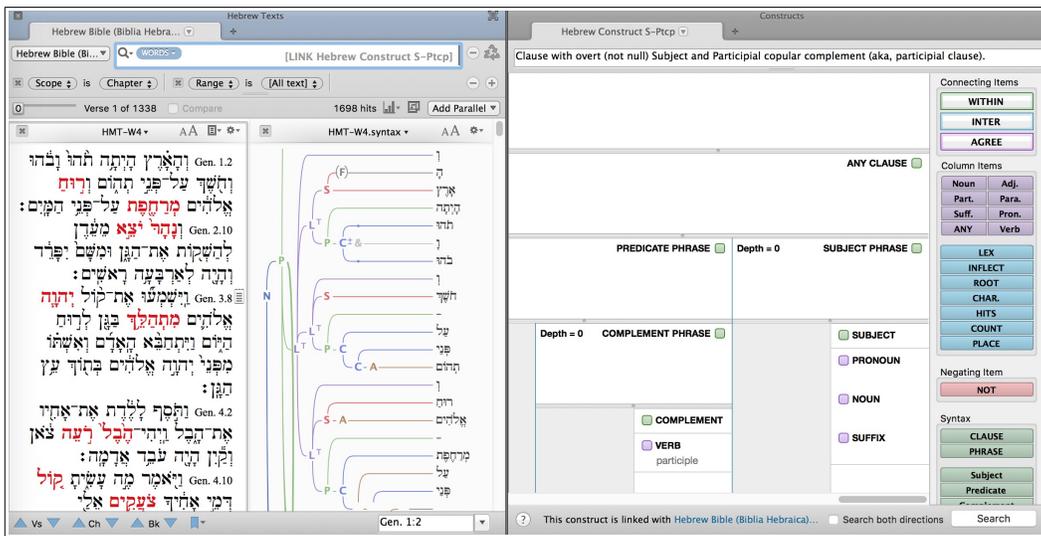
5.9. Infinitive “Absolute” used Verbally

Cases of the infinitive absolute used “verbally” can be found with a dependent clause search in which the Predicate Phrase contains a Predicate label is combined with a Verb that is specified as an Infinitive Absolute.

The screenshot displays the Accordance Hebrew Syntax Database interface. The left pane shows a list of Hebrew Bible verses with their corresponding syntax tree diagrams. The right pane shows the search results for the query 'Infinitive Absolute as Verb'. The search results are organized into a table with columns for 'Depth = 0', 'DEPENDENT', and 'PREDICATE PHRASE'. The search results show various Hebrew words and their corresponding syntax tree diagrams. The interface includes a search bar, a list of search results, and a detailed view of the search results.

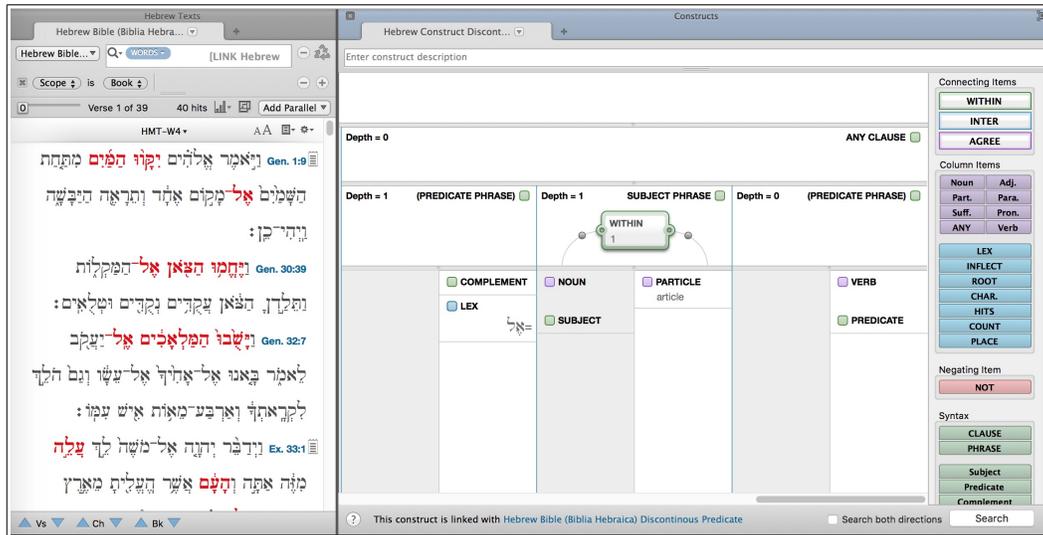
5.10. Subject-Participle Clause

As we indicated in the discussion of Search #5.1, we take participles to be the complement of a (typically null) copula. The following search illustrates how to look for “participial clauses” with an overt subject (i.e., a subject that is an NP, free pronoun, or cliticized pronoun).



5.11. Discontinuous Predicate

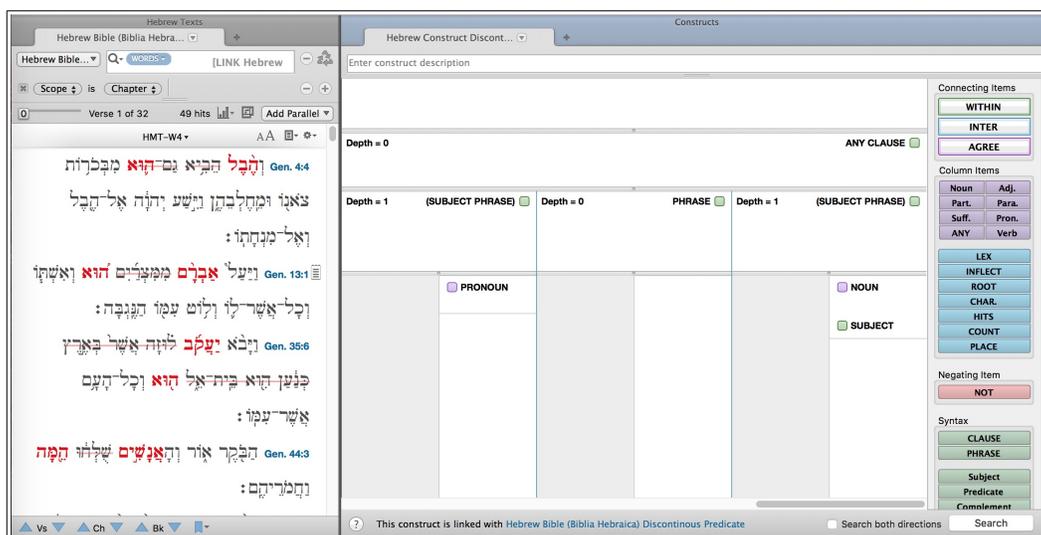
To look specifically for predicates that are split by an intervening subject, it is necessary to choose “One segment” within the “Phrase structure” option when inserting the Predicate phrase. This allows the insertion of another part of the Predicate phrase (also “One segment”) to specify the remaining components of the desired clause. Consider the search below:



In the example above, the desired clause has an overt verb followed by a subject NP that has a definite article, which is followed by the rest of the predicate (in this case, we specified a complement beginning with the preposition **אֶל**).

5.12. Discontinuous Subject

Just as predicates can be split into discontinuous parts, so also can subjects, complements, and even adjuncts. The search below illustrates how to set up a search for a discontinuous subject.



This search looked for a subject constituent that was separated from the remaining part of the constituent by any type of other phrase. The intervening phrase could be a predicate, complement, or adjunct. In this particular example, we added a pronoun into the second half of the subject constituent, to look for cases in which a subject NP was modified by a phrase beginning with a matching free pronoun, as in Gen 13:1, וַיֵּעַל אַבְרָם מִמִּצְרַיִם הוּא וְאִשְׁתּוֹ וְכָל-אֲשֶׁר-לּוֹ, וְאַבְרָם and Abram came up from Egypt, he and his wife and all that belonged to him’.

5.13. Compound Subjects

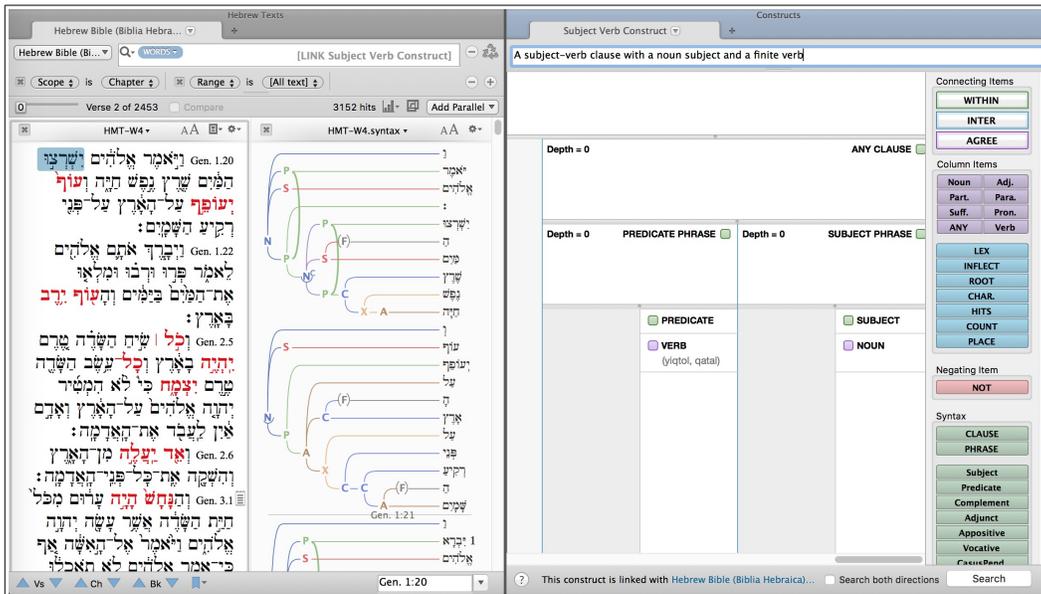
Compound constituents are those in which two or more items at the same syntactic level work together to form a single syntactic constituent. They are often coordinated by means of the ׀, but not always. The search below illustrates a simple compound Subject search, specifying

that the compound subject include a pronoun.

While the search above finds any compound subject with a pronoun (regardless of the number of parts of the compound), the variation below using a subject phrase finds only two part compounds of the type pronoun – noun.

5.14. Subject-Verb Clause

The first search below is set up as a simple subject-verb search.



To find the few cases of a subject preceding a *wayyiqtol*, simply change the VERB item to *wayyiqtol*. Similarly, the second search below finds all cases of subjects followed by verbs but with a ׀ intervening between the two.

The Accordance Hebrew Syntax Database

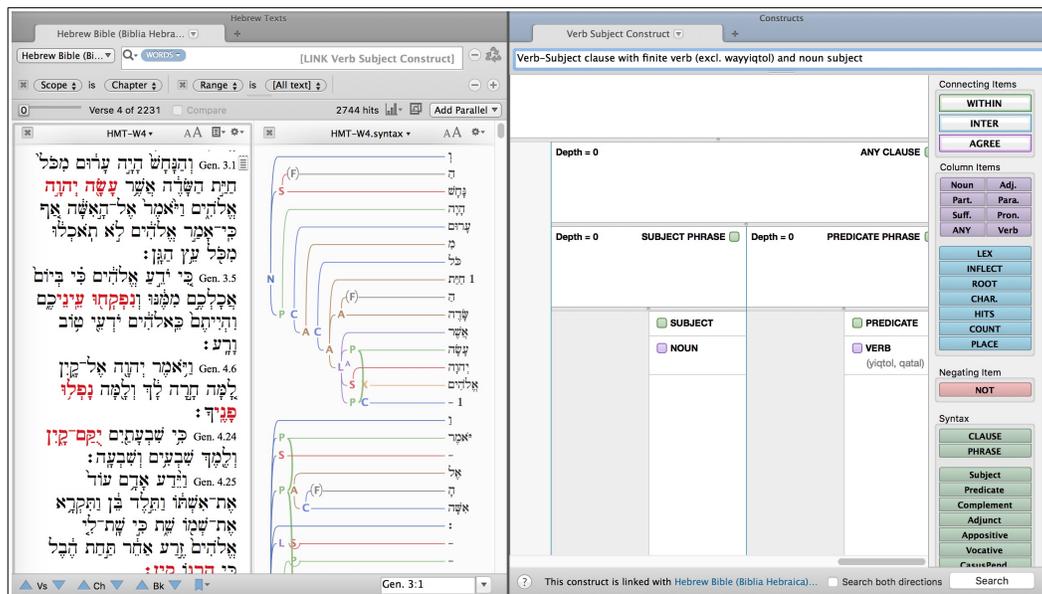
The screenshot displays the Accordance Hebrew Syntax Database interface. On the left, the Hebrew Bible text is shown with search results for the query "[LINK Subject Waw Verb Construct]". The results list verses such as Ex. 4.9, Ex. 30.38, Lev. 13.18, Lev. 13.58, Lev. 15.11, Lev. 17.15, Lev. 22.6, Lev. 23.29, Num. 5.30, and Num. 21.8, with the relevant words highlighted in red. On the right, the "Constructs" panel shows a hierarchical tree for the "Subject Waw Verb Construct". The tree starts with "ANY CLAUSE" at Depth 0, which branches into "PREDICATE PHRASE" (Depth 1) and "SUBJECT PHRASE" (Depth 0). The "PREDICATE PHRASE" further branches into "VERB" (with tags [yiqtol, qatal]) and "PLACE 1". The "SUBJECT PHRASE" branches into "NOUN". A "WITHIN 1" box is shown connecting the verb and the subject phrase. On the far right, there are various control panels including "Connecting Items" (WITHIN, INTER, AGREE), "Column Items" (Noun, Adj., Part., Para., Suff., Pron., ANY, Verb), "Syntax" (CLAUSE, PHRASE), and "Negating Item" (NOT).

The third search provides a template for a full clause structure, subject (with adjunct) – verb (with complement and adjunct). Simply add lexical or morphology tags or delete phrases as desired.

The screenshot displays the Accordance Hebrew Syntax Database interface for a "Full Clause Template" search. The left panel shows Hebrew text with search results for Gen. 6:20, Gen. 34:5, and Gen. 34:7, with relevant words highlighted in red. The right panel shows a hierarchical tree for the "Full Clause Template, with subject phrase modified by adjunct and predicate phrase including both complement and adjunct phrases". The tree starts with "ANY CLAUSE" at Depth 0, which branches into "PREDICATE PHRASE" (Depth 0) and "SUBJECT PHRASE" (Depth 0). The "PREDICATE PHRASE" branches into "COMPLEMENT PHRASE" (Depth 0) and "PREDICATE" (Depth 0). The "SUBJECT PHRASE" branches into "ADJUNCT PHRASE" (Depth 0) and "SUBJECT" (Depth 0). On the far right, there are various control panels including "Connecting Items" (WITHIN, INTER, AGREE), "Column Items" (Noun, Adj., Part., Para., Suff., Pron., ANY, Verb), "Syntax" (CLAUSE, PHRASE), and "Negating Item" (NOT).

5.15. Verb-Subject Clause

To perform a basic search for clauses with verb-subject order (excluding *wayyiqtol*), set it up like so:



For searches that specify, for example, that the predicate has a complement or adjunct, use the model of the discontinuous predicate illustrated in search #5.11.

5.16. Adjuncts

Adjuncts fall into two general categories: those that modify nouns (adjectives, relative clauses, PPs) and those that modify verbs. The two searches below illustrate each kind. The first search looks for NPs that are used to modify a verb, that is, adverbial NPs (sometimes called the “adverbial accusative”).

The Accordance Hebrew Syntax Database

The second search finds PPs that modify a noun, that is, NP-internal PPs. Typically PPs modifying verbs, but when they are within an NP, they function as a simpler version of a relative clause.

5.17. Apposition

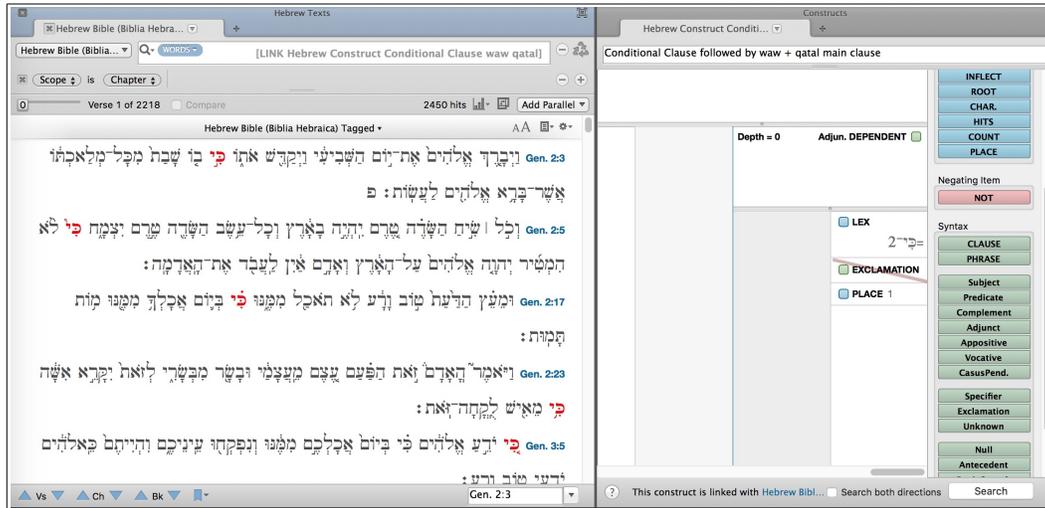
Note that in the Accordance syntax database, the X label is used for apposition because the appositional elements almost always share the same categorial (e.g., N, V) identity as the item they modify. Thus, the X signals a lack of separate identity while also allowing a very simple mechanism for finding appositives in the database. For most cases of apposition, using the simple APPOSITIVE label will suffice.

However, if one is looking for the apposition of particular types of phrases or clauses, it is necessary to use the “appositive” option within the phrase and clause entry boxes. The search below finds cases in which the appositive is an entire clause.

The screenshot displays the Accordance Hebrew Syntax Database interface. The search bar at the top contains the query "[LINK Hebrew Construct Appositive clause]". The search results show 379 hits. The main window displays a list of Hebrew verses with their corresponding syntax trees. The selected verse is Gen. 15:4, which reads: "וְהָיָה דְבַר יְהוָה אֵלָיו לֵאמֹר לֹא יִירָשְׁךָ יְהוָה כִּי אִם אֲשֶׁר יִצְאָ מִמֶּעֶיךָ הוּא יִירָשְׁךָ". The syntax tree for this verse shows a main clause (S) with a subject (N) and a predicate (A). The subject is a clause (S) that is appositional to the main subject. The appositional clause is also a main clause (S) with its own subject (N) and predicate (A). The interface also shows a search filter for "apposition of a clause" and a list of syntax categories including LEX, INFLECT, ROOT, CHAR, HITS, COUNT, PLACE, Negating Item (NOT), and various syntax types like CLAUSE, PHRASE, Subject, Predicate, Complement, Adjunct, Appositive, Vocative, CasusPend., Specifier, Exclamation, and Unknown.

5.18. Subordinate Clauses

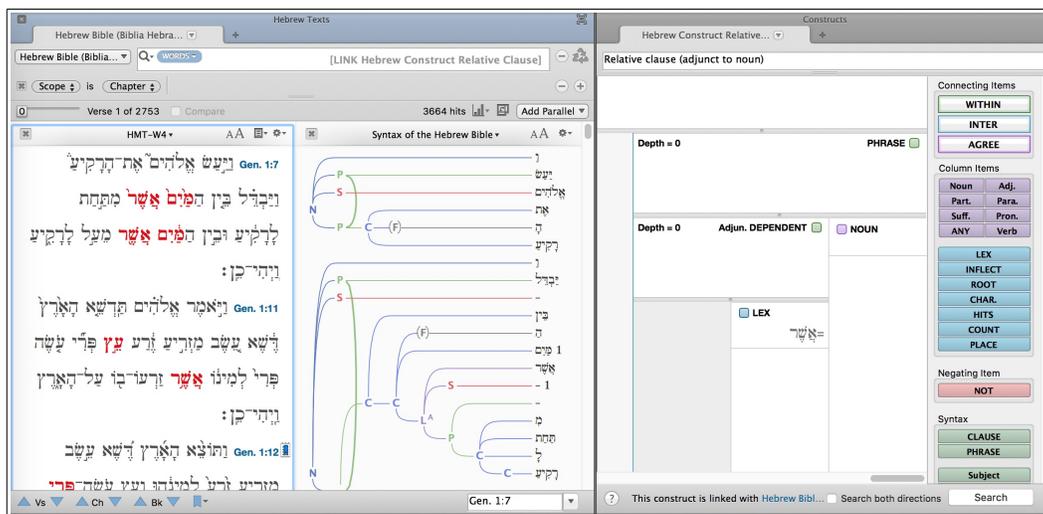
Subordinate clauses are clauses that are adjuncts within a larger constituents. Most types of subordinate clauses are adjuncts to a higher verb, as with the **כִּי** clauses found by the search below.



Note how the search is restricted to Adjunct Dependent clauses. There are examples of **כִּי** complement clauses, which could be found by changing the clause type. There are also independent clauses with a **כִּי** in the first position, as a disjunctive conjunction (“but”) or exclamative (“indeed!”). Note also that exclamative function of **כִּי** can be isolated or excluded by using the Exclamation label. And finally, the above search employs the PLACE function. Setting it =1 for this search isolates all **כִּי** adjunct clauses in which the **כִּי** is the first item (versus, for example, cases in which **כִּי** is part of a compound conjunction like **כִּי אֲדָּ** or **אֲפֹס כִּי**).

The one subordinate clause that is an adjunct not to a verb but to

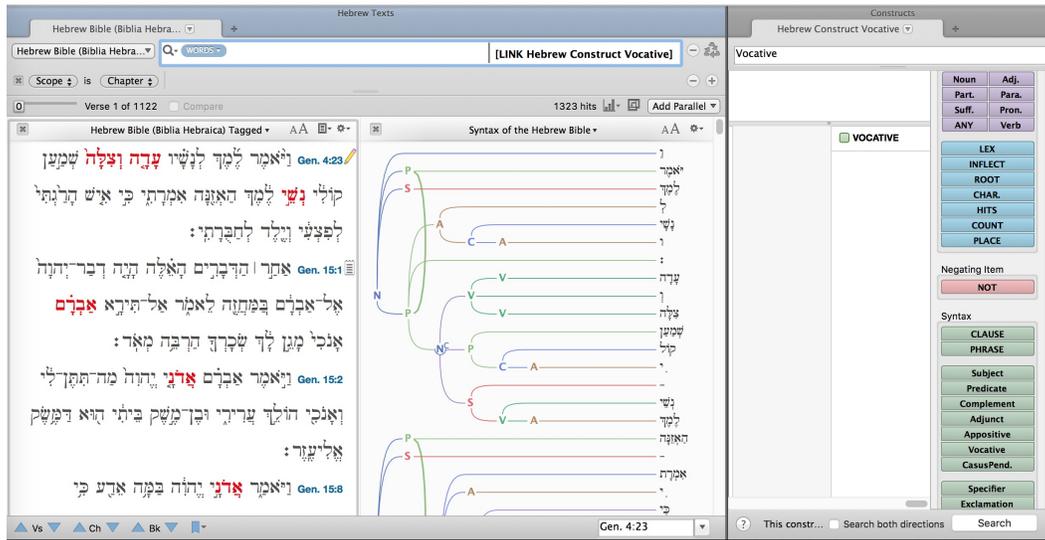
a noun is the relative clause. A search for a simple relative clause with an overt noun (not null) head is illustrated below.



5.19. Vocatives

Vocative elements are integrated into a clause in two fundamental ways. The first pattern, when a vocative follows its referential anchor (shown below) is essentially appositional. The second pattern is when the vocative does not follow an obvious anchor (and in some cases, such an anchor does not exist in the clause). Both are illustrated in the hits for a simple Vocative search.

The Accordance Hebrew Syntax Database

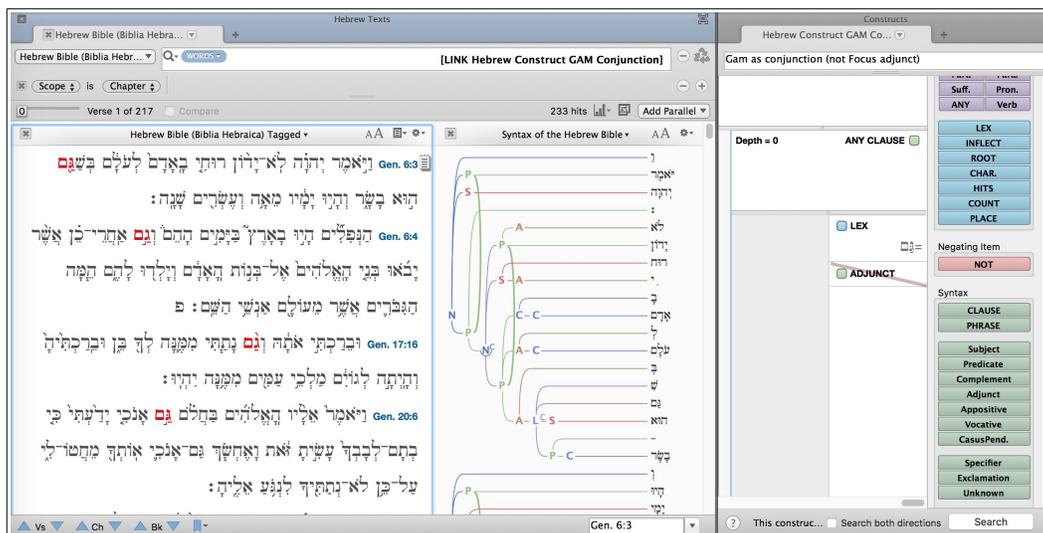


Note that in the first few examples, the vocative constituent follows either an overt or null anchor. In contrast, the vocative in Gen 15:2 does not have an anchor—it is the first constituent in Abram’s direct speech address to God.

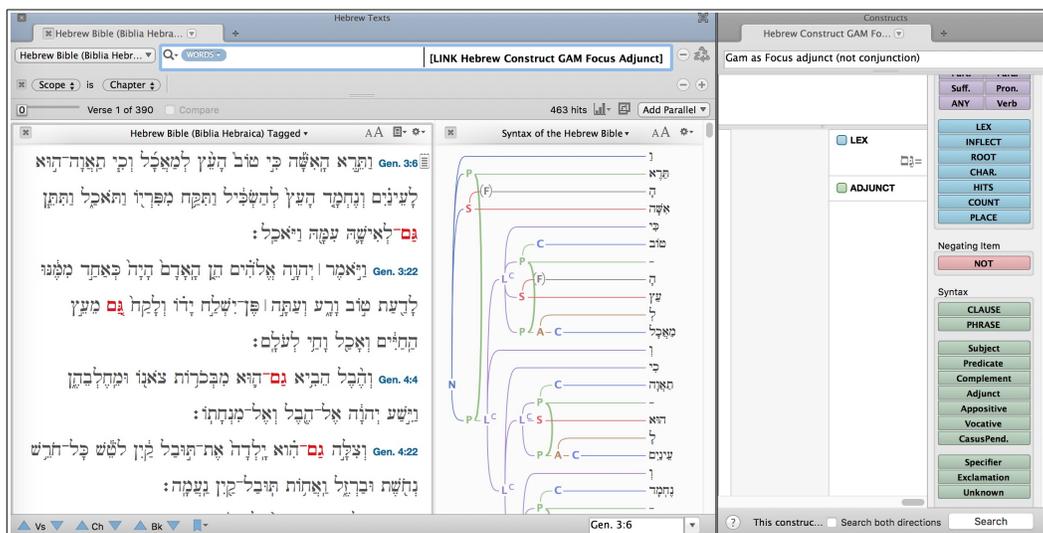
5.20. Conjunctions versus Focus Words

Some functions words in Hebrew, such as **וְ**, have both a conjunction use and a Focus word use. For multivalent functions like this (and **כִּי**, **אֲשֶׁר**, etc.), the syntax database distinguishes between their conjunctive use and whatever other use they have (e.g., as a Function word, as an Exclamative). The search below is set up to find the conjunction examples of these words, using **וְ** to illustrate.

The Accordance Hebrew Syntax Database

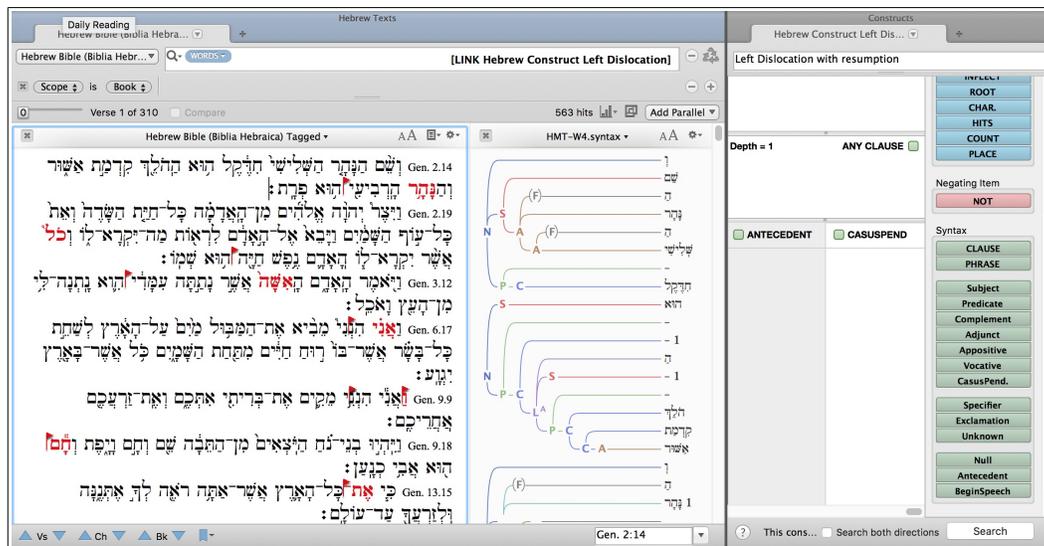


To find exclamative **כִּי**, simply insert EXCLAMATION and the lexical item **כִּי**. For the others, which are mostly Focus words, the database tags them as adjuncts. Thus, to find non-conjunctive uses of **וְ** requires a simple search like so:



5.21. Dislocation

Dislocation is the typically used linguistic term for the case-based term *casus pendens* ‘hanging case’. Dislocations are easily found in the syntax database with the CASUSPEN item. Moreover, a defining feature of dislocations is that they are linked to the clause by a coreferential constituent. In left dislocation (that is, a dislocated element at the front of a clause), this linkage occurs as a resumptive or anaphoric pronoun. In right dislocation (that is, a dislocated element at the end of a clause), the linkage occurs as a presumptive or cataphoric pronoun. The search below finds both left dislocations and their resumptive pronouns.



5.22. Ellipsis/Gapping

Ellipsis, or gapping, as it is commonly understood for Hebrew poetic syntax refers to the absence of a grammatically required constituent, the

content of which can be reconstructed by means of an overt constituent in a parallel poetic line filling the same syntactic function as the gapped word. A simple English example is “He drove the truck, she ___ the car,” where the verb “drove” is gapped or elided—it is not overt but must be understood—in the second clause. Ellipsis as studied concerning Hebrew combines the allowance of null constituents with the phenomenon of parallelism.

To find examples of ellipsis thus requires a search that incorporates both items: parallelism (two clauses within a single umbrella clause) and a null constituent that has an antecedent in the parallel line. The search below illustrates how to search for examples of verbal ellipsis—cases in which an overt verb in the first poetic line is elided or gapped in the second poetic line.

The screenshot displays the Accordance Hebrew Syntax Database interface. On the left, the 'Hebrew Bible (Biblia Hebraica)' search window is active, showing a search for '[LINK Verb Gapping]' with 410 hits. The results window shows three examples of verb gapping in Hebrew, each with a corresponding syntactic tree diagram. The examples are from Psalms 9:10, 16:2, and 17:13. The right-hand side of the interface shows the 'Constructs' window, which is used to define the search criteria. The search criteria include 'Verb Gapping' and 'Constructs'. The search results are displayed in a table with columns for 'Depth', 'ANY CLAUSE', 'PREDICATE PHRASE', 'NULL', 'PREDICATE', 'ANTECEDENT', and 'VERB'. The search results show that the search found 410 hits, with 1 hit for 'Verb Gapping' and 409 hits for 'Constructs'. The search results are displayed in a table with columns for 'Depth', 'ANY CLAUSE', 'PREDICATE PHRASE', 'NULL', 'PREDICATE', 'ANTECEDENT', and 'VERB'. The search results show that the search found 410 hits, with 1 hit for 'Verb Gapping' and 409 hits for 'Constructs'.

REFERENCES

- Andersen, Francis I., and A. Dean Forbes. 2003. Hebrew Grammar Visualized: I. Syntax. *Ancient Near Eastern Studies* 40:43–61.
- . 2012. *Biblical Hebrew Grammar Visualized*. LSAWS 6. Winona Lake, IN: Eisenbrauns.
- Bekins, Peter J. 2014. *Transitivity and Object Marking in Biblical Hebrew*. HSS 64. Winona Lake, IN: Eisenbrauns.
- Berg, Thomas. 2009. *Structure in Language: A Dynamic Perspective*. Routledge Studies in Linguistics. New York: Routledge.
- Boeckx, Cedric. 2006. *Linguistic Minimalism: Origins, Concepts, Methods, and Aims*. Oxford: Oxford University Press.
- . 2008. *Bare Syntax*. Oxford: Oxford University Press.
- Brown, Keith. 2010. Generative Grammar. Pp. 181-200 in *The Linguistics Encyclopedia*, 3rd ed; ed. Kirsten Malmkjaer. London: Routledge.
- Cheng, Lisa Lai-Shen. 2007. The Autonomy of Syntax. Pp. 209-26 in *Knowledge in Ferment: Dilemmas in Science, Scholarship and Society*, ed. Adriaan in 't Groen, Henk Jan de Jonge, Eduard Klasen, Hilje Papma, and Piet van Slooten. Leiden: Leiden University Press.
- Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.
- Cook, John A. 2008. The Hebrew Participle and Stative in Typological Perspective. *Journal of Northwest Semitic Languages* 34 (1):1-19.
- . 2014. Verbal Valency in Biblical Hebrew and the Case of מִלֵּא. Paper presented at the annual meeting of the Society of Biblical Literature, San Diego, CA, November 24.
- . Forthcoming. Verbal Valency: The Intersection of Syntax and Semantics. In *Malta and Chicago 2012 Colloquia*, ed. Alison Salveson and Tim Lewis. Perspectives in Linguistics and Ancient Languages 5. Piscataway, NJ: Gorgias.
- Creason, S. 1991. Discourse Constraints on Null Complements in Biblical Hebrew. *University of Chicago Working Papers in Linguistics* 7:18-47.
- Culicover, Peter W. and Ray Jackendoff. 2005. *Simpler Syntax*. Oxford: Oxford University Press.
- Crystal, David. 2008. *A Dictionary of Linguistics and Phonetics*. 6th ed. Oxford: Blackwell.
- Guevara, E. 2007. Binary Branching and Linguistic Theory: Morphological Arguments. Pp. 93-106 in *Proceedings of the XXXII Incontro di Grammatica Generativa, Firenze, 2–4 March 2006*, ed. M. C. Picchi and A. Pona. Alessandria: Edizioni dell'Orso.
- Holmstedt, Robert D. 2013. Pro-drop. Pp. 265-67 in *Encyclopedia of Hebrew Language*

The Accordance Hebrew Syntax Database

- and Linguistics, Volume 3: P-Z*, ed. Geoffrey Khan. Boston/Leiden: Brill.
- Kroeze, J. H. 2002. Developing a Multi-Level Analysis of Jonah Using HTML. Pp. 653-62 in *Bible and Computer: The Stellenbosch AIBI-6 Conference. Proceedings of the Association internationale Bible et informatique, "From Alpha to Byte," University of Stellenbosch, 17-21 July, 2000*, ed. J. Cook. Leiden: Brill.
- . 2006. Building and Displaying a Biblical Hebrew Linguistics Data Cube Using XML. In *Online Proceedings of ISCOL 2006, Haifa*. 25 pp. [Accessed 26/9/2009, http://mila.cs.technion.ac.il/english/events/ISCOL2006/ISCOL20060629_KroezeJH_XML_Paper.pdf]
- . 2008. Developing an XML-Based, Exploitable Linguistic Database of the Hebrew Text of Gen. 1:1-2:3. Unpublished Ph.D. thesis, University of Praetoria.
- Lowery, K. E. 2009. Hebrew Syntax: A Guide to the Westminster Hebrew Syntax Database. Unpublished ms. [Accessed 01/09/2009; <http://whi.wts.edu/blogs/kirk/wp-content/uploads/2009/01/guide2009-01-22.pdf>]
- Malmkjaer, Kirsten. 2002. Finite State (Markov Process) Grammar. Pp. 138-39 in *The Linguistics Encyclopedia*, 2nd ed; ed. K. Malmkjaer. London: Routledge.
- Miller, Cynthia L. 2007. Constraints on Ellipsis in Biblical Hebrew. Pp. 165-80 in *Studies in Comparative Semitic and Afroasiatic Linguistics Presented to Gene B. Gragg*, ed. Cynthia L. Miller. Chicago: The Oriental Institute of the University of Chicago.
- Naudé, Jacobus A. 1991. Qumran Hebrew as a Null Subject Language. *South African Journal of Linguistics* 9 (4):119-125.
- . 2001. The Distribution of Independent Personal Pronouns in Qumran Hebrew. *Journal of Northwest Semitic Languages* 27 (2):91-112.
- Petersen, U. 2004. Emdros — A Text Database Engine for Analyzed or Annotated Text. Pp. 1190-93 in *Proceedings of the 20th International Conference on Computational Linguistics: August 23-27, 2004, Geneva, Switzerland, Vol 2*. Morristown, New Jersey: Association for Computational Linguistics. [Accessed 26/9/2009, <http://emdros.org/petersen-emdros-COLING-2004.pdf>]
- Radford, A. 1997. *Syntactic Theory and the Structure of English: A Minimalist Approach*. Cambridge Textbooks in Linguistics. Cambridge: Cambridge University Press.