At the 2010 Society of Biblical Literature meeting in Atlanta, there was a special session devoted to discussing the new syntactic databases available in the Accordance Bible software. As the primary architect of the syntactic tagging scheme, I gave a paper outlining the issues of principles that I, my collaborator Prof. Martin Abegg (Trinity Western), and the chief programmer Roy Brown (OakTree Software) had to sort out.

(The other three presenters were Prof. John Cook (Asbury Theological Seminary), on the verbal valency issues we faced in the project and how we sort them out, Dr. Brown on the types of syntactic searches that users of Accordance can now perform, and Prof. Abegg on the type of issues in Qumran Hebrew our database can help to solve -- and yes, the overall project includes all ancient Hebrew, from epigraphic through Qumran).

As you might guess, there are numerous complexities involved with such a project. They range from issues of audience to theory (not to speak of the programming complexities, since they are beyond my ken). 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 require judgment, which is necessarily subjective, but also depend 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 happen 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 I have situated my particular research on Hebrew syntax within generative grammar, specifically as it is articulated within the program of Chomskyan minimalism, 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 I (and others on our team) have 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.

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’) 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:
Indeed, my son, God made the firmament of the heavens yesterday.

In my opinion, 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.

• 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 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 relationship are typically represented by brackets or trees:
[She ] [hit [her sister] [with the teddy bear]].

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

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.

**Binary versus Non-Binary**
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). Thus, at a basic level the hierarchy that we have followed is binary in nature.

Binary-branching is a basic principle to the minimalist program of Chomskyan generative linguistics, as well as many other generative frameworks. But the 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 (see here). Here, data-presentation outweighed analytical preference.

‘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 true in Hebrew, too; moreover, the converse is also true: occasionally multiple words represent a single syntactic constituent. This is the case with many proper nouns, such as לֶחֶם בֵּית Bethlehem ‘House of Bread’, but also true of complex prepositions, such as פְּנ”י מֵעַל, which is decomposable morphologically as ‘from.upon the.face.of’ but syntactically is taken as a single syntactic constituent ‘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, suffix, 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, and appositive.

"Where’s the Direct Object?"

R. D. Holmstedt, Nov 2010, page 6
No doubt some of you are looking through the short list of syntactic roles above and asking yourselves, “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 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 a ways below). Second, whereas direct objects in English are always in the Accusative (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 — the ‘object’ of the verb is labeled a C(omplement).

2. Non-Movement Approach to Constituent Discontinuity

Constituent movement is a hallmark of transformational generative grammar, 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
discontinuous constituents, that is, constituents that are divided into parts separated by un-related constituents. This happens less in English than in Hebrew, although it does occur with some English relative clauses, as below:

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

In this relative clause 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.

and saw God obj - 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 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, where discontinuity is signaled by lighter colored connecting lines, as with the tree for Gen 1:4:
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’).

3. Inclusion of Null/Covert Constituents
The third illustrative interaction with linguistic theory in our database production is the recognition of null constituents. 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 the first example below, and nearly as common in Hebrew is the use of a null copula in the Predicate position, the so-called verbless clause, as in the second example:

מִכָּלמְלַאכְתּוֹ הַשְּׁבִיעִי בַּיּוֹם וְיִּשְׁבֹּת __ __ ‘and (he) rested on the seventh day from all his work’ (Gen 2:2)

תַּהוֹם __ עלפְּנֵי עַד __ ‘and darkness upon the face of 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.

4. Final Comment: The Narrow Syntactic Focus of our Database
A final defining principle of the Accordance syntax database that I'll 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 כי 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 כי 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. 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, in the examples below 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.

(John Cook, the co-owner of this blog, presented an overview of the valency issues involved at the SBL session, and indeed I am indebted to his work for the last paragraph.)

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. 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.

But let me be absolutely clear: this decision on the narrow focus of our database was made for two practical reasons:

- First, every additional layer adds an increasing amount of subjectivity, and we want this research tool to be as broadly usable as possible.
- Second, the additional semantic and pragmatic layers would add a disproportionate number of years to the project. Whereas we are

R. D. Holmstedt, Nov 2010, page 10
confident that we will finish all our ancient Hebrew texts in the next 2-3 years, it would likely take a decade (or more) to produce a multi-layered database.

A theoretical issue that has absolutely nothing to do with the narrow focus of our project is the “autonomy of syntax” debate (see also here). 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 direct 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).

In future posts, I will begin describing how to use the syntax database within Accordance, a sort of user's manual-in-the-making.