The Logic of Searching the Andersen-Forbes Phrase Markers

A. Dean Forbes
16 July 2008

This document assumes that its readers have a working knowledge of Hebrew syntax.

NOTE WELL: The counts reported here are based on corrected data to which only we presently have access. The counts that you find should be close to those reported here but may not be identical.

After providing some orienting suggestions, this paper takes up four searches at three levels of difficulty: two basic, one intermediate, and one advanced.

Initial Observations:

1. Carrying out a non-trivial search in the phrase markers ("PMs") making up the Hebrew Bible is not an activity analogous to creating jazz. Sitting down and just "trying something" is a recipe for confusion and uncertainty.
2. This paper deals with the *logic of search*. The mechanical aspects of searching can be picked up by studying and mastering the contents of my paper "The Mechanics of Searching the Andersen-Forbes Phrase Markers." Also of help are the videos on the Logos website (http://www.logos.com/videos; scroll to "Syntax.").
3. A full search requires careful planning, execution, and documentation.
4. The operative principle is "decomposition," whereby a complex search is analyzed into a collection of many small, and one hopes, relatively simple sub-searches.
5. Do not assume that you can think of all the possibilities before doing the searches! You almost certainly cannot…After all, you are doing the searches to learn new things about the language. Hence, you likely will need to refine your searches iteratively.

Preparations:

1. It is assumed that readers of this white paper have ready access to the Logos Research Systems rendering of the Andersen-Forbes phrase markers (AFPMs). (Version 3.0e or later should work nicely.)
2. It is further assumed that they have gone through Part I of my companion paper "The Mechanics of Searching the Andersen-Forbes Phrase Markers" and created an *AFresources* workspace, or the like.
3. Users should call up our "A Systematic Glossary to the Andersen-Forbes Analysis of the Hebrew Bible" (found via "My Library": Find "Andersen") and become familiar with the "Basic Terms and Concepts" which initiate said document.
4. They should next scan down to the beginning of "Part III. Syntax Terms").
5. The reader should become *very familiar* with the terminology used to describe phrase markers: node, edge, in-degree, *et cetera*. A working knowledge of these concepts is vital to proper search.
6. Then, toggling between the glossary and a few selected phrase markers, the reader should exercise the glossary concepts until they are second nature.

7. For example, examine the PM for Jonah 2:2:
   a. How many nodes does the PM have?[1]
   b. How many edges?[2]
   c. How many clause immediate constituents (CICs)?[3]
   d. How many leaves?[4]
   e. How many segments?[5]
   f. How many root nodes?[6]
   g. What is the in-degree ("parent count") of the root node?[7]
   h. What is the out-degree ("child count") of the root node?[8]
   i. What structural form does the **mvt orig** CIC have?[9]
8. Note that the typical PM is "a multi-generation single-parent family." That is: it is rare for a node to have more than one mother.

Example #1 — Basic Searches for Nominalized Clauses:

The following shows how searches exploiting out-degree constraints can yield important results.

1. To see the two contexts in which nominalized clauses occur, take a look at Jer 23:6b. The final main clause in this verse contains two nominalized clauses, each involving a **nom / nom** *phrasal* node. The first has an overt nominalizer (אֲשֶׁר, 'which'), while the second does not.
2. Our strategy is to find how many nominalized clauses there are overall and then reckon how many are overtly marked by a preposed nominalizer and how many lack a nominalizer.
3. In the syntax search dialog left-hand panel, click on "phr" to install "Phrase 1".[10]
4. The phrasal node characteristics are next constrained in these two ways:
   a. In the right-hand panel, scroll down to the "Phrase" section and in the "Substantival" sub-section click "Nominalized Clause."
   b. Next, scroll down to the "Syntactic License" section and click "Nominalized Clause."[11] The left hand pane of the "Syntax Search" window will contain this very simple search specification:

Phrase 1: Phrase = Nominalized Clause AND License = Nominalized Clause

5. Carrying out the search that we have just specified, we find that the total number of **nom / nom** nodes in the PMs is 6,535.

---

[1] Twenty-three.
[2] Twenty-two.
[3] Five.
[4] Eleven.
[5] Eleven.
[6] One.
[7] Zero.
[8] Five.
[9] This is specified, as is always the case with CICs, by the label on its daughter node: **from + body prt**.
[10] The numerical indices attached to the nodes in the search specification are only for identification, are arbitrary.
[11] *Note well*: Specification of the licensing relation is critical since otherwise the search returns nom nodes involving the union relation, the inverse modification relation, *et cetera*.

6. How many of these are overtly marked? To find these, we further constrain the search by specifying in addition that the child count of the sought node is *two*, corresponding to the presence of an overt nominalizer plus the nominalized clause. We impose this constraint by calling up the syntax search afresh. It should still hold the specifications for the previous search. Scrolling to the very bottom of the right-hand panel, we encounter the "Parent Child Counts" section. We change "Child Count: Anything   " to "Child Count: = 2" and start the search. Now the search specification will look like this:

Phrase 1: Phrase = Nominalized Clause AND License = Nominalized Clause AND Child Count = 2

7. This more constrained search finds 5,808 nominalized clauses that are preceded by an overt nominalizer.
8. How many nominalized clauses lack a preposed nominalizer segment? To find these, we constrain the search by specifying that the child count of the sought node is *one*, corresponding to the presence only of a nominalized clause. We impose this constraint by calling up the syntax search afresh. It should still hold the specifications for the previous search. Scrolling to the very bottom of the right-hand panel, we encounter the "Parent Child Counts" section. We change "Child Count: = 2" to "Child Count: = 1" and launch the search. Now the search specification reads:

Phrase 1: Phrase = Nominalized Clause AND License = Nominalized Clause AND Child Count = 1

9. The search finds 727 instances that lack an overt nominalizer.
10. These results "add up" (5,808 + 727 = 6,535), so we probably need proceed no further as regards overall totals. But, to be very careful, we also search with the child count set to zero (no hits) and the child count set to ">2" (no hits). Had either of these protective searches yielded hits, we would have discovered errors in the database and would inform Logos of our discovery so that they could pass the word along to Andersen-Forbes.
11. We are now in position to carry out these three searches with limited "search ranges." We can find the counts (and search results) for the Torah, the Former Prophets, the Latter Prophets, the Poetry, and the (Other) Writings by suitably limiting the search ranges.[12] We can use the results to complete this table:

	Overt Nominalizer	Null Nominalizer	Overall Incidence
Torah			
Former Prophets			
Latter Prophets			
Poetry			
(Other) Writings			
Total	5,808	727	6,535

If you fill in the table, you may see why we have elsewhere called אֲשֶׁר a "prose particle."[13]

---

[12] The "Search Range" is specified toward the top of the "Syntax Search" pane.
[13] Note, however, that inferences based on the table counts will be shaky because the five parts of the Hebrew Bible are of differing lengths. There are, of course, ways to compensate for these differences.

Example #2 — Basic Searches for Coordinated Phrases:

The following shows how searches exploiting out-degree constraints can facilitate tallying correct counts of constituents. Suppose that, for whatever reason, we wish to produce a census of all of the coordinated simple noun phrases.

1. Our plan might be to specify a series of searches, with the number of conjuncts running from two up to, say, twenty—assuming that no coordination phrase has more than twenty conjuncts.
2. To specify the search for the simplest case, two conjuncts, we would:
    a. Click "Search" to call up the "Syntax Search" dialog.
    b. Click "Clear" to erase any left-over search specification.
    c. Click "Phrase" to specify the constraint that our search requires the presence of a phrase. In the right pane, check "Highlight this term in search results."
    d. Install a noun segment first conjunct:
        i. Click "Seg" to install "Segment 1."
        ii. In the right-hand pane, click "Must be an immediate child of parent."[14]
        iii. Under the "Parts of Speech" heading, click the "Clear" button below "Edit…".
        iv. Click the "Edit…" button.
        v. Check "Noun" at the top of the "Add Reference" dialog and click "OK."
    e. Install a coordinating conjunction:
        i. In the left-hand pane, click "Seg" to install "Segment 2."
        ii. In the right-hand pane, click "Must be an immediate child of parent."
        iii. Under the "Parts of Speech" heading, click the "Clear" button below "Edit…".
        iv. Click the "Edit…" button.
        v. Then check "Conjunction" at the top of the "Add Reference" dialog, check "coordinating," and then click "OK."
    f. Copy and position a noun segment second conjunct:
        i. Select (highlight) "Segment 1."
        ii. Click "Copy" in the bar above the left pane, making "Segment 3" appear, already being constrained to be a noun (since it is a copy of "Segment 1").
        iii. With "Segment 3" still highlighted, click the right-pointing green arrow in the bar above the pane to move the segment into its proper position as a daughter of "Phrase 1."
    g. At this point, we have constrained a two-conjunct search as follows:

Phrase 1 — Highlight

Add: Phr / Seg / Any / Gap / Group / Unordered Group / OR

→ **Segment 1**: ((Part of Speech = noun))

→ **Segment 2**: ((Part of Speech = conjunction AND Conjunction Subclass = coordinating (¡)))

→ **Segment 3**: ((Part of Speech = noun))

---

[14] It is essential to grasp the importance of this constraint. Suppose that you specify a clause immediate constituent (CIC), say, and then specify a segment. If you connect the segment to the CIC **without** adding the constraint that the segment is the "immediate child of parent," then the segment is only required to be somewhere—anywhere—in the part of the phrase marker dominated by the CIC. Only by requiring that the segment is the "immediate child of parent" does one require that the segment be immediately dominated by the CIC, its true daughter.

h. We might save the search specification by clicking "Save As…" at the bottom of the "Syntax Search" window, typing the desired name into the box to the right of "Name:" (I called my version of the search "Noun1andNoun2"), and clicking "Save."

i. When we run the search, the references of all 3,201 instances of "Noun1 and Noun2" coordination phrases are presented.

j. Alas, there is a massive *Musca domestica* in the unguent! In the "Syntax Search Results" panel for the search just completed, scroll down to the *two* entries for Gen 7:13. Clicking on the "chicken claw-print icon" to bring up the associated phrase marker, we see that the phrase being dealt with has three conjuncts, being שֵׁם־וְחָם וָיֶפֶת, "Shem and Ham and Yapeth." Our search found "Shem and Ham" and "Ham and Yapeth," by using *Ham* twice. The count is wrong…

k. We need to further constrain the search so that only two-conjunct coordination phrases are found. The mechanism for so doing is ready to hand.
   i.   Call up the faulty search just executed.
   ii. Highlight "Phrase 1."
   iii. Scroll the right-hand pane to the bottom.
   iv. Change "Child Count: Anything   " to "Child Count: = 3."
   The search specification now reads:

Phrase 1: Child Count = 3 — Highlight
Add: Phr / Seg / Any / Gap / Group / Unordered Group / OR
→ **Segment 1**: ((Part of Speech = noun))
→ **Segment 2**: ((Part of Speech = conjunction AND Conjunction Subclass = coordinating (ו)))
→ **Segment 3**: ((Part of Speech = noun))

l. Running this search yields 1,751 hits. Scrolling down through the list of hit citations, we see that Gen 7:13 is no longer present.

m. Calling up the search specification just executed, you should click "Save" at the bottom of the panel to over-write the faulty specification with this correct version.

Example #3 — Intermediate Searches of Suffix Pronouns:

The following shows how rather simple sequences of searches can be used to survey the uses of a part of speech. We ask: *How are pronoun suffixes used in Biblical Hebrew?* We adopt a divide and conquer strategy, proceeding as follows.

1. Overall count of pronoun suffixes:
   a. Initiate a search.
   b. Install a segment and require that it be highlighted.
   c. Scroll down on the right to "Parts of Speech" (or "Morphology" in old releases).
   d. Click "Edit…"
   e. In the "Add Reference" panel, click "Pronoun".
   f. In the "Conjunction, Pronoun, Miscellany Subclasses" sub-panel, click "suffixed", then click "OK".
   g. Running the search should yield about 45,587 hits.

h. Look at a few to verify that the search has found suffix pronouns. Notice that many of the suffix pronouns are parts of noun phrases or of prepositional phrases.

2. **Now, here is an absolutely critical point which holds for all searches:** *A structure may be a single segment, or it may be a phrase.* When searching, both possibilities must be taken into account. Otherwise, instances of the searched-for phenomenon will be missed.

3. We next determine that the suffixes divide into two large subsets. When we search for a CIC immediately dominating a suffix pronoun segment, we get 6,911 hits. When we search for a phrase immediately dominating a suffix pronoun segment, we get 38,676 hits. These results sum to 45,587, the grand total found earlier.

4. We next examine which CICs immediately dominate the suffix pronoun segments. We do this by repeatedly specifying a particular CIC type and running a search. For example, to see how often a direct object is simply a suffixed pronoun, we form a search involving a CIC with (immediate) daughter a suffixed pronoun.

   a. Repeat 1 a-f just above to install a pronoun suffix segment, use the "Add" menu to install a CIC node, and then use the little green arrows to move the CIC so that it is the mother of the segment.

   b. Highlight the CIC node in the left-hand panel (left click on it) and then click the "Direct Object" box in the right-hand panel.

   c. Running the search should find about 5,470 hits.

   d. Iterating this search strategy until each of the CIC categories has been search for, we obtain these results for the counts of CICs immediately dominating suffix pronouns:

CIC Type	Count	CIC Type	Single Location
Subject	1,422	Possessor	Num 30:5
Direct object	5,470	Agent	Isa 44:21
Second object	8	Beneficiary	Zeph 3:19
Indirect object	8		

5. We also sub-divide the 38,676 phrases that contain a suffix pronoun. Create a search consisting of a phrase dominating a suffix pronoun segment. (I.e., specify a phrase which is the mother of the suffix pronoun segment.)

   a. Using the "Occurrence" panel, require that the suffix pronoun segment be "first child of parent." This search yields 131 hits. Manually, or algorithmically, one may subdivide this yield into two sub-sets, 76 CICs and 55 non-CICs. Of the former, 61 are direct objects, 14 are subjects, and one is a beneficiary (Zech 7:5). These counts can be incorporated into the table above to create a census of all instances of CICs prominently[15] involving suffix pronouns:

CIC Type	Count	CIC Type	Single Location
Subject	1,436	Possessor	Num 30:5
Direct object	5,531	Agent	Isa 44:21
Second object	8		
Indirect object	8		
Beneficiary	2		

---

[15] By "prominently," we indicate that the suffix pronouns are central to the CICs here tallied.

The 55 non-CIC constituents having a suffix pronoun in first position all are personal pronoun phrases. The phrases are then themselves typically part of prepositional phrases (x42). See, for example, the direct object at the end of the phrase marker for Num 4:23. Thirteen times the phrases are represented as extended suffixes. See, for example, the phrase marker in Jer 51:35. These constitute a small weak spot in our representation.

    b. Return to the previous search and use the "Occurrence" panel to require that the suffix pronoun segment be "last child of parent." This search yields 38,545 hits.

6. If we wish, the results of the previous search can be sub-divided by adding a segment in front of the suffix pronoun segment and successively constraining this new earlier segment to be a major part of speech. When we do so, we find:

Lead Segment Part of Speech	Count
Simple Preposition	12,974
Pure noun participle	754
Noun[16]	24,432
Adverbial[17]	1

This leaves 384 instances unaccounted for. If we replace the lead segment by a lead phrase and run this search, we find 384 hits. By examining the hits and then constraining the lead phrase to be a compound preposition, we determine that the phrase always is a compound preposition.[18]

Example #4 — Advanced Searches on Subject-Verb Number Agreement:

> *NOTE: Developed below is a quite difficult and extensive search. Readers should not be dismayed if they find its particulars daunting. It is not easy sledding. As always, the key to success is — Decompose, decompose, decompose…*

We turn finally to a complicated set of searches that require great care in their formulation, execution, and assessment. The question whose answer we seek is: *How often (and where) is there number disagreement between a clause's explicit subject and its verb?*

A. Which subjects? We first assess for which of our four types of *subject* this inquiry makes sense:

    1. *Inadmissible subject types:* Two of the types of subjects specified for our data have no place in this inquiry:

        a. Subject of address – A speech has no meaningful number attribute.

---

[16] The actual search returns 24,431 hits. We increase the count by one to take account of the apparent nominal *lapsus calami* at Job 26:12 (to judge from the supplied *Qere* reading) not detected by the search.

[17] The suspicious parsing associated with this adverbial may be found at Jer 37:19. Reanalysis is called for.

[18] Note that the compound (inappropriately?) includes a negator in two cases: Job 18:15 and Job 39:16. These two loci need to be reanalyzed.

b. "Janus" subject-complement/subject – This CIC type never appears in clauses having finite verbs. Stated differently: a Janus node and a finite-verb node never are sisters.

2. *A rare, somewhat peculiar, subject type:* A third type of subject, the distributive subject, is fairly rare. Its examination will allow us to emphasize a very important general point about the structure of our data.

   a. To search for all instances of the distributive subject, call up the "Syntax Search" dialog, instantiate a "clause" ("Clause 1: Structure = Clause"), give the clause a "Clause IC" (CIC), and constrain the CIC to be a distributive subject.

   b. Your search should find about 313 instances. This grand total includes instances within verbless clauses (x15; see 2 Sam 20:1, final clause, for an example), within clauses whose predicator is a purely verbal participle (x19; see Hag 1:9, final clause, for an example), plus three oddities.

   c. Recall that the structure of the constituent making up a CIC is specified by the label on the daughter of that CIC. **Reminder:** *A structure may be a single segment, or it may be a phrase.* When searching, both possibilities must be taken into account. Otherwise, instances of the searched-for phenomenon will be missed.

      i. Have a look at the phrase marker for Num 2:34b. Here, the daughter of the **dist sbj** CIC is the segment אִישׁ, glossed 'each.' Note that the associated verb has plural number.

      ii. Next, look at the phrase maker for Num 4:49. Here, the label on the daughter of the **dist sbj** CIC is **n:humn / echo**. The constituent is an echo phrase (אִישׁ אִישׁ) having human semantics.

3. *The "standard" subject type:* Of course, the main contributor to our tallies will be the "standard" subject. To find out the grand total of explicit subjects, we carry out the search specified in A. 2. a. above, modified by constraining the CIC to be a (standard) subject. We get 31,834 hits, but many of these are in clauses that lack a finite verb. We need next to impose the constraints that the clauses have predicators and that the predicators are finite verbs.

B. Imposing the finite-verb constraint. When we require of a clause both that a standard subject be present and that a finite verb be present, two complications must be explicitly catered for: 1) the order of the constituents matters, and 2) the possibility of interpolated constituents must be explicitly taken into account. For the present situation, this means that we must carry out two searches, one constraining the clause to consist of

a subject (S) — "anything" (A) — finite-verb predicator (V)

and the second consisting of

finite-verb predicator (V) — "anything" (A) — a subject (S).

1. *The first full search (SAV):* To search for all instances of the SAV pattern, call up the "Syntax Search" dialog.

   a. Instantiate a "clause" ("Clause 1: Structure = Clause")

   b. Give the clause first a "Clause IC" (CIC), constrained to be a subject and highlighted.

c. Second, give the clause "Anything".

d. Give the clause third a "Clause IC" (CIC), constrained to be a finite verb and highlighted. The search specification looks like this:

```
Clause 2: Structure = Clause
Add: Clause / Clause IC / Phr / Seg / Any / Gap / Group / Unordered Group / OR
  → Clause IC 2: Structure = Subject — Highlight
  → Anything
  → Clause IC 3: Structure = Finite Verb — Highlight
```

e. When the search is run, around 5,905 hits should be returned. One should sample a few of the phrase markers found by the search to verify that they in fact meet one's expectations.

f. Once satisfied, it is wise to call up the search specification afresh and save it for later use. ("Save As…", Name: SAV, "Save".)

2. *The second full search (VAS):* To search for all instances of the VAS pattern:

a. Call up the "Syntax Search" dialog.

b. Use the up-green-arrow and down-green-arrow to swap the subject and finite-verb CICs with respect to "Anything." The search specification looks like this:

```
Clause 2: Structure = Clause
Add: Clause / Clause IC / Phr / Seg / Any / Gap / Group / Unordered Group / OR
  → Clause IC 3: Structure = Finite Verb — Highlight
  → Anything
  → Clause IC 2: Structure = Subject — Highlight
```

c. When the search is run, around 14,619 hits should be returned. One should sample a few of the phrase markers found by the search to verify that they in fact meet one's expectations.

d. Once satisfied, it is wise to call up the search specification afresh and save it for later use. ("Save As…", Name: VAS, "Save".)

C. Suspension/resumption of the subjects. At this point, we may decompose the makeup of the subjects in the searches by observing that toward the bottom of the right-hand panel in the syntax search, one is allowed to specify the "syntactic license" of the CIC. Does it involve *suspension*, *resumption*, or *neither*? Of our 20,524 candidate clauses (those having both an explicit subject and a finite verb, in whatever order), 347 involve suspended or resumed subjects (1.7%). One may elect to study all of the subjects together, to study the three sorts of subjects each-by-each, or to omit the suspended/resumed subjects from consideration. We shall not weigh in on this matter here.

D. Decomposition of the finite verbs. Recall our assertion in A. 2. c.: *The structure may be a single segment, or it may be a phrase*. This is true of finite-verb CICs. To determine the relative

incidence of segmental versus phrasal finite-verb CICs, call up each of our saved searches, extend the constraints on the finite-verb CIC, and search afresh. We proceed as follows:

1. SAV search decomposition, segmental CIC:
   a. Load the SAV search.
   b. Highlight the finite-verb CIC.
   c. Install a segment and make it the "immediate child of parent".
   d. Run the modified search.
   e. You should find about 5,873 clauses having a segmental finite-verb CIC in an SAV sequence (99.5%).
2. SAV search decomposition, phrasal CIC:
   a. Recall the search specification just run.
   b. Replace the segment by a phrase.
   c. Run this search.
   d. You should find 32 clauses wherein the finite-verb CIC has a phrasal structure. (See, for example, Nah 3:16 with its coordinated verbs.)
3. VAS search decomposition, segmental CIC:
   a. Load the VAS search.
   b. Highlight the finite-verb CIC.
   c. Install a segment and make it the "immediate child of parent".
   d. Run the modified search.
   e. You should find about 14,589 clauses having a segmental finite-verb CIC in a VAS sequence (99.8%).
4. VAS search decomposition, phrasal CIC:
   a. Recall the search specification just run.
   b. Replace the segment by a phrase.
   c. Run this search.
   d. You should find 31 clauses wherein the finite-verb CIC has a phrasal structure. (See, for example, Isa 45:25 with its coordinated verbs.)

One may elect to analyze the phrasal finite-verb CICs or leave them out of consideration. For the purposes of this paper, we choose to omit them from further consideration.

E. Segmental subjects, SAV sequence. We therefore impose the constraint that the finite-verb CICs are segmental and continue with the analysis. It is trivial to constrain the number of a segmental finite verb. Nor is it a challenge to constrain the number of a segmental subject. Hence, it is relatively simple to find instances where segmental subjects and segmental finite verbs do not agree in number.

1. Subj Seg – Any – Finite Verb Seg (SsegAVseg) search:
   a. Load the SAV search.
   b. Add a segment, immediate daughter of the finite verb.
   c. Similarly, add a segment, immediate daughter of the subject.
   d. Do a "Save As…", naming this SsegAVseg.
   e. Running the new search should yield around 2,608 hits.
2. Disagreeing Subj Seg – Any – Finite Verb Seg search:
   a. Load the SsegAVseg search.

b. **Important**: *Highlight (click on) the segment whose mother is the subject CIC node.*

c. On the right-hand side of the search dialog, scroll to the bottom.

d. Below "Agreement" you see:
This term's <u>feature</u> must be <u>in some relation to</u> that of <u>another term</u>.

e. Click on the red <u>feature</u> and select "**Number**". The red <u>feature</u> should be replaced by Number.

f. Click on <u>in some relation to</u> and select **different from**, which should replace the red words.

g. Click on <u>another term</u> and select *the entry that describes the segment mothered by the finite verb CIC*. This descriptor should replace the red text. At this point, my agreement statement reads:
This term's **Number** must be **different from** that of **Segment 1**.
Yours may differ in the last term because of differences in labeling.

h. **CRITICAL STEP**: To the right of the agreement statement is a white right-pointing arrow in a green box. Click this arrow. This exports the agreement statement, making it part of the search constraints. If you fail to export the constraint, the results will not reflect the agreement constraint!

i. Running the search should produce around 83 hits.

j. These 83 hits deserve careful scrutiny, as many of them may have resulted from faulty analyses. For example, in Song of Songs 8:11b, the CIC labeled **sbj** should be labeled **dist subj**. We have made the correction in our master files.

F. Segmental subjects, VAS sequence. We carry out analogous procedures to locate number disagreement when the CIC sequence is VAS.

1. Finite Verb Seg – Any – Subj Seg (VsegASseg) search:
   a. Load the VAS search.
   b. Add a segment, immediate daughter of the finite verb.
   c. Similarly, add a segment, immediate daughter of the subject.
   d. Do a "Save As…", naming this VsegASseg.
   e. Running the new search should yield around 6,053 hits.

2. Disagreeing Finite Verb Seg – Any – Subj Seg search:
   a. Load the VsegASseg search.
   b. **Important**: *Highlight (click on) the segment whose mother is the finite verb CIC node.*
   c. On the right-hand side of the search dialog, scroll to the bottom.
   d. Below "Agreement" you see
   "This term's <u>feature</u> must be <u>in some relation to</u> that of <u>another term</u>."
   e. Click on the red <u>feature</u> and select "**Number**". The red <u>feature</u> should be replaced by **Number**.
   f. Click on <u>in some relation to</u> and select **different from**, which should replace the red words.
   g. Click on <u>another term</u> and select *the entry that describes the segment mothered by the subject CIC*. This descriptor should replace the red text. At this point, my agreement statement reads:
   "This term's **Number** must be **different from** that of **Segment 2**."

Yours may differ in the last term because of differences in labeling.

h. **CRITICAL STEP**: To the right of the agreement statement is a white right-pointing arrow in a green box. Click this arrow. This exports the agreement statement, making it part of the search constraints. If you fail to export the constraint, the results will not reflect the agreement constraint!

i. Running the search should produce around 185 hits.

j. These 185 hits deserve careful scrutiny, as many of them may have resulted from faulty analyses. For example, in Exod 3:22, the CIC labeled **sbj** should be labeled **dist subj**. We have made the correction in our master files.

G. Phrasal subjects involving list phrases. Up to this point, our results have been potentially definitive. That is, by carrying out the various searches indicated for the segmental CICs, we are in a position to provide a definitive survey of segmental subject-segmental verb agreement.

When the subject is phrasal, things get tricky. This is because phrases can themselves be made up of other phrases and because information regarding number is not available at the phrasal nodes of our phrase markers. Take a look at the construction involved in the subject in the phrase marker for Num 32:11. That the subject has plural number is shown by the plural noun אֲנָשִׁים 'men' embedded three phrases deep in the subject. In this case, and in many others, searches must be designed to recover information deep down in constituents.

Fortunately, for some constructions searching is much more straight-forward. For example, to first approximation we expect union/mixed/joined lists to be plural. Consider the phrase marker in Ruth 1:4b. Here the verb is singular, but the subject is plural. We may easily search for subjects involving such lists. We proceed as follows:

1. Finite Verb Seg – Any – Subj UPhr (VsegASuphr) search:
   a. Load the VAS search.
   b. Add a segment, immediate daughter of the finite verb.
   c. Add a *phrase*, immediate daughter of the subject.
   d. Scroll down in the right-hand panel to "Syntactic License" and click on "Conjoined Phrases: Structural." This should put a green check in the three categories "Gathered," Joined", "Union". The search specification should look like this:

Clause 2: Structure = Clause

Add: Clause / Clause IC / Phr / Seg / Any / Gap / Group / Unordered Group / OR

→ **Clause IC 3**: Structure = Finite Verb
   → **Segment 1** — Highlight
→ *Anything*
→ **Clause IC 2**: Structure = Subject
   → **Phrase 1**: License = { Gathered (Mixed Lists), Joined, Union or Disjunction } — Highlight

   e. Do a "Save As…", naming this VsegASuphr.
   f. Running the new search should yield around 793 hits. These are all the instances where a (segmental) finite verb is followed at some point by a subject consisting of a list.

2. Finite Verb Seg – Any – Subj UPhr Disagreement (VsegASuphrDis) search:
   a. Load the VsegASuphr search.
   b. Highlight the segment which is the daughter of the finite verb CIC.
   c. Scroll down in the right-hand panel of the search dialog until you encounter "Parts of Speech" (or "Morphology" in older versions).
   d. Click "Edit…"
   e. In the "Add Reference" dialog that pops up, click "Verb", then "Singular", then "OK". (You have now constrained the verb to be singular.)
   f. Look at the specification of the search in the left-hand panel of the Syntax Search. It specifies that you seek a clause consisting of a finite verb CIC, Anything, and a subject CIC. The verb CIC consists of a segment that is constrained to be singular. The subject consists of a phrase that is constrained to be one of the three kinds of lists that we define. By implication, therefore, the number of the verb differs from that of the subject.
   g. Run the search. You should get about 526 hits. *This tells us that two-thirds of the clauses (526/793) of the specified structure exhibit disagreement in number.* Look them over to verify that they are as advertised…
   h. You may wish to perform a "Save As…" on the search. I call mine "VsegASuphrDis".
3. The corresponding Subj UPhr – Any – Finite Verb Seg (SuphrAVseg) search and Subj UPhr – Any – Finite Verb Seg Disagreement (SuphrAVsegDis) search can be specified and carried out along the obvious lines.

H. Phrasal subjects involving definite phrasal subjects. As a further small search, consider the case where the subject is a basic determined-noun phrase. First, find the total number of such instances for the VAS ordering.
   1. Finite Verb Seg – Any – Subj DefPhr (VsegASdphr) search:
      a. Load the VsegASuphr search.
      b. Highlight the *phrase*, immediate daughter of the subject.
      c. Scroll down in the right-hand panel to "Syntactic License" and click on "Conjoined Phrases: Structural." This should *remove* the green checks in the three categories "Gathered," Joined", "Union".
      d. Under "Tightly Joined Phrases", click "Definite".
      e. Do a "Save As…", naming this VsegASdphr, say.
      f. Running the new search should yield around 1775 hits. These are all the instances where a (segmental) finite verb is followed at some point by a subject consisting of a basic definite nominal.
   2. Finite Verb Seg – Any – Subj DefPhr Disagreement (VsegASdphrDis) search:
      a. Load the VsegASdphr search.
      b. Install a segment as immediate daughter of the definite phrase.
      c. Toward the top of the right-hand panel of the search dialog, open "Occurrence" and click "Must be last child of parent". This constrains the segment to be the nominal rather than the definite article.
      d. Highlight the daughter segment of the finite verb.
      e. On the right-hand side of the search dialog, scroll to the bottom.
      f. Below "Agreement" you see:

"This term's <u>feature</u> must be <u>in some relation to</u> that of <u>another term</u>."

g. Click on the red <u>feature</u> and select "**Number**". The red <u>feature</u> should be replaced by **Number**.

h. Click on <u>in some relation to</u> and select **different from**, which should replace the red words.

i. Click on <u>another term</u> and select *the entry that describes the segment mothered by the subject CIC's phrase node*. This descriptor should replace the red text. At this point, my agreement statement reads:

"This term's **Number** must be **different from** that of **Segment 2**."

Yours may differ in the last term because of differences in labeling.

j. **CRITICAL STEP**: To the right of the agreement statement is a white right-pointing arrow in a green box. Click this arrow. This exports the agreement statement, making it part of the search constraints. If you fail to export the constraint, the results will not reflect the agreement constraint!

k. Running the search should produce around 112 hits. These 112 hits deserve careful scrutiny, as many of them may have resulted from faulty analyses or involve collective/mass nouns.

I. Finishing up the searches. It should be pretty clear by now that the algorithmic finding all of the instances of disagreement where phrasal constituents are involved would be an arduous task. There comes a time when there is no better method of going through phrase markers than manual review …