Biblical Hebrew parsing on display:
The Role-Lexical Module (RLM) as a tool for Role and Reference Grammar

Nicolai Winther-Nielsen
Dansk Bibel-Institut
nwn@dbi.edu

Reference: Nicolai Winther-Nielsen, "Biblical Hebrew parsing on display: The Role-Lexical Module (RLM) as a tool for Role and Reference Grammar". SEE-J Hiphil 6 [http://www.see-j.net/hiphil] (2009). Accessed DD.MM.YYYY.

Abstract
The Role-Lexical Module (RLM) is a tool to build a lexicon of logical structure and semantic representation for Role and Reference Grammar (RRG). It was first presented in Winther-Nielsen (2008), and the present contribution explains improvements in the interface developed by project developer and designer Chris Wilson in 2008 and 2009. Using the purely structural information in the database of the Werkgroep Informatica at the Vrije Universiteit (WIVU) in Amsterdam, we display linguistic data bearing on the syntax-to-semantics linking algorithm at work in the structuralist-functional theory of RRG. In this paper, I supply further information on our new transliteration adhering to Nava Bergman's *Cambridge Workboook of Biblical Hebrew* (2005) as well as automated glossing and lexical selection. The main focus of this paper is, however, to explain how a Web-application can be used to display syntactic structure in tables containing the output of rewrite rules from a chart parser. The boxes in these tables display syntactic structure in a way which resembles the syntactic trees of a projectionist theory like RRG, and adding a true tree display to the RLM-tool is by now largely a matter of funding of programming. Syntactic examples are taken from our Genesis 1-3 corpus (i.e. Gen 1:1-3, 5, 16, 27; 2:5), and the issue of syntactic templates is brought up. For the initial sentence (Gen 1:1), I briefly discuss the logical structure and the semantic representation of the core combined with thematic roles. I conclude by proposing how this tool can be improved and used for other languages, and how it has teaching potential for online linguistic courses as well as learning tools using our database technology.

Key Words: Role and Reference Grammar; The Role- Lexical Module. Biblical Hebrew; Hebrew verb; Syntactic trees; parsing; computer-assisted analysis and technology. Logos. SESB. WIVU.

Introduction

Despite significant advances in modern computer technology, Biblical Hebrew scholarship still has much ground to cover before it can play its proper role in modern linguistics, chiefly because Hebrew scholars usually do not share their results in a familiar linguistic format. The Role-Lexical Module (RLM) is a Web-application presented for the first time in Winther-Nielsen (2008), and it is first and foremost a computational module for representation of semantic roles within a particular theory of grammar. It was devised in order to display linguistic data from the Hebrew Bible according to common linguistic conventions, and we are now able to display the nodes of syntax trees in tables. They are produced as output from a parser which in the processing literature is use for "a computer program that divides code up into functional components".[1] Wilson (2009) has explained the parser developed for our RLM-tool, and this paper will explain how our application can help the linguist to write rules that will parse the syntactic structure of simple clauses and display them for teaching and research. This parser is a very welcome addition to the central job of the RLM-tool which is to do classification of event structure for Hebrew verbs as well as semantic representation.

My goal in this particular contribution is to explain the interface of the RLM-tool, and show how it can display data from a restricted corpus of Biblical Hebrew and share results with fellow linguists. The following description does not pretend to offer the final version of a glossing of data from the Hebrew Bible but is an account of our work in progress. Unfortunately, our project has had no external funding for programming work which would have allowed Wilson to complete the Role-Lexical Module for Biblical Hebrew in a version 2.0 with full projection of syntactic constituents and information structure. The technical build-up of the system as well as the broader framework of LEX as a translation system is set out in Wilson (2009), but at present it is impossible to know when future funding may allow him to develop the tool further. In my current work I use the RLM-tool for lexical and semantic description of Biblical Hebrew RRG.[2] The ultimate goal – or perhaps utopian dream – is to offer an automated generation of linguistic data and display of syntactic structure for the complete corpus of Ancient Biblical Hebrew. To our knowledge, our project is the first one to provide an automated display of the RRG syntax for a text corpus, and certainly at least for a corpus of Biblical Hebrew, but the technology is designed to be extended to other languages and corpora with a similar database.

Since the RLM-tool was originally designed to work with the theory of RRG

[1] For the definition see http://www.wordreference.com/definition/parser.
[2] Accordingly, the present paper only describes how far we can get in exploiting data from a specific database, and I discuss the interim stage we have arrived at in early April 2009.

developed into its most recent statement in Robert D. Van Valin (2005; 2008), this theory will provide the reference point for display and processing of linguistic data. With years of spare time support from chief engineer and programmer Chris Wilson, Cambridge, as my sole technical developer and designer[3] I have conceived of the RLM project as a Web-application for RRG analysis of Hebrew and for storing and retrieval of linguistically annotated Hebrew text. We have designed the RLM-tool to display linguistic entities and syntactic structure by processing structural data from the Werkgroep Informatica database at the Vrije Universiteit in Amsterdam (WIVU).[4] The following discussion reflects what we may be able to do within the limits of this particular database in order to display syntax-to-semantics tagging for RRG, but we are well aware of the limitations of this and other databases, when data are exploited for uses for which they were not created.

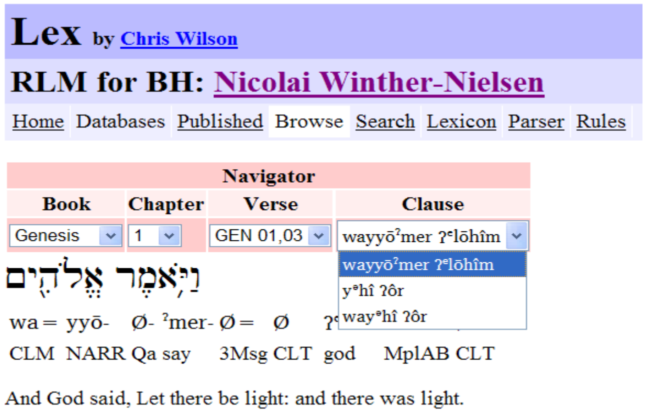


Figure 1 Role-Lexical Module (RLM) tool to select Biblical Hebrew (BH)

[3] Since version 1.0 of the RLM-tool published in Winther-Nielsen (2008), Chris Wilson has had four weeks of unfunded programming work on the project: two weeks at Christmas and New Year 2007-2008 and two weeks March 22-April 3 2009. I am sincerely grateful for all this expert programming by Chris without any funding. The project utterly depends on his work, and I hope that there will eventually be funding for him to continue future work on his Lex project.

[4] See Talstra and Sikkel (2000) and the further information on http://www.th.vu.nl/~wiweb/const/index.htm.

Fellow linguists are encouraged to inspect our data from the Hebrew Bible on display on our Web site (http://lex.qwirx.com/lex/clause.jsp). The RLM tools menu in Figure 1 may serve as a handy user's guide to the Web-application. I will explain in detail how the tool selects clauses in our corpus from Genesis 1-3 which at present is the only stretch of Hebrew text we display in full. The tool by default activates the primary function of *Browse* and allows the linguist to select any clause within the open corpus through the navigator. As a member of the team in Amsterdam I have access to the database of the entire Hebrew Bible by special permission, and I can produce data for the entire corpus.[5] The long-term goal of the project is to enrich the database with analysis of verb classes and semantic representations of clauses, and to display such research results for inspection under *Published*. Other instances of a particular verb can be found by *Search* using the MQL query language of Emdros (Sandborg-Petersen 2008).[6] *Parser* displays all possible outputs from the *Rules* which I write in order to construct syntactic trees for Hebrew. The link "RLM for BH: Nicolai Winther-Nielsen" points to the homepage of the RLM project (http://3bm.dk/index.php?p=3) where I will share news on the development of the RLM-tool and work on RRG. *Home* contains information on Unicode fonts to be downloaded for enhanced display of special characters, and additional *Databases* may also be offered in the future. Nothing new will be added to the discussion of *Lexicon* and *Wordnet* in Winther-Nielsen (2008).

The following presentation is designed first to treat transliteration and glossing of linguistic data from Hebrew, then proceeds to syntactic tree nodes in a box display of samples from Genesis 1-3. I introduce the basic syntax of Biblical Hebrew in an RRG account, and my evidence is clauses from Gen 1:1-3, 5, 16, 27; 2:5. The parsing of the first clause of the Hebrew Bible (Gen 1:1) will be discussed, and its logical structure will be analyzed as a small specimen of lexical representation enriched with thematic roles in order to supplement the first presentation of the tool mentioned above.

To sum up, the reader is about to enter the *Browse, Parser* and *Rules* links on the tools menu, and I will guide the reader into this tool for display of Biblical Hebrew syntax in a format useful for linguistics in general and RRG-researchers in particular. I will also provide one example of *Search* and *Published* in the format of the old

[5] I appreciate the continuing support of professor Eep Talstra, the director of the WIVU. I would also like to thank Dr Bertram Salzmann for conveying the decision of the license committee of the German Bible Society to give us permission to use the WIVU database in an email of November 14, 2008, and confirmed at a meeting in Boston later that month.

[6] The RLM project relies heavily on Emdros (http://emdros.org/), a textdatabase engine for storing and retrieving analyzed text in corpus linguistics and computational linguistics developed by Sandborg-Petersen (2008). Emdros was explicitly created to store multiple object types for linguistic monads as required by the theory of RRG (2008:49).

transliteration in order to briefly introduce the task of mapping logical structures into the lexicon of the RLM as final demonstration of the linguistic potential of the RLM Web-application.

Shortcomings of display in Bible software

At this day and age sophisticated Bible software is commercially available for scholars with a working knowledge of the Hebrew script and trained in traditional philological grammar and tagging. Biblical Hebrew is part of a long tradition of painstaking study by Jewish and Christian scholars, but this venerable philological tradition has only slowly developed towards current approaches in modern linguistics, even if Francis I Andersen, Eep Talstra and others opened the field in the 1970s (van der Merwe et al 1999: 18-21). Yet today the broader community of Hebrew scholars in many ways continues to work with old-fashioned philological approaches, and it uses the peculiar Hebrew script in presentations of data to fellow scholars. Therefore important research by Hebrew linguists has not been made available to the larger linguistic community, and unfortunately remains largely unknown.

This problem has not been addressed well by the new computational tools for the study of Biblical Hebrew. Within the last decade students and scholars have been able to buy high-quality and very useful software products that meet the needs of most Hebrew scholars and students of Theology. Employing these tools it is possible to look up a verse in the Hebrew Bible and have the text displayed on the screen of the computer with morphological and lexical information on each word. One case in point is the WIVU-based software by name of the Stuttgart Electronic Study Bible (SESB) published by the German Bible Society (Hardmeier, Talstra and Salzmann 2009).[7] In this software it is possible to place the cursor over any word in the text and activate a popup window which displays the linguistic information in a less accessible form.

To illustrate this, we will take a look at Gen 1:3 in Figure 2. In the creation account in Genesis the description of the first day of creation begins with the *inquit* clause *(And) God said*. The verb of saying is used as part of an extremely common quotation formula in order to convey who the speaker of the following quote is (Miller 1996: 52-61 *et passim*). This is the first very simple clause type in the Hebrew Bible and equivalent to clauses with a subject and a verb in other languages. Now, by way of example, for the first word, *wayyōʾmer*, the software informs the user that the word class is “conj” followed by “ אמרverb 1”. This information requires that the linguist is

[7] It uses the Libronix Digital Library System interface produced by Logos Research Systems. For an evaluation of version 1.0 see Kummerow (2005), and general information on the program homepage http://www.sesb-online.de/ (or in Danish at http://3bm.dk/index.php?p=18).

familiar with the Hebrew consonants. The last line with grammatical information on the verb of quotation will also pose problems for the general linguist, because he has to figure out the meaning of “qal” and “wayyiqtol” which are generally unknown outside Biblical scholarship.

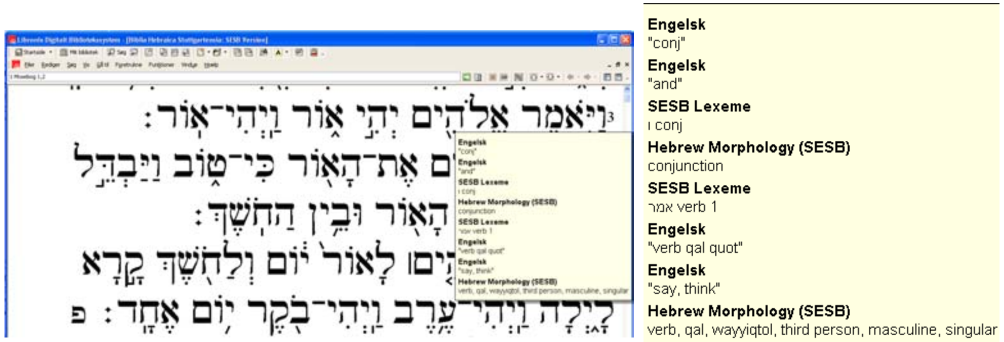


Figure 2 Information Window from SESB: word glosses

Another commercial tool illustrates how the non-specialist can get close to the Hebrew text without being a competent reader of Biblical Hebrew. He can use *The ESV English-Hebrew Reverse Interlinear Old Testament* edited by McDaniel and Collins (2006) and produced by Logos Research Systems, Inc for the Logos Bible Software version 3 (Figure 3). This interlinear display can be set up to start from the English text and then show the Hebrew text form in a numbering which reflects their sequential position in the interlinear display for the Hebrew Bible. Beneath this Hebrew line, another row in the table will list the dictionary words (lemmas) as Hebrew consonantal clusters. But again the general linguist must be able to decipher the Hebrew script. The last line is abbreviated grammatical code which would also be difficult to read, if it were not for the popup window which spells out the information in full. In this case the program uses familiar terms like “imperfect”, although the linguist might wonder why the verb is translated as past tense.

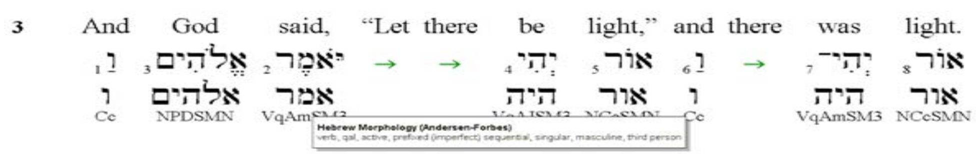


Figure 3 Interlinear display from Logos Bible Software

With the release of the Libronix Digital Library System in version 3.0 a couple of years ago these resources gave access to syntactic analysis. The new version of the SESB 3.0 released in April 2009 publishes more than 25 years of interclausal analysis for the WIVU-database (Talstra 2006). The constituency analysis is illustrated for Gen 1:3 in Figure 4. It displays units up to the level of the sentence, and contains clause level information on the structure of phrases and their function, without distinguishing between grammatical relations and hierarchical structure. This is solid sentence level tagging without discourse level information.

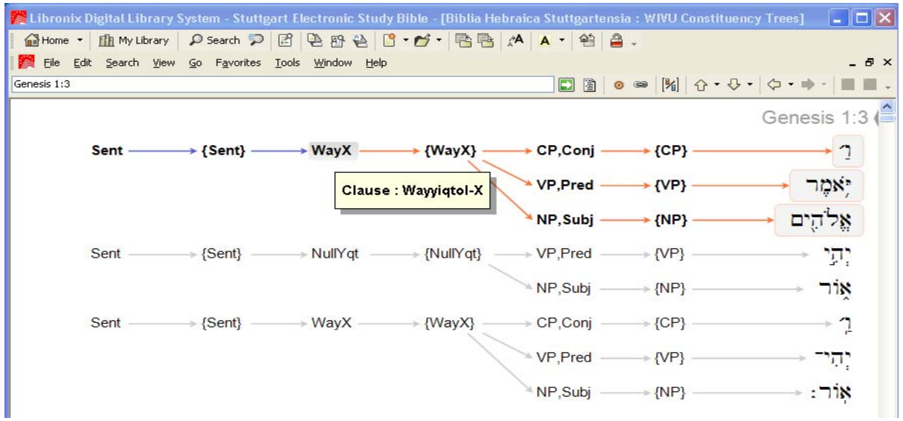


Figure 4 Constituent Structure in SESB (Talstra)

Another syntactic resource is the Andersen and Forbes (2008) discourse interpretation of the Hebrew text exemplified in Figure 5. In their display, all of Gen 1:1-3 is labeled as a top node “Supra-Clausal Structure : Sentence : Discourse” unit. Its immediate constituent is a node called “Clause Immediate Constituent : Time Point : Grammar” with scope over Gen 1:1-2 and this temporal complex sentence governs two clauses in Gen 1:3. The highest ranking node, directly dependent on the highest top node, is the direct speech quoted in Gen 1:3b and labeled as a “Clause (Predication) : Clause : Obliqueness”. The Gen 1:3a clause introduced above is here layered inside these nodes as another oblique. In effect then our *inquit* clause here ceases to introduce the direct speech quoted in Gen 1:3b. All these complex analyses are functional interpretations, and I imagine that many linguists and interpreters of the Hebrew Bible might disagree. Anyway, while SESB perhaps offers too little, Andersen-Forbes perhaps offers too much.

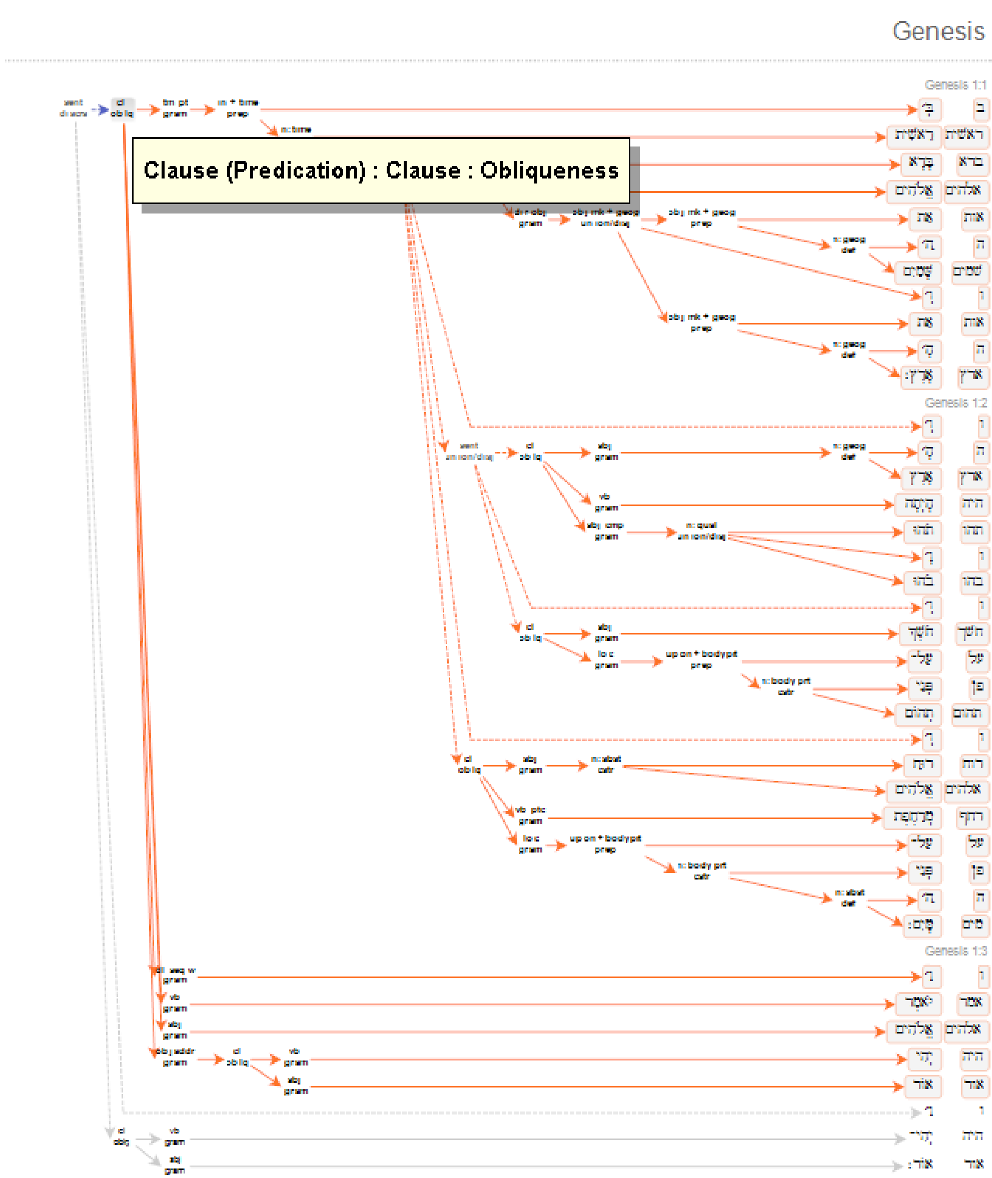


Figure 5 Constituent Structure in Logos (Andersen-Forbes)

All these examples show how limited even the most advanced and sophisticated programs are from a modern linguistic point of view, if you are not a trained Hebraist with competence on the Hebrew script and philological conventions. Even the most advanced and sophisticated commercial software does not meet the need for display of data in a format that can be easily exploited by the general linguist.[8] Inevitably this will restrict the exposure of Hebrew data to the linguistic community, even if a linguist with some skills in deciphering the Biblical Hebrew script could figure out much useful information for typological comparison. The goal of our project is, however, to follow common linguistic conventions and make data available in the standard style used by fellow linguists. Ultimately we hope to be able to integrate our tool and data with a future version of the SESB, provided that we can get funding for continued development of the RLM-tool.

Transliteration from the WIVU Emdros database

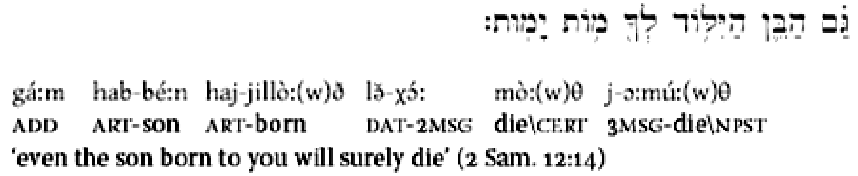
Now, if we accept the job to open the linguistic data of the Hebrew Bible for interaction with fellow scholars in general linguistics, our first task is to be able to display the data in a readable script and a useful format. The RLM-tool is designed to do this by offering selection and display of text and data by using the structural information of the WIVU database already commercially available in the SESB Bible software, and the initial challenge is to transliterate the consonantal text and its vowel points into a useful script.

When we started to move away from the language display habits of Hebrew Bible scholars, we considered to go along with Anstey (2006). His important dissertation explores the Leipzig Glossing conventions, and it presents the first step in his work on a morphological representation of Tiberian Hebrew as shown in Figure 6. In the future it will be interesting to follow Anstey's work on refining the representation of the Tiberian Hebrew of the Hebrew Bible for improved linguistic analysis and description.[9] His project of inventing a new morpho-phonological representation is

[8] Not even the *Biblical Analysis Research Tool* (BART), which I have earlier otherwise recommended for its text-analysis potential (Winther-Nielsen 2005:3 n. 10), has the proper linguistic representation of data or syntax trees, let alone a market distribution. I am not aware of any commercially acceptable solution, and I appreciate the fact that the market for general linguists with interests in the Hebrew Bible is too small. A Web application is a convenient and obvious choice for this kind of scholarly specialization, but electronic publishing or a digital resource for Libronix by the Personal Book Builder technology would be worth while exploring, if it was accessible to the SESB in the future.

[9] In his contribution on HiphiList Astey explains that his first proposal on glossing line was a mixture of orthographic, phonetic, and morpho-phonological information, but in his current

outside the scope of our own work, and his high-quality text representation would not be available for display in our project anyway, nor does it adhere to RRG conventions.


גַּם הַבֵּן הַיִּלּוֹד לְךָ מוֹת יָמוּת׃

gá:m hab-bé:n haj-jillò:(w)ð lɔ̄-χɔ́: mò:(w)θ j-ɔ:mú:(w)θ
ADD ART-son ART-born DAT-2MSG die\CERT 3MSG-die\NPST
'even the son born to you will surely die' (2 Sam. 12:14)

Figure 6 Display of Tiberian Hebrew in Anstey (2006)

We have therefore settled for a more conventional transliteration of data for syntactic analysis. It is also our impression that most linguists will be content with a coarse-grained glossing line exposing the crucial morpho-syntactic information, and we believe that our solution will work well as a useful solution for the syntax-semantics-pragmatics interface of RRG. We also try to address the needs of a broader group of Hebrew scholars and Israeli linguists who may have an interest in our transliteration.[10] It is a compromise between the needs of linguists and the needs of learners, and we hope to benefit both groups by the choices we have made.

The advantage of using the WIVU database is that fellow scholars and students can check our information independently by formulating queries on their own PC through the SESB Bible software and check our display. We only use the full database for linguistic processing and display the data we produce according to the the usual conventions of RRG. By way of example, we know from our database exemplified in Table 1 that the Hebrew word *wayyōˀmer* in Genesis 1:3 contains the lexical morpheme no. 34 of the Hebrew Bible in the EMdF database commonly known as Emdros (Sandborg-Petersen 2008:57 *et passim*), and described very well in Wilson (2009:9, 15-18).[11] The work on Hebrew text and syntax by the Werkgroep Informatica

work "this one line is replaced by three: orthographic, phonetic, and morphophonological. I am also in the process of adding a "morphoprosodic" line for verbs to indicate the prosodic morphology. These are followed by the IMG (interlinear morphemic gloss), adapted from Leipzig."

[10] This was argued in Winther-Nielsen (2008:466). At least some linguists have confirmed my hunch that they will prefer an ASCII representation which is as simple as possible and easy to use for typological work, but others will of course want the precision aimed at by Anstey. Our purpose is to build a framework that will look familiar to linguists as well as students of Biblical Hebrew.

[11] See also www.emdros.org and the presentation of Emdros applications at the *Mini-Conference on Persuasive Database Technology and Applications: Emdros for Learning and Linguistic Analysis* held at Aalborg University * Copenhagen Institute of Technology on March

in Amsterdam inspired Sandborg-Petersen (2008) to construct his Emdros database system as a perfect research platform for linguistic projects. This database technology and its MQL query language is now implemented for the WIVU database in Amsterdam as well as by Kirk Lowery for the Westminster Hebrew Syntax. It is also used in the Logos Bible Software and the Stuttgart Electronic Study Bible (SESB) 3.0.

In our project the WIVU Emdros database presents us with a rich array of information on object types, which are attached to every Hebrew word, and the database also contains information on the higher linguistic levels of phrases and clauses which we do not display here. The program can retrieve all the information we need, such as the graphical script, morphemes, part of speech, tense and much else of crucial importance for an automated representation of the text in the appropriate linguistic format.

The first task for the RLM-tool is to present the Hebrew data in a script familiar to the general linguist, because the graphical words in the database, in this case "J.O71>MER", are hardly useful for display. From the database we have access to the Unicode display of the text as יֹּאמֶר, and we include this surface text for the benefit of the Biblical scholar. This is simply our service to the Biblical scholar who is used to have access to the Hebrew text in commercial Bible software or as open source on the internet. It is a much more difficult challenge to automatically convert these Latin characters into a new system of transliteration that can be read both by students of Hebrew and of linguistics.

Our goal has been to produce a readable and sufficiently distinctive transliteration that will be intelligible to a general linguist and will allow him to interact with the linguistic evidence adduced from the original Hebrew text in question. The technical paper by Winther-Nielsen, Tøndering and Wilson (2009) describe the challenge of transliterating the *qamets* in the closed unstressed syllable as /o/, as well as distinguishing between *shewah quiescens* and *shewah mobile* based on the graphical information in the WIVU database. We also explain how we discussed various transliteration systems with the Swedish scholar Nava Bergmann of the University of Gothenburg who is a native speaker of Modern Hebrew and teaches Modern and Biblical Hebrew. We tested different solutions that would help the linguist to type Hebrew characters on the internet. We wanted all characters to be easily typed from the extended Latin character sets in Microsoft Windows or similar operating systems without installing peculiar Phonetic or Semitic fonts. However, in the end we choose to use the free and widely distributed SIL fonts (http://lex.qwirx.com/lex/index.jsp).

30 2009 (http://www.livssyn.hum.aau.dk/course/view.php?id=19),where our project is placed in a much broader perspective.

Table 1. The database information available for WIVU word no. 34

Word	34		
aramaic_definite_article	Absent	*paradigmatic_nominal_ending*	Not_applicable
Gender	Masculine	*paradigmatic_preformative*	!J!
graphical_aramaic_definite_article		*paradigmatic_pron_suffix*	Absent
graphical_aramaic_definite_article_plain		*paradigmatic_root_formation*	Not_ applicable
graphical_lexeme	>MER	*paradigmatic_verbal_ending*	[
graphical_lexeme_utf8	אמֶר	*Parents*	40777
graphical_locative		*part_of_speech*	Verb
graphical_locative_plain		*Person*	third_person
graphical_nominal_ending		*phrase_dependent_part_of_speech*	Verb
graphical_nominal_ending_plain		*pronoun_type*	None
graphical_preformative	J.O71	*Self*	34
graphical_preformative_plain	י	*State*	None
graphical_pron_suffix		*Stem*	Qal
graphical_pron_suffix_plain		*Suffix*	
graphical_root_formation		*suffix_gender*	None
graphical_root_formation_plain		*suffix_number*	None
graphical_verbal_ending		*suffix_person*	None
graphical_verbal_ending_plain		*surface_consonants*	J>MR
graphical_word	J.O71>MER	*surface_consonants_utf8*	יאמר
Language	Hebrew	*Tense*	Wayyiqtol
Lexeme	>MR[	*Text*	יֹּאמֶר
lexeme_utf8	אמר	*text_plain*	יאמר
lexical_set	Verb_of_quotation	*vocalized_lexeme*	>MR
Locative	Absent	*vocalized_lexeme_utf8*	אמר
noun_type	None	*word_number_within_book*	33
Number	Singular	*wordnet_gloss*	
old_lexeme	>MR[	*wordnet_synset*	0
old_lexeme_utf8	אמר		

Table 2. Transliteration of Hebrew Vowels and Consonants in RLM

CONS	א	בּ	ב	גּ	ג	דּ	ד	ה	ו	ז	ח	ט	י	כּ	כ	ל	מ	נ	ס	ע	פּ	פ	צ	ק	ר	שׂ	שׁ	תּ	ת
	ʔ	b	v	g	g	d	d	h	w	z	ḥ	ṭ	y	k	x	l	m	n	s	ʕ	p	f	ṣ	q	r	ś	š	t	t

VO-WEL	Gen	WIVU No.	Lexeme Encoding	April 2009 Transliteration	BHS Hebrew	Vowel	Bergm. 2005
1	2,21	1073	Y.AL:<OT**@**80**J**W	*ṣ-ṣalʕōtā^{y}w*	מִצַּלְעֹתָיו	ָי	**ā^{y}**
2	1:2	16	H@J:T**@H**	*hāytā^{h}*	הָיְתָה	ָה	**ā^{h}**
3	1:1	4	B.**@**R@74>	*bārā^{ʔ}*	בָּרָא	ָ	**ā**
4	1:1	8	C.@MA73Jim	*š-šāmayim*	הַשָּׁמַיִם	ַ	**a**
5	1:25	488	>**:A**D@M	*ʔ^{a}dāmā^{h}*	הָאֲדָמָה	ֲ	**-^{a}**
6	1:2	26	R74W.**XA**	*rû$_{a}$ḥ*	וְרוּחַ	חַ	**a-**
6	1:6	84	R@QI73J**<A**	*rāqî$_{a}$ʕ*	רָקִיעַ	עַ	**a-**
7	1:2	23	P.:N**;**74**J**	*p^{ə}nê*	פְּנֵי	ֵי	**ê**
8	1:10	70	MIQ:W**;**71**H**	*miqwē^{h}*	וּלְמִקְוֵה	ֵה	**ē^{h}**
9	1:1	3	R**;**>CI73JT	*rē^{ʔ}šît*	בְּרֵאשִׁית	ֵ	**ē**
	3:14	1456	XAJ.**E**75**J**k@00	*ḥayye^{y}xā*	חַיֶּיךָ	ֶי	**e^{y}**
11	1:11	192	<O70F**EH**	*ʕōśe^{h}*	עֹשֶׂה	ֶה	**e^{h}**
12	1:1	12	>@75R**e**y	*ʔāreṣ*	הָאָרֶץ	ֶ	**e**
13	1:2	5	>**:E**LOHI92Jm	*ʔ^{e}lōhîm*	אֱלֹהִים	ֱ	**-^{e}**
14	1:1	3	R;>C**I**73**J**	*rē^{ʔ}šît*	בְּרֵאשִׁית	ִי	**î**
15	1:1	8	C.@MA73Jim	*š-šāmayim*	הַשָּׁמַיִם	ִ	**i**
16	1:2	21	T:H**O**92**W**m	*t^{ə}hôm*	תְהוֹם	וֹ	**ô**
17	9:21	4421		*ʔhlh*	אָהֳלֹה	ֹה	**ō^{h}**
18	1:2	5	>:EL**O**HI92Jm	*ʔ^{e}lōhîm*	אֱלֹהִים	ֹ	**ō**
19	1:29	622	>**@**K:L@75H00	*ʔoxlā^{h}*	לְאָכְלָה	ָ	**o**
20	2:23	1138	LU75Q**:@**X@H&	*luq^{o}ḥāh*	לֻקֳחָה	ֳ	**-^{o}**
21	1:2	26	R74**W.**XA	*rû$_{a}$ḥ*	וְרוּחַ	וּ	**û**
22	1:28	570	KIB:C**U**92	*xivšuhā*	וְכִבְשֻׁהָ	ֻ	**u**
23	1:1	2	B.**:**-	*b^{ə}*	בְּרֵאשִׁית	ְ	**ə**

In the first version of the RLM-tool we used glossing conventions which fellow linguists often use when writing on modern Ivrith (Winther-Nielsen 2008).[12] For the new version we are now replacing the earlier quick-and-dirty experimental transliteration with a system that reflects the text of the Tiberian Hebrew more accurately. Rather than continue to experiment with new character sets we have now decided to follow the published proposal for transliteration of Bergman (2005:1, 14-15, 20). The Hebrew consonants and vowels are transliterated as shown in Table 2 from Winther-Nielsen, Tøndering and Wilson (2009:14).[13]This will be very useful for beginners studying Biblical Hebrew, and the general linguist will still be able to get an accurate impression of how Biblical Hebrew is pronounced and written by Israelis today, even if this system does not distinguish between the transliteration of the allophones גּ/ג, דּ/ד and תּ/ת. Furthermore, this decision will also support our common interest in cooperation on development of e-Learning tools.[14] The beauty of this solution is that users of the Bergman transliteration can copy and paste transliteration into their research papers and teaching material. However, the new system uses contextual rewrite rules for transliteration (Wilson 2009:13-15; Winther-Nielsen, Tøndering and Wilson 2009:6-8), and therefore several different transliterations could be associated with the same corpus of Biblical Hebrew.

Because the RLM-database and any other future Lex language project will use the Emdros database, we hope that our Role and Reference Grammar project will contribute to the technological development of similar language projects and linguistics outside of the Hebrew Bible Studies. Transliteration is an issue for introductory beginners' Hebrew learning, but it may also interest fellow linguists to know that a linguistic corpus stored in an Emdros database format allows teachers to use the data for e-Learning in the LMS system Moodle, following a project by Tøndering to build the *Ezer Emdros-based Exercise Tool* (3ET: http://3bm.dk/index.php?p=82) and Winther-Nielsen's development of the *Bereshit Basic Biblical Hebrew* (3BH: http://3bm.dk/index.php?p=91) implementing 3ET, RLM and other language technology in a Moodle environment.

To sum up, we believe that our proposal for transliteration in the RLM-tool will help linguists use and understand the Hebrew text corpus which has been stored and

[12] This paper describes the challenge of transliterating the *qamets* in the closed unstressed syllable as /o/, as well as distinguishing between *shewah quiescens* and *shewah mobile* based on the graphical information in the WIVU database.

[13] We cannot guarantee that the Hebrew Characters and transliteration will show up in all browsers, but we recommend download and installation of Ezra SIL Hebrew Unicode (http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=EzraSIL_Home) and Charis SIL (http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=CharisSILFont).

[14] See the Berman e-Learning project at http://3bm.dk/index.php?p=81.

annotated in the WIVU database. We hope that linguistic applications will support typological comparison between Hebrew linguistics and many other fields. This account shows how the database can be effectively used for ancient and modern language projects, and scholars may profit from sharing new tools across projects.

Word-level glossing

Transliteration of Hebrew characters is only a small part of the kind of service we owe the broader community of linguists, and a more important task is to display grammatical and semantic information in a readable format. Linguistic terms must adhere to the commonly accepted standards such as the current version of the theory of RRG rather than the traditional philological terms used in the database.

If we did not translate structural data into the format used in RRG, the clause from Gen 1:3 (Figure 2) would be displayed with the traditional grammatical labels, and lexical information would be missing. The clause display would look like example (1) which is a far cry from current linguistic convention, let alone RRG terminology.

(1) וַיֹּ֥אמֶר אֱלֹהִ֖ים

wa-	yyō-	Ø-	?mer-	Ø-	Ø	?ĕlōh-	îm-	Ø
CONJ	Wayyiqtol	(stem)		3ms	SFX	God	mp.absolute	SFX

The RLM-tool is therefore designed to transform the WIVU tags into current linguistic terms, and the linguist must be presented with information on all clauses or clause fragments in the RRG display format. Below the line with text in Hebrew characters in (2)a and in the transliteration from Table 1 in (2)b, we find the glossing line in (2)c. This third line contains grammatical glosses based on the morphological information available in the WIVU-database which scholars can get access to in the SESB Bible software. In our project we first of all translate this information into the appropriate standard notation according to the conventions of the RRG literature (Van Valin 2005: xviii-xxi), to the degree that the database will actually allow us to do so. Furthermore, we try to modulate those distinctions in the Hebrew verb which are most pertinent to argument projection and morpho-syntactic function, i.e. the marking of verbal stem as well as the operator morphemes for grammatical categories like aspect, modality and tense (Van Valin 2005:8). At the current stage of programming in our project we are able to use the database information to display a glossing line for the Hebrew clause in the RRG-friendly format in example in (2)c. For what is traditionally

called *binyanim* or ‘verb patterns’ according to Arad (2005), we use abbreviation labels reflecting the conventional Hebrew terminology, e.g. ‘Qa-’ for Basic stem called Qal, Pa’al, or Pattern 1 in current Hebrew terminology.

(2) a. וַיֹּ֥אמֶר אֱלֹהִ֖ים

b. wa= yyō- Ø- ˀmer- Ø= Ø ʔ^{e}lōh- îm= Ø wa=

c. CLM NARR Qa say 3Msg CLT god MplAB CLT CLM

d. *And God said,* (Let there be light:..).(Gen 1: :3)

One of the advantages of this display is that we are able to present some of the fundamental facts of Hebrew for the Hebrew learner as well as the linguist in a pedagogical manner. The *wa=* is a special form of the Hebrew clause marker (CLM) which does double duty as a prefix before the Hebrew verb in one of four finite verb forms which we call the serial narrative conjugation (NARR-).[15] The Hebrew verbal stem system can be marked by a separate consonant in a few forms, but is mostly umarked, and this fact is indicated by either zero (Ø-) or some prefix-morpheme. This is followed by the lexical kernel of the verb, e.g. *ˀmer.* It is followed by two sets of suffixes. The person, gender, number agreement suffix, which here is Ø for 3Msg, serves as the first argument in a pro-drop language for actor indexation (Winther-Nielsen 2008:469). Optionally the direct core argument can be pronominalized into a suffix on the verb, which here is labeled as a clitic (CLT) as proposed by Anstey (see note 15).

In the first version of the RLM-tool from January 2007 we used WordNet glosses for the Hebrew Bible, and this approach was discussed at length in Winther-Nielsen

[15] In the scholarly discussion on HiphiList Dr Matthew Anstey has raised several important issues concerning the glossing proposed here. In his view a more consistent glossing in terms of our project would be as follows:

w-ay-y-ō-ˀmer ʔ^{e}lōh-îm

CR-NARR-3MSG-Qal-say god.S-MPL

Following this finer grained analysis we could adopt *wa=y-yō-ˀmer* as the best solution within the WIVU framework. The initial *wa=* serves both as a marker of one particular serial coordination (NARR), but also retains its nature as an independent lexical item, the clause linkage marker (CLM). We follow Anstey and assume that the headmarking object suffixes on the verb and possessive suffixes on the noun are clitics. Anstey objects to the Ø-markings in this model, but we believe that the Ø for absence of a consonantal marking can be helpful for pedagogical reasons, and it works well in our use of the database.

(2008). However, since there is at present still no mapping from a dictionary to WordNet senses beyond Genesis 1-3, we have decided to move ahead without the time consuming task of manually entering WordNet glosses. This would be a project of its own merit, but it exceeds the primary focus on Hebrew RRG in our work at the moment. However, the WordNet entry technology could still be made available in the RLM-tool if we were at some stage to undertake a complete labeling of the Hebrew Bible to match the universal definitions available universally for many languages. Such work would require funding from outside sources which is not available for our project so far.

In the current version we now choose an approach which has been made possible thanks to our cooperation with the WIVU project in Amsterdam and the German and Dutch Bible Societies producing the SESB. As of April of 2009 we have permission to use the dictionary of Bosman, Oosting and Postma (2003) for automated lexical display. This dictionary is a German and English wordlist supplied commercially in the SESB program. The tool will automatically select the first entry from the WIVU display (Wivu) shown in Figure 7, e.g. the tool here glosses *ʔ*ᵉ*lōhîm* as "god", but the analyst may want to enter "God" as his preferred choice. Above this row there is a line where the analyst can enter his own choice of gloss for a particular word, and this choice will be used everywhere for this word in all of the Hebrew Bible as the default sense. Below this are meanings automatically inserted from the King James Version (KJV) based on word comparison.

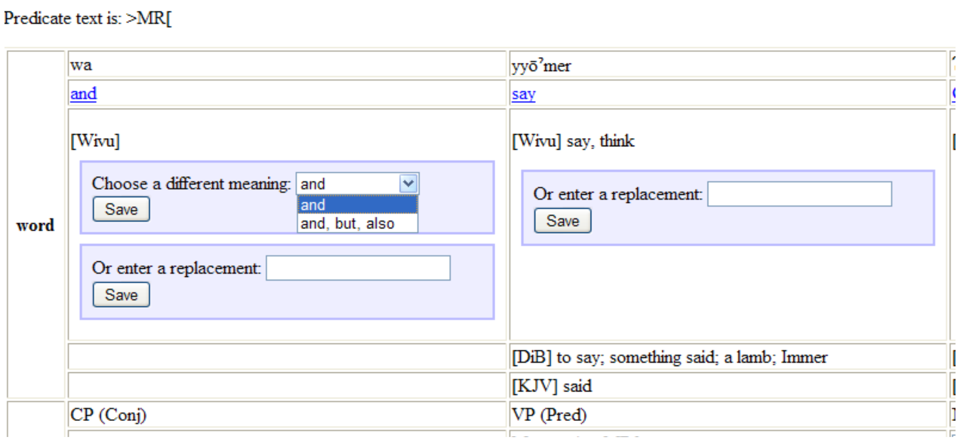


Figure 7. Selection of lexical information for display

Table 3. The DiB entries for the Hebrew root ʔmr (slightly simplified)

A	**StrongsNo**	**559**	**560**	**561**	**562**	**563**	**564**
B	Language	Hebrew	Aramaic	Hebrew	Hebrew	Aramaic	ProperN
C	TWOT	TWOT-118	TWOT-2585	TWOT-118a	TWOT-118a	TWOT-2585?	
D	Form	Verb	Verb	Noun Masculine	Noun Masculine	Noun Masculine	Proper Name Masculine
E	GkRelated	G114 G138 …		G2293 G2917…	G2981	G296	
F	Fuller Meaning	1) to say, speak, utter 1a)….	1) (P'al) to say, to speak, to command, to tell, to relate	1) utterance, speech, word, saying,	1) utterance, speech, word, saying, promise, command	1) lamb	Immer = "he hath said" 1) a priest in David's time 2) a priest …
G	**Unpointed Heb**	**אמר**	**אמר**	**אמר**	**אמר**	**אמר**	**אמר**
H	CALUnpointedAscii	)mr	)mr	)mr	)mr	)mr	)mr
I	TABSUnpointedAscii	AMR	AMR	AMR	AMR	AMR	AMR
J	Pointed Heb	אָמַר	אֲמַר	אֵמֶר	א ֹמֶר	אִמַּר	אִמֵּר
K	Transliteration	amar	amar	emer	omer	immar	Immer
L	Phonetic	aw-mar'	am-ar'	ay'-mer	o'-mer	im-mar'	im-mare'
M	Notes	a primitive root;	(Aramaic) corresponding to #559 …	from #559 …	the same as #561	(Aramaic) perhaps from #560	from #559 אָמַר
N	**Meaning**	**to say**	to say	**something said**	something said	**a lamb**	**Immer**
O	Full Meaning	to say (used with great latitude)	{to say (used with great latitude) }	something said	{something said }	a lamb	Immer, the name of five Israelites
P	Translation InAV	answer, appoint, avouch, bid, …	command, declare, say, speak, tell	answer, [idiom] appointed unto him…	promise, speech, thing, word.	lamb.	Immer.

At an earlier experimental stage we used an open source dictionary.[16] The information for DiB derives from the Microsoft Excel file hebrewDiB.xls, and is listed in excerpt for all Strong Numbers (559)-(564) in Table 3. This dictionary presents six meanings, four in Hebrew and two in Aramaic. For this particular root the program will collect the meaning specified in row N for the Hebrew verb in column (559), the nominal meaning in column (561), an Aramaic meaning in column (563) and a personal name in column (564). The linguist can use the RLM-tool to enter a new gloss based on this dictionary information and the King James translation, or he can rely on his own linguistic expertise.

When these facilities are explored for a consistent and canonical display we believe that the RLM-tool will offer useful linguistic data for typological research. It will ease linguistic work considerably, because it offers “glossing ready to go”, so to speak. This means that we can provide online parsing accessible through the internet, and linguists will by a simple copy-and-paste procedure have access to machine-consistent glossing to share within the linguistic community, and data can be hyperlinked or printed. The non-Hebraist linguist will be able to exploit our Hebrew corpus in a far more efficient way by consulting the corpus online, and he will have direct access to the morpho-syntactic features of Biblical Hebrew.

Finally, we assume that the RLM-tool will be useful in courses on advanced linguistic Hebrew as a resource to illustrate the solutions offered by RRG. This technology can be expanded to other languages and modern textual corpora with Latin characters or peculiar scripts. We therefore expect this transliteration and glossing technology to be of great value for linguistic work in RRG and other languages.

Clause constituents in “node-box” display

This introduction to transliteration, glossing and lexical description in progress paves the way for our main topic which is to discuss the parser of the RLM-tool developed by Wilson (2009). The parser takes a big step towards a computational processing of the morpho-syntactic features of Biblical Hebrew and display of the syntactic structure of individual clauses.

RRG is built around a smooth bidirectional linking between the representation of syntax and semantics within a discourse-pragmatic framework (Van Valin 2005:2).

[16] Ulrik Sandborg-Petersen shared with our project a program he developed when he produced a Greek-English dictionary based on the open source material from Crosswire at http://crosswire.org/~scribe/greekheb/hebrewDiB.xls. The main source for online tools is www.crosswire.org and the Sword project which is supported by the Society of Biblical Literature and includes the work of scholars.

Wilson has solved the programming challenge to implement this non-configurational and surface structural approach, and in the following I will explain how this works for Biblical Hebrew. The RLM-tool is designed to work with a stored corpus and must therefore work from syntax to semantics, since it is it is not meaningful to explore the semantics-to-syntax linkage for a dead language like Biblical Hebrew - who would be able to check its felicity, anyway? The new reinvented Modern Israeli, or Ivrit, cannot be used for semantics-to-syntax processing of the Hebrew Bible, since because it has a significantly different morpho-syntax. However, as a canonical corpus, and fixed for millennia, there is an important point in syntax-to-semantics processing, because it will allow us to represent Hebrew in a meta-language that can be used for translation and for corpus linguistic experimentation. A database of the corpus functions as a competent informer of what has been generated and how frequent linguistic items are used.

Table 4 Phrase function information in the WIVU database

Accordingly, for the syntax-to-semantics linking algorithm we need to determine the macroroles and other core elements of the clause as our first step (2005:149). From the WIVU database we have access to information on the clause functions of traditionally labeled constituents like conjunction “CP (Conj)”, verb phrase “VP (Pred)”, and noun phrase “NP (Subj)” displayed in

Table **4**. The RLM-tool uses this phrase structure information for automated display and selection of semantic roles.[17] From the database we could have used information on clause constituents on the phrase level, but in order to test and develop a parser we only use the morphemes as input for rule-based parsing. We even at this stage do not automatically calculate macrorole-information which could otherwise easily have been deduced from the WIVU database. However, at a later stage we may want to speed up the semantic analysis considerably by translating the “NP (Subj)” into actor and “NP (Do)” into undergoer for automated macrorole selection and fast

[17] Chris Wilson has developed the tool to use phrase_dependent_part_of_speech to generate the gloss lines, to choose base node types for parser input (e.g. "noun" and "proper_noun" have N/NUC, N/GNS and N/POS nodes, "conjunction" becomes "CR") and to automatically assign default macroroles and arguments.

forward RRG parsing for all of the Hebrew Bible. This would support online display of the entire Hebrew Bible for automated processing jobs.

The complete constituent projection options for RRG would look like Figure 8. In the layered structure of the clause in RRG, a SENTENCE has a CLAUSE as its constituent, and a CORE consists of the Nucleus (NUC), which is the Predicate (PRED) and is often expressed as the verb (V), as well as one or more optional arguments (ARG) which can be expressed syntactically by the noun phrase (NP), the preposition phrase (PP) or an adverb (ADV). Following the theory of the 'layered structure of the noun phrase', the NP embeds a nominal core ($CORE_N$) which often is a noun (N). We will explain the details by examples from our corpus.

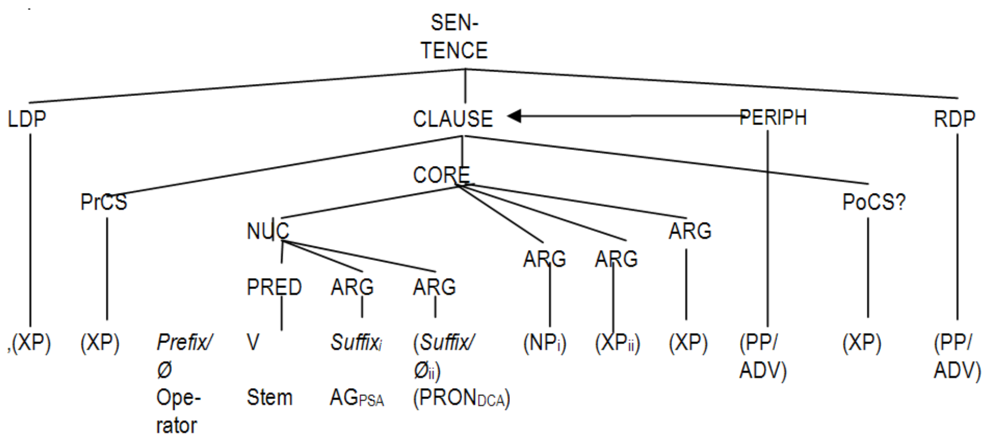


Figure 8 The constituent projection for Hebrew in RRG

Once all syntactic constituents have been parsed, a linguist in a true RRG-linkage system should have the option to assemble rules into the appropriate syntactic templates, and this would provide us with the true syntactic tree projection according to the principles of RRG. The preliminary solution at the moment is to offer a pseudo-RRG-projectionist representation of nodes as boxes, and it uses rewrite rules in a parser. All nodes and branches that are part of a complete parsing of all constituents are displayed in the format of Table 5 where boxes emulate the nodes of a canonical RRG tree stripped of branches.

In order to explain how this "node-box" display is created we need to understand the use of rewrite-rules in our parsing (Wilson 2009). The parser is initially set up to read lexical and grammatical morphemes with their functions as stored in the WIVU-database. The RLM-tool automatically labels this information with the appropriate RRG-terminology, and afterwards the linguist can select each morpheme as input in a

rewrite rule. For the glossing line in (2)c we have already explained the finite verb forms, or conjugations, which are marked by the presence or absence of the verbal prefix, as well as the two single and two serial conjugations in Biblical. For RRG parsing we assume that the prefixes and stems are part of the constituent structure, and therefore tense, mood and aspect morphemes as well as the *Aktionsart* stem prefix must be part of the operator projection (Van Valin 2005:8-12). The prefixes are therefore not included in the constituent projection.

Table 5 A display of a syntactic tree in "node-box" format

Parse finished with 1 possible trees. Showing the first 1:

<table>
<tr><td colspan="9">SENTENCE</td></tr>
<tr><td>CONJ</td><td colspan="8">CLAUSE</td></tr>
<tr><td>wa</td><td colspan="8">CORE</td></tr>
<tr><td></td><td colspan="5">NUC</td><td colspan="3">NP</td></tr>
<tr><td></td><td colspan="5">PRED</td><td colspan="3">$CORE_N$</td></tr>
<tr><td></td><td colspan="5">V</td><td colspan="3">NUC_N</td></tr>
<tr><td></td><td colspan="3">V_{Stem}</td><td>AG_{PSA}</td><td>$PRON_{DCA}$</td><td colspan="3">N</td></tr>
<tr><td></td><td>V_{TAM}</td><td>V_{STM}</td><td>V_{NUC}</td><td>Ø-</td><td>Ø</td><td>N_{NUC}</td><td>N_{GNS}</td><td>N_{POS}</td></tr>
<tr><td></td><td>yyō-</td><td>Ø-</td><td>ʾmer-</td><td></td><td></td><td>ʔ□lōh-</td><td>îm-</td><td>Ø</td></tr>
</table>

Click here to see and edit all 1 parses.

In the parser this is at present handled with rule no. (736) for verbal stem production in Table 6: From the WIVU morpheme features of prefix for aspect {V/TAM}, stem {V/STM} and verbal kernel {V/NUC} the linguist can create a rewrite rule declaring that these three morphemes combine to form a V/Stem, which is that part of the nuclear verb which does not primarily mark congruence and arguments. In RRG the predicate is the nuclear element in the layered structure of the clause, and its innermost element is the nucleus of the verb (V_{NUC}). A constituent projection node would have to go through the V_{NUC} which is the lexical kernel and the surface expression of the lexeme. This kernel expresses the Hebrew root which is often used as a synonym for the dictionary word, or the lemma.

Table 6 Parser rules for the Hebrew verb

642	NUC	{PRED}	Edit
556	PRED	{V}	Edit
563	V	{V/Stem} {AG/PSA} {PRON/DCA}	Edit
736	V/Stem	{V/TAM} {V/STM} {V/NUC}	Edit

After this preliminary step, the linguist must write a rule which will produce a syntactic projection of the Hebrew verb based on his analysis of Biblical Hebrew as a dependent-marking language with considerable head-marking features in its morpho-syntax (Winther-Nielsen 1995: 43-44. Van Valin 2005:16-19). The obligatory person, number and gender suffix, indexes what other theories often refer to as the "Subject", but in RRG terminology it is referred to as agreement of the verb with the 'privileged syntactic argument' (AG_{PSA}) (Winther-Nielsen 2008:468). Often this obligatory AG_{PSA} suffix will involve a double marking for the PSA, because an explicit lexicalized NP can be specified for discourse-pragmatic reasons of introduction, for resumption of a referent, or for textual disambiguation or some other pragmatic concern. Optionally, the 'direct core argument' (DCA) – in other theories known as the "(direct) object" – can be pronominalized as a $PRON_{DCA}$ clitic. Because the second argument almost always is marked either by pronominal inflection or by an explicit NP in the function of a direct core argument we categorize Biblical Hebrew as a pro-drop language similar to Spanish or Croatian rather than as a canonical head-marking language like Lakhota. This analysis will allow the linguist to build a verbal Nucleus node for Biblical Hebrew syntax: {V/Stem} + {AG/PSA} + {PRON/DCA} → V (663) is first created, then {V} → PRED (556), and the last rewrite rule {PRED} → NUC (642) completes this node.

The linguist can produce this syntactic analysis through the *Parser* link in the Navigator. The chart parser will generate the parsing output of all rewrite-rules which the linguist writes for his syntactic analysis, and for the Hebrew verb the rules discussed above are shown as pink coloured parts when the curser is moved to the NUC-box in Figure 9.[18] When clicking in one or more boxes, the node name in question will be entered as "Component node names (parts)", or an immediate constituent. The linguist can then label this grammatical feature by a node name adhering to the conventional labeling of constituents in RRG. As illustrated in this figure, the {NUC} constituent may serve as an immediate constituent of a core node if

[18] In this figure the following sigla are used: verb tense, aspect mode=V_{TAM}, verbal stem=V_{STM}, verb nucleus=V_{NUC}, obligatory PSA person, number and gender agreement suffix=$PRON_{PSA}$, and optional pronominalized direct core argument=$PRON_{DCA}$.

it occurs as the sole constituent of a clause, and we can then type "CORE" on the left hand side as "Top node name (symbol)" to label the nucleus as it constituent part. For now, this rule will be used by the parser and be displayed until changed or deleted. This display opens for an inspection of the output of all rewrite-rules, including the failed ones which did not result in a complete parsing of a clause. The linguist can write rules and test their output, and then change and improve until he has got his RRG analysis completely right. This output of all parser rules is available as an option referred to as "Click here to see and edit all 1 parses" underneath the "box-node" display or it is accessible directly from the *Parser* link in the tools menu.

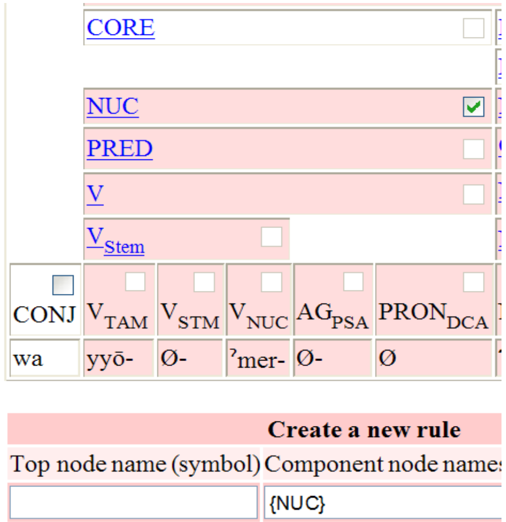


Figure 9 Output of parser rules

As for the privileged syntactic argument in Table 5, the divine name *Elohim* is analyzed as composed of the nucleus of the noun (N_{NUC}) *ʔᵉlōh-* followed by an *-îm* suffix for gender, number and status (N_{GNS}) which is glossed as masculine plural in the absolute (MplAB). In traditional Biblical Hebrew grammar the term 'absolute' is used for the form of the last modifying nominal element in a phrase with one or more preceding nominals functioning as the modified head(s). The head nominal is traditionally labeled the 'construct', and this head noun is modified by a following dependent adjunct noun which in turn together with its head may optionally serve as a new compound head for a second adjunct. Recursion can go on in several nouns, until the end of this kind of nominal compounding is reached. This final noun is in many cases morphologically marked as the absolute form, at least when there is a choice in form. Furthermore, possessive pronouns like *my, your, hers* and *there* in Hebrew are enclitic pronouns attached to nominal or prepositional heads. Instead of a modifying adjunct noun, a head noun can be inflectionally modified by pronominalized possessor suffixes on the noun (N_{Pos}). This is used to write parser rues for the NP node.

In this way the morpheme labels are read automatically from the database and they provide the basis for the linguist's writing of re-write rules for parsing. The RLM-tool helps the linguist write rules in a bottom-up fashion as shown in Figure 10 where the true rules are highlighted. It lists the linguistically correct rules that will result in a successful parsing of a sentence, and therefore qualifies as the only one – or one of a few – syntactically correct readings to be displayed by the browser in the "node box" format. However it will also list output of rules that will not produce a succesful parsing, and it is the job of the linguist to figure out the appropiate number and formulation of the fewest and relevant rules which results in a correct projection.

The boxes can be compared to crude visual representations of lines in the nodes of syntactic trees that can be created by the linguist in a gradual trial and error fashion. The linguist can test the appropriateness of his experimental roles, and he can refine and test again. Hovering the curser on a "node" in this display will highlight the output under the scope of this particular parsing rule. He can evaluate rules at work in random text and continuously improve, as new clauses pose new challenges. After some helpful learner experience and experimentation it was possible to create a set of rules for Hebrew that would produce the correct projection for all simple clauses. The parser produced the ambiguities we would expect, and it failed where it should – like in clause fragments, embeddings, ellipsis and the like. The right kind of syntactic trees are projected from rules that rely on the layered structure of the clause and database information on word class, and this system leaves semantic and pragmatic aspects of clause analysis to the other appropriate domains of the grammar. The parser does not rely on the dictionary or word senses, let alone various diverse pragmatic packing of the clause. Those issues are left to an investigation of the relational structure and the information structure.

RRG is a triple-projectionist theory of grammar (Van Valin 2005:170), but at this stage we do not plan to display the operator or focus projections. This would require work on design, which would be outside the scope of our project, but we believe that we can offer a useful way to build and display constituent structure. Even a pseudo-RRG rule-based processing without recourse to a true syntactic inventory can lead us to a single or a few possible parsing outputs, and predictable failures, true to the natural language data of Biblical Hebrew as well as the conventions in RRG. How this works we will explain by examples from our corpus.

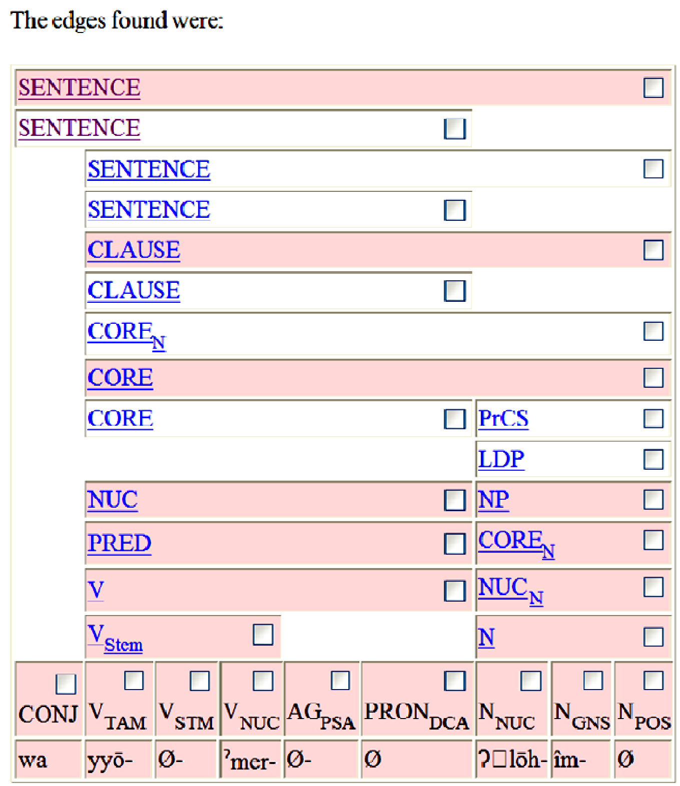


Figure 10 Output of Chart Parser for Gen 1:3

Parser rules at work in Genesis 1-3

The next step is to exemplify the parsing *Rules* from actual examples in the Hebrew Bible. We will introduce and elaborate on the very first screen in Figure 11, which opens when a guest visits our RLM-application and is exposed to the first sentence of the Hebrew Bible from the first chapter of the Book of Genesis.

As hinted at earlier in our discussion of Gen 1:3 (Figure 2 and Figure 3),[19] the

[19] The only slightly odd syntactic description for this clause is the overall problem contained in the Amsterdam database that the direct argument marker *ʔēt* is not distinguished from the preposition proper at the morphological level, and therefore both are at the moment glossed by P for preposition. We hope to be able to correct this display function by future programming,

first few clauses of the Hebrew Bible are not simple ones. The opening clause in Gen 1:1 is one of those cases where the program correctly registers two possible solutions from the structural data alone. In our non-graphical "node box" display of the two syntactic trees in Figure 12 there is one difference between the first and the second parsing pertaining to the preposed prepositional phrase *bᵊrēʔšît* 'in-beginn-ing' placed in front of the verb of creation. This first PP could on purely syntactic grounds be interpreted as either a 'pre-core slot' (PrCS) or a 'left detached position' (LDP), and this is the one difference between the two trees displayed.

Figure 11 The text display of part of Gen 1:1

In this case we will only be able to decide from semantic and pragmatic criteria, which in a functional and discourse-pragmatic adequate theory like RRG determines the correct interpretation of the constituents of the non-relational structure of the clause. Semantic and pragmatic considerations count when we need to explain the role of an element expressed in a position before the clause. In the case at hand, the preposed PP provides extra temporal information as a point of departure for the opening text of the Hebrew Bible and is thus a discourse-pragmatic feature. Furthermore, on semantic grounds alone, we know that this temporal information is not part of the inherent meaning of the clause as an argument of the verb. Finally, everything inserted in this sentence slot is a brand-new information, since no previous discourse is available in the opening of a new work, and it serves as the temporal

but have retained it in order to reflect the nature of the database and the challenges we solved automatically by programming at this point.

background of the following clause and has a sense like 'At an early time' (cf Winther-Nielsen 1992). The decission for a topicalized LDP setting element rather than a fronted PrCS is uncontroversial.

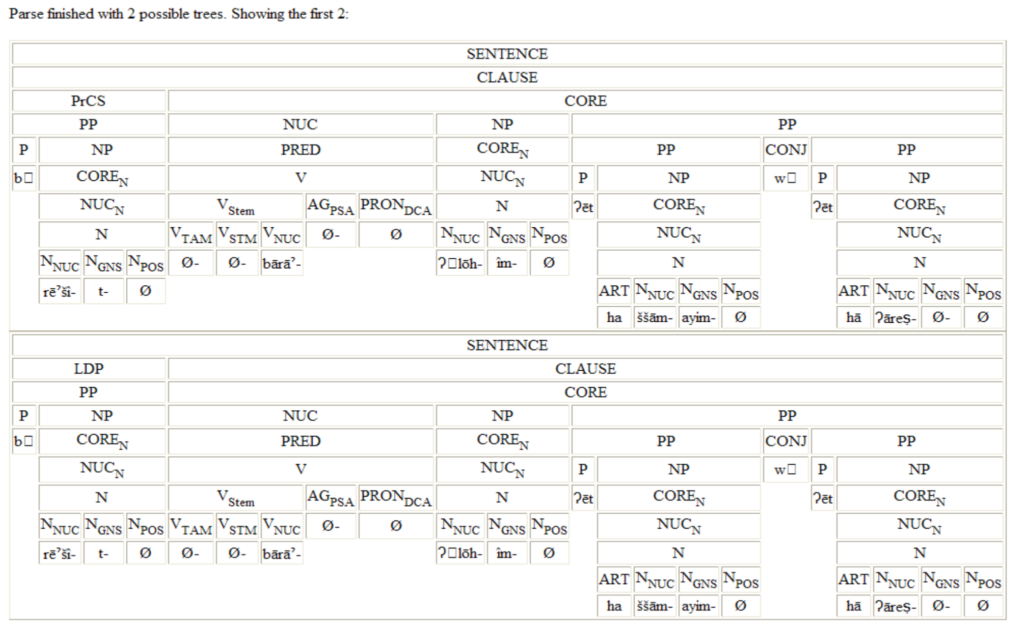


Figure 12 Screenshot of 2 "node-box" displays for Gen 1:1

We can explain the constituent projection for Hebrew in RRG by pedagogical examples in English to illustrate the choice of the LDP in Gen 1:1 in the light of the at least in principle other extra-Core positions available (PrCS, PoCS, and RDP) in example (3).

(3) Special positions in constituent structure for English

a. LDP In the beginning, God created the heaven and the earth
b. PrCS * In the beginning God created the heaven and the earth
c. PoCS ? He created the heaven and the earth, God, in the beginning,
d. RDP He created the heaven and the earth in the beginning, God

To understand the nature of the "node-boxes" in this kind of output display from the parser, we will now list the rules producing the nodes in our pseudo-projection tree. The topmost node is the sentence in example (4). The constituents of this node are clauses with or without the CLM, but it can also be preceded by the topicalized

LDP as in the case of Gen 1:1, or a constituent from the core of the clause can be fronted into the PrCS.

(4) The sentence node

a.	SENTENCE	{CLAUSE}
b.	SENTENCE	{CONJ} {CLAUSE}
c.	SENTENCE	{LDP} {CLAUSE}
d.	SENTENCE	{PrCS} {CLAUSE}
e.	SENTENCE	{CONJ} {PrCS} {CLAUSE}

Among the LDP rules in example (5), a PP constituent most often is the preferred phrase in Hebrew for the morpho-syntactic expression of a temporal, locative, or manner constituent, but we expect that an NP may also in rare cases occur in the detached position of the sentence.

(5) The LDP node

a.	SENTENCE	{LDP} {CLAUSE} (repeated from (4))
b.	LDP	{NP}
c.	LDP	{PP}

The opposite tendency is to be expected for the PrCS in the rules in example (6). One of the macroroles from the core will be fronted as a NP placed in front of the CORE. Because Hebrew often marks the direct core argument as a PP, this constituent can also be fronted, and so can an 'oblique core argument' (OCA).

(6) The PrCS node

a.	CLAUSE	{PrCS} {CORE} (repeated from (7))
b.	PrCS	{NP}
c.	PrCS	{PP}

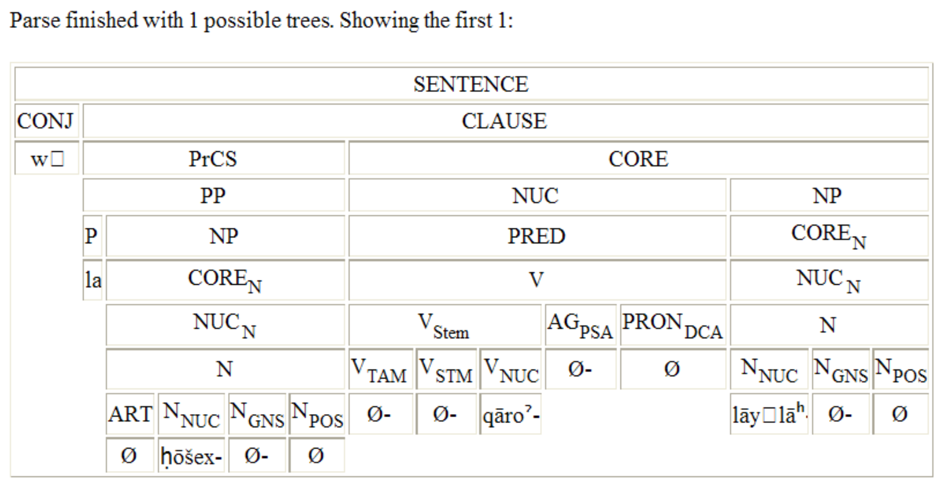


Figure 13 Screenshot of "node-box" display for Gen 1:5

The first simple fronted PrCS-constituent occurs in Gen 1:5 in Figure 13. For lack of space we will from now on not quote the linguistic glossing, but for every clause the linguist can use the navigator of the RLM-tool to inspect the glossed linguistic data online at our site, or easier yet, use the hyperlink provided in the accompanying notes.[20] The box display of the only successful parsing shows that in this case a contrastive focus is marked by preposing a DCA from the second pair part of two conjoined clauses. By implication from the discourse, this results in an adversative implicature: "but the darkness he called night".

The topmost part of the clause is projected by the rules in example (7). The most important node of the clause is the CORE which can be modified by a following additional predicating PERIPHERY with setting information on time, place and manner (cf Figure 15).

(7) The clause node

a.	CLAUSE	{CORE}
b.	CLAUSE	{CORE} {PERIPHERY}
c.	CLAUSE	{PrCS} {CORE}

The CORE node rules in example (8) at present are restricted to a small number,

[20] See http://lex.qwirx.com/lex/clause.jsp? book=Genesis&chapter=1&verse=5&clause=28748: In this and subsequent cases the linguist can enter this link into his browser in order to have the data displayed or to copy and paste from such data.

but Hebrew no doubt has more constituent configurations than this. However, the more constituents we propose, the more possible readings are presumably generated in output, but for the moment I try to keep the rules as few and simple as possible, and then only expand the rules cautiously, in order to leave the rest to the semantic and pragmatic domains of the grammar where they belong.

(8) The core the clause

a.	CORE	{NUC} {NP}
b.	CORE	{NUC} {NP} {PP}
c.	CORE	{NUC} {PP}
d.	CORE	{NUC} {NP} {PP} {NP}

The layered structure of the noun phrase calls at least for the rules in example (9). We need to be able to handle recurring elements such as the two NPs joined by a conjunction in Gen 1:1. Furthermore, the parsing of genetival constructions in Hebrew is a challenge. Complex noun phrases are formed around a head noun which in some forms is marked by a suffix to indicate that it is modified by one or more adjuncts as explained above for the construct and absolute forms. Examples like the NP *pᵊn-ê* ‘face-of’ and *tᵊhôm* ‘ocean’ in Gen 1:2 (Figure 14) were difficult to parse, but after some experimentation it became evident that a rule that allows for the CORE_N (as the “construct form”) to be modified by an embedded NP (as the “Absolute form”) will actually work in many cases, and this is the solution implemented in the RLM 2.0.[21]

(9) The layered structure of the NP

a.	NP	{NP} {CONJ} {NP}
b.	NP	{CORE/N}
c.	NP	{CORE/N} {NP}
d.	CORE/N	{NUC/N}
e.	CORE/N	{N} {NP}
f.	NUC/N	{N}

So far we have not mentioned a specific kind of clause which we will call the verbless clause, but nominal clause is another popular traditional term among Hebrew

[21] Wilson suggests that for future work it should be possible to write rules using the support for features and unification in the parser, maybe with some extension, rather than have a recursion where a CORE/N can contain an NP that itself contains a CORE/N, i.e. rules would look like this:

NP := {CORE/N.state=construct} {CORE/N.state=absolute}

Bible scholars (Winther-Nielsen 1995:36). This clause type covers the attributive, identificational, specificational and eqautional thematic relations, as well as the pure location structure (Van Valin 2005:55). It is used in Figure 14 for *ḥōšex ʕal-pᵊnê tᵊhôm* ‘darkness on-surface-of tehom’ which has the logical structure **be-on´** (x,y). In order to parse such clauses successfully I had to come up with a special tagging of a nominal predicate which is listed in example (10), but I have not included the rules for the use of NPs and adjectives as nominal predicates.

(10) The verbless clause node

a.	SENTENCE	{CONJ} {CLAUSE/Vbl}
b.	CLAUSE/Vbl	{NP} {NUC/PP}
c.	NUC/PP	{PRED/PP}
d.	PRED/PP	{PP}

The solution with a prepositional phrase as a predicate of a verbless clause is shown with the correct parser output in Figure 14.[22]

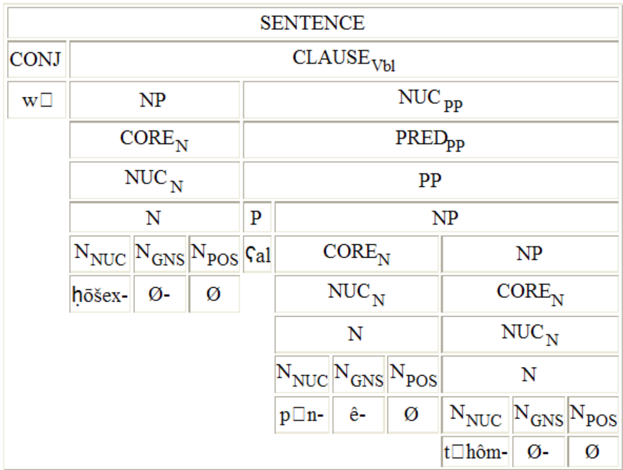


Figure 14 Screenshot of ”node-box” display for Gen 1:2

[22] See http://lex.qwirx.com/lex/clause.jsp? book=Genesis&chapter=1&verse=2&clause=28739. At this point I will not discuss the challenging problem of word order in the Hebrew Nominal Clause.

Outside of and after the core, RRG posits a PERIPHERY which is not an argument of the core, but instead is an adjunct which provides additional information on all the rest of the clause (Van Valin 2005:23). In most cases the pheriphery is a prepositional phrase, but since it is not part of the relational structure of the core with predicate and arguments, it is treated as an independent constituent in the layered structure of the clause. This kind of adjunct to the clause has a predicative function and must therefore have a constituent projection on its own. The rules for this type of PP is given in example (11).

(11) The periphery PP node
  a. PERIPHERY {PP/ADJT}
  b. PP/ADJT {CORE/P}
  c. CORE/P {NUC/P} {NP}

This is the correct and sole solution given by the parser for Gen 1:27 which has an additional manner element (Figure 15).[23]

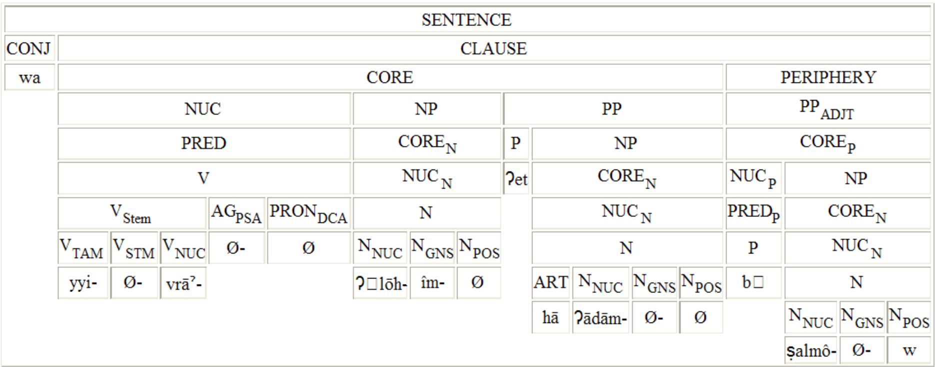


Figure 15 Screenshot of "node-box" display for Gen 1:27

[23] See http://lex.qwirx.com/lex/clause.jsp?book=Genesis&chapter=1&verse=27&clause=28739

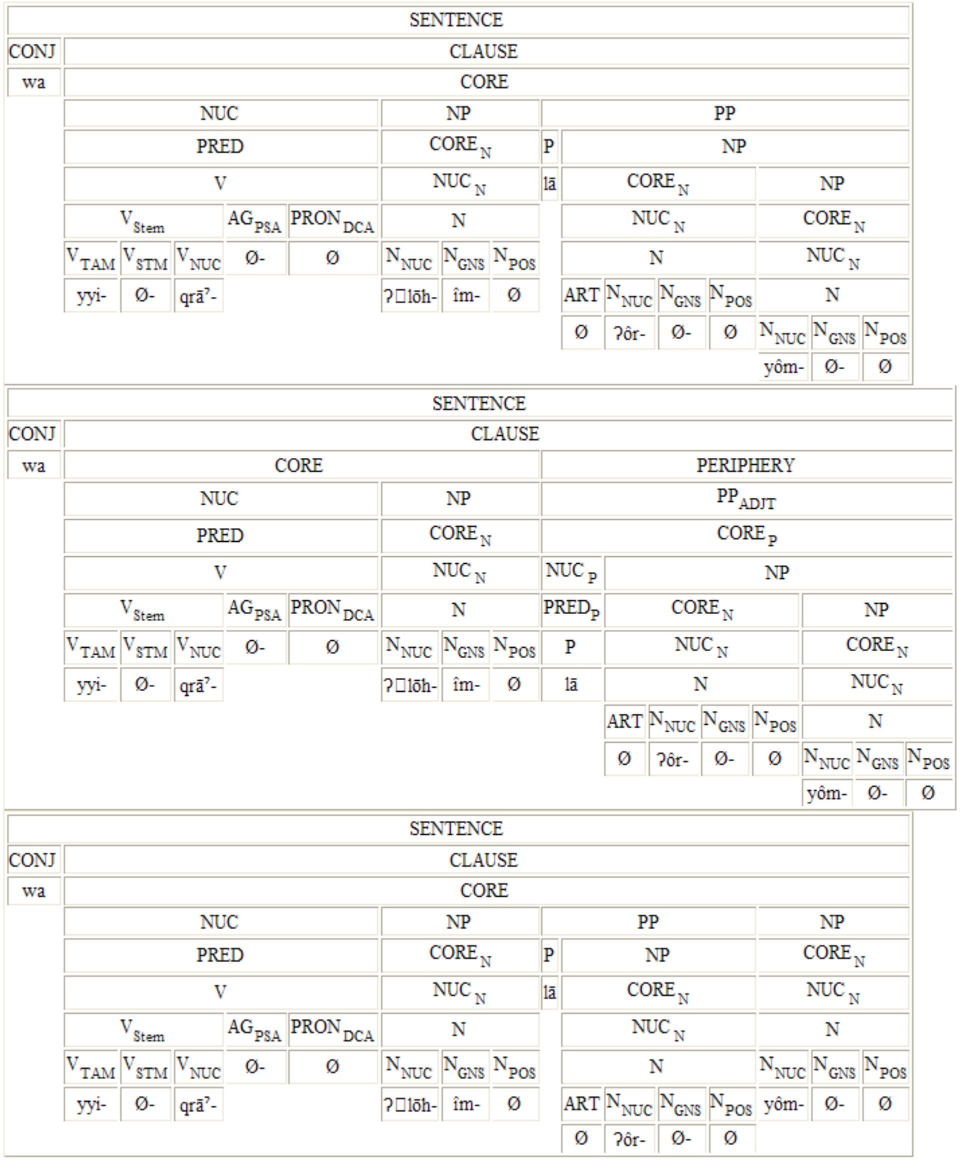


Figure 16 Screenshot of "node-box" display for Gen 1:5

RRG distinguishes this periperal adjunct from a prepositional phrase used within the core and generated by the rules in example (12). In this case a preposition governs an NP and modifies it.This constituent is treated as part of the relational structure of the clause with its predicates and arguments. This kind of entity is represented by *lā-ʔôr* ‘to the light’ (Gen 1:5) in the third solution in Figure 16. Here the first two solutions show the problem with the genitival construction mentioned above. On purely morpho-syntactic grounds the noun *yôm* ‘day’ can mistakenly be treated as part of a construct relation with *lā-ʔôr* and again from a purely structural point of view this PP can be treated as either a core argument or a periphery adjunct. Only when I wrote a core rule which allowed for the three arguments {NUC} + {NP} + {PP} + {NP} → CORE did the correct third solution appear as a possible parsing.

(12) The core PP node
  a. PP {PP} {CONJ} {PP}
  b. PP {P} {NP}

The case presented in Figure 16 is in my mind an important point in favor of the theory of RRG.[24] It very well illustrates the need for a grammar to handle the distinction between a semantically motivated relational PP argument and a pragmatically motivated non-relational PP adjunct. Without resort to semantic information in the lexicon of the logical structure of the verbs, this important distinction between parts of the layered structure of the clause and verb dependent OCAs cannot be handled by the parser’s output of the syntactic projection.

The adjective is another modification of the layered structure of the noun phrase. In this case we need rules which stipulate that the nucleus of the noun (NUC_N) can be modified by an adjective to form a periphery in the core layer of the noun (CORE_N). The rules in example (13) handle the determined as well as the indetermined forms of the adjective.

(13) The adjectival node
  a. CORE/N {NUC/N} {PERIPHERY/N}
  b. PERIPHERY/N {ADJ}
  c. ADJ {ART} {ADJ}

This solution works for Gen 1:16 on display in Figure 17 in the first of its possible parsings.[25] The reason for the three solutions is that in the WIVU database the

[24] See http://lex.qwirx.com/lex/clause.jsp?book=Genesis&chapter=1&verse=5&clause=28839
[25] See http://lex.qwirx.com/lex/clause.jsp?book=Genesis&chapter=1&verse=16&clause=28747

discourse marker *ʔet* is incorrectly treated as a preposition (see note on Gen 1:1 above).

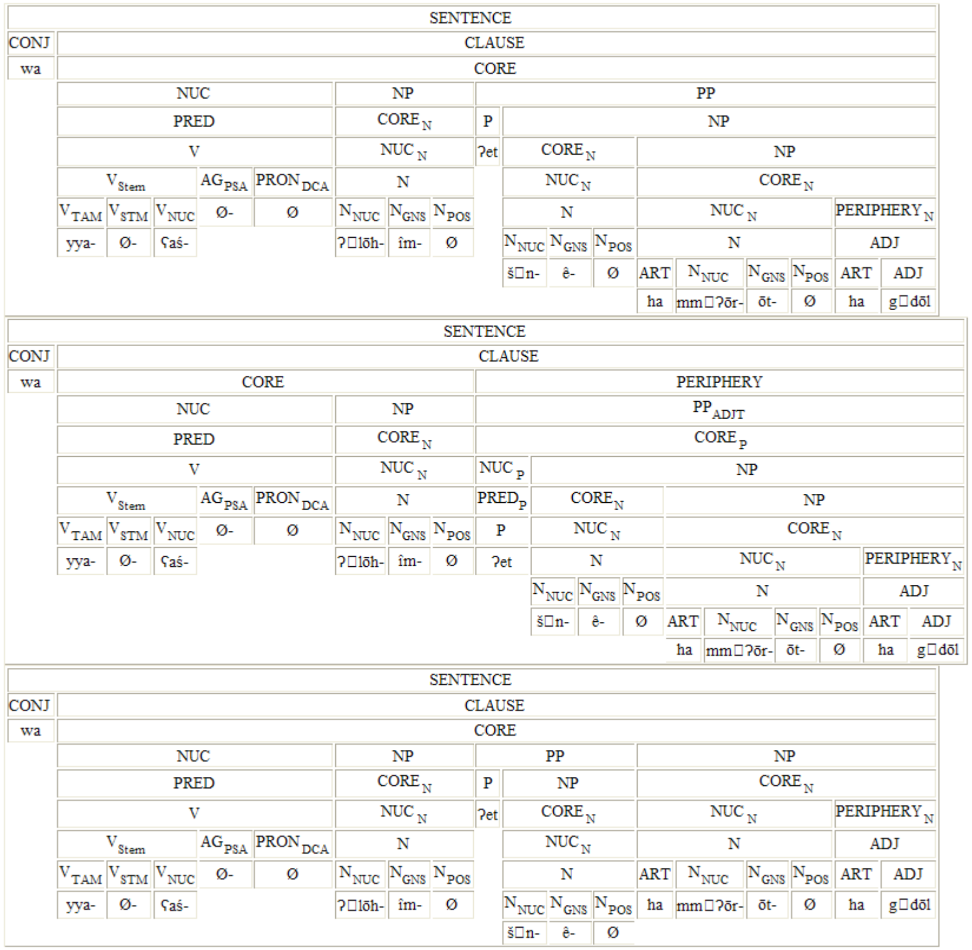


Figure 17 Screenshot of ”node-box” display for Gen 1:16

We have already discussed the parsing of the verb (Table 6), and the rules in example (14) simply summarize all the parser rules which are part of the nucleus node in the core.

(14) The nucleus node

a. NUC {PRED}
b. PRED {V}
c. V {V/Stem} {AG/PSA} {PRON/DCA}
d. V/Stem {V/TNS} {V/STM} {V/LEX}

The rules in example (15) show how we can parse the nucleus when it has been modified by a periphery with an adverb.

(15) The nuclear adverb node
a. PERIPHERY/V {ADV}
b. NUC {PERIPHERY/V} {NUC}

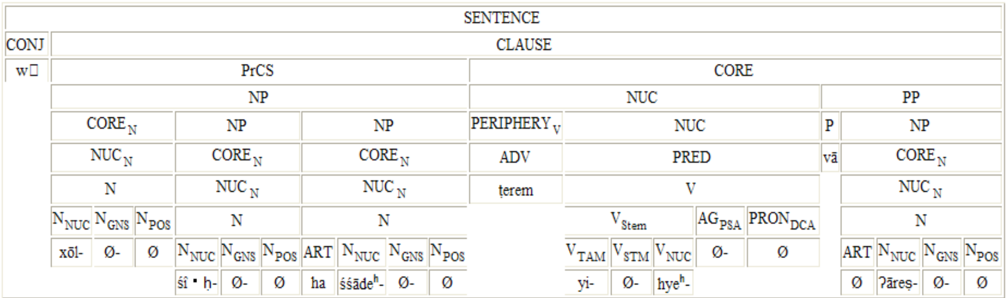


Figure 18 Screenshot of "node-box" display for Gen 2:5

This happens in the interesting case of the use of the adverb *ṭerem* 'not yet' in Gen 2:5 in Figure 18.[26]

Writing these rules allows the linguist to build automated syntactic parsings for all regular clauses in his data, and slowly widen their scope to cover the less frequent examples. All the rules can then be summarized and inspected following the summary of rules in Table 7. These rules are not exhaustive, but they cover the basic syntax of Hebrew, and can easily be expanded. It is not impossible to implement a technological solution for assembling these rules into syntactic the templates of RRG in future programming.

[26] See http://lex.qwirx.com/lex/clause.jsp?book=Genesis&chapter=2&verse=5&clause=28919

Table 7. Nodes and references for the parser rules for Biblical Hebrew

ID	Left-Hand	Right-Hand	Node & Ref	ID	Left-Hand	Right-Hand	Node & Ref
694	ADJ	{ART} {ADJ}	Adj (13)	**766**	NP	{CORE/N} {NP} {NP}	
750	CLAUSE	{CORE/Vbl}	CLAUSE (7)	**678**	NP	{NP} {CONJ} {NP}	LSNP (9)
730	CLAUSE	{CORE/V}		**715**	NUC	{PERIPHERY/V} {NUC}	Nucl Adv (15)
760	CLAUSE	{CORE}	CLAUSE (7)	**642**	NUC	{PRED}	NUC (14)
644	CLAUSE	{CORE} {PERIPHERY}	CLAUSE (7)	**629**	NUC/N	{N}	LSNP (9)
775	CLAUSE	{PrCS} {CORE}	PrCS (6)	**702**	NUC/N	{N} {NP}	
754	CLAUSE/Vbl	{NP} {NUC/PP}	Vbl (10)	**762**	NUC/P	{PRED/P}	
759	CORE	{NUC}		**753**	NUC/PP	{PRED/PP}	Vbl (10)
761	CORE	{NUC} {NP}	CORE (8)	**710**	PERIPHERY	{PP/ADJT}	PERIPH. (11)
521	CORE	{NUC} {NP} {PP}	CORE (8)	**693**	PERIPHERY/N	{ADJ}	Adj (13)
767	CORE	{NUC} {NP} {PP} {NP}	CORE (8)	**714**	PERIPHERY/V	{ADV}	Nucl Adv (15)
614	CORE	{NUC} {PP}	CORE (8)	**525**	PP	{PP} {CONJ} {PP}	PP (12)
741	CORE	{PRED/NP}		**536**	PP	{P} {NP}	PP (12)
700	CORE/N	{NUC/N}	LSNP (9)	**709**	PP/ADJT	{CORE/P}	PERIPH. (11)
711	CORE/N	{NUC/N} {PERIPHERY/N}	Adj (13)	**772**	PrCS	{NP}	PrCS (6)
751	CORE/N	{NUC} {NP}	LSNP (9)	**771**	PrCS	{PP}	PrCS (6)
707	CORE/P	{NUC/P} {NP}	PERIPHERY (11)	**556**	PRED	{V}	NUC (14)
749	CORE/Vbl	{NP} {PRED}		**705**	PRED/P	{P}	
776	Foo	{PP} {CORE/N} {PP}		**752**	PRED/PP	{PP}	Vbl (10)
774	LDP	{NP}	LDP (5)	**623**	SENTENCE	{CLAUSE}	S (4)
773	LDP	{PP}	LDP (5)	**744**	SENTENCE	{CONJ} {CLAUSE/Vbl}	S (4), Vbl (10)
738	N	{ART} {N/NUC} {N/GNS} {N/POS}		**728**	SENTENCE	{CONJ} {CLAUSE}	S (4)
737	N	{N/NUC} {N/GNS} {N/POS}		**655**	SENTENCE	{LDP} {CLAUSE}	S (4), LDP (5)
701	NP	{CORE/N}	LSNP (9)	**563**	V	{V/Stem} {AG/PSA} {PRON/DCA}	NUC (14)
704	NP	{CORE/N} {NP}	LSNP (9)	**736**	V/Stem	{V/TAM} {V/STM} {V/NUC}	NUC (14)

These rules cover regular verbal clauses (Figure 13 and Figure 15), as well as the Hebrew verbless clause (Figure 14). We have also shown that whenever a parsing fails the linguist will look for explanations in an improved syntactic analysis as is the case for the direct core marker misinterpreted as a preposition in Gen 1:16 (Figure 17). In other cases the linguist will know that this ambiguity can only be resolved once the syntax has access to a lexicon with information on the number of macroroles attached to a verb as in the case of Gen 1:5 in Figure 16. Semantic considerations will also play a role in cases where the linguist has to choose between a preposed LDP or a PrCS, but here pragmatic considerations of story setting and the like will also play a role as in Gen 1:1 (Figure 11). These rules are not exhaustive, but they cover the basic syntax of Hebrew, and can easily be expanded. It is not impossible to implement a technological solution for assembling these rules into syntactic the templates of RRG in future programming.

The RLM-tool deviates somewhat from the basic principles of RRG which assumes that a limited number of syntactic templates are stored in the syntactic inventory for any specific language (2005:11-16). Wilson has already done much of the ground-work for the design of an online syntactic parser and basically just needs to program the html-display which should be able to work for display of linguistics tree for any Emdros database. Since our project is currently on hold for lack of funding, we will have to wait for the full implementation until Wilson gets funding for full-time work on his Lex project. Alternatively, we could hope that other programmers within the Emdros community would develop an add-on module for syntactic projection that could work for any linguistic data stored in our database system, and use the chart parser developed for the RLM-tool.

Alternatively, if at some point we obtain substantial funding for Wilson it will be possible for him to develop a full interface for RRG with information structure parsing and all other features in the complete syntax-semantics-pragmatics interface of Role and Reference Grammar. Furthermore, we have been offered access to code from Guest (2008:441-442) who has developed her own chart parser independently of our project, and she uses the same technique of collapsing templates to rules (2008:436).

For now, however, we can emulate the true formal features of a tree for purposes of research into the structural constituents of the grammar.[27] At this stage our "node-box" displays can at least be manually produced by export into the 'Linguistic Tree

[27] Elizabeth Guest is offering her free tree drawing program *RRG draw* online for download at the RRG homepage (http://linguistics.buffalo.edu/people/faculty/vanvalin/rrg.html).

Constructor' (LTC) [28] by using the option to export the transliterated text in the link at the bottom of the *Browser* page (Figure 19).[29]

Download clause in GEN format for LTC:

- With Hebrew (right-to-left)
- Without Hebrew (left-to-right)

Figure 19 Clause display export function for LTC

The text is saved and can be opened as generic text type by the LTC tool. By means of this teaching and experimentation tool one may manually construct examples of the layered structure of the clause (LSC) and of the layered structure of the noun phrase (LSNP) among other things. It is possible to tag the words in the text as syntactic constituents in the manner shown in Figure 20.

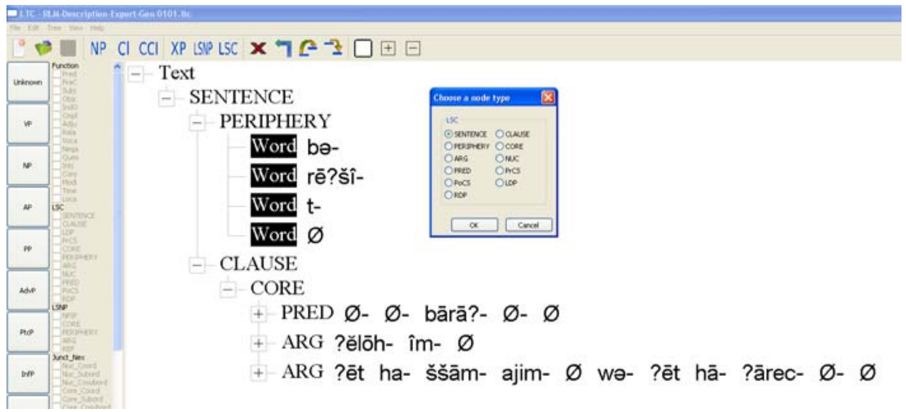


Figure 20 Manual linguistic construction of trees – LTC (old translit.)

The LTC also has a facility for export of horizontal trees in scalable zoom, so it is possible to combine automated box-displays with more conventional tree display (Figure 21).

[28] LTC has been constructed by Ulrik Sandborg-Petersen for building linguistic syntax trees in a point-and-click fashion and is available for download as a free program (http://ltc.sourceforge.net/).
[29] Please note that from now on we use the older transliteration from an earlier stage.

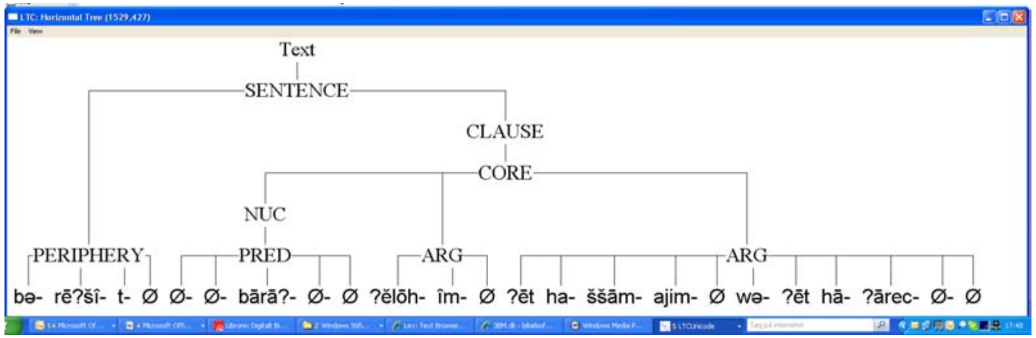


Figure 21 A horizontal tree display in LTC (old translit.)

This discussion has hopefully shown to what degree the RLM-tool can be used for parsing and for graphical display of syntactic trees. It is mainly a matter of time and design as well as cooperation before it will be possible to do syntactic trees online on the fly. Meanwhile we can explore and explain the syntax by the RLM-tool.

The Logical Structure and Lexical Roles for Gen 1:1

The RLM-tool was originally built as a module for lexical analysis in RRG. What we have described so far emerged gradually as a result of our curiosity, and the ability of the tool to explore the potential for building a parser as an add-on to the lexical analysis. Our original vision was to rely completely on the syntactic analyses in the Amsterdam database, but the present step helps us move beyond the WIVU by building a parser for RRG. In this way we can also more easily show how our tool can be used for many other languages, and we can choose any other open source Hebrew text without the data in the WIVU database, if we wanted to. However, to complete the picture and illustrate the logico-semantic potential, I will briefly introduce the role-lexical analysis for Gen 1:1. Petersen (2007) has analyzed the logical structure of Gen 1:1 in Conceptual Graphs, and Winther-Nielsen (2008) has discussed the Functional-Lexematic Module analysis of Gen 1:1, but the following discussion of the Logical Structure of Gen 1:1 is new.

In the RLM-browser, below the glossing and translation lines and above the lexical selection tools (Figure 7), a line mentions that the "Predicate text is: >MR [." This information from the WIVU database informs the linguist that the predicate of this clause is written with the consonants "BR>" in the peculiar consonantal text in the WIVU-text. By means of the transliteration key in Table 2, the linguist can figure out

that this is one way to write the verb ***bārā***$^{?}$ 'he created'. This information is very useful, because the linguist can now select *Search* from the menu line at the top of the RLM-tool and enter these consonants into the entry field displayed in Figure 22 in order to find all other instances of this verb in the corpus.[30] This figure also shows what happens when he chooses to press the Advance Help button in order to be able to write the consonantal text in the WIVU format or in the RLM-transliteration.

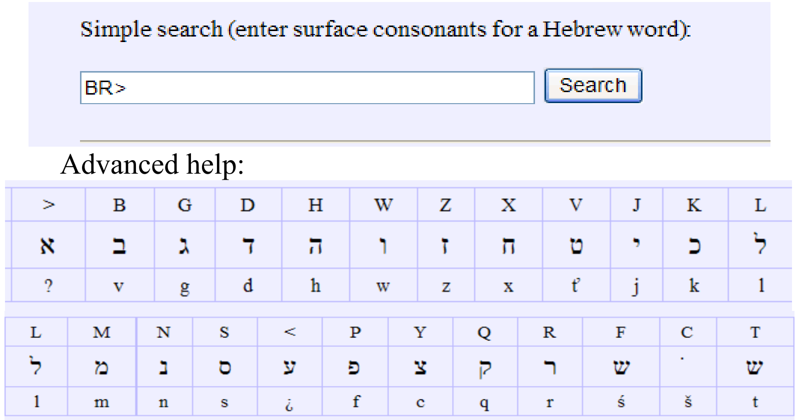


Figure 22 The Simple Search option and Advanced Help

The output of this particular search will return all clauses with the verb *bārā*$^{?}$ in Genesis 1-3. It is then easy for the linguist to check all these cases and inspect the available evidence in Figure 23.

The most central task of the RLM-tool is to build a lexicon as part of the syntax-to-semantics interface and expand it beyond the lexical glossing of the text display and in WordNet. RRG requires that the linguist can annotate his text corpus with semantic information and store his results in a semantic lexicon. The most important function of the RLM-tool is to enable the linguist to do this by linking from syntax to semantics via the lexicon. At the bottom part of the *Browser* screen he can provide this crucial semantic information displayed in Figure 24.

[30] Cf http://lex.qwirx.com/lex/search.jsp?q=BR%3E&simple=Search&aq=&max_results=100 .

Search Results for *BR>*

Displaying first 7 of 7 results.

Clause Text	Reference
בְּ רֵאשִׁית **בָּרָא** אֱלֹהִים אֵת הַ שָּׁמַיִם וְ אֵת הָ אָרֶץ bə- rē?šît **bārā?** ?ĕlōhîm ?ēt ha- ššāmajim wə- ?ēt hā- ?ārec	GEN 01,01
וַ **יִּבְרָא** אֱלֹהִים אֶת־ הַ תַּנִּינִם הַ גְּדֹל וְ אֵת כָּל־ נֶפֶשׁ לְ מִינֵהֶם wa- **yyivərā?** ?ĕlōhîm ?et ha- tannînim ha- gədōl wə- ?ēt kāl nefeš lə- mînēhem	GEN 01,21
וַ **יִּבְרָא** אֱלֹהִים אֶת־ הָ אָדָם בְּ צַלְמוֹ wa- **yyivərā?** ?ĕlōhîm ?et hā- ?ādām bə- caləmōw	GEN 01,27
בְּ צֶלֶם אֱלֹהִים **בָּרָא** אֹת bə- celem ?ĕlōhîm **bārā?** ?ōt	GEN 01,27
זָכָר וּ נְקֵבָה **בָּרָא** אֹת zākār ū- nəqēvāh **bārā?** ?ōt	GEN 01,27
אֲשֶׁר־ **בָּרָא** אֱלֹהִים ?ăer **bārā?** ?ĕlōhîm	GEN 02,03
בְּ **הִבָּרְאָם** bə- **hibārə?ām**	GEN 02,04

Figure 23 Output of search for BR> in Genesis 1-3 (old translit.)

Selected lexicon entry logical structure: do'(<x>, [create'(<x>:CREATOR, <y>:CREATION)]) & INGR exist'(<y>)

Linked logical structure: do'(?ĕlōhîm, [create'(?ĕlōhîm:CREATOR, ?ēt ha- ššāmajim wə- ?ēt hā- ?ārec:CREATION)]) & INGR exist'(?ēt ha- ššāmajim wə- ?ēt hā- ?ārec)

Notes

Add note: Create

Figure 24 Logical Structure and Semantic Representation (old translit.)

RRG works with logical structures like the one listed in the first line of example (16). The RLM-tool uses this kind of information for the selected lexicon entry, but combines the logical structure with a semantic characterization in terms of sets of roles which are defined technically as "thematic relations in terms of logical structure

argument positions" (Van Valin 2005:55).

(16) The lexicon in RRG and the role lexical module
- a. Logical Structure in RRG: **do'**(<x>, [**create**'(x, y)]) & **INGR exist'** (y)
- b. LS with thematic roles: do'(<x>, [create'(<x>:CREATOR, <y>:CREATION)]) & INGR exist'(<y>)
- c. Semantic representation: do'(?ĕlōhîm, [create'(?ĕlōhîm:CREATOR, ?ēt ha- ššāmajim wə- ?ēt hā- ?ārec:CREATION)]) & INGR exist'(?ēt ha- ššāmajim wə- ?ēt hā- ?ārec)

To produce these thematic roles and semantic representation by hand is a fairly challenging and time-consuming task, prone to error, and virtually impossible to do for a larger stretch of text. Example (16) shows how well the RLM-tool already at this stage can represent the very first clause of the Hebrew Bible by tagging and storing the logical structure and the semantic representation as required by the theory (2005:42-49).

RRG has implemented the Vendler-Dowty system of verb classification, but developed this approach in its own particular way. Among other things Van Valin has introduced a distinction in lexical aspect between the activity class and a particular group of verbs of movement, creation and consumption belonging to the active accomplishment class (Van Valin and LaPolla 1997:111). To the logical structure of activity is added the logical operator & "and then" and accomplishment BECOME before a **be-at**´, **exist**´ or **consumed**´ verb. In this way it is possible to distinguish between the activity *Big Bangs creates Universes* and the active accomplishment *In the beginning God creates.* Recently it has been suggested that active accomplishment involves a punctual end point (Van Valin 2005:44-45). In this case the logical structure should be represented as "& INGR **be-at**´/**exist**´/**consumed**´ (x)", but active accomplishment is retained as a classic term in the theory.

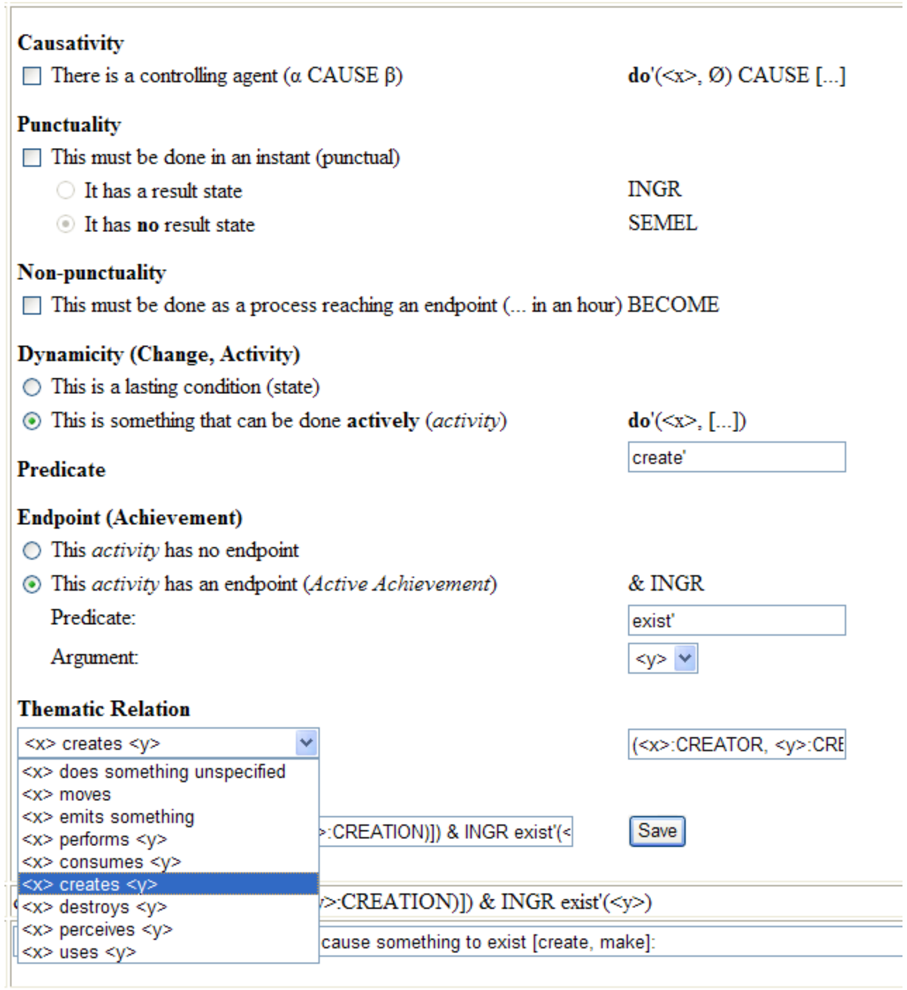


Figure 25 Lexical tests

The RLM-tool is designed to annotate and store such representations of meaning according to Logical Structure and perform a Semantic Representation which includes thematic roles like CREATOR and CREATION. The tool helps the linguist to answer seven crucial test questions that help expose the temporal structure of the verb in question (2005:34-41). These questions peel off the logico-semantic features bit by bit in an orderly procedure shown in the decision table for lexical tests in Figure 25. First

we need to isolate the operator-connective CAUSE[31] because, if present, it forces us to posit two LS components related by causation. In this case we would need to ask for a causative paraphrase of the verb class, but in Gen 1:1 the natural meaning is hardly that God has someone else cause the universe to be created. We then have to locate any punctual elements with or without result states as in achievement and semelfactive. This does not apply either, because creation is not necessarily by nature an instantaneous event for a split second, nor is this and accomplishment, i.e. **the earth created in an hour* does not make sense in the same way as *the earth dried in a year* after the Deluge.

Accordingly we are left with a decision between activity and state, and *create* is clearly an ongoing dynamicity in action which involves the actor actively during production. We click for activity and then enter “create'” as a primitive predicate belonging to the semantic language, and the program will store the logical structure “do'(<x>, [create'(<x>…”. Furthermore, RRG requires us to ask whether this verb is an activity of the variety that inherently reaches an endpoint in its basic sense, and this is the case with *create* in contrast to some senses of *make* and the unspecified sense of *do*. Since the activity of manufacture in most cases will reach an endpoint in order to qualify for a felicitous creation, we will click on “This *activity* has an endpoint (*Active Achievement*)”. We will then have to manually enter “exist'” and choose the second argument “<y>” to produce the last part of the Logical Structure “& INGR exist'(<y>)”. As the last step we now also have the opportunity to add the thematic relation labels for this particular Logical Structure by choosing “<x> creates <y>”. In an improved version this choice should actually suffice to select the “exist'” and “<y>” parts of the Logical procedure.

In this way our tool can help us produce both the logical structure and the semantic representation in the format of the display of example (16) which is enriched with thematic relations. Furthermore, the program will apply this reading for all the following instances of the verb in question.

[31] As correctly pointed out by Robert Van Valin (p.c.), the first box causation about is incorrect, because causation and having a controlling agent are not directly related or correlated. Pure activities with no causation can have a controlling agent (*Kim intentionally song so loudly that her boyfriend could hear her*), and causative event can lack one (*Kim accidentally/the rock broke the vase*). This can hopefully be discussed and changed in a future version of the the tool.

Lex by Chris Wilson

RLM for BH: Nicolai Winther-Nielsen

Home Databases Published Browse Search Lexicon Parser Rules Wordnet Database Dump Logged in as **nicolai**

Published Clauses

Displaying first 5 of 5 results.

Verb	Logical Structure	Reference
ברא	do'(?ělōhim, [create'(?ělōhim:CREATOR, ?ēt ha- ššāmajim wə- ?ēt hā- ?āreç:CREATION)]) & INGR exist'(?ēt ha- ššāmajim wə- ?ēt hā- ?āreç)	GEN 01,01
נכה	do'(<x>, Ø) CAUSE BECOME dead' (<x>:PATIENT)	EXO 02,11
נכה		EXO 02,12
נכה		IISA 23,10
נכה		IISA 23,12

© Lex Project 2005-2008.

Figure 26 List of Published Logical Structures

The logical structure tool is not available for trial by the public. Only partners in our project may get access to this tool, because it involves the copyright issues of the WIVU database, and we would not want visitors to interfere with our data-production. Project partners can, however, publish their data for the general linguistic community under *Published* as shown in Figure 24. At the present stage of our research there is no significant information published here, but our next step is to publish examples of significant verb classes from all the Hebrew in lists as part of our main project to produce a logical structure lexicon for Hebrew.

This final example illustrates how well our tool solves the central task in RRG to classify particular verbs and map their syntactic structure onto a logical structure. By using a program with a strict procedure for asking the right kind of test questions, the RLM-tool will help the linguist to discover and register the thematic relations as a function of argument positions. It all happens on the basis of non-arbitrary criteria, as claimed by Van Valin (2005:59). It also in our view proves that the RLM-tool is a very unique and useful Web-application for semantic analysis within the RRG theory for Hebrew as well as for any other corpus in any other languages. It may give us the opportunity to relate logical structure in different languages and compare across languages for logical structures and thematic relations for all projects which use the Emdros database and tools like the RLM which Chris Wilson wants to develop within the Lex project.

Conclusion

Since the introduction of to the RLM-tool in Winther-Nielsen (2008), Wilson (2009) has designed and programmed new features for the tool and has explained the Lex system in detail. In the present contribution I have explored the new linguistic features for linguistic analysis, and I have explained how the tool can bridge the gap between the data for a corpus of Ancient Hebrew text and the needs of the linguist working within the framework of Role and Reference Grammar. The displays show how the RLM-tool is designed to use the Hebrew text and the syntactic database built by the Werkgroep Informatica (WIVU) in Amsterdam, but now moves beyond the database by implementing a parser. We are furthermore in the process of developing our application into a tool for display of all of the syntax, semantics and pragmatics interface for a Role and Reference Grammar of Hebrew.

The presentation has focused on the power of our parser to build "node-box" displays for experimentation with syntactic trees. At this stage of development we are able to write rules that satisfy the basic syntactic parsing requirements of RRG and show the output in a preliminary format of tables. Designer and developer Chris Wilson has the ideas for syntactic projection, but we miss the funding. The linguist can parse regular clauses like the verbal (Gen 1:3) and verbless clauses (Gen 1:2), the periphery (Gen 1:27), and display the choice between an adjunct and an oblique core argument (Gen 1:5) as well as between LDP and PrCS (Gen 1:1). We hope that these cases will convince fellow linguists that our tool solves the task of constituent projection sufficiently well for Biblical Hebrew.

The main goal of the project follows in the next step when the linguist starts to analyze information on lexical aspect and argument realization in Biblical Hebrew. We want to build a logical structure lexicon for Hebrew and at the same time explore the verbal stem system of Hebrew, and we will publish such data for inspection by fellow linguists and Hebrew scholars through the RLM-tool as exemplified in our case for Gen 1:1. The goal is to classify the Hebrew verbs and map their syntactic structure onto a logical structure. The program is set up to ask the right kinds of test questions in a step-wise procedure and according to non-arbitrary criteria, as claimed by Van Valin (2005:59). This will help the linguist to discover and register the right verb classes, and the program will register everything in the Emdros database without human entry error. As an added bonus our tool can show thematic relations as a function of argument positions.

In our view the tool promises to offer a very unique and useful tool for semantic analysis of Biblical Hebrew within the RRG theory. Much further design development remains to be done, and we sincerely hope that lack of funding will not continue to slow down the process of programming and production. We hope to see the addition of syntactic trees to replace our node-box displays for projection on the fly on the internet and for export to personal research and publication. We are exploring how to construct a new lexicon based on the verb-specific thematic relations, but funding is a major issue for our project.

It is our hope that this description of the RLM-tool sufficiently well illustrates how corpora of linguistic data in other languages can be analyzed. Sandborg-Petersen (2008) explains how the Emdros database works for texts of a cultural heritage in Danish by analyzing the writings of poet and pastor Kaj Munk murdered by the Nazis in 1945. This corpus linguistic work should be carried out for many other projects as well. We hope that Chris Wilson can obtain contract funding for building similar databases for other RRG language projects under his LEX project, and we believe that this would enable us to relate logical structure across different languages and compare verb classification. It would enable research into semi-automated translation based on the language processing capacity of RRG and mapping from one metalangue to another.

The RLM-tool in our view is a very helpful tool for the modern study of Biblical Hebrew Linguistics. A major problem in the future is how to make the tool and our data available to fellow scholars and students beyond the corpus in Genesis 1-3 and the isolated clauses we publish. Our dream is to make our results commercially available as an add-on to the SESB Bible software or as part of a teaching and research program like LTC or 3ET, and to build a community for exchange of linguistic interpretations among students and researchers. Already at the present the Genesis 1-3 corpus ties into a related project to build the 3ET for introductory Hebrew learning (http://3bm.dk/index.php?p=82) and for persuasive learning technology. Furthermore, Winther-Nielsen, Tøndering and Wilson (2009) explain how we use the Bergman (2005) transliteration for the RLM-tool, and we hope to explore ways to build new teaching tool to supplement Nava Bergman's e-Learning material in case we can get the necessary funding for a Moodle implementation (http://3bm.dk/index.php?p=81). We suggest that the RLM-tool has great potential for teaching the linguistics of Biblical Hebrew, at least within the framework of Role and Reference Grammar.

Bibliography

Andersen, Francis I. and A. Dean Forbes. 2008. The Hebrew Bible: Andersen-Forbes Phrase Analysis. Version 0.80 (November 2008). Bellingham,WA: Logos Bible Software

Anstey, Matthew. 2006. Towards a Functional Discourse Analysis of Tiberian Hebrew. Canberra, Australia: AddColour Digital Pty Ltd.

–. FC. The Biblical Hebrew Qatal Verb: A Functional Discourse Grammar Analysis Interpersonal Grammar: A Crosslinguistic Perspective. To be published in a special edition of Linguistics.

Bergman, Nava. 2005. *The Cambridge Biblical Hebrew Workbook: Introductory Level*. Cambridge: Cambridge University Press.

Bosman, Hendrik-Jan; R. Oosting & Ferenc, Postma, eds. 2003. Hebrew and Aramaic Lexicon to the Old Testament : English and German. Stuttgart: German Bible Society, Stuttgart.

Guest, Elizabeth. 2008. Parsing for Role and Reference Grammar. In Van Valin, Robert D. Jr., 2008. 435-453.

Hardmeier, Christof; Eep Talstra & Bertram Salzmann. 2009. *SESB: Stuttgart Electronic Study Bible (Stuttgarter Elektronische Studienbibel)*, Stuttgart: Deutsche Bibelgesellschaft Bibelgesellschaft / Haarlem: Nederlands Bijbelgenootschap (SESB version 3).

Kumerow, David. 2005. Review Article: Stuttgart Electronic Study Bible. *SEE-J Hiphil 2* (available online at http://www.see-j.net/Default.aspx?tabid=70).

McDaniel, Chip and C. John Collins. 2006. *The ESV English-Hebrew Reverse Interlinear Old Testament* . Bellingham: Logos Research Systems, Inc.

Miller, Cynthia. 1996. *The Representation of Speech in Biblical Hebrew Narrative: A Linguistic Analysis*. Atlanta: Scholars Press.

Petersen, Ulrik. 2007. Genesis 1:1-3 in Graphs: Extracting Conceptual Structures from Biblical Hebrew. *SEE-J Hiphil* 4 [http://www.see-j.net/hiphil]. Accessed 20.01.07]

Sandborg-Petersen, Ulrik. 2008. *Annotated text databases in the context of the Kaj Munk Archive: One database model, one query language, and several applications*. Ph.D dissertation defended at the University of Aalborg June 27 2008. Available at http://www.hum.aau.dk/~ulrikp/PhD/

Talstra, Eep. 2006. *Biblia Hebraica Stuttgartensia. Werkgroep Informatica Constituency Tree Analysis*. Stuttgart: Deutsche Bibelgesellschaft

Talstra, E. and Sikkel, C.: 2000, Genese und Kategorienentwicklung der WIVU-

Datenbank, *in* C. Hardmeier, W.-D. Syring, J. D. Range and E. Talstra (eds), *Ad Fontes! Quellen erfassen - lesen - deuten. Was ist Computerphilologie?* , VU University Press, Amsterdam

van der Merwe, Christo H.J., Jackie A. Naudé, Jan H. Kroeze. 1999. *A Biblical Hebrew Reference Grammar*, Biblical Languages: Hebrew, no. 3. Sheffield: Sheffield.

Van Valin, Robert D. Jr., 2004 Lexical representation, co-composition, and linking syntax and semantics. Available at http://wings.buffalo.edu/linguistics/research/rrg/

–. 2005. *Exploring the Syntax-Semantics Interface*, Cambridge University Press

–. 2008, ed. *Investigations of the Syntax-Semantics-Pragmatics Interface* [SLCs 105]. Amsterdam and Philadelphia: John Benjamins.

Van Valin, Robert D. Jr, and LaPolla, Randy. 1997. *Syntax: structure, meaning and function.* Cambridge: Cambridge University Press.

Waltke, Bruce K., and Michael O'Connor. 1990. *An Introduction to Biblical Hebrew Syntax*. Winona Lake, Ind.: Eisenbrauns

Wilson, Chris. 2009. Lex: a Software Project for Linguists. *Technical Report SEE-J Hiphil* 6 [http://www.see-j.net/index.php/hiphil] (2009).

Winther-Nielsen, Nicolai. 1992. 'In the beginning' of Biblical Hebrew Discourse: Genesis 1:1 and the Fronted Time Expression. In Hwang, Shin Ja J. & William R. Merrifield, eds. *Language in Context: Essays for Robert E. Longacre*, 67-80. Dallas: SIL and the University of Texas at Arlington Publications in Linguistics 107

–. 1995. *A Functional Discourse Grammar of Joshua. A Computer-assisted Rhetorical Structure Analysis.* (CBOTS 40). Uppsala: Almquist & Wiksell.

–. 2005. Toward the Peak of Mount Sinai: A Discourse-Pragmatic Analysis of Exodus 19. *SEE-J Hiphil 2* [http://www.see-j.net/hiphil]. Published February 22, 2005.

–. 2008. A Role-Lexical Module (RLM) for Biblical Hebrew: A mapping tool forRRG and WordNet. In Van Valin, Robert D. Jr., 2008. 455-478.

Winther-Nielsen, Nicolai; Claus Tøndering & Chris Wilson, "Transliteration of Biblical Hebrew for the Role-Lexical Module" *Technical Report SEE-J Hiphil* 6 [http://www.see-j.net/index.php/hiphil] (2009).