Transliteration of Biblical Hebrew for the Role-Lexical Module[1]

Nicolai Winther-Nielsen, Claus Tøndering, and Chris Wilson
Contact nwn@dbi.edu, claus@ezer.dk, chris+lexreport@qwirx.com

Reference: Winther-Nielsen, Nicolai; Claus Tøndering & Chris Wilson, "Transliteration of Biblical Hebrew for the Role-Lexical Module" *Hiphil* 6 [http://hiphil.see-j.net] (2009). Accessed DD.MM.YYYY

Abstract

The Role-Lexical Module (RLM) is a syntax-to-semantics mapping tool for Biblical Hebrew described in Winther-Nielsen (2008, 2009) and Wilson (2009). It is the first implementation of a multilingual database system for the theory of Role and Reference Grammar proposed by Van Valin (2005). In this technical report on work in progress we explain how we have improved on the transliteration of Biblical Hebrew for a new and updated version of the RLM-tool. Our goal is to allow fellow scholars and students to be able to copy-and-paste linguistically glossed text from Hebrew Bible into papers and books. The transliteration can also support online teaching and studies of the Hebrew characters for students at the introductory level of learning Biblical Hebrew.

This Technical Report will show how we have used the transliteration system develop by Nava Bergman for *The Cambridge Workbook of Biblical Hebrew* (2005) and implemented this into an online-tool. We use the Hebrew Bible database produced by the Werkgroep Informatica of the Vrije Universiteit (WIVU) in Amsterdam, and commercially available in full in the commercial Bible Software product the Stuttgart Electronic Bible Study (SESB) 3.0. Our project has a license for research generously granted us by the German and Dutch Bible Societies.

[1] This technical report was written without any funding, and the coordinator Winther-Nielsen is deeply grateful for the project partners' many days of unpaid work for the project. We would also like to thank Bertram Salzmann for the permission to use the WIVU database (see further details in Winther-Nielsen 2009).

This report explains matters of conversion, programming and selection of fonts for transliteration and display of Biblical Hebrew in the Role-Lexical Module (RLM) available for online study and representation (http://lex.qwirx.com/lex/clause.jsp). The report will discuss how an IT system can emulate the sophisticated reading conventions developed fifteen hundred years ago for public reading of a stage of Biblical Hebrew which we can call Tiberian Hebrew (Anstey 2006).

The report describes our initial translation, the switch to the Bergman (2005) system, the contextual rewrite rules and the programming work as well as challenges still remaining. This work is part of the Lex project of Wilson (2009) who wants to be able to do bidirectional linking between syntax and semantics for translation purposes involving infrequent alphabets.

The initial transliteration

In the first publication on the RLM-tool by Winther-Nielsen (2008) it was argued that linguists within the Role and Reference Grammar framework as well as linguists in general need a useful representation of the graphical forms of Ancient Biblical Hebrew in a format that could be easily read and used for language-typological work. At this stage we therefore deliberately chose to represent Biblical Hebrew in characters that are available in the Times Roman symbol sets delivered with Microsoft Windows (2008:466). It was hoped that the ultimate transliteration system would enable the linguist to type sophisticated queries for our Emdros database in a character set that would not require installation of a peculiar phonetic or Semitic fonts, let alone be able to type them into a query.

Hebrew Bible scholars will usually display Biblical Hebrew in the original foreign scripts and use traditional grammatical labels, but this solution is not an option if we want to share the typological data of Hebrew with fellow linguists.

Initially we looked at the proposal of Anstey (2006) for a kind of morpho-phonetic transliteration, because this represented the most recent and well argued attempt to display Hebrew in a sophisticated transliteration of Tiberian Hebrew for fellow linguists. Anstey follows the Leipzig Glossing conventions as shown in example (1).

(1) Translitteration of Hebrew in Matthew Anstey (2006:107)

גַּם הַבֵּן הַיִּלּוֹד לָךְ מוֹת יָמוּת׃

gá:m	hab-bé:n	haj-jillò:(w)ð	lə-χó:	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)

However, Winther-Nielsen (2008) argued that this representation of Biblical Hebrew is not necessary for linguistic representation in a syntax-semantics-pragmatics interface, and it is not very familiar to the Biblical Scholar. For the programming of the RLM-tool he instead proposed to choose the transliteration of consonants and vowels listed in

Table **1**. This choice seemed to be an easy solution for a tentative representation of the Hebrew characters, but we assumed that the proposed format would be useful for the general linguist as well as sufficiently distinctive to allow the non-Hebraist scholar to appreciate the linguistic evidence adduced from the original Hebrew text in question. We also found similar representations in glossed text published by fellow linguists working on Modern Hebrew, or Ivrit.

Table 1. Transliteration of Hebrew consonants and vowels in RLM

Consonants

Hebrew	א	ב	ג	ד	ה	ו	ז	ח	ט	י	כ	ל	מ	נ	ס	ע	פ	צ	ק	ר			ת
Transl.	?	v	g	d	h	w	z	X	ť	y	k	l	m	n	s	¿	f	c	q	r	ś	š	t

Vowels

---			---			---				
ā	a	û	u	ō	ā	i	ē	e	ə	

The first preliminary transliteration served our purpose well. We could proceed right away to our exploration of the syntax-semantics-pragmatics interface for a Role

and Reference Grammar. We avoided the time-consuming task to do research on the complex phonetic system of Tiberian vocalization from the 6th-9th century AD vocalization which the common manuscripts of the Hebrew Bibles reflect. The invention of a much more complex system seemed better handled by scholars like Matthew Anstey who specialize in reconstructing the phonetic and morpho-syntactic glossing of Tiberian Hebrew.

The programmer and designer of the RLM-tool, Chris Wilson programmed the first open source code for *HebrewConverter* (http://rrg.qwirx.com/trac/lex). For the programming of the first version of the RLM-tool Wilson (2009) had access to the Hebrew database of the Workgroep Informatica at the Vrije Universtiteit (WIVU) in Amsterdam for graphical representation of the Hebrew text. Wilson could exploit this database to cull grammatical, lexical and even syntactic information from the database and build a parser for Biblical Hebrew.

Before any RRG reinterpretation of the structural data from the WIVU database we would display a clause like Gen 1:3 in a transliteration and glossing like example (2). Data from the Hebrew texts were transliterated in this format and discussed in Winther-Nielsen (2008).

(2) וַיֹּאמֶר אֱלֹהִים

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

Towards a new transliteration

Unfortunately, this first "quick and dirty" solution for transliteration did not stand the test of time. The preliminary transliteration has proved to be too inaccurate in relation to the very complex representation of Biblical Hebrew in the Tiberian vocalization of the manuscripts. We realize that an inaccurate transliteration may not serve the needs of the average Hebrew Bible scholar and student who will want a more precise rendition of the traditional text of the Hebrew Bible which is relevant for the diachronic stage of the language we use today.

In the Summer of 2008 Winther-Nielsen discussed the issue of transliteration with the Swedish scholar Nava Bergman of the University of Gothenburg who is both a native speaker of Modern Hebrew and works on web-based teaching of Biblical Hebrew. Winther-Nielsen and Bergman explored ways of moving the *Cambridge Biblical*

Hebrew Workbook (Bergman 2005) into a Moodle-based e-Learning environment and in that context explored how best to display and train transliteration of Hebrew characters in a Learning Management System.[2] During these discussions Bergman offered a solution which was based on Modern Israeli conventions for the transliteration of the Hebrew consonants in Israel today and shown in Table 2 Israeli transliteration of Consonants (Bergman).

Table 2 Israeli transliteration of Consonants (Bergman)

	א	ב	ב	ג	ד	ה	ו	ז	ח	ט	י	כ	כ	ל	מ	נ	ס	ע	פ	פ	צ	ק	ר	ש	ש	ת
Initial RLM	?	v		g	d	h	w	z	x	ťʼ	y	k		l	m	n	s	¿		f	c	q	r	ś	š	t
Bergman 2008	>	b	v				V		X	T		k	x					<	p	f		K		S		

Bergman's proposal seemed quite attractive, because this representation could be used on any PC, and the user would avoid the problem of installing foreign fonts and getting the Unicode to work on the internet and in the browser. However, during the Autumn of 2008 this proposal met with objections in a scholarly discussion group on HiphiList (http://e.dbi.edu/course/view.php?id=72)

In the end Winther-Nielsen therefore decided to revert to the original transliteration proposed by Bergman (2005:1,14-15, 20). This solution is supported by an international scholarly publication of high esteem and in general use, and it is based on decades of teaching experience by Bergman. This transliteration will be well received among new learners of Biblical Hebrew using Bergman's *Cambridge Workbook of Biblical Hebrew,* while the general linguist will not find it hard to identify the original script of the Hebrew manuscripts. Using this proposal as our guideline ensures an international status for our transliteration as well as its usefulness in linguistic circles for language-typological work involving Biblical Hebrew.

Also in the Summer of 2008 Winther-Nielsen initiated cooperation with an expert developer, the Danish IT consultant Claus Tøndering. Together they embarked on a Biblical Hebrew e-Learning project which has now been developed by Tøndering into the Ezer's Emdros-based Exercise Tool (3ET).[3] He also offered to assist Winther-Nielsen on the major challenge of how to turn the rules for reading of Tiberian Biblical

[2] See http://3bm.dk/index.php?p=81for details on the *Bergman eLearning of Biblical Hebrew for Beginners* at.

[3] For further information see 3ET exercise tool (http://3bm.dk/index.php?p=82) and the presentation Persuasive Learning Objects and Technology (Login as GUEST) (=http://www.livssyn.hum.aau.dk/course/view.php?id=19)

Hebrew into a programming algorithm that Wilson could use in order to implement this as code in the RLM-tool.

The new transliteration makes extensive use of the Unicode character set. This includes phonetic symbols such as ʕ and ə, characters with diacritics such as ê and ḥ, and subscript and superscript characters such as $_a$ and ^{h}. All of these characters are part of version 5.0 of the Unicode Standard[4]. On most PCs, support for some of these characters is, however, limited to a few character fonts. On Windows PCs, "Arial Unicode MS" is often a good font choice.

The transliteration aims to capture modern Hebrew pronunciation. This means that the transliteration is not reversible. For example, both ד and דּ are transliterated as "d" in accordance with Bergman's proposal and modern Hebrew pronunciation.

Contextual rewrite rules for transliteration

Automatic transliteration of Hebrew presents a number of challenges both from a linguistic point of view and from the point of view of a computer programmer.

For the computer programmer, the main challenge is that the proper transliteration of a Hebrew character can depend on the characters preceding and following it. For example, as any reader familiar with primers or grammars on Biblical Hebrew will know, the transliteration of the character פ depends on whether it is followed by[5] a *dagesh* or not. But in order to determine if the *dagesh* is a *dagesh forte* or a *dagesh lene*, the characters preceding the פ must be examined.

We explored the system and formulated the first informal rules in examples (3))(5). From a linguistic point of view, the proper transliteration of the characters *qamets* and *shewa* pose the greatest challenges. The *shewa* is transliterated ᵊ or omitted, depending on the characters surrounding it as specified in example (3)).The *qamets* is transliterated ā or o, depending on the characters surrounding it as specified in example (4).

(3) The pronunciation of the shewa:
- At the end of a word, the shewa is silent (typically under a final ך).
- In the context "consonant - shewa - consonant - shewa" at the end of a word (verbal ending in 2fs perf.) the shewas are silent.
- After the first consonant in a word, the shewa is pronounced.

[4] The Unicode Consortium: *The Unicode Standard 5.0*. Addision-Wesley, 2007. ISBN 0-321-48091-0. Also available at http://unicode.org.

[5] In Unicode encoding, the dagesh *follows* the consonant, although visually the consonant *contains* the dagesh.

- In the context "consonant - long vowel - consonant - shewa", the shewa is pronounced.
- In the context "consonant - accented qamets - consonant - shewa", the shewa is pronounced.
- After a consonant with a dagesh (forte or lene), except at the end of a word, the shewa is pronounced.
- In all other cases the shewa is silent.

(4) The pronunciation of the qamets:

- In the context "consonant - unaccented qamets - consonant - shewa", the qamets is pronounced o (except in a few verbal forms).
- In the context "consonant - unaccented qamets - consonant" at the end of a word, the qamets is pronounced o.
- In all other cases, a qamets is pronounced as a long a.

(5) The dagesh also requires special consideration:

- A dagesh in the letter ה at the end of a word is not really a dagesh but a mapiq. It causes this letter to be a full h, rather than a lengthening of the preceding vowel.
- A dagesh in the letter ו is not really a daghesh, but instead turns that letter into a long u.
- In the context "consonant - vowel (not reduced) - consonant with dagesh", the dagesh is a dagesh forte and should be transliterated as a doubling of the consonant.
- In other cases, the dagesh is a dagesh lene. This only occurs in the letters b, ג, ד, כ, פ and ת, and causes them to be transliterated b, d, g, k, p, and t. Without the dagesh, these letters are transliterated, v, d, g, x, f, and t, respectively.

Cantillation marks, which typically indicate a stressed syllable, are ignored in the transliteration, except when their presence is required to identify is a qamets is accented, as described above.

The new transliteration was initially specified by Tøndering in the form of a table containing contextual rewrite rules, and these rules were then implemented and tested by Wilson (2009) while programming the system. In this process Wilson corrected the contextual rewrite rules.

Table 3 reflects the corrected rules for Biblical Hebrew transliteration according to the system in Bergman (2005).

The table should be read as follows:

The two first columns serve merely to make visual perusal of the table easy.

The columns "source string" and "destination string" identify which Hebrew characters should be transliterated as which Latin characters. The columns "preceded by" and "followed by" identify in which context the particular source string should be found in order for the transliteration to take place.

The column "HTML" gives the HTML equivalent of the destination string.

In the "preceded by" and "followed by" columns,

- the characters ^ and $ indicate the beginning of a word and the end of a word, respectively,
- *cons* is any consonant character possibly followed by a dagesh,
- *consND* is any consonant character not followed by a dagesh,
- *vow* is any non-reduced vowel (that is, not hataph qamets, hataph patah hataph segol or shewa),
- *longwov* is any long vowel (that is, tsere, holam and shureq (waw with "dagesh"))

Example: The 5th line in the table states that if a shewa is preceded by a consonant, and is followed by a consonant, another shewa and the end of the word, the shewa should be ignored.

For each Hebrew character, the rows are listed in order of priority. For example, there are two lines listed under aleph, The first one (aleph preceded by a vowel and followed by a non-vowel) should be tried first. Only if aleph is not found in this context should the second line be used. For efficient implementation, each rule is converted into two compiled regular expressions, one matching against "preceded by" and the other matching against "character + followed by".

To transliterate a string, we iterate a pointer over the gaps between characters, starting before the first. At each position, we then apply each rule in turn at that position until one matches. When that happens, we consume the characters in the "character" column and output the ones in the "Destination string" column, and move the pointer forward by the number of characters consumed (at least one position). If no rule matches at a position, that is a transliteration failure, and the original character is output as a debugging aid.

Table 3 Contextual rewrite rules for transliteration

Hebrew character	Name	Preceded by	Source string	Followed by	Destination string	HTML
־	Maqaf		Maqaf		-	
׃	Sof pasuq		Sof pasuq		:	:
ְ	Shewa	^ cons	Shewa		ᵊ (superscript)	ᵊ
			Shewa	$	Ignore	
		cons	Shewa	cons shewa $	Ignore	
		longvow cons	Shewa		ᵊ (superscript)	ᵊ
		Qamets accent cons	Shewa		ᵊ (superscript)	ᵊ
		consND dagesh	Shewa		ᵊ (superscript)	ᵊ
		consND	Shewa		Ignore	
ֳ	Hataph Qamets		Hataph Qamets		ᵒ (superscript)	ᵒ
ֲ	Hataph Patah		Hataph Patah		ᵃ (superscript)	ᵃ
ֱ	Hataph Segol		Hataph Segol		ᵉ (superscript)	ᵉ
ֵ	Tsere		Tsere	$	ēʰ	ē ʰ
			Tsere, he, dagesh	$	ēh	ēh
			Tsere, yod		ê	ê
			Tsere		ē	ē
ָ	Qamets		Qamets	<no accent> cons shewa	o	o
			Qamets	<no accent> cons shewa (See note)	ā	ā
			Qamets	<no accent> cons <no accent> $	o	o
			Qamets, he	$	āʰ	ā ʰ
			Qamets, he, dagesh	$	āh	āh

Hebrew character	Name	Preceded by	Source string	Followed by	Destination string	HTML
			Qamets, yod		āʸ	ā ʸ
			Qamets		ā	ā
ַ	Patah		Ayin, patah	$	ₐʕ	ₐ ʕ
			Khet, patah	$	ₐḥ	ₐ ḥ
			Patah		a	a
ֶ	Segol		Segol, heh	$	eʰ	eʰ
			Segol, he, dagesh	$	eh	eh
			Segol, yod		eʸ	eʸ
			Segol		e	e
ִ	Hiriq		Hiriq, yod		î	î
			Hiriq		i	i
ֹ	Holam		Holam, waw		ô	ô
			Holam, he	$	ōʰ	ō ʰ
			Holam, he, dagesh	$	ōh	ōh
			Holam		ō	ō
ֻ	Qibbuts		Qibbuts		u	u
א	Aleph	vow	Aleph	<not vow>	ʔ (superscript)	&x02C0;
			Aleph		ʔ	ʔ
בּ	Bet	vow	Bet, dagesh		bb	bb
		<not vow>	Bet, dagesh		b	b
			Bet		v	v
גּ	Gimel	vow	Gimel, dagesh		gg	gg
		<not vow>	Gimel, dagesh		g	g
			Gimel		g	g

Hebrew character	Name	Preceded by	Source string	Followed by	Destination string	HTML
דּ	Dalet	vow	Dalet, dagesh		dd	dd
		<not vow>	Dalet, dagesh		d	d
			Dalet		d	d
הּ	He		He, dagesh		h	h
			He		h	h
וּ	Waw		Waw, dagesh		û	û
			Waw		w	w
זּ	Zayin		Zayin, dagesh		zz	zz
			Zayin		z	z
חּ	Khet		Khet		ḥ	ḥ
טּ	Tet		Tet, dagesh		ṭṭ	ṭ ṭ
			Tet		ṭ	ṭ
יּ	Yod		Yod, dagesh		yy	yy
			Yod		y	y
כּ	Kaph	vow	Kaph, dagesh		kk	kk
		<not vow>	Kaph, dagesh		k	k
			Kaph		x	x
לּ	Lamed		Lamed, dagesh		ll	ll
			Lamed		l	l
מּ ם	Mem		Mem, dagesh		mm	mm
			Mem		m	m
נּ ן	Nun		Nun, dagesh		nn	nn
			Nun		n	n
ס	Samek		Samek, dagesh		ss	ss
			Samek		s	s

Hebrew character	Name	Preceded by	Source string	Followed by	Destination string	HTML
ע	Ayin		Ayin		ʕ	ʕ
פּ ף	Pe	vow	Pe, dagesh		pp	pp
		<not vow>	Pe, dagesh		p	p
			Pe		f	f
צּ ץ	Tsade		Tsade, dagesh		ṣṣ	ṣ ṣ
			Tsade		ṣ	ṣ
ק	Qoph		Qoph, dagesh		qq	qq
			Qoph		q	q
ר	Resh		R		r	r
שׁ	Sin		Sin, dagesh		śś	ś ś
			Sin		ś	ś
שׂ	Shin		Shin, dagesh		šš	š š
			Shin		š	š
תּ ת	Tav	vow	Tav, dagesh		tt	tt
		<not vow>	Tav, dagesh		t	t
			Tav		t	t

Testing and developing a transliteration

In a project where a Hebrew Bible linguist cooperates with expert programmers with little or no expertise in Hebrew it is important to formulate a test case method as described in Wilson (2009).

In this project Tøndering had been taught introductory Biblical Hebrew, but Wilson still needed to be able to test the correct transliteration output from the rewrite rules used in the program. For this purpose we assembled cases of test words that would prove the effect of formulating new rules during the programming work, and our choice is shown in overview in Table 4. In the Vowel part of this table, columns (3) and (4) contain the lexeme numbers and the encoding of the words in Latin characters from the WIVU database. The fifth column lists the output of the RLM-tool in the transliteration arrived at by April 2009, while the sixth column is the Hebrew text to compare with. Columns (7)-(8) present the vowels in Hebrew and the transliteration proposed by Bergman (2008).

This test evidence helps us realize what we still need to solve for the final version of the transliteration. By way of example, the final he of **luqºḥāh* (no. 20) is used as a matris lectionis and therefore not pronounced, so it should have come out as *luqºḥāʰ*. The missing shewa in no. 2 (**hāytāʰ* instead of *hāyᵊtāʰ*) derives from a programming problem which we cannot locate at the present but it is related to the problem of the use of a qamets with a meteg in an open stressed syllable.

We have already listed the rule that Qamets followed by "<no accent> cons shewa" is normally transliterated /o/ in example (4). To test the program against this rule and guarantee the correct rendition of qamets as a qamets hatuf with /o/ pronunciation in open unstressed syllable we used the test set in example(6).

(6) Qamets hatuf in open stressed syllable

- Exod 28,3 חָכְמָה ḥoxmāʰ
- Gen 4,8 וַיָּקָם wayyāqom

Table 4. Transliteration according to Bergman (2005)

CONS	א	ב	ב	ג	ג	ד	ד	ה	ו	ז	ח	ט	י	כ	כ	ל	מ	נ	ס	ע	פ	פ	צ	ק	ר	ש	שׁ	ת	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@80JW	ṣ-ṣalʕōtāʸw	מִצַּלְעֹתָיו	◌ִי	$ā^{y}$
2	1:2	16	H@J:T@H	hāytāʰ	הָיְתָה	◌ָה	$ā^{h}$
3	1:1	4	B.@R@74>	bārāʾ	בָּרָא	◌ָ	ā
4	1:1	8	C.@MA73Jim	š-šāmayim	הַשָּׁמַיִם	◌ַ	a
5	1:25	488	>:AD@M	ʔᵃdāmāʰ	הָאֲדָמָה	◌ַ	-^{a}
6	1:2	26	R74W.XA	rûaḥ	וְרוּחַ	◌ַ	$_{a}$-
6	1:6	84	R@QI73J<A	rāqîaʕ	רָקִיעַ	◌ַ	$_{a}$-
7	1:2	23	P.:N;74J	pᵊnê	פְּנֵי	◌ֵי	ê
8	1:10	70	MIQ:W;71H	miqwēʰ	וּלְמִקְוֵה	◌ֵה	$ē^{h}$
9	1:1	3	R;>CI73JT	rēʾšît	בְּרֵאשִׁית	◌ֵ	ē
	3:14	1456	XAJ.E75Jk@00	ḥayyᵉyxā	חַיֶּיךָ	◌ֶי	e^{y}
11	1:11	192	<O70FEH	ʕōśeʰ	עֹשֶׂה	◌ֶה	e^{h}
12	1:1	12	>@75Rey	ʔāreṣ	הָאָרֶץ	◌ֶ-	e
13	1:2	5	>:ELOHI92Jm	ʔᵉlōhîm	אֱלֹהִים	◌ֱ-	-^{e}
14	1:1	3	R;>CI73J	rēʾšît	בְּרֵאשִׁית	◌ִי	î
15	1:1	8	C.@MA73Jim	š-šāmayim	הַשָּׁמַיִם	◌ִ	i
16	1:2	21	T:HO92Wm	tᵊhôm	תְּהוֹם	◌וֹ	ô
17	9:21	4421		ʔhlh	אָהֳלָה	◌ֳ-ה	$ō^{h}$
18	1:2	5	>:ELOHI92Jm	ʔᵉlōhîm	אֱלֹהִים	◌ֹ-	ō
19	1:29	622	>@K:L@75H00	ʔoxlāʰ	לְאָכְלָה	◌ָ	o
20	2:23	1138	LU75Q:@X@H&	luqºḥāh	לֻקָּחָה	◌ָ	-^{o}
21	1:2	26	R74W.XA	rûaḥ	וְרוּחַ	◌ּ	û
22	1:28	570	KIB:CU92	xivšuhā	וְכִבְשֻׁהָ	◌ֻ	u
23	1:1	2	B.:-	bᵊ	בְּרֵאשִׁית	◌ְ	^{ə}

However, in a few verb forms an accent is implied, even though a meteg to indicate secondary stress is missing, and this is the case with *hāyᵊtāʰ* discussed above, i.e. this is not a case of qamets hatuf in a closed unstressed syllabe ***hoytāʰ*). Wilson has successfully programmed the rewrite rules to handle this qamets pronounced as /ā/ in the common Ashkenazi pronunciation. We found the test cases in Table 6, to test the rewrite rules, and again this qamets is handled correctly in the present version of the transliteration. However, for some as yet unknown reason the following ultra short vowel, the shewa mobile /ᵊ/, in every instance has disappeared from the transliteration.

There are also various other challenges that need to be solved when we get the opportunity to continue the programming of the RLM-tool. From the above it is clear that we need an extra rule to handle the sequence of "vowel – waw with dagesh forte – vowel" in order to guarantee the reading *hiwwāsᵊrî* in stead of the misreading **hiûāsrî* which is impossible because waw followed by dagesh according to example (4) only can be read as the long vowel shureq (וּ transliterated as /û/) after a preceding consonant.

Table 5. Qamets as /ā/ in open stressed syllable without meteg

	Paradigm	Reference	Hebrew	April 2009	Comment
Qa PF 3fs	כָּתְבָה	Gen 4,11	פָּצְתָה	pāṣtāʰ	
3p	כָּתְבוּ	Gen 4,6	נָפְלוּ	nāflû	
NI IMPF 2fs	תִּכָּתְבִי	Ez 26,21	תִּמָּצְאִי	timmāṣʔî	2x in OT
3mp	יִכָּתְבוּ	Gen 18,29	יִמָּצְאוּן	yimmāṣʔûn	
2mp	תִּכָּתְבוּ	Lev 19,12	תִּשָּׁבְעוּ	tiššāvʕû	
INF 2fs	הִכָּתְבִי	Jer 6.8	הִוָּסְרִי	hiûāsrî	5x in OT
2mp	הִכָּתְבוּ	Gen 49,2	הִקָּבְצוּ	hiqqāvṣû	
COH 1s	אִכָּתְבָה	Gen 19,20	אִמָּלְטָה	ʔimmālṭāʰ	
1p		Isa 43,26	נִשָּׁפְטָה	niššāfṭāʰ	2x in OT

Another problem is that some Hebrew words are read differently from the consonantal text, and the pronunciation is not given in the WIVU database. The divine name יְהוָה is always read with the vowels from another word, אֲדֹנָי, *Lord*, because according to Jewish tradition the name of God could not be pronounced. Unfortunately this results in the distinctly ungrammatical transliteration *yᵊhwāʰ (Gen 2,4: 6828x), know from the old mispronunciation Jehovah. We believe, that ancient tradition supports the vocalization יְהוָה, and for this reason we will want to transliterate the name as Yahweh.

Among the other remaining problems are at present the following cases which will have to be corrected manually until the correction of the programming in the RLM-tool:

- The definite article preceding the noun incorrectly occurs as detached from the noun as a separate word, e.g. **ha šššāmayim* in stead of *haš-šāmayim* (Gen 1:1 and this problem is ubiquitous for all cases of determined nouns in the Hebrew Bible. There are similar problems for preposition + definite article
- Three out of four verbs in the Hebrew Bible occur in the so-called narrative form used for chaining of perfective verbs in stories, and it is characterized by a doubling of the initial prefix. Unfortunately this is at present displayed as an ungrammatical form with a separate word in front of the verb, e.g., *wa yyōʔmer* instead of *way-yōʔmer* (Gen 1,3).

There are no doubt other problems in the present version of our transliteration. We hope to be able to correct this in future programming soon, and meanwhile users will have to fix the errors manually. We will also appreciate feedback from users who would help call our attention to other errors than the ones mentioned above.

Conclusions

This technical report is part of the Role Lexical Module coordinated by Winther-Nielsen (2008, 2009) and programmed and designed by Wilson (2009), and in the transliteration effort we were greatly assisted by the programmer Claus Tøndering. We have also reported on our use of Bergman (2005) as our standard of reference, and we have discussed other interesting options.

This report explains the intricacies involved in automatically rewrite the Hebrew text into a readable linguistic script that can be used for syntactic and semantic analysis. Lead programmer Chris Wilson has now solved the problem of representing different alphabets by means of tables with rewrite rules, and we expect this to be helpful for minority languages wanting be involved in the translation efforts of the Lex project.

The report represents the state of art of the programming of the Role-Lexical Module. At present we do not know when we will be able to resume the project with some funding and programming assistance to Chris Wilson.

Bibliography

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

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

Wilson, Chris. 2009. Lex: a Software Project for Linguists. *Hiphil* 6 [http://hiphil. see-j.net] (2009).

Winther-Nielsen, Nicolai. 2008. A Role-Lexical Module (RLM) for Biblical Hebrew: A mapping tool forRRG and WordNet. In *Investigations of the Syntax-Semantics-Pragmatics Interface* [SLCs 105], R.D. Van Valin Jr (Ed.), 455-478. Amsterdam and Philadelphia: John Benjamins.

–. 2009. "Biblical Hebrew parsing on display." Forthcoming in *Hiphil 6* [http://hiphil.see-j.net] (2009).