

L'exégèse biblique et l'ordinateur

par

R. WONNEBERGER

Université d'Hamburg - R.F.A.

LES BESOINS DE NOTRE DISCIPLINE

Un projet de développement de la recherche dans le domaine de l'Ancien Testament devrait certainement avoir pour objectif principal la mise sur pied d'une banque de données, contenant le texte massorétique, y compris les voyelles, et des programmes standards pour chercher des mots et des combinaisons de mots⁽¹⁾. Si l'on veut un exemple de quelques questions de l'exégèse, pour lesquelles un tel projet pourrait être intéressant, on les trouvera chez M.H. Schult⁽²⁾. Mais il ne suffit pas de travailler sur la base des mots. Si l'on veut trouver des réponses aux questions posées par la Formgeschichte ou par la Redaktionsgeschichte, on a besoin d'une syntaxe générative, qui permet d'analyser complètement la structure syntaxique des phrases et en même temps décrire leur valeur stylistique⁽³⁾.

Bien qu'on dispose des techniques pour réaliser ce projet et qu'elles soient mises en oeuvre chaque jour dans d'autres contextes (comme par exemple dans les projets de la traduction automatique), un tel projet n'a pas encore vu le jour.

Quelles qu'en soient les raisons, en examinant les projets réalisés jusqu'à ce jour⁽⁴⁾, on arrivera probablement aux conclusions suivantes :

- 1) Bien qu'il y ait eu un grand nombre de projets, la recherche à l'aide de l'ordinateur ne joue pas de rôle essentiel dans l'exégèse de l'Ancien Testament.
- 2) Bien qu'il y ait eu des expériences au moins depuis 1965, les possibilités de l'ordinateur sont jusqu'à ce jour réservées aux groupes ou aux projets spécialisés en ce domaine.
- 3) Bien que les méthodes syntaxiques et sémantiques soient fournies par la linguistique, on se permet de les négliger.
- 4) Le but des projets est toujours la publication de certains résultats, jamais la publication des sources, des méthodes ou des programmes; ainsi le développement des méthodes est gravement entravé.

Au lieu de continuer avec de tels projets l'un après l'autre, j'aimerais proposer une alternative. Pourquoi mêler l'ordinateur aux questions de la recherche, puisque seulement très peu de gens se sont rendus maîtres de ces techniques ?

Si l'on partage notre idée selon laquelle l'application dynamique est plus appropriée à la nature de l'ordinateur, on aura comme objectif principal de faire profiter le plus grand nombre possible de chercheurs, des capacités de l'ordinateur. Evidemment, ce qui est le devoir commun à tous les chercheurs - quelle que soit leur recherche - c'est la composition des publications.

Dans le domaine de l'Ancien Testament, la publication des textes scientifiques comprend des difficultés supplémentaires, parce qu'il faut reproduire des textes en hébreu et en grec. Par conséquent, il serait souhaitable de faire des programmes d'éditeur accessibles à tout le monde et nous nous hâtons de donner la description d'un tel projet.

LA BASE DU SYSTEME TCF

Le système TCF⁽⁵⁾ a été développé en 1981 par M. Peter-Klaus Schilling de DESY⁽⁶⁾ et l'auteur⁽⁷⁾ sur la base de systèmes normaux installés au DESY à Hambourg.

Dans un grand laboratoire comme le DESY, les chercheurs et les groupes appartenant à l'institution ont beaucoup de documentation, allant de petites publications jusqu'aux thèses de doctorat ou aux publications officielles du centre. Tout d'abord, plusieurs personnes avaient entrepris de composer leurs documents à l'aide de programmes différents ou de systèmes éditeurs privés⁽⁸⁾.

Maintenant, on a loué un système commercial plus efficace pour composer toutes sortes de documents⁽⁹⁾. Les résultats du SCRIPT sont imprimés par un printer normal, sauf que celui-ci a une chaîne de minuscules en plus des majuscules. Evidemment, de tels résultats ne sont pas très satisfaisants, tout au moins, parce que le système donne l'occasion de produire une bonne qualité en écriture proportionnelle et en ensembles de caractères différents. Malheureusement, pour obtenir cette sorte d'output, il faut un laser printer, qui coûte très cher. Mais au DESY on a trouvé une solution élégante à ce problème.

Quand la composition des textes a commencé à augmenter, on avait déjà développé un grand réseau de terminaux graphiques pour permettre de voir tout de suite les résultats des calculs et des expériences⁽¹⁰⁾. On peut travailler avec les images graphiques et aussi en avoir des copies sur un plotter.

Il y a en principe deux méthodes pour la réalisation de l'image d'un caractère :

- 1) le quadrillage par des points ou des lignes parallèles;
- 2) la composition par une séquence de vecteurs. Bien que la première méthode soit connue de la télévision, elle n'est pas recommandée pour le codage des caractères, parce qu'il faut enregistrer un grand nombre de points. Si l'on représente les caractères alphabétiques par des lignes graphiques, on n'a besoin que d'un petit nombre de vecteurs qui peuvent être corrigés facilement et s'adaptent d'une manière naturelle aux changements de la mesure; en plus, ils donnent la possibilité d'organiser l'image comme une hiérarchie d'informations.

Ainsi, les images ne dépendent pas du hardware qui fait l'output. Elles sont réalisées par les programmes standard qui sont fournis pour le hardware. En cas d'un plotter, l'image composée par les vecteurs est soumise à un quadrillage, dont la qualité est déterminée par la résolution du plotter⁽¹¹⁾.

Bien que le système même ne contienne pas les signes diacritiques comme par exemple les accents français ou grecs, il donne la possibilité de les incorporer. On peut définir des lettres par une largeur de zéro (zero-width-characters), et le système les place dans les lignes d'output sans les compter selon la largeur de la ligne. C'est alors la tâche du programme suivant, CALLIGRA⁽¹²⁾ de les placer sur la même position que la dernière lettre normale précédente. L'application la plus connue, c'est le

français avec ses accents différents. Au lieu d'avoir un caractère spécial pour chaque combinaison(13), ces images sont composées dynamiquement.

En plus, le programme CALLIGRA donne la possibilité d'insérer des images qui ont été produites par d'autres systèmes(14).

LES CARACTERES HEBRAIQUES

La méthode peut être appliquée de la même façon pour les voyelles hébraïques et aussi pour la cantillation hébraïque.

A l'aide du programme EDITF(15), on peut dessiner les caractères hébraïques comme une succession de lignes dans une matrice de points. Pour aider et faciliter le développement des caractères, le programme procure les possibilités suivantes :

- 1) introduire un nouveau vecteur ou en détruire un autre (add, delete);
- 2) plier un vecteur en deux (split);
- 3) déplacer un point angulaire; les deux vecteurs s'adaptent automatiquement (move);
- 4) déplacer toute l'image (move);
- 5) zoomer toute l'image (zooming);
- 6) mettre plusieurs caractères dans une seule image (overlay).

La possibilité de faire un overlay de plusieurs caractères dans une seule image est avantageuse en deux points pour l'hébreu.

D'abord on peut, à partir d'une seule origine, former les lettres qui ont des parties communes. Par exemple, \aleph et \beth peuvent être construits avec \aleph .

Par le overlay, on peut aussi voir si les voyelles n'occupent pas le même espace que les consonnes, question très importante dans le cas de "lamed" (\aleph), parce qu'il faut laisser de l'espace pour le "cholem" (\aleph), le petit point sur la lettre (\aleph). Comme exemple, nous donnons le dessin de la lettre "lamed" (\aleph) dans l'illustration 1 et une table de tous les caractères dans l'illustration 2.

La qualité des caractères dépend de la manière de les dessiner. Par exemple, le \aleph , qui vient de l'ensemble des caractères HERSHEY(16), donne une impression différente des autres caractères, qui ne sont que le résultat d'un premier essai(17). Par des expériences, une meilleure impression peut être obtenue, comme le montre le \aleph .

L'input est fait par une table de traduction, méthode bien connue dans ce domaine; le changement de traduction est effectué par une variable du système SCRIPT(18).

LE RENVÈREMENT DES PARTIES HEBRAÏQUES

Le renversement des parties hébraïques est le point le plus difficile, parce qu'il ne figure pas dans les possibilités d'un éditeur comme SCRIPT. Le renversement ne peut être fait qu'au moment où la ligne est prête pour l'*output*. A ce moment-là, il n'est plus possible dans un tel système, d'influer sur la ligne. Par conséquent, il faut faire le renversement plus tard dans un autre programme. Comme on a besoin d'un tel programme, en tout cas pour simuler le laser printer, on a trouvé une solution simple à ce problème.

Le renversement n'est fait que dans le programme CALLIGRA, qui prépare les données pour le plotter. Ce programme obtient, pour chaque ligne, plusieurs records qui dépendent du nombre d'ensembles différents de caractères. Toutes les parties hébraïques d'une ligne sont alors contenues dans le même record. Comme les blancs normaux (hex 40) ont été remplacés par des blancs spéciaux, les blancs normaux qui restent encore indiquent la séparation de deux groupes de mots dans ce record.

En renversant les caractères d'un seul groupe de mots, il faut laisser les voyelles et la cantilation à leur propre place, afin qu'ils soient dessinés sur le caractère qui les précède dans l'input, car l'input est en séquence normale (de gauche à droite) et le dessin de la ligne se fait aussi dans cette direction. Grâce aux zero-width-characters, la solution est aussi simple qu'efficace : on trouve la largeur de chaque lettre dans une table hexadécimale; si elle est égale à zéro, on laisse la lettre devant la dernière lettre normale (dont la largeur est plus grande que zéro).

Cette solution a l'avantage supplémentaire de ne pas dépendre du programme. Pour introduire de nouvelles lettres ou signes diacritiques, il ne faut que changer cette table.

LES POINTS CARACTERISTIQUES DU SYSTEME

Ainsi, au lieu de construire un système spécial pour l'hébreu, on a pris les systèmes généraux et essayé de les adapter aux problèmes de l'écriture quadratique, qui vaut pour l'hébreu aussi bien que pour l'araméen et qui peut être utilisée comme transcription pour le syrien. Le système complet se compose ainsi :

- 1) un éditeur normal pour faire l'input;
- 2) un éditeur spécial pour faire l'output;
- 3) le programme EDITF et quelques autres programmes pour développer et manipuler les signes;
- 4) le programme CALLIGRA qui traduit les résultats de l'éditeur dans les dates graphiques, y compris le renversement des parties hébraïques du texte;

5) le terminal graphique pour contrôler les résultats;

6) le plotter pour faire des copies sur papier.

Evidemment, cette méthode a de grands avantages. On utilise le même software que la plupart des utilisateurs, donc il n'est pas nécessaire de s'occuper de cet aspect du problème. De même, le hardware est standard, et aussi pas trop coûteux.

Les résultats sont très satisfaisants et pourraient encore être améliorés si l'on avait plus l'expérience des méthodes de dessin des caractères.

Un avantage essentiel de ce modèle est sa transportabilité. L'output du programme CALLIGRA peut être organisé d'après les règles du AGF PLOTFILE(20). Par cette méthode, on peut produire l'output chez soi, et après transmettre les résultats à un autre centre d'ordinateur qui dispose d'une machine pour produire des microfiches ou microfilms -machine assez chère-, ce qui donne une résolution plus grande que le plotter.

L'APPLICATION DU SYSTEME

En travaillant avec ce système, on peut facilement mêler les citations grecques ou hébraïques au texte scientifique de l'exégèse (voir l'illustration 3), mais aussi les souligner (voir l'illustration 4) ou les mettre en forme de table (voir l'illustration 5 et l'illustration 6)(21).

BIBLIOGRAPHIE

Je remercie M. Jean-Marie Argelès, Paris, d'avoir corrigé le français de quelques parties de ce manuscrit.

Enderle/Giese/Krause/Meinzer Plotfile : Enderle G./Giese I./Krause M./Meinzer H.P.. The AGF Plotfile - toward a Standardization for Storage and Transportation of Graphics Information. Gesellschaft für Mathematik und Datenverarbeitung mbH, Bonn 1979.

Hardmeier Verwendung : Hardmeier Christof. Zur Verwendung von elektronischen Datenverarbeitungsanlagen in der alttestamentlichen Wissenschaft - Neue Möglichkeiten der Forschung am Alten Testament (Ein Projekt am Alttestamentlichen Seminar in Heidelberg). ZAW 82 (1970) 175-185.

IBM DCF : International Business Machines Corporation. Document Composition Facility : User's Guide - Program Product - Program Number 5748-XX9. IBM Publication SH20-9161-1. 2nd edition : April 1980.

Poswick Comparaison : Poswick R.-F. Au-delà du mot : la comparaison multilingue comme source d'étude de la structure des langues. Actes du Congrès International "Informatique et Sciences humaines, Liège 1981.

Schilling Graphics : Schilling Peter Klaus. Computer Graphics at DESY - Talk presented at the SEAS AM 79, Hamburg. Internal Report DV-79/03. DESY Hamburg, November 1979.

Schilling Implementing : Schilling Peter Klaus. Implementing and Using Hershey Fonts. Internal Report R2-81/3. DESY Hamburg, rev. vers March 20th, 1981.

Vandeweghe/van de Velde : Vandeweghe Willy / Van de Velde Marc. Bedeutung, Sprechakte und Texte - Akten des 13. Linguistischen Kolloquiums, Gent 1978 Band 2. LA 77. 1979. Wonneberger Stylistics.

Wonneberger Normaltext : Wonneberger R. Normaltext und Normalsynopse - Neue Wege bei der Darstellung alttestamentlicher Texte. (zur Publikation eingereicht).

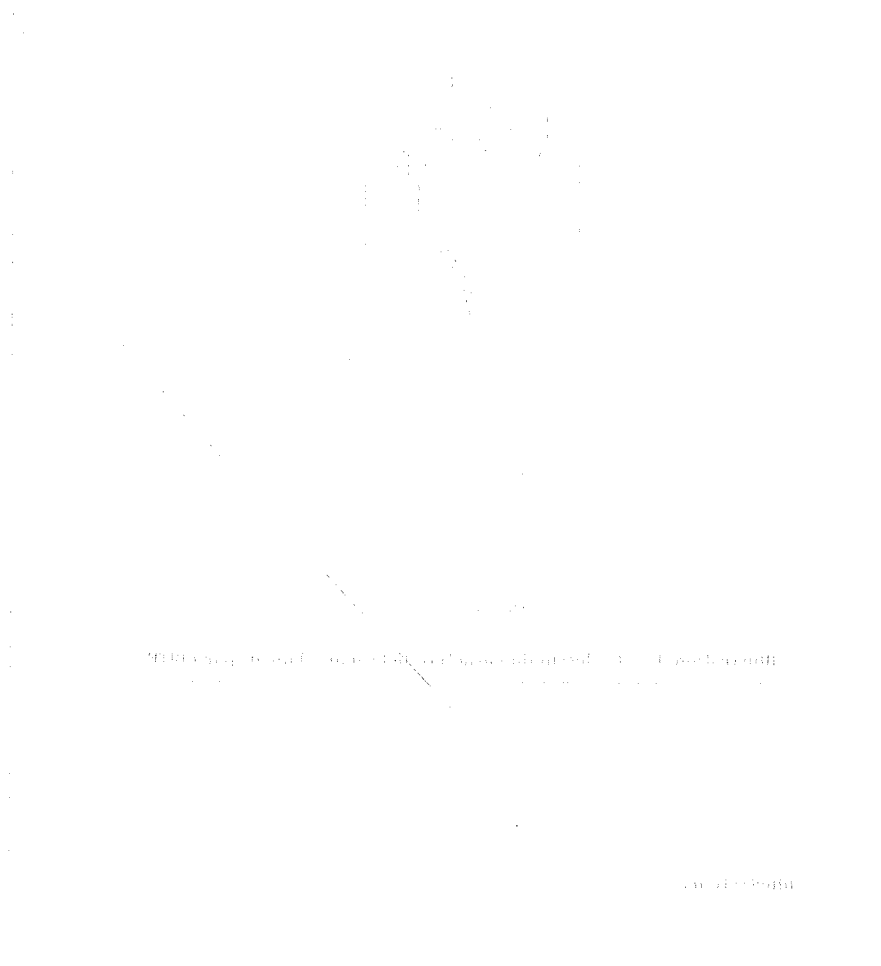
Wonneberger Scriptor : Wonneberger R. Scriptor - Ein Text-Editions-System. Skriptum des Rechenzentrums der Universität. Heidelberg 1976.

Wonneberger Stylistics : Wonneberger Reinhard. Generative Stylistics - An algorithmic approach to stylistic and source data retrieval problems based on generative syntax. Vandeweghe/Velde Bedeutung 389-399.

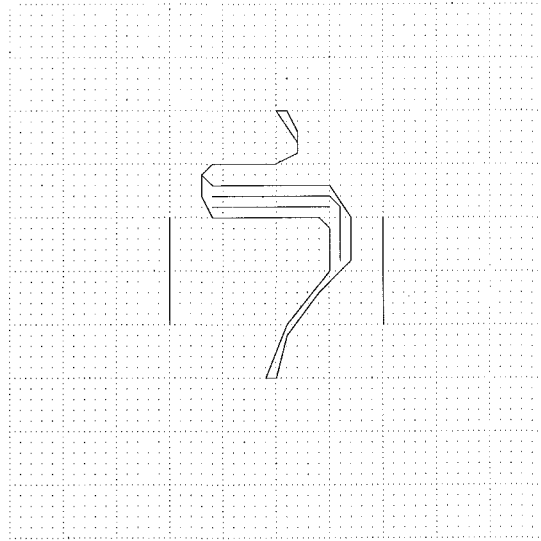
NOTES

- (1) Voir *Poswick Comparaison* pour la description d'un tel projet.
- (2) Dans *Hardmeier Verwendung* 179f.
- (3) Pour le grec, une telle grammaire se trouve dans *Wonneberger Syntax*; il n'y a rien de tel pour l'hébreu, à ma connaissance. Dans *Wonneberger Stylistics*, on trouvera comment se servir d'une telle grammaire pour décrire le style et pour poser des questions à une banque de données.
- (4) Comme je l'ai fait en préparant un compte-rendu de la recherche réalisée à l'aide de l'ordinateur, j'espère publier celle-ci dans quelque temps.
- (5) Text Composition Facility.
- (6) Deutsches Elektronen SYNchrotron, Hambourg.
- (7) L'auteur remercie le directeur de DESY de lui avoir donné la possibilité de profiter de son centre d'ordinateur et M.P.-K. Schilling pour sa collaboration amicale.
- (8) Par exemple, celui de l'auteur, voir *Wonneberger Scriptor*.
- (9) Le DCF (*document composition facility*), mieux connu sous le nom SCRIPT; voir *IBM DCF*.
- (10) Un survey sur ce système chez *Schilling Graphics*.
- (11) Avec le modèle installé actuellement, on obtient une solution 200 points/inch = 7,87 points/mm = 0,127 mm/point.
- (12) Voir *Schilling Fonts*.
- (13) Accent aigu : é à ù; circonflexe ê â û ô î; tréma : ë ü î; cédille : ç.
- (14) De cette manière, nous avons inséré les illustrations qui concernent le développement des caractères hébraïques, qui ont été produites par le programme EDITF; voir l'illustration 1 et l'illustration 2.
- (15) Voir *Schilling Implementing*.
- (16) Voir *Schilling Implementing* pour les détails.
- (17) Ces consonnes ont été dessinées par M. Kristian Schilling, les autres signes et quelques corrections ont été faits par M. Peter-Klaus Schilling, à l'aide d'une version du programme EDITF beaucoup moins confortable que la présente.
- (18) On pourrait aussi prendre des GML-TAGS, mais les variables donnent la possibilité supplémentaire de les regarder une fois comme des variables, et l'autre fois comme un texte normal, ce qui est avantageux pour des raisons internes au système.
- (19) Voir *Schilling Graphics* 1.

- (20) AGF est l'abréviation pour "Arbeitsgemeinschaft der Grossforschungseinrichtungen Deutschlands", dont le groupe "Koordinationsausschuss graphische Datenverarbeitung" a développé une structure de données pour transporter des données graphiques. "The AGF plotfile is a sequential file for storage and transportation of graphics information. It contains records describing graphics primitives (points, vectors, polylines, texts) and their attributes".
(Enderle/Giese/Krause/Meinzer Plotfile 1).
- (21) Ces exemples sont pris de *Wonneberger Normaltext*.



ILLUSTRATIONS



CH1693

Illustration 1. Le dessin du caractère hébraïque «amed» par EDITF

Illustrations

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00																(
10)
20																
30																
40																
50																
60																
70		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
80		p	q	r	s	t	u	v	w	x	y	z				
90																
AA																
BA	0	1	2	3	4	5	6	7	8	9						
CA																
DA																
EA																
FA	0	1	2	3	4	5	6	7	8	9						

FONT NR = 16

Illustration 2. Les caractères hébraïques

Illustrations

La phrase «הַיְשִׁלוּ הַלְלִים קָהָר הַנִּלְכָּעוּ»¹ par l'article se distingue de la phrase «הַיְשִׁלוּ»¹ avant le nom «נִלְכָּעוּ».

Illustration 3. Exemple d'un texte exégétique

Ri 6,8bβ-9a	#	1.S 10,18aβγb
כה-אמר יהוה אלהי ישראל	1	כה-אמר יהוה אלהי ישראל
אנכי העליתי אתכם ממצרים	2	אנכי העליתי את ישראל ממצרים
ואציא אתכם מכי"ת עבדים	3	
ואציל אתכם מיד מצרים	4	ואציל אתכם מיד מצרים
ומיד כל-לחצים אתכם	5	ומיד כל-דממלכות הלחצים אתכם

Illustration 4. Synopse Ri 6,8f || 1.S 10,18

1.S 31	R	1.Ch 10
וַיִּפֹּל עָלָיו	1	וַיִּפֹּל עָלָיו
וַיְשִׁלוּ הַלְלִים קָהָר הַנִּלְכָּעוּ	2	וַיְשִׁלוּ הַלְלִים קָהָר וְנִלְכָּעוּ
וַיִּשְׁמוּ אֶת-בְּלִי בֵּית עֲשֶׂתְרוֹת	3	וַיִּשְׁמוּ אֶת-בְּלִי בֵּית אֱלֹהֵיהֶם
וַיִּקְרְאוּ אֶת-עֲצֻמֹתָהֶם	4	וַיִּקְרְאוּ אֶת-עֲצֻמֹתָהֶם...
וַיִּקְרְאוּ תַחַת-הָאֵשׁל כִּי-גִבְשָׁה	"	... תַחַת הָאֵשׁל כִּי-גִבְשָׁה

Illustration 5. Ähnlichkeitsgrade (R) einer Synopse (Beispiele aus Sauls Tod)

G (Septuaginta)	#	Th (Theodotion)
ὅθ ταῦτα σέβεσθε,	1	ἴδετε τὰ σεβάσματα ὁμῶν
βασιλεῦ	2	

Illustration 6. Beispiel aus Dan 14,27

Illustrations